Software Requirements Specification

Version 1.0

May 21, 2020

Elementary School Pickup Services

by

Roneil Boyce

Liam Curtis

Elvis Mack

CSCI 5530 Software Engineering

**Table of Contents**

1. **Introduction**
   1. **Abstract**

The purpose of this manual is to describe a School Student Pickup Tool designed to efficiently and safely support a teaching staff while they board students to their parent’s vehicles. The manual displays diagrams depicting the multiple use cases. The features below describe the behavioral, functional, non-functional behavior of the program including the constraints and security involved with the system. This SRS manual provides documentation for software engineers and developers to edit the program. The contents of this manual should be provided for understanding amongst administration, staff, and parents. The program will be verified upon the standards laid out in the manual to provide the user with detailed insight on the programs functionality.

* 1. **Scope of Project**

The program described is designed to support an elementary school efficiently route parents of students through a pickup lane to safely retrieve their children from school. The concept is to develop a mobile application to be used by administrators, staff, and parents to verify the vehicles that are used to pick up students. The application will receive inputted data from the administrators, and staff to identify and confirm the student’s retrieval by their relative. The program’s security will designate the legitimacy of the vehicles and confirm the pickup in real time. Prior to this, the school was using a time-consuming method of verbal confirmation. With the introduction of the application, the pickup flow should become more efficient while improving the safety of the students. From the perspective of the administrator and staff, the application will display the list of cars currently in the pickup line, the students yet to be picked up, and the verified times of their departure.

* 1. **Glossary**

|  |  |  |
| --- | --- | --- |
| Term | Abbreviation | Description |
| Administrator | Admin | User with master key access to the program. Highest-level privileges granted. |
| Database | DB | Storage system of all cumulative project data. |
| Parent |  | User that provides data to Admin/Staff. |
| Pickup Time | PT | Time of student departure and event-based program. |
| Software Requirement Specification | SRS | Document to describe and visualize the entire functionality of a proposed system. |
| Staff |  | User with access to the Parent/Student data. Medium-level privileges granted. |
| User |  | Admin, Staff, or Parent |

* 1. **References**

IEEE. *IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications.* IEEE Computer Society, 1998.

* 1. **Overview**

The following chapter, System Description, will provide detailed descriptions to the unique functions of the program. It displays visualization documents and graphs intended to elaborate the conceptual context necessary for the more technical third chapter. This section provides understanding to the consumer who is less conventionally familiar with the programs software. In the final chapter, Requirements Specification, a detailed manual provides technical explanations for the breadth of the programs functions. This section is intended for interpretation by the developers for further advancement in the program.

1. **System Description**
   1. **System Environment**

|  |
| --- |
|  |

Elementary School Pickup Services has three active actors and a hierarchical system. The Admin, Staff, and Parent accounts can login to the system with varying degrees of software privileges.

The administrative access provides the tools to manage Staff, Parents, Students, and Vehicles as well as the program itself. The Staff accounts have access to view Parent, Student, and Vehicle data. They also provide regular input during the PT to verify the safe and timely destination of the Student. Parent accounts are capable of providing user input: Parent Name, Car Make, Car Model, License Plate #, etc. Parent accounts are not meant to have access to any of the systems data other than their own Parent/Student/Vehicle data.

* 1. **Functional Requirements Specification**

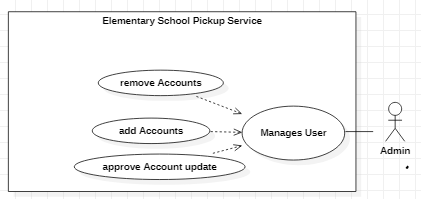
In the proceeding sections, outlines for each user’s use case is diagrammed.

