Software Requirements Specification

for

ClassRoom And Cycle Key Management System

Version: 1.0

Prepared by

Group Number: 4

Gedala Vamsi Vihari	**B220286CS**	**gedala_b220286cs@nitc.ac.in**
Chilekampalli Reddy Sekhar Reddy	**B220806CS**	**chilekampallli_b220806cs@nitc.ac.in**
Meragala Sujanesh Nandan	**B220389CS**	**meragala_b220389cs@nitc.ac.in**

Instructor:

Course: Software Engineering Lab

Date:

Contents

1 Introduction

1.1 Purpose

The purpose of this Software Requirements Specification (SRS) document is to provide a comprehensive and detailed outline of the software requirements for the Blood Bank Management System. It serves as a reference and guide for all stakeholders involved in the project, including developers, testers, project managers, and end-users. This document aims to establish a common understanding of the system's objectives, scope, and functionality, ensuring that all participants are aligned in their expectations and goals. Additionally, it acts as a foundation for the design, development, testing, and successful implementation of the Blood Bank Management System, enabling efficient blood bank operations and contributing to the enhancement of healthcare services

1.2 Product Scope

The **Classroom Key and Cycle Management System** is a software platform designed to address the inefficiencies associated with managing physical keys for classrooms and cycle-sharing within educational institutions. By automating the processes of requesting, transferring, and monitoring the use of keys and cycles, the system enhances transparency, accountability, and efficiency for students, Class Representatives (CRs), and administrators.

The system is intended to reduce the administrative burden on staff and provide users with an intuitive, user-friendly interface to manage resource requests. It also integrates automated notifications, approval workflows, and penalties to ensure proper usage of resources like cycles and classroom keys.

1.3 Intended Audience and Document Overview

Intended Audience:
The Blood Bank Management System Software Requirements Specification (SRS) is intended for a diverse group of stakeholders involved in the project. The primary audience includes:
● **Developers**: Those responsible for designing, implementing, and testing the Blood Bank Management System.
● **Project Managers**: Individuals responsible for overseeing the project's progress, resource allocation, and adherence to timelines.
● **Students (End-Users):** The general student body who will use the system to request and transfer keys and cycles, manage bookings, and view resource availability.
● **Class Representatives (CRs):** CRs who will handle classroom key management and have specific permissions to book classrooms and access keys, as well as manage cycle requests.
● **Administrators:** Individuals responsible for managing the system’s configuration, including user access, resource allocation, and overseeing all key and cycle transactions.
● **Tester**s: Individuals responsible for quality assurance and system testing.

● **Documentation Writers**: Authors responsible for creating user manuals, training materials, and user guides.
● **Client (Educational Institution):** The commissioning entity (school, college, or university) that requires the development and implementation of the Classroom Key and Cycle Management System.
● **Professor**: The academic instructor or course examiner responsible for evaluating the project.

This SRS document provides a comprehensive description of the functional and non-functional requirements for the Classroom Key and Cycle Management System. It is structured to allow each type of audience to easily find relevant sections of interest, including technical details for developers and project managers, as well as user-centric information for students, CRs, and administrators.

1.4 Definitions, Acronyms and Abbreviations

This section clarifies and standardizes the terminology used throughout the SRS document to ensure all stakeholders share a common understanding of key terms, acronyms, and abbreviations relevant to the **Classroom Key and Cycle Management System**.

S.No	Abbreviation/Term	Definition(s)
1	CR(Class Representative)	A student elected to represent their class, responsible for managing classroom keys and bookings
3	RDBMS	Relational Database Management System, such as MySQL, used to store and manage data.
4	SRS	Software Requirements Specification, the document you are currently reading.
5	GUI	Graphical User Interface, the visual components and interactions that users will employ to interact with the system.
6	API	Application Programming Interface, used for integrating with external systems or data sources.
7	User	Any person or entity interacting with the Blood Bank Management System, including donors, blood bank staff, and administrators.
8	Authentication	The process of verifying the identity of users before granting access to the system.
9	Authorization	The process of specifying what actions or resources users are allowed to access.
10	Classroom Key	A physical key that grants access to a classroom, managed by the system.
11	Cycle	A physical bicycle available for booking and borrowing through the system.

