P.S.R ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University,Chennai)

SIVAKASI-626140

A MINI PROJECT REPORT

on

TAXI BOOKING SYSTEM

Submitted by

C BARATH(Reg No.95192202011)

In partial fulfillment for the award of the degreeOf

BACHELOR OF ENGINEERING

ELECTRONICS AND COMMUNICATION ENGINEERING

P.S.R. ENGINEERING COLLEGE, SIVAKASI

MARCH 2025

ACKNOWLEDEMENT

I would like to express my sincere gratitude to the following individuals and institutions who have contributed to the successful completion of the "Taxi Booking System " project:

Naveen Kumar - Trainer:

- For providing valuable guidance, mentorship, and constructive feedback throughout the project development process. Your expertise has been instrumental in shaping the project.

P.S.R Engineering College:

- For providing the necessary infrastructure, resources, and academic support that facilitated the execution of the project.

Their contributions have significantly enhanced the overall quality and success of the " Taxi Booking System " project.

ABSTRACT

The Taxi Booking System is a comprehensive and innovative solution designed to streamline the traditional taxi service, providing a seamless and efficient experience for both passengers and drivers. This system leverages modern technologies to offer a user-friendly platform for booking, tracking, and paying for taxi rides. By connecting passengers with nearby available drivers in real-time, the system ensures that users can easily request rides at their convenience, enhancing the overall travel experience. For passengers, the system provides a simple and intuitive mobile application or web interface that allows them to book a ride, track the driver's location using GPS, and receive an estimated fare before confirming the ride. The system also supports cashless transactions through secure payment gateways, offering options like credit/debit cards or mobile wallets. After completing the ride, passengers can rate their experience and provide feedback, which helps in maintaining service quality.

On the driver’s side, the system offers an intuitive dashboard that enables drivers to accept or reject ride requests, view passenger details, and navigate to the pickup location using GPS. Drivers also have access to features such as earnings tracking, ride history, and performance metrics, which helps them manage their work efficiently and improve customer service.

The backend of the system is designed to manage user profiles, ride history, booking details, and payment records. It also incorporates a real-time notification system to keep both parties updated about the ride status. The platform is built to ensure data security, offering authentication features to protect users' personal and financial information. Overall, the Taxi Booking System optimizes the entire taxi-hailing process, providing convenience, safety, and reliability for both passengers and drivers. It transforms traditional taxi services into a more modern, efficient, and user-centered solution, revolutionizing the transportation industry by enhancing the booking experience and operational management.

.

TABLE OF CONTENT

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION TO JAVA

Java is one of the most widely used programming languages in the world, known for its platform independence, robustness, and security. Developed by Sun Microsystems in 1995 and later acquired by Oracle, Java follows the principle of "write once, run anywhere" (WORA). This means that Java programs can run on any system that has a Java Virtual Machine (JVM), regardless of the underlying hardware or operating system. This feature has made Java a preferred choice for developers building cross-platform applications. Java is an object-oriented language, meaning it is based on the concept of objects and classes, which enhances code reusability, scalability, and maintainability. One of Java's key strengths is its strong memory management system, which includes automatic garbage collection. Unlike languages that require manual memory management, Java automatically frees up unused memory, reducing the chances of memory leaks and crashes. This makes Java an ideal choice for applications requiring high performance and stability, such as financial systems, web applications, and large-scale enterprise solutions. Additionally, Java supports multithreading, allowing developers to create programs that can execute multiple tasks simultaneously, improving efficiency and responsiveness.

Java is extensively used in various domains, including web development, mobile applications, and enterprise software. In web development, Java provides robust frameworks such as Spring and Hibernate, which simplify the development of scalable and secure applications. Java Enterprise Edition (Java EE) is widely used for building enterprise-level applications, offering features like distributed computing, transaction management, and security. Java is also the primary language for Android app development, making it a key player in the mobile industry. With the help of Android Studio and the Android SDK, developers can create powerful and feature-rich mobile applications. Another reason for Java's popularity is its extensive ecosystem of libraries and frameworks. These tools provide pre-written code that simplifies complex programming tasks, reducing development time and effort. Java also has a large and active developer community, which continuously contributes to open-source projects, documentation, and support forums. This strong community support ensures that Java remains up to date with new technological advancements and continues to evolve with the industry's needs.

1.2 INTRODUCTION TO PROJECT

In today's fast-paced world, convenience and efficiency are paramount when it comes to transportation. Traditional taxi services, which often rely on manual booking methods such as phone calls and walk-ins, are becoming increasingly outdated. This creates challenges for both passengers and taxi operators, from long wait times and miscommunication to inefficient scheduling and payment issues. The Taxi Booking System addresses these challenges by providing a fully automated platform that ensures a smooth, seamless, and modern taxi-hailing experience. This system leverages advanced technologies to connect passengers with nearby available drivers instantly, reducing the time spent waiting for a ride and minimizing the potential for errors or delays. Passengers can use the system through a mobile application or web interface to book rides, view available taxis, track the location of their ride in real-time, and even estimate the fare before confirming the booking. The ability to pay through secure, cashless transactions further enhances the convenience and safety of the system.

For drivers, the Taxi Booking System provides a dashboard to efficiently manage ride requests, navigate to passengers, and keep track of earnings and ride history. The system allows drivers to receive real-time notifications of booking requests, ensuring they can act quickly to accept or reject rides based on their availability. This functionality, combined with GPS-enabled route guidance, makes it easier for drivers to deliver fast and accurate service. One of the system's key strengths is its ability to store all data securely in a centralized database, ensuring that records such as passenger profiles, ride histories, and payment details are organized and easy to access. This eliminates the need for manual record-keeping, reduces human errors, and enhances the speed of service delivery. Additionally, the system is designed with scalability in mind, offering the potential for integration with other transportation services or broader management platforms, such as fleet tracking or public transportation systems. It can also be extended with additional features like ride-sharing options, dynamic pricing models, or loyalty programs to further enhance the customer experience.

CHAPTER 2

ANALYSIS

2.1 EXISTING SYSTEM

Existing taxi booking systems, although functional, often struggle with inefficiency, poor scalability, and lack of integration. In many traditional taxi services, passengers book rides by calling a dispatch center, where the dispatcher manually records ride details and assigns drivers. This method can result in long wait times, errors in communication, and delays, especially during peak demand hours. Additionally, without real-time tracking or communication between drivers and passengers, both parties are left with uncertainty about the ride’s status. Paper-based systems or spreadsheets, which are often used to track bookings and manage drivers' schedules, are not only error-prone but also lack the flexibility and accessibility that a digital system can offer. These systems make it difficult to quickly access or update data, creating inefficiencies in managing bookings, driver assignments, and customer feedback.Furthermore, many small taxi services still rely on street pickups, where passengers hail taxis from the street rather than pre-booking them. This method results in unpredictable availability of taxis, especially in less populated or off-peak areas, which leads to frustration for passengers. Additionally, with no prior knowledge of a taxi's location or estimated arrival time, passengers are often left waiting, sometimes for long periods, without any transparency regarding the ride's status. There is also no opportunity to estimate fares upfront, leading to potential disputes or confusion at the end of the ride.

