**Braude**
College of Engineering, karmiel

**Capstone Project Phase A**

# <u>Dolt</u>

**An application for providing and consuming services with recommendation engine**

Project Code: 23-1-D-4

**Supervisor**

Mr. Alex Keselman

**alex@vir-tec.net**

Shmuel Shaul - 316256270

Shmuel.shaul@e.braude.ac.il

Aneel Amar - 322478892

aneel.ammar@e.braude.ac.il

# Abstract

Personal time optimization and the creation of additional income streams are two main aspects in the life of the average person and sometimes also contradict each other. In this project we suggest 'DoIt'. Doit is a platform for providing and consuming services based on user preferences using a recommendation engine. By using the application, users can make requests for services as much as they want, within the law, and offer a price in exchange for it. Other users can supply the requested services and earn money. The platform will also include a recommendation engine to match services to users based on their preferences. The project's goal is to create a user-friendly and efficient platform that will save time and money for both service providers and consumers. The model is similar to apps such as 'Uber' or 'Wolt'. In the following document we examine in depth the challenge raised by the problem and the existing solutions, we also present the engineering process that was carried out for the development of the product. What are the engineering challenges that arose throughout the process and how we overcome them. We present the research that was done to gain insight into which features are required to provide the best solution to the problem. The technological and logistical challenges that arose during the research are also described.

# Table of Contents

# 1. <u>Introduction:</u>

## 1.1 Motivation:

- Do you consider yourself a busy person? Who does not?
  How many times have you tried to get everything done even though you didn't have time? Take the dog for a walk, pick up a package from the P.O, buy a cake for the child's birthday in kindergarten, and all of this in the same morning.
  We have a solution that will save you a lot of time.
  Are you looking for additional income to help you pay the rent in Tel Aviv?
  Do you spend some time in the P.O to pick up your packages from Amazon or AliExpress? So why not earn some easy money? Our app will help you with that.

## 1.2 Our solution:

- Following the needs mentioned above we developed 'DoIt'.
  DoIt is a platform for providing and consuming services based on user preferences using a recommendation engine that we developed.
- DoIt has 2 types of users (Stakeholders):
  - The user who requested the service.
  - The user who provides the service.
- The app allows users to post any kind of service that they want someone to provide them, with the price they would like to pay for it (open for negotiation). For example: Bob needs to pick up a delivery from the P.O, but currently he is at work and the P.O closes before he finishes his work. Bob can post a request on 'DoIt' - 'Need someone to pick up my package from the P.O at Rothschild Avenue' and give it a price- 20 NIS.
  Alice, waiting in line to pick up her package at the Rothschild Avenue P.O, gets a notification about Bob's post on 'DoIt'. she accepts the request and earns 20 NIS without any effort.

## 1.3 Our added value:

- **An unlimited variety of services:** As mentioned earlier, we allow users to request any service they want, at any price they want, and to do it immediately to receive a response as soon as possible. On the other hand, users can also offer services on the platform and other users interested in these services can contact them.
- **recommendation engine**: Our application will integrate a recommendation engine that will improve the user experience and match the services that suit

each user according to previous use, geographic location, personal preferences, price levels and more.

- **Authentication:** Applications that dedicated to this purpose, contain authentication mechanisms in order to validate details of the service providers and not allow fraudsters to provide them. On the other hand, platforms like Facebook groups doesn't have this kind of mechanism. We will provide a secure authentication that will keep our users safe and happy.

### 1.4 Paper Organization:

- **Introduction:** introduce the problem we aim to solve and a general information about the solution.
- **Background and Related Work:** This section describes the background required for the research and development of the proposed solution.
It presents the existing solutions to the problem and what are the technological application options available for our solution.
- **Expected Achievements:** This part describes what we expect to achieve in the project: what features we will develop, what are the indicators of success and what are the technological challenges we expect to encounter.
- **Engineering Process:** This part describes the work process we carried out in the project. It describes the steps we performed for the development of the product. the challenges we faced, such as choosing the development technologies or the product architecture. We will also present the products we have produced so far in the form of use case and activity diagrams, user interface characterization and the software architecture.
- **Verification Plan:** This section describes the testing model we chose, iterative, and why it was chosen. The section also presents the test plan we planned for the project.
- **Bibliography :** Describes the sources we used to learn about the problem and the technological tools available to solve the problem.

