MissIt Rides: A Proof of Concept Application for MissIt Technology

Henry Caron and Nikhil Bhatia-Lin

May 14, 2021

MissIt Rides is a ride-share application built to demonstrate one potential instance of the utility of the low rate MissIt data communication channel. In this paper we detail the implementation of the MissIt Rides application, which is constrained by MissIt and our API interactions. With these constraints in mind, we have attempted to optimize user experience by introducing new features not typically found in ride-sharing apps. Our results show that this application has potential in specific use cases. In order to be more widely applicable, it is necessary to further develop its key components.

1 Introduction

In this paper, we present MissIt Rides, a novel application prototype based on the extremely low bit-rate communication channel MissIt. MissIt was developed by a small team led by Dr Fahad Dogar out of Tufts University. It uses missed calls as a data communication channel. MissIt modulates the duration of a missed call, allowing for a throughput of .3bps and error rate of less than 10^{-2} [2]. Our novel prototype, MissIt Rides, is a mobile application that allows a user to request a ride-share. It functions similarly to traditional ride-sharing applications. In the typical use case, a user selects pickup and drop-off locations, chooses from a list of potential rides, and confirms their ride. Due to the constraints of the MissIt network, the user can only pick from a set of predetermined pickup and drop-off locations. In addition, the application takes longer to find and confirm a ride than a traditional ride sharing app. The application also includes favorite and recent locations to minimize wait times for commonly selected destinations.

This paper will perform an in-depth explanation and analysis of the functionality of MissIt Rides. Then it will discuss our final user study, which investigated necessary elements of the app and peripheral app features. Finally, it will discuss challenges existing in the current iteration of the app that future researchers will need to solve.

2 Background

The MissIt Rides project began as an investigation into potential ways to optimize MissIt communication. On its own, MissIt acts as a standard messaging service that takes extended periods to send messages. This project set out to find specific use cases in which MissIt could be applied while minimizing sacrifices in wait time.

2.1 Smart Reply

The first burgeoning technology we analyzed for use with MissIt is smart reply, which was widely used first on Gmail and now has been incorporated into mobile messaging services [4]. Smart reply uses natural language processing to predict the responses that a user might want to make to a message. The best smart reply modules have high variability, relevance, and specificity [4].

It soon became clear that smart reply on its own is not viable for a messaging service. Conversations reliant of smart reply quickly degraded to simple messages without variability or relevance. For this reason, smart reply performs best as a supplement to preexisting messaging apps. For an app that relies on MissIt, it would be impossible for users to communicate effectively with it.

2.2 Text Similarity

Text similarity is a classic application of natural language processing. It a machine learning algorithm that scores the similarity of two phrases out of 100 [3]. A potential MissIt-based app that would use text similarity to function would have a database of pre-written messages. When a user enters a message, the app identifies the database message that scores highest on the text similarity algorithm. This message would then be sent.

While text similarity is good for short, general messages, the database required to handle long, complex, or specific messages quickly becomes difficult to maintain. In addition, there are words and phrases specific to users or situations that cannot be captured by a pre-written database of messages. For these reasons, we believe it is unsuitable to create a messaging app based solely on a text similarity algorithm. It would be necessary to create a custom algorithm that incorporates text similarity but also allows for phrases unique to the user.

2.3 Chat Bot

Chat bots are commonly found in online customer service bots, and combine text similarity algorithms with predetermined conversation paths to produce semi-realistic messaging with a computer user. These conversations are typically short and generic, and prepare the user to chat with a real customer service agent. Chat bot architecture is promising for a MissIt-based messaging application because it allows specific keywords to combine with a general text similarity algorithm. In addition, the structure of specific conversation paths reduces the variability of potential responses, which can cut down on data that is required to be sent.

For the reasons above, the initial idea we generated for a prototype app using MissIt was a chat bot-based messaging service. Since chat bots are used in highly constrained environments, our app would focus on a highly specific messaging scenario that feature no more than 5-6 messages between users. However, after some discussion, we decided to abandon this avenue. We believe such a messaging application would be too limiting within the scope of this project, and instead looked to applications of MissIt that were not purely messaging-based.

2.4 MissIt Rides

As an alternative to a messaging application, MissIt Rides was a promising project idea because ride sharing apps are commonly used around the world, it is feasible to create a ride-share app build with MissIt, and there are challenging technical obstacles to overcome. In addition, research revealed that no applications currently exist that have the same unique parameters as MissIt Rides: an app that allows for the free placement of rides for users without internet connection. This increases the likelihood for future use of MissIt Rides in developing regions. In addition, down the road, there is the possibility of including a simple chat interface that allows drivers and riders to communicate, using an architecture similar to chat bots described above. For the reasons described above, we set out to implement MissIt Rides.

3 Implementation

3.1 Users and Cached Data

A primary focus of our application is to minimize the data communicated from the offline device to online server. In a typical scenario, a user will select a pickup location, destination, and request a quote for a ride. In order to minimize the amount of data transferred between the device and server, the user sacrifices some autonomy. Instead of being able to choose any location in the world as a destination, the app only allows the user to pick from a set of predetermined nearby stops. These stops are generated automatically by an intelligent algorithm described below. Alternatively, the user can set favorites in advance, and has access to his or her recently visited locations. To this end, we associate a number of different attributes with each user.

