Software Requirements Specification

Version 1.0
<<Annotated Version>>

March 20,2021

Travel Agency Automation System

P Anurag Reddy
BV Abhiram

Submitted in partial fulfillment
Of the requirements of
CS29006 Software Engineering Lab

Table of Contents

Table of Contents 2
1. Introduction 3
1.1 Purpose 3
1.2 Document Conventions 3
1.3 Intended Audience and Reading Suggestions.. 3
1.4 Product Scope 3
1.5 References 3
2. Overall Description 4
2.1 Product Perspective 4
2.2 Product Functions 5
2.3 User Classes and Characteristics 4
2.4 Operating Environment 6
2.5 Design and Implementation Constraints 6
2.6 User Documentation 6
2.7 Assumptions and Dependencies 6
3. External Interface Requirements 6
3.1 User Interfaces 6
3.2 Hardware Interfaces 7
3.3 Software Interfaces 7
3.4 Communications Interfaces 8
4. System Features 8
4.1 To check the availability of a particular car.. 8
4.2 To check when can a specific car be available ……….. 8
4.3 To see all the available cars…………….. 8
4.4 Payment of advance …….. 9
4.5 Return a car ……….. 9
4.6 Lookup Database ……….. 9
4.7 Update-Database ……….. 9
4.8 Record data ………..10
4.9 Calculate stats ……….. 10
4.10 Set Price ……….. 10
4.11 Decide Fate of Car ………..10

5. Other Non-functional Requirements 11
5.1 Performance Requirements 11
5.2 Safety Requirements 11
5.3 Security Requirements 11
5.4 Software Quality Attributes 11
5.5 Business Rules 11
6. Other Requirements 12
Appendix A: Glossary 12
Appendix B: Analysis Models 12
Appendix C: To Be Determined List 12

1.Introduction

1.1 Purpose

The purpose of this document is to give a detailed description of the Travel Agency Automation System. It will explain the system's purpose and features, the interfaces of the system, what the system will do, the constraints under which it must operate, and how the system will react to external stimuli. This document is intended for both the stakeholders and the developers of the system.

1.2 Document Conventions

Name	Meaning
Owner	The person who owns the business which is going to use this software.
Customer	The person who is going to rent the car.
Database	The excel sheet in which all the data is stored.
Model	The model of the vehicle available in the travel agency for renting.
Excel	Microsoft Excel

1.3 Intended Audience and Reading Sessions

This project is a prototype for the Travel agency automation system, and it is restricted within the college premises. This has been implemented under the guidance of college professors and teaching assistants. This project is helpful for business owners as well as for customers.

1.4 Product Scope

This software system will be a Travel Agency Automation System for Travel Agencies interested in automating their functioning mode. This system will replace the normal ledger-based functioning of the business. This system acts as a subsidiary to the owner by providing a customer another way to access the data required by him to decide or book a vehicle. This will save the time of both the owner and customer. Hence making more money for the business and providing a smoother experience to the customer.

1.5 Reference

Ravi Bandakkanavar. (2018, July 4). *Software Requirements Specification document with example - Krazytech*. https://krazytech.com/projects/sample-software-requirements-specificationsrs-report-airline-database

