Sapienza University of Rome

Human Computer Interaction — A.Y 2022/2023

Applied Computer Science and Artificial Intelligence

A Crowd-Sourced Mobility Application

Authors:

Doku Martina
Iannotti Giuseppina
Locatelli Dennis
Muià Elena Maria
Loi Dario
Marincione Davide

May 25, 2023

Contents

Chapter 1

Introduction

1.1 Our Idea

The intent of our application is to give more precise information about possible delays or irregularities in the public transportation service. This could be achieved by the retrieval of information delivered directly by the community of users, who may have the chance to inform others about possible delays or overcrowding of the bus or the tram that they are taking. Furthermore, all the users will collect some credits for every contribution, which will be devolved into charity, in one of the NGOs chosen by the user, among those proposed by the application.

1.2 Existing Competitors

Google Maps: It is one of Italy's most frequently used mobility applications. It is a web service that provides detailed information about geographical regions and sites worldwide. In addition to conventional road maps, Google Maps offers aerial and satellite views of many locations. It works both for public transportation and private ones.

- **Pros**: It is very intuitive and offers many fundamental features. It delivers information about possible paths to reach the destination with their duration and arrival time. It specifies the possible expenses needed for every choice taken.
- **Cons**: The path duration is not defined by crowdsourced information or GPS tracking applications. It is based totally on statistical inferences which means that is almost never precise.

Moovit: It is one of the main mobility applications for public transportation only. It provides information about the statistically best path to reach a destination point. Describing when and where to take the transportation means in order to reach the destination as fast as possible.

- **Pros**: More reliable than Google Maps (based on users' interviews), the interface is very intuitive.
- **Cons**: Too many ads, the defined timing is almost never correct.

Probus: The application is designed for Android only and it only works with buses. It informs the user about the waiting time of a certain bus line and the fastest path to reach a destination.

- **Pros**: Useful because it is strictly focused on busses, and therefore is able to offer a more tailored experience to users.
- **Cons**: Although the time estimates are reliable within a single bus trip, the application does *not* give information about future ones, hence, users do not know how long they will have to wait for the next run.

Citymapper: Citymapper is a public transit app and mapping service which displays transport options, usually with live timing, between any two locations in a supported city. It integrates data for all urban modes of transport, including walking, cycling, and driving, in addition to public transport.

- **Pros**: An almost always accurate and comprehensive direction guide. Free for both Android and iOS. It provides a calories counter and specifies the expenses for every chosen path.
- **Cons**: Not available in many cities and it does not retrive crowdsourced information.

Transit: Transit is a mobile app packed with features that help you plan a trip on the bus. Real-time bus tracking and information, service alerts, and trip planners are some of the many useful features that make this app a favorite for transportation services.

- **Pros**: GPS tracking of public transportation in real-time, crowdsource support (tracking the user location when they use the app as a navigator), information about all the surrounding bus stops and possible paths to the destination.
- **Cons**: Many useful services are not free. It does not work very well in Italy.

Chapter 2

Need Finding and Storyboarding

2.1 Need Finding

2.1.1 The Interviews

In order to better understand what our users want, we first conducted a round of interviews. These allowed us to interact colloquially with our potential users and to gauge what they think are the major discomforts of public transportation. We also wanted to understand their approach to personal privacy and community-driven applications. We used data we obtained as a guide for our next steps in the design process.

Our Questions Our interviews were standardized around a set of ten questions that we designed, as a group, to be as open-ended as possible. We wanted to avoid leading the interviewees to answer in a particular way, have them act as designers, or figure out the specific purpose of the survey until later on, when the general questions were answered.

The Questions:

1. Did you commute via public transport in the last week? If so, what type?
2. What criteria do you consider when choosing your means of transportation?
3. What are some frustrating aspects about public transportation?
4. Do you use mobility apps (like Google Maps) while commuting? If so, which functionalities?
5. How much do you trust the information given by your app of choice?
6. Do you worry about giving authorizations to apps? Are there some you are more willing to share?
7. Are you concerned about organizations distributing your location based data to third parties?
8. Would you trust mobility info more if it were crowd-sourced? Would you participate in such a program?
9. Would a honor system, rewarding you based on the credibility of your contributions, incentivize you to participate more?
10. In a community-driven app, how interested are you in customizing and showing your profile?

The Outcomes To carry out the interviews, we split our groups into 3 teams of 2 people each. Each of those teams had a target of 10 interviews to reach, which was achieved in a few days (with some extra interviews to spare).

Overall, if we look at the general trends, we can draw the following conclusions:

- A majority of our interviewees use public transportation on a daily basis, and most of them commute via bus or metro.
- All of the interviewees use mobility apps, with Google Maps being the most popular, usually paired with an app such as Moovit to provide more accurate information.
- Most of the interviewees find frustration in three things:
  1. Overcrowding of the vehicles
  2. Lack of punctuality
  3. Unreliability of bus rides (which are often late or do not show up at all)
- Users are generally willing to share data that is required for the functioning of the app, and are only really concerned about their privacy if the issue is brought up.
- Users enjoy the prospect of a community-driven application, and would be willing to participate in such a program, especially if it were to improve the quality of the service.
- Users are generally not interested in customizing their profile, *especially* in the context of a mobility app, this varies from person to person, and seems to correlate with the

person's technical knowledge, but the general trend shows that the feature is not very popular.

- In a similar fashion, users seem not to be interested by a gamification system, rather seeming to prefer more direct rewards, either monetary, in the form of a better service, or through the possibility of devolving their rewards to charity.

2.1.2 Questionnaire

After the interviews, we had obtained an initial set of knowledge on the needs and wants of our userbase. We used this knowledge to produce a new set of focused questions that aimed to capture the opinions of a wider group. Firstly, we sent a test sample to 10 of our friends and relatives and we used their answers to retrieve information on how to refine our questionnaire. Then, we distributed the questions through Google Forms to various online groups, obtaining a sample size of ≈ 150 answers.

