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 :**
  + **CS251-2025-SectionNum-TA-ID1-ID2-ID3-ProjectName-SRSv1.0.docx**
* **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 fright click and then update.**

# Team

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Email** | **Mobile** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Document Purpose and Audience

Purpose:

The purpose of this paper is to make the requirements for software that have been decided upon by the stakeholders more understandable by outlining the features that the software must have in order to reduce development time and expense.

Audience:

* Software Development Team
* Stakeholders

# Introduction

## Software Purpose

## The purpose of the Budget Manager is to help users set and achieve financial goals while providing guidance for better financial planning and expense management.

## 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)**

Budget Manager implements a budget tracking system for individual users or even organizations, offering various functionalities for expense tracking, income management, setting financial goals, spending analysis, and budget planning. The app helps users monitor their financial activities, generate reports, and gain insights into their spending habits to make informed financial decisions.

## Definitions, acronyms, and abbreviations

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

|  |  |
| --- | --- |
| Phrase | Definition |
|  |  |
|  |  |

# 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**

|  |  |
| --- | --- |
| Requirement ID | Requirement State |
| FR01 | As soon as the user launches the application, a login/sign-up screen will appear. |
| FR02 | If the user chooses to log in, the application will provide a box for them to enter their username and password. |
| FR03 | If a user chooses to register, the app will provide a signup page where they can enter their personal information, including first name, last name, password, gender, country, and email address. And asking if the user would like to sync with their bank account. This registration process will enable personalized features and secure access to their financial data. |
| FR04 | The application shall have a panel for users to manage their budgets, consisting of:   1. **“**My Budget” tab where users can:    * Create a new budget by specifying income sources, expense categories, and budget limits.    * Edit or delete existing budgets.    * View a summary of their financial status, including total income, total expenses, and remaining budget. 2. **“**Transactions” tab where users can:    * Add new transactions by entering details such as category, amount, date, and payment method.    * Edit or delete recorded transactions.    * Filter transactions by date, category, or amount. 3. “Goals” tab where users can:  * View their financial goals * Set new financial goals by specifying target savings, timeframes, and desired spending limits. * Track progress toward their financial goals based on their recorded transactions. |
| FR05 | The application shall provide a Reports panel where users can:   1. View graphical representations (charts, graphs) of their spending and savings trends. 2. Generate summary reports of monthly or yearly income and expenses. 3. Export reports in PDF or Excel format for external use. |
| FR06 | The application shall provide notifications to help users stay on track with their financial goals:   1. Budget Alerts: Notify users when they are approaching or exceeding their budget. 2. Bill Reminders: Send reminders for upcoming bills and due payments. 3. Goal Progress Notifications: Notify users of their progress toward financial goals. |

## 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.** |

# 

|  |  |
| --- | --- |
| Measure | Details |
| Performance | * The Sign-in/sign-up screen should take about 3 seconds to load. * Every panel in the system should take less than 10 seconds to load. |
| Scalability | Up to 3000 users should be able to access the system at once without experiencing any performance issues. |
| Maintainability | For 72 hours, there should be an 80 percent chance that the system can be maintained. This implies that there is a 75% chance that a system component with a serious problem will be resolved in three days. |
| Reliability and availability | In a month, the system should function 90% of the time without any issues. Considering the system's dependability and maintainability, the user will have access to it 90% of the time. |

# // Making one more non func from wiki

# 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.**