Software Requirements Specification

Version 1.0

September 24, 2013

Vehicle Pooling System

Submitted in partial fulfillment
Of the requirements of
SE 309 Software Engineering

<<Any comments inside double brackets such as these are not part of this SRS but are comments upon this SRS example to help the reader understand the point being made>>

Refer to the SRS Template for details on the purpose and rules for each section of this document.

TABLE OF CONTENTS

1. Introduction

1.1. Purpose

The purpose of this document is to present a detailed description of the Vehicle Pooling System. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to the external stimuli. This document is intended for both the stakeholders and the developers of the system. The document is also crucial for the maintainability of the software.

1.2. Scope of Project

This software system will be a Vehicle Pooling System for a college student who travels to University by road. This system will be designed to minimize the cost of travelling to University for classes on a daily basis by providing tools to assist in automating the car pooling process, which would otherwise have to be performed manually. By maximizing the traveler's comfort and minimizing the cost, the system will meet the traveler's needs while remaining easy to understand and use.

More specifically, this system is designed to allow a vehicle owner to manage and communicate with a group of interested travelers to share the vehicle on daily basis. The software will facilitate communication between vehicle owners and travelers, via Android Based Smart Phone Application. The back-end services will be performed via private server (or a localhost) .

Do's:

1. Issue of login ids to the students. User Id will be the University Registration number and the password will be set by the students which may be changed by the student as and when (s)he requires to.
2. Create a profile for users after they have registered.
3. Maintain details of daily travelers, i.e. the vehicle owners and the travelers.
4. Maintain the routing details of the commuters on the daily basis.
5. Flash an error message on any invalid use of the system, making the system user friendly.
6. Make a combination such that every student travels to the university with minimum cost and maximum comfort.
7. Take entries at a predefined time every day (preferably), so as to make travelling arrangements one day before the travel.
8. Calculate fare share of each commuter.
9. Show the details of commuter along with the details of their companion, if any.
10. Allow the user to update their profile whenever needed.

Dont's:

1. Travelling arrangements cannot be made dynamically.

2. Routes cannot be defined by the user. Fare calculations will be done on the basis of commonly used routes. It is the vehicle owner’s responsibility to follow the shortest path.
3. Traveler cannot choose the vehicle owner nor can (s)he deny being picked up through the system.

Benefits:

1. Easy and convenient travel.
2. Reduction in the cost of travel.
3. Reduction in the number of vehicles on road as well as in the campus premises thus reducing the pollution level.
4. Increased interaction amongst students.

1.3. Definitions, Acronyms, and Abbreviations:

- VPS: Vehicle Pooling System
- SRS: Software Requirement Specification
- DEO: Data Entry Operator
- Php (): It is used to create dynamic web content and interact with SQL.
- SQL (Structured Query Language): It is used to interact with the databases on the web server.
- HTTP (Hyper Text Transfer Protocol): It is a transaction oriented client/ server protocol between a web browser and a web server.

1.4. Glossary

Term	**Definition**
System User	The person who is using the system in any way.
Administrator	The user who is having all the authorities and rights of the system.
Vehicle Owner	The traveler who owns a vehicle, prefers to travel by his own vehicle and is willing to share with other travelers.
Traveler	Person who does not prefer bringing his own vehicle or does not have his own vehicle and is willing to share with other travelers.
Database	Collection of all the information monitored by this system.
Commuter	Person who is either a Vehicle Owner or Traveler.
Member	A user registered with the Vehicle Pooling System.
Stakeholder	A person that is directly associated with the system, i.e. either

	uses it or develops it. Stakeholder involves the developers, the travelers and the University for which the system is developed.
Software Requirements Specification	A document that completely describes all the functions of a proposed system and the constraints under which it must operate.
User	Any person using the system who is not a developer.
Developer	The person who is into the technical development of the system.

1.5. References

- Object Oriented Software Engineering by Yogesh Singh and Ruchika Malhotra, PHI Learning private Limited,2012
- IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.

1.6. Overview

The next chapter, the Overall Description section, of this document gives an overview of the functionality of the product. It describes the informal requirements and is used to establish a context for the technical requirements specification in the next chapter.

