



# CmpE 451 - Milestone II Report

**Karpuz - Freelancer Platform**

Group 5

December 6, 2018

# Contents

<b>0</b>	<b>Accessibility</b>	<b>3</b>
<b>1</b>	<b>Executive Summary</b>	<b>3</b>
1.1	Project Introduction . . . . .	3
1.2	Work Done for Milestone 2 . . . . .	3
1.3	Road Ahead . . . . .	3
<b>2</b>	<b>List and Status of Deliverables</b>	<b>4</b>
<b>3</b>	<b>Evaluation of the Status of Deliverables and Their Impact on Plan</b>	<b>4</b>
<b>4</b>	<b>Coding Work</b>	<b>5</b>
<b>5</b>	<b>Requirements</b>	<b>6</b>
5.1	Glossary . . . . .	6
5.2	Functional Requirements . . . . .	7
5.2.1	User requirements . . . . .	7
5.2.2	System requirements . . . . .	8
5.3	Non-functional Requirements . . . . .	9
<b>6</b>	<b>Design</b>	<b>11</b>
6.1	Web . . . . .	11
6.1.1	Home Page . . . . .	11
6.1.2	Freelancer Bidding . . . . .	11
6.1.3	Client Bid Management . . . . .	12
6.1.4	Client Side Project . . . . .	12
6.1.5	Freelancer Side Project . . . . .	13
6.1.6	Client Projects . . . . .	14
6.1.7	Freelancer Projects . . . . .	14
6.1.8	User Profile . . . . .	15
6.1.9	Other User Profile . . . . .	16
6.1.10	Freelancer Rating . . . . .	17
6.1.11	Multiple Milestones . . . . .	18
6.2	Android . . . . .	18
6.2.1	Home Page . . . . .	18
6.2.1.1	Project Creation . . . . .	19
6.2.1.2	Profile . . . . .	19
6.2.1.3	Single Project Detail . . . . .	20
<b>7</b>	<b>Project Plan</b>	<b>21</b>
<b>8</b>	<b>GitHub (Issues, PR's, Projects and Milestone II)</b>	<b>22</b>
<b>9</b>	<b>Code Structure</b>	<b>22</b>
9.1	Backend . . . . .	22
9.2	Frontend . . . . .	22
9.3	Android . . . . .	22
<b>10</b>	<b>Evaluation of Tools and Management</b>	<b>23</b>
10.1	Tools . . . . .	23
10.1.1	Deployment . . . . .	23
10.1.2	Testing . . . . .	23
10.1.3	Android . . . . .	23
10.1.4	Backend . . . . .	23
10.1.5	Frontend . . . . .	23
10.1.6	API's . . . . .	23
10.1.7	Project . . . . .	23

10.2 Project Management . . . . .	23
10.2.1 Issues . . . . .	23
10.2.2 Pull Requests . . . . .	24
10.2.3 Meetings . . . . .	24
10.2.4 Communication . . . . .	24

## 0 Accessibility

You can reach our Web Application from [here](#). You can reach our Android Application APK 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 2

For our Milestone 2, our goals were:

- Web: Bidding
- Web: Milestones
- Web: Profile picture upload
- Web: Project attachment upload
- Web: Milestone attachment upload
- Android: Homepage
- Android: Project info display
- Android: User info display
- Android: Project creation
- Android: Design

We can proudly say we hit all our Milestone 2 goals. We're filling the backlog in a faster rate, and prioritizing tasks better. We still believe where we're going is much more important than how fast we're going. Android is behind web component in comparison but it's going strong.

### 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:

- Android caught up with frontend
- Basic annotation functionality implemented
- Basic search functionality implemented in all platforms
- E-mail confirmation integrated

and much more for our Final Milestone.

## 2 List and Status of Deliverables

Name	Delivery Date	Delivered
Web: Rating User based on Project	December 4, 2018	✓
Web: End to End Project Flow	December 4, 2018	✓
Web: Bidding Project and Managing Bids	December 4, 2018	✓
Web: Upload Attachment for portfolios, projects, milestones and user profile image	December 4, 2018	✓
Web: Project Multiple Milestone Section	December 4, 2018	✓
Web: User Projects Page	December 4, 2018	✓
Web: User Portfolio and Wallet Section	December 4, 2018	✓
Web: Project management menu	December 4, 2018	✓
Web: Other User Profile Page	December 4, 2018	✓
Android: Homepage	December 4, 2018	✓
Android: Project info display	December 4, 2018	✓
Android: User info display	December 4, 2018	✓
Android: Project creation	December 4, 2018	✓
Android: Design	December 4, 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 second 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	Create other user profile page with API connection. Create user's projects (client or freelancer) page with API connection. Add discarding, editing, finishing and deleting project popup with API connection. Help my team mate for Jenkins in deployment script and project structure. Design Attachment upload component. Then, implement user profile image upload functionality. Add portfolio for users and its upload functionality. Implement project flow end to end, finish and rating, etc. Implement listing, and managing bids. Implement wallet for users (withdraw and deposit money mock). Implement project attachment functionality. Implement upload project milestones' attachments.
Bahadır Öncel	Implement Android homepage for querying the backend for projects and displaying them in a list format. Implement user profile viewing (editing pending) Implement project creation. Implement project list refresh. Implement user logout. Fix design to be in line with frontend. Implement single project viewing. Lay foundation for project bidding.
Alp Kaan Usen	Restructure the url patterns and general architecture of the API, to make it more in line with the REST API conventions and fix some bugs. Add filtering to get projects endpoint, and by doing so provide basic text search. Implement edit project and get own projects endpoints. Implement update bid endpoint. Add wallet model and implement necessary endpoints. Integrate the payment system to the finish project endpoint.
Muhammed Emin Güre	Implemented portfolio, milestone, rating, bid models and endpoints and have some meetings to change the endpoints in order to catch the best practices for REST API. Migrate the API docs to Postman and create collections of requests to make the testing easy. Set up the Jenkins (Continuous Integration and Delivery Tool) for automatic deployment via GitHub commit hooks. Our system now has the ability to deploy itself when a commit pushed to master branch.
Muhammed Emin Ayar	Review and manually test all the endpoints in the end-to-end project process(i.e creation, bidding, rating etc.). Change the profile image upload function to general purpose file upload service. Design attachments file structure, implement attachment fields, re-write API response functions so that they include attachments, implement attachment deletion. Search and learn about Wiki-Data API service and design semantic tag models, search engine philosophy(not yet implemented just designed).
Güliz İrem Gökçeler	Implement dynamic forms for specific usages. Create, delete and edit Milestones of a project with API connection by using dynamic forms. Implement rating project feature. Making active the rate option after project is finished.
Batuhan Enes Çağlayan	For android side, researched for APIs that will help our team in analyzing user and application behaviour, keep specific statistics etc. Decided that Fabric will suit us well and should be implemented.
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 bidded 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 bided finished its bidding period, or the freelancer won the bid for a project that they bided.
      - **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 bided 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 in 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 Home Page

