Cairo University
Faculty of Computers and Artificial Intelligent

CS251 - Software Engineering I

Garage System

Software Requirements Specifications (SRS)

Mariam Osama Ahmed

Fouad Sayed Fouad

Mohamed Sayed Ali

5 & 2022

CS251: Phase 1 – Team Name

Software Requirements Specifications

Contents

Team

CS251: Phase 1 – Team Name

Software Requirements Specifications

ID	Name	Email	Mobile
20200519	Mariam Osama Ahmed	mo086388@gmail.com	01110981070
20200384	Fouad Sayed Fouad	foadsayed18@gmail.com	01210950773
20200450	Mohamed Sayed Ali	mohamedsayed415.ms@gmail.com	01020467154

Document Purpose and Audience

- This document tells us the requirements (functional/ nonfunctional), use case model, every use case description in this program.
- Audience → Customers who want to build a garage system.

Introduction

Software Purpose

- Build a strong infrastructure to the garage system.

Software Scope

- Owner or receptionist must log-in to the system to interact with the system.
- Each vehicle must be identified by model year, model name, unique identifier number.
- Each vehicle must have a depth and width.
- System should display available slot if owner or receptionist want to show them.
- System will calculate total time automatically by subtracting arrival time from departure time.
- System will calculate fees by multiplying number of hours *5.
- Owner must input number of slots at the beginning.

Software Requirements Specifications

Requirements

Functional Requirements

1. Log in
2. Park in
3. Park out
4. Identification
5. Display available slots
6. Calculate total time
7. Calculate total income
8. Calculate fees
9. Slot dimensions
10. Vehicle dimension
11. Check out

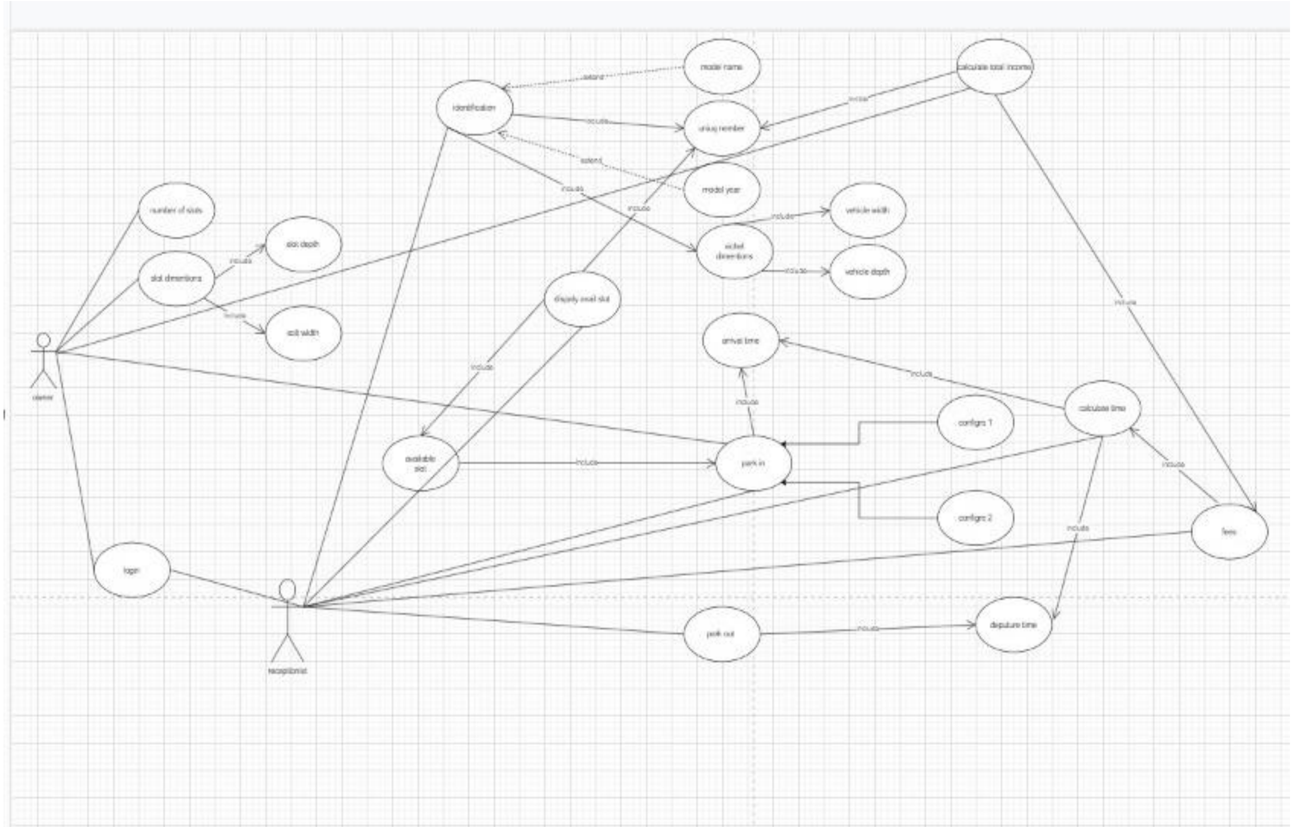
Nonfunctional Requirements

- usability:
  - Customers should be able to access all relevant use cases with the fewest amount of mouse clicks and keystrokes possible.
- Performance:
  - At least 100 customers and 1,000 reservations can be accommodated by the system. Over time, these figures should be multiplied tenfold or more.
  - The system should reduce database connection times and provide customers with a speedy and painless experience.
  - The system should always display the correct output to receptionist and customers.
- Reliability:
  - Through the use of persistent storage and regular backup, the system should not lose a reservation.
  - If the user interface disconnects from the system, the system should remember the details of the user's interaction.

