Faculty of Computers and Artificial Intelligence-
Cairo University (credit hours system)

Kher Wallet

Supervised by:

Dr. Khaled Wassif

TA. Ashraf Mohey

Implemented by	
20186031	Nada Mohamed
20186008	Sarah Khaled
20186043	Mark Rofaeel
20186007	Ayat Hany
20186049	Yasmine Shehab

Graduation Project

Academic Year 2021-2022

Table of Contents

List of Figures

List of Tables

List of Abbreviations

Abbreviation	Meaning
API	Application Programming Interface
GUI	Graphical User Interface
ERD	Entity Relationship Diagram

Table 1: List of Abbreviations

Chapter 1: Introduction

1.1 Motivation

Nowadays, there are a lot of organizations that accept donations from different people, either by donating money, clothes, devices, books or even by volunteering in many events.

It takes time and effort for people to reach to these organizations so they can donate. Also, people do not even know when any organization need help during event times.

Another problem is with the organizations! Organizations do not have an all-in-one platform to communicate on with people interested.

Considering this, our motivation is to solve these problems by implementing an Android/IOS application that gathers different kinds of organizations, that help people to donate in an easier way and notify them when there is a new post or event around the corner.

Kher Wallet may be the first donating platform in the Egyptian region.

1.2 Problem Definition

Our problem is that people spend so much time and effort to reach different organizations as charities, hospitals, and orphanage that has led to the shortage of the overall amount of donations.

Another key fact to remember is that those organizations do not have the capability to reach large amount of people to notify them with new posts or events.

1.3 Project Objective

The main objective of our application is filling the gap between different organizations and people by implementing Kher Wallet.

An application that gathers different types of organizations (charities, hospitals, and orphanage) which can be easily used by people to complete their donation process right at their own hands.

1.4 Gantt Chart of project time plan

Task Title	Start date	End date	Duration
Project definition	10/1/2021	10/8/2021	7
Idea brainstorming	10/22/2021	11/1/2021	10
Analysis	11/15/2021	12/15/2021	30
Requirements	12/18/2021	12/22/2021	4
Mid-year documentation	12/19/2021	2/22/2022	65
Changes in the documentation	2/11/2022	2/14/2022	3
Kher Wallet Logo	2/14/2022	2/20/2022	6
UI prototyping screens	2/15/2022	2/20/2022	5
Mid-year Presentation	2/20/2022	2/24/2022	4
Design and architecture	2/25/2022	7/4/2022	129
Flutter implementation	3/8/2022	7/4/2022	118
Data gathering	6/22/2022	7/4/2022	12
Acceptance	7/4/2022	7/7/2022	3
Application Testing	7/6/2022	7/10/2022	4
Final documentation	7/5/2022	7/13/2022	8
Final Presentation	7/9/2022	7/12/2022	3
Poster	7/10/2022	7/13/2022	3

Table 2: Project time plan

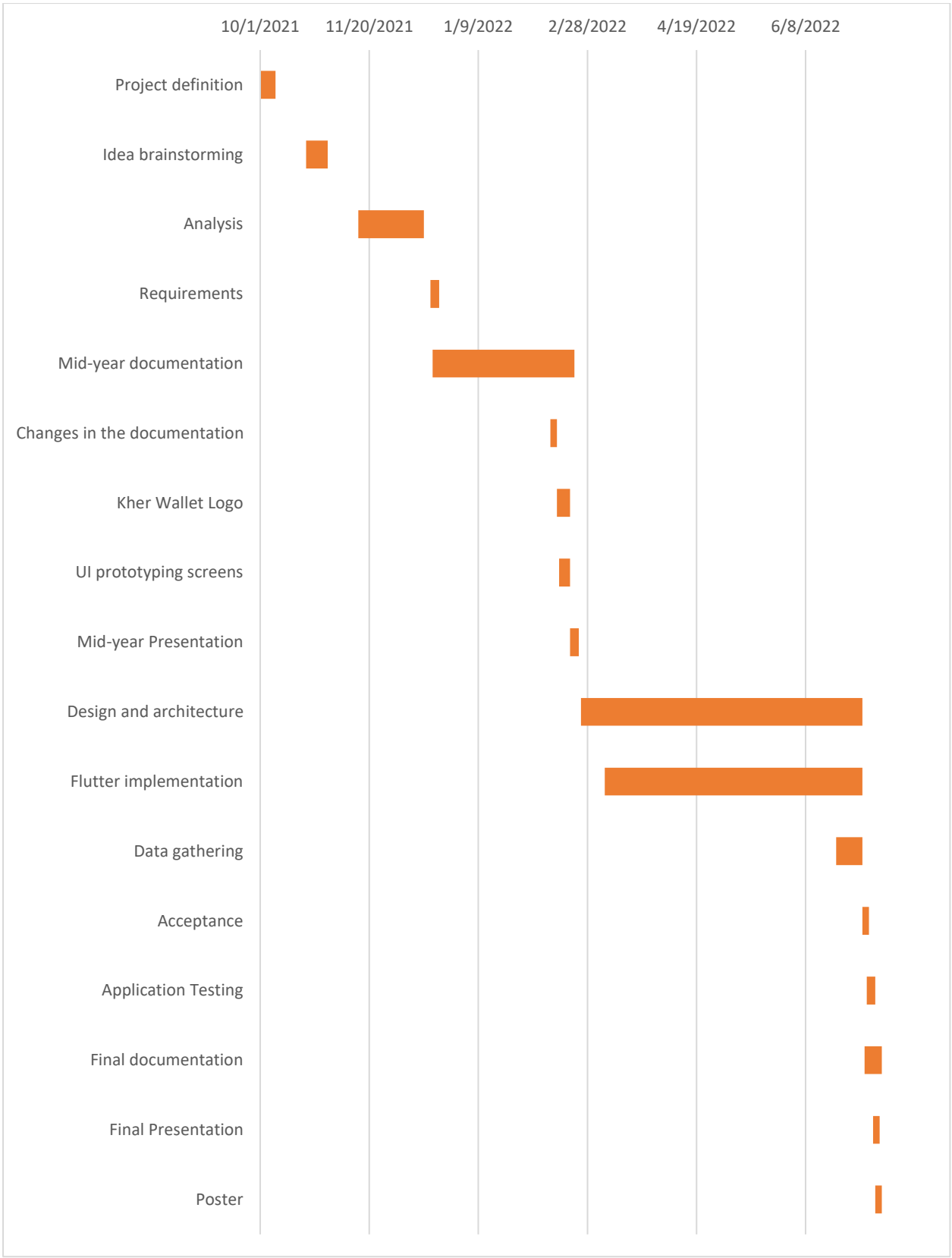


Figure 1: Gantt Chart

1.5 Project development methodology

Throughout the process of doing this project, we followed the Waterfall methodology and principles and the basic client-server architecture.

The application itself is our client, we used Android Studio as the application's platform.

For the server-side, we used Google Firebase services to deploy the needed functionality in the form of an API that the application is going to make use or communicate with. We used Cloud Google Firestore, Storage, Google Analytics, and Cloud messaging.

We used Flutter as a technology and the code is written in Dart programming language. We chose these technologies as it is one of the leading ones nowadays.

1.6 The used tools in the project

Software:

Flutter

Dart

Google Firebase

Android Studio Platform

GitHub

Google Forms

External libraries

Hardware:

Windows Laptops

Android Devices

1.7 Report Organization

In the next chapter, we will talk about the work related to our project.

After that, we will talk about the system analysis in chapter 3, which consists of functional, non- functional requirements and use case diagrams.

Then in chapter 4, we will be showing the system design in different diagrams as: system component diagram, system class diagrams, sequence diagrams, project ERD, and system GUI design.

And finally in chapter 5, we will be discussing the implementation and testing of our application.

Chapter 2: Related work

2.1 Introduction

In this section, we will be spotting the light on the different applications that are related somehow to our application either in functionality or in business or social context, so we will make a comparison between these applications and Kher Wallet and show what differences between both.

2.2 Waffarha vs Kher Wallet

- Waffarha is a local application that aim to make a great saving to our online customers by offering daily deals with big discounts. Couple of months ago, they added a new subsection in Bill Payments called Donation.

- In this donation subsection, they added several organizations (operators). By entering the amount needed for donation and personal mobile number, people can donate easily.

- After that one should click on 'Complete payment' to process the donation.

Here are some of differences between Waffarha and Kher Wallet:

Point of comparison	Waffarha	Kher Wallet
Business logic	The donation is not the main business logic of the application, it is just a subsection in the whole application.	The business logic is all about donations and requests for both customers and organizations.
Total number of organizations	Only 11 organizations are available to donate to.	Initially we started with 14 organizations, but definitely will increase soon.
Donation amount	Customer should enter the specific amount needed for donation.	No one knows the amount of donation except the representative sent by the organization. No need for anyone to know such thing.
UI for the organizations	No UI is provided for the organizations to know the number of donations.	What makes Kher Wallet special is the organization's UI. They can post, notify people when there is a new event or see and analyze the number of donations requested.

Table 3: Waffarha vs Kher Wallet

2.3 Summary

As seen in the table above, there are huge differences between the two applications. Kher Wallet is a donating application from A to Z. It has a lot of functionalities that goes around the idea of donating and volunteering. On the contrary, Waffarha is a saving application that has nothing to do with our application's idea. Waffarha only added one of the functionalities we made with much simple layout several months ago.

Chapter 3: System Analysis

3.1 Project specification

3.1.1 Functional requirement

Donator's functionalities:

1. Sign up:
   - Donator should be able to **create an account** on our application with their full name, email, password, address, and phone number.

2. Sign in:
   - Donator should be able to **sign into** the application by their email and password.

3. Forget/change password:
   Donator should be able to forget password during the sign in process. Donator should be able to change password from their profile. An email will be sent where password can be easily reset.

4. Sign out:
   - Signed-in donator should be able to **sign out** of their current account at any given time.

5. Update donator's profile:
   - Signed-in donator should be able to **update** any information in profile at any given time such as name, email, address, or phone number.

6. View notifications:
   - Donator should be able to **view notifications** whenever a new organization is added, or a new donation category is available, or when there is an event where help is needed, or when specific needed cases are posted by the organizations.

7. View posts:
    - Donator should be able to **view** the upcoming events where help is needed.

8. Expand post description:
    - Donator should be able to **view more** about a specific post with full description by expanding it.

9. Apply to a post/event:
    - Through a Google form, a donator can **apply** to an event or can subscribe to know more about a specific post.

10. Donate:
    - Donator should be able to **donate** by requesting a representative to donate money, clothes, devices, or books. The donation is then added to requests.

11. Remove from requests:
    - Donator should be able to **remove donations** only in the first 24 hours form the action.

12. View requests:
    - Donator should be able to **view** different donations in requests.

13. Filter posts/events:
    - Donator should be able to **filter** the different organizations to view their posts.

14. Write feedback:
    - Donator should be able to **write the feedback** of his own**.**

15. Request help:
    - Donator should be provided by the most **frequently asked questions** with provided answers.

16. Contact us:
    - Donator should be able to **contact us** to solve any problem via email.

Organization's functionalities:

1. Sign in:
    - Organization should be able to **sign into** the application by their email and password (provided by administrators).

2. Forget/change password:
    - Organization should be able to forget password during the sign in process. Organization should be able to change password from their profile. An email will be sent where password can be easily reset.

3. Sign out:
    - Signed-in organization should be able to **sign out** of their current account at any given time.

4. Update organization's profile:
    - Signed-in organization should be able to **update** any information in profile such as contact email, details or main branch and phone number.

5. View posts:
    - Organization should be able to **view** the posts it has already posted.

6. Expand post description:
    - Organization should be able to **view more** about a specific post with full description by expanding it.

7. Add post:
    - Organization should be able to **add** any upcoming post where help may be needed which is then send as a notification to donators.

8. Update post:
    - Organization should be able to **update** existing post such as event code or details.

9. Remove post:
    - Organization should be able to **remove** existing post.

10. View requests:
    - Organization should be able to **view** the requests made by donators.

11. Edit requests:
    - Organization should be able to **edit** any request's current state (in progress/ completed).

12. Analyze data:
    - Organization should be able to **analyze** the number of different types of donations made and a pie chart is generated based on these numbers. Also, the total number of requests is given to the organization.

13. Write feedback:
    - Organization should be able to **write** the feedback of his own.

14. Request help:
    - Organization should be provided by the most **frequently asked questions** with provided answers.

15. Contact us:
    - Organization should be able to **contact us** to solve any problem via email.

3.1.2 Non-functional requirement

1. Portability:
   - The software will be available for two platforms as it will run on both Android and iOS devices.

2. Usability:
   - Users can easily determine what a feature is and what it can do, also, users can easily navigate the application's interface.

3. Availability:
   - The application and the host database on cloud will be available to use 24/7.

4. Response time:
   - The application should load at not more than 5 seconds.

5. Scalability:
   - More requests and data can be handled as the host is on cloud and it can scale as we go.

6. Authentication:
   - The application will not accept users without authenticating them through Google's Firebase Authentication technology.

7. Authorization:
   - The application will not accept any organization from accessing the system without entering the specific email and password provided by administrators.

8. Security:
   - The password should contain at least 8 characters including at least one uppercase letter, at least one lowercase letter and a digit. The password is then hashed through Google's Firebase Authentication technology.
   - The email must be unique. The phone number of the user should not be less than 11 digit and must be unique.
   - Donator should agree to the Terms of Services and Private Policy.

