Social Network - Hadar

High Level Design

Submitted by:

Hussein Abu Jabal

Haneen Jeries

Haitham Kablan

Sami Hamud

Supervised by:

Eytan Singher

Table of Contents

Abstract

As time passes, notable demographic changes in cities are observed while families and individuals of close socioeconomic levels are condensed into the same neighborhood level. As life in high socioeconomic neighborhoods is stable, living in low socioeconomic neighborhoods become harsher. In particular, Hadar is a decent example of such neighborhoods.

Hadar is a relatively big neighborhood in Haifa, Israel. Located on the northern slope of Mount Carmel between the upper and lower city overlooking the Port of Haifa and Haifa Bay.
Hadar was once the commercial center of Haifa. But recently, the situation of the neighborhood has deteriorated due to wars, immigration and poverty.
Consequently, Hadar became home to high rates of helpless people, such as elders, single-parent families who need help with their kids, and other people who struggle to afford basic needs.

The recent advancement of technology and social networking pave the way to develop an app that could reduce the life difficulties in low socioeconomic neighborhoods.
This project proposes a communication app which poses a robust platform to support destitute and impoverished people. The main purpose of the application is to connect these people with relevant volunteers or assisting organizations. This is accomplished by matching the two sides according to a certain criterion. The entire process is supervised by an administrator.

All of this could serve as a potent environment to emphasize the necessity of the proposed application. Without loss of generality, the app holds also for other helpless neighborhoods in the world.

1. Introduction

The purpose of the proposed application is to match between users in need and volunteers that are willing to help them, which is accomplished in the following way:

- Users in need can add a help request, which appears on the feed of the relevant volunteers.
- Helpers view their feed and offer their help for relevant requests, so the user in need can contact them if he is interested.

In addition to all this, there is an admin for the application that his job is to supervise the system, such as verifying new users, to avoid complications.

1.1 General Project Description

This project presents a communication app that wisely connects helpless people in a community with volunteers in the same community. The app mainly serves elder, impoverished, disabled people and single-parent families to obtain their needs. The types of needs are classified into categories while helpless people could find their matched volunteers that aim to assist in the same category. A feed interface is developed to show the requests for the volunteer while each request is accompanied by a message between the two communicating people.

The overall activity is supervised by an administrator user, who is capable of adding new users and removing irrelevant ones. Helpless users are exposed to all categories, while volunteers are associated with their relevant assisting categories by the administrator user. Services of various assisting organizations could be integrated and delivered via the proposed app while the latter will serve as an advertisement platform for the organization services. The proposed app could be hosted by ios, android and web systems.

1.2 Programming Environment

We are going to implement the application using the cross-platform program flutter, which will make our application available for Android, IOS and Web users.
We will be running flutter using Android Studio IDE and will use Firebase as a cloud database.
We will write our code using Dart programming language mainly.

2. Theoretical Background

Hadar is a district of Haifa, located on the northern slope of Mount Carmel between the upper and lower city overlooking the Port of Haifa and Haifa bay [**Error! Reference source not found.**1].

Hadar has roughly around 40000 residents. It is also known as the most crowded district in Haifa by 14000 inhabitants per square kilometer. It is in flux with a large percentage of new immigrants from the former Soviet Union [3].

Up till the ’60s, it was the commercial center of Haifa. Just then, its residents started growing financially, which in result caused them to leave the neighborhood, to live in another that matches their new high socioeconomic status [2].

Therefore, the condition of the neighborhood was deteriorated, especially during the 2006 Lebanon War, when the neighborhood was hit by several rockets [3].

Today, Hadar is the home of a lot of elders, single mothers who need help with their kids, and other people who struggle to afford basic needs. therefore, a lot of organizations were created recently to fix the situation. One of the well-known active organizations in the hood is Lev-Hash; it offers many humanity services such as affordable items, dental services, distributing fresh food.