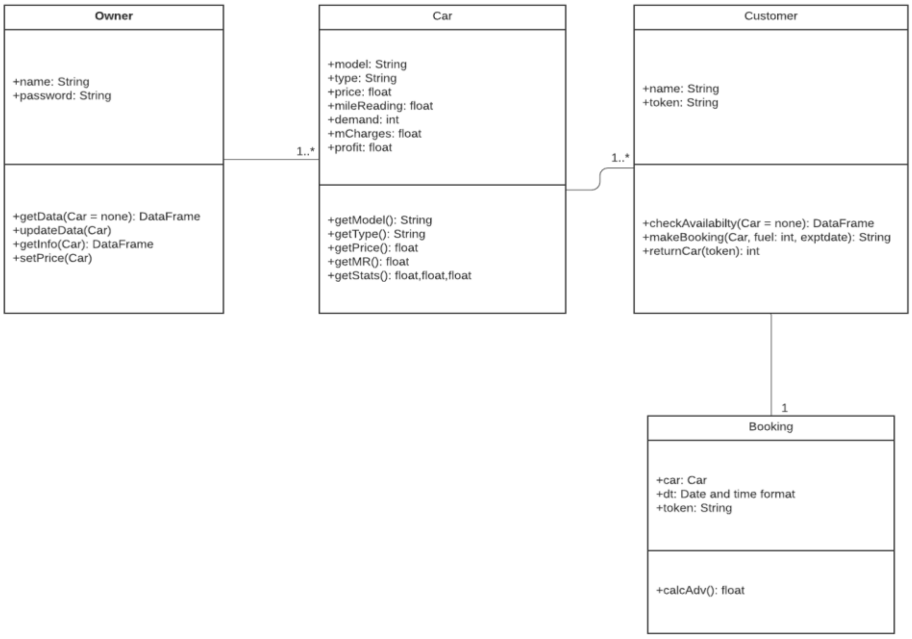
2. Overall Description

2.1 Product Perspective

Travel agency automation software (TAAS) automates the company's various operations, which lets the customer select and browse the cars on their own. Also, the owner can check the cars' statistics and choose which car to sell or buy according to the profit they make. It plays the role of a human receptionist at the company and provides data more conveniently.

