Table of Content

Task description and group members

Task Description

TASKS TWO AND THREE

Requirements and Requirements Analysis of A Passenger Positioning System

Title

Design and implementation of a Passenger Positioning System(Municipal Commuting Application)

Group Number: 6

Group Members

Name	Matriculation Number
Yimnai Nerus Zaumu	FE20A123
Tabot Charles Bessong	FE0A106
Bebongnchu Yannick Nkwetta	FE20A022
Balemba Junior Balemba	FE20A021
Tandongfor Shalom Changeh	FE20A111

Tasks content

Introduction

The present state of transportation within the community of Buea is in shambles. Though it often gets the job done, it is nonetheless with a lot of difficulty. I remember on April 12 having to stand for about thirty minutes, stopping without success more than 40 taxis with no one going in my direction. What makes it worse is that this had nothing to do with change or the lack of it thereof. When you factor this in, you cannot help but imagine how much worse it probably would have been. This problem is not unique to me. Most people will attest to facing these same difficulties albeit to varying degrees.

This is where the **Passenger Positioning System** comes into play.

Purpose

The main goal of this project is to solve this problem by greatly simplifying the process of getting a taxi for both the passengers and drivers. Passengers should be able to know which taxis are going in what direction and likewise the drivers should be able to know the same about the passengers.

Furthermore, passengers should be able to book taxis ahead of time. This will greatly improve the wait times of the taxis at each stop. This is going to cause a domino effect as it will in turn reduce traffic in some hotspots around the locality which will in turn just lead to better community life.

Project Requirements

Requirements outline

We identified six different kinds of requirements for this project which are

- Functional requirements
- Non Functional requirements
- Business requirements
- Regulatory requirements
- User requirements and
- Technical requirements

1) Functional Requirements

Functional requirements define the features the application must possess and functions it must perform. In short, they define the application in terms of its inputs, outputs and behaviours. Below are the functional requirements we have come up with for this application.

- Users(passengers) and Drivers should have the ability to Create accounts and log in to those accounts.
- Passengers should have the ability to view different drivers and their various locations
- Passengers should have the ability to book/cancel rides from the application
- Passengers should be able to pay for rides via Mobile money

- Passengers should have the ability to view their ride history from within the application
- Passengers should have the ability to track the drivers and the estimated time of arrival at their locations
- Drivers should have the ability to view active users and their locations
- Drivers should be able to view routes and their degrees of traffic
- Drivers should be able to know passenger concentration in different locations
- Drivers should be able to view statistics about their rides
- Drivers should be able to accept payments into their accounts
- Drivers should be able to withdraw payments into their local accounts
- Passengers and Drivers should be able to set their state to active or inactive in order not to confuse each other

2) Non Functional Requirements

Unlike functional requirements, non functional requirements describe how the system should perform instead on what it does. These are the key non functional requirements that we pinpointed which we believe are vital for this application's success.

- **Performance**

  The system should be able to manage with relative ease a huge number of interactions between drivers and passengers and concurrent transactions. This is

representative of how often passengers interact with drivers physically

- **Usability**

  The application should be easy and intuitive to use to both the passengers and the drivers. It should have clear and concise navigation with explicit error messages in cases where things don't go as planned. Users shouldn't have to imagine why things are behaving the way that they are.

- **Reliability**

  The system should be available and responsive at all times with minimal downtime and errors. When drivers are booked by passengers, they should be confident that those are real people they are going to carry and not just some bots spamming the system.

- **Security**

  User data should be well secured. Appropriate measures should be put in place to ensure this such as data encryption, authorization and authentication.

- **Scalability**

  The system should be able handle an increasing number of users(drivers and passengers alike) without significant change or degradation in its performance.

- **Compatibility**

  The application should be readily available on both android and IOS devices with similar functionality.

3) User Requirements

These requirements have to do with the expectations of the users of the system. In this case both the drivers and passengers.

- Accurate and real-time positioning information
- Easy to use interface for passengers
- Privacy protection for user data
- Compatibility with different devices and operating systems
- Reliable performance in different situations

4) Regulatory Requirements

- Compliance with data protection and privacy laws
- Compliance with transportation safety regulations
- Accurate and reliable positioning information for emergency response and rescue operations
- Integration with emergency response systems and protocols
- Compliance with environmental regulations and standards

5) Business Requirements

From a business perspective, the goals of this project is;

- Cost effective implementation and maintenance
- Scalability to accommodate a large number of passengers and vehicles

- Integration with existing transportation systems and infrastructure
- Ability to generate reports and analytics
- Compliance with industry standards and regulations

6) Technical Requirements

Technical requirements here refer to the hardware, software and infrastructure requirements of the project. In an ideal situation, the system will contain both a mobile application for passengers and drivers and a web dashboard for administrative control and even for account management by drivers. But in this case, we'll be focusing more on the interface for the driver and passenger. As a result, we'll be building just the mobile application that meets the needs of both the drivers and passengers. Of course, the system will be built in such a way that it can be expanded to allow many other interfaces to be integrated into it.
Also, another very important part of the technical requirements has to do with the tools and the reasons why we chose them.