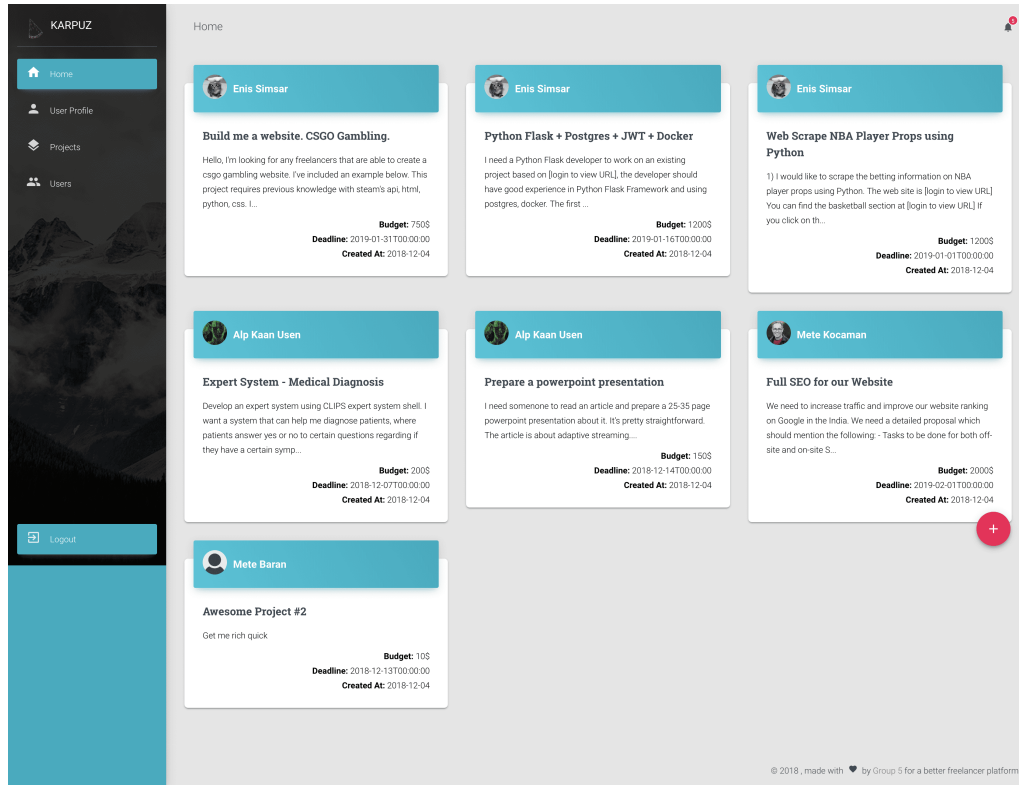


Figure 1: Home Page

#### 6.1.2 Freelancer Bidding

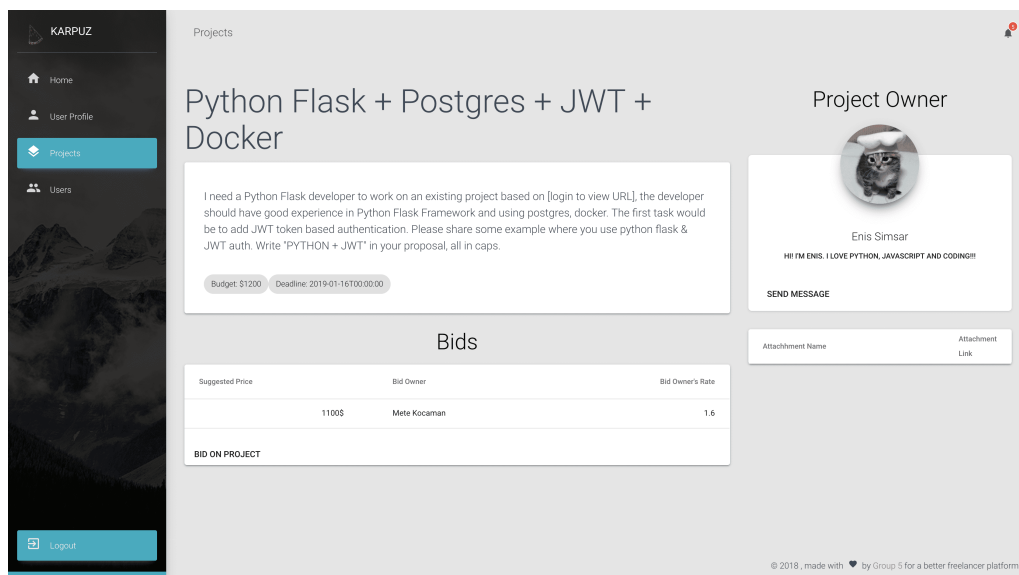


Figure 2: Freelancer Bidding

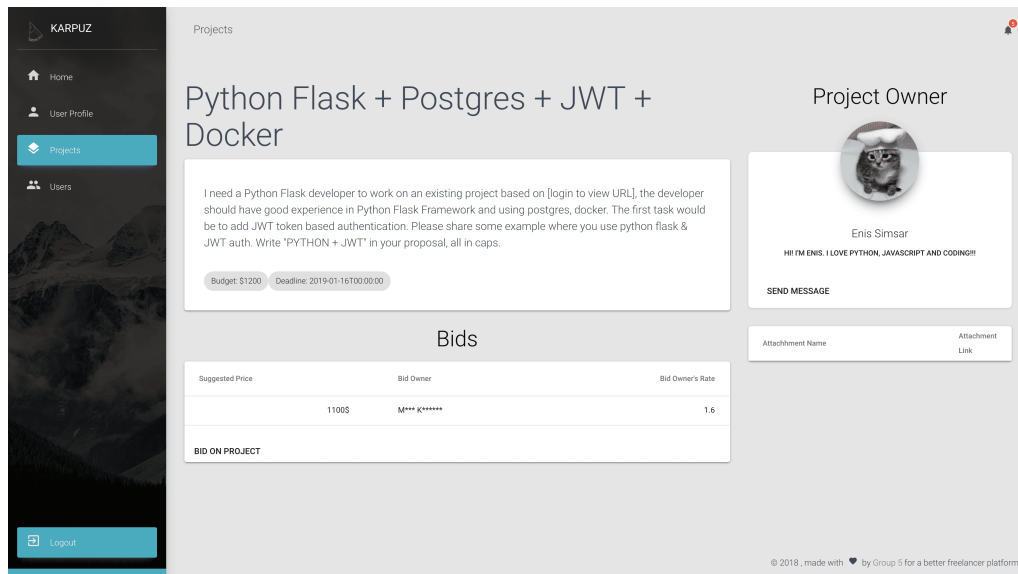


Figure 3: Freelancer Bidding Other

### 6.1.3 Client Bid Management

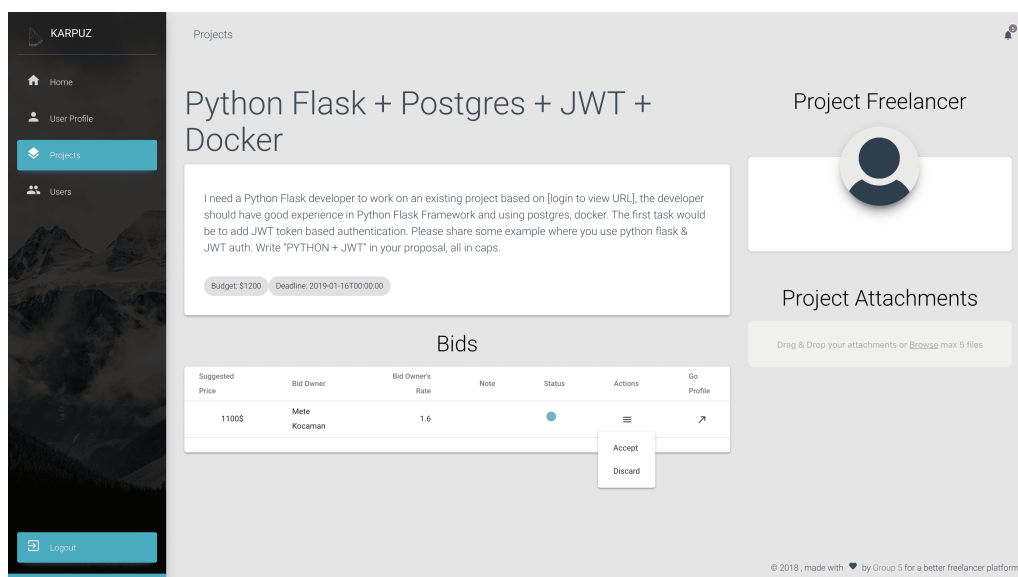


Figure 4: Client Bid Management

### 6.1.4 Client Side Project

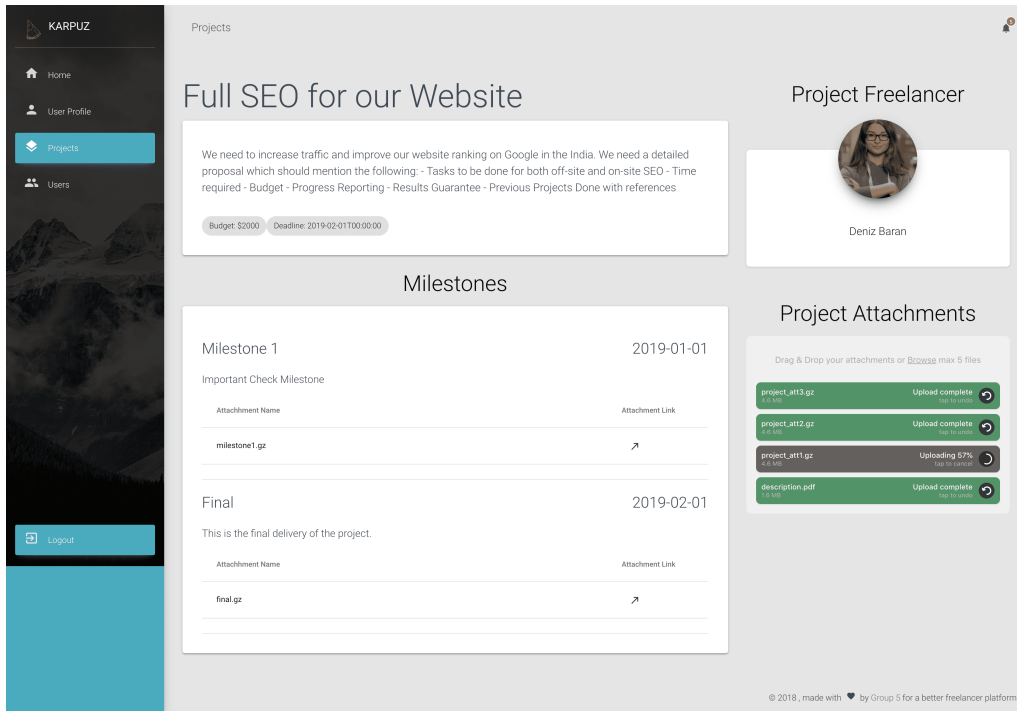


Figure 5: Client Side Project

