Independent Study Final Summary

Author: Ethan Perry
Advisor: Prof. Donna Bodeau
Clients: Kevin Young and Lois Young of the Chad William Young Foundation

Background

The Chad William Young Foundation (CWYF) has worked with Colorado School of Mines over the past several years in their goal of creating a software tool designed to aid cyclists and race organizers in keepings themselves and their constituents safe. During the early stages of the project, the primary deliverables included analysis methodology for evaluating cycling safety. This past fall, CWYF partnered with the department of computer science for a field session to make additional progress towards their goal. The team working on the project included Zoe Baker and Cole Wapelhorst, who were both working on the data science side of the project, and Ethan Perry, who worked on coding the mobile and server applications.

By the end of the project, a working Android mobile application and an accompanying server application programmed in python were produced. These apps were presented at the concluding seminar in December of 2021, and the project was put on pause for the Spring semester of 2022. Progress resumed as of June of the Summer 2022 semester.

Goals

The primary objective of this independent study was to improve both the mobile and server applications created as a part of the original field session project. These improvements were designed to target (1) the usability, defined as the ease of use associated with the application, (2) the scalability, defined as the ability to scale the application infrastructure with the size of its userbase, and lastly (3) the extensibility, defined as the ease of adding new features to the application. Several core deliverables, described in the sections that follow, were created to achieve these primary improvement goals.

Learnings

Three readings were self-assigned during this independent study:

1. Building Microservices
2. Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems
3. Design Patterns: Elements of Reusable Object-Oriented Software

Each of the three books are considered essential guides to achieving scalable cloud engineering, with the first book specifying patterns to improve the overall architecture of the back-end, the second describing the best practices for ingesting huge volumes of data and handling requests for such data, and the third highlighting ideal patterns for creating efficient programs and better-quality code. These books were read throughout the earlier stage of the independent study to gain technical background necessary to achieve the associated core deliverables.

Core Deliverables

The majority of core deliverables of this project were planned out from the beginning of the semester, with one deliverable being identified near the end of the independent study. All core deliverables were presented as a part of the final independent study presentation (attached at the end of this document).

Infrastructure Migration

Comprising a large part of the independent study period was an infrastructure migration from the previous monolithic back-end to a new microservice back-end hosted on Microsoft Azure. Several major changes were needed to make this migration possible, including a general restructuring and reorganization of the back-end service, the adoption of a new database technology, the containerization of the back-end, and the actual migration itself. The infrastructure migration was designed to improve the scalability and maintainability of the back-end, as well as lowering the associated cost of hosting.

Back-end Improvements and New Features

Beyond the changes needed for the infrastructure migration, several back-end changes were necessary for new product features and increased performance of the back-end. Specifically, the back-end was modified so that recordings of user rides and the data associated with these rides could be stored. This allowed for the addition of the new "activities" feature of the mobile application, where users can see and modify the rides they have recently taken.

For more details of the back-end improvements and new features, please see the presentation attached at the end of this document.

Mobile Improvements and New Features

The mobile (front-end) application experienced several major improvements to its design and also had a large feature update. The new "activities" feature tracks a user's ride and associated data of that ride. This includes the alerts which they created while riding, a description of that ride, and a title. To best visualize this data, a new activity view was created. The view includes an interactive map where the user can revisit the route which they took on their ride and the associated alerts created on that ride. Below the interactive map is an editable description of the ride, as well as a list of all alerts which the user created while riding. Clicking on an alert reveals an alert view page which is similar in structure to the activities page but contains content which is specific to that alert. This new activity feature is expected to aid in the user in their creation of detailed and helpful alerts.

Additional stylistic changes were made to the mobile application, including updated colors and font throughout the app. User menus and interactions were generally simplified to reduce the upfront strain of learning how to use the app, and additional textual descriptions of buttons and icons were included throughout the application, allowing for better understanding of what those buttons and icons are for.

For more specific details, images, and videos of the UI improvements, please see the presentation attached at the end of this document.

Algorithm Improvements

One area of improvement which was not initially planned for the independent study was the improvement of the alert time out algorithm. Previously, alerts would time out after a non-variable period which was set as an environmental configuration (4 days for example). This time out is now variable, described by the algorithm below:

Parameters:

- Base Timeout (T)
- Maximum Distance (M)
- Number of Nearby Alerts (N)
- Distance Multiplier (D)
- Popularity Multiplier (P)
- Alert Upvotes (F)
- Final Timeout (O)

