

Union College Trolley Tracking System

Grace Yang
CpE-498: Senior Capstone 2020
Advisor: Professor Cotter
11/25/2019

Table of Contents

1	Introduction	1
2	Background	3
2.1	GPS and Tracking	5
2.2	University Transportation Mobile Apps	4
2.3	Real-Time GPS Location Logging Methods	4
2.4	Mobile App Development	6
2.2.1	Native Apps	7
2.2.2	Hybrid Apps	7
2.2.3	Cross-Platform Apps	8
3	Design Requirements	8
4	Design Alternatives	8
4.1	Real-Time GPS Tracking Module	8
4.2	Mobile application design platform	10
5	Design	12
5.1	UI/UX Design Tools	13
5.2	Frontend and Backend Design Tools	13
5.3	Initial App Design	15
6	First Simple App	16
7	Implementation Schedule	17
	References	19

List of Figures

1	Union College Trolley Tracker	2
2	Trolley Stops and Operation Website	2
3	TransLoc Rider Mobile.....	4
4	TanLoc Rider Web	4
5	GDM/GPRS/GPS Module	6
6	GPSlogger.....	6
7	Block Diagram.....	14
8	Mobile App Prototype.....	15
9	Logo Design.....	16
10	First Simple App	16

List of Tables

1	React Native and Flutter Comparison	11
2	Implementation Schedule.....	17

1 INTRODUCTION

Union College trolley service was implemented to provide students with free transportation around the campus Sunday through Thursday 6pm - 2am; Friday and Saturday 6pm - 4am. This system not only makes it more convenient for students to get around campus, it also ensures student safety by providing a safe alternative to walking alone around campus at night. However, in an interview with Union College's Transportation Manager Rich Hewlett, he said "we did a survey this past Spring regarding student usage of the trolley. Disappointingly, the result showed that there is only an average of zero to one student each day who takes the trolley." (R. Hewlett, personal communication, Oct 2, 2019). It is obvious that only a small number of students uses the trolley due to the underdeveloped implementation of the system. The goal of my project is to develop a new trolley tracking system that is cost-effective, efficient, and user friendly.

The main reason why the trolley service is rarely used by students is that the current trolley tracking system is poorly designed. The only way for students to track the trolley is through a trolley tracking map (Figure 1) provided on the Trolley Stops and Operation website (Figure 2). The only information the tracking map provides is the real time location of the trolley and the time the trolley spent on the road or stopping. It does not specify the time it takes for the trolley to go from one location to another or the stopping time at each location. Therefore, although it is helpful to know the real time location of the trolley, it is hard for the users to estimate the arrival time at the pick-up or drop-off locations. Sometimes the trolley would take detours to other locations other than the listed pick-up locations, which makes it even harder for students who are waiting for the trolley to know whether the trolley will be arriving or not. Another reason why the current trolley system is inconvenient is that the trolley map is only available online or through the Union College Mobile

Application. First, it is inconvenient for users to go on the Trolley Stops and Operation website every time they need to use the trolley. Secondly, the Union College Mobile Application is not widely used by students.

The current trolley tracker is supported by a commercial vendor who tracks the movements of the trolley via a GPS unit in the trolley and displays the current trolley location on a map. The problem with this is that the cost of service is \$29 per month (R. Hewlett, personal communication, Oct 2, 2019), which is relatively high. An alternative GPS tracking system with reduced cost could be developed.

