Owl Walk

CS4850
Senior Project
Section 03

Kennesaw State University

College of Computing and Software Engineering (CCSE)

Department of Computer Science

Group: SP7
Juan Tapia, Andy Martinez-Reyes, Sam Perez,
Shawn Villacorta, Steven Holmes

Draft of Final Report
November 28, 2022

Overview

Kennesaw State University (KSU) has much artwork and landmarks throughout the Kennesaw and Marietta Campus. Most students may not know what artwork they may pass by or what the story behind it might be. With our mobile app, art will be located and documented; it will allow students and guests to learn about their environment. Allowing clients to request landmarks will keep the mobile app updated and allow a high level of interaction. For clarity, artwork and landmarks will be identified as “Points-of interest” as this can include artwork, locations, and monuments. These Points-of-Interest need to meet a specific set of requirements to be considered for the database.

Purpose

The purpose of this mobile application is to cause users to feel more connected to the university campuses by allowing users to locate interest points on campus. Being able to locate interest points allows users to get directions using these points of interest, locate events in the vicinity of the interest point, or admire interest points on campus.

Final Deliverables

For final deliverables, the following will be submitted:

1. Software Development Report
2. iOS/Android Compatible Mobile Application
3. Planning Document
4. Gannt Chart

Table of Contents

Background Information

Kennesaw State University (KSU) has much artwork and landmarks throughout the Kennesaw and Marietta Campus. Most students may not know what artwork they may pass by or what the story behind it might be. With our mobile app, art will be located and documented; it will allow students and guests to learn about their environment. Allowing clients to request landmarks will keep the mobile app updated

and allow a high level of interaction. For clarity, artwork and landmarks will be identified as “Points-of interest” as this can include artwork, locations, and monuments. These Points-of-Interest need to meet a specific set of requirements to be considered for the database.

The functionalities of the application include

- View the KSU maps to see interest points
- Interact with interest points to get more information of those points
- Have the ability to easier locate interest points
- Create an account
- Log into the user created account using unique credentials
- Make suggestions to interest point additions

Planning

During planning, our team focused on three major tasks. Task one is deciding the collection of development tools we plan on using for the development of our mobile application. Task two is determining what requirements will be feasible and are necessary for the functionality of the application. Task three is developing a project plan.

For task one, development tools are separated into categories; the categories for development tools are mobile application development, database development, prototype development, and user interface development. For mobile application development we concluded that React Native and Firebase are to be used. React Native will serve as the development tool for the mobile application’s base functionalities, and Firebase will serve as the application’s authentication development tool. Combined, React Native and Firebase will serve to complete all necessary functionalities of the mobile application. For database development, Firebase was originally decided as the database development tool; due to complications, MySQL is the development tool used to create the database. For prototype development, Adobe Photoshop, Canva, and Adobe XD are utilized to create screen prototypes for the user interface. For user interface development, React Native and Adobe XD are used to create the user interface of the mobile application.

For task two, the requirements for the mobile application are in a general list of requirements as shown in the table below. The general requirements list includes all requirements that will be considered in the development of the mobile application. Using the table, the Phase I and Phase II requirements lists are able to be derived.

Requirements	Phase I	Phase II
1. Retrieve interest points 2. Create prototype 3. Create map UI 4. Create map functionalities 5. Create interest points database 6. Add interest points to map 7. Create interactive login page 8. Create security system 9. Add login option on the map 10. Include interest point addition functionality 11. Test map functionalities 12. Add map icons 13. Format the UI 14. Add a pop-up tab for description 15. Monetization 16. Server Based Database	1. Retrieve interest points 2. Create prototype 3. Create map UI 4. Create map functionalities 5. Create interest points database 6. Add interest points to map 7. Create interactive login page 8. Create security system 9. Add login option on the map 10. Include interest point addition functionality 11. Test map functionalities 12. Add a pop-up tab for description	1. Server Based Database 2. Monetization 3. Format the UI 4. Custom Icons 5. Logo

Table 1.1: Requirements

Conflicts

During the development of our iOS and Android compatible mobile application, our team encountered many conflicts or inhibitors.

A major conflict encountered is the inability for all team members to download necessary development software. Due to this, we often participated in collaborative planning and software development. This included meetings being held with multiple people where development decisions were discussed and agreed upon. This allowed all parties involved to research relevant documents and find example code of the development decisions. In addition, all necessary code suggestions could be copied and pasted into the meeting chat and be tested in real-time.

