CMPE 352

Milestone Report

Ece Ata

Ahmet Yasin Alp

Cemal Aytekin

Volkan Yılmaz

Hatice Melike Ecevit

Akın İlerle

Öncel Keleş

Halil Samed Çıldır

Deniz Etkar

Contents

1. Introduction

This is a milestone report that describes the work, the deliverables and the status of work accomplished by group 1 throughout this semester.

2. Summary

Our project is a living history project which lets users to share their memories. The main idea is to make the culture live and be remembered. Users share posts about old times and habits which can be annotated by other users so it creates a cumulative memory repository.

What we have done so far in this semester is as follow,

- Met with our group members.
- Learned about GitHub and decided on a issue format that suits everyone in our group. To be able to do that we also learned about usage of Wiki and Wiki syntax.
- Determined our communication plan in order to have an efficient communication between group members.
- Stated our requirements, which defines what are going to offer to our users in our projects.
- Settled on a project plan to be more precise, since we work as group it is crucial to determine “When?” and “Who?” questions of any task.
- Created user stories and mock-ups.
- Created class, sequence and use case diagrams.
- Wrote test cases for our projects.
- Apart from these, we have designed and implemented a twitter project using Twitter api.

At the end of the semester this course made us realize that to work as a group has its own challenges and responsibilities. To overcome these, a group must be planned. Our project is going according to our project plan and we have finished planning part. Next semester we can start implementation immediately.

3. List and Status of Deliverables

- **Description:** Summary of the project.
    - **Status:** Completed.

- **Communication Plan:** Contains the information about communication and meeting plan of the group.
    - **Status:** Completed.

- **Requirements:** Glossary and functionalities of the project.
    - **Status:** Completed. However some additions/updates may be done according to the development of the application.

- **Project Plan:** Plan of the project and how the task divided between group members for the entire 2 semesters of the course.
    - **Status:** Completed. May change if any of the team members should decide not to take the related course next semester.

- **Use Case Scenarios:** 3 use case scenarios about people and possible ways of them using the application. The goal of these scenarios is to have a better look at the design of the application in real life uses, and prepare for users' way of approaching the project.
    - **Status:** Completed.

- **Mock Ups:** The draft of the application design and its pages. Contains designs of both web and android platforms.
    - **Status:** Completed.

- **Class Diagram:** Information about classes. Also shows their functions and relationship with each other. The coding of the application will be based on these classes shown in this diagram.
    - **Status:** Completed. However some additions/updates may be done according to the application.

- **Sequence Diagram:** Shows the sequence diagram of some scenarios like create memory, login etc. The diagrams represent the function relations between classes and user, and how they should interact with each other using the proper functions and variables.
    - **Status:** Completed

- **Use Case Diagram:** Shows the ways and relations between the end user or the admin and functions of the classes. Also shows the relations between classes and their own functions.
    - **Status:** Completed.

- **Test Cases:** Cases of various user actions and their reports. These cases show the necessary steps the application should run in order to achieve the main action user wants to happen. The tests are made logically, based on class and function interactions, and what they should yield. These tests are not made on the application yet.
    - **Status:** Halfway completed. Waiting for real life demos on the actual application.

4. Evaluation of the Status of Deliverables

All the deliverables that are planned to be finished by the time of this report are finished without any latency. Every member of the team concluded and submitted his/hers work on time. Thus, there were no effect or changes to the project plan.

5. A Summary of Coding Work Done By Each Team Member

Member	Work
Cemal Aytekin	Throughout this semester, I learned how to use lots of cool tools. Firstly, I have been assigned to create and maintain our high level project plan. So I used smartsheet to generate it and I regularly revise and update the plan. Also I used Proto.io to create some Mockups for both Web and Android platform. Another task was creating class diagram and some sequence diagrams. I used Lucidchart to create our class diagrams and some Sequence Diagrams. Finally, in the twitter API project, I used Python to generate our twitter bot “trendhist” which get first 10 trend topics and post a tweet including them by using API.

Öncel Keleş	In this course, I have got used to development tools for an application development project such as Lucidchart and Github. Github was probably the most useful, which we as a team shared our completed tasks and review each other's work, and I saw that Github is very developer friendly, especially for group projects. I used Github to create and edit the Wiki page, open&close issues and push commit to branches. I also tried out Lucidchart and Proto.io to create diagrams and mock ups for cases. I also got familiar with test cases and how they are done throughout team tasks and researches. With Volkan Yılmaz, we implemented the javascript part of the Twitter project to the front end.
Deniz Etkar	The only coding related activity I experienced throughout the span of this course was the Twitter Project. I contributed by writing the code for part of our project API where I and Akın worked in collaboration. We used java spring boot framework and the parts I was responsible for are the following: project controller, definition of some entities in our NoSQL database. I endorsed the use of a NoSQL database, MongoDB, when we were in the process of choosing a database management system. Because the format and the way in which it stores a record complements the use of dictionaries in JS (JavaScript). I have written a base testing code in the spring boot to create a basis for test related activities in our project.
Akın İlerle	The coding part of the course was in Twitter Project given. In that project, I implemented the web-server in the back-end with Deniz. I set up the project in spring boot, created a global exception handler and developed one of our two endpoints which returns the trending topics in a given day. I added Swagger to out project which visualises the endpoints and creates a documentation automatically. I created a MongoDB database in mLab(a database-as-a-service supplier) and a EC2 instance on AWS and deployed our web server there. Finally I reviewed team members code.

Hatice Melike Ecevit	Throughout this course, the coding work that I did was only in Twitter project. I have written a script for our Twitter bot trendhist to tweet and retweet on fridays about follow fridays. I also wrote some test cases to our web server in the back-end of our project using spring-boot. Before this project, I used proto.io for creating mockups on android platform and lucidchart for creating class diagram and some sequence diagrams. I also helped Cemal while creating our project plan so I've gotten familiar with smartsheet. This semester was the first time that I've used Github. I've learned about the wiki syntax and wiki pages. While we were opening issues, I learned about issue labeling and learned about git system also. I opened branches, merged branches, resolved conflicts and reviewed code.
Ece Ata	In this semester, I learned to use many tools and platforms. I tried to use GitHub for sharing our project to our group members; Lucidchart for creating project plan; Proto.io for mockups of our project. Also I contributed frontend part of our Twitter Project. I tried to use html and js. We created a page that shows trend topics of tweets.
Volkan Yılmaz	During this semester, I have learned a lot of things about the project management with my team. The first task assigned to me was creating the readme file so that people can easily understand the project roughly and surf through project in timeline on Github. I learnt which situations should we use issues and how labels should be so that each member can assign different priority to each task. With the help of this issues we can easily track our work. I also learned the planning makes clear our path which leads to final deliverable status of the project. We already planned the possible scenarios for users and we are going to test possible problems. Therefore we have already prepared the breaking points of this work. Moreover, I also learned how can we use the API of existing projects such as Google Maps, Facebook. While we are working on Twitter project I learned to read documentation and comment on how can I use so that the thought in my mind can be happened. I also notice sometimes the offering from Twitter does not work all the time. I searched that and find out sometimes when a lot request comes one after another Twitter restricts that. It can be originated from bot accounts and it needs to additional

