Cairo University
Faculty of Computers and Artificial Intelligent

CS251 - Software Engineering I

"Garagak"

Software Requirements Specifications (SRS)

By: The Straw Hats

1/6/2022

1st of June, 2022

CS251: Phase 2 – The Straw Hats

Software Requirements Specifications

Contents

CS251: Phase 2 – The Straw Hats

Software Requirements Specifications

Team

ID	Name	Email	Mobile
20200372	Fares Saad Abo ElSoud	faresaad74@gmail.com	01141496253
20200512	Marwan Mohamed Nabil Mohamed	marwan.m.nabil.03@gmail.com	01004039278
20200112	Bassem Mohamed Hassan Ismail	bassem8mohamed8@gamil.com	01203266637
20200638	Yahia Salah Zaki Mohamed	yahiaelsherif2002@gmail.com	01069559925

Document Purpose and Audience

-This document is a full and detailed description of a software system developed to ease a certain issue. Utilizing the best quality approaches, this document serves as a conceptual design specification and full overview of the preamble, all to bring out the fundamental functions of this project.

-Of course, this project software is so simple yet so beneficial. So, this document is designed so that it is easy for common readers such as CEOs and ordinary application users to understand.

Introduction

Software Purpose

-This software is developed to ease the tedious problem of finding a parking a slot in huge garages. It is designed to automatically assign a slot and unique ID to the user's vehicle (depending on its dimensions), this of course saves time and even the process of charging the user with the demanded fee.

Software Scope

- "Garagak" is designed specifically, as mentioned before, to ease the parking operation in large garages, and it does so by using many handy functions.

-It records the important details in this operation that is useful in the few processes of parking, such as the arrival and departure time, car dimensions, all while assigning a unique identification number to each vehicle to distinguish each one from another so that no issues could occur during checkout process.

-It should make the checkout process much efficient and secure, as everything is recorded from the moment the user is assigned a slot to the moment, he is at the gate checking out.

-Due to these factors, it should facilitate this process and make it easier for both the driver and the garage.

Software Requirements Specifications

Definitions, acronyms, and abbreviations

UCA	Use case for admin
UCU	Use case for user

Requirements

Functional Requirements

1. User

1.1. User should open the application and be able to initiate a “park-in” option in order for him to find the most suitable spot to park.
1.2. User can choose from 2 configurations that help find the perfect spot according to the dimensions of his car and the availability of slots; “first come first serve” which chooses the first free slot available from the parking garage slots, and “best-fit” which chooses the slot with minimum dimensions that suit the vehicle. When one is chosen and the slot is assigned and he is given a unique ID by the system, the arrival time is recorded.
1.3. When the User is leaving the slot, he/she should open up the app in order for him to see the amount he will be paying, so the system could recognize that the spot is free now.
1.4. The User should pay the amount demanded to the cashier.

2. Admin:

2.1. The admin is responsible for most of the functionality of the application.
2.2. The admin should authenticate the user’s info.
2.3. He is able to alter each slot’s maximum dimensions.
2.4. He should display the total amount of vehicles parked when the user opens the app.
2.5. He is able to set the maximum number of vehicles allowed in the garage at any time.
2.6. He has the ability to alter the hourly rate of parking.
2.7. He can set the maximum number of slots that can be used simultaneously.
2.8. He is always recording the total income of the garage.