### 6.1.5 Freelancer Side Project

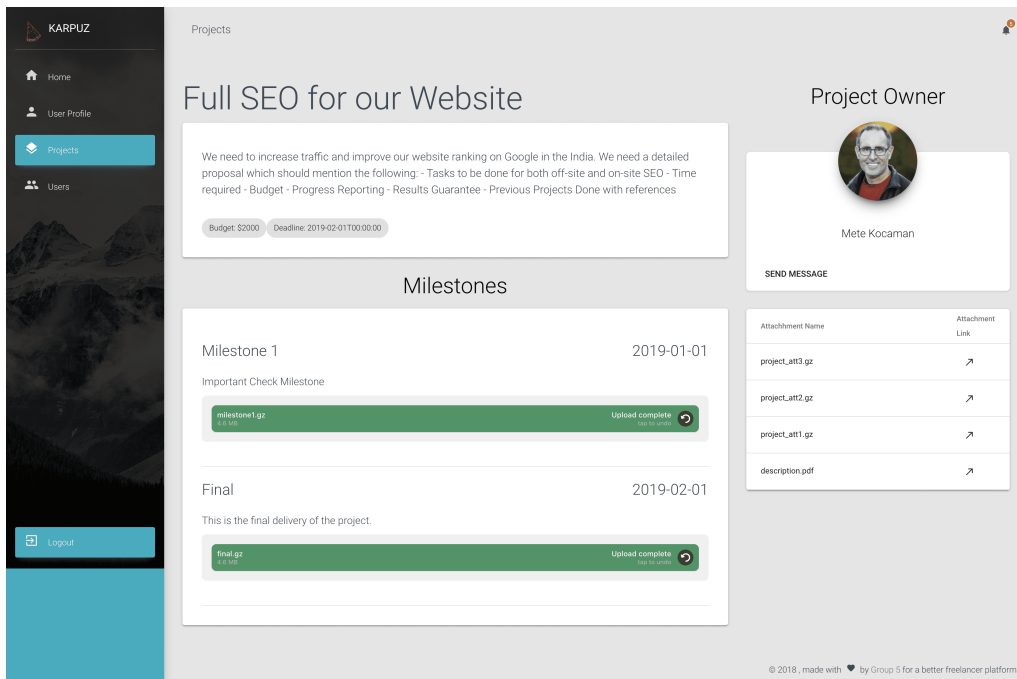


Figure 6: Freelancer Side Project

6.1.6 Client Projects

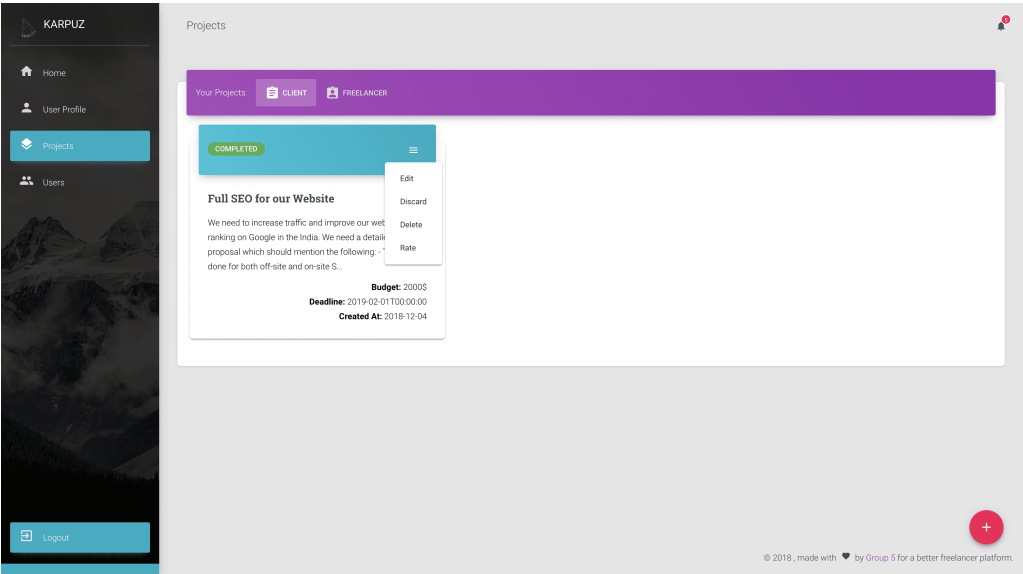


Figure 7: Client Projects

6.1.7 Freelancer Projects

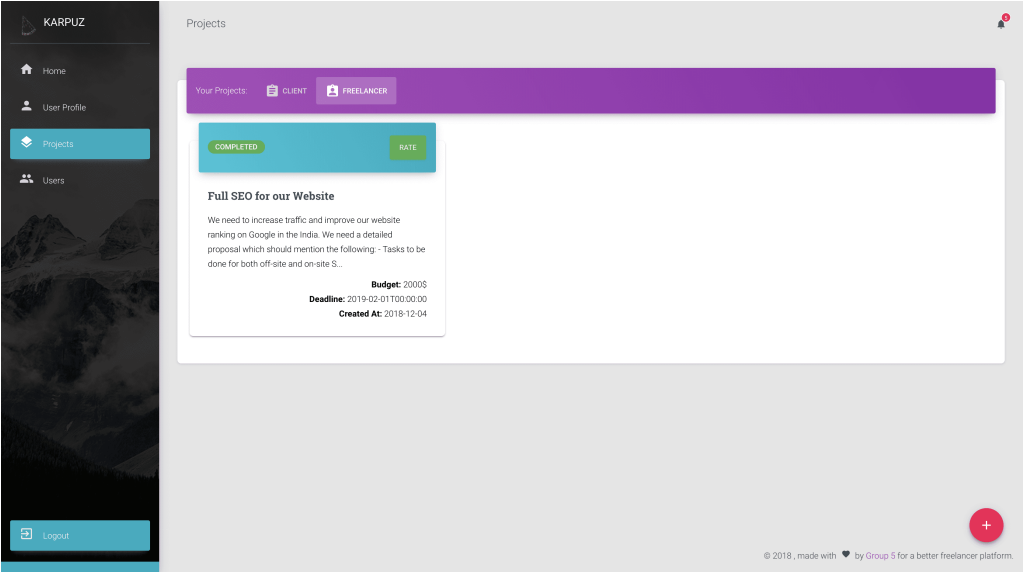


Figure 8: Freelancer Projects

### 6.1.8 User Profile

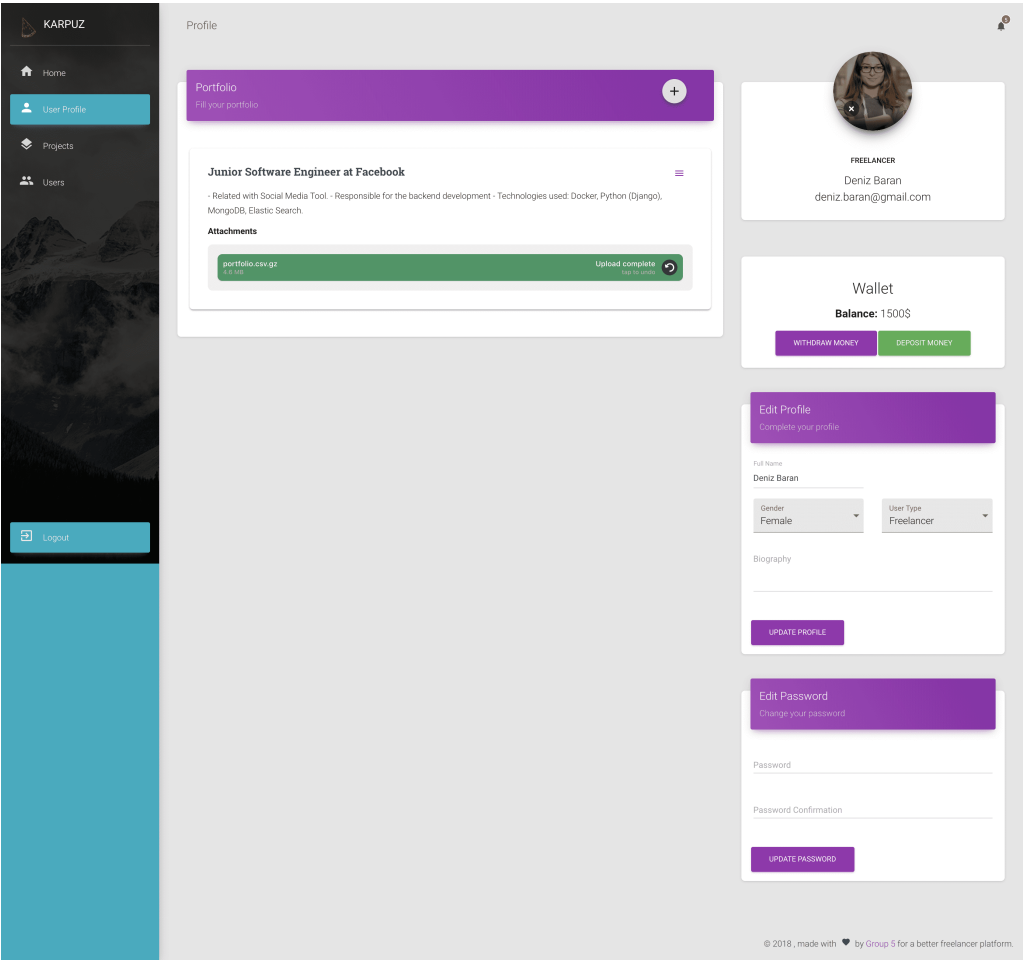


Figure 9: User Profile



### 6.1.9 Other User Profile

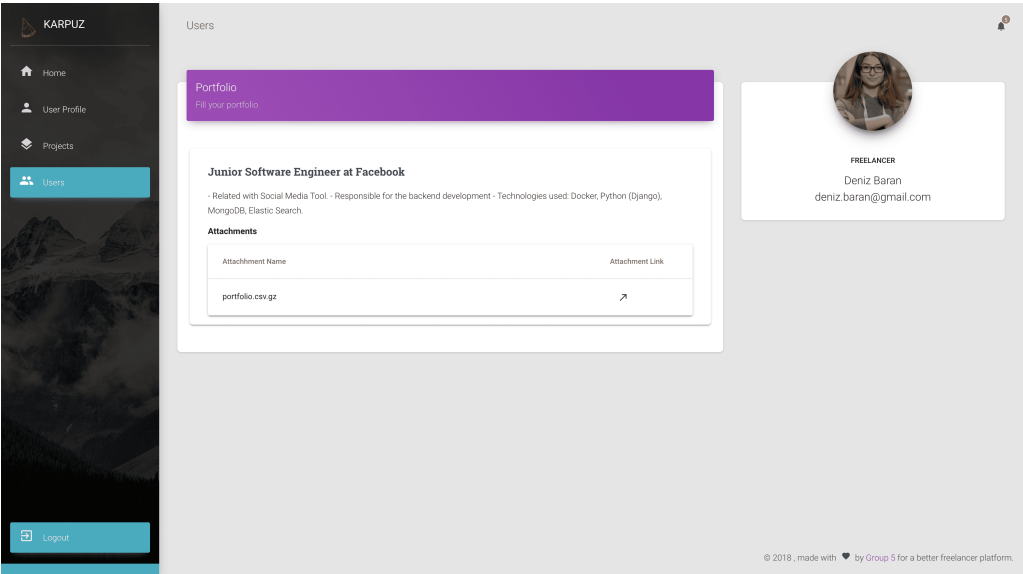


Figure 10: Other User Profile

6.1.10 Freelancer Rating

