

**SCHOOL OF BUSINESS AND ECONOMICS**

**DEPARTMENT OF BUSINESS TECHNOLOGY**

**SYSTEM ENGINEERING**

**Project Name:**

**AMBULANCE BOOKING SYSTEM**

**Student name: HABIYAREMYE LAURIER**

**Reg No: 222003068**

**Submitted to: Dr. BUGINGO Emmanuel**

**Date: xxx/ xxx 2023**

# Table of contents

Contents

[Table of contents 2](#_Toc140454198)

**CHAPTER 1: SYSTEM ENGENEERING……………………………………………………………………………………2**

[**1.1 Introduction 4**](#_Toc140454199)

**1.**[**2. Statement of the problem 5**](#_Toc140454200)

[**2.1 *Description of the existing system* 5**](#_Toc140454201)

[**2.*2 Problems of the current system* 5**](#_Toc140454203)

[***2.3 How the proposed system will work* 6**](#_Toc140454204)

[**2.4 Describe other alternatives 6**](#_Toc140454208)

[**3. Software requirement 7**](#_Toc140454209)

[**3.1 User Requirements 7**](#_Toc140454210)

[***3.1.1 Functional requirement* 7**](#_Toc140454211)

[**3.1.2 Non-function requirement 8**](#_Toc140454212)

[**Portability requirement 9**](#_Toc140454213)

[**Reliability requirement 10**](#_Toc140454214)

[**Usability requirement 11**](#_Toc140454215)

[**Space requirement 12**](#_Toc140454216)

[**Organization requirement 12**](#_Toc140454217)

[**Implementation requirement 14**](#_Toc140454218)

[**External environment requirement 15**](#_Toc140454219)

[**Privacy requirement 16**](#_Toc140454220)

[**Safety requirement 18**](#_Toc140454221)

[**3.2 System requirements 19**](#_Toc140454222)

[**3.2.1 Minimum End-user Hardware Requirements 19**](#_Toc140454223)

[**3.2.2 Minimum End-user Software Requirements 20**](#_Toc140454224)

[**3.3 Software specifications 21**](#_Toc140454225)

[**4. DATA FLOW DIAGRAM (LEVEL 0, LEVEL 1) 23**](#_Toc140454226)

[**4.1 DFD LEVEL 0 23**](#_Toc140454227)

[**4.2 DFD LEVEL 1 24**](#_Toc140454228)

[**5. Feasibility study 25**](#_Toc140454229)

[**5.1 Technical Feasibility 25**](#_Toc140454230)

[**5.2 Financial Feasibility 26**](#_Toc140454231)

[**5.3 Market Feasibility 26**](#_Toc140454232)

[**5.4 Economic Feasibility 26**](#_Toc140454233)

[**5.5 Legal and Regulatory Feasibility 26**](#_Toc140454234)

[**5.6 Operational Feasibility 27**](#_Toc140454235)

[**6. Description of the selected Process Model 27**](#_Toc140454236)

[**6.1 Cause of your selection 28**](#_Toc140454237)

[**7. Conclusion 29**](#_Toc140454238)

[**8. References 30**](#_Toc140454239)

**CHAPTER 2 : DATABASE TECHNOLOGY………………………………………………………………………….30**

**2.1 INTRODUCTION ………………………………………………………………………31**

**2.2 SECTION1: ... ………………………………………………………………………….31**

**2.2.1 ENTITIES ... ………………………………………………………………………………………….31**

**2.2.2 LOGICAL DATABASE MODEL …………………………………………………………………31**

**2.2.2ENTITY RELETIONSHIP DIAGRAM ……………………………………………………………31**

**2.3 SECTION2:SQL ….……………………………………………………………………31**

**2.4 SECTION3 ………………………………………………………………………………31**

**2.4.1 VIEWS, TRIGGERS AND PROCEDURES ………………………………………………….31**

**2.4 CONCLUSION ……………………………………………………………………...........34**

**CHAPTER3: JAVA PROGRAMMING ....................................................................................40**

**3.1 INTRODUCTION ………………………………………………………………………...40**

**3.2 TOOLS USED TO DEVELOP THIS SYSTEM IN JAVA PROGRAMMING …………40**

**3.3 FORMS DESCRIPTION …………………………………………………………………41**

**3.4 SPECIAL BUTTONS ...……………………………………………………………………52**

**3.5 CONCLUSION …………………………………………………………………………....5**

# **Introduction**

In today’s traffic world, ambulance plays a major role when an accident occurs on the road network and the need arises to save valuable human life. Transportation of a patient to an emergency hospital seems quite simple but in actuality, it is quite difficult and gets more difficult during peak hours.

In our Ambulance Booking System, people can easily book an ambulance. There are three major modules namely User, Ambulance, and Hospital. Users can register and log in using credentials. Users can edit their profile and change their password in an emergency. Any Upcoming Ambulance Booking details if anyone wants to Book an Ambulance or if there is an Emergency.

For booking an ambulance users have to select ambulance size, pick-up point & hospital, and date & time. In an emergency will automatically book the nearest ambulance & hospital. Users will get a list of All the bookings of Ambulances. The front-end involves Html, CSS, and JavaScript and the back-end involves Python. The framework used is Django and the database is MySQL.

# **Statement of the problem**

## 2.1 ***Description of the existing system***

## In today’s traffic world, ambulance plays a major role when an accident occurs on the road network and the need arises to save valuable human life. Transportation of a patient to an emergency hospital seems quite simple but in actuality, it is quite difficult and gets more difficult during peak hours.

In now days to get to an Ambulance, you have to dial 112as a tool free and then you wait for an ambulance to come. At that time you have no idea of where it is going to take you and the you wake up at the different hospital of your choice.

## 2.***2 Problems of the current system***

Transportation of a patient to an emergency hospital seems quite simple but in actuality, it is quite difficult and gets more difficult during peak hours,The challenges ranges from poor communication, poor road network and unstructured address to non-arrival of ambulances leading loss of lives that are preventable.

There are also some problems like :

Delay in ambulance responses

The delay of Ambulance drivers to get the full information

Loss of the patient’s information due to the manual management of information

## ***2.3 How the proposed system will work***

# In this system, there are four entities User, Ambulance and Hospital. The user must register and log in using a username and password. After logging in, the user can Book Ambulance, Book Hospital, View Nearby Hospitals, View Previous Booked Ambulances and Hospitals, and it can also change its password.

# When the user books an ambulance and hospital, a booking request is sent to the respective representatives of the ambulance and hospital. In view, Nearby Hospitals the user can view the nearest hospitals in their location. The ambulance driver has to register and then login in using a username and password.

# After logging in, the driver can view booking requests, nearby hospitals and their previous bookings i.e., previously accepted requests. In Booking requests, it can either accept or decline the user requests. The hospital has to register and log in using a username and password. After login in the hospital representative can view the booking request and either accept or decline the user request.

## **Describe other alternatives**

Third-Party Ambulance Booking Platforms:

There are third-party platforms and mobile applications that provide ambulance booking services. These platforms connect users with nearby ambulance services, allowing them to request emergency or non-emergency medical transportation.

Examples include platforms like "Uber Health" or "Zocdoc" that offer ambulance booking services as part of their healthcare offerings.

Custom-Built Solutions:

Organizations or healthcare providers may develop their own custom ambulance booking systems tailored to their specific requirements.

This option allows for full control over the system's features, integration with existing infrastructure, and customization based on unique needs.

Integrated Healthcare Management Systems:

Integrated healthcare management systems, such as hospital information systems or electronic medical record (EMR) systems, often include ambulance booking functionalities as part of their overall suite of services.

These systems offer a comprehensive solution for managing patient records, appointments, and medical transportation within a healthcare facility.

## **Software requirement**

## 3.**1 User Requirements**

### ***3.1.1 Functional requirement***

Certainly! Here are some functional requirements for an ambulance booking system project:

**User Registration and Authentication:** Users should be able to create an account and provide the necessary information for registration.The system should verify user credentials and authenticate users during login.

**Ambulance Availability and Search:** The system should provide a search functionality to allow users to check the availability of ambulances based on location, date, and time.Ambulance availability should be displayed in real-time, indicating the type of ambulance and its current location.

**Booking Request Management:** Users should be able to submit a booking request by providing the required details such as the pickup location, destination, date, and time.The system should validate and store the booking request in a database.

**Emergency Priority Handling:** The system should have a mechanism to identify and prioritize emergency cases over non-emergency cases.Emergency requests should be flagged and given higher priority in the dispatching process.

**Dispatching and Routing:** Once a booking request is received, the system should assign an available ambulance based on proximity, type, and availability.The system should generate an optimized route for the ambulance to reach the pickup location and then the destination.

**Real-time Tracking:** Users should be able to track the assigned ambulance in real-time using GPS or similar technologies.The system should provide updates on the estimated time of arrival (ETA) and any delays during transit.

**Communication and Notifications:** The system should facilitate communication between users and ambulance staff, such as through in-app messaging or SMS notifications.Users should receive notifications at various stages of the booking process, including confirmation, ambulance assignment, and ETA updates.

**Cancellation and Rescheduling:** Users should be able to cancel or reschedule their booking requests within a specified time frame.The system should handle cancellations and rescheduling requests, update the status accordingly, and potentially assign the ambulance to another request if necessary.

**Reporting and Analytics:** The system should generate reports and analytics to track key performance indicators (KPIs) such as response times, booking trends, and user feedback. This data can be used for performance evaluation, process improvement, and resource allocation.

**Administrator Dashboard:** The system should provide an administrative interface to manage and monitor the ambulance booking system.Administrators should have the ability to add or remove ambulances, manage user accounts, view reports, and perform system configuration tasks.

### **3.1.2 Non-function requirement**

Certainly! Here are some non-functional requirements for an ambulance booking system project:

**Performance:** The system should be able to handle a high volume of concurrent users and booking requests without significant performance degradation. Response times for user interactions, such as searching for ambulances or making a booking, should be within an acceptable range.

**Reliability:** The system should be highly reliable, ensuring that it is available and functional at all times.It should have mechanisms in place to handle system failures, such as backup servers or redundant components, to minimize downtime.

**Security:** The system should have robust security measures to protect user data, including personal information and medical details. User authentication and access control should be implemented to ensure that only authorized individuals can access sensitive information.

**Scalability:** The system should be designed to accommodate future growth and increasing user demands.It should be scalable both horizontally (adding more servers or instances) and vertically (increasing hardware resources) as needed.

**Usability:** The user interface should be intuitive, user-friendly, and accessible to users with varying levels of technical proficiency.The system should provide clear instructions and error messages to guide users through the booking process.

**Availability:** The system should have a high level of availability, minimizing scheduled maintenance windows and unplanned downtime.Adequate backup and disaster recovery measures should be in place to ensure data integrity and system availability.

**Compliance:** The system should comply with relevant regulations and standards, such as data protection laws and healthcare industry guidelines.It should adhere to privacy requirements and obtain necessary user consent for data collection and processing.

**Integration:** The system should be able to integrate with external services or systems, such as payment gateways for online payments or hospital information systems for patient data exchange.Integration should be seamless, ensuring smooth data flow and functionality between the ambulance booking system and other systems.

**Performance Monitoring and Logging:** The system should have mechanisms to monitor performance metrics, log system activities, and generate relevant logs for troubleshooting and analysis.This information can be used to identify and resolve issues promptly, optimize system performance, and ensure effective system monitoring.

**Maintenance and Support:** The system should have a well-defined maintenance and support plan in place.Regular updates, bug fixes, and security patches should be provided to address system issues and vulnerabilities.Adequate technical support should be available to assist users and address any system-related queries or concerns.

### **Portability requirement**

**Platform Independence:** The system should be developed using platform-independent technologies to ensure compatibility across various operating systems (e.g., Windows, macOS, Linux) and devices (e.g., desktops, laptops, tablets, smartphones).

**Web-Based Interface:** The system should have a web-based interface that can be accessed through standard web browsers, eliminating the need for platform-specific software installations.The interface should be responsive and adapt to different screen sizes and resolutions for optimal user experience on various devices.

**Mobile Application:** In addition to the web-based interface, the system can have a dedicated mobile application compatible with popular mobile operating systems (e.g., iOS, Android).The mobile application should provide similar functionality as the web interface and offer a seamless user experience on smartphones and tablets.

**Cross-Browser Compatibility:** The system should be compatible with major web browsers, such as Google Chrome, Mozilla Firefox, Microsoft Edge, and Safari.It should be tested on different browsers to ensure consistent behavior and functionality across them.

**Data Interoperability:** The system should support standard data formats and protocols to facilitate interoperability with other healthcare systems, such as hospital information systems or electronic medical record systems.It should be capable of exchanging data seamlessly and securely with other systems using standardized formats (e.g., HL7) or industry-specific APIs.

**Cloud Deployment:** The system can be designed for cloud deployment to enhance portability and scalability. It should be compatible with popular cloud platforms (e.g., Amazon Web Services, Microsoft Azure, Google Cloud Platform) to enable easy deployment and management in a cloud environment.

**Database Independence:** The system should support multiple database management systems (DBMS) to provide flexibility in choosing the database backend.It should be compatible with widely used DBMS, such as MySQL, PostgreSQL, Oracle, or Microsoft SQL Server.

**Third-Party Integration:** The system should allow for seamless integration with third-party services and APIs, such as payment gateways, mapping services (e.g., Google Maps), or SMS gateways.The integration should be platform-agnostic, enabling interoperability with different providers and services.

**Configuration and Customization:** The system should have configurable parameters and settings to accommodate varying deployment environments and user preferences.It should support customization options, such as branding, themes, or language localization, to meet the specific requirements of different deployments.

**Backup and Restore:** The system should have mechanisms in place to easily backup and restore data, ensuring data portability and recovery in case of system failures or migration to new environments.

These portability requirements ensure that the ambulance booking system can be deployed across multiple platforms, devices, and environments, allowing users to access the system conveniently from their preferred devices and operating systems.

### **Reliability requirement**

Here are some key considerations for reliability requirements in our ambulance booking system:

**System Availability:** The system should be available and accessible to users whenever they need to book an ambulance. It should have a high uptime and minimal downtime for maintenance or upgrades.

**Fault Tolerance:** The system will be designed to handle and recover from failures, errors, or disruptions. It will have mechanisms in place to detect faults, switch to backup systems if necessary, and minimize service disruptions.

**Data Integrity:** The system will ensure the integrity of the data being processed. This includes accurate storage, retrieval, and transmission of user information, booking details, and feedback. Measures such as data validation, encryption, and backups should be implemented to maintain data integrity.

**Performance Stability:** The system will consistently deliver the expected performance levels, regardless of the user load or system usage patterns. It should be capable of handling peak demand periods without significant degradation in response times or service quality.

**Error Handling and Logging:** The system will have robust error handling mechanisms in place to gracefully handle errors and exceptions. Additionally, it should log relevant error information to aid in troubleshooting, system improvements, and auditing purposes.

**Backup and Recovery:** The system will have appropriate backup and recovery mechanisms to ensure that critical data is protected and can be restored in the event of data loss or system failures. Regular backups, disaster recovery plans, and redundant infrastructure can contribute to system reliability.

**Monitoring and Proactive Maintenance:** The system will be continuously monitored to detect any potential issues or performance degradation. This includes monitoring server health, network connectivity, and other relevant system components. Proactive maintenance and timely system updates should be performed to address potential vulnerabilities and ensure system reliability.

**Scalability:** The system always be able to scale efficiently to accommodate increasing user demand. As the user base grows, the system should be capable of handling additional users, bookings, and interactions without compromising reliability or performance.

### **Usability requirement**

**User-Friendly Interface:** The system will have a clean and intuitive user interface that is easy to understand and navigate. Users should be able to perform booking-related tasks, enter information, and receive feedback without confusion or difficulty.

**Simple Registration and Login:** The registration and login process will always be straightforward and require minimal steps. Users should be able to create an account easily and access the system without unnecessary barriers.

**Clear Booking Process:** The process of booking an ambulance will be streamlined and easy to follow. Users should be guided through each step, and the required information should be clearly indicated. Clear instructions and error messages should be provided when necessary.

**Intuitive Location Selection:** The system will be able to provide a user-friendly way for users to specify their location. This can be achieved through various methods such as address search, map selection, or geolocation.

**Responsive Design:** The system will be designed to be responsive and compatible with different devices and screen sizes, including mobile devices and tablets. It should adapt to different resolutions and orientations to ensure a consistent user experience across platforms.

**Efficient Search Functionality:** The system should offer efficient search capabilities to allow users to find relevant information quickly. This includes the ability to search for available ambulances based on location, availability, and other relevant criteria.

**Clear Feedback and Confirmation:** The system will provide clear feedback and confirmation messages to users after they complete a booking or perform any other important action. Users should receive timely notifications about their bookings, including confirmation messages and estimated arrival times.

**Error Prevention and Handling:** The system will incorporate measures to prevent errors and guide users to correct any mistakes they make during the booking process. Clear error messages should be displayed when necessary, providing guidance on how to rectify the issues.

**Personalization and Preferences**: The system must allow users to personalize their experience by providing options to set preferences such as preferred ambulance type, preferred language, and notification settings.

**Accessibility:** The system will adhere to accessibility guidelines, ensuring that users with disabilities or impairments can access and use the system effectively. This includes providing alternative text for images, keyboard navigation support, and compatibility with screen readers.

### **Space requirement**

It is challenging to provide specific space requirements without detailed information about the expected user base and data volume. The server infrastructure and storage requirements should be estimated based on factors such as the expected number of users, booking frequency, data retention policies, and growth projections.

To accurately determine the space requirements, it is recommended to conduct a capacity planning exercise that involves analyzing historical data (if available), making informed assumptions about user behavior and system usage, and performing load testing to simulate different scenarios. This process can help identify the optimal server infrastructure and storage capacity needed for the ambulance booking system.

### **Organization requirement**

Organizational requirements for an ambulance booking system project encompass the necessary considerations related to the organizational structure, roles and responsibilities, and coordination among stakeholders. Here are some organizational requirements to consider:

**Project Team:** Establish a project team consisting of individuals with relevant expertise, including project managers, software developers, designers, testers, and system administrators.Define clear roles and responsibilities for each team member and ensure effective communication and collaboration throughout the project.

**Stakeholder Engagement:** Identify key stakeholders, such as ambulance service providers, hospitals, emergency response agencies, and regulatory authorities.Engage stakeholders early in the project to understand their requirements, gather feedback, and ensure their active involvement in the system development and implementation.