Questionnaire URL: https://forms.gle/oiWbZBUvY4aiPjvm9

Questionnaire Structure The answers are difficult to provide as a transcript, since we made use of Google Forms features to provide a slightly different questionnaire to users based on their ongoing answers. The single greates use of this feature was to not collect informations of users who did not utilize public transport recently, in order to avoid pollution of our dataset. In general, we can give an outline of the structure of our questionnaire:

1. Introductory questions — Age group, frequency of use of public transport, device brand
2. General questions on transit — Principal frustrations, interest in a crowd source system, etc...
3. Opinions on reward systems — Whether they prefer customization, charity, honor systems, etc... as forms of reward to their contributions

Testing Before sharing our form with multiple people we first gave it to a number of acquaintances, as to test whether everything was alright. This let us spot two problems:

1. When asked *In which **city** do you use public transit?*, some answered with lesser towns such as Fisciano or Monterotondo — This prompted us to change the question into *In which **province** do you use public transit?*.
2. Some were baffled by the presence of further questions on the sharing of data after they had already answered **No** to the question *Would you like to share (anonymously), your position to help the app inform others?* — Therefore we made sure to shadow those questions in case of such an occurrence.

Unfortunately this testing phase didn't let us catch another small problem (albeit one nonetheless), when prompted to share their age bracket, people were shown the following:

- < 18
- 18-25
- 25-35
- ...

What we didn't realize is that there is no guidance on how the extremes of these ranges should be considered! Therefore, without further instructions, the ranges could be considered in different ways, some such that the ranges would be overlapping. Fortunately, this was not a real problem as this question wasn't really fundamental to our analysis.

Conclusions We performed some descriptive statistics on our dataset in order to draw more informed conclusions, this consisted mostly of the production of a series of graphs that aimed to quantitatively describe certain metrics (*e.g:* preference for a certain reward system) while categorizing our samples in different groups (*e.g:* by age, device, etc...).

First thing first, let's check the age distribution of those who answered our form:

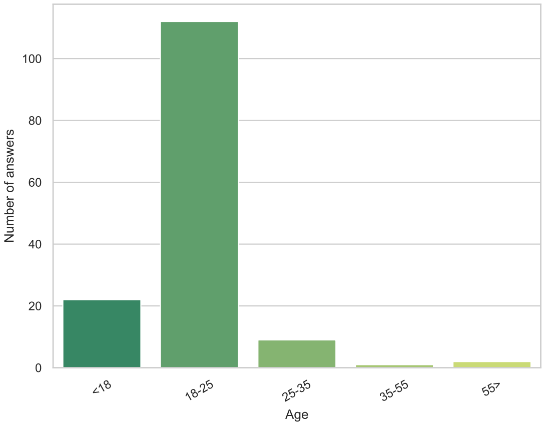


Figure 2.1: Age distribution of our respondents

The majority of our respondents are of ages 18-25, followed by a minority of users under 18 years. Those remaining are distributed across people aged 25-35 and > 55, with no respondent of age 35-55.

About the main province in which the respondents take public transport:

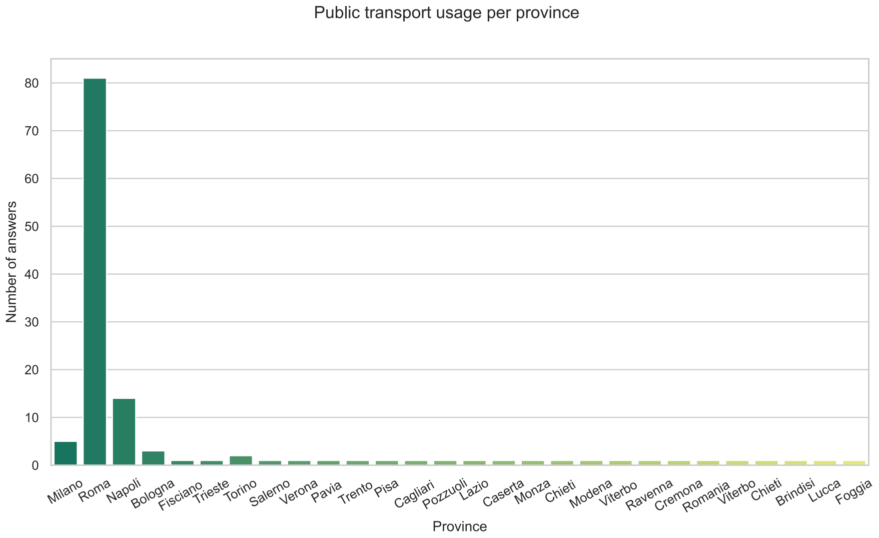


Figure 2.2: Public transport usage per province

As we can see, the majority uses public transport in Rome, with a minority in Milan and Naples. These results were obtained after having modified the questionnaire. As explained above, initially, we were asking for the city and not the province.

We are also interested in the most used operating system among the respondents:

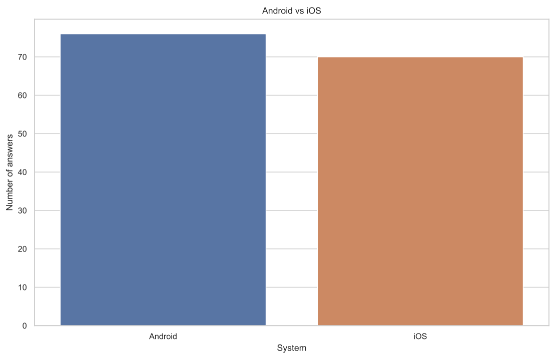


Figure 2.3: Android VS iOS

52% Android users and 48% iOS users. . . We were expecting an overwhelming incidence of Android users, yet it was *almost* a draw; nonetheless, as the majority of us carry Android devices, we decided to implement an Android application.