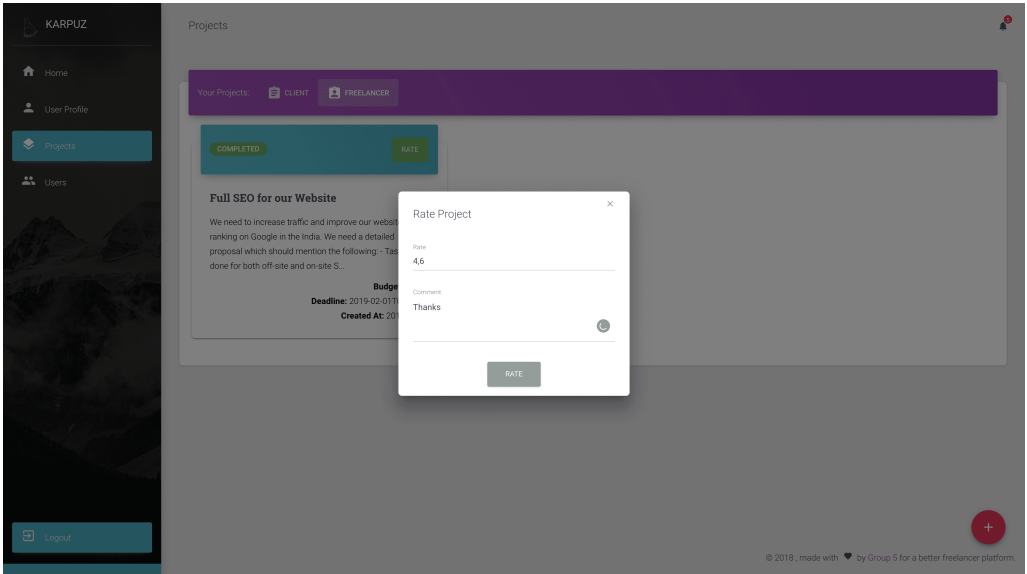


Figure 11: Freelancer Rating

6.1.11 Multiple Milestones

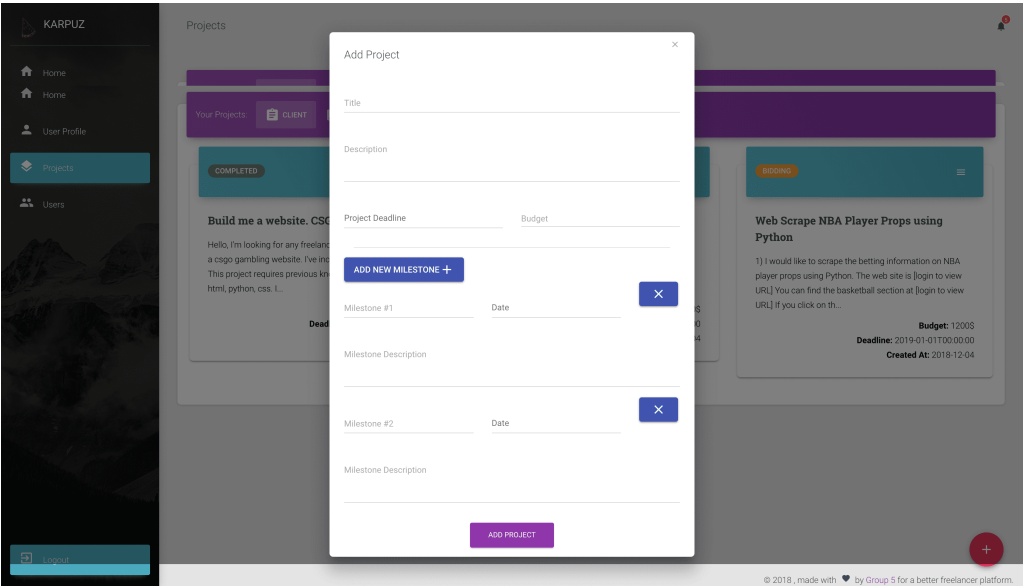


Figure 12: Multiple Milestones

6.2 Android

6.2.1 Home Page

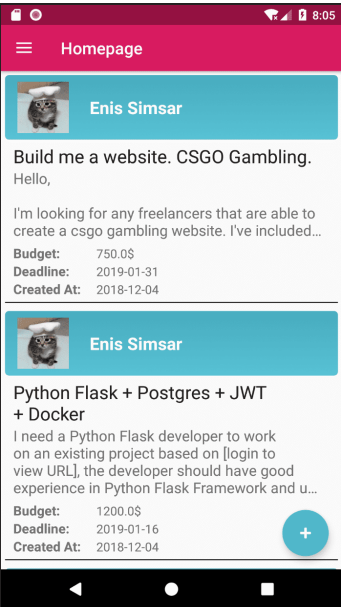
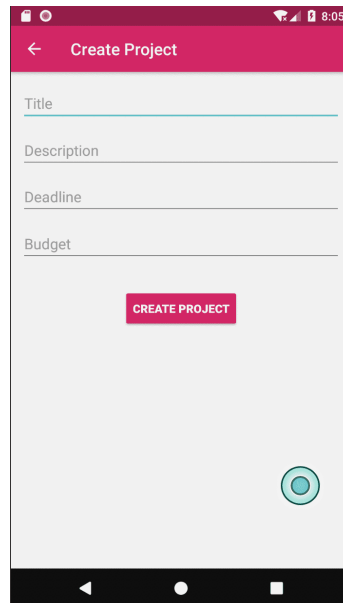


Figure 13: Homepage

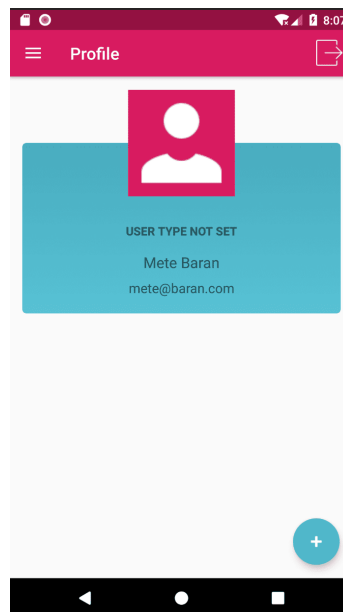
### 6.2.1.1 Project Creation



The screenshot shows a mobile application interface for creating a project. The top bar is red with a white back arrow and the text 'Create Project'. Below the bar, there are four text input fields labeled 'Title', 'Description', 'Deadline', and 'Budget'. A red button with the text 'CREATE PROJECT' is positioned below the input fields. At the bottom right, there is a circular icon with a green outline and a blue center. The bottom of the screen shows the standard Android navigation bar.

Figure 14: Project Creation

### 6.2.1.2 Profile



The screenshot shows a mobile application interface for a user profile. The top bar is red with a white hamburger menu icon, the text 'Profile', and a white document icon. Below the bar, there is a red square icon with a white person silhouette. Underneath the icon, the text 'USER TYPE NOT SET' is displayed. Below that, the name 'Mete Baran' and the email address 'mete@baran.com' are shown. At the bottom right, there is a circular icon with a blue outline and a white plus sign. The bottom of the screen shows the standard Android navigation bar.