3.1.1 Nearby Locations Algorithm

Nearby locations in the user's geographic proximity and are regenerated automatically whenever a user is more than ten miles away from the nearest location. These locations are produced via Google's Place Search API, specifically the Nearby Search request. In the standard implementation of this feature, the density of locations is highest closer to the users current location, and locations become more sparse the further away from the users location they are. Whenever a user travels to or from of these locations, the location is added as a recent location which allows it to be referenced as a single digit index. In addition, transportation locations are set as to be generated by default, as we expect users to want to travel to places like airports and train stations at a higher rate than other locations.

We track user type preferences in order to generate better lists of nearby locations over time. Each place Google returns comes with an associated "type;" there are 100 of these categories which range from "gym" to "convenience store" to "aquarium." Every time a user requests a ride to a nearby or recent place, we increment the count of each type associated with the place in the users database. Over time, this leads to the formation of favorite types, which we take into account when a new set of nearby locations is generated for a given user. This provides the user with a customized set of potential locations to travel to just based on their travel history.

3.1.2 Cached Data

In addition to normal locations, the user has access to locations that require far less wait time. These locations will be the places the user is most likely to travel to, minimizing the expected time a user will wait for any one ride request. Cached locations can be divided into two categories: favorite and recent. Each user of our application has the ability to select and save up to five favorite locations. Upon the addition of a new favorite location, the user is required to wait while the coordinates of the favorite are passed via the MissIt channel to the database. Once a location has been saved as a favorite, the cost in time of requesting a quote to get from one favorite to another is reduced significantly compared to a normal ride. This is because the system just passes a user id and two single digit favorite indices to the quote request route.

Similarly, the five most recently visited locations are saved on both the user's device and the server at the time that a ride is placed. These locations are cached with special indices, meaning if a user chooses to travel to one of these locations again, waiting time is considerably less than a typical ride.

3.2 User Interface Considerations

MissIt Rides adds unique constraints to a familiar app concept. For this reason it is essential that the user interface clearly and intuitively communicates the app's additional features. Some of the front-end solutions we implemented are described below.

3.2.1 Familiar User Interface

While MissIt Rides uses a communication method foreign to many of its users, ride sharing apps are heavily used. Therefore, it was important for the application to look and feel familiar. For that reason, the design and user interface of the app is based heavily off popular ride-sharing apps Uber and Lyft. The app includes a map from which to choose locations, menus to update favorite locations, a screen to view recent locations, and a screen to display potential rides.

3.2.2 Design of Pins

One of the largest drawbacks of using MissIt Rides is that it has the potential to take several minutes to communicate with a server about where the user is going. For that reason, we designed custom pins on the map that demarcate if a location will take a long time to retrieve or not. Our pins are designed as follows: These markers

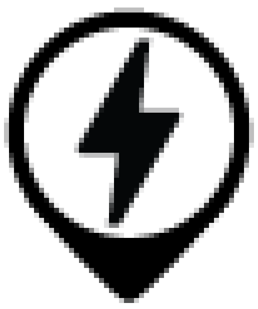


Figure 1: Fast Pin

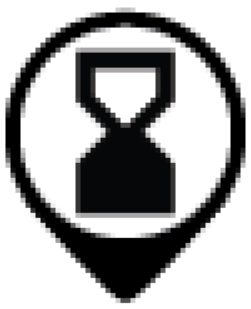


Figure 2: Slow Pin

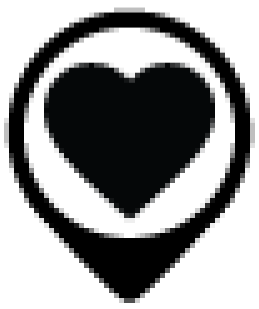


Figure 3: Favorite Pin

have been designed to provide subtle information to the user, without sacrificing space on the map or elsewhere in the app. After some feedback from our user study, we believe it is necessary to have a tutorial page or small key to explain what the markers represent. Many users failed to understand intuitively that a fast marker meant a shorter wait time than a slow marker.

3.2.3 Long Press to Get Nearest Location

MissIt Rides includes a feature for which the user long presses on their destination. The app will automatically find the nearest pin on the map to that location and set the destination to that point. This feature is helpful for users who prefer to choose their destination over selecting from a pin.

As with unique markers, users often missed this design feature. We believe the long press is not an intuitive action for selecting locations on a map. Additionally, the instructions written on screen were not prominent enough for users to read and understand. For this reason, it will be important for the next iteration of the app to take this feature and make it simple and intuitive for users to understand.

3.3 Ride Service API

For the purposes of this project we integrated with the Taxicode API for ride quotes and requests [1]. However, we hope that in the future, the app will be able to utilize a globally popular service like Uber or Lyft API to keep costs low for the user and to provide an experience that they will be more comfortable with.

4 User Study

4.1 Hypothesis

We predict that our app will allow users to schedule short range rides (within a 2300 meter radius) that requires less than 500 meters of total travel not spent in the ride. We predict medium-range rides (within a 6500 meter radius) that requires less than 1000 meters of travel not spent in the ride. We predict long-range rides (outside 6500 meter radius) will require more than 1500 meters of travel not spent in the ride.