12	Cycle Drop Point	A designated location on campus where cycles can be picked up or returned after use.
13	QR Code	A machine-readable code that students will scan to access cycles or record cycle transfers.
14	Booking	The act of reserving a classroom key or cycle for a specific time period.
15	Role-Based Access Control (RBAC)	A security model that assigns permissions based on the roles (e.g., student, CR, admin) of users.
16	Notification	Automated alerts sent to users regarding bookings, approvals, returns, and system updates.
17	Pre-Booking	The feature allowing CRs to book classroom keys in advance for scheduled events or classes.
18	Real-Time Monitoring	The ability for administrators to view current key and cycle transactions as they occur.

1.5 Document Conventions

● **Font**: The document uses Arial font with the following font sizes for various textual content:
● **Main Headings**: Font size 14
● **Subheading**s: Font size 12
● **Sub-subheadings and Paragraphs**: Font size 12 The SRS document aims to provide a clear and standardized presentation of information in accordance with the IEEE formatting requirements, ensuring readability and accessibility for all stakeholders involved in the Blood Bank Management System project

1.6 References and Acknowledgments

1. *IEEE Software Engineering Standards Committee, "IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications", October 20, 1998.*

2 Overall Description

2.1 Product Overview

*The **Classroom Key and Cycle Management System (CKCMS)** is a software solution designed to automate and streamline the processes involved in managing classroom keys and cycle-sharing within an educational institution. It addresses inefficiencies related to manual handling of these resources, ensuring transparency, accountability, and timely access. The system allows students, Class Representatives (CRs), and administrators to easily request, transfer, and monitor the use of classroom keys and cycles in real time.*
By introducing a seamless request and approval mechanism, CKCMS improves resource accessibility and reduces the likelihood of misuse or delays. The system uses secure authentication to protect resource access, and automated notifications keep users informed of their bookings, approvals, and pending returns. CKCMS also supports cycle management through QR code scanning at designated pickup and drop points, enhancing user convenience and ensuring accurate tracking.

2.2 Product Functionality

The product functionality of the ***Classroom Key and Cycle Management System*** includes:

- **Classroom Key Management:** CRs can request, book, and manage classroom keys, ensuring authorized access.
- **Pre-Booking for Keys**: Supports scheduling classroom key bookings to enhance access efficiency.
- **Cycle Management**: Students can reserve, borrow, and return cycles at designated stations using QR codes.
- **Cycle Availability Tracking**: Monitors cycle availability, streamlining pick-up and drop-off.
- **Approval System**: Transfers of keys and cycles require approval from current holders, ensuring accountability.
- **Real-Time Monitoring**: Admins track key and cycle transactions in real-time for better oversight.
- **Automated Notifications**: Sends alerts for bookings, approvals, returns, and late penalties to keep users informed.
- **Cycle Maintenance Feedback**: Students report maintenance issues, allowing admins to flag cycles for servicing.
- **Role-Based Access Control (RBAC)**: Different access rights for CRs, students, and admins ensure secure operations.
- **Data Security**: Enforces authentication and authorization to prevent unauthorized access.
- **Detailed Usage Logs**: Tracks all key and cycle transactions, providing insights and ensuring accountability.
- **Penalty System**: Automatically applies fines or warnings for late cycle returns to encourage timely usage.

2.3 Design and implementation constraints

Design and implementation constraints for the Classroom and Cycle Key Manageent System include:

- **Platform Compatibility:** The system needs to work seamlessly on multiple platforms (Windows, macOS, Linux) to accommodate diverse user environments.
- **Reliability and Availability:** The system must maintain a high level of reliability with at least 99.9% uptime, and scheduled maintenance activities should be communicated to users in advance.
- **Data Privacy Regulations:** Compliance with data privacy regulations (e.g., GDPR, HIPAA) is essential to protect donor and recipient data, which imposes constraints on data handling and storage.
- **Resource Limitations:** The system should be designed to operate efficiently with standard hardware and not require specific or high-end hardware components.
- **Scalability:** The architecture must support scalability to accommodate a growing number of users, donors, and blood units without compromising performance.
- **Integration Compatibility:** Integration with external systems or databases, such as RDBMS, should be seamless to ensure interoperability.
- **Legacy System Integration:** In some cases, integration with existing legacy systems may be necessary, which poses constraints on data mapping and communication protocols.
- **Usability Constraints:** The system's design should prioritize usability, providing an intuitive interface for users of varying technical proficiencies.
- **Regulatory Compliance:** Compliance with healthcare regulations and standards is crucial for blood bank management, which may necessitate specific design considerations and constraints.

