

High-Level Solution Design Document

Project Happy Heart for Client Eliyahu

Updated: 23/11/2022

Authors: Alon Kitin, Amir Alkalay, Elad Badikhi, Ido Magner, Linoy Keren, Matan Beigel, Tomer Katz, Uriya Hubara

Introduction

Happy heart is a charity foundation founded by Eliyahou Eliyovich, that aims to supply food, furniture, medical equipment and much more to people in need.

During COVID-19 pandemic, many people got sick and don't had the resources to take care of themselves, for example – in a bad scenario, COVID-19 hurt the breathing system and people needed to get oxygen concentrator as soon as possible, but they couldn't go to the hospital to get treatment!

In order to help those people, Eliyahou founded Happy Heart and currently help people in need all around the country.

Eliyahou came to us because he need a tool that will allow the foundation to organize all of the equipment and volunteers they currently have so they could keep helping people in more efficient way.

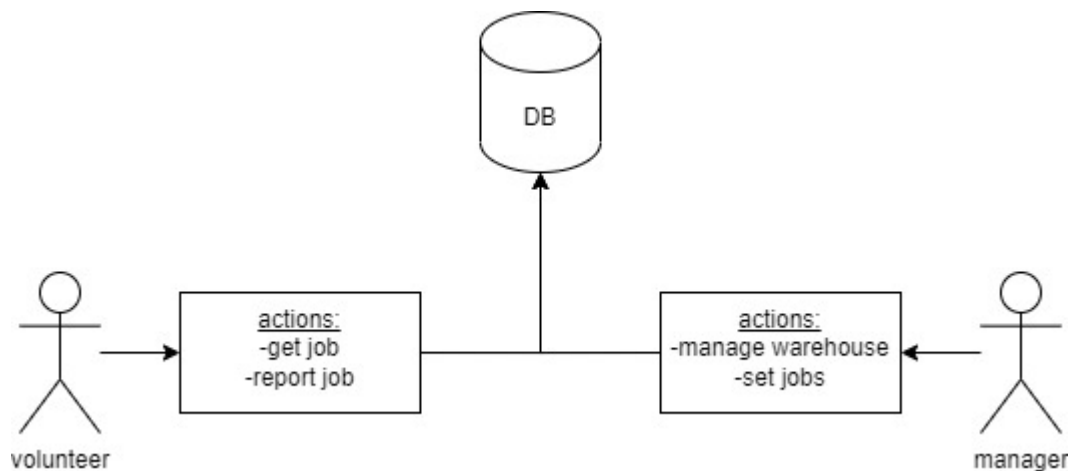
Requirement Summary

- A Database containing all of the medical equipment and volunteers' data.
- A mobile app for the members of the foundation that manage the equipment.
- A mobile app for the volunteers, that will allow them to see the needed transfers of equipment and fulfill them.

Assumptions and Prerequisites

After talking to Eliyahou, we understood that the managers and members of the foundation are preferring to use mobile devices, so we assume that all of the members and volunteers have a mobile device running either IOS or Android operating systems.

UML resource diagram



As shown in the diagram above, our system will include a database, which will contain information about all of the volunteers.

The database will also contain everything the member needs to know about the equipment the organization owns – number of each type, availability, current location, estimated returning date and more.

Furthermore, our system will include a mobile app for the management and the volunteers. For the management side, the app will allow them to manage the warehouse, the equipment in it and the volunteers currently associated with the foundation.

