# SCHOOL BUS MANAGEMENT SYSTEM

## High Level Design and Low Level Design

**Document Control:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **TOPIC: SBMS** | | | | | | | | |
| Guided by- **Rahul Tarkunde** |  |  | |  |  |  |  |  |
| **Date** | **Version** | **Author** | **Brief Description of Changes** | | | | **Approver Signature** | |
| December 07,2022 |  |  |  | | | |  | |
|  |  |  |  | | | |  | |
|  |  |  |  | | | |  | |

# Introduction

## PURPOSE

The purpose of this software or project is to show the requirements for School bus management system which creates direct contact between the Admin and Student menu. And it will provide an efficient and handy platform for the development of the proposed product or software. It applies proven methodologies and uses current software tools so that one can plan, control, and monitor student processes, and other components.

### Intended Audience:

This document is intended to be read by Admin and the Student Menu.

### Acronyms/Abbreviations:

|  |  |
| --- | --- |
| ADMIN | STUDENT |
|  | TO GIVE A CLEAR VIEW TO THE ADMIN AND THE STUDENT MENU ABOUT THE COMPLETION STATUS OF THE PROGRAM. |

* 1. **Project Purpose:**
     + The purpose of this document is to show the requirements for the School bus Management System, which gives a clear understanding to the admin and student of the completion of the tasks without the involvement of the mediators.
     + This software or project is to show the requirements for school bus management system which creates direct contact between the admin and the student menu. It applies proven methodologies and uses current software tools so that one can plan, control, and monitor student, processes and other components.

### Key Project Objectives:

* + - School bus management system will login and validate the username.
    - Add /remove/check status of the student.
    - Add bus Stop and print bus amount
    - Student will and request/check status .
    - The application will now update the task completion under user story.

### Project Scope:

### It can be implemented into the next step that involves School bus management system.

### Another user like Admin and Student menu can also be integrated with the future proposed system.

### The admin can see interested /add/remove students

### Student can also request/remove/check.

### Functional Overview: -

### Add Student: Add student in the database(csv) taking Student Id, firstName, lastName, username, password, amount. These data are updated in a csv file.

### Remove Student: Remove student from the database(csv) taking the unique Student Id.

### Update Students: The data of student csv got updated.

### Add busStop: Add busStop to the student while checking the status of available stop and add busStop.

### Display busStop: Show the available busStop and add busStop.

### Remove from bus: Taking student id and remove student from bus.

### Menu:show menu login page.

### Enter more than 5:It shows invalid operation

### Bus Amount: Print the bus amount.

### Exit:It's used to add custom menus to the standard menu bar.

### Final Report: At last, the final report will be given. The final report will include all the details of the student menu and the capacity of the bus.

### Design Overview:

Instant Chatters comprises of the following modules:

|  |  |
| --- | --- |
| Name of the Module | Void addStudent();  Void removeStudent(); |
| Handled by | Kishu Kumar |
| Description | This function takes the input from the user such as Student Id, firstName, lastName, userId, password, amount. Remove student from the database(csv) taking the unique Student Id. |

|  |  |
| --- | --- |
| Name of the Module | Void updateStudent();  Void addBusstop(); |
| Handled by | Vidyashree Ravindra Gurav |
|  | The data of student csv got updated.Add busStop to the student while checking the status of available stop and add busStop. |
| Name of the Module | Void displayBusstop();  Void removeFrombus(); |
| Handled by | Mohini Nandkumar Mole |
| Description | Show the available busStop and add busStopTaking student id and remove student from bus. |

|  |  |
| --- | --- |
| Name of the Module | Void menu();  Enter More Than 5 |
| Handled by | Yogita Vijaykumar More |
| Description | show menu login page.It shows invalid operation |

|  |  |
| --- | --- |
| Name of the Module | Void busAmount();  Void Exit(); |
| Handled by | Samiksha Gulabrao Khode |
| Description | Print the bus amount.It's used to add custom menus to the standard menu bar. |

#### Design Objectives:

This project aims to create and develop the School Bus Management System

In which there are two user the Admin and the Student, The system’s job is to update new busStop and to assign it to different busStop, the job of the student is to update the busStop completion status of the busStop that has been assigned to them. The application will then give the users a clear view of the amount of student presents and the amount of seats that is left.