Software Requirements Specifications

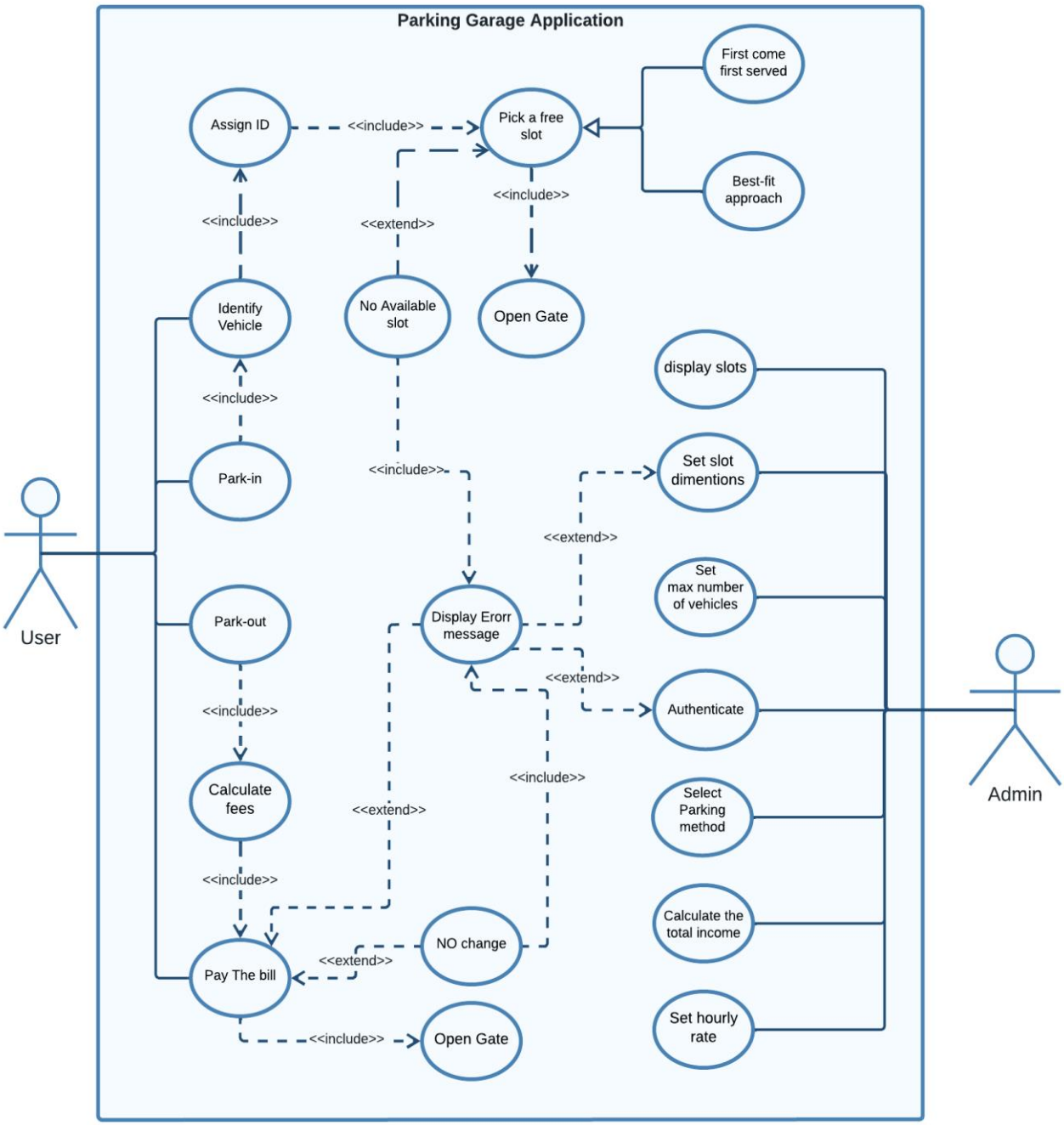
Non Functional Requirements

Attribute	Details
Usability	**-The system should be designed so that the user is quick to get around the application easily, and the functions are descriptive so the user can efficiently execute any task easily in his first encounters with the application so that errors that the user can commit are limited and possibly avoided.**
Maintainability	**-the application shouldn't contain many functions nor should it handle many variables simultaneously, so that it is easy to repair it if there are malfunctions.**
Correctness	**-the functionality of the system is simple; so it should be easy to meet all the functional requirements the application demanded.**
Testability	**-It should be easy to test it since –as we mentioned before- the application is simple and its functionalities are simple too.**
Documentation	**-As this document is made to help the users find their way around the application.**
Security	**-The system should be secure enough so that the interaction between both sides (system and user) doesn't have a third party involved, just these two.**

Software Requirements Specifications

System Models

Use Case Model



Software Requirements Specifications

Use Case Tables

Use Case ID:	UCA _function_1	
Use Case Name:	Set slot dimension	
Actors:	Admin	
Pre-conditions:	-Admin logs in and is authenticated successfully.	
Post-conditions:	-Slot's maximum and minimum dimension for each slot is updated or left as it is.	
Flow of events:	**User Action**	**System Action**
	1- user clicks on "Set slot dimension"	
		2- System asks admin to enter the slot information.
	3- User enters the number and the dimensions of slots.	
		4- System confirms admin and displays the slots.
Exceptions:	**User Action**	**System Action**
	1- User enters invalid number or invalid dimensions.	
		2- System refuse to add this Information. 3- System Display error message.
Includes:	-	
Notes and Issues:	-	