Additionally, we predict users will want to wait at most two minutes for our app to retrieve ride information before cancelling the ride request.

4.2 Pilot Study

Prior to our final user study, we performed a short, informal pilot user study with two individuals aged 18-24. This study was in interview style, with the user testing the application and talking through the motives behind their actions. We used this feedback to iterate on our user interface, specifically the design of our pins, the implementation of a press to get nearest preset location, and tweaking of buttons and navigation to make the app more intuitive.

4.3 Final User Study

4.3.1 Experimental Design

In order to test our implementation, we conducted a between-subjects study, in which we varied time to receive quotes and density of nearby locations generated. We tested 3 different timing delays (0, 30, and 120 seconds) and two different location densities for a total of 6 possible scenarios.

In the first case, nearby locations were generated with decreasing density as distance from the user's location increased. Specifically, in the first mile radius from the user locations were required to be at least 500 meters apart, in the second miles radius locations were required to be at least 1 km apart, in the four miles radius locations were required to be at least 2 km apart, and in the eight miles radius and beyond locations were required to be at least 4 km apart. In the second case, density of locations remained unchanged, with a minimum distance of 500 meters required between each location.

Half the subjects were assigned to use the version of the application with uniform location density, while the other half used the version which varied location density generation. Within each group, a third of the participants had a 0 second timing delay, a third had a 30 second delay, and a third had an 120 second delay. Subjects were randomly assigned to their condition and unaware of the versions outside of the one they used. We conducted our user study in person, with researchers monitoring the participants throughout the process and being available to answer any questions that arose.

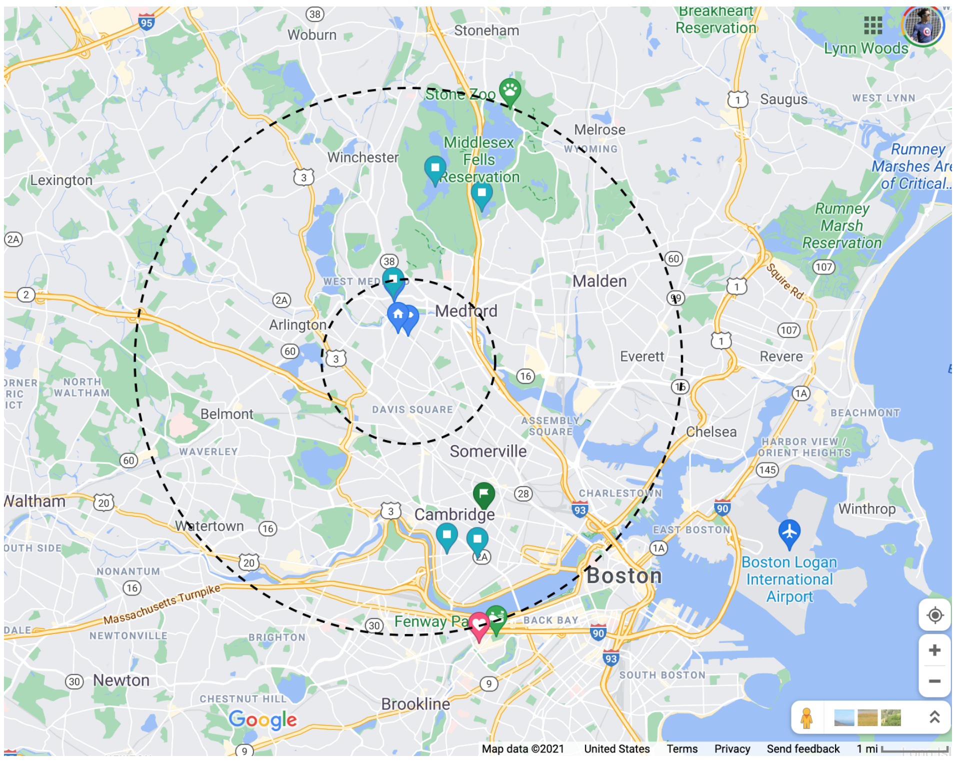


Figure 4: The map users were given to select a location from

At the start of the experiment, a participant was asked to choose 3 locations to travel to from 3 zones of increasing distance (see Figure 4). They were then asked to open an online survey. After filling out their personal information, they were handed a mobile device with MissIt Rides pre-opened and were asked to request a ride to a predetermined location within a 2300 meter radius. The user recorded both the location that was their target and the pinned location that they scheduled a ride to. They repeated this process two more times, with the second location being between 2300 and 6500 meters, and the third being more than 6500 meters away.

Upon completion of this section of the survey, the participant was asked to answer simple qualitative questions regarding their experience using the application. These questions focused on how far they would be willing to walk for a ride, as well as how long they would wait for an app to load ride information.

4.4 Participants

12 individuals (6 male, 5 female, 1 prefer not to specify) ages 18-24 participated in the study. All participants had experience using apps like Lyft and Uber, with 2 participants reporting 'a little' experience, and 9 reporting either 'a lot' or 'a great deal' of experience.