Tools

1) **UI Design:** Figma

   Reason: Besides the fact that it is really good and gets the job done, it is very easy to use, free and all team members have some minimum amount of experience with it.

2) **Mobile Application UI:** React Native(RN)

Reason: RN has a low barrier of entry so it is quite easy to pick up by someone with some Javascript or React knowledge no matter how basic. Also, it is cross platform and hence, we won't have to build a separate application for IOS devices. Furthermore, it has a very large community and ecosystem so it is very easy to seek and find help when you need it. Last but not the least, 2 of our team members have quite a bit of experience with it and so, it makes sense for us to go with something that some people already know instead of having to learn something different from scratch given the tight deadlines we have to work with.

3) **Styling UI:** SaSS and NativeWind

Reason: SaSS is a flavour of CSS while NativeWind is in simple terms, custom tailwind css but for RN. We chose these two so that we can focus less on how our system looks and more on what it actually accomplishes. It helps us to iterate much faster with the look and feel of the application.

4) **Database:** MySQL/Redis Cache

Reason: The entire team has been working with MSQL right from the start of our program at FET so we are fairly good with it. Also, MYSQl is vertically scalable and structured so suits our needs perfectly in this case. Redis cache could be learnt in an afternoon and because we'll be

using just for data caching, it will be quite easy for the team to follow. Most importantly, it is very good at getting things done. Another important reason for choosing MYSQL is that we can use an amazing MYSQL provider such as planetscale which provides us up to 1GB of free space which is more than sufficient for our use case. So, we won't have to build a local database before looking for means to port it to the cloud later.

5) **Backend Language/Framework:** Typescript/NestJs

Reason: NestJs is a NodeJs framework for building performant and scalable backend applications(APIS). The fact that it uses typescript also means it provides a high level of type safety for us thus catching a lot of errors at compile time instead of runtime. This makes it very suitable for our use case. More important is the fact that part of the team has some experience working with it so it won't require the entire team to learn an entire framework hence making us move faster.

6) **Deployment**

Backend: Azure Container Apps: We intend to host the final backend as an Azure Container application. This means we'll have to build the app as a docker container in the end before deployment. The reason for this choice is that one of our team members already has a functioning account on Azure and so it makes sense to use it instead of looking for something brand new. Also, it just works and is not too difficult to pick up by other team members.

Mobile App: Unfortunately, we will only be able to deploy on Android at this point via APKs as the google play store

will require a $25 fee for deployment and even worse than that, the Apple App store requires a $99 charge and doesn't even support APKs. Depending on the direction the other team members chose to go, we might consider these other options even after this course.

Documentation: This application will be documented with the help of Swagger documentation with the direction of the OpenApi specification. This is going to be very vital for the developers of the UI as they'll need guidance when consuming the API services that power the application.

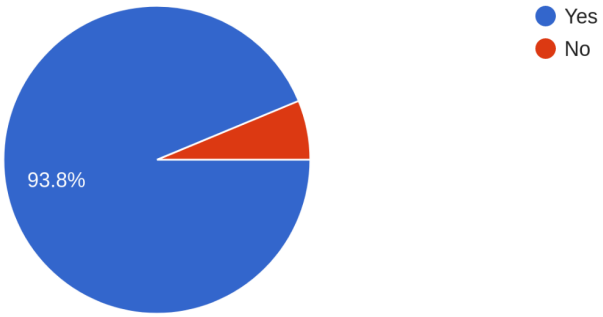
Analysis of Requirements

In order to make understanding of the above requirements and their specifications and in order to better analyse the system, we shared a survey with frequent passengers to get their thoughts and feelings on the entire process. We got quite a number of responses(X number to be exact). We asked passengers questions like how often they use the public transit system, their experience and what their feelings and expectations are from a digital system, in this case a mobile application that at least claims to solve the problems of the current system. The charts below represent some of the feedback we got from all the respondents to the survey we sent out.

Survey charts

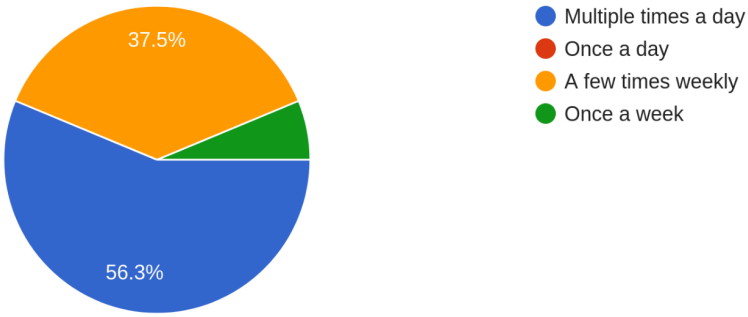
Are you a frequent user of public transportation in your locality?