	clicking to reach trending topic. I was on the frontend part of this project and I use javascript with Oncel Keles to connect backend with the website that we created. I read the documentation prepared by backend team of the project. Furthermore, I use several technologies to achieve tasks. For mockups, I use proto.io website which is very good at that. It has a lot of tools to make scenarios real, even we did not finish the project. I use lucidchart for my sequence diagram assessment and it makes easier these kind of planings with several tools inside. Finally, I realize this class is not just about the tasks and finishing those tasks before the deadline but rather the importance of group work and communication between us.
Ahmet Yasin Alp	I code only for Twitter Project although I learned a lot of useful tools. I implement a bot that pulls trending topics and their popular tweets from Twitter every night and pushes to our database. I choose Python and python-twitter library. For database connection I use pyMongo library. Some parts of code, we couldn't use the library so we swap to Twitter-RestfulApi. I use Proto.io for creating Android mockups
Halil Samed Çıldır	The main part of this course was to use software development tools like github and lucidchart. I have already used github before but in this course I have been able to use github with a team in Twitter API project. I helped in the UI part of our project and also I deployed the UI and bot parts of the project.

6. Documentation

Time: 17.09.2018

Objectives:

- Readme page is the initial page that summarizes all the work done throughout this project and it describes what this project is related to.
- Wiki page is the control page that allows surfing through the project in timeline. Also there are some personal wiki pages that describes each member of the group.
- Labels are crucial for communication professionally when there is a need to assigning or reviewing.
- Git Guide is designed to help members of this project and other people from other projects about GitHub.
- Issue styling is also important in order to tasks are clear.

Deliverables:

- README.md
- Group 1 Wiki
- Labels
- Git Guide
- Issue Styling

7. Communication

Communication Plan

Time: 24.03.2018

Objectives:

- In order to have a healthy and precise communication among the members of our group, determining rules and conventions that has been accepted by every member.
- Determining a communicator for our group in order to have efficient communication between the group and teaching staff.

Deliverables:

- Communication Plan

Meeting Type	Audience	Subject	Where	Delivery Method	Frequency
Weekly meeting	All team members	Weekly challenge discussion and task distribution	Any available room in BM	Face to face / Synchronous	Every Monday
Online meeting	All team members	Reviewing what we have done	Hangouts	Online / Synchronous	When needed
General	All team members	Weekly discussions about task of the week	Slack	Online / Asynchronous	Every week
General	All team members	Weekly building the agenda	Whatsapp	Online / Asynchronous	When needed to notify members
Project meeting	All team members	Weekly notes and problems	Github	Online / Asynchronous	When issues are created
Customer meeting	All team members and customer	Deciding on customer demands	Piazza	Online / Asynchronous	Ps hours

Figure 1: Communication Plan

8. Requirement Analysis

Time: 31.03.2018

Objectives:

- Objective #1: Identifying functional and non-functional requirements.
- Objective #2: Prepending a glossary of terms about the requirements.
- Objective #3: Getting feedback from the customer.

Deliverables:

- Requirements

Glossaries

- *Admin: A person who can manage other users' sharing and memories on the system.*

- *Unregistered User: A person who anonymously viewing and searching memories, but can't do any editing.*
- *Registered User: A person who has an account, can share memories and can see others' profile pages.*
- *Annotation: Explanatory content attached to an item. A comment or note that is left to a specified part of a memory which is relevant with the content.*
- *Username: A specific handle that is unique for each registered user.*
- *Password: A specific sequence of characters that will allow registered users to access to the system.*
- *Memory: Memory is the way users share a post in our app/website. It can contain images, videos, audios, texts and it has a specific time (this can be a time interval) and a location (this can be a region).*
- *Story: Text explanation of the memory.*
- *Profile Page: A page that is unique to each user where users can see the memories shared by the owner of that profile page.*

1.Functional Requirements

1.1. User Requirements

- *1.1.1. Unregistered Users*
  - *1.1.1.1. Register*
    - *1.1.1.1.1. Unregistered users shall be able to register to the system.*
    - *1.1.1.1.2. Registration form contains mandatory fields which are username, password and email address.*
    - *1.1.1.1.3. Unregistered Users shall be able to use authenticate with their Facebook accounts while registering.*
  - *1.1.1.2. Unregistered Users shall be able to search, browse for memories.*
  - *1.1.1.3. Unregistered Users should not be able to create new memories.*
  - *1.1.1.4. Unregistered Users should not be able to like or comment current memories.*

        - *1.1.1.5. Unregistered Users should not be able to do annotations on the memories.*
        - *1.1.1.6. Unregistered Users should not be able to visit any profile page.*
- *1.1.2. Registered Users*
    - *1.1.2.1. Registered users shall be able to login.*
    - *1.1.2.2. Registered users shall be able to logout.*
    - *1.1.2.3. Registered users shall be able to share memories.*
    - *1.1.2.4. Registered users shall be able to search in memories.*
    - *1.1.2.5. Registered users shall be able to change his/her password and email.*
    - *1.1.2.6. Registered users shall be able to leave a comment to a memory.*
    - *1.1.2.7. Registered users shall be able to like a memory.*
    - *1.1.2.8. Registered users shall be able to see a list of his/her memories in their profile page.*
    - *1.1.2.9. Registered users shall be able to add annotations to the memories.*
    - *1.1.2.10. Registered users shall be able to see a list of memories that he/she liked.*
- *1.1.3. Memory Features*
    - *1.1.3.1. Users shall be able to add location information to memories.*
        - *1.1.3.1.1. Location shall be able to have map support.*
    - *1.1.3.2. Users shall be able to add time information to memories.*
        - *1.1.3.2.1. Time information shall be able to contain the date or the time interval of the memory.*
        - *1.1.3.2.2. Time information shall be able to contain the date that the memory is added to the system.*
    - *1.1.3.3. Users shall be able to add photos to memories.*
    - *1.1.3.4. Users shall be able to add audio to memories.*
    - *1.1.3.5. Users shall be able to add video to memories.*
    - *1.1.3.6. Users shall be able to add story to memories.*
    - *1.1.3.7. Users should be able to add tags to memories.*
- *1.1.4. Search Conditions*
    - *1.1.4.1. Users shall be able to search memories based on location.*
    - *1.1.4.2. Users shall be able to search memories based on username.*

