Garbage Route Optimizer

Genesis Munoz

SUBSYSTEM REPORT

FRONTEND SUBSYSTEM

REVISION – Original

28 April 2021

SUBSYSTEM REPORT
FOR
Garbage Route Optimizer

TEAM 3

APPROVED BY:

Genesis Munoz	4/28/2021
Prof. Tyler Prof. Nowka	4/28/2021
Pranav Dhulipala	4/28/2021

Change Record

Rev.	Date	Originator	Approvals	Description
1	4/28/2021	Genesis Munoz		Final Release

Table of Contents

List of Tables

List of Figures

1. Subsystem Introduction

The purpose of the frontend subsystem is to provide the only means of interaction between a user and the information pertaining to the locations of bins a user will have to service. Through use of an iOS application it effectively showcases the information necessary for a user to navigate through a city, while at the same time being easy and safe to use while a user is driving. The application has three total pages for a user to view, a home page, search page, and navigation page.

1.1 Operational Description and Constraints

The user application was made to operate along with an internet connection. Therefore, a user must have a steady connection to the internet for the application to work. This user application was developed for iOS devices, therefore an iOS device will be required for users to be able to get directions as no Android version exists yet.

2. Design

The user application can be split up into three major components: Home Page, Search Page, and Navigation Page. The initial GUI sketch for the three components is depicted below:

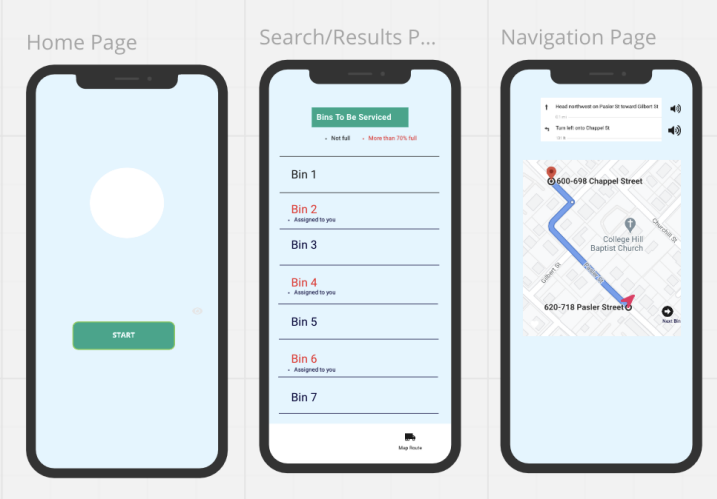

Figure 1: GUI Sketch for full application

The development of this sketch was made using Miro, the development of each of the pages into pure Swift will be explained in the following subsections.

2.1 Home Page

The Home Page, as seen in the aforementioned Figure 1, provides a simple and friendly first impression to a user. This page is meant to be used purely as a segue to the next page in the application, the search page. Upon the tap of the start button, a user will be redirected to the search page.

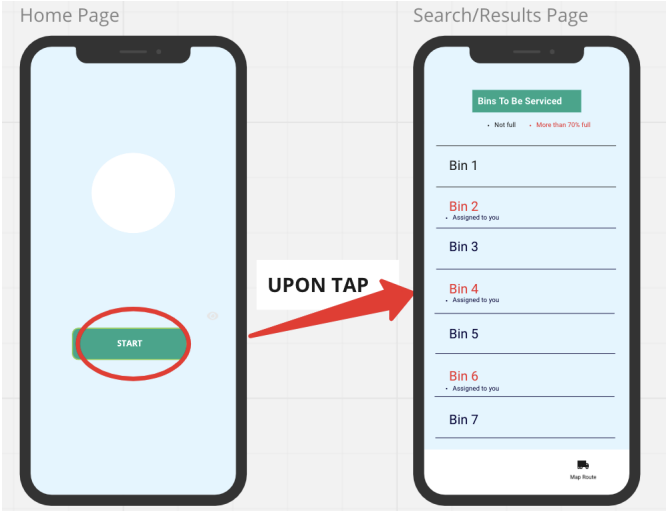

Figure 2: Transition to Search Page

As seen in Figure 2 above, upon tapping the start button a user is immediately redirected to the Search Page so he or she may proceed with their route.

