CHAPTER -3: SYSTEM DESIGN

3.1. Introduction

The system design will encompass a comprehensive perspective of the system from a functional standpoint. It is the process of defining elements of a system like modules, architecture, components, and their interfaces and data for a system based on the specified requirements. It is the process of defining, developing, and designing systems that satisfy the specific needs and requirements of the project or system.

The selam bus transport is a comprehensive software solution designed to streamline and enhance the efficiency of bus transportation services. This system aims to provide a user-friendly platform for passengers to easily plan their journeys, reserve seats, and make secure online payments. Simultaneously, it empowers bus operators with tools for efficient fleet management, route optimization, and real-time monitoring. It addresses the complexities of modern bus transportation by integrating advanced technologies to facilitate a seamless and convenient experience for both passengers and operators. The system encompasses modules for ticket booking, route management, fleet tracking, and robust reporting, ensuring a holistic approach to transportation management.This system aims to contribute to the overall improvement of public transportation services, enhancing accessibility and reliability for all stakeholders involved

3.2. Design Goals

The design goals of the system are derived mainly from the non-functional Requirements, that aims to create a unified and seamless implementation across multiple locations. The goal is to reduce complexity, optimize resources, and establish scalability in the transportation network.The four main quality viewpoints listed below are utilized to determine the system's design objectives.

1-Performance:

1.1-Response Time Optimization:

Goal: Minimize the time it takes for passengers to perform crucial actions, such as checking routes, booking tickets, and tracking buses in real-time.
Reasoning: Faster response times enhance the overall user experience, making it more convenient for passengers to plan and execute their journeys efficiently.

1.2-Scalability:

Goal: Design the system to handle increased demand during peak travel times and accommodate the addition of new routes and buses.
Reasoning: Scalability ensures that the system can support the growing number of passengers and transactions without degrading performance.

Dependability:

2.1-Reliability:

Goal: Ensure that the Bus Transport System consistently provides accurate and up-to-date information about routes, schedules, and bus locations.
Reasoning: Reliability is essential for building trust among passengers, as they rely on the system to plan their journeys and make informed decisions.

2.2-Fault Tolerance:

Goal: Implement mechanisms to detect and recover from faults, ensuring minimal disruption to the system in case of unexpected issues.
Reasoning: Fault tolerance enhances the system's resilience, reducing the impact of potential failures and maintaining continuous service availability.

3-Maintainability:

3.1-Code Modularity:

Goal: Design modular and well-documented code to facilitate easier maintenance and updates, allowing developers to make changes without affecting the entire system.
Reasoning: Code modularity simplifies troubleshooting and modifications, making it easier for the development team to manage and enhance the system over time.

3.2-Ease of Configuration:

Goal: Create a system that is easy to configure, enabling administrators to adapt to changes in routes, schedules, and operational requirements.
Reasoning: Easy configuration supports the adaptability of the system, allowing for efficient adjustments to accommodate changes in the transportation network.

4-End-User Criteria:

4.1-Usability:

Goal: Prioritize an intuitive and user-friendly interface for both passengers and administrators, allowing for easy navigation and interaction with the system.
Reasoning: Usability is crucial for ensuring that passengers can effortlessly access information and book tickets, contributing to overall satisfaction.

4.2-Accessibility:

Goal: Design the system to be accessible to users with diverse abilities, including those with disabilities, ensuring equal access to transportation information and services.
Reasoning: Accessibility promotes inclusivity, making the system usable by a wide range of passengers with different needs.

4.3-Feedback Mechanism:

Goal: Implement a feedback mechanism to gather user opinions, allowing passengers to provide input on their experiences and suggest improvements.
Reasoning: Regular feedback helps in addressing user concerns, enhancing user experience, and refining the system based on real-world usage.

4.4-Mobile Responsiveness:

Goal: Ensure that the system is responsive and accessible on various devices, especially mobile platforms, catering to the mobile-centric behavior of modern users.
Reasoning: Mobile responsiveness provides flexibility for passengers to access the system on the go, making it convenient to check schedules and book tickets from their mobile devices.

3.3. Proposed Software Architecture

3.3.1. Overview

Because the program is web-based, a web browser will be used to access it. The system design recommended using a database and a three-tier client/server architecture
to build this online system. A subsystem called the server, which is part of the client/server architecture, offers services to instances of other subsystems known as the clients, which are in charge of communicating with the user.
Through a network of client-server architecture, web browsers may access web-based applications. To communicate with the system, users will utilize their browser software.
Through the graphical user interface, users will communicate with the database system.
Through the use of a proper network connection, all data inputs made on the client PC are sent to the web server. The architecture of the system will be Model-View-Controller (MVC).
Practically speaking, the web browser will serve as the user interface, controllers will explain services, and data will be described in models.

