Milestone 2 Report

Group 2

3 December 2018

Contents

1 Executive Summary

1.1 Description

Our project, WorkHub, is a platform that allows clients to ask for freelance work on countless categories, aiming to cover as many job types as possible. Once a user signs up, they're able to create jobs as a client, or take on jobs as a freelancer as they wish.

1.2   Project Status

We've managed to make quite a bit of progress since our last milestone, focusing on the core functionalities of the project: Those being job creation and job bidding. Jobs can be created by any client, and a freelancer can bid on it, allowing the client to either accept or reject it. Jobs successfully become closed for bidding once a bid has been accepted.

These main actions also send notifications to the affected user, but for now these notifications are limited to the back-end and android. They will soon be accessible from the front-end as well. Other than that, all 3 sub-groups are working at an almost identical pace.

We've managed to insert Continuous Integration into the project setup, which allows our code to be rebuilt on the server and checked for errors easily, letting us take action as needed.

We've also set up a file storage server that will allow us to store user profile pictures and similar files in the future.

From this point on, we plan to add a general search function, make categories fully accessible (alongside adding a few default categories), enable users to switch between client/freelancer profiles, and add annotations.

2   Deliverables

All three groups are following each other closely in terms of deliverables.

1. Sign up : Works properly, user is saved to database.

2. Log in : Works properly, also generates a session token for the user.

3. Log out : Same as login, destroys the existing session using the token.

4. Get profile information : Works the same on both front-end and Android.

5. Job creation : Works properly, jobs are added to the database, and are displated on the site. Additionally, a job-category relation is added for every category that the job contains.

6. Job display : Works properly, but hasn't been refined to cater to the user's past experiences. It simply lists all jobs.

7. Job bidding : Works properly, bid is listed on the site, and is saved in the database.

8. Job deletion : Implemented in back-end.

9. Bid acceptance : Works properly, job is set as closed, and the bid is set as accepted.

10. Bid rejection : Job status doesn't change and bid status is set to 'rejected'

11. Notifications : Implemented in back-end. A notification is automatically sent for bid acception and rejection.

3 Each Team Member's Work

Team Member	Job	Description
Ali Torğutalp (Back-End)	Back-End models and overall database system setup	Bid creation, updating, acception, rejection. Updating database model.
Bartu İnce (Back-End)	Feature Implementation	Adding the notification feature, creating automatic notifications for bid acception and rejection. Updating database model. General bugfixes.
Nilsu Pamuk (Back-End)	Feature Implementation	Adding bid listing and job deletion features.
Ergün Erdoğmuş (Front-End)	Integration Handling, Server Management, Design	Integrated Now for deploying Pull Requests and CircleCI integration for CI/CD. Handled AWS server configuration and Continuous Deployment pipeline. Also, integrated Prettier (Code formatter) and added a pre-commit hook for formatting committed code. Created initial Login page. Completed homepage redesign, added job detail page with job biddings listed
Huriye Özdemir (Front-End)	Feature implementation	Designing and enhancement of Job Detail page
Şükrü Can Gültop (Front-End)	Design sign-up, and connecting login and signup to backend, create bid page	Sign up design before we changed UI kit, and connecting those two pages to backend and validations of fields.Design and implement fields and connection of create bid page.
Ahmet Ozan Tatlısu (Android)	Implementation of accept bid, reject bid, update profile and Handling error messages.	Implementing some exception class to get error messages from server and connect them with our functions and adding accept bid, reject bid and update profile utilities and design relevant pages.
Zeki Tel (Android)	Implementation of create job, list jobs, list bids and create bid functionalities.	Creating functions of create job, list jobs, lists bids and create bid , connect them to back-end and design relevant pages.

4 Requirements

4.1 User Requirements

Guests:

1. Guests shall see the landing page with search bar included.

2. Guests shall be able to use the search functionality for public freelancer profiles and specific projects.

3. Guests shall be able to see the projects' only headlines, brief descriptions, base price of the project and they shall not be able to see any bidding information.

4. Guests shall be able to navigate with direct link to a public freelancer profile, a specific search page and a project page.

5. Guests shall be able to navigate to sign-up view anytime.

6. Guests shall be able to register with name, surname, email and password.

7. Guests shall receive an email upon register to verify their email address.

8. Guests shall not be able to login to app with a non verified email.

9. Guests shall be able to login to app with email and password.

Users (who signed up)

Freelancers

1. Freelancers shall be able to update their informations at any time (name, surname, profile photo, bio, skills, experience, IBAN number to receive payments) and login credentials (email, password).

2. Freelancers shall be able to annotate the texts inside their skills, experience, bio fields while adding or editing them.

3. Freelancers' public information shall contain skills, experience, bio, description, jobs taken on this site, rating, reviews and photo.

Jobs (as Freelancers)

1. Freelancers shall set a preference about their skills to see the type of job that they desire the most in the job feed in the home page.

