CS 4720 Final App Documentation

Device Name: Shellder (Nexus 7)   Platform: Android

Name: Jake Wilson   Computing ID: jaw4fz

Name: Stephen Thiringer   Computing ID: sjt7zn

App Name: UViaggio

Project Description:

Our app, UViaggio, provides a way for students to calculate their time navigating between classes to better design their schedule around possible routes between buildings on Grounds. Deciding classes can be a struggle when there are only certain times classes are available, so with this app we are trying to alleviate the concerns of those students who need to travel across grounds to get to their classes so they can more accurately predict what the best schedule design will be for them.

We have created an app that does the following::

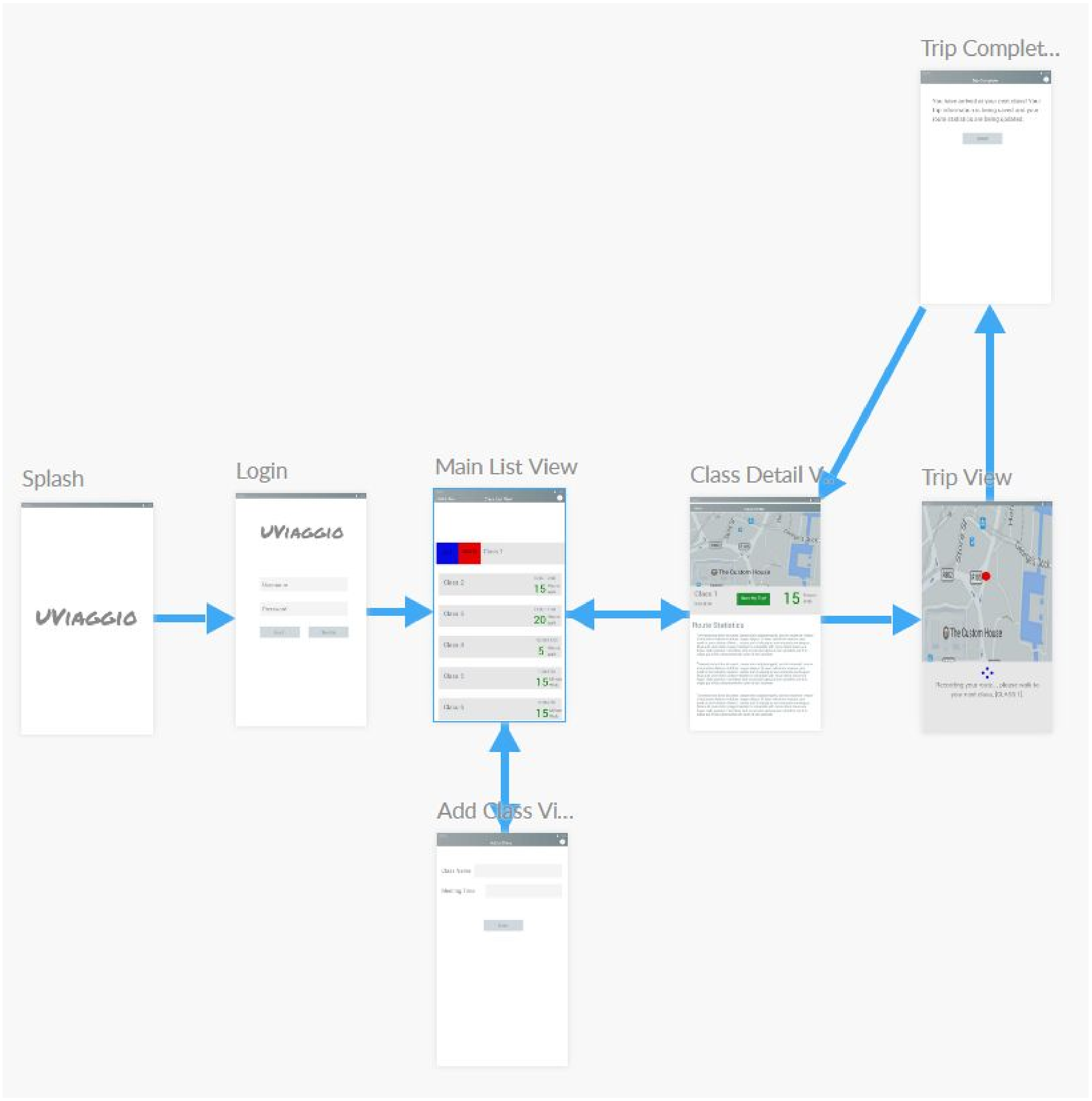
- The system shall allow a student to log in to the service using Firebase
- The system shall allow a student to enter a potential schedule
- The system shall access GPS to to track a students route from class to class across Grounds to more accurately predict their route timing between classes
- The system shall access Lous List API to be able to access the classes and their respective building locations on grounds
- The system shall perform calculations to determine distance between two buildings given two sets of coordinates, and calculate/store time taken and other metrics