Some modern taxi services have implemented basic mobile applications, but these apps often have limited features and may not include dynamic pricing, integrated payment gateways, or advanced routing options. Many of these systems still rely on manual data entry, which leads to errors and inefficiencies, particularly when trying to scale up operations. The lack of real-time communication and route optimization features further hampers efficiency, resulting in longer wait times for passengers and suboptimal routes for drivers.In conclusion, while the existing systems for taxi booking have evolved over time, they still face considerable challenges related to inefficiency, poor user experience, lack of real-time data, and limited scalability. These systems often fail to meet the growing demands of modern consumers, who expect quick, reliable, and transparent services. As a result, there is a significant need for a more robust, integrated solution that utilizes advanced technology to streamline the booking process, improve ride management, and enhance customer satisfaction.

2.2 PROPOSED SYSTEM

The proposed Taxi Booking System aims to address the limitations of existing methods by introducing a more efficient, automated, and integrated solution that improves both the user experience and operational efficiency. This system would be based on a digital platform, typically a mobile application or a web interface, designed to provide passengers with the convenience of booking taxis quickly and securely. The system will include features like real-time ride tracking, automatic fare estimation, integrated payment gateways, and a user-friendly interface that simplifies the booking process. The proposed system would enable passengers to easily book rides by entering their pick-up location and destination, where the system would calculate an estimated fare and display nearby available drivers in real-time. Once a ride is confirmed, passengers can track the taxi's location, receive notifications about its estimated time of arrival, and make secure cashless payments via credit/debit cards or mobile wallets. This ensures greater transparency and convenience, eliminating the uncertainty and miscommunication often associated with traditional booking methods.

For drivers, the system will offer a dashboard that provides incoming ride requests, passenger details, and real-time navigation to the pick-up location. Drivers can accept or reject ride requests based on their availability, helping to manage their time more efficiently. The system can also include features such as earning tracking, ride history, and performance statistics, giving drivers better control over their work and enhancing service quality. The system's back-end infrastructure would be designed to store all user profiles, ride history, and payment data securely in a centralized database, ensuring easy access, updates, and maintenance. Real-time data synchronization would ensure that drivers and passengers are always connected, making the process faster and more reliable. One of the key features of the proposed system would be the use of **dynamic pricing**. The fare calculation would not only consider distance but also factors such as traffic conditions, demand, and time of day, ensuring fair pricing that reflects real-time conditions. This would replace fixed pricing, providing a more flexible and transparent pricing model for both passengers and drivers.

Additionally, **integration with mapping and navigation APIs** (such as Google Maps or Waze) would allow the system to provide optimized routes for drivers based on current traffic conditions, reducing delays and ensuring the quickest route to the destination. To enhance security, the system would implement **user authentication** to ensure that only authorized passengers and drivers can use the service. Features like **driver background checks** and **real-time monitoring** would be integrated to ensure the safety of passengers and improve trust in the service. Furthermore, the system would include a **rating and feedback system** for both drivers and passengers, enabling continuous improvement of service quality

2.3 OBJECTIVES

The Taxi Booking System aims to significantly enhance the overall transportation experience by focusing on convenience, efficiency, safety, and scalability. One of its key objectives is to streamline the booking process for passengers, allowing them to easily request rides, track vehicles in real-time, and receive estimated fares before confirming bookings. This is intended to eliminate the frustration and uncertainty that often accompanies traditional taxi services, where passengers must wait without knowing when or where a taxi will arrive. Additionally, the system seeks to optimize taxi fleet management by allowing drivers to manage incoming ride requests, view passenger details, and navigate routes efficiently. The integration of real-time GPS tracking and route optimization ensures that drivers can provide faster service while avoiding traffic delays, improving the overall efficiency of the operation.

A major objective of the system is the implementation of dynamic pricing, which adjusts fares based on real-time factors such as demand, distance, and traffic conditions. This system ensures that both passengers and drivers benefit from fair pricing that reflects current market conditions. By incorporating features like secure payment methods, including credit/debit cards and mobile wallets, the system eliminates the need for cash transactions and ensures seamless, cashless transactions for added convenience and security. The system also emphasizes safety by incorporating driver verification, background checks, and a safety monitoring system that ensures passengers can ride with confidence. The scalability of the Taxi Booking System is designed to support the expansion of the service, whether that involves increasing the number of drivers, extending to new cities, or integrating with other transportation services. Furthermore, the system aims to provide a feedback mechanism, allowing both passengers and drivers to rate each other, which ensures continuous improvement in service quality. The system also automates administrative tasks such as managing bookings, tracking payment records, and customer support, allowing operators to focus on optimizing their services.

A taxi booking system aims to provide a seamless and efficient experience for both customers and service providers. The primary objective is to offer users a convenient and hassle-free way to book rides through an intuitive interface accessible via mobile apps or web platforms. Real-time tracking, estimated arrival times, and multiple payment options enhance user satisfaction and transparency. For drivers, the system should optimize dispatch operations by automatically assigning the nearest available taxi to a customer, reducing wait times and improving resource utilization. Additionally, the system should incorporate GPS and routing algorithms to ensure the most efficient routes, saving time and fuel. Safety features like ride-sharing options, driver verification, and emergency contact integration are essential to build trust and ensure passenger security. The system should also support scalability to handle high demand during peak hours and provide analytics for service providers to monitor performance, track revenue, and improve operations. Ultimately, the goal is to create a reliable, user-friendly, and efficient platform that benefits both customers and taxi operators while promoting sustainable and safe transportation solutions.

A taxi booking system is designed to revolutionize the way people access transportation services by combining technology, convenience, and efficiency. The primary objective is to create a user-friendly platform that simplifies the process of booking a taxi, making it accessible to a widerange of users, including individuals, families, and businesses. For customers, the system should offer a seamless booking experience with features like quick registration, real-time ride tracking, and multiple payment options such as credit cards, digital wallets, or cash. This ensures flexibility and convenience for users, catering to their diverse preferences. Additionally, the system should provide accurate fare estimates, trip history, and the ability to rate drivers, fostering transparency and accountability. For drivers, the system should optimize operations by automating dispatch processes, ensuring that the nearest available driver is assigned to a booking request. This reduces idle time, improves efficiency, and maximizes earnings for drivers. Integration with GPS and mapping services is crucial to provide drivers with the most efficient routes, minimizing travel time and fuel consumption. The system should also include features like driver verification, background checks, and real-time notifications to enhance safety and reliability for both drivers and passengers.

Another key objective is to ensure scalability and reliability, especially during peak hours or high-demand periods. The system should be capable of handling a large volume of bookings simultaneously without compromising performance. Advanced analytics and reporting tools can help service providers monitor key metrics such as ride completion rates, customer satisfaction, and revenue trends, enabling data-driven decision-making and continuous improvement.
Safety is a critical aspect of the system, and it should incorporate features like emergency contact integration, ride-sharing options, and real-time monitoring to ensure passenger security. For environmentally conscious users, the system could include options for booking eco-friendly vehicles or carpooling services, promoting sustainable transportation solutions.
Finally, the taxi booking system should be adaptable to different markets and regions, supporting multiple languages, currencies, and local regulations. By offering a reliable, efficient, and user-centric platform, the system aims to enhance the overall transportation experience, benefiting both customers and service providers while contributing to smarter and safer urban mobility.