3.2 Use Case Diagrams

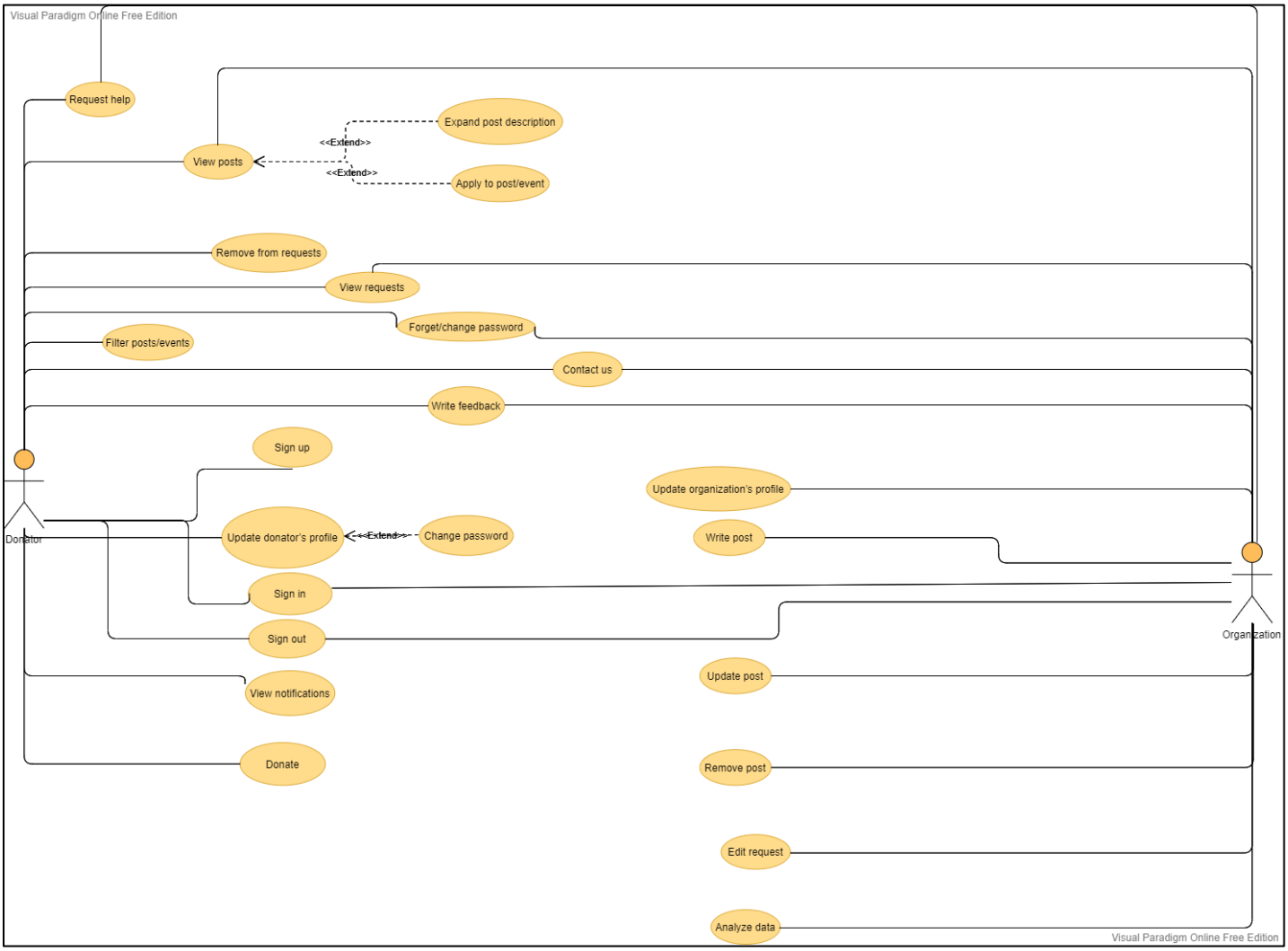


Figure 2: Use Case Diagram

In Figure 2, the use case shows the different functionalities that a donator or an organization can execute. Some functionalities are extended or included under other functionalities.

3.3 Use Case Tables

Use Case Name:	Donate	
Actors:	Donator	
Pre-conditions:	Donator signed in successfully and navigate to organization interested in.	
Post-conditions:	The donator can now donate by requesting an organization's representative and then the donation is added to the requests.	
Flow of events:	**User Action**	**Application Action**
	1- Donator signs into the application and click on places category then charities from the drawer.	
		2- Application shows different charities.
	3- Donator may select Resala Charity Organization.	
		4- Application lets the user choose what kind of donation he needs.
	5- Donator may select Money Donation.	
		6- Application shows different types of donations either using bank numbers or by requesting a representation.
	7- Donator select Request Representative.	
		8- Application adds the type of donation to the database and it can be viewed from the request section.

Table 4: Use Case Table 1

Use Case Name:	Write post	
Actors:	Organization	
Pre-conditions:	Organization signed in successfully, navigate to home and click on '+' button.	
Post-conditions:	The organization can write event easily.	
Flow of events:	**User Action**	**Application Action**
	1- Organization signs into the application.	
		2- Application verify the data from the database then show the home screen.
	3- Organization click on '+' button and write post details and code in the form provided.	
		4- Application store this post to the database, that then is shown to the donator's home screen and is also sent as a notification to the donator.
Exceptions:	**User Action**	**Application Action**
	1- Organization may lose internet connection while writing a post.	
		2- Application will not be able to store such post in the database.

Table 5: Use Case Table 2

Chapter 4: System Design

4.1 System Component Diagram

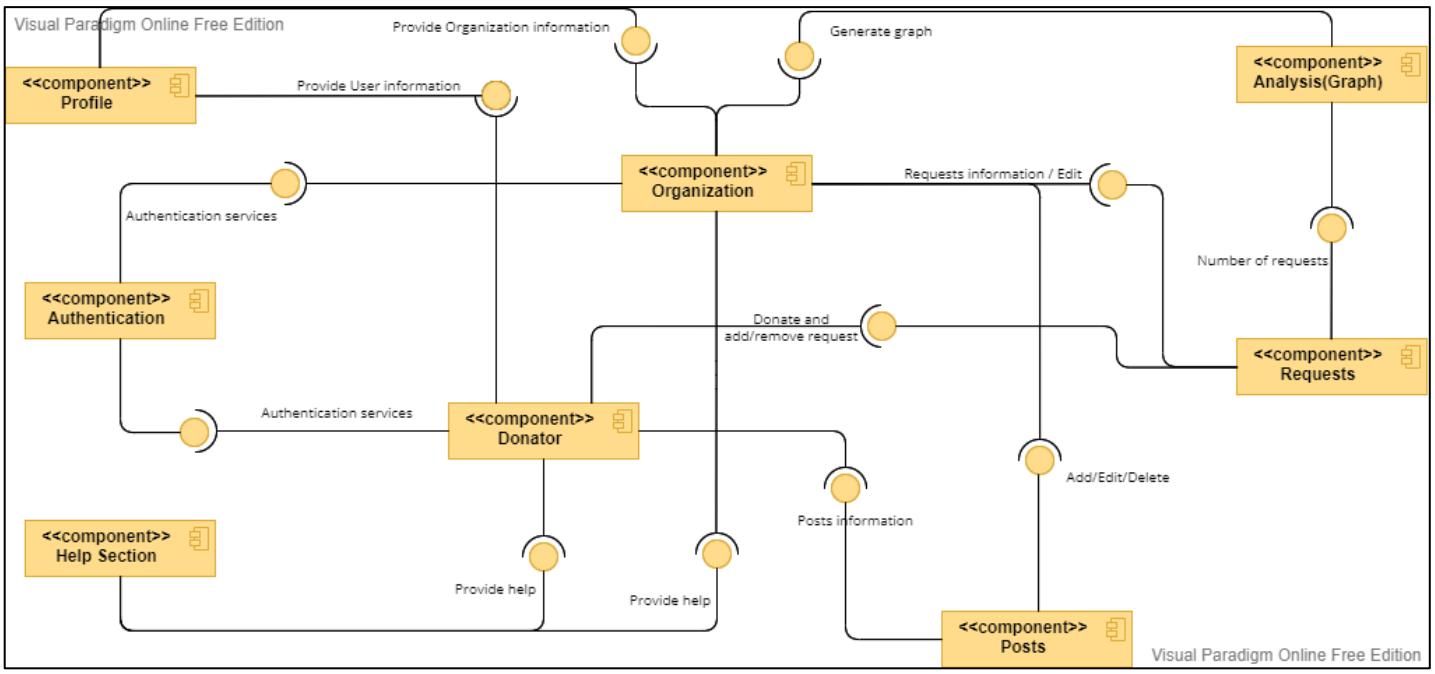


Figure 3: System Component Diagram

In Figure 3, the component diagram shows how components are wired together to form larger components or software systems.

4.2 System Class Diagrams

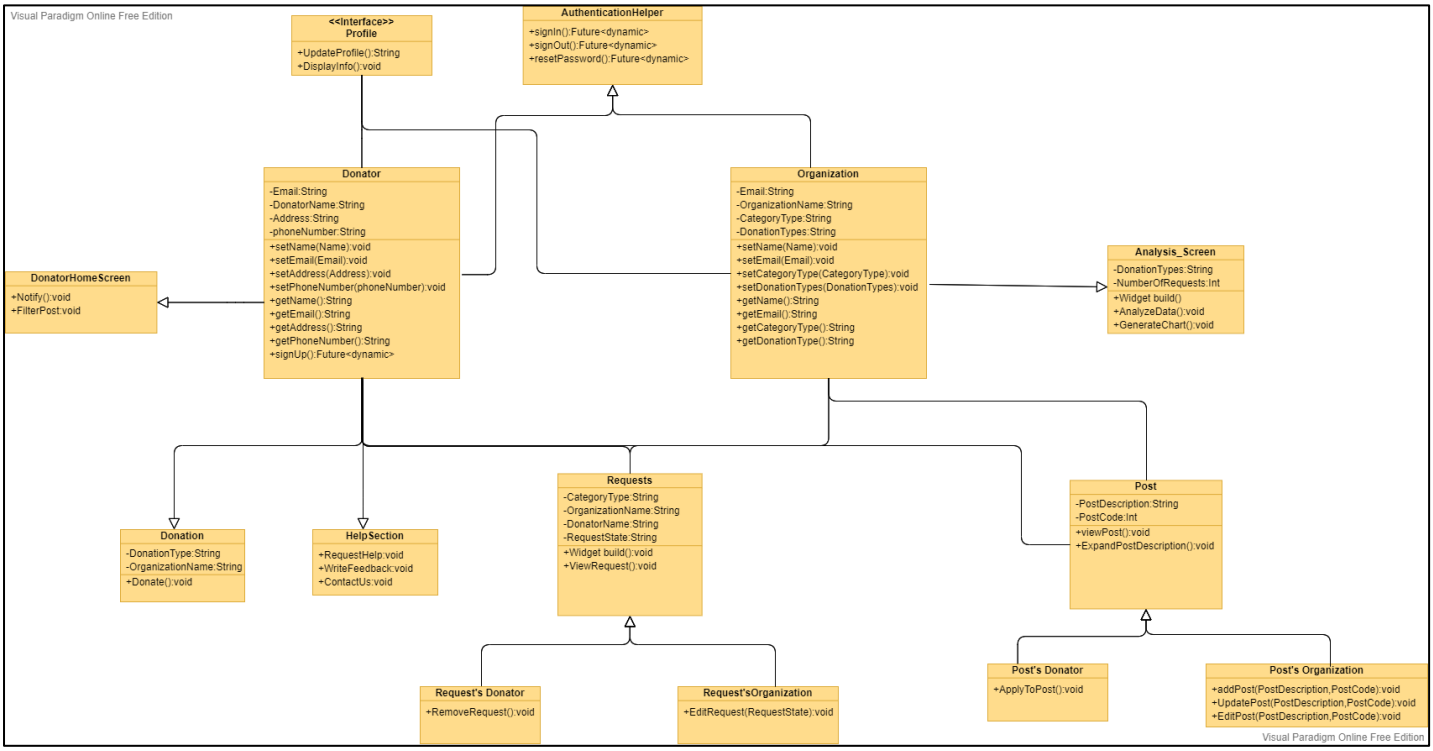


Figure 4: System Class Diagrams

In Figure 4, it shows the different classes and the relations between them. Here are the main classes (donator and organization) and the classes connected to them.

4.3 Sequence Diagrams

- Donating scenario
  - Donator signs into the application successfully. From the left drawer, category lists tab is selected and then places categories. Charities is then selected where Resala Charity is present. The donator chooses to donate clothes and then requests a representative to come to his doorsteps to complete the donation process. This request is saved to the database and is retrieved by the Resala charity anytime.