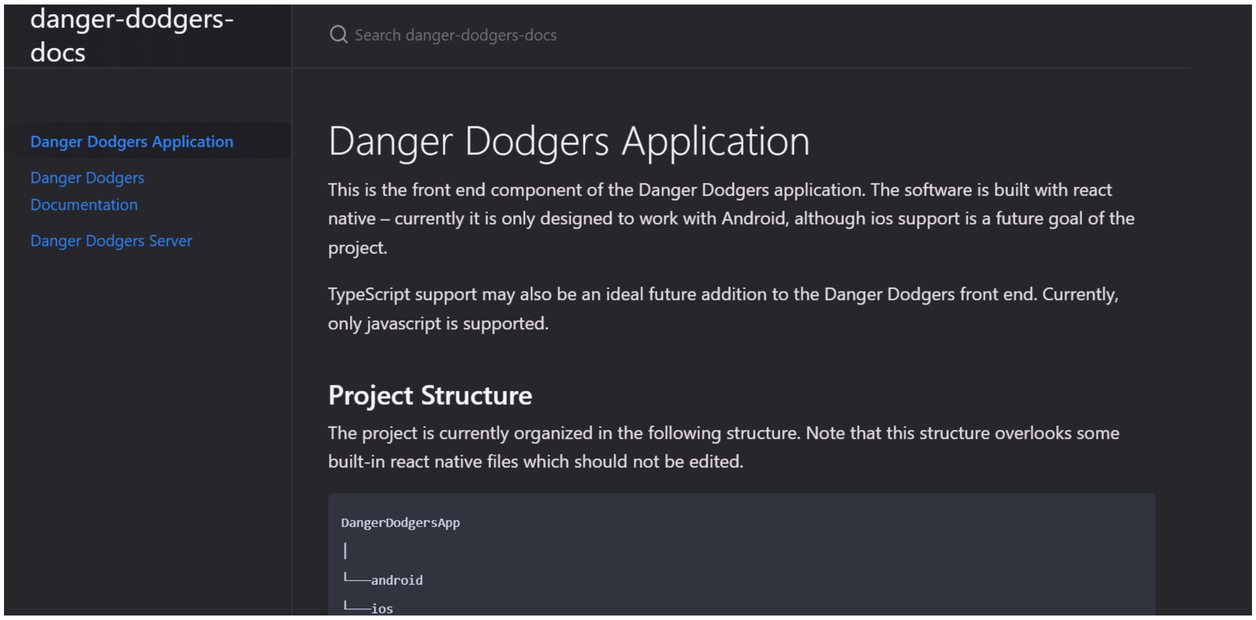
Algorithm:

- Search for the N closest alerts within distance M
    - If there are less than N alerts within distance M, this area is sparse, and O = T
- Compute the average distance from the current alert to the N closest alerts, with average distance denoted as A
- Final Timeout = T * (A * D) * (F * P)

The distance and popularity multipliers are set as environmental variables and will likely be subject to change based on future user research.

Developer Docs Website

The developer documentation website is an automatically generated documentation source for future developers working on mobile and back-end applications. The developer docs are easy to edit, add to, and improve, with all changes pushed to the cloud repository being automatically built into associated web pages (figure 1 showing an example of these webpages below). These developer docs are expected to serve as easy and useful onboarding resources for quick training of future application developers.


danger-dodgers-docs
Search danger-dodgers-docs
Danger Dodgers Application
Danger Dodgers Documentation
Danger Dodgers Server

Danger Dodgers Application

This is the front end component of the Danger Dodgers application. The software is built with react native – currently it is only designed to work with Android, although ios support is a future goal of the project.

TypeScript support may also be an ideal future addition to the Danger Dodgers front end. Currently, only javascript is supported.

Project Structure

The project is currently organized in the following structure. Note that this structure overlooks some built-in react native files which should not be edited.

```
DangerDodgersApp
|
└──android
└──ios
```

Figure 1. A screenshot of the developer documentation website. The documentation site is automatically built as developer docs are updated and includes an additional search feature for ease of use in navigating the documentation.

Conclusion

All core deliverables were completed during the duration of the 8-week independent study; however, there is still work to be done – application development is expected to be an ongoing process of testing and improvement until any large public release occurs. There are also several fundamental changes that were discussed during the final presentation – all of which may affect the outcome of the application, both in terms of the features it has to offer, its target audience, and the method of release. These are discussed in the future research section below.

Beyond the creation of the core deliverables, this project was an important learning experience. With working full time and taking an 8-week class simultaneously, new time management skills were required, and more importantly, the technical lessons learned were significant. Before beginning this independent study, I had little idea of how to do proper cloud engineering and wasn't as well versed in the essential software design patterns. After having begun my job, reading several books on the topic, and gaining experience by improving the application, I feel well versed in essential concepts and could replicate this effort on projects in the future.

Future Research

Throughout the course of the term, several areas of future research were identified. These have been classified on three different levels, depending on their scope: (1) feature level, (2) project level, and (3) product direction level.

Feature level changes are those that could be reasonably completed in around a month and are scoped to improve a single aspect of the application. Project level changes are those which would reasonably take a few months to complete and may include multiple related features. Finally, changes in the product direction are those which affect the entire product, and such changes are continuous over time.

Feature level changes:

1. Personalized hazard thresholds, filtering for and showing only the hazards which meet certain criteria

Project level changes:

1. Audible alerts (text to speech alerting)
2. IOs compatibility
3. Expanded tracking of other forms of exercise (running, swimming, etc.)
4. Allowing for user navigation, similar to the navigation included in google or apple maps for example

Product direction changes:

1. Only allowing official sources (such as race organizers and parks/recreation officials) to create activities
2. Specifically targeting race organizers as the constituents of the application