Hangar

Open Source Hackathon Platform

Tristen Even, tge160130@utdallas.edu; Razvan Preotu, rnp170130@utdallas.edu;
Tatum Hutton, tah180000@utdallas.edu; Michael Nguyen, mkn170030@utdallas.edu

CS 4485 / Spring 2021
Department of Computer Science
Erik Jonsson School of Engineering & Computer Science
The University of Texas at Dallas
Richardson, TX 75080, USA

Abstract

Hangar is an open-source hackathon management platform developed by American Airlines. Throughout the semester the UTD capstone team collaborated with a group of American Airlines developers to help them further develop Hangar. The team adopted an agile development strategy, prioritizing and categorizing issues based on a two-week implementation schedule. Assigned issue topics ranged from frontend development, backend development, discord/slack integration, testing, and documentation. The team's UT Design project completed in a six-week timeline aims to enhance the user experience of setting up Hangar. The project consists of the creation of a setup page, which is meant to be used by the user when configuring the app for the first time, and a 'Hangar for Dummies' page that gives user detailed navigation through app setup.

Keywords: UI/UX, Agile Development, Documentation

Architecture

Frontend
- React.js
- Docsify

Backend
- Node.js
- Express

Database
- PostgreSQL

Testing
- Jest

Impact

Hangar is designed to be easy to use and enjoyable for everyone. Our project contributes to this design principle by providing a simpler setup process to the user.

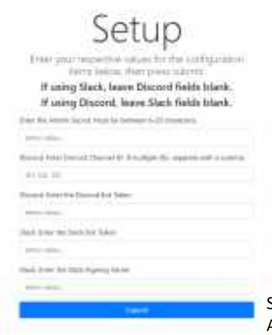
Before

- App didn't start if any of the required configuration entities were missing. User had to directly set these values in the database
- Setup instructions consisted of a simple markdown with few visuals

After

- App can start without any configuring entities and there is a setup page to set these values
- "Hangar for Dummies" page acts as a full user setup guide with visuals

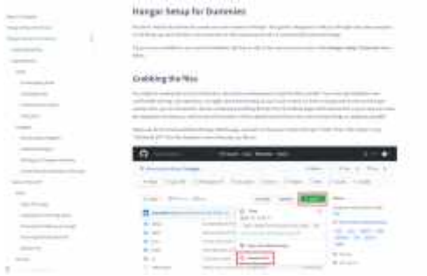
Results



Setup

If using Slack, leave Discord fields blank.
If using Discord, leave Slack fields blank.

Setup Page on App Startup



Hangar Setup for Dummies

Grabbing the files

Full User Guide with Visuals

Performance

- All undertaken tasks completed.
  - Initial estimate of 20 tasks was cut significantly in favor of tasks with larger scope.
- All functional expectations met.
- Content of each task considered valuable by team and company.

Summary

- Removed the need for certain external tools to begin using app
- Overhauled and streamlined app setup
- Created a full installation guide for lay users that requires zero technical knowledge
- Helped overhaul testing suite to minimize future tech debt

Access My Research

Group 2

Jibi Bijumon, jpb170330@utdallas.edu; Riya Patel, rgp170030@utdallas.edu; Jestina George, jxg172430@utdallas.edu;
Binali Patel, bxp170030@utdallas.edu; Milan Patel, mkp170430@utdallas.edu; Anthony Guardia, afg170000@utdallas.edu;
Muhammad Khatri, mkk170000@utdallas.edu; Carlos Guerrero, cjg18000@utdallas.edu

CS 4485 / Spring 2021
Department of Computer Science
Erik Jonsson School of Engineering & Computer Science
The University of Texas at Dallas
Richardson, TX 75080, USA

Abstract

To increase the accessibility of research to all people, Access My Research (AMR) is a nonprofit alternative to ResearchGate and Academia.edu that removes the barriers such as expensive and highly fragmented academic access while promoting collaboration, and levels the playing field for all. AMR is working to develop a web app that makes it easier for individuals and organizations to access and share research. The objective for our team was to continue development on the full product and make advancements so it could possibly be released to production.

Keywords: Research, Web App, Social Platform, UI, Infrastructure

Architecture

Frontend

- Vue JS

Backend

- Amazon Web Services (AWS)
  - Elasticsearch
  - Lambda
  - S3
- Google Domains
- Python
- Node.JS

Impact

Our focus for AccessMyResearch was working on both frontend and backend of the existing website.

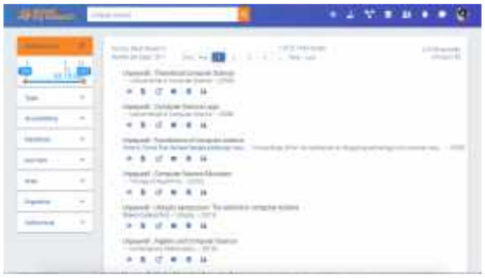
Before

- Issues with filter pane
- No actual research papers
- User unable to add their own research

After

- Connected a google domain to the existing Github
- Created a new filter pane that pulled data from a database instead of article information stored internally
- Connected website to AWS so a user could upload files
- Extracted articles from databases and added to Elasticsearch

Results



Website with new filter pane

Kibana User Upload

Sign-in Module

Performance

- 91% story completion in concluded sprints
- Significant advancements in the product
- Content is dynamically populated

Summary

- Provided a search engine hub that aggregates other research databases periodically while offering researchers advanced sharing, collaboration, and showcasing tools for maximum effortless visibility.
- Increased the ease-of-access to research for disadvantaged users, organizations, and communities.

AccessMyResearch: Feature Integration on a Web Application

Sanjana Sankaran, Mythri Challa, Albert Jean, Yousuf Jazzar, Shariff Habboush, Khalid Dakak
sxs179530@utdallas.edu, mxc170030@utdallas.edu, asj170000@utdallas.edu, yxj170003@utdallas.edu,
sih170000@utdallas.edu, knd170000@utdallas.edu

CS 4485 / Spring 2021
Department of Computer Science
Erik Jonsson School of Engineering & Computer Science
The University of Texas at Dallas
Richardson, TX 75080, USA

Abstract

AccessMyResearch is a non-profit social platform with an advanced search engine and database connecting academia to industry and the public. Its aim is to allow for sharing research among networks in the otherwise expensive, fragmented bubble that is academic research.

Throughout the semester, our ECS Senior Design team has been assisting AccessMyResearch (AMR) to develop their web application portal and build the non-profit platform.

Keywords: UI/UX, Searching and Indexing, Messaging

Architecture

Frontend

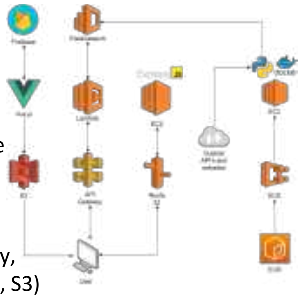
- Vue.js
- Bootstrap Libraries

Database and Search Engine

- Firebase
- Elasticsearch & Kibana

Backend

- AWS (Cognito, Gateway, EC2, ECR, ECS, Lambda, S3)
- Express.js



Impact

AccessMyResearch aims to make research available to everyone at significantly lower personal costs. Our work this semester was the next important step to completing this objective and preparing the website.

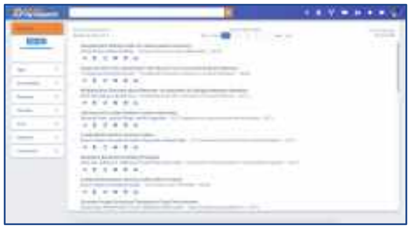
Before

- Incomplete UI design of web pages
- Limited functionality of important components of the website
- Basic and rigid indexing routines were used and run locally

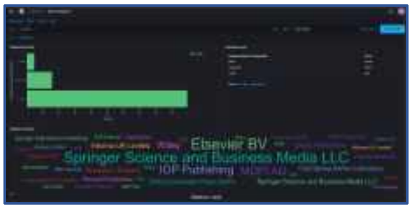
After

- Completed UI design of multiple pages and components
- Working messaging system with database integration
- Functional searching of research articles and filtering
- Upgraded indexing routines and deployed them to AWS

Results



Website



Elasticsearch Kibana Dashboard

Performance Metric

- 23 user stories over 6 sprints (in progress & shipped)
- Average complexity of user stories was 13/21 points
- Completed assigned user stories with functional expectations met (completed functions do not have bugs or interruptions)

Summary

- Created a functional backend to support the search engine features and databases.
- Improved usability and added functionality to many of the features on the site including messaging, user profile, searching, and downloading
- Beautified front end experience by applying a color and organization theme across all pages on the site

Asterisk Solutions

Live Site Registration

Akhil Indur (axi160330@utdallas.edu), Brice May (Brice.May@utdallas.edu), Sanghun Mok (Sanghun.Mok@utdallas.edu)

CS 4485 / Spring 2021
Department of Computer Science
Erik Jonsson School of Engineering & Computer Science
The University of Texas at Dallas
Richardson, TX 75080, USA

Abstract

The Live Site is an event hosting platform for virtual events. Initially the website allowed events to display custom pages and content including live streams and speakers. The teams' goal was to expand the website to allow a way to identify attendees, including a QR based system for registration to generate attendee QR codes. Another goal was to update the current live site to verify the QR code and allow the attendees to make a mini profile. The teams' solution to completing these goals was to use React + Redux, Node.js, and other computer science principles to successfully complete the Live Site Registration Project.

Architecture

Frontend/Backend and Database
- NodeJS
- MongoDB
- mailgun

Keywords
- Web Development
- Full Stack Development
- Frontend/Backend Development

Other Applications
- Gitlab
- Docker
- Visual Studio Code

Impact

Asterisk Solutions can implement our solution for any organization/company that wants to host a virtual live event on their website. The organizers can generate a specific event they want to host and edit the page settings to their preferences. Before publishing the live site, the event organizers can generate attendee QR codes by using their emails.

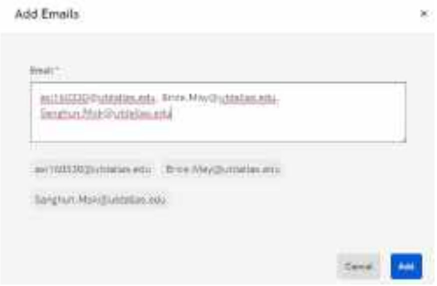
Performance

The overall performance was efficient and met every goal given. The functional expectations were met with no bugs or interruptions for the Live Site Registration. The UI was simple and easy to use and navigate as well.

The weekly tasks and progress measurements were being kept up with the team on a weekly basis for about 80% - 90%.

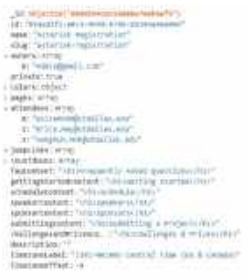
Goals were met and the company sponsor mentor was pleased with the work performed this semester.

Results



axi160330@utdallas.edu, Brice.May@utdallas.edu, Sanghun.Mok@utdallas.edu

axi160330@utdallas.edu
Brice.May@utdallas.edu
Sanghun.Mok@utdallas.edu



```
_id: ObjectId("[illegible]")
id: "[illegible]"
name: "Asterisk Registration"
slug: "asterisk-registration"
owners: Array
   0: "[illegible]@gmail.com"
private: true
colors: Object
pages: Array
attendees: Array
   0: "axi160330@utdallas.edu"
   1: "Brice.May@utdallas.edu"
   2: "Sanghun.Mok@utdallas.edu"
jumplinks: Array
[illegible]: Array
faqContent: "<h1>Frequently Asked Questions</h1>"
gettingStartedContent: "<h1>Getting Started</h1>"
scheduleContent: "<h1>Schedule</h1>"
speakerContent: "<h1>Speakers</h1>"
sponsorContent: "<h1>Sponsors</h1>"
submittingContent: "<h1>Submitting a Project</h1>"
challengesContent: "<h1>Challenges & Prizes</h1>"
description: ""
timezoneLabel: "(UTC-06:00) Central Time (US & Canada)"
timezoneOffset: -6
```

The "Add Emails" is an option on the Live Site Registration for where the event host admins can add emails of the attendees.

Those same emails can be seen in the backend database using MongoDB as objects under the array of attendees.

Summary

- Expanded the Asterisk Solutions Live Site Registration website to allow event hosts to identify attendees, including a QR based system for event registrations.
- Efficiently used the teams' skillset to use the main given tools: Node.js and React + Redux.
- Some of the skills learned throughout the project were: Full Stack Development, Agile/Scrum Process, and an overall understanding of Frontend/Backend Architecture and Database Architecture

Asterisk

Asterisk Events

Chase Dannen, cmd170230@utdallas.edu; **Faris Nasoetion**, frn170130@utdallas.edu;
Huy Nguyen, htn19000@utdallas.edu, **Justin Ok**, jjo170030@utdallas.edu;
Veerendranath Korrapati, vck170030@utdallas.edu;

CS 4485 / Spring 2021
Department of Computer Science
Erik Jonsson School of Engineering & Computer Science
The University of Texas at Dallas
Richardson, TX 75080, USA

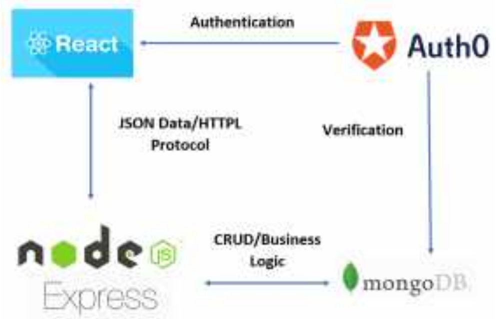
Asterisk

Abstract

This project aims to develop Asterisk Events which is an event platform to make event discovery and event management quick and easy for event organizers and attendees. This web application utilizes the MERN stack (MongoDB, Express.js, React, and Node.js) to help organizers create and manage their events. Asterisk Events also connects organizers to other tools and services offered by Asterisk and other event technology teams.

Keywords: Event platform, Web technology, Event Managing Software, Event Virtualization, Hybrid events, MERN

Architecture



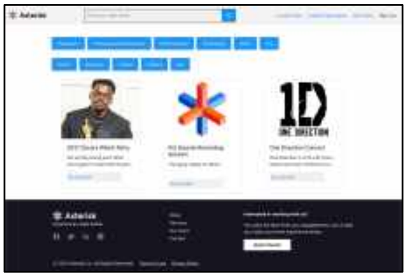
Impact

- Increase in Asterisk tool usage by event organizers
- Centralization of events in the hackathon and technology space
- Increase in speed and ease of use in event management
- Connect with participants in educational opportunities, networking events, and other types of gatherings.

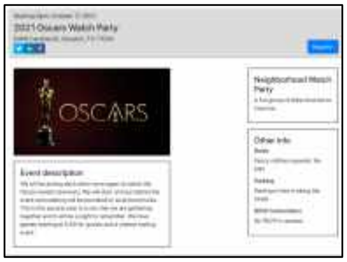
Performance

- The Asterisk Events web application meets industry performance standards for event management technology
- Meets user authentication and authorization security standards
- Meets latency and response time standards of web applications

Results



Discovery Page



2021 Oscars Watch Party

Event description

Neighborhood Watch Party

Other Info

Event Page

Event Dashboard

Edit Basic Details

Event Dashboard

Summary

The team was able to complete the development of the Asterisk Events web application which greatly improves the process of event management and discovery. Asterisk Events not only helps connect event organizers to tools that make event management easier but also helps connect attendees to the events they love.

Asterisk Solutions

Event Fund Project

Jared Stull, jbs160330@utdallas.edu; **Juan Perez**, jjp170130@utdallas.edu; **Rafael Raymos**, rjr150030@utdallas.edu;
Eunjun Kim, exk170230@utdallas.edu; **Mahtab Hossain**, mmh150030@utdallas.edu

CS 4485 / Spring 2021
Department of Computer Science
Erik Jonsson School of Engineering & Computer Science
The University of Texas at Dallas
Richardson, TX 75080, USA

Abstract: The Event Fund site is a CRM to connect event organizers and sponsors who are dissatisfied with the lack of organization, professionalism, and ability to maintain relationships during and after events. The site will bring companies and sponsors together by allowing each to view information about the other and easily establish communication. Our solution will make it easier for event hosts to find, maintain communication, and share information with potential sponsors.

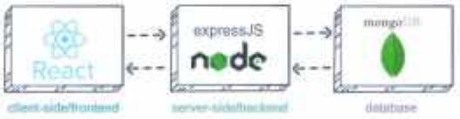
Keywords: Customer Relationship Management, Web Application, Communication

Architecture:

Event Fund uses the MERN stack design.

MERN is a full-stack, following the traditional 3-tier architectural pattern,
including the front-end display tier (React.js), application tier
(Express.js and Node.js), and database tier (MongoDB).

MongoDB - document database
Express(.js) - Node.js web framework
React(.js) - a client-side JavaScript framework
Node(.js) - the premier JavaScript web server



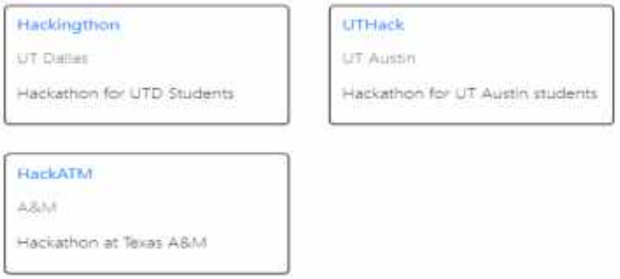
Impact:

- Allows event hosts to quickly set up event and sponsor information.
- Make the process of sponsoring and acquiring sponsors easier for organizers.
- Simplifies communication between hosts and sponsors.

Performance:

- Total Project Cost: $0
- 90% project requirements completed and functional
  - Website will save events and relevant information
  - Foundation for sponsor leads and intake forms established

Results:

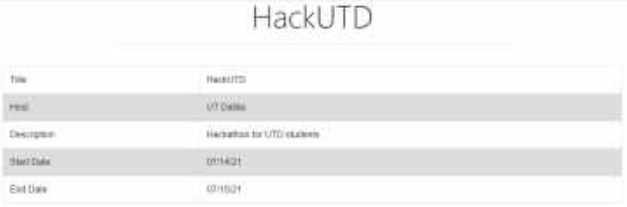


Hackingthon
UT Dallas
Hackathon for UTD Students

UTHack
UT Austin
Hackathon for UT Austin students

HackATM
A&M
Hackathon at Texas A&M

Event List



HackUTD

Title	HackUTD
Host	UT Dallas
Description	Hackathon for UTD students
Start Date	07/14/21
End Date	07/15/21

Display Event Info

Summary:

- The Event Fund website displays a list of public events for sponsors to browse.
- Potential sponsors can choose an event from the list to view more detailed information and see the specifics regarding prices and benefits to supporting that event.
- Once a sponsor has decided to support an event, the host can set up email reminders for the various tiers. Future Slack integration will also help to facilitate communication between both sides.

Asterisk

Live Site Interactive

Christos Kokkinomagoulos, cxk171730@utdallas.edu; Danny Phung, dmp190003@utdallas.edu; Zamaan Bawa, zsb160030@utdallas.edu; Dylan Hua, dch170030@utdallas.edu

CS 4485 / Spring 2021
Department of Computer Science
Erik Jonsson School of Engineering & Computer Science
The University of Texas at Dallas
Richardson, TX 75080, USA

Abstract

The Asterisk: Live Site Interactive project offers a real-time method of interaction for hackathon participants who are unable to attend hackathon events in person. To accommodate online participation, event organizers may at times, provide live video streams of the activities taking place. Our project aims to bridge the interaction gap between in-person participants and online participants by adding a *live chat* solution to Asterisk's Live Site platform by using tools such as *ReactJS.*

Key Words: Live Chat, PubNub, ReactJS

Architecture

- *ReactJS*
- Node.js
- Tailwind CSS
- PubNub chat API

Impact

The *live chat* interface makes it possible for participants, spectators, and event organizers to interact with each other to enhance morale, offer feedback, and provide support.

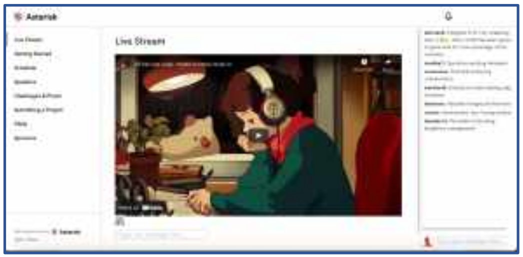
Performance

- Weekly task completion rate: 92%
- Company mentor feedback: goals were met and pleased with work performed this semester
- Functional expectations were met (completed functions do not have bugs or interruptions)

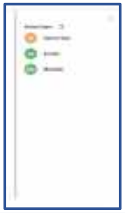
Summary

- Integrated *PubNub* chat API
- Added active users panel to view the current users in the chat
- Added private messaging feature for users to have 1 on 1 chats

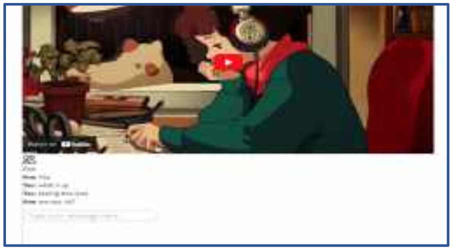
Results



Live chat



Active users panel



Private DM

Cobalt Health Status Dashboard

Agatha Lam, ajl170230@utdallas.edu; Daniel Wang, dxw171030@utdallas.edu; Michael Xu, mxx171430@utdallas.edu; Samuel Ma, sjm170130@utdallas.edu

CS 4485 / Spring 2021
Department of Computer Science
Erik Jonsson School of Engineering & Computer Science
The University of Texas at Dallas
Richardson, TX 75080, USA

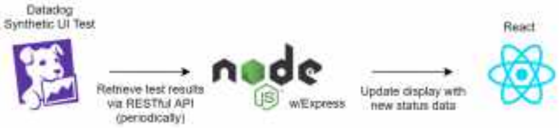
Abstract

The BAL Cobalt Health Status Dashboard is an external facing dashboard web application that displays the results of tests that check the health status of various components of Cobalt, the service that BAL Global's clients use. The necessity for this project comes from the fact that clients currently have no way of knowing if a Cobalt feature is not working besides manually logging in and coming across the outage directly. This dashboard would allow clients to gain an overall view of the system status at a quick glance and make decisions accordingly.

Keywords: Dashboard, Status, Javascript

Architecture

- JavaScript stack
- Web app using a React frontend with Express to call the Datadog API.
- Node.js with Express backend
- Datadog used to conduct synthetic UI tests on various Cobalt services


Datadog
Synthetic UI Test
Retrieve test results via RESTful API (periodically)
node
JS
w/Express
Update display with new status data
React

Impact

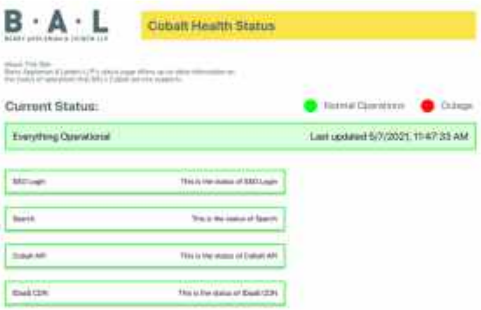
BAL Global customers now have a simple way of checking the health of various Cobalt components at a glance. This reduces the time and potential frustrations that customers may face as well as service calls that BAL may have to answer in case of an outage.

Performance

Weekly Task Completion Rate: 95%
Company Mentor Feedback: Goals and Expectations were met accordingly.

Results

Online Health Status Dashboard:



Cobalt Health Status

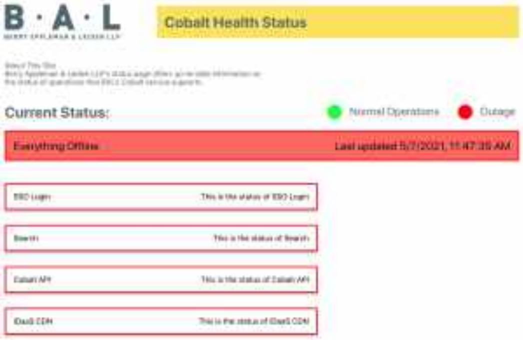
About This Site
Berry Appleman & Leiden LLP's status page offers up-to-date information on the status of operations that BAL's Cobalt service supports.