As follows, we show the number of public transport users by their device's operating system

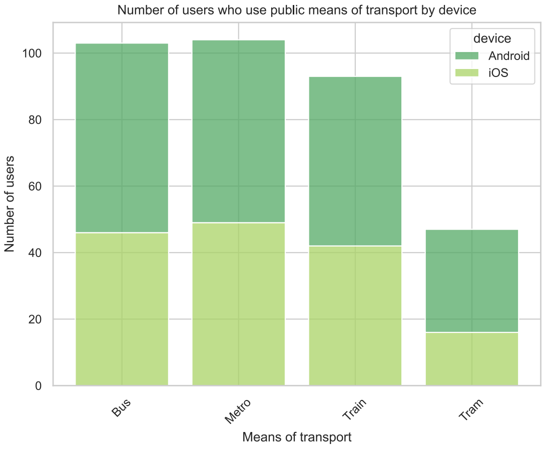


Figure 2.4: Users by means of transport by device

Let's look at the number of public transport users by province:

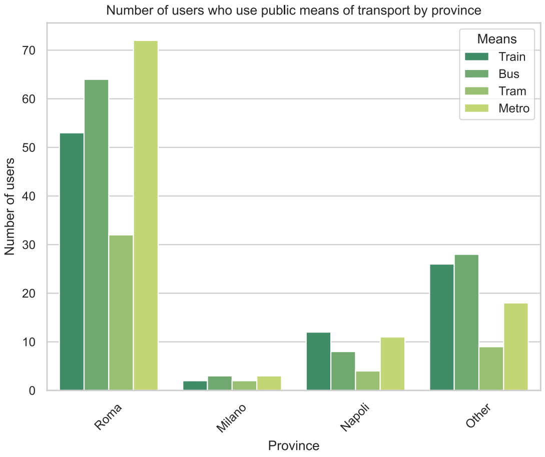


Figure 2.5: Users by means of transport by province

Since the majority of the respondents use public transport in Rome, Naples and Milan, we have collected the statistics of other provinces in the 'Other' column. It is evident that, in each province, users take the metro and the bus mostly.

At the basis of our reasoning, the most frequent disservices caused by public transport are:

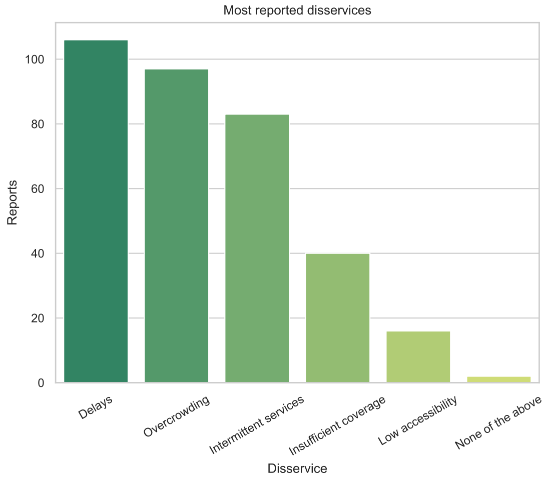


Figure 2.6: Disservices reports

This analysis is fundamental to our project. It highlights the most frustrating aspects of public transport. Those are the ones on which we should focus, in order to improve people's experience while using public transport. As shown, the majority of the respondents report about delays and overcrowding. The obtained results give us a further confirmation of our objective.

In the form, we ask some questions in order to understand users' preferences and whether they would be interested in our idea. Crowdsourcing is at the basis of our application. Retrieving information, about possible delays or transport overcrowding, directly from the community of users will make transportation more efficient. But, are users willing to partecipate? If so, would they be more incentivized having some kind of reward (e.g., charity, features and customization unlocking)? Let's look at some statistics. . .

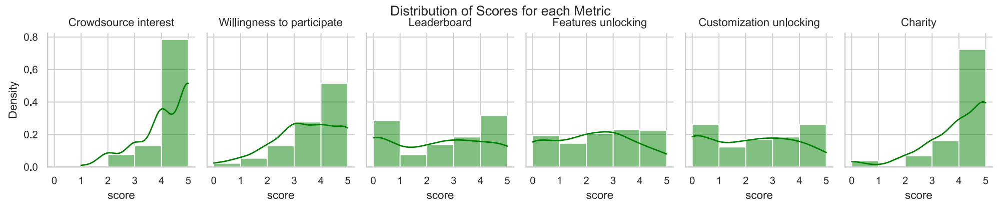


Figure 2.7: Overall distribution of respondents' interest

Without considering the age of the respondents we can see that the overall interest in a crowdsourced app and the willingness to participate in it is fairly high. For the functionalities, on the other hand, the opinions are mixed if not negative to everything except the idea of charity.

Let's look at these same statistics broken down by age ranges (there were not enough respondents in the 35-55 gap, thus a plot was unfeasible).

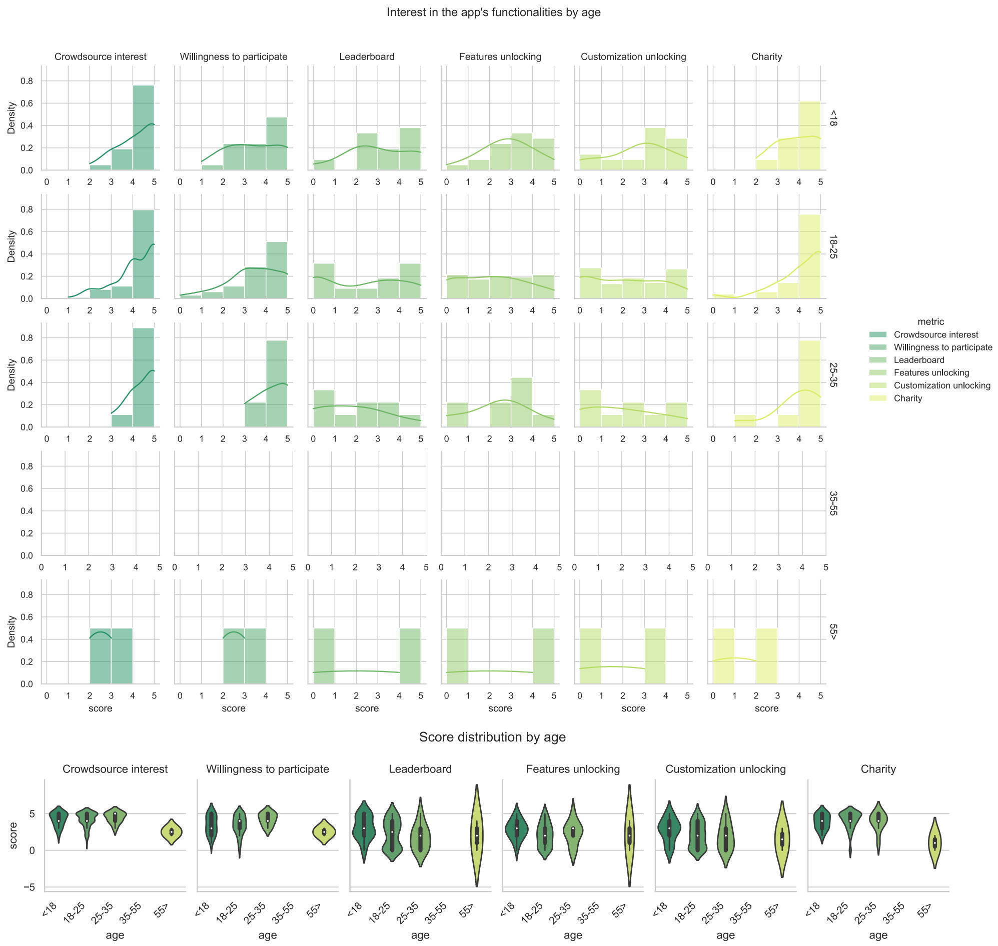


Figure 2.8: Respondents' interest by age

From this breakdown we can see that both the willingness to participate and the interest in the offered functionalities is slightly higher in the younger population, strengthening our interest in that target. Indeed, here's a focus on the 18-25 gap:

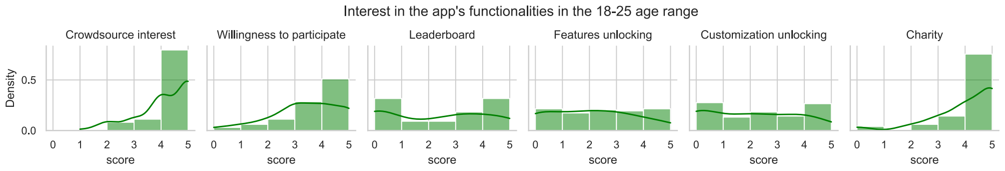


Figure 2.9: Interest in app functionalities 18-25

Now, to draw some conclusions, we arbitrarily set a threshold for *interested/not interested* respondents to each metric at a score of 3.

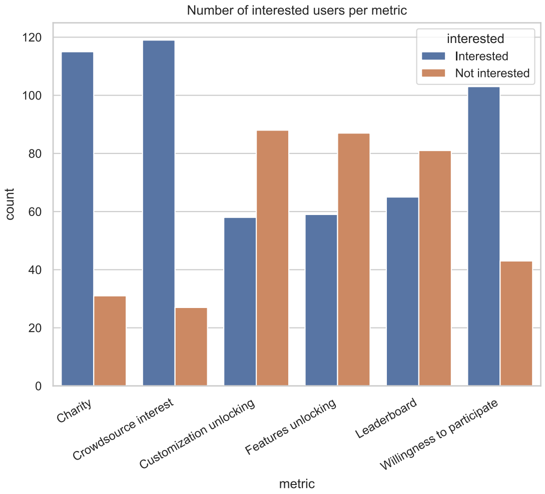


Figure 2.10: Interested users per metric

As we can see, the majority is not interested in a leaderboard of the users who have reported the greatest number of information (e.g., delays, overcrowding). In the same fashion, reward systems such as features and costumization unlocking have not captured the attention of the respondents. However, their crowdsource interest and willingness to partecipate in such a community give us a positive feedback. Charity donations incentivize them more.

Lastly, but not in terms of importance, let's check how the users would like to report the vehicles they are currently on.

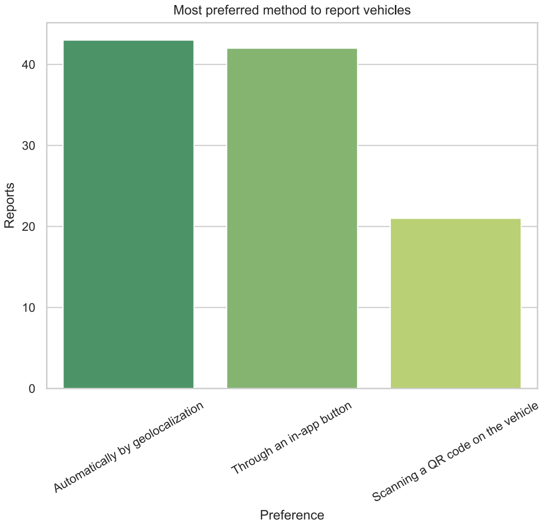


Figure 2.11: Most preferred method to report vehicles

From this, we deduce (since the GPS method is probably the most intrusive one), that the best course of action would be to create a system without geolocalization but that would also be as simple and annoyance-free as possible.

2.2 Storyboarding

Tasks After our questionnaires, we wanted to clearly identify a set of *core* tasks that should be supported by the application.

The Tasks:

- Looking for a route to a destination
- Reporting public transport delays to other users
- Reporting overcrowding in a public transport to other users
- Spending points for a donation to a non-profit organization

For each of those we then proceeded to produce a storyboard detailing the sequence of interactions that a user must perform in order to complete the task.

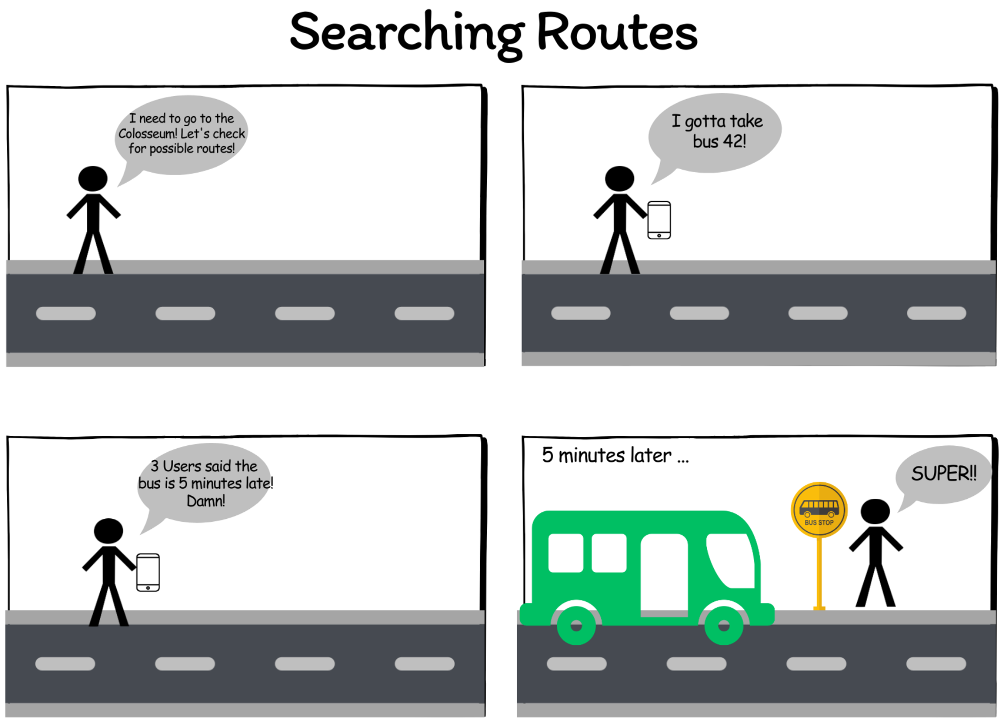

Searching Routes
I need to go to the Colosseum! Let's check for possible routes!
I gotta take bus 42!
3 Users said the bus is 5 minutes late! Damn!
5 minutes later ...
BUS STOP
SUPER!!