The third chapter, Requirements Specification section, of this document is written primarily for the developers and describes in technical terms the details of the functionality of the product.

Both sections of the document describe the same software product in its entirety, but are intended for different audiences and thus use different language.

2. Overall Description

The Vehicle Pooling System has three active actors and one cooperating system. The Vehicle owner and the traveler accesses the VPS via android based smart phones which interacts with the server through the Internet. Any communication between the actors is through the application.
<< The division of the Vehicle Pooling System is into two component parts, the Application interface and the private server on the internet. >>

The system will have to maintain the following information:

- User Details
- Route Details
- Vehicle Details
- Daily time and location details of the commuters

The users will have to provide following information to the system:

- University Registration number
- Name
- Location from where the user will travel
- Vehicle Details, if any
- Gender
- Gender Preferences, etc.

The DEO will perform functions like maintaining the route details in the database, etc.

The administrator will be the overall controller and operator of the system. He will have special privileges, like maintaining login details, along with the access to the complete functionality of the system.

2.1 Product Perspective

The VPS shall be developed using client/server architecture and will be compatible with Android Operating System. The front-end of the application will be developed using Java and ADT bundle tools. My SQL will be used for the back end management of the system.

2.1.1 User Interfaces

The VPS will have the following user friendly interfaces:

(A) **Login:** to allow the entry of only authorized users through valid login ID and password.
(B) **Traveler Details:** To maintain the details of the travelers.
(C) **Vehicle Owner Details:** To maintain the details of the vehicle owner as well as the vehicles used by the vehicle owners to travel to the University.
(D) **Vehicle Details:** Details of the vehicles used by the Vehicle owners with details such as the mileage will be stored in the system for activities like fare calculation and the maximum number of travelers can be selected by the Vehicle Owner.
(E) **Time of travel:** Each user has to enter the time on the daily basis at which he wishes to travel to the University, preferably during a proper time interval.
(F) **Gender Preference:** A traveler will be visible to a Vehicle Owner only if the gender preference of both the traveler and the Vehicle Owner matches the requirement.

The software should generate the following information:

(a) List of travelers who live at a shorter distance from the University as compared to the Vehicle owner so that he can select as per his convenience, along with the name and location of the traveler, the time at which he will reach that location will be provided.

(b) When the Vehicle Owners selects the traveler and moves on with the system, the fare share for each of the passengers is generated and displayed.

(c) Contact information of the vehicle owner and the information of vehicle is sent to each traveler.

(d) Contact information of the travelers is sent to the Vehicle Owners.

2.1.2. Hardware Interfaces

(a) A device which operates on Android.
(b) Connection to internet when the application is used.

2.1.3. Software Interfaces

(a) User's device should be operating on Android 2.2 or above.
(b) MS SQL Server for back-end.
(c) Eclipse for developing the application.
(d) Php for handling the database.

2.1.4. Communication Interface

In the VPS, communication is via web-enabled services.

2.1.5. Memory Constraints

At least 128 MB of RAM and 200 MB of memory on the memory card will be required to run the software.

2.1.6. Site Adaption Requirements

The terminal at the client site will have to support the hardware and software interfaces specified in sections 2.1.2 and 2.1.3, respectively.

2.2. Product Functions

The VPS will allow access only to the authorized users with specific roles (System Administrator, DEO, Vehicle owner and traveler). Depending on the role of the user, he/she will be able to access specific modules of the system.

A summary of major functions that the VPS shall perform includes:

- A login facility for enabling authorized access to the system.
- The system administrator and the DEO will be able to add, modify, delete or view routing details and user details, if needed.
- The commuters will have to provide their details and register themselves on the system, while using the system for the first time.
- The traveler will be able to register for the next day along with the time at which (s)he will reach a specific location.

- The traveler will be able to get the contact information of the Vehicle owner who selects him to travel along.
- The Vehicle Owner may select travelers which will be most suitable for him to pick up in the way to the University.
- The Vehicle owner will get the contact information of the travelers he has selected.
- Once a traveler has been selected by a Vehicle Owner, he will no longer be visible to other Vehicle owners, hence, the system works on first come first serve basis.