Current Status: ● Normal Operations ● Outage

Everything Operational Last updated 5/7/2021, 11:47:33 AM

SSO Login This is the status of SSO Login
Search This is the status of Search
Cobalt API This is the status of Cobalt API
IDaaS CDN This is the status of IDaaS CDN

Offline Health Status Dashboard:



Cobalt Health Status

About This Site
Berry Appleman & Leiden LLP's status page offers up-to-date information on the status of operations that BAL's Cobalt service supports.

Current Status: ● Normal Operations ● Outage

Everything Offline Last updated 5/7/2021, 11:47:35 AM

SSO Login This is the status of SSO Login
Search This is the status of Search
Cobalt API This is the status of Cobalt API
IDaaS CDN This is the status of IDaaS CDN

Summary

The BAL Health Dashboard creates an easier way for customers to view the status of the Cobalt components. This project comes with immediate usability for any clients that previously found checking the status of Cobalt to be cumbersome. With this project, we were able to create a health dashboard to meet the needs of BAL Global. This would effectively solve the issues that BAL has been encountering due to their lack of such a tool to communicate the statuses of their services to their clients with.

Containerize Cobalt Apps and Set Up Orchestration

Gabriel Goldstein gjg180000@utdallas.edu, Emily Lam ejl170001@utdallas.edu,
Jessie Luo jxl170005@utdallas.edu, Samuel Nitsch smn180000@utdallas.edu, Megan Tran mjt170002@utdallas.edu

CS 4485 / Spring 2021
Department of Computer Science
Erik Jonsson School of Engineering & Computer Science
The University of Texas at Dallas
Richardson, TX 75080, USA

Abstract

Berry Appleman & Leiden (BAL) seeks to containerize their Cobalt application to make software deployment and management more robust and efficient. Containerization achieves these goals by providing consistent developer environments, greater scalability, and ease of maintenance. We utilized Docker to create and manage the application container whilst maintaining the original integrity of Cobalt. This process establishes a framework for future continuous integration and development advancements.

Keywords: Containerization, Docker, CI/CD

Architecture

- Docker
- .NET Application (Cobalt)

Impact

Since containerization of this scale has not been performed before at BAL, we explored a new avenue that will help BAL determine how to plan future development of their solutions for easier containerization and subsequent deployment.

Performance

- Weekly goal completion rate: 100%
- Demo session held with B.A.L. engineers
- All deliverables were finished on time or earlier than planned
- Mentor feedback: pleased with the progress
- No changes in code functionality

Results

- An image (model for subsequent containers) and containers of the application were created
- Minimal, non-breaking changes to code and functionality
- Multiple containers were successfully run in parallel, showcasing the scalability of the app after containerization
- Future deployment capability using containers was verified by testing the app in a running container using authorization tokens

Summary

This was a highly functional proof-of-concept demonstrating that in the future, BAL can containerize their Cobalt application as well as other services for future deployment. Through our work, we have created a maintainable Docker image that allows programmers to easily build containers running Cobalt for any purpose. Many versions of this base image can be easily written to accommodate future enhancements to the code.

Our solution will benefit BAL by enabling cross-platform capability and providing increased flexibility in development and deployment.

SQL to Postgres Database Conversion

Database Migration With AWS SCT & DMS

Michael Chang, mxc171130@utdallas.edu; Isaac Chou, ihc180000@utdallas.edu;
Eden Harris, efh190000@utdallas.edu; Joseph Schinnerer, jcs170008@utdallas.edu;

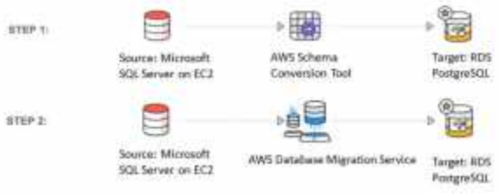
CS 4485 / Spring 2021
Department of Computer Science
Erik Jonsson School of Engineering & Computer Science
The University of Texas at Dallas
Richardson, TX 75080, USA

Abstract

AWS Aurora databases have many advantages that make them a desirable replacement over BAL's current SQL Server database solution for their Cobalt application. Some of these benefits include scalability, speed, reliability, and cost savings. This project worked to migrate the current Cobalt SQL Server databases to AWS Aurora for PostgreSQL using AWS Database Migration Service and AWS Schema Conversion Tool. Significant progress was made on the conversion of the database schemas and on the data migration. The work done on this project has the potential to positively impact BAL well into the future as they take advantage of the benefits Aurora offers.

Keywords: Amazon Web Service (AWS), Aurora PostgreSQL, SQL Server, Databases, Virtualization, Relational Database Structure

Process


STEP 1:
Source: Microsoft SQL Server on EC2
AWS Schema Conversion Tool
Target: RDS PostgreSQL
STEP 2:
Source: Microsoft SQL Server on EC2
AWS Database Migration Service
Target: RDS PostgreSQL

First, the databases were connected to the SCT with the SQL Server database being the source database and the Aurora PostgreSQL database being the target. Then, the schema conversion tool performed an automatic conversion of the schema. This conversion resulted in many issues, with most of the issues being with the database code objects. The issues unresolved by the SCT were then manually resolved. Once the schema had been fully converted, it could be applied to the target databases and a DMS task was created to migrate the corresponding data over to the new database. The DMS task was successful for the large majority of tables, with two having additional issues.

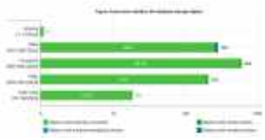
Impact

This project serves as proof of concept for:

- Modernization of the Cobalt application database solution
- Long-term cost savings and management upkeep
- Improved performance of the Cobalt application
- Reduced downtime for Cobalt due to Aurora's high availability
- Reduced administrative workload on BAL due to the fact that Aurora is fully managed by AWS Relational Database Service (RDS) which takes care of tasks such as backups, provisioning hardware, and setup
- Reduced security risks for BAL because AWS RDS is highly secure and keeps multiple backups

The positive impacts of this project will continue to benefit BAL well into the future, as long as the Aurora database remains in use.

Results



- The SCT was successful in automatically converting 99.8% of database storage objects, and 79% of the database code objects
- 161 code objects required complex actions due to 94 conversion issues with the SCT
- 214 of the 478 needed complex manual translation issues were resolved, fixed, or deprecated
- The schema definition for the tables was successfully applied to the Aurora PostgreSQL database
- Conversion of 175 functions was completed, and 11,059 out of 23,087 procedure issues were converted
- Data migration task was successful on 293/295 tables

Performance

- Conversion using SCT and DMS provides the basic structure for a fully functioning AWS RDS PostgreSQL database copy
- Virtualization and other benefits of using AWS RDS over a locally hosted server apply, including I/O throughput and security
- Schema conversion process duration was calculated to be about 4000 person-hours based on current status of completion and rates instead of a fully manual 11000 person-hours without SCT ad DMS
- Database translation/virtualization was determined to be feasible

Summary

Tools Used:
Amazon Web Service: Relational Database Service (AWS RDS)
Schema Conversion Tool (SCT)
Database Migration Service (DMS)

Total Cost: $1871.01

Result Summary:

- 100% of the table schema was converted successfully
- 47.9% of the procedure errors were resolved
- 100% of the functions were converted successfully
- 93.4% of the view errors were resolved
- 99.3% of the tables were successfully transferred

Future work will include completing the manual conversion of database code objects and extensive testing of the Aurora database functionality to ensure that it matches that of the source SQL Server database and remains secure.

Automated Collection of Event or Location Specific Social Media

Benjamin Stark, Bradley Franklin, Hope Church, Bhavi Shah, Brian Contreras
bjs170002@utdallas.edu, blf180001@utdallas.edu, ptc170030@utdallas.edu, bds180002@utdallas.edu, bac170630@utdallas.edu

CS 4485 / Spring 2021
Department of Computer Science
Erik Jonsson School of Engineering & Computer Science
The University of Texas at Dallas
Richardson, TX 75080, USA

BlackSwan Cybersecurity

Abstract

This project aims to gather social media posts related to events or locations and store them for access later to aid in the investigation of these events. This will involve a social media scraper and a searchable database. This will create a full view of any given event that was captured on social media.

Keywords : Social media, Event view, investigation

Architecture

Application

- Python Libraries
  - Tweepy
  - PyQt
  - GeoPy
  - Wget

Database

- MySQL

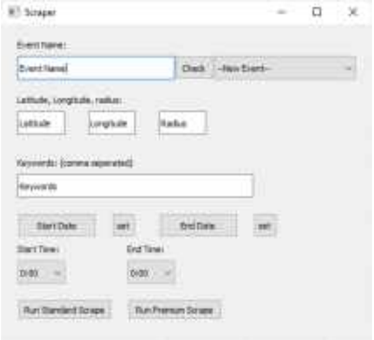
Impact

This project's applications aims toward development for ease in social media investigations, by providing investigators with tools to find online posts related to current events. Our work was to show the beginning phase of viability for a product which aims to assist investigators with the overflowing amount of data they would need to sift through.

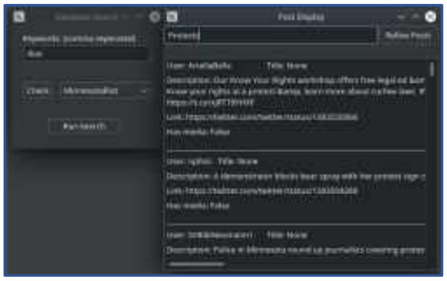
Performance

- All tasks completed by set deadlines
- Positive feedback from company sponsor
- Time from Scrape to Database Insertion: O(N)
- Time from Search to User Output: O(N*M)
  - Where N is Keywords Searched and M is Posts

Results



Scraper UI



User: [illegible] Title: None
Description: [illegible]
Link: https://twitter.com/twitter/statuses/[illegible]
Has media: False

User: [illegible] Title: None
Description: [illegible]
Link: https://twitter.com/twitter/statuses/[illegible]
Has media: False

User: [illegible] Title: None
Description: [illegible]

Search UI

Summary

- Full scrape of social media using event data input
- Location, time, and keywords as event search data
- Stored media posts and additional media into MySQL database
- Media posts searchable by event name or keywords
- Posts filterable by keywords

Developing Chatbots for Hotel Websites

Natalie Kendall
nmk160430@utdallas.edu

Rijad Hadzic
rxh140630@utdallas.edu

Joel Kurian
jtk130230@utdallas.edu

Pablo Magana Carmago
pxm180130@utdallas.edu

Christopher Marsden
cmm160330@utdallas.edu

CS 4485 / Spring 2021
Department of Computer Science
Erik Jonsson School of Engineering & Computer Science
The University of Texas at Dallas
Richardson, TX 75080, USA

Cyberweb Hotels, LLC

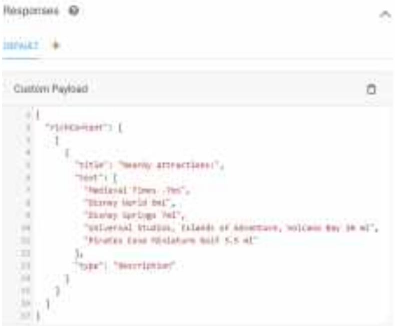
Abstract

As a group, we were tasked to complete twenty-five functional chatbots for twenty-five distinct hotels. The chatbots were all created on Google's DialogFlow platform to maintain consistency. The group quickly adapted to the platform through individual learning guided by our Sponsor, Cyberweb, who provided thorough tutorials and demonstrations. We maintained constant contact to ensure cohesive outcomes and assert equal quality standards. In order to guarantee the highest quality, we spent a considerable amount of time perfecting the design of the initial chatbot, as it would later become the template for the rest. The remainder of the work consisted of less architectural work and, instead, more delicate reiterative work. Having now finished the task, our chatbots are live—or they are soon to be—and ready to help guests all around the country! Overall a learning experience and a wonderful contribution to the hospitality industry.

Keywords: Chatbots, Service Industry, Automation, Artificial Intelligence

Architecture

Every Chatbot in the DialogFlow application consists of two things: Intents and Responses. Intents are the user questions and Responses are the Chatbot output



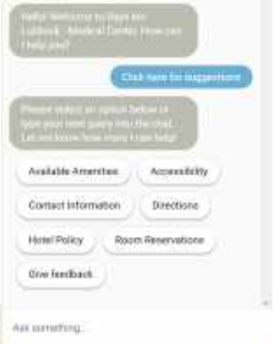
```
{
  "richContent": [
    [
      {
        "title": "Nearby attractions:",
        "text": [
          "Medieval Times [illegible]",
          "Disney World 8mi",
          "Disney Springs 7mi",
          "Universal Studios, Islands of Adventure, Volcano Bay 16 mi",
          "Pirates Cove Miniature Golf 5.5 mi"
        ],
        "type": "description"
      }
    ]
  ]
}
```

Performance

Due to our communication efforts each of our Chatbots contain the collection of knowledge and work done by the entire team. Questions were shared, collaborated and improved upon by all members to generate the best possible Chabot performance.

Results

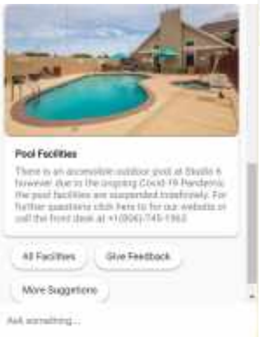
An example of one of our Chatbots Guiding the conversation for the customer



Hello! Welcome to Days Inn Lubbock - Medical Center. How can I help you?

Click here for suggestions

Please select an option below or type your hotel query into the chat. Let me know how I can help!



Pool Facilities

There is an accessible outdoor pool at Studio 6 however due to the ongoing Covid-19 Pandemic the pool facilities are suspended indefinitely. For further questions click here to for our website or call the front desk at +1(806)-745-1963

And another showing the results a customer requested

Impact

Our Chatbots are currently online on their respective hotel's website. They will help any customers that have questions about their stay and hopefully ease the workload of the hotel's customer service workers.

Summary

Working with Cyberweb we created 25 Chatbots all equipped with:

- Hotel Policy Information
- Useful and up-to-date links
- Images of the hotel
- Guided Conversation Patterns
- Transparent feedback mechanisms

Developing Chatbots for Hotel Websites

Toby Chow, Sarah Conaway, Tian Jiang, Kevin Liu, Zihan Wang
txc160330@utdallas.edu, slc170230@utdallas.edu, txj180011@utdallas.edu, kxl173730@utdallas.edu, zxw170530@utdallas.edu

CS 4485 / Spring 2021
Department of Computer Science
Erik Jonsson School of Engineering & Computer Science
The University of Texas at Dallas
Richardson, TX 75080, USA

Abstract:

The purpose of this project is to build a chatbot for hotel websites which are being managed by Cyberweb Hotels, LLC. The purpose of a chatbot is to make it easier for customers to find the information for which they are looking. The customer can ask the chatbot questions, and the chatbot will reply with automated responses based on keywords given by the customer. We used a Google service called DialogFlow to create these chatbots and hosted their UIs on remote websites. The final result of the project is the completion of 25 different chatbots.

Keywords: Chatbot, Google Cloud, DialogFlow, Customer Service

Architecture:

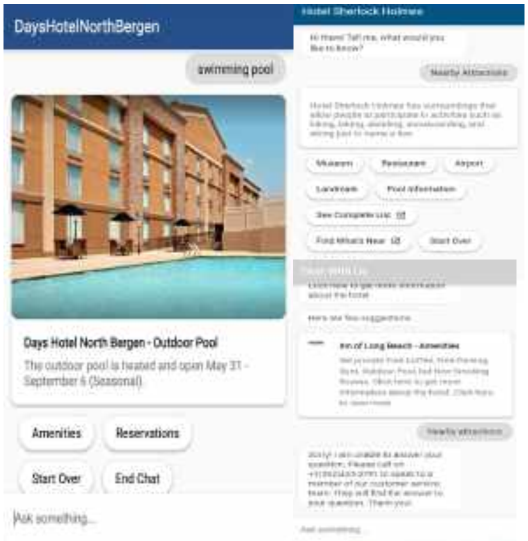
Front End

- HTML
- CSS
- JavaScript

Back End

- DialogFlow
- JSON

Results



Days Hotel North Bergen - Outdoor Pool

The outdoor pool is heated and open May 31 - September 6 (Seasonal)

Hi there! Tell me, what would you like to know?

Hotel Sherlock Holmes has surroundings that allow people to participate in activities such as hiking, biking, skating, snowboarding, and skiing just to name a few.

Click here to get more information about the hotel

Here are few suggestions

Inn of Long Beach - Amenities

[illegible]

Sorry! I am unable to answer your question. Please call on [illegible] to speak to a member of our customer service team. They will find the answer to your question. Thank you!

Impact:

Having chatbots for websites help enrich user experience as well as save companies time and resources by automating the questioning answer process.

Finishing this project has allowed us to provide these benefit to 25 clients of Cyberwebhotels.

Performance:

Goals were met and the company sponsor was pleased with work performed this semester from weekly meetings with sponsor and team advisor. Delivered chatbot model which functions well and is user friendly.

Summary:

The developed chatbots can reduce workloads of the hotel frontdesk. It can automatically answer general questions and provide instructions for perspective guest.

DragonRuby
Game Samples

Stephen Duffy, spd170330@utdallas.edu; Bowen Jiang, jxb170011@utdallas.edu; Nigel Lacewell, nll150030@utdallas.edu; Matthew Rachele, mor170030@utdallas.edu.edu; Jacob James, jrj170130@utdallas.edu;

CS 4485 / Spring 2021
Department of Computer Science
Erik Jonsson School of Engineering & Computer Science
The University of Texas at Dallas
Richardson, TX 75080, USA

DragonRuby

Abstract

Creation of video game sample applications utilizing the DragonRuby Engine to display the functionality and features, as well as create new ones in the production of commercial video games made through the Engine. DragonRuby uses ruby as its native scripting language and acts as a display and console while in the process of editing. Our sample applications will be used to further the development of DragonRuby as an IDE and provide a backbone for future developers utilizing the platform.

Keywords: Applications, Game Engine, DragonRuby

Architecture

DragonRuby Engine
The DragonRuby Engine acts as a console or terminal overlay that is also a Ruby REPL. By saving a file as a .rb and running it using a batch file, the DragonRuby console displays the current results.

Impact

- **Dragonruby has benefitted from project functions pushing the limits of the engine**
- **Users of the Dragonruby engine can examine new physics simulating functions through sample apps**
- **Users of the Dragonruby engine can examine visual effects in through sample apps**

Performance

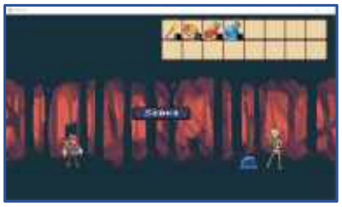
- **Able to calculate physics for looping and wall jumping**
- **Able to create a map editor**
- **Able to implement background parallax, easing, screen shake, and zoom**
- **Able to implement ray casting to add shadow casting to the engine**

Summary

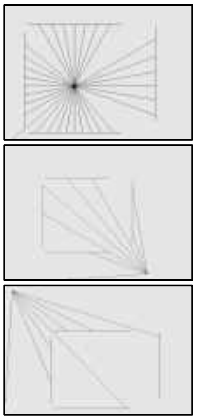
- **These templates provide some features that DragonRuby didn't have before.**
- **Beginners will find it easier to make similar games by using these templates.**
- **DragonRuby provides easy access to beginner-friendly cross-platform 2D game development.**

Results

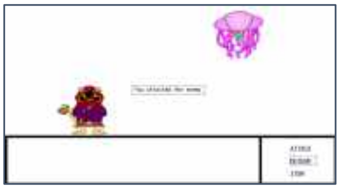
Inventory Interact RPG with Screen Shake and Zoom



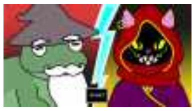
Shadow Casting Presentation Using Ray Casting



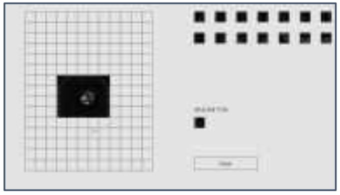
1. **A Pokemon-like Battle Scene**



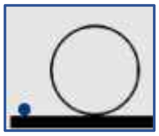
2. A Pokemon-like Battle Scene

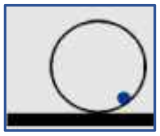


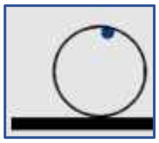
A Map Editor

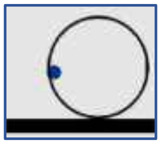


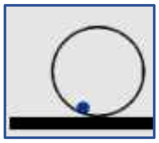
Sonic-like Loop



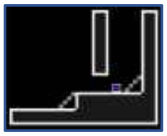


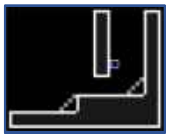


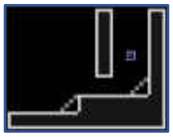




Simple Platformer with Walljumping







Wag!

Mobile Application Utilizing Rest API's through Django and Xamarin.Forms

Alec Dosier - ald170830@utdallas.edu | Jonathan Harvey - jeh180000@utdallas.edu | Jason Uwaeze - jason.uwaeze@utdallas.edu
Arsenios Scrivens – acs170003@utdallas.edu | Nabil Ajmeri - naa190002@utdallas.edu

CS 4485 / Spring 2021
Department of Computer Science
Erik Jonsson School of Engineering & Computer Science
The University of Texas at Dallas
Richardson, TX 75080, USA

Abstract

Create an Pet Adoption Mobile Application that enables animal lovers to browse, adopt, and interact with available sheltered dogs. Our solution was to create a mobile app using xamarin forms and C# for the front end involves and Django as our backend. Our solution involves capturing user information and preferences in the application, and storing the information in a SQLite database with a RESTful API using Django and Python. Xamarin Forms, a Microsoft visual studio extension, to create a solution for both Android and IOS. The purpose for our solution is to create a more efficient way to complete the adoption place, creating a space where people to search for pets, complete required documents, and interacted with the shelters all within the same app.

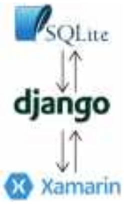
Keywords: Mobile App, Xamarin, Django, Pet Adoption

Architecture

Client Application
Xamarin.Forms is an open source cross-platform development framework used for creating IOS, Android, and windows app with .NET from a single codebase or project.

RESTful API using Django
The back end API utilizes the Django rest framework written in Python. JWT is used to provide authentication tokens for access control on each service request. Django also provides administrative tools to oversee and control the back end.

SQLite Database
SQLite is lightweight and is embedded into the API that makes secure transfer from the database to the API unnecessary due to its immediate location in the API.



Persistent Storage Using SQLite

RESTful API Using Django

Xamarin Forms Cross-Platform Front End

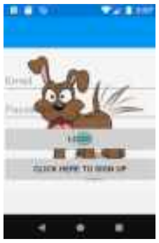
Impact

- Majority of shelters within Dallas-Fort Worth are using outdated software, our solution will help modernize their technology.
- Our solution holds as framework for later monetization such as advertisement and expedition of services
- Virtual pet adoption option for those with Covid-19 restrictions
- Pet Lovers and Shelters can experience a faster pet adoption process.
- Application can potentially decrease the amount of animals kept in shelters costing everyone less money.
- Pet Lovers can become more information with an interactive application.