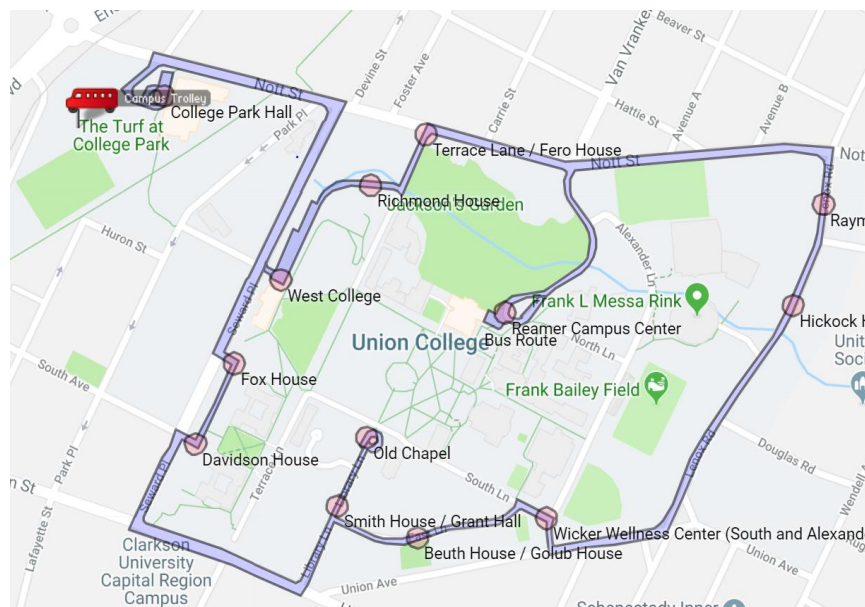


Figure 1: Union College Trolley Tracker [1]

TROLLEY STOPS AND OPERATION

The trolley runs in a continuous route "clockwise" on and around the campus (each trip takes approximately 30 minutes). A trolley will arrive roughly every 30 minutes, except during required driver breaks. Service is 6 pm to 2 am daily during the academic term.

To further serve the Union College community, a dispatcher can be reached by dialing (518) 248-5111. The phone will alert you to delays. Pickups will be made at designated locations only – these are marked with a "T."

The trolley begins its route at College Park Hall.

Stops on Trolley Route

College Park Hall

West College

Richmond House

OFFICERS ARE ON DUTY 24/7

Trolley tracker

Dispatch: (518) 248-5111



Figure 2: Trolley Stops and Operation Website [1]

My project will address these aforementioned problems by redesigning the Trolley Stops and Operation Website and creating a mobile application that is user friendly. The features of the website and the mobile application will include the real time location of the trolley and the users; the arrival and departure time of the trolley at each stop; notifications that inform users of the trolley's status; and an interface that informs the driver of the number of users at each location.

2 BACKGROUND

2.1 GPS and Tracking

The Global Positioning System (GPS) is a satellite-based navigation system that consists of at least 30 satellites circling Earth, ground stations, and receivers. GPS satellites circle the Earth in precise orbit and continuously emit coded high-frequency radio signals that are received and computed by the receivers. These signals contain information about the exact orbits of the satellites and the time of the atomic clocks on the satellites. The ground stations use radar to make sure the satellites are at the correct locations. When signals from three or more satellites are received, the GPS receiver

will compute the location of the user. Once the user's position is determined, the receiver can calculate other information such as speed, bearing, track, sunrise and sunset time.

2.2 University Transportation Mobile Apps

Most universities provide students with public transportations to ensure student safety, reduce on-campus traffic, and provide a cost-effective way of traveling between classes and dorms that is time optimized. Considering transportation apps such as Uber and Lift, universities started taking initiatives in revolutionizing their public transportation services through mobile apps that turn on-campus public transportation into a more user-friendly and on-demand service.

University of Kentucky developed a mobile app called TransLoc Rider in 2011 that offers free iPhone and Android apps for real-time GPS tracking of on campus buses as shown in figure 3 [2]. The app includes features such as the ability to select routes or individual stops, save user's favorite stops, provide feedback on user's experience, and get notified of an estimation of the arrival and departure time to and from each stop [3]. In 2015, 21,708 users used the app to track campus buses. This was a promising result. Other universities such as Duke University, Binghamton University, and Yale also uses similar on-campus transportation apps.