|  |  |
| --- | --- |
| FR#0 | System includes login functionality |
| FR#1 | Admin can add or remove any Parent or Staff account |
| FR#2 | Admin can edit any Vehicle or Student information |
| FR#3 | Admin can send alerts to Parent and Staff accounts |
| FR#4 | Admin can verify additional relatives and their Vehicles |
| FR#5 | Admin can edit the calendar for sending daily alerts |
| FR#6 | Admin can send Emergency Shutdown alerts |
| FR#7 | Staff can edit the vehicles in the list |
| FR#8 | Staff views the list updates in real time |
| FR#9 | Staff can confirm a vehicle by inputting it’s make, model, color, and license plate # |
| FR#10 | Staff can view the daily PT for a unique student and how many students remain |
| FR#11 | Staff can mark a student “Absent” preventing the daily email from being sent to the parent |
| FR#12 | Staff can confirm a student has boarded the correct vehicle |
| FR#13 | Staff will receive an update for whether the Parent is Confirmed, Unconfirmed, Late, Absent |
| FR#14 | Staff can view vehicles in the car rider line |
| FR#15 | Parent can check into the car rider line |
| FR#16 | Parent can sign up for an account upon receiving an invitation |
| FR#17 | All Parent accounts receive an email 30 minutes before PT |
| FR#18 | Parents can respond Confirmed to the email. “Unconfirmed” will be the status until the response is received |
| FR#19 | Parent can notify the Staff that they are running late via the email alert |
| FR#20 | Parent can request to add multiple relatives/vehicles for one account |
| NFR#21 | Data is stored permanently |
| NFR#22 | Data will update in real time |
| NFR#23 | The system is only accessible on campus |
| NFR#24 | Admin and Staff accounts are the only ones permitted to view Parent, Student, or Vehicle Data (exception: Parents personal data.) |
| NFR#25 | The systems Date/Time will take into account leap year and daylight savings time |

* + 1. **Admin Use Cases**

Use case: Add, Approve or Remove Accounts From Users Elementary School Pickup Service:

**Diagram:**



**Brief Description**

The Admin decides which users will be part of the Elementary School Pickup Service by manually adding, approving updates and removing accounts.

**Initial Step-By-Step Description**

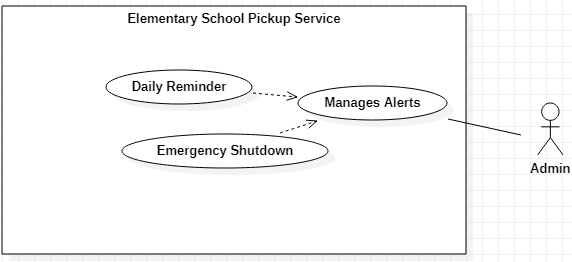
Before this use case can be initiated, the Admin has already accessed the Program and has been authorized as the Admin.

1. The Admin chooses to edit the User database.
2. The system displays the current Accounts to the Admin.
3. The Admin adds and/or removes desired Accounts and approves Accounts update requests.
4. The system presents the updated User database.

#### 

Use case: AdminManages Alerts for Elementary School Pickup Service:

**Diagram:**



**Brief Description**

The Admin decides when to signal each alert. Each of these alerts are situational based and will be part of the Elementary School Pickup Service.

**Initial Step-By-Step Description**

Before this use case can be initiated, the Admin has already accessed the Manage Alertsand has been authorized as the Admin.

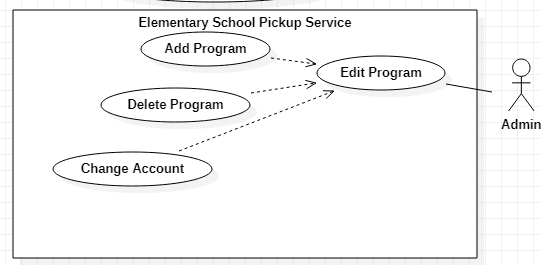
1. The Admin gains access to managing alerts.
2. The system displays the list of alerts to the Admin.
3. The Admin chooses to implement a set of dates to not receive any daily notification.
4. The system updates the set of dates for daily reminders to the User database.
5. The Admin writes Emergency Alert scenarios.
6. The System updates the setting for the Emergency Alert

#### 

#### Use case: Add or Delete Programs and change Account from Programs:

**Diagram:**

The Admin confirms his Program



**Brief Description**

The Admin decides which program to add or to delete, and what days should be used for daily reminders. This will be part of the Elementary School Pickup Service.

