**Software Requirements Specification**

**for**

**Carpool Management System**

**Version 1.0 approved**

**Prepared by Usaid Bin Rehan, Umer Wasi and Huzaifa Jawad**

**NUCES**

**4 December 2022**

**Table of Contents**

**Table of Contents ii**

**Revision History ii**

**1. Introduction 1**

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Product Scope 1

1.5 References 1

**2. Overall Description 2**

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Classes and Characteristics 2

2.4 Operating Environment 2

2.5 Design and Implementation Constraints 2

2.6 User Documentation 2

2.7 Assumptions and Dependencies 3

**3. External Interface Requirements 3**

3.1 User Interfaces 3

3.2 Hardware Interfaces 3

3.3 Software Interfaces 3

3.4 Communications Interfaces 3

**4. System Features 4**

4.1 System Feature 1 4

4.2 System Feature 2 (and so on) 4

**5. Other Nonfunctional Requirements 4**

5.1 Performance Requirements 4

5.2 Safety Requirements 5

5.3 Security Requirements 5

5.4 Software Quality Attributes 5

5.5 Business Rules 5

**6. Other Requirements 5**

**Appendix A: Glossary 5**

**Appendix B: Analysis Models 5**

**Appendix C: To Be Determined List 6**

**Revision History**

| **Name** | **Date** | **Reason For Changes** | **Version** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |

# **Introduction**

## **Purpose**

It contains the software requirements specified for Carpool Management System version 1 with entire scope of the project covered.

## **Document Conventions**

Content:

Font style: Arial

Font size: 11

Font decoration: none

Headings:

Font style: Times New Roman

Font size: 14

Font decoration: Bold

## **Intended Audience and Reading Suggestions**

It is intended for the developers, project managers, testers, documentation writers and the course instructor. It contains the overall description, external interface requirements, system features, and other non-functional requirements of the project and is organized in the standard IEEE format. The suggested sequence for reading the document is from start to end to get the complete overview of the project in order to avoid ambiguity of the scope and limitations of the project.

## **Product Scope**

The Carpool Management System is a social website with the purpose of allowing the private commuters of NUCES to connect with each other for benefits that come with ridesharing like division of cost, economical transport regardless of ownership of vehicle and bond with unfamiliar commuters. The objective is to connect through a unified and secure social platform where they can offer or search for economical rides efficiently. The goal is economical deployment of a software that has potential to be updated to fulfill all the needs of commuters to any organization.

## **References**

https://github.com/Usaid-Bin-Rehan/NUCES\_Carpool\_MVP

# **Overall Description**

## **Product Perspective**

This is a new self-contained product.

## **Product Functions**

The software allows NUCES commuters to sign-up/in, edit their profile, and opt as driver to create a group that others can join as passengers or join one or view and leave a joined group as passenger with the option of filtering the group by the commute details, that are stored as history to be accessible to admin as evidence who may want to block violators by removing their credentials from the database or use it to persuade advertisement clients for monetization that covers deployment costs, or the driver’s preferences. It also allows developers to add features of notification and routes visualization.

## **User Classes and Characteristics**

The user class is implicitly divided into user and database administrator based on privilege level where the administrator has the authority to view history of transactions for evidence of blocking violators and deal with advertisement clients for payment of infrastructure hosting. The most important user class is the commuters to NUCES which is explicitly divided into passengers and drivers that takes precedence over other commuters with whom there may be a mutually beneficial relationship but with security taking precedence, they are not allowed to register on the website.

## **Operating Environment**

The server and the database will operate on a cloud environment and will have client accessible on Windows, Linux, Mac, Android and IOS devices.

## **Design and Implementation Constraints**

The regulatory policy of using a data intensive framework, not allowing development using a compute intensive framework, and with necessary features make it economical for the website to be hosted using advertisement revenue in developing countries. The features discarded are: timed deletion of group, administrative login, requesting for a group, chat system, penalty system, password recovery system, review system and bargaining system. Currently, lacking update and delete group, stop driver from join and leave group, notify, stop re-signup and visualize routes.

