CONTEST-AZM
تو نشانِ عزمِ عالیشان

Aptech
Unleash your potential

Software Requirements Specification

Version 1.0

Theme: eAmbulance

Project Name: Rapid Rescue

Category: Web Application Development

Table of Contents

1.1 Background and Necessity for a Web Application

An Ambulance Booking System Application is a critical solution designed to streamline and optimize the process of requesting and dispatching ambulance services in times of medical emergencies. This innovative application leverages technology to provide rapid, efficient, and life-saving assistance to individuals whenever necessary. It offers a user-friendly interface for users to request ambulance services swiftly, connecting them with nearby medical facilities and emergency response teams. The Web application not only enhances response times but also ensures the timely and effective deployment of ambulances, ultimately contributing to improved healthcare outcomes and public safety.

1.2 Proposed Solution

The proposed solution is a Web Application called **Rapid Rescue** that allows users to quickly request ambulance services, track the ambulance's location in real-time, and provide essential medical information to Emergency Medical Technicians (EMTs) before they arrive at the scene.

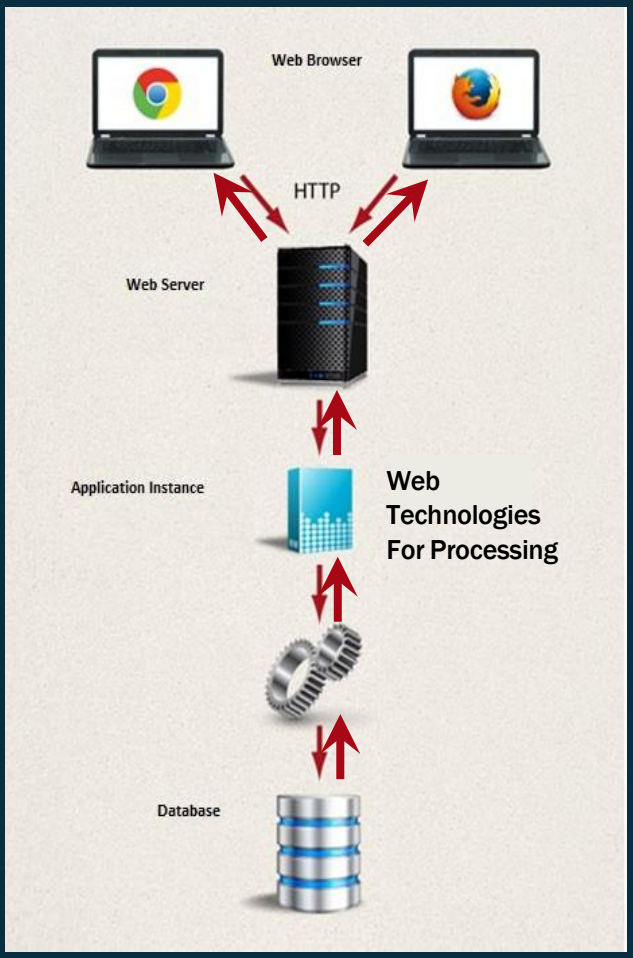
1.3 Purpose of the Document

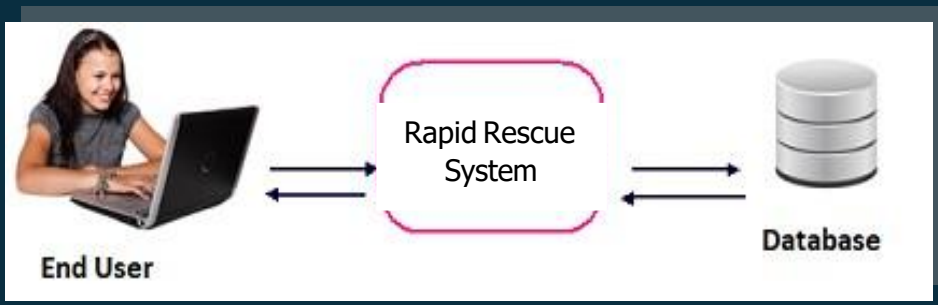
This document presents a detailed description of the **Rapid Rescue** Web application, explaining its features, purpose, scope, and limitations. It is intended for both stakeholders and developers of the application.