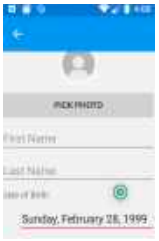
Performance

- Out of 40 people surveyed, 30% experience bug related issues with the application
- Out of 40 people surveyed, 90% would recommend Wag! to a friend
- Out of 40 people surveyed, 82.5% prefer Wag! Over current pet adoption services in their local shelter
- Able to run SQL Commands to store and collect information in the database with 100% accuracy

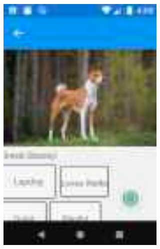
Results



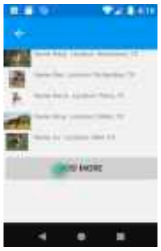
User Login Page
First time users can elect to sign up



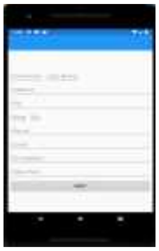
User Profile Page
User can create a profile to be viewed by shelters



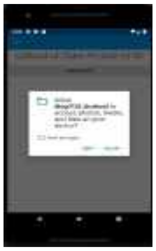
Browse Page
User can browse available pets



Saved Pet Page
User can view the pets they saved



Adoption Form Page
User can fill out applications for pet adoption



Proof of ID Page
User can upload their ID as proof to the shelters

Summary

- Wag! is a mobile application that allows for an easier pet adoption process
- Wag is aspiring to create an inviting and smooth user interface while maintaining a strong communication with shelters
- User will be able to choose and interact with available dogs from shelters from all over the United States.
- Database to save user preferences for future usage.
- User able to see available pets at local shelters.
- Wag! looks to increase the number of adoptions per year by allowing the user to complete pet adoption forms and communicate with the shelter from the same application
- Majority of shelters within Dallas-Fort Worth are using outdated software that is hard to navigate and not very user friendly.
- Applications like petfinder.com and petango.com are typically user friendly but lack accuracy and communication with various shelters

Canine Activity Recognition

Halo Collar

Austin Harris, ash170000@utdallas.edu; Christopher Delagarza, cmd170001@utdallas.edu;
Taylor Smith, tws180001@utdallas.edu; Ranveer Singh, rxs174730@utdallas.edu;
Faculty Advisor: Dileep Godithi, dileepgangadhar.godithi@utdallas.edu

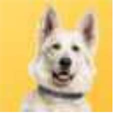
CS 4485 / Spring 2021
Department of Computer Science
Erik Jonsson School of Engineering & Computer Science
The University of Texas at Dallas
Richardson, TX 75080, USA

Abstract

We are working with Paws, LLC to improve the functionality of their Halo Collars by adding Activity Recognition ability of the system. This will increase the collars' utility by allowing dog owners to better monitor, train, and provide care for their dogs. We collected video and sensor data on dogs and annotated the data in a custom Python application to create a training dataset. We trained models on the dataset using sklearn. Finally, we used emlearn to port our machine learning model from Python to C to use it for collar's embedded systems.

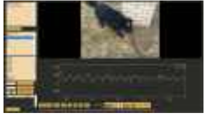
Keywords: Machine Learning, Activity Recognition. Embedded Systems

Architecture and Workflow



Collect dog sensor data

- Use of custom firmware to log collar sensor data
  - Jerked motion used for synchronization
- Real Time video feed for annotation only



Data Annotation using sensor data and video

- Custom Annotation Software developed in Python
  - Video Feed and Collar Data combined using Synchronization points
  - Sensor data visualized using Matplotlib
  - Set of labels provided for annotation
  - Shortcuts provided for easy labelling
  - Creates Annotation File for Labels

Deploy Model to Collar and Predict

- The model C header file added to the firmware
- Model predicts labels using the function defined in the header file

C Model using ***Emlearn***

- Python library emlearn
  - Converts sklearn Naive Bayes model to C header file combining Sensor data file with labelled data file

Data Preprocessing, Feature Extraction, and Model Training

- Data Preprocessing
  - Combining Sensor data file with labelled data file
- Feature Extraction
  - Feature extraction and aggregation of training data
- Model Training
  - Training a Naive Bayes Model in sklearn

Impact

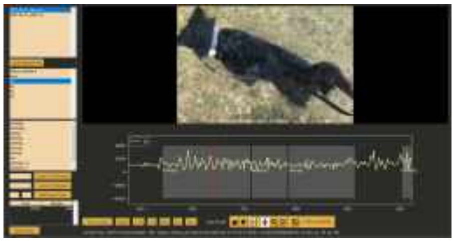
Prior to our project, Halo Collars didn't have any activity recognition for dogs. By the end of the semester we were able to create annotations software for easy data annotation, and identify a subset of activities reliably. Paws will use our workflow and tools to further scale up the training data for the model to better predict a dog's activities.

Performance

- The team kept the schedule set at the beginning of the semester.
- Created intuitive data annotation software for labelling
- Model Performance
  - Accuracy: 0.75 - 0.8
  - F-Score: 0.75 - 0.8
  - Size: 7KB
  - Speed: Real Time predictions

Results

- Documented workflow for gathering data and a working model
- Programs and Software:
  - Annotation GUI
  - Data processing pipeline
  - Python & C model code
- Set of annotated data, including video captures



Summary

We developed a workflow to gather training data and produce successful activity prediction models, along with several pieces of software to use in the workflow. Data is collected and then annotated using a GUI. The annotated data is used to generate a model which can be ported onto the collar's embedded systems.

Halo Collar

Data Analytics

Manoj Kumar Selvaraj, mxs170065@utdallas.edu | Emanuel Urbina, exu170030@utdallas.edu | Mark Mondt, markmondt3@gmail.com | Joel George, joel.george@utdallas.edu | Saketh Gollapudi, sxg170029@utdallas.edu

CS 4485 / Spring 2021
Department of Computer Science
Erik Jonsson School of Engineering & Computer Science
The University of Texas at Dallas
Richardson, TX 75080, USA

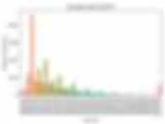
Abstract

Halo is a pioneer in the realm of smart fence solutions for pets. The Halo Collar coupled with the Halo App provides virtual on the go fences. The Halo Data Analytics project focused on gathering insights from various Halo data stores. The Halo cloud data store and the product database contain the user generated data and the smart collar generated data. The team's solution was to query the data from the various data stores and perform analytics via python utilizing different data science libraries. The outcome of the project was key insights and metrics regarding the product and users which can help improve the technical functionalities and business operations of the product.

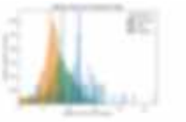
Keywords: IoT, Cloud, Data Analytics, Data Mining

Results

Business Report



Internal Dashboard



External Dashboard

Due to the nature of the project, the deliverables will be confidential, as requested by the sponsors.

Architecture

Data Analysis

- Python
  - Pandas
  - Matplotlib
  - Seaborn
- KQL
- SQL

Deliverables

- Power BI

Performance

- Schedule Performance Index: 85% of scheduled tasks were completed in accordance with the outlined project timeframe and schedule. Some tasks took more time than anticipated due to technical complexity.
- Analytics Satisfiability Rate was 82%, in that most of the questions (~21 out of 25) were correctly completed. Some questions were not able to be answered due to technical complexity and data unavailability.
- Overall, the sponsor was satisfied with the insights provided and the presentation of data metrics that were delivered.

Impact

The project's main focus was on understanding the product usage. The insights formed such as the average time spent on each tab or how many fences were created can help better understand important patterns and details. This can lead to streamlining certain features or introducing additional functionalities, all of which can help deliver a better product experience for the end user. In addition, the company can hone in on components that are the most beneficial to users and prioritize those elements. From a business perspective, the user and pet demographics insights can be used for marketing campaigns and improving sales funnels.

Summary

- Provided insights on product usage, user demographics, and marketing metrics.
- Improved analytics to allow for a sustained and more deeper form of analysis of data from multiple sources.
- Generated dashboards that supported advanced data visualizations and provided key technical and business information via business report.

Halo Collar

Wireless Leash

Bryan Stitt, bryan.stitt@utdallas.edu; Nguyen Nguyen, nguyen.nguyen6@utdallas.edu; Jose Lopez, jose.lopez9@utdallas.edu; Adrian Varela, adrian.varela@utdallas.edu; Abdul Moiz Aqil, abdulmoiz.aqi@utdallas.edu;

CS 4485 / Spring 2021
Department of Computer Science
Erik Jonsson School of Engineering & Computer Science
The University of Texas at Dallas
Richardson, TX 75080, USA

Abstract

The Halo Collar – Wireless Leash project is aimed at creating a wireless leash that allows users to keep their dogs within a certain distance during a walk. The project focused on research and data collection for different hardware found within the collar, the hardware included: Bluetooth, GPS, and accelerometer.

For all the listed hardware, the team was responsible for finding the most accurate one and alternatives if the most accurate one became unreliable or unavailable. Using a combination of these three would allow the application to ascertain the distance with reasonable accuracy between the user and the dog. We would develop an algorithm that is able to satisfy this goal.

Keywords: Geopositioning, Wireless Collar, Data Analysis, RSSI

Architecture

Android and iOS Applications
- SensorLog
- Accelerometer Analyzer
- Halo Collar
- NRF Connect

Backend and Database
- RealTerm - Microsoft
- CoolTerm - MacOS
- Microsoft Excel

Impact

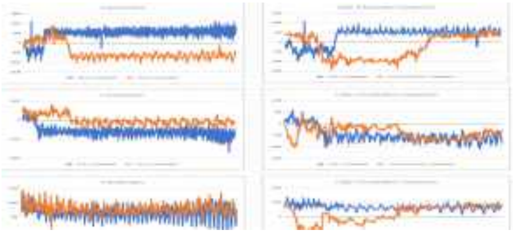
The purpose of this project was to find the most accurate method in finding distance between the company's product (Halo Collar) and the user's phone (Android or iOS). From data collection and analysis, we have concluded that the best method for distance measurement that is less than 3 feet is by using Bluetooth and the best method for distance measurement that is further than 3 feet is via GPS. With this information, we hope to save the company time and resources in data collection and analysis.

Performance

- Weekly task completion rate: 90%
- Data analysis is complete but algorithmic analysis is left uncompleted

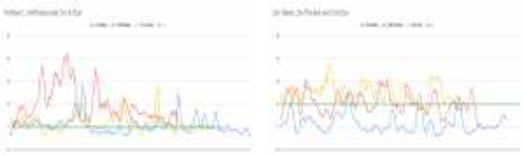
Results

Accelerometer



GPS

Bluetooth



Summary

- The percent error for GPS increases as distance decreases, therefore GPs is the best method for long distance measurement.
- The best method for short distance measurement is using Bluetooth, although its signals do through many bridges.
- Accelerator can be used to a certain measure when the dog is moving or just sitting.

Adding Navigation Autonomy to a Robotic Platform using Machine Learning

John Marshall Morton (CS) jmm170030 @utdallas.edu

Sam Ali (CS) sba170030 @utdallas.edu

Nicholas Dobmeier (CS) njd170130 @utdallas.edu

Seyed Maziar Mirrokni (CS) sxm178031 @utdallas.edu

George Yu (CS) gxy170330 @utdallas.edu

Johnathan Schneider (CS) jws170030 @utdallas.edu

James Tran (EE) tranj @utdallas.edu

Orobosa Ogbemudia (EE) omo170000 @utdallas.edu

Rishab Nair (EE) rmn170330 @utdallas.edu

Student Supervisor:
Umakanthan Lokeswari

Faculty Advisor:
Neal Skinner, P.H.D

Industry Sponsor:
Karl Kevilus

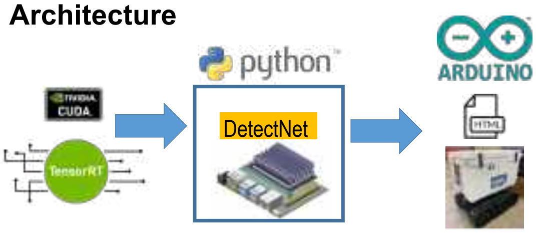
CS 4485 / Spring 2021
Department of Computer Science
Erik Jonsson School of Engineering & Computer Science
The University of Texas at Dallas
Richardson, TX 75080, USA

Abstract

The need for autonomous delivery has risen due to the pandemic. HEB seeks innovation by developing an autonomous delivery system that does not suffer from the space and weight constraints as other solutions. The company provided us with a radio-controlled prototype of the TANK and requested a fully autonomous solution using object detection tools such as cameras, sensors, and computer vision processing boards. Over the course of the semester the CS and ECE teams worked on acquiring and implementing the necessary hardware, and the CS team worked on software development in parallel with the hardware acquisition.

Keywords: Computer Vision, Machine Learning, Autonomous Navigation

Architecture



Phase 1: **TensorRT** is used for deploying efficient neural networks using graph optimization, kernel fusion and FP16/INT18 precision. **TensorRT** is built on the parallel programming model **CUDA.**

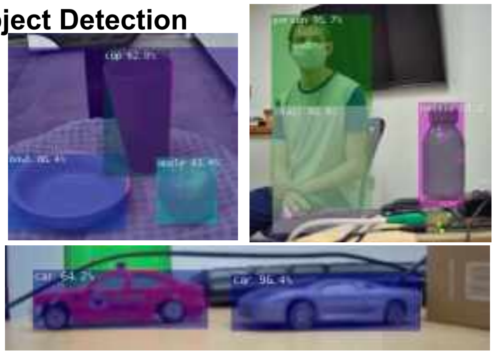
Phase 2: DetectNet inherits from TensorNet which comes from the **TensorRT SDK.** DetectNet is a vision primitive implemented in **Python.**
Phase 3: Through a serial connection, the **Jetson Nano** connects to the **Arduino Uno WIFI Rev 2** An Arduino library, called **WiFiNina**, can create a wireless access point, to which a simple **HTML** interface can sends a byte stream to Arduino's WiFi card for interpretation.

Impact

The growth of grocery store pick up and online retail has fueled the demand for more *efficient* and *economical* ways to deliver products to customers. Autonomous delivery robots offer a solution to the problem; however, current options have *significant* size and weight limitations.

HEB aims to provide an additional contactless service option to customers, as well as lower delivery costs, reduced manpower required for deliveries, and a solution to size and weight limitations of current delivery robots.

Object Detection



Performance Metrics

Qualitative Metrics:
Weekly meetings with the faculty advisor, company sponsor, and ECE team to determine how much progress has been achieved with the hardware, and to plan for the software development allowed by the hardware in place. Completed elementary stages of object recognition through a live cam feed and ultrasonic sensors. Provided a solid basis for further development towards a fully autonomous solution.

Quantitative Metrics:

- Frames per second: Over the course the semester, the fps was improved from 2 to 10 to around 20 FPS.
- Accuracy of object detection: With a threshold of 0.5, we can achieve 50% to 80% confidence on live footage.

Hardware

- Jetson Nano: Object detection processing board
- Arduino Uno Wi-Fi Rev2: Control the motor of the TANK by sending commands to the Sabretooth controller using HTTP
- Logitech Camera: Provide live feed footage for the purpose of object detection and path finding
- Ultrasonic Sensor: Detect distance to objects (0.3-14 m range)
- GPS and Compass: Determine current position of the TANK

Summary

- Added additional hardware to the TANK prototype.
- Partially converted the TANK prototype from a radio-controlled device to being dependent on computer vision.
- Implemented object detection from a live video feed.
- Established communication between the Python program, Jetson Nano, and the Arduino Uno.

Mapping Distributed Binary Objects to their Code Projects

Bilal Tola, Jacob Kitchens, Nathan Price, William Su, Musab Ashraf, Jesus Galera
Mbt170000@utdallas.edu, jak170430@utdallas.edu, nmp170330@utdallas.edu, wxs171030@utdallas.edu, mxa177730@utdallas.edu, jgg170230@utdallas.edu

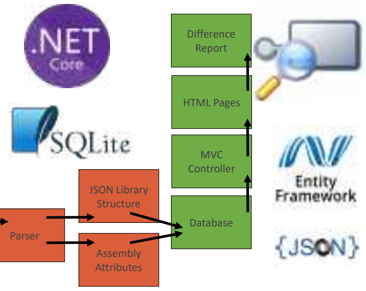
CS 4485 / Spring 2021
Department of Computer Science
Erik Jonsson School of Engineering & Computer Science
The University of Texas at Dallas
Richardson, TX 75080, USA

Abstract
Provide solution to enable split-second impact analysis for IT assets. Generate cross reference report to help determine likeness between two or more compiled objects. Provide web interface for easily accessed informational overview and database management.

Keywords: .NET DLL, Decompiler, .NET Core, Entity Framework

Architecture

- .NET Core
- Entity Framework
- ILSpy API
- SQLite Database
- Razor pages



Impact

Blazing-Fast Binary Mapper provides IN-COM clients split-second impact analysis for IT assets.

Before:

- Manual library comparison during product compilation
- Potentially looking through several different versions of referenced assemblies

After:

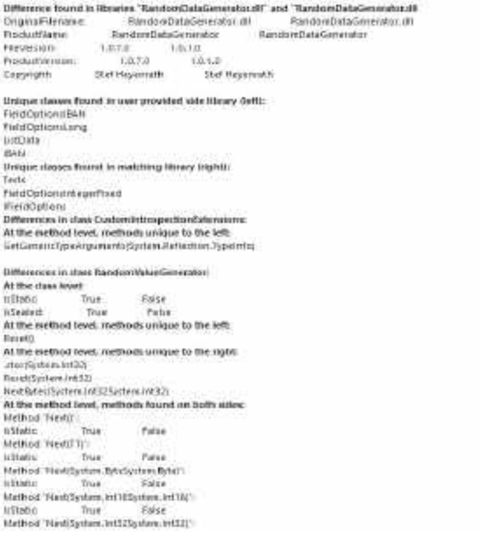
- Automatic library comparison to assist in selecting the most updated or useful library to the compilation

Performance

- Report is generated efficiently using best practices with layered architecture
- Code binaries are decompiled using a reliable and efficient library to ensure a performant decompilation
- Database queries are optimized internally through SQLite
- Web interface gives difference report by using the intersect and except extension methods

Results

We created the product with a simple web interface. It accepts a dll file as input and prints a comparison report to the screen. If there are multiple database matches, then multiple reports are generated one after the other. The report shows the difference in each section (classes, fields, methods, properties). It shows the unique values in either library and the differences at the lower levels.



```
Difference found in libraries "RandomDataGenerator.dll" and "RandomDataGenerator.dll"
OriginalFilename:        RandomDataGenerator.dll        RandomDataGenerator.dll
ProductName:           RandomDataGenerator           RandomDataGenerator
FileVersion:         1.0.7.0        1.0.1.0
ProductVersion:         1.0.7.0        1.0.1.0
Copyright:         Stef Heyenrath         Stef Heyenrath

Unique classes found in user provided side library (left):
FieldOptionsIBAN
FieldOptionsLong
ListData
IBAN
Unique classes found in matching library (right):
Texts
FieldOptionsIntegerFixed
IFieldOptions
Differences in class CustomIntrospectionExtensions:
At the method level, methods unique to the left:
GetGenericTypeArguments(System.Reflection.TypeInfo)

Differences in class RandomValueGenerator:
At the class level:
IsStatic          True          False
IsSealed          True          False
At the method level, methods unique to the left:
Reset()
At the method level, methods unique to the right:
.ctor(System.Int32)
Reset(System.Int32)
NextBytes(System.Int32System.Int32)
At the method level, methods found on both sides:
Method 'Next()'
IsStatic          True          False
Method 'Next(T[])'
IsStatic          True          False
Method 'Next(System.ByteSystem.Byte)'
IsStatic          True          False
Method 'Next(System.Int16System.Int16)'
IsStatic          True          False
Method 'Next(System.Int32System.Int32)'
```

Summary

- Essentially, by creating relationships between DLLs in a stored database, we enable for simpler compilations for IT assets
- Difference reports will present information to IT assets that will assist in adding the most recent or usable libraries into an enterprise product
- Moving forward, the future of our product would expand to multi-language support, being able to decompile several different languages to gather assembly attributes and structure for further comparison

Intelligent Occupancy Management Using Air Quality Measurement

Atharv Jain, axj170019@utdallas.edu; Sivam Patel, sxp180045@utdallas.edu; Gautam Sapre, gss170001@utdallas.edu, Sanjana Sivakumar, sxs170002@utdallas.edu

CS 4485 / Spring 2021
Department of Computer Science
Erik Jonsson School of Engineering & Computer Science
The University of Texas at Dallas
Richardson, TX 75080, USA

Abstract

Murata Manufacturing Co. has tasked the UTDesign Team with creating a low-maintenance system that successfully predicts occupancy in a building through non-intrusive sensing technology. This multi-semester project has consisted of three teams that have worked on this occupancy estimation algorithm. Last semester, the CS team created an algorithm that estimates the occupancy using the hardware. This semester we focused on bettering that algorithm and bringing it closer to prediction. We also put efforts towards creating a user interface that will help view our data and algorithm results better.

Keywords: Machine Learning, Occupancy, Prediction, Hardware

Architecture

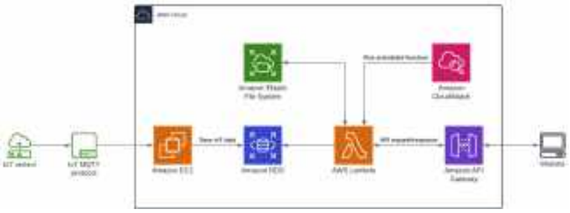


Figure 1: Flowchart of AWS services

Front-End
- React JS

Back-End
- Python
- Tensorflow

Database
- MySQL

Server
- Amazon Web Services

Impact

Over the past year, we have seen the need for occupancy management due to the Coronavirus. However, there are many other applications. From enhanced cognitive function for students to simply creating healthier public spaces, intelligent air quality management can improve people's lives. Whether it is monitoring air quality in areas affected by natural disasters such as wildfires or managing occupancy and ventilation to prevent the spread of disease, air quality management can improve people's health. By being able to accurately predict occupancy, we are also able to save money on HVAC and other resources costs by planning ahead.

Performance

- Prediction model had mean accuracy of 80% (20% error)
- Estimation model had mean error of 4 occupants
- REST API response time under 300 ms

Results

Figure 2: Module 1 data page

Figure 3: Sensor values vs Time graph page

Figure 4: Device status page

Several advancements were made this semester towards our project goals. First, we created some of our first models for predicting environmental factors such as C02. In addition, we created a website that allows stakeholders and team members to easily see the deployed IoT modules, their status, and the trends in their data. Furthermore, they can see the performance of future teams' prediction and estimation models, allowing them to evaluate model accuracy and weaknesses quickly. In addition, we have set up the cloud infrastructure needed for future semesters, allowing them to focus on model improvement rather than infrastructure. In addition, the cloud infrastructure contains our API, database, and additional jobs for monitoring the status of our IoT devices and continuously testing our estimation model to see how the model performs over time in various environments.

Summary

Overall, we were able to hit most of our objectives this semester. We were able to create a UI and the corresponding API. In our machine learning task, we began researching prediction and creating our first prediction models, giving us valuable insight into the problem space and possible next steps. Lastly, we were able to use cloud services to centralize our resources, enable continuous mode evaluation, and add additional requested features are such as module status checking.

References

An electric utility's 3-part guide to peak shaving with neural networks. (n.d.). Retrieved from https://www.brownanalytics.com/load-forecasting/

McElwee, K. (2020, January 24). Predict daily electric consumption with neural networks. Retrieved from https://towardsdatascience.com/predict-daily-electric-consumption-with-neural-networks-8ba59471c1d

Commodity Trading Systems

with Blockchain, Cloud and AI

Alan Liao, alan.liao@utdallas.edu; Andrew Jensen, andrew.jensen@utdallas.edu;
Carter Poe, cep150330@utdallas.edu; Nikhil Mishra, nxm171830@utdallas.edu; Varun Akula, varun.akula@utdallas.edu

CS 4485 / Spring 2021
Department of Computer Science
Erik Jonsson School of Engineering & Computer Science
The University of Texas at Dallas
Richardson, TX 75080, USA

PetroInnovation
Labs

Abstract

The Commodity Trading Platform requires the implementation of a blockchain network for consensus which, based upon the smart contract implementation, allows the platform to have trusty transactions and records. The ledger, also in each node, provides the means for persistent and redundant data storage. In the context of cloud development, each provider has a different platform for a similar technology, varying in terms of user flow, default security parameters and access control which makes the difficulty of implementation vary cloud platform to cloud platform.