2.4 Assumptions and Dependencies

Assumptions

- ***User Roles****: It is assumed that all users (students, CRs, and administrators) will adhere to their assigned roles and responsibilities when interacting with the system.*
- ***Reliable Internet Access*** *The project assumes that users will have a stable internet connection to access the system for booking cycles or classroom keys.*
- ***Availability of QR Code Scanners:*** *It is assumed that students will have access to devices with QR code scanning capabilities to facilitate cycle borrowing and returns*
- ***Key and Cycle Availability****: It is assumed that keys and cycles will be available and maintained in good condition for use, with proper tracking mechanisms in place.*
- ***User Training:*** *It is assumed that users will receive adequate training to navigate the system and understand the procedures for key and cycle bookings.*
- ***CR and Admin Engagement:*** *The assumption is that CRs and administrators will actively participate in managing classroom keys and overseeing cycle bookings.*

Dependencies

- ***External Integration****: The system depends on the successful integration with external systems and databases, such as those of hospitals or healthcare facilities, for the exchange of donor and medical information.*
- ***Database Availability****: The project relies on the continuous availability and reliability of the underlying database management system (e.g., MySQL) for storing user, booking, and resource data.*

- ***IT Infrastructure**: The project depends on a stable IT infrastructure, including reliable servers, internet, and backup systems, to ensure uninterrupted access and performance.*
- ***QR Code Technology:** The system depends on the functionality of QR code technology for cycle borrowing and return processes.*
- ***Resource Allocation:** The system is dependent on adequate resources and funding for its development, deployment, and ongoing maintenance.*

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

The **Classroom Key and Cycle Management System** will feature a user-friendly web-based interface. It will include the following components:

- **Dashboard:** Provides an overview of available classroom keys, cycle reservations, and pending requests. Allows quick access to key functionalities for both CRs and students.
- **Key Booking Page:** Enables CRs to book and manage classroom keys, specifying details such as room, time, and duration.
- **Cycle Booking Page:** Allows students to reserve cycles, scan QR codes for borrowing/returning, and view cycle availability.
- **Approval Interface:** Provides an interface for CRs and administrators to approve key transfers and cycle borrowings.
- **Maintenance Feedback Form:** A form to report any issues with cycle maintenance for administrators to flag and manage repairs.
- **Notification System:** Sends alerts for bookings, approvals, returns, and penalties, displayed on users' dashboards.

3.1.2 Hardware Interfaces

The Classroom Key and Cycle Management System will be designed to be accessible via standard web browsers (e.g., Chrome, Firefox) on a variety of devices including desktops, laptops, tablets, and smartphones. The system does not require any specialized hardware beyond a stable internet connection. Users will interact with the system using devices that may have QR code scanning capabilities for cycle booking and returns.

3.1.3 Software Interfaces

The system will interact with the following software components:

- **Database Management System (DBMS):** Utilizes MySQL for secure storage and retrieval of key, cycle, and user data.
- **Web Server:** Runs on Spring Boot to handle backend services, client-server interactions, and API endpoints.
- **Operating System:** Compatible with Windows, macOS, and Linux environments.
- **Programming Languages:** Likely to be Developed using HTML, CSS, and JavaScript for the front-end, and Java with Spring Boot for the back-end processing.

3.2 Functional Requirements

3.2.1 F1: Role-Based Login

- **CR:** Class Representatives (CRs) can log in using their credentials to access features like borrowing classroom keys, bicycles, and managing requests.
- **Non-CR Students**: Students can log in with their credentials to borrow bicycles.
- **Administrator:** Admins can log in to manage key and cycle status, view borrowings, and feedback.

3.2.2 F2: Borrow a Classroom Key (CRs Only)

The system shall allow CRs to view classroom availability and book a classroom key if available. CRs can also request keys from other CRs if the classroom is already in use.

3.2.3 F3: Borrow a Bicycle (CRs and Non-CR Students)

The system shall allow users to borrow a bicycle by scanning a QR code at a designated drop point. Users can see the available cycles at the center before confirming the borrowing action.

