



# CmpE 352 - Report

Karpuz - Freelancer Platform

Group 5

May 23, 2018

# Contents

<b>1 Executive Summary</b>	<b>3</b>
1.1 Project Introduction . . . . .	3
1.2 Work Done in the First Half . . . . .	3
1.3 Road Ahead . . . . .	4
<b>2 List and Status of Deliverables</b>	<b>4</b>
<b>3 Evaluation of the Status of Deliverables and Their Impact on Plan</b>	<b>4</b>
<b>4 Coding Work</b>	<b>5</b>
<b>5 Requirements</b>	<b>5</b>
5.1 Glossary . . . . .	5
5.2 Functional Requirements . . . . .	7
5.2.1 User Requirements . . . . .	7
5.2.1.1 Guest User . . . . .	7
5.2.1.2 Admin User . . . . .	7
5.2.1.3 Registered User . . . . .	7
5.2.2 System Requirements . . . . .	9
5.2.2.1 Search . . . . .	9
5.2.2.2 Recommendation . . . . .	9
5.2.2.3 Notification . . . . .	9
5.2.2.4 Payment . . . . .	9
5.2.2.5 Wallet . . . . .	9
5.2.2.6 Landing Page . . . . .	10
5.3 Non-functional Requirements . . . . .	10
5.3.1 Usability . . . . .	10
5.3.2 Availability . . . . .	10
5.3.3 Annotations . . . . .	10
5.3.4 Third Party Integration . . . . .	10
5.3.5 Security . . . . .	10
5.3.6 Reliability . . . . .	11
5.3.7 Scalability . . . . .	11
5.3.8 Performance . . . . .	11
5.3.9 Legal . . . . .	11
<b>6 Mockups</b>	<b>11</b>
6.1 Personas . . . . .	11
6.1.1 Berkcan - The Backend Developer . . . . .	11
6.1.2 Mete - The Ambitious Entrepreneur . . . . .	12
6.1.3 Nazli - The Extrovert Designer . . . . .	13
6.2 User Stories & Mockups . . . . .	14
6.2.1 User Story 1 . . . . .	14
6.2.2 User Story 2 . . . . .	17
6.2.3 User Story 3 . . . . .	21
<b>7 Software Design</b>	<b>24</b>
7.1 Class Diagram . . . . .	24
7.2 Sequence Diagrams . . . . .	24
<b>8 User Tests</b>	<b>29</b>
<b>9 Project Plan</b>	<b>38</b>
<b>10 Software Development Infrastructure</b>	<b>40</b>

<b>11 Evaluation</b>	<b>41</b>
11.1 Tools . . . . .	41
11.2 Project Management . . . . .	41

# 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 in the First Half

In the first half of the Software Engineering course, we were assigned to a group of 10 people, and most of us didn't know each other at the beginning. In high spirits we arranged our first meeting and shook hands. It was awesome that at the first 3-4 meetings everybody was there. We were thrown homework after homework but we completed them all with ferocity and joy. After some time, the attendance linearly decreased and the effort invested too. This was expected, since weather is so nice and everyone is having fun outside, but we weren't about to back down. As the last effort, we're collectively writing this report and can't wait for the second half of the course (after a well earned summer break, of course).

Our group had all kinds of junior developers - some of us worked in startups and knew what it was all about, some of us didn't have any clue about the software development pipeline, and some of us were in the middle. This made our group the perfect group for learning, we had all kinds of students. The experienced ones wisely guided the other ones around the area, while the inexperienced ones quickly picked up new stuff. The contribution was not all equal as expected, but everyone put equal effort into the project and the outcome satisfied the whole team.

In the first half of the course, the team went through 11 meetings. Not every meeting had a specific homework assigned to it but most of them did. Details for individual meetings can be found in the repo wiki. The team successfully completed crafting, including but not limited to following:

- Model Repos
- Communication Plan
- Project Requirements
- Project Plan
- Project Guidelines (Issue, Comment, Label, Git guides)
- Readme Page that evolved in time
- Project Name
- Project Logo
- Personas
- User Stories
- Mockups
- Test Cases

### 1.3 Road Ahead

We're periodically updating our project plan and plan to not drift away too much. But as a summary, Karpuz team aims to (after a long and well earned summer break) implement the Karpuz platform, with its backend, web app and Android counterparts. Karpuz team aims to turn the theoretical knowledge obtained in the first half of the course to practical knowledge, but doing Agile development and User Experience Centric Design (Lean UX). We know dividing development work will not be easy, and we know that we PR reviewing experience we had from the Twitter API project was not enough. But our experienced members are more than happy to bear with and teach the inexperienced ones all the development pipeline, since it's a nonzero sum game. The experienced ones will put deeper insight to their skills by teaching to their peers, and the inexperienced ones will learn better approaches from their more experienced peers. As a team, we believe we'll create a product that people will love and use with ease, and we're not intimidated by the course load we have. Our team did value communications over processes and quality over quantity in the first half of the course, and we aim to keep our ways for the development part.

## 2 List and Status of Deliverables

Name	Delivery Date	Delivered
Project Plan	Feb 26, 2018	✓
User Stories & Mockups	March 5, 2018	✓
Requirements	March 12, 2018	✓
Design Diagrams & Test Cases	March 26/ Apr 2, 2018	✓
Karpuz Twitter App	Apr 13/ May 7, 2018	✓

Table 1: List All deliverables and Their Status

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

- **Project Plan:** We made a realistic project plan and complied with it almost all of the time. Balanced distribution of tasks and appropriate deadlines gave us a good outline to follow. We may need to alter the original project plan, but a major change is not needed. Without planning things get messy especially when you have a team of 10 people.
- **User Stories & Mockups:** All the team members contributed to mockups and despite that we have managed to come up with very similar looking mockups. We have covered the most important functionalities of the platform with the user stories. User stories and mockups helped us to look into our future users' eyes. To make the platform usable we needed to think as we are to use this platform.
- **Requirements:** We have made a lot of revisions to come up with the first version and nearly finalized the requirements, other than any changes that might be needed during the course of the project. Requirements document is the foundation stone of the project. We couldn't and can't go forward without this document. Therefore it is safe to say that this document has the most impact in our project.
- **Design Diagrams & Test Cases:** We communicated well during the process of designing class and sequence diagrams, and the results is satisfactory. We could have had more people working on the test cases, however we still managed to cover most of the essential test cases.
- **Karpuz Twitter App:** We could have better contribution from all team members, still some of the functions that were implemented was impressive. This app helped us create an outline for implementing and deploying our platform.

## 4 Coding Work

Team Member	Contributions
Enis Simsar	Searched good template for Python Tornado Web Framework. Created a boiler plate codes. Research about Twitter REST API (in the past, I used Stream API and little REST API). Wrote Singleton for Twitter API calls. Implemented a location based searching tweets. Reviewed my teammates Pull Requests and also they reviewed my PRs. They reported some bugs and I also found some bugs. Then, solved them. Deployed our app to the Digital Ocean Instance. Created a LaTeX template for this report.
Bahadir Öncel	Implemented functionality that when two valid twitter usernames given, the app finds the common followers of the two users. I've never used Python Tornado Framework before so that was a good start. Reviewed and commented PR's, communicated some good approaches to teammates.
Alp Kaan Usen	Implemented a Twitter API function that takes the user screen name as input and shows that user's timeline, using Enis' template. I had not used RESTful APIs or the Tornado framework before, so I gained some experience with them. Reviewed some of the pull requests.
Muhammed Emin Güre	Implemented a new function that finds the common followings of two valid users who have followings less than 5000 due to Twitter API rate limits. I found a css template to show twitter profiles in a good design and made some arrangements to make it even better. It was the first time that I used Python to create a Web app and this gave me a new look to this incredible programming language. Reviewed almost all PR's and reported some bugs.
Muhammed Emin Ayar	Implemented a new function that finds the latest likes of a user when the user's screen name is given. It also takes the number of tweets that user wants to see on the result page. I used twitter's javascript to show tweets. When an invalid screen name is given as input or searched user has a private profile the result page simply prints error. When merging my pull request there were some conflicts and I fixed them.
Batuhan Enes Çağlayan	Implemented a function that searches users based on a query that consists of the name of user. Its result is limited to 18 users since I wanted a symmetrical interface all over the page. This was some quality experience for me since I have never used any APIs in my programming career before but have a project in my mind that cannot be realized without the use of an API, let alone the Python experience I have gained.
Ihsan Berkan Balaban	Implemented search tweets "Enter a query and get the tweets!"
Güliz İrem Gökçeler	-
Oğuzhan Kırlar	Followed up requirements regularly
Utku Alhan	-

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 be either Freelancers or Clients, explained below.
- **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 that wants to take on Projects for Clients in exchange for payment.
- **Client:** A user that wants post Projects to be done by Freelancers.
- **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 bid 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.
- **Profile Page:** A page that contains information about a Freelancer and a Client.
- **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 , that the Freelancer chooses to display on their Profile Page.
- **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.
- **Home Page:** The page that is shown to the Registered User when they log in.

## 5.2 Functional Requirements

### 5.2.1 User Requirements

#### 5.2.1.1 Guest User

- **5.2.1.1.1.** Guest users shall be able to sign up anytime.
- **5.2.1.1.2.** Guest users shall be able to search for projects.
  - **5.2.1.1.2.1.** Guest users should be able to filter the searching results according to the rating and the price.

#### 5.2.1.2 Admin User

- **5.2.1.2.1.** Admin users shall be able to view any registered user account, project, rating, comment, and report.
- **5.2.1.2.2.** Admin users shall be able to block or remove any registered user account.
- **5.2.1.2.3.** Admin users shall be able to view messages between registered users.
- **5.2.1.2.4.** Admin users shall be able to message any registered user.
- **5.2.1.2.5.** Admin users shall be able to edit, block or remove any project.
- **5.2.1.2.6.** Admin users shall be able to search for projects.
- **5.2.1.2.7.** Admin users should be able to generate a report about projects, freelancers, and clients.

#### 5.2.1.3 Registered User

- **5.2.1.3.1.** Registered users shall be able to log in.
- **5.2.1.3.2.** Registered users shall be able to update their personal information and password.
- **5.2.1.3.3.** Registered users shall have a virtual wallet.
  - **5.2.1.3.3.1** Registered users shall be able to deposit money to and withdraw money from their virtual wallet.
- **5.2.1.3.4.** Registered users shall be able to search for projects and users.
  - **5.2.1.3.4.1** Registered users should be able to filter the searching results according to the rating and the price.
- **5.2.1.3.5.** Registered users shall be able to view a project in its entirety
- **5.2.1.3.6.** Registered users shall be able to view any registered user profile page.
- **5.2.1.3.7.** Registered users shall be able to message any registered user.
- **5.2.1.3.8.** Registered users shall be able to report any registered user account and any projects.
- **5.2.1.3.9.** Registered users on their home page:
  - **5.2.1.3.9.1** Shall be able to access their profile page via a link.
  - **5.2.1.3.9.2** Shall be able to search for projects and users.
  - **5.2.1.3.9.3** Shall be able to see notifications.
  - **5.2.1.3.9.4** Shall be able to access their wallet.
- **5.2.1.3.10.** Registered users shall be able to create a profile page.
  - **5.2.1.3.10.1** Registered users shall be able to add a profile picture to their profile page.

