

**REPUBLIC OF CAMEROON**

**Peace-Work-Fatherland**

**MINISTER OF HIGHER EDUCATION**

**FACULTY OF ENGINEERING**

**AND TECHNOLOGY**

**REPUBLIQUE DU CAMEROON**

**PAIX-Travail-Patrie**

**MINISTRE DE L’ENSEIGNEMENT SUPERIEUR**

**FACULTE D’INGINERIE**

**ET TECHGNOLOGIE**

**CEF440:**

**INTERNET AND MOBILE PROGRAMMING**

**Design and implementation of a Passenger positioning system (Municipal commuting App)**

**Presented by:**

**HEUYA NDJANSEB KEVIN KENNEDY FE20A049**

**Tendo Kris Diogwi FE20A114**

**LEKEAKA DELAND ACHANANK Fe20A056**

**MAFONGANG SOUPGUI DENISE VANELLA FE20A061**

**YUVEN BRENDARD WIYFOFE FE20A124**

**INSTRUCTOR**: **Dr. NKEMENI VALERY**

**March 2023**

**PROJECT DESCRIPTION**

**Design and implementation of a Passenger positioning system (Municipal commuting App)**

This is an app that is used by both passengers and drivers, where the passengers use it to specify their positions at a given time, whereas the drivers use it to locate where the passengers are found in a given time. This can help both drivers and passengers in the following ways:

→ It reduces the amount of time passengers have to wait for a taxi.

→ Drivers can optimize fuel consumption as the app will guide them to move to locations where there are more potential customers.

**CONTENT**

**I. ANALYSIS**

**1. Market Research**

**2. User Needs Analysis**

**3. Technical Feasibility**

**4. Design and Development**

**5. Testing and Quality Assurance**

**6. Launch and Deployment**

**7. Maintenance and Upgrades**

**II. REQUIREMENTS**

**→ Functional Requirements**

**GPS Integration**

**User Registration**

**Booking system**

**Notification System**

**Analytics Capabilities**

**Localization Support**

**Customer Support**

**→ Non Functional Requirements**

**Performance**

**Security**

**Reliability**

**Scalability**

**Usability**

**Compatibility**

**Accessibility**

**Maintainability**

**Cost-effectiveness**

**I. ANALYSIS**

The taxi industry is one of the most important industries in the world, and it plays a crucial role in the transportation sector. In recent years, the use of technology in the taxi industry has increased significantly. One of the most significant technological advancements in the taxi industry is the development of software that helps taxi drivers locate passengers. This report aims to analyze the user needs for building software that helps taxi drivers locate passengers.

To build a software that helps taxi drivers locate passengers, the following analysis can be made:

**1. Market Research:**

A market research was conducted to identify the demand for such a software, the competition in the market, and the target audience.

The taxi industry is an essential part of the transportation sector, and it has undergone significant changes in recent years. With the advent of ride-hailing services such as Uber and Lyft, traditional taxi services have faced tough competition. However, the taxi industry still holds a significant market share, and there is room for innovation and improvement.

One potential area for improvement is the use of technology to enhance the passenger experience. One way to achieve this is by developing software that helps taxi drivers locate passengers more efficiently. This report presents the findings of market research conducted to analyse the feasibility of building such software.

Methodology:

The research was conducted through a combination of primary and secondary sources. Primary research involved conducting interviews with taxi drivers, passengers, and industry experts. Secondary research involved reviewing existing literature on the taxi industry and technology trends.

Findings:

The research found that there is a significant demand for software that helps taxi drivers locate passengers more efficiently. Many passengers reported frustration with the current system, where they have to rely on hailing a taxi on the street or calling a taxi company to book a ride. They expressed a desire for a more convenient and reliable way to hail a taxi.

Taxi drivers also expressed interest in using such software, as it would enable them to find passengers more easily and increase their earnings. They noted that the current system often results in wasted time and fuel as they drive around looking for passengers.

Industry experts also confirmed that there is a market for such software, as it would address a significant pain point for both passengers and taxi drivers. They noted that similar software already exists in the ride-hailing industry and that it has been successful in improving efficiency and customer satisfaction.

Conclusion:

Based on the research findings, it is clear that there is a demand for software that helps taxi drivers locate passengers more efficiently. The development of such software has the potential to improve the passenger experience, increase taxi driver earnings, and enhance the overall efficiency of the taxi industry. Therefore, it is recommended that further research and development be conducted to bring this software to market.

**2. User Needs Analysis:**

Identifying the needs and pain points of both the driver and the passenger to develop features that cater to their requirements.