2. Freelancers shall see the projects page and detailed description of projects.

3. Freelancers shall be able to bid on a project.

4. Freelancers shall be able to accept a job until a specified amount of time that is determined by client for freelancer to approve the accepted job is passed.

5. Freelancers shall be able to drop (sign off) a project.

6. Freelancers shall be able to search for clients that have posted jobs including or outside of their skill.

7. Freelancers shall be able to receive their payment after they have completed their job.

8. Freelancers shall be able to see its' active jobs.

Clients

1. Clients shall be able to update their informations at any time (name, surname, profile photo, bio, payment information) and login credentials (email, password).

2. Clients shall be able to annotate the texts inside their bio field while adding or editing them.

3. Clients' public information shall contain bio, jobs given on this site, rating, reviews and photo.

Jobs (as Clients)

1. Clients shall be able to create a project for freelancers to bid on.

2. Clients shall be able to add headline, description, base bidding price, image, video and due date to a project while creating it.

3. Clients shall be able to add annotations to headline, description, image and video while creating the project.

4. Clients shall be able to search projects, other clients and freelancers.

5. Clients shall be able to select if their job is visible to other clients while creating the project.

6. Clients shall be able to cancel a job when there are no freelancers that have taken the job even if there are biddings.

7. Clients shall be able to offer the job to a freelancer that is bidded on the project.

8. Clients shall be able to offer the job to a freelancer that bidded on the project even if the bidding period is not over.

9. Clients shall be able to object to a project and report it to platform owners while it is currently being worked on by a freelancer.

10. Clients shall be able to edit a project's headinline, description, image and video until bidding period is over.

11. Clients shall be able to make a payment, using the site as a middleman.

12. Clients shall be able to see its' created projects and its' active jobs.

Communication

1. Freelancers and clients shall be able to communicate with each other with a direct messaging system.

Rating and Reviews

1. Freelancers shall be able to leave a rating and a review about their client when; after the job is finished, after the job is dropped off by freelancer, after the job is cancelled by client.

2. Clients shall be able to leave a rating and a review about freelancer when; after the job is finished, after the job is dropped off by freelancer and after the job is cancelled by client.

3. The search mechanism shall prioritise freelancers that have received better overall ratings.

All Users

1. All Users shall be able to flag inappropriate use of the site and breaching the code of conduct.

2. Upon logging in, all users shall be redirected to their main page, showing their currently active jobs, followed by jobs that are recommended to them.

4.2 System Requirements

Recommendation Mechanism

1. The system shall provide a recommendation mechanism for clients that offers freelancers according to their past job requests.

2. The system shall provide a recommendation mechanism for freelancers that offers jobs according to their profile and their past searches.

Messaging Mechanism

1. There shall be a messaging mechanism to maintain the communication between freelancers and clients during processes.

2. Both parties shall be able to exchange texts.

3. Phone number and email sharing shall be forbidden.

Complaint Mechanism

1. There shall be a flagging system to notify administrators of misbehavior.

2. There shall be a system for reporting complaints with a checkbox with possible complaints and an "other" box.

Bidding Mechanism

1. Other bids shall be visible to bidders.

2. All bids shall be invisible to guests.

Search Mechanism

1. The system shall provide support for user-defined tags that allows jobs to be filtered.

2. The system shall contain an advanced search mechanism that lets the user set restraints for creator name, date, topic and budget.