- **5.2.1.3.10.2** Registered users shall be able to share their personal information on their profile page.
  - **5.2.1.3.10.2.1** Registered users shall be able to share their name and surname on their profile page.
  - **5.2.1.3.10.2.2** Registered users shall be able to share their age on their profile page.
  - **5.2.1.3.10.2.3** Registered users shall be able to share their gender on their profile page.
  - **5.2.1.3.10.2.4** Registered users shall be able to share a short text about themselves on their profile page.
- **5.2.1.3.11.** Registered users shall be able to switch between two user types, which are freelancer and client.
  - **5.2.1.3.11.1 Freelancer**
    - **5.2.1.3.11.1.1** Freelancers shall be able to create a profile page.
      - **5.2.1.3.11.1.1.1** Freelancers shall be able to add semantic tags to their profile page for their interests and their skills.
      - **5.2.1.3.11.1.1.2** Freelancers shall be able to create a portfolio to be added to their profile page upon account creation.
    - **5.2.1.3.11.1.2** Freelancers shall be able to view their ongoing projects.
    - **5.2.1.3.11.1.3** Freelancers shall be only able to rate and comment on a client after a project is finished.
    - **5.2.1.3.11.1.4** Freelancers shall be able to receive their payments via the virtual wallet.
    - **5.2.1.3.11.1.5** Freelancers shall be able to choose to add the project to his portfolio after a project is finished.
    - **5.2.1.3.11.1.6** Freelancers should be able to connect their LinkedIn profile to their account.
    - **5.2.1.3.11.1.7** Freelancers shall be able to bid on a project.
  - **5.2.1.3.11.2 Client**
    - **5.2.1.3.11.2.1** Clients on their home page:
      - **5.2.1.3.11.2.1.1** Shall be able to create a project.
        - **5.2.1.3.11.2.1.1.1** Clients shall be able to choose a topic for their project.
        - **5.2.1.3.11.2.1.1.2** Clients shall be able to choose a deadline for their project.
        - **5.2.1.3.11.2.1.1.3** Clients shall be able to specify an estimated budget for their project.
        - **5.2.1.3.11.2.1.1.4** Clients shall be able to upload attachments about their project.
        - **5.2.1.3.11.2.1.1.5** Clients shall be able to associate semantic tags with their project.
      - **5.2.1.3.11.2.1.2** Shall be able to view projects with same or similar semantic tags while creating a project.
      - **5.2.1.3.11.2.2** Clients shall be able to choose a freelancer's bid for a project.
      - **5.2.1.3.11.2.3** Clients shall be able to send a push notification of their project to a specific freelancer.
      - **5.2.1.3.11.2.4** Clients shall be able to change the visibility of their projects to guest users, choosing either public or private.
      - **5.2.1.3.11.2.5** Clients shall be able to pay freelancers with the balance of their virtual wallet.
      - **5.2.1.3.11.2.6** Clients shall be able to rate and comment on a freelancer only after their project has been completed by that freelancer.
      - **5.2.1.3.11.2.7** 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.
  - **5.2.2.1.1.1.** The system shall rank the results for project searches according to client name.
  - **5.2.2.1.1.2.** The system shall rank the results for project searches according to topic.
  - **5.2.2.1.1.3.** The system shall rank the results for project searches according to associated semantic tags.
  - **5.2.2.1.1.4.** The system shall rank the results for project searches according to estimated budget.
  - **5.2.2.1.1.5.** The system shall rank the results for project searches according to project status.
  - **5.2.2.1.1.6.** The system shall rank the results for project searches according to deadline.
- **5.2.2.1.2.** The system shall rank the results for user searches.
  - **5.2.2.1.2.1.** The system shall rank the results for user searches according to user name.
  - **5.2.2.1.2.2.** The system shall rank the results for user searches according to associated semantic tags.
- **5.2.2.1.2.** 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. The system shall show them on clients' home page.
- **5.2.2.2.2.** The system shall generate project recommendations for freelancers according to the portfolio information and semantic tags they provided. The system shall show them on freelancers' home page.

### **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 Landing Page**

- **5.2.2.6.1.** The system shall show the total number of projects, freelancers, clients and total earnings of freelancers.
- **5.2.2.6.2.** The system shall list public projects.
- **5.2.2.6.3.** The system shall list most bided projects.
- **5.2.2.6.4.** The system shall exhibit user testimonials.

## **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**

- **5.3.2.1** The system shall have a Web application that supports:
  - **5.3.2.1.1** Chrome browser version 63 and up
  - **5.3.2.1.2** Firefox browser version 58 and up
  - **5.3.2.1.3** Opera browser version 50 and up
- **5.3.2.2** The system shall have an Android application that supports version 6.0 Android Marshmallow and later (used by 57.7% of Android Users according to February 5, 2018, Android Dashboard).

### **5.3.3 Annotations**

- **5.3.3.1** 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% uptime

### 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 Mockups

### 6.1 Personas

#### 6.1.1 Berkcan - The Backend Developer



## **Demographics**

- Bogazici University CMPE Senior Student
- Loves Javascript / Angular.js
- Likes Python / Django
- 21 years old
- Registered to the system as a freelancer

## **Goals**

- Wants to earn money from freelancer projects
- Wants to find projects that fit his portfolio

## **Corresponding User Story**

As a freelancer, Berkcan can find appropriate projects from searching python, backend tags so that he can bid on the projects

### **Definition of Done (Acceptance Criteria)**

- Freelancer is able to search with tags (5.2.1.3.4, 5.2.2.1.3)
- Freelancer is able to filter search results (5.2.1.3.4.1)
- Freelancer is able to see project's budget (5.2.1.3.5)
- Freelancer is able to bid on the project (5.2.1.3.11.1.7)

## **6.1.2 Mete - The Ambitious Entrepreneur**

### **Demographics**

- Bogazici University Management 2016 Graduate
- Has limited budget for projects
- 26 years old
- Registered to the system as a client

### **Goals**

- Wants to get his unicorn app implemented
- Wants to find a cheap freelancer for his startup

## **Corresponding User Story**

As a client, Mete can post his project and get bids to that he can choose a freelancer to get it implemented, finally finish the project, give a comment and rate to the freelancer and display his wallet to pay money.

### **Definition of Done (Acceptance Criteria)**

- Client is able to post projects (5.2.1.3.11.2.1.1)
- Client is able to get bids for his projects (5.2.1.3.11.2.2)
- Client is able to see the profiles of the freelancers (5.2.1.3.6)
- Client is able to finish the project and rate & comment to the Freelancer (5.2.1.3.11.2.7)
- Client is able to see his transactions in his wallet (5.2.2.5.1)



#### 6.1.3 Nazli - The Extrovert Designer

##### Demographics

- Unemployed
- 27 years old
- Talented UI/UX designer

- Freelancer

### Goals

- Wants to find a platform to take jobs as a freelancer
- Wants this platform to be trustworthy for payments
- Wants this platform to be trustworthy for project management

### Corresponding User Story

As a guest, Nazli can see statistics about the system, most bided projects, some user testimonials, and payment protocols on the landing page so that she can trust the platform and sign up

#### Definition of Done (Acceptance Criteria)

- Guest is able to see system statistics (5.2.1.1)
  - Guest is able to see a total number of projects
  - Guest is able to see a total number of clients
  - Guest is able to see a total number of freelancers
- Guest is able to see most bided projects (5.2.1.4)
- Guest is able to see implemented payment protocols (5.2.1.5)
- Guest is able to see user testimonials (5.2.1.6)

## 6.2 User Stories & Mockups

### 6.2.1 User Story 1

As a freelancer, I can find appropriate projects from searching tags so that I can bid to the projects.

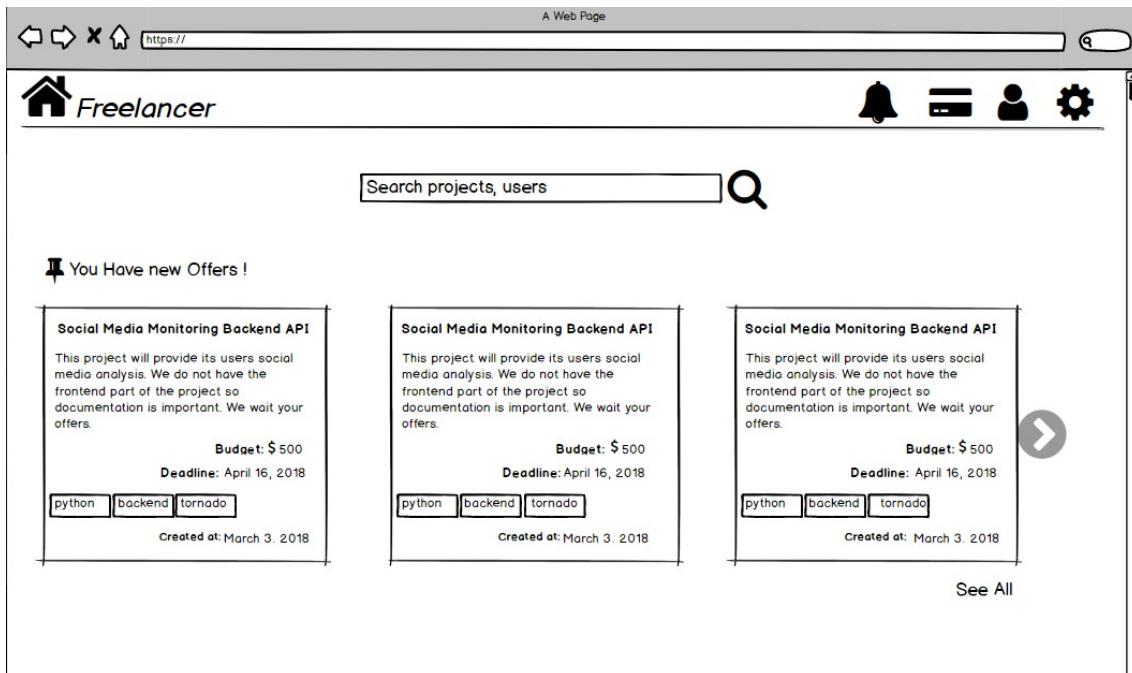
#### Flow

- Freelancer enters to the website. Enters their credentials to login.

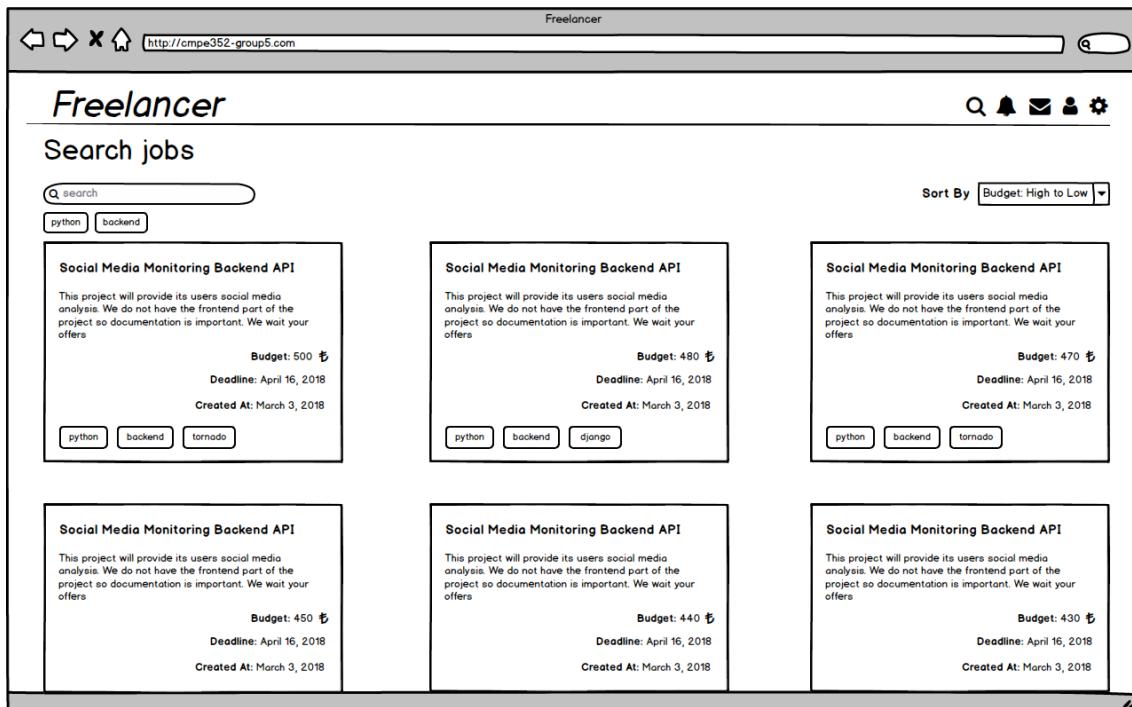
The mockup shows a browser window with the following details:

- Header:** Shows the title "Freelancer" and the URL "https://cmpe352-group5".
- Content Area:**
  - Log In Form:** Contains fields for "E-mail", "Password", and "Re-enter Password", along with a "Sign Up" button.
  - Search Bar:** A placeholder text "Search now: The new Instagram" is visible.
  - Statistics:** Three summary boxes at the bottom:
    - "Projects Completed": 1247
    - "Clients": 2156
    - "Freelancers": 3592

- Freelancer is greeted by the homepage. They can see the hottest projects.