3. Basic System Functionalities

The application provides a variety of functionalities for different types of users.

3.1 Users

First, let us define the types of users:

1- **Admin:** The administrator of the system, has the authority to add new users to the application and make judgments upon all the events that take place in the system.
2- **User In-need:** A user that can request help.
3- **Volunteer:** A user who wants to offer his help to a **User In-need**.
4- **Unregistered User:** A guest who has not been registered in the system yet.
5- **Organization:** Basically, considered a **Volunteer**.
6- **Registered User:** Any user of the above types who is not an **Unregistered User.**

3.2 Functionality

We will now enlist the basic functionalities and features of the application, according to user types:

Unregistered User:

- An **Unregistered User** who wants to sign up for the application must send a registration request in the Main Menu screen and wait to be verified by the **Admin**.
- An **Unregistered user** can request to join as a **User In-need** or as a **Volunteer**.

Registered User:

- A **Registered User** can sign into the application using the Main Menu.
- A **Registered User** has an option to stay signed in the application even when he is not using it, by specifying this in his profile.
- A **Registered User** can view and edit the details of his profile. The profile button is found in the lower bar of the application.
- A **Registered User** can log out of the application by pressing the Log-Out button in his profile.
- A **Registered User** can mute or unmute the notifications by specifying this in his profile.

Admin:

- An **Admin** gets notified when an **Unregistered User** asks to register in the application.
- An **Admin** can authorize an **Unregistered User's** registration according to a criterion that he decides, or otherwise, reject his request.
- An **Admin** can view a feed, which includes joining requests which have not yet been responded to, and can confirm or deny each request separately.
- An **Admin** can view a set of the **Registered Users** in the system or a filtered subset of it.
- An **Admin** can search for a specific **Registered User** in the system and view his activity.

User In-need:

- A **User In-need** can add a request for specific help from a pre-defined list of needs, with an option to specify an unlisted need, and can write a short description of what he needs.
- A **User In-need** gets notified when a **Volunteer** declares that he wants to help him. If he wants to accept his help, he should press the accept button to connect with the **Volunteer**.
- A **User In-need** can view a feed which includes his previous and current help requests with information about each request, such as if someone has offered to help him or if the need has been satisfied.
- A **User In-need** can request help by renewing a previous help request in the help-request feed.
- A **User In-need** can communicate with a **Volunteer**.

Volunteer:

- A **Volunteer** gets notified when a **User In-need** requests help which he is authorized to volunteer in. Then he can let the **User In-need** know of his intent to help him by notifying him.
- A **Volunteer** can view a feed that includes help requests that he is authorized to help with, which has not been satisfied yet, and offer his help to one of them by notifying the **User In-need** who requested it.
- A **Volunteer** can communicate with the **User In-need** whom he wants to help to ask him for details.

4. Software Implementation

In this section, we will describe the main implementation details of the application.

4.1 Modules

Registration:
The Registration module is responsible for all the user authentication activities in the system, which includes:

- Sign-up request
- Sign in
- Log out

Help requests:
This module manages the matching process conclusively, it provides the following functionalities:

- Adding help request

- Offering help
- Feed system

Administration:
The administration module is crucial for ensuring the app security, managing the users, and monitoring their activity.

- Verifying new users
- Classifying users
- Monitoring the system
- Adding new help request types

Database:
The Database module defines the data collections that we will use in the application. We will manage 4 database collections in our implementation:

1. Users collection:
   This collection will save the registered users in the system.
2. Help requests collection:
   This collection will hold the help requests posted by all the Users In-need.
3. Registration requests collection:
   This collection will hold the unresolved registration requests.
4. Help requests types collection:
   This collection will hold the types of help requests that can be provided. This will allow us to add new types of help requests dynamically without having to release a new app version.

4.1.1 Communication between modules:

Registration -> Administration:

- When a new user is trying to sign up, the admin receives a notification about it and decides whether to accept the request or reject it.