Notification Mechanism

1. The system shall notify clients when their job has received a new bid.

2. The system shall notify both types of users when they receive a direct message.

3. The system shall notify a freelancer when one of their current jobs had its description changed by its client.

4. The system shall notify a freelancer when they win a bidding.

5. The system shall notify a freelancer about a change in the job description.

4.3 Non-functional Requirements

Security

1. Users shall register with valid information.

2. The system shall support the "Secure HyperText Transfer Protocol".

3. The database that archives user passwords shall be hashed and encrypted.

4. Users shall be forced to have a complex password that is least 8 characters long.

5. The system shall have 3D-Secure Payment for payment processes.

6. The product shall notify customers of changes to its information policy.

Reliability

1. In case of a downtime, the system shall recover in under 48 hours.

2. Data shall be backed up regularly to minimize loss of information.

Performance

1. The system shall respond to requests within 5 seconds to avoid interruptions.

2. The system should be able to handle at least 250 online users at any time.

3. The system shall be able to process 70 payment transactions per second.

Usability

1. The website shall be fully available in English.

Availability

1. The system shall meet or exceed 99% uptime.

2. Less than 20 seconds shall be needed to restart the system after a failure.

3. The system shall be able to support Chrome browser version 60 and Firefox 57("quantum") and their newer versions.

4. The system shall have an Android application which should work at Android 6.0 and newer versions.

Annotation

1. The W3C Web Annotation Data Model shall be used to annotate items which are job descriptions, images, messages.

Flexibility

1. The system shall be designed in a modular manner so that new features could be introduced easily.

2. The system shall be designed in a readable manner so that modifications can be done easily.

3. The system shall be designed to add new features easily.

Accessibility

1. System shall include accessibility components that encourage an inclusive user base by enabling people with visual impairments, hearing loss, limited dexterity, and other disabilities to access.
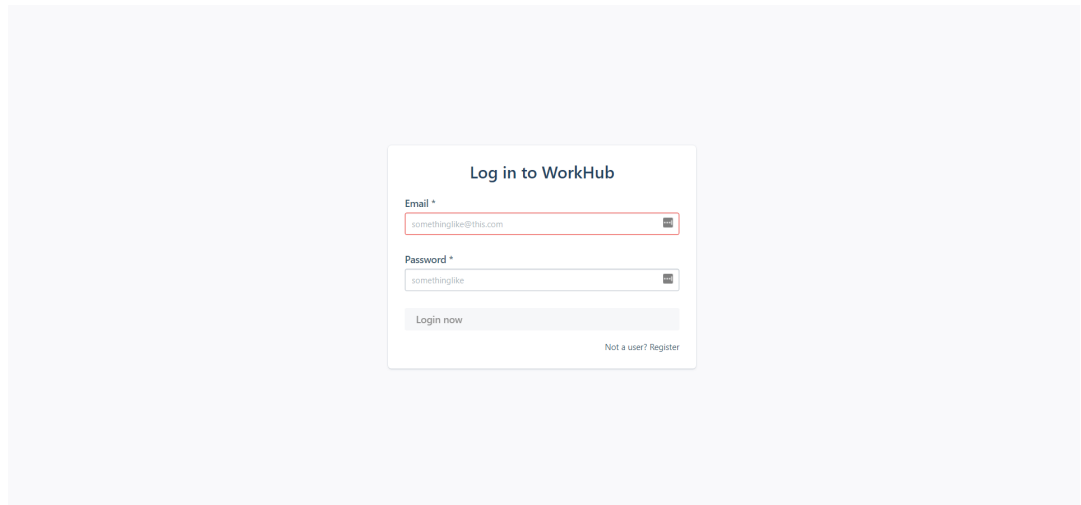
5 Design
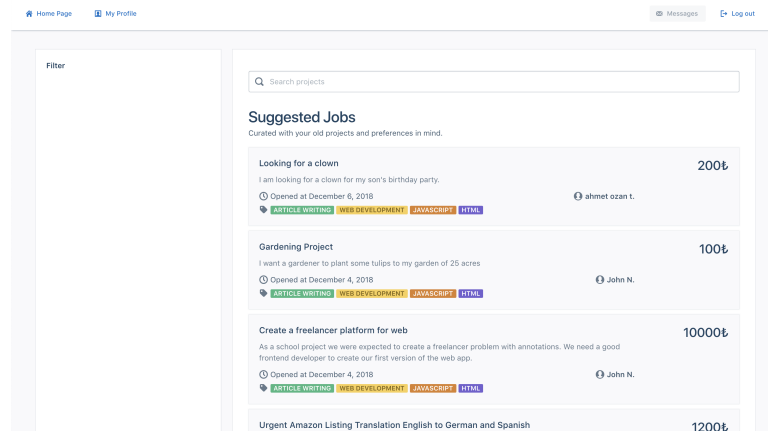