- Freelancer searches for a project in their capacity.



- Freelancer clicks on a project and sees the details of it. Here he can bid to the project and see who bided to it before.

Freelancer

http://cmpe352-group5.com

# Social Media Monitoring Backend API

Budget: 500      Deadline: April 16, 2018

This project will provide its users social media analysis. We do not have the frontend part of the project so documentation is important. We wait your offers.

Details

We are a small but visionary internet company and we want to be effective on social media platforms. We need to monitor our social media accounts and activities in detail. We use Facebook, Twitter, Instagram and LinkedIn. So you need to work with these platforms' APIs. We want you to have experience on APIs especially on these specific ones. Task will be coding only backend but it must be well documented. Deadline is strict, but budget may be negotiated.

python    backend    tornado    api

Project Owner

Mete Kocaman  
CEO at Sosyal Bilisim    [Send Message](#)

Bids

Mert Kizil    450    ★★★★☆

Bid Price: 550

Note: I am an experienced backend dev!

[Place Bid](#)

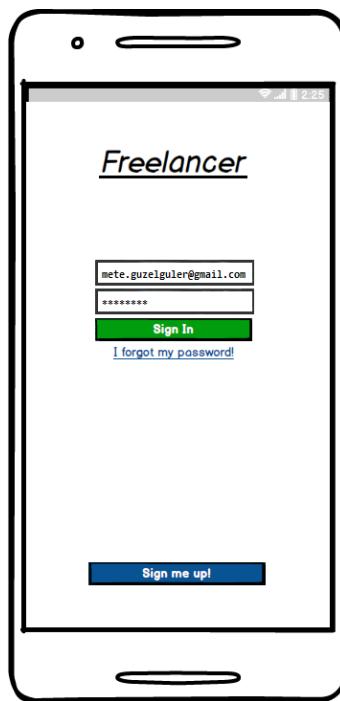
The screenshot shows a web browser window for a 'Freelancer' platform. The URL is http://cmpe352-group5.com. The main content is a project listing for a 'Social Media Monitoring Backend API'. It includes a budget of 500 and a deadline of April 16, 2018. The description details the need to monitor social media accounts (Facebook, Twitter, Instagram, LinkedIn) using their respective APIs. The task involves only backend coding and requires good documentation. A bid from 'Mert Kizil' is displayed, offering 450 with a note stating they are an experienced backend developer. The interface includes standard browser controls, a search bar, and user profile links for the project owner.

### 6.2.2 User Story 2

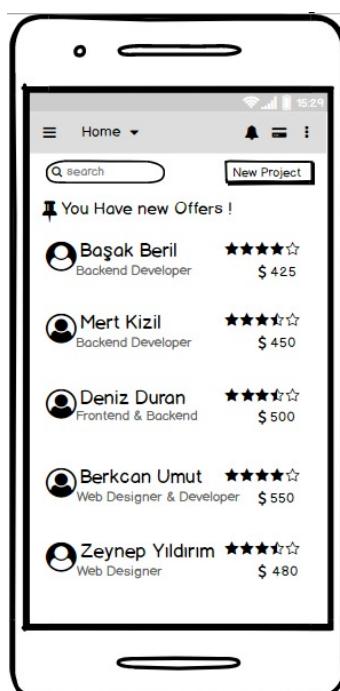
As a client, I can post his project and get bids to that I can choose a freelancer to get it implemented, finally finish the project, give a comment and rate to the freelancer and display his wallet to pay money.

#### Flow

- Client opens the Android app (the first time). They are greeted with the login page.



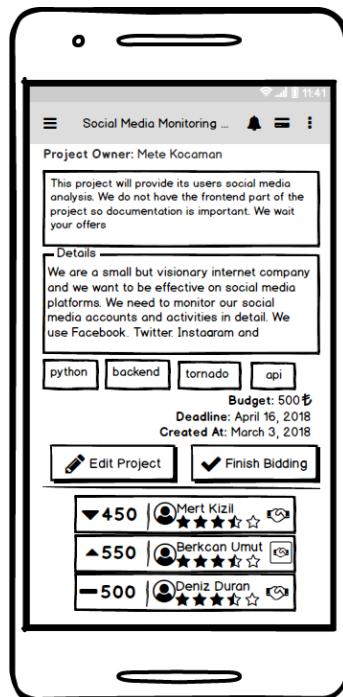
- Client logs in using their credentials. They see the homepage customized by them with freelancers based on their current and previous posted projects.



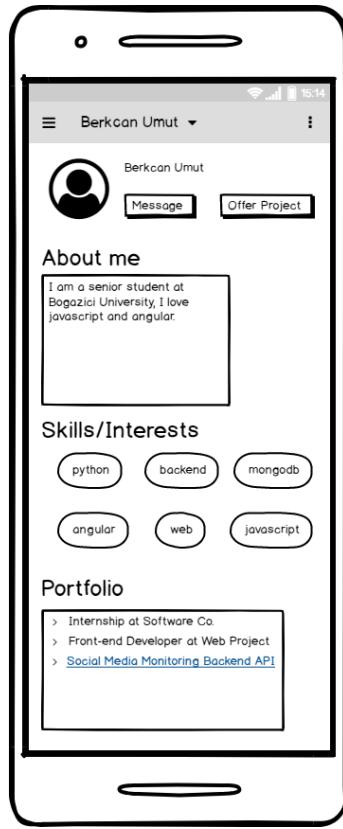
- Client decides to create a project. Opens the project creation page and enters the details of the project.



- Client created the project and collects the bids to it.



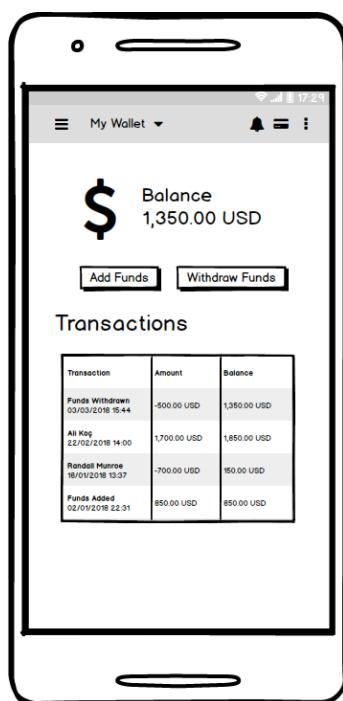
- Client taps one of the bidders profiles to see it.



- Client completes the project and they're taken to the profile rating screen. Here they can rate and comment the freelancer.



- Client displays their wallet to see the amount they have usable.

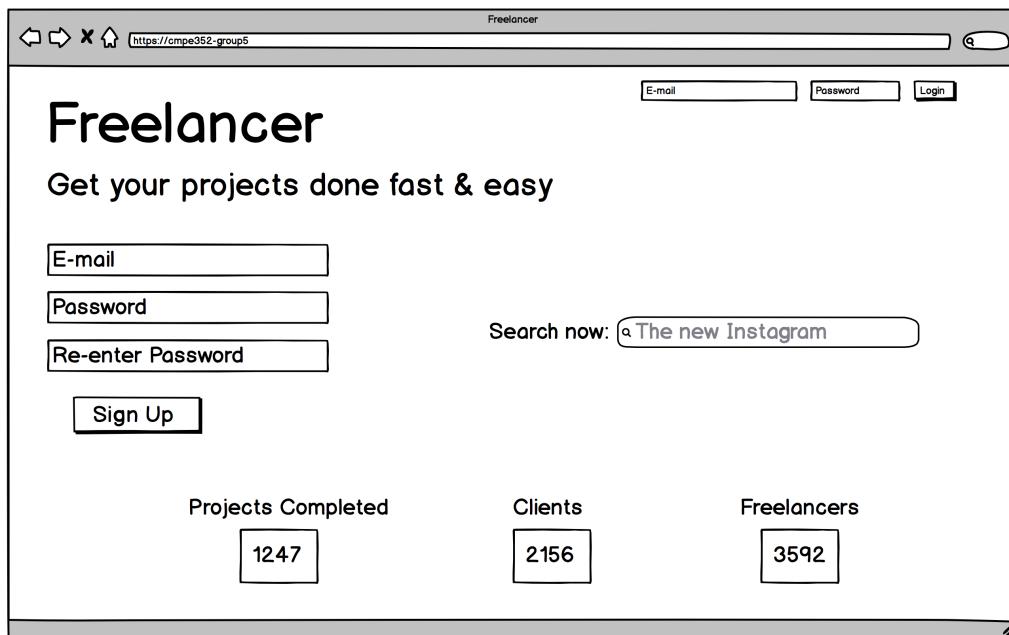


### 6.2.3 User Story 3

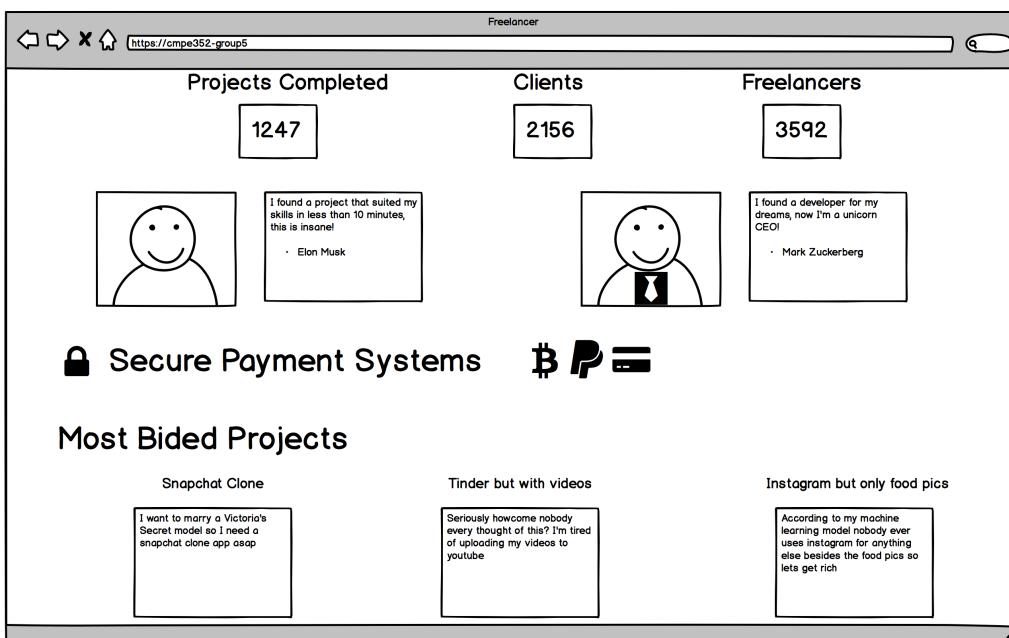
As a guest, I can see statistics about the system, most bided projects, some user testimonials, and payment protocols on the landing page so that I can trust the platform and sign up.