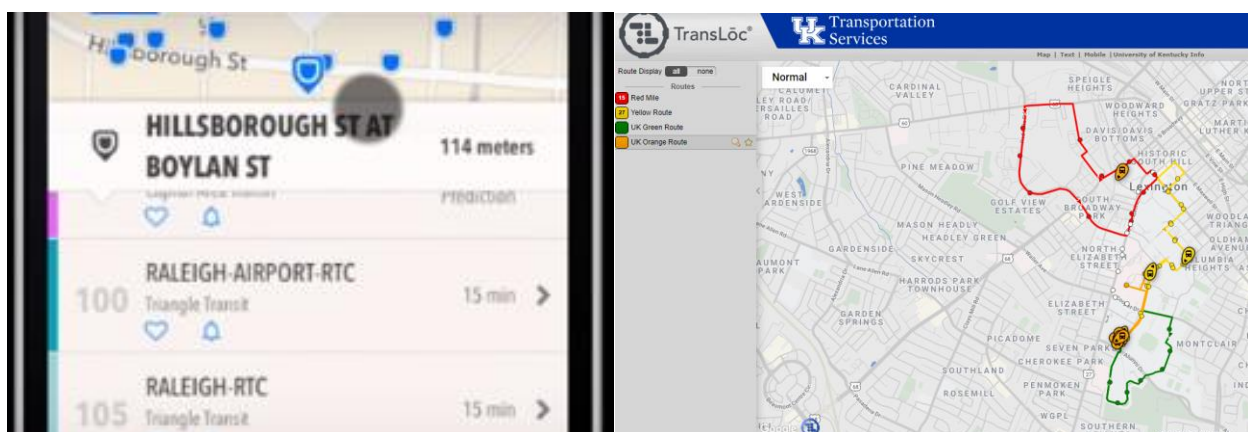


Figure 3: TransLoc Rider Mobile[2]

Figure 4: TransLoc Rider Web [2]

2.3 Real-Time GPS Location Logging Methods

There are two parts to storing GPS data: obtain and analyze the real-time GPS data, and importing the data to a mobile application and website. To obtain the real-time GPS data, we could either achieve it using a GSM/GPRS module (Figure 5) or using a mobile phone. Some conventional ways of using mobile phone to track and log GPS data are using Android apps to extract data, using the API of Google Location Sharing to retrieve the location history of the phone, and using Find My Device service on either an iPhone or Android. There are many existing research papers on implementing real time GPS tracking via a mobile phone. Among them, the Chittagong University's bus tracker project stood out to me. The system monitors the buses and updates the changes of their status via android application. The app in the vehicle receives the location from the satellite and transmits it to a software called Firebase. Firebase updates the data every few milliseconds and transmits it to the users. This method is proven to be cost-effective, easy to maintain, and provides faster location updates [4]. There are many existing free Android apps on the market that logs the GPS data and relation information of the phone. For example, an app called GPS logger logs the phone's coordinates to GPS format files at regular intervals (figure 6) [5]. To track the real-time location of the trolley, an Android phone with a GPS logging app running in the background could be mounted on the trolley. This method is effective as long as the phone is charged and has a stable service.



Figure 5: GDM/GPRS/GPS Module [6]

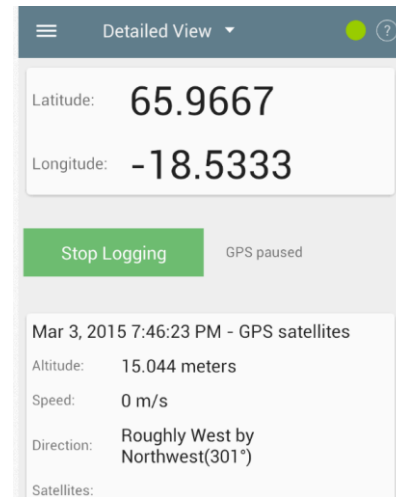


Figure 6: GPSlogger [5]