3.2.4 F4: Submit Classroom Key (CRs Only)

CRs holding classroom keys can submit them back after usage. The system will update the key status and remove it from their holding list.

3.2.5 F5: Submit Bicycle with Feedback (All Students)

The system shall prompt users to submit feedback when they return a borrowed bicycle. The system will mark the bicycle as returned and make it available for others.

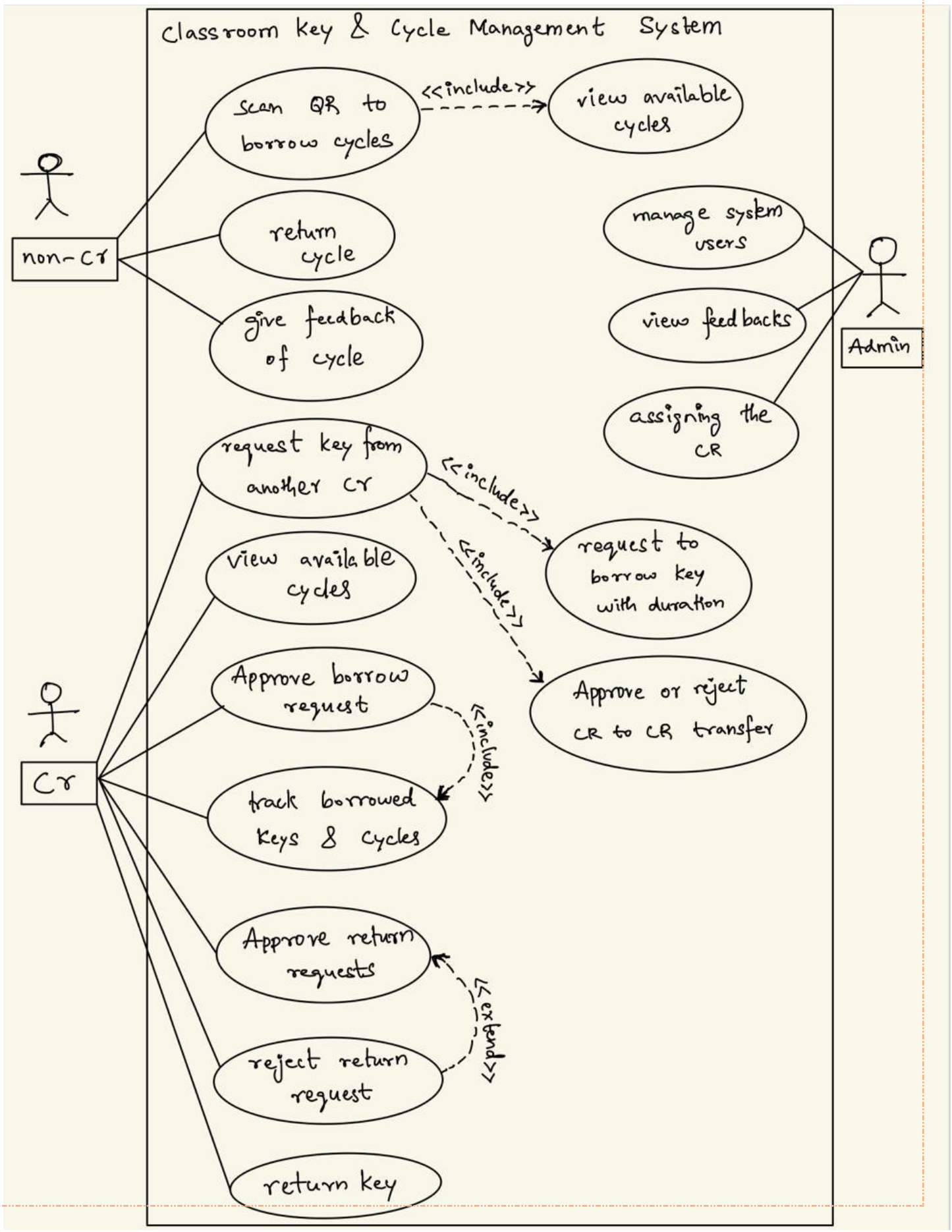
3.2.6 F6: View Requests and Sent Requests (CRs Only)

CRs can view pending and fulfilled key and cycle requests. They can also track their sent requests for classroom keys from other CRs.