In addition, we faced conflicts with database related development. During the initial planning of Owl Walk, Firebase was decided as our database development software. Throughout development, we were advised to utilize a different database due to React Native and Firebase having version conflicts and due to a dependency issue. Due to this, database development had to be changed to MySQL, thus causing a delay in database development.

Design

For the application, we followed the formatting of a Google Map. To do this, we utilized the Google API for React Native. This allows for the map default details to be automatically included in an empty map. The default details include roads, hills, buildings, etc. Using the Google Maps API, we are able to initialize the opening position upon application startup. For the Kennesaw and Marietta maps, the initial load-up location is a notable interest point.

For the database, we utilized MySQL due to technical complications. The formatting of the data in the database utilizes the following attributes: name, location(coordinates), description, and images. The

formatting of the interest points and their respective attributes follows Json data marshaling. This ensures that the data is not corrupted or misread during execution on different devices.

Architecture

The architecture of how the application calls functions and sends data is shown below. The pages on the mobile application are

1. Front Page
2. Login/User Creation
3. Maps
4. Dynamic Page
5. Landmark Request

The user can either be a guest or an account user.

The Guest has the options to

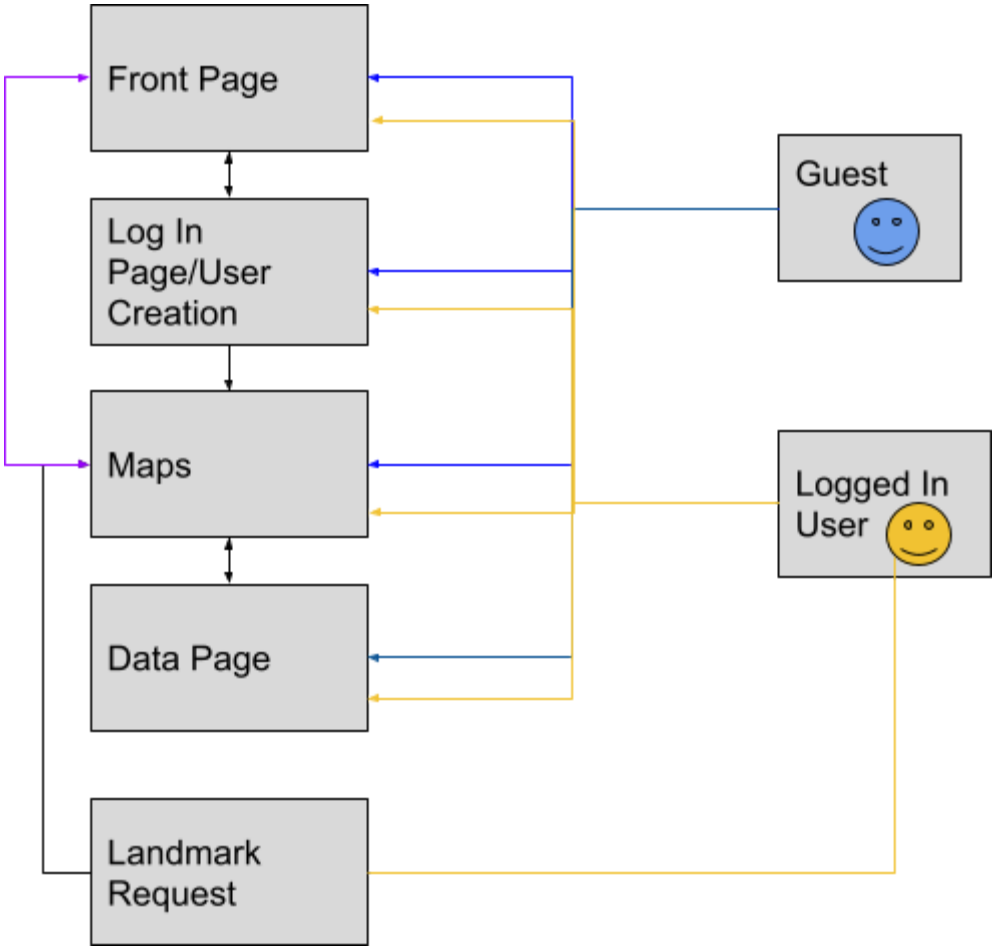
1. View the front page
2. view the login page/create account
3. View the maps
4. View the data page