        - *1.1.4.3. Users shall be able to search memories based on a specific string*
        - *in stories.*
        - *1.1.4.4. Users shall be able to search memories based on the time of the memory.*
        - *1.1.4.5. Users shall be able to search memories based on tags.*
    - *1.1.5. Homepage*
        - *1.1.5.1. Memories shall be listed based on popularity which is determined by comments/likes in homepage.*
        - *1.1.5.2. Memories shall be shown based on memory location in homepage.*
        - *1.1.5.3. Users can go to a specific location and see all the memories there in the map.*
    - *1.1.6. Profile page*
        - *1.1.6.1. User's own profile page*
            - *1.1.6.1.1. Users shall be able to see their posted memories in their profile pages.*
            - *1.1.6.1.2. Users should be able to edit his/her memories in their profile pages.*
            - *1.1.6.1.3. Users should be able to delete his/her memories in their profile pages.*
            - *1.1.6.1.4. Users should be able to edit his/her bio in their profile pages.*
        - *1.1.6.2. Other user's profile pages*
            - *1.1.6.2.1. Users shall be able to see the memories other users posted in their profile page.*
    - *1.1.7 Annotations*
        - *1.1.7.1. Users shall be able to add annotation to images in a memory.*
        - *1.1.7.2. Users shall be able to add annotation to a specified part of a story in a memory.*
        - *1.1.7.3. Users shall be able to add annotation to a specified part of a comment in a memory.*

1.2. System Requirements

- *1.2.1. Recommendation*
    - *1.2.1.1. System shall recommend memories.*
        - *1.2.1.1.1. Recommendations shall be based on likes.*
        - *1.2.1.1.2. Recommendations shall be based on location.*

        - *1.2.1.1.3. Recommendations shall be based on tags.*
        - *1.2.1.1.4. Recommendations shall be based on the date of the memory.*
        - *1.2.1.1.5. Recommendations shall be based on the story text itself.*

2. Nonfunctional Requirements

- *2.1. Security*
    - *2.1.1. Database system shall be protected.*
    - *2.1.2. Privacy of user data shall be protected.*
    - *2.1.3. Passwords shall be at least 8 characters long and contain at least one number and one letter.*
    - *2.1.4. Passwords shall be stored in hashed format.*
- *2.2 Availability*
    - *2.2.1. The application is expected to have a web and mobile (Android) client.*
    - *2.2.2. The language of the application shall be in English.*
    - *2.2.3. Users shall navigate to every page within the application using at most 3 links.*
- *2.3 Annotation*
    - *2.3.1. The W3C Web Annotation Data Model shall be used to annotate items.*
- *2.4 Performance*
    - *2.4.1 The system should be able to respond to requests within 3 seconds with at least 16 mbit internet speed.*

9. Project Plan

Time: 25.02.2018

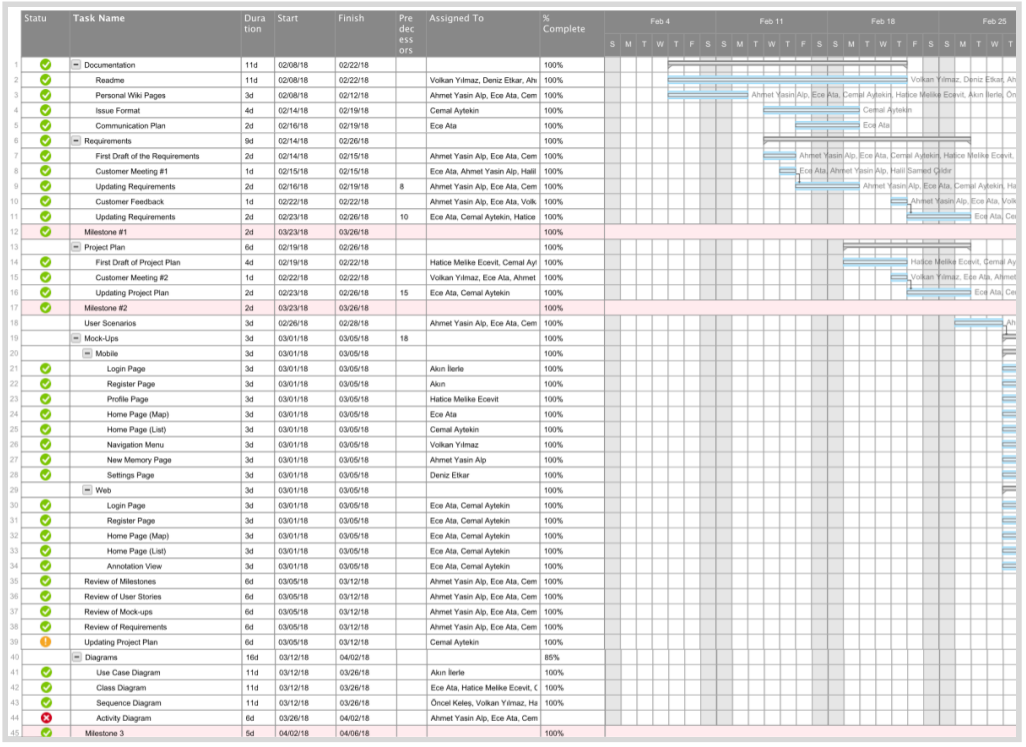
Objectives:

- Create a high level project plan that outlines the major activities and milestones, which helps us to track our progress with our plan as well as keep our plan up to date.
- The goal of the plan is to organize the work by estimating the effort and resources needed complete the tasks and to compare the execution (real life progress). This way we are able to assess how we are progressing.
- Also it helps figure out the following questions:
    - According to our plan, where should we be in our project now?

    - Where are we?
    - What is the difference?
    - What are our planning to do to make up the difference?
- Note that this plan will be a very high level plan at this point.