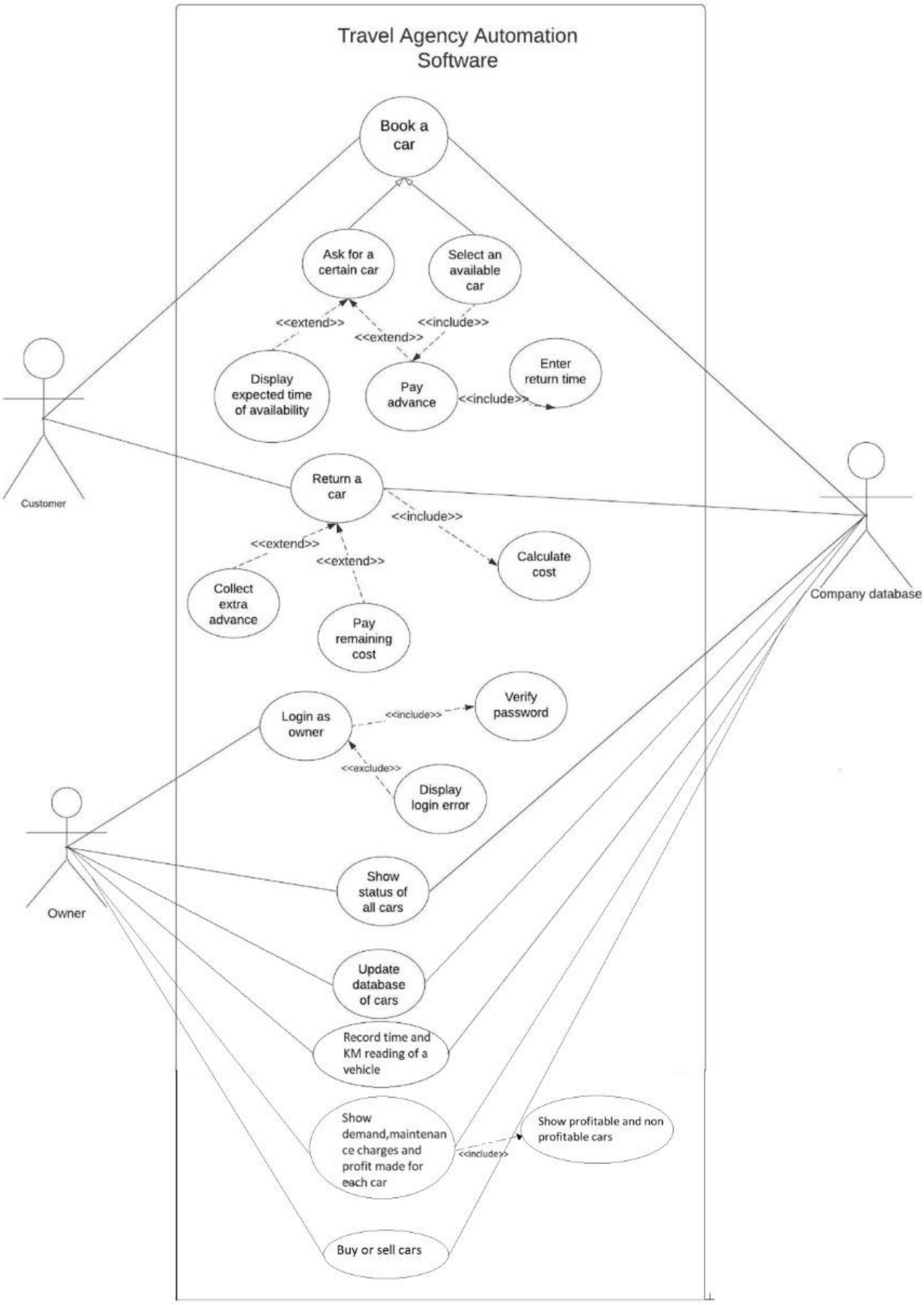
2.3 User Classes and Characteristics

The two user classes which are in the product are the customer and the owner. The customer is the one who wants to take a car for rent, and the owner is the one who can log in as an admin and has unique access to statistics of the cars and the database.
To understand the functions and other attributes used in the classes of TAAS, its class diagram is shown below;



2.2 Product Functions

For a better understanding of functions of TAAS, its use case diagram is shown below;



2.4 Operating Environment

The operating environment for the travel agency automation system is listed below

- Operating system: windows
- Access to the database: pandas library in python
- Platform: Python

2.5 Design and Implementation Constraints

Tools to be used are python, excel, and windows. Requires Python 3.4 or higher. Made packages of individual classes and made them into a package installed in each pc of the business for implementation.

2.6 User Documentation

Must know how to use the command prompt in windows for running the package.

15 Windows Command Prompt (CMD) Commands You Must Know, from https://www.makeuseof.com/tag/15-cmd-commands-every-windows-user-know/

