Cairo University  
Faculty of Computers and Artificial Intelligence

**كل التعليمات الحمراء تقرأ بعناية و تنفذ ثم تمسح و يكتب مكانها المطلوب**

**CS251**

**Intro. to Software Engineering**

Project Name

Software Requirements Specifications

Version 1.0

Team Names and Emails and Phone of Team Leader

Month & Year

Contents

[Instructions [To be removed] اقرأ هذا الجزء ثم احذفه 3](#_Toc128921813)

[Team 3](#_Toc128921814)

[Document Purpose and Audience 3](#_Toc128921815)

[Introduction 3](#_Toc128921816)

[Software Purpose 3](#_Toc128921817)

[Software Scope 3](#_Toc128921818)

[Definitions, acronyms, and abbreviations 4](#_Toc128921819)

[Requirements 4](#_Toc128921820)

[Functional Requirements 4](#_Toc128921821)

[Non Functional Requirements 4](#_Toc128921822)

[System Models 5](#_Toc128921823)

[Use Case Model 5](#_Toc128921824)

[Enriched User Stories 5](#_Toc128921825)

[System Navigation Map 7](#_Toc128921826)

[Tools 7](#_Toc128921827)

[Ownership Report 7](#_Toc128921828)

[Policy Regarding Plagiarism: [To be removed] اقرأ هذا الجزء ثم احذفه 8](#_Toc128921829)

# Instructions [To be removed] اقرأ هذا الجزء ثم احذفه

* **IMPORTANT. Rename this document to :**
  + **CS251-2023-YourGroup-TA-LeaderID- DraftToffeeSRSv0.0.docx for draft version and**
  + **CS251-2023-YourGroup-TA-LeaderID- FinalToffeeSRSv1.0.docs for final version**
* **Generate pdf version for submission in course page**
* **Remove the following notes and any red notes in this color. امسح كل التعليمات الحمراء بعد تنفيذها**
* **This document is the template document for your SRS.**
* **For further guidelines and information, READ project description.**
* **After finishing the document, update the table of contents by clicking right click and then update.**

# 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 this document is about.**
* **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... If audience doesn’t know it, let’s clarify it.**

# 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 professor**
  + **Make sure to go in the missing details for the mentioned features/components**
    - **Discuss with TA and with product owners**
  + **Limit yourself to the needs of the client and do not volunteer to expand the project 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.**
* **Requirements statements should be Measurable and Specific**

## Non Functional Requirements

* **Non-functional requirements describe how the system works (performance, quality, platform, etc.)**
  + **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**](http://en.wikipedia.org/wiki/Non-functional_requirement)**. 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 suitable 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**
* **Write a definition for each actor and what his role is.**

## Enriched User Stories

* **Using below table template, for each requirement write an enriched user story specifying the details of each use case and showing the interaction to implement this use case.** 
  + **If one requirement is so big, you could divide it to more than one user story.**
  + **If some requirements are not major, you could plugin them in other user stories.**
* **Flow of events should be very detailed**
* **User Story #1**

|  |  |
| --- | --- |
| **User Story ID** | US #1 |
| **User Story Name** |  |
| **Actors** |  |
| **Description** | **As** a …..  **I like** to be able to …..  **So** ………….. |
| **Per condition** |  |
| **Post condition** |  |
| **Acceptance Criteria** | **Describe when we can decide that this user story is correctly implemented and accepted. For example:**  **Given** I’m a logged-out system user and I’m on the Sign-In page  **When** I fill in the “Username” and “Password” fields with my authentication credentials and I click the Sign-In button  **Then** the system signs me in |

* **Scenarios**

**Normal Scenario**

|  |  |
| --- | --- |
| **Actor Action** | **System Response** |
| 1- User Enter Card and Password.  2- Click **Submit** |  |
|  | 3- System Verify user data  4- System displays list of Mobile companies |
| 5- User Select Vodafone from the list |  |
|  | 6- System retrieves Vodafone bills |
| 7- And so on |  |
|  |  |
|  |  |
|  |  |

**Exceptional Scenario**

|  |  |
| --- | --- |
| **Actor Action** | **System Response** |
| 1- User Enter Card and Password.  2- Click **Submit** |  |
|  | 3- Card is invalid  4- Systems rejects card and displays an error message |
|  |  |
|  |  |

* **Screen Design**

**Give a draft design of the screen(s) on which this user story will be implemented.**

**Do it as a as wireframe or a mockup. Use a tool to do that. Give each screen a number and name.**

* **Data Dictionary:**

| **Element Label** | **Type/Length** | **Data Validation / Business Rule** |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

* **User Story #2**

|  |  |
| --- | --- |
| **User Story ID** | US #2 |
| **User Story Name** |  |
| **Actors** |  |
| **Description** | **As** a …..  **I like** to be able to …..  **So** ………….. |
| **Per condition** |  |
| **Post condition** |  |
| **Acceptance Criteria** | **Describe when we can decide that this user story is correctly implemented and accepted. For example:**  **Given** I’m a logged-out system user and I’m on the Sign-In page  **When** I fill in the “Username” and “Password” fields with my authentication credentials and I click the Sign-In button  **Then** the system signs me in |

…………………

…………………

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

## System Navigation Map

* **Draw a navigation map that show how the screens are related (See example at** <https://stuff.mit.edu/afs/sipb/project/android/docs/training/design-navigation/wireframing.html>)

# Tools

* **Write a list of all tools used to develop the design (e.g., ArgoUML, Visual-Paradigm, mocqus, etc.)**

# Ownership Report

* **Remove the following notes and any red notes**
* **For every item in this document, write the owners.**
* **Team leader must verify the table with the team members.**

|  |  |
| --- | --- |
| **Student** | **Items he created** |
| **Mohammad Ali Sayed** | **Part of Use Case Model, Non-Functional Requirements, and User Stories #1 and #2.** |
|  |  |
|  |  |

# Policy Regarding Plagiarism: [To be removed] اقرأ هذا الجزء ثم احذفه

* **Remove this part and all red instructions**
* **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.**