Figure 15: User Profile

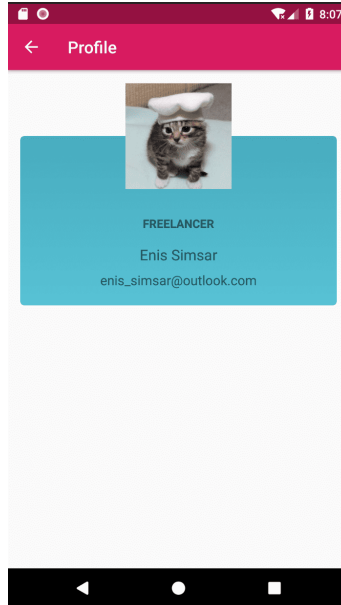


Figure 16: Other User Profile

#### 6.2.1.3 Single Project Detail

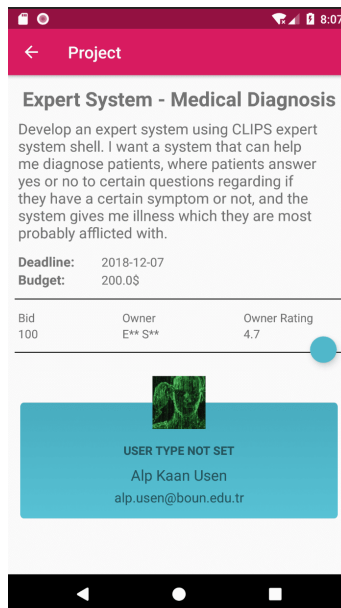


Figure 17: Project Detail

## 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		<b>Implementation for Part I</b>	<b>15 days</b>	<b>10/16/18 8:00 AM</b>	<b>11/5/18 5:00 PM</b>		
2		<b>Backend</b>	<b>12 days</b>	<b>10/16/18 8:00 AM</b>	<b>10/31/18 5:00 PM</b>		
3		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 Güre
4		<b>Auth</b>	<b>5 days</b>	<b>10/22/18 8:00 AM</b>	<b>10/26/18 5:00 PM</b>		
5		Register End Point	2 days	10/22/18 8:00 AM	10/23/18 5:00 PM		Alp Kaan Usen
6		Login End Point	2 days	10/22/18 8:00 AM	10/23/18 5:00 PM		Muhammed Emin Güre
7		Get Profile End Point	2 days	10/22/18 8:00 AM	10/23/18 5:00 PM		Muhammed Emin Ayar
8		Update Profile End Point	3 days	10/24/18 8:00 AM	10/26/18 5:00 PM		Muhammed Emin Güre
9		Logout End Point	3 days	10/24/18 8:00 AM	10/26/18 5:00 PM		Alp Kaan Usen
10		<b>Project</b>	<b>6 days</b>	<b>10/24/18 8:00 AM</b>	<b>10/31/18 5:00 PM</b>		
11		Create Project	3 days	10/24/18 8:00 AM	10/26/18 5:00 PM		Muhammed Emin Ayar
12		Discard Project	3 days	10/24/18 8:00 AM	10/26/18 5:00 PM		Muhammed Emin Güre
13		Get Project	2 days	10/29/18 8:00 AM	10/30/18 5:00 PM		Alp Kaan Usen
14		Finish Project	3 days	10/29/18 8:00 AM	10/31/18 5:00 PM		Muhammed Emin Ayar
15		Get Projects by Filter	2 days	10/30/18 8:00 AM	10/31/18 5:00 PM		Alp Kaan Usen
16		<b>Frontend</b>	<b>12 days</b>	<b>10/16/18 8:00 AM</b>	<b>10/31/18 5:00 PM</b>		
17		Boiler Plate Code	4 days	10/16/18 8:00 AM	10/19/18 5:00 PM		Enis Simsar;Guliz Irem Gokceler;Ihsan Berkan Balaban
18		Landing Page	2 days	10/22/18 8:00 AM	10/23/18 5:00 PM		Enis Simsar
19		<b>Auth</b>	<b>5 days</b>	<b>10/22/18 8:00 AM</b>	<b>10/26/18 5:00 PM</b>		
20		Register Page	2 days	10/22/18 8:00 AM	10/23/18 5:00 PM		Guliz Irem Gokceler
21		Login Page	2 days	10/22/18 8:00 AM	10/23/18 5:00 PM		Ihsan Berkan Balaban
22		Profile Page	3 days	10/24/18 8:00 AM	10/26/18 5:00 PM		Enis Simsar
23		Edit Profile Page	3 days	10/24/18 8:00 AM	10/26/18 5:00 PM		Guliz Irem Gokceler
24		<b>Project</b>	<b>6 days</b>	<b>10/24/18 8:00 AM</b>	<b>10/31/18 5:00 PM</b>		
25		Home Page	3 days	10/24/18 8:00 AM	10/26/18 5:00 PM		Ihsan Berkan Balaban
26		Search Project Page	3 days	10/29/18 8:00 AM	10/31/18 5:00 PM		Enis Simsar
27		Project Page	3 days	10/29/18 8:00 AM	10/31/18 5:00 PM		Guliz Irem Gokceler
28		<b>Android</b>	<b>12 days</b>	<b>10/16/18 8:00 AM</b>	<b>10/31/18 5:00 PM</b>		
29		Boiler Plate Code	4 days	10/16/18 8:00 AM	10/19/18 5:00 PM		Bahadır Oncel;Batuhan Enes Caglayan;Oguzhan Kirlar
30		Landing Page	2 days	10/22/18 8:00 AM	10/23/18 5:00 PM		Bahadır Oncel
31		<b>Auth</b>	<b>5 days</b>	<b>10/22/18 8:00 AM</b>	<b>10/26/18 5:00 PM</b>		
32		Register Page	2 days	10/22/18 8:00 AM	10/23/18 5:00 PM		Batuhan Enes Caglayan
33		Login Page	2 days	10/22/18 8:00 AM	10/23/18 5:00 PM		Oguzhan Kirlar
34		Profile Page	3 days	10/24/18 8:00 AM	10/26/18 5:00 PM		Bahadır Oncel
35		Edit Profile Page	3 days	10/24/18 8:00 AM	10/26/18 5:00 PM		Batuhan Enes Caglayan
36		<b>Project</b>	<b>6 days</b>	<b>10/24/18 8:00 AM</b>	<b>10/31/18 8:00 PM</b>		
37		Home Page	3 days	10/24/18 8:00 AM	10/26/18 5:00 PM		Oguzhan Kirlar
38		Search Project Page	3 days	10/29/18 8:00 AM	10/31/18 5:00 PM		Bahadır Oncel
39		Project Page	3 days	10/29/18 8:00 AM	10/31/18 5:00 PM		Batuhan Enes Caglayan
40		<b>Testing for Above Functionalities</b>	<b>3 days</b>	<b>11/1/18 8:00 AM</b>	<b>11/5/18 5:00 PM</b>		
41		Unit Test for Backend	3 days	11/1/18 8:00 AM	11/5/18 5:00 PM		Each Member
42		SnapshoI Test for Frontend	3 days	11/1/18 8:00 AM	11/5/18 5:00 PM		Enis Simsar;Guliz Irem Gokceler;Ihsan Berkan Balaban
43		Milestone 1	1 day	11/6/18 8:00 AM	11/6/18 5:00 PM	1	
44		<b>Implementation for Part II</b>	<b>23 days</b>	<b>11/7/18 8:00 AM</b>	<b>12/27/18 5:00 PM</b>		Each Member
45		<b>Backend</b>	<b>23 days</b>	<b>11/7/18 8:00 AM</b>	<b>12/27/18 5:00 PM</b>		
46		<b>User</b>	<b>6 days</b>	<b>11/7/18 8:00 AM</b>	<b>11/14/18 5:00 PM</b>		
47		Get Average Rating	3 days	11/7/18 8:00 AM	11/9/18 5:00 PM		Muhammed Emin Ayar
48		Get Ratings	3 days	11/7/18 8:00 AM	11/9/18 5:00 PM		Muhammed Emin Güre
49		Report User	3 days	11/12/18 8:00 AM	11/14/18 5:00 PM		Alp Kaan Usen
50		Delete User	3 days	11/12/18 8:00 AM	11/14/18 5:00 PM		Muhammed Emin Ayar
51		<b>Admin</b>	<b>6 days</b>	<b>11/12/18 8:00 AM</b>	<b>11/19/18 5:00 PM</b>		
52		Get Reports for Projects	3 days	11/12/18 8:00 AM	11/14/18 5:00 PM		Muhammed Emin Güre
53		Get Reports for Users	3 days	11/15/18 8:00 AM	11/19/18 5:00 PM		Alp Kaan Usen
54		Block User	3 days	11/15/18 8:00 AM	11/19/18 5:00 PM		Muhammed Emin Ayar
55		<b>Project</b>	<b>12 days</b>	<b>11/15/18 8:00 AM</b>	<b>11/30/18 5:00 PM</b>		
56		Get Recommended Projects	3 days	11/15/18 8:00 AM	11/19/18 5:00 PM		Muhammed Emin Güre
57		Add Bid	4 days	11/20/18 8:00 AM	11/23/18 5:00 PM		Alp Kaan Usen
58		Accept Bid	4 days	11/20/18 8:00 AM	11/23/18 5:00 PM		Muhammed Emin Ayar