## 2. <u>Background and Related Work:</u>

### 2.1 The market today:

- After research, we found that these are the leading or very similar competitors in the apps markets (Google Play & Apple Store) that offer similar services:

| Name | Description | Google Play notes | Apple Store notes |
|---|---|---|---|
| **Facebook groups** | probably the largest platform in this list.<br>Today, there is a group on Facebook for any topic. in Israel, for example, there is a group called "need something"? which counts 256K people. In this group, people post demands or questions and other people provide the required services/answers.<br>The problem with this type of platform is that it is very messy, unfocused, and not personally tailored to the user. | - | - |
| **Thumbtack** | Thumbtack allows users to find the right people for all their projects and get a service done. The service providers are professionals. | Over 1 million downloads | 33.3K Ratings with 4.5/5 grade |
| **UrbanClap** | UrbanClap is the largest home service provider that connects customers to service professionals to meet their daily needs. | Over 1 million downloads | 531.9K Ratings with 4.7/5 grade |
| **ServisHero** | deals with different home services and has a database of over 5000 professionals catering to on-demand services across countries | Over 100K Downloads | - |

| | | | |
|---|---|---|---|
| | like Malaysia, Singapore, and Thailand. | | |
| **Bizzby** | Services right from domestic cleaning and ironing to repairment of appliances as well as gardening, junk removal, health, and beauty, to the child and personal care, is catered to by this particular application | Over 10K downloads | 73 Ratings with 2.2/5 |

## 2.2 Main Disadvantages of competitors:

- **Offer professionals:** the big apps like Thumbtack or UrbanClap offer only professionals to provide the services, not unprofessional people. Also the offered services were predetermined, that's means that the flexibility in the types of services we offer is a significant advantage for day-to-day users.
- **Not in Israel:** as far as we research, we didn't find any local app that provide this kind of service in Israel.
- **Focus on inner-house tasks:** apps like 'Bizzby' that allow unprofessional people to offer service is mainly focus on home-services like cleaning. We offer the users to choose whatever they want to provide or get provided.

## 2.3 Mobile development

- The field of mobile application development involves the conceptualization, design, and creation of software programs for mobile devices, primarily utilizing the iOS and Android operating systems. These applications can be pre-installed on the device at the time of purchase, obtained through a mobile application marketplace, or accessed via a mobile web browser.
- Compared to their web-based counterparts, native mobile applications generally exhibit superior performance and aesthetic qualities. This is largely attributed to the fact that native mobile applications are specifically designed and optimized for a specific operating system, whereas web-based applications must be adaptable to multiple platforms.

- **mobile development over web development**
  - **Mobile platform feature sets:** Utilization of device-specific features such as the camera, sensors, and payment methods to enhance the overall user experience.

o **Operating system features**: Access to operating system-specific features such as push notifications, which allows for direct promotion of services and events.

o **Offline access:** Offline functionality, which allows users to access various areas of the application without the need for internet connectivity, thereby providing a more consistent and reliable experience.

o **Convenience user access:** Greater convenience for users, as mobile apps can be accessed directly from the device's home screen, eliminating the need for web browsers or manual entry of web addresses, making the user experience more seamless.

**2.4 Cross-platform development:**

**2.4.1** When talking about cross platform development we need consider two side-users side and developers side.

- **Users point of view:** From the perspective of users, cross-platform development primarily pertains to the ability to utilize a single application across multiple mobile devices, regardless of their underlying operating system. This capability, to have a consistent user interface across multiple devices, is a significant advantage for individuals who desire to utilize the same app on different devices..

- **Developers point of view:** From the perspective of developers, cross-platform development allows for the creation of a single codebase that can be utilized across multiple platforms with minimal modifications. This approach to development streamlines the process by utilizing a single set of code, reusable features, plugins, and components, thus accelerating the development timeline and reducing the time required to bring the product to market.

**2.4.2 Flutter:**

- Flutter, developed by Google, is an open-source technology that enables the creation of mobile, desktop, and web applications utilizing a single codebase. Unlike other commonly used solutions, Flutter is not limited to being a framework or library, but instead, is a comprehensive software development kit (SDK).

**Differences between SDK and framework:**

o **Library:** A library is a collection of pre-written and reusable code that can be integrated into an application to perform a specific function.

o **Framework:** A framework, on the other hand, is a set of tools that serve as a foundation for an application, providing a structure and

basic functionality that the developer must then build upon with their own code to achieve the desired outcomes.

- **SDK:** An SDK, in contrast, encompasses a broader range of tools, including libraries, documentation, application programming interfaces (APIs), and potentially frameworks. It provides developers with all the necessary components for software development.

- Flutter SDK:

"Since Flutter is a full-fledged SDK, it includes a rendering engine, ready-made widgets, testing and integration APIs. The three main architectural layers of Flutter are:

- **Embedder:** A platform-specific embedder provides an entry point; coordinates with the underlying operating system for access to services like rendering surfaces, accessibility, and input; and manages the message event loop. The embedder is written in a language that is appropriate for the platform.

