

[**1. Introduction:**](#_oodyo44y26c) **1**

[1.1. Problem Statement](#_igl4gbv0e9og) 2

[**3. User Interface:**](#_ov3hxeo5gaqo) **3**

[**4. Dataflow diagram:**](#_q3bknylzyixq) **4**

[**5. Technical stack:**](#_hexb4w2y6fqe) **5**

[**6. Architecture:**](#_pi0vmocc3a8d) **6**

[**7. Testing and Quality:**](#_ey8zzqvpiz2h) **7**

[**8. Method of validation or approval of data:**](#_wvqz6lf19yvq) **8**

[**9. Project Timeline:**](#_uffg2bz2imrf) **9**

[**10. Conclusion:**](#_35wro7pp1tln) **10**

# Introduction:

We are basically building an app for the community that will help drivers locate areas of passengers with higher concentration and for customers to board or locate taxis quickly. Here passengers seeking taxis can easily be found via the map seen by drivers which can now quickly pinpoint the shortest location to their desired clients.

After detailed and careful analysis we were able to come out with the following problem descriptions.

### **1.1 Problem Statement**

1. Passengers do not know any information about the current location of the taxi which can result in long waits and thus affect the readiness of passengers to board a taxi.
2. Passengers do not know at what time a taxi might show up at their current location.
3. There’s overcrowding at various points in town in the early hours of the day due to the fact that passengers' arrival time at the expected place is not recognized that is all passengers be it short, long or medium distance arrive the road at the same time,
4. The distress of passengers about the delay of a taxi since there is no information about the delay of the taxi may take their time since the hope of waiting might be fruitless.
5. With the recent increase in the price of fuel, drivers will really want to optimize the consumption of this expensive fuel.
6. App features:

This section describes all the app features of the system, and how they are used and described.

Real-time tracking: The app should allow passengers and transportation

providers to track the location of the vehicle in real-time, using GPS

technology.

1. **Booking and reservations**: The app should allow passengers to book and reserve transportation services in advance, with the ability to specify pickup and drop-off locations and times.
2. **Driver information:** The app should provide passengers with information about the driver, such as their name, photo, and contact details, and allow them to contact the driver directly.
3. **Payment processing**: The app should allow passengers to make payments for transportation services, with support for various payment methods such as credit cards, PayPal, or mobile wallets.
4. **Notifications and alerts:** The app should send notifications and alerts to passengers and transportation providers, such as pickup and drop-off reminders, arrival notifications, or driver updates.
5. **Reviews and ratings:** The app should allow passengers to leave reviews and ratings for the driver and the transportation service, with feedback on the driver's performance and the overall quality of the service.
6. **Customer support:** The app should provide customer support for passengers and transportation providers, with support for various channels such as phone, email, or chat.
7. **In-app messaging**: The app should provide an in-app messaging system for passengers and drivers to communicate directly with each other, with support for text, voice, and video messaging.
8. **Localization and multi-language support**: The app should support localization and multi-language support for passengers and transportation providers in different countries and regions.
9. **Integration with other apps and services:** The app should integrate with other transportation providers apps and services, such as payment gateways, mapping services, or weather APIs.