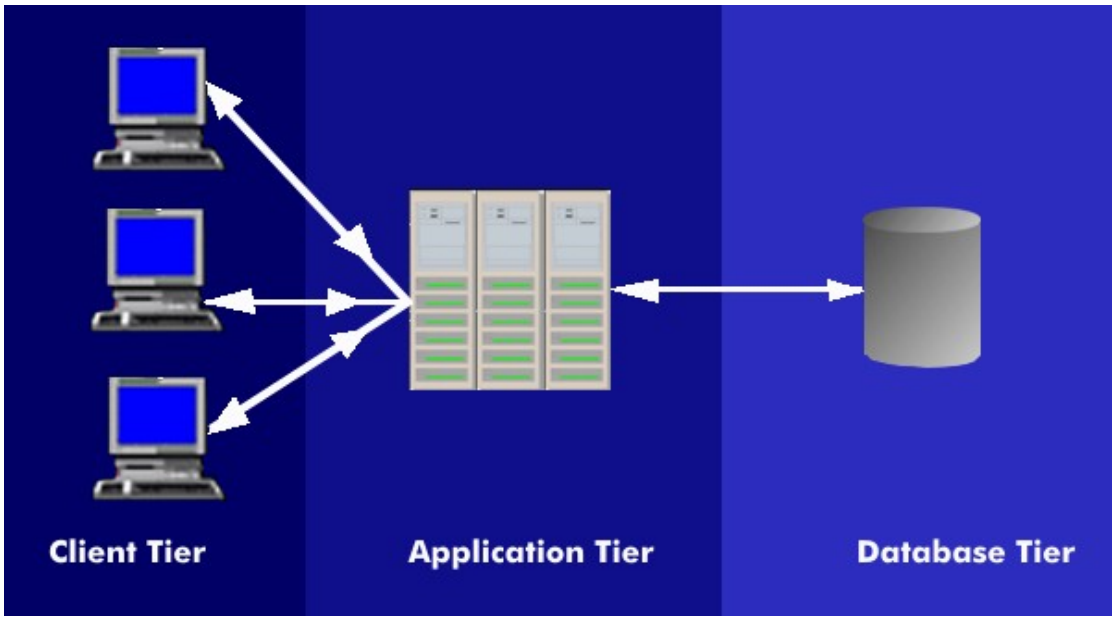

Figure 3.1.1 Selam bus system archirecture

3.3.2. Subsystem Decomposition

Subsystem decomposition is a process in software architecture where the system is broken down into smaller, manageable subsystems or modules. Each subsystem is designed to perform a specific set of functions, and together they contribute to the overall functionality of the system. In the context of the proposed software architecture for a Bus Transport System. The Selam Bus Transport System comprises several interconnected subsystems to effectively manage and streamline its operations. Each subsystem plays a crucial role in ensuring the smooth functioning of the entire transportation platform. The subsystem decomposition is as follows:

1. Passenger Management Subsystem:
Manages passenger profiles, registration, and authentication.
Handles passenger-related operations such as updating profiles, changing passwords, and resetting passwords.

2. Ticket Management Subsystem:
Manages ticket creation, reservations, and cancellations.
Handles seat availability checks and provides relevant ticket information to passengers.

3. Route Management Subsystem:
Manages routes, including creation, modification, and cancellation.
Coordinates with other subsystems for seat availability, ticketing, and overall route information.

4. Driver Management Subsystem:
Manages driver information, including addition and verification.
Coordinates with the Bus Management Subsystem for driver and bus assignments.

5. Bus Management Subsystem:
Manages bus-related information, including addition, capacity checks, and updates.
Coordinates with the Driver Management Subsystem for driver and bus assignments.

6. Ticket Officer Management Subsystem:
Manages ticket officers, including registration and authentication.
Handles ticket-related operations, such as cancellation and issue resolution.

7. General Manager Subsystem:
Manages high-level administrative tasks and coordinates with other subsystems for route modifications, cancellations, and officer additions.

8. Operation Manager Subsystem:
Manages day-to-day operations and tasks.
Coordinates with subsystems for driver and route additions/modifications.

9. Notification Subsystem:
Manages the generation and sending of notifications to users.
Provides updates on ticket reservations, cancellations, and other relevant events.