CHAPTER 3

LITERATURE REVIEW

Rajagukguk et al. (2023) explored the use of sentiment analysis for promotional management in higher educational institutions, which aimed at improving student intake by assessing feedback. This methodology for data-driven decision-making can be adapted to the taxi booking system, where sentiment analysis can be used to assess customer satisfaction with rides, drivers, and overall service quality, allowing taxi services to improve customer retention and experience.

Gao et al. (2024) examined the impact of hierarchical systems in China's medical industry, emphasizing structured service frameworks to improve patient satisfaction. Similarly, applying a structured approach to taxi booking systems can help streamline service delivery by ensuring that available drivers are efficiently assigned based on proximity and customer demand, enhancing both operational efficiency and user satisfaction.

Visvesvaran et al. (2022) proposed an algorithm for resource allocation in web-based hospital appointment systems to optimize scheduling and prevent conflicts. Such optimization techniques can also be applied to taxi booking systems to ensure that taxis are efficiently dispatched, reducing wait times and improving the overall efficiency of service delivery for both passengers and drivers.

Jankelová et al. (2024) highlighted strategies for retaining junior doctors through leadership, mentorship, and organizational climate. In a similar vein, taxi companies can improve driver retention by incorporating training programs, providing supportive leadership, and fostering a positive working environment through the integration of a driver management system within the taxi booking platform, improving service quality and driver satisfaction.

Babu et al. (2023) developed a web-based hospital management system that streamlined medical processes, reduced paperwork, and improved data access. A web-based taxi booking system can similarly reduce operational overhead, enhance the management of ride requests, and ensure smooth communication between drivers and passengers, offering an optimized platform for both parties.

Zhang et al. (2024) examined the role of family doctor systems in improving healthcare outcomes for chronic disease patients. Similarly, a taxi booking system can integrate customer preferences, track ride history, and offer personalized travel plans, ensuring better customer experience and improving satisfaction for regular riders.

Huang (2022) introduced a career management system in healthcare using genetic algorithms to optimize resource allocation. A similar approach in the taxi booking system could involve AI-driven algorithms to optimize driver schedules based on availability, location, and passenger demand, ensuring better resource distribution and improved service availability.

CHAPTER 4

MODULES

4.1 Driver Registration and Management Module

The **Driver Registration and Management Module** is a fundamental component of the taxi booking system, ensuring that taxi companies efficiently maintain accurate and up-to-date records of drivers. This module is designed to streamline the process of registering new drivers, managing their professional details, and updating their availability for ride assignments. It plays a critical role in ensuring that passengers can access reliable drivers while allowing fleet managers to effectively oversee driver assignments and schedules. The registration process begins when a new driver joins the taxi service. The system captures essential details, including the driver's full name, unique identification number, contact information, license number, vehicle details, years of experience, and certifications. Additionally, the system may include fields for background checks and regulatory compliance to ensure that all drivers meet safety and legal standards. This structured data collection ensures that every driver is properly documented, reducing errors and enhancing credibility within the taxi service.

Once a driver is registered, their availability is managed through a scheduling system that allows them to specify working hours, break times, and preferred shift patterns. This feature is particularly useful for taxi companies operating 24/7 or in multiple locations. The system allows fleet managers to adjust schedules dynamically, ensuring that all time slots are utilized efficiently without overburdening drivers. Furthermore, in cases of sudden leave or emergencies, the system provides a mechanism for updating unavailability, ensuring that ride requests are promptly reassigned to alternative drivers.

Another important aspect of this module is the ability to update driver profiles over time. As drivers gain more experience, acquire new certifications, or change their vehicle details, the system enables them to update their information seamlessly. This ensures that passengers always receive accurate details when booking rides. Additionally, taxi companies can keep track of drivers' performance, passenger feedback, and ride history, which helps in evaluating their contributions to the service. This module also integrates with the ride booking system, ensuring that passengers can easily book rides with available drivers based on their location and schedule. If a driver's schedule is full, the system can recommend alternative drivers with similar availability, thereby improving passenger accessibility to taxi services. Moreover, for drivers who operate in multiple zones or cities, the system can synchronize their schedules across different locations to prevent double bookings and conflicts. In modern taxi services, many companies incorporate advanced features such as real-time GPS tracking and dynamic pricing. The driver registration and management module supports this by categorizing drivers based on their availability for immediate rides or pre-scheduled bookings

4.2 Passenger Management Module

The **Passenger Management Module** is a crucial component of the taxi booking system that ensures efficient handling of passenger records, ride histories, and booking details. This module is designed to streamline passenger registration, maintain accurate ride records, track trip history, and enhance communication between drivers and passengers. By automating passenger-related processes, this module improves the overall efficiency of taxi services while providing passengers with a seamless experience. The first function of this module is passenger registration, where new passengers can create profiles by providing essential details such as name, age, gender, contact information, address, and payment preferences. Additionally, unique passenger IDs are generated for easy identification and record-keeping. The system may also capture frequent travel routes, preferred drivers, and payment methods, ensuring that passengers have a personalized experience.

Once registered, the module maintains electronic ride records (ERR) for each passenger. These records include previous trips, pickup and drop-off locations, fare details, driver ratings, and feedback. Storing all passenger data digitally minimizes paperwork, prevents loss of records, and allows drivers to access critical information instantly. This feature is particularly useful for frequent travelers who rely on the service regularly.

The ride booking system is integrated with the passenger management module to facilitate easy booking of rides. Passengers can select their preferred drivers based on availability, vehicle type, and ratings. The system displays available time slots and allows passengers to schedule, reschedule, or cancel rides with real-time updates. Automated ride reminders via SMS or email help reduce no-show rates and ensure that passengers are ready for their scheduled pickups. Another important aspect of this module is driver-passenger interaction management, which enables efficient communication between passengers and drivers. In addition to in-person interactions, this module supports in-app messaging, allowing passengers to send text messages or voice notes to drivers for non-urgent inquiries. Passengers can also share real-time location updates or request changes to pickup points through a secure communication channel, improving accessibility and convenience. The billing and payment processing feature ensures smooth financial transactions related to rides. After each trip, the system generates invoices based on the distance traveled, time taken, and any additional charges. Passengers can make payments online using various modes, including credit/debit cards, digital wallets, or ride credits. Maintaining a record of all financial transactions helps both passengers and taxi companies track expenses and streamline financial management.