#### Flow

- The guest enters the website and is greeted by the landing page. They can see the signup form, search bar and some statistics about the platform.



- Scrolling down, the guest can see some user testimonials, implemented payment methods and the most bided projects.



- Guest searches for some projects to see the platform's structure

The screenshot shows a search results page for 'Freelancer'. The search bar contains 'python backend'. The results are sorted by 'Budget: High to Low'. Each project listing includes the title 'Social Media Monitoring Backend API', a brief description, budget (e.g., 500 ₺), deadline (e.g., April 16, 2018), creation date (e.g., March 3, 2018), and tags (e.g., python, backend, tornado or django).

Project Title	Budget	Deadline	Created At	Tags
Social Media Monitoring Backend API	500 ₺	April 16, 2018	March 3, 2018	python, backend, tornado
Social Media Monitoring Backend API	480 ₺	April 16, 2018	March 3, 2018	python, backend, django
Social Media Monitoring Backend API	470 ₺	April 16, 2018	March 3, 2018	python, backend, tornado
Social Media Monitoring Backend API	450 ₺	April 16, 2018	March 3, 2018	python, backend, tornado
Social Media Monitoring Backend API	440 ₺	April 16, 2018	March 3, 2018	python, backend, tornado
Social Media Monitoring Backend API	430 ₺	April 16, 2018	March 3, 2018	python, backend, tornado

- Guest views some projects' details

The screenshot shows a detailed view of a project titled 'Social Media Monitoring Backend API'. The project has a budget of 500 ₺ and a deadline of April 16, 2018. The description states: 'This project will provide its users social media analysis. We do not have the frontend part of the project so documentation is important. We wait your offers.' The 'Details' section provides more information: 'We are a small but visionary internet company and we want to be effective on social media platforms. We need to monitor our social media accounts and activities in detail. We use Facebook, Twitter, Instagram and LinkedIn. So you need to work with these platforms' APIs. We want you to have experience on APIs especially on these specific ones. Task will be coding only backend but it must be well documented. Deadline is strict, but budget may be negotiated.' The project is tagged with python, backend, tornado, and api.

**Project Owner:** Mete Kocaman, CEO at Sosyal Bilisim, Send Message

**Bids:**

User	Bid	Rating	Action
Mert Kizil	450	★★★★☆	Handshake icon

- Guest decided to register to the platform.

The screenshot shows a web browser window for a 'Freelancer' platform. The URL in the address bar is <https://cmpe352-group5>. The page has a header with the word 'Freelancer'. On the left, there are input fields for 'Email' (containing 'nazli\_angel@hotmail.com'), 'Password' (containing '\*\*\*\*\*'), and 'Confirm Password' (containing '\*\*\*\*\*'). To the right of these fields is a 'Login' button. Below the input fields is a 'Sign Up' button. A search bar at the top right contains the placeholder text 'Search now: The new Instagram'. At the bottom of the page, there are three summary boxes: 'Projects Completed' (1247), 'Clients' (2156), and 'Freelancers' (3592).