5 Results

5.1 Distance and Time Analysis

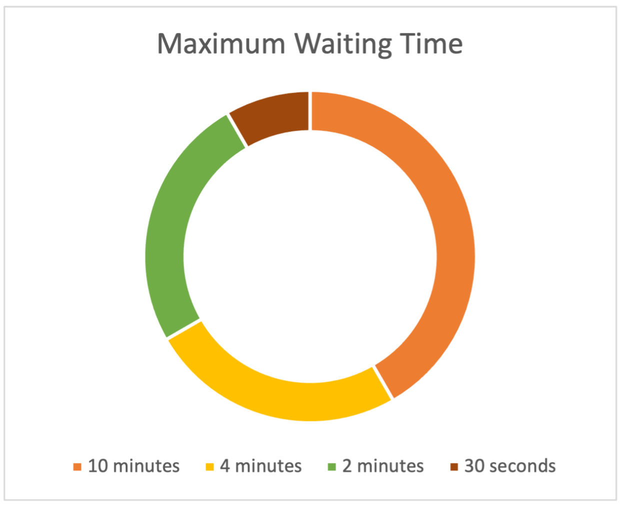
	Version 1	Version 2	Overall
$<$ 2300m ride	319	545	**432**
$<$ 6500m ride	1674	907	**1290**
$>$ 6500m ride	765	985	**875**

Table 1: Average Distance in Meters Between Target Location and MissIt Rides Dropoff Pin. Version 1 is with variable density of nearby locations, Version 2 is with constant density

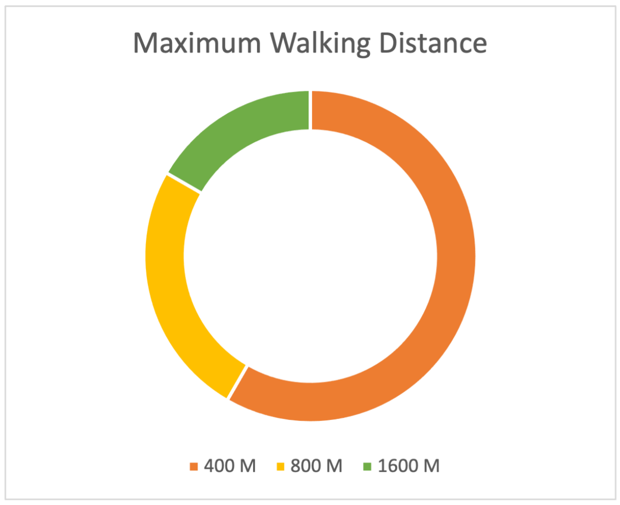
	Version 1	Version 2	Overall
$<$ 2300m ride	50%	16%	**33%**
$<$ 6500m ride	0%	16%	**8%**
$>$ 6500m ride	67%	33%	**50%**

Table 2: Percentage of rides which were called directly to the final destination. Version 1 is with variable density of nearby locations, Version 2 is with constant density

For each ride requested by our users, we calculated the distance between their final destination and the location they were able to request a ride to using MissIt Rides. Contrary to our hypotheses, we were unable to discern any significant trends when comparing the averages across different versions of the application. This is likely due to the randomness of whether or not users chose locations to travel to that were available to travel to directly via MissIt Rides (Table 2). Notably, a significant (30%) number of rides overall were able to be called directly to the user's target destination.



(a) The maximum time users reported being willing to wait for quotes imagining they had no internet access



(b) The maximum distance users reported being willing to walk from their drop-off to their final destination

Figure 5: Maximum distance and time trade-offs as reported by users

We surveyed users to gain more clarity on two key design constraints of our application: how long users would be willing to wait for quotes and how far users would be willing to walk to get to their final destination. These questions are inherently tied to the key trade-off we were considering in our design: balancing utility (number of available locations) with performance (the amount of data we need to store on either end and how long this takes to transmit via MissIt).

5.2 Qualitative User Feedback

When asked whether they ran into any issues during their experience, 5 of 12 users reported no. Seven users reported having some form of issue, which varied. Recurring issues users experienced are as follows:

- It was hard to find locations using just a map.
- "To" and "From" selection buttons switched after location selection in a way that was counter-intuitive.
- Long press feature was unclear/confusing.

6 Discussion

6.1 Contributions

We believe that the features we have built into our application make it feasible for users to access some of the more popular locations in the Somerville/Medford area. We noticed a trend of users wanting to travel to transportation hubs such as Logan Airport in Boston (5/12 Users) and local cultural centers like Davis Square (7/12 Users). This observed user activity supports our implementation, which prioritized commonly visited locations and transit centers. The fact that many of our users picked these destinations is promising for some of our main use cases: people wanting to access the locations described above.

Our results reveal that the median distance users will be willing to walk from where they are dropped off to their final destination is $1/4$ mile, or about 400 meters. For rides that are less than 2300 meters, the average distance between target location and drop-off pin is 432 meters, meaning it is on the border of being seen as worthwhile by the average user. Rides that are beyond the 2300 meter cutoff have a difference of more than double 400 meters between destination and drop-off location. This result reveals that the current version of the app is unfeasible for longer distance rides that are not to a favorite or recent location. For this reason, the current optimal use case for our app is for a user to travel under 2300 meters, to a somewhat close destination. In order to increase the utility of our application for longer rides we will need to either increase the density with which locations are generated or figure out a more effective method to predict and supply locations for users to travel to at these distances.