**Project Management:** Assign a project manager responsible for overseeing the entire project, ensuring adherence to timelines, managing resources, and coordinating with stakeholders.Establish a project management framework to track progress, monitor risks, and facilitate effective decision-making.

**Governance:** Establish a governance structure to provide oversight and decision-making authority for the project.Define roles and responsibilities of governance entities, such as steering committees or project boards, to ensure alignment with organizational objectives and strategic priorities.

**Communication and Collaboration:** Establish effective channels of communication within the project team and with stakeholders.Implement collaboration tools and platforms to facilitate seamless communication, document sharing, and version control.

**Change Management:** Develop a change management strategy to address the organizational impact of implementing the ambulance booking system.Identify potential resistance to change and plan appropriate strategies for training, communication, and support to ensure smooth adoption of the system.

**Training and Documentation:** Provide training programs to users, administrators, and support staff to familiarize them with the ambulance booking system.Develop comprehensive documentation, including user manuals, system administration guides, and troubleshooting resources, to support ongoing system usage and maintenance.

**Legal and Regulatory Compliance:** Ensure that the ambulance booking system complies with relevant legal and regulatory requirements, including data privacy laws, healthcare industry regulations, and emergency response guidelines.Establish mechanisms to address compliance-related concerns, such as data protection, consent management, and audit trails.

**Support and Maintenance:** Define support and maintenance processes, including service level agreements (SLAs) and escalation procedures, to address user queries, system issues, and ongoing system enhancements.Assign dedicated support personnel to provide timely assistance and ensure system availability and reliability.

**Continuous Improvement:** Foster a culture of continuous improvement by soliciting user feedback, analyzing system performance, and identifying areas for enhancement.Establish mechanisms for gathering and incorporating user suggestions and requirements for future system updates and iterations.

These organizational requirements are crucial for effective project management, stakeholder engagement, and successful implementation of the ambulance booking system within the organizational context.

### **Implementation requirement**

The implementation requirements for this project involve the technical aspects and processes needed to deploy and integrate the system effectively. Here are some implementation requirements to consider:

**System Architecture:** Define the system architecture, including the choice of technology stack, database management system, and web server infrastructure.Determine the required hardware and software components for the system's implementation.

**Development Environment:** Set up a development environment with appropriate tools and frameworks for coding, testing, and debugging.Provide access to version control systems, project management tools, and collaboration platforms for efficient development and coordination.

**System Configuration:** Configure the necessary system parameters, such as server settings, database configurations, and security configurations, to ensure optimal system performance and security.

**Database Design and Setup:** Design and create the database schema to store system data efficiently.Set up the database management system and establish data connection and access controls.

**Application Development:** Develop the ambulance booking system application, following the defined system requirements and design specifications.Implement the required functionality, including user registration, ambulance availability search, booking request management, and real-time tracking.

**User Interface Design:** Design an intuitive and user-friendly interface for the ambulance booking system.Ensure responsive design to support multiple devices and screen sizes.

**Integration:** Integrate the ambulance booking system with external services and systems, such as payment gateways, mapping services, or SMS gateways.Establish secure connections and implement data exchange protocols as per integration requirements.

**Testing and Quality Assurance:** Conduct thorough testing to ensure the system functions as intended, including unit testing, integration testing, and system testing.Implement quality assurance processes to identify and fix bugs, optimize system performance, and validate system requirements.

**Data Migration:** If applicable, plan and execute the migration of existing data from legacy systems or previous versions of the ambulance booking system to the new system.Ensure data integrity and accuracy during the migration process.

**Deployment and Rollout:** Prepare the system for deployment, including configuring production servers, setting up security measures, and establishing system monitoring and logging mechanisms.Develop a deployment plan, considering factors such as downtime minimization, data backup, and rollback procedures.Roll out the system in a controlled manner, ensuring proper user access and support channels.

**Training and User Adoption:** Provide training sessions to users and administrators to familiarize them with the ambulance booking system's features and functionalities.Offer documentation and user guides to support users in understanding and effectively using the system.

**Post-Implementation Support:** Establish mechanisms for ongoing system support and maintenance, including bug fixing, performance optimization, and regular updates.Implement monitoring and logging tools to proactively identify and address system issues.

These implementation requirements form the technical foundation for successfully deploying the ambulance booking system and ensuring its functionality, usability, and reliability in a production environment.

### **External environment requirement**

The external environment requirements for an ambulance booking system project involve the factors and dependencies outside the system itself that can impact its operation and effectiveness. Here are some external environment requirements to consider:

**Internet Connectivity:** The system requires a stable and reliable internet connection to ensure seamless communication between users and the system, as well as integration with external services.Users accessing the system remotely or in areas with limited connectivity may require alternative means of access or offline functionality.

**Geographical Coverage:** The system should be designed to accommodate the geographical coverage area where ambulance services are available.Consider the specific regions, cities, or areas where the ambulance booking system will be deployed, ensuring that the system supports the corresponding ambulance service providers.

**Mapping and Geolocation Services:** Integration with mapping and geolocation services, such as Google Maps or other similar platforms, can provide accurate location information for ambulances and facilitate efficient routing.Ensure compatibility and access to reliable mapping services for the system to accurately display ambulance availability and track their movements.

**Emergency Response Agencies:** Collaborate with local emergency response agencies to align the ambulance booking system with their protocols and processes.Obtain necessary permissions, approvals, or certifications from relevant agencies to ensure compliance and effective coordination during emergency situations.

**Telecommunications Infrastructure:** The system relies on telecommunication infrastructure, such as mobile networks or landline connections, to facilitate communication between users and emergency service providers. So, we have to Consider the availability and reliability of telecommunication services in the deployment area to ensure uninterrupted communication and response.

**Legal and Regulatory Requirements:** Comply with legal and regulatory requirements specific to the healthcare industry, emergency services, and data privacy. We will Ensure adherence to local, regional, and national laws related to healthcare information management, patient confidentiality, and emergency response protocols.Payment Gateway Integration:Because our system involves online payment for ambulance services, we have to integrate with reliable and secure payment gateways that comply with applicable financial regulations. So, we will Consider the specific payment options and mechanisms preferred by users in the deployment area.

**Language and Localization:** Depending on the target user base, our system will provide language localization options to support multilingual users. We will Localize the system's user interface, notifications, and communication to match the preferred language(s) of the deployment area.

**External APIs and Services:** The system willIdentify and integrate with external APIs or services that enhance the functionality and usability of the ambulance booking system. This may include SMS gateways for emergency notifications, weather APIs for real-time weather updates, or healthcare provider APIs for seamless data exchange.

**Regulatory Reporting:** Determining if the system needs to generate regulatory reports or provide data for statistical analysis or compliance purposes.Ensuring if the system can generate the required reports and support any data submission processes mandated by regulatory authorities.

Considering and addressing these external environment requirements will help ensure that the ambulance booking system is well-aligned with the surrounding ecosystem, regulatory landscape, and infrastructure necessary for its effective operation and integration.

### **Privacy requirement**

Privacy requirements for an ambulance booking system project are crucial to protect sensitive user information and comply with data privacy regulations. Here are some privacy requirements to consider:

**Data Confidentiality:** The system willEnsure that all user data, including personal information and medical details, is treated as confidential and securely stored.The system will Implement appropriate access controls and encryption mechanisms to safeguard user data from unauthorized access.

**Consent Management:** Obtaining explicit consent from users to collect, process, and store their personal information and medical data. The system will Provide clear and transparent consent mechanisms, such as consent checkboxes or consent forms, and allow users to easily withdraw their consent if desired.

**Data Minimization:** The system will Collect and store only the minimum necessary data required for the ambulance booking system's operation.The system will Avoid storing sensitive or unnecessary information to minimize the potential risks associated with data breaches or unauthorized access.

**Anonymization and Pseudonymization:** The system will Implement anonymization techniques to remove or irreversibly de-identify personal information from user data, whenever possible.Use pseudonymization techniques, such as assigning unique identifiers or tokens, to replace direct user identifiers, ensuring that data cannot be easily attributed to specific individuals.

**User Access Controls:** The system will Implement role-based access controls to limit access to user data to authorized personnel only. The system will Differentiate access privileges based on user roles, ensuring that healthcare professionals and administrators can access relevant data only on a need-to-know basis.

**Secure Communication:** The system will Employ secure communication protocols (e.g., HTTPS) to protect data transmission between users and the ambulance booking system.The system will Use encryption mechanisms to safeguard sensitive data during transmission, preventing unauthorized interception or tampering.

**Audit Logs and Monitoring:** The system will Implement logging mechanisms to record user activities, system access, and data modifications.The system will Regularly monitor and review audit logs to identify and investigate any potential privacy breaches or suspicious activities.

**Data Retention and Disposal:** The system will Define data retention policies that outline the duration for which user data will be stored and specify the procedures for data disposal once it is no longer required. The system will Ensure that data disposal processes comply with applicable regulations and industry best practices, such as secure deletion or anonymization of data.

**Third-Party Data Handling:** If the ambulance booking system shares data with third-party services or partners, the system will ensure that appropriate data protection agreements are in place.The system will Conduct due diligence to assess the privacy practices and data security measures of third parties involved in the system's ecosystem.

**Compliance with Data Protection Regulations:** The system will deal with relevant data protection and privacy regulations, such as the General Data Protection Regulation (GDPR) or other local privacy laws.The specific requirements and obligations imposed by these regulations are incorporated into the system's design and operations.

**User Notification and Transparency:** The system will Inform users about the data collected, how it will be used, and the privacy practices implemented by the ambulance booking system.The system will Provide users with clear and easily accessible privacy policies, terms of service, and data handling practices, ensuring transparency regarding data usage and sharing.

These privacy requirements aim to protect user privacy, maintain the confidentiality of personal and medical data, and ensure compliance with relevant data protection regulations. By incorporating these requirements into the ambulance booking system, you can establish a privacy-centric approach that fosters user trust and confidence in the system.

### **Safety requirement**

Safety requirements for an ambulance booking system project are essential to ensure the safety of users, emergency response personnel, and the general public. Here are some safety requirements to consider:

**System Reliability:** The ambulance booking system should be designed and implemented to operate reliably, minimizing system downtime and ensuring continuous availability.Implement redundancy measures, fault tolerance mechanisms, and backup systems to mitigate the risk of system failures and ensure uninterrupted service.

**User Authentication and Authorization:** We will Implement secure authentication and authorization mechanisms to verify the identity of users and prevent unauthorized access to the system.The system will Ensure that only authorized users, such as medical professionals or emergency response personnel, can access sensitive functionalities or perform critical actions.

**Emergency Response Priority:** Incorporate features and algorithms that prioritize emergency requests over non-emergency bookings.Implement mechanisms to expedite the allocation of ambulances for emergency situations, considering factors such as distance, severity, and availability.

**Data Security:** The system will Implement data security measures to protect user information and prevent unauthorized access, tampering, or disclosure.The system will use encryption techniques, secure data storage, and secure communication protocols to safeguard sensitive data.

**System Monitoring and Alarms:** The system will Implement real-time monitoring mechanisms to detect system failures, anomalies, or performance issues.Set up alarms or alerts to notify administrators or support personnel about critical system events or abnormal behavior that may affect safety or system integrity.

**Data Backup and Disaster Recovery:** In this system we will Regularly back up system data to ensure its availability and integrity in the event of data loss or system failures. The system will Establish a disaster recovery plan with backup procedures, backup storage, and restoration processes to recover the system and data in case of unforeseen incidents.

**User Training and Support:** The system will Provide comprehensive training programs to users, administrators, and support staff to ensure they are proficient in using the ambulance booking system and are aware of safety protocols and emergency response procedures.We will Offer user support channels to address any concerns or issues promptly, ensuring users can access assistance during critical situations.

**System Performance and Scalability:** The team willEnsure that the system can handle the expected user load and booking volumes without significant performance degradation.There will be scalability measures, such as load balancing, caching, or cloud infrastructure, to accommodate increased usage or sudden spikes in demand.

**Compliance with Safety Standards:** The system will Comply with applicable safety standards and regulations specific to healthcare systems, emergency response protocols, and data security.We will Familiarize with industry best practices and regulatory requirements, such as HIPAA (Health Insurance Portability and Accountability Act) or local safety guidelines, and incorporate them into the system's design and operations.

**User Safety Guidelines:** The mechanism will Provide clear guidelines and instructions to users on how to use the ambulance booking system safely and responsibly. The team will Educate users about potential risks and safety precautions, such as providing accurate location information, following instructions from emergency personnel, and using the system responsibly during emergencies.

By incorporating these safety requirements into the ambulance booking system, We can ensure that the safety of users, emergency response operations, and the overall community. Regular monitoring, system resilience, and adherence to safety standards will contribute to a reliable and secure system that supports efficient and safe ambulance bookings.

## **3.2 System requirements**

### **3.2.1 Minimum End-user Hardware Requirements**

The minimum end-user hardware requirements for an ambulance booking system project may vary depending on the type of platform (web or mobile) and the complexity of the application. Here are some minimum hardware requirements for both web and mobile platforms:

***Minimum End-User Hardware Requirements for Web Platform:***

**Desktop or Laptop:** Operating System: Windows, macOS, or Linux.

Processor: Intel Core i3 or equivalent.

RAM: 4 GB or more.

Storage: At least 100 MB of free disk space.

Web Browser: Latest version of popular browsers such as Google Chrome, Mozilla Firefox, Safari, or Microsoft Edge.

**Internet Connection:** Stable and reliable internet connection with a minimum speed of 1 Mbps for smooth browsing and data exchange.

***Minimum End-User Hardware Requirements for Mobile Platform:***

**Smartphone or Tablet:** Operating System: Android or iOS.

Processor: Dual-core or higher.

RAM: 2 GB or more.

Storage: At least 100 MB of free storage space.

Screen Size: Preferably 4.7 inches or larger for better user experience.

Screen Resolution: At least 720p for clarity in displaying app content.

**Internet Connection:** Stable and reliable internet connection with 3G, 4G, or Wi-Fi connectivity for real-time data exchange.

With our system **(AMBULANCE BOOKING SYSTEM),** we will take care of these important minimum hardware requirements, and for optimal performance and user experience, it is recommended to have better hardware specifications. Additionally, the ambulance booking system's development will also focus on responsive design to ensure that the application can adapt to different screen sizes and devices, accommodating a broader range of hardware configurations.

### **3.2.2 Minimum End-user Software Requirements**

***Minimum End-User Software Requirements for Web Platform:***

**Operating System:** The latest version of Windows, macOS, or Linux operating system.

**Web Browser:** The latest version of popular web browsers such as Google Chrome, Mozilla Firefox, Safari, or Microsoft Edge.Ensure that the browser supports modern web technologies, HTML5, CSS3, and JavaScript.

**Browser Plugins:** Ensure that the necessary plugins, such as Adobe Flash Player or Java Runtime Environment (JRE), are up to date if the system requires their use.

***Minimum End-User Software Requirements for Mobile Platform:***

**Operating System:** For Android: The ambulance booking system will support a specific minimum version of the Android operating system, such as Android 5.0 (Lollipop) or higher.

For iOS: The system will support a specific minimum version of iOS, such as iOS 10 or higher.

**App Store:** If the ambulance booking system is distributed through app stores (Google Play Store for Android or Apple App Store for iOS), end users will be required to ensure that they have access to the respective app store to download and install the application.

**Updates:** End users are encouraged to keep their operating system and installed applications updated with the latest security patches and bug fixes to ensure compatibility and security.

It is important to consider on these minimum software requirements, and the specific requirements may vary based on the technology stack, frameworks, and libraries used in the development of the ambulance booking system. Additionally, it is recommended to follow clear instructions of end users regarding the supported software versions and any additional dependencies or prerequisites needed to run the system smoothly.

## **Software specifications**

The system comprises 3 major modules with their sub-modules as follows:

**USER:**

**Registration**: The user can register using personal details.

**Login**: The user can log in using credentials.

**Change** **Password**: User can change their password.

**Book** **Ambulance**: Users can book an ambulance by giving its location, hospital location, and timings.

**Book** **Hospital**: The user can book Hospital.

**View** **Previous** **Booked** **Ambulance**: The user can view the previously booked ambulance.

**View** **Nearby** **Hospitals**: Users can view nearby hospitals.

**Previous** **Hospital** **Bookings**: Users can view previous hospital bookings

**AMBULANCE DRIVER:**

**Registration**: Ambulance drivers can register using personal details and ambulance number

**Login**: Ambulance Driver login in using credentials.

**Change** **Password**: Ambulance drivers can change their password.

**View** **Booking** **Request**: The ambulance driver can view the booking request.

**View** **Nearby** **Hospitals**: Ambulance drivers can view Nearby Hospitals.

**Previous** **Bookings**: Ambulance Driver can view Previous Bookings.

**HOSPITAL MANAGER:**

**Registration**: The Hospital manager can register using personal details.

**Login**: The Hospital manager can log in using credentials.

**Change** **Password**: The Hospital manager can change their password.

**View** **Booking** **Request**: The Hospital manager can view the booking request.

**Previous** **Bookings**: The Hospital manager can view previous booking requests.

**ADMINISTRATOR:**

**Login**: The Hospital manager can log in using credentials.

**Change** **Password**: The Hospital manager can change their password.

**Manage and monitor:** The admin will monitor and manage all information in the system

**View** **Booking** **Request**: The Hospital manager can view the booking request.

**Previous** **Bookings**: The Hospital manager can view previous booking requests

**Add or Remove;** The Admin will have full access to add or remove ambulance, doctor, etc.

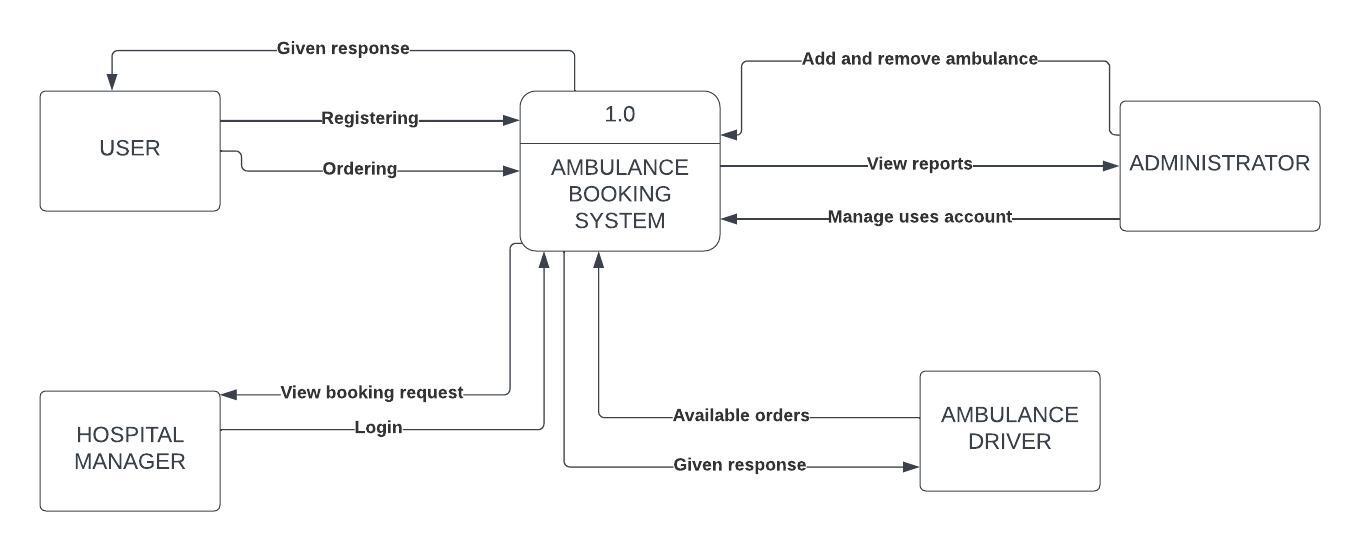
**View reports;** The Admin will have full access to view all reports and information across the system.

