Cairo University  
Faculty of Computers and Artificial Intelligent

**CS251 - Software Engineering I**

Project Name

Software Requirements Specifications (SRS)

Team Names

Month & Year

Contents

[Instructions [To be removed] 3](#_Toc101814799)

[Team 3](#_Toc101814800)

[Document Purpose and Audience 3](#_Toc101814801)

[Introduction 3](#_Toc101814802)

[Software Purpose 3](#_Toc101814803)

[Software Scope 3](#_Toc101814804)

[Definitions, acronyms, and abbreviations 3](#_Toc101814805)

[Requirements 4](#_Toc101814806)

[Functional Requirements 4](#_Toc101814807)

[Non Functional Requirements 4](#_Toc101814808)

[System Models 4](#_Toc101814809)

[Use Case Model 4](#_Toc101814810)

[Use Case Tables 5](#_Toc101814811)

[Ownership Report 6](#_Toc101814812)

[Policy Regarding Plagiarism: 6](#_Toc101814813)

# Instructions [To be removed]

* **IMPORTANT. Rename this document according to the naming style stated in the project description.**
* **Remove the following notes and any red notes.**
* **This document is the template document for your SRS.**
* **For further guidelines and information, READ project details document (C251-Project Description-ParkingGarage-v1.0).**

# Team

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Email** | **Mobile** |
|  | 1st name is team leader |  |  |
|  |  |  |  |
|  |  |  |  |

# Document Purpose and Audience

**What is this document?**

This document shows the purpose and the scope of the software, explained the requirements (functional and non-functional) of the software, and illustrates the system models using the use case diagram and the use case tables.

**Who is excepted to read it?**

The project Manager and the customer.

# Introduction

## Software Purpose

It helps the users to reduce the time spent in the parking zone. Its purpose is to automate and maximize the use of parking zones, as well as to meet all of the customers' requirements.

## Software Scope

* **Any software could have too many components / Major features .. but we should implement specific things...this is the scope**
* **In simple points, what is the software scope (focus on components / Major features, not tiny things)**

## Definitions, acronyms, and abbreviations

* **In a table, list all needed ones. Consider the audience**
* **Think as following: Document has abbreviation ATM..IFF audience doesn’t know it, let’s clarify it.**

# Requirements

## Functional Requirements

* The user must be able to provide personal and vehicle information.
* The system should capture automatically the arrival time ofa vehicle if there is an available slot.
* During the park-in function the system shall pick a free slot based on the active slot configuration.
* The system should capture automatically the departure time ofa vehicle from the garage.
* The system shouldcalculate the total income as well as the total number of vehicles that used the parking garage at any given point in time.
* The system should display the available parking slots.
* The system should handle the exceptions that can happen during user interaction and through any other calculations and displays an error message.

## Non Functional Requirements

|  |  |
| --- | --- |
|  | **Details** |
| **Performance** | * During the park-out calculating the parking-fees should be done within 20 second |
| **Usability** | * The system is designed to be user-friendly and simple to use. * User should be notified if there is no available slot |
| **Reliability** | * All orders should be stored in the database without any errors. |
| **Security** | * Users cannot access the database, do not read and write the information. |
| **Scalability** | * System should be able to support up to 1000 Cars. |