In addition, users revealed that they would be willing to wait for a median of four minutes for an app to load potential rides in their area. Using our design, if a user selected two non-cached locations, they could expect to wait 288 seconds, or just under five minutes. For each cached selection the user chose, they could expect to wait for around 48 seconds less. If a user were to choose two cached locations, they would wait for 192 seconds, or just over 3 minutes. Based on our user study, this wait time would be reasonable for the user. For this reason, we believe our current wait time implementation could be feasible for users without internet access.

6.2 Limitations

Our application is targeted towards users in developing regions. All user study participants are American citizens attending Tufts University, who do not have experience with intermittent or limited internet connection. We attempted to mitigate this issue by asking users to imagine themselves in a non-internet environment. Regardless, we believe our results are biased due to our sample population. A more accurate user study would have been performed with users from a developing region, but due to time and lack of connections, our only option was with people around Tufts.

Additionally, we found minor issues with the methodology of our study. One of our questions asked the maximum time a user would wait for the app to load ride options. Despite being told to assume they had no access to internet, some participants were confused about whether that should apply to this question.

Our user study prioritized the effectiveness of MissIt Rides' location generation algorithm and wait times. Due to time constraints and experimental consistency, we opted not to test for other important features, such as recent and favorite locations, as well nearby location refresh. These parts of the app are necessary for proper functionality, so it is important for them to be properly user tested in the future.

6.3 Future Work

Our user study revealed that our application is several key areas to improve upon in further iterations. For this reason, all code will be made open-source for a future development team to build upon.

6.3.1 Increase Power of Location Algorithm

The current state of our nearby location generation is constrained by the limited parameters of the Google Places API. Specifically, the Google Places API only allows for a maximum of 20 results to be returned at a time, which means that in order to retrieve results over a larger area we need to manually make multiple individual calls to the API. While the high cost of this operation is offset by caching these locations once they are retrieved, decreasing the cost of this process would allow for more frequent and dynamic retrieval of locations which could be more tailored to user preferences.

6.3.2 Location Search Feature

A location search feature is one of the most important features our application is missing. This would allow the user to input their destination, and the app would automatically find the nearest pickup or drop-off spot. Our user study revealed numerous instances when a user would name a location to which they would like to travel and be unable to locate it on a map. We observed many of our users opening up the list view of location selection and either scrolling to try and find a search function or explicitly asking if there was a search feature. If this app were to be used by an individual in a new area (like a tourist or business traveller), they would have this same issue. Additionally, it is ubiquitous for popular ride sharing apps to include a location search feature. For these reasons, we believe it is essential for a final version of this app to allow users to have typed input. However, due to the nature of the application, it would be challenging for a user's mobile device to store every address in their region without access to the internet. Common features like auto-complete and multiple address options would also be difficult to produce without access to the internet.

We would recommend that the next development team store all street and landmark names, along with their respective coordinates, instead of every possible address. This would allow the user to search by street or major landmark near to their desired location, and use that context to find exactly where they want to travel. This would be a compromise as compared to typical internet-enabled search features found on modern ride sharing apps. Regardless, including this feature will be critical for the app's usability down the line.

6.3.3 Offline map

Currently, our map feature uses a module that connects to Google Maps over the internet. Since the application needs to remain offline, this feature must evolve for the app to continue to be usable. There are already numerous free offline map download APIs existing online. We believe that this feature will not be difficult to implement, but it is necessary for the app to work properly.

6.3.4 User Interface

As touched upon above, the user interface for MissIt Rides includes unique features. These include special icons to denote wait times for certain types of locations and long press to find the nearest location to you. While each of these features are promising, neither are intuitive. Specifically, we noticed in our observing our participants over the course of our user study that despite us including instructions for the long press on the home screen, none of our participants noticed or used the feature. In future iterations of MissIt Rides, it will be essential to make the meaning of special icons clear and intuitive from the outset. Additionally, the feature that allows the user to tap any place on a map and have the app automatically find the nearest preset location needs to be redesigned for better user experience.

7 Conclusion

Based on user feedback, we believe there is potential for MissIt Rides to be a feasible use case for MissIt. The application is still very early stage and requires development in both user experience and back-end logic. Regardless, our user study revealed that the basic use case of MissIt, selecting pickup and drop-off locations from predetermined pins in conjunction with long wait times, is realistic for use in developing regions. We look forward to seeing where this project goes next.

References

[1] [n.d.].

[2] Fahad R. Dogar, Ihsan Ayyub Qazi, Ali Raza Tariq, Ghulam Murtaza, Abeer Ahmad, and Nathan Stocking. 2020. MissIt: Using Missed Calls for Free, Extremely Low Bit-Rate Communication in Developing regions. (2020), 12. https://doi.org/10.1145/3313831.337625

[3] Wael Gomaa and Ali Fahmy. 2013. A Survey of Text Similarity Approaches. (2013), 18. https://doi.org/10.5120/11638-7118

[4] Anjuli Kannan, Karol Kurach, Sujith Ravi, Tobias Kaufmann, Andrew Tomkins, Balint Miklos, Greg Corrado, Laszlo Lukacs, Marina Ganea, Peter Young, and Vivek Ramavajjala. 2016. Smart Reply: Automated Response Suggestion for Email. (2016), 10.

