**Sri Lanka Institute of Advanced Technological Education Higher National Diploma in Information Technology**

**Higher National Diploma in Information Technology**



**System Requirement Specification Document**

**“LS Transport”**

(Desktop Application for Cab Service)

|  |  |  |
| --- | --- | --- |
| Name Of the student | Register Number | Signature |
| H.A.L.E.S.Gunasekara | ANU/IT/2019/F/0055 |  |

(HNDIT2404)

Project(individual)

Supervisor: Mr. Pavaridu sahansith

Academic Year: 2019/2021 Year 2-Semester 02(Fulltime)

Submission Date: (12/10/2021)

**TABLE OF CONTENT**

1. Introduction……………………………………………………………………….03
2. Scope of the project…………………………………….........................................04
3. Functional requirements……………………………………………………….… 05
   1. Actor description………………………………………………….………….05
   2. Use case diagram…………………………………………………………….06
   3. Case description……………………………………………………...………07
      1. Case 01…………………………………………………………………..07
      2. Case 02…………………………………………………………………..08
      3. Case 03…………………………………………………………………...09
      4. Case 04…………………………………..………………………………10
      5. Case 05…………………………………………………………..………11
      6. Case 06……………………………………..……………………………12
      7. Case 07…………………………………………………………..………13
      8. Case 08………………………………………………………….……….14
      9. Case 09………………………………………………………….……….15
      10. Case 10…………………………………………………….…………..16
      11. Case 11……………………………………………………..………….17
4. Non-functional requirements………………………………………….…………..18
5. **Introduction**

The SRS document is submitted to meet software development requirements of the individual project module conduct by the Sri Lanka Institute of Advanced Technological Education (SLIATE).

This SRS document describes the objectives, requirements and nature of the project intended to develop the software. There are a number of guidelines to follow when creating this document.  this includes the purpose, scope, functional and nonfunctional requirements, it also contains the information about environmental conditions required, safety and security requirements, software quality attributes of the project etc.

This is a desktop application designed for a car rental company “LS transport”, today people use vehicles for their daily commute, but there is no formal procedure for doing so, once this application is set up, you will be able to perform those tasks with minimal need.

1. **Scope of the project**

LS transport is a user- friendly fancy computer application that can be connected through any computer.

Here the admin can add users will not be able to connect to this software without the inclusion of users. The admin will enter the username and password after entering the username and password. User can then enter it and link to the app.

Use can perform activities such as vehicle registration, driver registration, booking, customer registration

User can negotiate with their clients and reserve vehicles at their discretion.

Here will find a summary of drivers and clients currently registered with the software, as well as incomes of current software users and existing vehicles.