Projects Completed	Clients	Freelancers
1247	2156	3592

# 7 Software Design

## 7.1 Class Diagram

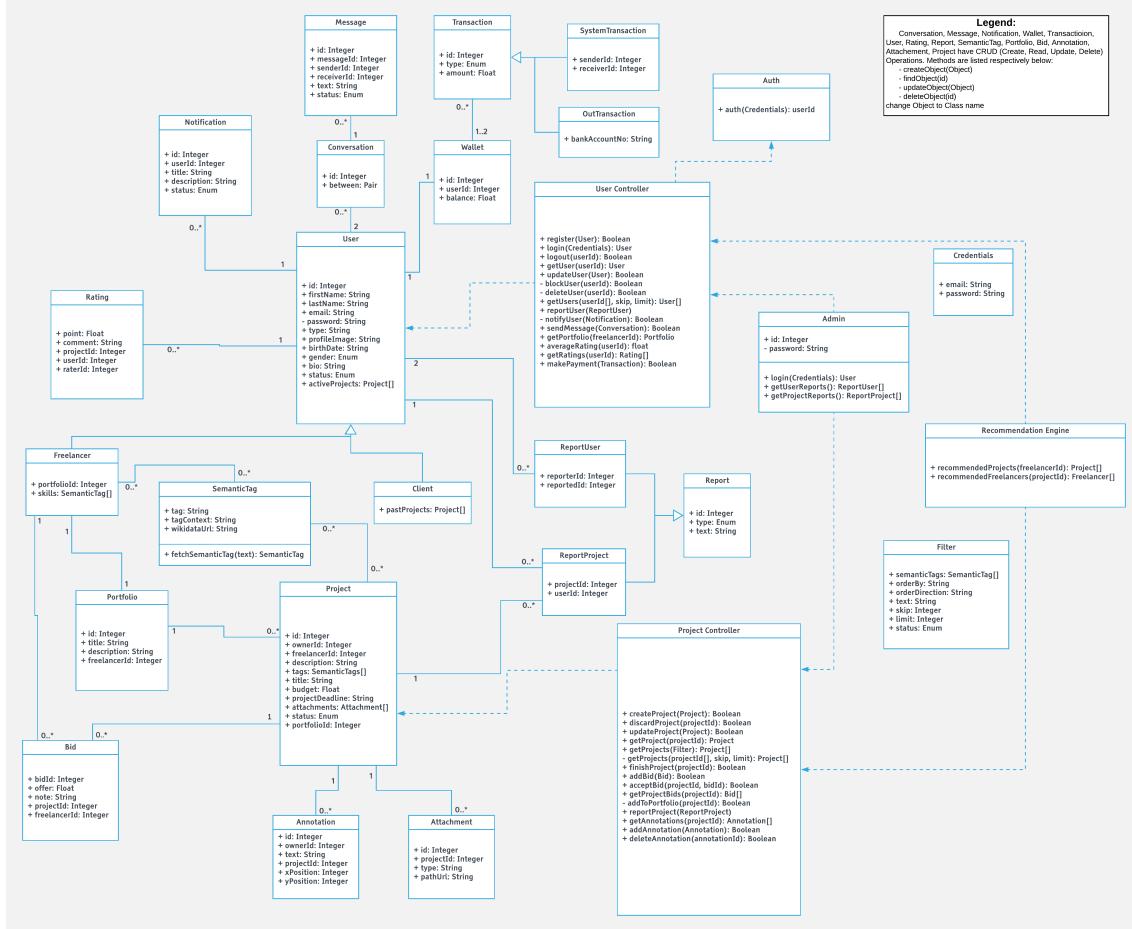


Figure 1: Class Diagram

## 7.2 Sequence Diagrams



Figure 2: Sign Up

## LOGIN

Batuhan | March 25, 2018

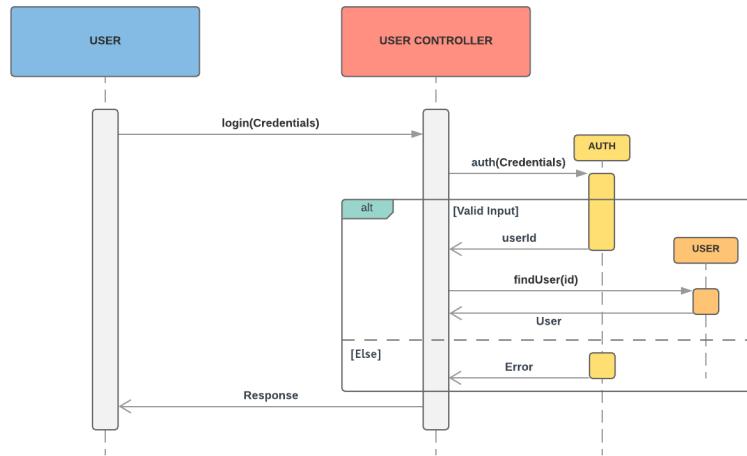


Figure 3: Login

## SD LOGGED IN CLIENT CREATE PROJECT

Enis Simsar | March 25, 2018

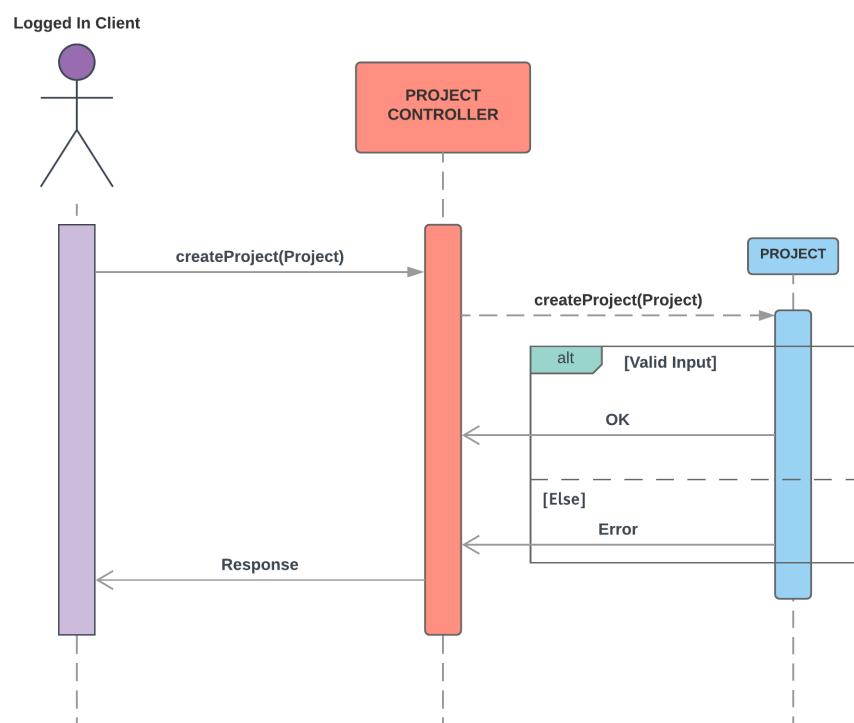


Figure 4: Create Project

## SD LOGGED IN FREELANCER BID ON PROJECT

alp.usen | March 25, 2018

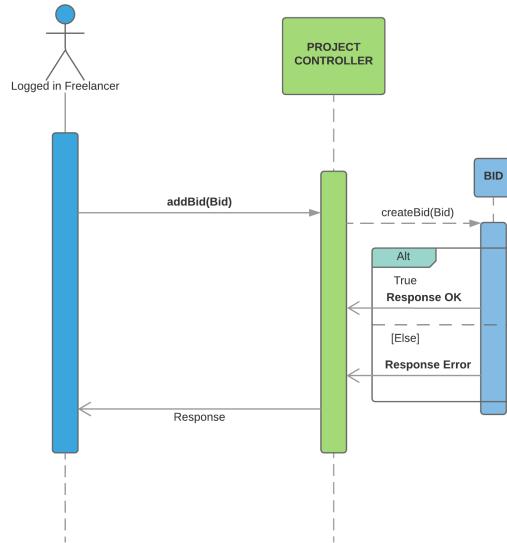


Figure 5: Bid on Project

## SD MESSAGE TO USER

Guliz Irem Gokceler | March 25, 2018

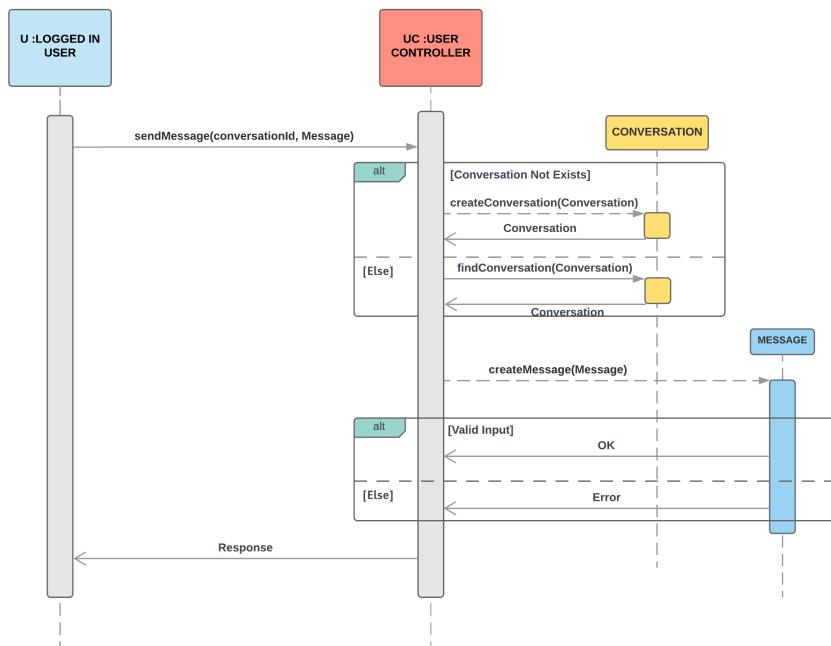


Figure 6: Send a Message

## SD LOGGED IN FREELANCER RECOMENDED PROJECTS

Enis Simsar | March 25, 2018

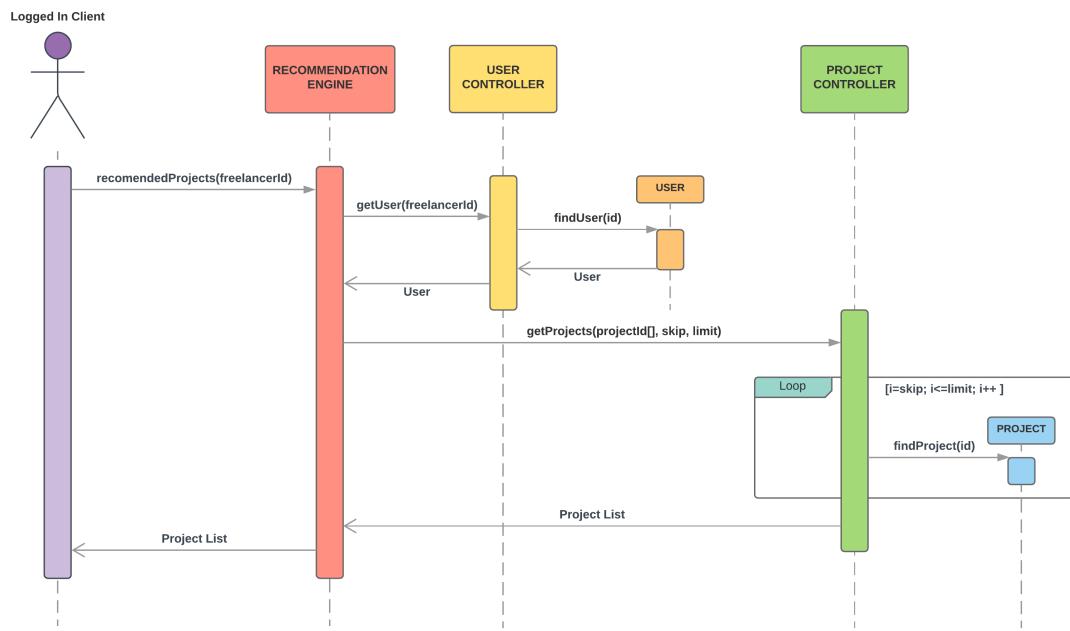


Figure 7: Recommended Projects

## SD LOGGED IN CLIENT RECOMMENDED FREELANCERS

Enis Simsar | March 25, 2018

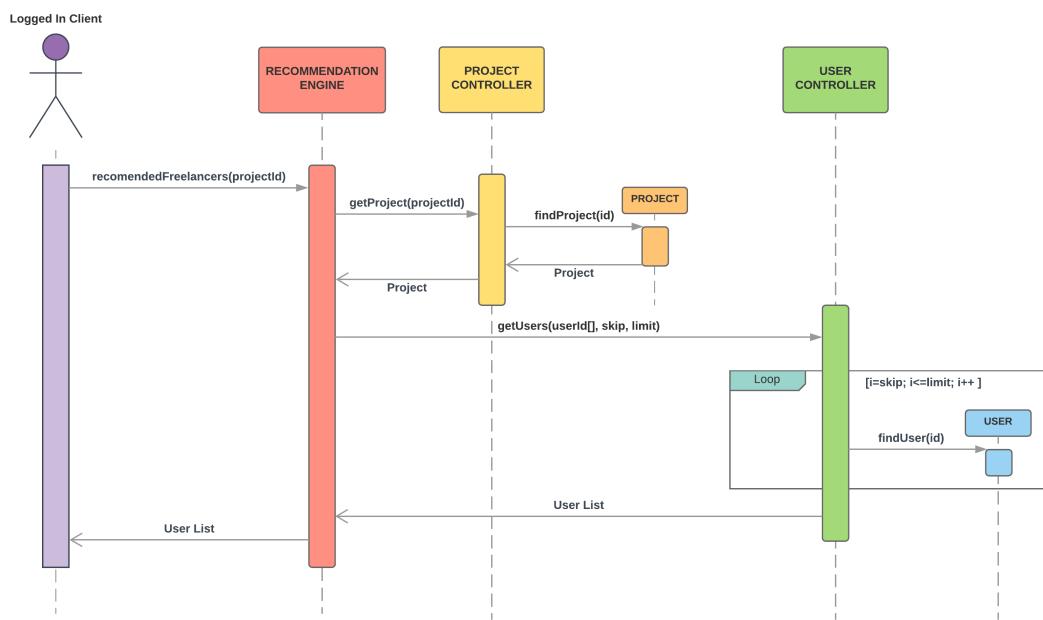


Figure 8: Recommended Freelancers

## SEARCH PROJECT

Bahadir Oncel | March 25, 2018

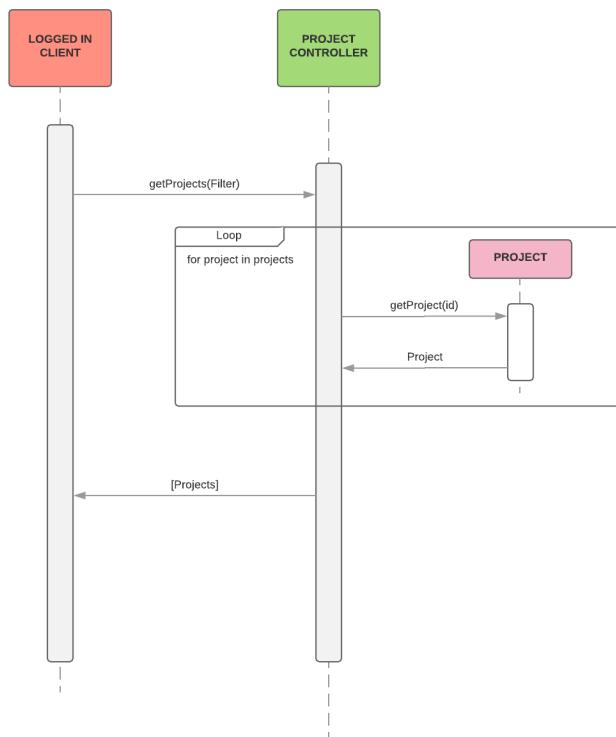


Figure 9: Search Projects

## REPORT PROJECT SEQUENCE DIAGRAM

Muhammed Emin Güre | March 25, 2018

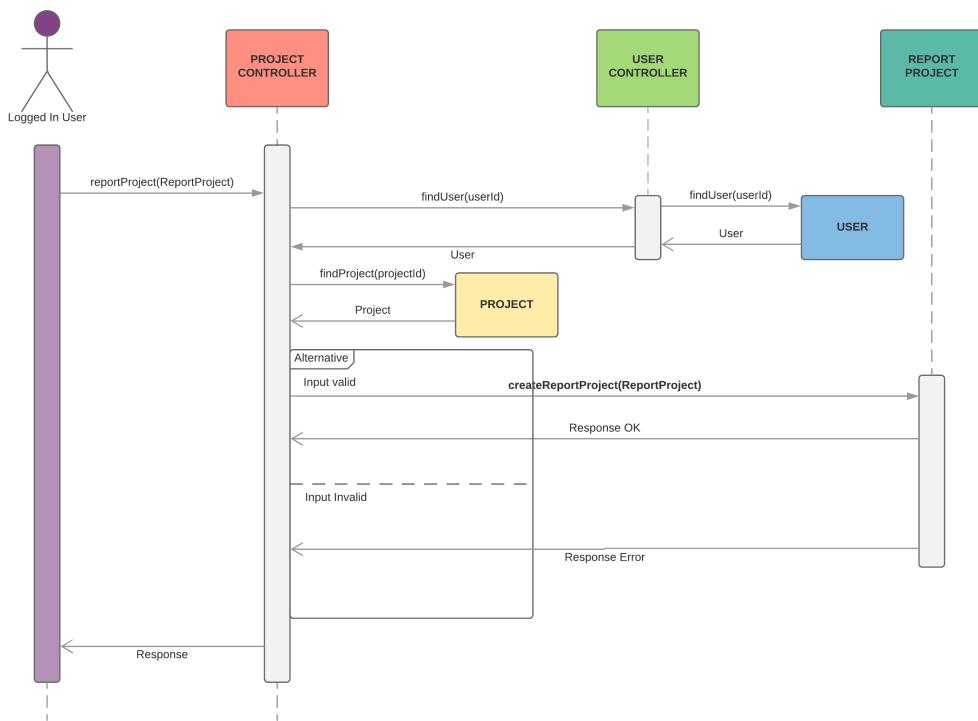


Figure 10: Report Projects

## 8 User Tests

**Note:** Please, add prefix 5.2 to all reference to requirements. For example, "Reference to Requirements: 1.3.7" means "Reference to Requirements: 5.2.1.3.7".

### Test Case ID: CREATE\_PROJECT

Reference to Requirement: 1.3.11.2.1.1.

Module Name: Project

Priority: High

Title: Project Creation Test Case

Designed by: Muhammed Emin Ayar

Tested by:

Design Date: 01.04.2018

Test Date:

Preconditions: User shall be signed in and in client mode.