Appendices

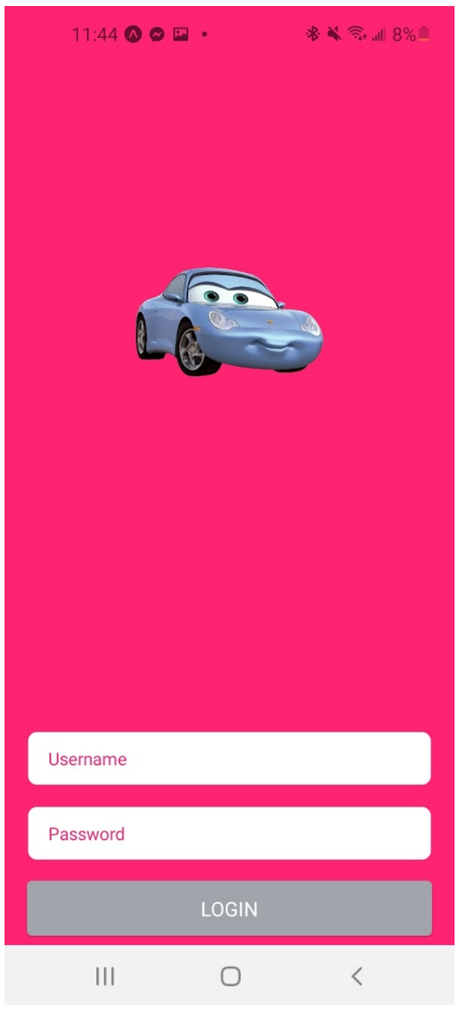
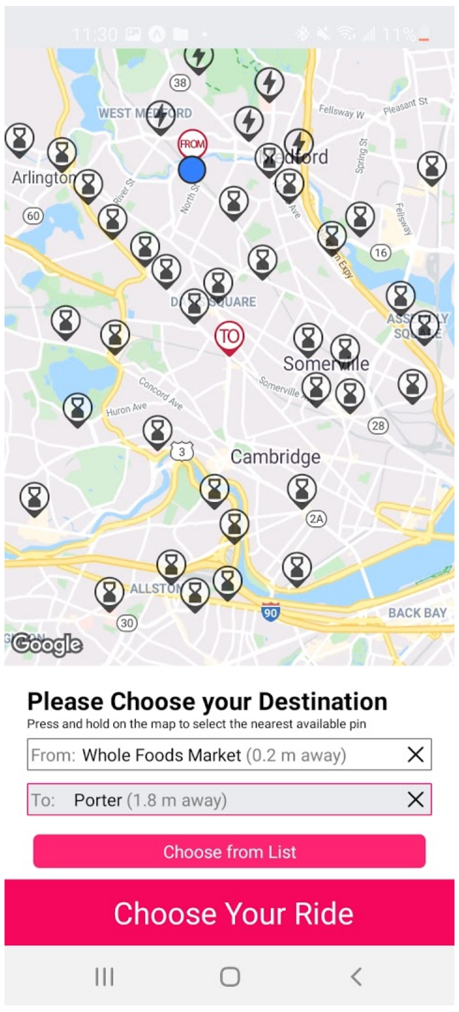


Figure 1: Login Screen



Please Choose your Destination
Press and hold on the map to select the nearest available pin

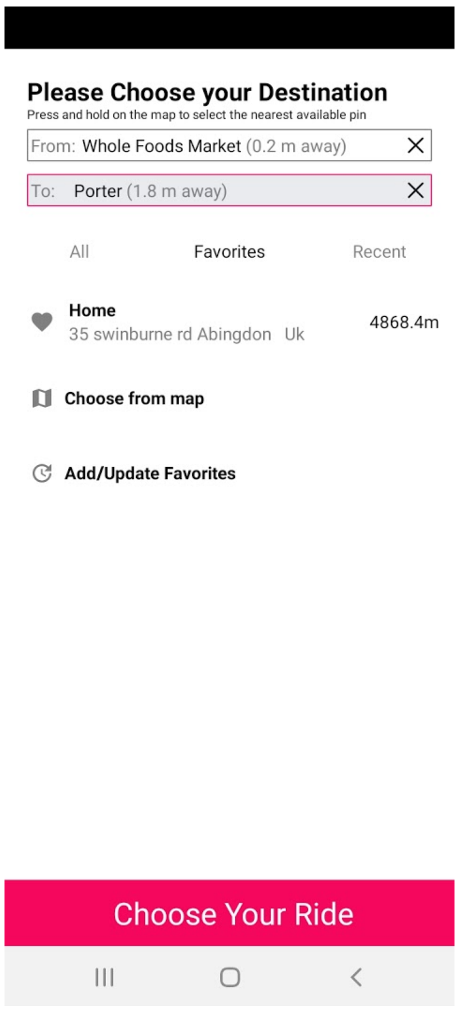
From: Whole Foods Market (0.2 m away)

To: Porter (1.8 m away)

Choose from List

Choose Your Ride

Figure 2: Home Screen (Map View)



Please Choose your Destination
Press and hold on the map to select the nearest available pin

From: Whole Foods Market (0.2 m away)