Figure 1: Front Page as a guest
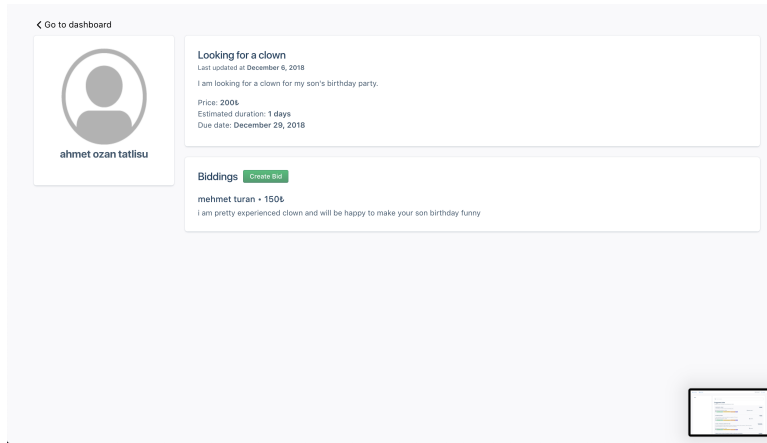

Figure 2: Home Page, with jobs visible
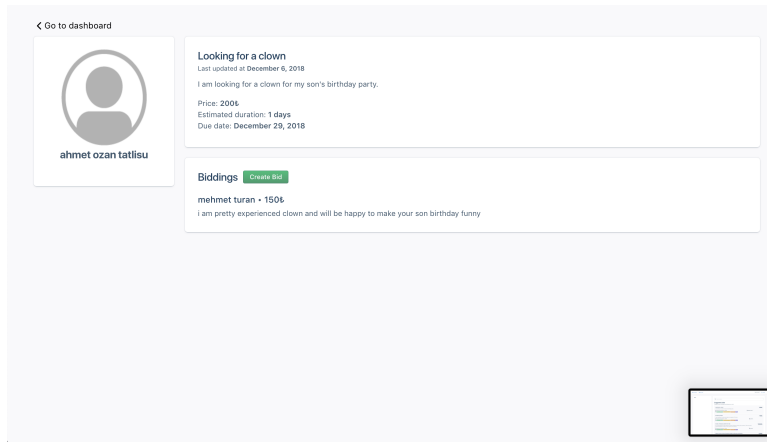
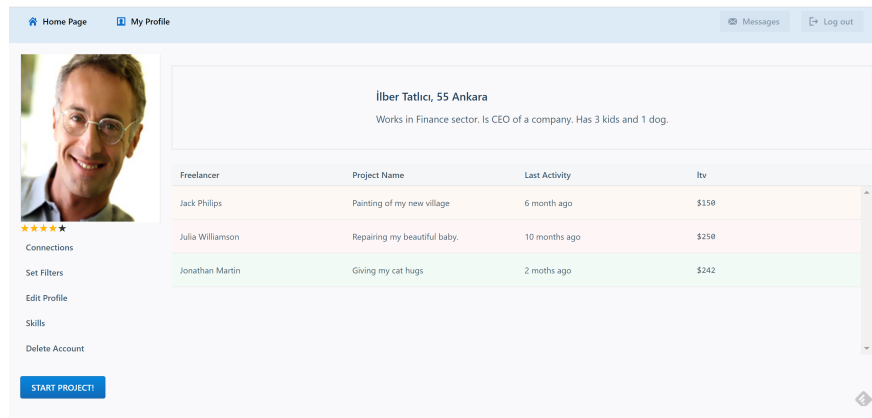Figure 3: Job detail page