Case ID	Test Scenario	Steps of Testing	Test Input	Expected Output	Actual Output	Pass\Fail
CREATE_PROJECT_V	Create a project with valid parameters	<ul style="list-style-type: none"><li>Go to home page as a client</li><li>Click on create project</li></ul>	<ul style="list-style-type: none"><li>Project Title: Snapchat Clone</li><li>Description: I want to marry a Victoria's Secret model so i need a snapchat clone app asap</li><li>Tags: ios,android</li><li>Budget: 2500\$</li><li>Deadline:01.04.2019</li><li>Attachments: Adriana_Lima.png</li></ul>	Project will be created		
CREATE_PROJECT_I	Try creating a project without entering budget	<ul style="list-style-type: none"><li>Go to home page as a client</li><li>Click on create project</li></ul>	<ul style="list-style-type: none"><li>Project Title: Snapchat Clone</li><li>Description: I want to marry a Victoria's Secret model so i need a snapchat clone app asap</li><li>Tags: ios,android</li><li>Deadline:01.04.2019</li><li>Attachments: Adriana_Lima.png</li></ul>	Project creation will fail		
CREATE_PROJECT_I	Try creating a project with invalid deadline	<ul style="list-style-type: none"><li>Go to home page as a client</li><li>Click on create project</li></ul>	<ul style="list-style-type: none"><li>Project Title: Snapchat Clone</li><li>Description: I want to marry a Victoria's Secret model so i need a snapchat clone app asap</li><li>Tags: ios,android</li><li>Budget: 2500\$</li><li>Deadline:01.04.2017</li><li>Attachments: Adriana_Lima.png</li></ul>	Project creation will fail		

Figure 11: Create Project Test Case

## Test Case ID: BID\_ON\_PROJECT

Reference to Requirement: 1.3.11.1.7, 2.3.1.2

Module Name: Project

Priority: High

Title: Bid On Project

Designed by: Alp Kaan Usen

Tested by:

Design Date: 31.03.2018

Test Date:

Preconditions: User shall be signed in and in freelancer mode.

Case ID	Test Scenario	Steps of Testing	Test Input	Expected Output	Actual Output	Pass\Fail
BID_ON_PROJECT_V	Try to bid on a project with valid input	<ul style="list-style-type: none"><li>• Open an available project's view page</li><li>• Input price and note</li><li>• Click place bid button</li></ul>	<ul style="list-style-type: none"><li>• Bid price: 100</li><li>• Note: "Pick me!"</li></ul>	Bid will be placed		
BID_ON_PROJECT_I	Try to bid on a project with invalid input	<ul style="list-style-type: none"><li>• Open an available project's view page</li><li>• Input price and note</li><li>• Click place bid button</li></ul>	<ul style="list-style-type: none"><li>• Bid price: -100</li><li>• Note: "I will give you money if you choose me!"</li></ul>	Bid will not be placed		

Figure 12: Bid Project Test Case

## Test Case ID: ADD\_FUNDS

Reference to Requirement: 1.3.3.1.

Module Name: Project

Priority: Low

Title: Add Funds To The Wallet

Designed by: Muhammed Emin Ayar

Tested by:

Design Date: 01.04.2018

Test Date:

Preconditions: User shall be signed in.

Case ID	Test Scenario	Steps of Testing	Test Input	Expected Output	Actual Output	Pass\Fail
ADD_FUNDS_V	Add funds with valid credit card information	<ul style="list-style-type: none"><li>• Go to the wallet page</li><li>• Click on add funds</li><li>• Type the amount of money</li></ul>	<ul style="list-style-type: none"><li>• Credit Card Holder: Mete Kocaman</li><li>• Credit Card Number: 5269 1234 0769 5500</li><li>• Expiration Date: 07/2022</li><li>• Security Code: 123</li></ul>	Funds will be added		
ADD_FUNDS_I	Try adding funds with invalid credit card information	<ul style="list-style-type: none"><li>• Go to the wallet page</li><li>• Click on add funds</li><li>• Type the amount of money</li></ul>	<ul style="list-style-type: none"><li>• Credit Card Holder: Mete Kocamann</li><li>• Credit Card Number: 5269 1234 0769 550</li><li>• Expiration Date: 07/2017</li><li>• Security Code: 123</li></ul>	Funds will not be added		

Figure 13: Add Funds Test Case

## Test Case ID: BID\_NOTIFICATION

Reference to Requirement: 2.3.1.1.1, 2.3.1.2

Module Name: Project

Priority: High

Title: Bid Notification

Designed by: Alp Kaan Usen

Tested by:

Design Date: 01.04.2018

Test Date:

Preconditions:

- Two different users shall be signed in, one in freelancer mode and the other in client mode.
- For BID\_NOTIFICATION\_F\_W and BID\_NOTIFICATION\_F\_L, freelancer shall have a bid placed on the project that is the subject of the test.
- For BID\_NOTIFICATION\_F\_W and BID\_NOTIFICATION\_F\_L, there shall be other bids on the project.
- For BID\_NOTIFICATION\_C, the freelancer shall not have placed a bid on the project previously.

Case ID	Test Scenario	Steps of Testing	Test Input	Expected Output	Actual Output	Pass\Fail
BID_NOTIFICATION_F_W	Notify the freelancer when the project's bidding period has ended	<ul style="list-style-type: none"> <li>Open the project view page as the client, who created the project</li> <li>Accept the freelancer's bid</li> </ul>	-	The freelancer will receive a notification, stating that they have won the bid		
BID_NOTIFICATION_F_L	Notify the freelancer when the project's bidding period has ended	<ul style="list-style-type: none"> <li>Open the project view page as the client, who created the project</li> <li>Accept the a bid other than the testing freelancer</li> </ul>	-	The freelancer will receive a notification, stating that the bidding period is over		
BID_NOTIFICATION_C	Notify the client when the freelancer places a bid on the project	<ul style="list-style-type: none"> <li>Open the project view page as the freelancer</li> <li>Input price and note</li> <li>Click place bid button</li> </ul>	<ul style="list-style-type: none"> <li>Bid price: 100</li> <li>Note: "Choose me!"</li> </ul>	The client will receive a notification, stating that a bid has been placed on their project		

Figure 14: Bid Notification Test Case

## Test Case ID: RECOMMEND\_PROJECTS

**Reference to Requirement:** 1.3.11.1.2 , 2.1.2

**Module Name:** Project

**Priority:** High

**Title:** Recommend Projects To Freelancers

**Designed by:** Muhammed Emin Ayar

**Tested by:**

**Design Date:** 01.04.2018

**Test Date:**

**Preconditions:** Users shall be signed in and in Freelancer mode.

- For RECOMMEND\_PROJECT\_P The user shall have python semantic tag and python projects in his/her portfolio.
- For RECOMMEND\_PROJECT\_A The user shall have android semantic tag and android projects in his/her portfolio.

Case ID	Test Scenario	Steps of Testing	Test Input	Expected Output	Actual Output	Pass\Fail
RECOMMEND_PROJECT_P	See recommended projects	• Go to home page	none	Recommended Project List: • Social Media Monitoring Backend API • ... • Backend To A School APP		
RECOMMEND_PROJECT_A	See recommended projects	• Go to home page	none	Recommended Project List: • Snapchat Clone • ... • App To See Where My Boyfriend Is		

Figure 15: Recommend Projects Test Case

## Test Case ID: SEARCH\_FOR\_PROJECTS\_BY\_RANKING

**Reference to Requirement:** 1.1.7., 1.2.6., 1.3.4., 2.1.1.6.

**Module Name:** Project

**Priority:** High

**Title:** Search for Projects by Ranking according to Deadline

**Designed by:** Güлиз İrem Gökçeler

**Tested by:**

**Design Date:** 01/04/2018

**Test Date:**

**Preconditions:** Every user(registered or guest) is able to do search operation

Case ID	Test Scenario	Steps of Testing	Test Input	Expected Output	Actual Output	Pass\Fail
SEARCH_FOR_PROJECTS	Search for a project with a keyword	<ul style="list-style-type: none"> <li>• Click search bar</li> <li>• Input a keyword</li> <li>• Click on search button</li> <li>• Select the ranking type</li> </ul> <ul style="list-style-type: none"> <li>• Keyword:"backend"</li> <li>• Rank by: deadline</li> </ul>		Search results will be displayed		

Figure 16: Search for Projects by Ranking Test Case

## Test Case ID: FINISH\_PROJECT\_GIVE\_COMMENT&RATE

Reference to Requirement: 1.3.11.2.7

Module Name: Project

Priority: High

Title: Finish Project and Give Comment&Rate

Designed by: Enis Simsar

Tested by:

Design Date: 01.04.2018

Test Date:

Preconditions: User shall be signed in and in client mode.

Case ID	Test Scenario	Steps of Testing	Test Input	Expected Output	Actual Output	Pass\Fail
FINISH_PROJECT_GIVE_COMMENT&RATE_V	Finish the project having a freelancer, and can give a comment and rate.	<ul style="list-style-type: none"> <li>Open her/his open project's view page</li> <li>Click finish project Button</li> <li>Open a window with project's freelancer information.</li> <li>Choose rate and type comment</li> <li>Click send rate&amp;comment button</li> </ul>	<ul style="list-style-type: none"> <li>rate: 4.5</li> <li>Comment: "Good Job!"</li> </ul>	Rating will be send and affect the freelancer rating.		
FINISH_PROJECT_GIVE_COMMENT&RATE_I	Finish the project not having a freelancer, and cannot give a comment and rate.	<ul style="list-style-type: none"> <li>Open her/his open project's view page</li> <li>Click finish project Button</li> <li>Open a window with a message "this project does not have a freelancer".</li> </ul>		Client will not rate.		

Figure 17: Finish Project & Give Comment and Rate Test Case

## Test Case ID: RECOMMEND\_FREELANCERS

Reference to Requirement: 1.3.11.2.6, 2.1.1

Module Name: Project

Priority: High

Title: Recommend Freelancers

Designed by: Batuhan Enes Çağlayan

Tested by:

Design Date: 31.03.2018

Test Date:

Preconditions:

- User shall be signed in and in client mode.
- For RECOMMEND\_FREELANCERS\_JS, user should have a project with the semantic tag "Javascript" and there should exist a user with the semantic tag "Javascript" in his/her portfolio.
- For RECOMMEND\_FREELANCERS\_IOS, user should have a project with the semantic tag "iOS" and there should exist a user with the semantic tag "iOS" in his/her portfolio.

Case ID	Test Scenario	Steps of Testing	Test Input	Expected Output	Actual Output	Pass\Fail
RECOMMEND_FREELANCERS_JS	Client is offered with recommended freelancers on home page	1. Go to home page	None	Recommended Freelancers List: <ul style="list-style-type: none"><li>• Ryan O' Reily</li><li>• ...</li><li>• Ivan Locke</li></ul>		
RECOMMEND_FREELANCERS_IOS	Client is offered with recommended freelancers on home page	1. Go to homepage	None	Recommended Freelancers List: <ul style="list-style-type: none"><li>• Tony Sopranos</li><li>• ...</li><li>• Ivan Karamazov</li></ul>		

Figure 18: Recommend Freelancers Test Case

## Test Case ID: CONNECT\_LINKEDIN\_PROFILE

Reference to Requirement: 1.3.11.1.6.

Module Name: Project

Priority: High

Title: Connect Linkedin Account

Designed by: Güliz İrem Gökçeler

Tested by:

Design Date: 01/04/2018

Test Date:

Preconditions:

- User shall have a valid Linkedin account and in freelancer mode.
- User shall be signed in and on his/her profile page. User will see a link to add a Linkedin account to his profile.

Case ID	Test Scenario	Steps of Testing	Test Input	Expected Output	Actual Output	Pass\Fail
CONNECT_LINKEDIN_PROFILE_V	He/she wants to connect his/her Linkedin profile to his/her account	<ul style="list-style-type: none"><li>• Click on "Add a Linkedin Account"</li><li>• Enters his/her Linkedin email</li><li>• Enters the password</li><li>• Clicks connect</li></ul>	<ul style="list-style-type: none"><li>• Email:"mertkizil@gmail.com"</li><li>• Password: mert0Kizil4</li></ul>	Connection is succesfull.		
CONNECT_LINKEDIN_PROFILE_I	He/she wants to connect his/her Linkedin profile to his/her account	<ul style="list-style-type: none"><li>• Click on "Add a Linkedin Account"</li><li>• Enters non-existing email</li><li>• Enters the password</li><li>• Clicks connect</li></ul>	<ul style="list-style-type: none"><li>• Email:"mrtkizil@gmail.com"</li><li>• Password: mert0Kizil4</li></ul>	System will show an error message.		