2.4 Mobile App Development

App development is normally referred to developing for mobile devices such as smartphones, phablets, and tablets. Most people's mobile apps run on iOS and Android. As of August 2019, Android controls around 51.8% of the mobile device market and iOS owns 47.4% [7]. iOS is developed and supported by Apple whereas Android is developed and supported by Google. Android is an open source operation system, which means it can be used by other device makers and it is easier to make alteration to the system. On the other hand, iOS can only be used by Apple and it is fairly restricted. Unfortunately, native iOS apps and Android apps cannot work together. They are built separately with different tools and source codes. This raises the question of how we could build an app that works for both iOS and Android? To answer this question, we need to understand the three main types of mobile apps: native app, hybrid app, and cross-platform app. The very first decision that an app developer will face is which type of app to build. There are benefits and disadvantages to these different types of mobile apps and there is no clear winner. When deciding about which type to choose, our choice depends different factors: the budget, the

size of the app, performance requirements, deadline, etc.. It is important to first have a thorough understanding of these different types of apps.

2.4.1 Native App

A native app is a software application built for specific device platform using different programming language and SDK. For example, Swift or Objective-C is used for iOS and Java is used for Android. iOS uses XCode as its development tool whereas Android uses Android Studio. Therefore, a native app of iOS can't be used on an Android device and vice versa. This introduces us to the disadvantages of developing a native app: We need different source code for iOS and Android, it has a higher cost of development, it takes longer time to build, and we need to use specific SDK for the development process. Despite its disadvantages, it has obvious benefits as well: a native app has fast response time, wider device functionality, more robust, and matching UI/UX to platform conventions [8].

2.4.2 Hybrid Apps

An alternative to native apps is hybrid apps, which compensates the biggest disadvantage that native apps have: different source codes for different software applications. A hybrid app is like a web app in disguise. We can install it like a native app, but the core of it is a web app. It is built with HTML, CSS, and JavaScript. Like a web app, it also run in WebView, a simplified browser within the app. The obvious advantages of a native app are: one codebase across all platform, budget and time friendly, have a sufficient amount of access to device features, time and cost efficient [9]. With advantages comes its disadvantages: it does not perform as well as native app, it takes longer to load, it has limited capabilities due to the need to use plugins to access the built-

in features of a device, and poor user experience [10]. Some examples of hybrid app platforms are PhoneGap/Cordova, Canvas, Ionic, and OnsenUI.

2.4.3 Cross-Platform Apps

Like hybrid apps, cross-platform apps enable developers to create applications that work across platforms. They also enable companies develop time and cost-effective app that are stable, easy to maintain, and stable. However, they are unable to offer as good of a user experience and performance as native apps [11]. Some cross-platform development platforms are Xamarin, React Native, Titanium, and Flutter.

Since the main users of the trolley mobile app that I will develop are students on campus and most of them use both iPhones and Android phones. I wish to develop an app that could be used on both devices. Since this is an individual project and I only have two trimesters to complete it, developing natives apps might not be time efficient or cost effective. In addition, the app that I will develop only require basic user interface and simple graphics. That means developing a cross-platform app could be the best choice for this project.

3 DESIGN REQUIRMENTS

Performance

- The trolley tracking mobile app will be able to provide real-time location of the trolley
- The mobile app will efficiently notify the driver the location of the students and the number of students waiting at each location. It will notify the students whether the trolley has reached its capacity, the arrival time of the trolley to the desired location, the waiting time

of the trolley at each location, and the next location the trolley is heading. It will also notify the students when the trolley takes a detour somewhere off campus and an estimated arrival time of the trolley to campus.