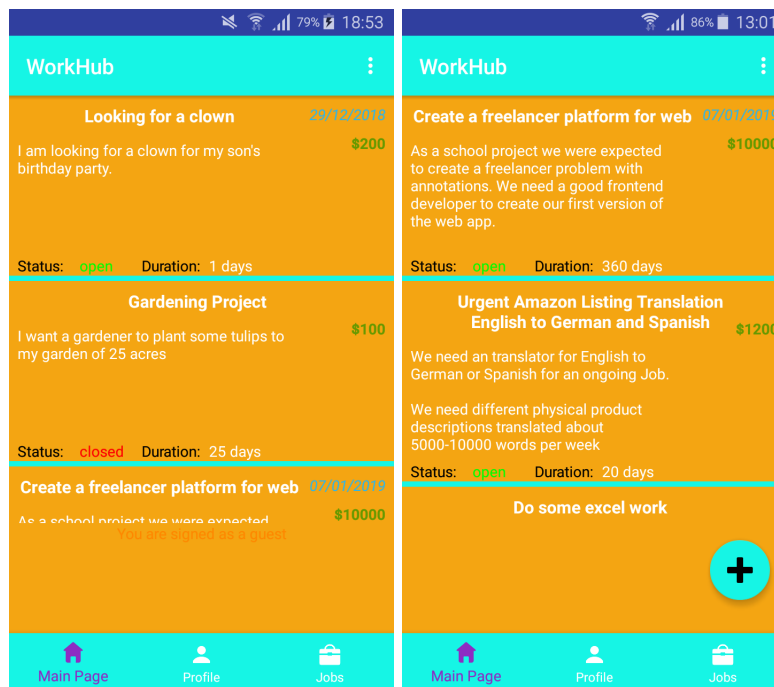Figure 4: Job bidding dialog

Figure 5: Profile Page

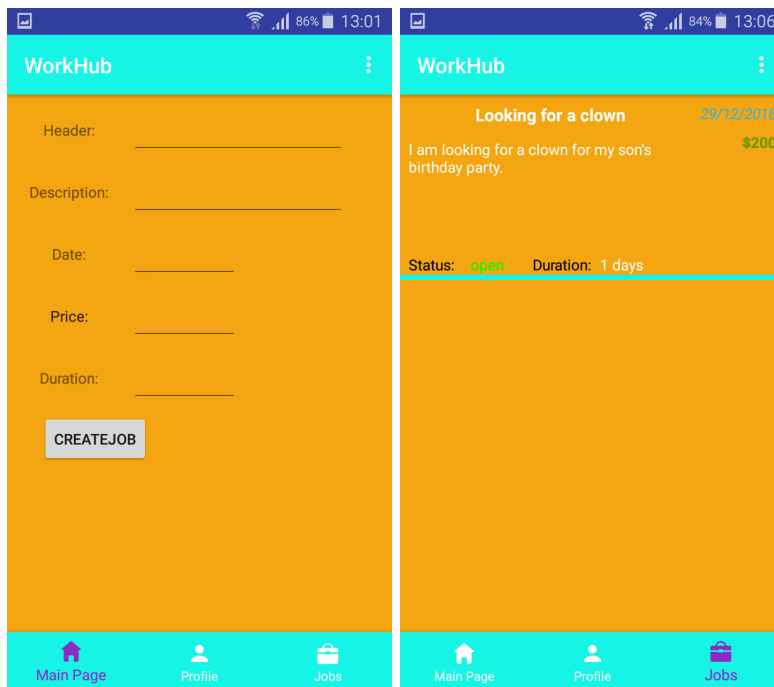Figure 6: Android Main Page as a guest and client
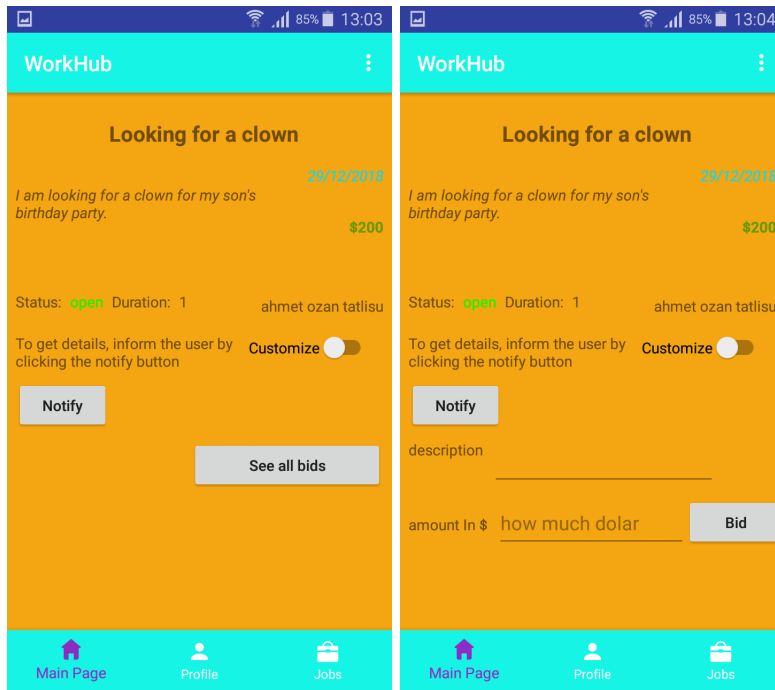
Figure 7: Job create page and jobs page
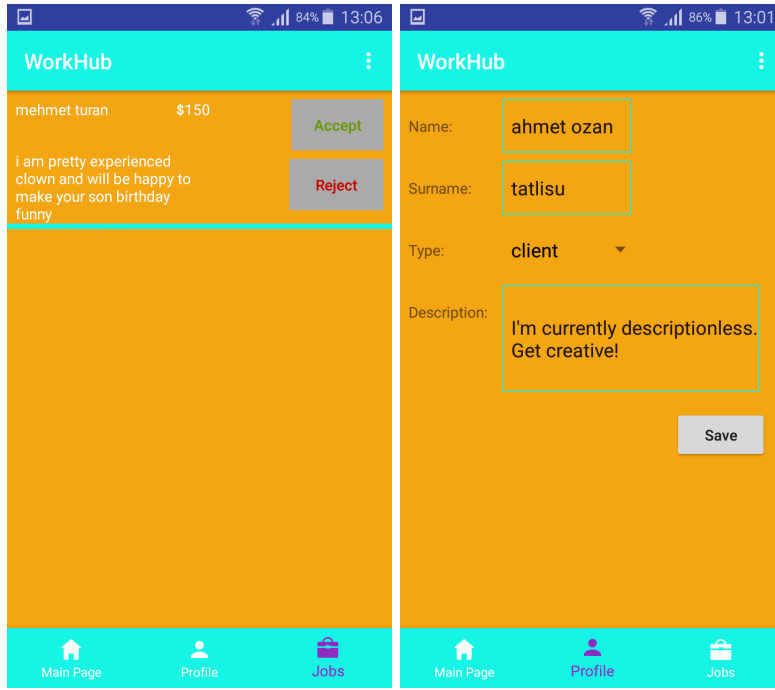
Figure 8: Jobs detail page for client and freelancer

