



# CmpE 451 - Milestone I Report

Karpuz - Freelancer Platform

Group 5

November 1, 2018

# Contents

<b>0 Accessibility</b>	<b>2</b>
<b>1 Executive Summary</b>	<b>2</b>
1.1 Project Introduction . . . . .	2
1.2 Work Done for Milestone 1 . . . . .	2
1.3 Road Ahead . . . . .	2
<b>2 List and Status of Deliverables</b>	<b>3</b>
<b>3 Evaluation of the Status of Deliverables and Their Impact on Plan</b>	<b>3</b>
<b>4 Coding Work</b>	<b>4</b>
<b>5 Requirements</b>	<b>5</b>
5.1 Glossary . . . . .	5
5.2 Functional Requirements . . . . .	6
5.2.1 User requirements . . . . .	6
5.2.2 System requirements . . . . .	7
5.3 Non-functional Requirements . . . . .	8
<b>6 Design</b>	<b>10</b>
6.1 Web . . . . .	10
6.1.1 Landing Page . . . . .	10
6.1.2 Register & Login Page . . . . .	11
6.1.3 Home & Example Project Page . . . . .	12
6.1.4 User Profile Page . . . . .	13
6.2 Android . . . . .	14
6.2.1 Launch Page . . . . .	14
6.2.2 Welcome Page . . . . .	14
6.2.3 Register Page . . . . .	15
6.2.4 Login Page . . . . .	15
6.2.5 Home Page . . . . .	16
<b>7 Project Plan</b>	<b>17</b>
<b>8 GitHub (Issues, PR's, Projects and Milestone I)</b>	<b>18</b>
<b>9 Code Structure</b>	<b>18</b>
9.1 Backend . . . . .	18
9.2 Frontend . . . . .	18
9.3 Android . . . . .	18
<b>10 Evaluation of Tools and Management</b>	<b>19</b>
10.1 Tools . . . . .	19
10.1.1 Deployment . . . . .	19
10.1.2 Testing . . . . .	19
10.1.3 Android . . . . .	19
10.1.4 Backend . . . . .	19
10.1.5 Frontend . . . . .	19
10.1.6 API's . . . . .	19
10.1.7 Project . . . . .	19
10.2 Project Management . . . . .	20
10.2.1 Issues . . . . .	20
10.2.2 Pull Requests . . . . .	20
10.2.3 Meetings . . . . .	20
10.2.4 Communication . . . . .	20

# 0 Accessibility

You can reach our Web Application from [here](#).

## 1 Executive Summary

### 1.1 Project Introduction

Karpuz is a freelance job platform, where anyone can sign up as a freelancer or contractor, and post job listing and/or apply to them. Contractors (also referred as Clients) can put job information, such as the total pay of the job, job description, assets needed for the task, deadline, and so on. Freelancers have an open platform for showing off their job portfolio and accept jobs. Karpuz's intelligent recommendation engine (which is just a keyword matcher in current plan) will suggest freelancers to clients and vice versa. Karpuz is a platform not just for software developers, but also for UI/UX designers and anyone doing a job related to software lifecycle.

### 1.2 Work Done for Milestone 1

For our Milestone 1, our goals were:

- Decide and document API using UML diagrams created during project planning
- Create & structure backend project
- Implement the most necessary endpoints from API, which were decided to be user register, user login, user logout, project listing, project creation.
- Choose Cloud provider, install necessary components and get server up & running
- Choose technology to use for frontend
- Create and structure frontend project
- Implement at least landing page and homepage for frontend
- Implement register, login, logout, project listing, and project creation functionalities in frontend
- Create and structure Android project
- Implement Network components and Mock networking for Android
- Implement register, login, and persistent login functionalities on Android
- Implement Homepage structure

We can proudly say we hit all our Milestone 1 goals. We're filling the backlog as we advance in the project, and prioritizing tasks before diving into them. We believe where we're going is much more important than how fast we're going.

### 1.3 Road Ahead

We're getting used to our compass and gaining momentum, so we're aiming to get faster in time. We aim for all components of the project (Server, Frontend and Android) to advance in nearly same pace, and we make it a priority to document our progress. We believe communication comes before processes, so we aim to use Lean approach in our development. We aim to have:

- Basic annotation functionality implemented in all platforms
- Basic search functionality implemented in all platforms
- E-mail confirmation integrated

and much more for our Milestone 2.

## 2 List and Status of Deliverables

Name	Delivery Date	Delivered
Web: Landing Page	October 29, 2018	✓
Web: Login & Register Page	October 29, 2018	✓
Web: Home Page	October 29, 2018	✓
Web: User Profile Page	October 29, 2018	✓
Web: Project Page	October 29, 2018	✓
Android: Landing Page	October 29, 2018	✓
Android: Login & Register Page	October 29, 2018	✓
Android: Home Page	October 29, 2018	✓

Table 1: List All deliverables and Their Status

## 3 Evaluation of the Status of Deliverables and Their Impact on Plan

We were able to deliver everything we planned for the first milestone, but we aim to speed up for the rest of the way. Our production was slow, since it was the beginning of the project, but now that we spent some time working on it, we have laid the foundation that is necessary for us to work efficiently, and we hope to continue in this path to improve our both speed and quality of work. Also, we will continue to work on features that are delivered to improve their quality as well.

## 4 Coding Work

Team Member	Contributions
Enis Simsar	Construct the frontend boiler plate in React. Create Redux and Saga environments for API calls. Design Landing Page components and theme. Design Login Page and create links between register, login and landing pages. Design User Profile Page and LoggedIn User Profile Page components and general layout for logged in users. Create links between User Profile and Home. Implement Login and Register api connection. Implement API token storing mechanism. Implement not logged in user redirection to login page. Implement logout logic for logged in user. Implement the connection logged in user profile, projects and create project API connections. Setup the docker environment for frontend and backend. Deploy the system in the Digital Ocean instance.
Bahadir Öncel	Create and structure Android project in Android Studio. Implement Network structure and mock network framework. Design and implement launch screen, welcome screen, login screen, register screen, and initial home screen. Implement login persist functionality using Android system Account Manager.
Alp Kaan Usen	Create the project model. Implement the endpoints of the API that deals with retrieving and editing a user's profile. Overall minor modifications and bug fixes to the API.
Muhammed Emin Güre	Implement the endpoints of the API that deals with project model like crate, get etc. Code review of entire backend codebase with manual testing of every endpoint. Minor modifications and bug fixes to the API.
Muhammed Emin Ayar	Create the user model. Design of the authentication functionalities. Implement the endpoints of the API that deals with authentication like login, register, logout. Code review of user and project functionalities. Some modifications and bug fixes. Implement image upload and download functionalities.
Güliz İrem Gökçeler	Implement the frontend side of register page. Design Project Page for both unregistered and registered user. Create formValidator component to check Register form validation.
Batuhan Enes Çağlayan	-
İhsan Berkan Balaban	-
Oğuzhan Kırlar	-