2.3. User Characteristics

Qualification: The user should be a registered student of the University and should be comfortable with English.
Experience: Should be well versed/informed about the process and functioning of the VPS.
Technical Experience: Elementary knowledge of how to use applications on Android based smart phones.

2.4. Constraints

- There will be only one administrator.
- The user sjould preferably enter his requirements for the day only at a specific time.
- The system maintains the details of the specific routes and the locations, which is static.

2.5. Assumptions and Dependencies

- The University will provide details of the students for validations.
- The students will do authentic entries and will not create ambiguity.
- The login ID and password of the student should be kept confidential with himself/herself and should not be given to any other student/non-student whomsoever in any situation.

- The vehicle owner will provide the true mileage of his(her) vehicle.

3. Specific Requirements

This section contains the software requirements in detail along with the various forms to be developed.

3.1. External Interfaces

Login

This form will be available to all the users and can be used to login to the system. Various fields available on this form will be:

- Username: User name consisting of alphabets, numbers or special characters as created by the user during creation of the account is to be entered here.
- Password: Password consisting of alphabets, numbers or special characters as created by the user during creation of the account is to be entered here.
- Sign In: This button is to be clicked if the user has entered the username and the password.
- Sign Up: This button is to be clicked if the user wants to create a new account.

Sign Up

This form will be available when the user wants to register for the VPS. Various fields available on this form will be:

- Name: The user should enter his name which he wants to use as the username. It is highly recommended that the name registered with the University should be entered in this section.
- Email: Email id should be entered which is to be used to update the user about new advancements in the system as well as to inform the new password in case the user forgets his already existing one.
- Password: A password is to be entered by the user which may contain of alphabets, numerals and special characters.
- Confirm Password: The password entered in the above section is to be reentered and the account will be created only if the passwords match.
- Phone Number: Correct phone number is to be entered consisting of 10 digits.
- Gender: Radio button, either male or female is to be selected.

- Gender Preference: Pop-up button. This section is to select the gender preference, if the user has objection in travelling with the other gender.
- Source: Commuter has to select the place from where he will be travelling from.
- Category: User has to select if he is a Vehicle Owner or Traveler/traveler or can login as both.
    - In case the user choses Vehicle Owner or both in the above section, he gets to enter the following information as well.
    - Vehicle Registration Number: This contains the Vehicle Registration number which the user wants to use while travelling to the university.
    - Vehicle Capacity: This contains the capacity of the vehicle which the user wants to use while travelling to the university.
    - Vehicle mileage: This contains the mileage of the vehicle which the user wants to use while travelling to the university.
    - Clear Vehicle Info: Used to clear all the information about a vehicle.
    - Add Vehicle: To be clicked if the user wants to enter another vehicle.
- Create Account: To be clicked when all the information entered by a user is correct and the user wants to end the process by account creation.
- Reset Entries: This button is to be clicked when the user wants to redo all the entries.

Register for next day

This form will be available to the user after he logs into the system. Depending whether he is the traveler or vehicle owner or he has registered as both.

- Traveler

Various fields available on this form will be:

- Select location: Clicking this section will push down a list of locations from which the traveler can be picked by a vehicle owner. A traveler can select only one location.
- Time: Traveler has to enter the time at which he will reach the above mentioned location.
- Done: This button is to be clicked by the traveler only if he has filled the above details correctly. This completes the entry part by the traveler.

- Edit details: This button is for editing of the above filled details other than the primary key.

- Vehicle Owner

Various fields available on this form will be:

- Select Vehicle: Clicking this button will push down a list of vehicles owned by the vehicle owner out of which he can select any one.
- Select location: Clicking this section will push down a list of locations from which the vehicle owner can start his journey. A vehicle owner can select only one location.
- Time: Vehicle owner has to enter the time at which he will reach the above mentioned location.
- Select traveler: This will show the list o travelers who may fall in the route of the vehicle owner along with their location and time to reach that location. The vehicle owner may select a traveler as per his convenience.
- Done: This button is to be clicked by the vehicle owner only if he has filled the above details correctly. This completes the entry part by the vehicle owner.
- Edit details: This button is for editing of the above filled details other than the primary key.