**Manage user accounts;** The Admin will have full access to manage, control, remove or add a user’s account.

**Perform system configuration tasks;** The Admin will have full access to make some additional configurations.

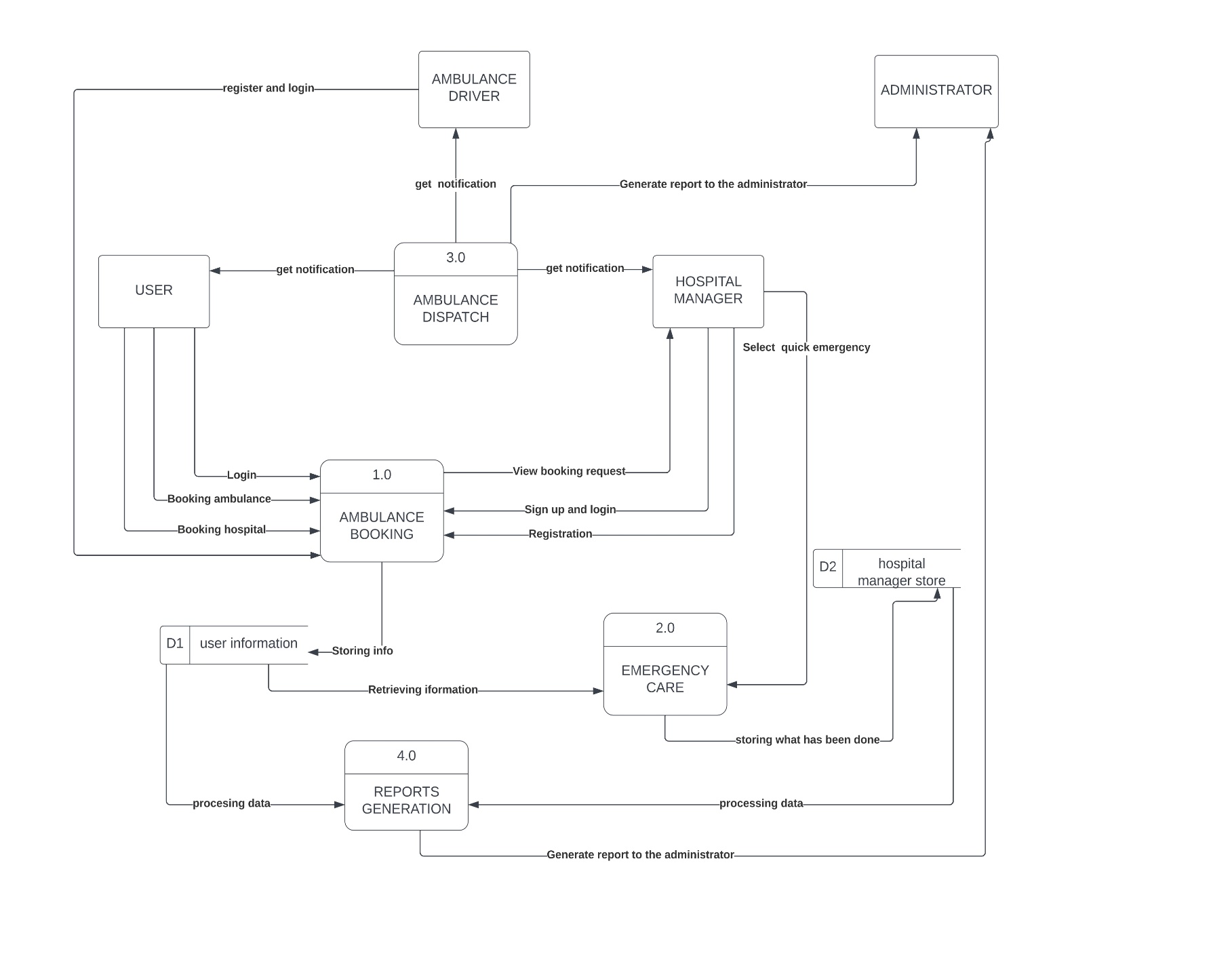
# DATA FLOW DIAGRAM (LEVEL 0, LEVEL 1**)**

4.1 DFD LEVEL 0



This figure defines the boundary between the system, or part of a system, and its environment, showing the entities that interact with it, this diagram is a high-level view of a system.

## 4.2 DFD LEVEL 1



This diagram shows the whole system is represented as a single process. A level 1 DFD notates each of the main sub-processes that together form the complete system.

# **Feasibility study**

5**.1 Technical Feasibility**

This study evaluates whether the proposed project can be successfully implemented from a technical perspective. It assesses factors such as technical requirements, availability of resources, infrastructure, and expertise needed to execute the project.

## **5.2 Financial Feasibility**

This study analyzes the financial aspects of the project, including the estimated costs, potential revenues, return on investment (ROI), and profitability. It helps determine if the project is financially viable and if the expected benefits outweigh the costs.

## **5.3 Market Feasibility**

**Market Demand:** Evaluated demand for an ambulance booking system in the target market. We made a Research and analyze factors such as population density, healthcare infrastructure, transportation challenges, and existing ambulance services. We Identified that there is a gap or opportunity for an improved and efficient ambulance booking system.

**Target Market Size:** The size and potential growth of the target market will be more expanding by the power of media and marketing. We will Consider factors such as the number of potential users, frequency of ambulance requests, and projected market expansion.

**Competitive Analysis:** The competitive landscape to existing players offering similar services will be taken as intense competition. We Analyzed their strengths, weaknesses, pricing models, and market share.

**User Acceptance and Adoption:** According to market research to gauge user acceptance and adoption of an ambulance booking system we got good considerations and good responses.

**Marketing and Distribution Channels:** We Developed a comprehensive marketing strategy to create awareness and promote your ambulance booking system.

## **5.4 Economic Feasibility**

The Project Online Ambulance Booking System uses freely available development tools and provide the system as an open-source system. Only the Maintenance Cost will be charged from the system users i.e., JSP Software Libraries that are used in this system are free open-source libraries.

Being a web application Online Ambulance Booking System will have an associated hosting cost. Since the system does not contain any multimedia data transfer, bandwidth required for the operation of this application is very low. The system will follow the freeware software standards. No cost will be charged from the potential customers. Bug fixes and maintaining tasks will have an associated cost. At the initial stage the potential market space will be the local people with lots of emergencies and redundant seat booking will be automatically cancelled thus automating some manual tasks which will be greatly useful for the ambulance system..So, the project is economically feasible.

## **5.5 Legal and Regulatory Feasibility**

The project is legally & ethically for the Citizens of India throughout. The project is being designed including the study concerning contracts, liability, violations and legal other traps frequently unknown to the technical staff. The data processing system complies with data protection act and the user data is kept secured so the project is legally feasible.

The project is legally & ethically for the Citizens of India throughout. The project is being designed including the

study concerning contracts, liability, violations and legal other traps frequently unknown to the technical staff. The

data processing system complies with data protection act and the user data is kept secured so the project is legally

feasible.

## 5.6 Operational Feasibility

Resources that are required for this project includes:

Programming Device (laptop & desktop)

Hosting Space (Freely available)

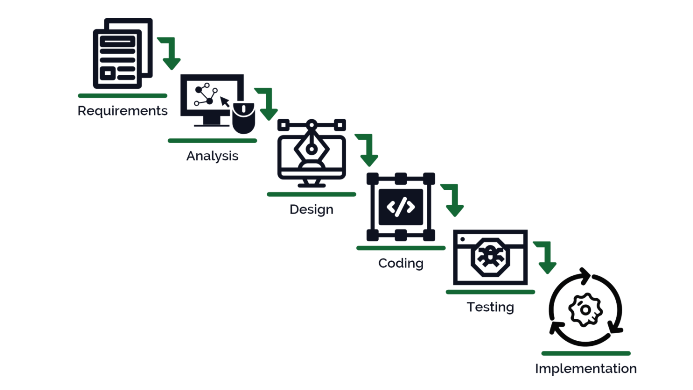
Programming tools (free open-source server)

Programming, designing, planning individuals

A proper need of planning and execution by the efficient members under a proper guidance is needed. We are building the project on Windows 10 platform which is capable of making & testing of the software product. As basically windows versions are preferred mostly in workplaces so it will be easy to consult the problems that arise or the required improvements. The software will be made more user friendly and easier to handle from the back end so that users are comfortable using it. As all the mentioned needs are met thus, we can say that our project is operationally feasible.

# Description of the selected Process Model

The waterfall model is a classical model used in the system development life cycle to create a system with a linear and sequential approach. It is termed a waterfall because the model develops systematically from one phase to another in a downward fashion. The waterfall approach does not define the process to go back to the previous phase to handle changes in requirements. The waterfall approach is the earliest approach that was used for software development.



## **Cause of your selection**

The waterfall model is a sequential software development process that follows a linear and structured approach. Here are some reasons why the waterfall model was best for our project:

**Sequential and Predictable:** The waterfall model provides a clear and sequential flow of development phases, including requirements gathering, design, implementation, testing, and deployment. Each phase has well-defined objectives and outputs, making it predictable and easier to manage.

**Clear Documentation:** The waterfall model emphasizes documentation at each stage, ensuring that requirements, designs, and test cases are thoroughly documented. This documentation serves as a valuable reference for future maintenance, updates, or audits.

**Risk Management:** The waterfall model encourages upfront planning and risk identification early in the project lifecycle. By addressing risks and potential issues during the planning and design phases, it helps minimize risks and uncertainties throughout the development process.

**Client Engagement:** The waterfall model allows for comprehensive client engagement during the requirements gathering and design phases. Clients can review and provide feedback on the documentation and design artifacts before development begins, reducing the chances of misunderstandings or costly changes later in the process.

**Well-suited for Stable Requirements:** The waterfall model works well when the project requirements are stable and unlikely to change significantly. It assumes that the requirements are fully understood and documented at the beginning, minimizing the need for frequent changes during development.

**Resource Allocation:** The waterfall model facilitates resource planning and allocation. Since each phase has distinct deliverables, resources can be assigned to specific tasks and phases based on the project schedule and requirements.

**Regulatory and Compliance Requirements:** In certain industries or projects with strict regulatory or compliance requirements, the waterfall model can provide a structured and traceable development process. The documentation and sign-off at each phase help ensure compliance with standards and regulations.

# **Conclusion**

The efficient alternative to ambulance software program offers a comprehensive solution to enhance emergency medical services by leveraging technology, communication, and optimization algorithms. By streamlining the emergency response process, this program aim somaticizes response times, improve patient outcomes, and save lives.

In conclusion, an ambulance booking system project is a critical application that aims to facilitate the efficient and reliable booking of ambulance services. Throughout the project, various requirements need to be considered to ensure the system's success and usability.

Reliability requirements are crucial to maintain system availability, fault tolerance, data integrity, and performance stability. The system should be designed to handle failures, maintain data accuracy, and deliver consistent performance to meet user expectations. Additionally, backup and recovery mechanisms, error handling, and proactive maintenance should be implemented to enhance reliability.

Usability requirements are equally important to ensure a user-friendly experience. The system should have an intuitive interface, simple registration and login processes, and a clear booking process. Efficient search functionality, clear feedback and confirmation messages, and error prevention and handling contribute to a positive user experience. Personalization options and adherence to accessibility guidelines are also important considerations.

To successfully develop an ambulance booking system project, it is essential to engage in user research, usability testing, and continuous user feedback. This helps in understanding user needs, identifying usability issues, and refining the system accordingly.

Overall, by addressing reliability and usability requirements, an ambulance booking system can provide a reliable, efficient, and user-friendly platform for users to request and receive prompt ambulance services when they need them the most.

# **References**

[*https://www.researchgate.net/publication/342154636\_Ambuitec\_Ambulance\_Booking\_Application\_for\_Emergency\_Health\_Response\_Blood\_Inventory*](https://www.researchgate.net/publication/342154636_Ambuitec_Ambulance_Booking_Application_for_Emergency_Health_Response_Blood_Inventory)

[*https://www.researchgate.net/publication/309888533\_Mobile-Based\_Medical\_Emergency\_Ambulance\_Scheduling\_System*](https://www.researchgate.net/publication/309888533_Mobile-Based_Medical_Emergency_Ambulance_Scheduling_System)

*https://www.irjmets.com/uploadedfiles/paper/issue\_4\_april\_2022/21801/final/fin\_irjmets1651499822.pdf*

CHAPTER 2 : DATABASE TECHNOLOGY

2.2 SECTION1

2.2.1 ENTITIES

**User :** A User Entity will contain all information that concerns to patients that created an account in the system. There attributes will be as follow:

***UserID***: A unique identifier for each user.

***Name:*** The name of the user.

***PhoneNumber:*** The phone number of the user.

***Email:*** The email address of the user.

***Address:*** The address of the user.

***DateOfBirth:*** The date of birth of the user.

***Gender:*** The gender of the user.

***MedicalHistory:*** Information about the user's medical history.

***InsuranceInfo:*** Information about the user's insurance .

**AMBULANCE Entity:** An ambulance entity will hold all information about Ambulances in the system and it will contain information about the ambulance drivers. The attributes will be as follows:

***AmbulanceID:*** A unique identifier for each ambulance.

***VehicleNumber:*** The number or plate number of the ambulance.

***Type:*** The type or category of the ambulance.

***Capacity:*** The maximum number of passengers the ambulance can carry.

***Equipment:*** Information about the equipment available in the ambulance .

***CurrentLatitude:*** The current latitude coordinate of the ambulance's location.

***CurrentLongitude:*** The current longitude coordinate of the ambulance's location.

***AvailabilityStatus:*** The status indicating whether the ambulance is available.

***DriverName:*** The name of the ambulance driver.

***DriverContact:*** The contact information of the ambulance driver.

***DriverCertification:*** Certification details of the ambulance driver.

**BOOKING REQUEST Entity:** This entity will contain all information about the bookings that has been made by the users, it will contain information about the Hospitals that have been chosen as the destination of the patient. The attributes will be as follow:

***BookingID:*** A unique identifier for each booking request.

***UserID:*** The ID of the user making the booking (foreign key referencing User table).

***PickupLatitude:*** The latitude coordinate for the pickup location.

***PickupLongitude:*** The longitude coordinate for the pickup location.

***DestinationLatitude:*** The latitude coordinate for the destination.

***DestinationLongitude:*** The longitude coordinate for the destination.

***HospitalName:*** The name of the hospital (if applicable).

***RequestedTime:*** The date and time when the booking was requested.

***PriorityLevel:*** The priority level of the booking.

***Status:*** The status of the booking.

***Reason:*** The reason for the booking .

***AdditionalNotes:*** Additional notes or comments related to the booking .

**TRIP ENTITY**: This entity will hold all information that concerns to the Travels and trips done according to the bookings that were made. The attributes will be as follow;

***TripID:*** A unique identifier for each trip.

***BookingID:*** The ID of the booking associated with the trip (foreign key referencing BookingRequest table).

***AmbulanceID:*** The ID of the ambulance assigned to the trip (foreign key referencing Ambulance table).

***PickupTime:*** The date and time of the trip's pickup.

***DropoffTime:*** The date and time of the trip's drop-off.

***DistanceTraveled:*** The distance traveled during the trip.

***EstimatedArrivalTime:*** The estimated arrival time at the destination.

***Fare:*** The fare or cost of the trip.

***PaymentStatus:*** The payment status of the trip.

**PAYMENT ENTITY:** This Entity will contain information about the payments that will be done according to the trips taken by the users, it show whether the trip was paid or not. The attributes will be as follow;

***PaymentID:*** A unique identifier for each payment.

***TripID:*** The ID of the trip associated with the payment (foreign key referencing Trip table).

***UserID:*** The ID of the user making the payment (foreign key referencing User table).

***PaymentMethod***: The method used for payment.

***Amount:*** The amount paid.

***PaymentDateTime:*** The date and time of the payment.

**EMERGENCY CONTACTS ENTITY:** This entity will hold information about the emergency contacts of the patients signed up in the system, These contacts will be used incase there is any issue that needs to be told to the family of the patient. The attributes will be as follow;

***ContactID***: A unique identifier for each emergency contact.

***UserID***: The ID of the user associated with the emergency contact (foreign key referencing User table).

***ContactName:*** The name of the emergency contact.

***Relationship:*** The relationship between the user and the emergency contact.

***ContactPhoneNumber:*** The phone number of the emergency contact.

**FEEDBACK AND RATINGS Entity:** This entity will show and contain all feedbacks of the users and it will help to see the ratings of our users about the service we delivered to them. The attributes will be as follow;

***FeedbackID:*** A unique identifier for each feedback and rating.

***TripID:*** The ID of the trip associated with the feedback and rating (foreign key referencing Trip table).

***UserID:*** The ID of the user providing feedback and rating (foreign key referencing User table).

***Rating:*** The rating provided by the user.

***Comments:*** Comments or feedback from the user.

***FeedbackDateTime:*** The date and time when the feedback was provided.

**ADMIN ENTITY:** This will contain all information about the admins in the system, this will differentiate a Dispatcher from Coordinator or an administrator in the system. The attributes will be as follow;

***AdminID:*** A unique identifier for each admin.

***Name:*** The name of the admin.

***PhoneNumber:*** The phone number of the admin.

***Email:*** The email address of the admin.

***Role:*** The role or position of the admin.

**NOTIFICATION ENTITY:** This entity will show and contain the notifications about the actions in the system, this will show if a user has received a notification, this will help us to see if there is any new user created, if an ambulance is available or any action done in the system. The attributes will be as follow;

***NotificationID:*** A unique identifier for each notification.

***UserID:*** The ID of the user associated with the notification (foreign key referencing User table).

***Content:*** The content of the notification .

***Timestamp:*** The timestamp indicating when the notification was sent.

***Status***: The status of the notification.

**REPORTS AND ANALYTICS ENTITY:** This entity will contain the report, analytics and views of the administrators as there are the coordinators and auditors in the system. The attributes will be as follow;

***ReportID:*** A unique identifier for each report.

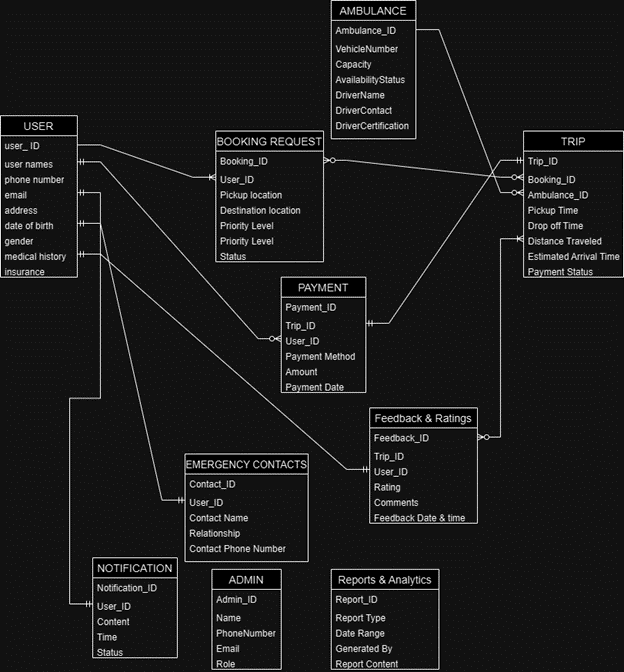
***ReportType:*** The type or category of the report.