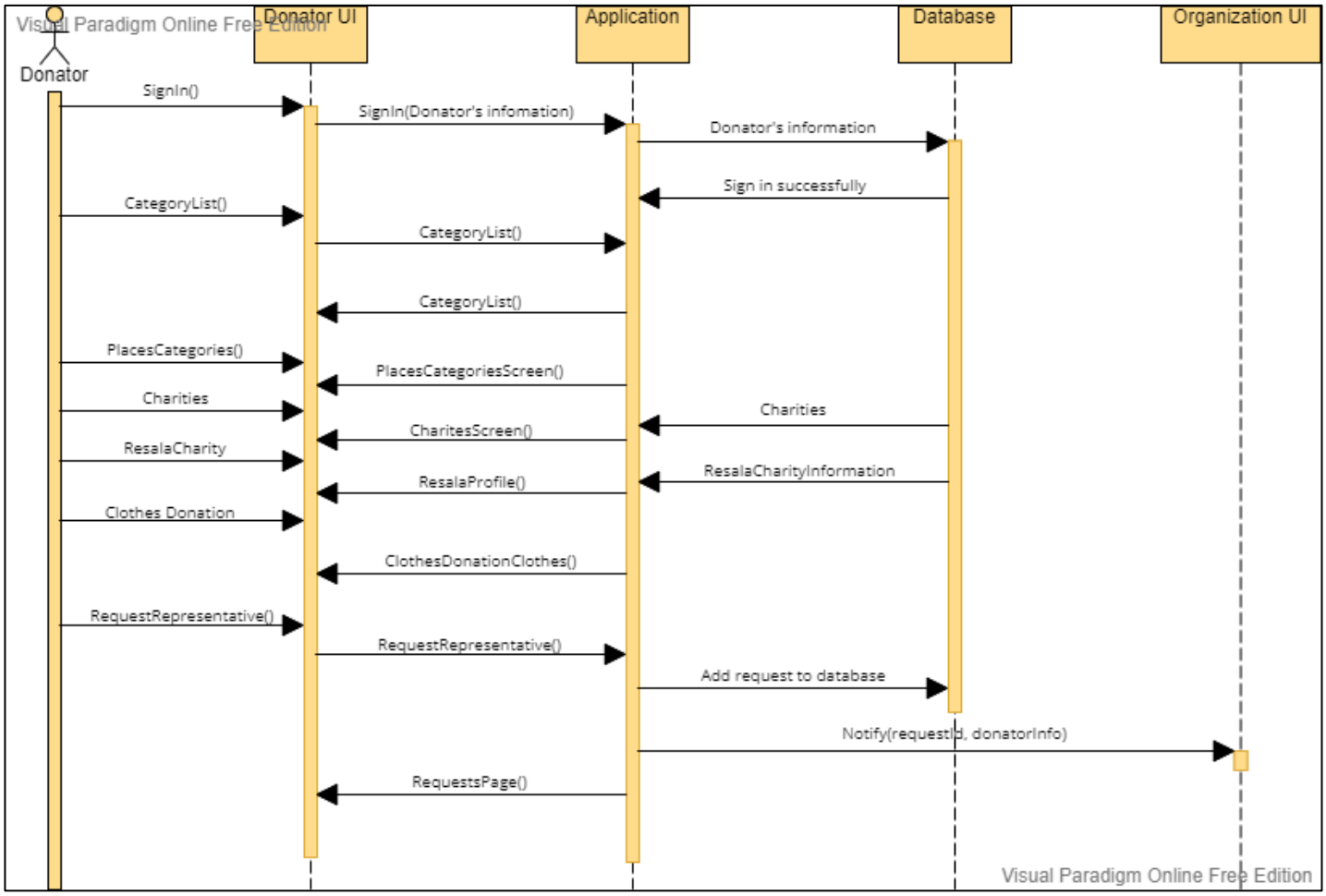


Figure 5: Sequence Diagram - Donating scenario

- Adding new post scenario
    - Organization signs into the application successfully. From the Home screen, the organization clicks the “+” button to add new post. The organization add the post code and details and then click on “post” button. This post is saved to the database and is retrieved by the any donator anytime.

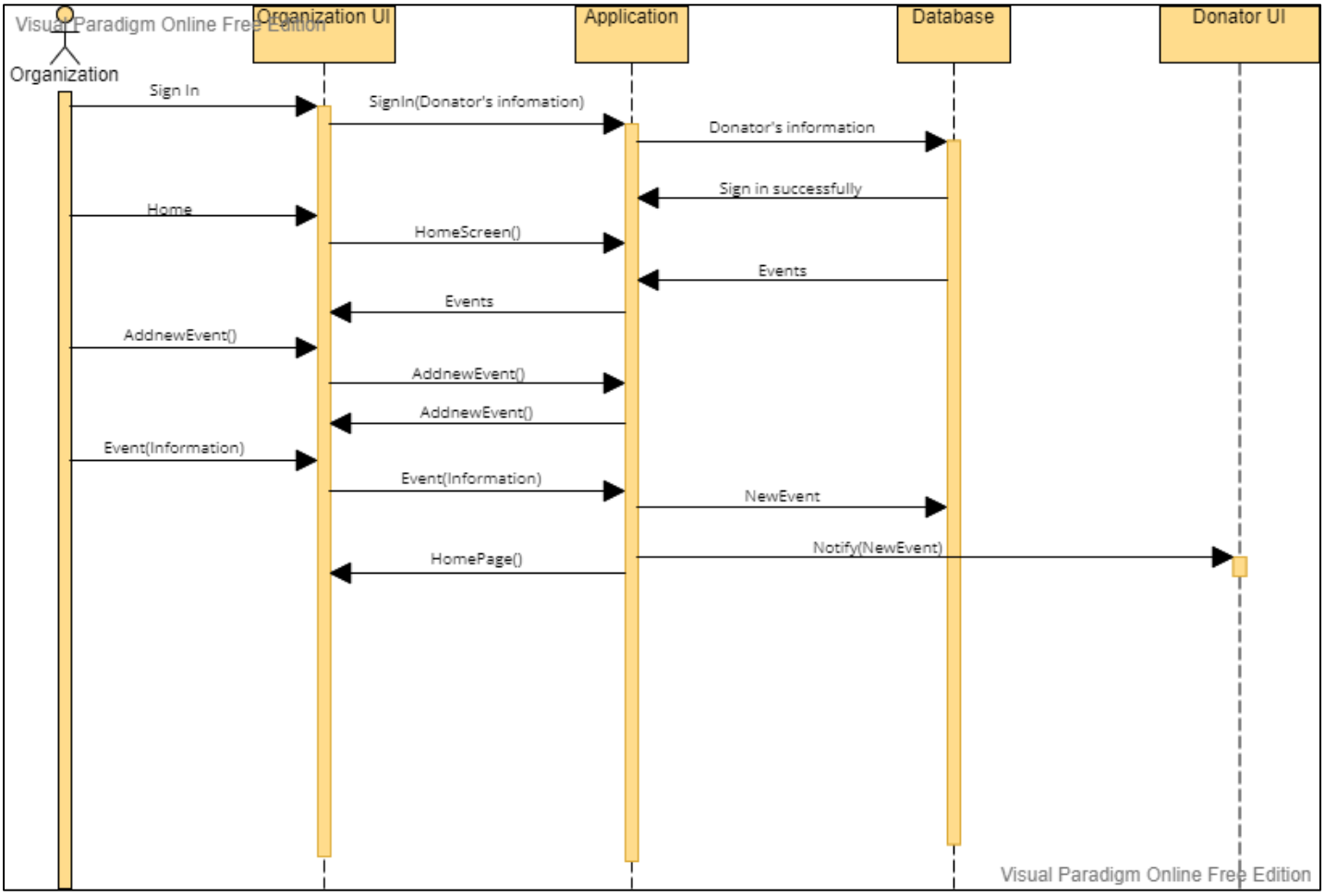


Figure 6: Sequence Diagram - Adding new post scenario

4.4 Project ERD

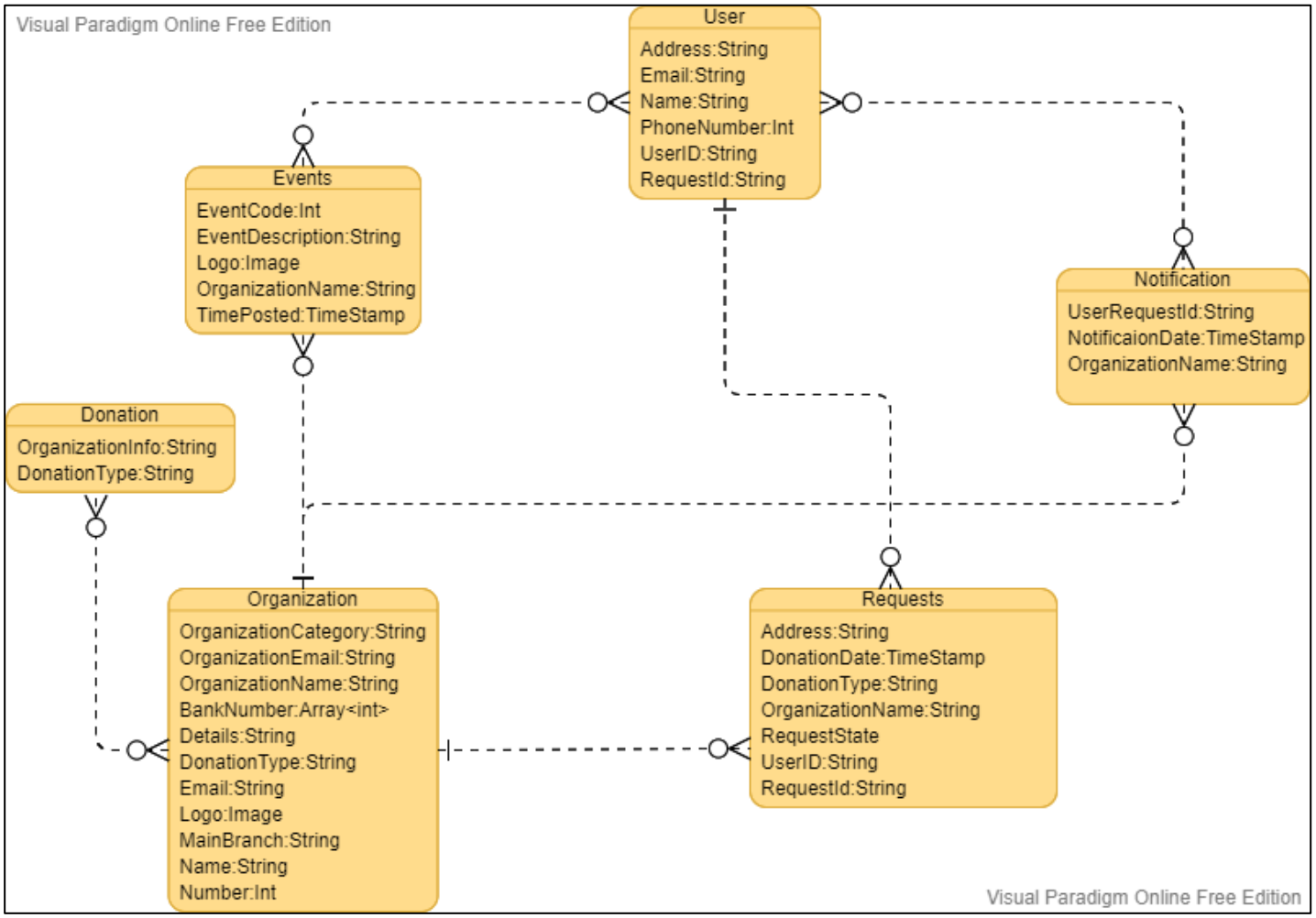


Figure 7: Entity Relationship Diagram

In Figure 7, it shows the different tables in the database the relations between them. There are Many-To-Many, Many-To-One, One-To-Many and One-To-One relationships.