Administration -> Registration

- If the admin decides to accept the request, he then assigns the user type to **Volunteer** or **User In-need** and then notifies the user back about his acceptance.

Registration ↔ Database:

- The **sign-up request** will append the new user data to the **Registration requests collection** in the Database module.
- The **sign in** and **Log out** functionalities will access the **Users collection** to authenticate the user.

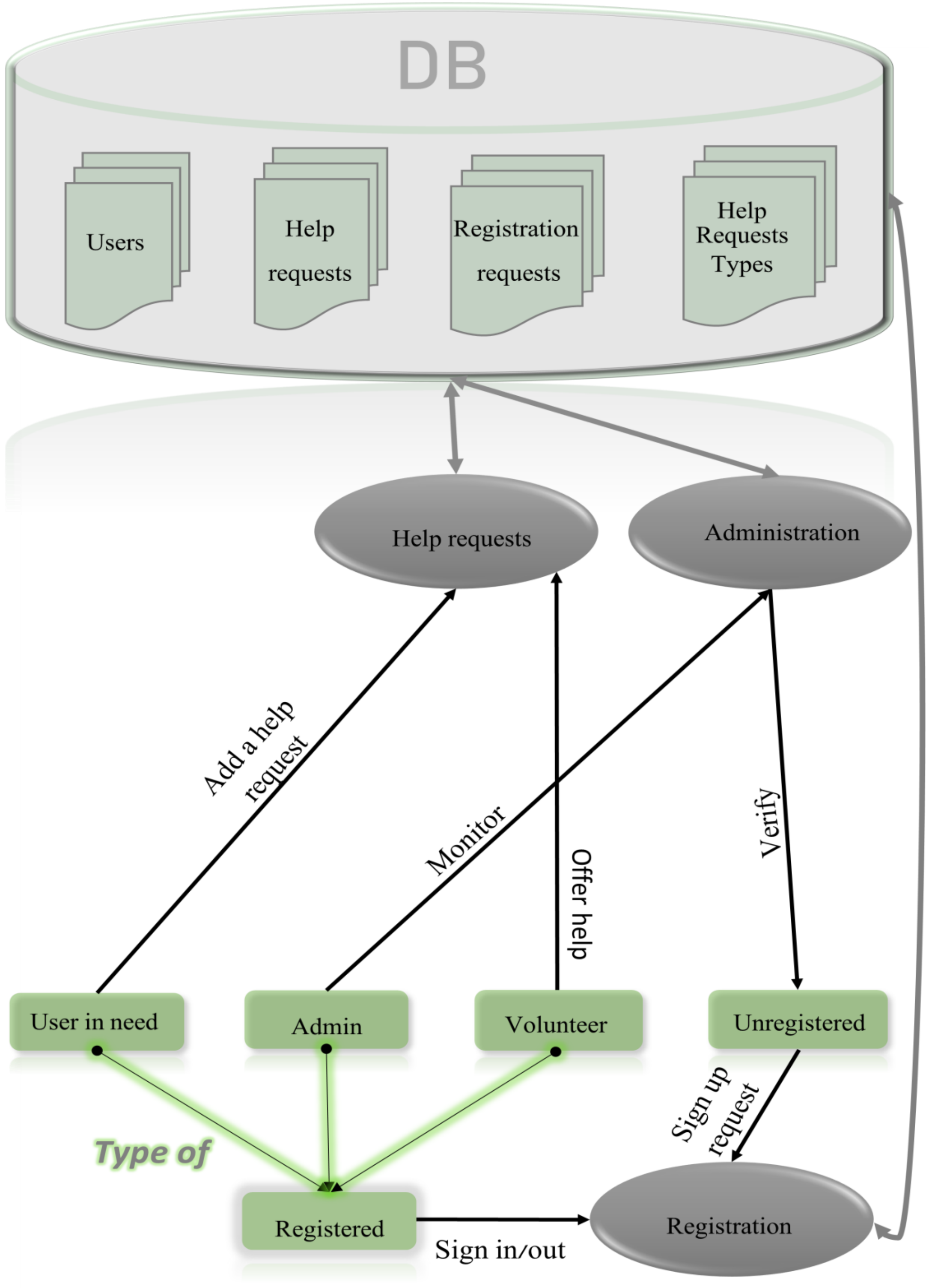
Help Requests ↔ Database:

- The **Adding help request** functionality will append the new help request to the **Help requests collection**.
- The **Feed system** will access the **Help requests collection** to fetch the relevant helping requests.
- The **Offering help** functionality will edit a specific help request in the **Help requests collection** to specify that the current Volunteer is willing to help.

Administration ↔ Database:

- The **Verifying new users** functionality fetches new registration requests from the **Registration requests collection.**
- The **Classifying users** feature accesses the **Users collection** to filter users by a specific criterion.
- **Monitoring the system** function accesses the **Users collection** and **Help requests collection** to present user’s data and activities.
- **Adding new help request types** functionality appends a new type to the **Help requests types collection**.

4.2 Top-Level View



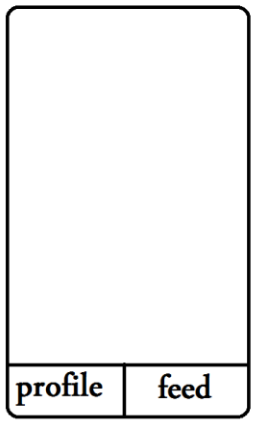
4.3 Main Menu

4.3.1 User Interface

Each type of user has a unique Main Menu user interface. We will present them here.

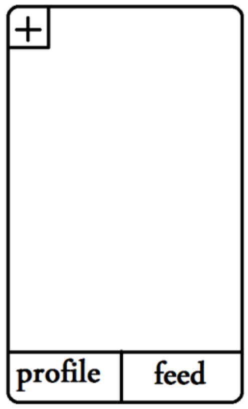
Volunteer Main Menu:

A volunteer has main 2 button; one that allow him to view his profile, and another one that allows him to view the help-request feed that is relevant to him. Here is a sample of the user interface:



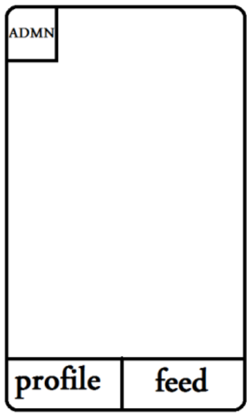
User In-need Main Menu:

A User In-need has main 3 button; one that allow him to view his profile, and another one that allows him to view a feed of his previous and current help requests and the last button allows him to add a new help request. Here is a sample of the user interface:



Admin Main Menu:

A User In-need has main 3 button; one that allow him to view his profile, and another one that allows him to view a feed of unresolved verification requests and the last button allows him to view the admin panel which allows him to monitor the system. Here is a sample of the user interface:



4.3.2 Features.

- Display feed:
  This feature differs from a type user to another, as described above.
- Add help request:
  This feature is available only for User In-need, it opens a new view which allows him to add a new request.
- Display profile
  This feature is available for every user, it switches to the user profile.
- Admin Panel
  This feature is available only for the admin, it allows him to monitor and control the system including the users.

4.4 Home page

The Home page has two possible views, showing each of them depends on the status of the user.

- If the user is not registered yet or is not signed in, then the Home page shows a view that includes two buttons; sign up and sign in.
- Otherwise, if the user is already signed in, then he will be transferred to the Main Menu view.

5. References

1. https://en.wikipedia.org/wiki/Hadar_HaCarmel
2. https://haipo.co.il/item/60005
3. https://www.hamichlol.org.il/%D7%94%D7%93%D7%A8_%D7%94%D7%9B%D7%A8%D7%9E%D7%9C