Furious Monkeys

Presents

Front img.PNG

The Furious Monkeys

Andrew Laramee 27050925

Eric Payette 27008058

Hoang Khang Nguyen 27079427

Jacqueline Luo 26938949

Michael Mescheder 27202202

Piratheeban Annamalai 27755708

**Table of contents**

Introduction ……….………………………………………………………………………………… 4

Requirements ……………………………………………………………………………………….. 4

Backlog ……………………………………………………………………………………………….. 9

Constraints………………………………………………………………………………………….. 12

Architecture ……………………………………………………………………………………….. 13

Design Pattern Implementation….………………………………………………………….. 13

Domain Model …………………………………………………………………………………….. 14

Class Diagram ……………………………………………………………………………………… 15

Component Diagram ……….…………………………………………………………………… 16

Risk Management ……..………….……………………………………………………………… 17

Release Plan …………………..…………………………………………………………………… 21

Test Plan ……………………….…………………………………………………………………….25

Unit Testing ………………….……………………………………………………………………. 26

Acceptance Testing ……….…………………………………………………………………….. 34

System Testing ……………..…………………………………………………………………….. 37

Defect Tracking ……………..……………………………………………………………………..41

Quality Metrics ……...……..…………………………………………………………………….. 43

User Interface ……………………………………………………………………………………... 47

**Introduction**

Furious Monkeys has been tasked with creating a mobile application that reduces exhaust emissions caused by select public transportation services. mTransit is an application that focuses on decreasing the quantity of buses driving at night with few to no passengers aboard. The project currently focuses on managing buses of the Société de Transport de Montreal (STM) and is limited to the island of Montreal.

The STM currently works in such a way that night buses have a set route to drive and predetermined times on which to depart. Whether there are 10, 20 or no passengers on said bus has no effect on whether it will be running on any given night. This issue leads to empty buses running and extra exhaust emissions that could be avoided entirely if there were a system in place to notify drivers of potentially useless trips.

mTransit aims to solve this issue by having potential bus riders sign into the application and plan their own trip. Users will input their pickup and drop off location into the application and buses will be routed accordingly to pick them up along with other users on the same bus line. If a certain passenger threshold is not met on a given bus route, the bus will be held back and a nearby bus will be rerouted in order to pick up any passengers that may have been waiting for it.

The purpose of this document is as follows:

* Describe the requirements of the application
* Display the architecture of the system
* Declare and prioritize risks and produce strategies to mitigate them
* Show a rundown of how sprint 4 was carried out
* Explain how the project’s features plan to be tested
* Display the quality metrics for past and present iterations and show how they have improved
* Present user interface prototypes for the project

**Requirements**

|  |
| --- |
| **Sprint 4 changes** |
| Removed non-essential requirements to focus on the application's main functionality |

The following requirements were elicited from the product owner and have been turned into user stories approved by the product owner. **~~15~~ 7** epics have been elicited for a total of **~~64~~ 31** user story points after scoping out the less necessary features.

|  |  |
| --- | --- |
| EP-1 | As a rider, I want to login through Facebook~~, Twitter and Google+~~ so that I don't need an mTransit account |
| USP | ~~12~~ 4 |
| Priority | High |
| Description |  |

|  |  |
| --- | --- |
| ~~EP-2~~ | ~~As a rider, I want to be able to have my registration authenticated through SMS so that it is more secure.~~ |
| ~~USP~~ | ~~4~~ |
| ~~Priority~~ | ~~Low~~ |
| ~~Description~~ | ~~This will ensure that the rider is actually using a mobile device. The SMS will be sent once the user has entered their social media credentials and tries to register.~~ |

**\*\*Scoped out by product owner**

|  |  |
| --- | --- |
| ~~EP-3~~ | ~~As a rider, I want to know how long it will be until I am picked up so that I can plan my time accordingly.~~ |
| ~~USP~~ | ~~4~~ |
| ~~Priority~~ | ~~Medium~~ |
| ~~Description~~ |  |

**\*\*Scoped out by product owner**

|  |  |
| --- | --- |
| EP-4 | As a Rider, I’d like to be able to select my destination bus stop so that I can plan my trip. |
| USP | 5 |
| Priority | High |
| Description | After the user a logged into their account, they will be able to send a request for a ride. When doing so, they will be able to select the drop off station from the map. Once they do this, the app will generate the best route to produce fewer emissions. |

|  |  |
| --- | --- |
| ~~EP-5~~ | ~~As a rider, I want to change my drop-off location while on the bus so that I can get to a different destination.~~ |
| ~~USP~~ | ~~4~~ |
| ~~Priority~~ | ~~Medium~~ |
| ~~Description~~ | ~~Users have to set a destination point, but if they no longer want to go there, and while they are on route, they may change their destination point.~~ |

**\*\*Scoped out by product owner**

|  |  |
| --- | --- |
| EP-6 | As a rider, I’d like to be able to see all bus stops near me so that I can plan my trip accordingly. |
| USP | 5 |
| Priority | Medium |
| Description |  |

|  |  |
| --- | --- |
| EP-7 | As a rider, I’d like to be able to rate the route I have taken so that I can give my feedback. |
| USP | 3 |
| Priority | High |
| Description | The rider may rate the route that they have taken by saying how they thought the overall experience was; if they waited too long or if the route itself was good or not. |

|  |  |
| --- | --- |
| EP-8 | As a driver, I’d like to take the fastest and most economical route so that I can save time for riders and produce fewer emissions. |
| USP | 10 |
| Priority | High |
| Description | If there is no one (or almost no one) waiting for the bus, then another bus that has a similar route will pick up those few waiting for the bus and drop them off at their desired destinations. After dropping them off, the bus will go back on its original route. Although this might delay the length of the bus ride, the bus would produce fewer emissions than if both buses went on their own route. |

|  |  |
| --- | --- |
| ~~EP-9~~ | ~~As a driver, I’d like to log in with a code given to me by my employer so that I am verified as an employed STM driver.~~ |
| ~~USP~~ | ~~2~~ |
| ~~Priority~~ | ~~Medium~~ |
| ~~Description~~ | ~~The code will ensure that the driver is actually an employee of the STM.~~ |

**\*\*Scoped out by product owner**

|  |  |
| --- | --- |
| ~~EP-10~~ | ~~As a driver, I’d like to know how many people are waiting at each stop so that I can plan my stops accordingly.~~ |
| ~~USP~~ | ~~4~~ |
| ~~Priority~~ | ~~Medium~~ |
| ~~Description~~ | ~~This is for the safety of the bus and the passengers. If the number of people waiting for the bus exceeds the bus’ capacity, then another bus may pass to pick up those waiting.~~ |

**\*\*Scoped out by product owner**

|  |  |
| --- | --- |
| ~~EP-11~~ | ~~As a driver, I’d like to cover for another driver when there are not enough people on that line so that there are no buses that are needlessly running.~~ |
| ~~USP~~ | ~~8~~ |
| ~~Priority~~ | ~~High~~ |
| ~~Description~~ | ~~Every bus has a minimum amount of people calling for it before it should run. If a bus does not, another bus can pick up its passengers.~~ |

**\*\*Scoped out by product owner**

|  |  |
| --- | --- |
| EP-12 | As a user, I’d like to be able to change the language so that I can use the language that is most familiar to me. |
| USP | 2 |
| Priority | Low |
| Description |  |

|  |  |
| --- | --- |
| ~~EP-13~~ | ~~As a driver, I’d like to be able to know which stop’s total waiting time is the highest so that I know how my route is going to be adjusted.~~ |
| ~~USP~~ | ~~5~~ |
| ~~Priority~~ | ~~Medium~~ |
| ~~Description~~ | ~~Each stop will track the total wait time of people that are at that stop. This will affect how the initial algorithm calculates the route that the driver will take by giving high priority to stops with high total wait times.~~ |

**\*\*Scoped out by product owner**

|  |  |
| --- | --- |
| ~~EP-14~~ | ~~As a rider, I’d like to be able to select a wheelchair friendly bus to pick me up.~~ |
| ~~USP~~ | ~~2~~ |
| ~~Priority~~ | ~~Low~~ |
| ~~Description~~ |  |

**\*\*Scoped out by product owner**

|  |  |
| --- | --- |
| EP-15 | As a rider, I want to be able to search an area of Montreal in order to reduce the amount of bus stop markers on the map. |
| USP | 3 |
| Priority | Medium |
| Description | When a user is in the process of searching for a destination bus stop, we will allow the use of a search bar to shift the map over to the general area they search rather than having them drag the map the entire way. Bus markers will only displayed within a certain distance of the area searched. |