**Initial Step-By-Step Description**

Before this use case can be initiated, the Admin has already accessed the Program and has been authenticated as the Admin.

1. The Admin chooses to edit the Programs.
2. The system displays the current list of Programs, and accounts to receive these alerts.
3. The Admin adds and/or removes desired programs and decides which Account will have access to each program.
4. The system presents the updated status to the Admin.

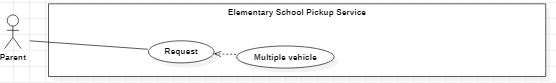
#### 

#### Parent Use Cases

#### Use case: A Request for multiple vehicle From Parent:

**Diagram:**

The Parent confirms his selection



**Brief Description**

The Parent sends a Request to the admin to register multiple vehicles which is part of the Elementary School Pickup Service

**Initial Step-By-Step Description**

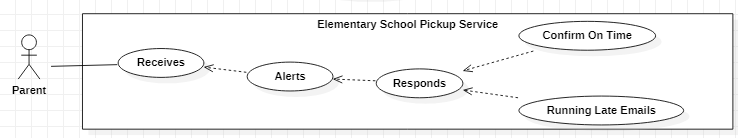
Before this use case can be initiated, the Parent must create their account and known by the system as a parent.

1. The Parent log in to their account.
2. The system displays the current options to the Parent.
3. The Parent request to add another vehicle.
4. The system sends the request to the Admin.

#### 

#### 

#### Use case: Parent receives Alerts:

**Diagram:**

**Brief Description**

The Parent receive an Alert depending upon their response.

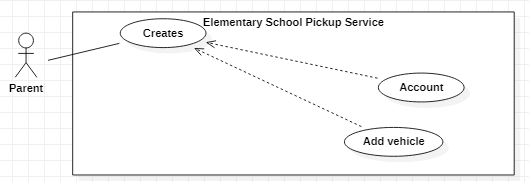
**Initial Step-By-Step Description**

Before this use case can be initiated, the Parent must have access to their account.

1. The Parent log in to their account.
2. The system displays the Parent current account.
3. The Parent goes to their setting to enable or disable alerts.
4. If the Parent enables the alert then the system will send alerts to the Parent email.
5. Once the Parent get the alert, they must respond to confirm on time pick up.

#### Use case: Creating Account for Parent:

**Diagram:**



**Brief Description**

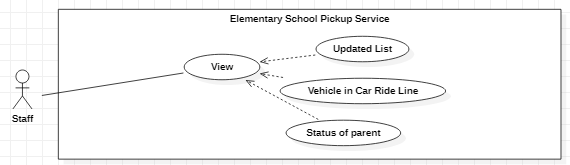
The Parent will sign up for an account and register their vehicle.

**Initial Step-By-Step Description**

To initiate this use case the parent must first create an account.

1. The Parent chooses to create an account.
2. The system asks for the Parent’s information and car information.
3. The Parent fills in the necessary information.
4. The system saves the Parent’s information and saves it to the database.
   * 1. **Parent Use Cases**

#### Use case: Views which vehicle are in the Car Ride Line:

**Diagram:**

**Brief Description**

The Staff will be able to view the status of the line and which vehicle belongs to who and position where in the line..

**Initial Step-By-Step Description**

Before this use case can be initiated, the Staff must have access to the view.