Table 2: Coding Work for Each Team Member

## 5 Requirements

### 5.1 Glossary

- **User:** A person that opens the Web application or Android client of the system. This person can be registered or unregistered.
- **Guest User:** A user that is unregistered, or not logged in to the system.
- **Admin User:** A user that is specially chosen by the developers, or other Admin Users, in order to maintain the system's content. They have more powers than other users, outlined in the Functional Requirements section below.
- **Registered User:** A user that has logged in to the system. They can choose the role of Freelancer, Client or both.
- **Role of Freelancer:** A role that wants to take on Projects for Clients in exchange for payment.
- **Role of Client:** A role that wants post Projects to be done by Freelancers.
- **Password:** A string of characters that each Registered User has in order to log in to the system.
- **Valid Password:** A Password that consists of at least 8 alphanumeric characters.
- **Freelancer:** A user whose role is Freelancer
- **Client:** A user whose role is Client.
- **Project:** A set of tasks defined by a Client to be done by a Freelancer of the Client's choosing (provided that the Freelancer has made a Bid for that Project), before a certain deadline of the Client's choosing.
- **In Progress Project:** A project that has been assigned to a freelancer; a project which the owner (client) chose a freelancer who bidden for it
- **Virtual Wallet:** A field that signifies how much money a Registered User has in the system.
- **Bid:** The act, performed by Freelancers on available Projects, showing the Freelancer is interested in taking on the Project. Each Bid comes with an amount of money that the Freelancer demands completing the Project.
- **Message:** A method of communication between two Registered Users, visible only to those two Registered Users and Admin Users.
- **Block:** The act, performed by Admin Users, of suspending any activity by or regarding the blocked party. Registered Users and Projects may be blocked.
- **Report:** The act, performed by Registered Users, of calling Admin Users to attention about content violating the Terms of Service.
- **Terms of Service:** The rules of the system that are necessary to accept in order to become a Registered User, and to comply with in order keep Registered User status. (To be defined later.)
- **Rate:** The act, performed by Clients on Freelancers, or by Freelancers on Clients, of giving integer points between 1 and 5 (inclusive), that corresponds to how well the latter adhered to the Project specifications. This is a one-time occurrence at the end of each completed Project.
- **Comment:** The act, performed by Clients on Freelancers, or by Freelancers on Clients, of giving feedback about the Project. This is a one-time occurrence at the end of each completed Project.
- **Semantic Tag:** A string that signifies some quality of the attached content.

- **Portfolio:** A certain set of Projects that were completed by a Freelancer and past experience of the Freelancer.
- **Seed Payment Method:** A payment method where Client sends some part of the total money to the Freelancer when a milestone is reached by the Freelancer.

## 5.2 Functional Requirements

### 5.2.1 User requirements

- **5.2.1.1.** Users shall be able to sign up anytime with their username, password and email.
- **5.2.1.2.** Users shall be able to search for projects.
  - **5.2.1.2.1.** Users should be able to filter the searching results according to the rating and the price.
- **5.2.1.3. Admin User**
  - **5.2.1.3.1.** Admin users shall be able to view any registered user account, project, rating, comment, and report.
  - **5.2.1.3.2.** Admin users shall be able to block or remove any registered user account.
  - **5.2.1.3.3.** Admin users shall be able to view messages between registered users.
  - **5.2.1.3.4.** Admin users shall be able to message any registered user.
  - **5.2.1.3.5.** Admin users shall be able to edit, block or remove any project.
  - **5.2.1.3.6.** Admin users shall be able to search for projects.
  - **5.2.1.3.7.** Admin users should be able to generate a report about projects, freelancers, and clients.
- **5.2.1.4. Registered User**
  - **5.2.1.4.1.** Registered users shall be able to log in with their username and password.
  - **5.2.1.4.2.** Registered users shall be able to update their personal information and password.
  - **5.2.1.4.3.** Registered users shall have a virtual wallet.
    - **5.2.1.4.3.1.** Registered users shall be able to deposit money to and withdraw money from their virtual wallet.
  - **5.2.1.4.4.** Registered users shall be able to search for projects and users.
    - **5.2.1.4.4.1.** Registered users should be able to filter the searching results according to the rating and the price.
  - **5.2.1.4.5.** Registered users shall be able to view a project in its entirety.
  - **5.2.1.4.6.** Registered users shall be able to view any registered user profile information.
  - **5.2.1.4.7.** Registered users shall be able to send messages any registered user.
  - **5.2.1.4.8.** Registered users shall be able to report any registered user account and any projects.
  - **5.2.1.4.9.** Registered users shall be able to annotate
    - **5.2.1.4.9.1.** text areas such as freelancer portfolios, project description and project title with text or image.
    - **5.2.1.4.9.2.** image areas such as project attachments with text or image.
  - **5.2.1.4.10** Registered users shall be able to edit or delete their own annotations.
  - **5.2.1.4.11.** Registered users:
    - **5.2.1.4.11.1.** Shall be able to access their profile.
    - **5.2.1.4.11.2.** Shall be able to search for projects and users.
    - **5.2.1.4.11.2.** Shall be able to access their wallet.

- **5.2.1.4.12.** Registered users shall be able to create their profile information.
  - **5.2.1.4.12.1.** Registered users shall be able to add a profile picture to their profile.
  - **5.2.1.4.12.2.** Registered users shall be able to share their name, surname, age, gender and short text about themselves on their profile.
- **5.2.1.4.13.** Registered users shall be able to switch between two user types, which are freelancer and client.
  - **5.2.1.4.13.1.** Freelancer
    - **5.2.1.4.13.1.1.** Freelancers shall be able to create a profile.
    - **5.2.1.4.13.1.1.1.** Freelancers shall be able to add semantic tags to their profile for their interests and their skills.
    - **5.2.1.4.13.1.1.2.** Freelancers shall be able to create a portfolio to be added to their profile upon account creation.
    - **5.2.1.4.13.1.2.** Freelancers shall be able to view their ongoing projects.
    - **5.2.1.4.13.1.3.** Freelancers shall be only able to rate and comment on a client after a project is finished.
    - **5.2.1.4.13.1.4.** Freelancers shall be able to receive their payments via the virtual wallet.
    - **5.2.1.4.13.1.5.** Freelancers shall be able to choose to add the project to his portfolio after a project is finished.
    - **5.2.1.4.13.1.6.** Freelancers should be able to connect their LinkedIn profile to their account.
    - **5.2.1.4.13.1.7.** Freelancers shall be able to bid on a project.
  - **5.2.1.4.13.2.** Client
    - **5.2.1.4.13.2.1.** Clients shall be able to create a project.
    - **5.2.1.4.13.2.1.1.** Clients shall be able to choose a topic for their project.
    - **5.2.1.4.13.2.1.2.** Clients shall be able to choose a deadline for their project.
    - **5.2.1.4.13.2.1.3.** Clients shall be able to specify an estimated budget for their project.
    - **5.2.1.4.13.2.1.4.** Clients shall be able to upload attachments about their project.
    - **5.2.1.4.13.2.1.5.** Clients shall be able to associate semantic tags with their project.
    - **5.2.1.4.13.2.2.** Clients shall be able to select one of the bids from his/her project's bid list.
    - **5.2.1.4.13.2.3.** Clients shall be able to offer their projects to a specific freelancer via push notifications.
    - **5.2.1.4.13.2.4.** Clients shall be able to change the visibility of their projects to guest users, choosing either public or private.
    - **5.2.1.4.13.2.5.** Clients shall be able to pay freelancers with the balance of their virtual wallet.
    - **5.2.1.4.13.2.6.** Clients shall be able to rate and comment on a freelancer only after their project has been completed by that freelancer.
    - **5.2.1.4.13.2.7.** Clients shall be able to change the details of the project within the bidding period.