- The mobile app will be downloaded by at least 50% of the students on Union campus

Cost

- The cost for purchasing software tools should be within \$100
- The cost for purchasing an Android phone should be under \$50
- The cost for publishing the app on the Apple App Store and Google Play Store should be around \$125
- The app will be free for all users

Aesthetics

- The website should have a smooth flow, pleasant color, and good balance
- The images on the website should send the right signals
- The design should be creative and simple

Maintainability

- The code for the app is well documented, bugs are easy to fix
- The software is easy to understand, easy to target what need to be changed, easy to make changes, and easy to check that the changes have not introduced additional bugs
- The software system can adapt to a changed environment

4 DESIGN ALTERNATIVES

4.1 Real-Time GPS Tracking Module

As mentioned in the background section. There are two ways of logging real time GPS data. One is to place an Android phone in the trolley and extract the GPS coordinates of the phone via an existing mobile application. The alternative method is to design a simple GPS tracking and data logging device. This could be accomplished by using a TinyDuino Processor Board paired with a GPS TinyShield. The GPS data could then be processed and monitored using Arduino IDE [12]. This design is easy to implement, and the components are low cost. However, this system might be too fragile for the long term use on a trolley. A more robust design could be implemented with a GM862-GPS cellular quad band module and a microcontroller [13]. The GM862-GPS is a satellite-based service that combines GPS and GPRS to achieve a more accurate measurement of the real time tracking. A microcontroller will be used together with the GM862-GPS to activate and analyze the GPS data [14]. Although implementing GPS/GPRS module is doable, it is less time efficient compared to using an app on the Android phone. It also requires more work for hardware maintenance. Therefore, using an Android app to log real-time GPS data is a more desirable choice for my project.

4.2 Mobile application design platform

As mentioned in the background, there are three types of mobile apps: native apps, hybrid app, and cross-platform apps. After thinking through the pros and cons of each type of app, I decided that developing a cross-platform app is most suitable for my project. There are two major software development kits for cross-platform apps: React Native and Flutter.

React Native is backed by Facebook and has been used by Instagram, Facebook, Airbnb, Walmart, Tesla, UberEATS, etc. [15] Flutter is backed by Google and has been used by Alibaba, AppTree,

Tencent, Google Ads, etc. [15] Both software development kits share some similarities despite their differences.

Both React Native and Flutter are backed by giant tech communities: Facebook and Google. Both are open source and free SDKs. Both have strong communities keeping documentation and comprehensive resources up-to-date. Although their performance can't be compared with that of native apps, they have their own ways of utilizing fundamental UI building elements and widgets to provide nearly native experiences. Another feature that they both share that's worth discussing is the hot reload function. The hot reload function allows the code to be simultaneously reflected on the console [15]. Although they share many similarities, there are major differences between them as well.

React Native uses React library and JavaScript, whereas Flutter uses a programming language called Dart, a language developed by Google that is object-oriented and modern. React Native has a stronger and more mature community support compare to Flutter because Flutter is relatively new. However, Flutter offers extremely well documented resources and support backed by Google to compensate this shortcoming. In terms of front-end support, React Native uses native components and achieve attractive UIs, whereas Flutter uses built-in widgets to achieve attractive UIs. Below is a table comparing these two development kits (Table 1).

	Flutter	React Native
Language	Dart	JavaScript
Developers	Google	Facebook
Initial Release	2017	2015

Native Performances	Superior	Good
Time-to-Market of App	Faster	Slower than Flutter
Documentation	Precise, clear, and up-to-date	Up-to-date and imprecise
Who Uses	Alibaba, Reflect, Google, Greentea, Tencent, Google Ads, App Tree	Facebook, Instagram, Pinterest, Uber, Tesla, Walmart, Wix.com, Baidu Mobile, Artsy
Competitive Advantages	Simpler, faster, and elegant native apps using easy-to-use Flutter SDK	More than 3 years in the market with the ability to deliver native-like app experiences

Table 1: React Native and Flutter Comparison [16]

After examining the pros and cons of each software development kit, I decided to use Flutter for my project instead of React Native for the following reasons: Elegant and object-oriented programming language, excellent and complete documentation and resources by Google, and near native user experience.