We have incorporated the following optional features:

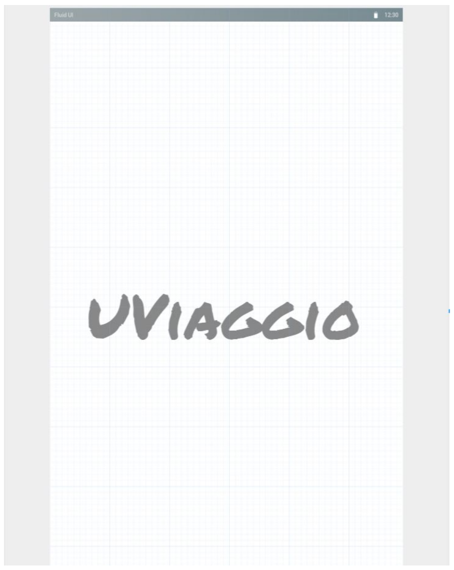
- Gps / Location Awareness - We will let the user indicate when to start tracking them to a destination building from their schedule for that day. Upon arriving, the system will record the time taken to walk to class and cease GPS tracking, notifying that tracking has ended and displaying the adjusted leaving time based on their time taken.
- Consume a pre-built web service - We will consume the Stardock Lou's List API to find latitude and longitude coordinates for building locations. As the API is outdated, we provide a reference list of classes below in the Optional Features description section.

- Data Storage using SQLite - We will store a student's potential schedule using SQLite so they can return and modify their schedule when they need to. Also stored would be metrics regarding a student's routes.
- Build and consume your own web service using a third party platform - We will use Firebase to allow users to setup an account and login.

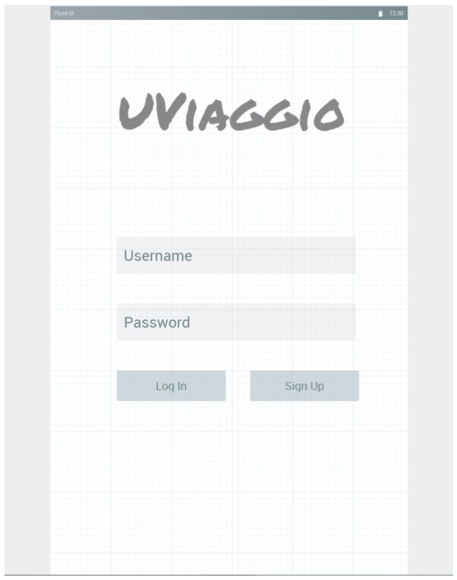
Initial Project Wireframe


Trip Complet...
Splash
UViaggio
Login
UViaggio
Main List View
Class 1
Class 2
Class 3
Class 4
Class 5
Class 6
Class Detail V...
The Custom House
Class 1
Route Statistics
Trip View
The Custom House
Add Class Vi...
Class Name
Meeting Time