# User Interface:

Designing a user interface for a passenger positioning system requires

Consideration of the needs and preferences of the users, as well as the

Features and functionality of the system. Here are some key elements

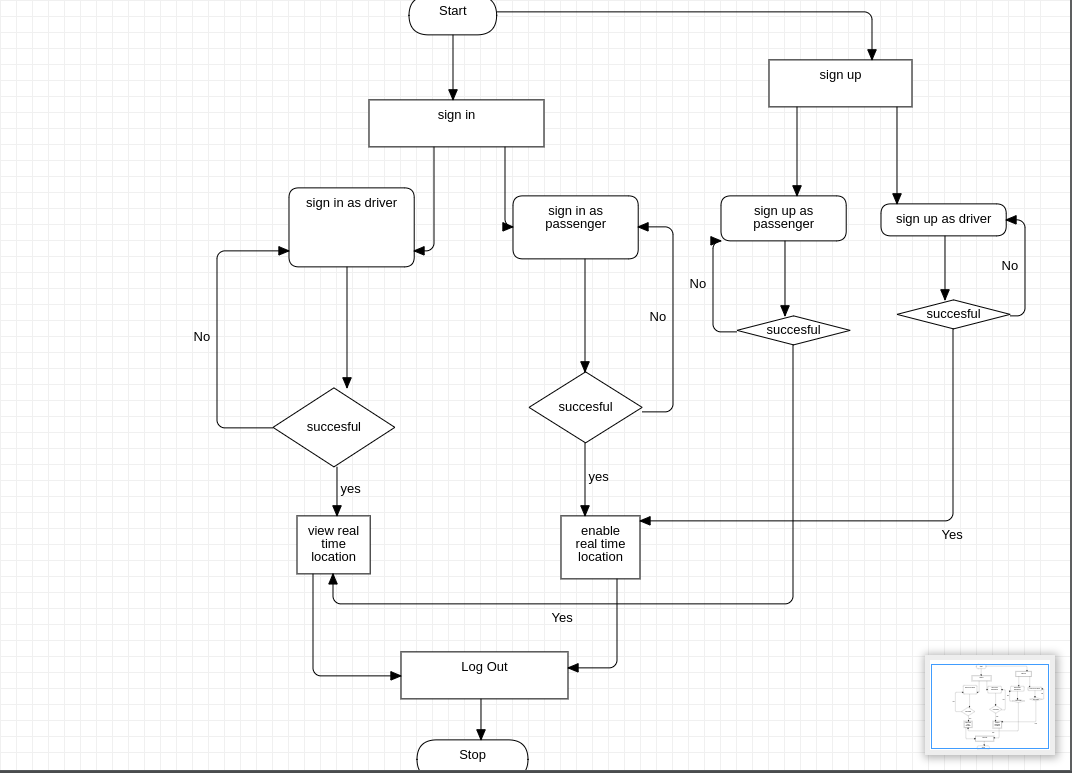
When designing a passenger positioning system:

* **Home screen**: The home screen should provide a clear and concise overview of the key features and functionality of the app, such as real-time tracking, booking, and reservations, and driver information.
* **Map view**: The map view should provide a visual representation of the vehicle's location and route, with support for zooming and panning, and real-time updates.
* **Booking and reservations**: The booking and reservations screen should allow passengers to specify their pickup and drop-off locations and times, select their preferred vehicle and driver, and make payments for the service.
* **Driver information**: The driver information screen should provide passengers with information about the driver, such as their name, photo, and contact details, and allow them to contact the driver directly.
* **Notifications and alerts**: The notifications and alerts screen should provide passengers with real-time updates and alerts about their trip, such as pickup and drop-off reminders, arrival notifications, and driver updates.
* **Reviews and ratings**: The reviews and ratings screen should allow passengers to leave reviews and ratings for the driver and the transportation service, with feedback on the driver's performance and the overall quality of the service.
* **Customer support:** The customer support screen should provide passengers with access to customer support, with support for various channels such as phone, email, or chat.
* **In-app messaging:** The in-app messaging screen should provide passengers and drivers with an easy way to communicate directly with each other, with support for text, voice, and video messaging.
* **Account settings**: The account settings screen should allow passengers to manage their account settings, such as their personal information, payment methods, and preferences.
* L**ocalization and multi-language support**: The user interface should support localization and multi-language support for passengers and for passengers and transportation providers in different countries and regions.

Overall, the user interface for a passenger positioning system should be intuitive, user-friendly, and aesthetically pleasing, with a focus on providing a seamless and convenient user experience for passengers and transportation providers.

# Data Flow diagram:

Viewing the overall system specifications of our system, We were able to have a general overview of our system using a flow diagram.



# Technical stack:

These are the various technologies that are to be used by our system:

Front-end:

* React Native for the passenger app
* Bootstrap for CSS
* Html for templating pages.

Back-end:

* Django or Flask for the web server and API development
* PostgreSQL or MySQL for the database
* Redis or Memcached for caching

Other technologies:

* Amazon Web Services (AWS) for cloud infrastructure, such as EC2 for hosting the servers and S3 for storing media files
* Twilio for SMS and voice communication
* Google Maps API or OpenStreetMap for maps and location data
* Stripe or PayPal for payment processing
* Firebase or OneSignal for push notifications
* Docker or Kubernetes for containerization and deployment

# Architecture:

* Client-side app: This is the mobile app used by passengers to

request a ride, track the driver's location, and pay for the ride. It could be

developed using native code or a cross-platform framework like React

Native.