## **User Documentation**

https://github.com/Usaid-Bin-Rehan/NUCES\_Carpool\_MVP

## **Assumptions and Dependencies**

The assumption that could affect the requirements stated in the SRS is that, this version as an MVP is only intended to be functional as long as the product version is not deployed therefore minimal consideration of UI and UX principles, that are to modified in the final product individually for Android, IOS, and Windows / Linux and Mac, where the other assumption is to use a cross-platform language / framework for development for economic development. Another assumption is minimal maintenance once the final version deployed hence brainstorming scenarios can delay it.

# **External Interface Requirements**

## **User Interfaces**

The UI when a user visits the website is that there are two options, sign-in and sign-up, with blank fields where the user has to enter appropriate credentials or else the respective errors will be displayed. If the user correctly enters the credentials, he is routed to edit profile UI that is blank if signed-up or filled if signed-in. The navigation bar has semester and once group tabs where group boxes, with basic details, can be scrolled, drop-down filtered or opened to view more details or join. There are also create and view joined group buttons where latter also allows leave the group.

## **Hardware Interfaces**

The logical and physical characteristics of interface between the client’s device and server are that the client’s browser runs the front-end whereas multiple CPU nodes throughout the world run the server. The supported device types are Android, IOS, Windows, Mac and Linux. The communication protocols used are that used by NodeJs.

## **Software Interfaces**

NodeJs allows browser to use JavaScript for server-side rendering while the templating engine EJS provides rendering of client-side templates. The cloud vendor provides interface between the server and the database, which is outside the scope of this document. The interface to NPM packages from packages.json is provided by NPM Package Manager. The data items coming into the system are the POST requests and going out of the system are the GET requests that are handled asynchronously by NodeJs. The nature join and leave communication is transaction.

## **Communications Interfaces**

Not Applicable

# **System Features**

## **Sign-in/up**

4.1.1 Description and Priority

Medium Priority

4.1.2 Stimulus/Response Sequences

User visits the website and then enters the credentials and wait for it to be verified or deal appropriately with a respective error.

4.1.3 Functional Requirements

Verification of Sign-up: Conditional Statements in Js

Verification of Sign-in: Fetch from MySQL and Conditional Statements in Js

## **Edit Profile**

4.1.1 Description and Priority

High Priority

4.1.2 Stimulus/Response Sequences

After passing sign-in/up, the user enter details or skips the edit profile if not blank.

4.1.3 Functional Requirements

Enter Details: Conditional Statements in Js

Store in Database: Post to MySQL and Conditional Statements in Js

## **Create Group**

4.1.1 Description and Priority

High Priority

4.1.2 Stimulus/Response Sequences

The driver checkboxes to set profile as driver when editing profile, then clicks on create group button and fill the necessary details.

4.1.3 Functional Requirements

Checkbox: Conditional Statements in Js

Enter Details: Post to MySQL and Conditional Statements in Js

## **Filter to Join and View Joined to Leave Group**

4.1.1 Description and Priority

Low Priority

4.1.2 Stimulus/Response Sequences

Passenger visits either “Semester” or “Once” group and may click on dropdown filters to view groups of his choice and then clicks on a group to see the preferences of driver and click on the join button. The user can also leave the joined group after clicking on “view joined” button and then the leave group button.

4.1.3 Functional Requirements

Filter the database: Conditional Statements in Js and MySQL

Perform transactions: Post to MySQL and Transactions in Js

# **Other Nonfunctional Requirements**

## **Performance Requirements**

No explicit performance requirements except that a data-intensive framework must be used.

## **Safety Requirements**

The safety requirement is to take appropriate precautions for travelling with a relevant commuter.

## **Security Requirements**

The requirements related to NUCES email are handled by NUCES itself. Make sure to use a SIM that is not used for any activity other than carpooling and removed to avoid being contacted.