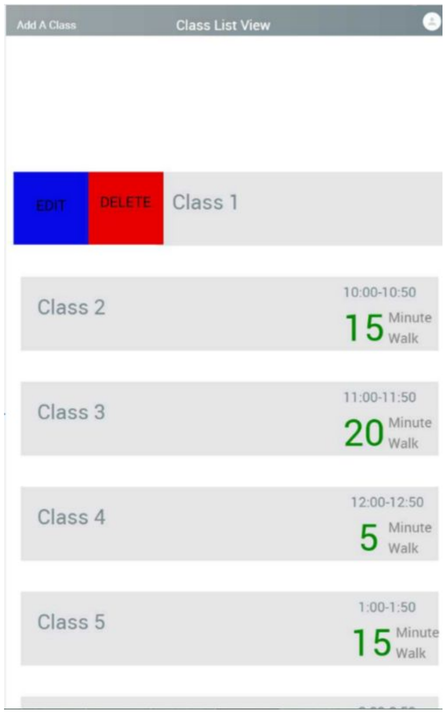
Individual Screens

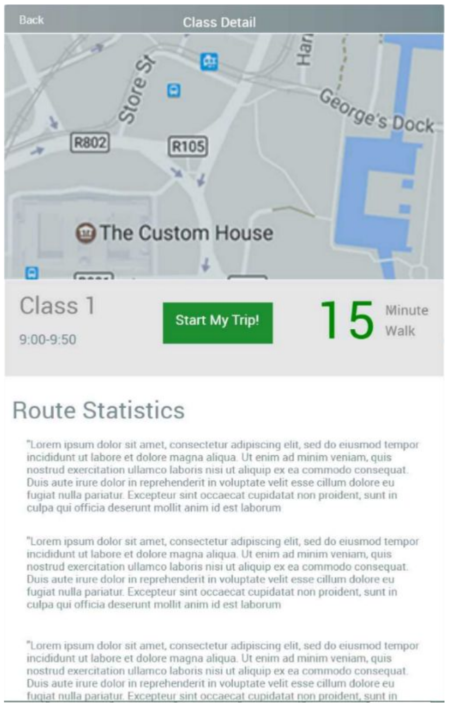


Splash



Login





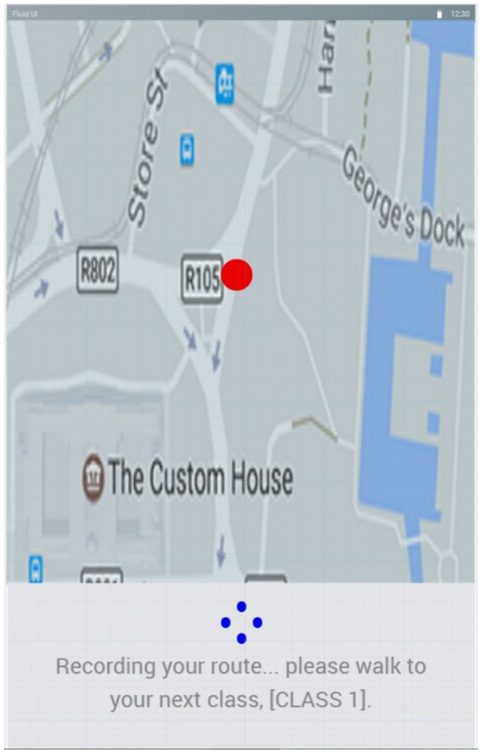
"Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

"Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

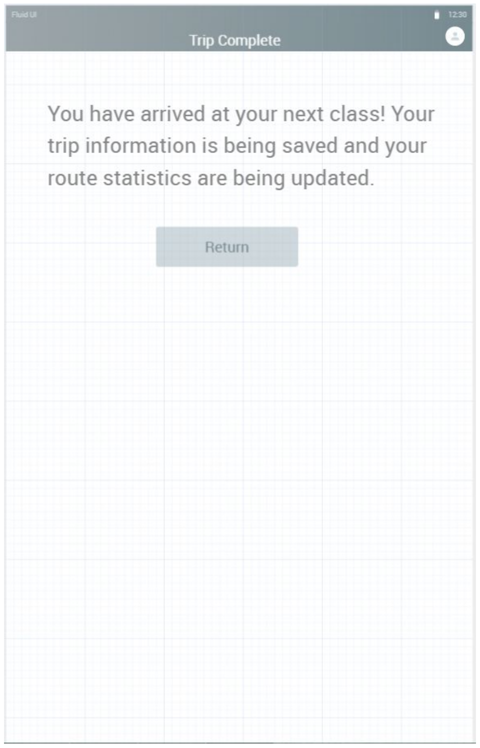
"Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in

Class List View

Route Detail View



Recording your route... please walk to your next class, [CLASS 1].