Figure 2.12: The user can find all the possible paths leading to the defined destination point, moreover, he/she can be informed about the statistically calculated waiting time as well as any reported delays regarding the proposed bus lines.

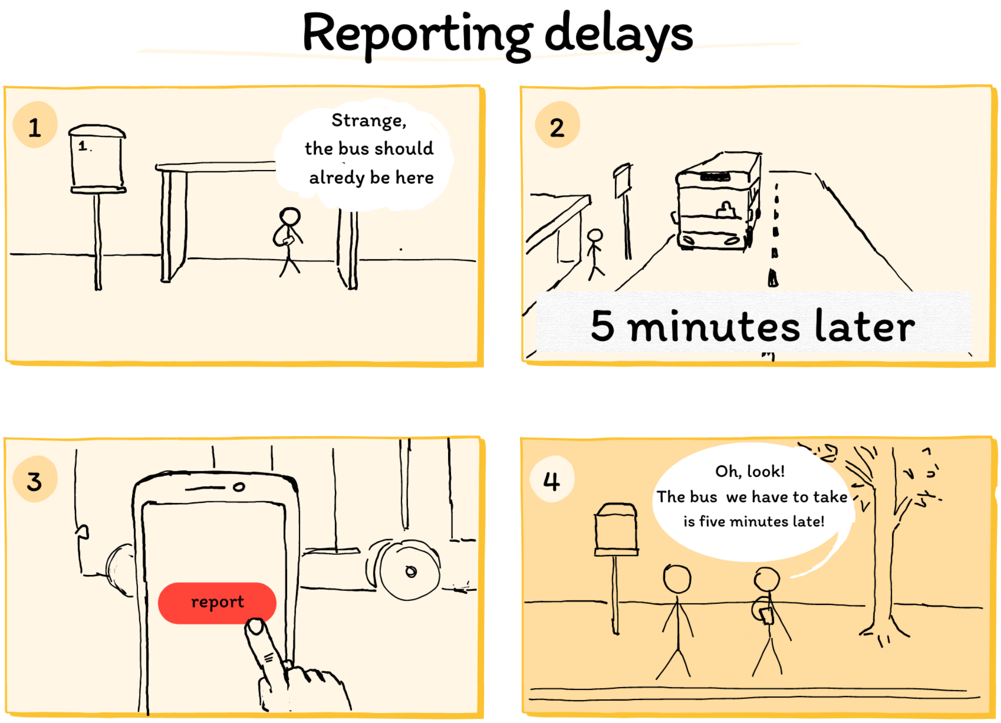

Reporting delays
1
Strange, the bus should alredy be here
2
5 minutes later
3
report
4
Oh, look! The bus we have to take is five minutes late!

Figure 2.13: The user can visualize the real time statistics on the expected waiting times for a certain bus line. If the statistics are not correct he/she has the chance to inform others about the actual arrival time of the bus, leading to a recalculation of the waiting times.