Figure 9: All bids page for client and profile edit page

6 Project Plan

		Name	Duration	Start	Finish	Predecessors	Resource Names
1		**Initial Requirements and Project Planning**	**30 days**	**2/5/18 8:00 AM**	**3/16/18 5:00 PM**		**Everyone**
2		Researching Github	5 days	2/5/18 8:00 AM	2/9/18 5:00 PM		Everyone
3		Adding Personal Github wiki pages	1 day	2/7/18 8:00 AM	2/7/18 5:00 PM		Everyone
4		Creating README File	0.5 days	2/7/18 8:00 AM	2/7/18 1:00 PM		Ali Torutalp;Huriye Öz...
5		Deciding Communication Plan	1 day	2/14/18 8:00 AM	2/14/18 5:00 PM		Everyone
6		Creating initial Contribution Guide	3 days	2/14/18 8:00 AM	2/16/18 5:00 PM		Everyone
7		Meeting with Customer for the first time	0.25 days	2/15/18 8:00 AM	2/15/18 10:00 AM		Bartu nce;Nilsu Pamuk;...
8		Creating first draft of the requirements	5 days	2/15/18 8:00 AM	2/21/18 5:00 PM		Everyone
9		Creating the high-level project plan	5 days	2/21/18 4:00 PM	2/28/18 4:00 PM		Everyone
10		Discussing the first draft of the requirements with .	1 day	2/22/18 8:00 AM	2/22/18 5:00 PM	8	Everyone
11		Comparing the requirement draft with similar plat...	3 days	2/28/18 8:00 AM	3/2/18 5:00 PM	8	Everyone
12		Discussing and finalizing our unique features	5 days	3/5/18 8:00 AM	3/9/18 5:00 PM	8;11	Everyone
13		Finalizing the requirement and publishing the repo	5 days	3/12/18 8:00 AM	3/16/18 5:00 PM	8;9;10;11;12	Everyone
14		Milestone #1	0 days	3/12/18 8:00 AM	3/12/18 8:00 AM		
15		Revising the requirements	3 days?	3/19/18 8:00 AM	3/21/18 5:00 PM	1	
16		**UI Design**	**12.5 days**	**3/5/18 8:00 AM**	**3/21/18 1:00 PM**		**Everyone**
17		Creating user stories (Creating use case diagram)	5 days	3/5/18 8:00 AM	3/9/18 5:00 PM		Everyone
18		Creating mobile mockup	2.5 days	3/5/18 8:00 AM	3/7/18 1:00 PM		Ali Torutalp;Harun Eren
19		Creating web mockup	1.25 days	3/5/18 8:00 AM	3/6/18 10:00 AM		Ergün Erdomu;Fatih ...
20		Discussing mockups with customer	1 day	3/12/18 8:00 AM	3/12/18 5:00 PM	18;19	Everyone
21		Deciding on the theme of the platform and creatin..	1 day	3/19/18 8:00 AM	3/19/18 5:00 PM		Bartu nce;Ergün Erdo...
22		Designing the UI mobile app	2.5 days	3/19/18 8:00 AM	3/21/18 1:00 PM	18;19;20	Ali Torutalp;Harun Eren
23		Designing the UI web app	1.25 days	3/19/18 8:00 AM	3/20/18 10:00 AM	18;19;20	Ergün Erdomu;Fatih ...
24		Milestone #2	0 days	3/19/18 8:00 AM	3/19/18 8:00 AM		
25		Revising the designs	1 day?	3/21/18 1:00 PM	3/22/18 1:00 PM	16	
26		**High level design of the system**	**10 days**	**4/4/18 8:00 AM**	**4/17/18 5:00 PM**		**Everyone**
27		Creating class diagrams	5 days	4/4/18 8:00 AM	4/10/18 5:00 PM		Everyone
28		Creating sequence diagrams	5 days	4/11/18 8:00 AM	4/17/18 5:00 PM		Everyone
29		Creating activity diagrams	5 days	4/11/18 8:00 AM	4/17/18 5:00 PM		Everyone
30		Milestone #3	0 days	4/11/18 8:00 AM	4/11/18 8:00 AM		
31		**Design of the system**	**23 days**	**4/25/18 8:00 AM**	**5/25/18 5:00 PM**		**Everyone**
32		Database design	7.5 days	4/25/18 8:00 AM	5/4/18 1:00 PM		Ali Torutalp;Harun Eren
33		Server side design	7.5 days	5/2/18 8:00 AM	5/11/18 1:00 PM		Zeki Tel;ükrü Can Gült...
34		Mobile architecture design	15 days	5/2/18 8:00 AM	5/22/18 5:00 PM		Harun Eren
35		Web architecture design	15 days	5/5/18 8:00 AM	5/25/18 5:00 PM		Ergün Erdomu
36		Milestone #4	0 days	5/5/18 8:00 AM	5/7/18 5:00 PM		
37		Revising the system design	1 day?	5/28/18 8:00 AM	5/28/18 5:00 PM	31	
38		**Implementation**	**28 days?**	**10/3/18 8:00 AM**	**11/9/18 5:00 PM**		
39		**Authentication pages**	**12 days?**	**10/3/18 8:00 AM**	**10/18/18 5:00 PM**		
40		**Login page**	**7 days?**	**10/3/18 8:00 AM**	**10/11/18 5:00 PM**		
41		Server implementation	7 days?	10/3/18 8:00 AM	10/11/18 5:00 PM		
42		Webapp implementation	7 days?	10/3/18 8:00 AM	10/11/18 5:00 PM		
43		Android implementation	7 days?	10/3/18 8:00 AM	10/11/18 5:00 PM		
44			**7 days?**	**10/10/18 8:00 AM**	**10/18/18 5:00 PM**		
45		Server implementation	7 days?	10/10/18 8:00 AM	10/18/18 5:00 PM		
46		Webapp implementation	7 days?	10/10/18 8:00 AM	10/18/18 5:00 PM		
47		Android implementation	7 days?	10/10/18 8:00 AM	10/18/18 5:00 PM		
48		**Profile page**	**7 days?**	**10/17/18 8:00 AM**	**10/25/18 5:00 PM**		