5 DESIGN

The mobile application for the Union College trolley service is called Trill. It contains functionalities such as real-time GPS tracking, arrival and departure time of trolley to and from each station, push notifications, daily trolley schedule display, and the head count of the number of students at each stop. As mentioned in section 4.1, the real-time GPS tracking of the trolley is

done by placing an Android phone in and trolley and use the GPSlogger Android app on the phone to record the GPS data (Figure 7 (1)). The data is then used in the development of Trill. The development of Trill includes UI/UX design, frontend design, and backend design. The software tools used for UI/UX design are Sketch, inVision, and Adobe Illustrator. The software tools used for frontend and backend design are Flutter and Android Studio. The users could download the app from either Apple App Store or Google Play Store for free. The Trill app is not only informative, but also interactive. When users are at a certain stop, they could use the app to inform the drivers which stop they are at and the driver will get notified of the number of students at each stop (Figure 8 (3)).

5.1 UI/UX Design Tools

Sketch is a digital design application used for UI/UX, mobile, web and icon design. inVision is a prototyping tool that allows users to create mockups for their designs and can be easily shared with their team or clients. Sketch and inVision are usually used together to create well designed prototypes with customized graphics. Adobe Illustrator is a vector graphics editor used to create illustrations and digital artworks. I use Sketch and inVision for the prototyping of the Trill app and Adobe Illustrator for the logo design.

5.2 Frontend and Backend Design Tools

As mentioned in section 4.2, I chose Flutter to be the software development kit for the frontend and backend design of Trill. In order to use Flutter, I need to install Android Studio as well because Flutter cannot run on its own, it serves as a plugin on Android Studio. Android Studio is an

integrated development environment for Google's Android operating system and is usually used for developing native Android applications.