- **Flutter engine:** the core of flutter which is mostly written in C++ and supports the primitives necessary to support all Flutter applications. The engine is responsible for rasterizing composited scenes whenever a new frame needs to be painted. It provides the low-level implementation of Flutter's core API, including graphics, text layout, file and network I/O, accessibility support, plugin architecture, and a Dart runtime and compile toolchain.

- **Flutter framework**: which provides a modern, reactive framework written in the Dart language. It includes a rich set of platform, layout, and foundational libraries, composed of a series of layers."

(Quoted from 'Flutter architectural overview')

- Dart

In order to use flutter, we need to write code in Dart.

Dart is a programming language that is specifically designed and optimized for the creation of high-performance applications on a variety of platforms.

The advantages of dart are:

- **High performance:** Programs written in Dart tend to run faster than programs created in JavaScript.

- **OOP support:** It is an object-oriented programming language with support for inheritance, interfaces, and optional typing features.

**AOT and JIT compilation: "**During development, a fast developer cycle is critical for iteration. The Dart VM offers a just-in-time compiler (JIT) with incremental recompilation (enabling hot reload), and rich debugging

support. When apps are ready to be deployed to production—whether you're publishing to an app store or deploying to a production backend—the Dart ahead-of-time (AOT) compiler can compile to native ARM or x64 machine code." (Quoted from official Dart documentation).

### 2.4.3 Firebase

- Firebase is a BaaS (Backend As A Service) platform offering a wide variety of features and components for enhanced mobile and web application development.
  Firebase reduces the need for users to develop APIs and manage servers. Firebase is a toolset. This toolset is the services that we, as developers, would normally have to build them. This includes things like analytics, authentication, databases, configuration, file storage, push messaging and more.

- **Google Firestore :** is a part of the Google Firebase app development platform. It is a cloud-hosted NoSQL database option for the storage and synchronization of data.
  It allows 2 main features:
  - Real-Time Updates – Firestore utilizes synchronization for updating data across connected devices.
  - Offline Synchronization – Firestore performs caching of data being used by an app to let it read, write, query, and listen to data even with an offline device.

## 2.5 Agile development

Agile development is a software development approach that emphasizes flexibility and rapid iteration. It involves frequent collaboration between cross-functional teams, including developers, designers, and project managers, to deliver high-quality software incrementally.

In Agile development, teams work in short cycles called "sprints," typically lasting one to four weeks. At the beginning of each sprint, the team identifies a set of features or requirements that they will work on during the sprint. The team then collaborates to design, develop, and test these features, with the goal of delivering a usable product increment at the end of each sprint.

Agile development relies on regular communication and collaboration among team members, as well as a focus on continuous improvement and adapting to change. It also relies on lightweight documentation and planning, with a focus on delivering working software quickly.

Overall, Agile development aims to deliver value to customers and stakeholders quickly and efficiently, while also being flexible and responsive to changing requirements and market conditions.

### 2.6 Recommendation system

In the field of recommendation systems, there are three main categories:

- **Content-based filtering:** this type utilizes the similarity of products, services, or content features, as well as user-specific information, to generate recommendations.
- **Collaborative filtering:** on the other hand, relies on the preferences of similar users to make recommendations for a specific individual.
- **Hybrid recommender systems**: integrate multiple recommendation strategies, capitalizing on the strengths of each approach to generate recommendations.

# 3. <u>Expected Achievements</u>

In our project we expect to create a cross platform mobile app that will work on iOS and Android mobile phone and provide the option of providing and consuming services based on user preferences using a recommendation engine that we developed.

**3.1 Goals:**

- Develop a user-friendly mobile app that will allow users to consume and provide services based on their needs.
  The consuming user will have to enter the details about the service he is interested in into a form. After it, he will need to enter the price he is willing to pay.
  After confirming the details, a service event will appear on a map pinned on the area where the service is required.
  The providing user has 2 options:
  The first is to Enter a dedicated screen that will offer him services he can provide. The offers will be adapted to the user in relation to the details he entered in the past and geographical proximity.
  The second option is to enter a screen containing a map on which services that can be provided are marked.
- Allow users to register to the system using identifiers. Using the identifying details, the system will perform authentication in order to confirm the reliability of the users.
- Integrate a ML based recommendation system in the app that improve the user experience and suggest more accurate and fit services for each user.
- Allow users an easy internal communication between of them.