- Both

Under this section, the user will be given a choice if he wants to travel as a traveler or as a vehicle owner for the next day and will be directed to the respective page.

3.2. Functions

3.2.1 Login

- **Use case Description**

Introduction This use case documents the steps that must be followed to log in to the VPS.

Actors • System Users
Pre-condition The user must be a registered member having his/her login ID and password.
Post-condition If the use case is successful, then the actor is logged into the system, otherwise the system state is unchanged.
Basic flow Starts when the actor wishes to log into the VPS. 1. The actor enters his/her login ID and password. 2. The system validates the above information. 3. The actor enters into the system.
Alternate flows **Alternate Flow 1 : Invalid Login ID or password** If in the login flow, the actor enters an invalid login ID and/or password or leaves one or both of the fields empty, the system displays an error message. The actor returns to the beginning of the basic flow. **Alternate Flow 2 : Forgot Password** If the actor forgets his/her password, the actor can go to Forgot Password use case. **Alternate Flow 3 : User exits** This allows the user to exit during the use case. The use case exits.
Special requirements None
Associated use cases Forgot Password

- **Validity Checks**
  - Every user will have a unique login id which will be the student's university registration number

- Login ID can’t be blank
- Login ID will not accept blank spaces and special characters
- Password can’t be blank
- Password shall contain atleast one capital letter and one number.
- Password will not accept blank spaces
- The minimum length of the password must be 6.

- **Sequencing Information**
    - None

- **Error Handling/ Response to abnormal situations**
    - If any of the validation flows do not hold true, appropriate error message will be prompted to the user for doing the needful

3.2.2. Candidate Registration

- **Use case Description**

Introduction This use case documents the steps that must be followed in order to register with the VPS application
Actors • Student
Pre-condition None
Post-condition If the use case is successful, then the student is registered with the system, otherwise the system state is unchanged.
Basic flow Starts when the candidate wishes to register with the VPS. 1. The system requests that the actor specifies the information, like his name, email id, university registration number, gender, gender preferences, phone number, password, type etc. 2. Once the actor provides the requested information, the system validates the provided information and the information is stored.

3. The system then sets the university registration number as the username and registers the student.
Alternate flows **Alternate Flow 1 : Invalid Entry** If the actor enters an invalid phone number, email or other details, the system displays an error message. The actor returns to the beginning of the basic flow **Alternate Flow 2 : User exits** This allows the user to exit during the use case. The user exits.
Special requirements None
Associated use cases None

- **Validity Checks**
    - Only the new candidates will be allowed to register with the system.
    - Every member will have a unique university reg. no and Password.
    - Login ID and password cannot be blank.
    - Password can accept special characters.
    - Email id cannot be blank and should be in the correct format.
    - Phone number cannot be blank.
    - Password should contain atleast one number and one capital alphabet.
    - Password should be greater than 6 digits.
    - If the type is vehicle owner, then the details of the vehicle must be provided.

- **Sequencing Information**

  None

- **Error Handling/ Response to abnormal situations**

  If any of the validation flows do not hold true, appropriate error message will be prompted to the user for doing the needful.

3.2.3. Register for the next day- traveler

- **Use case Description**

Introduction This use case description documents the steps that the traveler should follow in order to use the VPS for traveling the next day.
Actors • Traveler
Pre-condition The traveler must be logged onto the system before the usecase begins.
Post-condition If the use case is successful, then the traveler's information is added into the database. Otherwise, the system state is unchanged.
Basic Flow 1 This use case starts when the traveler wants to travel the next day to the University and the information is to be added in the back-end database. 1. The system requests the user to enter the location at which he wishes to be picked up by a vehicle owner. 2. Once, the location is selected by the traveler, the system requests the traveler to enter the time at which he reaches the above mentioned location. 3. Next the system confirms the details of the traveler and ends the use case with a update in the back-end database. 4. The user may have been picked by a car owner and can check for his updated profile. The user also gets an option to edit his details after he is logged into the system. This is explained in the basic flow 2. **Basic Flow 2** This use case starts when the user wants to edit his pre-entered details other than