## 5.2.2 System requirements

- **5.2.2.1.** Search
  - **5.2.2.1.1.** The system shall rank the results for project searches according to multitude of parameters such as client name, topic, semantic tags, budget, project status or deadline.
  - **5.2.2.1.2.** The system shall rank the results for user searches according to multitude of parameters such as user, and semantic tags.

- **5.2.2.1.3.** The system shall support search with keywords
- **5.2.2.2. Recommendation**
  - **5.2.2.2.1.** The system shall generate freelancer recommendations for clients with similar semantic tags and portfolio information to their projects.
  - **5.2.2.2.2.** The system shall generate project recommendations for freelancers according to the portfolio information and semantic tags they provided.
- **5.2.2.3. Notification**
  - **5.2.2.3.1.** The system shall be able to send notifications to:
    - **5.2.2.3.1.1.** the freelancer, when:
      - **5.2.2.3.1.1.1.** a project that the freelancer bidden finished its bidding period, or the freelancer won the bid for a project that they bidden.
      - **5.2.2.3.1.1.2.** the freelancer got a message from a registered user.
      - **5.2.2.3.1.1.3.** a client notified the freelancer about a project.
      - **5.2.2.3.1.1.4.** a project that the freelancer is involved in changed status.
    - **5.2.2.3.1.2.** the client, when:
      - **5.2.2.3.1.2.1.** a freelancer bidden to a project that the client opened.
      - **5.2.2.3.1.2.2.** the client got a message from a registered user.
      - **5.2.2.3.1.2.3.** an in progress project that the client owns changed status.
- **5.2.2.4. Payment**
  - **5.2.2.4.1.** The system shall allow users to use seed payment method.
- **5.2.2.5. Wallet**
  - **5.2.2.5.1.** The system shall show all the transaction history to the registered users.
- **5.2.2.6. General**
  - **5.2.2.6.1.** The system shall be able to recommend similar projects while client creating a new project.
  - **5.2.2.6.2.** The system shall show the total number of projects, freelancers, clients and total earnings of freelancers.
  - **5.2.2.6.3.** The system shall list public projects.
  - **5.2.2.6.4.** The system shall list most bided projects.
  - **5.2.2.6.5.** The system shall exhibit user testimonials.
  - **5.2.2.6.6.** The system shall show recent and popular projects freelancer users.

### **5.3 Non-functional Requirements**

- **5.3.1. Usability**
  - **5.3.1.1.** The language of the application shall be English.
- **5.3.2. Availability**
  - **5.3.2.2.1.** The system shall have a Web application that supports:
    - **5.3.2.2.1.1.** Chrome browser version 63 and up
    - **5.3.2.2.1.2.** Firefox browser version 58 and up
    - **5.3.2.2.1.3.** Opera browser version 50 and up
  - **5.3.2.2.2.** The system shall have an Android application that supports version 6.0 Android Marshmallow and later (used by 57.7)
- **5.3.3. Annotation**

- **5.3.3.1.** 3.1. The system shall utilize annotations in compliance with the latest W3C Web Annotation Protocol (accessed January 2018).
- **5.3.4. Third Party Integration**
  - **5.3.4.1.** Android API shall be used for the native Android app.
  - **5.3.4.2.** Wikidata API shall be used for fetching semantic tags.
  - **5.3.4.3.** Push notifications shall be used to notify users of possible changes of contracts.
    - **5.3.4.3.1.** Chrome push notifications API for Web
    - **5.3.4.3.2.** Google push notifications API for Android
  - **5.3.4.4.** LinkedIn API should be used for connecting user accounts to LinkedIn accounts and fetching information.
  - **5.3.4.5.** Smart Moderation API shall be used for preventing swear words on the user's comments.
- **5.3.5. Security**
  - **5.3.5.1.** The system shall protect user passwords in a database using SHA-256 encryption.
  - **5.3.5.2.** The system shall block sources that send more than 5 requests per second, to prevent DDoS attacks.
  - **5.3.5.3.** The system shall check user queries against SQL injection attacks.
  - **5.3.5.4.** User account email shall be unique.
  - **5.3.5.5.** User account password shall be valid.
  - **5.3.5.6.** The system should use 3D Secure Payment for payments.
- **5.3.6. Reliability**
  - **5.3.6.1.** The database shall be backed up daily and weekly.
    - **5.3.6.1.1.** Weekly backups shall be full backups.
    - **5.3.6.1.2.** Daily backups shall be differential backups.
  - **5.3.6.2.** The system should have monthly maintenance period of 3 hours.
  - **5.3.6.3.** The system shall recover fully from a crash in at most 24 hours.
  - **5.3.6.4.** The system shall have at least 99
- **5.3.7. Scalability**
  - **5.3.7.1.** The system should handle at least 10000 online registered users at any time.
- **5.3.8. Performance**
  - **5.3.8.1.** The system should return all queries within at most 3 seconds to users with at least an 8 Mbps internet connection.
  - **5.3.8.2.** The system should draw every frame at most 33ms (1000 ms/30 fps) to avoid being perceived as laggy.
- **5.3.9. Legal**
  - **5.3.9.1.** Clients shall be able to choose a license for their projects.