**3.2 Success Criteria:**

- The app will work on both Android and iOS devices without functional differences.
- The app will contain the options to provide and consume services according to the functional and non-functional requirements.
- The integrated recommendation system model will recommend service with high accurate.
- The internal chat will be implemented, and 2 users can send and receive messages (at least text) and media.

### 3.3 Unique Features:

- Internal Chat: Create an Internal chat in order to provide users an easy way to communicate and negotiate without using third-party chat apps.

- Services history: Allow each user to see an history of his provided/consumed service in order to improve the user experience.

- Recommendation system: the app will include an integrated ML model that will suggest relevant services for users based on preferences.

- User profile: each user will have a personal profile that will help him describe the services he interested to supply/consume, his ranking and more details.

- Ranking: the app will allow to rank user based on services that already done. For example, if Bob, used services that Alice provide, then Bob can review Alice service and rank it in scale of 1-10.

- Notifications: The App will use notifications in order to update users about new services that created near to them. Also, when a service is provided, the supplier will mark the service as 'done' and the consumer will get a notification about it.

## 4. <u>Engineering Process</u>

### 4.1 Process:

**4.1.1 Define App purpose:** in the first stage, we defined the purpose our app. We identified the problem that the app will solve: save people time and give an extra income for people who need.

**4.1.2 Identify app audience:** on this stage we modeled our users. We did research using the social networks like Facebook and Instagram and ask users to answer about questions that we prepared. The questions help us to understand the following things about the potential users:

- What is their age range?
- What will be their frequency of use?
- What features would they like to have?
- What are their concerns about such a service?
- What are the devices (computers, mobile phones or smart watch) from which such a service would be used?

We analyzed the data we collected and used it in the next stages.

**4.1.3 Research the competitions:** we looked for a similar services on the different platforms – Facebook, Google Play and App store. We made a list of potential competitors and listed their advantages and disadvantages.

**4.1.4 Define key features and functionality:** on this stage we had a good grasp of what our app I need to have in order to compete with the apps on the market. Additional to the basic features like register, add event, and supply event, we thought about features that were missing to others app, like recommendation system, cross-platform app, unlimited service categories.

**4.1.5 Define required set of skills and tools:** in this stage we understood that we need our app to run on iOS and Android mobile devices. considering out limited resources like number of developers and time that it will take us to learn Kotlin (Android) and Swift (iOS) we decided that we need one code base for both OS's. we searched for this kind of developing options and found 2: React Native and Flutter. In the end, we choose Flutter.

- **Hot Reload:** Flutter has a feature called "hot reload" which allows developers to make changes to the code and see the results immediately in the app, without having to manually stop and restart the app. This can speed up the development process and make it easier to test and debug the app.

- **Customization:** Flutter allows developers to create custom widgets and designs, which gives them more control over the look and feel of the app.