the University Registration Number in the back-end data. 1. The system empties all the fields except the name and University Registration number. 2. The system requests the user to enter all the details in the registration page. If the details entered in the registration page are valid then the back-end data is updated.
Alternative Flows **Alternative Flow 1: Invalid Entry** If in the Select Location or Select Time sections, the user does not select any choice, then the system shows a message to choose a correct choice. The actor returns to the basic flow and may re-enter the invalid entries. **Alternative Flow 2: Already registered** If the user has already registered himself for the next day, then on clicking the DONE button, an appropriate error is displayed and the actor returns to his home page where he may proceed with editing his pre-entered details.
Special requirements None
Associated use cases None

➢ **Validity Checks**

- Only the traveler will be authorized to register in the above mentioned way.
- Traveler cannot edit his/her primary key.
- Location is to be selected only from the list of locations available in the system.
- Time should be entered in the 24-hour format.
- A traveler cannot register more than once for a day.
- No column can be left blank.

➢ **Sequencing Information**

None

- **Error Handling/ Response to abnormal situations**

  If any of the validation flows do not hold true, appropriate error message will be prompted to the user for doing the needful

3.2.4. Register for the next day- Vehicle owner

- **Use case Description**

Introduction This use case description documents the steps that the traveler should follow in order to use the VPS for traveling the next day.
Actors • Vehicle owner
Pre-condition The vehicle owner must be logged onto the system before the usecase begins.
Post-condition If the use case is successful, then the vehicle owner's information is added into the database. Otherwise, the system state is unchanged.
Basic Flow 1 This use case starts when the vehicle owner wants to travel the next day to the University and the information is to be added in the back-end database. 1. The system requests the user to enter the location at which he wishes to start his journey. 2. Once, the location is selected by the vehicle owner, the system requests the vehicle owner to enter the time at which he reaches the above mentioned location. 3. Now, the system displays the list of travelers who may fall in the way of the vehicle owner and requests him to select among that list.

4. The system generates the fare share for each commuter using fare calculation use case. 5. The vehicle owner gets the contact information of all the travelers he has selected. 6. Next the system confirms the details of the vehicle owner and ends the use case with update in the back-end database. The user also gets an option to edit his details after he is logged into the system. This is explained in the basic flow 2. **Basic Flow 2** This use case starts when the user wants to edit his pre-entered details other than the primary key in the back-end data. 1. The system empties all the fields except the name and University Registration number. 2. The system requests the user to enter all the details in the registration page. If the details entered in the registration page are valid then the back-end data is updated.
Alternative Flows **Alternative Flow 1: Invalid Entry** If in the Select Location or Select Time sections, the user does not select any choice, then the system shows a message to choose a correct choice. The actor returns to the basic flow and may re-enter the invalid entries. **Alternative Flow 2: Already registered** If the user has already registered himself for the next day, then on clicking the DONE button, an appropriate error is displayed and the actor returns to his home page where he may proceed with editing his pre-entered details.
Special requirements None
Associated use cases Fare Calculation

- **Validity Checks**

  - Only the vehicle owner will be authorized to register in the above mentioned way.
  - Vehicle owner cannot edit his/her primary key.
  - Location is to be selected only from the list of locations available in the system.
  - Time should be entered in the 24-hour format.
  - A vehicle owner cannot register more than once for a day.
  - No column can be left blank.
  - Actor can view the list of only those travelers whose gender preferences can be matched.

- **Sequencing Information**

  None

- **Error Handling/ Response to abnormal situations**

  If any of the validation flows do not hold true, appropriate error message will be prompted to the user for doing the needful.

3.2.5 Maintain Routing Details

- Use case Description

Introduction This use case documents the steps that must be followed in order to add/delete/ update the details of the routes in the system.
Actors • Administrator • DEO
Pre-condition The user must be logged into the system before the use case begins.
Post-condition If the use case is successful, then the routing details are updated/added/deleted

otherwise the system state is unchanged.