1. User-Friendly Interface: The software should have a user-friendly interface that is easy to navigate. The interface should be intuitive and straightforward, so that taxi drivers can easily use it without any technical knowledge.

2. Real-Time Location Tracking: The software should have real-time location tracking features that allow taxi drivers to track the location of passengers in real-time. This feature will help taxi drivers to locate passengers quickly and efficiently.

3. Integration with GPS: The software should be integrated with GPS systems to provide accurate location information. This integration will help taxi drivers to navigate to the passenger's location quickly and efficiently.

4. Multiple Language Support: The software should support multiple languages to cater to the needs of taxi drivers who speak different languages. This feature will help taxi drivers to use the software easily and efficiently.

5. Availability on Multiple Platforms: The software should be available on multiple platforms, including mobile devices and desktop computers. This availability will help taxi drivers to access the software from anywhere and at any time.

6. Secure Payment System: The software should have a secure payment system that allows passengers to pay for their rides securely. This feature will help taxi drivers to avoid cash transactions and reduce the risk of theft.

7. Feedback System: The software should have a feedback system that allows passengers to rate their rides and provide feedback on the service provided by the taxi driver. This feature will help taxi drivers to improve their service and provide better customer experience.

Conclusion:

In conclusion, building software that helps taxi drivers locate passengers requires careful consideration of user needs. The software should have a user-friendly interface, real-time location tracking, integration with GPS, multiple language support, availability on multiple platforms, secure payment system, and a feedback system. These features will help taxi drivers to provide better service to their customers and improve their overall experience.

**3. Technical Feasibility:**

Analyzing the technical feasibility of integrating GPS, real-time updates, secure payment gateway, and other features into the software.

1. Mobile Application Development: The software should be developed as a mobile application that can be installed on smartphones. The application should be compatible with different operating systems, including Android and iOS.

2. Real-Time Location Tracking: The software should have real-time location tracking features that allow taxi drivers to track the location of passengers in real-time. This feature requires integration with GPS systems and the development of a robust backend system to process location data.

3. User Authentication: The software should have a user authentication system that ensures only authorized users can access the application. This system requires the development of a secure login system that protects user data and prevents unauthorized access.

4. Payment Gateway Integration: The software should have a payment gateway integration system that allows passengers to pay for their rides securely. This feature requires the development of a secure payment gateway system that complies with industry standards.

5. Scalability: The software should be scalable to accommodate an increasing number of users and transactions. This feature requires the development of a robust backend system that can handle high traffic volumes and data processing.

6. Data Security: The software should have robust data security measures to protect user data from unauthorized access and cyber threats. This feature requires the implementation of encryption protocols, firewalls, and other security measures.

7. Cloud Infrastructure: The software should be hosted on cloud infrastructure to ensure high availability and scalability. This feature requires the development of a cloud-based architecture that can handle high traffic volumes and provide reliable service.

Conclusion:

In conclusion, building software that helps taxi drivers locate passengers requires careful consideration of technical feasibility. The software should be developed as a mobile application, have real-time location tracking features, user authentication, payment gateway integration, scalability, data security, and cloud infrastructure. These features will ensure the software is reliable, secure, and can handle high traffic volumes.

**4. Design and Development:**

Designing and developing the software with a user-friendly interface, notification system, analytics capabilities, localization support, and data privacy compliance.

1. User Interface Design: The software's user interface should be designed in a way that is easy to use and understand for both taxi drivers and passengers. The design should be intuitive, visually appealing, and responsive to different screen sizes.

2. System Architecture: The software's system architecture should be designed to handle high traffic volumes and ensure reliable service. The architecture should be scalable, fault-tolerant, and have redundant components to ensure maximum up time.

3. Database Design: The database design should be optimized for fast data retrieval and storage. The database should be designed to handle large amounts of data, including passenger information, payment details, and ride history.

4. API Integration: The software should integrate with different APIs to provide additional features such as weather information, traffic updates, and map data. The integration should be seamless and efficient to ensure a smooth user experience.

5. Testing and Quality Assurance: The software should undergo extensive testing and quality assurance to ensure it meets industry standards and is free of bugs and errors. The testing should include functional testing, performance testing, security testing, and user acceptance testing.

6. Agile Development Methodology: The software development process should follow an agile methodology that emphasizes collaboration, flexibility, and iterative development. This approach allows for quick feedback and adjustments based on user feedback.

Conclusion:

In conclusion, designing and developing software that helps taxi drivers locate passengers requires careful consideration of various factors. These include user interface design, system architecture, database design, API integration, testing and quality assurance, and agile development methodology. By following these guidelines, the software can be designed and developed to ensure maximum reliability, security, and usability.