1.4 Scope of the Project

The **Rapid Rescue** Web application will be a responsive and user-friendly platform for individuals and emergency services. Users can request ambulances, track their location, provide medical information, and receive basic first-aid instructions. Dispatchers can manage and assign ambulances, while EMTs can access patient information and update their status. The application will not handle billing or insurance claims, as these are beyond the scope of this system.

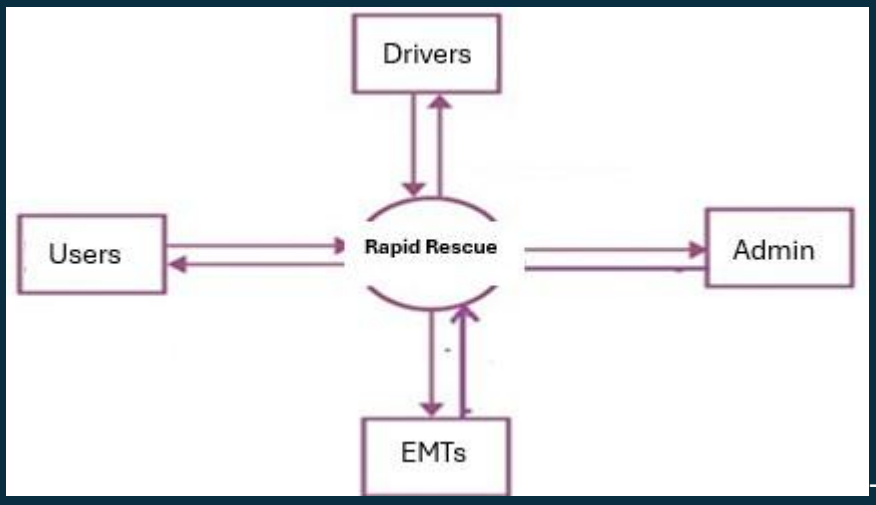
Architecture Diagram for Rapid Rescue





Flow diagrams

A sample flow diagram depicting the interaction between various entities and the application is shown here:



1.5 Constraints

The development of the Rapid Rescue Web application must adhere to several constraints to ensure its successful implementation and operation. Technically, the application must be compatible with major Web browsers and responsive across various devices. You may encounter constraints related to data storage, data synchronization, and backup procedures. The usage of various images and videos may be subject to licensing agreements and copyright restrictions. It is important to understand and comply with these constraints to avoid legal issues.

1.6 Functional Requirements

Application will be designed with a set of forms/pages with menus representing the choice of activities to be performed.

For Users:

- **Home Page:** Provide quick access to emergency services and information.
- **Account Registration:** Allow users to create accounts with Email Id and Password.
- **Profile Management:** Users can modify their Profile Information and implement password change.

- **Emergency Request**: It is a new booking facility by providing the hospital name, address, customer mobile number, pickup address, and selecting the type (Emergency/Non-Emergency).
- **Real-time Tracking**: Show the dispatched ambulance's location and estimated arrival time.
- **Medical Profile**: Allow users to input and update their medical history, allergies, and emergency contacts.
- **First-Aid Instructions**: Provide basic first-aid guidance while waiting for the ambulance.
- **Feedback**: Allow users to submit feedback on the service received.

For Admin:

- **Login:** Login with secure access to the dispatch system.
- **Ambulance Management**: Add, modify, or remove ambulances from the fleet.
- **Driver Profiles**: Add, Modify, and Delete Driver details.
- **Dispatch Control**: Assign and track ambulances for emergency requests.
- **Real-time Monitoring**: View all active emergency requests and ambulance locations.
- **Communication:** Send messages to EMTs and update users on status changes.

For Driver/EMTs:

- **Login**: Login with secure access to the system.
- **Patient Information**: Access to the patient's medical profile and emergency details.
- **Status Updates**: Ability to update their status (en-route, arrived, or transporting patient).

Common Features:

Implement GPS functionality as applicable.

- **Search/Filter**: Enable searching and filtering of emergency requests, ambulances, or patient records.
- **Notifications:** Send alerts for new requests, status changes, or important updates.

- About Us: Description of services offered by the app.
- Image Gallery: Gallery showcasing different types of ambulances.
- Contact Us: A form to accept user queries with fields for name, email, contact number, and message.
- Ambulance Type: List of ambulance types, sizes, and equipment.
- Feedback Form: A form for users to provide feedback.
- Ambulance Costs: Information on the costs associated with ambulance services.
- Driver's List: A hardcoded list of drivers, their locations, and contact information.