Basic flow

Starts when the actor wishes to update/add/delete the details of the route.

1. The actor specifies the function he/she would like to perform (add, delete etc.)
2. Once the actor provides the requested information, one of the following flows are executed:
    - If the actor selects “Add candidate details”, Add route details flow is executed.
    - If the actor selects “Delete candidate details”, Delete route details flow is executed.
    - If the actor selects “Update candidate details”, Update route details flow is executed.

Basic Flow 1 : Add Route Details

1. The system requests the actor to enter the location and the distance of that location from the University.
2. The details are added to the system after the validation.

Basic Flow 2 : Update Route Details

1. The actor enters the location whose details are to be updated.
2. The system validates the above information, if it exists.
3. The actor makes the desired changes to the details.
4. Once the actor updates the information, the system updates the route details with the updated information.

Basic Flow 3 : Delete Route Details

1. The actor enters the location whose details are to be deleted.
2. The system validates the above information, if it exists.
3. The system deletes the record.

Alternate flows

Alternate Flow 1 : Invalid Details

If in the view, delete or update route details flow, the actor enters an invalid location or leaves the fields empty, the system displays an error message.

Alternate Flow 2 : Record already exists

If in the add details flow, a location with the specified information already exists, the system displays an error message. The actor returns to the beginning of the basic flow.

Alternate Flow 3 : Update cancelled
If in the update route details flow, the actor decides to cancel the update, the update is cancelled. The actor returns to the beginning of the basic flow.

Alternate Flow 4 : Deletion cancelled
If in the delete route details flow, the actor decides to cancel the deletion, the deletion is cancelled. The actor returns to the beginning of the basic flow.

Alternate Flow 5 : Deletion not allowed
If in the delete route details flow, the user details are used for form filling, form status, etc. , the deletion is cancelled. The actor returns to the beginning of the basic flow.

Special requirements
None

Associated use cases
None

- Validity Checks
  - Only the Administrator/DEO will be allowed to maintain the route details.
  - Location cannot be left blank and can't accept special characters.
  - Distance must be provided.

- Sequencing Information

  None

- Error Handling/ Response to abnormal situations

  If any of the validation flows do not hold true, appropriate error message will be prompted to the user for doing the needful

3.2.6. View Profile

- **Use case Description**

Introduction
This use case documents the steps that must be followed in order to view the profile.

Actors • Student
Pre-condition The user must have been logged into the system before the use case begins
Post-condition If the use case is successful, then the student is able to view his profile details.
Basic flow 1. The actor logs into the system to view his/her profile 2. The system displays all the details of the actor.
Alternate flows **Alternate Flow 1 : System failure** If due to some technical problems, the application is unable to load the user details, the system returns to the beginning of the use case and tries to reload the profile. **Alternate Flow 2 : User exits** This allows the user to exit during the use case. The user exits
Special requirements None
Associated use cases None

- **Validity Checks**
    - Only a registered student can view his/her profile.

- **Sequencing Information**

    None

- **Error Handling/ Response to abnormal situations**

    If any of the validation flows do not hold true, appropriate error message will be prompted to the user for doing the needful

3.3. Performance Requirements

(a) Should work on any smartphone which works on Android 2.2 or above.
(b) Should run on 200MHz, 128 MB RAM machine or above.

3.4. Design Constraints

None

3.5. Software System Attributes

- **Usability**
  The application will be user-friendly and easy to operate, and the functions will be easily understandable.

- **Reliability**
  The application will be available to the commuters throughout their enrollment in the university and has a high degree of tolerance.

- **Security**
  The application will be password protected. Users will have to enter correct login ID and password to access the application.

- **Maintainability**
  The application will be designed in a maintainable manner. It will be easy to incorporate new requirements in the individual modules.

- **Portability**
  The application will be easily portable on any android-based system that has the required hardware and software.

3.6. Logical Database Requirements

The following information will be placed in a database :

Table name	Description
Users	Records the details of the user
Routes	Records the various routes
Travellers	Records the daily time and location of travellers along with the vehicle owners' details, if any

3.7 Other Requirements

None