The Account user has the options to

1. View the front page
2. view the login page/create account
3. View the maps
4. View the data page
5. View the request a landmark page

The following chart shows what pages are accessible from other pages

Pages	Front	Login/User	Maps	Data	Landmark
Front		✅	✅	✅	
Login/User	✅	✅			✅
Maps	✅			✅	
Data			✅		
Landmark		✅	✅		



UI Discussion

For the user interface, there were many different designs discussed. A major design discussed throughout phase I are depicted in the images. After the completion of phase I, we held a meeting to discuss potential updates to the UI we could make. A major update to make is to ensure the buttons are not overcrowding the screen. Making the buttons smaller will support not overcrowding and also make the UI more user friendly. In addition, we want to refine what font is being used. The font currently being used is substantial; however, we want a font that more clearly depicts a welcoming and fun mobile application while also maintaining the user friendly appearance. Finally, we had many discussions on the logo we used. A major issue with the logo is ensuring that we made the color of it contrast enough with the background of the screen showing. In addition, due to the many name changes of our app, we continuously had to change the logo to depict the new name.

Operation Summary

The following is a description of how to operate the main Expo mobile application.

To download all dependencies to run the application, the following installation steps are needed:

1. Open a command prompt/terminal
2. Use this link to install Node.js and install the npm package installer
   https://docs.expo.dev/get-started/installation/
3. Start a template build using
   `npx init app_name`
4. Use this link to download screen utilities
   https://reactnavigation.org/docs/getting-started
5. Run the command
   `npm install @react-navigation/native-stack`
6. Use this link to install react-native maps
   https://docs.expo.dev/versions/latest/sdk/map-view/

After following these steps, all dependencies and requirements to host the mobile Expo application are installed.

To run the application, follow these steps:

1. Install the expo application on your phone or have an emulator running on your computer
2. Run the command

   `npx expo start`
3. Scan the QR code using the application or run the application using the emulator

During the installation, there are possible errors that may arise. When running, ensure that you do not use yarn and npm to run the application. To run, use only one of the two options, yarn or npm. In addition, do not install the legacy global CLI or the developer client unless you are knowledgeable about their functionalities. Installing these without being familiar may cause undefined behavior.

The following is a description of how to operate the MySQL Database.

To download all dependencies to run the database, the following installation steps are needed:

1. Download MySQL using this link (download for your specific machine)
   https://dev.mysql.com/downloads/

   The following video will give a full tutorial of how to install MySQL and all dependencies
   https://www.youtube.com/watch?v=7S_tz1z_5bA

2. Run the SQL statement
   ALTER USER root@localhost IDENTIFIED BY PASSWORD 'Your_Password_Used''

To use the database along with React Native

1. Change the directory to the root folder of the application
2. Run the commands

   ```
   npm install express
   npm install body-parser
   npm install mysql
   ```

To run the database

1. In the root folder of the project, write

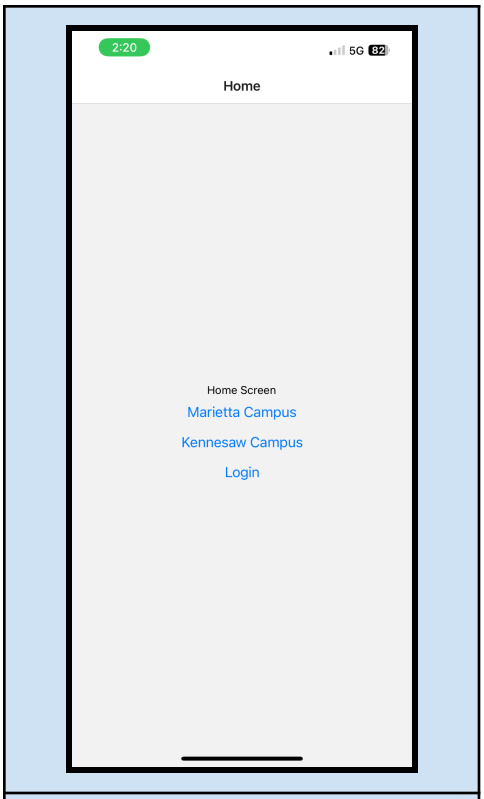
   ```
   node KennesawDatabseConnection.js
   ```

   in the command prompt window to run the SQL Connection. This must be ran before the mobile application