## 6 Design

### 6.1 Web

#### 6.1.1 Landing Page

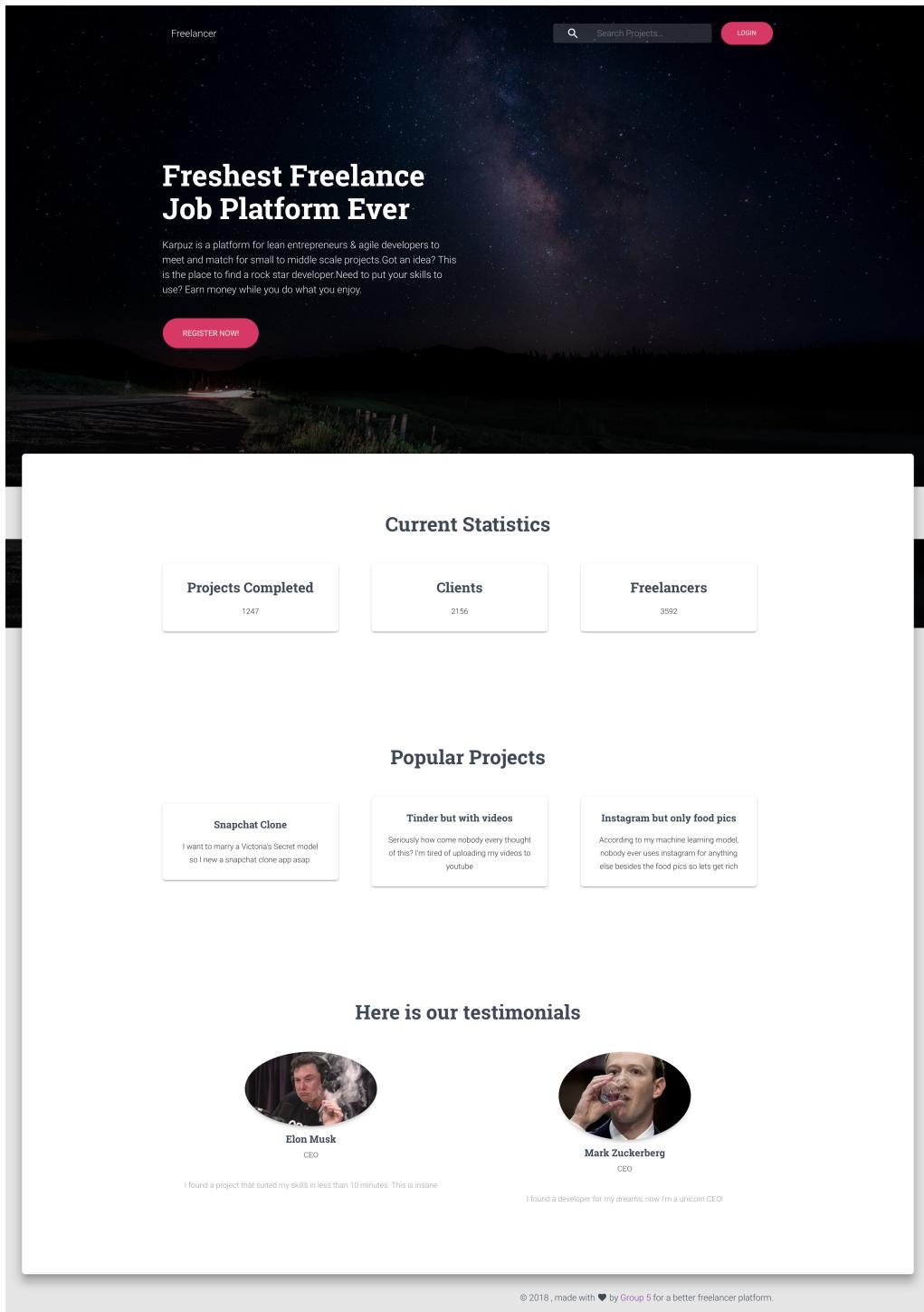
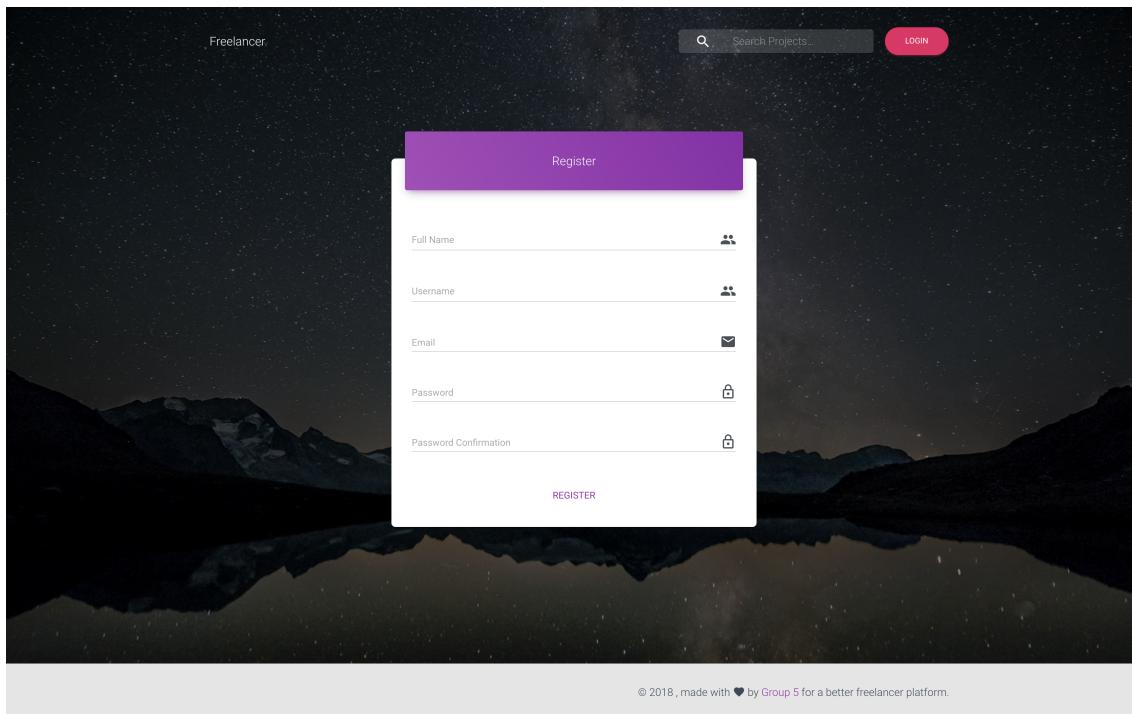
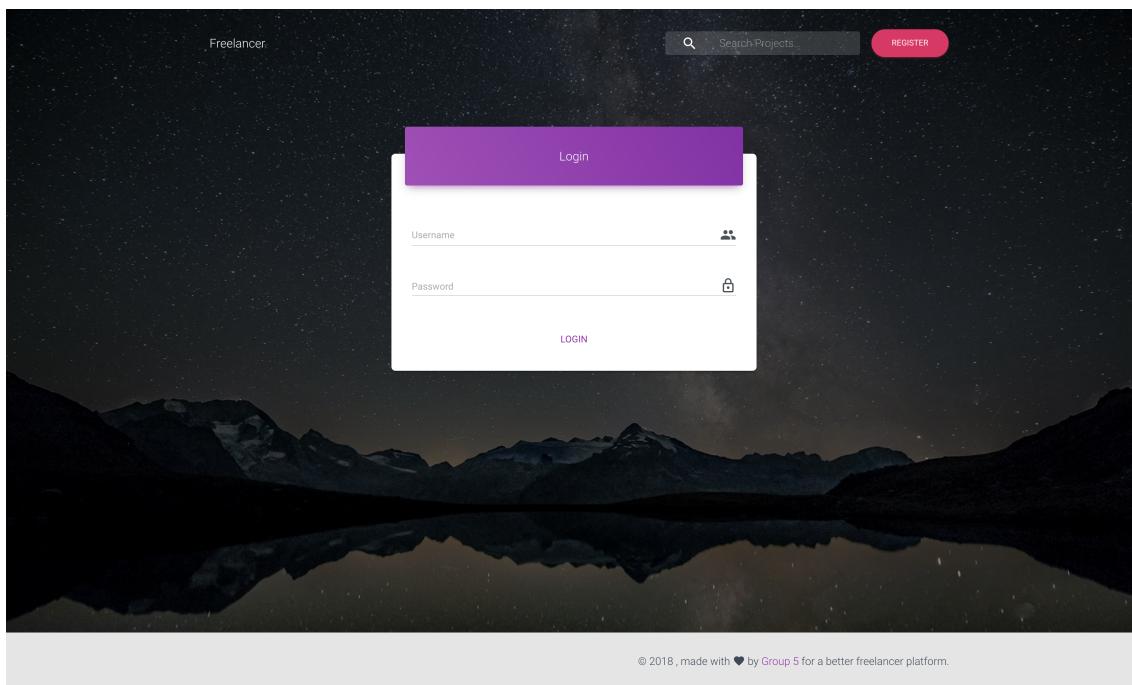