Deliverables:

- Project Plan



	Statu	Task Name	Duration	Start	Finish	Predecessors	Assigned To	% Complete
1	✓	Documentation	11d	02/08/18	02/22/18			100%
2	✓	Readme	11d	02/08/18	02/22/18		Volkan Yılmaz, Deniz Etkar, Ah	100%
3	✓	Personal Wiki Pages	3d	02/08/18	02/12/18		Ahmet Yasin Alp, Ece Ata, Cem	100%
4	✓	Issue Format	4d	02/14/18	02/19/18		Cemal Aytekin	100%
5	✓	Communication Plan	2d	02/16/18	02/19/18		Ece Ata	100%
6	✓	Requirements	9d	02/14/18	02/26/18			100%
7	✓	First Draft of the Requirements	2d	02/14/18	02/15/18		Ahmet Yasin Alp, Ece Ata, Cem	100%
8	✓	Customer Meeting #1	1d	02/15/18	02/15/18		Ece Ata, Ahmet Yasin Alp, Halil	100%
9	✓	Updating Requirements	2d	02/16/18	02/19/18	8	Ahmet Yasin Alp, Ece Ata, Cem	100%
10	✓	Customer Feedback	1d	02/22/18	02/22/18		Ahmet Yasin Alp, Ece Ata, Volk	100%
11	✓	Updating Requirements	2d	02/23/18	02/26/18	10	Ece Ata, Cemal Aytekin, Hatice	100%
12	✓	Milestone #1	2d	03/23/18	03/26/18			100%
13		Project Plan	6d	02/19/18	02/26/18			100%
14	✓	First Draft of Project Plan	4d	02/19/18	02/22/18		Hatice Melike Ecevit, Cemal Ayt	100%
15	✓	Customer Meeting #2	1d	02/22/18	02/22/18		Volkan Yılmaz, Ece Ata, Ahmet	100%
16	✓	Updating Project Plan	2d	02/23/18	02/26/18	15	Ece Ata, Cemal Aytekin	100%
17	✓	Milestone #2	2d	03/23/18	03/26/18			100%
18		User Scenarios	3d	02/26/18	02/28/18		Ahmet Yasin Alp, Ece Ata, Cem	100%
19		Mock-Ups	3d	03/01/18	03/05/18	18		100%
20		Mobile	3d	03/01/18	03/05/18			100%
21	✓	Login Page	3d	03/01/18	03/05/18		Akın İlerle	100%
22	✓	Register Page	3d	03/01/18	03/05/18		Akın	100%
23	✓	Profile Page	3d	03/01/18	03/05/18		Hatice Melike Ecevit	100%
24	✓	Home Page (Map)	3d	03/01/18	03/05/18		Ece Ata	100%
25	✓	Home Page (List)	3d	03/01/18	03/05/18		Cemal Aytekin	100%
26	✓	Navigation Menu	3d	03/01/18	03/05/18		Volkan Yılmaz	100%
27	✓	New Memory Page	3d	03/01/18	03/05/18		Ahmet Yasin Alp	100%
28	✓	Settings Page	3d	03/01/18	03/05/18		Deniz Etkar	100%
29		Web	3d	03/01/18	03/05/18			100%
30	✓	Login Page	3d	03/01/18	03/05/18		Ece Ata, Cemal Aytekin	100%
31	✓	Register Page	3d	03/01/18	03/05/18		Ece Ata, Cemal Aytekin	100%
32	✓	Home Page (Map)	3d	03/01/18	03/05/18		Ece Ata, Cemal Aytekin	100%
33	✓	Home Page (List)	3d	03/01/18	03/05/18		Ece Ata, Cemal Aytekin	100%
34	✓	Annotation View	3d	03/01/18	03/05/18		Ece Ata, Cemal Aytekin	100%
35	✓	Review of Milestones	6d	03/05/18	03/12/18		Ahmet Yasin Alp, Ece Ata, Cem	100%
36	✓	Review of User Stories	6d	03/05/18	03/12/18		Ahmet Yasin Alp, Ece Ata, Cem	100%
37	✓	Review of Mock-ups	6d	03/05/18	03/12/18		Ahmet Yasin Alp, Ece Ata, Cem	100%
38	✓	Review of Requirements	6d	03/05/18	03/12/18		Ahmet Yasin Alp, Ece Ata, Cem	100%
39	!	Updating Project Plan	6d	03/05/18	03/12/18		Cemal Aytekin	100%
40		Diagrams	16d	03/12/18	04/02/18			85%
41	✓	Use Case Diagram	11d	03/12/18	03/26/18		Akın İlerle	100%
42	✓	Class Diagram	11d	03/12/18	03/26/18		Ece Ata, Hatice Melike Ecevit, C	100%
43	✓	Sequence Diagram	11d	03/12/18	03/26/18		Öncel Keles, Volkan Yılmaz, Ha	100%
44	✗	Activity Diagram	6d	03/26/18	04/02/18		Ahmet Yasin Alp, Ece Ata, Cem	
45	✓	Milestone 3	5d	04/02/18	04/06/18			100%

Figure 2: A part of project plan

10. View Analysis

View Analysis

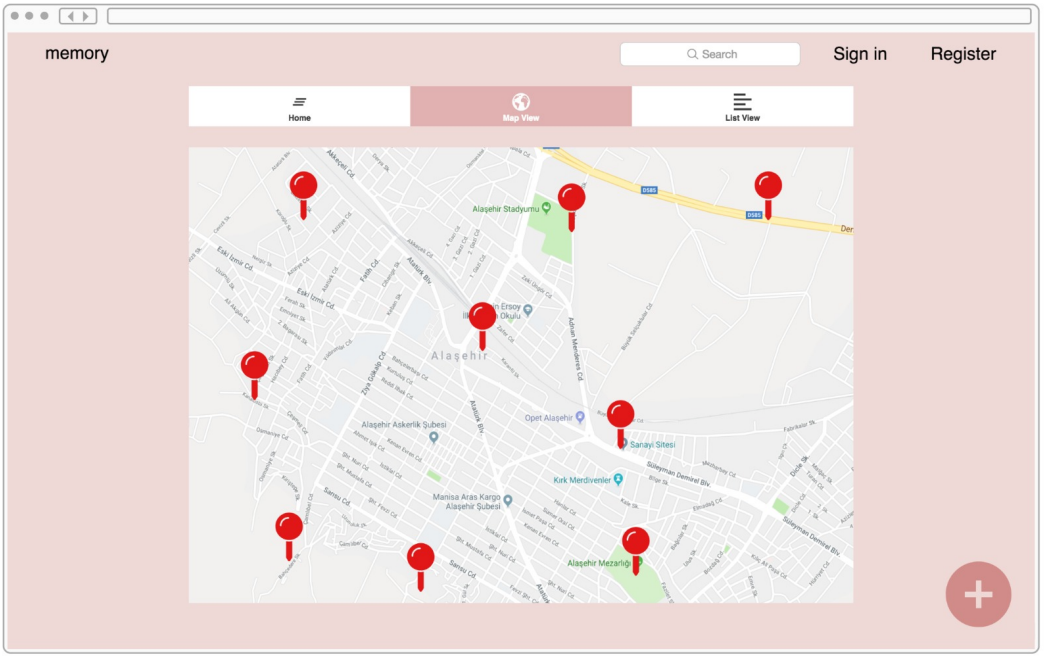
Time: 19.05.2018

Objectives:

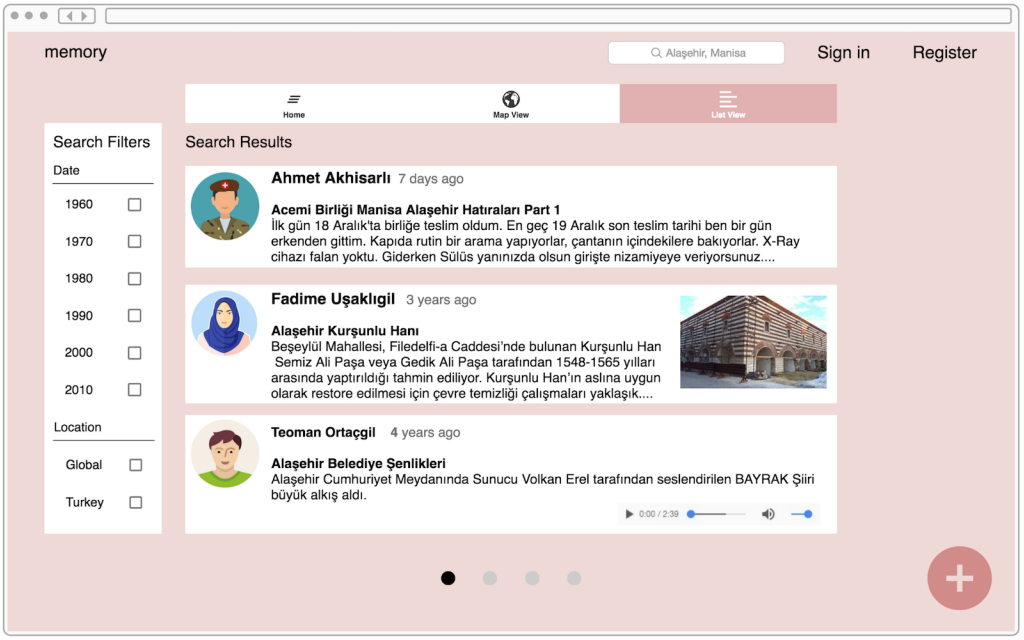
- In order to get into the context of the application as much as possible, we created mock-ups.
- To determine and understand our application's audience, we created user personas. And stated their acceptance criteria.
- With these criterias we wrote use case scenarios to help us understand our project's requirements.

Deliverables:

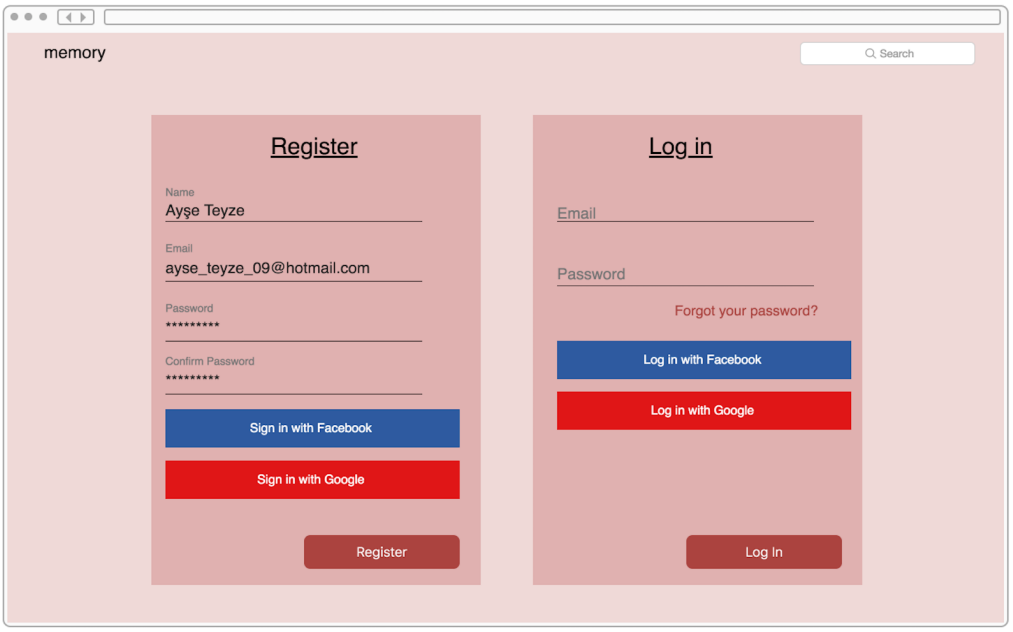
- Mockups:
  - Web
  - Homepage Map View for Unregistered User



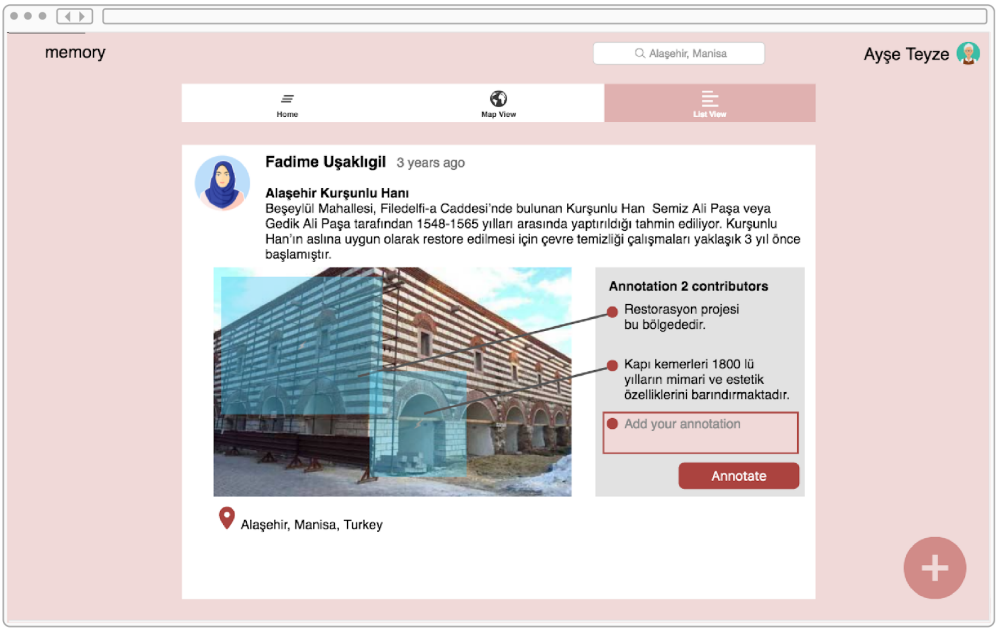
- Search Results Page for Unregistered User



