

**TOLL BOOTH COLLECTION**

**High Level Design & Low-Level Design**

The purpose of this document is to provide a template for documenting both HLD & LLD.

### Document Control:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Project Revision History** | | | | |
|  | | | | |
| **Date** | **Version** | **Author** | **Brief Description of Changes** | **Approver Signature** |
| 25-02-2023 | HLD\_LLD  Design Document V0.1 | Ashwitha Mula | Changes in edit bus function. | Sikander |
| 24-02-2023 | HLD\_LLD  Design Document V0.2 | Nikita Laxman Patil | Changes in class of main menu. | Sikander |
| 23-02-2023 | HLD\_LLD  Design Document V0.3 | Totakura Sravani | Changes in view ticket function. | Sikander |
| 22-02-2023 | HLD\_LLD  Design Document V0.4 | Sudharshan Diliprao Kshirsagar | Changes in admin menu. | Sikander |
| 27-02-2023 | HLD\_LLD  Design Document V0.5 | Vasuki Nalla | Changes in delete ticket function | Sikander |
| 23-02-2023 | HLD\_LLD  Design Document V0.6 | Sudharshan Diliprao Kshirsagar | Changes in user registration | Sikander |

1. [1 INTRODUCTION 4](#_bookmark0)
   1. [INTENDED AUDIENCE 4](#_bookmark1)
   2. [ACRONYMS/ABBREVIATIONS 4](#_bookmark2)
   3. [PROJECT PURPOSE 4](#_bookmark3)
   4. [KEY PROJECT OBJECTIVE 5](#_bookmark4)
   5. Project Scope and Limitations 5
      1. [In Scope 5](#_TOC_250015)
   6. [FUNCTIONAL OVERVIEW 5](#_bookmark5)
   7. [ASSUMPTIONS, DEPENDENCIES & CONSTRAINTS 5](#_bookmark6)
   8. [RISKS 6](#_bookmark7)
2. [DESIGN OVERVIEW 6](#_bookmark8)

THE SYSTEM CONSISTS OF THREE ENTITIES: ADMIN , COORDINATOR AND CLIENT 6

* 1. [DESIGN OBJECTIVES 6](#_bookmark9)
     1. *Recommended Architecture 6*
  2. [ARCHITECTURAL STRATEGIES 7](#_bookmark10)
     1. [*Reuse of Existing Common Services/Utilities 7*](#_bookmark11)
     2. [*User Interface Paradigms 7*](#_bookmark12)
     3. [*System Interface Paradigms 7*](#_TOC_250014)
     4. [*Error Detection/ Exceptional Handling 7*](#_TOC_250013)
     5. [*Performance 8*](#_bookmark13)
     6. [*Security 8*](#_bookmark14)

1. BUS RESERVATION SYSTEM FLOW CHART AND ER DAIGRAM **8**
   1. [FLOW CHART DIAGRAM 9](#_TOC_250012)
      1. *Internal Interfaces 10*
      2. [*External Interfaces 10*](#_TOC_250011)
2. [DETAILED SYSTEM DESIG12](#_bookmark15)
   1. [KEY ENTITIES 10](#_TOC_250010)
3. [ENVIRONMENT DESCRIPTION 11](#_TOC_250009)
   1. [TIME ZONE SUPPORT 11](#_TOC_250008)
   2. [Language Support 11](#_TOC_250007)
   3. [User Desktop Requirements 11](#_TOC_250006)
      1. [Deployment Considerations 11](#_TOC_250005)
      2. Application Service Desk Place 11
4. [CONFIGURATION 12](#_TOC_250004)
   1. Operating System 12
   2. [Database 12](#_TOC_250003)
   3. [Desktop 12](#_TOC_250002)
5. [REFERENCES 12](#_TOC_250001)
6. [**CONCLUSION 12**](#_TOC_250000)

### 1.Introduction

The Toll Booth Collection project that stimulates a toll booth where vehicles can pass through and pay toll fee. The Project is designed to keep track of the number of vehicles that pass through the toll booth and the total amount of money collected.

* Therefore, the toll booth collection was designed. It’ll reduce the stress and workload of the employee.
* Now it hardly takes only few minutes to pass through the toll and paying for it. wherever the vehicle passes through the toll then the all details of that particular vehicle is their in this toll booth collection system, you can store the vehicle details, we can view the vehicle details.
* Only admin can add vehicles and add or remove employees.

### Intended Audience

|  |  |
| --- | --- |
| BU Authority |  |
|  |  |

### Acronyms/Abbreviations

|  |  |
| --- | --- |
| TBC | Toll Booth Collection |

|  |  |
| --- | --- |
| UML | Unified Modeling Language |

### Project Purpose

The sole intention behind the consideration of this project is to pay and pass through the Toll without wasting the time and effort of an employee.

### Key Project Objective

* To reduce the delay in processing time.
* To reduce the delay in records updating.
* To provide the user-friendliness in all possible ways.
* To provide greater flexibility.