Backlog

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Epic ID | Name | USP | Priority |
| US-1 | EP-1 | Login through Facebook | 3 | High |
| ~~US-2~~ | ~~EP-1~~ | ~~Login through Google+~~ | ~~3~~ | ~~High~~ |
| ~~US-3~~ | ~~EP-1~~ | ~~Login through Twitter~~ | ~~3~~ | ~~High~~ |
| US-4 | EP-1 | Facebook registration UI | 1 | High |
| ~~US-5~~ | ~~EP-1~~ | ~~Google+ registration UI~~ | ~~1~~ | ~~High~~ |
| ~~US-6~~ | ~~EP-1~~ | ~~Twitter registration UI~~ | ~~1~~ | ~~High~~ |
| ~~US-7~~ | ~~EP-2~~ | ~~SMS Authentication UI~~ | ~~1~~ | ~~Low~~ |
| ~~US-8~~ | ~~EP-2~~ | ~~Integrate SMS Authentication~~ | ~~3~~ | ~~Low~~ |
| ~~US-9~~ | ~~EP-3~~ | ~~Time-to-pickup timer integration~~ | ~~3~~ | ~~Medium~~ |
| ~~US-10~~ | ~~EP-3~~ | ~~Time-to-pickup UI design~~ | ~~1~~ | ~~Medium~~ |
| US-11 | EP-4 | Bus Route Selection UI | 1 | High |
| US-12 | EP-4 | Integrate custom pins on map | 1 | High |
| US-13 | EP-4 | Limit a maximum of 1 selected pin on the map | 1 | High |
| US-14 | EP-4 | Integrate a connection path from current location to destination pin | 2 | High |
| ~~US-15~~ | ~~EP-5~~ | ~~Drop-off Change Integration~~ | ~~4~~ | ~~Medium~~ |
| US-16 | EP-6 | Map UI Design | 1 | Medium |
| US-17 | EP-6 | Google Map API integration | 2 | High |
| US-18 | EP-6 | Add Bus stops node | 1 | High |
| US-19 | EP-7 | Rate App UI | 1 | Medium |
| US-20 | EP-7 | Integrate rating system for the application | 2 | High |
| US-21 | EP-8 | Efficient Route Selection | 10 | High |
| ~~US-22~~ | ~~EP-9~~ | ~~Driver Registration UI~~ | ~~1~~ | ~~High~~ |
| ~~US-23~~ | ~~EP-9~~ | ~~Driver Code Authentication~~ | ~~1~~ | ~~Medium~~ |
| ~~US-24~~ | ~~EP-10~~ | ~~Ride Counter UI~~ | ~~1~~ | ~~Medium~~ |
| ~~US-25~~ | ~~EP-10~~ | ~~Ride Counter~~ | ~~3~~ | ~~Medium~~ |
| ~~US-26~~ | ~~EP-11~~ | ~~Route Change Constraint~~ | ~~8~~ | ~~High~~ |
| US-27 | Ep-12 | French Language Selection | 2 | Low |
| ~~US-28~~ | ~~EP-13~~ | ~~Calculate total wait time of riders for each bus stop~~ | ~~3~~ | ~~Medium~~ |
| ~~US-29~~ | ~~EP-13~~ | ~~Integrate total wait time into route algorithm calculation~~ | ~~2~~ | ~~Medium~~ |
| ~~US-30~~ | ~~EP-14~~ | ~~Add wheelchair accessible button~~ | ~~2~~ | ~~Low~~ |
| US-31 | EP-15 | Implement search feature on map page | 1 | Medium |
| US-32 | EP-15 | Calculate max distance markers will be shown using latitude and longitude | 1 | Medium |
| US-33 | EP-15 | Parse stm\_stops.json for stops within calculated max distance of searched location | 1 | Medium |
| Total |  |  | **~~64~~ 31** |  |

Epic Legend

|  |  |
| --- | --- |
| ID | Epic Name |
| EP-1 | As a rider, I want to login through Facebook so that I don't need an mTransit account |
| ~~EP-2~~ | ~~As a rider, I want to be able to have my registration authenticated through SMS so that it is more secure~~ |
| ~~EP-3~~ | ~~As a rider, I want to know how long it will be till I am picked up so that I can plan my time accordingly~~ |
| EP-4 | As a rider, I’d like to be able to select my destination bus stop so that I can plan my trip |
| ~~EP-5~~ | ~~As a rider, I want to change my drop-off location while on the bus so that I can get to a different destination~~ |
| EP-6 | As a rider, I’d like to be able to see all bus stops near me so that I can plan my trip accordingly |
| EP-7 | As a rider, I’d like to be able to rate the route I have taken so that I can give my feedback |
| EP-8 | As a driver, I’d like to take the fastest and most economical route so that I can save time for riders and produce fewer emissions |
| ~~EP-9~~ | ~~As a driver, I’d like to log in with a code given to me by my employer so that I am verified as an employed STM driver~~ |
| ~~EP-10~~ | ~~As a driver, I’d like to know how many people are waiting at each stop so that I can plan my stops accordingly~~ |
| ~~EP-11~~ | ~~As a driver, I’d like to cover for another driver when there are not enough people on that line so that there are no buses that are needlessly running~~ |
| EP-12 | As a user, I’d like to be able to change the language so that I can use the language that is most familiar to me |
| ~~EP-13~~ | ~~As a driver, I'd like the total wait time of riders at each bus stop be known so that the bus route can be optimized to pick up riders in a timely manner~~ |
| ~~EP-14~~ | ~~As a rider, I’d like to be able to select a wheelchair friendly bus to pick me up~~ |
| EP-15 | As a rider, I want to be able to search an area of Montreal in order to reduce the amount of bus stop markers on the map |

**Constraints**

|  |
| --- |
| **Sprint 4 changes** |
|  |

Several constraints have been identified for the mTransit project. It is crucial to consider these constraints when we are planning the design and building the application so that no time is wasted implementing functionality out of the scope of the project. The list of constraints for the mTransit project is as follows:

* Riders can only plan a route that would take a single bus. Riders are not able to select a destination that requires multiple buses.
* If a rider is further than 1km from the nearest bus stop, the user will not be able to select a route. They will be notified that they are too far away from the nearest bus stop.
* If a user does not have a ~~social media~~ Facebook account, then he will have to create one to use the functionalities of the mTransit app.
* A driver can also be a rider. Not all riders can be drivers since they need to be authenticated (they need a driver’s code provided by the STM).
* If a user does not own a smart android phone, they will not be able to use the functionality of the mTransit application.
* If a user, while in the middle of their trip, changes their destination to a new location in the opposite direction that the bus is traveling, the change will be denied. The user will be notified that the change is not made.

**Architecture**

|  |
| --- |
| **Sprint 4 changes** |
|  |

**Design Pattern Implementation**

**Singleton**

The MapPage class has been implemented with the singleton design pattern. This was due to the fact that the application needs a maximum of one map to display bus stop markers and show the user’s route. The MapPage also handles requests from multiple disparate parts of the application and it needs to be assured that another instance of MapPage does not exist.

**Observer**

The observer pattern has been implemented onto the google-maps.ts service which uses the native Google maps geolocation service to watch the user’s position and measure distances to nearby bus stops. The service is subscribed to which sends a constant stream of data based on when the user’s location changes and this data is observed and acted upon.

**Domain Model**

The domain model begins at the HomePage which interacts with STM and social media authentication components. Once authenticated, the homepage sends the user to the MapPage, which has a different display based on being a Rider or Driver. To start as a Rider, the Rider needs authentication from Facebook through Firebase, and then has to do a one-time authentication through SMS. To start as a Driver, the Driver needs to be a certified driver from the STM and be authenticated by the STM database. The driver’s information and credentials are stored in DriverCatalog. Riders can also rate the quality of their trip, which is stored with the bus route that was taken.

The MapPage has access to BusCatalog and BusStopCatalog, which store Bus and BusStop information respectively. BusCatalog also has access to BusStopCatalog in order to add a stop to the bus’s current route. BusCatalog can access the BusRouteCatalog as well in order to give a predetermined route to a bus.



**Class Diagram**

The class diagram follows the same structure as the domain model but has additional implementation. The MapPage has all the methods for a Driver and Rider. What is displayed on the map is different if the user is a Driver or a Rider.

BusCatalog is an arraylist that holds Bus objects. A bus object has a variable that marks if the bus is wheelchair accessible. A Bus Object has a BusRoute object as a variable which is initially taken from the BusRouteCatalog. BusStopCatalog is an arraylist that holds BusStop objects, which store the location and id of the stop, the current number of passengers waiting at the stop and the combined waiting time of the passengers at the busStop. BusRouteCatalog is an arraylist of busRoute objects, which contains an arraylist of busStops as well as all the user feedback for the route.

BusCatalog contains methods to calculate how long until a certain stop is reached (timeUntilStop) and to calculate which bus should pick up a rider based on pickup and drop-off chosen (findAppropriateBus). BusStop Catalog contains a method to find the bus stops nearest a given location (findNearBusStops).



**Component Diagram**

The mTransit UI component is responsible for displaying the information and visuals to the user, which is gotten from the mTransit Logic component. This component is responsible for the core logic within the system, such as first-time authentication, optimizing the bus routes and finding the nearest bus stops to the user. These are handled by the Authenticate, Bus Route and Bus Stops components respectively.

The mTransit Logic component depends on external social media and STM components for authentication from a third party tool called auth0. The Ionic Subsystems component, which uses HTML, CSS and JavaScript, and the mTransit Database component allows storage and retrieval of data, can then be used after authentication.



**Risk Management**

|  |
| --- |
| **Sprint 4 changes** |
| **Risk resolved:** TO-1: Most members of the team are unfamiliar with the Ionic framework |
| **Risk resolved:** TO-2: Team members unfamiliar with testing frameworks |
| **Risk resolved:** TO-6: Team using unfamiliar database technology (MongoDB) |
| **Risk resolved:** PE-1: Teammates not pulling their weight in the project |
| **Risk resolved:** PE-3: Teammates dropping the class |
| **Risk resolved:** RE-3: Product owner wants to add/change/remove requirements |
| **Risk resolved:** RE-4: Requirements are too complex to meet deadline |
| **Risk resolved:** RE-5: Team fails to consider certain requirements |

The tables below depict how Furious Monkeys plan to manage all possible risks that could occur during the development of mTransit. The first table shows a direct probability to impact assessment of the possible risks that may occur. The second table deals with a more detailed analysis of each risk and the strategies that Furious Monkeys will assume in order to handle them. Each risk is assessed qualitatively and each level is explained below:

**High**: Extremely likely to occur / Represents a High adverse impact

**Moderate**: Somewhat likely to occur / Represents a moderate adverse impact

**Low**: Unlikely to occur / Represents a low adverse impact

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Probability | High |  |  | TO-2 | TO-1, TO-6, PE-4, RE-4 | RE-3 |
| Medium | RE-1, TE-2 |  | TE-1, PE-2 | PE-1, RE-5 | TO-5. RE-2 |
| Low |  |  | PE-5 | TO-3 | TO-4,  PE-3 |
|  | | Minimal | Minor | Moderate | Significant | Severe |
| Impact | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Risk Assessment | | | | Risk Management | |
| Risk ID | Description | Probability | Impact | Resolved in Sprint | Strategy and Effectiveness |
| Technology Risks | | | | | |
| TE-1 | Technology used is not scalable | Medium | Moderate | - | Switch to technologies that are more scalable.  Strategy: Avoidance  Effectiveness: Medium |
| TE-2 | Technology cannot be used by people with certain conditions (color blind) | Medium | Minimal | 3 | Switch to colours that can be seen by everyone. Also, add physical lines that can be seen by everyone.  Strategy: Avoidance  Effectiveness: High |
| Tools Risks | | | | | |
| TO-1 | Most members of the team are unfamiliar with the Ionic framework | High | Significant | 4\*\* | Team members will have to watch and learn from online tutorials and help each other  Strategy: Acceptance  Effectiveness: Medium |
| TO-2 | Team members unfamiliar with testing frameworks | High | Moderate | 4\*\* | Team members will have to watch and learn from online tutorials  Strategy: Acceptance  Effectiveness: Medium |
| TO-3 | Team members unfamiliar with Jira | Low | Significant | 1 | The class as a whole had a tutorial session on Jira  Strategy: Mitigation  Effectiveness: High |
| TO-4 | Social Media login API stop functioning | Low | Severe | 3 | Implement our own login functionality  Strategy: Acceptance  Effectiveness: Medium |
| TO-5 | Team member unable to install the Ionic framework on computer | Medium | Severe | 2 | Team member will have to uninstall programs or use another computer.  Strategy: Acceptance  Effectiveness: High |
| TO-6 | Team using unfamiliar database technology (MongoDB) | High | Significant | 4\*\* | Team members will have to learn how to use the technology.  Strategy: acceptance  Effectiveness: Medium |
| People Risks | | | | | |
| PE-1 | Teammates not pulling their weight in the project | Medium | Significant | 4\*\* | Tasks given to small teams of two.  Strategy: Mitigation  Effectiveness: High |
| PE-2 | Teammates not at meetings and therefore unaware of their tasks | Medium | Moderate | 1 | Tasks and issues are shown on both github and Jira so team members are still aware of what needs to be done.  Strategy: Mitigation  Effectiveness: High |
| PE-3 | Teammates dropping the class | Low | Severe | 4\*\* | Work will be distributed according to number of members remaining in team.  Strategy: Acceptance  Effectiveness: Medium |
| PE-4 | Teammates have conflicting schedules | High | Significant | 1 | Work around team members’ schedules.  Strategy: Acceptance  Effectiveness: Moderate |
| PE-5 | Lack of communication between team members | Low | Moderate | 1 | Team members communicate through discord which is available on a mobile device.  Strategy: Avoidance  Effectiveness: High |
| Requirement Risks | | | | | |
| RE-1 | App is too complicated for older phones | Medium | Low | 1 | After having the meeting with the client, he wanted the app to be compatible with new phones.  Strategy: Mitigation  Effectiveness: High |
| RE-2 | Team misunderstands requirements | Medium | Severe | 1 | Team had a meeting with the client so the requirements were clarified.  Strategy: Mitigation  Effectiveness: High |
| RE-3 | Product owner wants to add/change/remove requirements | High | Severe | 4\*\* | If there are any changes made to the requirements, then the work will be distributed among members of the team.  Strategy: Acceptance  Effectiveness: Medium |
| RE-4 | Requirements are too complex to meet deadline | High | Significant | 4\*\* | Team members will do what they can to meet all requirements, starting with high risk high value and ending with high risk low value.  Strategy: Acceptance  Effectiveness: Medium |
| RE-5 | Team fails to consider certain requirements | Medium | Significant | 4\*\* | Team members will add these new requirements in a future sprint.  Strategy: Acceptance  Effectiveness: Medium |

**\*\*added/updated in current sprint**

**Release Plan**

|  |
| --- |
| **Sprint 4 changes** |
| Updated Sprint 4 release plan with scoped out epics |
| Updated project velocity for sprint 4 |
| Updated retrospective with summarized thoughts |

**Sprint 4 summary**

The goal of Sprint 4 was to finish as many of the important functionality based features as possible so that the product owner could create an API with our work. This allowed us to scope out many of the ‘bells and whistles’ type features to focus on getting a working template ready by the end of the sprint. We started with 50 user story points that needed to be completed and ended with 22 which encompass the most crucial epics of the software.

We got SonarQube fully functioning and took a full scan of the system we had at the end of Sprint 3 to compare with our current results. With major refactoring, including vulnerability and bug removal, we were able to reach an acceptable level of quality across our system. Our code coverage grew much over the course of the sprint as we had a team of individuals working solely on creating and running unit tests. We did not succeed in reaching the desired 80% code coverage as there were difficulties in learning how to create certain unit tests, but we progressed a lot from its initial state.

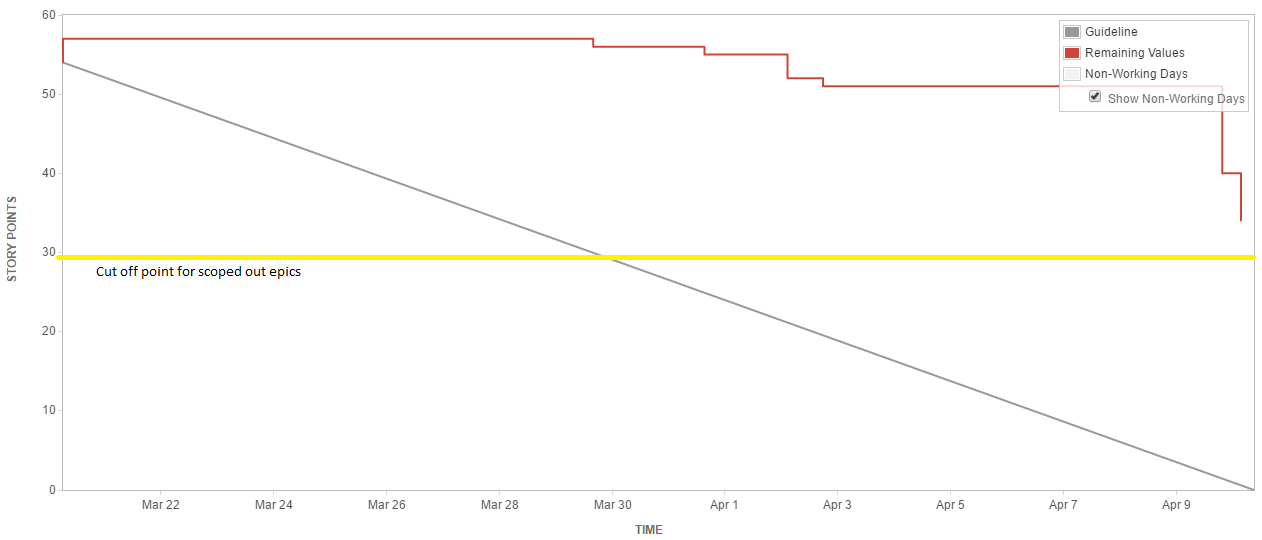
With the progress we’ve made, we have a solid base with which and API may be created for nighttime bus services in Montreal. The bus route creation has been developed with both the google directions services to find the fastest route as well as stm data in order to add the proper bus stop locations. The autocomplete search functionality will help users find their destination more efficiently through whatever service the api is used for.

Through the high and low points we’ve had in these 4 sprints we have learned a lot and have greatly developed our coding and software management skills. These will help us greatly in the capstone project coming next year and throughout our careers as well.

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Description** | **USP** | **Status** |
| ~~EP-3~~ | ~~As a rider, I want to know how long it will be till I am picked up so that I can plan my time accordingly~~ | ~~4~~ | Scoped out |
| EP-4 | As a Rider, I’d like to be able to select my drop off stations so that I can get to my destination. | 4 | **Completed** |
| ~~EP-5~~ | ~~As a Rider, I’d like to be able to change my drop off station so that I can get to a different destination.~~ | ~~2~~ | Scoped out |
| EP-6 | As a Rider, I’d like to be able to see all bus stops near me so that I can choose which one to go to. | 5 | **Completed** |
| EP-7 | As a rider, I’d like to be able to rate the route I have taken so that I can give my feedback | 3 | **Incomplete** |
| EP-8 | As a driver, I’d like to take the fastest and most economical route so that I can save time for riders and produce fewer emissions | 10 | **Completed** |
| ~~EP-9~~ | ~~As a driver, I’d like to log in with a code given to me by my employer so that I am verified as an employed STM driver~~ | ~~2~~ | Scoped out |
| ~~EP-10~~ | ~~As a Driver, I’d like to know how many people are waiting at each stop so that I can plan my stops accordingly.~~ | ~~5~~ | Scoped out |
| ~~EP-11~~ | ~~As a driver, I’d like to cover for another driver when there are not enough people on that line so that there are no buses that are needlessly running~~ | ~~8~~ | Scoped out |
| ~~EP-13~~ | ~~As a driver, I'd like the total wait time of riders at each bus stop be known so that the bus route can be optimized to pick up riders in a timely manner~~ | ~~5~~ | Scoped out |
| ~~EP-14~~ | ~~As a rider, I’d like to be able to select a wheelchair friendly bus to pick me up~~ | ~~2~~ | Scoped out |
| EP-15 | As a rider, I want to be able to search an area of Montreal in order to reduce the amount of bus stop markers on the map | 3 | **Completed** |
| Total user story points | | 22/25 ~~50~~ |  |

**Project velocity after 4 sprints: 17.21 (+4.79 from S3)**

**Sprint 4 Burndown Chart**



**Sprint 4 Retrospective**

**Things that helped:**

* Weekly in-person team meetings: These meetings highlights the accomplished task that the team is currently doing or have finished.
* Daily SCRUM meeting: Each meeting gives the team a stronger understanding of our progression. Additionally, it encourages the teammates to assign themselves tasks to do.
* Consistent use of reporting tools: This will help us immensely keep track of our progression and a more accurate time of completion.
* Transfer of knowledge: ​So the entire team benefits from the effort and time spent learning a tool or a work around.
* Pair programming to ensure everyone pulled their weight.
* Finding workarounds for feature blocking issues.

**Things that hindered us:**

* Getting stuck on single issues that last multiple days without solid progress.
* Not doing enough research on frameworks we chose to use before implementing them.

**Test Plan**

|  |
| --- |
| **Sprint 4 changes** |
| **Updated acceptance tests to reflect completed epics and added a note on which have been signed by the product owner and which are to be signed in the near future** |
| **Updated defect tracking section with new Sprint 4 bugs and resolutions** |
| **Updated unit testing section with all unit tests run through karma** |
| **Updated System tests to include all relevant passed/failed epics** |

**Tools**

All unit tests will be executed using Karma, a Javascript test runner, through the Jasmine testing framework. Ionic uses AngularJS for a lot of the core functionality and both Karma and Jasmine are the recommended tools for unit testing with AngularJS. The Karma-Jasmine combination is the most popular testing duo for ionic and angular projects as they work seamlessly together and can be used with a variety of different browsers.