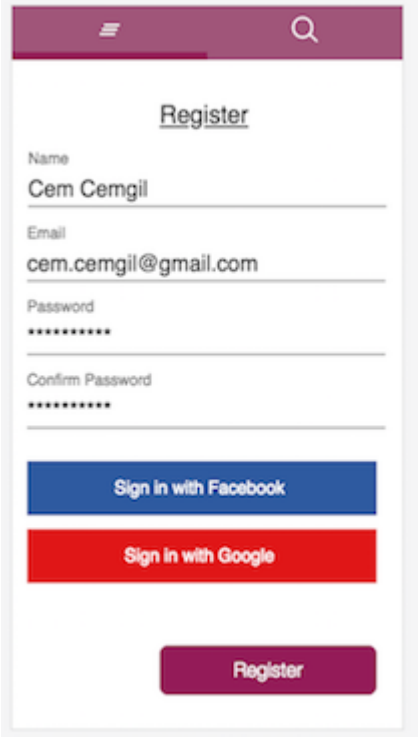
- Register and Login Page



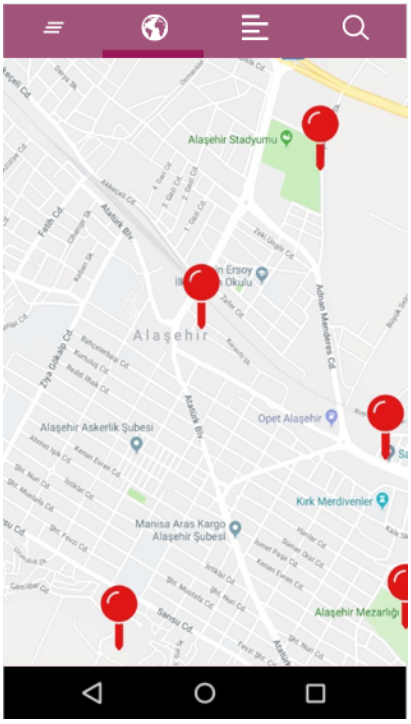
- Annotation Sample

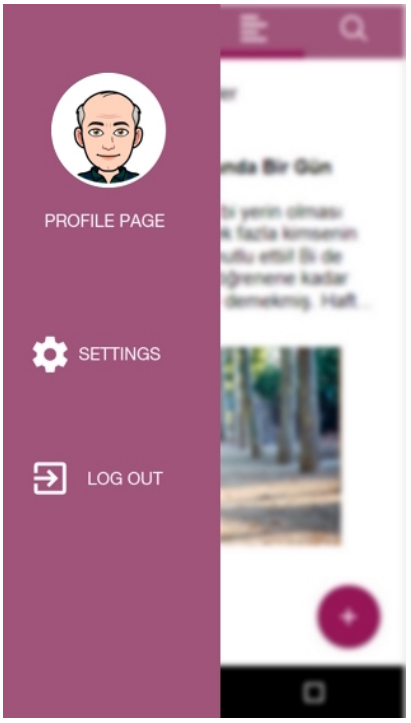


- [Mobile]


Register
Name
Cem Cemgil
Email
cem.cemgil@gmail.com
Password
Confirm Password
Sign in with Facebook
Sign in with Google
Register

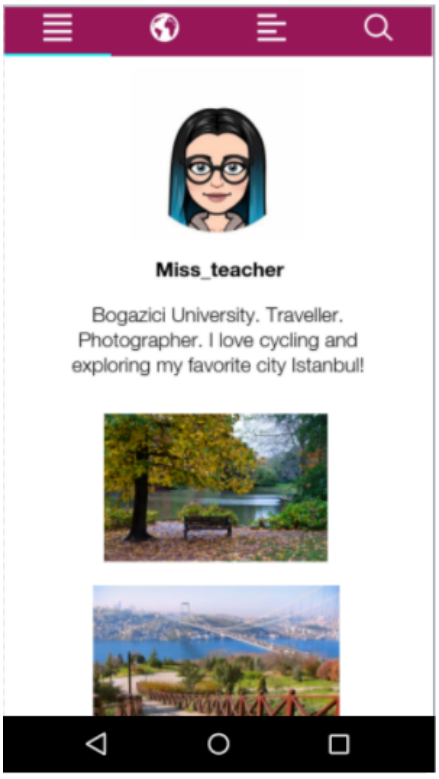

Cem Cemgil
Professor @unibogazici. I am very interested in old technology and old technological devices. You can check out my tech blog cemgillovestech.com

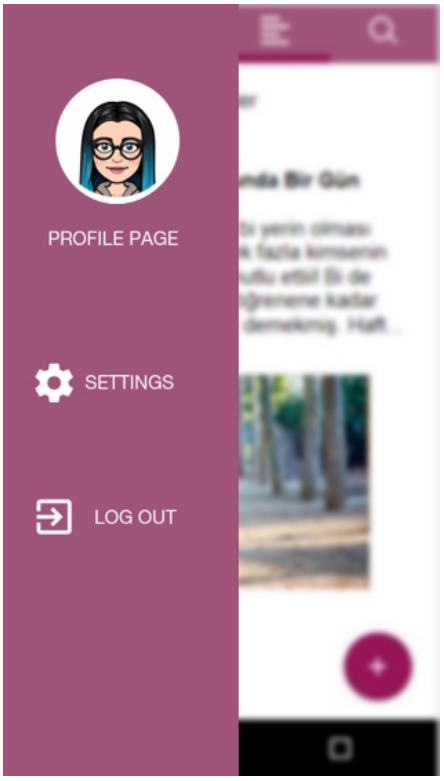

Alaşehir Stadyumu
Alaşehir
Alaşehir Askerlik Şubesi
Opet Alaşehir
Kırk Merdivenler
Manisa Aras Kargo Alaşehir Şubesi
Alaşehir Mezarlığı