* **Server-side API:** The API is the interface between the client-side app and the various backend services. It handles authentication, validation, and processing of data, and communication with external services. It could be built using a web framework like Django or Flask.
* **Database**: The database stores data about passengers, drivers, rides, payments, and other related information. It could use a relational database like PostgreSQL or MySQL.
* **Ride management system**: This is the core system that manages the allocation of rides to drivers, tracking the location of drivers and passengers, and handling payments. It could use a message queue system like RabbitMQ or Kafka to distribute tasks across multiple servers.
* **Payment gateway**: The payment gateway handles online payments, including verifying credit card information, processing transactions, and managing refunds. It could use a payment processing service like Stripe or PayPal.
* **Notification system:** The notification system sends push notifications to passengers and drivers, keeping them informed about the status of their rides. It could use a service like Firebase or OneSignal.
* **Map and geolocation services:** Map and geolocation services provide real-time location data to the passenger app and ride management system. It could use a service like Google Maps API or OpenStreetMap.
* **Cloud infrastructure:** The system could be hosted on a cloud infrastructure like Amazon Web Services (AWS) or Google Cloud Platform (GCP), providing scalability, reliability, and flexibility.

Overall, this architecture would provide a scalable and reliable system

that can handle a high volume of requests and transactions, while

providing a seamless experience to passengers and drivers.

# Testing and Quality:

Testing and quality assurance are essential components of any software development process, including the development of a passenger positioning system. Here are some testing and quality assurance considerations for such a system:

* **Unit testing**: Develop unit tests to test individual components of the system, such as APIs, database queries, and functions. This helps identify bugs and issues early in the development process.
* **Integration testing**: Develop integration tests to test how the different components of the system work together. This helps identify issues that may arise when different components are combined.
* **Performance testing**: Conduct performance testing to evaluate the system's response time, throughput, and scalability. This helps ensure that the system can handle a high volume of requests and transactions.
* **User acceptance testing**: Conduct user acceptance testing to ensure that the system meets the requirements and expectations of end-users. This helps identify any issues or discrepancies between what the system delivers and what users expect.
* **Security testing**: Conduct security testing to evaluate the system's security posture, including authentication, authorization, and data protection. This helps identify vulnerabilities that could be exploited by attackers.
* **Quality assurance**: Develop a quality assurance process to ensure that the system meets quality standards and best practices. This includes code reviews, automated testing, and peer reviews.
* **Continuous testing:** Implement continuous testing to ensure that the system is always being tested, even after it is deployed. This helps catch any issues that may arise as the system evolves over time.

By incorporating these testing and quality assurance considerations into the development process, you can ensure that the passenger positioning system is reliable, performant, and meets the needs of end-users.

# Method of validation or approval of data:

# Information from end users:

A total of ten(10) taxi drivers and fifteen(15) passengers were questioned.

 99% of the drivers were glad and super excited about the system. They said and I quote “if that kind app go be that mean say we go make plenty money since man no dey turn turn for find passenger,  a go fit just open the app for my phone check side wey customer dey then i go dey”

97% of the passengers were as well pleased with the initiative and said they couldn’t wait to have such a system in place.

# Project Timeline:

Developing a passenger positioning system can be a complex and time-consuming project, with many factors influencing the timeline. Here is a general project timeline for such a system:

**Requirements gathering: 1-2 weeks.**

This includes defining business and user requirements, functional and non-functional requirements, and technical requirements.

**Analysis and planning: 2-3 weeks.**

This includes designing the system architecture, developing a data flow diagram and use case diagram, creating wireframes and mockups of the user interface, and planning the development process.

**Development: 4-5 weeks.**

This includes developing the front-end and back-end of the system, integrating external services like payment gateways and geolocation services, and conducting unit and integration testing.

**Quality assurance and testing: 2-3 weeks.**

This includes conducting performance testing, security testing, user acceptance testing, and continuous testing.

**Deployment and launch: 1-2 weeks.**

This includes deploying the system to a production environment, performing final testing and quality assurance, and launching the system to end users.

Overall, the timeline for developing a passenger positioning system can range from 9-10 weeks, depending on the complexity of the system that is in the case of wanting to add functionalities, the size of the development team, and other factors. It's important to note that unexpected delays and challenges can arise during the development process, so it's important to build in some buffer time to accommodate these issues.

# Conclusion:

In conclusion, a passenger positioning system is a complex and important tool that can greatly benefit transportation companies, drivers, and passengers. By utilizing geolocation technology and a user-friendly interface, this system can provide real-time information about vehicle locations, routes, and estimated arrival times, which can improve passenger safety, satisfaction, and overall transportation efficiency.

Developing such a system requires a thorough understanding of business and user requirements, as well as technical expertise in software development and integration. It's important to carefully consider the system's features, architecture, and technology stack, as well as its budget, timeline, and quality assurance plan.

Ultimately, a well-designed and well-implemented passenger positioning system can offer significant benefits to both transportation companies, drivers and their passengers and can be a valuable asset in today's fast-paced and technology-driven world.