**Why Jasmine**

Jasmine is known for keeping an extremely detailed and well-structured format for all executed tests. It also includes test doubles through the use of “spies” to control a specific function’s behavior and to see how it is being used. This feature is only available in Mocha through an external test double library called Sinon. While the Mocha-Sinon duo provides a small amount of extra testing functionality we decided that we would not need any more than Jasmine provides. Seeing as none of our team members have ever worked with a dedicated testing framework before, the abundance of Jasmine tutorials and clear documentation were a large selling point in our decision to use it.

**Why Karma**

The choice to use Karma was based on a few criteria that our team deemed important and we did not compare it to any other test runners. We researched Karma to find out that it was both created and recommended by the AngularJS team, it was lightweight and there were countless tutorials and examples of professionals using it together with Jasmine.

**Organization**

The tests will be structured in a standardized scheme with a test class per class being tested and a nested class for each method being tested. This scheme keeps things organized as all tests for a method are grouped together and the structure will not be affected when run through Karma.

**Unit Testing**

**Unit Testing Tools**

Unit testing will be done using Jasmine and Karma. Jasmine was used to create unit tests and Karma was used to run them.

All our tests are run at the same time, within a testing folder.

**Relevant Units To Be Tested**

All the core modules of application need to be tested. Each page of the application need to be able to be opened. The mobile application needs to work according to the user’s interactions with the system. This means that when the user wants to change languages, the function will be applied.

**Unit Test Report**

For this current sprint, we are testing to see that the pages actually do open up and that the swapping of language functions properly for the users.

|  |  |
| --- | --- |
| **Unit Test (App Component)** | **Result** |
| **RootComponent.is created** | Pass |
| **RootComponent.HomePage** | Fail |
| **Summary** | 1/2 tests passed |

|  |  |
| --- | --- |
| **Unit Test (Home Page)** | **Result** |
| **HomePage.is created** | Pass |
| **HomePage.goToMap()** | Pass |
| **HomePage.goToDriverLogin()** | Pass |
| **HomePage.goToRating()** | Pass |
| **HomePage.swapLanguage().exists** | Pass |
| **HomePage.swapLanguage()** | Pass |
| **HomePage.translateService.use(‘en’)** | Pass |
| **HomePage.translateService.use(‘fr’)** | Pass |
| **HomePage.facebookLogin().exists** | Pass |
| **Summary** | 9/9 tests passed |

|  |  |
| --- | --- |
| **Unit Test (Map Page)** | **Result** |
| **MapPage.is created** | Pass |
| **MapPage.loadMap().exists** | Pass |
| **MapPage.loadMap()** | Pass |
| **MapPage.goToRating().exists** | Pass |
| **MapPage.goToRating()** | Pass |
| **Summary** | 5/5 tests passed |

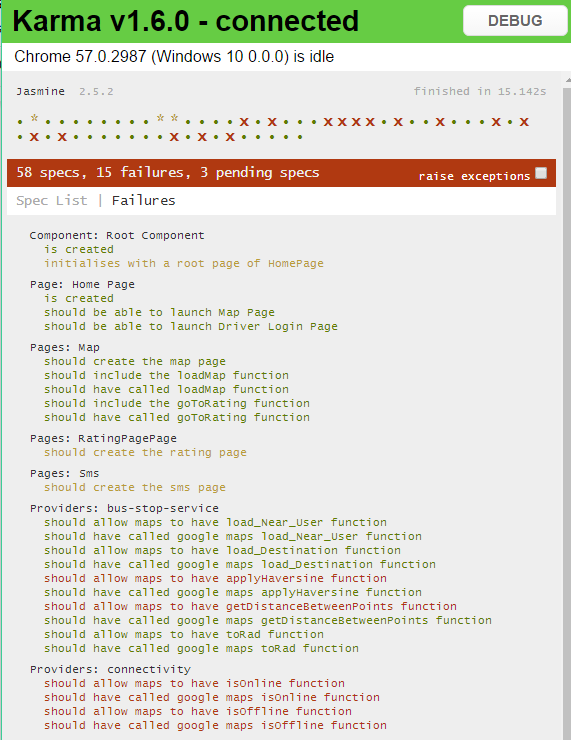
|  |  |
| --- | --- |
| **Unit Test (Rating Page)** | **Result** |
| **RatingPagePage.is created** | Fail |
| **Summary** | 0/1 tests passed |

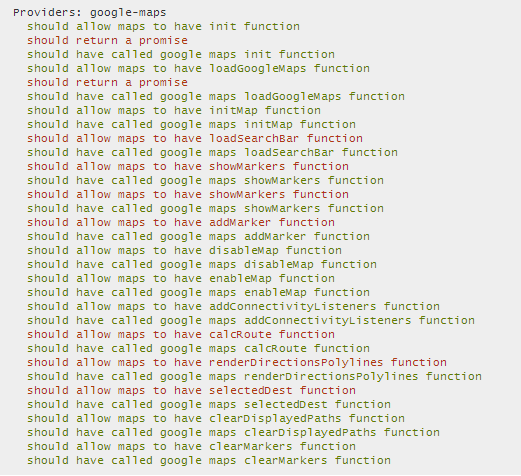
|  |  |
| --- | --- |
| **Unit Test (SMS Page)** | **Result** |
| **SmsPage.is created** | Fail |
| **Summary** | 0/1 tests passed |

|  |  |
| --- | --- |
| **Unit Test (BusStopService Provider)** | **Result** |
| **bus-stop-service.load\_Near\_User().exists** | Pass |
| **bus-stop-service.load\_Near\_User()** | Pass |
| **bus-stop-service.load\_Destination().exists** | Pass |
| **bus-stop-service.load\_Destination()** | Pass |
| **bus-stop-service.applyHaversine().exists** | Fail |
| **bus-stop-service.applyHaversine()** | Pass |
| **bus-stop.service.getDistanceBetweenPoints().exists** | Fail |
| **bus-stop.service.getDistanceBetweenPoints()** | Pass |
| **bus-stop.service.toRad().exists** | Pass |
| **bus-stop.service.toRad()** | Pass |
| **Summary** | 8/10 tests passed |

|  |  |
| --- | --- |
| **Unit Test (Connectivity Provider)** | **Result** |
| **Connectivity.isOnline().exists** | Fail |
| **Connectivity.isOnline()** | Fail |
| **Connectivity.isOffline().exists** | Fail |
| **Connectivity.isOffline()** | Fail |
| **Summary** | 0/4 tests passed |

|  |  |
| --- | --- |
| **Unit Test (Google-Maps Provider)** | **Result** |
| **google-maps.init().exists** | Pass |
| **google-maps.init(): Promise<any>** | Fail |
| **google-maps.init()** | Pass |
| **google-maps.loadGoogleMaps().exists** | Pass |
| **google-maps.loadGoogleMaps(): Promise<any>** | Fail |
| **google-maps.loadGoogleMaps()** | Pass |
| **google-maps.initMap().exists** | Pass |
| **google-maps.initMap()** | Pass |
| **google-maps.loadSearchBar().exists** | Fail |
| **google-maps.loadSearchBar()** | Pass |
| **google-maps.showMarkers().exists** | Fail |
| **google-maps.showMarkers()** | Pass |
| **google-maps.addMarker().exists** | Fail |
| **google-maps.addMarker()** | Pass |
| **google-maps.disableMap().exists** | Pass |
| **google-maps.disableMap()** | Pass |
| **google-maps.enableMap().exists** | Pass |
| **google-maps.enableMap()** | Pass |
| **google-maps.addConnectivityListeners().exists** | Pass |
| **google-maps.addConnectivityListeners()** | Pass |
| **google-maps.calcRoute().exists** | Fail |
| **google-maps.calcRoute()** | Pass |
| **google-maps.renderDirectionsPolylines().exists** | Fail |
| **google-maps.renderDirectionsPolylines()** | Pass |
| **google-maps.selectedDest().exists** | Fail |
| **google-maps.selectedDest()** | Pass |
| **google-maps.clearDisplayedPaths().exists** | Pass |
| **google-maps.clearDisplayedPaths()** | Pass |
| **google-maps.clearMarkers().exists** | Pass |
| **google-maps.clearMarkers()** | Pass |
| **Summary** | 22/30 tests passed |





**Acceptance Testing**

All of the user stories presented in the backlog will go through rigorous acceptance testing once their associated tasks have been completed. This will assure that the business goals set by the product owner are met and approved.

**Sprint 4**

|  |  |
| --- | --- |
| **AT-1** | **EP-1 As a Rider, I want to be able to login with a social media account** |
| **Acceptance Criteria** | I am able to select Facebook as my desired login gateway. I am logged into mTransit and able to access all of the application’s “Rider” functionality. |
| **Result** | **PASS** |
| **Comments** | **Passed in Sprint 3 - Signed by product owner** |

|  |  |
| --- | --- |
| **AT-6** | **EP-6 As a Rider, I’d like to be able to see all bus stops near me** |
| **Acceptance Criteria** | From the main map page I am able to select a bus route and see all of its bus stops as individual markers on the map. |
| **Result** | **PASS** |
| **Comments** | **Passed in Sprint 4 - Signed by product owner** |

|  |  |
| --- | --- |
| **AT-12** | **EP-12 As a user, I’d like to be able to change the language** |
| **Acceptance Criteria** | From the sidebar I am able to select “French” to change the application’s language to French. |
| **Result** | **PASS** |
| **Comments** | **Passed in Sprint 3 - Signed by product owner** |

|  |  |
| --- | --- |
| **AT-4** | **EP-4 As a Rider, I’d like to be able to select my destination bus stop** |
| **Acceptance Criteria** | As a rider logged into the application, I am able to pick a destination station from existing bus stops. |
| **Result** | **PASS** |
| **Comments** | **Passed in Sprint 4 - To be signed by product owner** |

|  |  |
| --- | --- |
| **AT-8** | **EP-8 As a driver, I’d like to take the fastest and most economical route** |
| **Acceptance Criteria** | As a driver, I am able to take the quickest path to pick up all of the waiting riders on my route. |
| **Result** | **PASS** |
| **Comments** | **Passed in Sprint 4 - To be signed by product owner** |

|  |  |
| --- | --- |
| **AT-15** | **EP-15 As a rider, I want to be able to search an area of Montreal in order to reduce the amount of bus stop markers on the map** |
| **Acceptance Criteria** | From the map page, I am able to search for any given location in Montreal and have the map zoom into that location and show all of the bus stops within a constant diameter of my position. |
| **Result** | **PASS** |
| **Comments** | **Passed in Sprint 4 - To be signed by product owner** |

|  |  |
| --- | --- |
| **AT-7** | **EP-7 As a Rider, I’d like to be able to rate the application** |
| **Acceptance Criteria** | From the menu sidebar, I am able to select “Rate the app” and I am brought to the rating page. I can select between 1 and 5 stars and I am able to add an optional comment. |
| **Result** | **Incomplete** |
| **Comments** | Rating page is created and accessible but does not connect properly to the backend |

**System Testing**

System tests will be executed by simulating the workflow a real user would take while using the application, both as a Rider and a Driver.