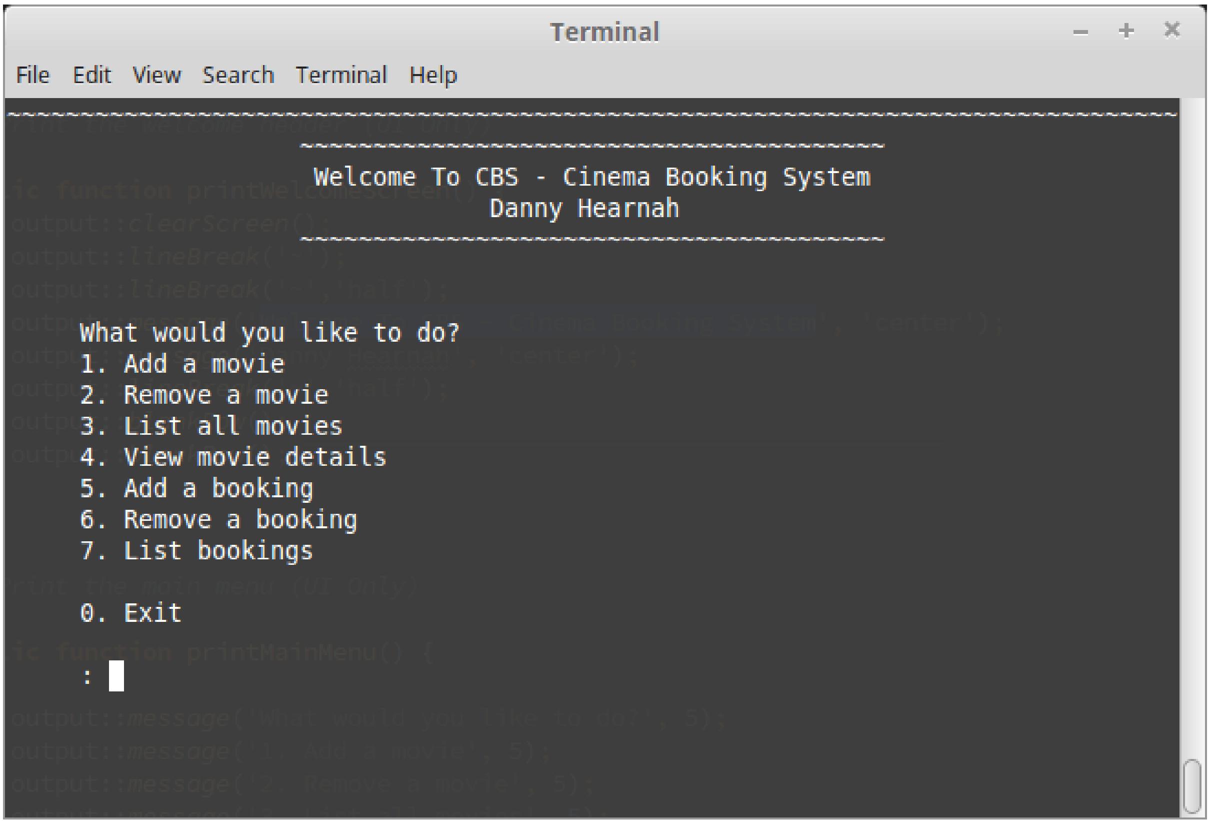
2.7 Assumptions and Dependencies

Excel is required to be installed on all the PCs of the business. The PCs' operating system is assumed to be Windows, but it would not cause any problem to the application. Python 3.4 or higher versions need to be installed on the PCs for the application to run.

3.External Interface Requirements

3.1 User Interfaces

Front-end: Text-based interface.



```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                    ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
                     Welcome To CBS - Cinema Booking System
                                 Danny Hearnah
                    ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~


     What would you like to do?
     1. Add a movie
     2. Remove a movie
     3. List all movies
     4. View movie details
     5. Add a booking
     6. Remove a booking
     7. List bookings

     0. Exit

     : 
```

Text-based interface example, source google images.

Back-end: Python, mainly Pandas library.

3.2 Hardware Interfaces

Devices of the windows operating system are supported. The data is stored in the form of excel workbooks or spreadsheets, which are accessed through python and used as necessary.

3.3 Software Interfaces

Software Used	Description
Windows	We both use windows-based laptops. So, our application is windows compatible but will also run other Operating Systems.
Python	Python of version 3.4 and higher is preferred for running the application.
Excel	We used MS Excel for storing data about models, renting records and for easy access using pandas library of python.
Pandas library	Pandas of version 1.2.1 and higher is preferred for running the application.

3.4 Communication Interfaces

There is no communication between any two users, and the software is being designed to be deployed locally on a pc with all the data. So, there is no communication with the internet too.

4.System Features

4.1 To check the availability of a particular car

4.1.1 The customer can be able to ask for the availability of a particular car which he could take for rent
4.1.2 In the options after selecting 'book a car,' there will be an option called 'ask for availability.
After selecting it, the system will ask the user to enter the car's name and if they need it with AC or non-AC. Then the system will output if it is available or not. If the ordered car is available, the user can proceed to payment.
4.1.3
REQ1: the input, which is a car name, should be among the cars which the company provides; otherwise, the system will print" Our company does not provide **________** cars" (with the car name in the blank)
REQ2: The system should have access to the company database

4.2 To check when can a specific car be available

4.2.1 From feature 4.1, if the asked car is not available at the moment, with this feature, the user will be able to ask the system when can the car be available
4.2.2 There will be an option after the system says the car is not available at the moment. The system checks the returning time of the previous customers of the car and outputs the nearest time.
4.2.3
REQ1: The system should have access to the company database

4.3 To see all the available cars

4.3.1 The user can choose this feature other than 4.1. To see all the cars available at the moment and choose from them.
4.3.2 After selecting the 'Book a car' option, this option is called 'see all available cars. ' If the user selects the option, then the system will output all the available cars. The user can choose any car from it then the user will proceed to payment.
4.3.3
REQ1: The system should have access to the company database

4.4 Payment of advance

4.4.1 The user will pay a minimum advance to take the required car for rent
4.4.2 When the user proceeds to payment, the system will display the minimum advance required to pay. If the customer pays it, the system notes down the car's initial readings and asks the user for an expected time of return. The system updates the database, and the car will be the customers to take for rent.
4.4.3
REQ1: A valid expected time of return should be inputted; otherwise, "invalid return time" will display on the screen
REQ2: The system should have access to the company database