End users can log out

1. **Functional requirements**
   1. **Actor description**

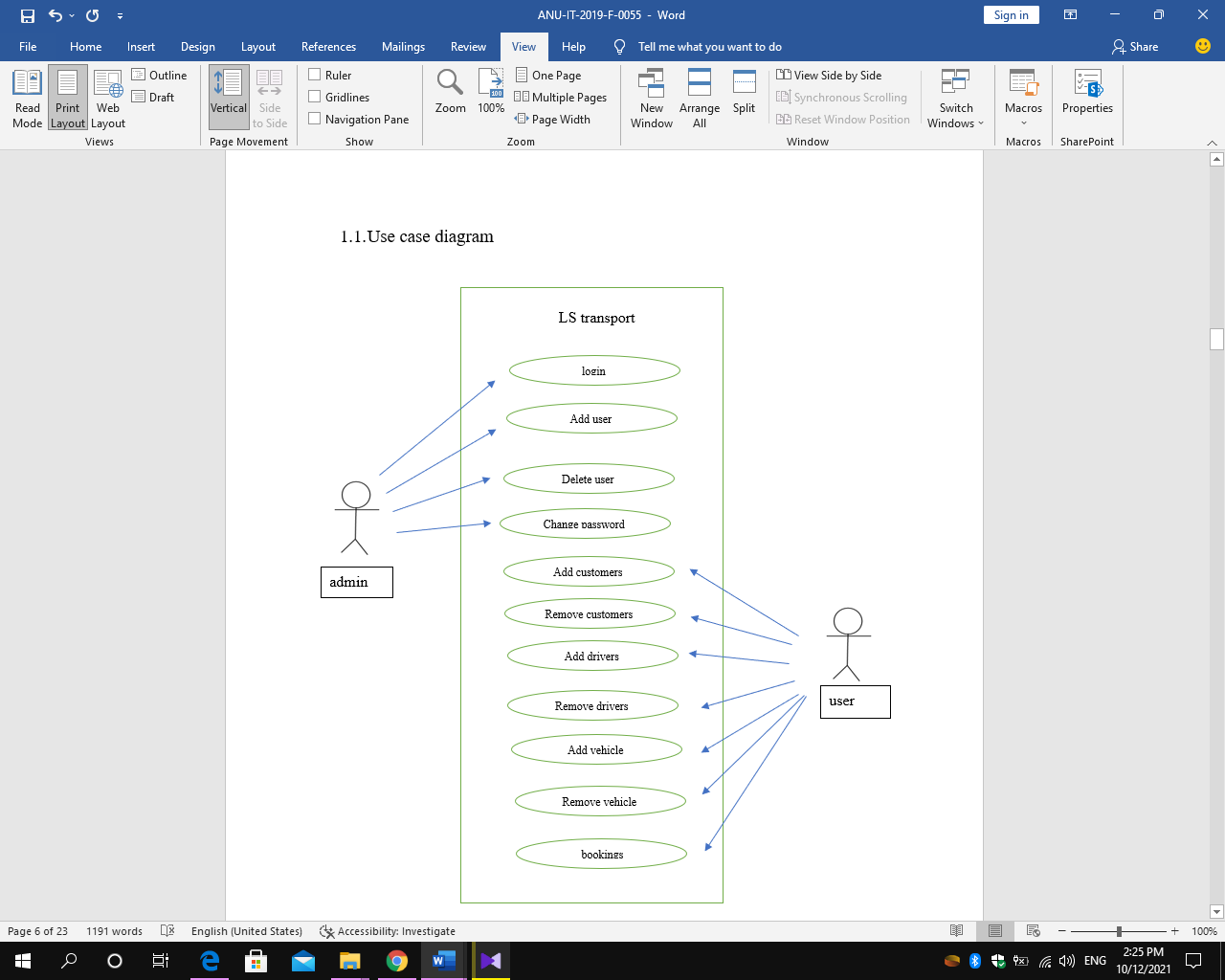
**Admin**

Admin is the most important component of the system. Here the user is the one who adds to the system.

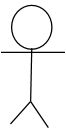
**User**

The user has a large role to play in the system, for example, entering vehicles, entering drivers, and making vehicles bookings. Simply put, the user performs all the functions of the system after adding the admin user.

* 1. Use case diagram



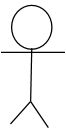
* 1. Case description
     1. Case 01



admin

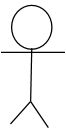
|  |  |
| --- | --- |
| Case name | login |
| actor | admin |
| overview | Admin need to login to the application |
| description | * Open application * Go to login page * Fill login information * Click login button |
| Pre- conditions | * Admin should open the application * Admin must enter the password first |
| Post-conditions | Display the message and redirect to the users add page. |
| Alternative flows | At the time of login, you enter the login button without a password error or password, display relevant error message. |

* + 1. Case 02



admin

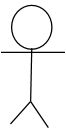
|  |  |
| --- | --- |
| Case name | Add user |
| actor | admin |
| overview | Admin can add users |
| description | * Go to users add page. * insert users details * Admin provides password and click save button |
| Pre-condition | Admin must log in to the application |
| Post-condition | View the added users. |

3.3.3 Case 03

admin

|  |  |
| --- | --- |
| Case name | Delete user |
| actor | admin |
| overview | Admin can delete user |
| description | * Go to users add page. * After admin can delete users |
| Pre-condition | * Admin must log in to the application * And users must have pre-entered |
| Post-condition | Users remove the application |