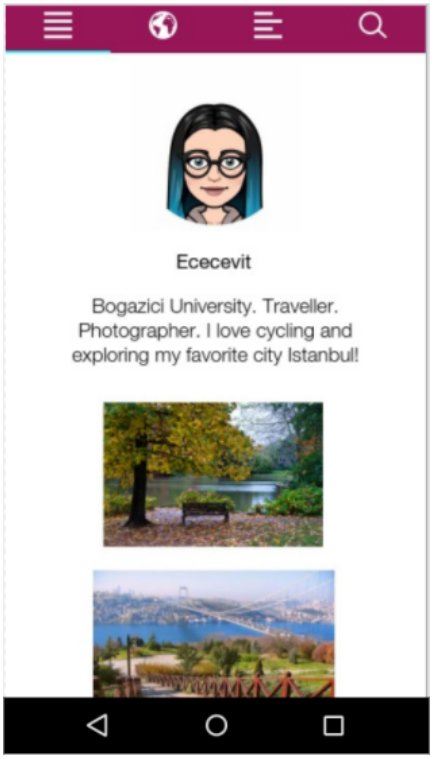

PROFILE PAGE
SETTINGS
LOG OUT

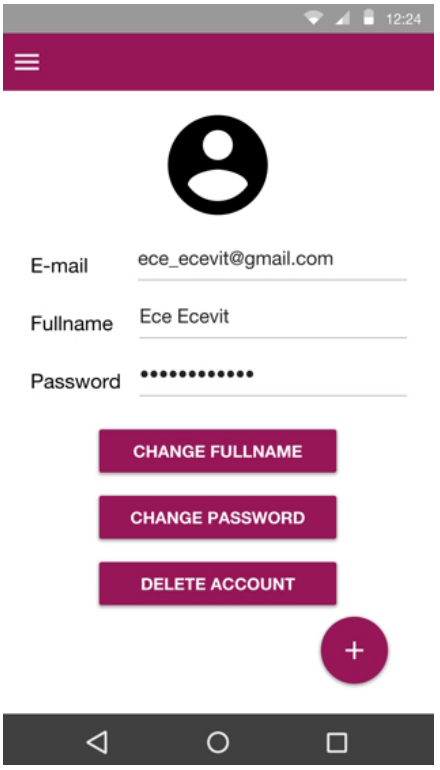

13:07
Header
Floppy disk
Date
1971 - 2007
technology
computer
Story
A floppy disk, also called a floppy, diskette, or just disk, is a type of disk storage composed of a disk of thin and flexible magnetic stor...


Cemre Kalender
Posted 28.02.2018
Atatürk Arberotumunda Bir Gün
Nisan 2016
İstanbul içinde böyle bi yerin olması bizi çok mutlu etti. Pek fazla kimsenin bilmemesi daha da mutlu ettii! Bi de "arboretum" demeyi öğrenene kadar canım çıktı.. Ağaç ev demekmiş. Ha...
Devamını Oku
13k
65
John Doe
27.02.2018


Miss_teacher
Bogazici University. Traveller. Photographer. I love cycling and exploring my favorite city Istanbul!


PROFILE PAGE
SETTINGS
LOG OUT





11. Design

Time: 06.04.2018

Objectives:

- Create diagrams for 3 use case scenarios.
- Create class diagrams
- Create Sequence Diagrams

Deliverables:

- Use Case Scenarios

Scenario #1: Ayşe Teyze

Persona

Ayşe Teyze is a middle aged person. Her age is about 55-60. She lives in Nazilli, Aydın. She enjoys living in Nazilli because she loves nature and hates

big cities. She is interested in history. She likes reading books and watching movies.

Story

After E-devlet announced the family trees, she looked from which villages their parents are coming. Then, she learnt they used to live in Alaşehir, Manisa and shared this information with her grandchild and then she wanted to find out more about Alaşehir. Her grandchild suggested her our app. That's how she came across the Living History project. She entered the app on WEB, then searched for words 'Alaşehir', 'Manisa'. Our app returned some memories according to applied filter. She recognized one memory from her childhood and picked this memory. Afterwards, she read some annotations for that particular memory. Then, she wanted to create annotation about her recollection of this memory, however she could not. It asked from her to register. As a final act, she registered.

Acceptance Criteria

As a middle aged person, I want to search for memories, view memories and annotations, add annotations to memories so that I can find out more about my ancestral history.

Scenario #2: Cem Cemgil

Persona

Cem Cemgil is a professor at Bogazici University, department of computer engineering. His is about 45 years old. He enjoys sharing old computer and technology stuff and he writes his old memories in his blog about technology and computer science. He is mostly interested in these topics. He even has a collection of 80's computer disks to write codes. He tells stories about how programming codes were once written in cards on his lectures.

Story

One student of him read his blogs and suggested him to register to the Living History project. He downloaded the app to his Android smartphone. He registered and created 2 memories about Netaş which he had planned before to share on his blog. He added some tags to his memories. He realized that he unfortunately had some typo in it. He clicked to navigator menu to reach his profile page, then selected his memory to edit it.

Acceptance Criteria

As a professor and a computer geek I want to register, create memories with tags, edit memories from my profile page so that I can share my computer related knowledge with other people without any errors.

Scenario #3: Ece Ecevit

Persona

Ece is a student at Bogazici University-Pred department. She is 21 years old. She enjoys living in Istanbul because she finds it to be such an amazing city. Every weekend she visits new attractions around Istanbul. She enjoys taking pictures and cycling. She would love to be a blogger/influencer someday.