4.5 Return a car

4.5.1 The customer can return a car using this feature. The customer will proceed to the payment section
4.5.2 There will be an option to return a car in the interface. After selecting it, the system will ask the user for the name of the car and token number. The system will ask the user to enter the mile reading of the user's car. Then the system calculates the cost for the car and outputs it.
4.5.3
REQ1: The customer must input the mile-reading and token number correctly; otherwise system outputs" Invalid Input."
REQ2: The system should have access to the company database

4.6 Lookup Database

4.6.1 The owner would be able to access the database to look at the number of cars and their present status. i.e., in repair, available, or rented.
4.6.2 Choose the Owner in the login options and enter a password. After that, choose the option of lookup database in the given options. Can enter a model if you want to look at the data of one car only or look at the complete database.
4.6.3
REQ1: Password is a unique string.
REQ2: If the password is wrong, then outputs message" Wrong password."
REQ3: If the wrong model is entered, then software outputs a message "Wrong model entered" and prints the list of all models available.

4.7 Update-Database

4.7.1 The owner would be able to update the number of cars in the fleet. If he buys or sells a car, the same can be reflected in the database.
4.7.2 There would be an option to update databases in the options. Choose that option, enter model, choose to buy or sell cars to fleet, enter the number of cars.
4.7.3
REQ1: If the wrong model is entered, then software outputs a message "Wrong model entered" and prints the list of all models available.

REQ2: The number of cars that must be bought or sold is an integer number.

4.8 Record data

4.8.1 The software records to date and time, mile reading, and other valuable metrics when a car is rented out.
4.8.2 When a customer makes a booking, the system will record necessary data. The owner has to do nothing.
4.8.3
REQ1: The Recorded data will be stored in another excel sheet indexed with a token number.

4.9 Calculate stats

4.9.1 The software calculates average demand, maintenance charges, money earned, and other valuable stats for each car model.
4.9.2 There would be an option to calc stats in the options. Choose that option, enter a model.
4.9.3
REQ1: If the wrong model is entered, then software outputs a message "Wrong model entered" and prints the list of all models available.
REQ2: Previous data and some dummy data which satisfies the industry standards are entered first to get fair stats.

4.10 Set Price

4.10.1 The software helps in calculating a profitable and competitive price for each model of car.
4.10.2 There would be an option to set the price in the options. Choose that option, enter model, the sys will show required stats for deciding price, enter the price to set.
4.10.3
REQ1: If the wrong model is entered, then software outputs a message "Wrong model entered" and prints the list of all models available.
REQ2: The sys will only display all the required stats for making this choice but will not choose us.

4.11 Decide Fate of Car

4.11.1 The software helps decide which model of car to buy and which to sell off so that the business can make a consistent amount of profit.
4.11.2 There would be an option to decide the fate of a model in the options. Choose that option, enter model, the sys will tell him if the model is worth investing in or should he reduce the number of cars of that model in his fleet. An optimum answer is displayed.
4.11.3
REQ1: If the wrong model is entered, then software outputs a message "Wrong model entered" and prints the list of all models available.

REQ2: The system gives a ballpark number which is the number optimum of cars of that model when owned will give max profit.

5. Other Non-functional Requirements

5.1 Performance Requirements

Requirements such as RAM, CPU speed, etc., will help the software run smoothly. This helps the load time for user interface screens to be minimum and for also quick responses from the system

5.2 Safety Requirements

If there is extensive damage to the company database, such as file corruption or disk damage. It is recommended to backup all this data to an internet cloud storage to avoid these problems So we can restore the data if any of it happens

5.3 Security Requirements

The software doesn't communicate with any other device for data transfer. So, the data is secure from the software. But the data can be directly accessed and manipulated. For which the software is not liable. The owner has a unique password using which he can log in to the system and access its features. Anyone can access the features of the customer.

5.4 Software Quality Attributes

· Usability: The company should have more cars which have more demand
· Maintainability: The company should keep their vehicles, and customers should return the vehicles in good condition
· Reliability: They should always keep the database updated and correct.

5.5 Business Rules

If the system crashes, then the owner can easily re-run the application and start using it again. If the system shows the wrong amount of payment to the customer while returning the car, then it is the owner's responsibility to check if the database has not been tampered with. If the system shows that a model is available, but in reality, it is unavailable, then it is the owner's responsibility to rectify the problem.

6. Other Requirements

Appendix A: Glossary

None

Appendix B: Analysis Models

None

Appendix C: To Be Determined List

None