		Name	Duration	Start	Finish	Predecessors	Resource Names
49		Server implementation	7 days?	10/17/18 8:00 AM	10/25/18 5:00 PM		
50		**Freelancer public page**	**7 days?**	**10/17/18 8:00 AM**	**10/25/18 5:00 PM**		
51		Webapp implementation	7 days?	10/17/18 8:00 AM	10/25/18 5:00 PM		
52		Android implementation	7 days?	10/17/18 8:00 AM	10/25/18 5:00 PM		
53		**Client public page**	**7 days?**	**10/17/18 8:00 AM**	**10/25/18 5:00 PM**		
54		Webapp implementation	7 days?	10/17/18 8:00 AM	10/25/18 5:00 PM		
55		Android implementation	7 days?	10/17/18 8:00 AM	10/25/18 5:00 PM		
56		**Job creation page**	**7 days?**	**10/24/18 8:00 AM**	**11/1/18 5:00 PM**		
57		Server implementation	7 days?	10/24/18 8:00 AM	11/1/18 5:00 PM		
58		Webapp implementation	7 days?	10/24/18 8:00 AM	11/1/18 5:00 PM		
59		Android implementation	7 days?	10/24/18 8:00 AM	11/1/18 5:00 PM		
60		**Job detail page**	**7 days?**	**10/31/18 8:00 AM**	**11/8/18 5:00 PM**		
61		**Showing job details**	**7 days?**	**10/31/18 8:00 AM**	**11/8/18 5:00 PM**		
62		Server implementation	7 days?	10/31/18 8:00 AM	11/8/18 5:00 PM		
63		Webapp implementation	7 days?	10/31/18 8:00 AM	11/8/18 5:00 PM		
64		Android implementation	7 days?	10/31/18 8:00 AM	11/8/18 5:00 PM		
65		**Listing current bids**	**7 days?**	**10/31/18 8:00 AM**	**11/8/18 5:00 PM**		
66		Server implementation	7 days?	10/31/18 8:00 AM	11/8/18 5:00 PM		
67		Webapp implementation	7 days?	10/31/18 8:00 AM	11/8/18 5:00 PM		
68		Android implementation	7 days?	10/31/18 8:00 AM	11/8/18 5:00 PM		
69		Revision of first milestone	7 days?	11/1/18 8:00 AM	11/9/18 5:00 PM		
70		Adding end to end tests	7 days?	11/1/18 8:00 AM	11/9/18 5:00 PM		
71		Milestone #2.1	0 days?	11/1/18 8:00 AM	11/1/18 8:00 AM		
72		**Bid on a job feature**	**7 days?**	**11/8/18 8:00 AM**	**11/16/18 5:00 PM**		
73		Server implementation	7 days?	11/8/18 8:00 AM	11/16/18 5:00 PM		
74		**Handling notifications**	**7 days?**	**11/8/18 8:00 AM**	**11/16/18 5:00 PM**		
75		Server implementation	7 days?	11/8/18 8:00 AM	11/16/18 5:00 PM		
76		Webapp implementation	7 days?	11/8/18 8:00 AM	11/16/18 5:00 PM		
77		Android implementation	7 days?	11/8/18 8:00 AM	11/16/18 5:00 PM		
78		**Search feature**	**7 days?**	**11/15/18 8:00 AM**	**11/23/18 5:00 PM**		
79		Server implementation	7 days?	11/15/18 8:00 AM	11/23/18 5:00 PM		
80		Webapp implementation	7 days?	11/15/18 8:00 AM	11/23/18 5:00 PM		
81		Android implementation	7 days?	11/15/18 8:00 AM	11/23/18 5:00 PM		
82		**Job listing - freelancer feed**	**7 days?**	**11/22/18 8:00 AM**	**11/30/18 5:00 PM**		
83		**Creating a recommendation mechanism**	**7 days?**	**11/22/18 8:00 AM**	**11/30/18 5:00 PM**		
84		Server implementation	7 days?	11/22/18 8:00 AM	11/30/18 5:00 PM		
85		Webapp implementation	7 days?	11/22/18 8:00 AM	11/30/18 5:00 PM		
86		Android implementation	7 days?	11/22/18 8:00 AM	11/30/18 5:00 PM		
87		Server implementation	7 days?	11/22/18 8:00 AM	11/30/18 5:00 PM		
88		Webapp implementation	7 days?	11/22/18 8:00 AM	11/30/18 5:00 PM		
89		Android implementation	7 days?	11/22/18 8:00 AM	11/30/18 5:00 PM		
90		**Messaging feature**	**7 days?**	**11/29/18 8:00 AM**	**12/7/18 5:00 PM**		
91		Server implementation	7 days?	11/29/18 8:00 AM	12/7/18 5:00 PM		
92		Webapp implementation	7 days?	11/29/18 8:00 AM	12/7/18 5:00 PM		
93		Android implementation	7 days?	11/29/18 8:00 AM	12/7/18 5:00 PM		
94		**Adding report message feature**	**7 days?**	**11/29/18 8:00 AM**	**12/7/18 5:00 PM**		
95		Server implementation	7 days?	11/29/18 8:00 AM	12/7/18 5:00 PM		
96		Webapp implementation	7 days?	11/29/18 8:00 AM	12/7/18 5:00 PM		