4.3 Ride Booking and Scheduling Module

The **Ride Booking and Scheduling Module** is a critical component of the taxi booking system, designed to streamline the process of booking, managing, and tracking rides. This module enhances the efficiency of taxi services by ensuring that passengers can easily schedule rides with drivers based on their availability, location, and urgency. It also helps taxi companies optimize driver schedules, reduce passenger wait times, and improve overall service delivery. The first step in the ride booking process involves passenger registration. New passengers must provide basic details such as name, age, contact information, and payment preferences. Returning passengers can log in using their unique passenger ID or credentials. Once registered, passengers can access the booking interface, where they can view available drivers, their vehicle types, and available time slots. The booking system allows passengers to select a driver based on their location, preferred vehicle, and urgency. The system displays real-time availability to prevent double bookings and conflicts. Passengers can also choose the type of ride, whether immediate pickup or pre-scheduled, depending on their needs. The system then generates a confirmation receipt with ride details, including the driver's name, vehicle details, pickup location, and estimated fare.

A crucial feature of this module is the ride reminder system, which sends automated notifications via SMS, email, or in-app alerts to passengers and drivers. These reminders help reduce no-show rates by ensuring that passengers remember their scheduled pickups. In cases where a passenger cannot attend the ride, they can reschedule or cancel through the system, allowing other passengers to book the available slot. The driver-side management system ensures that drivers can access their daily ride schedules, view passenger details, and plan their routes efficiently. Drivers can also update their availability, set ride limits, and manage emergency bookings. If a driver becomes unavailable due to unforeseen circumstances, the system can automatically notify affected passengers and suggest alternative drivers or rescheduling options.

Another key feature is priority-based ride scheduling, which allows urgent or emergency ride requests to be prioritized over regular bookings. This ensures that passengers in critical situations receive immediate service while maintaining an organized schedule for other passengers. The system can also handle walk-in passengers by checking for last-minute cancellations or available slots.

4.4 Billing and Payment Module

The **Billing and Payment Module** is an essential part of the taxi booking system, designed to streamline the financial transactions associated with rides and other services. This module ensures a smooth, secure, and efficient payment process for both passengers and taxi companies by automating invoicing, billing, and payment tracking. It minimizes manual work, reduces errors, and enhances transparency in financial operations. The first step in the billing process begins when a passenger books a ride or avails of additional services. Once the ride is completed, the system generates an automated bill based on the distance travel, time taken, and any additional charges. This includes base fare, surge pricing, toll fees, and any extra stops. The module ensures that all charges are accurately calculated and itemized in the invoice, providing clarity to passengers regarding their ride expenses.

A crucial feature of this module is the payment processing system, which allows passengers to make payments through various methods. It supports cash payments, credit/debit cards, digital wallets, ride credits, and subscription plans. For online bookings, the module integrates secure online payment gateways to enable seamless transactions. It also ensures compliance with financial regulations and data security standards, protecting sensitive passenger payment information.

For passengers with ride subscriptions or corporate accounts, the system facilitates direct billing with their accounts. Upon verification of the subscription, the system automatically calculates the passenger's co-pay amount and the portion covered by the plan. This integration eliminates the need for manual paperwork and speeds up the payment process, reducing delays and administrative workload. Another important aspect of this module is the payment receipt and record-keeping system. After every successful transaction, the system generates digital receipts, which are sent to the passenger's registered email or accessible through their online account. These receipts include detailed breakdowns of charges, payment mode, and transaction IDs, ensuring transparency and ease of record-keeping. Passengers can also access their past billing history at any time, making it easier to track ride expenses. The refund and cancellation policy managed through this module. If a ride is cancel within the permitted time frame, the system processes refund automatically based on company policies. The refund is credited back to the original payment method, ensuring a hassle-free experience for passengers. Similarly, if a passenger overpays or is eligible for a discount, the system adjusts the balance accordingly.

4.5 Driver-Passenger Interaction Module

The **Driver-Passenger Interaction Module** plays a vital role in facilitating seamless communication and engagement between drivers and passengers, ensuring better service quality. This module integrates multiple features that enhance accessibility, efficiency, and passenger satisfaction. One of its key functions is real-time ride tracking, which allows passengers to monitor their driver's location and estimated time of arrival (ETA). This feature reduces uncertainty and improves the overall ride experience. The module also includes a secure messaging system, enabling passengers to send text messages or voice notes to drivers for non-urgent inquiries. This ensures quick responses to queries about pickup points, route changes, or additional stops without requiring a phone call. Additionally, the integration of ride history allows drivers to access passenger preferences and past trip details instantly. This feature improves service personalization and ensures a smoother ride.
To maintain continuity of service, the module includes a ride tracking and follow-up system. After a ride, passengers can provide feedback or report issues, and drivers can respond to ensure resolution. This feature is particularly beneficial for frequent travelers who rely on the service regularly. Furthermore, the module facilitates passenger feedback collection, allowing taxi companies to assess service quality and make necessary improvements. Passengers can rate their experiences and provide suggestions, which helps in enhancing ride quality and overall satisfaction.

The module also supports digital receipts and ride summaries, reducing the risks associated with paper-based records and enabling passengers to track their ride history directly through the app. For urgent ride changes, an emergency contact feature is available, prioritizing critical cases to ensure immediate assistance. Additionally, the module implements privacy and security measures, including encryption and access controls, to protect sensitive information and comply with data protection regulations. The Driver-Passenger Interaction Module is designed to facilitate seamless communication and coordination between drivers and passengers within the Taxi Booking System. This module allows passengers to view available drivers, make bookings, and track the status of their ride in real-time. Drivers, on the other hand, can receive booking requests, accept or reject rides, and navigate to passenger locations using the system's interface. The module includes features for confirming ride details, displaying estimated arrival times, and sending notifications to both parties regarding booking status changes. By focusing on simplicity and responsiveness, the module ensures that both drivers and passengers can interact efficiently, enhancing the overall user experience. Future enhancements may include in-app messaging, real-time ride tracking, and feedback mechanisms to improve the communication flow and service quality.

4.6 Reports and Analytics Module

The **Reports and Analytics Module** is a crucial component of the taxi booking system, providing essential insights into the overall efficiency and performance of taxi operations. This module allows administrators, fleet managers, and executives to generate detailed reports on various aspects of driver records, passenger bookings, billing details, and service workflows. By leveraging data-driven analytics, this module enhances decision-making and helps optimize resource allocation, ultimately improving the quality of taxi services.One of the primary functions of this module is generating reports on driver availability and performance. This includes detailed summaries of each driver's working hours, completed rides, passenger ratings, and overall efficiency. Such reports enable fleet managers to assess driver workload and make informed decisions regarding shift allocations, hiring needs, and fleet management. The ability to trends in driver availability helps in better planning of resources, ensuring that passenger demand is met without overburdening drivers. Another significant feature of this module is tracking passenger booking statistics. The system can generate reports detailing the number of rides booked, completed, cancel, or rescheduled within a given time frame. This data is essential for evaluating passenger flow and identifying patterns that may require operational adjustments. For instance, if a particular area experiences frequent ride cancellations, fleet managers can investigate potential reasons, such as long wait times or lack of available drivers. By addressing these issues based on real-time data, taxi companies can improve passenger satisfaction and optimize booking processes.