59		Get Project Bids	4 days	11/20/18 8:00 AM	11/23/18 5:00 PM		Muhammed Emin Güre
60		Add Project to Portfolio	5 days	11/26/18 8:00 AM	11/30/18 5:00 PM		Alp Kaan Usen
61		Report Project	5 days	11/26/18 8:00 AM	11/30/18 5:00 PM		Muhammed Emin Ayar
62		<b>Annotation</b>	<b>10 days</b>	<b>11/26/18 8:00 AM</b>	<b>12/7/18 5:00 PM</b>		
63		Annotate Project	5 days	11/26/18 8:00 AM	11/30/18 5:00 PM		Muhammed Emin Güre
64		Delete Annotation for Project	5 days	12/3/18 8:00 AM	12/7/18 5:00 PM		Alp Kaan Usen
65		Annotate Portfolio	5 days	12/3/18 8:00 AM	12/7/18 5:00 PM		Muhammed Emin Ayar
66		Delete Annotate for Portfolio	5 days	12/3/18 8:00 AM	12/7/18 5:00 PM		Muhammed Emin Güre
67		<b>Frontend</b>	<b>23 days</b>	<b>11/7/18 8:00 AM</b>	<b>12/27/18 5:00 PM</b>		
68		<b>User</b>	<b>8 days</b>	<b>11/7/18 8:00 AM</b>	<b>11/16/18 5:00 PM</b>		
69		Show User Rating Info	8 days	11/7/18 8:00 AM	11/16/18 5:00 PM		Enis Simsar
70		Report User Page	8 days	11/7/18 8:00 AM	11/16/18 5:00 PM		Guliz Irem Gokceler
71		<b>Admin</b>	<b>15 days</b>	<b>11/7/18 8:00 AM</b>	<b>11/27/18 5:00 PM</b>		
72		Reported Users Page	8 days	11/7/18 8:00 AM	11/16/18 5:00 PM		Ihsan Berkan Balaban
73		Reported Projects Page	7 days	11/19/18 8:00 AM	11/27/18 5:00 PM		Enis Simsar
74		<b>Project</b>	<b>15 days</b>	<b>11/19/18 8:00 AM</b>	<b>12/7/18 5:00 PM</b>		
75		Show Project Bids	7 days	11/19/18 8:00 AM	11/27/18 5:00 PM		Guliz Irem Gokceler
76		Send Project Bid Popup	7 days	11/19/18 8:00 AM	11/27/18 5:00 PM		Ihsan Berkan Balaban
77		Report Project Page	8 days	11/28/18 8:00 AM	12/7/18 5:00 PM		Enis Simsar
78		<b>Annotation</b>	<b>8 days</b>	<b>11/28/18 8:00 AM</b>	<b>12/7/18 5:00 PM</b>		
79		Project Annotation Support	8 days	11/28/18 8:00 AM	12/7/18 5:00 PM		Guliz Irem Gokceler
80		Portfolio Annotation Support	8 days	11/28/18 8:00 AM	12/7/18 5:00 PM		Ihsan Berkan Balaban
81		<b>Android</b>	<b>23 days</b>	<b>11/7/18 8:00 AM</b>	<b>12/27/18 5:00 PM</b>		
82		<b>User</b>	<b>8 days</b>	<b>11/7/18 8:00 AM</b>	<b>11/16/18 5:00 PM</b>		
83		Show User Rating Info	8 days	11/7/18 8:00 AM	11/16/18 5:00 PM		Bahadır Oncel
84		Report User Page	8 days	11/7/18 8:00 AM	11/16/18 5:00 PM		Batuhan Enes Caglayan
85		<b>Admin</b>	<b>15 days</b>	<b>11/7/18 8:00 AM</b>	<b>11/27/18 5:00 PM</b>		
86		Reported User Page	8 days	11/7/18 8:00 AM	11/16/18 5:00 PM		Oguzhan Kirlar
87		Reported Projects Page	7 days	11/19/18 8:00 AM	11/27/18 5:00 PM		Bahadır Oncel
88		<b>Project</b>	<b>15 days</b>	<b>11/19/18 8:00 AM</b>	<b>12/7/18 5:00 PM</b>		
89		Show Project Bids	7 days	11/19/18 8:00 AM	11/27/18 5:00 PM		Batuhan Enes Caglayan
90		Send Project Bid Popup	7 days	11/19/18 8:00 AM	11/27/18 5:00 PM		Oguzhan Kirlar
91		Report Project Page	8 days	11/28/18 8:00 AM	12/7/18 5:00 PM		Bahadır Oncel
92		<b>Testing</b>	<b>6 days</b>	<b>12/10/18 8:00 AM</b>	<b>12/17/18 5:00 PM</b>	1;44	
93		Unit Tests	2 days	12/10/18 8:00 AM	12/11/18 5:00 PM		Each Member
94		User Tests	2 days	12/12/18 8:00 AM	12/13/18 5:00 PM		Each Member
95		Debugging	2 days	12/14/18 8:00 AM	12/17/18 5:00 PM		Each Member
96		Milestone 2	1 day	12/18/18 8:00 AM	12/18/18 5:00 PM	92	Each Member
97		Launch Presentation	1 day	1/2/19 8:00 AM	1/2/19 5:00 PM		Each Member
CmpE 451 - Group 5							

Figure 18: Project Plan

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

- 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 II 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 continued with React for Frontend project. We changed the development branch, combining all develop branches in to 'develop' branch. We added necessary packages to 'Package.json'. We tried to break up all pages into small pieces and we implemented all small parts for the pages. We created issues for those small pieces, features such as API connection and bugs. We implemented our codes on the different branches. For example, we develop the design of the portfolio section in the 'f-portfolio' branch and when the implementations is finished, we created a pull request to develop branch that is the main development branch for project. Our review process were not enough, so we will try improve it. Our merged codes are automatically deployed with Jenkins when they went to 'master' branch.

### 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. We're using Kotlin and we became a fairly big fan of it. It's intuitive and readable as opposed to Java. We're fairly happy that Android got itself a new lover.

#### 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. 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 ProjectLibre 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**

PR reviewing was not done properly for the first 2 milestones we've delivered. We hope to achieve better in the coming weeks, but we don't see a clear way forward with this issue, we simply don't have enough heads for work.

### **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 still 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.