***DateRange:*** The date range covered by the report.

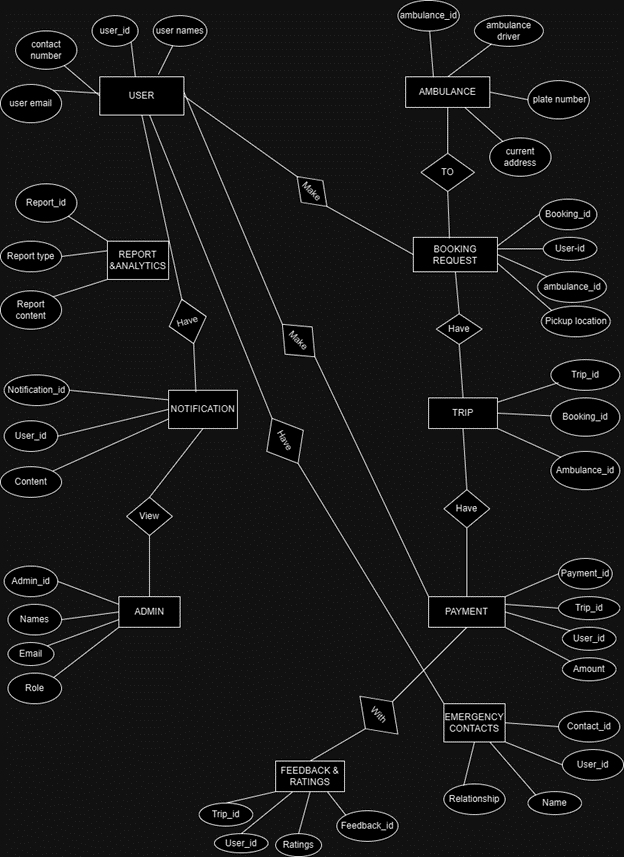
***GeneratedBy:*** The ID of the admin who generated the report (foreign key referencing Admin table).

***ReportContent:*** The content or details of the report .

2.2.2 LOGICAL DATABASE MODEL

****

2.2.2ENTITY RELETIONSHIP DIAGRAM

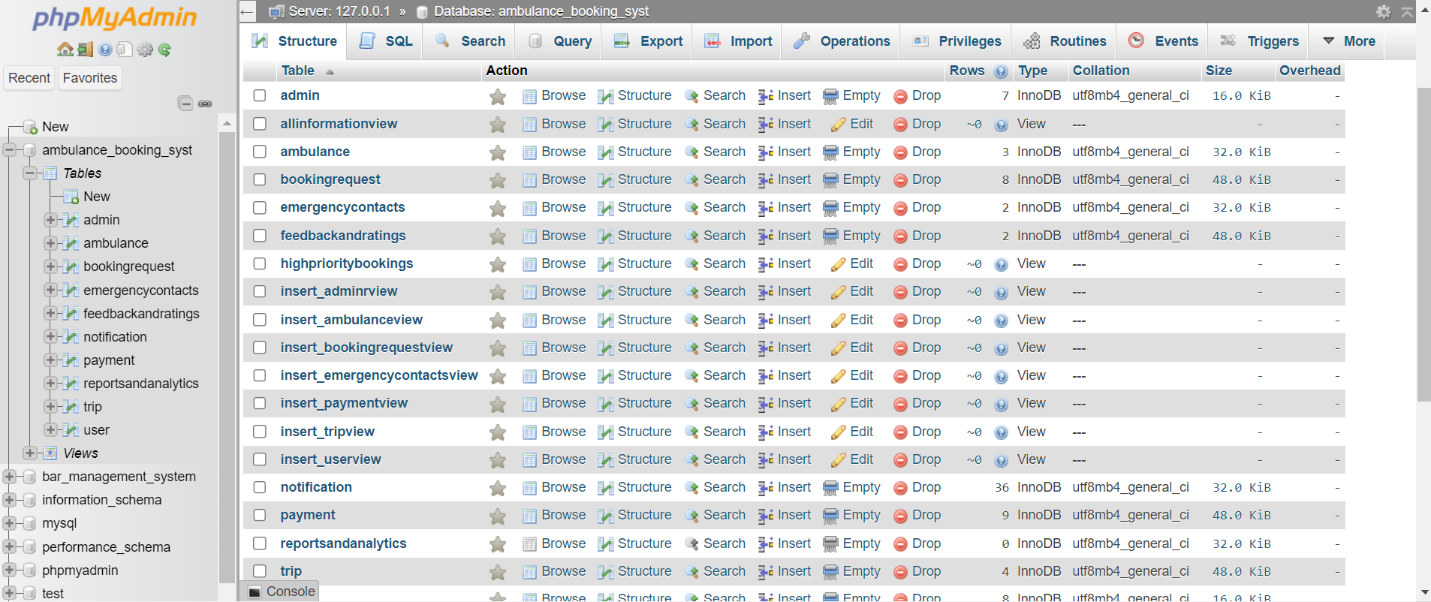


SECTION2:SQL

1.DATABASE :

Create the database of your system

CREATE DATABASE Ambulance\_booking\_syst;



Q2. Write queries to create all the tables and relationships of your system

**1.USER TABLE**

CREATE TABLE User (

UserID INT AUTO\_INCREMENT PRIMARY KEY,

Name VARCHAR(255),

PhoneNumber VARCHAR(20),

Email VARCHAR(255),

Address VARCHAR(255),

DateOfBirth DATE,

Gender VARCHAR(10),

Password VARCHAR(255),

InsuranceTEXT

);

**2.AMBULANCE TABLE**

CREATE TABLE Ambulance (

AmbulanceID INT AUTO\_INCREMENT PRIMARY KEY,

VehicleNumber VARCHAR(20),

Type VARCHAR(50),

Capacity INT,

Equipment TEXT,

CurrentLocation VARCHAR(50),

DriverName VARCHAR(255),

DriverContact VARCHAR(20),

DriverCertification VARCHAR(255)

);

**3.BOOKING REQUEST TABLE**

CREATE TABLE BookingRequest (

BookingID INT AUTO\_INCREMENT PRIMARY KEY,

UserID INT,

PickupLocation VARCHAR(50),

Destination VARCHAR(50),

HospitalName VARCHAR(50),

RequestedTime DATETIME,

PriorityLevel VARCHAR(20),

Reason TEXT,

AdditionalNotes TEXT,

FOREIGN KEY (UserID) REFERENCES User(UserID)

);

**4.TRIP TABLE**

CREATE TABLE Trip (

TripID INT AUTO\_INCREMENT PRIMARY KEY,

BookingID INT,

AmbulanceID INT,

PickupTime DATETIME,

DropoffTime DATETIME,

DistanceTraveled VARCHAR(50),

Fare VARCHAR(50),

PaymentStatus VARCHAR(20),

FOREIGN KEY (BookingID) REFERENCES BookingRequest(BookingID),

FOREIGN KEY (AmbulanceID) REFERENCES Ambulance(AmbulanceID)

);

**5.PAYMENTE TABLE**

CREATE TABLE Payment (

PaymentID INT AUTO\_INCREMENT PRIMARY KEY,

TripID INT,

UserID INT,

PaymentMethod VARCHAR(20),

Amount VARCHAR(50),

PaymentDateTime DATETIME,

FOREIGN KEY (TripID) REFERENCES Trip(TripID),

FOREIGN KEY (UserID) REFERENCES User(UserID)

);

**6. EMERGENCY CONTANCT TABLE**

CREATE TABLE EmergencyContacts (

ContactID INT AUTO\_INCREMENT PRIMARY KEY,

UserID INT,

ContactName VARCHAR(255),

Relationship VARCHAR(50),

ContactPhoneNumber VARCHAR(20),

FOREIGN KEY (UserID) REFERENCES User(UserID)

);

**7.FEEDBACK AND RATINGS TABLE**

CREATE TABLE FeedbackAndRatings (

FeedbackID INT AUTO\_INCREMENT PRIMARY KEY,

TripID INT,

UserID INT,

**Rating\_Stars** VARCHAR(50),

Comments TEXT,

FeedbackDateTime DATETIME,

FOREIGN KEY (TripID) REFERENCES Trip(TripID),

FOREIGN KEY (UserID) REFERENCES User(UserID)

);

**8.ADMIN TABLE**

CREATE TABLE Admin (

AdminID INT AUTO\_INCREMENT PRIMARY KEY,

Name VARCHAR(255),

PhoneNumber VARCHAR(20),

Email VARCHAR(255),

Role VARCHAR(50),

Password VARCHAR(50)

);

**9.NOTIFICATION TABLE**

CREATE TABLE Notification (

NotificationID INT AUTO\_INCREMENT PRIMARY KEY,

UserID INT,

Content TEXT,

Timestamp DATETIME,

Status VARCHAR(20),

FOREIGN KEY (UserID) REFERENCES User(UserID)

);

**10. REPORTS AND ANALYTICS TABLE**

CREATE TABLE ReportsAndAnalytics (

ReportID INT AUTO\_INCREMENT PRIMARY KEY,

ReportType VARCHAR(50),

DateRange VARCHAR(50),

GeneratedBy INT,

ReportContent TEXT,

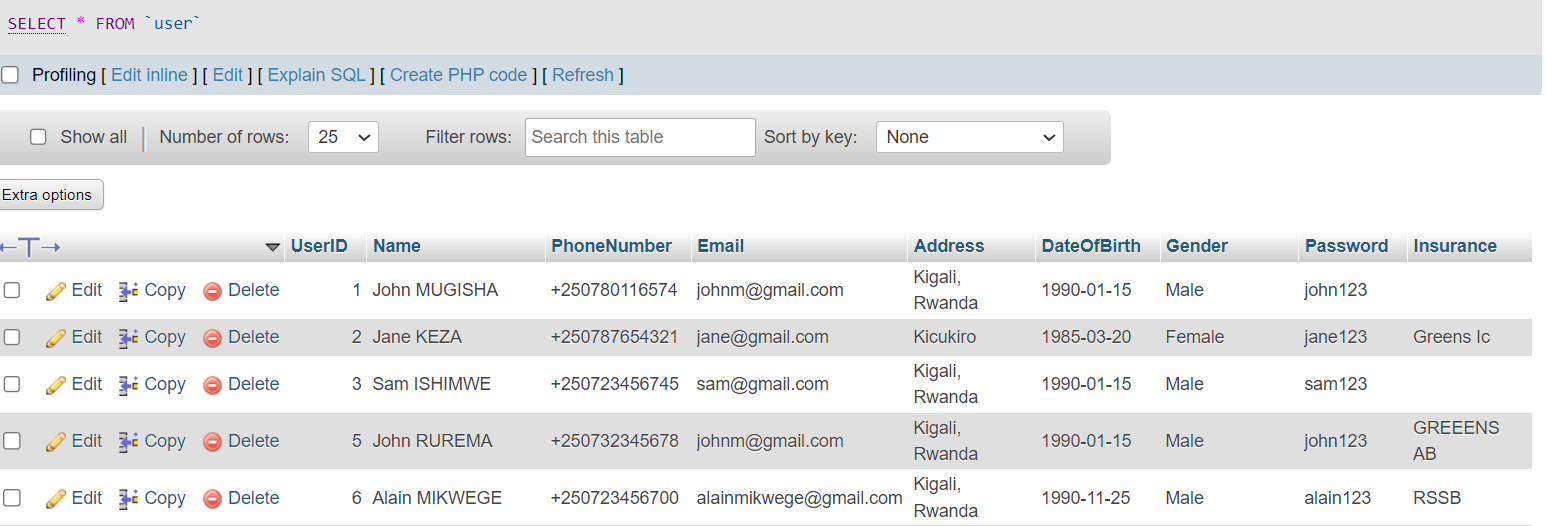
FOREIGN KEY (GeneratedBy) REFERENCES Admin(AdminID)

);

***Data for User table***

INSERT INTO User (Name, PhoneNumber, Email, Address, DateOfBirth, Gender, InsuranceInfo, Password)

VALUES ('John MUGISHA', '+250 788 657 842', 'johnmugisha@gmail.com', 'NR1 Street', '1998-02-11', 'Male', 'UAP', '123456');



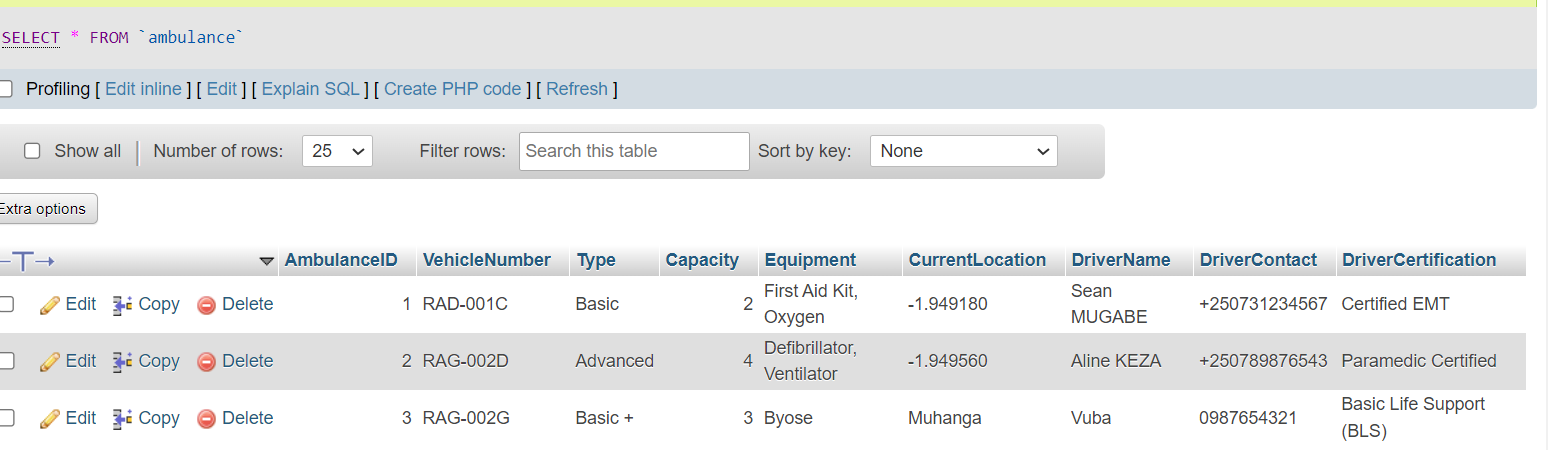
***Data for Ambulance table***

INSERT INTO Ambulance (VehicleNumber, Type, Capacity, Equipment, CurrentLatitude, CurrentLongitude, AvailabilityStatus, DriverName, DriverContact, DriverCertification)

VALUES

('RAD-001C', 'Basic', 2, 'First Aid Kit, Oxygen', -1.94918, 30.05836, 'Available', 'Sean MUGABE', '+250731234567', 'Certified EMT'),

('RAG-002D', 'Advanced', 4, 'Defibrillator, Ventilator', -1.94956, 30.05974, 'Available', 'Aline KEZA', '+250789876543', 'Paramedic Certified');



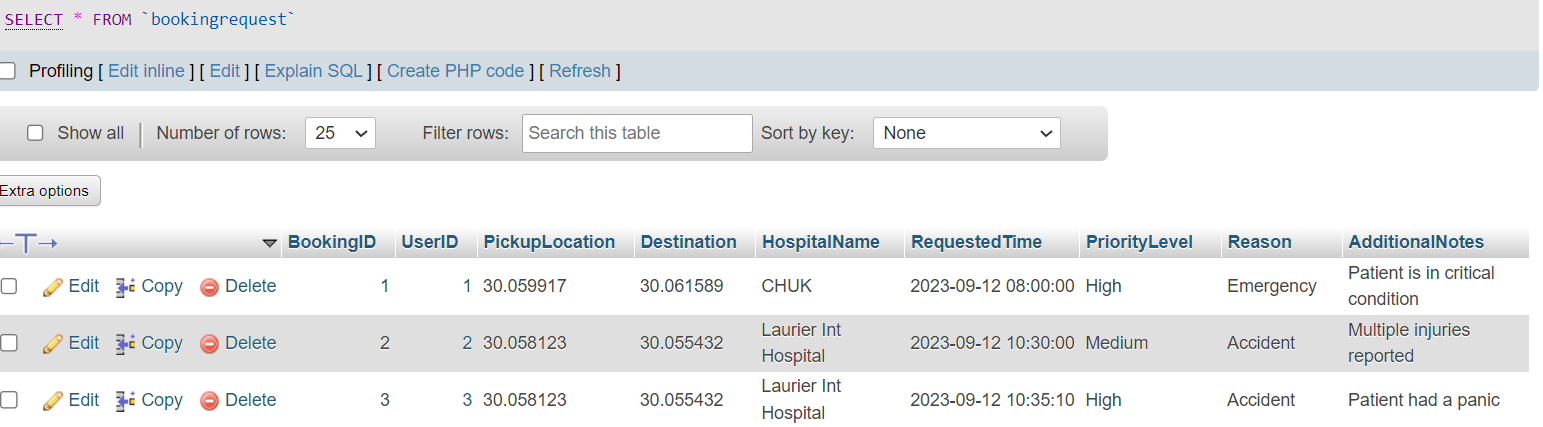
***Data for BookingRequest table***

INSERT INTO BookingRequest (UserID, PickupLatitude, PickupLongitude, DestinationLatitude, DestinationLongitude, HospitalName, RequestedTime, PriorityLevel, Status, Reason, AdditionalNotes)