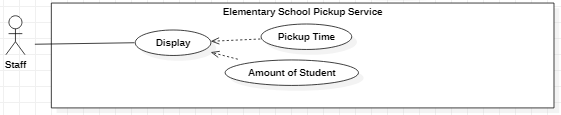
1. The Staff Request to see the view of the Car Ride Line.
2. The System presents the view and updates at real time.
3. When a Parent car has left the Staff will tell the system to change the Parent Status.
4. The System will change the parent status from Still here to Picked up.
5. The System will then update the list of vehicles in the line automatically.
6. The Staff will look back at the list and search for the next available pick up.

#### 

#### 

#### 

#### Use case: Display what time the Parent will arrive and the amount of student to Staff:

**Diagram:**

**Brief Description**

The Display will show the specify time for each parental pickup and the amount of student that are at school.

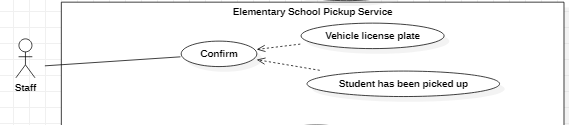
**Initial Step-By-Step Description**

Before this use case can be initiated, the Staff must have access to the Pickup Time and Amount of Student.

1. The Staff will look at the display.
2. The System will Display the amount of Student and the Pickup Time for each Parent
3. The System will update itself when a student leaves early or does not arrive at all.
4. The System will also update if the Parent decides to Pickup their student late.

#### 

#### Use case: Staff confirming vehicle:

**Diagram:**

**Brief Description**

The Staff confirms which student ride has arrived by the vehicle license plate.

**Initial Step-By-Step Description**

Before this use case can be initiated, the Staff must have access to the list of cars for each student.

1. The Outside Staff will confirm which car has arrived for each student.
2. The System will ping the selected car that Outside Staff have found.
3. The Inside Staff will receive the ping from the System and deliver the student outside.
4. The system will then update itself.
   1. **User Characteristics**

Admin are expected to be internet literate and the overall overseer of the application and has the power to add and remove accounts with different roles. The admin should be familiar with the application as a whole as they are responsible for making sure the lists and system is supported by both parents and staff. All admins will have the highest privileges of all users and must understand the severity of their actions when removing accounts or sending emergency alerts.

Staff are expected to be internet literate and are the primary user of the application. Staff must be able to look into an updating list of pickups for students, filter options within the list, click buttons to generate reports and if need be download the reports themselves. They must be able to also confirm pick up for students.

Parents are expected to have at least simple functionality with the internet and a primary user of the application. Parents will be able to create their own account, add new vehicles, and check into the application to tell the staff they are ready to pick up their student. Parents must understand the process of adding their vehicle and checking in.

# Non-Functional Requirements

# The Elementary School Pickup Services will be contained on a server with high speed internet access. The program uses Alpha Software, a tool for developers to combine front end development with their database. The speed of the internet connection relies on the hardware of the user and not the system itself. The data will be stored permanently and will not be deleted under any circumstances. The data will only be accessible via the wireless internet connection; user’s will not be able to access Student data from off campus.

1. **Software Specification**
   1. **Requirement Specification**
      1. **Parent Sign Up**

|  |  |
| --- | --- |
| Use Case Name | Parent Registration |
| Trigger | Admin clicks the “Add User” button  * Non-Registered user clicks the “Add User” Button |
| Precondition | Admin: Is logged in and on the *Manage Users* sections of the admin page.  * Non-Registered User: None |
| Basic Path | User enters information including name, username, password, and email.  * *For Admin Only*: Admin will then select the Parent role. * The user submits the information using the *Submit* Button. * The system registers the new Parent user and sends an email to the user’s specified address. |
| Alternative Paths | Error message regarding the issues and the user is asked to provide valid information regarding adding the user. |
| Postcondition | The email is sent requesting the new user to log in. |
| Exception Paths | The user information already exists within the system.  * The attempt may be abandoned at any step of the process. |
| Other | None |

