
Software Requirements Specification

for

Cars365

Prepared by Team 7

BE CSE G1 PSGTECH

19-08-2021

Table of Contents

Table of Contents	1
Revision History	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	3
2.1 Product Perspective	3
2.2 Product Functions	4
2.3 User Classes and Characteristics	4
2.4 Operating Environment	5
2.5 Design and Implementation Constraints	5
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	6
3.3 Software Interfaces	6
3.4 Communications Interfaces	7
4. System Features	7
4.1 Description and priority	7
4.2 Stimulus/response sequences	7
4.3 Functional requirements	7
5. Other Nonfunctional Requirements	7
5.1 Performance Requirements	7
5.2 Safety Requirements	8
5.3 Security Requirements	8
5.4 Software Quality Attributes	8
5.5 Business Rules	8
6. Other Requirements	8
Appendix A: To be determined list	9
Appendix B: Analysis Models	9

Revision History

Name	Date	Reason For Changes	Version

--	--	--	--

1. Introduction

1.1 Purpose

The purpose of this project is to create an online platform (Cars365) for buying, selling and renting used cars.

1.2 Document Conventions

This document uses the following conventions.

- DB Database
- ER Entity Relationship
- ML Machine learning
- DV Data visualizations

1.3 Intended Audience and Reading Suggestions

This project is a prototype for a car trading online platform and it is temporarily restricted within the college premises. This has been implemented under the guidance of college professors. This project is useful for virtual and authenticated communication between buyers and sellers.

1.4 Product Scope

The purpose of the cars365 online platform is to ease car trading and to create a convenient and easy-to-use application for car buyers and sellers. The system is based on a database with its buy, sell, rental and price prediction functions. We will have a database server supporting hundreds of major cities around India as well as hundreds of cars by various automobile manufacturers. Above all, we hope to provide a comfortable user experience along with convincing price predictions available.

1.5 References

- <https://krazytech.com/projects>
- Fundamentals of database systems by ramez elmarsi and shamkant b.navathe.

2. Overall Description

2.1 Product Perspective

The Cars365 online platform stores the following information.

- **Buyer details :**

It includes the buyer's details such as their name, email Id, phone number, address, and their preferred car attributes like car type, seating capacity, fuel type, and a lot more.

- **Seller details:**

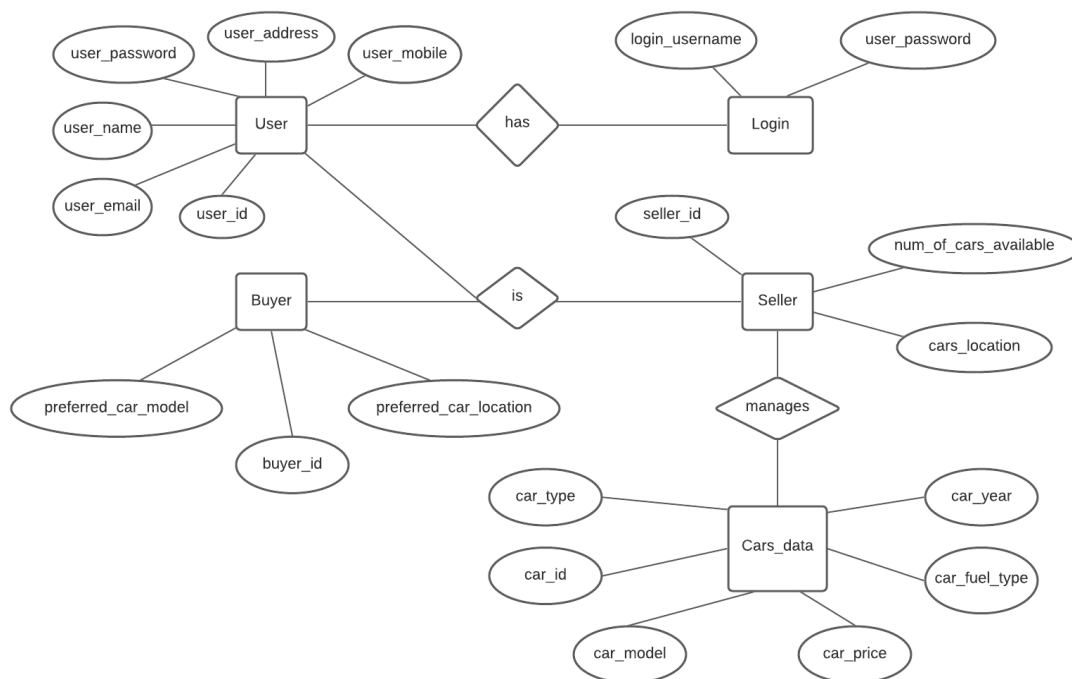
It includes the seller's details such as their name, email Id, phone number, address, and their owned car attributes like car type, car model, fuel type, and a lot more.

- **General Car data:**

It includes past to present information about car trading, trends and the statistics related to cars.

2.2 Product Functions

The major features of the Cars365 online platform is shown in below entity-relationship model (ER model)



2.3 User Classes and Characteristics

Users can register / login to the Cars365 portal by providing details like email ID, username, password, etc. Users of the system should be able to retrieve car information from the available car sellers with their owned car attributes from the database.