### 4.1.6 Challenges during the process:

- **Choose the most appropriate development platform:** As we mentioned earlier, we had a dilemma between React Native and Flutter, in the end, we choose Flutter because of the following reasons:
  - **Hot Reload:** Flutter has a feature called "hot reload" which allows developers to make changes to the code and see the results immediately in the app, without having to manually stop and restart the app. This can speed up the development process and make it easier to test and debug the app.
  - **Customization:** Flutter allows developers to create custom widgets and designs, which gives them more control over the look and feel of the app. This can be particularly useful for creating apps with unique or complex UI requirements.
  - **Performance:** Flutter uses the Dart programming language, which is compiled ahead-of-time (AOT) into native code. This can result in faster app performance compared to React Native, which uses JavaScript and is interpreted at runtime.
  - **Ecosystem:** Flutter has a strong ecosystem, with a wide range of plugins and packages available for various purposes, such as accessing native device features, integrating with third-party APIs, and more.
- **Choose the right model of the recommendation system:** In order to maximize user experiences, we must choose the most appropriate model of the recommendation system, that is, choose between Content-based filtering and Collaborative filtering, eventually we choose Content-based filtering for these reasons:
  - A key advantage of the content-based filtering model is its ability to generate user-specific recommendations without requiring data on other users. This attribute greatly facilitates scalability to a large user base, making it an ideal choice for systems with a large number of users.
  - the model is able to capture the individual preferences of a user, allowing for the recommendation of specialized items that may be of interest to a small number of users.
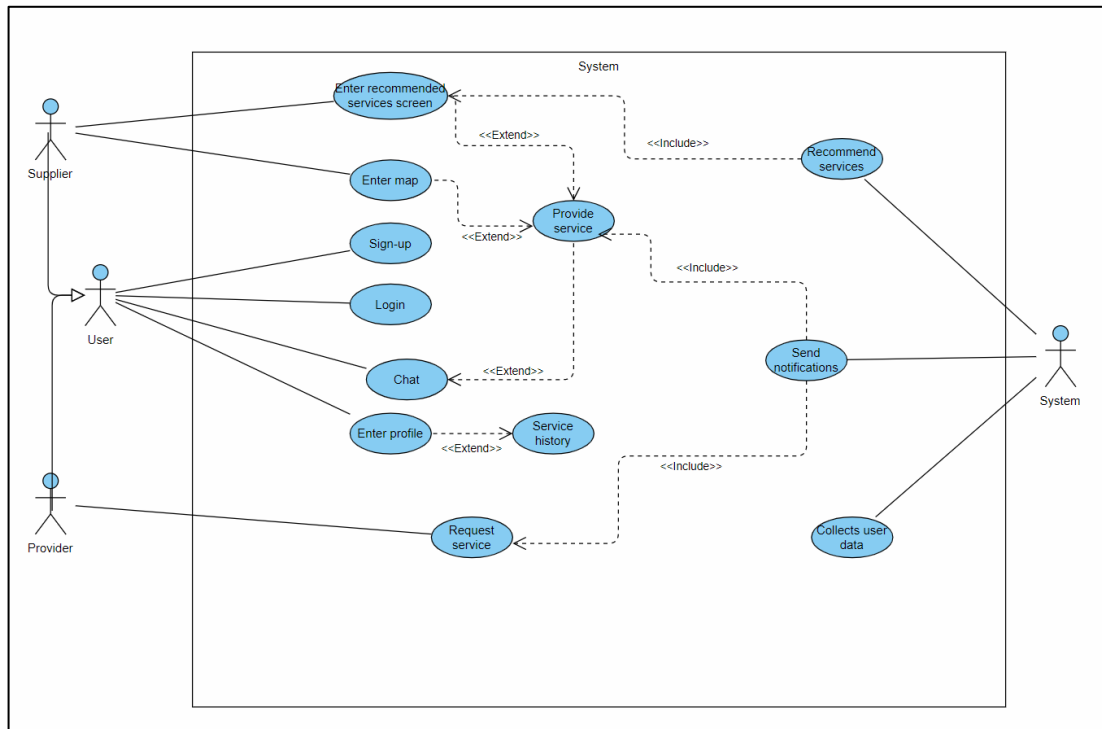
## 4.2 Product

### 4.2.1 Use case diagram:



Figure 1.1

### 4.2.2 Activity diagram

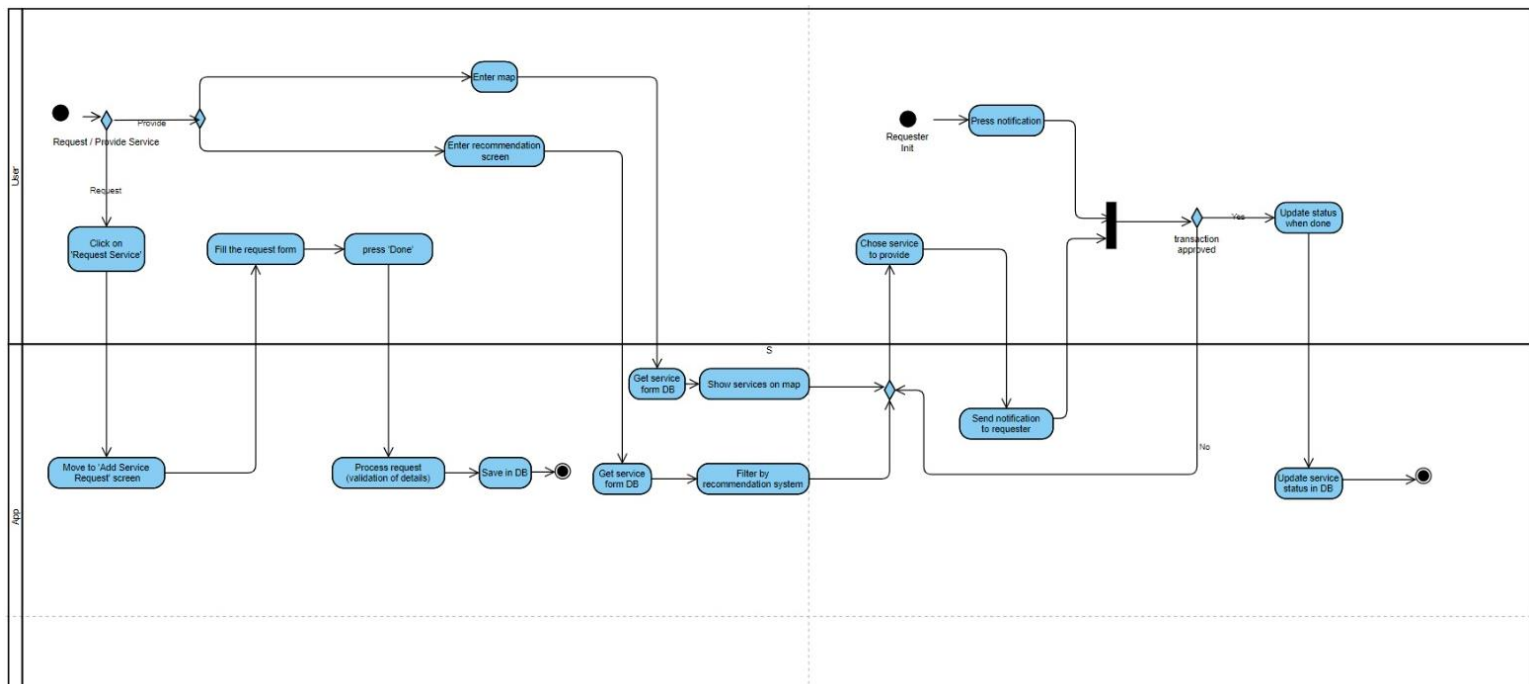- This activity diagram describes the flow of the creating and searching services.