4.5 System GUI Design

4.5.1 Donator's GUI

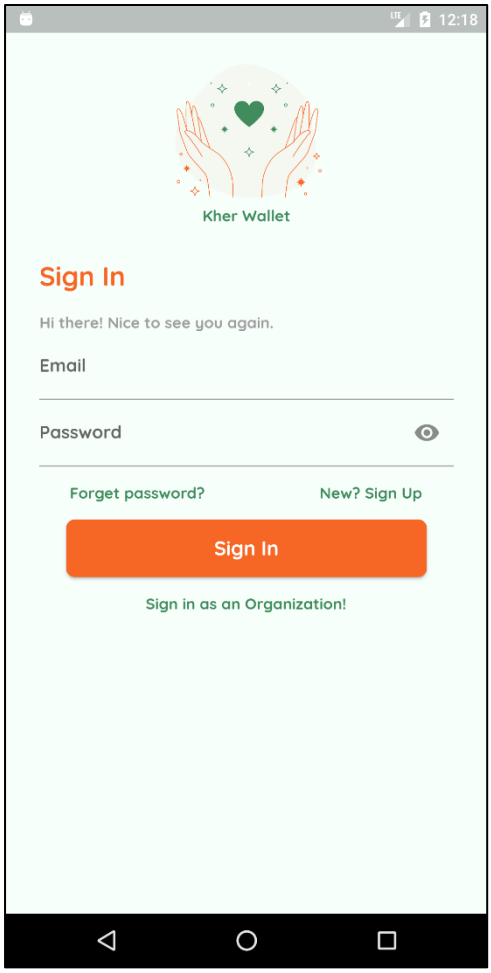


Figure 8: Donator's sign-in screen

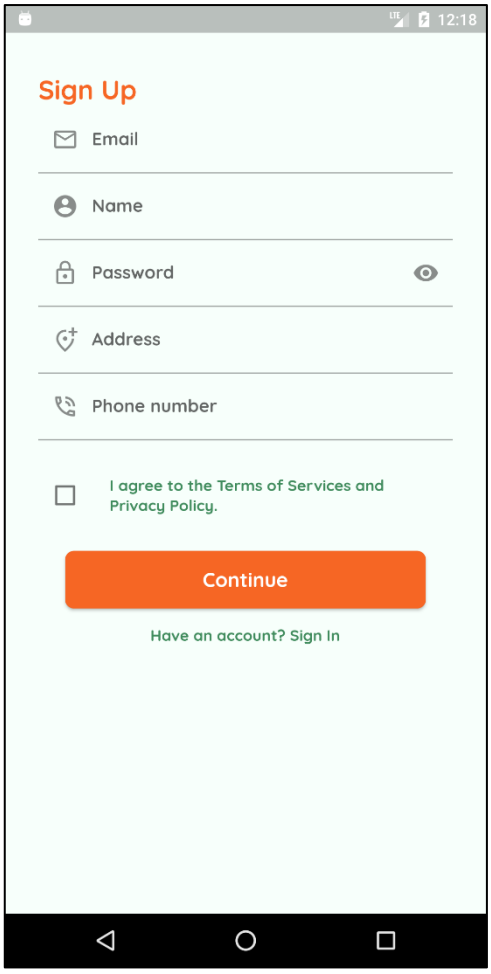


Figure 9: Donator's sign-up screen

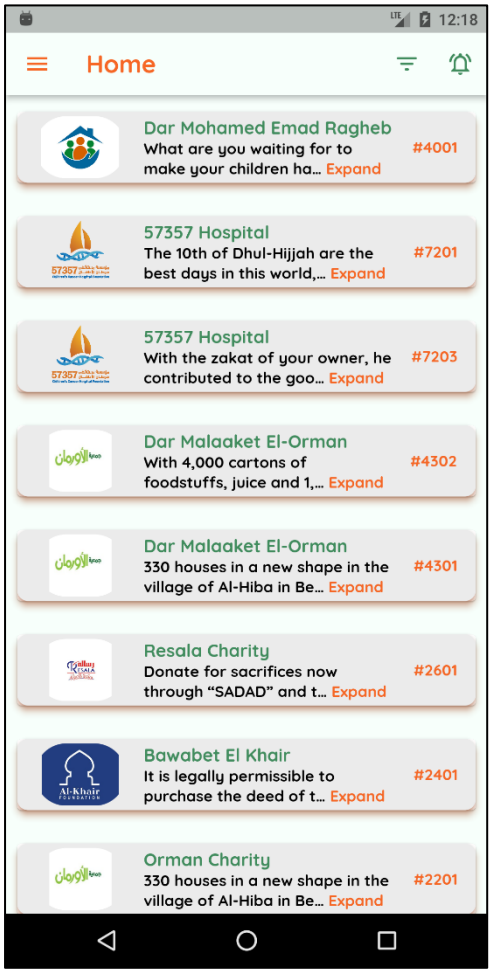


Figure 11: Donator's home screen

Figure 10: Expand post functionality

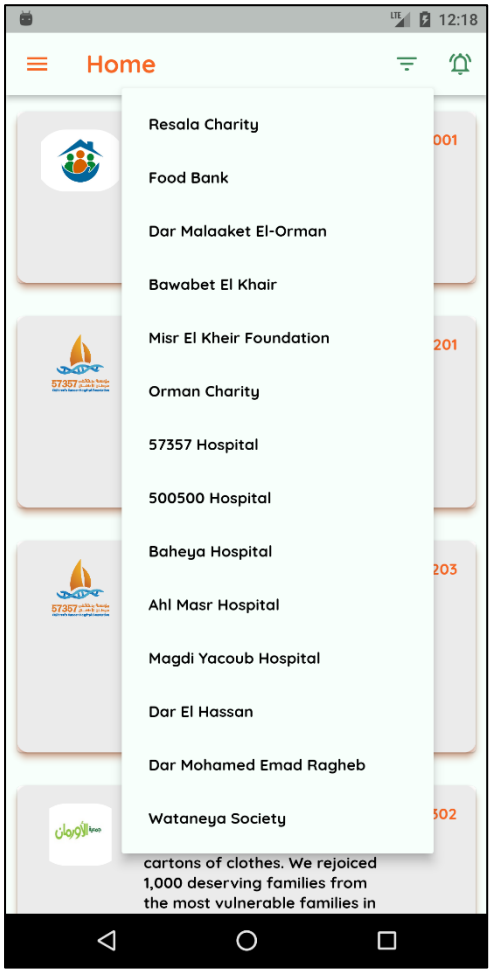


Figure 13: Filter functionality

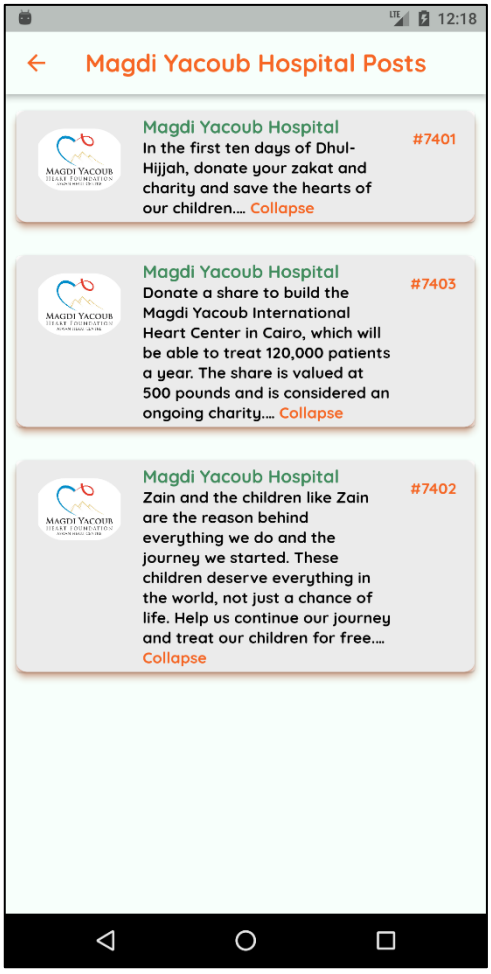


Figure 12: Magdi Yacoub Hospital posts

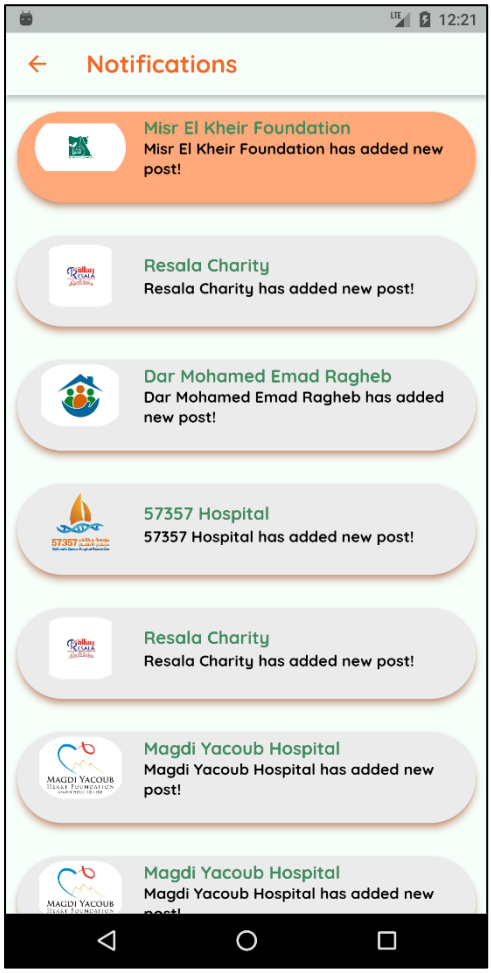


Figure 15: Donator's notifications screen

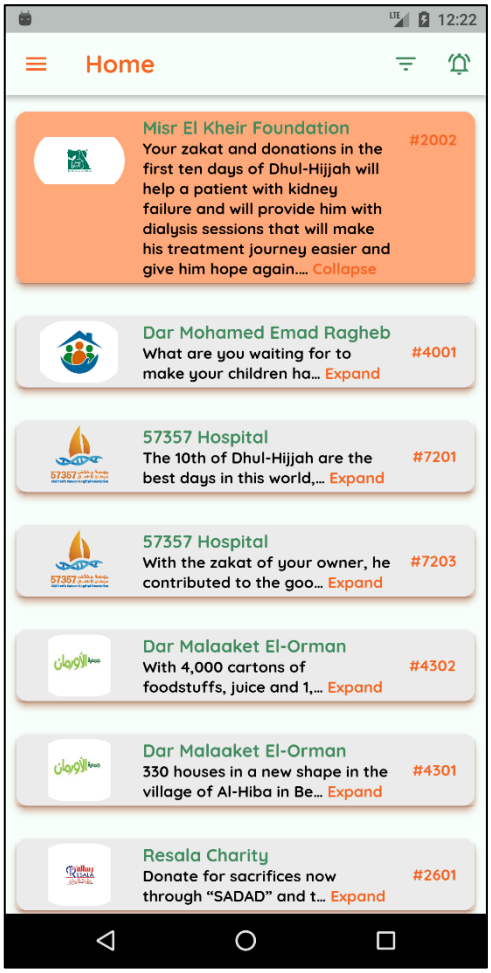


Figure 14: New notification alert

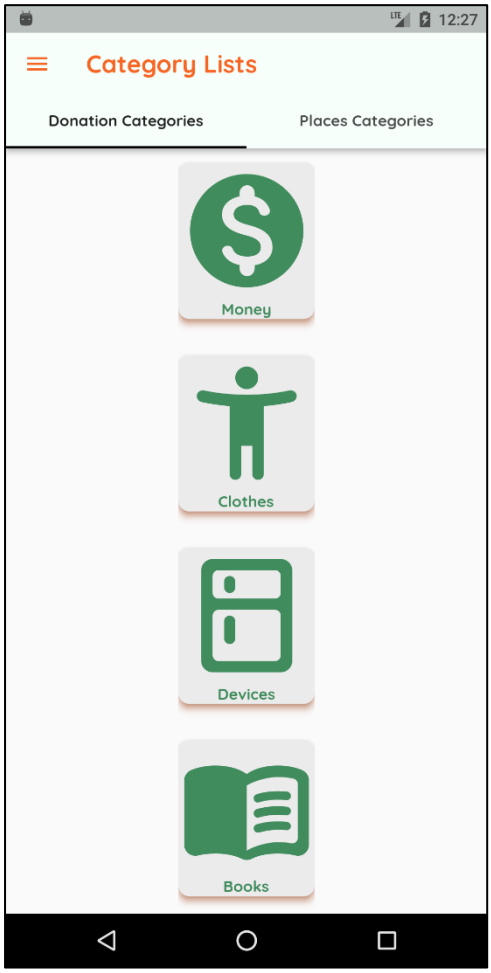


Figure 17: Donation categories

Figure 16: Places categories

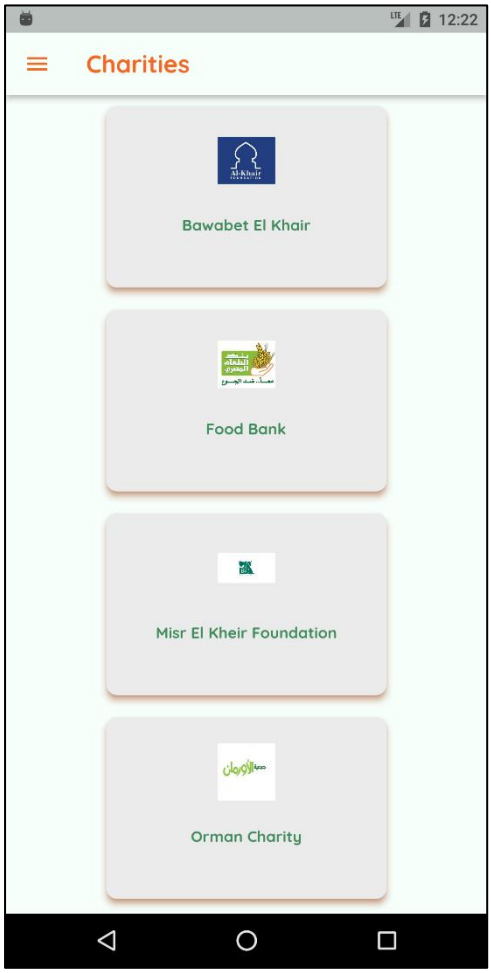


Figure 19: Example of charities

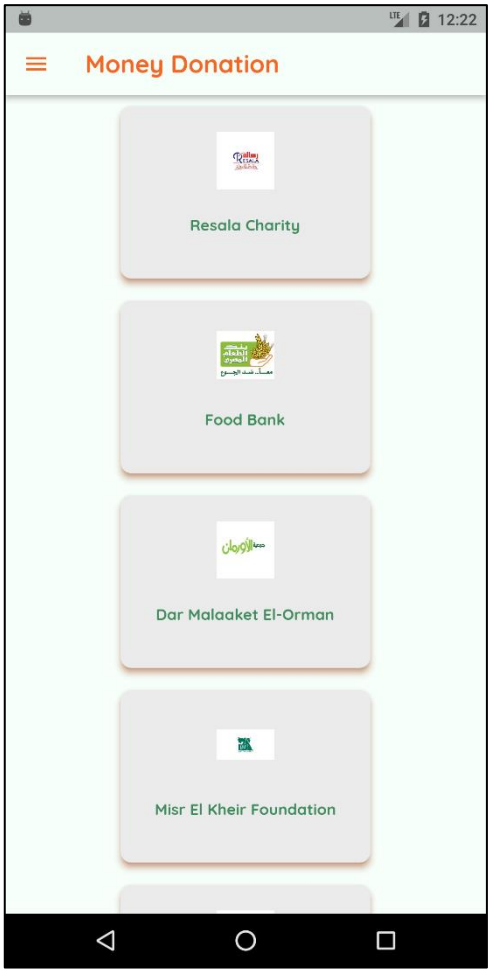


Figure 18: Organizations that accept money donation

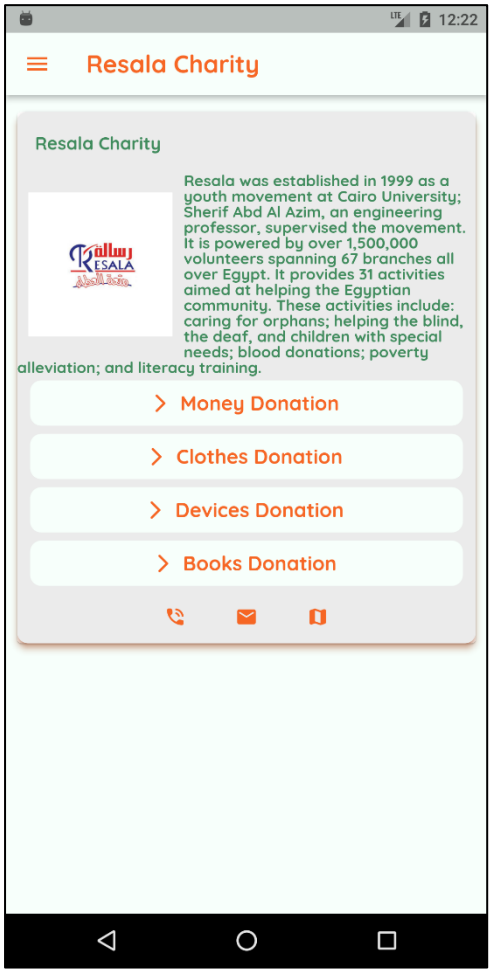


Figure 21: Resala Charity profile

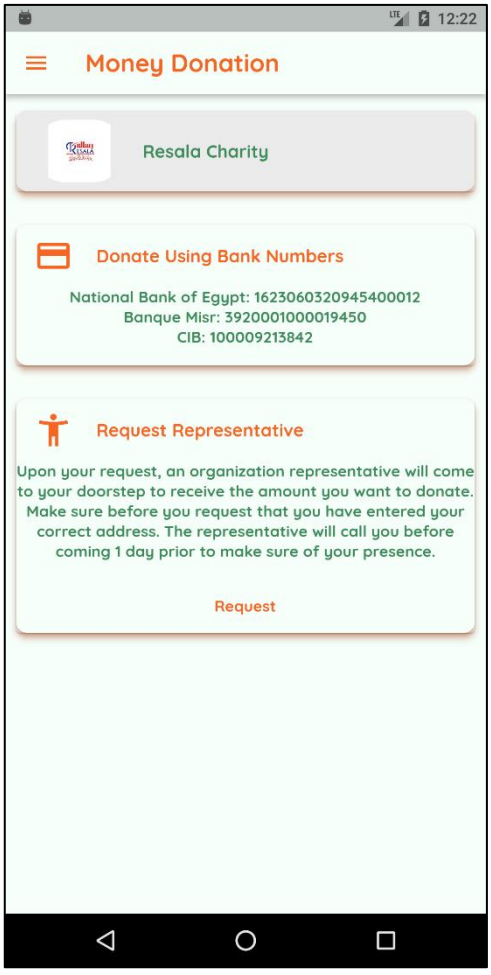


Figure 20: Money donation option in Resala

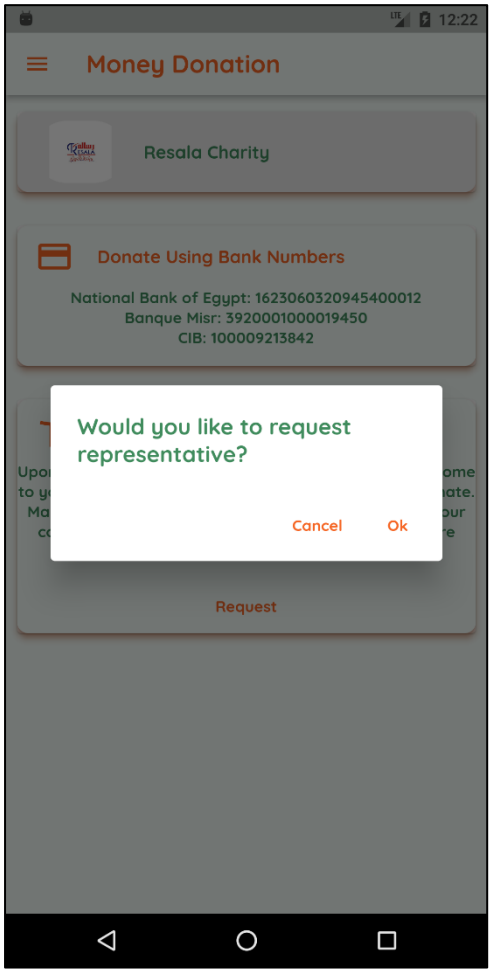


Figure 23: Alert dialog to request

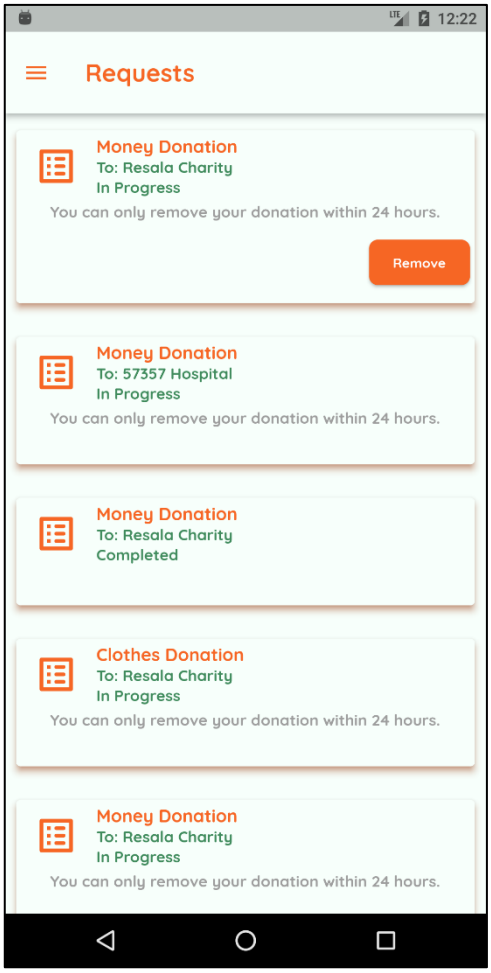


Figure 22: Donator's request screen

Figure 25: Help screen

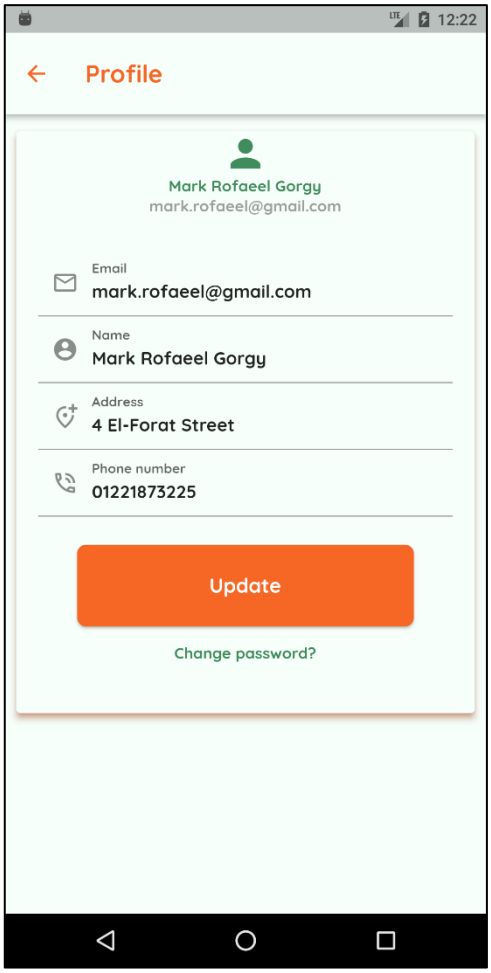


Figure 24: Donator's profile screen

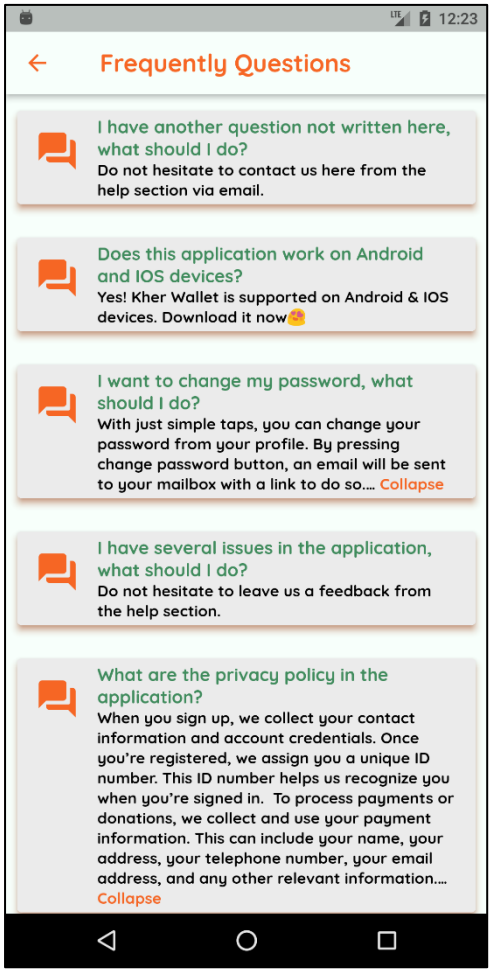


Figure 27: FAQ screen

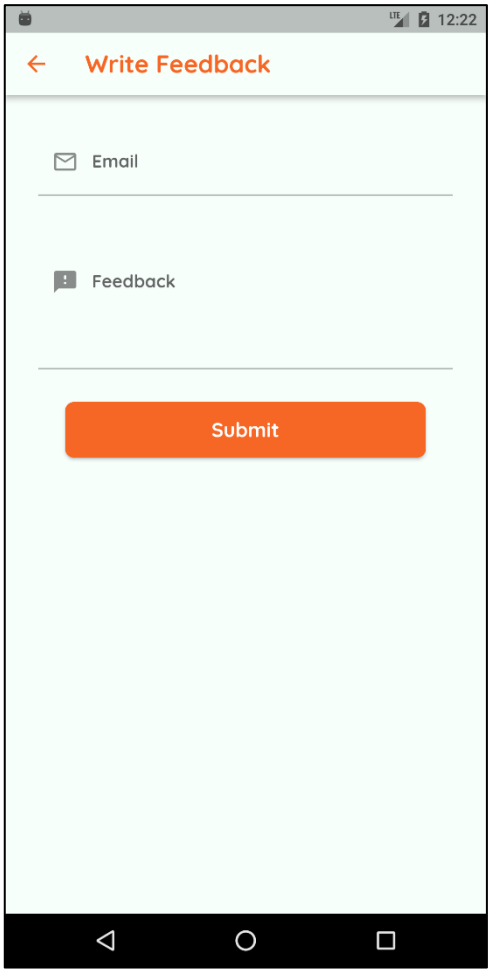


Figure 26: Write feedback screen

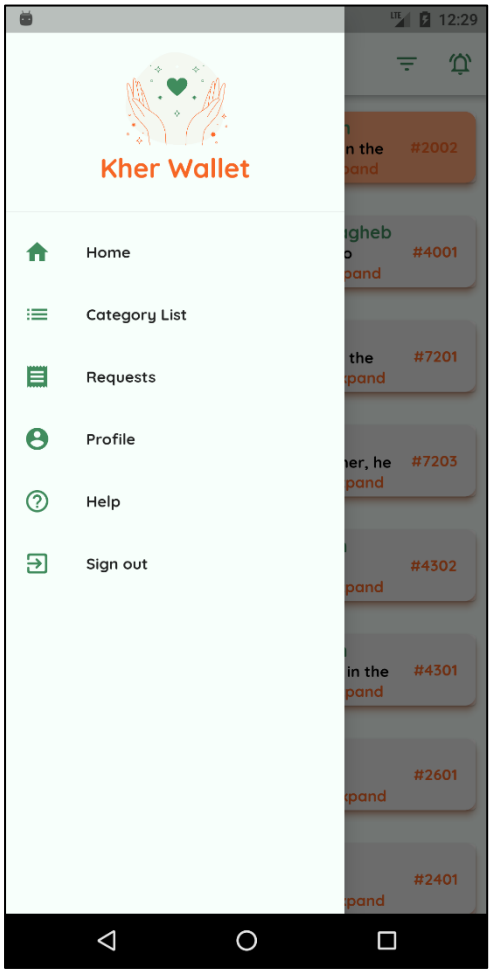


Figure 29:Donator's Navigation Panel

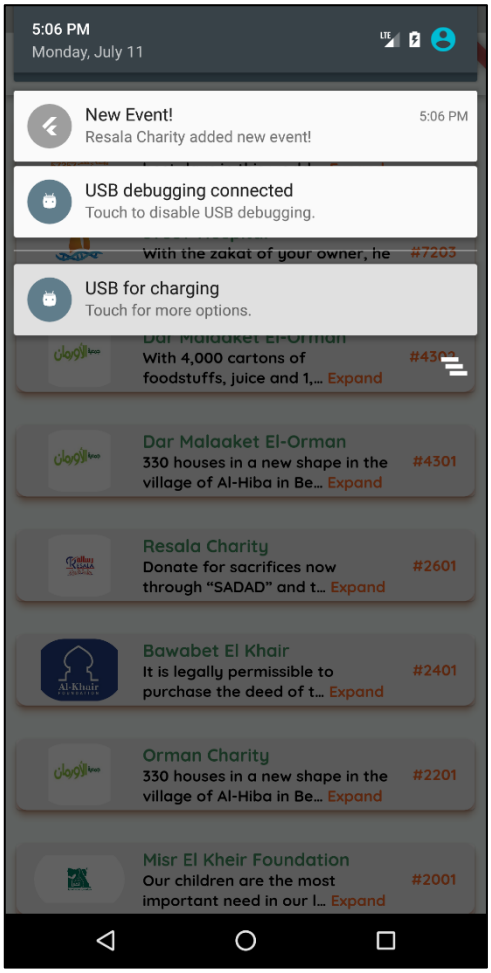


Figure 28: Donator's background notifications

4.5.2 Organization’s GUI

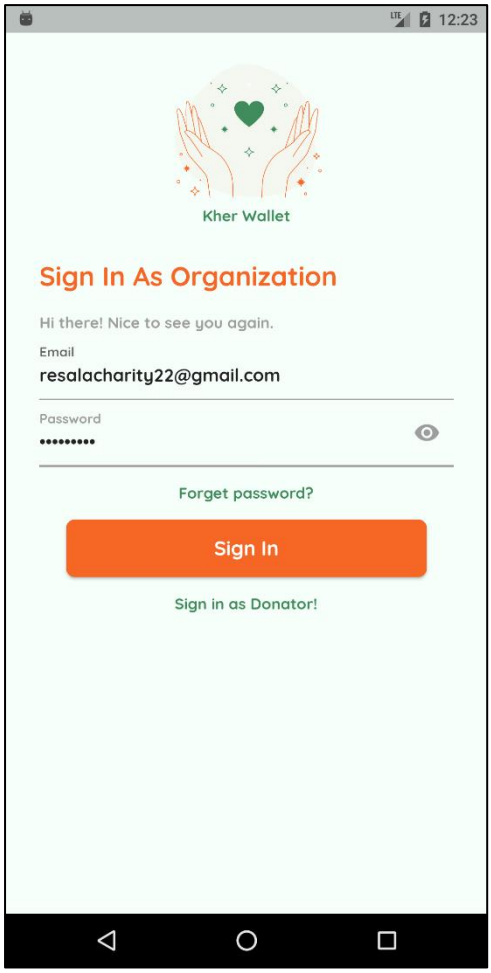


Figure 31: Organization's sign-in screen

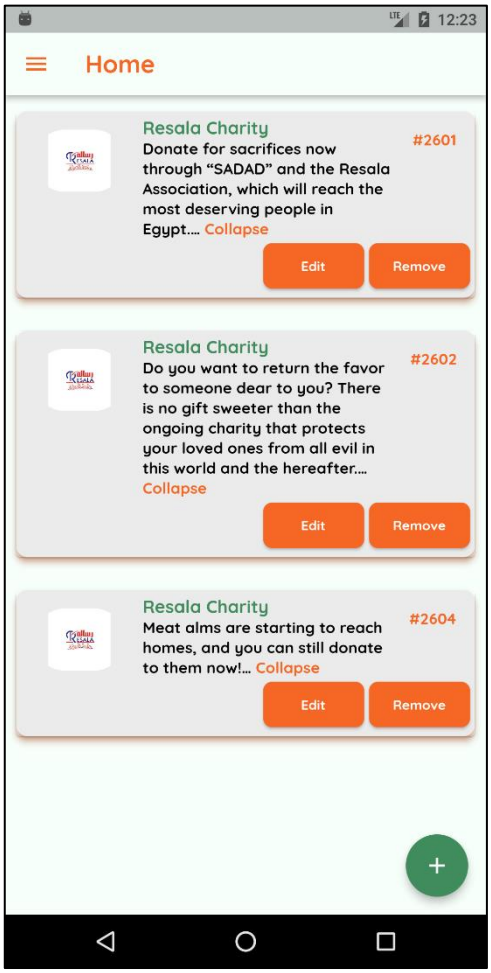


Figure 30: Organization's home screen

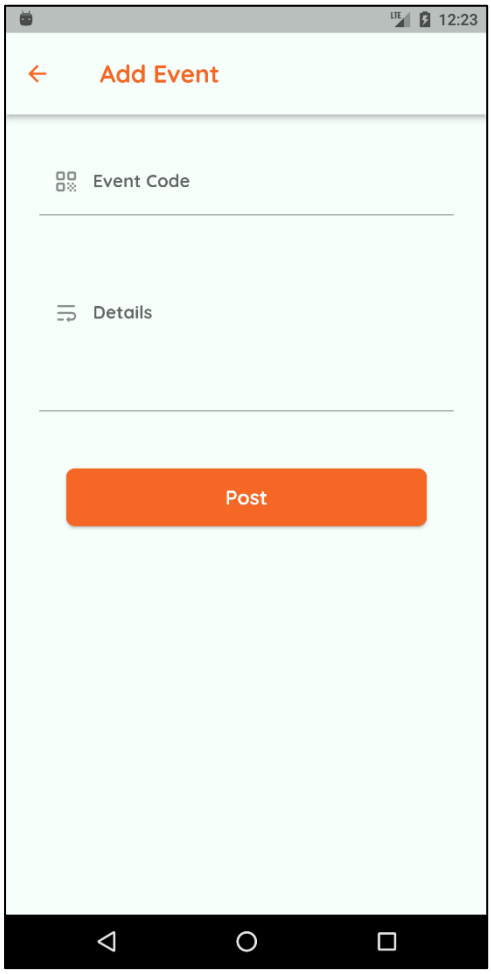


Figure 33: Add event screen

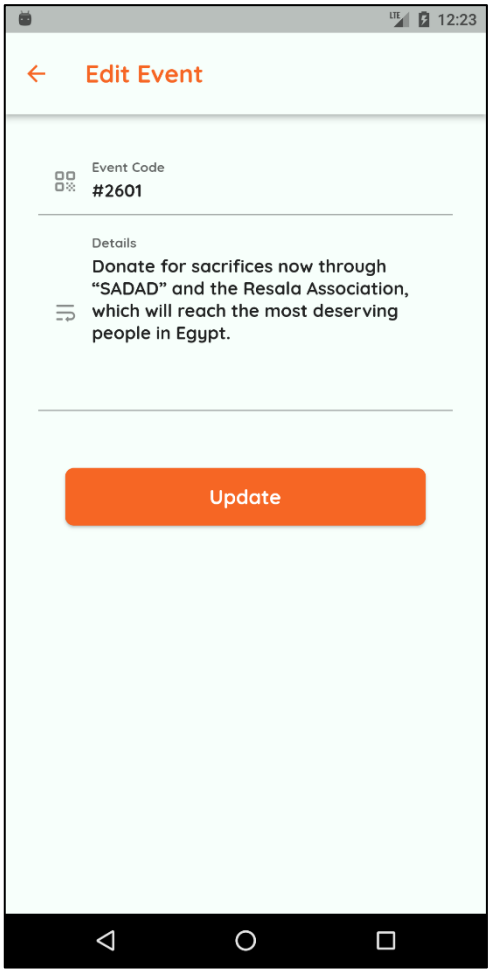


Figure 32: Edit event screen

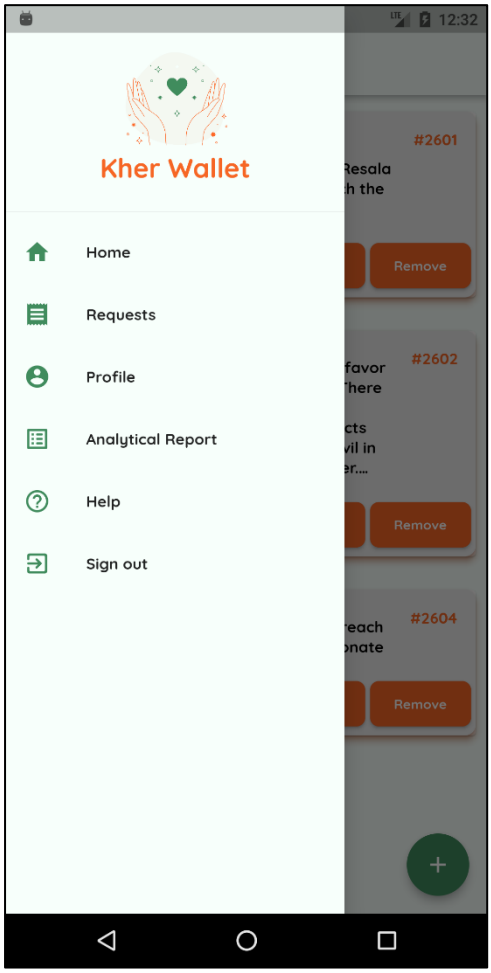


Figure 35: Organization's Navigation Panel

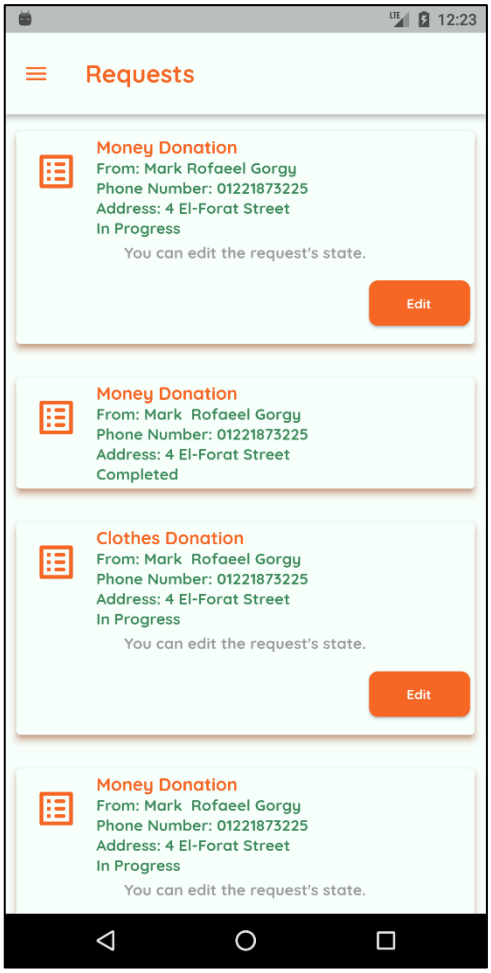


Figure 34: Organization's requests screen

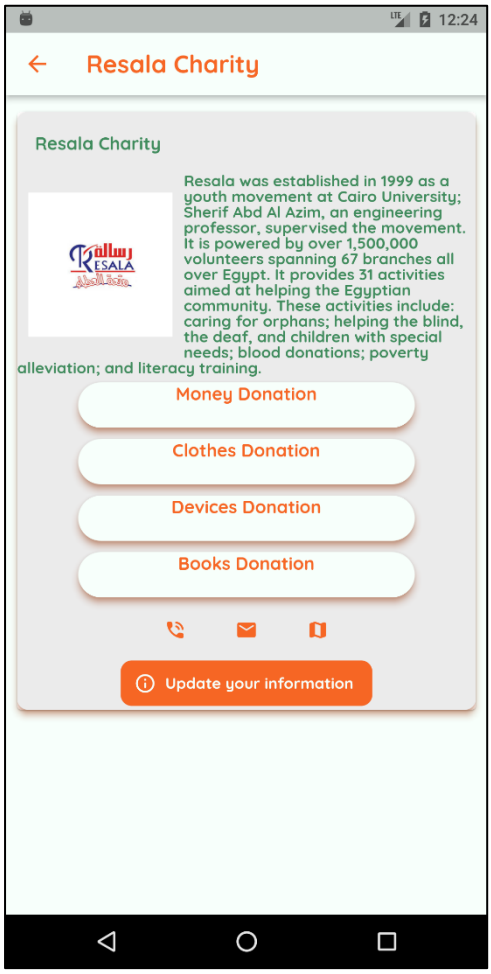


Figure 37: Organization's profile screen

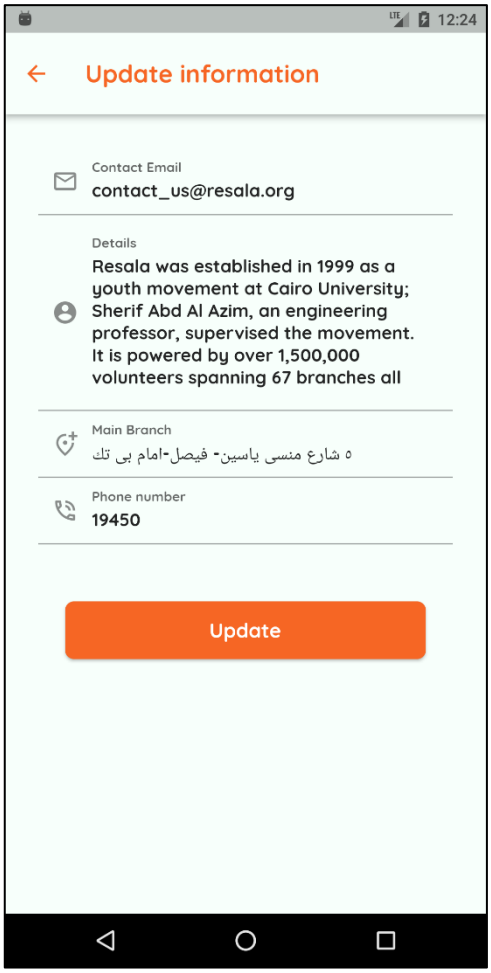


Figure 36: Organization's update profile screen

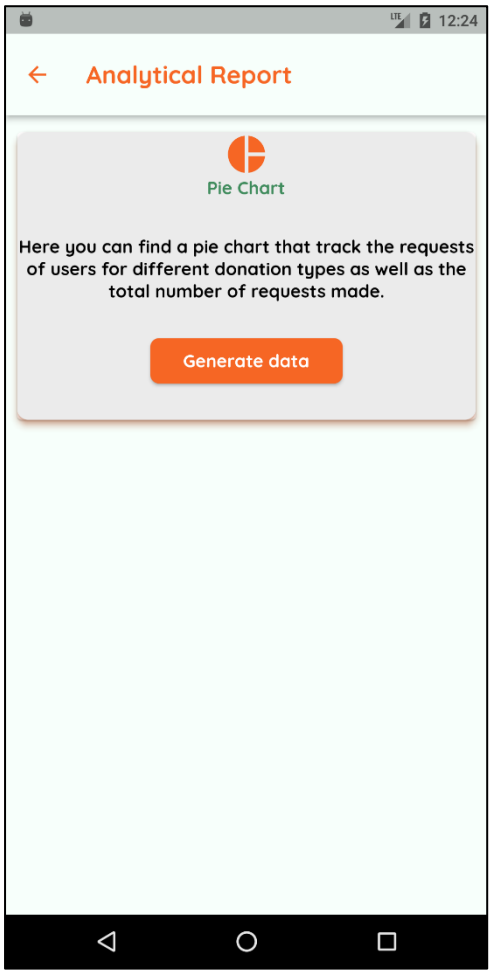


Figure 39: Analytical report screen

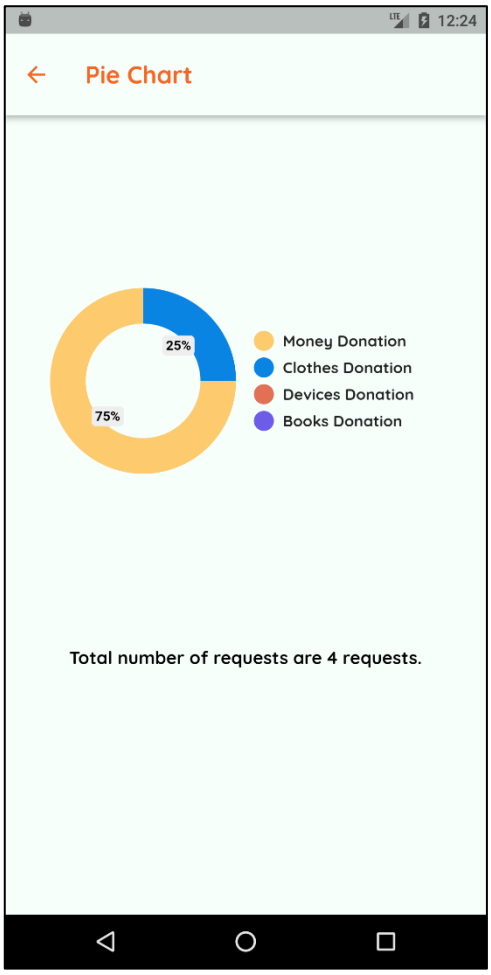


Figure 38: Pie chart screen

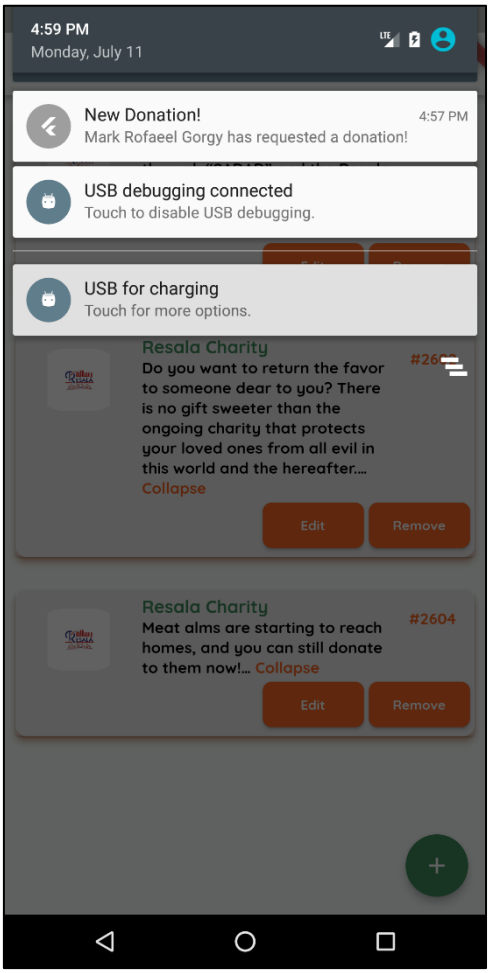


Figure 40: Organization's background notifications

Chapter 5: Implementation and Testing

The testing strategy for the Kher Wallet System is covered in this chapter. To make sure all the modules are operating properly, we will give test scripts for some of our application's features in the following sections. Regarding the integration of the entire system, our testing strategy was to treat the entire system as a functionality and evaluate its performance in various scenarios. We'll go over a few of the tests that were run before we assess the results.

In this section, we'll discuss a few of the test scripts we created for our mobile application to make sure that its modules' core functionality is working as it should.

While designing the test scripts we were keen on making it comprehendible and traceable, so we provided each of the test cases with the following attributes:

• **Test Case ID** along with the Test Case Name for better traceability.

• **Test Scenario** to identify the feature/functionality we are focusing on testing.

• **Test Priority** to identify the severity of the test case

• **Pre-condition** or **Prerequisites** for the test functionality, as it may impact the test result.