	⑩	Name	Duration	Start	Finish	Predecessors	Resource Names
97		Android implementation	7 days?	11/29/18 8:00 AM	12/7/18 5:00 PM		
98		Revision of milestone 2	3 days?	11/29/18 8:00 AM	12/3/18 5:00 PM		
99		Adding end to end tests	3 days?	11/29/18 8:00 AM	12/3/18 5:00 PM		
100		Milestone #2.2	0 days?	11/29/18 8:00 AM	11/29/18 8:00 AM		
101		**Client, freelancer accepting-declining feature**	**7 days?**	**12/5/18 8:00 AM**	**12/13/18 5:00 PM**		
102		Server implementation	7 days?	12/5/18 8:00 AM	12/13/18 5:00 PM		
103		**Handling notifications**	**7 days?**	**12/5/18 8:00 AM**	**12/13/18 5:00 PM**		
104		Server implementation	7 days?	12/5/18 8:00 AM	12/13/18 5:00 PM		
105		**Creating necessary views**	**7 days?**	**12/5/18 8:00 AM**	**12/13/18 5:00 PM**		
106		Webapp implementation	7 days?	12/5/18 8:00 AM	12/13/18 5:00 PM		
107		Android implementation	7 days?	12/5/18 8:00 AM	12/13/18 5:00 PM		
108		**Freelancer delivering project feature**	**7 days?**	**12/12/18 8:00 AM**	**12/20/18 5:00 PM**		
109		Server implementation	7 days?	12/12/18 8:00 AM	12/20/18 5:00 PM		
110		Webapp implementation	7 days?	12/12/18 8:00 AM	12/20/18 5:00 PM		
111		Android implementation	7 days?	12/12/18 8:00 AM	12/20/18 5:00 PM		
112		**Freelancer drop off feature**	**7 days?**	**12/19/18 8:00 AM**	**12/27/18 5:00 PM**		
113		Server implementation	7 days?	12/19/18 8:00 AM	12/27/18 5:00 PM		
114		Webapp implementation	7 days?	12/19/18 8:00 AM	12/27/18 5:00 PM		
115		Android implementation	7 days?	12/19/18 8:00 AM	12/27/18 5:00 PM		
116		Adding all necessary tests	3 days?	12/19/18 8:00 AM	12/21/18 5:00 PM		
117		Revision of final milestone	3 days?	5/11/18 8:00 AM	5/15/18 5:00 PM		

7 Code Structure

7.1 Branching

Each of our features were developed in their own branch, to prevent any conflicts. They were all contained in a general branch called "feature", followed by a sub-branch named after what they intend to do. Bug-fixes were kept under another branch by the name of "bugfix", following the same structure as "feature". This was done separately for all 3 workgroups (back-end, front-end, and Android), to prevent any feature name mixups or code conflicts.

7.2   Pull Requests

We utilized Github's pull request reviewing system very strictly. None of the code was directly pushed into the main branch unless it was an extremely minor fix, and even then, it was done under the supervision of someone from another subgroup, making sure there won't be any conflicts.

8   Evaluation of the Tools Used

We have used a multitude of different tools for our project, mostly with great success.

- **SourceTree** : The main software we have used to create, edit and delete our branches while also quickly being able to push, pull and commit code as needed. It has a somewhat steep learning curve, but it has proven to be very helpful in our project.

- **Node.js Express** : The framework used for back-end development. It's too early in the development phase to see if the choice was absolutely correct, but so far it did not cause any problems during coding. Express' ability to deploy a working app very quickly is a boon, as well.

- **Sequelize** : The framework that allows us to access our database, and send queries easily using its built-in functions, instead of requiring to type out the query by hand. Aside from the fact that its documentation website seems to be down more often than not, it's been very useful.

- **Android** : We have used native Java to implement the Android part of the project. Java is simple to implement and has many easily accessible sources on various websites.

- **Now** : We've deployed each of our pull requests with Now and it helped us test whether the changes we have made were acceptable without the need to download, apply the patch and build the code base all over again for each branch.

- **CircleCI** : We've used this as the CI/CD tool. Its configuration is pretty easy and it works at acceptable speeds. So, we can deploy our changes almost instantly.

- **Prettier** : Prettier is a tool used for conforming code to a single standard, greatly increasing readability in group projects like ours. As minor as it may seem, it's been quite a bit of help.