#### Performance:

The system will work on the admin terminal. The performance depends on the hardware component of the admin’s system.

#### Maintenance:

If maintenance demands consistently keep the Team from completing their Sprint Plan, stop planning for so much. In Sprint Planning, leave some headroom – an allowance for maintenance. Reduce the forecast for new feature work. The size of the allowance may be easy to determine from past Sprints or it may take experimentation. If the Team does not need all the time budgeted in a Sprint, they can use it for more feature work or payment of technical debt.

### Environment Description:

* 1. **Time Zone Support:** IST- Kolkata
  2. **Language Support:** English

#### User Desktop Requirements:

* + 1. 64-bit processor, 1 GHz or faster
    2. At least 2 GB free hard drive space
    3. At least 1 GB RAM

#### Server-Side Requirements:

* + 1. 64-bit processor, 1 GHz or faster
    2. At least 1 GB free hard drive space
    3. At least 1GB RAM

#### Deployment Considerations:

* + - 1. Easy setup: no session storage daemon, use memory caching to enhance performance.
      2. Local storage is used.
      3. No network latency to consider.
      4. To scale buys a bigger CPU, more memory, larger hard drive, or additional hardware.

#### Application Server Disk Space:

No such disk space is required as the program is fully functional on online IDE(s) as well. The Local Operating System is required and one text file to store the records of processes.

#### Database Server Disk Space:

No such disk space is required as the program is fully functional on online IDE(s) as well.

The Local Operating System is required and one text file to store the records of processes.

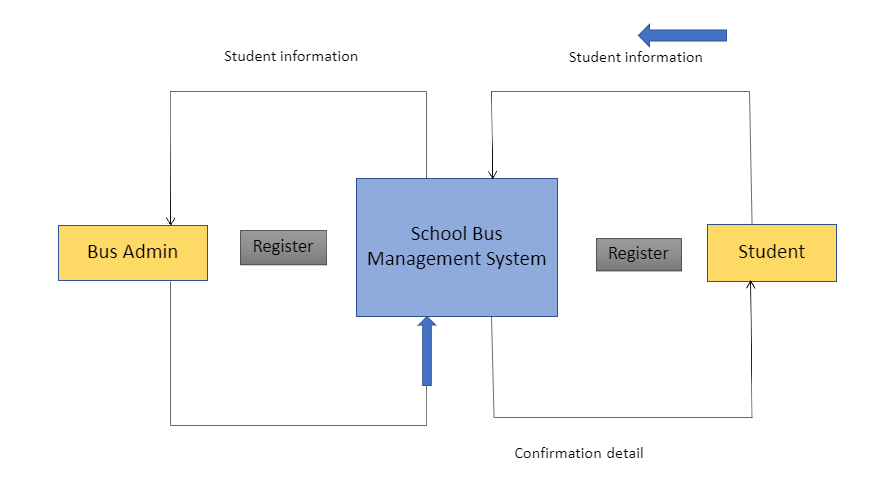
#### Integration Requirements:

1. Language: C
2. Tools: Valgrind, Makefile
3. Complier: gcc
4. Linux Environment

#### Configuration:

**3.5.1: Operating System**: Linux environment.

### Data Flow Diagram: -



**Admin Menu:**

* + Add Student
  + Remove student
  + Add Bus Stop
  + Bus Amount
  + Exit

#### Student Menu:

#### Request add to bus queue.

#### Remove from bus queue.

#### Check status.

#### Exit.

#### 5 Flow Chart Diagram:

#### 