• **Test Case Description** to describe the scenario and environment of this test case.

• **Test Data** needed for this test case.

• **Test Steps** that the tester/developer needs to go through to perform this test.

• **Expected Output** that the test result will be evaluate against.

5.1 Donator functionalities test cases:

Donator sign up positive scenario						
Test Scenario	Sign up				Test case ID	T-A1
Test Case	Sign up positive scenario				Test Priority	1
Pre-condition	None					
Test case description	Donator sign up in the application with full name, email, password, address, and phone number.					
Test steps	Test data	Expected output	Actual output	Test Browser	Test Result	Test comment
1. Sign up screen 2. Donator enters personal information	Personal information	Navigate to Home screen	Navigate to Home screen	Mobile android emulator	Pass	-

Table 6: Donator sign up positive scenario

Donator sign up negative scenario						
Test Scenario	Sign up				Test case ID	T-A2
Test Case	Sign up negative scenario				Test Priority	1
Pre-condition	None					
Test case description	Donator sign up in the application with full name, email, password, address, and phone number.					
Test steps	Test data	Expected output	Actual output	Test Browser	Test Result	Test comment
1. Sign up screen 2. Donator enters personal information	Personal information	Display error authentication message	Display error authentication message	Mobile android emulator	Pass	-

Table 7: Donator sign up negative scenario

Update profile positive scenario						
Test Scenario	Update profile				Test case ID	T-A5
Test Case	Update profile positive scenario				Test Priority	2
Pre-condition	Donator signed in successfully and navigate to profile.					
Test case description	Signed-in donator should be able to update any information in profile at any given time such as name, email, address, or phone number.					
Test steps	Test data	Expected output	Actual output	Test Browser	Test Result	Test comment
1. Donator profile screen 2. Donator change any information	Donator's name, email, address, or phone number	Navigate to profile screen	Navigate to profile screen	Mobile android emulator	Pass	-

Table 8: Update profile positive scenario

Update profile negative scenario						
Test Scenario	Update profile				Test case ID	T-A6
Test Case	Update profile negative scenario				Test Priority	2
Pre-condition	Donator signed in successfully and navigate to profile.					
Test case description	Signed-in donator should be able to update any information in profile at any given time such as name, email, address, or phone number.					
Test steps	Test data	Expected output	Actual output	Test Browser	Test Result	Test comment
1. Donator profile screen 2. Donator change any information	Donator's name, email, address, or phone number	Display error validation message	Display error validation message	Mobile android emulator	Pass	-

Table 9: Update profile negative scenario

Donate positive scenario						
Test Scenario	Donate				Test case ID	T-A7
Test Case	Donate positive scenario				Test Priority	1
Pre-condition	Donator signed in successfully and navigate to organization interested in.					
Test case description	Donator should be able to donate by requesting a representative to donate money, clothes, devices, or books.					
Test steps	Test data	Expected output	Actual output	Test Browser	Test Result	Test comment
1. Donator chooses an organization. 2. Donator chooses the donation type.	-	Donation is added to requests screen	Donation is added to requests screen	Mobile android emulator	Pass	-

Table 10: Donate positive scenario

Donate negative scenario						
Test Scenario	Donate				Test case ID	T-A8
Test Case	Donate negative scenario				Test Priority	1
Pre-condition	Donator signed in successfully and navigate to organization interested in.					