As different functionalities of the application become completed, tests will be created for each workflow that can be followed by the user. Each test will be broken down into detailed steps that lead to the specified functionality being thoroughly examined. Steps will be given a pass or fail result and each failure will spark the creation of a bug in Jira to reference it.

**Sprint 4 System Testing**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ST-1** | **EP-1 As a Rider, I want to be able to login with a social media account** | | **Expected Output** | **Result** |
| **Reproduction Steps** | **1** | Open application on Android device | Application is opened to the login page | Pass |
| **2** | Select Facebook, Twitter or Google+ as login choice | Application prompts for credentials of selected social media service | Pass |
| **3** | Enter credentials for selected social media service | Application verifies credentials through the social media API and logs the user into mTransit | Pass |
| **Final Result** | Pass | | | |
| **Comments** | **Passed in Sprint 3** | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ST-12** | **EP-12 As a user, I’d like to be able to change the language** | | **Expected Output** | **Result** |
| **Reproduction Steps** | **1** | Open the application | The application loads the login screen | Pass |
| **2** | Select “French” from top right of the screen | The application language switched from english to french | Pass |
| **Final Result** | Pass | | | |
| **Comments** | **Passed in Sprint 3** | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ST-6** | **EP-6 As a rider, I want to be able to see all bus stops near me so that I can plan my trip accordingly** | | **Expected Output** | **Result** |
| **Reproduction Steps** | **1** | Open the application | The application loads the login screen | Pass |
| **2** | Follow the login procedure | The application logs in with the given credentials and loads the map page with all of the nearby bus stops | Pass |
| **Final Result** | Pass | | | |
| **Comments** | **\*\*Passed in Sprint 4** | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ST-15** | **EP-15As a rider, I want to be able to search an area of Montreal in order to reduce the amount of bus stop markers on the map.** | | **Expected Output** | **Result** |
| **Reproduction Steps** | **1** | Open the application | The application loads the login screen | Pass |
| **2** | Follow the login procedure | The application logs in with the given credentials and loads the map page with all of the nearby bus stops | Pass |
| **3** | Search Kirkland | Map displays Kirkland area | Pass |
| **Final Result** | Pass | | | |
| **Comments** | **\*\*Passed in Sprint 4** | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ST-4** | **EP-4 As a Rider, I’d like to be able to select my destination bus stop so that I can plan my trip.** | | **Expected Output** | **Result** |
| **Reproduction Steps** | **1** | Open the application | The application loads the login screen | Pass |
| **2** | Follow the login procedure | The application logs in with the given credentials and loads the map page with all of the nearby bus stops | Pass |
| **3** | Tap the screen close to Kirkland | Bus stops that are near show up as light blue pins | Pass |
| **4** | Double tap a light blue pin | The tapped blue pin turns yellow. There is also a blue path created from your location to a bigger purple pin that represents the bus stop that you will be picked at. The path continues from the big purple pin to the yellow pin that represents the bus’ route. | Pass |
| **Final Result** | Pass | | | |
| **Comments** | **\*\*Passed in Sprint 4** | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ST-8** | **E-P 8 As a driver, I’d like to take the fastest and most economical route** | | **Expected Output** | **Result** |
| **Reproduction Steps** | **1** | Open the application | The application loads the login screen | Pass |
| **2** | Follow the login procedure | The application logs in with the given credentials and loads the map page with all of the nearby bus stops | Pass |
| **3** | Tap the screen close to Kirkland | Bus stops that are near show up as light blue pins | Pass |
| **4** | Double tap a light blue pin | The tapped blue pin turns yellow. The path from the big purple pin to the yellow pin that represents the bus’ route which is the most economical. | Pass |
| **Final Result** | Pass | | | |
| **Comments** | **\*\*Passed in Sprint 4** | | | |

**Defect Tracking**

|  |
| --- |
| **Sprint 4 changes** |
|  |

**Sprint 4 report:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Defect ID** | **Description** | **Discovered** | **Resolved** | **Status** |
| FM-76 | Login stuck at ‘OK’ page | Sprint 2 | Sprint 3 | **RESOLVED** |
| FM-79 | The language button requires two key presses to be executed each time the application is loaded | Sprint 3 | - | **INCOMPLETE due to low priority** |
| FM-80 | Geolocation of user's current position isn't accepted by distance calculation function | Sprint 3 | Sprint 3 | **RESOLVED** |
| FM-86 | Error: Unexpected token / in JSON at position 404 | Sprint 3 | Sprint 4 | **RESOLVED** |
| FM-87 | Ratings page loads blank screen on mobile | Sprint 4 | Sprint 4 | **RESOLVED** |
| FM-88 | Can't pass SMS authentication page | Sprint 4 | - | **INCOMPLETE** |
| FM-89 | Star rating on rating page does not update until 'Continue' button is pressed | Sprint 4 | - | **INCOMPLETE** |
| FM-90 | Read JSON file page must be run prior to map opening for markers to display | Sprint 4 | Sprint 4 | **RESOLVED** |
| FM-91 | Cancel Search does not remove existing Markers | Sprint 4 | Sprint 4 | **RESOLVED** |
|  |  |  |  |  |

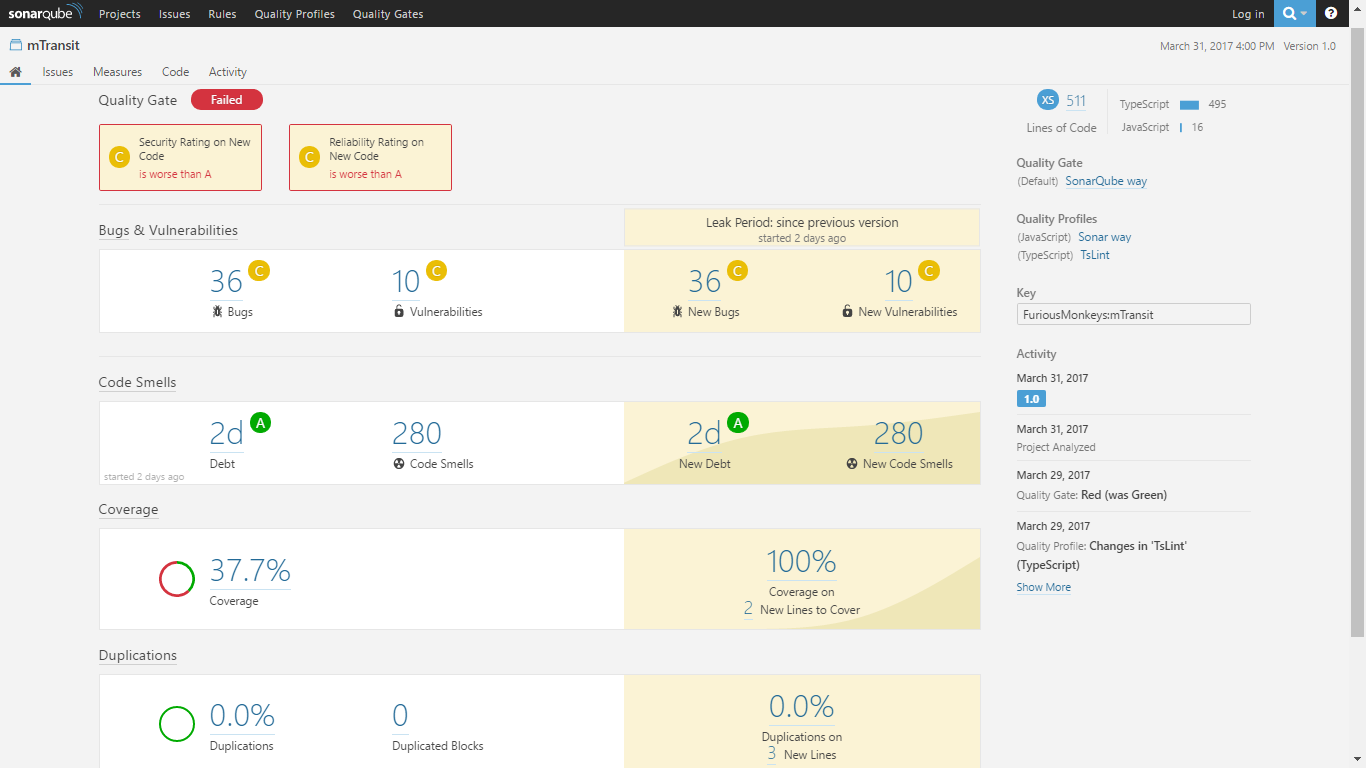
**Quality Metrics**

|  |
| --- |
| **Sprint 4 changes** |
| Added full SonarQube report from end of Sprint 3 to create comparable quality metrics |

We used SonarQube to keep track of the quality of our coding. It uses several metrics to evaluate the quality of the coding. SonarQube does not support the analyzation of typescript files. However, there exists a very recent open source plugin that does it (<https://github.com/Pablissimo/SonarTsPlugin>).