Note: Boilerplate or readymade HTML template can be used, provided it is only for design aspect and not for implementing application functionality.

1.7 Non-Functional Requirements

There are several non-functional requirements that should be fulfilled by the application. They include:

- **Safe to use**: The application should not result in any malicious downloads or unnecessary file downloads.
- **Accessibility**: The application should have clear and legible fonts, user-interface elements, and navigation elements.
- **User-friendliness**: The application should be easy to navigate with clear menus and other elements and easy to understand.
- **Operability**: The application should be reliable and efficient.
- **Performance**: The application should demonstrate high value of performance through speed and throughput. In simple terms, the application should have minimal load time and smooth page redirection.
- **Scalability**: The application architecture and infrastructure should be designed to handle increasing user traffic, data storage, and feature expansions.
- **Security**: The application should implement adequate security measures such as authentication. For example, only registered users can access certain features.
- **Availability**: The application should be available 24/7 with minimum downtime.
- **Compatibility**: The application should be compatible with latest browsers and various devices.

These are the bare minimum expectations from the project. It is a must to implement the FUNCTIONAL and NON-FUNCTIONAL requirements given in this SRS. Once they are complete, you can use your own creativity and imagination to add more features if required.

1.8 Interface Requirements

Hardware

- Intel Core i5/i7 Processor or higher
- 8 GB RAM or higher
- Color SVGA
- 500 GB Hard Disk space
- Mouse
- Keyboard

Software

IDE:

- Appropriate IDE as per the platform

Frontend:

- HTML5, CSS3, Bootstrap (optional), ReactJS 18 or higher/AngularJS/Angular 9 or higher, JavaScript, jQuery, and XML

Backend:

- Java 16 or higher, Java EE 7 or higher/Jakarta EE 9 or higher, with Apache NetBeans IDE/Eclipse latest version, Apache Tomcat 10.0 or higher, Glassfish 6.0 or higher, and related libraries

OR

- C# 7.2 with ASP.NET MVC and ASP.NET MVC Core (optional), Visual Studio 2019 or higher

OR

- PHP 7.0 or higher version with Apache Web Server and Laravel Framework Homestead (optional)

OR

- Python 3.0 or higher with PyCharm IDE, Django/Flask framework

For local hosting (optional):
XAMPP latest version

Database:

- MySQL 8.0 or higher/SQL Server 2019 or higher

Maps and Geolocation:

- Google Maps API or OpenStreetMap for location services

Database Design

Based on the given specifications, you will define suitable entities, attributes for these entities, and identify relationships between the entities. For example, some entities along with their attributes can be identified as follows:

Users
userid (primary key)
firstname
lastname
email
phonenumber
password
date_of_birth
address

Emergency Requests
requestid (primary key)
userid (foreign key referencing users)
request_time
status (pending, completed, or cancelled)

Ambulances
ambulanceid (primary key)
vehicle_number
equipment level (basic, advanced)
current_status (available, on call, or maintenance)

EMT
emtid (primary key)
firstname
lastname
certification
phonenumber
email

Drivers
driverid (primary key)
firstname
lastname
phonenumber

Similarly, you can define other entities and relationships between entities and methods representing activities on the entities.

Note: These are just examples, you do not have to adhere to these structures and can design your own table structure with different columns.

1.G Project Deliverables

You will require to design and build the project and submit it along with a complete project report that includes:

- Problem Definition
- Design Specifications
- Diagrams such as Flowcharts for various Activities, Data Flow Diagrams, and so on
- Database Design
- Test Data Used in the Project
- Project Installation Instructions (if any)
- **User Credentials for all Types of Users with Passwords**

Documentation is considered as a very important part of the project. Ensure that documentation is complete and comprehensive.

Documentation should not contain any source code.

The consolidated project will be submitted as a zip file with a ReadMe.doc file listing assumptions (if any) made at your end and SQL scripts files (.sql) containing database and table definitions.

Note: Preferably, host the working Web application on a Website and share the URL for evaluation.

Submit a video (.mp4 file) demonstrating the working of the Web application, including all features under Functional Requirements. This is MANDATORY.

Over and above the given specifications, you can apply your creativity and logic to improve the system.

Sitemap: To understand the flow of Rapid Rescue Web Application, you will have to create a Sitemap and add it to the home page of your application.

****