Cairo University
Faculty of Computers and Artificial Intelligent

CS251 - Software Engineering I

Project Name

Software Requirements Specifications (SRS)

Team Names

Month & Year

CS251: Phase 1 – Team Name

Software Requirements Specifications

Contents

CS251: Phase 1 – Team Name

Software Requirements Specifications

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

- **Any document anywhere should tell us 2 things: (1) what this document is and (2) who is excepted to read it.**
- **Write in simple notes: What is this document?**
- **List the target audience to read this document (e.g. CEO? Project Manager? Customer...?)**

Introduction

Software Purpose

- **Summarize the purpose of the software**

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.**

CS251: Phase 1 – Team Name

Software Requirements Specifications

Requirements

Functional Requirements

- **This is the most critical part... functional requirements describe what the system should do**
  - **E.g. an ATM allows you to enter Card, enter user name password and withraw a money**
- **List all the system requirements, respecting the problem statement giving by your TA**
  - **Make sure to go in the missing details for the mentioned features/components**
    - **Discuss with TA**
  - **Going beyond them (e.g. adding new complete major feature / component) is breaking the statement scope**
- **Each requirement should be clearly described, such that it can be understood without the presence of the one who wrote it.**
- **This part is the basis for writing the contract with client and estimating the size, time and cost of developing the software.**

Non Functional Requirements

- **Non-functional requirements describe how the system works**
  - **E.g. Withdraw operation will be done within 20 second. Network is using secured protocols. System allows up to 30,000 withdrawals per minute.**
  - **Think about the operation / system quality**
- **There are too many non functional requirements. Read in wiki. Pick the suitable ones for your system. Non-functional requirements must be VERIFIABLE, i.e., MEASURABLE.**
  - **Some Types as just examples: Usability, Reliability, Performance, Security, Scalability, Portability, Maintainability**
  - **Select the sutiable ones, for each one write the details**
  - **Be realistic ☺**

	Details
Performance	• **Withdraw operation will be done within 20 second**
Scalability	• **System should be able to support up to 1000 simultaneous game players.**

System Models

Use Case Model

- **Using UML, write the use case model expressing the system actors & operations**

Software Requirements Specifications

Use Case Tables

- **Using below table template, for each requirement write a use case table that shows user/system interaction**
  - **If one requirement is so big, you could divide it to more than table**
  - **If some requirements are not major, you could plugin them in other senario**
    - **E.g. you may not do LogIn Usecase table as it is simple functionality**
- **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:		

CS251: Phase 1 – Team Name

Software Requirements Specifications

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.