Keywords: Blockchain, Cloud, AI, Kubernetes

Architecture

- Application
  - Node
  - React
    - Rendered page views for UI
  - Express
    - Middleware functionality
  - Nginx
    - Web server for reverse proxy
- Blockchain
  - IBM Blockchain Platform
    - HyperLedger-Fabric
    - Kubernetes

Impact

PetroInnovations Labs aims to reduce loss in global trade. Our work this semester was towards deploying the application microservices and a Hyperledger Fabric blockchain for a trustless, immutable trading platform.

Before

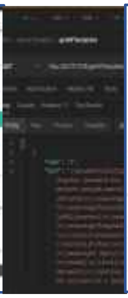
- Application out of date
- Could only be run locally
- No WebApp implementation or testing suite

After

- Fully deployed microservices
- Multiple blockchain deployments available
- WebApp documentation linked to Postman client

Results

WebApp deployed using Postman



Successful Postman API code test

IBM resource list on IBM private network

Blockchain Platform services on IBM

Performance

Most of the goals for this project were big-picture objectives. Thus, we have no performance metrics as a product of that. Our work either functioned or didn't. Optimization, for us, meant simply that the code was implemented properly.

We kept the application runnable. We proposed a three tier architecture with a modular user system and code. We were able to find workable deployments to get the application running and we added API documentation.

Summary

We were able to correct broken dependencies that would not correct via normal updates. We were able to test the application locally after that. We have an API documentation and testing suite that was built in Postman which should simplify the preexisting documentation. The current url is set for the floating ip for a vpc we have the app deployed on but this can be modified to any address we see fit. We have a couple different deployments for a hyperledger fabric. The first was leveraging the IBM blockchain extension with the remote development toolkit extension in VSCode to deploy a small microfabric on the VPC previously mentioned. The second Blockchain deployment is through the IBM Blockchain Platform where we have a similar network configured.

PonyUp Breeding App

David Cordoba(dcc170001@utdallas.edu), **Shaz Dattoo** (shaz@utdallas.edu), **Alexis Zepeda** (amz170130@utdallas.edu), **Jacob Waller**(jacob.waller@utdallas.edu)

CS 4485 – Spring 2021
Department of Computer Science
Erik Jonsson School of Engineering & Computer Science
The University of Texas at Dallas
Richardson, TX 75080

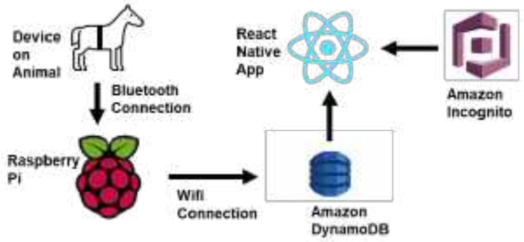
UTD
THE UNIVERSITY OF TEXAS AT DALLAS
Erik Jonsson School of Engineering and Computer Science

Abstract

The PonyUp Breeding App is designed to show real-time data from the Birthing Monitor device created by the Engineering Team. Our application is compatible on both iOS and Android, and includes various functionality that allows the user to create an account, register/edit new devices, and display live data uploaded to our AWS database. The purpose of this build is to connect breeders with a device that allows them to see the status of their animal during their birthing phases. The data that is visible through the device and application include, but is not limited to, temperature, contractions, accelerometer, and gyroscope data. Throughout the course of this build, we have successfully created an interactive application that displays accurate information that is current to the data uploaded into the database. For testing purposes, a dummy device has also been created to simulate the birthing monitor device and uploads this forged data directly to our database.

Keywords: Mobile App, API, iOS, Android, AWS, DynamoDB, Expo, React Native, Java

Architecture



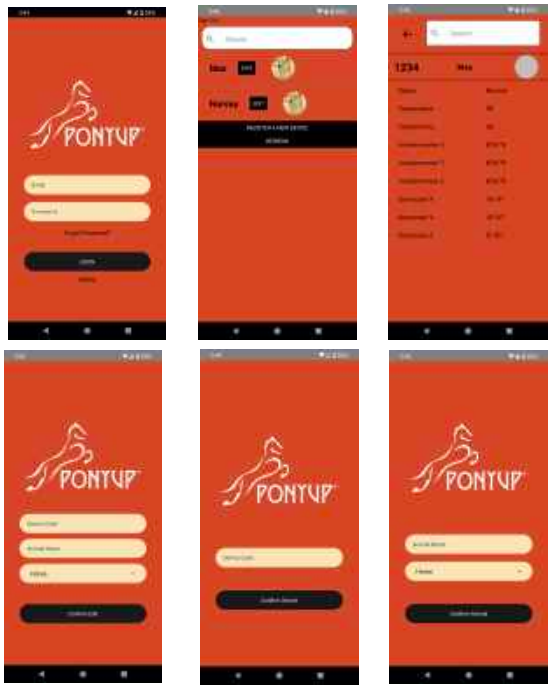
Impact

- Offer a simple way to allow breeders to check the temperature, contractions, and other vitals of an animal.
- An increase in the probability of survivability of the animal and offspring during the birthing process.

Performance

- The PonyUp App through help of a dummy device can present animal data to user
- React Native development ensures that user experience is consistent across the Android & iOS platforms.
- Engineering team is given all raw data from app for ease of trouble shooting
- Minimalist App Design Keeps Project Intuitive to Users

Results



Summary

The PonyUp Breeding App,

- Allows users login or create new accounts securely.
- Users are able add their device and put info like name and type of animal
- Connects to a database where the animal data is stored.
- Screens Added to include editing of device to fix any changes in user error
- Dummy device gives PonyUp App all possible forms of data from hardware of Engineering Team

Metrics

- Our metrics are based on our completed requirements.
- We completed all of our planned core requirements.
- We were unable to fully implement the search bar functionality

Transistor Load Pull Data Analysis

API and GUI Application

Adhiraj Sen, *adhiraj.msm@gmail.com*; Ashby Joseph, *ashbyjosephpro@gmail.com*; Hanna Kessler, *hanna.kessler10@gmail.com*; Rutvij Shah, *rutvij@rutvijshah.dev*; Subhash Reddy Balam, *bsubhashrdy@gmail.com.*

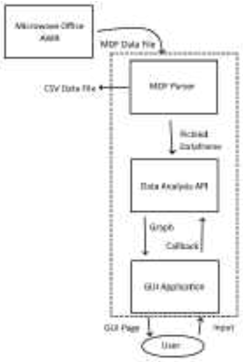
CS 4485 / Spring 2021
Department of Computer Science
Erik Jonsson School of Engineering & Computer Science
The University of Texas at Dallas
Richardson, TX 75080, USA

Abstract

We utilized Pandas, NumPy, and Matplotlib Python libraries to create an API and GUI application that cleans, interpolates, analyzes, and plots transistor load pull data. This will allow Qorvo engineers access to other Python libraries while working with their data, as well as a functional GUI for those engineers who do not wish to code.

Keywords: Python, Pandas, NumPy, Load Pull, PAE, API, Gain Compression

Architecture



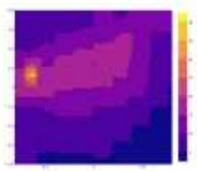
Impact

The impact on Qorvo is access to a nice way to analyze their load pull data in order to make good decisions related to that data. The graphical visualizations can be easily used to digest the data quickly and can be shared in internal meetings and discussions.

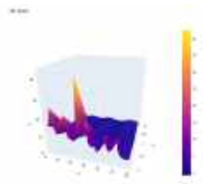
Performance

This project is usable in its current state. The API and GUI are functional, and some analysis can be done with the data. There is certainly room for improvement in adding more ways to analyze data, however our code is solid in its usability and stability.

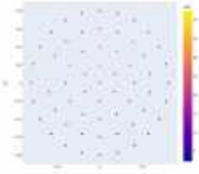
Results



Real vs Imaginary Gamma Points and the PAE are the Isolines in the Contour Plot.



Real vs Imaginary Gamma Points and Gain forms the Iso-surfaces in the 3D-Surface Plot.



Real vs Imaginary Gamma Points, within the complex plane and the PAE is the third dimension

Summary

The purpose of this project is to analyze transistor load pull data. Specific functions include parsing the data generated by Microwave Office (AWR) into a usable format, filtering data that meets certain power levels, interpolating and slicing data, G compression, and visualizing data with contour maps. This functionality is available through an installable Python API which gives access to other Python libraries to those engineers at Qorvo who want to dig deeper into their data. It is also available through a GUI application which allows engineers who do not wish to be involved in coding to simply and easily visualize their data.

CS 4485 / Spring 2021
Department of Computer Science
Erik Jonsson School of
Engineering & Computer Science
The University of Texas at Dallas
Richardson, TX 75080, USA

ML Control of Load Pull Simulations

Andrew Fu,
axf170008@utdallas.edu;
Nolan Kuo,
nxk170023@utdallas.edu;
Hemal Salvi
hns170002@utdallas.edu;
Nicholas Sherman De Morales
njs170000@utdallas.edu;
Priyanshu Swar
pxs170020@utdallas.edu;

Abstract

RF devices ("Radio-Frequency Devices") are used to wirelessly transmit signals between two pieces of electronic equipment, and they're used everywhere to enhance our day-to . As one of the lead producers of RF Circuitry, Qorvo needs to constantly test and verify the design of a variety of circuits. To do this, a technique called RF transistor load pull is used to measure the performance of the device. Using software, load pulls can be simulated, but this takes a large amount of time based on the number of measurement points.

This project aims to lower simulation time by applying machine learning models to a lower amount of points to determine the parameters of a circuit with high accuracy.

Keywords:Machine Learning, Load Pull, RF Circuits

Architecture

Data Sourcing and Cleaning

We used Cadence AWR to run Load Pull simulation measurements, and Python to export and parse the data from the simulation.

Machine Learning

We used the Scikit-Learn library to test and apply various machine learning models to our initial datasets, and used Polynomial Regression from Scikit-Learn for our final application.

Application

We used TKinter to build our Application's UI and PyInstaller to create our final executable file.

Results

Cadence AWR, while extremely versatile, suffers from long simulation timings when applied to transistor load pulls. Our work lowers the measurement time needed per simulation to verify devices/circuits.

Before

- Slow script analysis of data based in Cadence

After

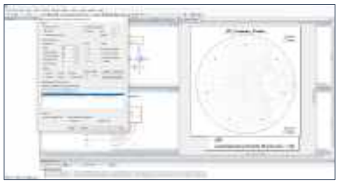
- Full backend analysis of data
- Automatic communication with Cadence AWR

Impact

- Offloaded simulation measuring to a backend script running at faster speed
- Increased the ease and speed of validating the robustness of a circuit/device
- Easy usability and application for any user

Performance

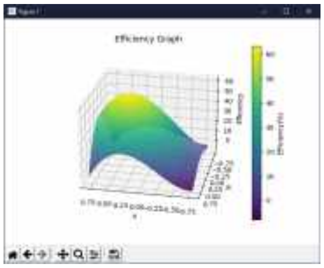
Our team was able to pull data from Cadence AWR into Python, and was able to show the results of running multiple machine learning models on sample sets of the data. We were additionally able to create an application incorporating our most efficient model (polynomial regression), thus fulfilling our obligations to the sponsor.



Cadence AWR Simulation

Application

Machine Learning Model



Summary

Over this semester, we've:

- Managed to get a method to run load pulls in AWR and collect the resulting data information (with help from the previous team and Team 29, the other team working with Qorvo))
- Trained multiple machine learning models on our data to check for accuracy, and got linear regression as our optimal machine learning model
- Developed an application to allow Qorvo engineers the ability to calculate the points of maximum power and efficiency for their load pull simulations

UTDrone

Unmanned Aerial System using Image Recognition and Machine Learning

Christian Huang:CSH140030@UTDallas.edu Garrett Hornok: GMH170001@UTDallas.edu
Kevin Lin: KXL180020@UTDallas.edu Kevin Nguyen: KHN170030@UTDallas.edu

CS 4485 / Spring 2021
Department of Computer Science
Erik Jonsson School of Engineering & Computer Science
The University of Texas at Dallas
Richardson, TX 75080, USA

Abstract

The Computer Science (CS) Team, sponsored by Raytheon Intelligence and Space, has developed a software package targeted to a flying drone platform capable of performing advanced computer vision, object classification, and navigation tasks. This system will be delivered on a Raspberry Pi 4 platform enabling real-time data processing and Machine Learning (ML) inference on the drone platform. We are cooperating with another team at UTD consisting of ECE students who are developing the hardware platform and flight system. Our work integrates image recognition to the drone, allowing it to generate a navigation path to the target.

Keywords: Machine Learning, Artificial Intelligence, Image Recognition

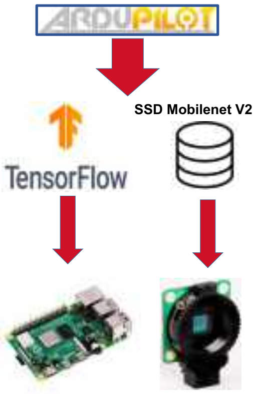
Architecture

SSD MobileNet v2
SSD MobileNet50 is a state-of-the-art convolutional neural network for object-detection with the base trained model boasting a median-average precision (mAP) of 22.2 using the COCO17 Metric and a speed of 22 ms for inference time (The second fastest among all SSD models)

TensorFlow Keras
TensorFlow Keras is an open course end-to-end machine learning platform with libraries that allow development and training of machine learning models.

Raspberry Pi 4
The Raspberry Pi 4 is the latest edition of the world's most popular small single-board computer (SBC). Programs capable of running on the Pi are also capable of being ported to other SBC's such as the Jetson Nano or TX2.

Ardupilot
Ardupilot is an open source autopilot software which allows us to implement flight libraries to fly the drone autonomously.



Impact

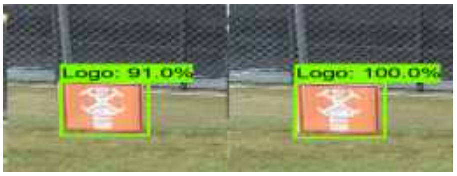
Image recognition in Unmanned Aerial Systems is currently used for autonomous flying, which can aid in medical transportation, resource delivery, transportation, and national disaster support. One UAS can hold a significant impact in crime reduction, as cities have shown decreasing rates by almost 10%. In disaster relief, emergency responses are faster with UASs, but also cheaper, allowing better aid to access smaller communities. For example, after a hurricane struck New Orleans, helicopters and water vessels assisted 24,135 rescues and evacuated 9,409 people from hospitals [1]. These helicopters can mean the difference between life and death, and if they are UAS's, they can be sent out within minutes of notice to begin a rescue.

1. "Coast Guard: Observations on the Preparation, response, and Recovery missions related to Hurricane Katrina." [Online]. Available: https://www.govinfo.gov/content/pkg/GAOREPORTS-GAO-06-903/html/GAOREPORTS-GAO-06-903.htm. [Accessed: 28-Apr-2021].

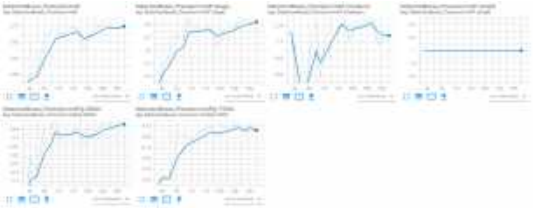
Performance

The model can accurately detect the UTDrone logo with a 84% confidence at ranges of 15-30 meters in validation runs and upwards of 99% confidence in point-blank to 10 meter ranges. This is decent enough to detect the logo from the viewpoint of the drone.

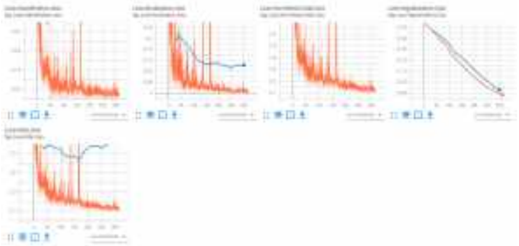
Results



Inference testing demonstrates accurate detection. Left picture shows inference. Right picture shows the ground truth.



Mean average precision (mAP) measures how well the model can detect true positives and avoid false positives.



Loss is the standard means of predicting the accuracy of a particular model as less loss means more accuracy. The loss function looks at the localization loss (which looks at the height, width, and center point of the bounding boxes) and classification loss loss (looks at false positives). The total loss of the model is 0.1159.

Summary

During the semester research was done to discover the current best practices in terms of model development and drone distance calculations. This led us to the SSD MobileNet V2 Model along with the Raspberry Pi execution engine. Our work will be used and put to the test in a Raytheon sponsored competition between UT-Dallas, UT-El Paso, and UT-Arlington. At this competition we will showcase our drones ability to autonomously fly and land at a specified distance, search for and land on the UTDrone logo, and navigate through an obstacle course.

ML Compression

Noah Alexander, noa170001@utdallas.edu; Mythreyan Ganesh, mrg180010@utdallas.edu; Andrew Miller, axm171230@utdallas.edu; Abhijith Venkat, axv180059@utdallas.edu; Jacob Richard, jcr180005@utdallas.edu

CS 4485 / Spring 2021
Department of Computer Science
Erik Jonsson School of Engineering & Computer Science
The University of Texas at Dallas
Richardson, TX 75080, USA

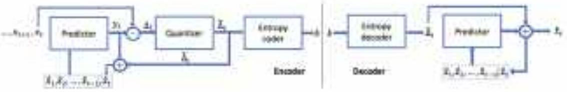
Abstract

This proposal describes the plan for developing a remote office surveillance event classification project. The most rigorous work will be put towards formulating a novel, state-of-the-art data compression technique, with an emphasis on video compression, using machine learning. Leading compression algorithms do not generally make use of machine learning. This offers an area of great potentiality assuming machine learning can transform this field as it has transformed many others. Through researching existing data compression techniques and adding our own new insights, we aim to create a new video codec to rival those in existence.

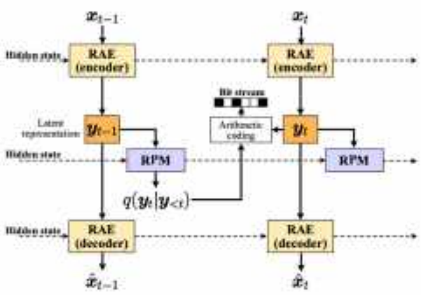
Keywords: Machine Learning, Compression, Video, IoT

Architecture

Sensor compression



Video compression



Impact

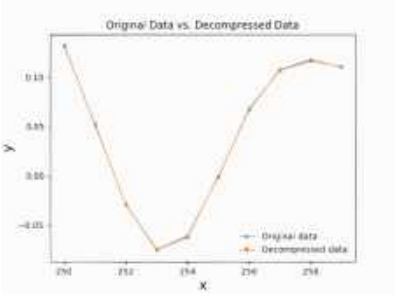
Our work on this project will leave Telaverge with two functioning implementations for two types of data compression: a machine learning, Recurrent Learned Video Compression program for video compression, and an effective non-machine learning program for IoT. These implementations will serve as the starting points for further development and optimization.

Performance

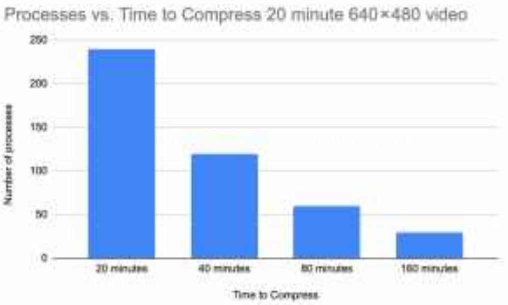
The sensor compression achieved a good compression ratio, time to compress, and losslessness.
The video compression achieved a good compression ratio and losslessness. The time to compress was slow but we propose parallel processing and compression of the model in future work. We implemented static data types for the arithmetic encoding.

Results

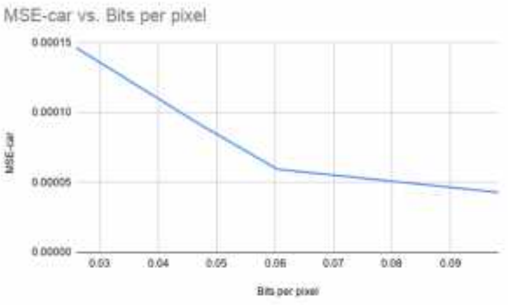
Sensor data results



Proposed amount of time to compress video for n processes



Mean squared error vs bits per pixel to compress video



hCiketi

Collaborative Application for Physician Consulting

Adrika Khan, Mark Luka, Farishah Nahrin, Maxx Tiger, Melecia Young
Adrika.khan@utdallas.edu, mark.luka@utdallas.edu, farishah.nahrin@utdallas.edu,
Maxx.tiger@utdallas.edu, melecia.young@utdallas.edu

CS 4485 / Spring 2021
Department of Computer Science
Erik Jonsson School of Engineering & Computer Science
The University of Texas at Dallas
Richardson, TX 75080, USA

Abstract

In the medical industry, two current problems revolve around the fact that doctors cannot share patient data between EMRs easily and that locations with low resources do not have access to specialized consults.

The hCiketi application is a telemedicine service that doctors can use to interact and consult with each other as well as use and share patient data through the browser interface.

Keywords: Full Stack, Video Conferencing, UI/UX

Architecture

Frontend
- React
- Material UI

Backend
- Node.js
- Meteor

APIs
- OpenEMR
- WebRTC

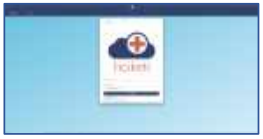
Impact

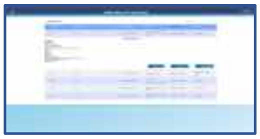
The hCiketi application aims to bring together some of the missing features that currently provide a hurdle to those working in the medical industry. Our work this semester was essential in bringing this app to life from its previous state as a few disjoint components.

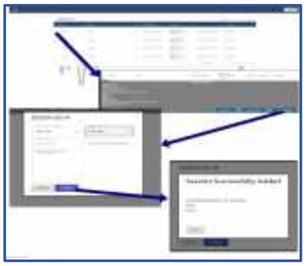
Before
- Multiple separate components such as paginated table
- All data is hardcoded
- No consistency in design

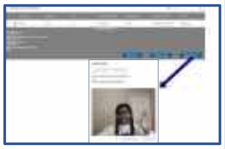
After
- Databases, login, hardware connections implemented
- Components connected, follow similar design language
- Application follows medical security standards

Results


Login Page


Home Page


Session Setup Flow


Starting a Scheduled Session

Performance Metric

- Bi-Weekly Scrum Sprint Task Completion Rate: 100%
  - All tasks assigned by sponsor were met
- App is cohesive and mostly ready for sponsor purposes
  - Backend and frontend are both complete, OpenEMR and WebRTC APIs were not fully implemented due to timing constraints

Summary

- Setup full stack secure login authentication and logout via Meteor and routing based on user.
- Revamped User Interface and implemented MaterialUI for consistency
- Setup support for adding patients, sessions, as well as session settings.
- Replaced all hardcoded values with actual data from the databases

Innovator Hub Data Acquisition Tool

Texas Instruments

Justin Haindel, Scott Jacobsen, Diego Perez, Patrick Soisson, Uyen T. Le
jlh170430@utdallas.edu, jacobsenscott07@gmail.com, dap170030@utdallas.edu,
pbs170030@utdallas.edu, utl170030@utdallas.edu

CS 4485 / Spring 2021
Department of Computer Science
Erik Jonsson School of Engineering & Computer Science
The University of Texas at Dallas
Richardson, TX 75080, USA

Abstract

Create a Graphical User Interface to help engineers visualize the data on their MSP432-P401R launchpad. This project will ultimately help engineers gather data measurements and provide analysis in an effective and cost-efficient manner. The major goal is to design a frontend system capable of displaying the data from the launchpad by utilizing an HTML5 solution that is easily accessible. This tool will make it easier for launchpad users to diagnose problems and tinker with their launchpad

Keywords: GUI, HTML5, UI/UX, CSS, JS

Architecture

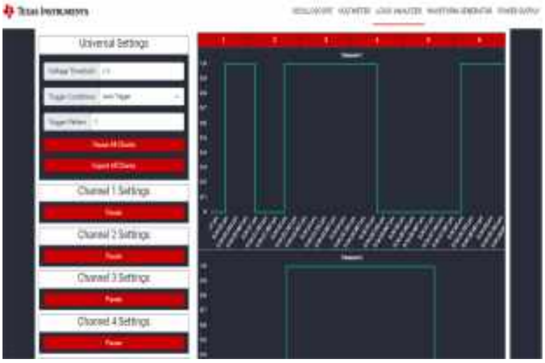
Front-End:
The front-end was built using HTML, CSS, JS

Hardware:
The MPS432 LaunchPad and the CC3100 Wi-Fi Booster Pack hardware are use.