Test case description	Donator should be able to donate by requesting a representative to donate money, clothes, devices, or books.					
Test steps	Test data	Expected output	Actual output	Test Browser	Test Result	Test comment
1. Donator chooses an organization. 2. Donator chooses the donation type.	-	Donation is not added to requests screen	Donation is not added to requests screen	Mobile android emulator	Pass	-

Table 11: Donate negative scenario

Remove request positive scenario						
Test Scenario	Remove request				Test case ID	T-A9
Test Case	Remove request positive scenario				Test Priority	2
Pre-condition	Donator signed in successfully and navigate to requests screen.					
Test case description	Donator should be able to remove donations only in the first 24 hours form the action.					
Test steps	Test data	Expected output	Actual output	Test Browser	Test Result	Test comment
Donator navigates to requests screen.	-	Request removed successfully	Request removed successfully	Mobile android emulator	Pass	-

Table 12: Remove request positive scenario

Remove request negative scenario						
Test Scenario	Remove request				Test case ID	T-A10
Test Case	Remove request negative scenario				Test Priority	2
Pre-condition	Donator signed in successfully and navigate to requests screen.					
Test case description	Donator should be able to remove donations only in the first 24 hours form the action.					
Test steps	Test data	Expected output	Actual output	Test Browser	Test Result	Test comment
Donator navigates to requests screen.	-	Request cannot be removed successfully	Request cannot be removed successfully	Mobile android emulator	Pass	24 hours has passed for this request or request state is completed already.

Table 13: Remove request negative scenario

Filter posts/events positive scenario						
Test Scenario	Filter posts/events				Test case ID	T-A11
Test Case	Filter posts/events positive scenario				Test Priority	2
Pre-condition	Donator signed in successfully and navigate to home screen.					
Test case description	Donator should be able to filter the different organizations to view their posts.					
Test steps	Test data	Expected output	Actual output	Test Browser	Test Result	Test comment
Donator navigates to requests screen.	Specific organization name	Posts/events filtered by organization name	Posts/events filtered by organization name	Mobile android emulator	Pass	-

Table 14: Filter posts/events positive scenario

Filter posts/events negative scenario						
Test Scenario	Filter posts/events				Test case ID	T-A12
Test Case	Filter posts/events negative scenario				Test Priority	2
Pre-condition	Donator signed in successfully and navigate to home screen.					
Test case description	Donator should be able to filter the different organizations to view their posts.					
Test steps	Test data	Expected output	Actual output	Test Browser	Test Result	Test comment
Donator navigates to requests screen.	Specific organization name	Posts/events filtered by organization name	Posts/events are not filtered by organization name	Mobile android emulator	Fail	-

Table 15: Filter posts/events negative scenario

5.2 Organization functionalities test cases:

Organization sign in positive scenario						
Test Scenario	Sign in				Test case ID	T-B1
Test Case	Sign in positive scenario				Test Priority	1
Pre-condition	None					
Test case description	Organization should be able to sign into the application by their email and password (provided by administrators).					
Test steps	Test data	Expected output	Actual output	Test Browser	Test Result	Test comment
1. Organization sign in screen 2. Organization enters email and password.	Email and password	Navigate to organization home screen	Navigate to organization home screen	Mobile android emulator	Pass	-