2.2 Search Page

The Search Page, as seen in Figure 2, will provide the user with the respective information pertaining to trash bins in their area. Since there's no backend connected to this subsystem, the information displayed regarding bin capacity is composed of dummy values, to only showcase the functionality of the page. A user will be able to see and scroll through the total amount of bins that need servicing that day. He or she will also be able to see which bins correspond to them to service that day, as depicted by the subtitle under bins. Bins denoted with a red font will be the bins that are above seventy percent capacity, the rest will be displayed as not full.

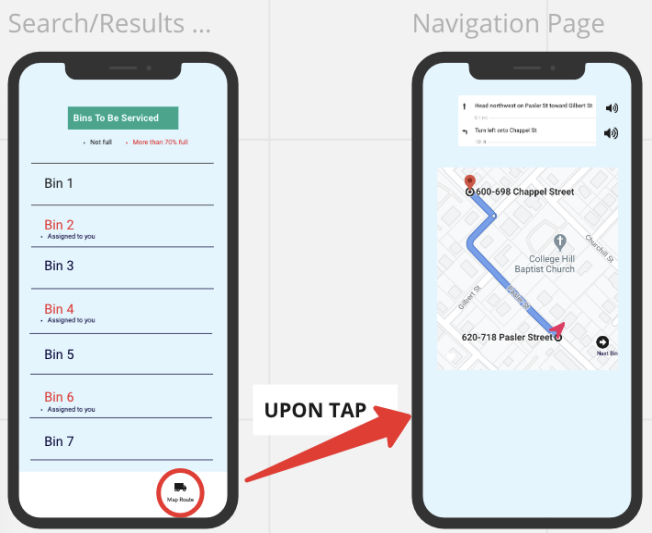

Figure 3: Transition to Navigation Page

As depicted in Figure 3, once a user has inspected the bins corresponding to them they will be able to press the "map route" button and begin navigating to their first bin destination.

2.3 Navigation Page

The Navigation Page, as seen in Figure 3, will provide directions to users so they may service all the bins that have been assigned to them that day. Two forms of displaying directions will be made available to a user, directions displayed in text and directions narrated through the use of a speech synthesiser. Upon servicing one bin, a user will be able to press the “Next Bin” button, as seen in Figure 4 below, to get directions to the next bin in their route from their current location.

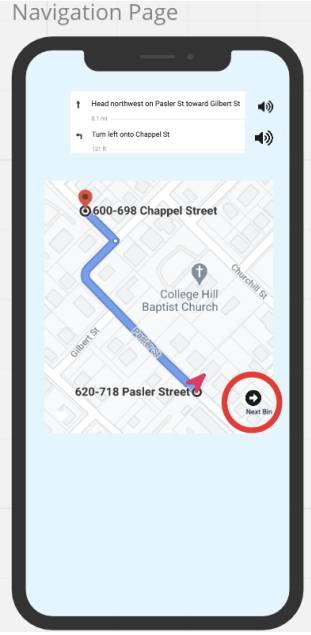

Figure 4: Next Bin button in Navigation Page

3. Implementation and Operation

The three components of the user application were built using pure Swift through an Xcode editor, making use of Apple's MapKit API in the Navigation Page. All design specifications were met when building the Home, Search, and Navigation pages. In the following subsections you will find comparisons between the design sketch of each page and it's resulting version.

3.1 Home Page

The resulting Home Page was rendered and tested on an iPhone 8 model, not an iPhone 11 as depicted in the GUI sketch. Colors in the resulting version were made more vibrant and a logo was included to bring about a more complete look to the page, along with a welcome message for the user to feel more comfortable using the application.