**5. Testing and Quality Assurance:**

Testing the software to ensure it functions smoothly, is free from bugs, and meets quality standards.

The process of building software that helps taxi drivers locate passengers involves several stages, including testing and quality assurance. .

1. Testing Plan: A testing plan should be developed before beginning the testing process. This plan should include the testing objectives, testing methods, and testing schedule.

2. Functional Testing: Functional testing should be conducted to ensure that the software meets its intended purpose. This includes testing the software's ability to locate passengers, provide directions, and calculate fares.

3. Performance Testing: Performance testing should be conducted to ensure that the software can handle a high volume of requests without crashing or slowing down.

4. Security Testing: Security testing should be conducted to ensure that the software is secure and protected against hacking and other security threats.

5. Compatibility Testing: Compatibility testing should be conducted to ensure that the software works on different devices and operating systems.

6. Usability Testing: Usability testing should be conducted to ensure that the software is user-friendly and easy to navigate.

7. Regression Testing: Regression testing should be conducted after any changes or upgrades to ensure that the software still functions correctly.

8. Quality Assurance: Quality assurance should be conducted throughout the development process to ensure that the software meets the required standards and specifications

Conclusion:

In conclusion, building software that helps taxi drivers locate passengers requires a thorough testing and quality assurance process. This process should include functional testing, performance testing, security testing, compatibility testing, usability testing, regression testing, and quality assurance. By following these guidelines, the software can be developed to meet the needs of its users and provide a seamless experience for both drivers and passengers.

**6. Launch and Deployment:**

Launching the software in the market and deploying it on various devices and operating systems.

After designing and developing software that helps taxi drivers locate passengers, the next step is to launch and deploy it.

1. Market Research: Before launching the software, it is crucial to conduct market research to identify potential users, competitors, and market trends. This research will help in determining the pricing strategy, marketing plan, and target audience.

2. Pricing Strategy: The pricing strategy should be designed to attract users while ensuring profitability. The pricing model can be based on a subscription fee, commission-based model, or a combination of both.

3. Marketing Plan: The marketing plan should be designed to create awareness and generate interest among potential users. The plan can include online advertising, social media marketing, influence marketing, and content marketing.

4. User On-boarding: The user on-boarding process should be designed to make it easy for new users to sign up and start using the software. The process should be intuitive, simple, and require minimal effort from the user.

5. Training and Support: The software should come with training materials and support to help users understand how to use it effectively. The support should be available through different channels such as email, phone, or chat.

6. Continuous Improvement: After launching the software, it is crucial to continuously improve it based on user feedback. This feedback can be collected through surveys, reviews, and analytics data. The improvements can include adding new features, improving existing ones, or fixing bugs and errors.

Conclusion:

In conclusion, launching and deploying software that helps taxi drivers locate passengers requires careful consideration of various factors. These include market research, pricing strategy, marketing plan, user on-boarding, training and support, and continuous improvement. By following these guidelines, the software can be launched and deployed successfully to attract users and ensure profitability.

**7. Maintenance and Upgrades:**

Providing regular maintenance and upgrades to ensure the software remains up-to-date and relevant to user needs.

After launching and deploying software that helps taxi drivers locate passengers, the next step is to ensure its maintenance and upgrades.

1. Regular Maintenance: The software should undergo regular maintenance to ensure it is functioning optimally. This includes fixing bugs and errors, updating security features, and optimizing performance.

2. User Feedback: User feedback should be collected regularly to identify areas that require improvement. This feedback can be collected through surveys, reviews, and analytics data.

3. Upgrades: The software should be upgraded regularly to add new features and improve existing ones. These upgrades should be based on user feedback and market trends.

4. Testing: Before implementing any upgrades or changes, the software should undergo thorough testing to ensure it is functioning correctly.

5. Documentation: All changes and upgrades should be documented to ensure transparency and accountability. This documentation should include the reason for the change, the implementation process, and the expected outcome.

6. Training and Support: Any changes or upgrades should be accompanied by training materials and support to help users understand how to use them effectively.

Conclusion:

In conclusion, maintaining and upgrading software that helps taxi drivers locate passengers requires regular maintenance, user feedback, upgrades, testing, documentation, and training and support. By following these guidelines, the software can continue to function optimally and meet the changing needs of its users.

**II. REQUIREMENTS**

**FUNCTIONAL REQUIREMENTS:**

**1. GPS Integration:**

The software should be able to integrate with the GPS system to provide real-time location updates of both the driver and the passenger.