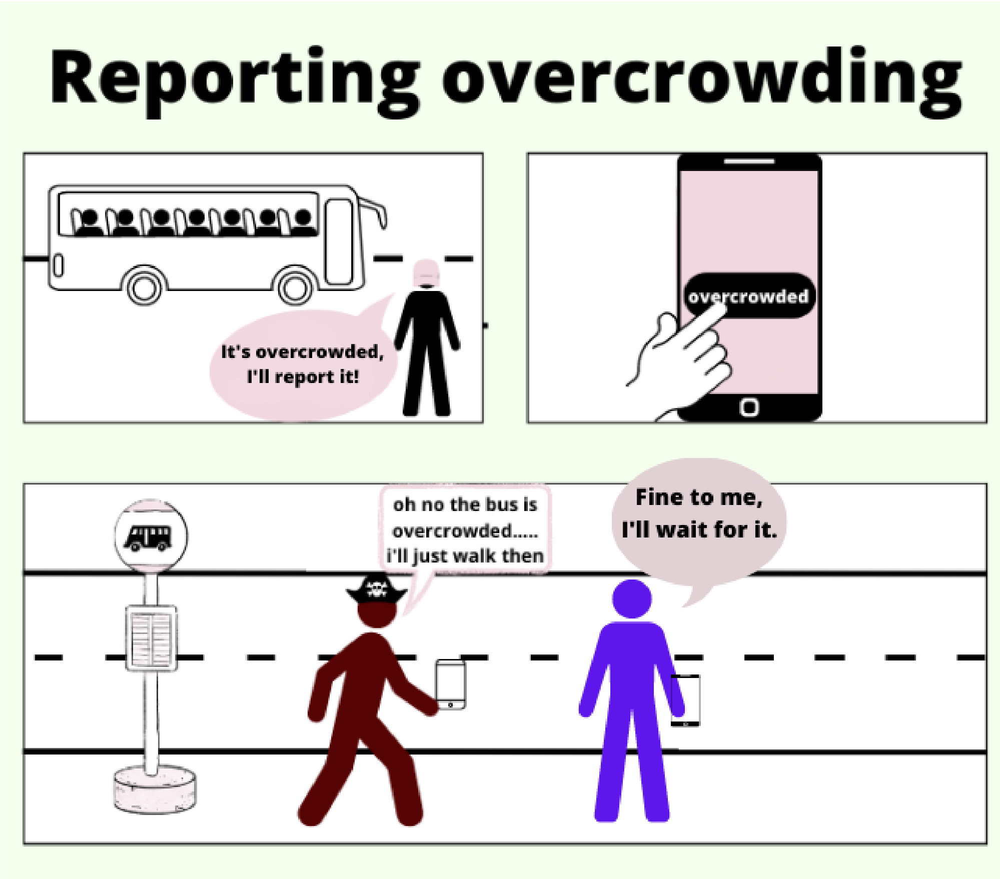
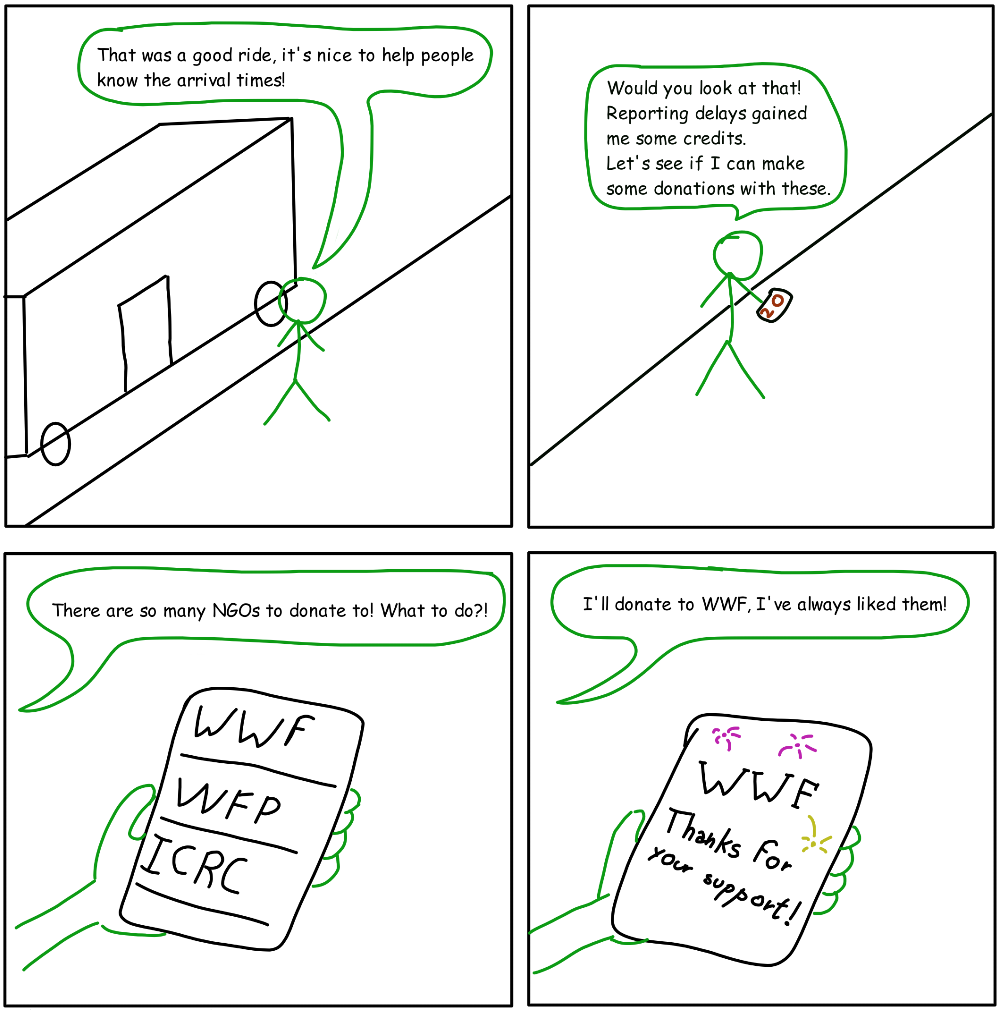

Reporting overcrowding
It's overcrowded, I'll report it!
overcrowded
oh no the bus is overcrowded..... i'll just walk then
Fine to me, I'll wait for it.

Figure 2.14: The user can also report if the bus that is passing is overcrowded, thus, even if the bus arrives in time, he can give useful information to other users, who may benefit from this information and choose a different route to their destination.


That was a good ride, it's nice to help people know the arrival times!
Would you look at that! Reporting delays gained me some credits. Let's see if I can make some donations with these.
There are so many NGOs to donate to! What to do?!
WWF
WFP
ICRC
I'll donate to WWF, I've always liked them!
WWF
Thanks for your support!

Storyboard by: Giuse & Davide

Drawings by: Davide

Figure 2.15: In the end, the user gains a certain amount of credits for the reported departures. After n reports, he/she can devolve their credits as charity to one of the NGOs proposed by the application. Thus, every free contribution to the service may be of further help to society.

Chapter 3

Prototyping

Tools In order to be able to test our ideas in a flexible context, we decided to prototype our application through the popular prototyping tool *Figma*. This allowed us to quickly iterate on our ideas, as well as to test them with users in a more realistic context than a simple mockup.

Figma URL: https://www.figma.com/file/79iXdbhOZP1pKwH8kEN14G/HCI?type=design&node-id=0%3A1&t=NvYIj1oicjiQwJ4P-1

Evaluation Techniques For each major iteration of our prototype, we tested the application with groups of users of various sizes, using two different evaluation techniques:

1. **Think Aloud**: The user is asked to perform a series of tasks, while explaining their thought process, interactions with the application and expected outcomes out loud.
2. **Cooperative**: Similar to think aloud, but there is a constant feedback loop between the user and the tester.

In both cases, we recorded the audio of the evaluation session, in order to be able to better analyze the results at a later time.