3.2.7 F7: Borrowing History The system shall allow CRs and students to view their borrowing history for classroom keys and cycles, including dates, times, and feedback (for cycles).

3.2.8 F8: Administrator View of Key and Cycle Status The system shall provide administrators with a dashboard to view the current status of classroom keys and bicycles, including who is holding them, their availability, and the request status**.**

3.2.9 F9: Administrator View of Feedback The system shall allow administrators to view feedback submitted by users regarding bicycles. Admins can flag bicycles for maintenance based on the feedback.


Classroom key & Cycle Management System
scan QR to borrow cycles
<<include>>
view available cycles
non-CR
return cycle
give feedback of cycle
manage system users
view feedbacks
assigning the CR
Admin
request key from another CR
<<include>>
request to borrow key with duration
<<include>>
Approve or reject CR to CR transfer
view available cycles
Approve borrow request
<<include>>
CR
track borrowed keys & cycles
Approve return requests
<<extend>>
reject return request
return key

4 Other Non-functional Requirements

4.1 Performance Requirements

To avoid any issues, the DBMS software produced should be able to work efficiently to provide information when needed and to store data without any latency. Among the many aspects that influence performance, the system resources must be adequate and meet the baseline requirements for the software to execute smoothly.
To avoid problems when a request comes in, the performance should be accurate and the response time should be as short as possible. To avoid data loss when a server fails or a corrupted file causes a total data loss, the data should be backed up as log files on a regular basis. To be more effective, the DBMS should be able to manage a huge amount of data and conduct activities in less time. To connect to the software, the password and username will be matched to the password and name kept in the database, allowing only authenticated users to login.

4.2 Safety and Security Requirements

For reasons of safety and security, user information will be kept private and will not be shared with any other third-party organizations, ensuring that user privacy and information are protected. The data is backed up on a regular basis to ensure that it is not lost in the event of a database crash or other data loss event. The data is also saved on a private storage system, which means it cannot be viewed from the outside. For security reasons, the database is secured, and system users have varied restrictions on accessing it. Users cannot update the database; only administrators are permitted to do so. Admins and users should have distinct accounts so that only admins can make changes to the database

4.3 Software Quality Attributes

- **Adaptability** - This developed DBMS software is adaptable by any organization.
- **Availability** - The availability of the software is easy and for everyone.
- **Correctness** - The results of the function are pure and accurate.
- **Flexibility** - The operation may be flexible and reports can be presented in many ways.
- **Maintainability** - After the deployment of the project if any error occurs then it can be easily maintained by the software developer.
- **Portability** - The software can be deployed at any machine.
- **Reliability** - The performance of the software is better which will increase the reliability of the software.
- **Reusability** - The data and record that are saved in the database can be reused if needed.
- **Robustness**- If there is any error in any window or module then it does not affect the remaining part of the software.
- **Usability-** To perform any operations and to understand the functioning of software is very easy.

- **Productivity**- This software will produce every desired result with accuracy.
- **Timelines**- The time limit is very important. It will save much time and provide fast access.

5 Other Requirements

5.1 Hardware Requirements

- Processor: 1GHz
- RAM: 512 MB
- Storage: 1 GB or higher
- Internet connectivity for communication interfaces.

5.2 Software Requirements

Operating System	***Compatible with Windows, Linux, and MacOS.***
RDBMS	***MySQL or equivalent.***
Web Browser	***Latest versions of Google Chrome, Mozilla Firefox, and Safari.***

Appendix – Data Dictionary

- ***HTML****: HYPER TEXT MARKUP LANGUAGE.*
- ***CSS****: CASCADING STYLE SHEETS.*
- ***BOOTSTRAP, TAILWIND CSS****: OPEN-SOURCE CSS FRAMEWORK.*
- ***JS****: JAVA SCRIPT.*
- ***SPRINGBOOT:*** *A FRAMEWORK BUILT ON TOP OF SPRING (JAVA)*
- ***MYSQ****L: OPEN-SOURCE RELATIONAL DATABASE MANAGEMENT SYSTEM.*
- ***SQL****: STANDARD QUERY LANGUAGE FOR RDBMS.*