## **Software Quality Attributes**

Reusability over Adaptability over Availability over Usability over Reliability over Correctness

## **Business Rules**

The requirement is to create a MVP that can be used for developing an economic, cross-platform, data-intensive product as well as be used till the product is deployed, where both are hosted by advertisement revenue and require minimal maintenance and administration.

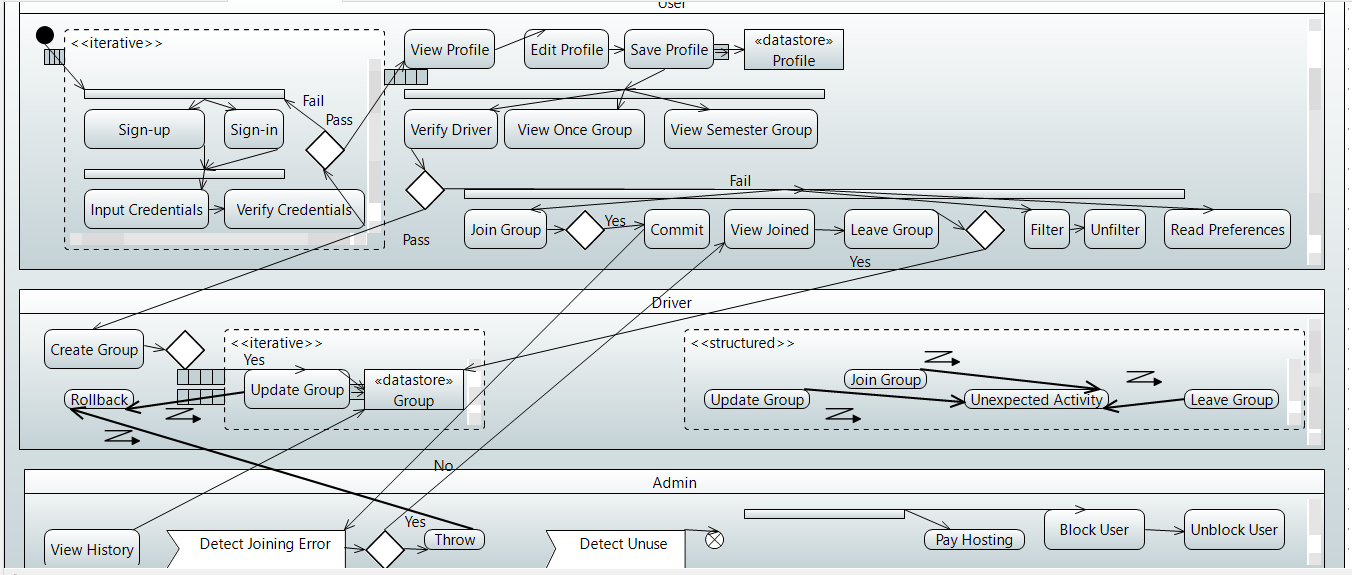
# **Other Requirements**

**Appendix A: Glossary**

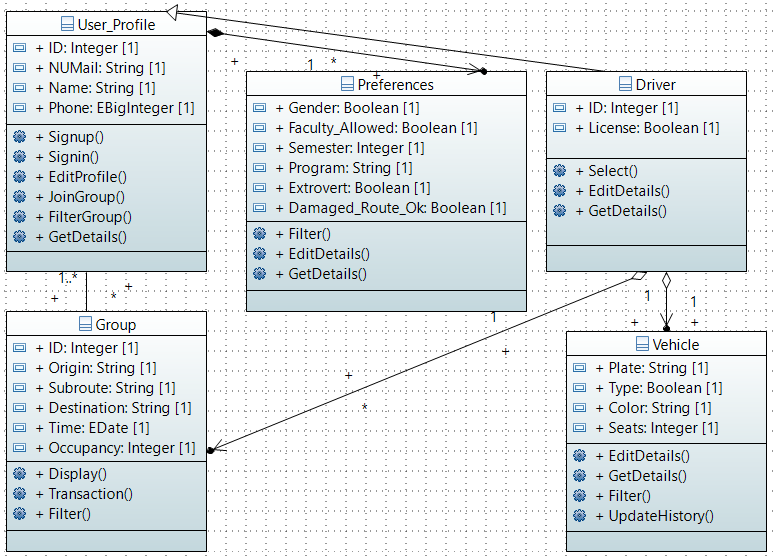
NUCES: National University of Computer and Emerging Science  
Js: JavaScript