VALUES

(1, -1.949285, 30.059917, -1.932574, 30.061589, 'CHUK', '2023-09-12 08:00:00', 'High', 'Pending', 'Emergency', 'Patient is in critical condition'),

(2, -1.942345, 30.058123, -1.926789, 30.055432, 'Laurier Int Hospital', '2023-09-12 10:30:00', 'Medium', 'Pending', 'Accident', 'Multiple injuries reported');



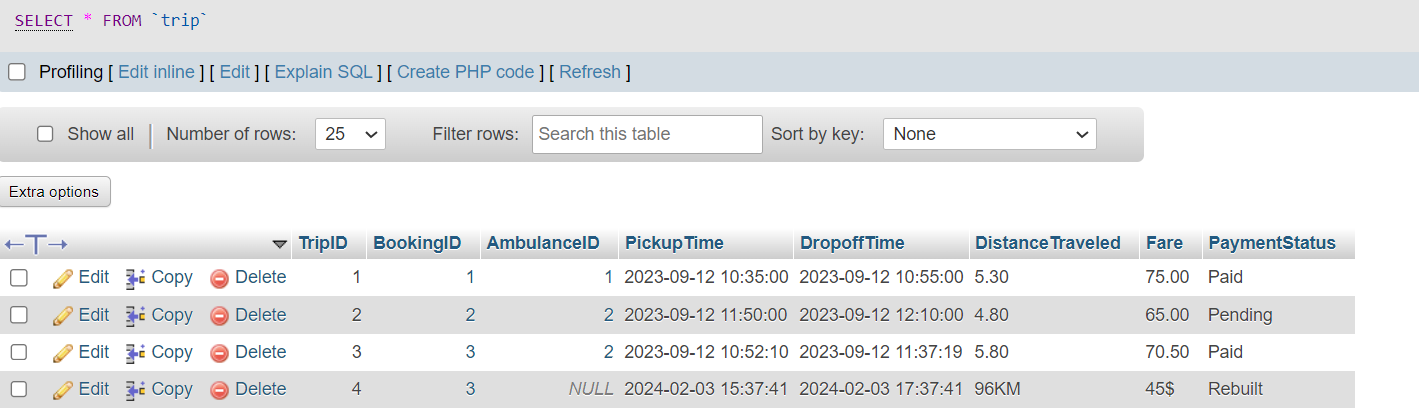
***Data for Trip table***

INSERT INTO Trip (BookingID, AmbulanceID, PickupTime, DropoffTime, DistanceTraveled, EstimatedArrivalTime, Fare, PaymentStatus)

VALUES

(1, 1, '2023-09-12 10:35:00', '2023-09-12 10:55:00', 5.3, '2023-09-12 10:55:00', 75.00, 'Paid'),

(2, 2, '2023-09-12 11:50:00', '2023-09-12 12:10:00', 4.8, '2023-09-12 12:10:00', 65.00, 'Pending');



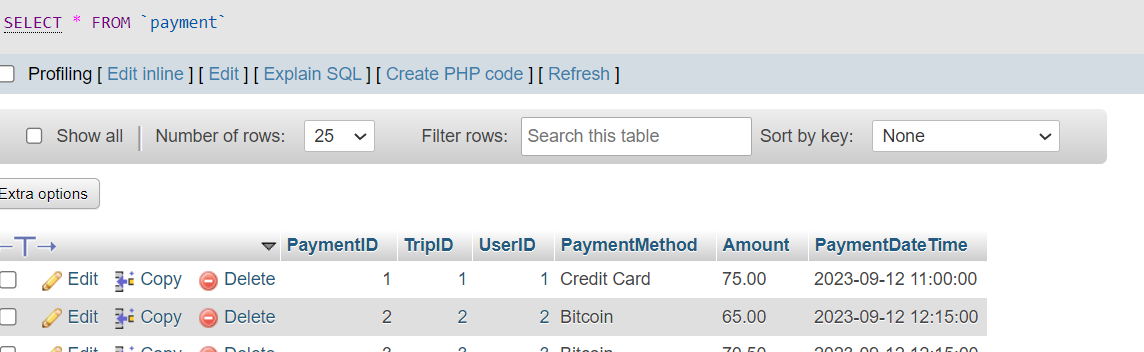
***Data for Payment table***

INSERT INTO Payment (TripID, UserID, PaymentMethod, Amount, PaymentDateTime)

VALUES

(1, 1, 'Credit Card', 75.00, '2023-09-12 11:00:00'),

(2, 2, 'Bitcoin', 65.00, '2023-09-12 12:15:00');



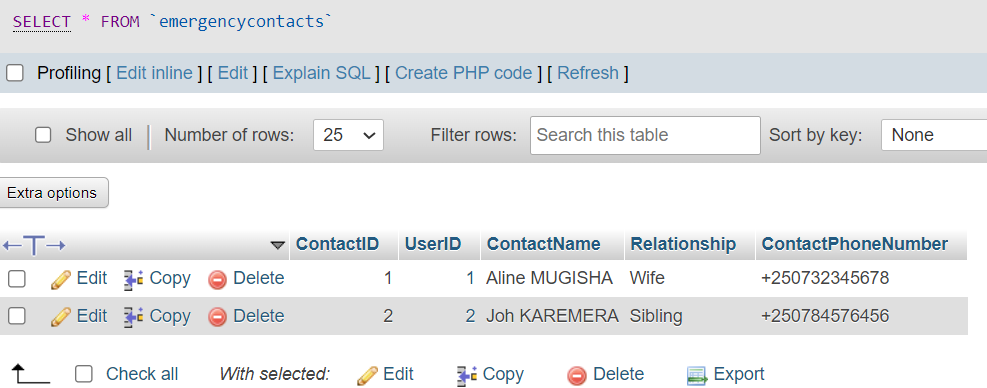
***Data for EmergencyContacts table***

INSERT INTO EmergencyContacts (UserID, ContactName, Relationship, ContactPhoneNumber)

VALUES

(1, 'Aline MUGISHA', 'Wife', '+250732345678'),

(2, 'Ane KEZA', 'Sibling', '+250780123456');



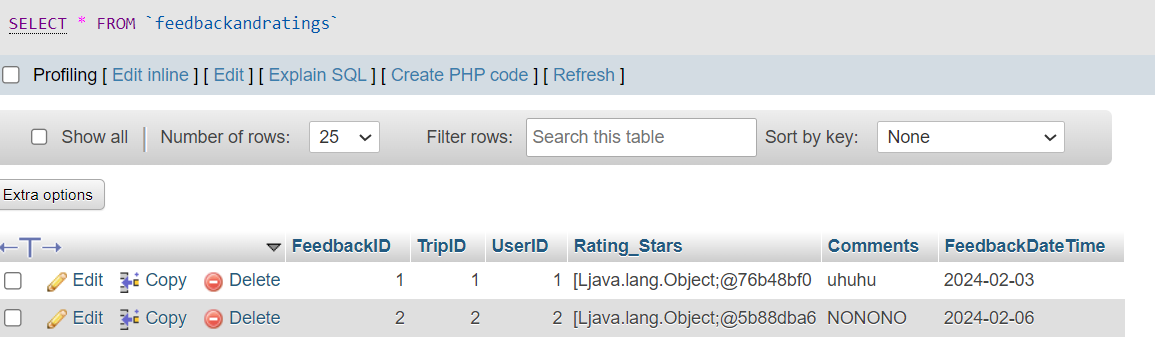
***Data for FeedbackAndRatings table***

INSERT INTO FeedbackAndRatings (TripID, UserID, Rating, Comments, FeedbackDateTime)

VALUES

(1, 1, 5, 'Excellent service! Mukomereze aho.', '2023-09-12 11:30:00'),

(2, 2, 4, 'Badutabaye mugihe cyihuse, murakoze.', '2023-09-12 12:30:00');



***Data for Admin table***

INSERT INTO Admin (Name, PhoneNumber, Email, Role)

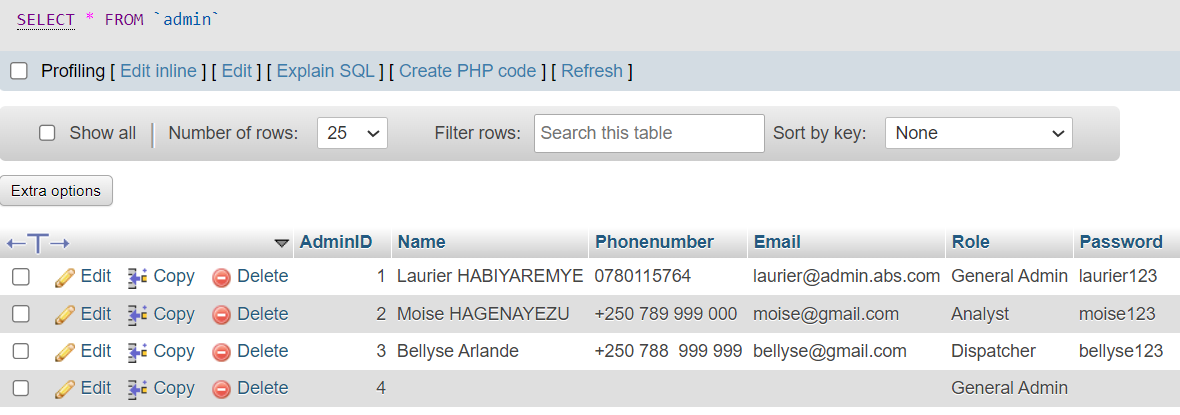
VALUES

('Jean MURONGO', '+250781111111', 'admin1@gmail.com', 'Administrator'),

('Alice MBONGO', '+250722222222', 'admin2@gmail.com', 'Dispatcher'),

('Laurier Greens RUGWIZA', '+250733333333', 'admin3@gmail.com', 'Coordinator'),

('Martin MIHIGO', '+250782111111', 'admin1@gmail.com', 'Administrator');



2.4.1 VIEWS, TRIGGERS AND PROCEDURES

**create view insert\_Userview AS select\*from User where UserID=4;**

**create view insert\_Adminrview AS select\*from Admin where AdminID=1;**

**create view insert\_Ambulanceview AS select\*from Ambulance where AmbulanceID=3;**

**create view insert\_Tripview AS select\*from Trip where TripID=3;**

**create view insert\_Paymentview AS select\*from Payment where PaymentID=3;**

**create view insert\_BookingRequestview AS select\*from BookingRequest where BookingID=3;**

**create view insert\_EmergencyContactsview AS select\*from EmergencyContacts where ContactID=3;**

1. create one view of your choice that considers sub-query.

CREATE VIEW HighPriorityBookings AS

SELECT \*

FROM BookingRequest

WHERE PriorityLevel = 'High';

SELECT \*

FROM HighPriorityBookings;

2.4 CONCLUSION

In conclusion, the Ambulance Booking System represents a crucial step forward in optimizing emergency response services. Through the development and implementation of this database, several key objectives have been achieved.

Firstly, the system provides a streamlined process for managing ambulance bookings, ensuring that emergency requests are promptly addressed and efficiently coordinated. By centralizing booking information and automating various administrative tasks, the system reduces the risk of errors and delays, ultimately enhancing the overall responsiveness of emergency services.

Additionally, the database facilitates effective communication and collaboration among stakeholders involved in the ambulance booking process. Emergency dispatchers, ambulance crews, and healthcare facilities can easily access and update booking details, allowing for real-time coordination and decision-making. This collaborative approach fosters a more cohesive and integrated emergency response ecosystem, leading to improved patient outcomes and satisfaction.

Moreover, the database incorporates robust security measures to safeguard sensitive patient information and ensure compliance with relevant data protection regulations. Access controls, encryption protocols, and audit trails are implemented to mitigate the risk of unauthorized access or data breaches, thereby maintaining the confidentiality and integrity of patient records.

Overall, the Ambulance Booking System represents a valuable tool for enhancing the efficiency, effectiveness, and safety of emergency medical services. By leveraging modern technology and best practices in database management, the system empowers healthcare providers to deliver timely and responsive care to those in need, ultimately saving lives and improving community well-being.

As we move forward, continuous monitoring, evaluation, and refinement of the database will be essential to address emerging challenges, accommodate evolving needs, and capitalize on new opportunities for innovation. By remaining agile and adaptive, we can ensure that the Ambulance Booking System continues to serve as a reliable and indispensable resource for emergency response operations.

**CHAPTER3: JAVA PROGRAMMING**

**3.1 INTRODUCTION**

In today's fast-paced world, efficient emergency medical services (EMS) are crucial for ensuring the well-being of individuals in need of urgent medical attention. The Ambulance Booking System (ABS) is a Java-based application designed to streamline the process of requesting and managing ambulance services effectively. This innovative system leverages modern technology to provide a seamless and reliable platform for both users and emergency service providers.

The ABS offers a user-friendly interface accessible via desktop and mobile devices, allowing users to quickly request ambulance services with just a few clicks. Whether it's for medical emergencies, non-emergency transport, or pre-scheduled appointments, the ABS ensures prompt and efficient dispatch of ambulances to the requested locations.

Key features of the Ambulance Booking System include:

1. User Registration and Authentication: Users can create accounts securely and log in to access the system, ensuring data privacy and user authentication.
2. Ambulance Request Management: Users can submit ambulance requests, providing essential details such as location, type of emergency, and special requirements.
3. Real-time Tracking: The system enables real-time tracking of ambulance location and estimated time of arrival (ETA), allowing users to monitor the status of their requests.
4. Emergency Alerts and Notifications: The ABS sends instant notifications and alerts to users and emergency responders, ensuring timely communication and coordination during emergencies.
5. Admin Dashboard: Administrators have access to a centralized dashboard for managing ambulance bookings, dispatching vehicles, and monitoring system activities.
6. Data Analytics and Reporting: The system generates comprehensive reports and analytics to track ambulance utilization, response times, and other key performance metrics, enabling continuous improvement of EMS operations.

The Ambulance Booking System aims to revolutionize the way emergency medical services are accessed and delivered, promoting efficiency, accessibility, and reliability in healthcare delivery. By harnessing the power of Java technology, this application empowers users and emergency responders alike to collaborate effectively in times of need, ultimately saving lives and improving outcomes in emergency situations.

**3.2 TOOLS USED TO DEVELOP THIS SYSTEM IN JAVA PROGRAMMING**

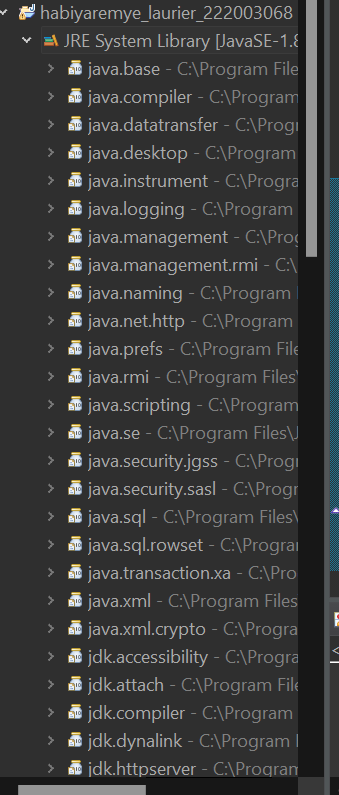
**Eclipse IDE:** an integrated development environment used in computer programming. It contains a base workspace and an extensible plug-in system for customizing the environment. It is the second-most-popular IDE for Java development, and, until 2021, was the most popular.

**Java Runtime Environment (JRE)**

The Ambulance Booking System (ABS) relies on the Java Runtime Environment (JRE) to execute its Java-based code and ensure cross-platform compatibility. The JRE provides the necessary runtime environment for running Java applications, including the ABS, on various operating systems without requiring recompilation.

By utilizing the JRE, the ABS offers seamless deployment across different platforms, including Windows, macOS, and Linux, ensuring that users can access the system regardless of their operating system preferences. This approach simplifies the installation process and enhances accessibility, allowing users to benefit from the system's features without the need for complex setup procedures or platform-specific configurations.

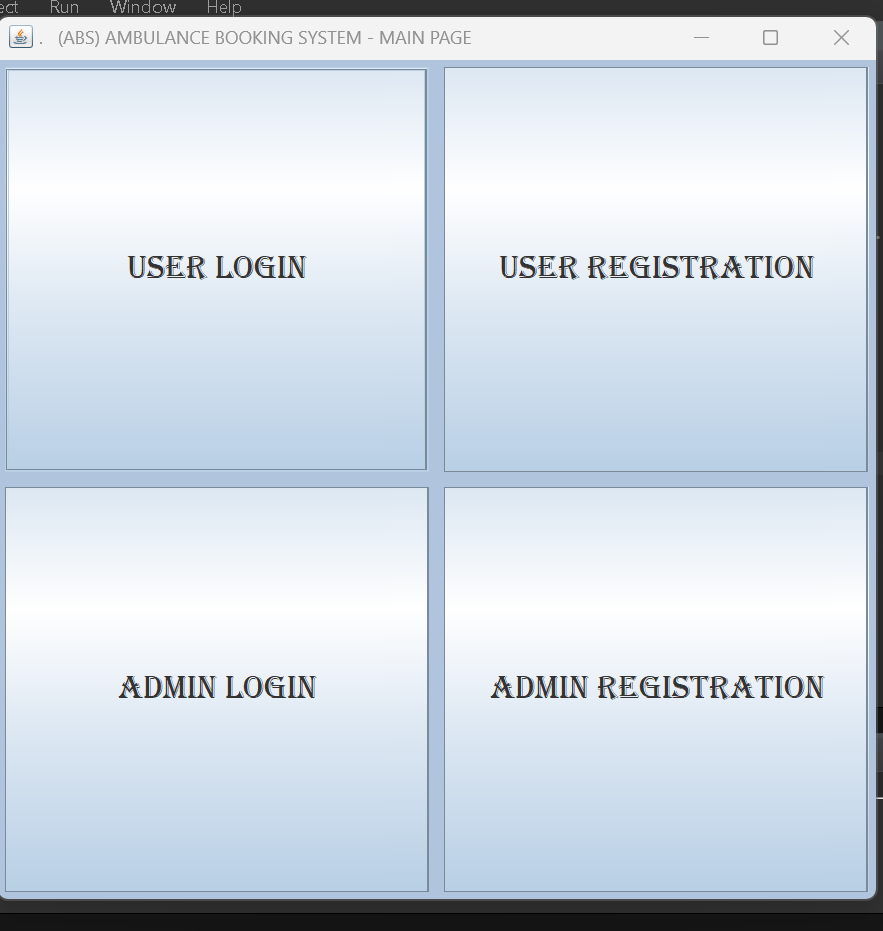
Furthermore, the JRE ensures optimal performance and stability for the ABS, enabling efficient execution of critical functions such as user authentication, ambulance request processing, real-time tracking, and administrative tasks. By leveraging the capabilities of the JRE, the ABS delivers a robust and reliable solution for managing ambulance services effectively, contributing to enhanced emergency medical care and improved patient outcomes.