Table 16: Organization sign in positive scenario

Organization sign in negative scenario						
Test Scenario	Sign in				Test case ID	T-B2
Test Case	Sign in negative scenario				Test Priority	1
Pre-condition	None					
Test case description	Organization should be able to sign into the application by their email and password (provided by administrators).					
Test steps	Test data	Expected output	Actual output	Test Browser	Test Result	Test comment
1. Organization sign in screen 2. Organization enters email and password.	Email and password	Display error authentication message	Display error authentication message	Mobile android emulator	Pass	-

Table 17: Organization sign in negative scenario

Update profile positive scenario						
Test Scenario	Update profile				Test case ID	T-B3
Test Case	Update profile positive scenario				Test Priority	2
Pre-condition	Organization signed in successfully and navigate to profile.					
Test case description	Signed-in organization should be able to update any information in profile at any given time such as contact email, details or main branch and phone number.					
Test steps	Test data	Expected output	Actual output	Test Browser	Test Result	Test comment
1. Organization profile screen 2. Organization changes any information	Organization's contact email, details, main branch, or phone number	Navigate to profile screen	Navigate to profile screen	Mobile android emulator	Pass	-

Table 18: Update profile positive scenario

Update profile negative scenario						
Test Scenario	Update profile				Test case ID	T-B4
Test Case	Update profile negative scenario				Test Priority	2
Pre-condition	Organization signed in successfully and navigate to profile.					
Test case description	Signed-in organization should be able to update any information in profile at any given time such as contact email, details or main branch and phone number.					
Test steps	Test data	Expected output	Actual output	Test Browser	Test Result	Test comment
1. Organization profile screen 2. Organization changes any information	Organization's contact email, details, main branch, or phone number	Display error validation message	Display error validation message	Mobile android emulator	Pass	-

Table 19: Update profile negative scenario

Add post positive scenario						
Test Scenario	Add post				Test case ID	T-B5
Test Case	Add post positive scenario				Test Priority	1
Pre-condition	Organization signed in successfully, navigate to home and click on '+' button.					
Test case description	Organization should be able to add any upcoming post where help may be needed which is then send as a notification to donators.					
Test steps	Test data	Expected output	Actual output	Test Browser	Test Result	Test comment
Organization Home screen and click '+' button	Organization's post code and description.	Navigate to Home screen and the added post appears.	Navigate to Home screen and the added post appears.	Mobile android emulator	Pass	-

Table 20: Add post positive scenario

Add post negative scenario						
Test Scenario	Add post				Test case ID	T-B6
Test Case	Add post negative scenario				Test Priority	1
Pre-condition	Organization signed in successfully, navigate to home and click on '+' button.					
Test case description	Organization should be able to add any upcoming post where help may be needed which is then send as a notification to donators.					
Test steps	Test data	Expected output	Actual output	Test Browser	Test Result	Test comment
Organization Home screen and click '+' button	Organization's post code and description.	Navigate to Home screen and the added post does not appear.	Navigate to Home screen and the added post does not appear.	Mobile android emulator	Pass	Connection issues from client side.