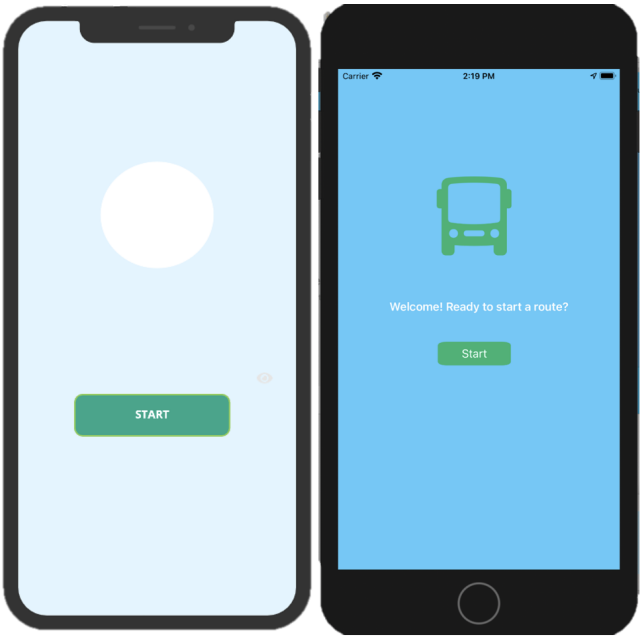

Figure 5: Design Home Page Sketch (left) vs Resulting Home Page (right)

Start Button: The main functionality of this page is to provide a transition to the search page, this was accomplished by making use of the start button. This button was wired up inside the Xcode storyboard to segue over to the Search Page.

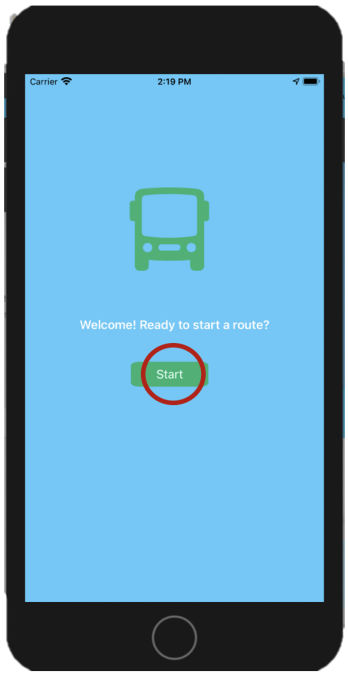

Figure 6: Start button on Home Page

3.2 Search Page

The resulting Search Page was rendered and tested on an iPhone 8 model, not an iPhone 11 as depicted in the GUI sketch. Inside the Search Page, a user is able to scroll through the bins that are in need of servicing that day. The functionality of going back to the Home Page is very obvious to the user in this and the upcoming navigation page, as a user can just drag down the page to go back to the previous. The integrity of the original design is kept with the showcasing of bin capacity using red font and a default capacity for bins denoted as full. The user is told by the subtitles under some bins, which will correspond to them to service that day.

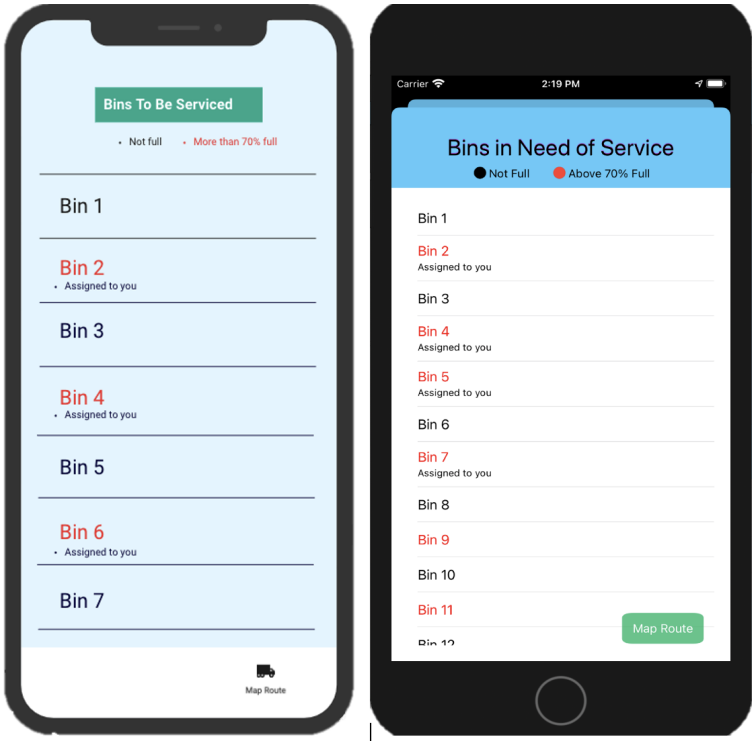

Figure 7: Design Search Page Sketch (left) vs Resulting Search Page (right)

Map Route Button: Apart from showcasing bin information to the user, the Search Page provides a button to take them to the Navigation Page so that they may be routed to the location of the first bin they should service. This button, much like the start button from the Home Page, was wired up in Xcode storyboard to segue over to the Navigation page.