Figure 1.2

### 4.2.3 UI Characterization

- **Login Screen**

    On this screen you can login in 3 different ways – by clicking on the 'sign-in with Email', 'Google' and 'Apple ID' buttons. All of these options will bring you to enter your user details.

    If you are a new user, you can register to the app by clicking on the 'Sign up' button.
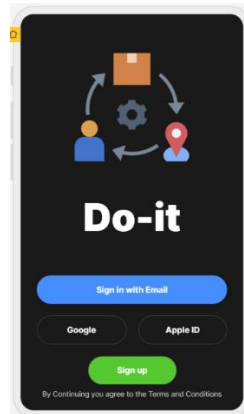


Figure 2.1

- **Profile Screen**

    On this screen the user can edit his profile details like email, picture, etc. He can do it by clicking on the 'edit profile' button.

    By clicking on the 'turn on/off notification' he can change the notification status.

    He can invite users to the app by clicking the 'invite' button.

    By clicking on the 'logout' button he can logout from the app.

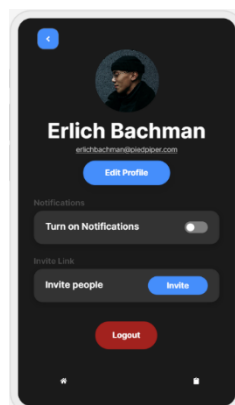    Pressing on the 'back' button on the top will return to the previous screen.



Figure 2.2

- **Home Screen**

This is the main screen of the app.

The user can click on 'overview' button and see all the services he chooses to do today.

Below that there are the services divided into categories.

By clicking on history, the user will move to the 'history screen'.

By clicking on the plus button, the user will move to the 'add service screen'.

By clicking the 'map search button' the user will move to 'map screen' to search for services.

By clicking on the 'board' button in the right bottom side the user will move to the 'requests screen' to watch his own service requests.

By clicking on the profile picture, the user will move to the 'profile screen'.

By clicking on the 'search' button near to the profile picture the user can search service in his services list.
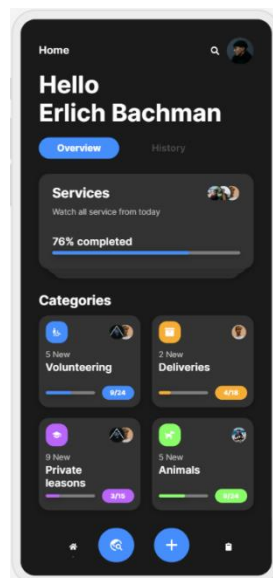


Figure 2.3

- **Map Screen**

  This screen shows a map of the area you are in. Places where service is needed are marked on the map.

  Clicking on a particular service on the map will open a screen with details about the service.

  In the search field you can enter the name of a place and jump to it on the map.

  Pressing on the 'back' button on the top will return to the previous screen
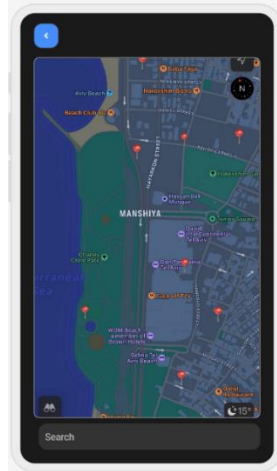


Figure 2.4

- **Recommended for you Screen**

  This screen shows all of the services that recommendation system recommends to the user by his preferences and history of use.