Table 21: Add post negative scenario

Update post positive scenario						
Test Scenario	Update post				Test case ID	T-B7
Test Case	Update post positive scenario				Test Priority	1
Pre-condition	Organization signed in successfully and navigate to home and click on specific post's update button.					
Test case description	Organization should be able to update existing post such as event code or details.					
Test steps	Test data	Expected output	Actual output	Test Browser	Test Result	Test comment
Organization Home screen and click update button of specific post	Organization's post code and description.	Navigate to Home screen and the post is updated.	Navigate to Home screen and the post is updated.	Mobile android emulator	Pass	-

Table 22: Update post positive scenario

Update post negative scenario						
Test Scenario	Update post				Test case ID	T-B8
Test Case	Update post negative scenario				Test Priority	1
Pre-condition	Organization signed in successfully and navigate to home and click on specific post's update button.					
Test case description	Organization should be able to update existing post such as event code or details.					
Test steps	Test data	Expected output	Actual output	Test Browser	Test Result	Test comment
Organization Home screen and click update button of specific post	Organization's post code and description.	Navigate to Home screen and the post is not updated.	Navigate to Home screen and the post is not updated.	Mobile android emulator	Pass	Connection issues from client side.

Table 23: Update post negative scenario

Remove post positive scenario						
Test Scenario	Remove post				Test case ID	T-B9
Test Case	Remove post positive scenario				Test Priority	1
Pre-condition	Organization signed in successfully and navigate to home and click on specific post's remove button.					
Test case description	Organization should be able to remove existing post.					
Test steps	Test data	Expected output	Actual output	Test Browser	Test Result	Test comment
Organization Home screen and click remove button of specific post	-	Navigate to Home screen and the post is removed	Navigate to Home screen and the post is removed	Mobile android emulator	Pass	-

Table 24: Remove post positive scenario

Remove post negative scenario						
Test Scenario	Remove post				Test case ID	T-B10
Test Case	Remove post negative scenario				Test Priority	1
Pre-condition	Organization signed in successfully and navigate to home and click on specific post's remove button.					
Test case description	Organization should be able to remove existing post.					
Test steps	Test data	Expected output	Actual output	Test Browser	Test Result	Test comment
Organization Home screen and click remove button of specific post	-	Navigate to Home screen and the post is not removed	Navigate to Home screen and the post is not removed	Mobile android emulator	Pass	Connection issues from client side.

Table 25: Remove post negative scenario

Edit requests positive scenario						
Test Scenario	Edit requests				Test case ID	T-B11
Test Case	Edit requests positive scenario				Test Priority	2
Pre-condition	Organization signed in successfully and navigate to requests and click on specific request's edit button.					
Test case description	Organization should be able to edit any request's current state (in progress/ completed).					
Test steps	Test data	Expected output	Actual output	Test Browser	Test Result	Test comment
Organization Requests screen and click edit button of specific request	Request's state (in progress/ completed)	Navigate to Request screen and the request is updated.	Navigate to Request screen and the request is updated.	Mobile android emulator	Pass	-

Table 26: Edit requests positive scenario

Edit requests negative scenario						
Test Scenario	Edit requests				Test case ID	T-B12
Test Case	Edit requests negative scenario				Test Priority	2
Pre-condition	Organization signed in successfully and navigate to requests and click on specific request's edit button.					
Test case description	Organization should be able to edit any request's current state (in progress/ completed).					
Test steps	Test data	Expected output	Actual output	Test Browser	Test Result	Test comment
Organization Requests screen and click edit button of specific request	Request's state (in progress/ completed)	Request cannot be edited.	Request cannot be edited.	Mobile android emulator	Pass	Request state is completed already.

Table 27: Edit requests negative scenario

5.3 Donator/organization functionalities test cases:

Write feedback positive scenario						
Test Scenario	Write feedback				Test case ID	T-C1
Test Case	Write feedback positive scenario				Test Priority	3
Pre-condition	Donator/organization signed in successfully and navigate to help screen and write feedback button.					
Test case description	Donator/organization should be able to write the feedback of his own.					
Test steps	Test data	Expected output	Actual output	Test Browser	Test Result	Test comment
Donator/organization navigates to help screen and clicks feedback button.	Donator/organization email and written feedback	Kher Wallet receives user's feedback	Kher Wallet receives user's feedback	Mobile android emulator	Pass	-

Table 28: Write feedback positive scenario

Write feedback negative scenario						
Test Scenario	Write feedback				Test case ID	T-C2
Test Case	Write feedback negative scenario				Test Priority	3
Pre-condition	Donator/organization signed in successfully and navigate to help screen and write feedback button.					
Test case description	Donator/organization should be able to write the feedback of his own.					
Test steps	Test data	Expected output	Actual output	Test Browser	Test Result	Test comment
Donator/organization navigates to help screen and clicks feedback button.	Donator/organization email and written feedback	Kher Wallet does not receive user's feedback	Kher Wallet does not receive user's feedback	Mobile android emulator	Pass	Connection issues from client side.

Table 29: Write feedback negative scenario

Forget/change password positive scenario						
Test Scenario	Forget/change password				Test case ID	T-C3
Test Case	Forget/change password positive scenario				Test Priority	1
Pre-condition	Donator/organization click on forget password button before sign in.					
	Donator/organization signed in successfully and navigate to profile and click on change password button.					
Test case description	Donator/organization should be able to forget password during the sign in process. Donator/organization should be able to change password from their profile. An email will be sent where password can be easily reset.					
Test steps	Test data	Expected output	Actual output	Test Browser	Test Result	Test comment
Donator/organization click on either forget password or change password button	-	Email is sent to email entered previously.	Email is sent to email entered previously.	Mobile android emulator	Pass	-

Table 30: Forget/change password positive scenario

Forget/change password negative scenario						
Test Scenario	Forget/change password				Test case ID	T-C3
Test Case	Forget/change password negative scenario				Test Priority	1
Pre-condition	Donator/organization click on forget password button before sign in.					
	Donator/organization signed in successfully and navigate to profile and click on change password button.					
Test case description	Donator/organization should be able to forget password during the sign in process. Donator/organization should be able to change password from their profile. An email will be sent where password can be easily reset.					
Test steps	Test data	Expected output	Actual output	Test Browser	Test Result	Test comment
Donator/organization click on either forget password or change password button	-	Email is not sent to email entered previously.	Email is not sent to email entered previously.	Mobile android emulator	Pass	Connection issues from client-side.

Table 31: Forget/change password negative scenario

5.4 System Black Box Testing

Black box testing measures the performance and quality of our application system as a donating application with different test cases, we are mostly concerned about this in testing:

- The donator must be authenticated by Google Firebase Authentication technology.
- The request must be specified with the specific organization.
- The notification pushed to the specific organization.

Test case scenario:

This is a simple test case in which:

- A donator registers to the system with valid information.
- Google Firebase Authentication technology validate this information, and the donator is now registered successfully to the application.
- The donator is able to choose the preferred organization by going through the different places categories.
- The donator, now, can donate money, clothes, books, or devices by requesting a representative to come to his doorsteps to complete the donation process.
- This request is saved to the database and is retrieved by the specific organization anytime.
- A notification is sent to the organization specifying the donator's name, address, phone number and the donation type made.
- The organization can view the request made by this donator in the request section. And the organization can, now, send the representative to the donator peacefully.

Input Details:

- Donator registration information.
- Donator authentication.
- Request information to the organization.

Output Details:

- Request is being processed by the system.
- Notification is pushed to the specific organization.
- Donator and organization can reach each other.

Conclusion:

To conclude what we have made, Kher Wallet's major goal is to bridge the gap between various groups of people and organizations. Our application that compiles several organizations that users can utilize to simply finish their donation procedure at their own fingertips.

We are so proud of what we have achieved so far, we hope one day that this application will be used by a lot of people through these digital evolution days! Throughout the next period, we are looking forward to the next milestone of developing and evolving this application.

Poster:

Cairo University, Faculty of Computers and Artificial intelligence

Kher Wallet

KHER WALLET
Mobile App

Nada Mohamed, Sarah Khaled, Mark Rofaeel,
Ayat Hany and Yasmine Shehab
Supervisor: Dr. Khaled Wassif

Abstract

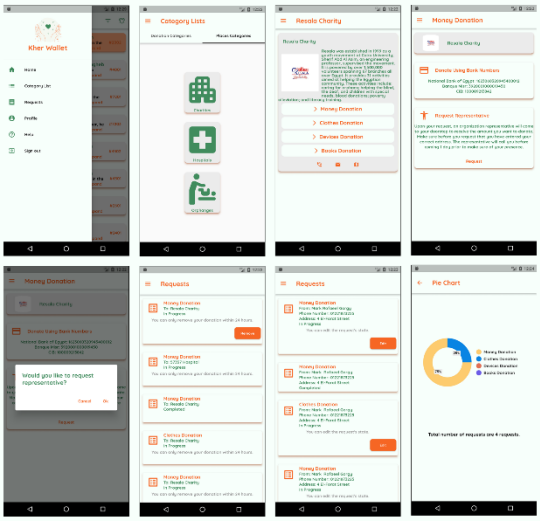
Nowadays, there are a lot of organizations that accept donations from different people, either by donating money, clothes, devices, books or even by volunteering in many events.
It takes time and effort for people to reach to these organizations so they can donate. Also, people do not even know when any organization need help during event times.
As well as the organizations! Organizations do not have an all-in-one platform to communicate on with people interested.
Considering this, our motivation is to solve these problems by implementing an Android/IOS application that gathers different kinds of organizations, that help people to donate in an easier way and notify them when there is a new post or event around the corner.
Kher Wallet may be the first donating platform in the Egyptian region.

Introduction

Our problem is that people spend so much time and effort to reach different organizations as charities, hospitals, and orphanage that has led to the shortage of the overall amount of donations.
Another key fact to remember is that those organizations do not have the capability to reach large amount of people to notify them with new posts or events.
The main objective of our application is filling the gap between different organizations and people by implementing Kher Wallet. An application that gathers different types of organizations which can be easily used by people to complete their donation process right at their own hands.

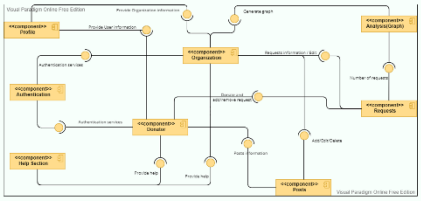
Methods

Throughout the process of doing this project, we followed the **Waterfall methodology** and principles and the basic **client-server architecture**.



Primarily Design

The donator can choose the preferred organization by going through the different places categories. The donator, now, can donate money, clothes, books, or devices by requesting a representative to come to his doorsteps to complete the donation process. This request is saved to the database and is retrieved by the specific organization anytime. A notification is sent to the organization specifying the donator's name, address, phone number and the donation type made. The organization can view the request made by this donator in the request section. And the organization can, now, send the representative to the donator peacefully. A pie chart is generated with the total number of requests and the percentage of every donation type.



The component diagram shows how components are wired together to form larger components or software systems.

Conclusion

To conclude what we have made, Kher Wallet's major goal is to bridge the gap between various groups of people and organizations. Our application that compiles several organizations that users can utilize to simply finish their donation procedure at their own fingertips. We are so proud of what we have achieved so far, we hope one day that this application will be used by a lot of people through these digital evolution days! Throughout the next period, we are looking forward to the next milestone of developing and evolving this application.

Contact us via email: **kherwallet@gmail.com**

References:

1. Waffarha Application:

https://waffarha.com/en

2. Flutter Documentation:

https://docs.flutter.dev/

3. Google Firebase Documentation:

https://firebase.google.com/docs

4. External packages:

- Firebase authentication: https://pub.dev/packages/firebase_auth
- Firebase messaging: https://pub.dev/packages/firebase_messaging
- Firebase analytics: https://pub.dev/packages/firebase_analytics
- Cloud firestore: https://pub.dev/packages/cloud_firestore
- Firebase core: https://pub.dev/packages/firebase_core
- Flutter local notifications: https://pub.dev/packages/flutter_local_notifications