16 responses



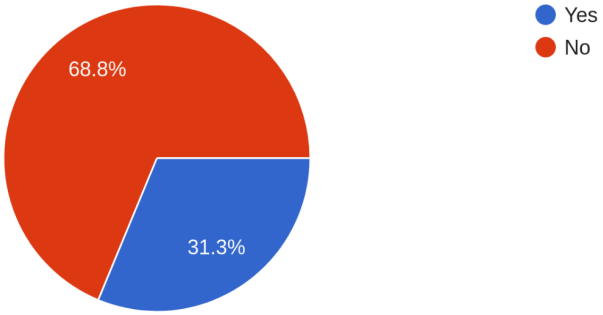
How often do you use public transportation?

16 responses



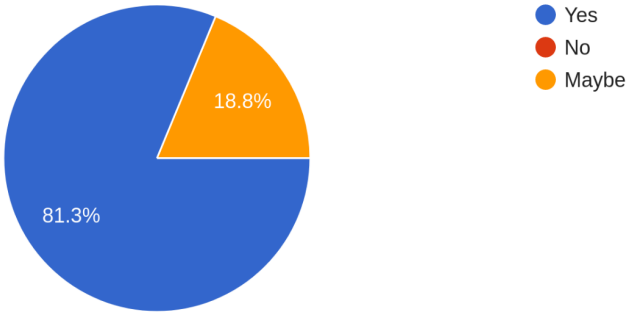
Do you find the current process efficient?

16 responses



Would you be interested in using a digital system that enables you easily find drivers and vice versa?

16 responses



How important is it to you that the taxi service is reliable and punctual?

16 responses

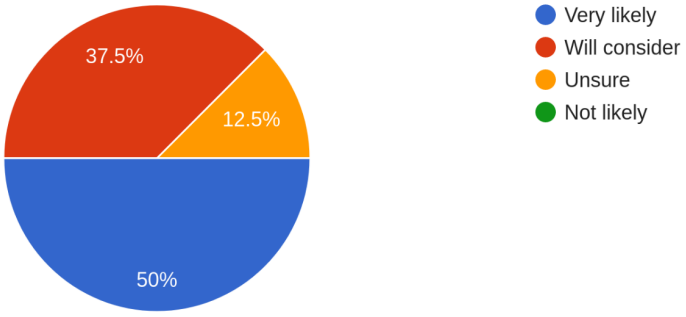
What are some of the problems you experience with the current format of transportation within your locality?

14 responses

- Too much congestion
- Punctuality and cash issues
- Difficult to find empty taxis at times
- Inefficiency of system and lack of trust
- Unavailability of car proportionally in respect of the locality and time(late in the night)
- Change. Time to walk to the road and get a taxi
- Rude drivers
- We are often too tight in a taxi or bus

If there was an application that served the purpose, how likely are you to install and use it if it made it easier to use the transportation system?

16 responses



The following link is to a spreadsheet that gives more detail on user feedback.
You can check it out at;
https://docs.google.com/spreadsheets/d/1jlLOzc7jj5gS2ITPK93_7pTybV6RUoeipCWpd4KlJoI/edit?resourcekey&usp=forms_web_b#gid=500060370

The survey doesn't contain sufficient data to draw a definite conclusion, it nonetheless drives us in the right direction. It is clear users want a better solution and at this point, all roads lead to a mobile application and that's where our solution comes in.
Furthermore, user feedback has led us to prioritise our non-functional requirements better. It is impossible and impractical to satisfy all non functional requirements because there will always be trade-offs. In this case, we will prioritise them in the following order:

- Usability
- Reliability
- Performance
- Scalability
- Security

Also, in order to ease understanding of our requirements, we came up with the following diagrams that better explain the project and what we intend to accomplish with it. In order to achieve this, we have to first come up with the stakeholders of the system and the system components mentioned in the requirements.

Stakeholders of system:

- Passengers
- Drivers
- Developers

System Components:

- **Mobile application:** It is the main means of interaction of the application. It is the meeting point of all transactions that take

place on the platform. Passengers are able to interact with drivers directly and vice versa.

- **Payment Gateway:** It enables the drivers to be able to subscribe to the system to be able to have any sort of access. It also lets the drivers withdraw their money from their account platforms. Also, passengers who want to book directly from the system or pay drivers through the system will need to make use of it as well.
- **Database as a service provider:** In this case, we will make use of Planetscale. Their free tier is quite generous with up to 1GB of free MYSQL storage. We wouldn't have to worry for one bit about hosting our database locally
- **Hosting service:** Azure Container Registry will serve all of our purposes regarding deploying our backend. Also, the $200 they offer for their free tier is more than sufficient for our needs.
- **Testing:** Jest testing library will be used extensively throughout our development process in order to write unit tests for our application.

Conclusion

Because of the results we got from the survey, together with all of the work that went into research by all the team members into this requirements and and requirements analysis, we can confidently say that we are on the right track and are on the road to develop an application that solves the problem it was initially conceived to solve.