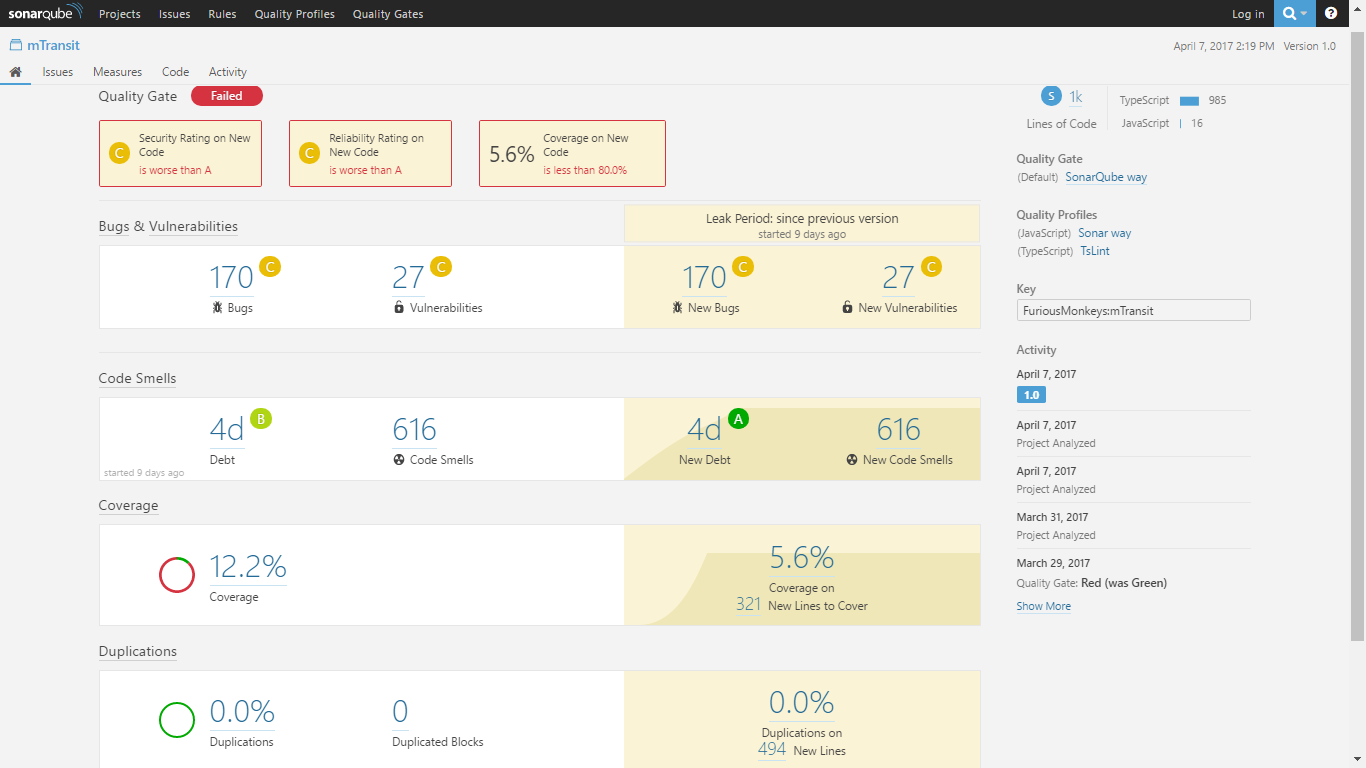
We integrated the different plugins needed to measure code coverage, in our case it was the use of an lcov.info file created by the ng test --code-coverage service. No duplications were found throughout our code as a segment of code needs to be at least 10 lines long and identical to another segment of code for it to be considered a duplication.

**Sprint 3 SonarQube Quality Report**



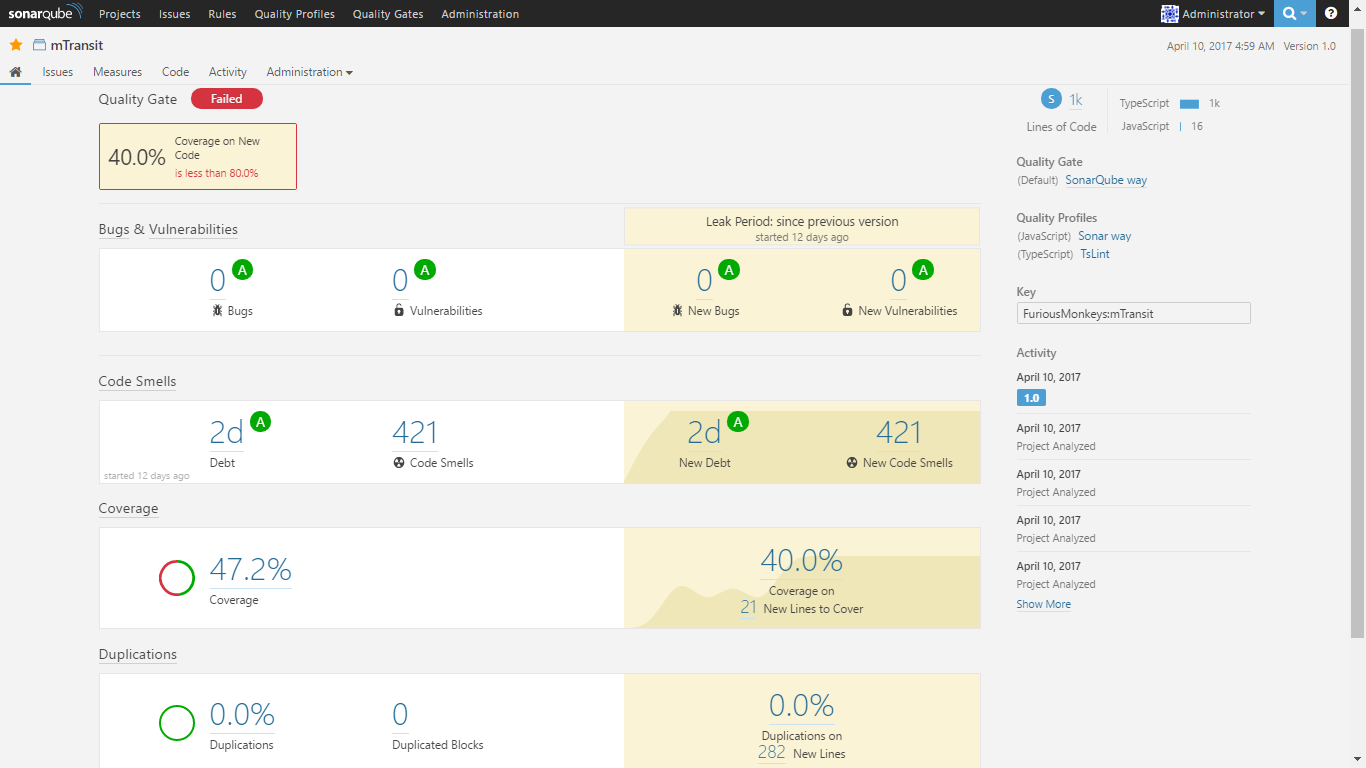
**Sprint 4 SonarQube Quality Report**

The below image shows our SonarQube results after adding in more completed user stories and tasks but before optimization and refactoring.



**Sprint 4 post refactoring results**

Below are the results of our final SonarQube scan with all of our finished user stories included and major refactoring having been done. We were able to clean the code and reach a quality level of A for lack of bugs and A for lack of vulnerabilities. Our code smells were also greatly lowered by properly formatting the code and allowing for easier maintenance. The code coverage was not ideal as we had some issues learning how to write the proper unit tests to manage line and branch coverage. We believe our low code coverage score may also be caused by our lcov data not being updated when our karma tests ran.



**User Interface Design**

|  |
| --- |
| **Sprint 4 changes** |
| **Added UI flow section** |
| **Updated UI images** |

Our target demographic are teens and young adults. Most of them enjoy going to the house of their friends or party downtown via bus transportation. Here are a few examples of the our target demographic:

1. Tyler is a 27 year old graduate in computer science. He wishes to hang out at a popular bar with his friends late in the evening, downtown. He lives in a neighbourhood where buses rarely pass when it is late. He needs a convenient and fast way to go downtown. Tyler loves that he can call a bus at a bus stop and can track it on his mTransit app. He can effectively know when it will appear so he can plan his walk to his selected bus stop accordingly. He enjoys the freedom to modify his destination.
2. Sam is a 21-year-old freshman in University. She recently moved into a downtown apartment and often spends her weekend evenings out with her friends. Neither Sam nor her friends have a car so they need to take the bus whenever they want to hang out with each other. Sam usually meets with her friends around 11 pm, so the mTransit service is especially useful to her. Sam is able to use the mTransit app to get picked up by a bus whenever she wants to meet up with friends, and can plan when she leaves based on the bus’s arrival time. Sam also really enjoys that she can meet her friends on the bus as the bus will detour to her friends’ bus stops instead of sending a second bus. This lets Sam and her friends all arrive together improving their night out. When Sam needs to get home, she is able to use the mTransit service again during the late-night hours. Sam is able to enjoy her night with her friends not having to worry about missing the last bus since the mTransit system is active throughout the night.
3. Billy is a 19 years old working downtown as a dishwasher. He can only leave work once the last customer leaves. So Billy generally leaves at 3am. Billy is also colorblind thus cannot distinguish red from green. For that reason, Billy avoids utility apps that uses red/green in their core functionality. Fortunately for him, mTransit uses a colorblind friendly palette. The colour layout of mTransit is Blue and White. While using the Map, the Bus stops and path is coloured black and the selected drop off bus stop is Blue. This way, Billy can use the app and still distinguish his drop off Bus stop to a default one.
4. Denise is a professional accountant who drives to her work everyday, as she does not like the time public transit takes. However, sometimes she feels like helping the planet, so she occasionally takes the public transit in the evenings. However, her main mode of transportation is still her car. She is bilingual, but as a native Quebecer, she likes it when french language options are available.
5. Ed is a wheelchair-bound teenager who has piano lessons every Wednesday evenings. His parents believe that he should be independent and capable so they refuse to coddle him by driving him to these lessons. Therefore, Ed takes the bus every Wednesday evening. However, the buses on his route are not always wheelchair friendly which greatly saddens Ed. That is why mTransit has the option of selecting a wheelchair friendly bus to pick someone up.

Our UI was designed with the above personas in mind. We chose colors that are colorblind friendly, covering the most common types of colorblindness (red/green and blue/yellow). We have also added a language change option since many people in Montreal are francophone and would love to use the app in their native language. This is in the login page, the first page one has access to as the user would like to switch as soon as possible. The font we have chosen is standard and professional looking. We want to avoid fonts like “Comic Sans MS” or “Papyrus” because those look childish and thus makes our app look amature and undesireable. We decided to put the search option at the top of the app, because we did not want people accidentally pressing on it if they were busy using the app while also doing something else, such as interacting with their friends or if they were intoxicated. This is also why UI elements will be easy to press and not need precision pressing.