Table 3.1: Test Rounds Sample Sizes

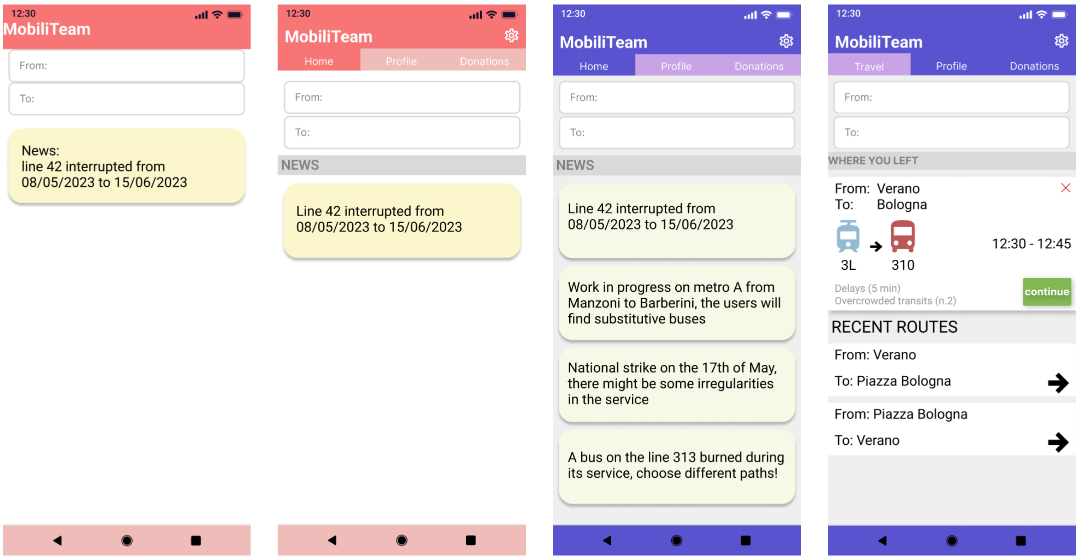
Iteration	Number of Tests
First	4
Second	7
Third	13

The number of tests increases as we iterated on our design, however, in the beginning, the duration of the tests was longer, due to the suboptimal User Experience of the application, which led to more time being spent on each task.

3.1 Prototypes

In the following section, we are going to show a number of areas of our app and their evolution through the various iterations of our design process.

3.1.1 The Home Screen



News:
line 42 interrupted from 08/05/2023 to 15/06/2023

Line 42 interrupted from 08/05/2023 to 15/06/2023

Line 42 interrupted from 08/05/2023 to 15/06/2023

Work in progress on metro A from Manzoni to Barberini, the users will find substitutive buses

National strike on the 17th of May, there might be some irregularities in the service

A bus on the line 313 burned during its service, choose different paths!

From: Verano
To: Bologna
12:30 - 12:45
Delays (5 min)
Overcrowded transits (n.2)

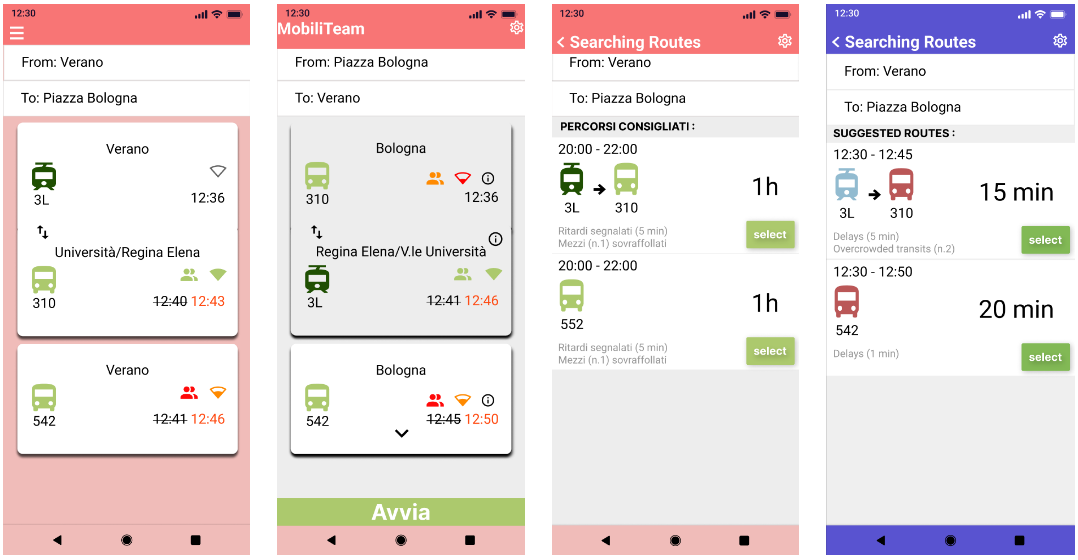
From: Verano
To: Piazza Bologna

From: Piazza Bologna
To: Verano

(a) First Iteration, Early (b) First Iteration, Late (c) Second Iteration (d) Third Iteration

Figure 3.1: The *home* is the central screen of the application, in fig. 3.1a, we can see the very first iteration of the home screen, which lacked the top navigation bar, and a separation of the news section, the iteration that was actually deployed in tests is fig. 3.1b, here, it emerged from tests that the lack of news was confusing. This was fixed in fig. 3.1c, where we populated the news section with more articles, from now on, a change in color palette can also be noticed, surprisingly, people found having more news even *more* confusing, as they felt it made the app lose its focus on its main objective, thus, in the third iteration (fig. 3.1d), it was replaced with a more utilitarian feature, a selection of recent routes and the possibility to resume a route that was previously interrupted.

3.1.2 Searching for Routes



12:30

From: Verano

To: Piazza Bologna

Verano

3L 12:36

Università/Regina Elena

310 ~~12:40~~ 12:43

Verano

542 ~~12:41~~ 12:46

12:30

MobiliTeam

From: Piazza Bologna

To: Verano

Bologna

310 12:36

Regina Elena/V.le Università

3L ~~12:41~~ 12:46

Bologna

542 ~~12:45~~ 12:50

Avvia

12:30

Searching Routes

From: Verano

To: Piazza Bologna

PERCORSI CONSIGLIATI :

20:00 - 22:00

3L 310 1h

Ritardi segnalati (5 min)
Mezzi (n.1) sovraffollati

select

20:00 - 22:00

552 1h

Ritardi segnalati (5 min)
Mezzi (n.1) sovraffollati

select

12:30

Searching Routes

From: Verano

To: Piazza Bologna

SUGGESTED ROUTES :