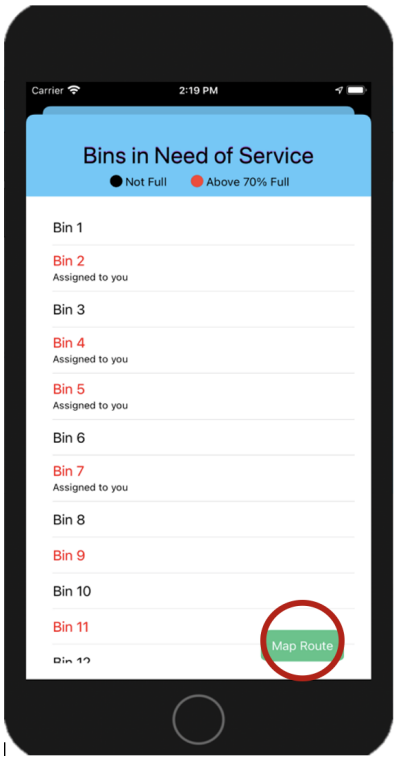

Figure 8: Map Route button on Search Page

3.3 Navigation Page

The resulting Navigation Page was rendered and tested on an iPhone 8 model, not an iPhone 11 as depicted in the GUI sketch. When a user is on the Navigation Page, he or she will immediately receive directions both via text on the screen and via speech synthesizer to get to their first bin. Since there's no backend connected to this subsystem, set coordinates were instantiated to represent the total bins a user would have to service that day. From the list of set bin coordinates, in order to get directions from one bin to another, I made use of the MapKit API to send direction requests for each coordinate set and have them be outputted in the top label of the screen. At the same time, directions are narrated to a user with a speech synthesizer.

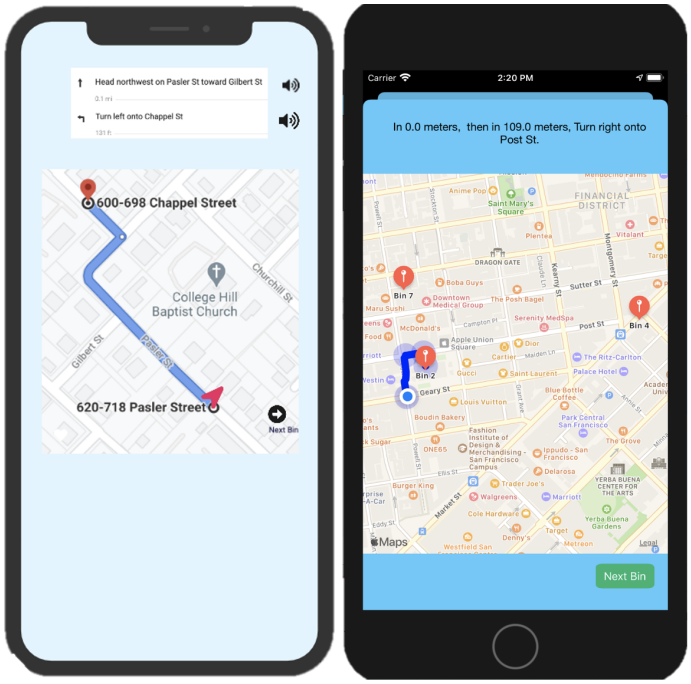

Figure 9: Design NavigationPage Sketch (left) vs Resulting Navigation Page (right)

User Location: Through the usage of the Mapkit API, the user location is displayed and allowed access to from the permissions enabled through Xcode. Prior to showing the map to the user, the screen will ask for permission to make use of the user's location. This prompt will always show up, unless the option for "Allow While Using App" is selected by a user. For the purpose of demo the starting location of the driver is a predefined set of coordinates inside Swift code, and the simulation will always start off at that same location, which is why the simulation is also taking place in San Francisco, CA.