### 

### Login

|  |  |
| --- | --- |
| **Use Case Name** | Login |
| **Trigger** | The user wants to access their account. |
| **Precondition** | The User is on the *Login* page of the application and has account information to allow them to log in (username/password). |
| **Basic Path** | * The system requests that the user enter their information. * The user enters the username in the username box * The user enters the password in the password box * The user presses the *Login* button. * The system validates the information by checking the database of credentials. * The system approves of the credentials and authenticates the user and logs the user into the system. * The system directs the user to the home page with added information available to them. |
| **Alternative Paths** | If the user provides invalid credentials, the user will not be approved access.   * The system presents a message to show that the login has failed and ask the user to input valid information. * The user has a choice to attempt to login again or request a password change. |
| **Postcondition** | The user can access the system according to their role. |
| **Exception Paths** | The user may logout or close the connection at any time. |
| **Other** | None |

### Add Vehicle

|  |  |
| --- | --- |
| **Use Case Name** | Add Parent Vehicle |
| **Trigger** | The parent decides to add a vehicle onto their account |
| **Precondition** | The parent is already logged in and is on the *Add Vehicle* page. |
| **Basic Path** | * The user is asked to re-input their account username and password. * The user enters the username in the username box * The user enters the password in the password box * The user presses the *Login* button. * The user inputs Plate #, Plate Type, State of Registration, Year, Maker, Model, Color, and Body. * The user clicks the *Submit* Button * The user registers a new vehicle on their account and sends an email to the specified address to confirm. |
| **Alternative Paths** | * If the parent provides invalid credentials, the parent will not be approved access requesting the parent to input correct information. * Error message regarding the vehicle information and the user is asked to provide valid information regarding adding the vehicle. |
| **Postcondition** | An email is sent to confirm the new vehicle on the account. |
| **Exception Paths** | The user may abandon the addition of the vehicle at any time. |
| **Other** | Re-inputting account information is for security reasons. |

### 

### Edit Calendar for daily alerts

|  |  |
| --- | --- |
| **Use Case Name** | Edit Calendar |
| **Trigger** | The admin wishes to update the calendar with new daily alerts for all users. |
| **Precondition** | The admin is logged into the system and is on the dashboard. |
| **Basic Path** | * The user clicks *Calendar*. Leading them to the Calendar page. * The user clicks *Edit*. * The user updates the calendar with new information that was requested. * The user clicks *Save*. |
| **Alternative Paths** | None |
| **Postcondition** | The calendar updates immediately. |
| **Exception Paths** | The user may abandon the operation at any time. |
| **Other** | None |

### View Student Pickup List

|  |  |
| --- | --- |
| **Use Case Name** | View Pickup List |
| **Trigger** | The staff member wants to know the updated pickup list for students getting ready to be picked up. |
| **Precondition** | The staff member logs into their account and is on the dashboard. |
| **Basic Path** | * The user clicks *Pickup*. * The user views the entire pickup list with parents either there or on the way. * The user selects filter 1 by status. * The system filters status by “on the way” and “waiting”. * The user selects filter 2 by Date/Time. * The system filters by both status and time starting from waiting earliest to none latest. |
| **Alternative Paths** | None |
| **Postcondition** | The user has an updated list of parent pickups with relevant information for their arrival. |
| **Exception Paths** | None |
| **Other** | None |

### Send Arrival Notification

|  |  |
| --- | --- |
| **Use Case Name** | Send Arrival Notification |
| **Trigger** | The parent wish to notify the staff that they have arrived for the pickup of their child. |
| **Precondition** | The parent has logged in to the system and is on the dashboard. |
| **Basic Path** | * The user clicks *Pickup*. * The user confirms the vehicle being used for pickup within the available vehicles on the account. * The user clicks *Send Pickup Notification*. * The system notify that the user should be in the lineup to be waiting for their student before submitting. * The user clicks *Continue*. * The system sends the notification to the staff and sends the user to a page to show the eta for the pickup. |
| **Alternative Paths** | None |
| **Postcondition** | A staff member is notified of the pickup request and the user is sent to a specific pickup page to prepare for the pickup. |
| **Exception Paths** | The user can cancel the pickup request at any time if they are not ready for the pickup. |
| **Other** | None |