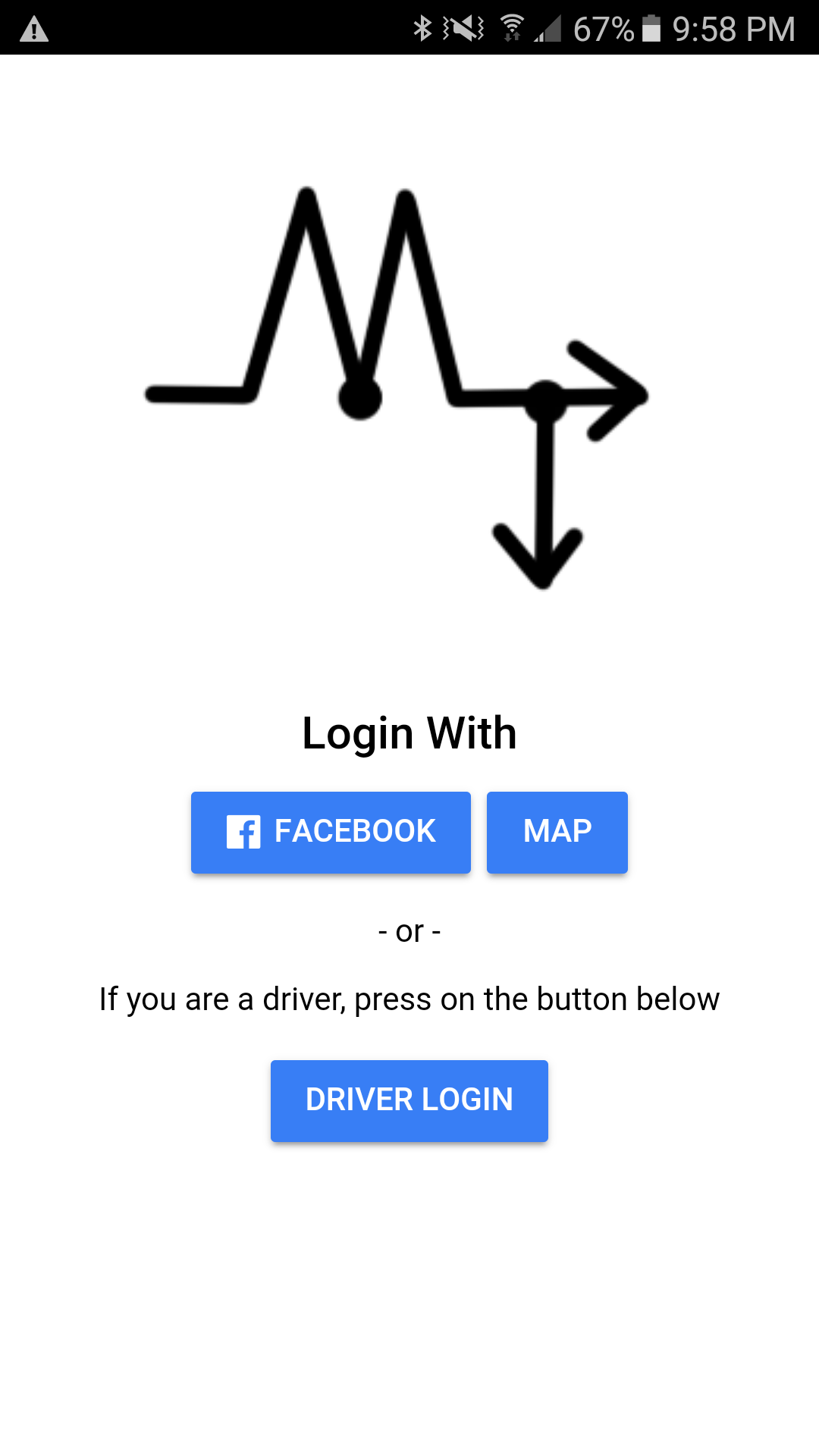
1. Login Page

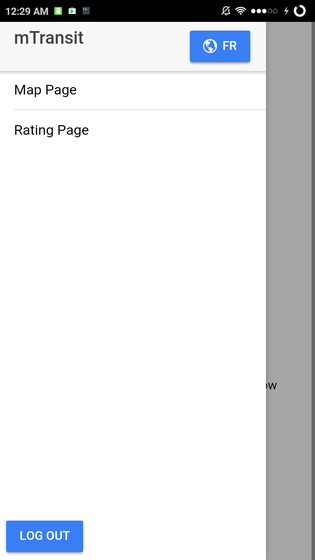
EP-1: As a rider, I want to login through Facebook~~, Twitter or Google+~~ so that I don't need an mTransit account.

EP-9: As a driver, I’d like to log in with a code given to me by my employer so that I am verified as an employed STM driver.

EP-12: As a user, I’d like to be able to change the language so that I can use the language that is most familiar to me.

EP-1: Initial Login Page EP-12:Login Page (French)





**Description:** A sidebar that users may bring up by swiping to the right. They can use this to navigate to the map page and the rating page. Additionally, they may switch the language of the app by pressing on the language button.

When the Facebook button is pressed, the first time, the user will be prompted to let the app have access to their Facebook account. Each subsequent time, pressing the button simply lead them to the map page.

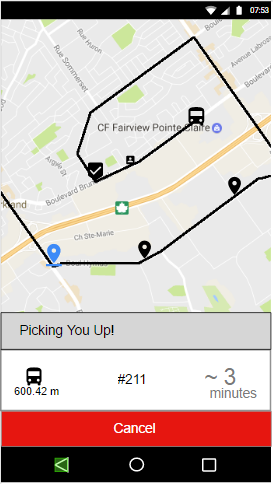
The user will also be able to select press the language button to alternate between English and French text. The change language button can be found if the user puts their finger on the left border of their mobile phone and swipe to the right. This phenomenon is displayed in the last image.

2. Waiting for Bus Page

EP-3: As a rider, I want to know how long it will be until I am picked up so that I can plan my time accordingly.

EP-5: As a rider, I want to change my drop-off location while on the bus so that I can get to a different destination.

EP-3: Rider Page (Waiting for Bus)



**Description:** The bus icon on the map represents the current location of the bus that is going to pick up the rider. The person icon represents the rider’s current location. The check mark icon represents the location of the bus stop where the rider will be picked up. The black pins represent the bus stops on the same bus route as the bus picking up the rider. The underlined blue pin represents the selected destination. The black line on the roads of the map represent the route of the bus picking up the passenger.

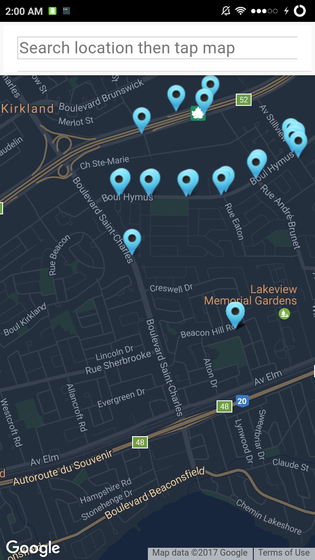
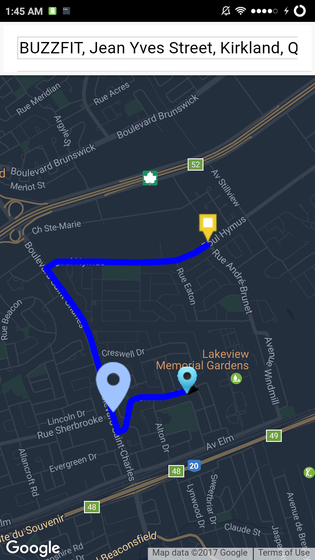
3. Choosing Destination

EP-4: As a Rider, I’d like to be able to select my destination bus stop so that I can plan my trip

EP-6: As a rider, I’d like to be able to see all bus stops near me so that I can plan my trip accordingly

EP-4: Choosing a Destination Stop

EP-6: Viewing nearby bus stops EP-4: Confirming the Destination Stop

**Description:** The light blue pins represents the bus stops on the map. The user taps a part of the screen to have the bus stops in that area appear. The user can also drag the map around to see other areas as well as using the search bar to find a location. This is shown in the image on the left. This will help users by allowing them to search for where they want to go via Google’s autocomplete then tap the area they want to go to see the pegs.

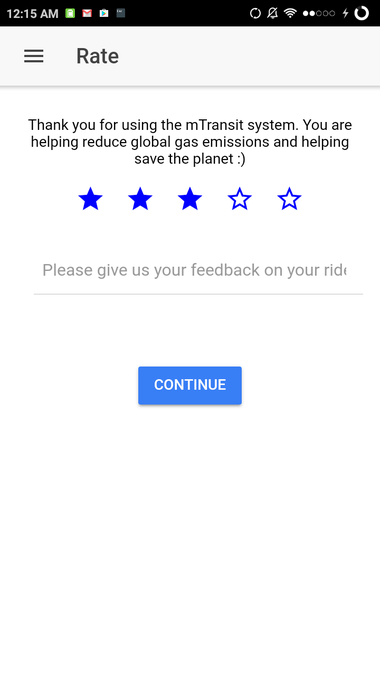
When the rider double taps a light blue pin, it will turn yellow and a new big purple pin will appear. The purple pin represents where the user needs to go to be picked up, and the square yellow pin is the destination. A path is also drawn between the user’s current location, their pick up stop (big purple pin) and the destination (square yellow pin) showing both the route to the pick-up stop as well as the bus’s route to your destination.

4. Rating Trip Experience

EP-7: As a rider, I’d like to be able to rate the route I have taken so that I can give my feedback.

EP-7: Leaving a Comment

EP-7: Giving a Star Rating



**Description:** The rider rates his experience using our app on a scale of 5 stars. He does this by touching a star, the star to the far right is the best score. When he touches a star, all stars on the left gets filled in blue he can leave a comment in the text section below. After entering their feedback, they press the continue button.