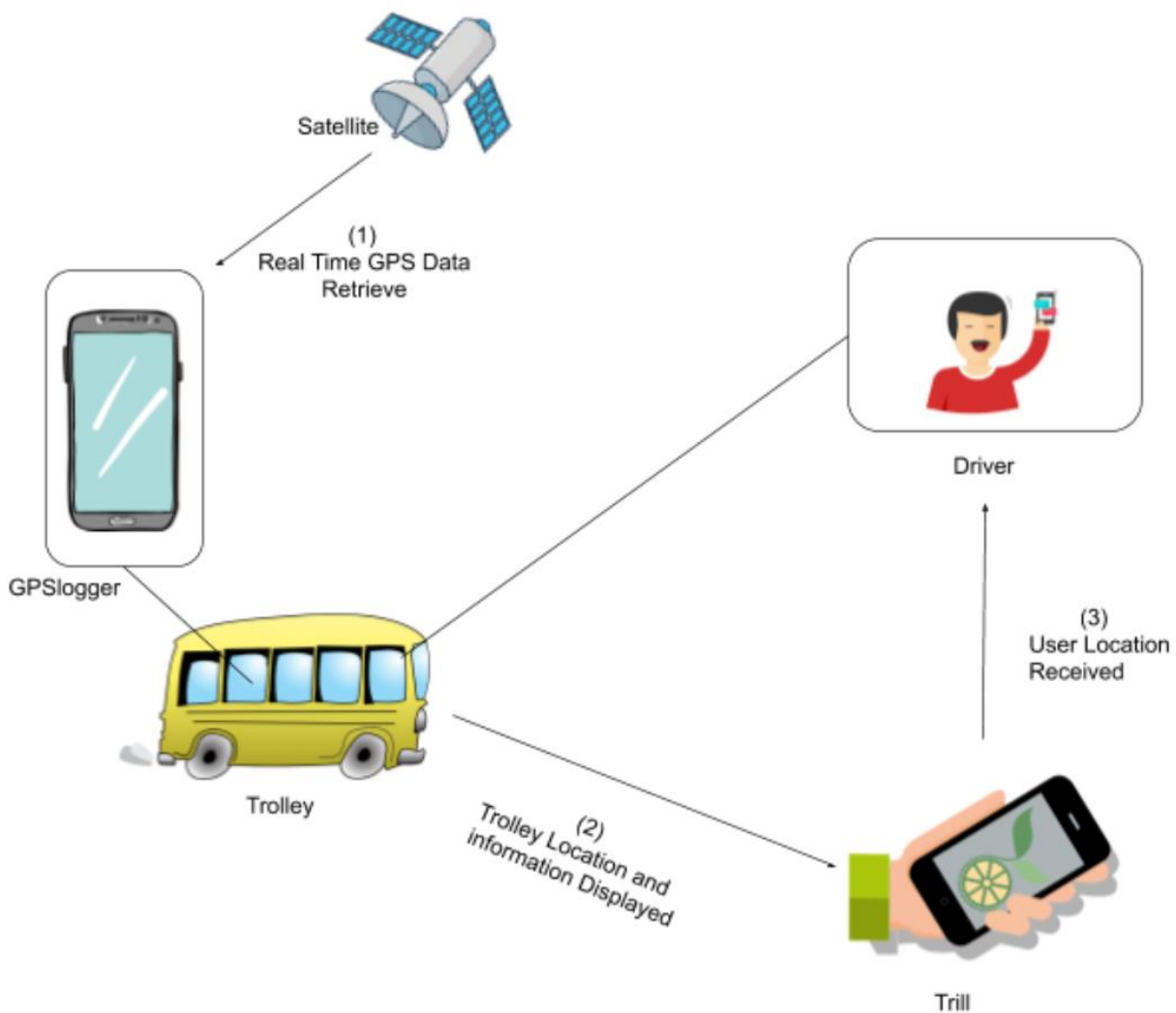


Figure 7: Block Diagram

5.3 Initial App Design

The mobile app prototype is shown in Figure 8. Upon opening the app, a map with the trolley's real time location shows up (Frame 1). If the users swipe up from the bottom of the screen, they could be brought to two different screens depending on the status of the trolley. If the trolley is off

campus, the users will see an informative message (Frame 3). If the trolley is on campus, the users will see the different stops and the trolley's arrival time to each stop (Frame 2). If the trolley is currently at the stop, its departure time will be displayed. The users could click on each stop to see more information about the trolley (Frame 4) such as whether the trolley has reached its full capacity or not. If the users swipe right from the left edge of Frame 1, the users will see the daily schedule of the trolley (Frame 5). If the users swipe left from the right edge of Frame 1, they will be brought to Frame 6 which shows the number of students at each stop.



Figure 8: Mobile App Prototype

6 FIRST SIMPLE APP

Using Adobe Illustrator, I have designed the logo for the trolley app named "Trill" (Figure 9). Using Flutter and with the assistance of an online Flutter course on AppBrewery, I have designed a simple app with one image in it (Figure 10). When you click on the app icon, an image of Trill's logo will show up. For the design process of the first simple app, I first downloaded Android Studio on my laptop and set up Flutter as a plugin to Android Studio. Then I used the Dart programming language to program the app. While programming the app, I opened an Android screen emulator to see how the app would look like on a real phone. Any update in the code will be reflected on the emulator with a 2 to 3 seconds delay. This functionality of Flutter is helpful because it allows me to make quick adjustment to my code without wasting time on waiting for the results.



Figure 9: Logo Design



Figure 10: First Simple App

7 IMPLEMENTATION SCHEDULE

Week	Tasks
Week 1	<ul style="list-style-type: none">• Collect real time GPS data and import it to Flutter• Start redesigning the first frame of the prototype on Sketch and inVision
Week 2	<ul style="list-style-type: none">• Program the first frame of the prototype on Flutter• Redesign the second frame of the prototype on Sketch and inVision
Week 3	<ul style="list-style-type: none">• Program the second and third frame of the prototype on Flutter• Redesign the third and fourth frame of the prototype on Sketch and inVision
Week 4	<ul style="list-style-type: none">• Program the fourth and fifth frame of the prototype on Flutter• Redesign the fifth and sixth frame of the prototype on Sketch and inVision
Week 5	<ul style="list-style-type: none">• Program the sixth frame of the prototype on Flutter.• Test the finished app

