**SRS Project – School Bus Logic**

Team Members: Zed Walter

Description of project: This project will consist of a SQL database which will hold information about students, bus drivers, and the bus routes. The database will be linked to a website that will allow users, mainly the bus drivers or school administrators, to see which students have gotten on which busses.

**Section 1**

Introduction:The website will allow for greater safety of students by tracking which school bus they get on. This will allow school administrators to better respond to emergency situations, such as a car accident involving school busses and contact the parents of students involved in the accident.

Purpose: The project is geared toward giving parents more peace of mind with the safety of their children, and school administrators a better way to determine who is on a school bus in case of an accident. As well as giving administrators a more direct way to access student health information in case of an emergency and quick access to parent contact information.

Scope: The website will allow bus drivers to update the status of students on their bus and allow school administrators to see student information including their bus status.

Technologies Used: Python, HTML, CSS, SQL

**Section 2a**

Must Have Requirements:

The system shall allow the user to update the “on bus” status of an individual student

The system shall allow the user to update the “on bus” status of all students currently on the bus to “False”

The system shall allow the administrator to create a new student

The system shall allow the administrator to update student information

They system shall allow the administrator to delete a student

The system shall allow the administrator to view all information about a student

The system shall allow the administrator to create a new bus route

The system shall allow the administrator to edit a bus route

The system shall allow the administrator the delete a bus route

The system shall allow the administrator to view all the students on a bus route

The system shall allow the user to see the names of students and their “on bus” status

**Section 2b**

Stretch Requirements:

The system may provide a login system generating different views for administrators and users.

The system may provide the user with a search function for student names

The system may inform the user if a student is on the wrong bus

**Section 2c**

Weekly schedule:

|  |  |  |  |
| --- | --- | --- | --- |
| Week | Task | Hours | Competition Criteria |
| 1 | Research projects | 5 | Project proposal draft |
| 2 | Finalize project idea & draft | 10 | Completed project proposal |
| 3 | Research app vs web | 10 | Decide on end user UI |
| 4 | Outline Database design | 15 | Completed ERD |
| 5 | Download & implement libraries | 15 | Environment setup (“hello world”) |
| 6 | Requirements draft | 10 | SRS draft |
| 7 | Requirements Analysis | 15 | Completed SRS |
| 8 | Backend coding & library implementation | 15 | Full database implementation |
| 9 | Utilize flask to display queries to front end | 15 | Complete connection from database to front end |
| 10 | CSS and HMTL formatting update | 10 | Complete stylization update and finalize graphics |
| 11 | Overall testing | 10 | All must have requirements implemented and |
| 12 | Stretch challenge attempts + stretch challenge testing | 10 | Login screen and differing account views setup. Or finish must have’s if not done |
| 13 | Instructor meeting prep + presentation | 5 | Final turn in complete |

**Section 3**

**Design Overview of the Product**

Workflow: The user will be presented with buttons correlating to bus routes. Clicking on a button will lead them to a page where they can change the status of students tied to that route to be “on bus” or “off bus”.

Resources: Visual Studio, open-source libraries (flask/SQLAlchemy), Heroku to host my database + website

Data at Rest: Data will be stored on the Heroku database which is a PostgreSQL database.

Data on the Wire: User will interact with the server which will make requests to the database to store or retrieve information. This information will then be returned to the server to be displayed to the user.

Data State: All data will be stored on Heroku’s PostgreSQL database and queried through our server API.

HMI/HCI/GUI:

<https://wireframe.cc/I3QmWr>

<https://wireframe.cc/GSSHSM>

<https://wireframe.cc/1vj1yM>

<https://wireframe.cc/3J8yNe>

Pictures/ Diagrams:

**Section 4**

**Verification**

**Demo:** To make sure the system meets all the requirements I will be testing each section. The user section to alter student’s bus status, and the administrator section to add/edit student information. I will create demo bus routes and students to ensure that all the requirements are met.

**Testing:**

|  |  |
| --- | --- |
| **Must have requirements & stretch requirements** | **Acceptance testing criteria** |
| The system shall allow the administrator to create a new student | The administrator clicks on the new student button and is brought to a forum to enter new student information. After they click submit, it adds a new student to the database. |
| The system shall allow the administrator to update student information | The administrator clicks on the update student button and is brought to a forum to update a student’s information. After they click submit, it updates that student’s information. |
| The system shall allow the administrator to delete a student | The administrator clicks on the delete student button and is brought to a forum to enter a students information. After they click submit, it removes that student from the database. |
| The system shall allow the administrator to view all information about a student | The administrator clicks on the search student button and is brought to a forum to search for a student. After they click submit, it shows all the information about a student. |
| The system shall allow the administrator to create a new bus route | The administrator clicks on the add route button and is brought to a forum to enter the information for a new bus route. After they click submit, a new bus route is added to the database. |
| The system shall allow the administrator to delete a bus route | The administrator clicks on the delete route button and is brought to a forum to enter the information to find a bus route. After they click submit, the bus route specified is deleted from the database |
| The system shall allow the administrator to view all students on a bus route | The administrator clicks on the search route button and is brought to a forum to enter information to find a bus route. After they click submit, the bus route is displayed with all the students. |
| The system shall allow the administrator to edit a bus route | The administrator clicks on the edit route button and is brought to a forum to enter new information for the route. After they click submit, the bus route will be updated in the database with the new information. |
| The system shall allow the user to see the names of students on a bus route and their “on bus” status | The user clicks on a bus route and is brought to a screen that shows the students names and whether they are on the bus or not. |
| The system shall allow the user to update the “on bus” status of an individual student | The user will search on a drop-down bar for a student’s name and after selecting and hitting submit, it will update the student’s information in the database and on the screen to be “on bus”. |
| The system shall allow the user to update the “on bus” status of all students currently on the bus to “False” | The user can select a “Drop off all” button which will update the status of all students on the bus to be False which will display as “Off bus” |
| The System may provide a login system, generating a different view for administrators and users | When you first open the website, you will be prompted with a login screen. Using a user vs administrator login will provide you with different views past the login screen. |
| The System may provide the user with a search function for student names | In the user view, while looking at a bus route, the user will be able to use a search bar for all students instead of a drop down for just students tied to the bus. |
| The system may inform the user if a student is on the wrong bus | The program will prompt the bus driver with a message if the student they are trying to alter the “on bus” status is not tied to the route they are on. |

**Sources/Citation/Resources** Links:

<https://www.postgresql.org/docs/>

<https://www.heroku.com/what>

<https://flask.palletsprojects.com/en/2.2.x/>