Figure 19: Connecting LinkedIn Profile Test Case

## Test Case ID: SEND\_MESSAGE

Reference to Requirement: 1.3.7.

Module Name: Project

Priority: High

Title: Send Message to a User

Designed by: Gülistan İrem Gökçeler

Tested by:

Design Date: 01/04/2018

Test Date:

Preconditions:

- User shall be signed in and on his/her profile page.
- User shall not be blocked by the user whom want to send message.
- Input username shall be valid and exist on the system.

Case ID	Test Scenario	Steps of Testing	Test Input	Expected Output	Actual Output	Pass\Fail
SEND_MESSAGE_V	User wants to send a message to another user.	<ul style="list-style-type: none"> <li>• Clicks on message button</li> <li>• Enters the username whom message will be sent</li> <li>• Enters the text</li> <li>• Clicks send button</li> </ul>	<ul style="list-style-type: none"> <li>• To:"Deniz Duran"</li> <li>• Text:"Hello!"</li> </ul>	System will send the message successfully		
SEND_MESSAGE_I	User tries to send message to a non-existent user	<ul style="list-style-type: none"> <li>• Clicks on message button</li> <li>• Enters non-existent username</li> <li>• Enters the text</li> <li>• Clicks send button</li> </ul>	<ul style="list-style-type: none"> <li>• To:"Derya Deniz"</li> <li>• Text:"Hello!"</li> </ul>	System will show:"No such an user exists!"		
SEND_MESSAGE_B	User tries to send message to an user that blocked him earlier.	<ul style="list-style-type: none"> <li>• Clicks on message button</li> <li>• Enters the username</li> <li>• Enters the text</li> <li>• Clicks send button</li> </ul>	<ul style="list-style-type: none"> <li>• To:"Mete Kocaman"</li> <li>• Text:"Hello!"</li> </ul>	System will show:"You can not message to that user!"		

Figure 20: Send Message Test Case

## 9 Project Plan

	Name	Duration	Start	Finish	Predecessors	Resource Names
1 ✓	Documentation	11 days	2/5/18 8:00 AM	2/19/18 5:00 PM		
2 ✓	Repository Initialize	6 days	2/5/18 8:00 AM	2/12/18 8:00 PM	Bahadir Oncel;Batuhan Enes Caglayan;Enis Simsar;Muhammed Emin Gure	
3 ✘ ✓	Repository Search	4 days	2/7/18 8:00 AM	2/12/18 8:00 PM	Each Member	
4 ✘ ✓	Personal Pages	2 days	2/8/18 8:00 AM	2/9/18 5:00 PM	Each Member	
5 ✓	Guidelines	4 days	2/14/18 8:00 AM	2/19/18 5:00 PM		
6 ✘ ✓	Issues	3 days	2/14/18 8:00 AM	2/16/18 8:00 PM	Guliz Irem Gokceles	
7 ✘ ✓	Commits	3 days	2/14/18 8:00 AM	2/16/18 8:00 PM	Guliz Irem Gokceles	
8 ✘ ✓	Labels	4 days	2/14/18 8:00 AM	2/19/18 8:00 PM	Batuhan Enes Caglayan;Ihsan Berkana Balaban	
9 ✘ ✓	Git	2 days	2/16/18 8:00 AM	2/19/18 8:00 PM	Enis Simsar	
10 ✓	Requirements, Communication And Project Plan	13 days	2/14/18 8:00 AM	3/2/18 8:00 PM		
11 ✘ ✓	Communication Plan	3 days	2/14/18 8:00 AM	2/16/18 8:00 PM	Bahadir Oncel	
12 ✓	The First Draft of the Project Requirements	3 days	2/14/18 8:00 AM	2/16/18 5:00 PM		
13 ✘ ✓	Glossary	3 days	2/14/18 8:00 AM	2/16/18 8:00 PM	Utku Alinan	
14 ✓	Functional Requirements	3 days	2/14/18 8:00 AM	2/16/18 5:00 PM		
15 ✘ ✓	User Requirements	3 days	2/14/18 8:00 AM	2/16/18 5:00 PM		
16 ✘ ✓	Guest User Requirements	1 day	2/14/18 8:00 AM	2/14/18 8:00 PM	Alp Kaan Usen;Enis Simsar	
17 ✘ ✓	Admin Requirements	1 day	2/15/18 8:00 AM	2/15/18 8:00 PM	Bahadir Oncel;Batuhan Enes Caglayan	
18 ✘ ✓	Freelancer Requirements	1 day	2/16/18 8:00 AM	2/16/18 8:00 PM	Alp Kaan Usen;Enis Simsar	
19 ✘ ✓	Client Requirements	1 day	2/16/18 8:00 AM	2/16/18 8:00 PM	Bahadir Oncel;Batuhan Enes Caglayan	
20 ✓	System Requirements	3 days	2/14/18 8:00 AM	2/16/18 5:00 PM		
21 ✘ ✓	Search Requirements	1 day	2/14/18 8:00 AM	2/14/18 8:00 PM	Guliz Irem Gokceles;Muhammed Emin Ayar;Muhammed Emin Gure	
22 ✘ ✓	Recommendation Requirements	1 day	2/15/18 8:00 AM	2/15/18 8:00 PM	Guliz Irem Gokceles;Muhammed Emin Ayar;Muhammed Emin Gure	
23 ✘ ✓	Notification Requirements	1 day	2/16/18 8:00 AM	2/16/18 8:00 PM	Guliz Irem Gokceles;Muhammed Emin Ayar;Muhammed Emin Gure	
24 ✘ ✓	Payment Requirements	1 day	2/16/18 8:00 AM	2/16/18 8:00 PM	Guliz Irem Gokceles;Muhammed Emin Ayar;Muhammed Emin Gure	
25 ✘ ✓	Non-Functional Requirements	1 day	2/16/18 8:00 AM	2/16/18 8:00 PM	Oguzhan Kirar;Utku Alinan;Ihsan Berkana Balaban	
26 ✓	Revise Project Requirements	1 day	2/19/18 8:00 AM	2/19/18 8:00 PM		
27 ✘ ✓	Project Plan	2 days	2/2/18 8:00 AM	2/23/18 8:00 PM	12	Each Member
28 ✓	The First Version of the Project Requirements	2 days	2/22/18 8:00 AM	2/23/18 5:00 PM	12	
29 ✘ ✓	Update wrt. Customers Feedbacks	1 day	2/22/18 8:00 AM	2/22/18 8:00 PM	Alp Kaan Usen;Bahadir Oncel;Muhammed Emin Ayar	
30 ✘ ✓	Update wrt. Group Feedbacks	1 day	2/23/18 8:00 AM	2/23/18 8:00 PM	Alp Kaan Usen;Bahadir Oncel;Muhammed Emin Ayar	
31 ✓	Revise Project Plan	1 day	2/26/18 8:00 AM	2/26/18 8:00 PM	27	Each Member
32 ✘ ✓	The First Version of the Project Plan	2 days	3/1/18 8:00 AM	3/2/18 8:00 PM	31	
33 ✓	Update wrt. Customer Feedbacks	2 days	3/1/18 8:00 AM	3/18/18 5:00 PM	Enis Simsar	
34 ✓	Update wrt. Group Feedbacks	2 days	3/1/18 8:00 AM	3/18/18 5:00 PM	Enis Simsar	
35 ✘ ✓	Mock-ups	7 days	2/28/18 8:00 AM	3/7/18 8:00 PM	31	
36 ✓	Web Mock-ups	4 days	2/28/18 8:00 AM	3/5/18 8:00 PM		
37 ✘ ✓	Welcome Page, Sign-up and Login	2 days	2/28/18 8:00 AM	3/1/18 5:00 PM	Batuhan Enes Caglayan	
38 ✘ ✓	User Profile	2 days	2/28/18 8:00 AM	3/1/18 5:00 PM	Oguzhan Kirar	
39 ✘ ✓	Home Page	2 days	3/1/18 8:00 AM	3/2/18 5:00 PM	Guliz Irem Gokceles	
40 ✘ ✓	Feed and Search Page	2 days	3/1/18 8:00 AM	3/2/18 5:00 PM	Enis Simsar	
41 ✘ ✓	Project Creation	2 days	3/2/18 8:00 AM	3/5/18 5:00 PM	Ihsan Berkana Balaban	
42 ✘ ✓	Project Page	2 days	3/2/18 8:00 AM	3/5/18 5:00 PM	Muhammed Emin Gure	
43 ✘ ✓	Statistics for Guest	2 days	3/2/18 8:00 AM	3/5/18 5:00 PM	Bahadir Oncel	
44 ✘ ✓	Wallet Page	2 days	3/2/18 8:00 AM	3/5/18 5:00 PM	Utku Alinan	
45 ✘ ✓	Finish and Delete Project	2 days	3/2/18 8:00 AM	3/5/18 5:00 PM	Ihsan Berkana Balaban,Oguzhan Kirar	
46 ✓	Mobile Mock-ups	2 days	3/2/18 8:00 AM	3/5/18 8:00 PM		
47 ✘ ✓	Welcome Page, Sign-up and Login	2 days	3/2/18 8:00 AM	3/5/18 5:00 PM	Batuhan Enes Caglayan	
48 ✘ ✓	User Profile	2 days	3/2/18 8:00 AM	3/5/18 5:00 PM	Alp Kaan Usen	
49 ✘ ✓	Home Page	2 days	3/2/18 8:00 AM	3/5/18 5:00 PM	Guliz Irem Gokceles	
50 ✘ ✓	Feed and Search Page	2 days	3/2/18 8:00 AM	3/5/18 5:00 PM	Enis Simsar	
51 ✘ ✓	Project Creation	2 days	3/2/18 8:00 AM	3/5/18 5:00 PM	Bahadir Oncel	
52 ✘ ✓	Project Page	2 days	3/2/18 8:00 AM	3/5/18 5:00 PM	Muhammed Emin Ayar	
53 ✘ ✓	Wallet Page	2 days	3/2/18 8:00 AM	3/5/18 5:00 PM	Utku Alinan	
54 ✘ ✓	Finish and Delete Project	2 days	3/2/18 8:00 AM	3/5/18 5:00 PM	Ihsan Berkana Balaban,Muhammed Emin Ayar	
55 ✘ ✓	Persons and User Scenarios	5 days	3/2/18 8:00 AM	3/6/18 8:00 PM		
56 ✓	Create Three Persons	3 days	3/2/18 8:00 AM	3/6/18 5:00 PM	Bahadir Oncel;Enis Simsar;Muhammed Emin Ayar;Muhammed Emin Gure	
57 ✘ ✓	Write Their Scenarios	1 day	3/7/18 8:00 AM	3/7/18 5:00 PM	Bahadir Oncel;Enis Simsar;Muhammed Emin Ayar;Muhammed Emin Gure	
58 ✘ ✓	Pairing with Mockups	2 days	3/7/18 8:00 AM	3/8/18 5:00 PM	36:46	Bahadir Oncel;Enis Simsar
59 ✘ ✓	Pairing with Requirements	1 day	3/8/18 8:00 AM	3/8/18 5:00 PM	31	Each Member
60 ✘ ✓	Revise Mockups	1 day	3/7/18 8:00 AM	3/7/18 5:00 PM	36:46	Each Member
61 ✘ ✓	Revise Project Plan	1 day	3/9/18 8:00 AM	3/9/18 5:00 PM	32	Enis Simsar;Muhammed Emin Ayar
62 ✘ ✓	Revise User Scenarios	1 day	3/9/18 8:00 AM	3/9/18 5:00 PM	56:57	Each Member
63 ✘ ✓	Revise Requirements	1 day	3/9/18 8:00 AM	3/9/18 5:00 PM	58:59	Each Member
64 ✘ ✓	Revise Wiki	1 day	3/9/18 8:00 AM	3/9/18 5:00 PM	60:61:62:63	Each Member
65 ✘ ✓	Diagrams	10 days	3/4/18 8:00 AM	3/27/18 5:00 PM		
66 ✘ ✓	Class Diagram	3 days	3/4/18 8:00 AM	3/16/18 5:00 PM	Alp Kaan Usen;Enis Simsar;Muhammed Emin Ayar;Muhammed Emin Gure	
67 ✘ ✓	Sequence Diagram	6 days	3/19/18 8:00 AM	3/26/18 5:00 PM	66	Enis Simsar
68 ✘ ✓	Revise Class & Sequence Diagrams	1 day	3/27/18 8:00 AM	3/27/18 5:00 PM	66:67	Each Member
69 ✘ ✓	Test Cases	4 days	3/29/18 8:00 AM	4/2/18 5:00 PM	28	Alp Kaan Usen;Enis Simsar;Guliz Irem Gokceles;Muhammed Emin Ayar
70 ✘ ✓	Revise Test Cases	2 days	4/4/18 8:00 AM	4/5/18 5:00 PM	69	Each Member
71 ✘ ✓	Milestone 1	1 day	4/5/18 8:00 AM	4/5/18 5:00 PM		
72 ✘ ✓	Twitter API	22 days	4/5/18 8:00 AM	5/4/18 8:00 PM		
73 ✘ ✓	Research Twitter Public API End Points	4 days	4/5/18 8:00 AM	4/10/18 5:00 PM	Each Member	
74 ✘ ✓	Create A Boiler Plate Code for Setup	2 days	4/10/18 8:00 AM	4/11/18 5:00 PM	Enis Simsar	
75 ✘ ✓	Create Karpuz Twitter APP	2 days	4/11/18 8:00 AM	4/12/18 5:00 PM	73:74	Each Member
76 ✘ ✓	Review Pull Requests and Merge Them	2 days	4/12/18 8:00 AM	4/13/18 5:00 PM	75	Each Member
77 ✘ ✓	Deploy Karpuz Twitter APP	1 day	4/13/18 8:00 AM	4/13/18 5:00 PM	76	Enis Simsar
78 ✘ ✓	Find and Report Bugs	4 days	4/30/18 8:00 AM	5/3/18 5:00 PM	76:77	Each Member
79 ✘ ✓	Revise Karpuz Twitter APP	1 day	5/4/18 8:00 AM	5/4/18 5:00 PM	77:78	Each Member
80 ✘ ✓	Milestone 2	1 day	5/7/18 8:00 AM	5/7/18 5:00 PM	72	Each Member
81 ✘ ✓	Report About Project	6 days	5/16/18 8:00 AM	5/23/18 5:00 PM		
82 ✘ ✓	Test Cases for Karpuz Twitter App	4 days	5/21/18 8:00 AM	5/24/18 5:00 PM	72	Each Member