**3.3 FORMS DESCRIPTION …………………………………………………………………41**

**1.AMBULANCE BOOKING SYSTEM**

This Ambulance Booking System (ABS) main page form serves as the central hub for accessing various functionalities within the application. Here's how it works:



1. **User Interface**: Upon launching the application, the user is presented with a graphical user interface (GUI) containing four buttons, each representing a distinct action related to user and admin interactions with the system.
2. **Button Functionality**:
   * **User Login**: Clicking this button prompts the user to log in with their credentials through a separate user login form.
   * **User Registration**: Clicking this button initiates the user registration process, allowing new users to create an account by filling out a registration form.
   * **Admin Login**: Clicking this button directs administrators to log in using their credentials via a dedicated admin login form.
   * **Admin Registration**: Clicking this button enables administrators to register for a new account by providing necessary details through an admin registration form.
   * **Form Navigation**: Upon clicking a button, the respective form associated with the action is opened, allowing users and administrators to perform the desired tasks within the ABS.

Overall, this main page form serves as the entry point for users and administrators to interact with the ABS, providing a user-friendly interface for accessing its functionalities.

**2.Admin Registration**

The **AdminRegistration** class facilitates the registration process for new administrators within the Ambulance Booking System (ABS) application. Upon launching the registration form, administrators are prompted to enter their personal details, including name, phone number, email address, role, and password. The form provides text fields and dropdown menus for inputting this information. After filling out the required fields, administrators can click the "REGISTER" button to submit their details. Upon submission, the system validates the information and inserts it into the database. If the registration is successful, a confirmation message is displayed, welcoming the new administrator. Additionally, the form includes a "GO TO LOGIN" button, allowing administrators to navigate to the login page if they already have an account. Overall, the **AdminRegistration** form streamlines the process of onboarding new administrators into the ABS application, ensuring efficient management of administrative roles and responsibilities.

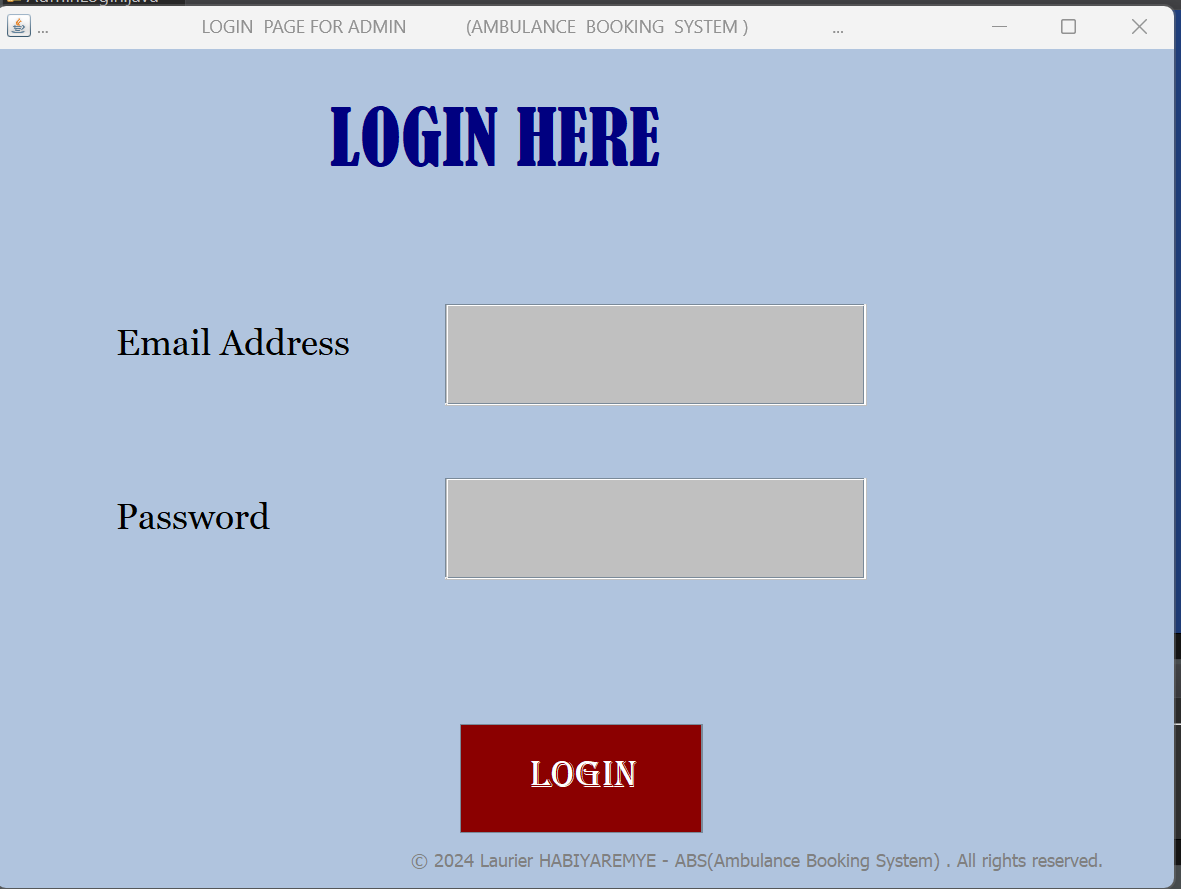
A screenshot of a computer

Description automatically generated

**2.Admin Login**

The **AdminLogin** class serves as the gateway for administrators to access the Ambulance Booking System (ABS) application. It presents a login interface where administrators are prompted to enter their email address and password. The form provides text fields for inputting the credentials, with the email address labeled as "Email Address" and the password field labeled as "Password". Upon entering the required information, administrators can click the "Login" button to authenticate their credentials. The system then validates the provided details by querying the database. If the credentials are correct, the administrator is granted access to the system's home page, represented by the **AdminHome** class. Conversely, if the provided credentials are incorrect, an error message is displayed, notifying the administrator of the incorrect username or password. Overall, the **AdminLogin** form ensures secure access to the ABS application for authorized administrators while maintaining robust authentication measures to protect sensitive system data.

Top of Form



**4.Admin Home**

The **AdminHome** class serves as the central dashboard for administrators within the Ambulance Booking System (ABS) application. It provides a comprehensive interface where administrators can perform various actions related to managing the system. The dashboard features buttons for accessing different functionalities, such as viewing details, viewing database tables, making payments, handling booking requests, managing emergency contacts, registering ambulances, and updating data. Additionally, the dashboard offers options for logging out and changing passwords to ensure security and account management. Overall, the **AdminHome** class enhances administrative efficiency by offering a centralized hub for accessing and managing key aspects of the ABS application.

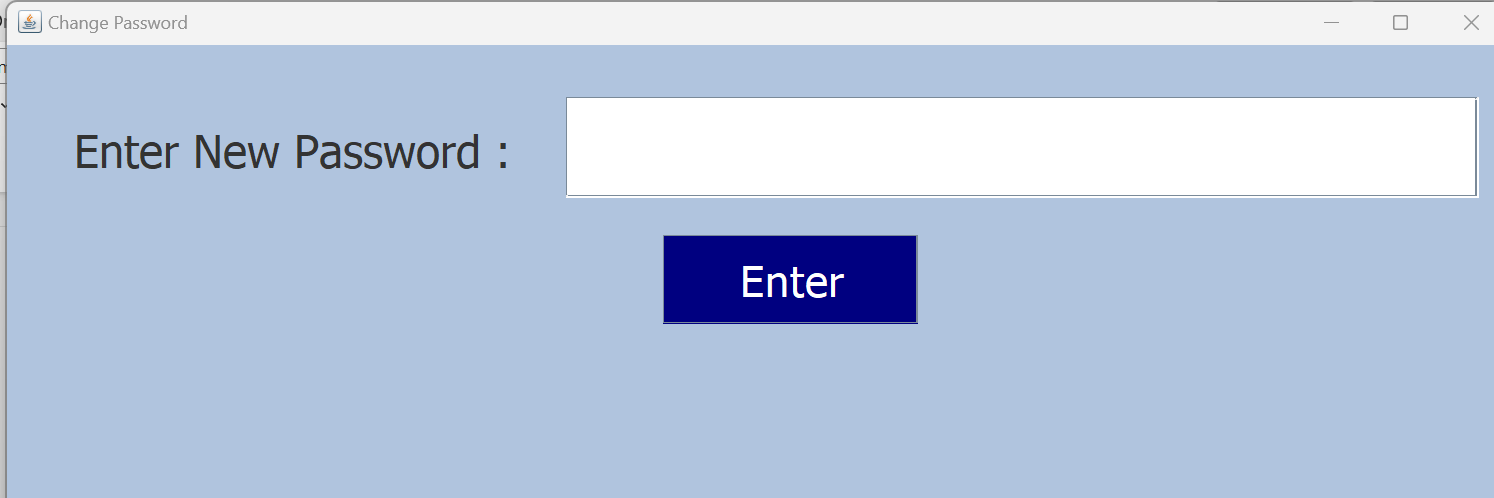
Top of Form

A screenshot of a computer

Description automatically generated

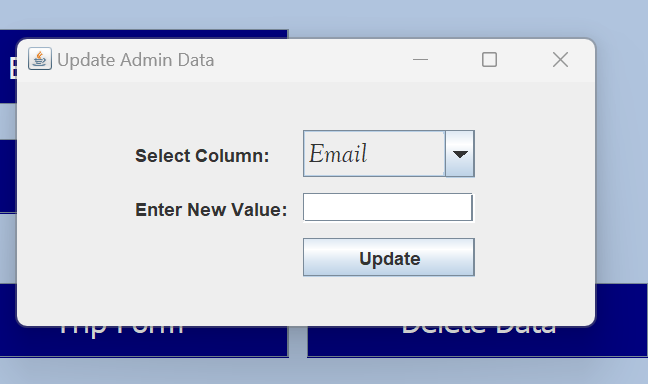
**5.Admin Change Password**

The **AdminChangePassword** class provides functionality for administrators to change their passwords within the Ambulance Booking System (ABS) application. It presents a simple user interface where administrators can enter their new password and confirm the change. Upon clicking the "Enter" button, the application updates the password in the database associated with the provided email address. If the update is successful, a confirmation message is displayed; otherwise, an error message is shown. This class enhances the security and flexibility of the ABS application by allowing administrators to manage their account passwords efficiently.



**6.Admin UpdateData Form**

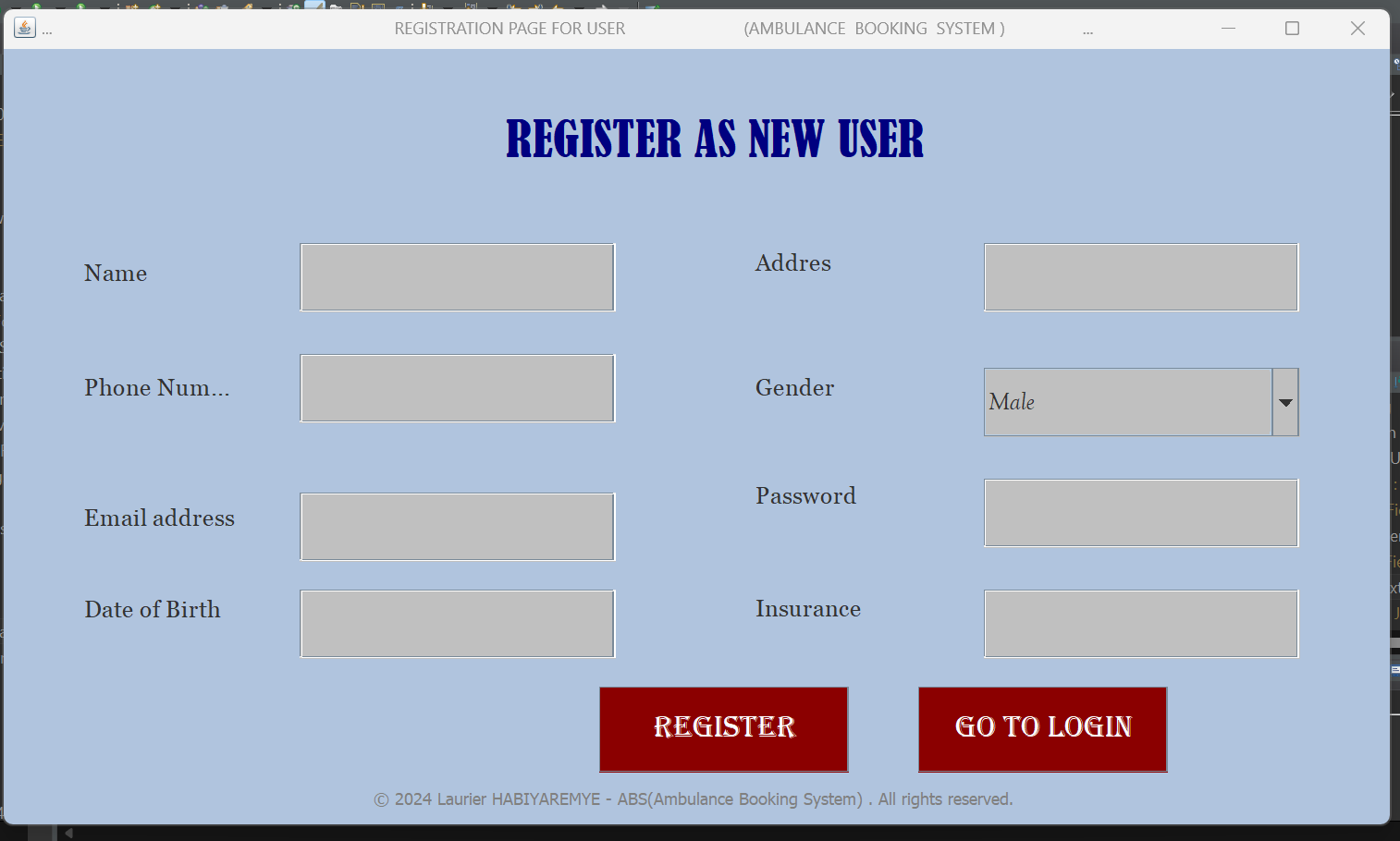
The **AdminUpdateDataForm** class facilitates the modification of administrator data within the Ambulance Booking System (ABS) application. It presents a user-friendly interface where administrators can select the column they wish to update from a dropdown menu and input the new value in a text field. Upon clicking the "Update" button, the application updates the specified column's value in the database associated with the provided email address. If the update is successful, a confirmation message is displayed; otherwise, an error message is shown. This class enhances the functionality and flexibility of the ABS application by allowing administrators to manage their data effectively.



**7.UserRegistrationForm**

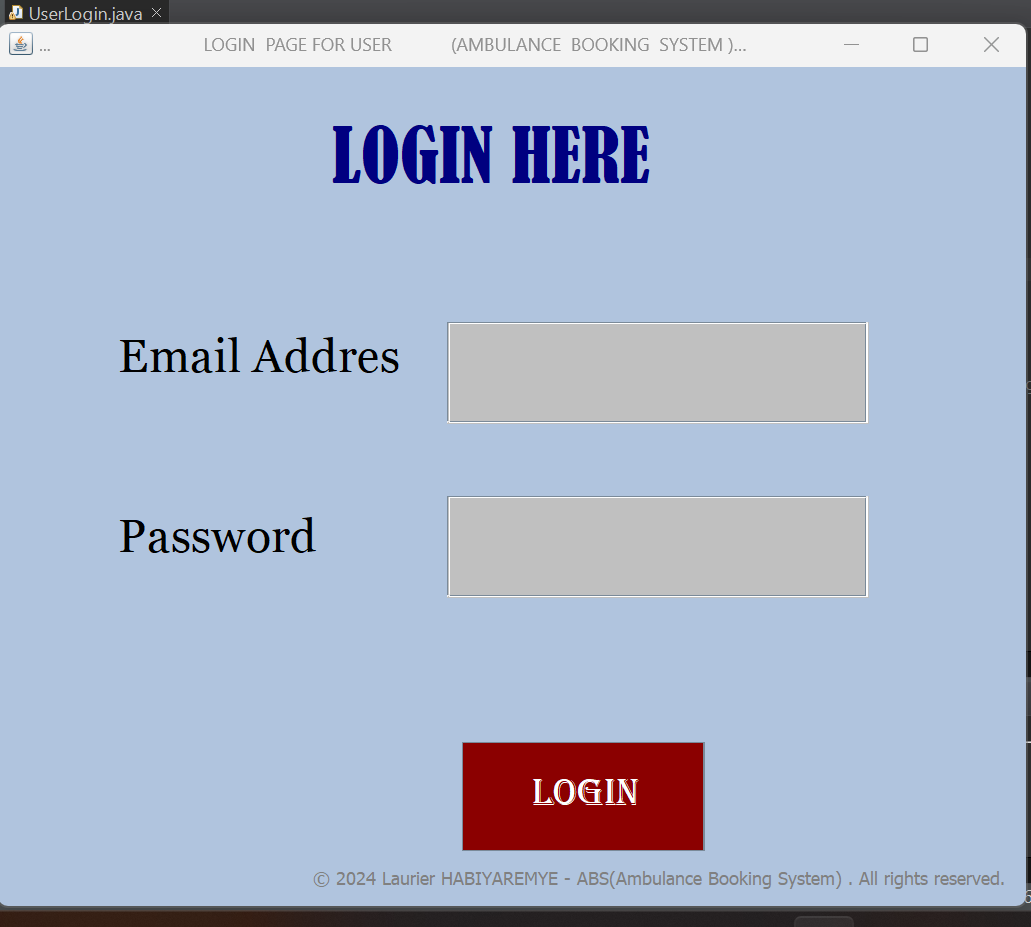
The **UserRegistration** class orchestrates the registration process for new users within the Ambulance Booking System (ABS) application. It offers a comprehensive form where users can input their personal details such as name, phone number, email address, date of birth, address, gender, insurance information, and password. The form provides options for selecting gender from a dropdown menu. Upon completion, users can click the "Register" button to submit their details, which are then stored in the ABS database. A confirmation message is displayed upon successful registration, and users can navigate to the login page using the "GO TO LOGIN" button. This class empowers users to easily join the ABS platform and access its services while ensuring their information is securely stored.

