

# SOFTWARE REQUIREMENTS SPECIFICATION

FOR

COINKEEPER

APRIL 2023

## TABLE OF CONTENT

### 1.INTRODUCTION

1.1 PURPOSE .....	02
1.2 SCOPE.....	02
1.4 STRUCTURE .....	03

### 2.SYSTEM OVERVIEW

2.1 PRODUCT PERSPECTIVE .....	03
2.2 PRODUCT FEATURES .....	04
2.3 USER ROLES AND CHARACTERISTIC.....	04
2.4 OPERATING ENVIRONMENT .....	05
2.5 ASSUMPTIONS AND DEPENDENCIES .....	05

### 3.USE CASE ..... 06

### 4.FUNCTIONAL REQUIREMENT ..... 07

### 5.EXTERNAL INTERFACE REQUIREMENTS

5.1 USER INTERFACE .....	09
--------------------------	----

5.2 HARDWARE INTERFACE .....	09
5.3 SOFTWARE INTERFACE .....	09
5.4 COMMUNICATION INTERFACE .....	10

## 6.NON-FUNCTIONAL REQUIREMENTS

6.1 PERFORMANCE REQUIREMENTS .....	10
6.2 SAFETY REQUIREMENTS .....	10
6.3 SECURITY REQUIREMENTS .....	10
6.4 SOFTWARE QUALITY ATTRIBUTES.....	11

## SOFTWARE REQUIREMENTS SPECIFICATION

### 1.Introduction

CoinKeeper, the revolutionary application designed to help individuals track and manage their expenses in a more effective and efficient manner, With CoinKeeper, users will be able to set a target amount of money to save at the end of each month, and input their daily expenditures accordingly. The application will provide users with real-time feedback on their spending habits, ensuring that they do not exceed their budget and stay within their means by the most innovative aspect of CoinKeeper is its notification system

One of the key features of CoinKeeper is its ability to provide users with analytical data on their expenses through the use of charts and graphs.

#### 1.1 purpose

The purpose of this software is analyze and manage people's expenses in an easy and friendly way.

#### 1.2 Scope

The "coinKeeper" is a great app designed to help users to manage their finances, save their money, track bills and purchases and know the expenses. The app offers a range of features such as the ability to add, adjust the amount of salary and the amount the user wants to save. In addition, determine a daily amount to help you reach your goal. Also, you can set a pre-plan for shopping within a specific budget. This app can be used by anyone to track their expenses and this helps them to reach their financial goals.

## 1.4 Structure

1. Introduction: represent the concept of the software and some key features. Also, the purpose and scope of this software .

2. System Overview: Here we give a view of product perspective, features provided, target users , work in which OS ,and a number of assumptions .

3. use case diagram: we illustrate how to deal with the system in order to get the maximum benefit of the software

3. Functional Requirements: give a detailed description for each software functionality, its component ,and services it provide

4. External interface requirements: These requirements identify external user interface components including user interface, software, hardware, communication interfaces).

5. Non-functional requirements: Non-functional requirements in the software development requirements document imply system functionality's general criteria. Thus, they include performance requirements, safety and security requirements, safety quality attributes (adaptability, flexibility, usability, etc.).

## 2. System Overview

### 2.1 Product Perspective

The idea of managing money through applications is not new, but these applications often lack the element of arrangement and alert, and also lack consistency and flexibility . The way the software deals with users input and output will give them the reason to visualize their expenses and let them know if they are under or above the average limit. through CoinKeeper, Anything related to expenses will be in one place . Do we provide a calculator? yes we do, do we provide a shopping list with a predicted value of purchases? Also yes we do . By the functions we provide it would cover all users needs.

## 2.2 Product features

1)explicit input of :

A)the salary.

B)the saving amount.

C)user daily limit (if user does not prefer the default choice).

default choice: by taking the salary and saving amount the user wants, we can calculate the daily limit by this. equation : subtracting from the salary the amount to be saved and then dividing the result by thirty.

2)notification: The power of allowing notification will motivate the user to be in touch with his goal.[1]

3)transactions: User input transaction into the application in three main ways A)

Copy the transaction and then paste it in specified field

B) Scan the bill

C) manually by typing

D)automatically by use automation in shortcut app[2]

4) View :by the use of charts and graphs will give users better understanding . This information can help users identify areas where they may be overspending and make adjustments to their budget accordingly.

5)Account settings page: It allows the user to freely modify his personal data, for example, image, theme, and even the currency

## 2.3 User rules and Characteristics

Our application accessible for all, it is designed to be easy to use and understand wherever possible for young people and adults who want to save money, as well as those that cannot manage their spending effectively, do not require them to be a technology expert if users can use the mobile phone, they can use our application easily.

[1] the work of notification system:

## **2.4 operating environment**

Our app targets smart devices that run on Android and iOS operating systems, and has been built through the Flutter framework to design UI for both Android and iOS without having to write a separate code base for each framework, it provides faster Writing code, reduced testing time, a rich documentation, ease of learning and use and the app loads quickly and runs smoothly. For the database "MySQL" will be used, it provides full support for each application development requirement.

Also for our application, it provides a good notification system for notifying users and reminding them of their purpose.

## **2.5 Assumptions and Dependencies**

We try to make our application integrated and free of problems that can affect the application's usage or functioning, And we also make every effort to anticipate problems and find solutions, Such as difficulty of use or privacy and user information, we strive to make the app clearer and easier and do not require difficult requirements. Our app must have permission to the date and time, Camera, and notification system to function, if the user doesn't give permission to the app, it won't function as it should, also the user cannot take advantage of all features of the app.

the diagram shows the use case of CoinKeeper app which illustrate relations and constraint diagrammatically so users understand the situations and be more knower of functionalities the app provide

## **4. Functional Requirements**

### **4.1 User story :**

As a user I must be able to register and enter my account.