**Appendix B: Analysis Models**

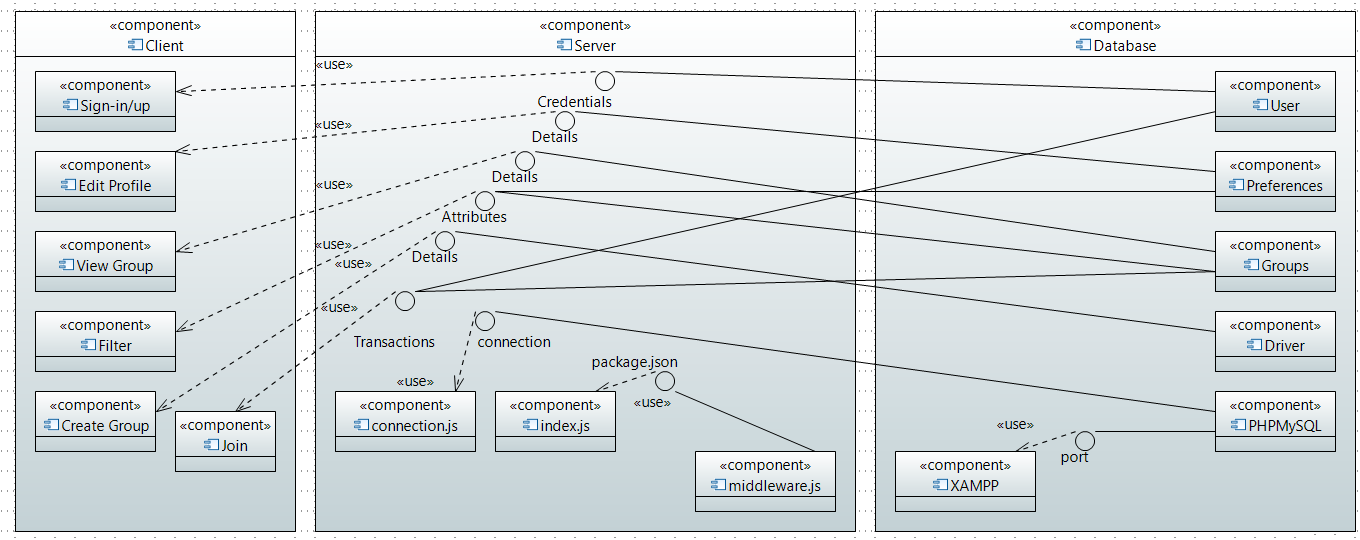
Activity



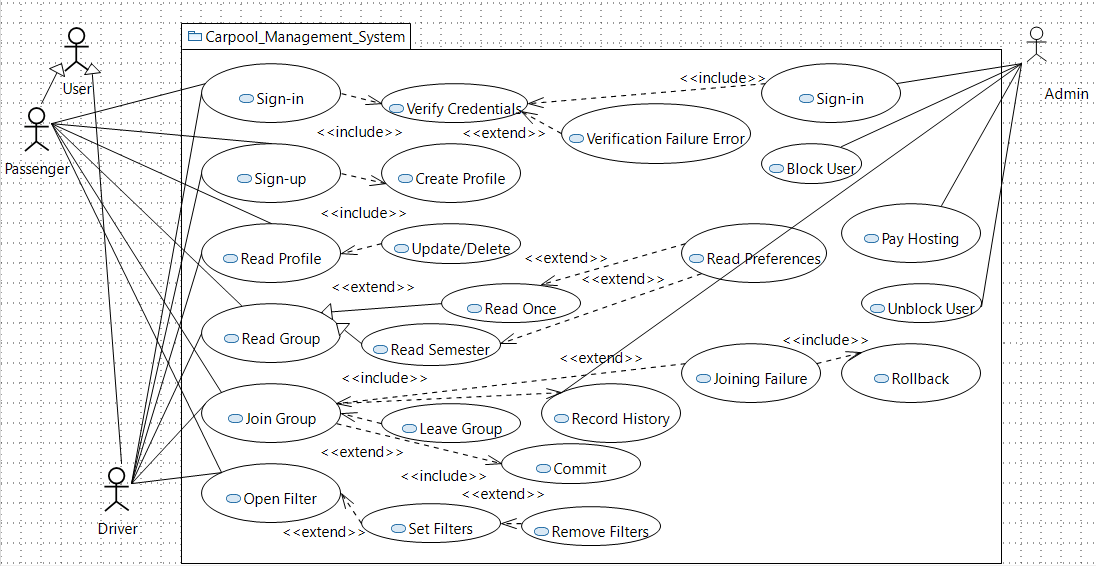
Class



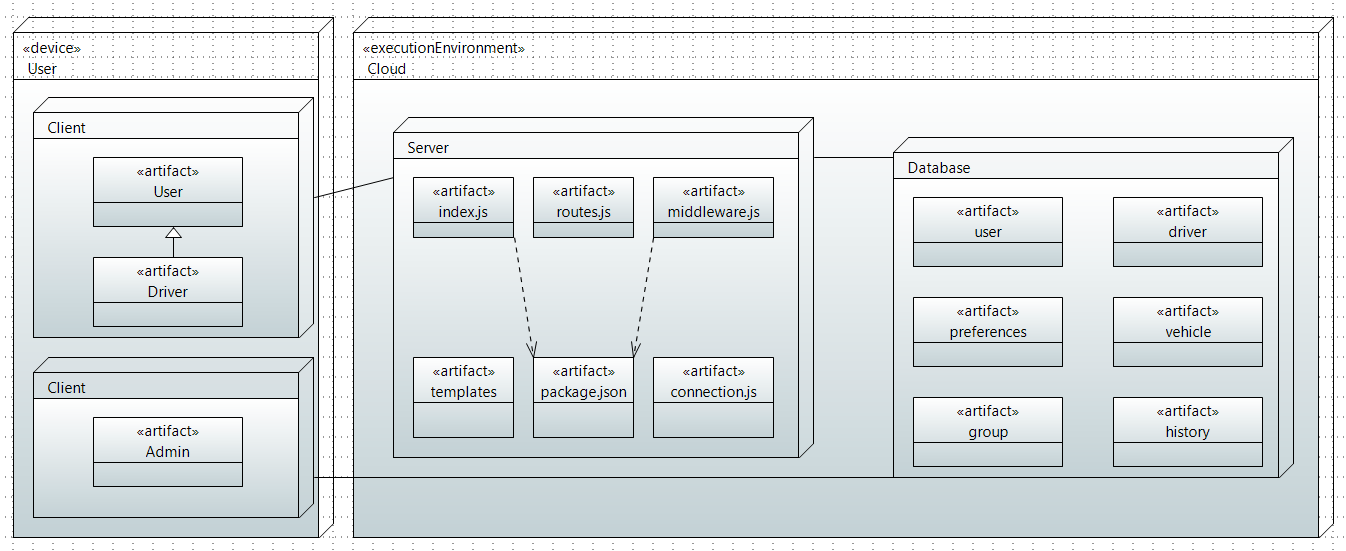