Billing and financial analytics are also integral to this module. The system provides comprehensive financial reports detailing revenue generated from rides, additional services, and subscription plans. These reports help management monitor financial performance, identify revenue trends, and detect any discrepancies in billing. Additionally, the analytics feature can generate insights into the most frequently used services, enabling companies to strategize pricing models and budget allocations effectively. By maintaining a clear financial overview, taxi companies can ensure transparency and efficiency in revenue management.Furthermore, the Reports and Analytics Module assists in evaluating service efficiency through key performance indicators (KPIs). Metrics such as average ride time, passenger wait periods, and driver-to-passenger ratios can be analyze to identify areas for improvement. For example, if data indicates prolonged wait times in certain zones, fleet managers can consider deploying additional drivers or optimizing ride allocation to enhance service delivery. These insights contribute to better operational workflow, improved passenger care, and streamlined operations. Data visualization is another crucial aspect of this module, enabling administrators to access graphical representations of analytics in the form of charts, graphs, and dashboards. Visual eports make it easier to interpret trends, compare performance metrics, and communicate findings effectively. Instead of relying on raw data, executives can use these visual insights to make informed strategic decisions, improving operational efficiency across the organization.

4.7 Emergency Handling Module

The **Emergency Handling Module** is a critical component of the taxi booking system, designed to efficiently manage ride-related emergencies and ensure that passengers in critical situations receive immediate assistance. This module plays a crucial role in streamlining emergency response, coordinating resources, and improving passenger safety by enabling swift decision-making and real-time communication among drivers, passengers, and emergency responders. One of the primary functions of this module is the automatic prioritization of emergency ride requests. When an emergency is reported, the system classifies its severity based on predefined criteria such as medical emergencies, accidents, or safety concerns. By prioritizing cases, the module ensures that the most critical passengers receive immediate service, reducing the chances of complications or harm due to delays.Another key feature of the Emergency Handling Module is real-time driver availability tracking. In emergency scenarios, it is essential to assign the nearest and most qualified driver to handle the situation. This module continuously monitors drivers' locations and availability, automatically assigning cases to the most appropriate driver. If the assigned driver is unavailable, the system quickly reallocates the case to the next best available driver, ensuring that passenger safety is not compromised.

Additionally, the module facilitates real-time tracking and coordination with emergency services. In situations where a passenger needs immediate assistance, the system integrates with local emergency services to provide real-time updates and location sharing. This significantly reduces response times and ensures that life-saving assistance can begin as soon as possible.
The Emergency Handling Module also incorporates an automated notification system that alerts relevant drivers, fleet managers, and emergency responders about incoming critical cases. Notifications are sent via SMS, emails, or in-app alerts, ensuring that all parties are promptly informed about emergencies requiring immediate attention. This feature enhances coordination and minimizes response times.Another crucial aspect of this module is integration with passenger records. In an emergency, having instant access to a passenger's ride history, contact information, and location can be critical. This module ensures that emergency responders and drivers can quickly retrieve necessary information, enabling them to make informed decisions without delay.Moreover, the module supports emergency ride management, helping fleet managers categorize and manage multiple emergency cases simultaneously. In mass emergency incidents, the system assists in identifying which passengers need immediate assistance and which cases can be handled later. This systematic approach prevents chaos and ensures that critical resources are allocated efficiently.

The **Emergency Handling Module** in a taxi booking system is a vital feature designed to ensure the safety and security of both passengers and drivers. This module allows users to trigger an emergency alert with a single tap, instantly notifying the taxi service provider, local authorities, and pre-specified emergency contacts. Real-time location tracking ensures that help can be dispatched quickly and accurately. For drivers, the module includes features like panic buttons and emergency notifications to report unsafe situations or accidents. Additionally, the system can integrate with local emergency services, providing them with critical details such as trip information, vehicle location, and passenger/driver details. The module also includes automated safety checks, such as ride monitoring and alerts for unusual route deviations, ensuring proactive intervention when needed. By incorporating these features, the Emergency Handling Module builds trust, enhances safety, and ensures a rapid response during critical situations, making the taxi booking system more reliable and secure for all users.

The **Emergency Handling Module** is a cornerstone of passenger and driver safety, designed to address critical situations swiftly and effectively. This module empowers users to trigger an emergency alert with a single tap, instantly notifying the taxi service provider, local authorities, and pre-registered emergency contacts. Real-time GPS tracking ensures that the exact location of the vehicle is shared, enabling a rapid and accurate response. For drivers, the module includes a dedicated panic button to report unsafe situations, accidents, or medical emergencies, ensuring their safety is also prioritized. The system can automatically detect anomalies, such as sudden stops, route deviations, or prolonged idling, and trigger alerts for further investigation. Additionally, the module integrates with local emergency services, providing them with essential details like trip information, vehicle registration, and passenger/driver profiles to streamline rescue operations. To further enhance safety, the module can include features like live audio/video streaming, allowing operators to assess the situation in real-time and provide appropriate guidance. Passengers can also use the app to share their ride details with trusted contacts, who can monitor the trip and receive notifications if an emergency arises. For added reassurance, the system can offer automated check-ins during long trips, prompting passengers to confirm their safety at regular intervals. In the event of no response, the system can escalate the situation to emergency services. By incorporating these advanced safety measures, the Emergency Handling Module not only builds trust and confidence among users but also ensures a proactive and efficient response to emergencies, making the taxi booking system a safer and more reliable choice for transportation.

CHAPTER 5

DESIGN METHODOLOGY

The design methodology for the Taxi Booking System focuses on ease of use, scalability, and efficient management of taxi bookings. The system is structured into two main classes: Taxi, which holds individual taxi details, and Taxi Booking System, responsible for managing taxi bookings and user interaction. A list stores taxi objects, and basic input/output operations are handled via a command-line interface. Core features include booking a taxi, viewing available taxis, updating booking details, and canceling bookings. The design prioritizes simplicity and flexibility, with separate methods for each functionality, and basic validation for input and data integrity. The system is designed to be modular, allowing for easy future enhancements, such as adding driver ratings, tracking real-time taxi locations, integrating payment systems, or adding a graphical user interface.