Figure 1: Web Landing Page

### 6.1.2 Register & Login Page



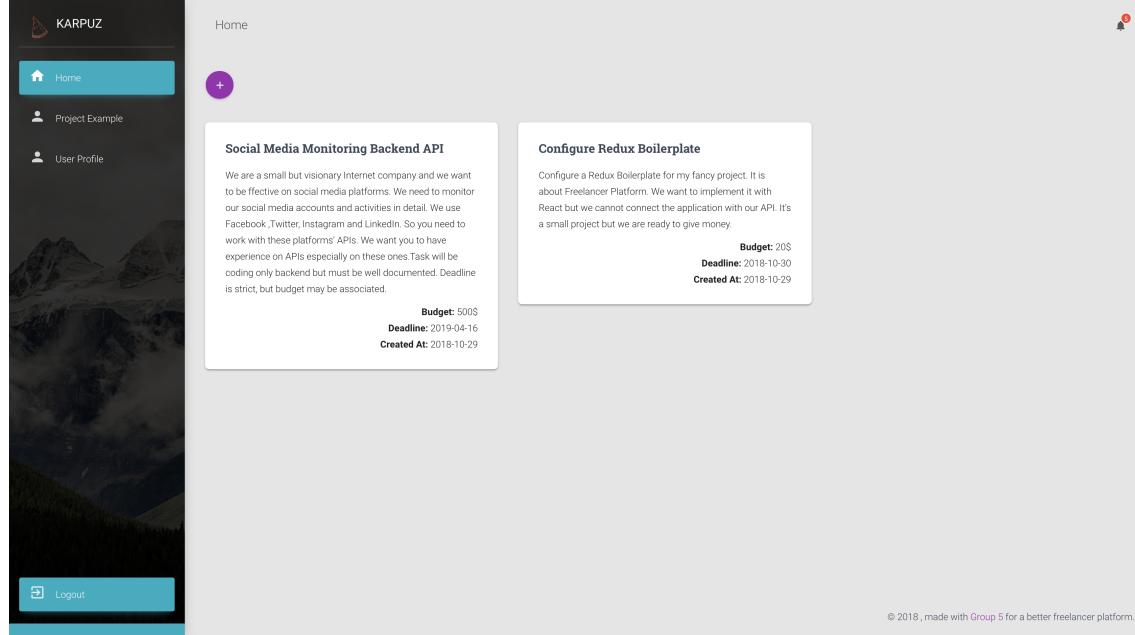
(a) Register Page



(b) Login page

Figure 2: Authentication Pages for Web

### 6.1.3 Home & Example Project Page



(a) Home Page

The screenshot shows the details of the 'Social Media Monitoring Backend API' project from the previous image. The top navigation bar says 'Project'.

**Project Owner**: Mete Kocaman, CEO AT SOSYAL BİLİŞİM. There is a 'SEND MESSAGE' button.

**Bids**: A table showing three bids:

Suggested Price	Bid Owner	Bid Owner's Rate
450	Mert Kızıl	3.5
400	Selin Sezen	4.5
425	Deniz Gökçe	4

**BID ON PROJECT**

At the bottom right, there is a copyright notice: "© 2018, made with Group 5 for a better freelancer platform."