Component



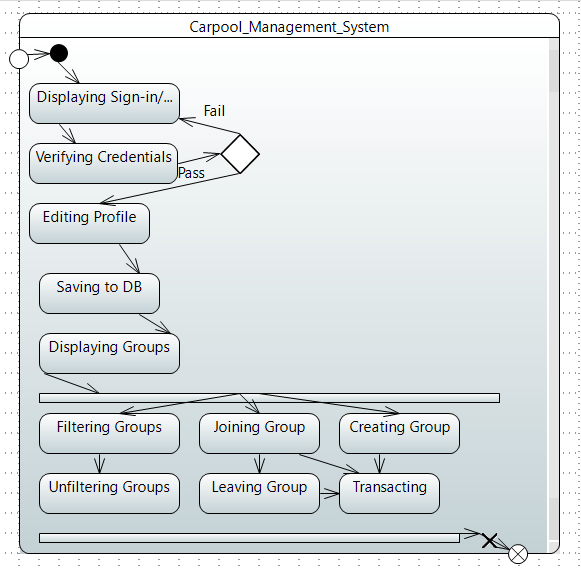
Use-Case



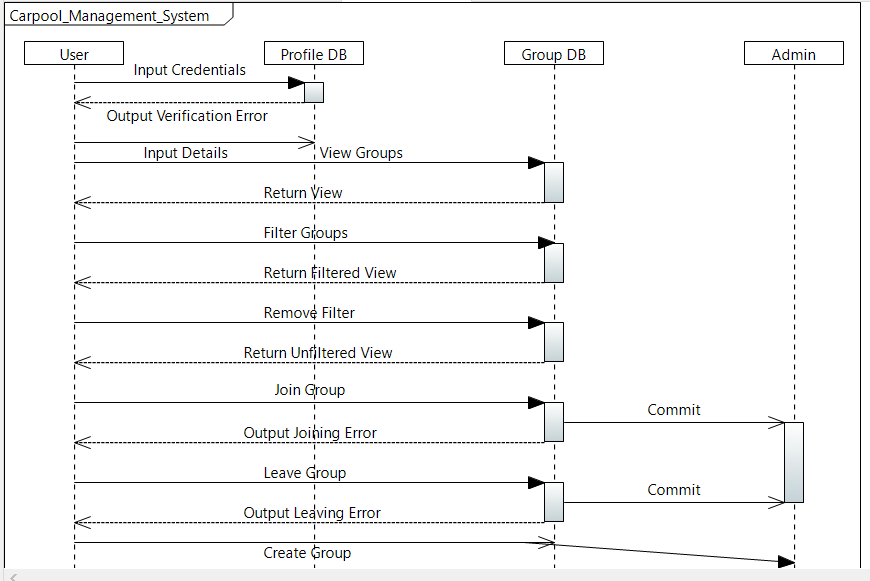
Deployment



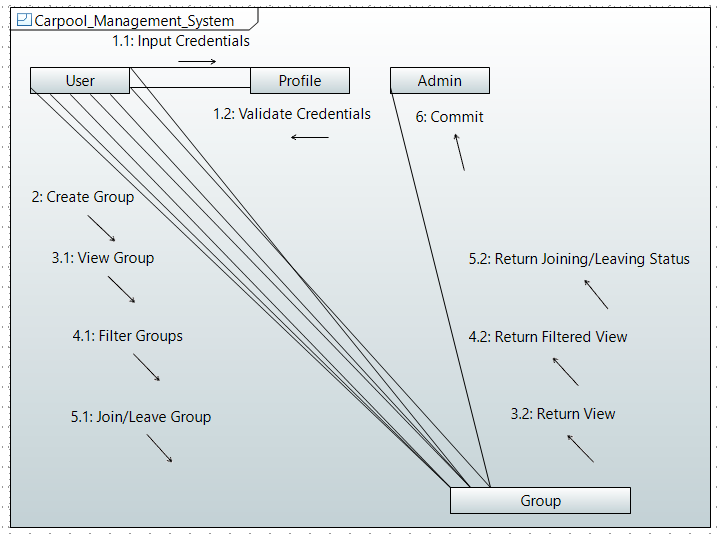
State



Sequence



Communication / Collaboration



**Appendix C: To Be Determined List**

https://github.com/Usaid-Bin-Rehan/NUCES\_Carpool\_MVP