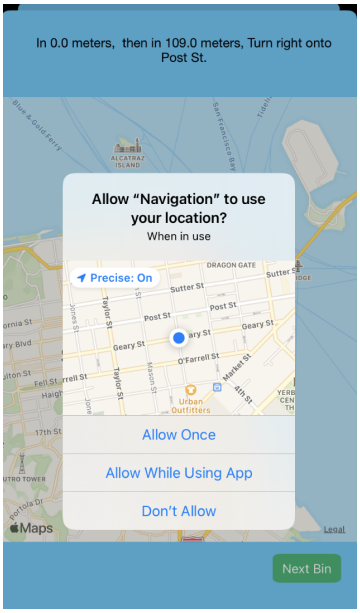

Figure 10: Allow for usage of user location prompt

Next Bin Button: Once a driver has successfully serviced a bin, tapping the “Next Bin” button will give them directions to the next bin in their route. Now, since the MapKit user location is read-only, there wasn’t a way to simulate the user (blue dot on screen) to move along the route. What was done to best show off the route, was adding a blue circle of a small radius on every turn the user would have to make to get to their destination. Upon completion of more bins, the blue line will grow and represent the full route when a driver has serviced all of their bins. The final route displayed is meant to represent information that would be sent back to a database, were this subsystem connected to one. It is displayed to give a more complete and functional look to the page.

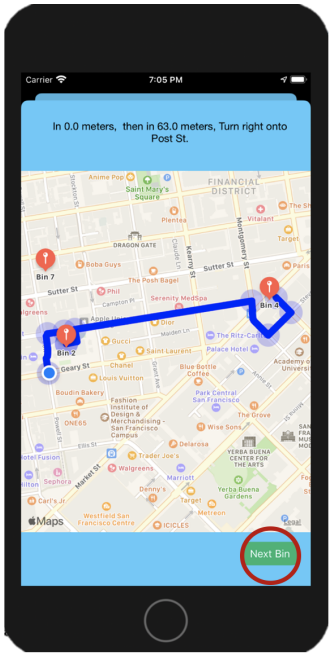

Figure 11: Navigation Page upon tapping of Next Bin button

4. Validation Plan and Execution Plan

4.1 Performance on Execution Plan

The execution plan depicted in the table below has been executed completely. There were some tasks that overlapped at times with others due to conflicts involving workload from other classes, and delays to get the right functionalities working. Despite this the execution plan was completed on time before any testing on the subsystem was conducted. The subsystem was completed in full, despite changes stemming from not having a backend to connect to anymore.