### Generate Vehicle Pickup Information

|  |  |
| --- | --- |
| **Use Case Name** | Generate Report |
| **Trigger** | The staff member wishes to review the vehicle information for the pickup. |
| **Precondition** | The staff member has logged into their account and selects *Generate Pickup Report* after selecting a parent in the *Pickup* menu. |
| **Basic Path** | * The user selects which information they need for the report including, name, plate #, car make, model, color, etc. * The user clicks *Generate Report*. * The user views the report in their browser. |
| **Alternative Paths** | Instead of viewing the report in their browser, the user has the option to download a PDF of the report. |
| **Postcondition** | The user reviews the report for pickup. |
| **Exception Paths** | The user may abandon the operation at any time. |
| **Other** | None |

### Confirm Pickup

|  |  |
| --- | --- |
| **Use Case Name** | Confirm Pickup |
| **Trigger** | A staff member wants to confirm the pickup between a parent and student. |
| **Precondition** | The staff member has logged into their account and is on the *Pickup* page. |
| **Basic Path** | * User selects the parent from the list on the database. * User clicks *Pickup Confirm*. * The system notifies the user about the confirmation of the pickup. * User clicks *Confirm.* * The system confirms the pickup and the pickup is removed from the pickup list. |
| **Alternative Paths** | None |
| **Postcondition** | The *Pickup* page reflects the new pickup being removed from the list. |
| **Exception Paths** | The user can abandon the operation at any time. |
| **Other** | None |

### Update Emergency Alerts

|  |  |
| --- | --- |
| **Use Case Name** | Update Emergency Alerts |
| **Trigger** | The admin wants to send an emergency shutdown alert to reflect a problem at the school and wants to alert all users in the system about the problem. |
| **Precondition** | The admin has logged in to the system and is on the dashboard. |
| **Basic Path** | * The user clicks on *Emergency Alert.* * The system notifies the user that this should only be used on emergency cases only. * The user clicks *I understand.* * The user inputs a description, who should receive the alert and any extra information needed. * The user clicks *Send*. * The system sends the alert to all users that fits the requirements. |
| **Alternative Paths** | None |
| **Postcondition** | The system sends the alert to all fitted users immediately. |
| **Exception Paths** | The user can abandon the alert at any time before submission. |
| **Other** | None |

## 

## Detailed Non-Functional Requirements

### Non-Functional Requirements List

|  |  |
| --- | --- |
| NFR1 | Data is stored permanently |
| NFR2 | Data will update in real time |
| NFR3 | The system is only accessible on campus |
| NFR4 | Admin and Staff accounts are the only one permitted to view Parent, Student, or Vehicle Data(exception: Parents personal data) |
| NFR5 | The systems Date/Time will take into account leap year and daylight savings time |

### Security

### Administrators, Staff, and Parents will all login using username/password credentials. All users will be required to verify via email account. All users will choose their own passwords and have a distinct role in the system (Admin, Staff, Parent). Administrators will be the only users able to access this high-privilege data making them the only ones capable to reset passwords. The Parent accounts will only have access to their personal account information and any alerts sent to them by the school. The Staff will only be able to access the information to the Students and their corresponding Parent or Vehicle. Admins have the ability to terminate an account, denying access to an old user’s privileges to guarantee that previous employees do not remain active on the application after departure. The application will only be usable within the perimeter of the school, this will be established via the schools existing wireless internet. The passwords will be encrypted and stored in a database via a one-way cryptographic hash function. In order to protect from an attack on the database, input sanitization must occur to remove special characters from any text input. The attributes of the program will be enclosed in quotes (“”) to decrease risk of attacks on unquoted attributes.

### 