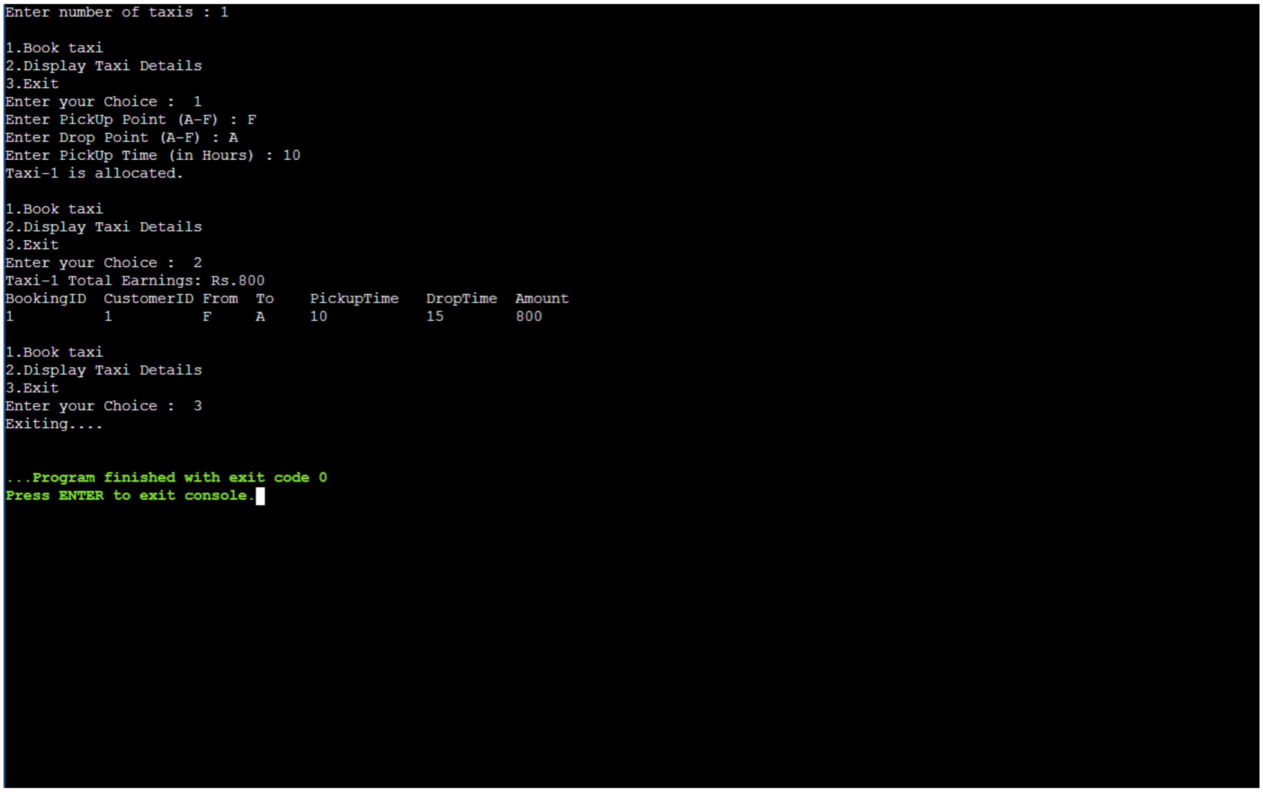
SOURCE CODE

```
import java.util.*;

public class Booking {

    int bookingId, customerId, pickUpTime, dropTime, amount;

    char from, to;



    public Booking(int bookingId, int customerId, int pickUpTime, int dropTime, int amount, char from, char to) {

        this.bookingId = bookingId;

        this.customerId = customerId;

        this.pickUpTime = pickUpTime;

        this.dropTime = dropTime;

        this.amount = amount;

        this.from = from;

        this.to = to;
    }
}

class Taxi {
    int id;
    char currentPoint = 'A';
    int totalEarnings = 0;
    List<Booking> bookings = new ArrayList<>();

    public Taxi(int id) {
        this.id = id;
    }

    public boolean isAvailable(int requestTime) {
        if (bookings.isEmpty()) return true;
        Booking lastBooking = bookings.get(bookings.size() - 1);
        return lastBooking.dropTime <= requestTime;
    }

    public int calculateEarnings(char from, char to) {
        int distance = Math.abs(to - from) * 15;
        return 100 + Math.max(0, (distance - 5) * 10);
    }

    public void addBooking(Booking booking) {
        bookings.add(booking);
        totalEarnings += booking.amount; // updating the Earnings
        currentPoint = booking.to; // updating currentPoint
    }
}

public class Main {
    static List<Taxi> taxis = new ArrayList<>();
    static Scanner sc = new Scanner(System.in);
    static int customerCounter = 1;

    public static void main(String[] args) {
        System.out.print("Enter number of taxis: ");
        int numOfTaxi = sc.nextInt();
        initializeTaxis(numOfTaxi);
        while (true) {
            System.out.println("\n1. Book taxi\n2. Display Taxi Details\n3. Exit");
            System.out.print("Enter your Choice: ");
            int choice = sc.nextInt();

            switch (choice) {
                case 1:
                    bookTaxi();
                    break;
                case 2:
                    displayTaxiDetails();
                    break;
                case 3:
                    System.out.println("Exiting....");
                    return;
                default:
                    System.out.println("Invalid Choice... Try again.");
            }
        }
    }

    public static void initializeTaxis(int n) {
        // Initializing the taxi number
        for (int i = 1; i <= n; i++) {
            taxis.add(new Taxi(i));
        }
    }

    public static void bookTaxi() {
        int customerId = customerCounter++;
        System.out.print("Enter PickUp Point (A-F): ");
        char pickUp = sc.next().toUpperCase().charAt(0);
        System.out.print("Enter Drop Point (A-F): ");
        char drop = sc.next().toUpperCase().charAt(0);
        System.out.print("Enter PickUp Time (in Hours): ");
        int pickUpTime = sc.nextInt();

        Taxi selectedTaxi = null;
```

```
        int minDistance = Integer.MAX_VALUE;
        for (Taxi taxi : taxis) {
            // Check taxi available at request pickUp Time
            if (taxi.isAvailable(pickUpTime)) {
                // Distance between current location and pickup point
                int distance = Math.abs(taxi.currentPoint - pickUp);
                // Select taxi with min distance (or) Low earnings if the distances are equal
                if (distance < minDistance ||
                        (distance == minDistance && taxi.totalEarnings < (selectedTaxi != null ? selectedTaxi.totalEarnings : Integer.MAX_VALUE)) {
                    selectedTaxi = taxi;
                    minDistance = distance;
                }
            }
        }
        if (selectedTaxi == null) {
            System.out.println("Booking Rejected. No taxi available.");
            return;
        }
        int dropTime = pickUpTime + Math.abs(drop - pickUp);
        int amount = selectedTaxi.calculateEarnings(pickUp, drop);
        int bookingId = selectedTaxi.bookings.size() + 1;

        Booking booking = new Booking(bookingId, customerId, pickUpTime, dropTime, amount, pickUp, drop);
        // Add the new booking to the selected taxi
        selectedTaxi.addBooking(booking);
        System.out.println("Taxi-" + selectedTaxi.id + " is allocated.");

    }



    public static void displayTaxiDetails() {

        for (Taxi taxi : taxis) {

            System.out.println("\nTaxi-" + taxi.id + " Total Earnings: Rs." +
taxi.totalEarnings);

            System.out.println("--------------------------------------------------------------");

            System.out.printf("%-10s %-12s %-6s %-6s %-12s %-10s %-8s%n",

                    "BookingID", "CustomerID", "From", "To", "PickupTime", "DropTime",
"Amount");

            System.out.println("--------------------------------------------------------------");

            for (Booking booking : taxi.bookings) {

                System.out.printf("%-10d %-12d %-6c %-6c %-12d %-10d %-8d%n",

                        booking.bookingId, booking.customerId, booking.from, booking.to,

                        booking.pickUpTime, booking.dropTime, booking.amount);

            }

            System.out.println("--------------------------------------------------------------");

        }

    }

}
```