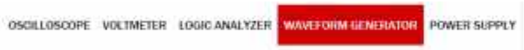
Impact

- Created a signal analysis tool that can be easily acquired, used, and maintained by any engineer.
- Made the user interface accessible and easily navigable while not compromising performance or information that the user may want to access.
- Created a much more informative and accessible dashboard for the TI-Innovator.

Results



Logic Analyzer Tab



Different Pages

Performance Metrics

- Weekly task completion rate: 90%
  - Some task took longer than expected.
- Project Scope was met (Functional expectations)
- Company mentor feedback
  - Goals were met and pleased with work performed.
- Successfully completed on time, scope, and budget.

Summary

- Designed a Graphical-User-Interface (GUI) for a TI tool that incorporated the following features: Oscilloscope, Voltmeter, Logic Analyzer, Waveform Generator, and Power Supply Controls.
- Each of these core features has its own unique page with its own relevant information and additional features such as downloading the data to a csv.

Wi-SUN Network Simulator

Phase 2

Kuanlin Liu, Kavon Ramsey, Raufil Dhuka, Aj Shah, Austin Nguyen, Jeong Lee, Mary Strebel
kxl180023@utdallas.edu, Kjr170230@utdallas.edu, rrd170130@utdallas.edu, ajs170530@utdallas.edu,
austin.nguyen1@utdallas.edu, jeongmin.lee@utdallas.edu, mary.strebel@utdallas.edu

CS 4485 / Spring 2021
Department of Computer Science
Erik Jonsson School of Engineering & Computer Science
The University of Texas at Dallas
Richardson, TX 75080, USA

Abstract

To study the performance and feasibility of Wireless Smart Utility Networks, Texas Instruments is creating a simulation tool that will emulate the Wi-SUN Standard in large networks. The simulation tool will allow TI personnel to determine the optimal system parameters for a Wi-SUN network.

The objective for Phase 2 was to continue the implementation of Wi-SUN features and to implement a way to take simulation metrics.

Keywords: Smart Utilities, Network Simulation, Internet of Things

Technology Overview

Standards

- 6LoWPAN
- RPL
- IEEE 802.15.4 MAC

Tools

- Contiki-NG
- Cooja
- Wireshark

Impact

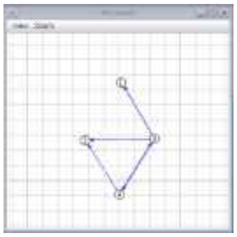
Smart Ubiquitous Networks aim to bring utility meters into the age of IoT in order to reduce infrastructure overhead by leveraging Field Area Networks. Our work this semester was the second step to completing this objective.

Accomplishments

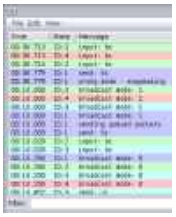
- Implemented unslotted channel hopping on the MAC layer
- Explored collect view and other methods of taking network metrics within Cooja

Results

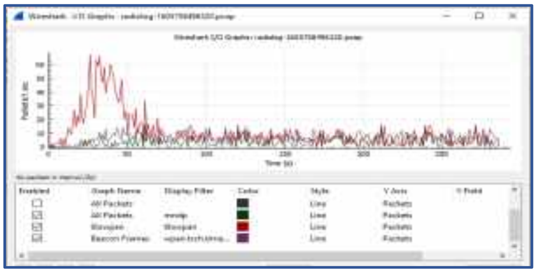
Graphical view of USCH simulation



USCH log output



Analyzing packet distribution with Wireshark



Performance Metric

- Performance was measured through weekly reports and bi-weekly meetings with company mentor.
- Company mentor feedback: goals were met and pleased with work performed this semester
- Functional expectations were met (completed functions do not have bugs or interruptions)

Summary

- Continued integration of Wi-SUN MAC in Contiki-ng/Cooja
- Studied network metrics and taking metrics within Cooja
- Compared features of Contiki and Contiki-NG

Intelligent CAN Fuzzing Test Automation (ICFTA) Prototype- Phase 3

Cathleen Peji, Lakshmi (Priyanka) Punukollu, Madison King, Reagan Davis, Sarah Burris, Shaira Rahman, Sydney Ward
cdp150030@utdallas.edu, lpp170030@utdallas.edu, mbk190001@utdallas.edu, rld170330@utdallas.edu, seb170005@utdallas.edu
sxr172230@utdallas.edu saw170030@utdallas.edu

CS 4485 / Spring 2021
Department of Computer Science
Erik Jonsson School of Engineering & Computer Science
The University of Texas at Dallas
Richardson, TX 75080, USA

Abstract

The TMNA Vehicle Security Team supports the research and development of a prototype CAN (controller area network) Intelligent Fuzzing test automation tool.

The objective of phase 3 is to refactor the legacy code by reducing code redundancies, correcting critical algorithmic errors, and increasing readability and documentation. Additionally, collected data shall be integrated into Elasticsearch, where it will be displayed and graphically analyzed with Kibana. Finally, the experiment will be stress-tested to detect potential vulnerabilities.

Keywords: CAN, refactoring, Elasticsearch, stress-testing

Architecture

Hardware:

- Toyota 17CY T0, T1, T2 Testbench
- 2018 Toyota Camry D171 Pioneer head unit
- Inno-maker USB-CAN-HAT-RPI

Software:

- Python-CAN
- Docker
- VirtualBox
- Ubuntu
- SSH Client
- Elastic Stack- Kibana

Impact

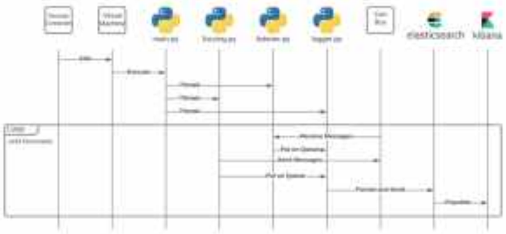
- System allows for TMNA to test the ECU's within their vehicles for backdoor vulnerabilities by sending large amounts of fuzzed data. This data can be generated by either brute force or permutation with arbitration ID's of their choice.
- The system could be used as a Red-Team tool to Pentest vehicles using a variety of algorithms and to discover arbitration IDs within the network.
- Graphical tracking and analysis of CAN received and sent messages (made available locally) can be used for long term tracking of message/arbitration ID changes.

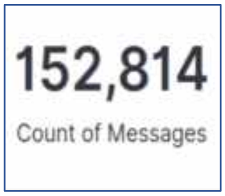
Performance

- Internal Group Milestones met: internal group requirements/deliverables achieved at tri-weekly basis
  - 100% attendance and goal completion rate for at least ⅔ meetings

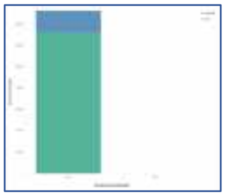
Results

Sequence Diagram of Application

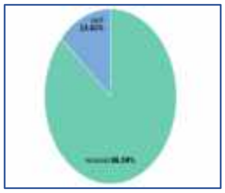




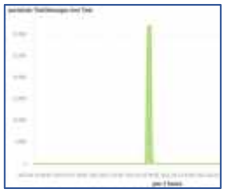
Total # messages for Permutation Fuzz mode for 24 hrs



Sent vs Received messages per Arbitration ID



Total sent messages vs received messages percentage



Total Sent Messages Over Time

Summary

- Software tool created to:
  - Automate CAN messages to a vehicle's CAN bus in brute force, and permutation fuzzing or listen-only modes from command line arguments.
  - Display and log the sent/received messages from CAN bus.
  - Send data to Elastic Stack for data storage and data graphical representation over time to serve as a GUI.
- Heavy refactoring done to improve readability and reusability.
- Documentation and encapsulation of software finished for phase 4.

Electronic Medical Records

Project Members: Jaewon Han, Kwonyoung Lee, Heejae Yoo, Boxuesong Zhao
jaewon.han@utdallas.edu, kwonyoung.lee@utdallas.edu, hxy172830@utdallas.edu, bxz160230@utdallas.edu
Supervisors: Dr. Sumana Nanjundachar, Dr. Prakash Kabbur

CS 4485 / Spring 2021
Department of Computer Science
Erik Jonsson School of Engineering & Computer Science
The University of Texas at Dallas
Richardson, TX 75080, USA

Abstract

Many hospitals in resource poor areas in India currently use paper systems to enter medical data including procedures. With this system, it is difficult to track and analyze patient data including procedures and physician performance.

The objective of this project is to create a frontend, backend, and database. Train and Help Babies organization (TaHB) aims to provide a medical procedure log and track physician skills/performance (procedures, outcomes, with patient data) for hospitals in India through our electronic medical record system.

Keywords: Database, UI/UX, Hospital

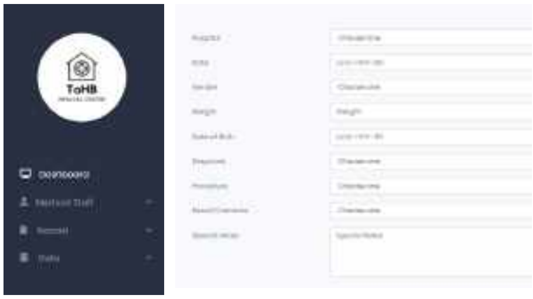
Architecture

Frontend
- Javascript
- HTML
- CSS
- jQuery
- Bootstrap

Backend and Database
- PostgreSQL
- Golang
- Gorilla
- AWS

Results

Medical Record List Page



Sidebar and Add Medical Record Page

Impact

- With this project, TaHB aims to provide a portal for entry of medical records for few hospitals in India, electronically. We will be able to track the quality of medical care with analysis of details of procedures and physician performance.
- The project this semester contributed to this goal.

Performance

- Functional Expectations were met (Website is ready for use)
- Website updates the database quickly (loads/adds records in less than a second)
- 90% of functional deliverables were met. Last 10% can be implemented with relative ease due to project foundation.

Summary

- Created and designed database.
- Created frontend website.
- Created Backend to tie database and frontend together.
- Created a website that implements a user system with 3 different users: Admin, Hospital Admin, and User.
- Created the ability to store and access previously created medical records.

Tyler Technologies Integration Framework – T1 Corrections Import

Shahneel Lalani, sxl105220@utdallas.edu; Anika Ahmed, ama170530@utdallas.edu; Hassana Saade, has170001@utdallas.edu;
Affan Chowdhury, aac140930@utdallas.edu; Muhammad Ahmad msa190004@utdallas.edu

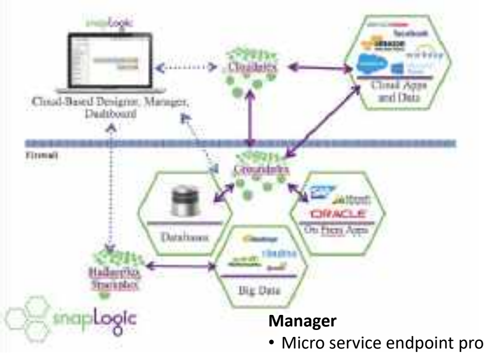
CS 4485 / Spring 2021
Department of Computer Science
Erik Jonsson School of Engineering & Computer Science
The University of Texas at Dallas
Richardson, TX 75080, USA

Abstract

To automate the way secure data from 3rd party court and justice clients is integrated into Tyler Technologies system by using the SnapLogic framework. SnapLogic is a deployable integration platform that automates the process of development by having an intuitive User Interface [UI], doing data automation, and having pre-built connectors to speed up the design process. Our solution was to read in 3rd party files and format into either JSON to XML or vice versa. Then that newly formatted data was imported to match Tyler Technologies system format.

Keywords: SnapLogic, Pipeline, JSON, XML

Architecture



Designer

- Univeral Access via 100% HTML5 based UI
- Minimal-coding with snap and configure
- Endpoint introspection
- Data preview

Manager

- Micro service endpoint provisioning
- Time and event-based scheduling
- User management
- Organization and project management

Dashboard

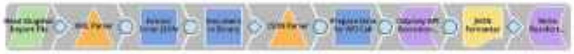
- Streamline mobile experience with HTML5 UI
- Visual Statistics
- Coarse and granular metrics

Performance

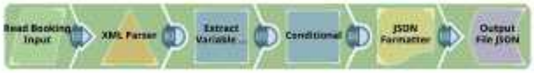
- Feedback and evaluation from everyone involved team members, supervisors, and mentors
- Pipelines connected to the Odyssey API to transfer data into the Tyler Technology system
- Pipelines were able to adjust format the imported data to be properly read in the Tyler Technology system
- All pipelines validated successfully and ran without interruptions

Results

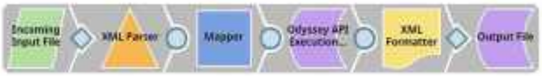
Mugshot Pipeline



Booking Pipeline



Party Caution Flag Pipeline



Impact

- SnapLogic lowers the cost and time of development process
- Transfers data reliably to Tyler Technologies system
- Reuses executions to process multiple documents
- SnapLogic eases usability for non-technical people

Summary

- The pipelines we created:
  - Imported a mugshot into the system
  - Imported booking information
  - Imported party caution flag
- Convert files from XML to JSON and vice versa.
- Format data to fit Tyler Technologies criteria for their correction system
- Used SnapLogic pipelines to automate the import process
- Future enhancements: Pipelines to be able to handle any type of input file

Tyler Corrections Export

Team 2

Alexander Vo, ahv160030@utdallas.edu; **Hansu Kim**, hxk170004@utdallas.edu; **Han Nguyen**, hmn180000@utdallas.edu; **Hareesh Parchuri**, hxp173530@utdallas.edu; **Karthik Sekar**, kxs167130@utdallas.edu

CS 4485 / Spring 2021
Department of Computer Science
Erik Jonsson School of Engineering & Computer Science
The University of Texas at Dallas
Richardson, TX 75080, USA

Abstract

Tyler Technologies required a framework by which Tyler corrections data could easily and seamlessly be exported to third-party systems in the required data format.

The solution needed to be adaptable to any currently required format, as well as formats that may be required in the future.

It also needed to be simple enough for a non-programmer to make effective use of.

Keywords: Data export, Pipelining, Cloud Computing

Architecture

- Snaplogic
  - Cloud-based integration platform
  - Create highly modular and intuitive data pipelines
- Python
  - Scripting

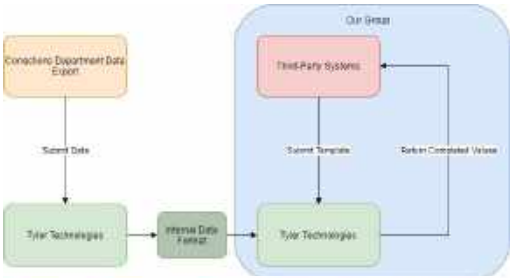
Impact

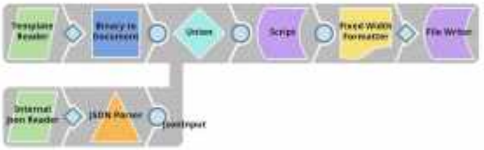
Using this system, exporting data to third party systems will be easy and seamless. Minimal programmer resources will be required to adapt to new data formats.

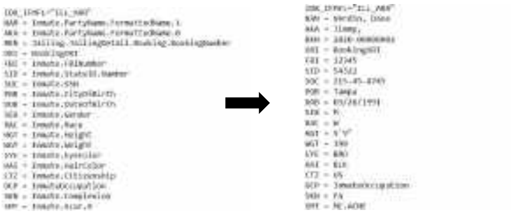
Performance

- The pipeline runs quickly and efficiently
- No programming required to implement new data formats
- Saves person-hours spent manually entering data

Results







```
IDB_IFMFt="Ili_NAF"
NAM = Inmate.PartyName.FormattedName.1
AKA = Inmate.PartyName.FormattedName.0
BKN = Billing.BillingDetail.Booking.BookingNumber
ORI = BookingORI
FBI = Inmate.FBINumber
SID = Inmate.StateID.Number
SOC = Inmate.SSN
POB = Inmate.CityOfBirth
DOB = Inmate.DateOfBirth
SEX = Inmate.Gender
RAC = Inmate.Race
HGT = Inmate.Height
WGT = Inmate.Weight
EYE = Inmate.EyeColor
HAI = Inmate.HairColor
CTZ = Inmate.Citizenship
OCP = InmateOccupation
SKN = Inmate.Complexion
SMT = Inmate.Scar.0
```

```
IDK_IFMFt="ILI_NAF"
NAM = Wardin, Isaa
AKA = Jimmy,
BKN = 2020-00000001
ORI = BookingORI
FBI = 12345
SID = 54322
SOC = 215-45-8749
POB = Tampa
DOB = 03/26/1991
SEX = M
RAC = W
HGT = 5'9"
WGT = 190
EYE = BRO
HAI = BLK
CTZ = US
OCP = InmateOccupation
SKN = FA
SMT = MC,ACNE
```

Summary

- Utilizes SnapLogic to create a modular and easy to understand pipeline
- Parses Tyler data and maps it onto the custom template provided by the customer
- Is easily adaptable to any data format required in the future without the need for a programmer to rewrite code

Tyler SnapLogic Integration Framework - T3

George Noga, gsn160030@utdallas.edu; **Khalid Abusultan**, khalid.abusultan@utdallas.edu; **Rayan Imam**, rayan.imam@utdallas.edu; **Trent Gibson**, tag160430@utdallas.edu; **Vishal Singh**, vishaldeep.singh@utdallas.edu

CS 4485 / Spring 2021
Department of Computer Science
Erik Jonsson School of Engineering & Computer Science
The University of Texas at Dallas
Richardson, TX 75080, USA

Abstract:

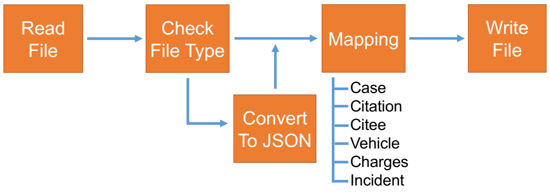
Our team's project for Tyler Technologies was to create a framework to transfer data from multiple different court citation systems. This will enable non-technically oriented people to leverage our solution to transfer said data in an easy to understand and time sensitive manner using the SnapLogic technology framework. We leveraged our individual software development strengths and skills to uniquely solve the problems presented using the SnapLogic platform in an optimal manner. All project goals were met including parsing XML and CSV data for court citations into JSON, formatting JSON data and mapping it across various fields which are pertinent to end users dealing with court citations, and the goal of outputting this data into an XML file which matches a given XSD format. Through use of JavaScript and a unique and flexible use of SnapLogic Snaps to create a versatile pipeline, our team has implemented a functional framework that Tyler Technologies can use in future implementations that use SnapLogic technology.

Keywords: SnapLogic, JSON, XML, XSD, CSV, Citations, JavaScript

Performance:

- All tasks for all bi-weekly agile sprints were completed on time
- All project goals were met on time

Architecture:



Impact:

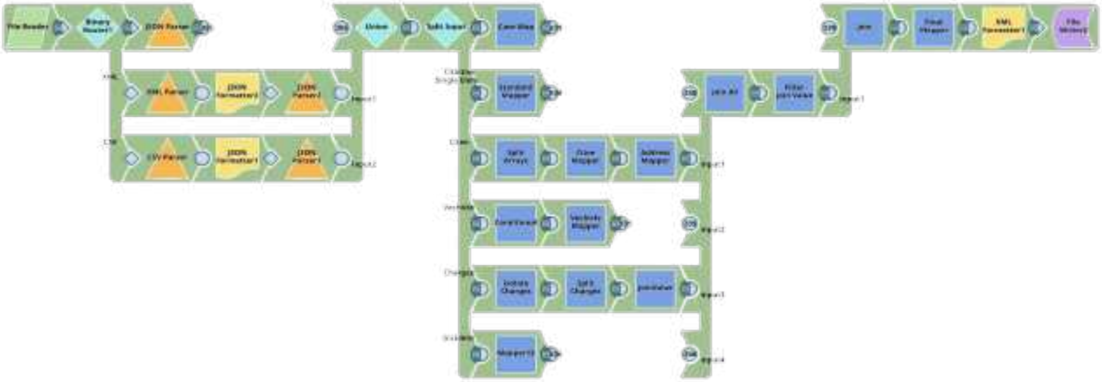
- Efficient citation standardization
- Easy future expandability
- Reduction of entry knowledge
- Easier file Ingress

Summary:

The Final SnapLogic Pipeline Has:

- Adaptable robust mapping
- Effective citation transfer
- Handles complex nested JSON arrays
- Easy handling for anyone non-technical
- Supports input flexibly with JSON, XML, and CSV citation formats

Final Pipeline:



Tyler Technologies

SnapLogic Integration Framework - T4

Kent Templin, kent.templin@utdallas.edu; Rajesh Nallamottu, rajesh.nallamottu@utdallas.edu;
Ambika Bastola, ambika.bastola@utdallas.edu; Alphonsus Nguyen, alphonsus.nguyen@utdallas.edu;
Phuc Duong, phuc.duong@utdallas.edu

CS 4485 / Spring 2021
Department of Computer Science
Erik Jonsson School of Engineering & Computer Science
The University of Texas at Dallas
Richardson, TX 75080, USA

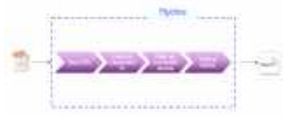
Abstract

To improve the efficiency and the overall amount of time it takes for Tyler Technologies to export files to Collection Agencies, our team has created a pipeline that simulates data integration for collection agencies. The pipeline takes in the debtor information (csv file) and converts it to JSON and XML file format or keep it in a CSV format. The pipeline also receives a list agencies to give the debtor information. The pipeline validates the debtor information, checking for any missing entries and invalid values. The pipeline then assigns cases for each agency, that is present in the agency list. After the assignment of cases to agencies, a file is created for each agency, which contains the debtor information assigned to the agency. The file format of the file is determined by the agencies preferred file format.

Keywords: Data flow, csv, json, xml, pipeline, Collection Agencies

Architecture

SnapLogic has an Elastic Integration Platform or Intelligent Integration Platform, which can deal with complex use cases and allows for high development productivity. The platform works to integrate applications and data with snaps.
We used the software SnapLogic to build a pipeline that controls the data flow.
The advantage to SnapLogic is that implementation of coding is 80% faster than manual coding.
Pipeline are made by sequence of snaps; snap is an object that introspects the source and target and performs a single complete function, such as read, write, or act on data.



Impact

This pipeline will work to improve the reliability and efficiency of integrations between Tyler Technologies and their respective clients. The framework that was established will help to save Tyler Technologies thousands of hours and will no longer interfere with the scheduled project development.