(b) Example Project page

Figure 3: Home and Example Project Pages for Web

#### 6.1.4 User Profile Page

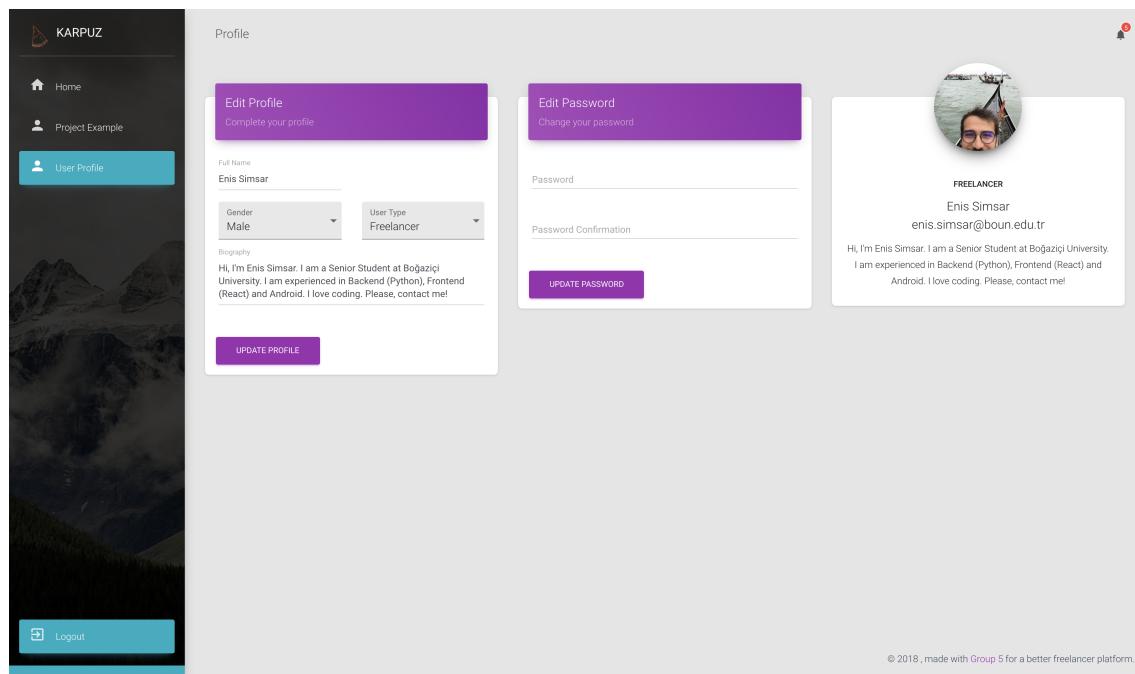


Figure 4: Web User Profile Page

## 6.2 Android

### 6.2.1 Launch Page

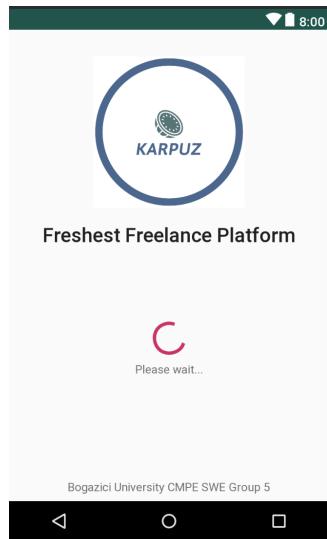


Figure 5: Android Launch Page

### 6.2.2 Welcome Page

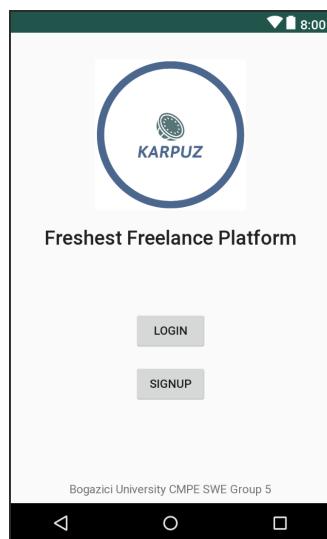


Figure 6: Android Welcome Page

### 6.2.3 Register Page

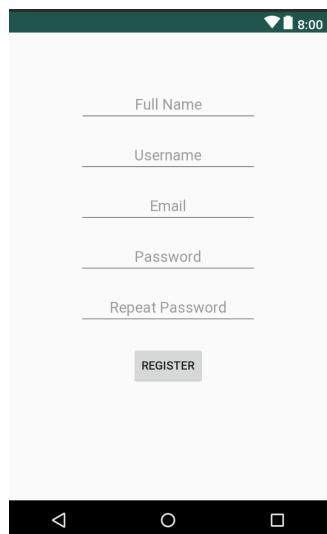


Figure 7: Android Register Page

### 6.2.4 Login Page

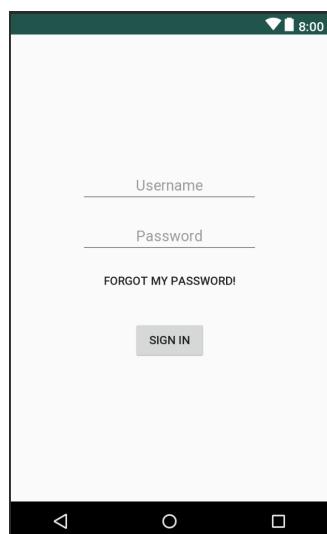
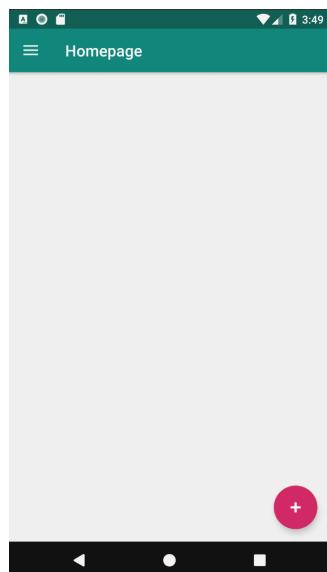
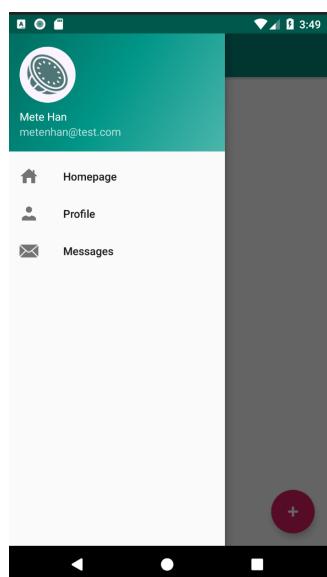


Figure 8: Android Login Page

### 6.2.5 Home Page



(a) Home Page



(b) Home Page - Drawer Open

Figure 9: Homepages for Android

## 7 Project Plan

We added the project plan as png. Also, you can reach it from [here](#).