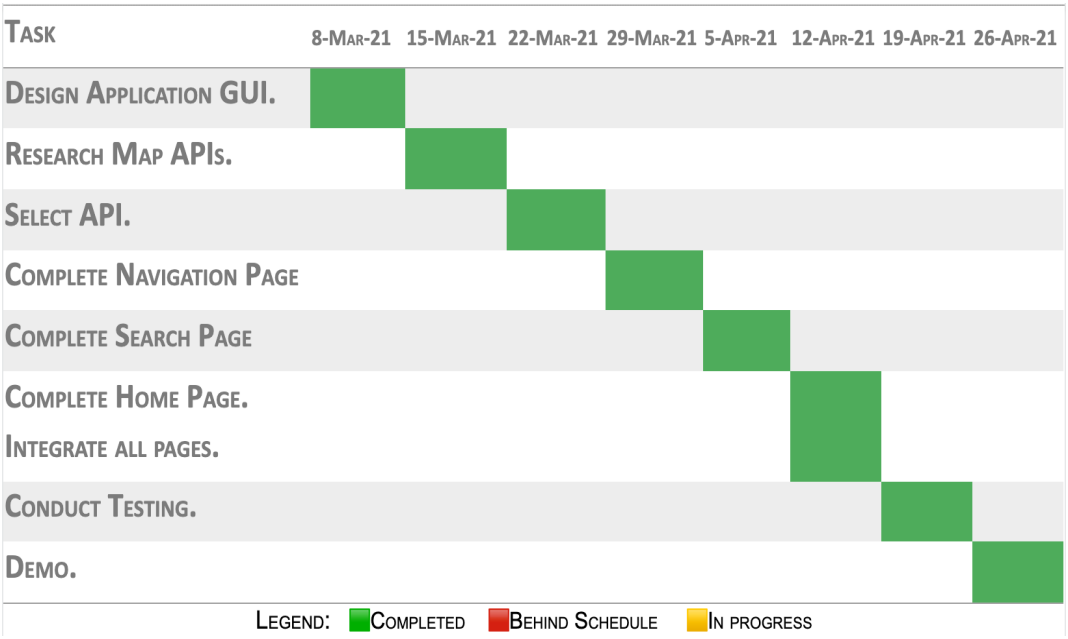

Table 1: Tasks in Execution Plan

4.2 User Application Component Validation

Different features on each page were tested before validating. Some failed prior to the final demonstration of this subsystem, but were fixed and subsequently denoted as passed. Apart from the features that failed initially, all the other features were tested and validated in a timely manner to remain in track with the project. Find a list of each of the features validated for each of the application pages in the table below.

Home Page	**Status**
Screen Visibility	Passed ✔
Button functionality to go to next page	Passed ✔
Search Page	Passed ✔
Screen Visibility	Passed ✔
Show which bins correspond to driver	Passed ✔
Scrolling functionality to go through bin list	Passed ✔
View list of bins denoting capacity	Passed ✔
Ability to go back to previous page	Passed ✔
Navigation Page	Passed ✔
Screen Visibility	Passed ✔
View bins to be serviced on map	Passed ✔
Zoom into map functionality	Passed ✔
Button functionality to go to next bin	Passed ✔
Display user location	Passed ✔
Show/map route as directions are displayed	Passed ✔
:put route directions with speech synthesizer	Passed ✔
Output route directions with text on screen	Passed ✔
Ability to go back to previous page	Passed ✔

Table 2: Feature Validation Tests/Status

5. Subsystem Conclusion

Overall, despite encountering some issues along the way this subsystem functioned well and as expected. There are still, however, some limitations in the subsystem that could be addressed were this project to be extended for development another semester.

5.1 Navigation Page Limitations

5.1.1 User Location Limitations

Due to only having tested the application through simulation, there may be faults that might make themselves present if the application were tested on a real iOS device. Testing on an actual iOS device would've also helped to validate whether the user location was working properly. In subsection 3.3, we talked about how the property of the user location is read-only which is why there was no way to simulate a driver actually following the generated directions for a route. In an actual iOS device it would've been possible to follow along with the directions and see the blue dot representing a user actually move inside the map.

5.1.2 Route Displaying Limitations

A feature that couldn't be implemented in the page, is the showing of the entire route prior to receiving directions. We talked about how as a driver proceeds from bin to bin a full route is generated and how theoretically that information could be sent back to a database had this subsystem been connected to one. That idea came from the inability of being able to show the full route without generating directions at the same time.

5.2 Impacts

The main goal of this subsystem was to give garbage truck drivers a means to efficiently and easily service trash bins in their respective cities. This was successfully accomplished. Impacts of this subsystem include:

- Time Saving: a driver can now save time on getting to each bin destination by simply selecting the "Next Bin" button as the location for the next bin will already be stored in the application.
- Resources Saving: a driver won't need to waste resources going to bins that aren't above seventy percent full.
- Environmentally Conscious: by avoiding bins that aren't above seventy percent full a driver's carbon footprint will not be as big.
- Driver Safety: the functionality of going back to a previous page is implemented by simply allowing a user to drag down on the screen. This permits the use of only one hand, so the driver might keep one hand on the steering wheel at the same time, avoiding possible accidents.

6. Software Requirements and Documentation

Swift 5.1 was used to develop the application for this subsystem in an Xcode 12.4 editor. Swift was used to both style and add the navigation functionality to the application with usage of the MapKit API. The MapKit API was what let us enable user location, get and display directions to the user, as well as allowing us to display a line on the map. To make the initial GUI sketches, Miro was utilized. The following table includes information pertaining to the software used in this subsystem including the version used and links to see documentation.

Software/Editor/API	Version	Documentation
Xcode	12.4	https://developer.apple.com/documentation/xcode-release-notes/xcode-12_4-release-notes
Swift	5.1	https://swiftdoc.org/
MapKit	included with Swift 5.1	https://developer.apple.com/documentation/mapkit/
Miro	n/a	https://miro.com/

Table 3: Software used and Documentation