Performance

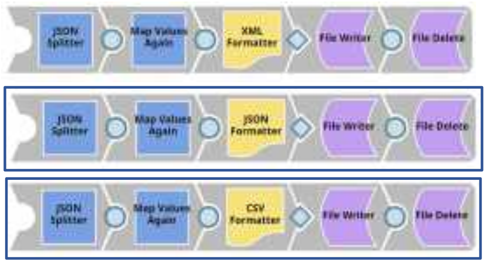
The performance is fast. Only takes less than 5 seconds to execute a csv file that has approximately 175,000 rows of data. The information is split among N number of agencies which is read from a list of agencies. The output files will take the agencies names as the file names, and the file format will be set according to the agency's preference.

Results

Our pipeline is capable to taking in the debtor information (csv file) that has approximately 175,000 rows.
The pipeline successfully converts it to json and xml file formats. Then the pipeline will write the case information to a file which designated with the Agency's name

Pipeline:

Sub-pipelines:



Summary

- We used the software SnapLogic to build a pipeline that controls the data flow.
- We established a framework which Tyler Technologies will be able to improve the efficiency of project development.
- The pipeline we made takes in the debtor information (csv file) and converts it to json and xml file formats.
- The pipeline can split information and assign pieces of information to different agencies.

UTD Explore Apollo

“That’s one small step for man, one giant leap for mankind”

Tanvir Sarker – Tanvir.Sarker@UTDallas.edu
Kha Nguyen – Kha.Nguyen@utdallas.edu
Sthitadhi Sengupta - sthitadhi.sengupta@utdallas.edu
Rohan Manda – Rohan.Manda@utdallas.edu

CS 4485 / Spring 2021
Department of Computer Science
Erik Jonsson School of Engineering & Computer Science
The University of Texas at Dallas
Richardson, TX 75080, USA

Sponsor: Dr. John Hansen
Project Staff: Meena Mudha Chandra Shekhar

1. Abstract

- **Goal:** Add ability to play multiple channels for website "ExploreApollo"
- **Our solution:**
  - Use current audio player module with adjustments to support playing up to three channels at the same time
  - Optimize UI and data management to improve performance
  - Add more data models to postgres and routes to Ruby API

Keywords : Full-Stack Development , Database , Web-Application

2. Architecture

- **Client Applications:**

React + HTML + CSS + JS.

- **Web Application:**

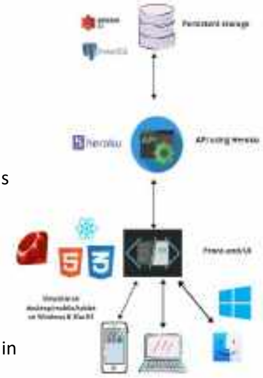
Responsive web application supporting most devices that have capabilities to access browsers and supported JavaScript plugins.

- **RESTful API:**

Backend API built using Ruby on Rails and hosted on Heroku

- **PostgreSQL Database:**

PostgreSQL provides persistent storage.
Backend API communicates with database.
Database communicates with audio in S3 buckets of AWS.



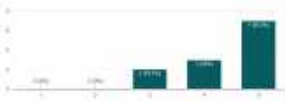
3. Impact

- Creation of interface to play multiple channels at the same time
- Playback of audios from multiple missions, rather than just a single channel of a single mission

4. Performance

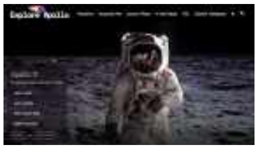
UTD CRSS Explore Apollo Website Survey

- Test suite run on code to ensure implementation accuracy.
- “Submit Feedback” form used to measure user satisfaction
  - Average of 4.4/5 stars from 12 respondents

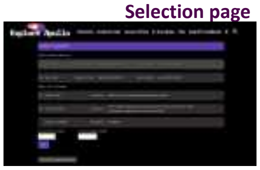


5. Results

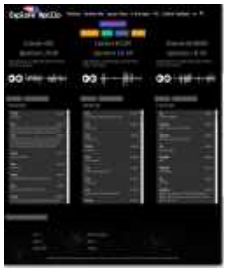
Explore Apollo Homepage



Channels Selection page

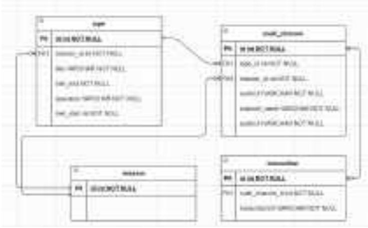


Multiple Channel Playbacks Page



Visit the Explore Apollo website using this QR code or go to app.exploreapollo.org!

Schema



6. Summary

- Capable of adding and playing multiple channels for Apollo 11 mission, along with the potentiality of adding audio and navigation for future/new missions
- **Tasks Accomplished:**

1. Added data models to postgreSQL
2. Added resources and routes to Ruby API
3. Resolved issue on Heroku which makes the web application couldn't connect to API
4. Added multi-channel playback from Apollo 11 mission audio
5. Created navigation between tapes and selection of channels
6. Scalable solution for future additions of missions

GUI for Drive Partition Recovery System

Puya Gharahi, Tushitha Jujjuri, Noah Monaghan, Felicity Pawlowski

Puya.Gharahi@utdallas.edu, txj170000@utdallas.edu,
Noah.Monaghan@UTDallas.edu, felicity.pawlowski@utdallas.edu

CS 4485 / Spring 2021
Department of Computer Science
Erik Jonsson School of Engineering & Computer Science
The University of Texas at Dallas
Richardson, TX 75080, USA

Abstract

This project aims to provide a graphical user interface (GUI) that allows users to visually access and navigate an existing drive partition recovery system. The existing system helps recover files and directories that have been either intentionally or unintentionally deleted. Applications, among others, include recovering deleted files, corrupted files, or aiding in law enforcement investigations which seek to recover files and directories that have been previously deleted by potential criminals. With this GUI, the recovery process for Linux and NTFS partitions will be accessible to all users through simple and intuitive design. The GUI will be implemented using PyQt5.

Keywords: Data Recovery, Linux, UI/UX

Architecture

Front-End
The front-end was built using the PyQt library and Python programming language.

Back-End
The back-end was implemented using the C programming language and contains all the file recovery processing.

Platform
Our GUI was developed for the Linux operating system and its various distributions.

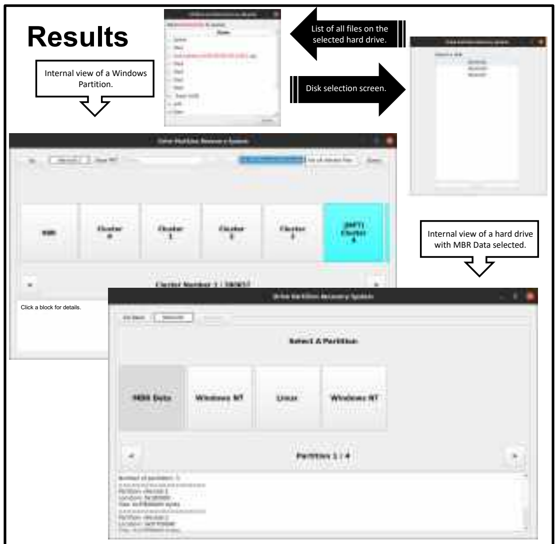
Impact

This project provides a user-friendly GUI that displays low-level storage structures of Linux and Windows partitions that can easily be navigated.

Additionally, this GUI allows users to list and select permanently deleted files and directories which can then be recovered to an external location.

The creation of this GUI was a critical step in empowering non-experts with the ability to perform low-level storage analysis and recovery operations. Potential users include digital forensics specialists, law enforcement, and general users who are seeking to recover lost files.

Results


Internal view of a Windows Partition.

List of all files on the selected hard drive.

Disk selection screen.

Internal view of a hard drive with MBR Data selected.

Performance Metric

- Front-end can load millions of internal file storage structures without making the user wait.
- Able to instantly search anywhere within a disk partition.
- All functional requirements were met for Windows file recovery.
- Fast response times with low latency for back-end API calls.

Summary

The GUI,

- Provides key information needed to recover files
- Provides the whole view of the internal structure of the drives.
- Includes dynamic scrolling and go-to feature to improve usability
- Displays billions of blocks without compromising the speed of the application
- Uses PyQt5 cross-platform GUI library that can be used on both Windows and Linux OS.
- Interacts with the backend written in C.

Personalized Electronic Photo Album UTD-CS

Alexander Sadowski, ajs170230@utdallas.edu; Grayson Smith, ges160030@utdallas.edu; Brendan Lesperance, bal180000@utdallas.edu; Rizwan Akhwand, rxa122230@utdallas.edu; Ty Phan, ttp180001@utdallas.edu

CS 4485 / Spring 2021
Department of Computer Science
Erik Jonsson School of Engineering & Computer Science
The University of Texas at Dallas
Richardson, TX 75080, USA

Abstract

To save the amount of time it takes a user to search through a large quantity of photos to pick out the best ones for a photo album, we created a program to streamline this process. Our program provides an easy to use experience where a user will upload a folder full of photos to a database. Based on the quality of the photos and tags applied to categorize them, the user can efficiently choose which photos they want and build an album.

Keywords: Python Database, User Interface, Data Analysis

Architecture

For our architecture we created a Python GUI that runs off of a Python Database.

Python GUI:

- Tkinter GUI package
- PIL (Python Image Library) to process images

Python Database

- Brisque Image Quality Assessment to give a mathematical value for an images quality
- SQLite for a local database solution.

Impact

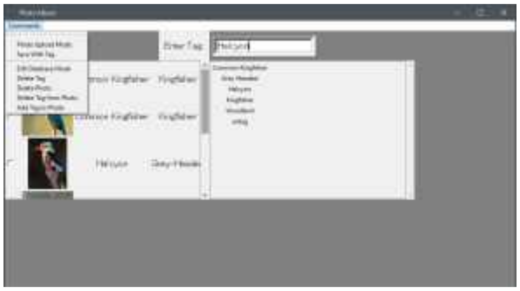
Our project can be used by UTD or other users for events where a large quantity of photos are taken to go through them more efficiently and effectively and thus save time on a previously time consuming task

Performance

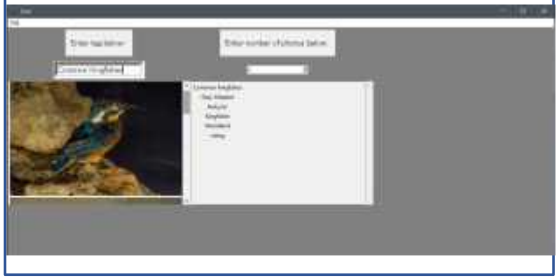
- We have a decently functioning final product
- Our sponsor approves of our product and is pleased with our work
- Most of the goals that were set by our sponsor and ourselves were met for this semester

Results

Admin Screen:



User Screen:



Summary

- We created a personalized photo album creation tool
- This tool takes an input of photos and gives the user the ability to search through them in a more efficient manner
- Uses an embedded tag system to assign tags to photos so that a group of photos can be searched for based on a tag
- Uses an image quality assessment library to further narrow down possible photo options for the user
- Provides an overall user friendly experience that makes generating a photo album a quick and easy task

Degree Audit Tool Phase II

UT Design

Eliseo Robles, Noel Romero, Ram Panda, Noman Syed
eliseo.robleslopez@utdallas.edu , noel.romero@utdallas.edu, ram.panda@utdallas.edu, noman.syed@utdallas.edu

CS 4485 / Spring 2021
Department of Computer Science
Erik Jonsson School of Engineering & Computer Science
The University of Texas at Dallas
Richardson, TX 75080, USA

Abstract

The ECS degree audit tool will simplify students' degree plans and allow them to visualize their progress and their intended course plan until graduation. The tool is accessible for new freshmen, transfer students, and current students. It is intended to reduce academic advisors' workload and provide students' an approachable and intuitive format to viewing degree plans. Freshmen & Prospective computer science students can choose a default degree plan to guide them on planning their future courses. Students will have their credits loaded automatically by uploading their unofficial pdf transcripts from galaxy. Students can move around courses and customize their degree plan, being warned when prerequisite constraints will not allow them to take certain courses. Finally, students and advisors can share their unique link to their advisor or other students in order to make collaboration on their degree plan easier.

Keywords: PHP, React, UI/UX, Laravel

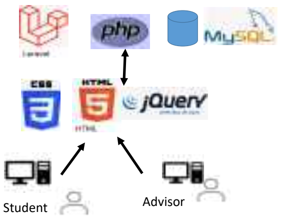
Architecture

Web Browser Application

- JQuery
- CSS
- HTML

Backend and Database

- MySQL
- PHPMyAdmin
- PHP
- Laravel



Impact

The Degree Audit Tool exists to improve the course planning experience by allowing students to share their entire planned course progression with their advisor.

Before

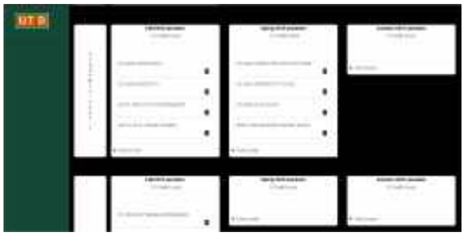
Students could only see course progression on paper with a confusing degree plan flow chart. The process was perplexing for students and put excess work on advisors

After

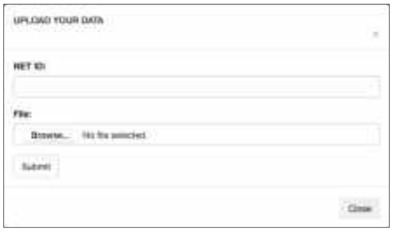
Students can see their own course plan in an organized and easy to interpret chart format. They can plan future courses in any way they please, being warned of prerequisite limits. The new system has special functionality for new and Transfer students.

Results

Course Dashboard



Transcript Upload



UPLOAD YOUR DATA

NET ID:

File:

Browse... No file selected.

Submit

Close

Performance Metrics

- Parser works for Transfer and Non Transfer Students reliably
- Company mentor feedback: Great time management and steady progress made throughout the semester.
- All major functional expectations were met.
- Code snippets and instructions written to the entirety for future team to add other degrees

Summary

- Created web application created to simply the degree plan process
- Friendly and intuitive User Interface builds a degree view that is easy on the eyes
- Students can upload unofficial transcripts to application to populate it
- Students can make changes to their degree plan and collaborate with advisors and peers
- Student's information is associated with netid and maintained throughout their college experience

JSOM: Analytical Hierarchy Process Tool Phase 2

Benjamin Goff, Austin McGuire, Srihaas Saripalli, Emory Soper

benjamin.goff@utdallas.edu, austin.mcguire@utdallas.edu, srihaas.saripalli@utdallas.edu, emory.soper@utdallas.edu

CS 4485 – Spring 2021
Department of Computer Science
Erik Jonsson School of Engineering & Computer Science
The University of Texas at Dallas
Richardson, TX 75080, USA

Abstract

The purpose of this project is to develop a website that implements the Analytical Hierarchy Process, a technique used to objectively make complex decisions. Our team iterated on the previous team's work by adding more features and overhauling the user interface. During this phase of the project, the website became more accessible and easier to use. The website is hosted using Amazon Web Services with a serverless web stack. The tool is deployed on UTD servers so that it can be used by students and staff at the University of Texas at Dallas's Naveen Jindal School of Management (JSOM).

Keywords: AHP, AWS, UI/UX

Architecture

Front end:
- AWS S3
- Vue.js
- Vuetify framework
- Serverless Stack

Back end:
- DynamoDB
- AWS Lambda
- Python

Impact

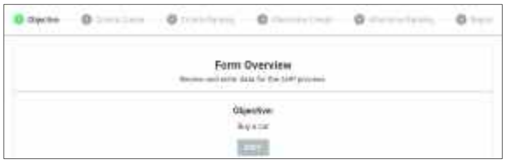
Previous Website:
- Focused on the functionality of multiple users working on the AHP process together.
- User experience required an excessive amount of page navigation and sending links to work well.

Updated Website:
- Focused on improving the UI and UX.
- Each step is streamlined and explained.
- All users work together concurrently.
- Support for entering quantitative data for the ranking step of the algorithm

Performance

- Weekly sponsor meetings to test and showcase new features and receive feedback.
- Site deployed using AWS on JSOM servers
- Site is able to scale to meet demand of millions of concurrent users.

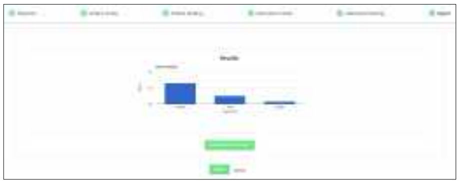
Results


Form Overview

Objective:

JSOM: Analytical Hierarchy Process
Admin Form

JSOM: Analytical Hierarchy Process
Response Form



Summary

- Improved upon the previous team's website.
- Made the AHP process accessible and easy to use.
- Worked with sponsor to ensure the user experience was excellent for new users.

Air Quality Live Website

UTD MINTS

Veronica Ramirez, vir170130@utdallas.edu; Fawaz Khurram, fxk170000@utdallas.edu; Kevin Flores, kef170130@utdallas.edu; Daniel Yustana, dsy170030@utdallas.edu; Sidney Evans, sce190001@utdallas.edu

CS 4485 / Spring 2021
Department of Computer Science
Erik Jonsson School of Engineering & Computer Science
The University of Texas at Dallas
Richardson, TX 75080, USA

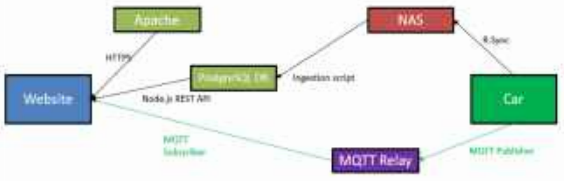
Abstract

The UTD MINTS Air Quality Live Website displays current air quality conditions around the Dallas area. The data is being collected by a vehicle that records air quality as it drives around Dallas. There are two main objectives in building and improving the website. Firstly, through user interface changes that make it easier to access information. This will be done through aesthetic improvements, bug fixes, and adding new features that make the website straightforward and simple to use. The second objective is to map historical data on the website through the use of data we have already gathered from the sensor and adding an option to overlay data from a certain time range.The ultimate goal is to create a fully furnished website that users can use to easily access air quality data.

Keywords:

Web development **user interface**
Front-end development **Node.js**

Architecture



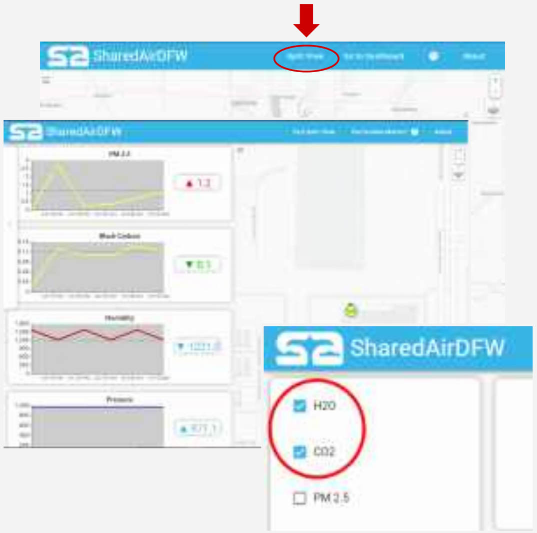
Impact

The changes to the front end of the MINTS Air Quality Live Website allows for easier consumption of information through the split screen feature as well as adding more information such as the H2O and CO2 levels in the dashboard.

Features added:

- Improved dashboard arrow color, color now turns green when values go down and red when values go up. Done for particles that increase air quality when decreasing.
- Improved dashboard chart color by adding a gradient of colors allowing the chart colors to look more natural as the values change.
- Added split screen feature, allowing dashboard and map to be viewed at same time
- Obtained GPS data from multiple sensors, averaged the position of the vehicle, and then displayed this result at the map

Results



Performance

- Met our weekly goals of progress throughout the semester.
- Improved data quality through incorporating multiple sensors and averaging data.
- Improved user interface through a fully functional split screen feature between dashboard and map.
- Improved data readability through improvement of colors on the dashboard chart and arrows.
- Added more information to the dashboard not previously available.
- Completed all prioritized goals assigned during the project timeframe.

Summary

There are other websites which display air quality for regions in which the user lives, however, the advantage that the UTD MINTS Air Quality Website has is that the website reports close to current air quality statistics by the minute. In order to better achieve its goal, the website required new aesthetic and functional features. Some of those features (including but not limited to the splitscreen home/dashboard feature and additional dashboard charts) were implemented during the trajectory of this project while ensuring that the website behaved properly, and resolving existing issues as they appeared.

Project ADELE

Automated Documentation to Extract Learning from Electroencephalogram Data

Cristian Garces, cxg150730@utdallas.edu; Bradley Krakar, bjk170630@utdallas.edu;
Jesse Ladyman, jdl170130@utdallas.edu; Rami Jaber, rxj180009@utdallas.edu

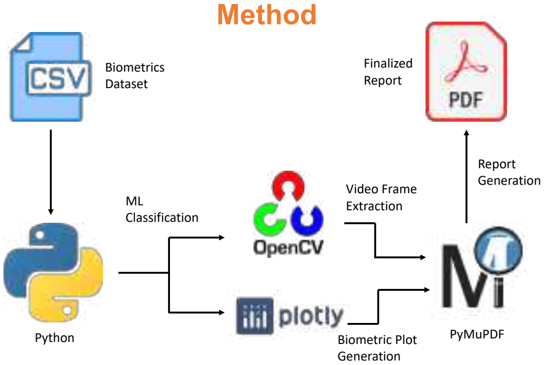
CS 4485 / Spring 2021
Department of Computer Science
Erik Jonsson School of Engineering & Computer Science
The University of Texas at Dallas
Richardson, TX 75080, USA

Abstract

Project ADELE provides a quick and fully automated solution for generating reports for biometric data analysis. Given EEG data and a set of other related biometric data, such as heart rate and pupil diameter, Project ADELE fits the data into discrete events with an unsupervised classification machine learning algorithm in Python. From this collection of events, ADELE generates a report on each event in PDF format, eliminating the need for manual data classification and report writing and formatting. The reports generated are designed to be user-friendly and readable, providing a clear overview of the data and increasing the efficiency of any research involving this data.

Keywords: Biometrics, Machine Learning, Automated Documentation

Method



Impact

- Increases efficiency by eliminating need for manual classification and documentation of EEG and related biometric data
- Generated reports offer researchers a clear overview of biometric data
- Machine learning classification provides researchers a starting point for further data analysis

Performance

- All functional requirements for Project ADELE were met
- Project divided into 3 phases – all phase goals were completed on time
- Accuracy of reports was manually assessed
- Autogenerated reports deemed clear and of high quality by sponsors

Results

Event: Scrolling Twitter

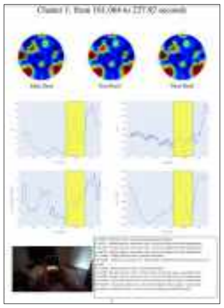
Clustering: KMeans, n_clusters = 5

UTD - [illegible]

Cristian Garces, Bradley Krakar, Jesse Ladyman, Rami Jaber

– Table of Contents –

[illegible]



Cluster 1: From 161.064 to 227.92 seconds

SAMPLE REPORT

Summary

- ML classification algorithm divides dataset into discrete events
- Information about each event is compiled into single page Event Reports
- Final autogenerated reports contain:
  - Title Page
  - Hyperlinked Table of Contents
  - Event Reports
- Event Reports contain:
  - EEG heatmap visualizations
  - Plots of related biometric data
  - Video frame from event
  - Summary of stimulated brain areas during event

Ochem Rank 2

Organic Chemistry Ranking Application

Developers: Zacharia Shoaib, Zacharia.Shoaib@utdallas.edu; Raheel Ahmed, Raheel.Ahmed@utdallas.edu; Narayana Yenukonda, Narayana.Yenukonda@utdallas.edu; Umar Kazi, Umar.Kazi@utdallas.edu; Saman Laleh, Saman.Laleh@utdallas.edu

Company Sponsors: Dr. Mihaela C. Stefan UTD Organic Chemistry Lab, Dr. Michael C. Biewer UTD Organic Chemistry Lab
Faculty Sponsor: Dr. Ovidiu Daescu. Special thanks to Justin Miller, Hanghang Wang, Sean Kennedy, Dr. Jey Veerasamy, and Dr. Miguel Razo-Razo

CS 4485 / Spring 2021
Department of Computer Science
Erik Jonsson School of Engineering & Computer Science
The University of Texas at Dallas
Richardson, TX 75080, USA

Abstract

Our project is a continuation of the development of the UTD Chemistry and Biochemistry Department's Organic Chemistry Ranking Application. Our objective is to refactor and to make the current development more efficient. We plan to further this project to help university students taking Organic Chemistry 1 or 2 from either UT Dallas or perhaps from other collegiate institutions. Our approach to this project is to delegate ourselves to two primary tasks: to ensure that we can add more problems to the set of questions already available and to improve the user interface and overall experience during the use of the web application. This project will allow UT Dallas to further promote student success within the scope of the Organic Chemistry Department.