		Name	Duration	Start	Finish	Predecessors	Resource Names
1	Backend	<b>Implementation for Part I</b>	15 days	10/16/18 8:00 AM	11/5/18 5:00 PM		
2	Backend	Backend	12 days	10/16/18 8:00 AM	10/31/18 5:00 PM		
3	Auth	Boiler Plate Code	4 days	10/16/18 8:00 AM	10/19/18 5:00 PM	Alp Kaan Usen;Muhammed Emin Ayar;Muhammed Emin Gür	
4	Auth	Auth	5 days	10/22/18 8:00 AM	10/26/18 5:00 PM		
5	Auth	Register End Point	2 days	10/22/18 8:00 AM	10/23/18 5:00 PM	Alp Kaan Usen	
6	Auth	Login End Point	2 days	10/22/18 8:00 AM	10/23/18 5:00 PM	Muhammed Emin Gür	
7	Auth	Get Profile End Point	2 days	10/22/18 8:00 AM	10/23/18 5:00 PM	Muhammed Emin Ayar	
8	Auth	Update Profile End Point	3 days	10/24/18 8:00 AM	10/26/18 5:00 PM	Muhammed Emin Gür	
9	Auth	Logout End Point	3 days	10/24/18 8:00 AM	10/26/18 5:00 PM	Alp Kaan Usen	
10	Project	<b>Project</b>	6 days	10/24/18 8:00 AM	10/31/18 5:00 PM		
11	Frontend	Create Project	3 days	10/24/18 8:00 AM	10/26/18 5:00 PM	Muhammed Emin Ayar	
12	Frontend	Discard Project	3 days	10/24/18 8:00 AM	10/26/18 5:00 PM	Muhammed Emin Gür	
13	Frontend	Get Project	2 days	10/29/18 8:00 AM	10/30/18 5:00 PM	Alp Kaan Usen	
14	Frontend	Finish Project	3 days	10/29/18 8:00 AM	10/31/18 5:00 PM	Muhammed Emin Ayar	
15	Frontend	Get Projects by Filter	2 days	10/30/18 8:00 AM	10/31/18 5:00 PM	Alp Kaan Usen	
16	Frontend	<b>Frontend</b>	12 days	10/16/18 8:00 AM	10/31/18 5:00 PM		
17	Frontend	Boiler Plate Code	4 days	10/16/18 8:00 AM	10/19/18 5:00 PM	Enis Simsar;Guliz Irem Gökceler;İhsan Berkhan Balaban	
18	Frontend	Landing Page	2 days	10/22/18 8:00 AM	10/23/18 5:00 PM	Enis Simsar	
19	Frontend	Auth	5 days	10/22/18 8:00 AM	10/26/18 5:00 PM		
20	Frontend	Register Page	2 days	10/22/18 8:00 AM	10/23/18 5:00 PM	Guliz Irem Gökceler	
21	Frontend	Login Page	2 days	10/22/18 8:00 AM	10/23/18 5:00 PM	İhsan Berkhan Balaban	
22	Frontend	Profile Page	3 days	10/24/18 8:00 AM	10/26/18 5:00 PM	Enis Simsar	
23	Frontend	Edit Profile Page	3 days	10/24/18 8:00 AM	10/26/18 5:00 PM	Guliz Irem Gökceler	
24	Frontend	<b>Project</b>	6 days	10/24/18 8:00 AM	10/31/18 5:00 PM		
25	Frontend	Home Page	3 days	10/24/18 8:00 AM	10/26/18 5:00 PM	İhsan Berkhan Balaban	
26	Frontend	Search Project Page	3 days	10/29/18 8:00 AM	10/31/18 5:00 PM	Enis Simsar	
27	Frontend	Project Page	3 days	10/29/18 8:00 AM	10/31/18 5:00 PM	Guliz Irem Gökceler	
28	Frontend	Android	12 days	10/16/18 8:00 AM	10/31/18 5:00 PM		
29	Frontend	Boiler Plate Code	4 days	10/16/18 8:00 AM	10/19/18 5:00 PM	Bahadir Oncel;Batuhan Enes Caglayan;Oguzhan Kirlar	
30	Frontend	Landing Page	2 days	10/22/18 8:00 AM	10/23/18 5:00 PM	Bahadir Oncel	
31	Frontend	Auth	5 days	10/22/18 8:00 AM	10/26/18 5:00 PM		
32	Frontend	Register Page	2 days	10/22/18 8:00 AM	10/23/18 5:00 PM	Batuhan Enes Caglayan	
33	Frontend	Login Page	2 days	10/22/18 8:00 AM	10/23/18 5:00 PM	Oguzhan Kirlar	
34	Frontend	Profile Page	3 days	10/24/18 8:00 AM	10/26/18 5:00 PM	Bahadir Oncel	
35	Frontend	Edit Profile Page	3 days	10/24/18 8:00 AM	10/26/18 5:00 PM	Batuhan Enes Caglayan	
36	Frontend	<b>Project</b>	6 days	10/24/18 8:00 AM	10/31/18 5:00 PM		
37	Frontend	Home Page	3 days	10/24/18 8:00 AM	10/26/18 5:00 PM	Oguzhan Kirlar	
38	Frontend	Search Project Page	3 days	10/29/18 8:00 AM	10/31/18 5:00 PM	Bahadir Oncel	
39	Frontend	Project Page	3 days	10/29/18 8:00 AM	10/31/18 5:00 PM	Batuhan Enes Caglayan	
40	Frontend	<b>Testing for Above Functionalities</b>	3 days	11/1/18 8:00 AM	11/5/18 5:00 PM		
41	Testing	Unit Test for Backend	3 days	11/1/18 8:00 AM	11/5/18 5:00 PM	Each Member	
42	Testing	Snapshot Test for Frontend	3 days	11/1/18 8:00 AM	11/5/18 5:00 PM	Enis Simsar;Guliz Irem Gökceler;İhsan Berkhan Balaban	
43	Testing	Milestone 1	1 day	11/6/18 8:00 AM	11/6/18 5:00 PM	1	Each Member
44	Testing	<b>Implementation for Part II</b>	23 days	11/7/18 8:00 AM	12/7/18 5:00 PM		
45	Testing	Backend	23 days	11/7/18 8:00 AM	12/7/18 5:00 PM		
46	Testing	User	6 days	11/7/18 8:00 AM	11/14/18 5:00 PM		
47	Testing	Get Average Rating	3 days	11/7/18 8:00 AM	11/9/18 5:00 PM	Muhammed Emin Ayar	
48	Testing	Get Ratings	3 days	11/7/18 8:00 AM	11/9/18 5:00 PM	Muhammed Emin Gür	
49	Testing	Report User	3 days	11/12/18 8:00 AM	11/14/18 5:00 PM	Alp Kaan Usen	
50	Testing	Delete User	3 days	11/12/18 8:00 AM	11/14/18 5:00 PM	Muhammed Emin Ayar	
51	Testing	<b>Admin</b>	6 days	11/12/18 8:00 AM	11/19/18 5:00 PM		
52	Testing	Get Reports for Projects	3 days	11/12/18 8:00 AM	11/14/18 5:00 PM	Muhammed Emin Gür	
53	Testing	Get Reports for Users	3 days	11/15/18 8:00 AM	11/19/18 5:00 PM	Alp Kaan Usen	
54	Testing	Block User	3 days	11/15/18 8:00 AM	11/19/18 5:00 PM	Muhammed Emin Ayar	
55	Testing	<b>Project</b>	12 days	11/5/18 8:00 AM	11/30/18 5:00 PM		
56	Testing	Get Recommended Projects	3 days	11/15/18 8:00 AM	11/19/18 5:00 PM	Muhammed Emin Gür	
57	Testing	Add Bid	4 days	11/20/18 8:00 AM	11/23/18 5:00 PM	Alp Kaan Usen	
58	Testing	Accept Bid	4 days	11/20/18 8:00 AM	11/23/18 5:00 PM	Muhammed Emin Ayar	
59	Testing	Get Project Bids	4 days	11/20/18 8:00 AM	11/23/18 5:00 PM	Muhammed Emin Gür	
60	Testing	Add Project to Portfolio	5 days	11/26/18 8:00 AM	11/30/18 5:00 PM	Alp Kaan Usen	
61	Testing	Report Project	5 days	11/26/18 8:00 AM	11/30/18 5:00 PM	Muhammed Emin Ayar	
62	Testing	<b>Annotation</b>	10 days	11/26/18 8:00 AM	12/7/18 5:00 PM		
63	Testing	Annotate Project	5 days	11/26/18 8:00 AM	11/30/18 5:00 PM	Muhammed Emin Gür	
64	Testing	Delete Annotation for Project	5 days	12/3/18 8:00 AM	12/7/18 5:00 PM	Alp Kaan Usen	
65	Testing	Annotate Portfolio	5 days	12/3/18 8:00 AM	12/7/18 5:00 PM	Muhammed Emin Ayar	
66	Testing	Delete Annotate for Portfolio	5 days	12/3/18 8:00 AM	12/7/18 5:00 PM	Muhammed Emin Gür	
67	Frontend	<b>Frontend</b>	23 days	11/7/18 8:00 AM	12/7/18 5:00 PM		
68	Frontend	User	8 days	11/7/18 8:00 AM	11/16/18 5:00 PM		
69	Frontend	Show User Rating Info	8 days	11/7/18 8:00 AM	11/16/18 5:00 PM	Enis Simsar	
70	Frontend	Report User Page	8 days	11/7/18 8:00 AM	11/16/18 5:00 PM	Guliz Irem Gökceler	
71	Frontend	<b>Admin</b>	15 days	11/7/18 8:00 AM	11/27/18 5:00 PM		
72	Frontend	Reported Users Page	8 days	11/7/18 8:00 AM	11/16/18 5:00 PM	Ihsan Berkhan Balaban	
73	Frontend	Reported Projects Page	7 days	11/19/18 8:00 AM	11/27/18 5:00 PM	Enis Simsar	
74	Frontend	<b>Project</b>	15 days	11/7/19/18 8:00 AM	12/7/18 5:00 PM		
75	Frontend	Show Project Bids	7 days	11/19/18 8:00 AM	11/27/18 5:00 PM	Guliz Irem Gökceler	
76	Frontend	Send Project Bid Popup	7 days	11/19/18 8:00 AM	11/27/18 5:00 PM	Ihsan Berkhan Balaban	
77	Frontend	Report Project Page	8 days	11/28/18 8:00 AM	12/7/18 5:00 PM	Enis Simsar	
78	Frontend	<b>Annotation</b>	8 days	11/28/18 8:00 AM	12/7/18 5:00 PM		
79	Frontend	Project Annotation Support	8 days	11/28/18 8:00 AM	12/7/18 5:00 PM	Guliz Irem Gökceler	
80	Frontend	Portfolio Annotation Support	8 days	11/28/18 8:00 AM	12/7/18 5:00 PM	Ihsan Berkhan Balaban	
81	Frontend	<b>Android</b>	23 days	11/7/18 8:00 AM	12/7/18 5:00 PM		
82	Frontend	User	8 days	11/7/18 8:00 AM	11/16/18 5:00 PM		
83	Frontend	Show User Rating Info	8 days	11/7/18 8:00 AM	11/16/18 5:00 PM	Bahadir Oncel	
84	Frontend	Report User Page	8 days	11/7/18 8:00 AM	11/16/18 5:00 PM	Batuhan Enes Caglayan	
85	Frontend	<b>Admin</b>	15 days	11/7/18 8:00 AM	11/27/18 5:00 PM		
86	Frontend	Reported User Page	8 days	11/7/18 8:00 AM	11/16/18 5:00 PM	Oguzhan Kirlar	
87	Frontend	Reported Projects Page	7 days	11/19/18 8:00 AM	11/27/18 5:00 PM	Bahadir Oncel	
88	Frontend	<b>Project</b>	15 days	11/19/18 8:00 AM	12/7/18 5:00 PM		
89	Frontend	Show Project Bids	7 days	11/19/18 8:00 AM	11/27/18 5:00 PM	Batuhan Enes Caglayan	
90	Frontend	Send Project Bid Popup	7 days	11/19/18 8:00 AM	11/27/18 5:00 PM	Oguzhan Kirlar	
91	Frontend	Report Project Page	8 days	11/28/18 8:00 AM	12/7/18 5:00 PM	Bahadir Oncel	
92	Frontend	<b>Testing</b>	6 days	12/10/18 8:00 AM	12/17/18 5:00 PM	1:44	
93	Testing	Unit Tests	2 days	12/10/18 8:00 AM	12/11/18 5:00 PM	Each Member	
94	Testing	User Tests	2 days	12/12/18 8:00 AM	12/13/18 5:00 PM	Each Member	
95	Testing	Debugging	2 days	12/14/18 8:00 AM	12/17/18 5:00 PM	Each Member	
96	Testing	Milestone 2	1 day	12/18/18 8:00 AM	12/18/18 5:00 PM	92	Each Member
97	Testing	Launch Presentation	1 day	1/2/19 8:00 AM	1/2/19 5:00 PM		Each Member