### Project Scope and Limitation

The Bus Reservation System helps admin or employee to collect the fare and let vehicle

pass through the Toll.

### Limitation:

* + - Limited for type of vehicles.

### 1.5.1 In Scope

The scope of this project should be used in as a toll booth collection system a facility which is used to pass a vehicle.

### Functional Overview

The functions that are used for admin part is mentioned below:

* *Admin ():* This function admin() is used to login as a admin. He’ll manage the vehicle which are passing. They are:

admin can add the vehicles, view all vehicle details and can add or remove employees.

* *Employee:* Employee can only do few operations like he can add the vehicle,

show about the type of vehicle and hourly analysis.

### Assumptions, Dependencies & Constraints

Assumption, dependency & constraints - system require network connection. We are storing the vehicle data which are passing through the Toll based on vehicle Number.

### Risk

* + - No Risk (As it is for educational purpose)

# Design Overview

The entire activity is to automate the toll collection process of day activities of system library like:

* + - Admin login.
    - Adding vehicles.
    - View vehicle details.
    - View the Revenue.
    - Show all data.
    - Hourly analysis.
    - Remove/Add employee.
    - Employee login.
    - View Vehicle Details.
    - Hourly Analysis.

### Design Objectives

* + - The Bus Reservation enables admin to manage the details of add Vehicle, add employee/remove, view vehicles and can view the Revenue.
    - The user can start the program whenever they want.

### Recommended Architecture Client-side hardware interface:

* + - * Ubuntu/Linux machine
      * Terminal

### Tools to be used:

* g++ compiler
* Valgrind.
* Splint.
* Unit testing
* IT testing

# Architectural Strategies

* + - The architectural strategy used in this project is an admin - admin can manages the details of the Vehicles.

### Reuse of Existing Common Services/Utilities

* + - * Toll Booth Collection System by using ubuntu/Linux.

### User Interface Paradigms

* + - * Linux/Ubuntu machine
      * g++ complier

### System Interface Paradigms

* + - * Operating system: LINUX/Ubuntu
      * Linux Kernel version: GNU/Linux 5.10.16.3-microsoft-standard-WSL2 x86\_64
      * Bash shell: 5.0.16(1)-release
    1. **Error Detection / Exceptional Handling**
       - We used Valgrind for error detection and exceptional handling.
       - We have integrated the project again and again to resolve the errors. We must manage exceptional handling for all types of vehicles which are passing through toll.

## 2.2.5 Performance

* + - * To make the application run faster and smoother the code has to be written in optimized manner. The performance is based on the configuration of the system.