Keywords: Android, iOS, UI/UX, Mobile App, MySQL, AWS, React

Architecture

Question Format

- Ranking questions consist of a question stem and 5 compounds to be ranked by some chemical property.
- Every question was categorized by class (Organic Chemistry I or II) and primary topic (acidity, basicity, resonance, nucleophilicity, etc.)
- There are 133 different questions to choose from with more being added (goal is 200 questions by the end of spring 2021).

Design Guidelines

- The app works on Android, iOS, PC, and MacOS devices.
- Portrait and landscape modes are supported on mobile devices.
- Hint pages are shown per category when the user enters an incorrect answer.

React

- React is an application-building library that utilizes a combination of JavaScript and CSS to create viewable, customizable components.
- JavaScript and HTML were used to determine what components should be displayed, and CSS was used to design those components (color, sizing, position, etc.)

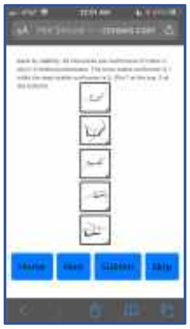


Figure 1: Question Overview in Safari

Impact

- This project is and will continue to be a free website that comes with user friendly features to help anyone learn the subject of Organic Chemistry (Ochem). Whether someone is starting out as a beginner, or they think they're an expert, this will help them test any and most Ochem compound ranking questions taught globally no matter where they are.
- Previously students had to physically access the questions from the professors and search through physical pages; this application allows students to digitally access the questions whenever and wherever they desire and quickly input their needed question parameters so they can study more frequently and more efficiently.
- The design of the application and data base allows for question set expansion, as well as immediate deployment of new questions to students. This allows the professors to easily adapt to any deficits in the curriculum and have more immediate impact on the ability of their students to prepare for exams.

Results

- With more of the back-end and front-end implementation already synchronized by the original creators, we used more of our time optimizing the UI/UX for a better user experience. This allowed us to add new features including:
- a smooth image-blow-up feature which would allow a user to easily enlarge a compound and move it to the side by clicking and holding it
- a "Try again" button when seeing the solution to allow the user to seamlessly remaster the question in that moment.
- 10+ questions to help master the concepts from different angles

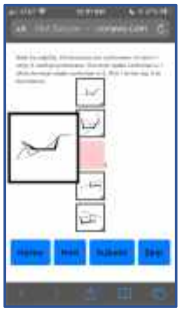


Figure 2: Question Overview with mouse-click feature in Safari

Figure 3: Question solution view in Safari

Figure 4: "Try again" button feature in Safari

Performance

- The app meets 100% of our clients' criteria, with low API latency and high accuracy/precision.
- Simple, intuitive interface is aesthetically pleasing and accessible for average user.
- Exam scores of OChem Students can be taken before and after using this app to measure impact.
- Feedback page was implemented for user suggestions/critique
- Consistent performance across Android, iOS, and PC/Mac (Web) devices.

Summary

- OChemrank.com is a website that can be used for chemistry students to practice organic chemistry problems. The objective of the project is to improve the existing Organic Chemistry ranking website.
- Questions are categorized by topic and semester; randomizing enhances retention.
- Easy-to-use UI across multiple devices makes it an accessible learning tool.
- Lowers student use of outdated and unreliable Organic Chemistry resources.
- Convenient storage of images in SVG format allows simple translation from ChemDraw to database.
- Flexible database and infrastructure allows for long-term support and improvements.

Home Sleep Quality Monitoring

Divya Gubba, dsg170130@utdallas.edu; Raashi Kulshrestha, rxk170530@utdallas.edu; Jonathan Ho, jyh170130@utdallas.edu; Nicolas Gover, nsg170030@utdallas.edu; Patrick Rustandi, pnr180000@utdallas.edu

CS 4485 / Spring 2021
Department of Computer Science
Erik Jonsson School of Engineering & Computer Science
The University of Texas at Dallas
Richardson, TX 75080, USA

Abstract

Sleep apnea is one of the most common sleep disorders and is a condition marked by abnormal breathing during sleep. Sleep apnea is said to affect 2 – 9% of the population and if left undiagnosed and untreated, it can increase the risk of other health problems, but it can be monitored and measured.
Our goal is to create an application using a wearable ECG device, placed on the user's chest, which will be used to monitor and measure sleep apnea. The mobile app will read the ECG signal via Bluetooth and be responsible for transmitting the information to the cloud for storage to then be processed by an algorithm. This processed information will then be able to be viewed by the users as a report in their iPhone application once they have created their user login and have connected their ECG device to the iPhone.

Keywords: Sleep Apnea, Apnea-Hypopnea Index (AHI), Electrocardiogram Device (ECG), Bluetooth, Azure IOT, Azure SQL, Expo Mobile App

Architecture

ECG Device
- VivaLNK

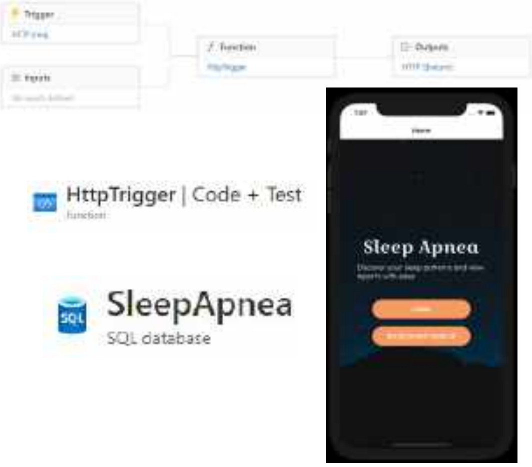
Mobile Application
- Expo
- Android Studio

Backend and Database
- Azure
  - Azure Function
  - Azure SQL Database

Performance

Final product has all desired features implemented.
Azure functions were tested thoroughly
Speed performance was at an acceptable level
Data was not lost in transactions with the database

Results



Impact

This project aims to simplify the process of measuring one's AHI.
Previously, even the most simple sleep tests use invasive tubes and wiring
With this new application, the user only needs to wear a heart rate monitor while they sleep, and still find their AHI with an acceptable degree of accuracy.

Summary

- Created a user-friendly mobile application
  - Connects to ECG device through Bluetooth
  - Connects to Azure Web Services through the internet
- Implemented Azure Web Services

Indoor Air Monitoring

Android App Utilizing Machine Learning Model and IoT Devices

Alec Borman, jjc171230@utdallas.edu; Hayden Iske, hji170000@utdallas.edu; Dang Truong, dmt180000@utdallas.edu; Jay Castro, jjc171230@utdallas.edu; Angel Torres, axt190026@utdallas.edu

CS 4485 / Spring 2021
Department of Computer Science
Erik Jonsson School of Engineering & Computer Science
The University of Texas at Dallas
Richardson, TX 75080, USA

Abstract

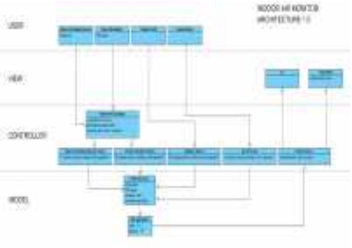
Improved an Android application that takes information from the user's devices, including an indoor air monitoring device and peak flow meter, and local weather data and alerts the user to when he/she will fall sick and need to take medicine using a machine learning model.The machine learning model relies on data from two IoT devices, user-answered questions via mobile app, and local weather data gathered via a web API. The user's data is backed into a cloud-based server for the model to access and then improve based on previously asked questions.

Keywords: Mobile App, Cloud-Based, Machine Learning

Architecture

Core

The core of the app was built using a Model-View-ViewModel (MVVM) interface. With this design, the business model of the application is split from the presentation, maintaining clean separation between application logic.



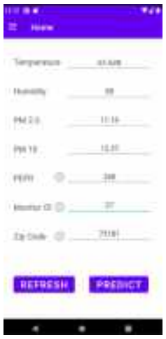
Software

- Android Studio
  - Kotlin / Gradle
- Cloud Server
- Purple Air API
- Amazon AWS
- DynamoDB
- Amazon REST API
- Peak Flow Meter

Impact

- User can connect their devices to the app and receive predictions on when to take their medication to prevent health problems
- Model uses predictive learning to continuously improve behavior and become more efficient and handling illness flare-ups
- Full backend implementation and integration with android mobile applications

Results



The user enters values for PEFR, Monitor ID, and Zip Code. Refreshing the app updates the values.

New Settings page where the user can save their Monitor ID to local storage.

Performance

- Polling new data to the app refreshes the app once every 20 seconds
- Objective of updating the app with latest API's and new features was accomplished
- Company mentor feedback: Met the requirements and was pleased with the the new additions and improvements of the app
- Weekly task completion rate: 90 %
  - Some features were not completely implemented

Summary

- Refurbished MVVM to improve design and overall clean architecture for the app which helps developers build onto the app with ease
- Demonstrated a functional UI portion for the app which shows relevant data needed and retrieved from the device and model (PM2.5, PM10, etc.) as well as the questionnaire
- Updated Purple Air API to have a functional device that measures air quality,temperature, and humidity in an environment
- Created AWS Database and REST API for App Use
- Refactored using RetroFit2
- Integrated Machine Learning Model into the app for predictive learning

Open Source Event Manager for ECS Student Organizations

CS 4485 / Spring 2021
Department of Computer Science
Erik Jonsson School of Engineering & Computer Science
The University of Texas at Dallas
Richardson, TX 75080, USA

Mustafa Sadriwala mustafa.sadriwala@utdallas.edu
Shannen Barrameda sib170130@utdallas.edu
Siddartha Devic Sid.devic@utdallas.edu
Siddharth Naik Siddharth.naik1@utdallas.edu
Jiwon Topper Jiwon.topper@utdallas.edu

Abstract

Key Words: React, Web Development, C#

ECS UTD Events (available at https://ecs-utdevents.web.app/) is a web application that centralizes all information about events hosted by ECS student organizations. Representatives of ECS organizations upload event information onto ECS UTD Events, and all students can easily preview upcoming ones from daily, weekly, and monthly views. Previously, students relied on Mr. Jerry Alexander's (ECS Assistant Dean for Student Development) weekly emails and UTD applications like OrgSync / Presence, which become increasingly difficult to keep track of. ECS UTD Events aims to promote larger turnout and reduce planning conflicts by centralizing event marketing. Our project, like UTD Grades, is open source and will facilitate code contributions from the broader UTD community.

Introduction

- The leaders of the 25+ ECS student organizations have always had a difficult time to both plan non-conflicting events and garner large turnouts.
- Each student organization has their own separate social media accounts, discord servers, and slack channels for advertising and marketing.
- For a student -- especially one who is new to UTD -- this can easily become a lot to keep track of.

Organizations
Women Who Compute
Artificial Intelligence Society
Association for Computing Machinery

- SOC verified organizations are eligible to register on our site and become listed on our home page, as well as begin posting events.

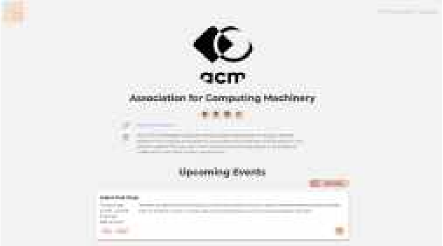
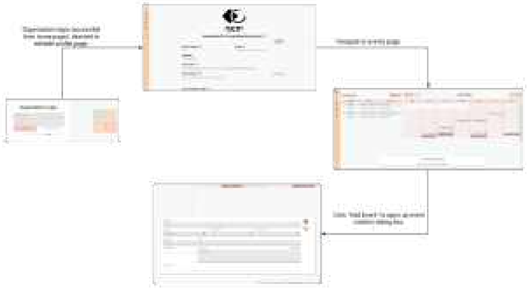


Figure 3: Organization profile page accessible from the organization list on the home page. Organizations may modify this to list their different social media sites, and all their upcoming and past events are automatically listed.

Workflow for Adding an Event

- Organizations add events to their profile page through streamlined user interface.
- Importantly, events can be assigned "collaborators", which are simply other organizations signed up on the site.
- Collaborators have full access to edit events, encouraging collaboration.
- Events can also be assigned "tags" which provide tidbits of information describing an event.
  - For example, the "industry" tag can be used for industry recruitment events, or the "womxn" tag for events focused on inclusion and diversity.
- On the events page, ALL events from ALL organizations are visible. This helps minimize event overlap / schedule conflicts!



Contact and Acknowledgments

Contact: utdecsevents@gmail.com.

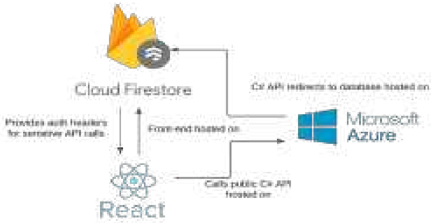
We would like to thank Dr. Miguel Razo for advice and approval of this project. We also wish to acknowledge the amazing work over the past decade of Mr. Jerry Alexander, who is retiring this year. Thank you so much for everything you have done for the ECS community Mr. Alexander!

Software Architecture

Front end
- React web development framework [1]; CSS.

Backend
- C# API querying a database hosted on Google Cloud Firestore [2].

Hosting
- Google Cloud Firestore hosting and database. C# backend API is hosted on Microsoft Azure web app hosting [3].


Cloud Firestore
C# API redirects to database hosted on
Provides auth headers for sensitive API calls
Front-end hosted on
Microsoft Azure
Calls public C# API hosted on
React

Performance Metrics

- Weekly task completion rate: 98%
  - All features for our MVP were met after only 1.5 months
  - 2/3 features from stretch goals were implemented
- Mentor feedback
  - MVP goals were met very early in production
  - Proud of work ethic and impact project had on ECS
  - Great UI with a modern twist
- Functional expectations
  - Completed functions have very little to no bugs
- Points of Contact (POC) Feedback
  - (Retrieved feedback from organization representatives every month)
  - Implemented primary features POC prioritized the most
  - Application
    - Visually appealing
    - Intuitive and easy to use

Impact

Public API
- Our production site has a public API: https://ecsutdevents.azurewebsites.net/.
- E.g. GET https://ecsutdevents.azurewebsites.net/api/events/all gets all events.
- Also open source: https://github.com/ecs-utd-events/ecs-utd-events-backend.
  - Sensitive API calls (e.g. deleting / adding an event) are restricted with an authentication header accessible only through our main site.
- Advantages of an open public-facing API:
  - if an organization lists all their events on our site, they could simply use our public API to populate their own website / calendar with exactly the same events. Changes need to only happen on our site and will be reflected everywhere the API is used!
  - A public API also facilitates the creation of additional software on top of ours. For example, a discord bot could be created which automatically retrieves all events for a particular organization and pings everybody in a channel once a week.

Open Source
- Application is open-source so anyone can contribute!
- This project has become a part of the Senior Design Project and will be maintained by future Senior Design students

Design
- Push for intuitive and accessible design with a modern twist
- Mobile support for most commonly used functions
- User-centered

UTDesign

Makerspace User Tracking

Alejandro Arredondo, axa117030@utdallas.edu; Sanya Macadam, szm170030@utdallas.edu;
Eliot Partridge, eliot.partridge@utdallas.edu; Chase Uphaus, cju170030@utdallas.edu

CS 4485 / Spring 2021
Department of Computer Science
Erik Jonsson School of Engineering & Computer Science
The University of Texas at Dallas
Richardson, TX 75080, USA

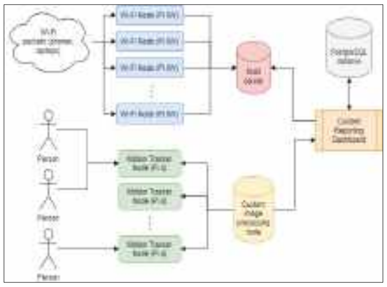
Abstract

A hybrid method of Wi-Fi nodes and motion sensors using multiple cameras to track users within UTD's Makerspace area. Tracking reports how many students are in Makerspace, as well as traffic surrounding equipment such as 3D printers. Motion sensing cameras will focus on how many people are in the space, and the Wi-Fi nodes will act as a heat map to track the movement of users. Combined, double counting is less likely while still giving a picture of movement from one area to another, allowing Makerspace to know how many students use their facility, what the peak times and days are, as well as which equipment is used most and how often.

Keywords: User Tracking, Motion Detecting, Wi-Fi nodes

Architecture

- Wi-Fi approach is detection of packets via devices like phones and laptops via Raspberry Pi 0W nodes pushed through find3
- Motion sensor approach detects people via cameras and Raspberry Pi 4 followed by image processing


Wi-Fi packets (phones, laptops)
Wi-Fi Node (Pi 0W)
Wi-Fi Node (Pi 0W)
Wi-Fi Node (Pi 0W)
Wi-Fi Node (Pi 0W)
find3 server
PostgreSQL instance
Custom Reporting Dashboard
Person
Person
Person
Motion Tracker Node (Pi 4)
Motion Tracker Node (Pi 4)
Motion Tracker Node (Pi 4)
Custom image processing tools

Impact

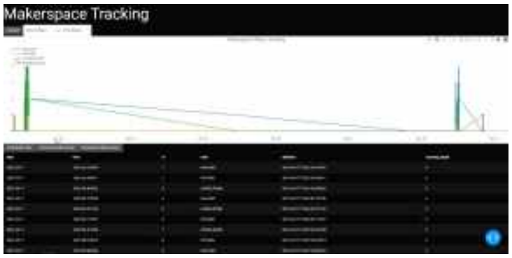
Allows Makerspace team to know:

- How many students generally use their facilities
- What the peak times and dates are for heavier traffic
- Know which equipment is most popular/used more frequently to help with ordering more

Performance

- Combined approach can help counteract double counting from students that carry more than one Wi-Fi enabled device
- Cameras can detect people with 50% confidence level
- Data from both sides is reported to the dashboard within 12 seconds

Results



Dashboard that reports traffic (test data used)

User Tracking Nodes up close and placed within Makerspace

Summary

- Hybrid approach can both count entry and exit but also act as a heat map to gauge popularity of certain equipment
- This project can be applied to places besides Makerspace with very minor adjustments
- Reports can be generated at varied intervals including daily, weekly, monthly, semesterly, etc.
- Future improvements can include a full interface for the reports, complete with charts and analysis to describe when traffic is highest and when it is down compared to previous months

UTD Tool Crib

Daniel Lopez (del170000@utdallas.edu); Gerald Torrijos (gxt170002@utdallas.edu);
Jacob Darin Glenny (jdg180001@utdallas.edu); Sudarshana Jagadeeshi (sxj180060@utdallas.edu);

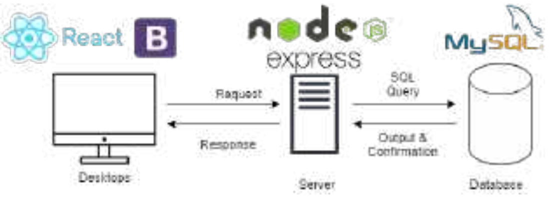
CS 4485 / Spring 2021
Department of Computer Science
Erik Jonsson School of Engineering & Computer Science
The University of Texas at Dallas
Richardson, TX 75080, USA

Abstract

The UTDesign Fabrication Shop wants to set up an inventory system for their items that they rent and sell to students needing supplies in their projects. This system will automate and improve efficiency of several tasks such as creating expense reports and handling overdue tools. We will accomplish our goal by creating a front end React based webapp that will run locally on their systems which will create query calls to their MySQL database via an Express/Node backend server. This project will not pose any costs and will not require meeting in person. The app will benefit the Fabrication Shop long term by having an efficient system to help students in their projects.

Keywords: UI/UX, CRUD, Web Application

Architecture



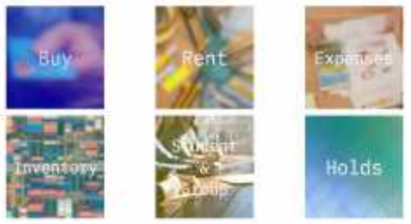
Impact

This project will help the UTD student body build and explore practical applications of their learned skills. In particular, the EPICS and Capstone senior students will be able to rent tools and purchase parts in a convenient way. The ability to manage the tools will prevent any one group from hogging resources, allowing equal opportunity for all interested. The ability to collect metrics about tool usage will help improve the quality of the inventory over time.

Performance

- On-time delivery: (# of Features Completed On-Time / Total # of Attempted Features) * 100
    - (8 features/10 attempted) * 100 = 80%
- Total commits:
    - Backend: 210, for an average of 15 commits/week
    - Frontend: 240, for an average of 17.1 commits/week
- Sponsor feedback was overall positive.

Results



Inventory Page

Student & Group

Summary

Our system was built using nodeJS, with React and Bootstrap in the frontend, and expressJS in the backend. The backend will query a mySQL database and return the appropriate information to be displayed. Features supported include renting, buying, expense reports, inventory management, and spreadsheet ingestion.

Our project will yield huge benefit to the Tool Crib as they continue to grow and attract more UTD students. The approach used will be more reliable and powerful than the previously-used spreadsheet approach. In addition, some additional features like tool statistics and expense reports can help not only Tool Crib workers, but management. They can now identify commonly rented tools, simplify billing, and gain insight into their busy weeks in a semester.

Overall, the project can be considered a success as the majority of features(80%) were delivered, and the remaining 20% were properly communicated to be unfeasible with our sponsor. While future groups certainly have plenty to improve upon, our final software is self-sufficient, finished and usable.

UTDesign

Training Verification System

Simeon Soname, Brandon Komplin, Wesley Ireland, Autumn Pin
sts170002@utdallas.edu, bjk170130@utdallas.edu, wti160030@utdallas,edu, abp170003@utdallas.edu

CS 4485 / Spring 2021
Department of Computer Science
Erik Jonsson School of Engineering & Computer Science
The University of Texas at Dallas
Richardson, TX 75080, USA

Abstract

The UTDesign Training Verification System is developed to keep track and CRUD the student database, and grant or deny access to equipment based on instruction level. This system allows administrators to know which students have the proper training for equipment use. Furthermore, the system provides administrators the ability to manage the shop effectively and efficiently. By developing a database, this system allows for logging-in, controlling access to the shop, and retrieving necessary data for each student in an efficient and timely manner.

Keywords: SpringBoot, Angular, Client-Server, REST, API, Database, Web-app

Architecture

Frontend:
- Spring boot

Backend and Database:
- Spring
- MySQL

Impact

The UTDesign Training Verification System is set up to provide a barrier between students and the heavy machinery within the makerspace. With the potential for injury, this application prevents untrained users from making an irreversible mistake within UTDesign's grounds.

Before: There was no solution for untrained and misinformed individuals from entering into the lab and working in the absence of a staff members strict supervision.