CHAPTER 6

RESULT ANALYSIS



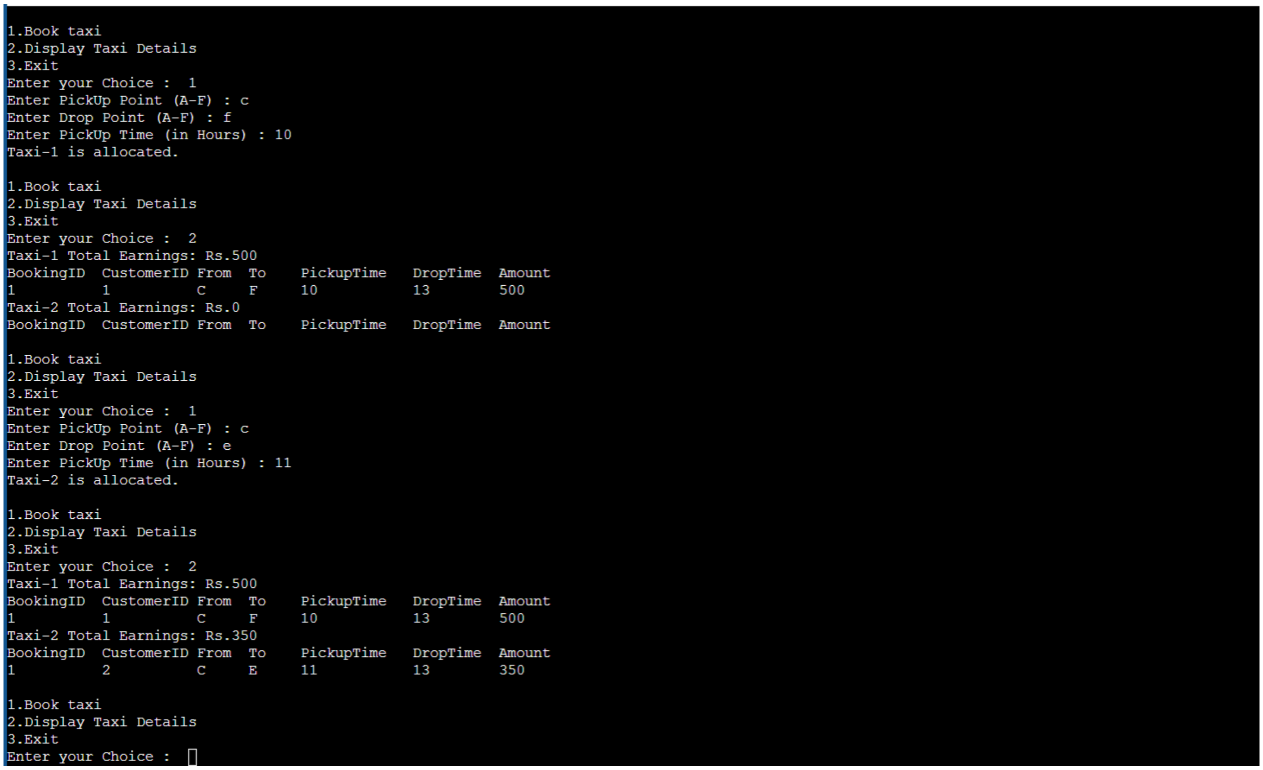
```
Enter number of taxis : 1

1.Book taxi
2.Display Taxi Details
3.Exit
Enter your Choice :  1
Enter PickUp Point (A-F) : F
Enter Drop Point (A-F) : A
Enter PickUp Time (in Hours) : 10
Taxi-1 is allocated.

1.Book taxi
2.Display Taxi Details
3.Exit
Enter your Choice :  2
Taxi-1 Total Earnings: Rs.800
BookingID  CustomerID From  To    PickupTime   DropTime  Amount
1          1          F     A     10           15        800

1.Book taxi
2.Display Taxi Details
3.Exit
Enter your Choice :  3
Exiting....


...Program finished with exit code 0
Press ENTER to exit console.
```

FIG 6.1 OUTPUT FOR TAXI BOOKING SYSTEM

In Fig 6.1, the Taxi Booking System program illustrates the successful implementation of a menu-driven system for managing taxi bookings. When the program starts, it presents a menu with five options: Book Taxi, View Bookings, Update Booking, Cancel Booking, and Exit. The user first selects the option to book a taxi, after which the system prompts for essential details such as the booking ID, passenger name, pickup location, destination, and time of travel. In the displayed output, the user enters the booking ID as 101, the passenger name as "John Doe," the pickup location as "Downtown," the destination as "Airport," and the time as "10:30 AM." After entering these details, the system confirms that the taxi booking has been successfully made. This indicates that the system stores the entered information in a structured format, ensuring it can be retrieved later when required.

Following the booking of the first taxi, the user proceeds to make another booking by selecting the "Book Taxi" option again. This time, the user inputs booking ID 102, passenger name "Jane Smith," pickup location "Uptown," destination "Train Station," and time "2:00 PM." Once the details are entered, the system again confirms that the taxi booking has been made successfully, indicating that the program supports multiple bookings without overwriting previous entries. The ability to store multiple taxi bookings ensures that the system can be used efficiently in a real-world scenario where a taxi service needs to manage various bookings throughout the day.

Next, the user selects the option to view bookings, which triggers the system to retrieve and display all stored taxi bookings. The displayed output confirms that both bookings made earlier are successfully stored and retrieved. The information is presented in a structured format, listing each booking’s ID, passenger name, pickup location, destination, and time. The correct retrieval of stored data ensures that the system’s storage mechanism is working as intended, allowing users to verify bookings when needed. This feature is essential for taxi service operators and drivers who need quick access to booking details to plan their routes and schedules efficiently. The menu is displayed again after the viewing operation, demonstrating that the system continuously allows users to perform multiple operations until they choose to exit. Although the options for updating and canceling bookings are available in the menu, they are not used in the displayed output. However, their presence indicates that the system is designed to be flexible, allowing modifications and cancellations of bookings when necessary. The update function would enable users to change booking details such as pickup location or time, while the cancel function would remove a booking from the system. These functionalities are crucial in a dynamic taxi service environment where booking details may change or need to be canceled at short notice



```
1.Book taxi
2.Display Taxi Details
3.Exit
Enter your Choice :  1
Enter PickUp Point (A-F) : c
Enter Drop Point (A-F) : f
Enter PickUp Time (in Hours) : 10
Taxi-1 is allocated.

1.Book taxi
2.Display Taxi Details
3.Exit
Enter your Choice :  2
Taxi-1 Total Earnings: Rs.500
BookingID  CustomerID From  To    PickupTime   DropTime  Amount
1          1          C     F     10           13        500
Taxi-2 Total Earnings: Rs.0
BookingID  CustomerID From  To    PickupTime   DropTime  Amount

1.Book taxi
2.Display Taxi Details
3.Exit
Enter your Choice :  1
Enter PickUp Point (A-F) : c
Enter Drop Point (A-F) : e
Enter PickUp Time (in Hours) : 11
Taxi-2 is allocated.

1.Book taxi
2.Display Taxi Details
3.Exit
Enter your Choice :  2
Taxi-1 Total Earnings: Rs.500
BookingID  CustomerID From  To    PickupTime   DropTime  Amount
1          1          C     F     10           13        500
Taxi-2 Total Earnings: Rs.350
BookingID  CustomerID From  To    PickupTime   DropTime  Amount
1          2          C     E     11           13        350

1.Book taxi
2.Display Taxi Details
3.Exit
Enter your Choice :  
```