Top of Form



**8.User Login**

The **UserLogin** class facilitates the login process for users accessing the Ambulance Booking System (ABS) application. It presents a user-friendly interface where users can input their email address and password to authenticate their identity. The login form is adorned with appropriate labels and input fields for email and password. Upon clicking the "Login" button, the provided credentials are verified against the database records. If the credentials match, the user is directed to the home page (**UserHome**) of the ABS application, where they can access various features. A success message is displayed upon successful login, while an error message is shown if the credentials are incorrect. This class ensures a secure and seamless login experience for users of the ABS platform.



**9.User Home**

The **UserHome** class serves as the dashboard for users within the Ambulance Booking System (ABS) application. It provides users with various functionalities accessible through buttons, facilitating interactions with the system. Here's an overview of the features available on the user dashboard:

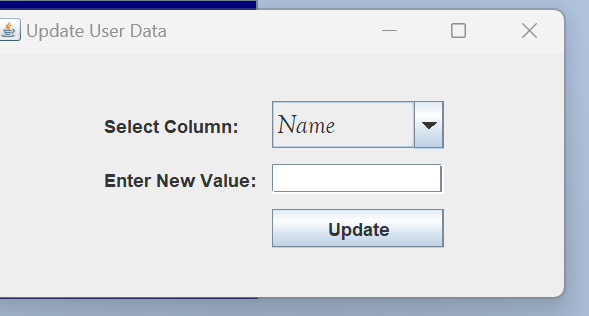
1. **View Profile Details**: Users can view their profile information such as name, email, address, date of birth, gender, insurance, and password by clicking the "Profile" button.
2. **Make Payment**: Users can initiate the payment process by clicking the "Make Payment" button, which likely redirects them to a payment form to complete the transaction.
3. **Request Trip**: Users can submit a request for an ambulance trip by clicking the "Request Trip" button, which opens a booking request form for them to fill out.
4. **Update Profile**: Users can update their profile information by clicking the "Update Profile" button, which opens a form allowing them to modify their details.
5. **Add Emergency Contact**: Users can add emergency contact information by clicking the "Add Emergency Contact" button, opening a form where they can input emergency contact details.
6. **Change Password**: Users can change their password by clicking the "Change Password" button, which likely opens a form prompting them to enter their current password and then set a new one.
7. **Logout**: Users can log out of their account by clicking the "Logout" button, which prompts them with a confirmation dialog before redirecting them to the login page (**UserLogin**).

A screenshot of a computer

Description automatically generated

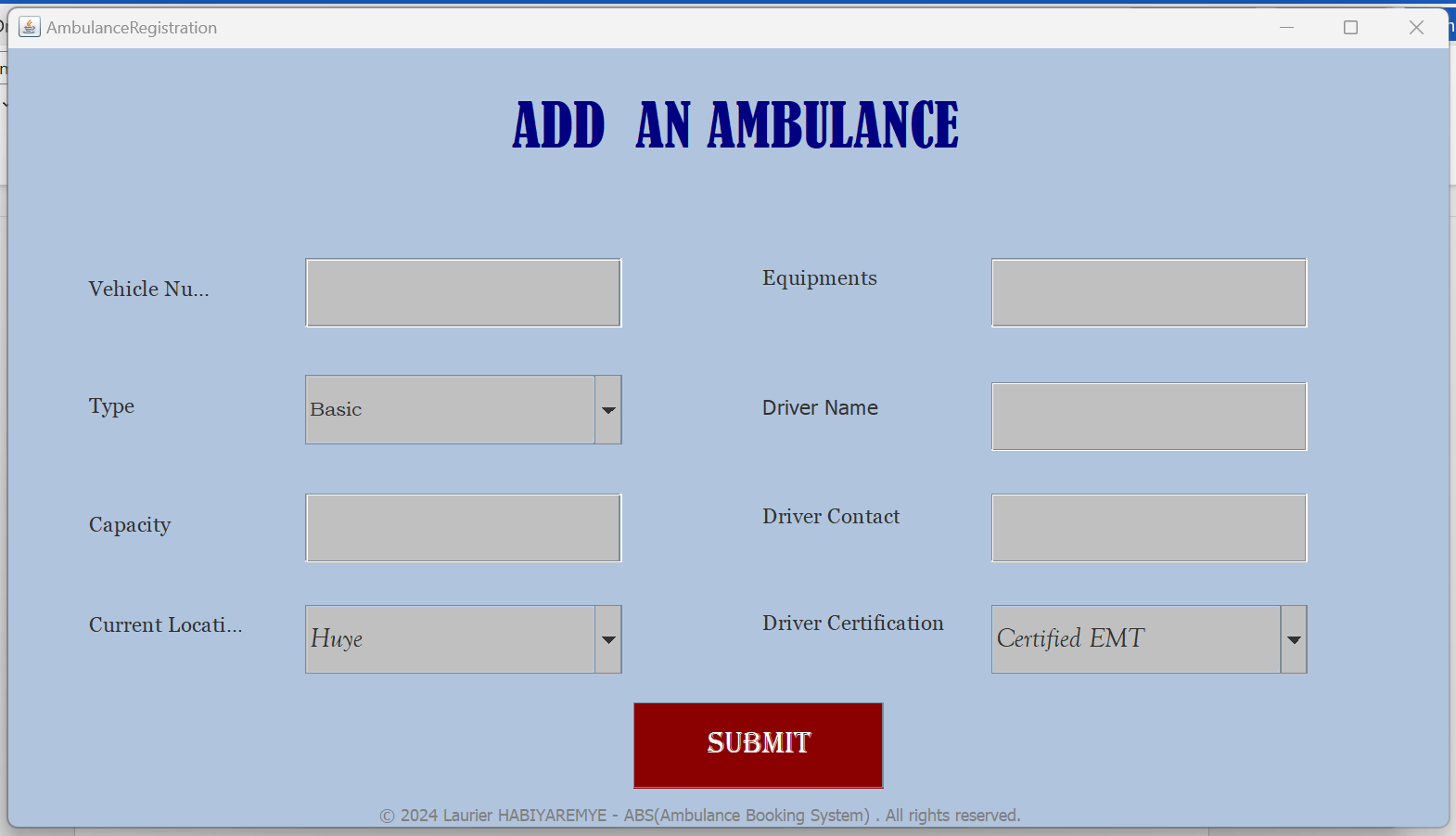
**10.User UpdateData Form**

The **UserUpdateDataForm** class provides users with a convenient interface to update their personal information within the Ambulance Booking System (ABS) application. Upon launching the form, users can select the attribute they wish to update from a dropdown menu, enter the new value in a text field, and click the "Update" button to save the changes. The form dynamically populates the dropdown menu with options such as name, email, address, date of birth, gender, insurance, and password, allowing users to choose the attribute they want to modify. Behind the scenes, the application connects to the database, executes an SQL update query to modify the specified attribute for the user identified by their email address, and provides feedback to the user regarding the success or failure of the operation through message dialogs. Once the update is completed, the form is automatically closed to provide a seamless user experience.



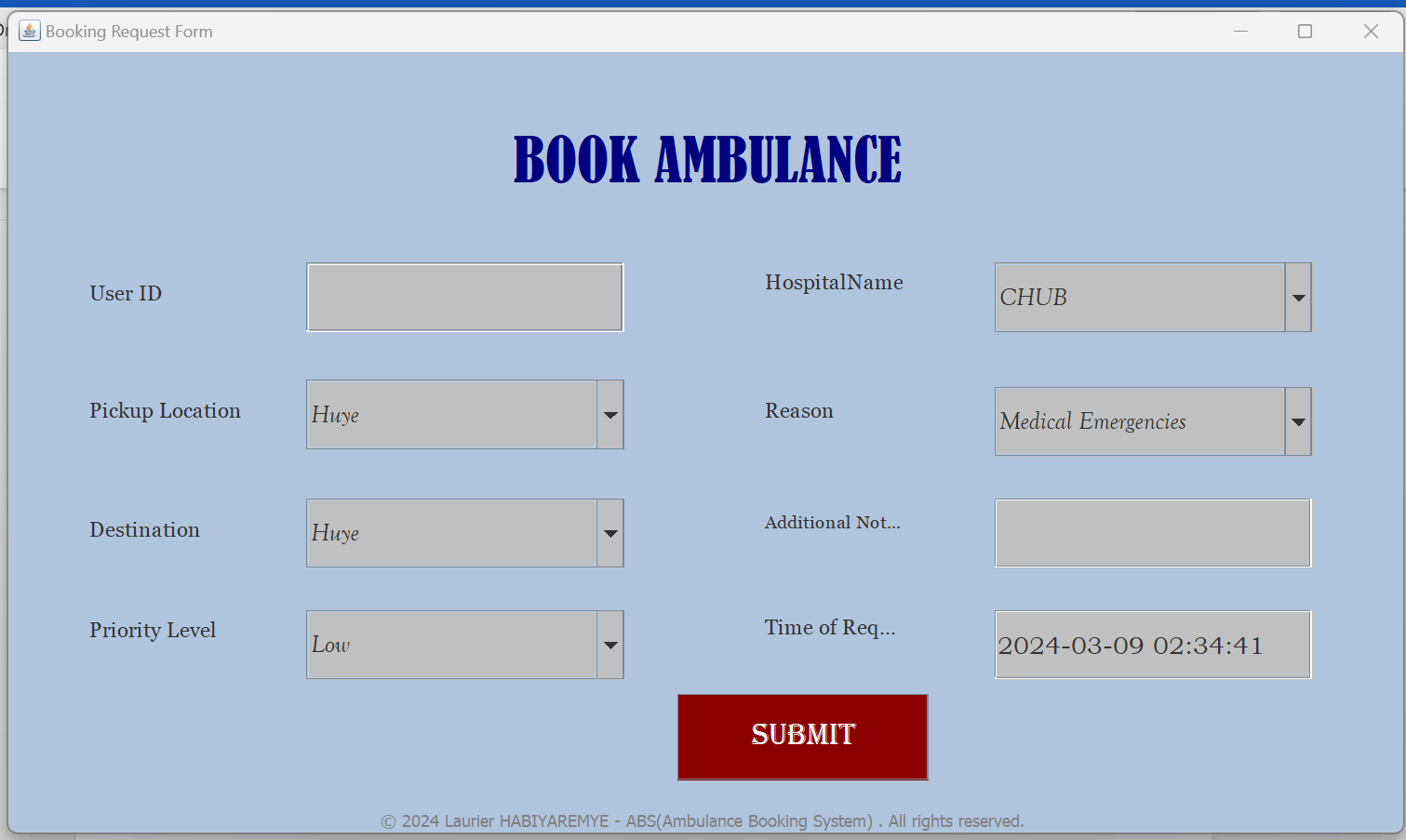
**11.Ambulance Registration**

The **AmbulanceRegistration** class facilitates the registration process for adding a new ambulance to the Ambulance Booking System (ABS). It presents users with a form containing fields to input details such as the vehicle number, type of ambulance, capacity, equipment, current location, driver name, driver contact, and driver certification. Users can select the ambulance type and driver certification from dropdown menus, while other information is entered via text fields. Upon submission, the data is validated and stored in the database. If the registration is successful, a confirmation message is displayed to the user; otherwise, an error message alerts them to any issues encountered during the registration process. The form is designed with a user-friendly interface and adheres to the system's aesthetic standards, providing a seamless experience for ambulance registration.



**12. Booking Request Form**

The **BookingRequestForm** class provides a user-friendly interface for users to request ambulance services within the Ambulance Booking System (ABS). It features input fields for the user ID, pickup location, destination, hospital name, priority level, reason for the request, additional notes, and the time of the request. Users can select the pickup location, destination, hospital name, and reason from dropdown menus, while other information is entered via text fields. The form also automatically populates the current time for the request. Upon submission, the data is validated and stored in the database. If the request is successfully processed, a confirmation message is displayed to the user; otherwise, an error message alerts them to any issues encountered during the request submission process. The form adheres to the system's aesthetic standards and provides a seamless experience for booking ambulance services.



**13.Emergency Contacts Form**

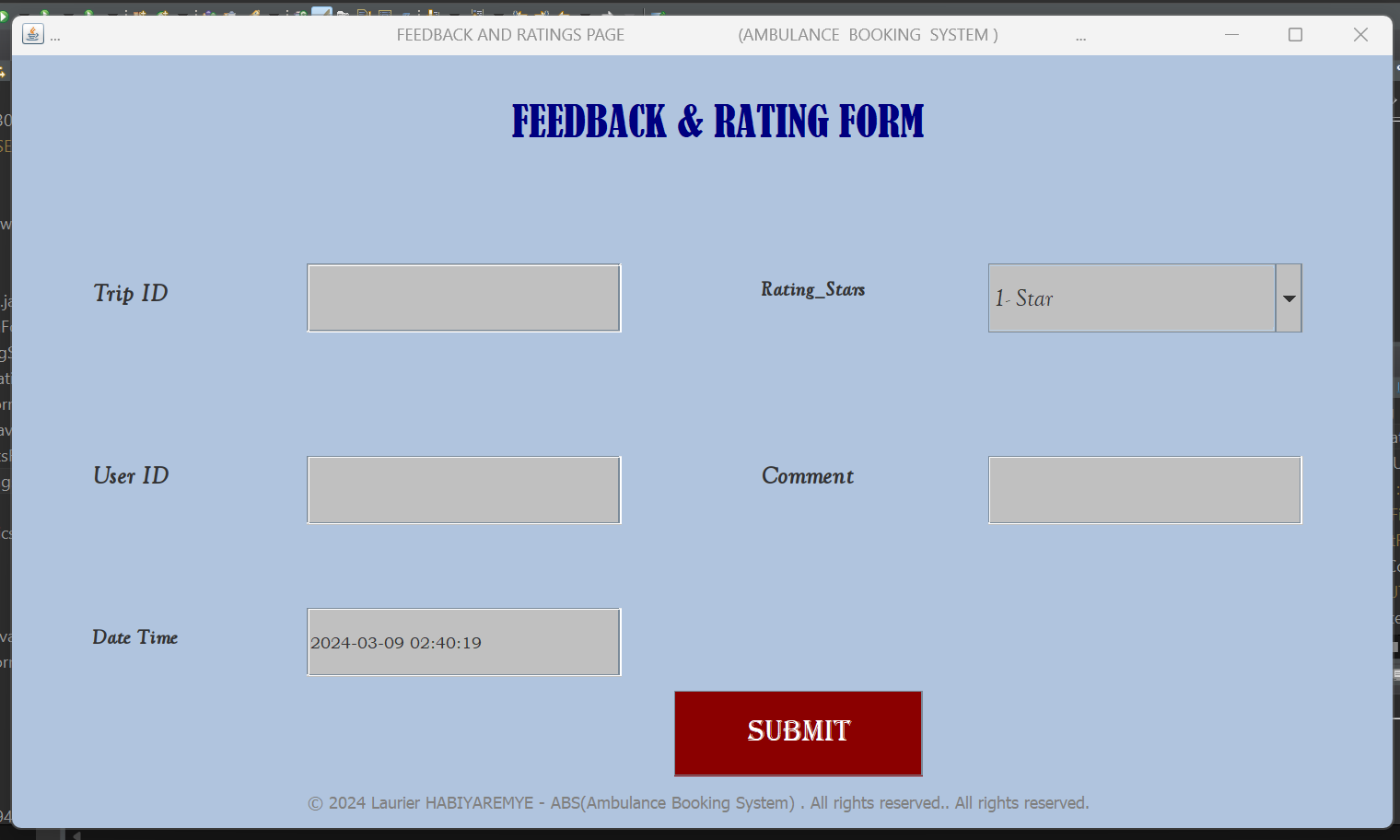
The **"EmergencyContactsForm"** class provides a streamlined interface for users to input their emergency contact information within the Ambulance Booking System (ABS). It includes fields for the user ID, contact name, relationship to the user, and contact phone number. Users can select the relationship from a dropdown menu while entering other information directly into text fields. Upon submission, the data is securely stored in the database. If the submission is successful, a confirmation message is displayed; otherwise, an error message alerts the user to any issues encountered during the submission process. The form adheres to the system's design standards and ensures a user-friendly experience for entering emergency contact details.