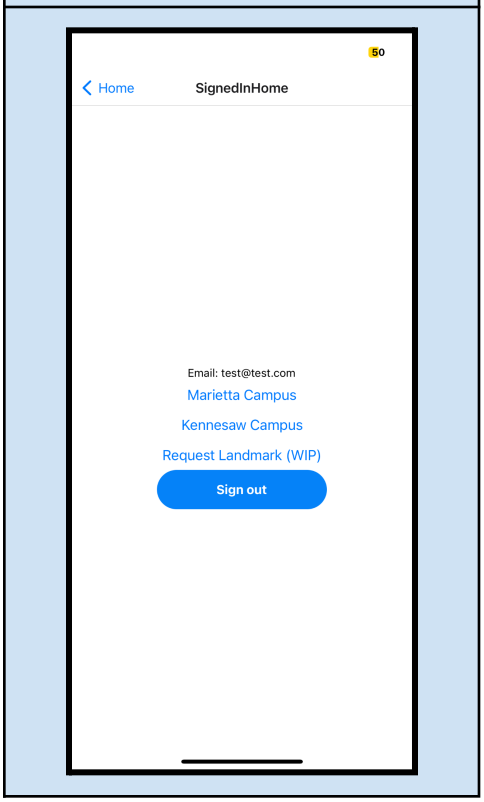
2. Follow the "to run the application" steps for how to run the Expo mobile application

If the application refuses to connect or gives an authentication error, ensure that the proper names are being used in the connection. In addition, the JavaScript files need the proper IP Addresses in the URL of the machine being used. If a remote PC were used for development, this would be defaulted. Since the application needs to be run manually, ensure that the IP address is changed each time; otherwise, there will be an error and the mobile application will not run.

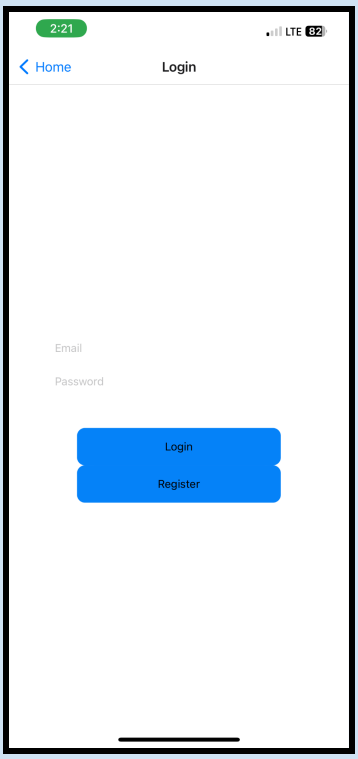
Once all steps are complete, the mobile application should have the expected outputs.



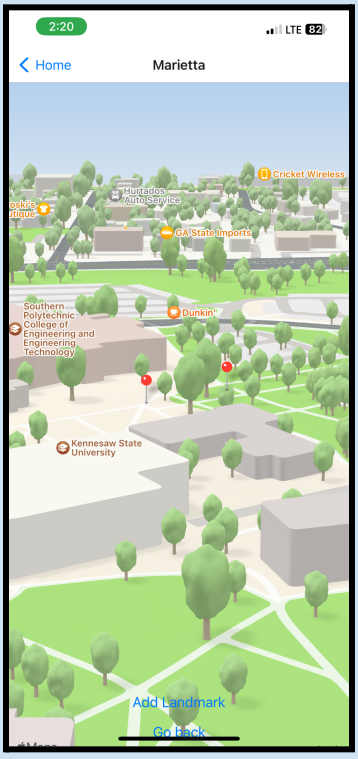
The home screen prior to the user logging in or creating an account



The home page after the user logging in or creating an account



The login screen



The Marietta Map

The Kennesaw Map

QR Code

Planning and Management

Collaboration tools utilized by our team are Microsoft Teams, Discord, and if needed, SMS Messaging. These collaboration tools were utilized during mandatory weekly meetings and optional collaboration meetings that are scheduled by an as-need basis. Our meeting schedule was determined by gathering all available times of our team members and deciding between the two most available times. The final decision for the meeting time and day was Tuesdays at 7:00pm.; however, we ensured to adjust the

meeting time depending on other devotions or complications. Image 1(Appendix) depicts the availability of our team members.

The following chart shows a summary of major talking points during meetings we held.