# 

# System Models

## Use Case Model

## Use Case Tables

* **Flow of events should be very detailed**

|  |  |  |
| --- | --- | --- |
| Use Case ID: |  | |
| Use Case Name: |  | |
| Actors: |  | |
| Pre-conditions: |  | |
| Post-conditions: |  | |
| Flow of events: | **User Action** | **System Action** |
| 1- User Enter Card and Password. |  |
|  | 2- System Verify user data |
| 3- User Select Vodafone from the list |  |
|  | 4- System retrieves Vodafone bills |
| and so on |  |
| Exceptions: | **User Action** | **System Action** |
| 1- User Enter Card and Password. |  |
|  | 2- Card is invalid and unreadable.  3- System rejects cars. |
| Includes: |  | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: |  | |
| Use Case Name: | Park-out | |
| Actors: | Driver | |
| Pre-conditions: | The driver asks to get out of the garage | |
| Post-conditions: | The driver gets out of the garage | |
| Flow of events: | **User Action** | **System Action** |
| 1- The driver asks to get out of the garage |  |
|  | 2- System calculates the parking fees |
| 3- The driver pays the parking fees |  |
|  | 4- System opens the exit door |
| 5- The driver gets out of the garage |  |
| Exceptions: | **User Action** | **System Action** |
| 1- User Enter Card and Password. |  |
|  | 2- Card is invalid and unreadable.  3- System rejects cars. |
| Includes: | Pay fees | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: |  | |
| Use Case Name: | Pay fees | |
| Actors: | Driver, Bank | |
| Pre-conditions: | The driver asks to pay parking fees | |
| Post-conditions: | The driver gets out of the garage | |
| Flow of events: | **User Action** | **System Action** |
| 1- The driver asks to pay parking fees |  |
|  | 2- System shows the parking fees and payment methods (cash – credit card) |
| 3- The driver Choose the payment method and pay them |  |
|  | 4- system confirms the success of the payment process |
|  |  |
| Exceptions: | **User Action** | **System Action** |
| 1- The driver does not have enough money |  |
|  | ??? |
| Includes: |  | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: |  | |
| Use Case Name: | Park-in | |
| Actors: | Driver | |
| Pre-conditions: | The driver enters his vehicle’s info | |
| Post-conditions: | The driver gets out of the garage | |
| Flow of events: | **User Action** | **System Action** |
| 1- The driver enters his vehicle data |  |
|  | 2- The system selects a suitable free slot for the vehicle depends on his info |
| 3- The driver park his vehicle in garage |  |
|  | 4- system set time of entering automatic |
|  |  |
| Exceptions: | **User Action** | **System Action** |
| 1- The driver enters his vehicle data |  |
|  | 2- no slots available  3- no slots has enough size |
| Includes: | Enter vehicle’s info | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: |  | |
| Use Case Name: | vehicle’s info | |
| Actors: | Driver | |
| Pre-conditions: | The driver enters his vehicle’s info | |
| Post-conditions: | System save the data | |
| Flow of events: | **User Action** | **System Action** |
| 1- The driver enters his vehicle’s info |  |
|  | 2- The system save vehicle’s info |
|  |  |
|  |  |
|  |  |
| Exceptions: | **User Action** | **System Action** |
| 1- The driver enters wrong info |  |
|  | ?? |
| Includes: | Enter vehicle data | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: |  | |
| Use Case Name: | Pick free Slot | |
| Actors: | system | |
| Pre-conditions: | The driver enters his vehicle’s info | |
| Post-conditions: | System Pick free Slot | |
| Flow of events: | **User Action** | **System Action** |
| 1- The driver enters his vehicle’s info |  |
|  | 2- The system Pick free Slot depends on come first served slots and slot with the minimum dimension to hold the vehicle |
|  |  |
|  |  |
|  |  |
| Exceptions: | **User Action** | **System Action** |
|  | 1- not found suitable slot  2- not found available slot |
|  |  |
| Includes: | Enter vehicle data | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: |  | |
| Use Case Name: | Display Available parking slots | |
| Actors: | system | |
| Pre-conditions: |  | |
| Post-conditions: |  | |
| Flow of events: | **User Action** | **System Action** |
|  | 1-Display Available parking slots |
|  |  |
|  |  |
|  |  |
|  |  |
| Exceptions: | **User Action** | **System Action** |
|  | 1- not found slot available |
|  |  |
| Includes: |  | |
| Notes and Issues: |  | |

|  |  |  |
| --- | --- | --- |
| Use Case ID: |  | |
| Use Case Name: | Display error messages | |
| Actors: | system | |
| Pre-conditions: |  | |
| Post-conditions: |  | |
| Flow of events: | **User Action** | **System Action** |
| 1- when user enters wrong data |  |
|  | 2- system Display error messages |
|  |  |
|  |  |
|  |  |
| Exceptions: | **User Action** | **System Action** |
|  |  |
|  |  |
| Includes: |  | |
| Notes and Issues: |  | |

# Ownership Report

* **Remove the following notes and any red notes**
* **For every item in this document, write the owners. If someone is owner of something, s/he understands it 100%**
* **Team leader must verify the table with the team members.**

|  |  |
| --- | --- |
| **Item** | **Owners** |
|  |  |
|  |  |

# Policy Regarding Plagiarism:

**Students have collective ownership and responsibility of their project. Any violation of academic honesty will have severe consequences and punishment for ALL team members.**

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.