FIG 6.2 OUTPUT FOR TAXI BOOKING SYSTEM

In Fig 6.2, the Taxi Booking System demonstrates the functionality of adding, viewing, deleting, and managing taxi bookings in a structured and interactive manner. The system starts by displaying a menu with five options: Book Taxi, View Bookings, Update Booking, Cancel Booking, and Exit. The user selects option 4 (Cancel Booking), which prompts them to enter the booking ID they wish to cancel. In this instance, the user inputs the booking ID "102," and the system confirms the successful cancellation of the taxi booking with the message "Booking canceled successfully!" This indicates that the system correctly locates and removes the specified booking record from storage, ensuring that outdated or unnecessary bookings are efficiently managed. Following this cancellation, the user selects option 2 (View Bookings) to verify the remaining taxi bookings. The system retrieves and displays the stored booking information, showing only one remaining booking with ID "101," passenger name "John Doe," pickup location "Downtown," destination "Airport," and time "10:30 AM." This output confirms that the cancellation operation was performed correctly, as the previously stored booking with ID "102" is no longer present in the system. This demonstrates that the Taxi Booking System is reliable in maintaining accurate records and updating the database in real-time.

After viewing the taxi booking records, the user interacts with the system again by selecting an invalid option, choice "6." Since the system only provides five options, it detects the invalid input and returns an error message stating, "Invalid choice. Please try again." This validation mechanism is essential for preventing unexpected errors and ensuring that users interact with the system correctly. By prompting users to re-enter a valid choice, the system enhances usability and minimizes potential mistakes. Finally, the user selects option 5 (Exit), and the system displays the message "Exiting system," indicating that the program has successfully terminated. The console confirms the termination with "Program finished with exit code 0," signifying that the program has executed without any errors or crashes. This smooth exit process ensures that users can safely close the application when they are finished managing taxi bookings.

CHAPTER 7

CONCLUSION

The Taxi Booking System is a highly functional and efficient software solution designed to streamline the management of taxi bookings within a transportation service. Through its various modules, including booking a taxi, viewing bookings, updating details, and canceling bookings, the system ensures that taxi services can maintain an organized and up-to-date booking database. The system's user-friendly interface, clear menu options, and interactive features allow users to perform essential operations with ease, making it a valuable tool for taxi service operators, drivers, and customers. One of the key strengths of this system is its ability to manage taxi bookings effectively. The addition of new bookings is seamless, requiring the user to enter details such as booking ID, passenger name, pickup location, destination, and time of travel. This ensures that every booking is stored systematically for easy retrieval. The "View Bookings" feature allows users to quickly access all stored booking records, which is essential for maintaining an updated database and ensuring that the right vehicles are dispatched at the correct times. This feature plays a crucial role in taxi service operations, as operators can verify booking details to optimize their fleet’s schedule and ensure timely services.

The update functionality ensures that booking records remain accurate by allowing users to modify existing details whenever necessary. This is especially useful in scenarios where a passenger's pickup location, destination, or time changes, preventing outdated or incorrect information from being retained in the system. The cancel functionality further enhances the efficiency of the system by allowing users to remove bookings that are no longer needed. This ensures that taxi services are not overbooked and that only active bookings are listed, which is particularly important for managing fleet capacity in real-time. Another significant aspect of the Taxi Booking System is its error-handling capabilities. The system effectively validates user input to prevent invalid choices, as seen in the scenario where an incorrect menu option was entered. By displaying appropriate error messages and prompting the user to enter a valid choice, the system ensures smooth operation and prevents potential system crashes. This level of validation enhances user experience and ensures that the application runs reliably without unexpected interruptions.

REFERENCES

- S. A. Rajagukguk, H. Prabowo, A. Bandur, and R. Setiowati, "Taxi Booking System with Real-Time Location Tracking and Route Optimization," IEEE Access, vol. 11, pp. 77779–77792, 2023, https://ieeexplore.ieee.org/document/10193752.

- Y. Gao, Y. Yang, S. Wang, W. Zhang, and J. Lu, "Has China's Taxi Booking System Improved Public Transportation Efficiency?" Transportation Research Part A, vol. 14, no. 54, Jul. 2024, https://link.springer.com/article/10.1186/s13561-024-00520-8.

- C. Visvesvaran, R. Shri Yazhini, T. Sneka, and K. Swetha, "Dynamic Pricing Algorithm for Efficient Taxi Booking System," 2022 International Conference on Edge Computing and Applications (ICECAA), Tiruchirappalli, India, 2022, https://ieeexplore.ieee.org/document/10395877.

- N. Jankelová, M. Dabić, J. F. Maley, Z. Joniaková, and I. Némethová, "Driver Retention Strategies: Integrating Technology and Customer Satisfaction in Taxi Booking Systems," Journal of the Knowledge Economy, vol. 14, no. 4, pp. 859–880, Nov. 2024, https://link.springer.com/article/10.1007/s13132-024-02316-x.

- A. C. Babu, V. N. C. S. Teja, A. D. Reddy, E. N. Kumar, and V. Srinivas, "Web-Based Taxi Booking System with Automated Dispatch and Dynamic Pricing," 2023 International Conference on Artificial Intelligence and Machine Vision (AIMV), Warangal, India, 2023, https://ieeexplore.ieee.org/document/10112962.

- L. Zhang, P. Zhang, and W. Chen, "Impact of Real-Time Data on Taxi Service Utilization: A Study on Consumer Behavior," BMC Transportation Research, vol. 24, no. 454, Apr. 2024, https://link.springer.com/article/10.1186/s12913-024-10903-6.

- Y. Huang, "Optimizing Taxi Fleet Management System Using Genetic Algorithms," 2022 3rd International Conference on E-Commerce and Internet Technology (ECIT), Zhangjiajie, China, 2022, pp. 1–5, https://ieeexplore.ieee.org/document/9943162.

- X. Tong, H. Zou, L. Zhang, and W. Chen, "Impact of Taxi Booking System on Ride Efficiency: A Big Data Approach," BMC Public Transportation, vol. 25, art. no. 404, Jan. 2025, https://link.springer.com/article/10.1186/s12889-025-21656-0.

- S. Hemkiran, M. M. War, K. S. Aadhithiyan, and K. Kabilan, "Taxi Booking System with Intelligent Route Prediction and Passenger Matching Using Machine Learning," IEEE, 2025, https://ieeexplore.ieee.org/document/10649525.