To: Porter (1.8 m away)

All Favorites Recent

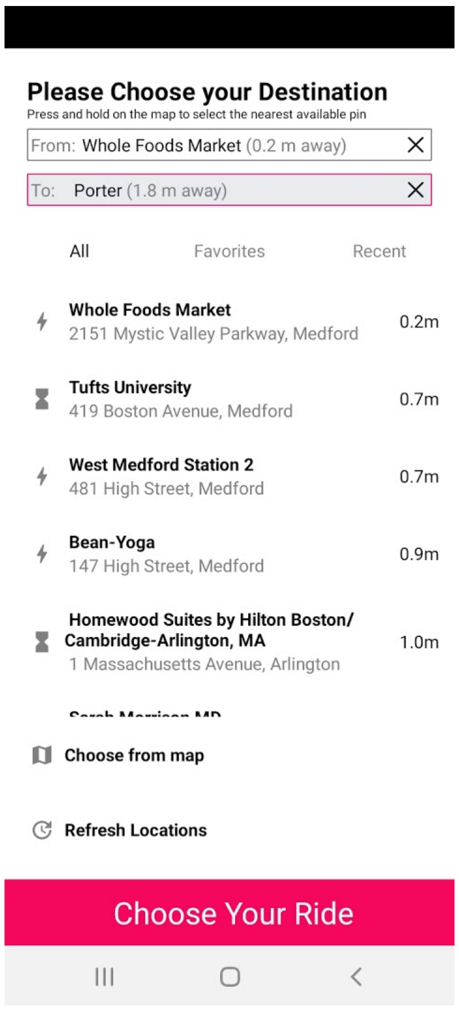
Home 4868.4m
35 swinburne rd Abingdon Uk

Choose from map

Add/Update Favorites

Choose Your Ride

Figure 3: Home Screen (Favorites View)



[redacted]

Please Choose your Destination

Press and hold on the map to select the nearest available pin

From: Whole Foods Market (0.2 m away) ✕

To: Porter (1.8 m away) ✕

All | Favorites | Recent

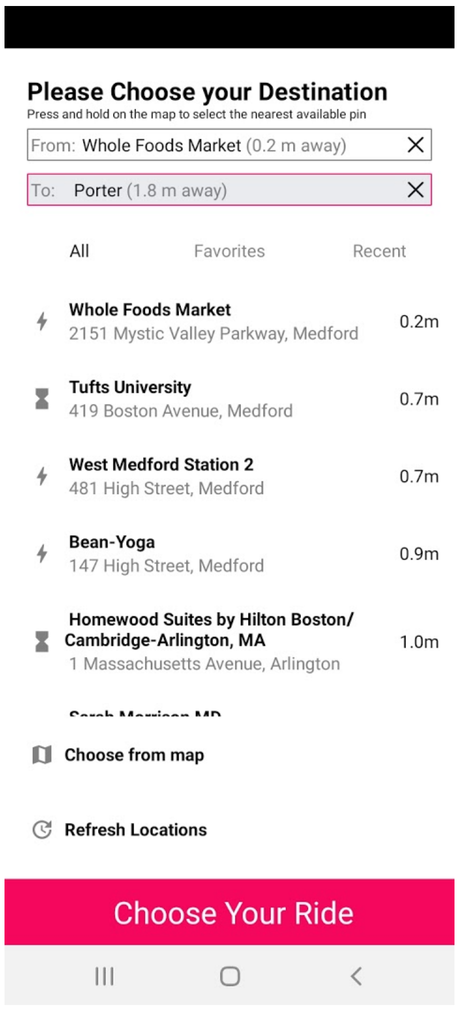
	Whole Foods Market 2151 Mystic Valley Parkway, Medford	0.2m
	Tufts University 419 Boston Avenue, Medford	0.7m
	West Medford Station 2 481 High Street, Medford	0.7m
	Bean-Yoga 147 High Street, Medford	0.9m
	Homewood Suites by Hilton Boston/ Cambridge-Arlington, MA 1 Massachusetts Avenue, Arlington	1.0m
	Sarah Morrison MD	

Choose from map

Refresh Locations

Choose Your Ride

Figure 4: Home Screen (All Locations View)



[redacted]