You have arrived at your next class! Your trip information is being saved and your route statistics are being updated.

Trip View

Trip Complete View

Additional Discussion:

Platform Justification: The benefits of using android for our app include development, language knowledge, more people use android worldwide, and it was easier to implement google services on Android. Neither of us own Macs, so Android was a logistical choice to make it less of a struggle to develop our app. Our combined knowledge of Java and SQL meant that Android Studio came easier to us after our struggles with the mini-app and we were hoping to take the things we learned from the mini-app and apply them to the final app. If we were to release our app onto the Play or App store, we thought about the possible user base and opted to go with Android on sheer amount of users alone. We also knew we wanted to implement

Google Play Services to aid us in our GPS representation and Android is a google platform, so it only made sense.

Major Features and Screens: The login and create account activity are our user validation screens that are connected to our Firebase server. Our Main activity is a recyclerview of cards that contain all of our class information and are auto updated when you add or remove a class from the list or press the begin route button on each and finish our GPS activity. Our GPS activity takes in the coordinates that are passed from the individual class from the recyclerview through an intent and shows a map of your current location with a marker for the location of the building which the class is in. There's a button on the map that starts the tracking functionality and records the length of time it took for you to get from the start location to the destination. Our add class activity takes user input and makes a call to the Lou's list api to obtain all of the information about the class, including its GPS coordinates.

Optional Features: Our login and create account activities use Firebase for user validation. To test this, try to create an account with an invalid email, or a short/nonexistant password. Also, create two separate users and add classes for each of them and swap between the two; their data will remain unique on each account. - 15 points

Our GPS activity tracks your location as you walk to a class and records the time it took. To test this, add a class and begin the activity by pressing the begin route button and hit start tracking. Once you reach the destination the activity will finish and jump back to the recycler view and will have an updated recommended leave time using your average walking time and the time the class starts - 15 points

In order to store classes, we take the values of our class constructor and place them in the SQLite database with each column representing a field of the class. In order to test this, add classes and remove classes (by long clicking the item in the recyclerview) and go between activities or close and reopen the app and the classes will persist - 20 points

Our add class activity uses the Lou's list api and creates classes with the information those json requests provide. In order to test this, add a class using the FAB and see the new class in the recyclerview (note these classes are from the past, since the data in the api is slightly outdated) - 10 points

Reference Class List

- CS 4720 001 - Rice Hall 130
- CS 1111 001 - Mech 205
- CS 1110 101 - Olsson 001
- PHYS 1010 001 - Physics Bldg 203
- CHE 2215 100 - Chem Eng 005
- MUSI 1310 001 - Old Cabell 107

- HIUS 1510 001 - Nau Hall 142

Total = 60 points

Testing Methodologies: For testing the core GPS functionality of our app, we developed it to a state where the app would calculate our location and put in an estimated proximity calculation. Then we walked around grounds testing different proximity calculations until we landed on one we felt matched up with when you feel like you've arrived at a building. For the other general usage parts of our app we did debugging with logs to make sure the functionality of each activity aligned with what we wanted. We tested mostly on the emulator, and after incorporating major features we tested on our hardware device to ensure quality.

Usage Info: You may create an account on the create account screen or you may use our account we used for testing - Username = test@example.com Password = pleasework12345! Be sure to allow location tracking when prompted. To delete a class, press and hold the associated card.

Lessons Learned: We've learned a lot about the mobile development process by doing this project. The wireframe was very helpful in laying out the functionality we were striving to provide in our app and it enabled us to set a goal that was reachable without being too simple. Writing the code inside Android Studio was more intuitive this second time around since we have had more exposure to mobile development code in general. We also learned that it is easier to setup the database with all of the necessary fields after you start writing all of the data passing and manipulation since it's easier to have a better understanding of what data is necessary in order to get the functionality to work properly. Another thing we learned is the feeling of accomplishment when you test the main functionality of your app and it works, the feeling of walking into Wilson Hall and having it finish and update our data was awesome.