  By clicking on the service, the user can watch the details about the service.

  Near to the service box, there is the distance from the user to the service place.

  Pressing on the 'back' button on the top will return to the previous screen
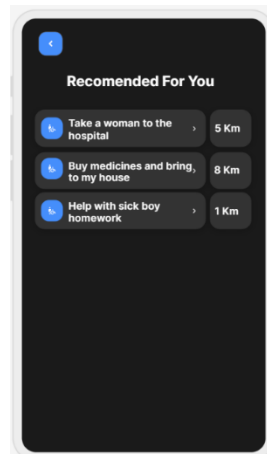


-

Figure 2.5

**History Screen**

On this screen the user can watch the history of the services he has done.

By clicking on a specific service, the user can watch all the details about this service.

Pressing on the 'back' button on the top will return to the previous screen.
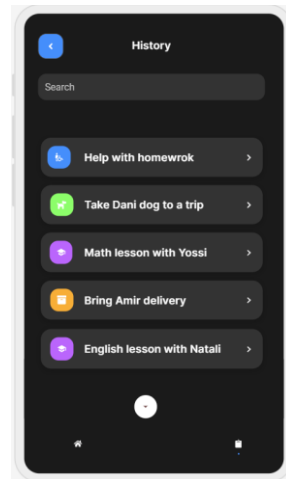


Figure 2.6

- **Add service request Screen**

  On this screen, the user has the option to request a new service.

  He must enter the details as requested in the fields.

  When finished, he must press the 'done' button.

  A popup will open asking for his approval once more. After confirmation, the user will move to new screen with a summary of the service.

  Pressing on the 'back' button on the top will return to the previous screen.
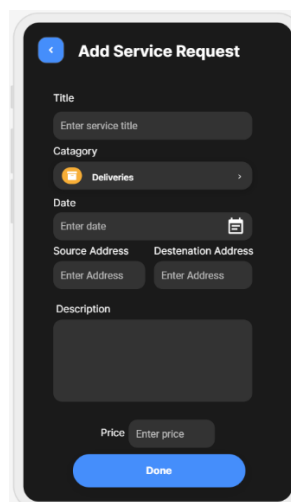


Figure 2.7

- **Chat Screen**
  On this screen there is an option to talk to the service provider/requester.
  The user can enter text in the field and click 'Send'.
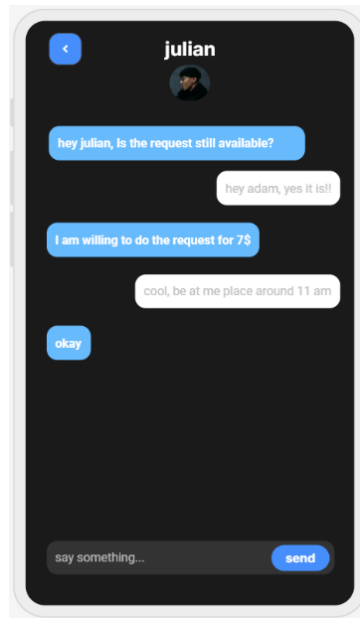  Pressing on the 'back' button on the top will return to the previous screen.