A screenshot of a computer

Description automatically generated

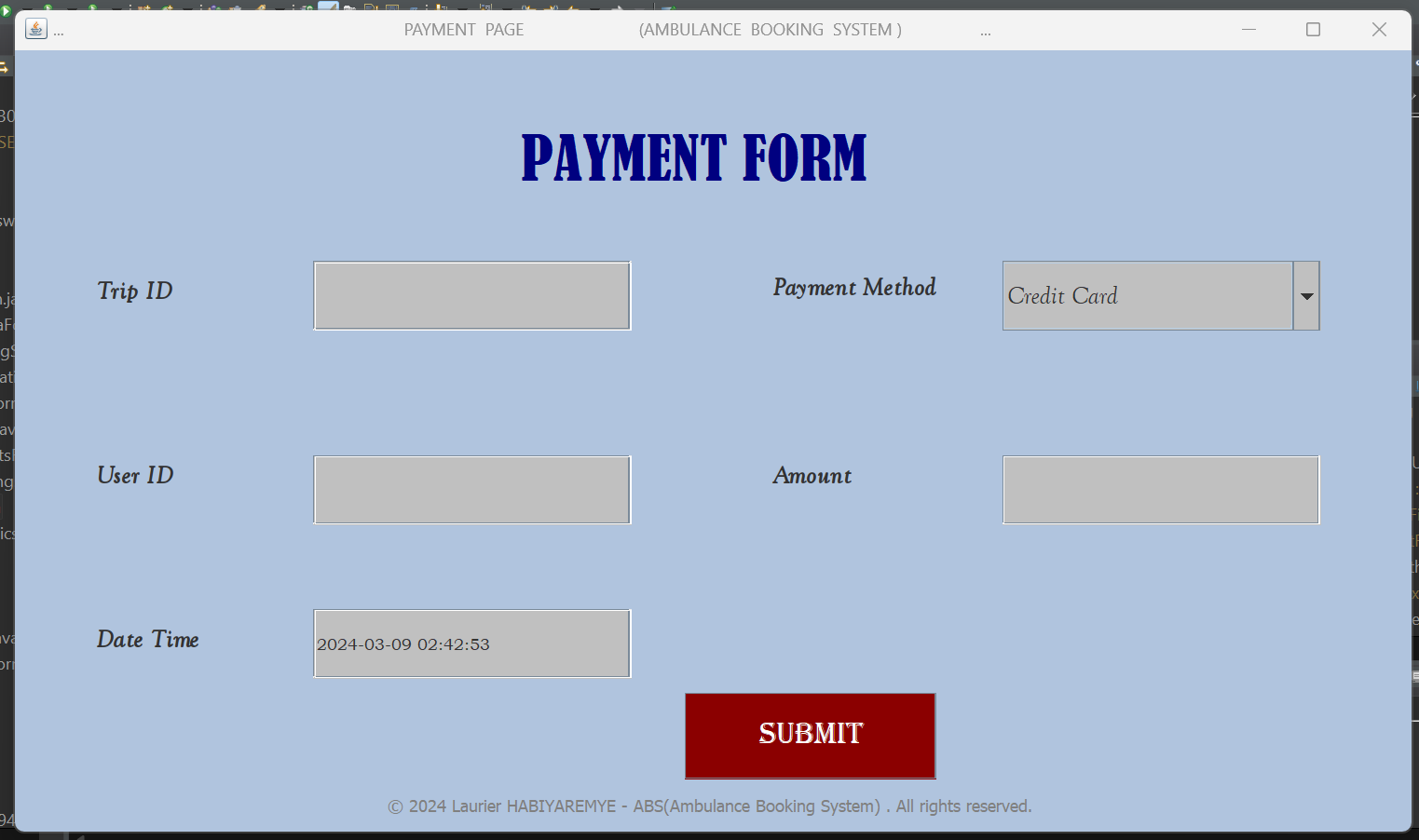
**14.Feedback And Ratings Form**

The "**FeedbackAndRatingsForm**" class facilitates the submission of feedback and ratings within the Ambulance Booking System (ABS), enhancing user engagement and service quality. It presents a structured form with fields for the trip ID, user ID, rating stars, comments, and feedback date-time. Users can select the rating stars from a dropdown menu and provide additional comments in a text field. The form automatically captures the current date-time, simplifying the feedback submission process. Upon submission, the data is securely stored in the database. If the submission is successful, a confirmation message is displayed, assuring the user that their feedback has been received and will be addressed by the ABS team. The form's design promotes ease of use and encourages users to provide valuable feedback to improve the overall service experience.



**15.Payment Form**

The **"PaymentForm"** class provides a streamlined interface for users to make payments within the Ambulance Booking System (ABS), ensuring smooth transactions and efficient service delivery. The form includes fields for the trip ID, user ID, payment method, amount, and payment date-time. Users can choose their preferred payment method from a dropdown menu and input the amount to be paid. The form also automatically captures the current date-time, simplifying the payment process for users. Upon submission, the payment details are securely stored in the database for record-keeping and transaction tracking purposes. A confirmation message is displayed to the user, acknowledging the successful payment and assuring them that the ABS team will follow up accordingly. The user-friendly design of the form encourages users to complete their payments promptly, contributing to the overall effectiveness and reliability of the ABS.



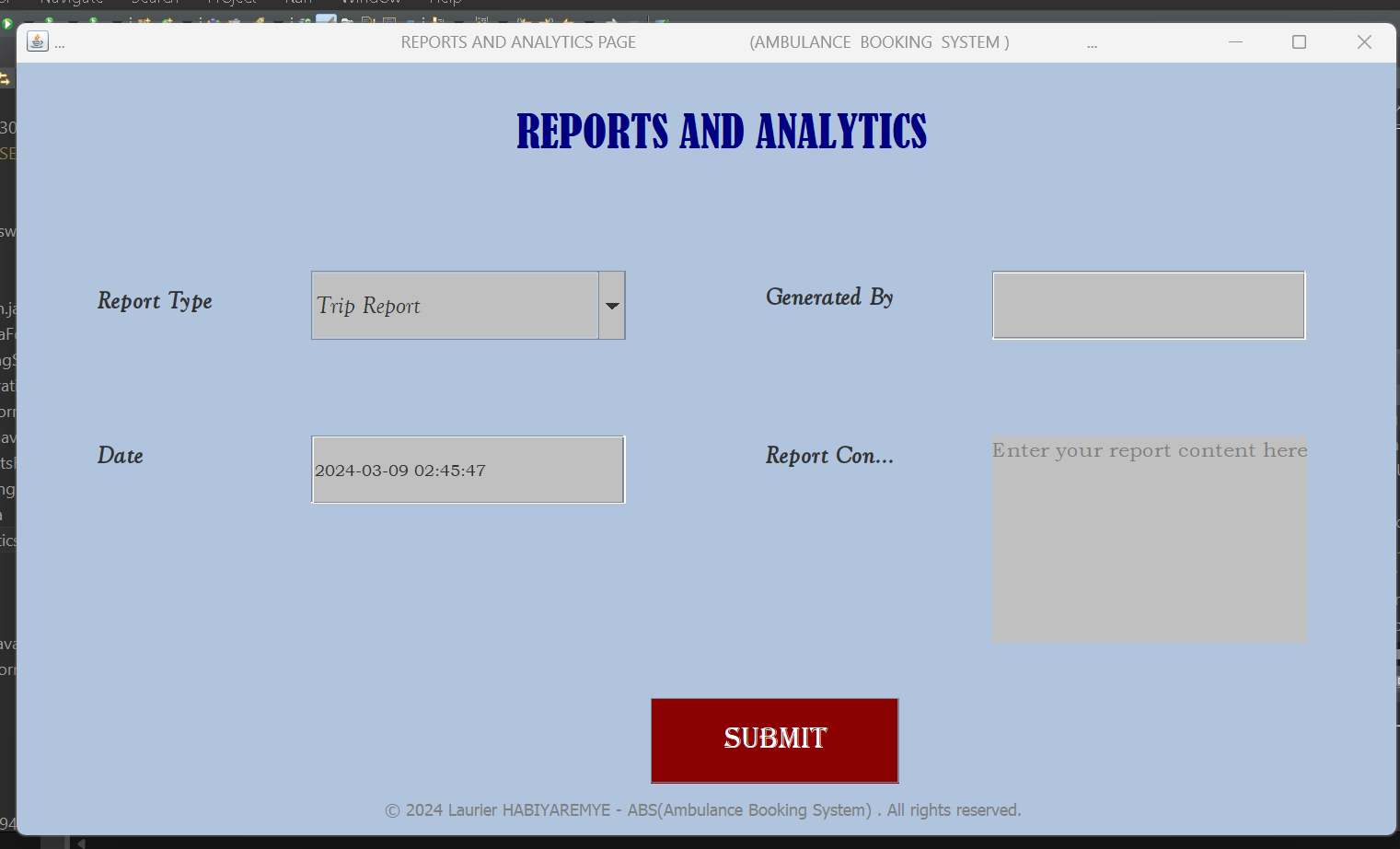
**16.Reports And Analytics FORM**

The "ReportsAndAnalyticsForm" class facilitates the generation and submission of various reports and analytics within the Ambulance Booking System (ABS), enhancing data-driven decision-making and performance monitoring. Users can select the type of report from a dropdown menu, such as trip report, ambulance report, payment report, system analytics report, incident report, monthly/yearly report, or compliance report. Additionally, users can specify the date range for the report and the entity generating the report.

The form includes a text area where users can enter the content of the report. To provide a user-friendly experience, the text area initially displays a prompt instructing users to enter their report content. When the text area gains focus, the prompt disappears, allowing users to input their report content seamlessly. Moreover, the form automatically populates the date range field with the current date and time, simplifying the process of specifying the report's timeframe.

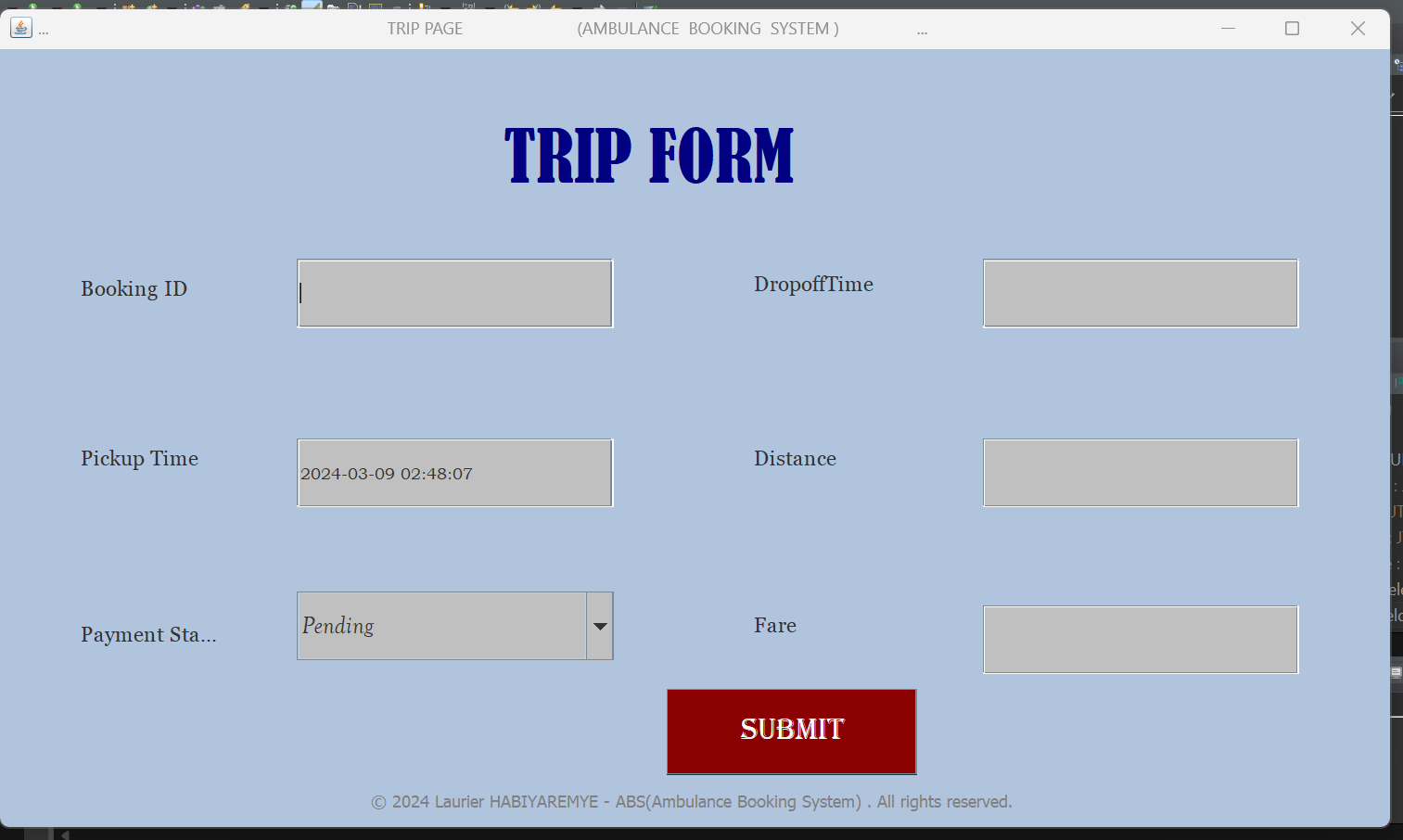
Upon submission, the report details, including the report type, date range, entity generating the report, and report content, are securely stored in the database. Users receive a confirmation message indicating the successful submission of the report, ensuring transparency and accountability in the reporting process. This robust reporting mechanism empowers ABS stakeholders to gain valuable insights, monitor key metrics, and make informed decisions to optimize service delivery and operational efficiency.

Top of Form



**17.Trip Form**

The "**TripForm**" class provides a user-friendly interface for entering trip details within the Ambulance Booking System (ABS). Users can input essential trip information such as the booking ID, pickup time, dropoff time, distance traveled, fare, and payment status. The form includes dropdown menus for selecting the payment status, which can be pending, canceled, or rebuilt, enhancing flexibility and accuracy in trip management. Additionally, the form automatically populates the pickup time field with the current date and time, streamlining the process of recording trip details. Upon submission, trip details are securely stored in the database, and users receive a confirmation message indicating the successful creation of a new trip. This robust trip management system ensures efficient handling of ambulance bookings, contributing to the seamless delivery of emergency medical services.



* 1. SPECIAL BUTTONS

**1.View Data Button**

A blue sign with white text

Description automatically generated

The **"View Data"** button in the admin user home interface provides administrators with a quick and convenient way to access detailed information about users, bookings, or other essential data within the system. Upon clicking the "View" button, administrators can navigate to dedicated pages or modal windows displaying comprehensive views of the selected data, facilitating efficient monitoring and analysis of system activities. This feature empowers administrators to make informed decisions and take appropriate actions based on real-time insights, ultimately enhancing the overall management and effectiveness of the system.

A screenshot of a computer

Description automatically generated

**2.Update Button**

**A blue sign with white text

Description automatically generated**

The **"Update"** button serves as a valuable tool for administrators to modify existing user profiles, booking details, or system configurations with ease. By clicking the "Update" button, administrators gain access to intuitive and user-friendly interfaces where they can make necessary adjustments to the selected information. Whether it involves updating user contact information, modifying booking schedules, or adjusting system settings, the "Update" button streamlines the process of maintaining accurate and up-to-date data within the system. Administrators can leverage this functionality to ensure that the system reflects the most current and relevant information, thereby optimizing operational efficiency and user satisfaction.

A screenshot of a computer

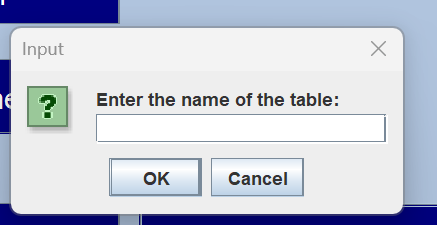
Description automatically generated

**3.Delete Button**

**A blue sign with white text

Description automatically generated**

The **"Delete"** button in the admin user home interface empowers administrators to remove unwanted or obsolete data from the system efficiently. By clicking the "Delete" button, administrators can initiate the deletion process for selected records, such as user accounts, bookings, or other entities. This feature typically prompts administrators to confirm their intention to delete the chosen data, helping to prevent accidental removals and ensure data integrity. Upon confirmation, the system executes the deletion operation, permanently removing the specified records from the database. The "Delete" button thus provides administrators with a powerful tool to maintain a clutter-free and organized system environment, promoting streamlined operations and improved data management practices.

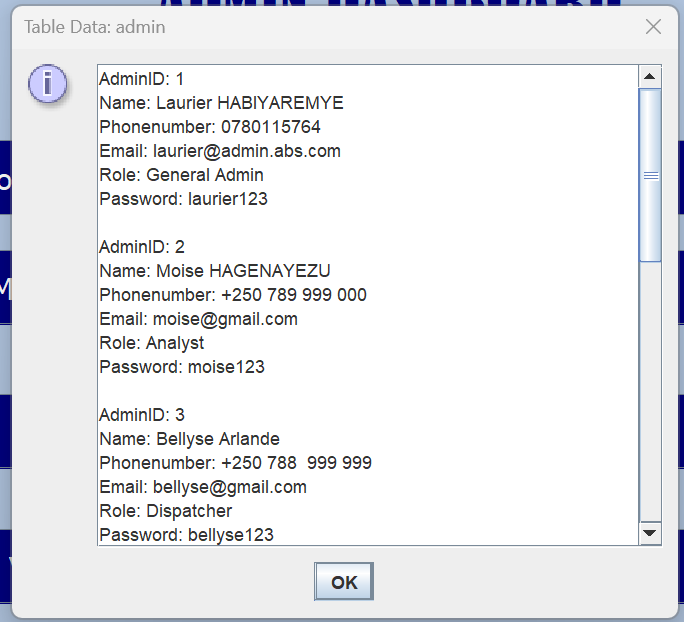
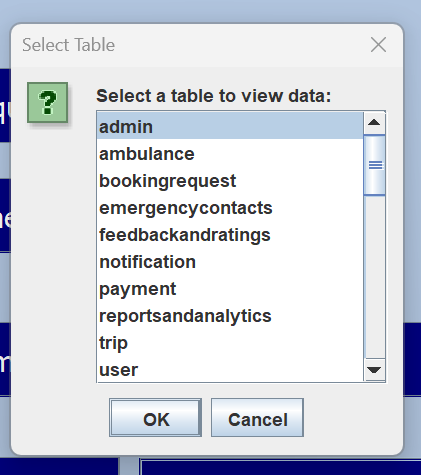


**4.View Database Button**

**A blue sign with white text

Description automatically generated**

The **"View Database"** button offers administrators a comprehensive overview of the entire database or specific datasets within the system. By clicking the "View Table" button, administrators can access structured tables displaying various data attributes, including user details, booking information, or system logs. This functionality typically presents data in a tabular format, facilitating easy navigation, sorting, and filtering of records based on specific criteria. Administrators can utilize the "View Table" feature to conduct in-depth analyses, track trends, and identify patterns within the dataset, enabling informed decision-making and strategic planning. Additionally, this functionality may include options to export data for further analysis or reporting purposes, enhancing the system's usability and versatility for administrators.



3.5 CONCLUSION

In conclusion, the Ambulance Booking System represents a comprehensive solution tailored to streamline emergency medical transportation logistics. By leveraging modern technology and user-friendly interfaces, the system ensures swift and efficient ambulance dispatch, improving response times and ultimately saving lives. With features such as trip scheduling, user management, payment processing, and comprehensive reporting, the system caters to the diverse needs of both administrators and end-users. Its robust architecture, coupled with advanced functionalities, enhances the overall efficiency of ambulance services while maintaining data integrity and security. As an indispensable tool for healthcare organizations and emergency responders, the Ambulance Booking System stands as a testament to the transformative power of technology in optimizing critical healthcare services.