SriLanka Highway Management System

Software Requirements Specification

Functional Requirements

1. Admin Requirements

The **Admin** shall be able to:

1. Log in to the administrative side of the system.
2. View, add, and delete user accounts.
3. View and delete vehicle logs.
4. Add and remove tolls.
5. View and delete vehicle reports.
6. View and delete other reports.

---

2. Cashier Requirements

The **Cashier** shall be able to:

1. Log in to the billing side of the system.
2. View temporary vehicle logs.
3. Add vehicle log entries (Entrance/Exit) with cost details.
4. Add accident reports.
5. Add other operational reports.

Non-Functional Requirements

1. **Performance**
    - The system shall provide a fast and responsive user experience.
2. **Security**
    - The system shall ensure the security and confidentiality of user data.

3. **User Interface**
    - The system shall provide a comfortable dark-themed graphical user interface (GUI) for users.
4. **Reliability**
    - The system shall be reliable and operate without unexpected failures during normal usage.
5. **Usability**
    - The system shall be simple and easy to use for all users.
6. **Input Efficiency**
    - The system shall minimize user input by using dropdown menus and default values where applicable.

Object-Oriented Programming (OOP) Concepts We Used For Our Software

1. Encapsulation

Encapsulation is used to group data and related methods into a single class.

- We used classes for every function in our system

2. Inheritance

Inheritance is used to create new classes from existing classes.

- AuthServiceBase>>>AuthAdmin,AuthCashier
- AdderBase>>>UserAdder,CashierAdder
- DeleterBase>>>CashierDeleter,UserDeleter
- TableLoader>>>UserTableLoader

3. Polymorphism

Polymorphism allows the same method to behave differently based on the object.

- Override=DeleterBase>>>CashierDeleter,UserDeleter

4. Abstraction

Abstraction is used to hide complex implementation details and show only essential features.

- Our Database operations are handled internally,users only interact through the system interface.
- AdderBase
- We use Database class objects to transfer data to the sql server
- In cashier menu we used CashierReportAdder objects

And we used validations to all necessary places in the code!