Application Modules

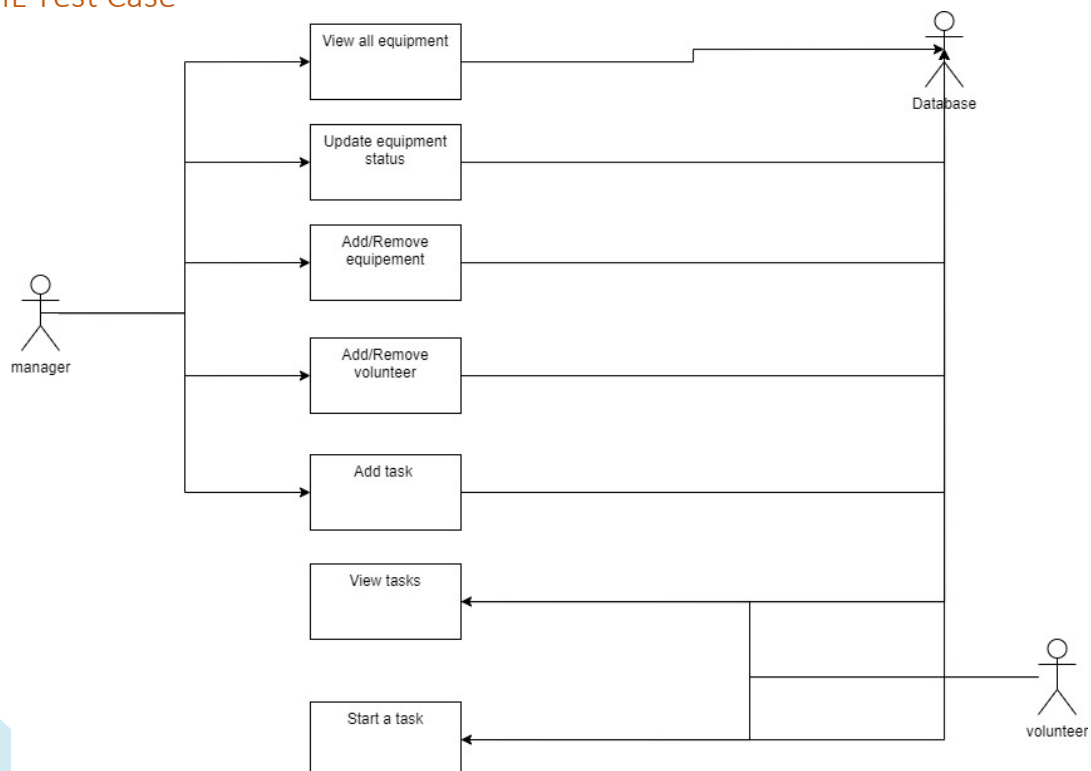
- Core components- DataBase, Application.
- Frameworks–
 - Firebase – we chose Firebase as our DB because it has a free use plan suitable with the project's scale. Moreover, Firebase has a very easy to use abstraction, and packages that we can use in Flutter SDK.
 - Flutter – We chose to use Flutter because most of our team has some experience in developing in Flutter, and because Flutter is a widely used SDK, the other part of the team could find a lot of tutorial videos to help them catch-up.

User Stories

After talking to Eliyahou, we gathered the following user stories:

- As a manager I want to be able to view the history of all of the missions.
- As a manager I want to be able to add a new volunteer.
- As a user I want to be able to login from IOS and android platforms.
- As a user I want to be able to add a new equipment to the system.
- As a user I want to be able to view the data on our supply.
- As a user I want to be able to mark a certain machine as loaned.
- As a user I want to be able to view information about the current loan.
- As a user I want to be able to view the history of loans of a certain machine.
- As a user I want to be able to mark a loaned machine as returned.
- As a user I want to be able to mark the machine as broken and unavailable.
- As a user I want to be able to set a time limit on a loan of a certain.
- As a user I want to be able to add a mission for the volunteers.
- As a user I want to be able to export my database to csv.
- As a user I want to be able to set a reminder to call a person after a set amount of time from the loan start – optional.
- As a volunteer I want to be able to see all of the open missions.
- As a volunteer I want to volunteer for an open mission.
- As a volunteer I want to be able to mark a mission as complete.
- As a volunteer I want to get virtual points for my missions.

UML Test Case



TDD

In this project we would like to stick to the test-driven development style, the main tests we want to our app to pass are:

- Constant availability of warehouse storage.
- Consistent list of volunteer tasks.
- System updates should always be available to all users.