Figure 10: Project Plan

## 8 GitHub (Issues, PR's, Projects and Milestone I)

- You can reach our issues from [here](#).
- You can reach our pull requests from [here](#).
- You can reach our GitHub project boards from [here](#).
- You can reach our GitHub Milestone I from [here](#).

## 9 Code Structure

### 9.1 Backend

We are using Django Framework and its Model-View-Template structure for the implementation of the API, and MongoDB as our database. We are using a docker environment to make the team work easier. We started with the default project template of Django and worked our way from there. We are using JSON Web Tokens for authentication. When a user is logged in, a token is generated and this token is used for the requests that need authentication. We are getting requests as JSON format and we are responding requests with JSON format. We are using mongoengine to integrate MongoDB to our project. Since MongoDB is a non-relational database, it allows us to be more flexible during development. We are making use of new branches for every new functionality we implement and after it is merged we delete the branch. We are using static media files for any media that might be used in the project. We did not implement any unit tests yet, but we plan to implement them for every endpoint.

### 9.2 Frontend

We used React for Frontend project. We start with the create-react-app project that is the official boiler plate for React Web Apps. We have the root of the Frontend project, there are docker-compose file and docker file for deployment of frontend, Package.json file for dependencies, src folder for out all source codes, and public for our main html file. We tried to break up all pages into small pieces and we implemented all small parts for the pages. We created issues for pages, features such as API connection and bugs. We implemented our codes on the different branches. For example, we develop the design of the home page in the home-page branch and when the implementations is finished, we created a pull request to develop/frontend branch that is the main development branch for Frontend project. We merge the home-page branch with develop/frontend branch, then we deleted the home-page branch, because we do not want unused branches. Periodically, we merged the develop/frontend branch with master. Then, after merging these two branches, we pull the all changes to the Digital Ocean Instance, if pull request comment contains "Require Build" text, we build the frontend project in the Digital Ocean Instance (Docker build) and restart the Docker container, otherwise we restart the Docker container.

### 9.3 Android

We're using Android Studio latest version, and our target API is 23 as stated in our Requirements. We're using MVVM (Model - View - ViewModel) structure for our Android project. We're using Retrofit for networking and RxJava whenever possible. Our aim is to have as much distinct components as possible, to prevent spaghetti code. We don't like singletons & are fans of dependency injection. Currently we don't have any tests but we aim to have at least 1 test for each PR. We always create & work in personal branches.

## 10 Evaluation of Tools and Management

### 10.1 Tools

#### 10.1.1 Deployment

We used Docker and Docker Compose for Backend and Frontend projects. We rent a machine from Digital Ocean and this machine configuration is enough for our project. We installed Docker and Docker Compose to this instance and created docker-compose and docker files for Backend and Frontend projects. We wrote a script that when triggering this script, it causes down all docker container and then up all containers. We have 2 .env files one of them for backend project and the other for frontend project. Before the deployment, developer who wants to deploy our app must be set this .env files. S/he can use our .env-example files when creating his/her .env files.

#### 10.1.2 Testing

We did not any tests for this Milestone. We will implement the tests about API and Android for next Milestone.

#### 10.1.3 Android

We're using Android Studio and are fairly happy with it. We're usually progressing fairly quickly, unless we're having a dependency collision. We try to keep Pull Requests as small as possible to ease review process. We try to get at least 1 approval before merging our PRs.

#### 10.1.4 Backend

We're using PyCharm as our IDE and we are following PEP-8 code standard. This way we have a better compatibility in our codes. We are branching for new functionalities and merging them with our backend master branch via pull requests. For now we are testing our functions manually but we will implement unit tests soon.

#### 10.1.5 Frontend

We're using Web Storm and VS Code for development. Both of them are good for coding. Web Storm is a messy IDE but it helps us for renaming, checking dependency and import statements and so on. On the other hand, we are more flexible in VS Code. So, we're using both of them. We are developing our codes in a feature branch and creating a pull request for each feature. We did not apply the getting at least 1 approval before merging it, because of the limited time to Milestone I. We will merge all pull request with at least 1 approval. We are using React Framework. This framework is hard to learn and maintain. Therefore, maybe we will change this framework with Vue or another Web Development Framework. For API calls, we are using Redux. Also this framework is messy to add new endpoint call. So, maybe we will change this package with axios package that is another API request package. Also we used Postman in order to test API calls before implementing them in the Frontend project.

#### 10.1.6 API's

We're using Swagger to document our API, and Postman to test it. Swagger has a bigger potential but since we don't need it, we are not aiming to integrate it to our backend.

#### 10.1.7 Project

We are using the Project Libre for Project Plan. Maybe, we will change this program with Open Project program. So, open project is available for Docker and it can run in our Digital Ocean instance.

## **10.2 Project Management**

### **10.2.1 Issues**

We are creating issues for bugs and new features (small tasks such as navigation bar of the logged in user), labeling all issues, assigning a team member and selecting the right project such as web, backend or android. We are using boards, sometimes we are creating some notes in the boards and then we are converting these notes to issues. So, we are keeping track our backlogs from GitHub boards.

### **10.2.2 Pull Requests**

We are not able to PR's fully efficient for this Milestone because we did not have enough time to review. I will PR's more efficiently in the next periods.

### **10.2.3 Meetings**

We're arranging meetings as we need them - we try to have at least 1 meeting a week but not regularly. Regular class works as a good meeting environment for us.

### **10.2.4 Communication**

We're having a hard time getting our less experienced friends into project. More experienced members have most of the burden of the project and it's not healthy. We aim to improve this situation by encouraging them more.