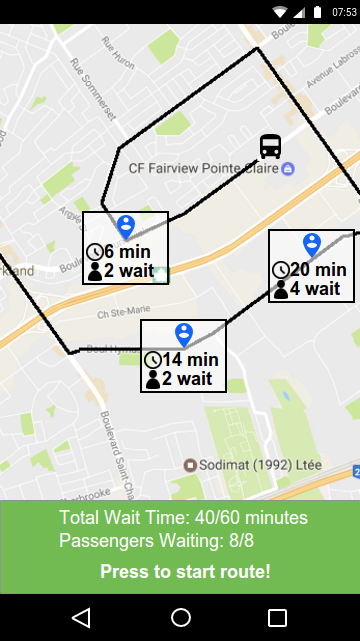
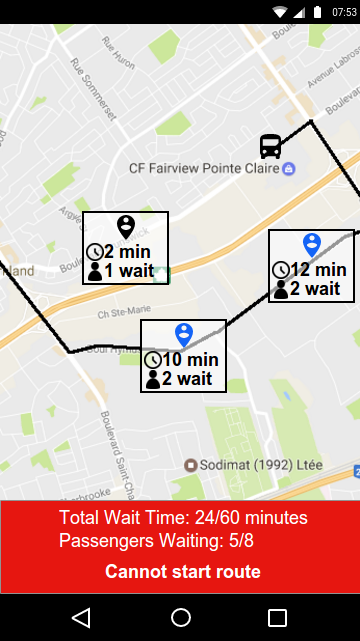
5. Driver View

EP-8: As a Driver, I’d like to take the fastest and most economical route so that I can save time for riders and produce fewer emissions.

EP-10: As a driver, I’d like to know how many people are waiting at each stop so that I can plan my stops accordingly.

EP-11: As a driver, I’d like to cover for another driver when there are not enough people on that line so that there are no buses that are needlessly running.

EP-10: Waiting to Start Route EP-10: Starting Route

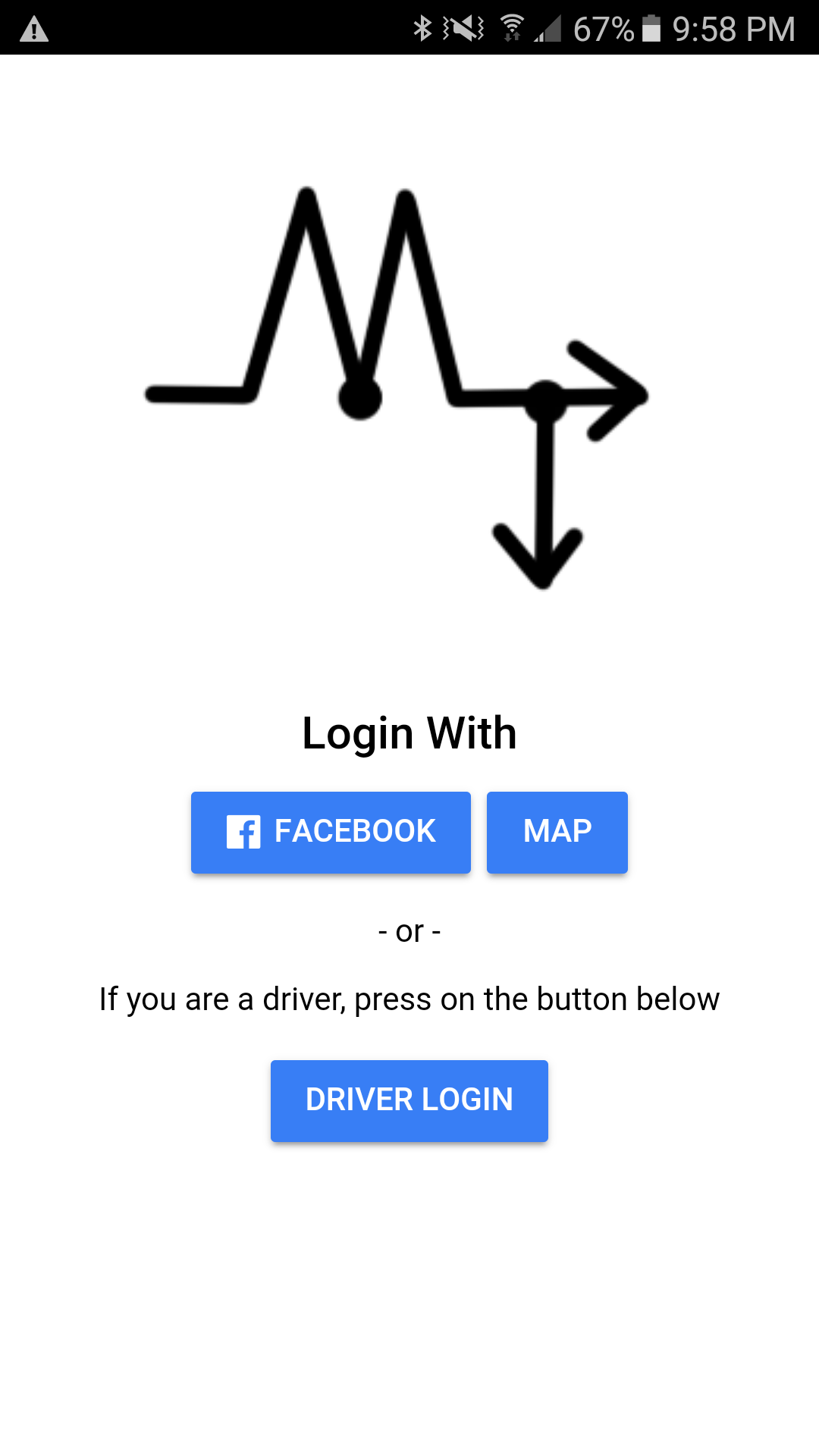
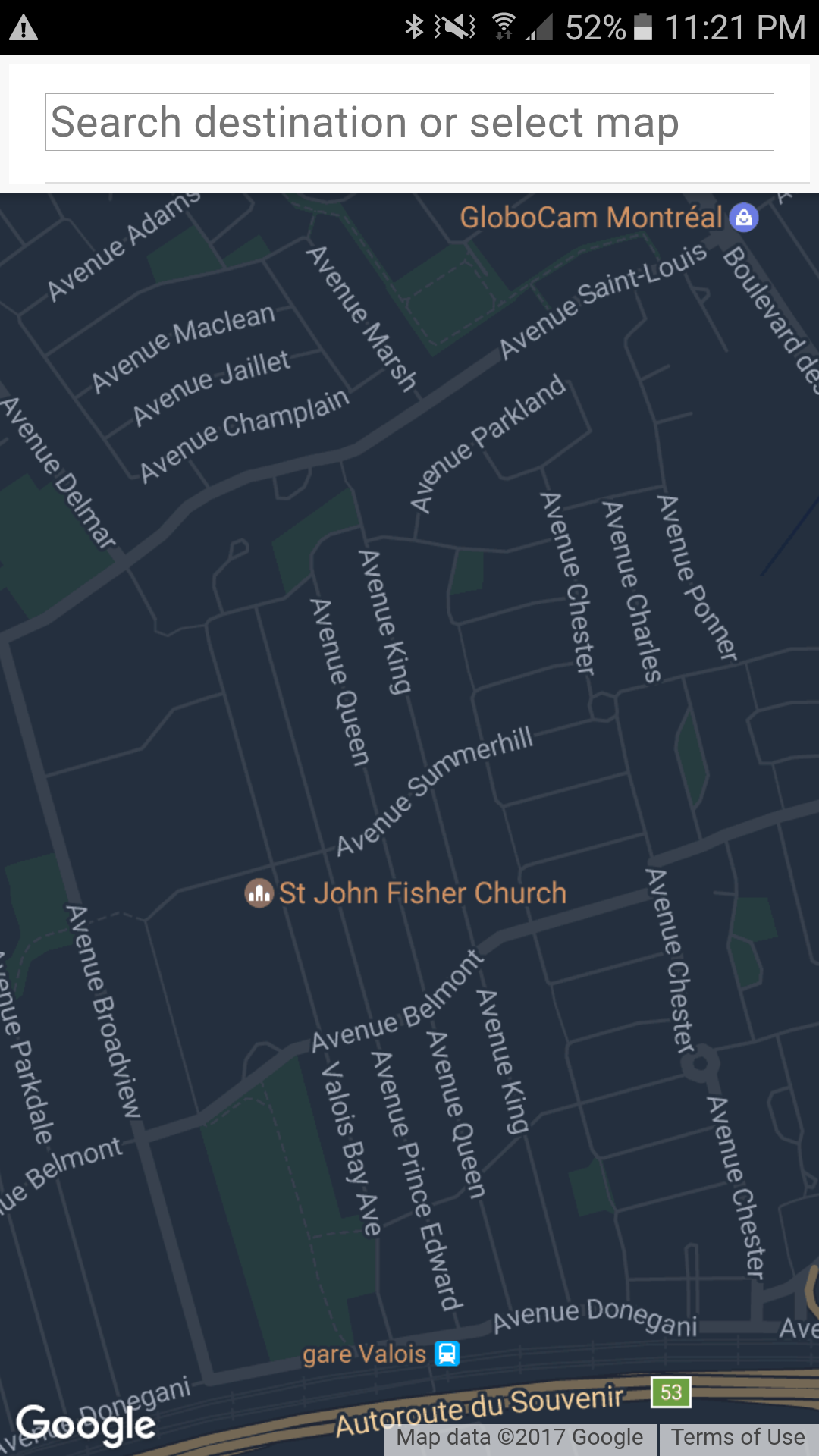


**Description:** The person pins represent riders that should be present at a bus stop and the number above the person pin is the number of riders at that bus stop. The blue person pins are the riders that are on the current route of the bus that is going to pick the rider up. However, the black person pins represent the riders that are not on the current route of the bus but will be added soon due to the bus rerouting algorithm.

The interface on the left is shown when the number of passengers waiting does not meet the required minimum, or the total passenger wait time has not exceeded the maximum. The red field tells the bus driver that they cannot start the route.

The interface on the right is shown when either the minimum number of passengers is met or the total passenger wait time has exceeded the maximum. The driver can then start their route by pressing on the green field at the bottom. The bus route to be taken is displayed in black on the roads. The black person pin has also become a blue pin and has been added to the route.

UI Flow

The first screen the user sees is the Login screen. On that screen, the user may choose to be authenticated through Facebook or may proceed to the map page directly.

The next screen the user sees is the main page, which features our map. When the user zooms in sufficiently on the screen and presses the map, they will see the map markers which indicates bus stops.

They may then press on the marker which will then turn yellow. This will be the destination point. After that, they must pick a starting marker, from which a route will then be generated. The user may choose to search for a location using the search bar on top of the screen.