Software Requirements Specifications

Use Case ID:	UCA_function_2	
Use Case Name:	Set max number of vehicles	
Actors:	Admin	
Pre-conditions:	Admin logs in and is authenticated successfully	
Post-conditions:	-Maximum number of vehicles is updated or left as it is.	
Flow of events:	**User Action**	**System Action**
	1- user clicks on Set max num.	
		2- System asks admin to enter the number of vehicles.
	3- User enters the number of vehicles.	
		4- System confirms admin and displays the slots.
Exceptions:	**User Action**	**System Action**
	1. User enters max number of vehicles more than the number of slots.	-
		2. System displays an error message “this number is greater than the maximum slots”
Includes:	-	

Software Requirements Specifications

Use Case ID:	UCA_function_3	
Use Case Name:	Choose parking method	
Actors:	Admin	
Pre-conditions:	Admin logs in and is authenticated successfully.	
Post-conditions:	-a parking method is chosen and user is assigned a slot and garage status is updated accordingly.	
Flow of events:	**User Action**	**System Action**
	1- User clicks on choose parking method.	
		2- System displays parking options, which are: 1-First come first serve. 2-Best-fit
	3- User selects the desired method.	
		4- System performs the selected method and continues with the “park-in” option.
Exceptions:	**User Action**	**System Action**
	-	-
Includes:	-	
Notes and Issues:	-	

Software Requirements Specifications

Use Case ID:	UCA_function_4	
Use Case Name:	Calculate total income	
Actors:	Admin	
Pre-conditions:	Admin logs in and is authenticated successfully.	
Post-conditions:	-Total income is updated.	
Flow of events:	**User Action**	**System Action**
	1- User clicks on calculate total income.	
		2-System retrieves the total income
		3.System Display the total income
Exceptions:	**User Action**	**System Action**
	-	-
Includes:	-	
Notes and Issues:	-	

Use Case ID:	UCU_function_1
Use Case Name:	Park out

Software Requirements Specifications

Actors:	User, admin	
Pre-conditions:	User parked successfully	
Post-conditions:	-Number of hours added successfully. -The gate opens. -Total income is updated.	
Flow of events:	**User Action**	**System Action**
	1-User clicks on user options	
		2- System prompts the user to choose one of the following: 1.Park in 2.Park out
	3- User clicks on Park out function.	
		4. The departure time is recorded 5. System calculates the total time (departure time – arrival time). 6. Total fee is calculated (total time x hourly rate). 7. The demanded amount is displayed to the user and demanded from him.
	8- User inserts the money.	
		9. System confirms the operation. 10. The gate is opened.
Exceptions:	**User Action**	**System Action**
	1- User enters the amount of money less than the demanded.	

		2-the cash is less than the total fees. 3. System doesn't open the gate. 4. User is prompted again.
	OR 1. User enters an amount that is more than the demanded.	
		2. There is no sufficient change 3. The system prompts the user "please enter the exact amount due to an insufficient amount of change.
Includes:	Calculate total fee , Open gate	
Notes and Issues:	-	

Use Case ID:	UCU_function_2
Use Case Name:	Park in
Actors:	user

Software Requirements Specifications

<table>
<tr><td>Pre-conditions:</td><td colspan="2">User is successfully logged in.</td></tr>
<tr><td>Post-conditions:</td><td colspan="2">-The system assign slot for the user.
-The gate opens.
-Garage status is updated.</td></tr>
<tr><td rowspan="9">Flow of events:</td><td>User Action</td><td>System Action</td></tr>
<tr><td>1-User clicks on user options</td><td></td></tr>
<tr><td></td><td>2-System prompts the user to choose one of the following:

1. Park in

2. Park out</td></tr>
<tr><td>3- User clicks on park in.</td><td></td></tr>
<tr><td></td><td>4-System asks the use for the car dimension.</td></tr>
<tr><td>5-user enters the car dimensions and car info.</td><td></td></tr>
<tr><td></td><td>5-System assigns a unique ID for the vehicle.

6-After the user chooses one of the 2 configurations (best-fit or first come first serve), the System assigns the result of the chosen method and assigns it to this vehicle.

7-The gate is opened.

8.Garage status is</td></tr>
</table>

		updated and arrival time is recorded.
	7-User enters the garage and park successfully.	
Exceptions:	**User Action**	**System Action**
	1-User enters the dimensions and the info.	
		2-There is no available slots for this particular vehicle. 3- System Displays error message "Your vehicle doesn't fit in any slot." 4-gate is still closed.
	OR 1. User enters the dimensions and info.	
		2-The garage is full and there aren't any slots. 3-System displays error message "Sorry the garage is full!"
Includes:	Identify vehicle, Open gate, Assign ID, Display error message.	
Notes and Issues:	-	