Functional requirements:

- 1: Sign up and register page - high priority -.
- 2: Login with Email or phone number.
- 3: Phone number verification.
- 4: "I forgot my password" section, by sending a link or code to the E-mail or the phone number.

#### **4.2 User story :**

I must be able to log out from my account or deactivate it.

Functional requirements:

- 1: from Account settings page, system allows the user to get out of his account. this because user may want to logout and go back to the main page to log in again or sign up in another account
- 2: source :Logout or deactivate buttons.

#### **4.3 User story :**

I must be able to edit my personal data such as profile picture, name, E-mail, phone number and password.

Functional requirements:

- 1: from Account settings page - as a high priority -.
- 2: Edit the name and the profile picture.
- 3: Modify the Email or the phone number.
- 4: Changing the password.
- 5: Edit the currency.
- 6: The ability to choose dark or light mode.

#### **4.4 User story :**

As a user I must be able to enter my budget data, modify them, delete them.

Functional requirements:

- 1: from Home page - high priority -.
- 2: Entering, modifying, adding, deleting different financial data. 3: Saving & downloading the data whenever needed.
- 4: Displaying the data of the account.

#### **4.5 User story :**

I must be able to enter and update the purchases and the expenses.

Functional requirements:

- 1: from Home page - high priority -.
- 2: Entering the purchase prices by typing, pasting or scanning the bill. 3: Modify the price (if there's a mistake).

#### **4.6 User story :**

I should be able to receive notifications that will keep me updated on my financial status.

Functional requirements:

- 1: Warning notification if I get close to the daily limit.
- 2: Notifications indicating that I'm not able to buy more (exceeding the limit).

#### **4.7 User story :**

I should be able to display and view the statistics.

Functional requirements:

- 1: Settings page - high priority -.
- 2: Finance summary charts to display the financial status.
- 3: Display the amount of spending and saving.

#### **4.8 User story :**

I should be able to calculate my purchases, savings or anything related to my financial data in the same application without the need of an external calculator.

Functional requirements:

- 1: Settings page (Calculator button).
- 2: Calculator page - medium priority -.
- 3: Copy and paste the data or the results.

#### **4.9 User story :**

I must be able to list the preliminary hypothetical purchases to plan and manage my financial status.

Functional requirements:



- 1: Shopping cart button in the home page - high priority -.
- 2: Adding the item.
- 3: Determine the expected price.
- 4: Display the result of the expected money to buy the shopping cart items.

## **5 External Interface Requirements**

### **5.1 User Interface**

Our user interface will be designed to be intuitive and easy to use and have a clean and modern design with clear navigation and labeling, also will be responsive and adaptable to different screen sizes and devices, and it has a dashboard where users can see an overview of their account and any important notifications or updates.

### **5.2 Hardware interface**

Our app is developed for mobile devices with IOS and Android operating systems .A server and an internet connection is required.

### **5.3 Software interface**

**Operating System** The software is developed for ios and android systems devices to make it Available to everyone.

**Database** Because of the high performance, data security and schema is fixed, we choose "MySQL".

we go with flutter because it's an easy way to

#### **Flutter**

build and deploy visually attractive, natively compiled applications for mobile

### **5.4 Communication interface**

The software supports all types of updates in the operating system. We'll use some

advanced tools to assist us design and implement application functionality ,for instance the push notification ,SMS messages ,and guide tips for the first use or on an email sent as newsletter .

## **6 Non-functional Requirements**

NFRs are a set of specifications that describe the system's operation capabilities and constraints and attempt to improve its functionality. This kind of requirement describes how the system should perform, rather than what it should do.

### **6.1 Performance Requirements**

1. The system is able to handle thousands of concurrent users.
2. When a user requests to view their expenses, CoinKeeper should load and display the information within 4 seconds.
3. the system immediately updates whenever the user spends money.
4. loading data while login should not take more than 3 seconds.

### **6.2 Safety Requirements**

CoinKeeper has a reliable backup and recovery system to prevent the loss of user data in the event of a system failure or crash. The application will automatically create backup copies of user data on a regular basis and provide an easy way for users to restore their data if needed from the last backup.

### **6.3 Security Requirements**

1. The system must be protected from suspicious access: the user must enter a correct email address and password to log into the account.
2. Sensitive data of users such as salary and transactions added , must be of the highest standards of privacy to ensure it's not manipulated by unauthorized people because it will affect The whole functionality of the software.
3. After a certain number of login attempts,a pop up will appear on the screen to let the user reset the password ,email address is required here to make sure that the real app's user is behind this.
4. for more security purposes add the two factor authentication.

## **6.4 Software Quality Attributes**

### **6.4.1 Usability**

CoinKeeper application provides services to the users With ease and clarity, and it

is in points:

1. Ease of user access to his data at any time,
2. Understand buttons and headings with ease and clarity.
3. The user can create his investment portfolio with ease.

#### **6.4.2 Availability**

The system in CoinKeeper is available to users at any time. In the event that users are experiencing problems, the pages should display a temporary notification showing when the system will return.

#### **6.4.3 Compatibility**

It is a type of software test to check if CoinKeeper is able to run on different operating systems, devices, apps and mobile devices. In addition, CoinKeeper should be compatible with devices running a recent operating system.

[see the prototype of the app in figma](#)