After: There is a tablet in the entranceway to the makerspace that students must log into, preventing the untrained or misinformed from entering into the lab

Performance

- The UTDesign Training Verification System Web Application meets the industry's performance standards due to its development method
- Cross-platform development is not a topic of this application, as the Browser-based interface is not platform exclusive
- Extremely fast response times due to its direct connection to the database
- Availability to have multiple connections to the same application

Results

User View

Admin View

Summary

- Gave UTD Fab Shop the ability to track student training levels automatically.
- Web-app architecture means that students don't have to download any additional software in order to access system.
- Student information, machines available for sign-in and use records are persisted by a database.
- Use records from the database can be downloaded into a report that displays how many students used each machine and how long machines were used in total.
- Use records are generated automatically from student sign-ins and sign-outs.
- Primarily visual tool that is designed to streamline the sign-in process as much as possible.

UTDesign Phone App

Cross-Platform Mobile Application Providing Access to Key UTDesign Resources

Sindhura Boppudi, sindhura.boppudi@utdallas.edu; Liora Davey, liora.davey@utdallas.edu; Mahimachand Nalluri, mahim@utdallas.edu;
Janrose Samson, janrose@utdallas.edu; Prathyusha Thiruvuri, prathyusha.thiruvuri@utdallas.edu

CS 4485 / Spring 2021
Department of Computer Science
Erik Jonsson School of Engineering & Computer Science
The University of Texas at Dallas
Richardson, TX 75080, USA

Abstract

The UTDesign phone application is a mobile provision of commonly used resources by UTDesign students. The purpose of this is to enable greater access and convenience to these student tools, with a goal of increasing productivity. The main tool we will be utilizing in the creation of this application will be Flutter, a software development kit from Google that is designed for cross-platform development. This allows our team to focus on writing robust and clean code while developing a software solution for Apple and Android devices, two of the most prevalent.

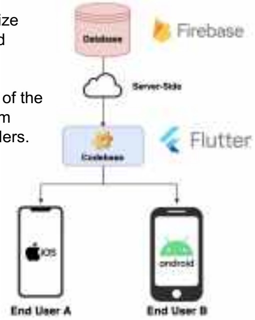
Keywords: Mobile App, UI/UX, Database Management

Architecture

The Flutter Software Development Kit (SDK) is a user interface-driven toolkit designed for building a robust and responsive application for a variety of operating systems. The main codebase was built using Flutter, specifically with Dart as the programming language building block for the application.

On the server-side, the team chose to utilize Firebase: a modern and powerful backend service by Google with a powerful Flutter integration. Firebase features a database platform for the Android and iOS versions of the application, with appropriate cross-platform functions such as notifications and reminders.

The FlutterFire integration comprises the entirety of the UTDesign phone application and enables parallel experiences to iOS and Android users. Modifications to the codebase are reflected in both end users, with a two-part database that contains the application data.



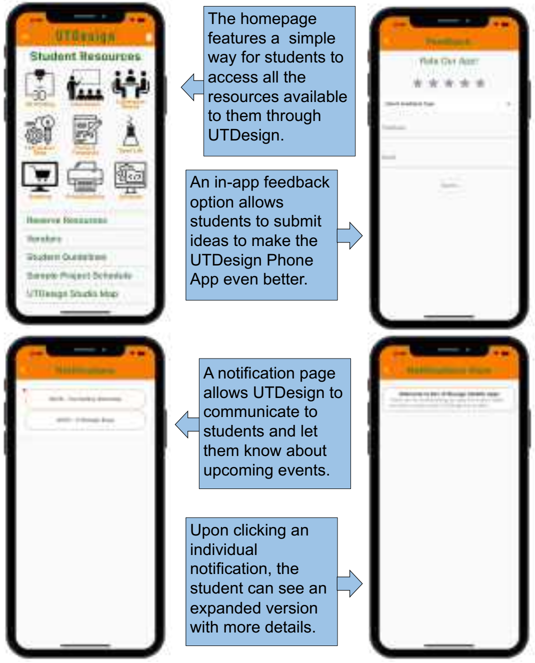
Impact

The overall impact of the application was the successful creation of a centralized hub of commonly used resources and services for UTDesign students. This features a main catalog of resources, reservation options, guidelines and policies that are relevant to students. In addition to this static component, students are provided with a dynamic notification function that will consistently send out alerts to upcoming workshops and UTDesign events, along with commonly needed reminders.

Performance

The reception of the application was generally positive based on results from a survey released after app deployment. These results were compared to an initial survey about the website. Many students find it to be a vast improvement in terms of convenience compared with the website, which was not designed with the mobile user in mind. The ease of access in having a portal to common reservations was appreciated by students, a feature that will increase as the campus resumes in-person operations after the pandemic.

Results



The homepage features a simple way for students to access all the resources available to them through UTDesign.

An in-app feedback option allows students to submit ideas to make the UTDesign Phone App even better.

A notification page allows UTDesign to communicate to students and let them know about upcoming events.

Upon clicking an individual notification, the student can see an expanded version with more details.

Summary

We created a mobile app that allows students to access the resources they need to be able to utilize UTDesign to its fullest. Flutter and Dart allowed us to make a cross-platform app using just one codebase, which let us focus more on features rather than having to recreate the app for Android and iOS. Firebase provides a great way for administrators to send push notifications and receive feedback about the app's functions.
The FlutterFire integration is simple and flexible so that administrators can easily learn and understand how to access and modify application data and deploy notifications.

Acknowledgements

The UTDesign Phone App Team would like to extend our utmost gratitude to Don Proeschel and Wayne Peterson for their continuous encouragement and guidance.

UTDesign Procurement Manager: Improvements Team

Olaoluwa Babatunde, oeb160030@utdallas.edu, Adam Lipscomb, adl180002@utdallas.edu, Kenny Huynh, kmh170230@utdallas.edu, Christopher Parekh, csp160030@utdallas.edu, Kumuda Aggarwal, kxa170630@utdallas.edu , Akash Prakash, axp161831@utdallas.edu

CS 4485 / Spring 2021
Department of Computer Science
Erik Jonsson School of Engineering & Computer Science
The University of Texas at Dallas
Richardson, TX 75080, USA

Abstract

The UTDesign Procurement Manager is an application designed to assist UTDesign and EPICS with their most important function: procurement. The application aims to make the procurement process easy for students and admins to work with and view the needed information. The additions of filtering and sorting will make finding relevant information easier, opening/closing all detail for each procurement gives the table a cleaner look, alongside other updates, will help to solve some of the issues that people who were in direct contact with the application had exhibited previously. The goal of the improvements team is to reduce anything that may lead to an inefficient experience for the user.

Keywords: Procurement, UI/UX, UTDesign

Architecture

Front-End
- React
- Material-UI

Back-End
- Axios
- REST API

Database
- MYSQL
- MariaDB
- Nginx

Impact

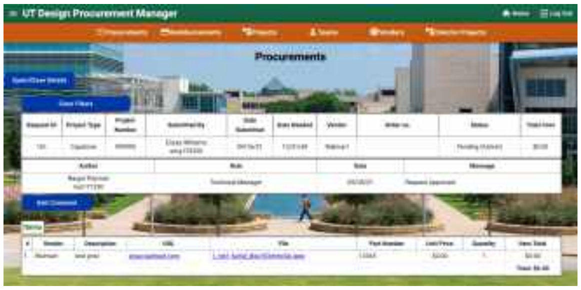
The UTDesign Procurement Tool aims to provide users (Admins, Technical Managers, and students) an easy and intuitive experience regarding the procurement process. Our work this semester added much needed improvements to the overall system that will lay the foundation for future improvements for the upcoming future teams.

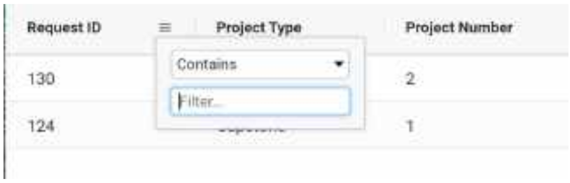
Improvements:
- Added advanced filtering and sorting options
- Added the functionality to clear filters
- Added a functionality to show and hide details for all procurements
- Improved UI/UX

Performance

- Completed 2 of the high-priority tasks and many of the low-priority tasks
- Successfully implemented sorting and filtering to improve UX/UI
- Added the functionality to show and hide details for all Procurement tabs
- Implemented the functionality to open the details of a procurement by clicking on the ID
- Added the functionality to clear filters

Results





Request ID	Project Type	Project Number
130		2
124		1

Summary

UTDesign Procurement Manager:

- The UPM tool provides assistance for UTDesign and EPICS projects with processing procurement requests.

- Implemented sorting and filtering to tables to provide faster access to select different procurement requests

- Implemented the ability for the user to click on the request ID to open the details of the selected procurement request.

UTDesign Procurement Manager (UPM) Security

Elissa Williams, (emg170330@utdallas.edu); **Julie Dam**, jdd150330@utdallas.edu; **Kate Malinis,** kate.malinis@utdallas.edu; **Nargol Payman**, nxp171230@utdallas.edu; **Ryan Peichel**, ryan.peichel@utdallas.edu

CS 4485 / Spring 2021
Department of Computer Science
Erik Jonsson School of Engineering & Computer Science
The University of Texas at Dallas
Richardson, TX 75080, USA

Abstract

The UTDesign Procurement Manager (UPM) is a Web-based application designed to aid in the procurement of materials for projects associated with UTDesign. The application allows student to submit procurement and reimbursement requests for the technical managers and faculty advisors associated with the project to approve or deny.

Our goal was to add security measures to the existing application's framework in order to better protect the user data stored within the system and reduce vulnerabilities. To achieve this, we transitioned the tool to HTTPS, implemented Shibboleth's Single Sign on, encrypted the database at rest, and ensured all communication took place on UTD's email servers.

Keywords: Cyber Security, HTTPS, Single Sign On (SSO), Encryption

Architecture

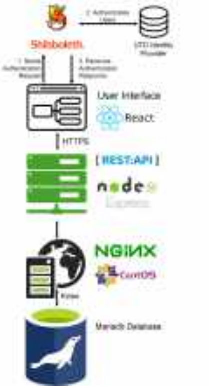
Shibboleth
- Service Provider V3
- UTD's Identity Provider

User Interface
- React.js Framework
- Calls the REST API using HTTPS

Rest API
- To serve data to the User Interface
- Implemented alongside Node.js with Expressress.js as backend
- Knex.js query builder

Server
- UTD Linux Server with CentOS 7
- Nginx web server

Database
- MariaDB



Impact

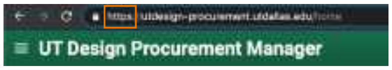
- All Communication from the Website are now being sent from a UTD email instead of a Gmail, making them more trustworthy and secure.
- The application is now on HTTPS instead of HTTP.
- The MariaDB Database has been fully encrypted in order to ensure the security of the student's personal information.
- The app is now set up with Shibboleth's Service Provider and UTD's IDP, and the sign in page now reflects this change.

Performance

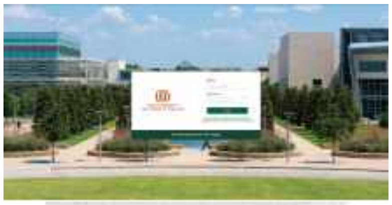
- UPM now meets most of the University's security requirements.
- All high priority tasks have been delivered in a timely manner.
- All stretch goals have been satisfied.
- The faculty advisor and technical manager were consistently kept informed and satisfied by the team's progress.

Results

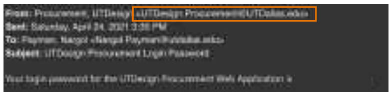
- **HTTPS**


https://utdesign-procurement.utdallas.edu/home
UT Design Procurement Manager

- **Shibboleth**



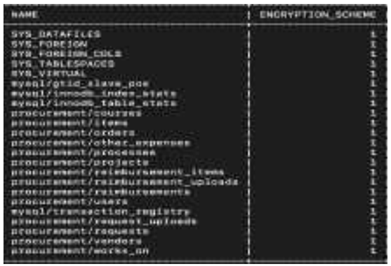
- **Email Notification From a UTD Email**



From: Procurement, UTDesign <UTDesign.Procurement@UTDallas.edu>
Sent: Saturday, April 24, 2021 3:38 PM
To: Payman, Nargol <Nargol.Payman@utdallas.edu>
Subject: UTDesign Procurement Login Password

Your login password for the UTDesign Procurement Web Application is

- **Database Encryption**



```
 NAME                                          | ENCRYPTION_SCHEME
-----------------------------------------------+------------------
 SYS_DATAFILES                                 |                 1
 SYS_FOREIGN                                   |                 1
 SYS_FOREIGN_COLS                              |                 1
 SYS_TABLESPACES                               |                 1
 SYS_VIRTUAL                                   |                 1
 mysql/gtid_slave_pos                          |                 1
 mysql/innodb_index_stats                      |                 1
 mysql/innodb_table_stats                      |                 1
 procurement/courses                           |                 1
 procurement/items                             |                 1
 procurement/orders                            |                 1
 procurement/other_expenses                    |                 1
 procurement/processes                         |                 1
 procurement/projects                          |                 1
 procurement/reimbursement_items               |                 1
 procurement/reimbursement_uploads             |                 1
 procurement/reimbursements                    |                 1
 procurement/users                             |                 1
 mysql/transaction_registry                    |                 1
 procurement/request_uploads                   |                 1
 procurement/requests                          |                 1
 procurement/vendors                           |                 1
 procurement/works_on                          |                 1
```

Summary

- By moving our website to HTTPS we were able to ensure that all data communicated to and from the server would be encrypted.
- We also encrypted the data at rest on the server to ensure the security of the student data.
- Our team ensured that all communications from the app were reliable and off a UTD email instead of a general Gmail.
- We implemented the latest format of Nginx on the server and configured it such that it would communicate with Shibboleth and FastCGI modules.
- As part of the transition to Shibboleth Service Provider, we set up responder and authorizer sockets and connect it with UTD's Identity Provider.
- We replaced the old login page with UTD's standard login and used cookies in order to pass the Shibboleth attributes into the API to get the correct view based on the role of the user.

Movement Disability Application

UT Southwestern

Connor Hulla, cjh170230@utdallas.edu; Ariel Kramer, aek170030@utdallas.edu; Nicholas Leung, nml170001@utdallas.edu; Ezinne Megwa, enm170130@utdallas.edu; Marie-Claire Salha, mjs170530@utdallas.edu; Adam Yarneau, any160130@utdallas.edu

CS 4485 / Spring 2021
Department of Computer Science
Erik Jonsson School of Engineering & Computer Science
The University of Texas at Dallas
Richardson, TX 75080, USA

Abstract

Many movement disabilities can be hard to diagnose and treat due to their infrequent or short occurrences. A brief doctors visit may not catch these movement disabilities in action. This means that when they do occur, a doctor may not be around to analyze the episode.

Our team's solution was to create apps for both Android and IOS that allowed participants to quickly record, rate, and submit videos of their movement disability episodes to UTSW to review. Doctors will use a website to review and rate video submissions as well.

This will help doctors get a better understanding of their participant's movement disabilities. It will also give participants some agency and control over their disability. Our team has been able to implement the main functionalities of both apps, the website, and database.

Keywords: Android, IOS, Database Systems, Web Development

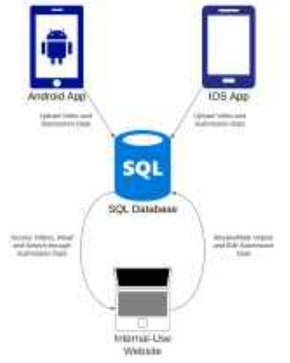
Architecture

Android Mobile Application
- Android Studio - Java

IOS Mobile Application
- Front End- xCode, Swift
- Library - SwiftUI

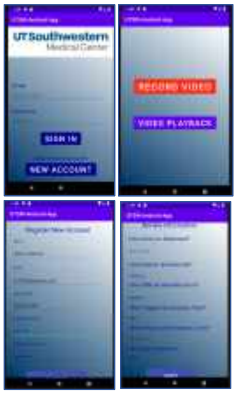
Website
- Front End - React.js, Bootstrap
- Back End - Node.js, Axios

Database
- MySQL Workbench - SQL
- MySQL Server
- JDBC - Java

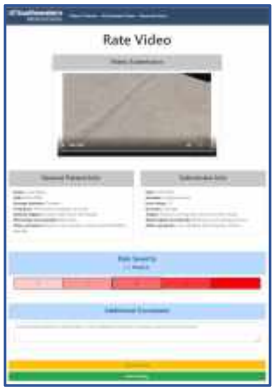

Android App
IOS App
Upload Video and Submission Data
Upload Video and Submission Data
SQL
SQL Database
Internal-Use Website

Results

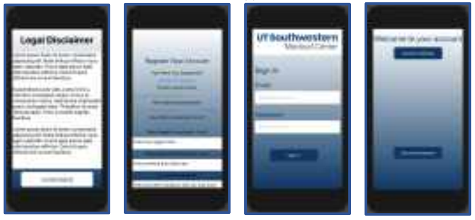
Mobile App: Android



Website



Mobile App: iOS



Impact

- Rare, infrequent movement disorder episodes can be recorded and studied at a later time.
- Obtain the necessary legal and medical permissions from a study's participants.
- Allows participants to review, rate, and comment on their episodes.
- Allows doctors to further analyze a participant's movement disorder, record their observations, and even change inaccurate information.
- Store all video and related medical information in an organized and efficient manner.

Performance

- Weekly team meetings between the student team to discuss progress, issues, and future long and short term goals.
- Progressed measured by the amount of implemented features and goals.
  - All features related to front end design were successfully completed.
  - Back end features were either completed or set up for simple future implementation.
  - System structure and framework was tested and finalized.

Summary

Goals
- Help capture rare or infrequent movement disorder episodes.
- Let doctors analyze these recordings to better understand a participant's movement disorder.
- Give participants an easy, quick way to submit recordings of their episodes.
- Also give participants some agency in their diagnosis or treatment of their movement disorder

Accomplishments
- Built a solid framework for storing, uploading, and accessing video submissions between mobile apps, an internal-use website, and a database.
- All front end designs have been finished and polished.
- Mobile apps were specifically made to be easy-to-use and intuitive so anyone can use the app with minimal confusion.
- Created a flexible framework for a website that can access and interact with a database.
- Designed a robust and organized database that can store video submission information

App Development for Situational Awareness Wearables

Jonathan Tygielski, jrt170130@utdallas.edu Josh Guzman, jrg170130@utdallas.edu
Vincent Vu, vincent.vu@utdallas.edu Tuan Nguyen, tmn180000@utdallas.edu

CS 4485 / Spring 2021
Department of Computer Science
Erik Jonsson School of Engineering & Computer Science
The University of Texas at Dallas
Richardson, TX 75080, USA

Abstract

A mobile application was developed for Tac-Alert's wearable device to provide a simple and effective user interface. The application provides enhanced situational awareness for tactical users by managing detection data, radiation-safety levels, and alerts. Location and status information of nearby users will be implemented in the future. The app wirelessly communicates with the device via the Bluetooth low energy protocol and runs on both Android and iOS for maximum compatibility. The app is both modular and configurable so it can scale properly.

Keywords: Mobile App, Bluetooth, UI/UX

Architecture

Android and IOS applications

- React Native Expo
- JavaScript
- Figma

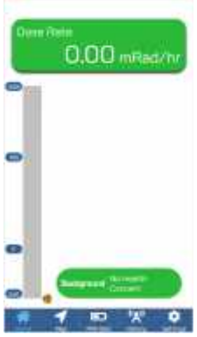
Impact

This app is beneficial because it enables law enforcement agencies to track radiation levels. It also provides clear instructions on how to respond to dangerous situations for officer and public safety.

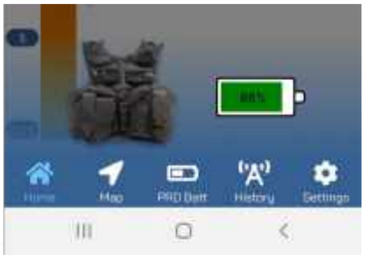
Performance

- Cross-platform development ensures the app has consistent performance across both Android and IOS.
- Fast response times for receiving data from the wearable device
- Company mentors evaluated the app and found it to meet the specifications

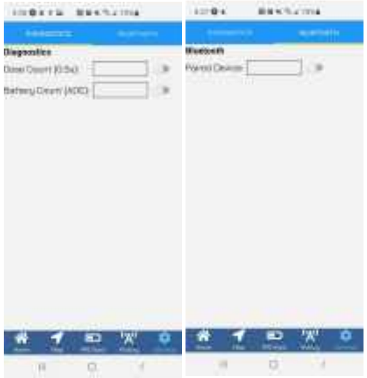
Results



Home Screen



Battery Display



Settings Page

Map/History Page

Summary

The Tac-Alert App:

- Provides dynamic radiation dose readings and instructions
- Shows battery level of the connected wearable device
- Diagnostic mode for testing
- Future Bluetooth connectivity with wearable device
- Map and History pages will be additional features
- Intuitive UI built for law enforcement agencies

Asterisk Solutions

Live Site Widget Store

Hisham R Alsamarrai, HRA180000@utdallas.edu; **Devraj Gajurel**, DXG190004@utdallas.edu; **Haris Gusic**, HXG170006@utdallas.edu; **David Burgwin**, DAB180001@utdallas.edu; **Ahmad Kadado**, AXK170006@utdallas.edu

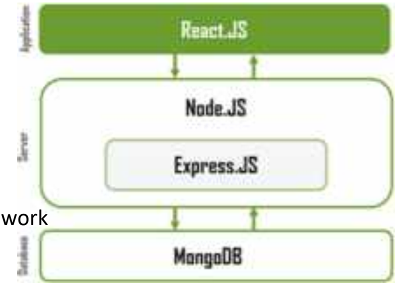
CS 4485 / Spring 2021
Department of Computer Science
Erik Jonsson School of Engineering & Computer Science
The University of Texas at Dallas
Richardson, TX 75080, USA

Abstract: The group has been tasked with developing a Widget Store for Asterisk Solutions' live event site. The site allows users to build and customize temporary websites for hosting Online events. Since Asterisk Solutions currently does not have a suitable Plugin Store that they can port to their live event site, this project is unique. To develop the application, the MERN (MongoDB, Express, React, Node) stack will be utilized with a REST framework for the Application Programming Interface to communicate with the database. This application will improve the current live site as users will be able to use and share widgets that will allow them to further customize their events. Furthermore, Asterisk Solutions can use the system in the future to distribute their own various widgets and plugins to their customer base.

Keywords: Widget Store, Web Application, MERN stack, RestFul API

Architecture:

- MERN Stack
  - MongoDB
  - Express
  - React
  - Node
- API
  - Rest Framework
- Load Widget Dependencies
  - Unpkg/Skypack



Keywords: Unpkg/Skypack, Mongo, Express, React, Node

Performance:

- All the features that require user interaction are fast and responsive.
- Code is readable, formatted, and modularized to allow future developers to edit easily.
- Files are separated into directories and sorted according to their purpose.
- Request and response to/from the API hitting database is quick and efficient.
- Our application meets the industry performance standard.

Results:

Widgets

Impact:

- A User-friendly UI design allows users to easily add, manage, search, and preview Widgets from other users.
- Seamless integration between UI, backend, and database.
- Users will have access to fast and functional Widgets that allow them to have a better user experience when customizing their website.
- The current build will allow Asterisk Solution's to further develop the Widget Store in the future.

Metrics:

- 90% completion of project goals due to a lack of expertise in the domain and tools used when the project began.
- Weekly standups occurred between the team and the mentor where the team members expressed what has been completed and goals for the upcoming standup.
- Constructive feedback was given during each meeting by the mentors.
- Functional Requirements were fully met except for the loading of widget dependencies which was partially implemented.

Summary:

- The API is implemented with CRUD capabilities on Widgets and their Dependencies.
- Widgets and their Dependencies fit a schema made in the backend when inserted into the collection.
- The Backend has the capability of creating a database schema for Widgets and their Dependencies on the Mongo Database.
- Can save all widget dependencies and variables used in the widget in the database.
- Implements a Json parser to parse all the Dependencies and their information from the package file of the Widget.
- Displays a User-friendly UI that allows users to add and manage widgets that will be used on their websites.