Figure 21: Plan Table

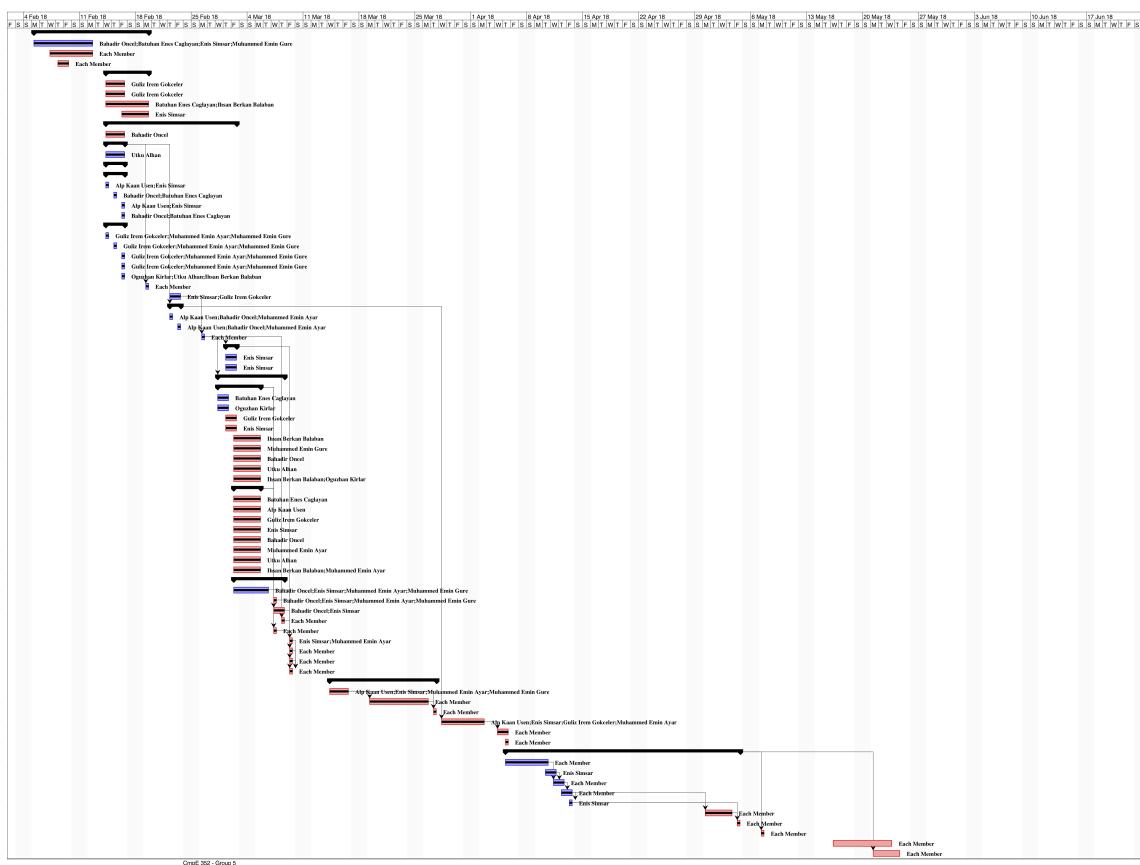


Figure 22: Gantt Chart

## 10 Software Development Infrastructure

We conducted a meeting about Karpuz Twitter APP. In the meeting, we discussed two frameworks, one of them is JavaScript NodeJS and the other one is Python Tornado. We could not decide one of them. Therefore, we took a poll after the meeting. The winner of the poll was Python Tornado and we researched about Python Tornado Framework and its example templates.

We found a good example for Tornado and started to create a boiler plate setup code. We created a baseline for our APP. Then, we started to research about Twitter Public API, its authentication, endpoints and so on. Some of us selected some endpoints. Then, they created their own apps in Twitter and started to try these endpoints in the Postman or Python codes that were already in the Course Piazza Page.

In the boiler plate setup code, we wrote a singleton which connects the Twitter Public API. We wanted to use this singleton design pattern because we are eager to learn some new information for us. And also, we created a homepage for our APP which contains 10 placeholder areas because we wanted to encourage participation whole team.

After writing boiler plate setup code, each team member created a new branch about her/his work name and each member started to work in her/his branch. We created a .env file which contains tokens. We added this file to the gitignore. Each team member add her/his Twitter Application tokens to the .env file. We used tokens in the project from this file.

First step, each team member created a handler file and added a endpoint to the APP. Brief explanation, when user calls the endpoint, endpoint redirects this requests to its handler and after some computation, this handler will return some response to the user.

Second step, each team member implemented her/his idea, Twitter Public API endpoint. We imposed a restriction about the this implementation. This implementation contains some logics and we want to test these logics in the future, before the exam. However, we created an APP which has a UI and our handlers return a HTML response. Therefore, we separated the logic parts from handlers and we put them into a new file which is called 'logic.py'. Each team member wrote his/her logic part into this file.

In the feature, if we want to create an API, we can easily create some new endpoints which return JSON type responses. This design makes this purpose easy and convenient because we have already logic parts that return JSON type responses.

Third step, after these Python codes, each team member started to wrote a html file which contains the results. We used Embedded Tweet from Twitter for tweets and some prepared templates from web for Twitter Users.

Fourth step, each team member committed and pushed her/his codes to the our GitHub repository. Then, s/he created a pull request about her/his work, and added all team members to this pull request as a reviewer. Reviewers read codes and gave some comments to the codes. After two approvals, we merged pull requests. Sometimes, we faced the merge conflicts. In these situations, firstly we solved the merge conflicts then again committed and pushed the version which solves the merge conflicts. After merging pull requests operations, we removed the related branches.

Fifth step, each team member tested (primitive testing) her/his functionality in her/his local-host. These tests include only this feature works or not. We created a readme file, which contains installation and package requirements of our app, for our Karpuz Twitter APP.

After completing the pull requests, we created an instance in the Digital Ocean. This instance runs on Ubuntu 16.04 and we installed miniconda (Python 3 environment). Then, we cloned out repository to this instance, installed requirements and run the APP. Lastly, we added the instance IP to the our readme file and wiki page.

## 11 Evaluation

### 11.1 Tools

For Project Management we used:

- **Git:** Best project management tool. But it has a lot of sub features. We used issues for tracking the project related tasks, Wiki for documenting our works, pull requests for merging branches and reviewing codes and readme for making announcements about our project. It becomes team work easy and more convenient.
- **Markdown:** We learned markdown language for writing wiki pages.
- **Project Libre:** A Project Plan Creation Tool. It is open source but it is not a good program. Its interface very poor and it does not provide a easy change in plan.
- **Balsamiq:** A Mockup Creation Tool. It is easy to use. It provides both mobile and web templates.
- **Lucid Chart:** A Chart Creation Tool. We created all diagrams with this program such as Class Diagram and Sequence Diagrams. It is a paid program but free for students. It is easy to use and has a powerful user interface. Also, it provides collaborative works.

For Karpuz Twitter APP we used:

- **Tornado:** A Web Framework for Python. It handles both BackEnd and FrontEnd.
- **PyCharm:** An IDE for Python. It provides PEP8 standards and we used this standard in our coding works. With this standard, all of us used the same convention. We did not face with a problem about tab style or etc.
- **SourceTree:** It provides an UI for Git in MacOs and Windows platforms. It is helpful for understanding Git Versioning system.
- **Postman:** It provides an UI to call some endpoints with Headers, Body and show these endpoints' responses.

### 11.2 Project Management

As stated in the Executive Summary part, the team consisted of members with all kinds of experience levels. We didn't plan any leadership in the team nor we intend to do, but more experienced members naturally emerged as project managers/team leaders and it's going nicely. The teacher aspect of the more experienced members helps the inexperienced ones by improving their skills, both computer engineering wise and the other skills needed in project development, such as communication and responsibility.

Communication part was both the funniest and the hardest part of the team work. People tend to be very accessible or not accessible at all - this made communication unpredictable. All team members should be equally accessible - and that should be every time, everyday for a startup. This course does not teach the startup life-cycle but acting as one is preferable since we need to move fast & move precisely.

As with all the teams, we also have some members who tend to ignore their responsibilities and leave their work for others. That's unacceptable but it's also a natural part of software development, especially in startups where hiring & firing should be done fast. We can't fire our team members, but we learned who to trust with important tasks and who to not to.

We believe we've nailed all the assignments and built a core team who're either skilled to do work or open to learning new things. We believe the next part of the course will be much smoother for us. We're aiming for lean development so changes (both in code and in management) will be easier.