Figure 2.8

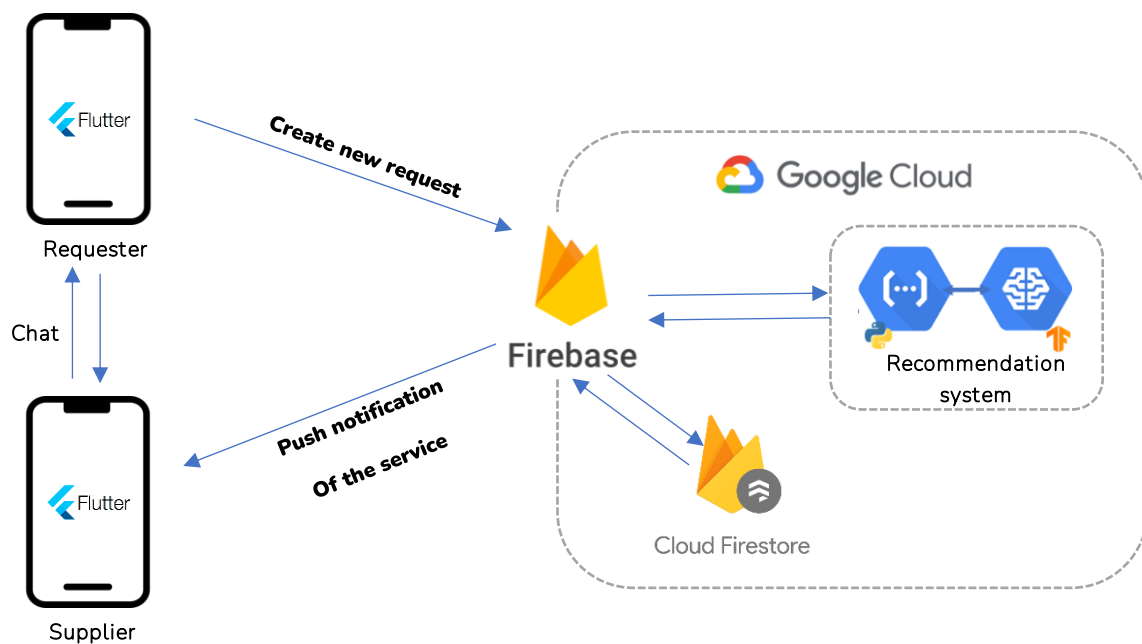## 4.3 Software Architecture



Figure 3.1

# 5. <u>Verification Plan</u>

Since we chose the agile development method, we will have to test the different functions that the application features during the implementation phase. Every sprint we test the features that were developed during the sprint. We created a list of functional and non-functional that we would like to test.

| Req. Number | Requirement | Description | Expected Result |
|---|---|---|---|
| 1 | The system allows users to sign-in to the system. | The user has to enter the app and press sign-in on the home page. Then enter his details. | System validates the details and if they valid, save the details in the DB. Else show an error message on the screen. |
| 2 | The system allows the user to login to the system. | The user has to enter the app and press login on the home page and insert credentials. | The system validates the credentials, if valid and user exist, move the user to the home screen. |
| 3 | The system allows the user to enter his profile page. | The user has to click on his profile picture. | The system switch screen to the user profile screen. |
| 4 | The system allows user to enter a map that shows services. | The user has to press on the map icon in the main screen. | The system switch screen to the map screen and show marked services on the map. |
| 5 | The system allows the user to see details about services on map. | The user has to click on a service he wants the details about. | The system raise a popup window with detail about the |

| | | | service the user pressed on. |
|---|---|---|---|
| 6 | The system allows the user to choose service from the map. | The user has to click on 'provide' button inside the service popup window. | The system open a chat with the requester in order to allow the users to agree on a deal. |
| 7 | The system allows the user to watch a recommended services list. | The user has to click the 'recommended for you' button on the map screen. | The system will switch screen to the 'recommended for you' screen and show a list of recommended service that provided by the recommendations system. |
| 8 | The system allows the user to see details about services on list. | The user has to click on a service he wants the details about. | The system raise a popup window with detail about the service the user pressed on. |
| 9 | The system allows the user to add new service request. | The user has to click on the 'Add service request' button in the main screen, fill the request form a press done. | The system validate the details. If valid, save on DB and send related notifications. Else show an error message on the screen. |
| 10 | The system allows the user to communicate with other users. | The user has to enter the chat room form different user profile or by providing service. | The system switch to the chat screen with the requested user. |

| 11 | The system allows the user to rank other users. | The user has to choose rating on different user profiles. | The system will save the rank in the DB and calculate the new average rating. |
|----|---|---|---|
| 12 | The system will be able to send notifications. | Notification will be sent by system after providing event or offering service event. | The system will send notification to the destination user with the relevant details. |
| 13 | The system will be able to recommend on services to users according to user preferences. | The system will be using a model to filter a service according to user preferences | The system will use the recommendations system to generate recommended services list and show the list in the 'recommended for you' screen. |

# 6. <u>References:</u>

1. Eleskovic Denis, (2020, Dec), A comparison of Hybrid and Progressive Web Applications for the Android platform.
2. Stevenson Doug, (2018, Sep 18) What is Firebase? https://medium.com/firebase-developers/what-is-firebase-the-complete-story-abridged-bcc730c5f2c0
3. Flutter architectural overview https://docs.flutter.dev/resources/architectural-overview
4. Flutter VS PWA: Which is Better for Mobile App Development? (2022, Oct, 17), https://cubettech.com/resources/blog/flutter-vs-pwa-which-is-better-for-mobile-app-development/
5. What are today's top recommendation engine algorithms?, (2020, Mar, 9), https://itnext.io/what-are-the-top-recommendation-engine-algorithms-used-nowadays-646f588ce639

GitHub repository of the project:

https://github.com/sshaulnv/Final-project-doit