- **Sellers Portal:**
This portal provides an interface for the sellers who can upload their cars' photos and information so that their cars can be posted on the buyers page.
- **Buyers Portal:**
This portal provides an interface for buyers who are searching used cars to buy and through this, they can connect to the sellers using the provided contact information.
- **Rental Portal:**
This page is for the users who wish to rent cars for a certain period of time.
- **Car price predictor:**
The car price predictor is a ML model which predicts the prices of the used cars based on the data provided by the sellers.
- **Comparison between cars:**
This page is handy for buyers who tend to compare multiple cars for selecting the best one out.
- **Popular cars dashboard and visualizations:**
This dashboard provides useful charts and tables of car sales and trends of buyers and sellers.
- **Vintage cars auctioning:**
Auctioning page is for the vintage car sellers who can post their cars and conduct real time auctions.

2.4 Operating Environment

Operating environment for the Cars365 online platform is as listed below.

- **Operating system:** any operating system which supports web browsers.
- **Client/Server system.**
- **Database:** Google Firebase
- **Platform:** Html, CSS, JavaScript, Python.

2.5 Design and Implementation Constraints

1. The global schema.
2. SQL commands for above queries/applications.
3. Implement the database using a centralised database management system
4. Implement the system using front-end languages (Html/CSS) and back-end language (JavaScript).
5. This software is responsible for providing seamless buyer/seller communication through the portal and cannot be used for making transactions of any kind.

2.6 User Documentation

- <https://github.com/sanjai22m/Cars365>

2.7 Assumptions and Dependencies

Let us assume that Cars365 online platform is used in the following application:

- A request for storing sellable cars and owners information and receiving predicted prices.
- A request for viewing the available list of cars and their prices in a location.
- A request for collecting owner's information after successful car selection.

Assuming that the user details and number of cars available for sale in a location are unknown, a database with dynamic storage size is created.

3. External Interface Requirements

3.1 User Interfaces

- Front End Software: Html, CSS, JavaScript
- Back End Software: JavaScript, Google Firebase

3.2 Hardware Interfaces

- Windows
- Browser which supports Html, JavaScript.

3.3 Software Interfaces

Software used in the Cars365 online platform are listed below

Operating System:

We have chosen the Windows operating system because of its compatibility, ease of use and reliability.

Database:

Google Firebase is being used to store all the necessary details of buyers and sellers and also cars data.

JavaScript:

To implement our project we have chosen JavaScript as the back-end programming language since it is secure, easy to write and most popularly used for web development.

3.4 Communications Interfaces

This project will be supported in all types of web browsers supporting HTML, CSS and JS since we are deploying a website and using simple electronic forms to store and retrieve information.

4. System Features

4.1 Description and Priority

The Cars365 online platform maintains and views information of buyers and sellers details, number of available cars for sale, locations and prices. So this project has a high priority because it is difficult for sellers to find buyers and buyers to buy cars remotely without a trustable interface and ease of access.

4.2 Stimulus/Response sequences

- Sellers store car details in the portal.
- Sellers get to know the predicted prices of their cars.
- Buyers search for cars in a location through the portal.
- The platform displays the list of all available cars in that particular location.
- Buyers select the cars and get to know the price of the car and details of the owners/sellers.

4.3 Functional requirements

- **Client/Server system:**
The term client/server refers primarily to an architecture or logical division of responsibilities, the client is the application (also known as the front-end), and the server is the DBMS (also known as the back-end).

A client/server system is a distributed system in which,

- Some sites are client sites and others are server sites.
- All the data resides at the server sites.
- All applications execute at the client sites.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- E-R Model
E-R model is used to describe the structure of database in the form of a diagram
- Normalization
Normalization is a database design technique that reduces data redundancy and

eliminates undesirable characteristics like Insertion, Update and Deletion Anomalies. Also Normalization rules divides larger tables into smaller tables and links them using relationships

5.2 Safety Requirements

If there is extensive damage to a wide portion of the database due to unprecedented crashes or failure, the recovery method restores a past copy of the database that was backed up by a local database which periodically stores an instance of the current database.

5.3 Security Requirements

There are no specific security requirements for this project but a secure and reliable database should be chosen for data storage.

5.4 Software Quality Attributes

Correctness :

The seller and car details should be legit since many buyers would be making contacts and enquiries.

Availability:

This car should be available in the listed location with the recorded attributes.

Maintainability :

The owners and sellers should maintain a correct availability list of cars with accurate attributes and locations.

Usability :

The car information should suffice the preferences and requirements of the buyers.

5.5 Business Rules

The buyers registered in the portal should be transparent and readily available to make deals with the contacting buyers and to transport the cars to their locations. At the same time, the buyers must cooperate with the sellers in terms of reachability and transactions to ensure healthy car trading.

6. Other Requirements

For documentation and modelling diagrams for this project, the tools used are:

- **Documentation:** MS Word and Google Docs
- **E-R Model design:** Lucidchart

Appendix A: To be determined list

The size of the database is to be determined based on the number of users (both buyers and sellers) and the cars data. The extension of features will be possible only after knowing the reach and responses of the active users.

Appendix B: Analysis Models

The entity-relationship model for the product functions have been shown in the above section (2.2 Product functions).