12:30 - 12:45

3L 310 15 min

Delays (5 min)
Overcrowded transits (n.2)

select

12:30 - 12:50

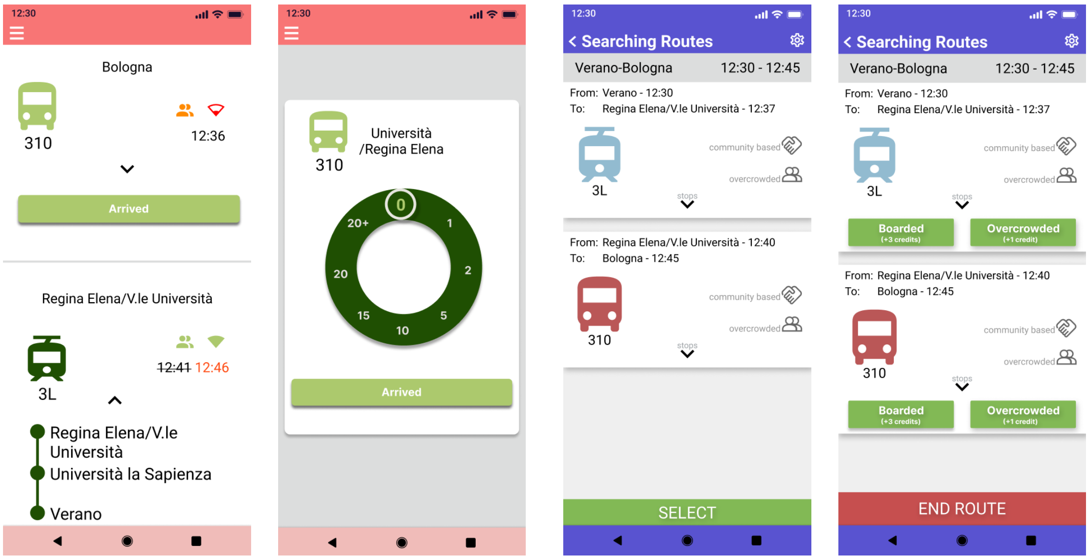
542 20 min

Delays (1 min)

select

(a) First Iteration (b) Second Iteration, Early (c) Second Iteration, Late (d) Third Iteration

Figure 3.2: The *navigation* screen was the one that presented the most criticalities during the earliest stages of testing, in fig. 3.2a, we can see the first iteration of the screen, which was very confusing, owing to the fact that the final stop is not present in any of the routes, the individual routes' cards are not clearly separated, the double arrow logo makes it look as if tapping it would switch the direction of the route, when it is in fact purely decorative, and the eccessive use of non-standard icons makes the screen look cluttered. In fig. 3.2b, the problem was exacerbated by the addition of an info button, which failed in his purpose by adding to the clutter. The design was simplified in fig. 3.2c, where the info button was removed, the routes were clearly separated, and many icons were replaced with text, this version was very successful in tests, and was only slightly modified in the third iteration (fig. 3.2d), which is mainly a color palette change, along with some slightly more realistic timestamps (and english text).



(a) Route Selection, Early (b) Roundel (c) Route Selection, Final (d) Reporting Screen

Figure 3.3: Another critical part of the project was the way that reporting delays and overcrowding was handled, initially, once a route was chosen (fig. 3.3a), the user would have to click the ambiguously-sounding button *arrived*, which then brought up a roundel (fig. 3.3b) that users could tap onto to report the time discrepancy between the report and the bus's arrival. This was *the* single most confusing aspect of the User Experience in our earlier iterations, and was subsequently replaced with a saner system, which, based on the assumption that this discrepancy is statistically insignificant, reduced the process to simply select the desired route (fig. 3.3c) and then tap on a button to report boarding/overcrowdedness (fig. 3.3d)

Chapter 4

Post-Evaluation and Development

4.1 Final Changes

After our last discussion with the professor, we proceeded to swiftly implement his suggestions, namely:

- **Reworking the reporting system**: Instead of having two separate buttons to report boarding and overcrowding, the actions have now been *merged*, the user is presented with two buttons, 'overcrowded' and 'not overcrowded'. This change was made to simplify the reporting process, as well as to make it more intuitive: this way, the user can report the status of the current vehicle with a single tap, while also giving us information on its position.
- **Various text fixes**: Some of the text in the application was incorrect, with some lines appearing twice and others having inconsistent timings.
- **Component consistency**: Buttons and cards were made more consistent with the material design philosophy all across the application.
- **Community based information**: The application now shows when some information is *not* community based, and instead comes from statistical inferences.

After these changes, the prototyping stage was considered complete, and we moved on to finalizing the application.

4.2 The Backend

4.2.1 The Simulation

In order to avoid having to deal with the complexities of a real-world application, such as integrating various Google APIs (Maps, Authentication, etc...), we decided to build a *simulation* of a transit system, which would allow us to test our application with "live" data in a shorter timespan.

In reality, there are two simulations running at the same time inside the backend- one is used to produce the "live" data and is omniscient to the status of the transits, governing each and every of their movements through simple routes (in which traffic is "simulated", such as to create possible delays, through some simplex noise functions) and also simulating the existence of passengers (of which some are users) which through multiple probability distributions can signal to our backend how crowded the transits they are on are.

The second simulation is one which would be needed even if we were to use the Google APIs- it *doesn't* know the transits' true current position, assumes a constant traffic, and is oblivious to how many people are onboard the transits. To approximate these informations it uses the signals produced by the "users" in the first simulation.

4.2.2 The Server

The server is a simple Flask application, Flask was chosen as a backend for its simplicity and ease of use, in line with the philosophy of the project. It uses both the second (non-omniscient) simulator, to keep track of the transits, and a SQLite database, to maintain information about the users (real users, not the simulated ones). A description of all the requests that can be done to the server can be found in form of an OpenAPI `.yaml` file.

4.3 The application

Built via Android Studio, our Kotlin application was programmed to be as close to the Figma as possible while trying to improve the user experience with animations and better responsiveness. The task of producing this app truly was a feat, as none of us had ever built a mobile application. . . because of this it is more than possible that we've done gross mistakes during its development, nonetheless, we are satisfied with the results that we reached, as we managed to make the application fully functional and to look as nice as we possibly could (arguably not a lot. . . we are computer scientists after all).

4.4 Conclusion

Possible Improvements The application, if developed further, could be improved by strengthening the backend by improving the ways in which overcrowding and delays are predicted based on the data collected from the users. The possibility to report other types of disservices could also be added, as long as it does not encumber the User Experience. Lastly, the application could be made available to a wider audience, by porting it to iOS through a cross-platform language such as Flutter.

Final Thoughts We believe our demo to be a solid showcase of the potential of the application, which we believe has the opportunity to change the way that people interact with public transportation, and to improve the quality of the service as a whole.