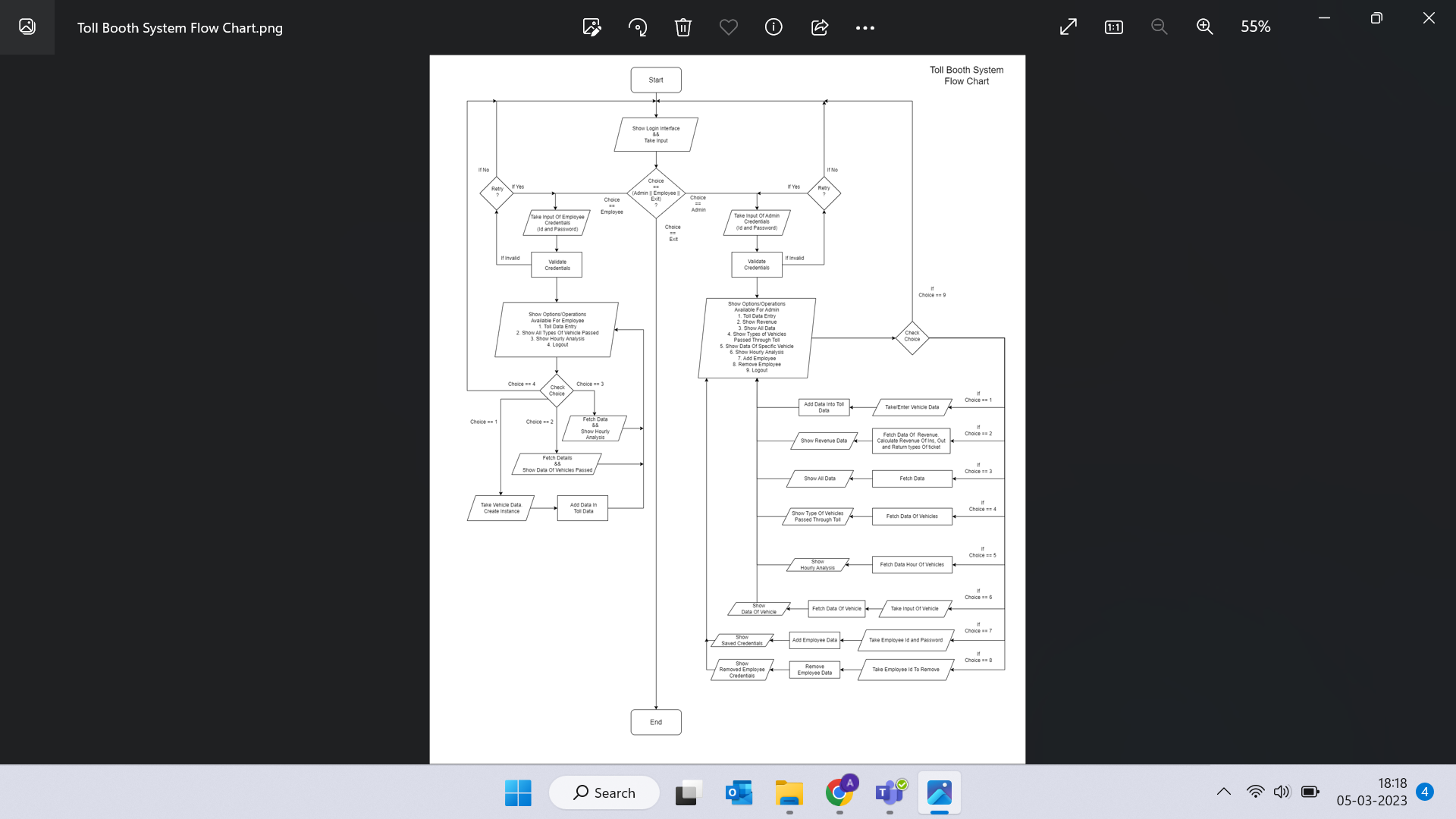
# 2.2.6 Security

* + - * For security purpose Admin will store the all-vehicle information. we must secure the data in the directory.

# Toll Booth Collection Flow Chart:

### 3.1 Flow Chart Diagram

Flow Chart Diagram is a diagrammatic representation of data movement through a system-manual or automated from inputs to outputs through processing. Flow chart helps in the analysis of the flow of data through a system and thus help identifying the system requirements.



### Internal Interface

The internal interfaces comprise interfaces through which the system interacts with the clients/users through which it provides them services.

### External Interfaces

The external interface comprises interfaces through which the users interact with the system.

* + - * Desktop or Linux Machine

# Detailed System Design

The Toll Booth Collection System web portal that enables vehicles to pass through.

and add the vehicle adding, viewing, updating and etc.

### 4.1 Key Entities

Key Entities are associated with the systems are: -

### Admin: -

The admin can manage add data of vehicle, view revenue collection, Hourly analysis etc.

### End User: -

The user can pass through the toll by paying the fare.

# Environment Description

* + - G++: In Linux, the G++ stands for GNU Compiler Collection. It is a compiler system for the various programming languages. It is mainly used to compile the C and C++ programs.
    - UBUNTU: Ubuntu is an open-source operating system (OS) based on the Debian GNU/Linux distribution.
    - Ubuntu incorporates all the features of a Unix OS with an added customizable GUI, which makes it popular in universities and research organizations. Ubuntu is primarily designed to be used on personal computers, although a server edition does also exist.
    - GITHUB: GitHub is a code hosting platform for version control and collaboration. It lets you and others work together on projects from anywhere. This tutorial teaches you GitHub essentials like repositories, branches, commits, and pull requests.

# Time Zone Support

It will support time zones as per Indian standard time (IST) in (GMT +5:30) and UST standard.

# Language Support

We are using C++ programming language is a high-level general-purpose programming language created by Danish computer scientist Bjarne Stroustrup as an extension of the C Programming language. The C++ programming language was initially standardized in 1988 as ISO/IEC 14882:1998. There is different version in C++ they are C++03, C++11, C++14, C++17 and C++20.This is current version.

# User Desktop Requirements

## Deployment Considerations

Deployment considerations are,

* + - * Linux Operating System
      * 10gb Free Space
      * 4GB RAM

### Configuration

NA

* 1. **Operating System**

Linux desktop editions with 4 GB RAM- A GUI-based LINUX system must be used

### Database

Vectors are used to store the necessary details.

### Desktop

# 7 References

* CPU: Intel i3/i5/i7 generation 3 and later
* RAM: 4GB or greater - For optimal performance, 6GB or 8GB are recommended if you will be running multiple browser tabs and/or multiple applications at the same time

1. [**Toll Booth Collection\_system GeeksforGeeks**](https://www.geeksforgeeks.org/traceroute-in-network-layer/) **2.http**[**s://www**](http://www.geeksforgeeks.org/bus-reservation-systemin-e-commerce/)**.gee**[**ksforgeeks.org/Toll-Booth-Collection-system in-e-commerce/**](http://www.geeksforgeeks.org/bus-reservation-systemin-e-commerce/)
2. **CONCLUSION**
   * Finally, in Toll Booth Collection System, we have developed a secure, user friendly Toll Collection System.
   * It contains Admin and Employees who will manage the whole system.
   * The user can travel through the toll by paying.