Meeting	Summary
8/22	This served as our initial meeting. We discussed expectations within the team and within the project. In addition, we discussed the information to be included into the project plan.
8/29	This served as our requirements review meeting. We discussed our requirements in regard to the project and what could be feasible in the amount of time we have with respect to our workload in other courses. Furthermore, this was the end to our planning sprint, sprint 1.
9/6	This served as the development planning meeting. We discussed issues with installing required software and setting up dependencies. This also served as the meeting where our lead developer, Juan, gave us information and a walkthrough on the APIs we will be utilizing during the development of our mobile application.
9/13	This served as the meeting that
9/20	This meeting served as the meeting where all major updates regarding development will be discussed. The topics discussed are updates on Firebase as the database. As of now, there are issues with the database that are not able to be solved.
9/27	Meeting canceled due to high stress and popular vote
10/4	Meeting canceled due to midterms, wanted to allow everyone to focus on their exams
10/11	This meeting served as the planning meeting for the milestone one readiness presentation to be held on 10/18. During this meeting, we ensured to get all information relevant in the PowerPoint and do a practice demo of the current mobile application.
10/18	This meeting served as a review meeting for the advice posed during our presentation. We decided then to utilize MySQL due to complications using Native and Firebase at specific versions. This also served as the final day of sprint 2.
10/25	This meeting served as a general status update.
11/1	This meeting served as a general status update
11/8	This meeting served as a user interface development discussion.
11/15	This meeting we discussed the development of the user interface. This meeting also served as the final day of sprint 3.
11/22	Meeting canceled for Thanksgiving Break
11/29	This meeting served as our final meeting before product deliverable, We discussed

Updated Project Plan

Project Planning

Overview

Kennesaw State University (KSU) has much artwork and landmarks throughout the Kennesaw and Marietta Campus. Most students may not know what artwork they may pass by or what the story behind it might be. With our mobile app, art will be located and documented; it will allow students and guests to learn about their environment. Allowing clients to request landmarks will keep the mobile app updated and allow a high level of interaction. For clarity, artwork and landmarks will be identified as “Points-of interest” as this can include artwork, locations, and monuments. These Points-of-Interest need to meet a specific set of requirements to be considered for the database.

Points-of-interests

- Must be a location, artwork, or Monument.
- Must be part of the Kennesaw State University Campuses (Kennesaw/Marietta).
- Must have a noticeable background a story behind its creation.
- Must be accessible to students to a certain degree (limits some indoor artwork)

Phase I Requirements

1. Retrieve interest points
2. Create prototype
3. Create map UI
4. Create map functionalities
5. Create interest points database
6. Add interest points to map
7. Create interactive login page
8. Create security system
9. Add login option on the map
10. Include interest point addition functionality
11. Test map functionalities
12. Add a pop-up tab for description

Phase II Requirements

1. Server Based Database
2. Monetization
3. Format the UI
4. Custom Icons
5. Logo

Final Deliverables

1. Owl Walk iOS/Android Compatible Mobile Application
2. Documentation
3. Operations Summary
4. Progress Timeline
5. Gantt Chart

Project Website

https://srperez0619.github.io/Senior-Project

Project Team

Role	Name	Responsibilities	Contact
Team Leader	Sam Perez	Advising Team when Needed, Software Development, Documentation, Data Gathering, Point of Contact	srperez0619@yahoo.com 7068768881
Team Members	Andy Martinez-Reyes	Software Development, UI Planning and Development, Data Gathering	Andy.m.reyes514@gmail.com 661-513-7901
	Steven Holmes	UI Planning and Development, QA	sholmes9461@gmail.com 706-659-5557
	Shawn Villacorta	Documentation, UI Planning and Development, Data Gathering	shawnv.villacorta@gmail.com 678-665-0458
	Juan Tapia	Software Development, QA, Security	malevolent_juan@yahoo.com 404-808-8603
Advisor	Sharon Perry	Facilitate project progress. advise on project planning and management.	Sperry46 in D2L !!

Future Meetings

With respect to all members' availability, we have tentative schedules posted for these date and times. These are subjected to change when necessary.