Use Case ID:	UCA_function_5
Use Case Name:	Display slots
Actors:	Admin

Pre-conditions:	Admin logs in and is authenticated successfully.	
Post-conditions:	-The GUI displays the slots available to the user and the user can initiate park-in.	
Flow of events:	**User Action**	**System Action**
	1- User clicks on “display admin options”.	
		2- System lists the following: 1.displays slots 2.set slot dimensions 3.authenticate 4.set max number of vehicles 5.select parking method 6.calculate total income 7.set hourly rate
	3-User selects “display slots”.	
		4-System displays the available slots to the user.
Exceptions:	**User Action**	**System Action**
	-	-
Includes:	-	
Notes and Issues:	-	

Use Case ID:	UCA_function_6
Use Case Name:	Authenticate
Actors:	Admin

Pre-conditions:	-User opens the application and enters his info.	
Post-conditions:	-User is now able to update the Garage information if they want.	
Flow of events:	**User Action**	**System Action**
		1-System prompts the user to choose one of The following : 1. Admin options. 2. User options.
	2-User clicks on "admin options"	
		3-System checks info to validate if the user is an admin 4-the user is validated and displayed the list of options he can perform.
Exceptions:	**User Action**	**System Action**
	1- User enters invalid info.	
		2- System Display error message "not an admin or invalid info".

Use Case ID:	UCA_function_7
Use Case Name:	Set slot dimension
Actors:	Admin
Pre-conditions:	Admin logs in and is authenticated successfully.

Post-conditions:	-user is able to update any of the slots' dimensions. If he desires.	
Flow of events:	**User Action**	**System Action**
	1- user clicks on Set slot dimension	
		2- System prompts the user to select which slot to update.
	3- User selects a slot	
		4- System prompts the user for the new dimensions of the selected slot.
	5-User enters the new max and min dimensions of the slot.	
		6-system confirms the numbers and updates the slots.
Exceptions:	**User Action**	**System Action**
	1- User enters invalid numbers or invalid dimensions.	
		2-System displays an error message "Invalid numbers" and prompts him again.

Use Case ID:	UCA_function_8
Use Case Name:	Set hourly rate
Actors:	Admin
Pre-conditions:	Admin logs in and is authenticated successfully.
Post-conditions:	-User can set new hourly rate

Software Requirements Specifications

Flow of events:	**User Action**	**System Action**
		1-System prompts the user to choose one of The following : 1-Admin options. 2- User options.
	2-User clicks on “admin options”	
		3-System prompts the user to choose one of the following: 1-set max slots 2-select parking method 3-calculate total income 4-set hourly rate 5-set slot dimensions 6-set max vehicles 7-display slots
	4-User selects “set hourly rate”	
		5-system prompts user for the new fee.
	6-User enters a valid number	
		7-system checks the number and validates it
		8-Hourly fee is updated
Exceptions:	**User Action**	**System Action**
	1-User enters 0 or negative number	
		2-System displays error message “invalid number please try again” and prompts him again.

Software Requirements Specifications

Includes:	-
Notes and Issues:	-

CS251: Phase 2 – The Straw Hats

Software Requirements Specifications

Ownership Report

Item	Owners
-Use case Descriptions (all)	Marwan Mohamed Fares saad Yahia salah
-Source Code	Fares saad Marwan Mohamed
-Use case diagram	Fares saad Marwan Mohamed Bassem Mohamed
-Class Diagram	Marwan Mohamed Fares Saad
-Purpose and introduction of document	Marwan Mohamed
-Functional and Non-functional requirements	Marwan Mohamed
-Sequence Diagram	Fares saad Marwan Mohamed
Class - Sequence Usage Table	Fares saad Marwan Mohamed Bassem Ismail Yahia Salah

Policy Regarding Plagiarism:

1. تشجع الكلية على مناقشة الأفكار و تبادل المعلومات و مناقشات الطلاب حيث يعتبر هذا جوهريا لعملية تعليمية سليمة
2. ساعد زملاءك على قدر ما تستطيع و حل لهم مشاكلهم فى الكود و لكن تبادل الحلول غير مقبول و يعتبر غشا.
3. أى حل يتشابه مع أى حل آخر بدرجة تقطع بأنهما منقولان من نفس المصدر سيعتبر أن صاحبيهما قد قاما بالغش.
4. قد توجد على النت برامج مشابهة لما نكتبه هنا أى نسخ من على النت يعتبر غشا يحاسب عليه صاحبه.
5. إذا لم تكن متأكدا أن فعلا ما يعد غشا فلتسأل المعيد أو أستاذ المادة.
6. فى حالة ثبوت الغش سيأخذ الطالب سالب درجة المسألة ، و فى حالة تكرار الغش سيرسب الطالب فى المقرر.