System Models

Use Case Model

usecase diagram
.drawio



Use Case Tables

Use Case ID:	1	
Use Case Name:	Log-in	
Actors:	Owner, Receptionist	
Pre-conditions:	Any one of them want to enter the system	
Post-conditions:	Any one of them will be able to interact with the system	
Flow of events:	**User Action**	**System Action**
	1- Any one of them open the log-in page	
	2- enter password	
		3- System will check the password
	4- Any one of them will be able to interact with the system	
Exceptions:	**User Action**	**System Action**
	1- Owner or Receptionist Enter Password	
		2- Enter invalid password 3- System will send message to the owner or receptionist to enter the password again
Includes:	None	
Notes and Issues:		

Use Case ID:	2	
Use Case Name:	Identification	
Actors:	Receptionist	
Pre-conditions:	Customer wants to park his car	
Post-conditions:	Customer can park in the garage	
Flow of events:	**User Action**	**System Action**
	1- Customer tells the receptionist vehicle's information	
	2- Receptionist enter the data of the vehicle	
		3- System stored the data
		4- System check available slots
		5- System retrieve available slots to the receptionist
	6- Receptionist will choose a slot	
Exceptions:	**User Action**	**System Action**
	1- Receptionist enter the data of the vehicle	
		2- he entered an unlogic data (ex. Model year: x23z) 3- System will send message to the receptionist to enter data again
Includes:	1- Unique identifier number 2- Vehicle dimensions	
Notes and Issues:		

Use Case ID:	3	
Use Case Name:	Park-In	
Actors:	Owner, Receptionist	
Pre-conditions:	Vehicle must be identified in the system	
Post-conditions:	Vehicle has been parked in the garage	
Flow of events:	**User Action**	**System Action**
	1- Owner will select a one type of two configurations	
		2- System stored the configuration
	3- Receptionist will make vehicle parked based on this configuration	
		4- System will store the arrival time automatically
		5- System give unique identifier number to the vehicle
Exceptions:	**User Action**	**System Action**
	1- Owner enter configuration number	
		2- Enter invalid password 3- System will send message to the owner to enter the password again
Includes	Arrival time	
Notes and Issues:		

Use Case ID:	4	
Use Case Name:	Calc time	
Actors:	Receptionist	
Pre-conditions:	Receptionist entered arrival and departure time	
Post-conditions:	System will retrieve a total time	
Flow of events:	**User Action**	**System Action**
		1- System will calculate a total time from subtracting departure time minus arrival time
Exceptions:	**User Action**	**System Action**
Includes:	1- Arrival time 2- Departure time	
Notes and Issues:		

Use Case ID:	5	
Use Case Name:	Fees	
Actors:	Receptionist	
Pre-conditions:	Vehicle must be parked out	
Post-conditions:	Customer can leave the garage by his vehicle	
Flow of events:	**User Action**	**System Action**
		1- System will calculate the time which vehicle has been parked at the garage
		2- System will calculate the total fees and retrieve it
	3- Receptionist ask guest to pay a fees and take it from him	
Includes:	Calc time	
Notes and Issues:		

Use Case ID:	6	
Use Case Name:	Park-Out	
Actors:	Receptionist	
Pre-conditions:	Vehicle must be parked in the garage	
Post-conditions:	Guest must go to pay a fees	
Flow of events:	**User Action**	**System Action**
	1- Receptionist check the unique identifier number	
		2- System will retrieve a vehicle’s place
		3- System will store the departure time automatically
Exceptions:	**User Action**	**System Action**
Includes:	Departure time	
Notes and Issues:		

Use Case ID:	7	
Use Case Name:	Total income	
Actors:	Owner	
Pre-conditions:	Owner log-in to the system	
Post-conditions:	Owner can know total income	
Flow of events:	**User Action**	**System Action**
	1- owner open the log-in page	
	2- enter password	
		3- System will check the password
	4- owner choose option total income	
		5- system retrieve total income
Exceptions:	**User Action**	**System Action**
	1- Owner Enter Password	
		2- Enter invalid password 3- System will send message to the owner to enter the password again
Includes:	1- fees 2- unique identifier number	
Notes and Issues:		

Ownership Report

Item	Owners
Document Purpose and audience/ Use case model/ Use case tables	*Mariam Osama Ahmed*
Software scope/ Use case tables	*Fouad Sayed Fouad*
Software purpose/ Functional & Nonfunctional requirements	*Mohamed Sayed Ali*