Week 6	<ul style="list-style-type: none"> • More user testing of the app
Week 7	<ul style="list-style-type: none"> • Implement the redesigned Sketch prototype to on Flutter
Week 8	<ul style="list-style-type: none"> • User testing of this new version • Publish it on Google Play Store
Week 9	<ul style="list-style-type: none"> • Gather user feedback
Week 10	<ul style="list-style-type: none"> • Final Report

Table 2: Implementation Schedule

REFERENCE

- [1] "Trolley Stops and Operation." *Union College*, www.union.edu/campus-safety/trolley-stops-and-operation.
- [2] "Bus Tracking App Helps Keep Campus Moving." *UKNow*, 29 Mar. 2016, <https://uknow.uky.edu/campus-news/bus-tracking-app-helps-keep-campus-moving>.
- [3] "Features." *TransLoc Rider*, <http://translocrider.com/features#howitworks>.
- [4] Alamgir, Mohammad & Jahan, Israt & Aktar, Nasrin. (2019). Chittagong University Teachers' Bus Tracking System Using Smartphone Application. 10.1109/ICEEICT.2018.8628098. https://www.researchgate.net/publication/330937599_Chittagong_University_Teachers'_Bus_Tracking_System_Using_Smartphone_Application
- [5] "GPSTracker for Android." *GPSTracker for Android*, <https://gpslogger.app/>.
- [6] "SIM808 GSM/GPRS/GPS Module." *SIM808 GSM/GPRS/GPS Module - ITEAD Wiki*, https://www.itead.cc/wiki/SIM808_GSM/GPRS/GPS_Module.
- [7] "Mobile OS Market Share in the U.S. 2019." *Statista*, <https://www.statista.com/statistics/266572/market-share-held-by-smartphone-platforms-in-the-united-states/>.
- [8] Saccomani, Pietro. "Native, Web or Hybrid Apps? What's The Difference?" *MobiLoud Blog*, MobiLoud, 4 June 2019, <https://www.mobiloud.com/blog/native-web-or-hybrid-apps/>.
- [9] Panchal, Mayur, et al. "These Hybrid App Examples Will Transform Your Perspective Towards Mobile Apps." *Excellent WebWorld*, 21 Nov. 2019, <https://www.excellentwebworld.com/hybrid-app-examples/>.
- [10] K., Vitaly, and Vitaly K. "Native vs. Hybrid App: Considering Pros and Cons of Each Option." *Cleveroad Inc. - Web and App Development Company*, 11 July 2018, <https://www.cleveroad.com/blog/native-or-hybrid-app-development-what-to-choose>.
- [11] "What Is Cross-Platform Development? Definition & More." *Sapho*, <https://www.sapho.com/glossary/cross-platform-development/>.
- [12] Instructables. "Tiny GPS Tracker." *Instructables*, Instructables, 19 Sept. 2017, <https://www.instructables.com/id/Tiny-GPS-Tracker/>.
- [13] "GM862 Cellular Quad Band Module with GPS." *CEL-07917 - SparkFun Electronics*, <https://www.sparkfun.com/products/retired/7917>.

- [14]. W. El-Medany, A. Al-Omary, R. Al-Hakim, S. Al-Irhayim and M. Nusaif, "A Cost Effective Real-Time Tracking System Prototype Using Integrated GPS/GPRS Module," *2010 6th International Conference on Wireless and Mobile Communications*, Valencia, 2010, pp. 521-525.
- [15]. Spec. "Flutter Vs. React Native: Detailed Comparison, Similarities, And Superiority." *Medium*, Codeburst, 28 Dec. 2018, <https://codeburst.io/flutter-vs-react-native-detailed-comparison-similarities-and-superiority-3e92b910fa6e>.
- [16]. Spec. "React Native vs. Ionic vs. Flutter: Comparison of Top Cross-Platform App Development Tools." *Medium*, Codeburst, 5 July 2019, <https://codeburst.io/react-native-vs-ionic-vs-flutter-comparison-of-top-cross-platform-app-development-tools-71c8011309ac>.