Please Choose your Destination

Press and hold on the map to select the nearest available pin

From: Whole Foods Market (0.2 m away) ✕

To: Porter (1.8 m away) ✕

All | Favorites | Recent

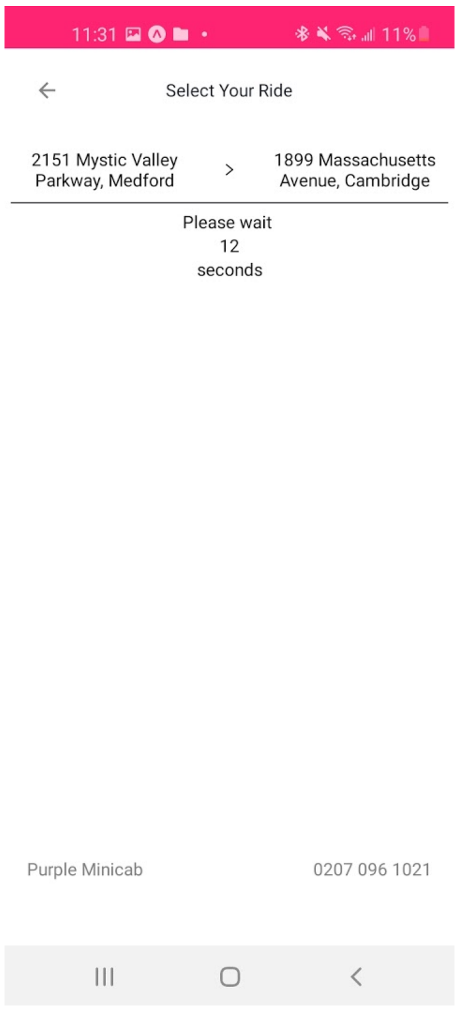
	Whole Foods Market 2151 Mystic Valley Parkway, Medford	0.2m
	Tufts University 419 Boston Avenue, Medford	0.7m
	West Medford Station 2 481 High Street, Medford	0.7m
	Bean-Yoga 147 High Street, Medford	0.9m
	Homewood Suites by Hilton Boston/ Cambridge-Arlington, MA 1 Massachusetts Avenue, Arlington	1.0m
	Sarah Morrison MD	

Choose from map

Refresh Locations

Choose Your Ride

Figure 5: Ride Screen (Waiting for Ride)


11:31 11%

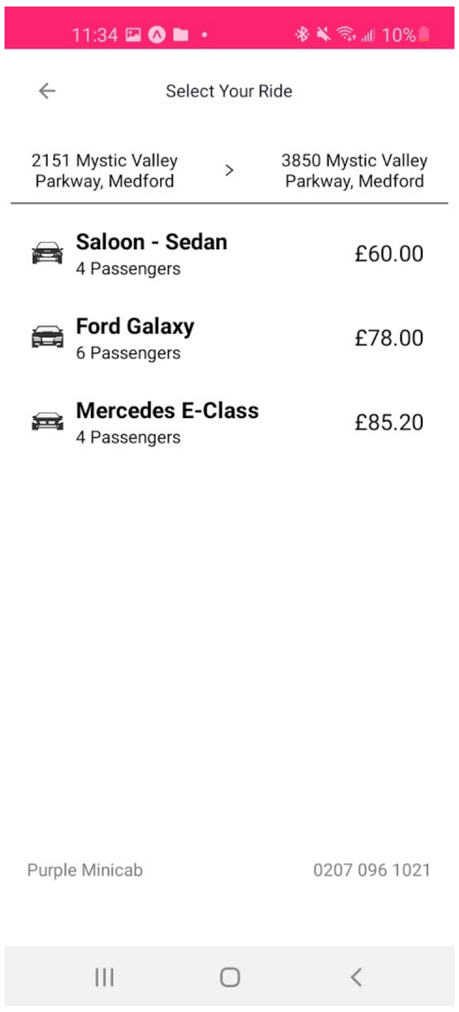
Select Your Ride

2151 Mystic Valley Parkway, Medford > 1899 Massachusetts Avenue, Cambridge

Please wait
12
seconds

Purple Minicab 0207 096 1021

Figure 6: Ride Screen (Waiting for Ride)


11:34 10%

Select Your Ride

2151 Mystic Valley Parkway, Medford > 3850 Mystic Valley Parkway, Medford

Saloon - Sedan 4 Passengers	£60.00
Ford Galaxy 6 Passengers	£78.00
Mercedes E-Class 4 Passengers	£85.20

Purple Minicab 0207 096 1021

Figure 7: Ride Screen (Waiting for Ride)

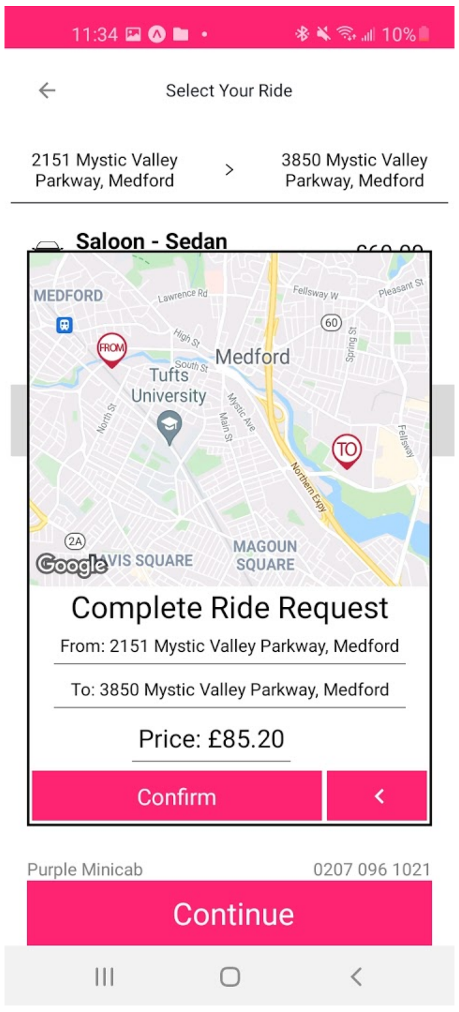


Figure 8: Ride Screen (Waiting for Ride)