GPS integration is a crucial functional requirement for taxi driver software as it enables real-time location tracking of both the driver and the passenger. This feature ensures that drivers can easily locate passengers and provide efficient and timely services. Without GPS integration, the software would not be able to provide accurate location tracking, which could result in delays, confusion, and frustration for both drivers and passengers. Therefore, GPS integration is a fundamental requirement that must be met when developing software for taxi drivers.

**2. User Registration:**

The software should allow users to register and create profiles, including their contact information and payment details.

User registration is another critical functional requirement for software designed to help taxi drivers locate passengers. This feature allows passengers to create accounts and provide necessary information such as name, contact details, and pickup location. This information is essential for drivers to identify the passenger and locate them accurately. Additionally, user registration enables drivers to maintain a record of their past rides, which can be helpful for future reference. Overall, user registration is a crucial requirement that must be included in taxi driver software to ensure smooth and efficient operations.

**3. Booking System:**

The software should provide a booking system that allows passengers to request a ride, and drivers to accept or decline the request.

Booking system is also a critical functional requirement for software designed to help taxi drivers locate passengers. This feature allows passengers to book rides in advance and specify their pickup and drop-off locations. The booking system enables drivers to plan their schedules and optimize their routes, resulting in more efficient operations. Additionally, the booking system can also provide passengers with estimated fares and allow them to make payments online, which can improve the overall user experience. Therefore, including a booking system in taxi driver software is essential for ensuring seamless and hassle-free operations.

**4. Notification System:**

The software should have a notification system that sends alerts to both the driver and the passenger about the booking status, location, and estimated time of arrival.

Another critical functional requirement for software designed to help taxi drivers locate passengers is a notification system. This feature allows drivers to receive real-time notifications about available rides and their pickup locations. The notification system can also inform passengers about the status of their ride, such as when the driver has arrived or if there are any delays. By providing timely and accurate information, the notification system can help reduce wait times and improve the overall user experience. Therefore, including a notification system in taxi driver software is crucial for ensuring efficient and effective communication between drivers and passengers.

**5. Payment Gateway:**

The software should have a secure payment gateway that allows passengers to pay for their ride through various modes of payment.

A payment gateway is a crucial functional requirement for building software that helps taxi drivers locate passengers. This is because the software needs to facilitate secure and seamless transactions between the passenger and the driver. The payment gateway should be integrated with various payment methods such as credit/debit cards, mobile wallets, and net banking to provide convenience to passengers. It should also ensure that all transactions are secure and comply with industry standards for data protection. A reliable payment gateway will help build trust between passengers and drivers, leading to increased usage of the software and higher customer satisfaction.

**6. Analytics Capabilities:**

The software should have analytics capabilities that allow administrators to track user behavior, booking patterns, and other metrics.

Analytics capabilities can also be considered as a functional requirement for software designed to help taxi drivers locate passengers. By analyzing data on ride requests, driver availability, and traffic patterns, the software can provide valuable insights to improve efficiency and optimize operations. For example, analytics can help identify popular pickup locations and peak hours of demand, allowing drivers to strategically position themselves for maximum profitability. Additionally, analytics can help track driver performance and identify areas for improvement, such as reducing wait times or increasing customer satisfaction. Therefore, including analytics capabilities in taxi driver software can help drive business success and improve the overall user experience.

**7. Localization Support:**

The software should support localization features such as language translation, currency conversion, and cultural norms.

This feature allows the software to display information in the local language, use local currency, and provide location-specific information such as landmarks and street names.

Without localization support, the software may not be able to cater to the needs of users from different regions, which can result in poor user experience and low adoption rates. Therefore, it is essential to ensure that the software has robust localization support that can handle various languages and cultural differences.

**8. Data Privacy Compliance:**

The software should comply with data privacy regulations such as GDPR and CCPA to ensure user data is protected.

Data privacy compliance is a crucial functional requirement for building software that helps taxi drivers locate passengers. This is because the software will be collecting and processing personal data such as the passenger's name, phone number, and location. To ensure compliance with data privacy regulations, the software must have appropriate security measures in place to protect the data from unauthorized access or disclosure. It should also provide users with clear information about how their data will be used and give them control over their personal information. Failure to comply with data privacy regulations can result in legal penalties and damage to the reputation of the software provider.

**9. Customer Support:**

The software should have a customer support system that allows users to report issues, provide feedback, and seek assistance.

Customer support is an essential functional requirement for building a software that helps taxi drivers locate passengers. The software should have a dedicated customer support team that can assist passengers and drivers in case of any issues or queries. The customer support team should be available 24/7 to provide assistance through various channels such as phone, email, chat, or social media.

The customer support team should be well-trained and equipped with the necessary tools and knowledge to handle different types of issues related to the software. They should also have access to a comprehensive knowledge base that contains information about the software's features, functionalities, and troubleshooting tips.