Sprint	Date	Time
Sprint 1	8/29/22	6:30pm-8:00pm
Sprint 2	9/21/22	6:30pm-8:00pm
	10/5/22	6:30pm-8:00pm
Sprint 3	10/19/22	6:30pm-8:00pm

Sprint	Date	Time
	11/2/22	6:30pm-8:00pm
Sprint 4	11/16/22	6:30pm-8:00pm
(CDAY)	11/20/22	6:30pm-8:00pm

Communication/Collaboration Plan

The communication tools we will utilize are Groupme for group messaging, phone numbers for emergency contacting, Discord for team meetings, Email for formal communication.

The collaboration tools we will utilize are Github for a shared code repository to upload and pull code, Discord can be collaboration meetings, and Expo for coding with Github compatibility. The Project Lead, Sam, will conduct bi-weekly meetings for updates and task assignments. She will also facilitate any communication between the advisor and the team.

Risk Assessment

Potential Delays

- API Implementation
    - Create a guide of the setup and software requirements
    - Examine issues and debug to understand issues
- Network Availability
- Team Availability
- Data Security
- Kennesaw Campus Information

If any significant delays occur the project plan will adapt accordingly. If constant delays occur a meeting will be held to get the project back on track.

Updated Gantt Chart

For our gantt chart, we have our expected and actual start/end dates for major tasks. Under each tasks are subtasks that were considered and focused on

Task	Expected Start Date	Expected End Date	Actual Start Date	Actual End Date
Retrieve Art Interest Points	9/5/2022	9/16/22	9/16/22	10/25/22
Create Mock Map UI	8/28/22	9/14/22	9/10/22	9/16/22
Create Map Template	9/8/22	9/30/22	8/28/22	9/14/22
Create Map Functionalities	9/21/22	11/4/22	8/28/22	10/25/22
Create Database for Interest Points	9/5/22	9/16/22	9/26/22	10/25/22
Add Interest Points to Map	10/24/22	11/10/22	9/26/22	10/25/22

Create Mock Login	8/29/22	9/20/22	9/10/22	9/16/22
Create Interactive Login Page	9/12/22	10/7/22	10/25/22	11/18/22
Include Security System	8/29/22	11/4/22	10/25/22	11/18/22
Put Login/Account Bubble on Map	9/26/22	10/14/22		
Include Interest Point Suggestions	9/26/22	10/21/22	9/10/22	10/25/22
Add Icons to Map	9/3/22	9/10/22		
Create Intractability to Points	10/11/22	10/19/22	9/26/22	10/25/22
Test Suggested Point Addition	10/20/22	10/21/22		
Integrated Details Page	11/7/22	11/18/22	9/26/22	10/25/22
Create Icons for Interest Points	9/29/22	10/5/22		
Automation	10/31/22	12/1/22		
Scroll Style Detail Page	10/24/22	11/4/22	9/26/22	10/25/22
Custom Logo	8/29/22	9/9/22	8/29/22	10/01/22
Testing	10/19/22	11/29/22	8/29/22	11/28/22
Documentation	8/29/2022 10/31/22	9/20/22 12/1/22	8/29/2022	11/28/22
Website Building	8/29/22	9/9/22	8/29/22	11/28/22

Version Control

For version control, GitHub was used. However, due to the large amount of technical conflicts and download issues, our version control was unkempt and out of date. For version control, we ensured to create multiple saves of the software on external devices, such as USB Drives. In addition, the software is stored on a remote server.

Testing Summary

Requirement	✅❌	Summary
Interest Points Show On Map	✅	Interest points submitted into the database show up as a red pin icon with the name
User Creation	✅	User creation is functional using any email
Login/Signout	✅	User can login and sign out as needed, account information is saved
Map UI Is Functional	❌	Map UI is not entirely functional
Database Pulls Data Points	✅	Database pulls all points inserted through MySQL
Home Page Includes All Functionalities	✅	Homepage includes functionalities to view the Marietta Campus, Kennesaw Campus, and User Creation
Security System Detects Correct/Incorrect Information	✅	Security system can detect incorrect login
Pop Up Description Works	✅	Description pops up if double clicked

Conclusion

Overall, our team is proud of the work we have done for this application. We have encountered many issues, but we were able to create a working mobile application compatible with iOS and Android. Our application contains many of our Phase I requirements and some of our Phase II requirements that we were heavily invested in. For the remainder of the semester, we will continue to work on this project until we feel as if there is nothing more to complete. Throughout this course, we learned much about the Software Development Life Cycle, importance of documentation, and how to work as a team to deliver a product in a limited amount of time. 4

Appendix

Image 1: Team Schedule