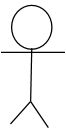
3.3.4 Case 04



admin

|  |  |
| --- | --- |
| Case name | Change password |
| Actor | admin |
| Overview | Admin can change password |
| Description | * Go to users add page. * After admin can change password * Click save button after change password. |
| Pre-condition | * Admin must log in to the application * It must have included users. |
| Post-condition | Changed the user’s password. |

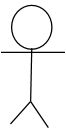
3.3.5 Case 05



user

|  |  |
| --- | --- |
| Case name | Add customer |
| Actor | user |
| Overview | User can add customers |
| Description | * User should login to the application using his own user name and password * Go to customer add page * User enter the customer information and click save button |
| Pre-condition | User must log in to the application |
| Post-condition | Add customers and show customers |

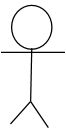
3.3.6 Case 06



user

|  |  |
| --- | --- |
| Case name | Remove customer |
| Actor | user |
| Overview | User can remover customer |
| Description | * Go to user add page * Remove customer * Click save button |
| Pre-condition | * User must log in to the application * It must have included customers |
| Post-condition | Deleted customer |

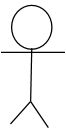
3.3.7 Case 07



user

|  |  |
| --- | --- |
| Case name | Add driver |
| Actor | user |
| Overview | User can add driver |
| Description | * User should login to the application using his own user name and password * Go to the driver add page * User enter the drivers information and click save button |
| Pre-condition | User must log in to the application |
| Post-condition | Add driver and show driver |

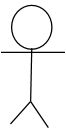
3.3.8 Case 08



user

|  |  |
| --- | --- |
| Case name | Remove driver |
| Actor | user |
| Overview | User can remover driver |
| Description | * Go to driver add page * Remove driver * Click save button |
| Pre-condition | * User must log in to the application * It must have included drivers |
| Post-condition | Deleted driver |

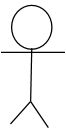
3.3.9 Case 09



user

|  |  |
| --- | --- |
| Case name | Add vehicle |
| Actor | user |
| Overview | User can add vehicle |
| Description | * User should login to the application using his own user name and password * Go to the vehicle add page * User enter the vehicle information and click save button |
| Pre-condition | User must log in to the application |
| Post-condition | Add vehicle and show vehicle |

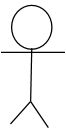
3.3.10 Case 10



user

|  |  |
| --- | --- |
| Case name | Remove vehicle |
| Actor | user |
| Overview | User can remove vehicle |
| Description | * Go to vehicle add page * Remove vehicle * Click save button |
| Pre-condition | * User must log in to the application * It must have included vehicle |
| Post-condition | Deleted vehicle |

3.3.11 Case 11



user

|  |  |
| --- | --- |
| Case name | booking |
| Actor | user |
| Overview | User can booking vehicle |
| Description | * User should login to the application using his own user name and password * Go to the booking page * Check availability of drivers and vehicles * User enter booking information and click save button |
| Pre-condition | User must log in to the application  It must have drivers and vehicles already available |
| Post-condition | Booking vehicle |

1. **Non-functional requirements**

**Non-Functional Requirement** (NFR) specifies the quality attribute of a software system. They judge the software system based on Responsiveness, Usability, Security, Portability and other non-functional standards that are critical to the success of the software system.

* **Performance**

System performance defines how fast a system can respond to a particular user’s action under a certain workload.

The application load time should not be more than one second for users.

* **Reliability**

Reliability is the probability and percentage of the software performing without failure for a specific number of uses or amount of time.

Applicants can access their resume 98% of the time without failure.

* **Availability**

This feature defines the amount of time the system is running, the time it takes to repair a fault, and the time between lapses.

* **Maintainability**

This feature indicates the average time and ease and rapidity with which a system can be restored after a failure.

* **Security**

Security measures ensure your software’s safety against espionage or sabotage.

Only the users with the role “admin” can view the applicant’s verified phone number.