In addition to providing assistance to passengers and drivers, the customer support team should also gather feedback from them on a regular basis. This feedback can help improve the software's performance and user experience.

**NON-FUNCTIONAL REQUIREMENTS:**

**1. Performance:**

The software should have fast response times and be able to handle a large number of users and requests simultaneously.

The software must be designed to perform efficiently and quickly, as taxi drivers need to find passengers in real-time. The performance of the software can be measured in terms of response time, throughput, and scalability.

Response time refers to the time taken by the software to respond to user requests. In the case of a taxi driver locating passengers, the response time should be minimal so that drivers can quickly find their next passenger.

Throughput refers to the number of requests that can be processed by the software in a given period. The software should have high throughput so that it can handle multiple requests from different drivers simultaneously.

**2. Security:**

The software should have robust security measures in place to protect user data and prevent unauthorized access.

Security is a crucial non-functional requirement when building a software that helps taxi drivers locate passengers. The software should ensure the privacy and confidentiality of both the driver and passenger's personal information, such as their name, phone number, and location. It should also have measures in place to prevent unauthorized access to the system and protect against cyber-attacks, such as data breaches or hacking attempts. Additionally, the software should have secure payment processing capabilities to ensure that financial transactions between drivers and passengers are safe and secure. Overall, security is an essential aspect of any software development project, especially when dealing with sensitive information like personal and financial data.

**3. Reliability:**

The software should be reliable and available 24/7 to ensure passengers can always find a ride when they need it.

It refers to the ability of the software to perform consistently and accurately without any errors or failures. In this context, reliability means that the software should be able to provide accurate information about passenger locations, pick-up times, and routes without any glitches or delays.

The reliability of the software can be achieved through various means such as rigorous testing, error handling mechanisms, and fault-tolerant design. The software should also be designed to handle unexpected situations such as network outages or hardware failures.

A reliable software system will not only improve the efficiency of taxi drivers but also enhance customer satisfaction by providing a seamless experience.

**4. Scalability:**

The software should be scalable to accommodate future growth and increasing demand.

Scalability refers to the ability of the software to handle an increasing number of users without compromising its performance. As more and more taxi drivers start using the software, it should be able to scale up its resources and handle the increased load without any issues.

**5. Usability:**

The software should be easy to use for both drivers and passengers, with intuitive interfaces and clear instructions.

The software must be easy to use and intuitive, with a user-friendly interface and clear navigation. Failure to ensure usability can lead to frustration for the user, decreased adoption, and loss of revenue. Usability testing should be conducted to ensure that the software meets the needs of its users and provides a positive user experience.

**6. Compatibility:**

The software should be compatible with different devices, operating systems, and browsers to ensure a seamless user experience.

Compatibility is a crucial non-functional requirement when building software that helps taxi drivers locate passengers. This is because the software needs to be compatible with the hardware and software systems commonly used by taxi drivers, such as smartphones, GPS devices, and mapping applications. Additionally, the software needs to be compatible with different versions of operating systems and web browsers. Failure to ensure compatibility can result in the software not functioning as intended, leading to poor user experience, loss of revenue, and a decrease in user adoption.

**7. Accessibility:**

The software should be accessible to users with disabilities, with features such as screen readers and keyboard shortcuts.

The software must be accessible to all users, including those with disabilities, such as visual or hearing impairments. This means that the software should comply with accessibility standards, such as WCAG, and provide features such as keyboard navigation, screen reader compatibility, and adjustable font sizes. Failure to ensure accessibility can lead to exclusion of certain users and potential legal issues.

**8. Maintainability:** The software should be easy to maintain and update, with clear documentation and well-structured code.

This requirement refers to the ease with which the software can be modified, updated, and repaired over time. A maintainable software system is one that can be easily adapted to changing requirements, fixed when errors occur, and enhanced with new features.

In the context of a taxi driver locating software, maintainability is important because it ensures that the system remains functional and up-to-date. As new technologies emerge or passenger needs change, the software must be able to adapt quickly and efficiently. This requires a well-structured codebase that is easy to understand and modify.

**9. Cost-effectiveness:** The software should be cost-effective to develop,deploy, and maintain, with efficient use of resources and minimal overheads.

The software should be designed in a way that maximizes its benefits while minimizing its costs. This means creating a solution that is affordable to develop and maintain, while still providing value to the end-users. Cost-effectiveness can be achieved through efficient software design, effective use of resources, and careful consideration of the software's scope and features. By prioritizing cost-effectiveness, developers can ensure the software is sustainable in the long run and can continue to provide value to taxi drivers and passengers alike.