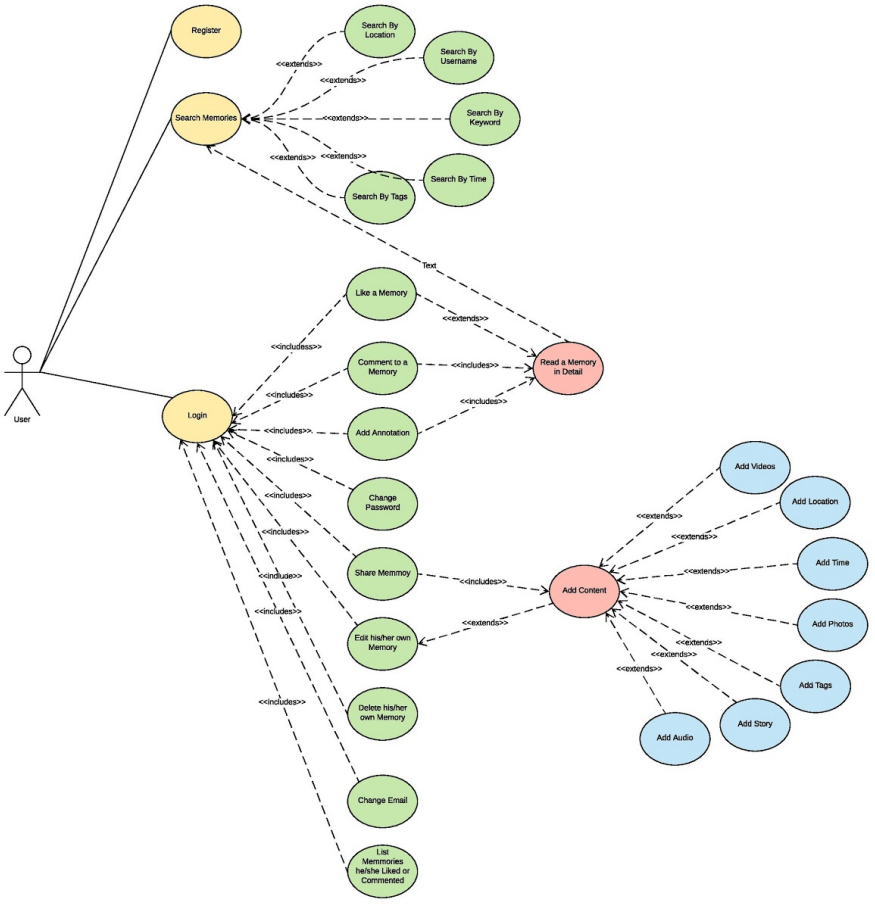
Story

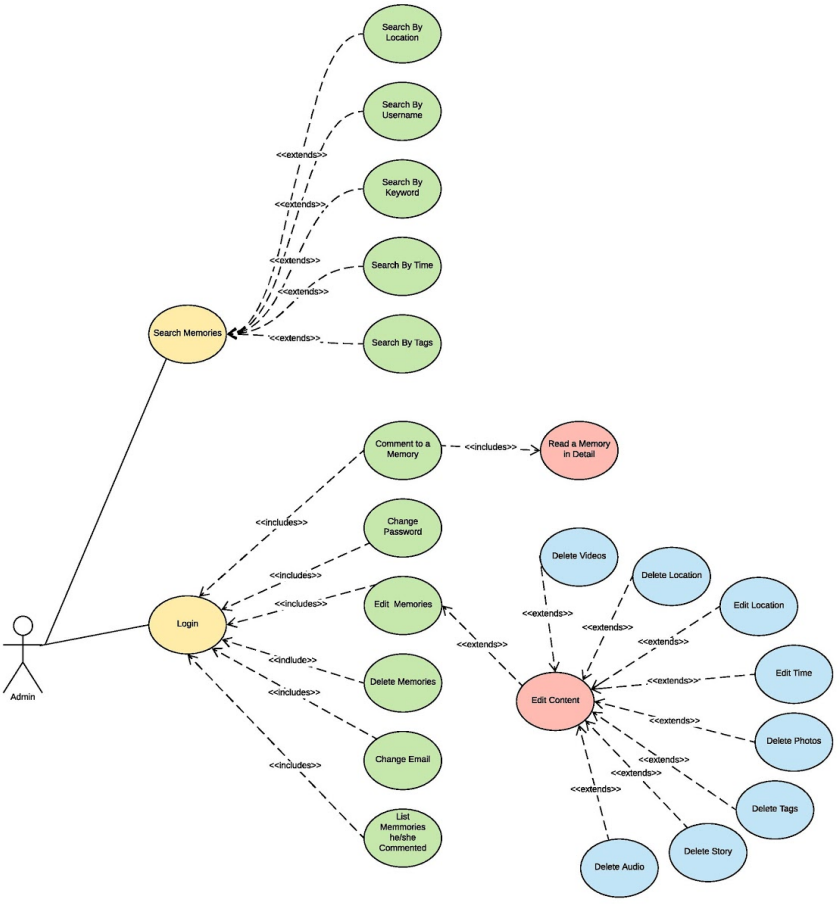
She was looking for an app to achieve the goal mentioned above, and found ours. She posted memories about mostly everything! She loves getting comments/likes. One day while hanging out on her phone, she opened the app. She immediately signed in and went to homepage. While cruising around the homepage she saw a popular post by a famous blogger which had a lot of likes. She thought she wanted to be famous just like her. She opened her profile page and gave herself a better bio and a better name, and now that it was time for her to be a famous blogger!

Acceptance Criteria

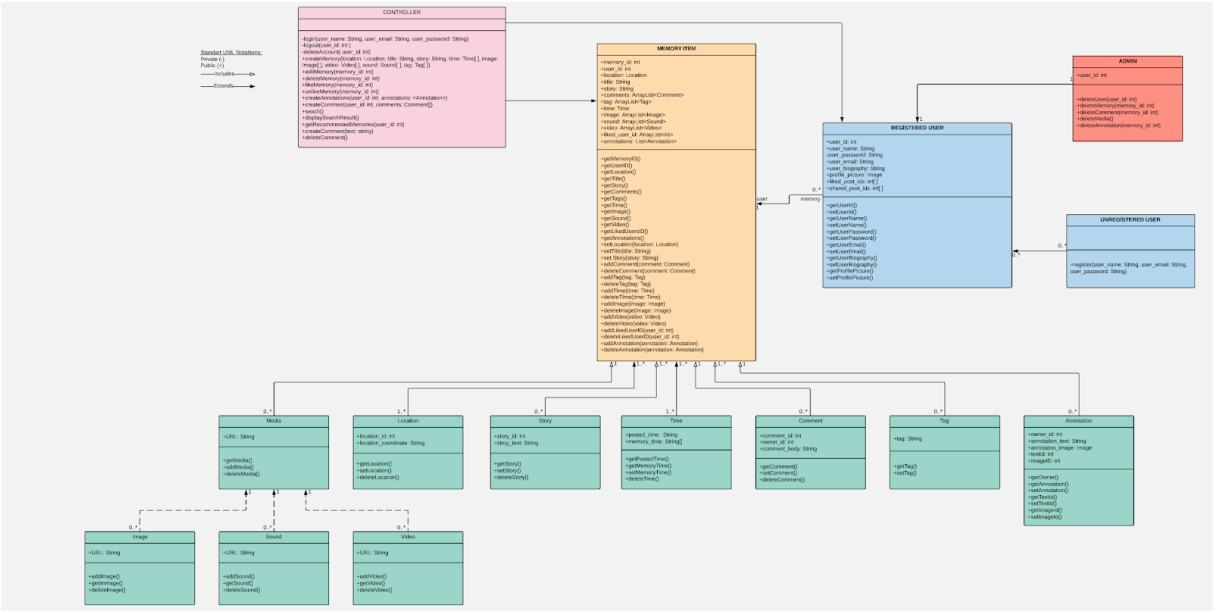
As a university student I want to sign in, see what is most popular, change my bio and name from profile page so that I can one day be the most popular user.

- Use Case Diagram


Register
Search By Location
Search By Username
Search Memories
Search By Keyword
Search By Time
Search By Tags
<<extends>>
Text
Like a Memory
Comment to a Memory
Add Annotation
Read a Memory in Detail
<<includes>>
User
Login
Change Password
Share Memmoy
Edit his/her own Memory
Delete his/her own Memory
Change Email
List Memmories he/she Liked or Commented
Add Content
Add Videos
Add Location
Add Time
Add Photos
Add Tags
Add Story
Add Audio



- [Class Diagram]



- Sequence Diagram

User Login Sequence Diagram

Actor:

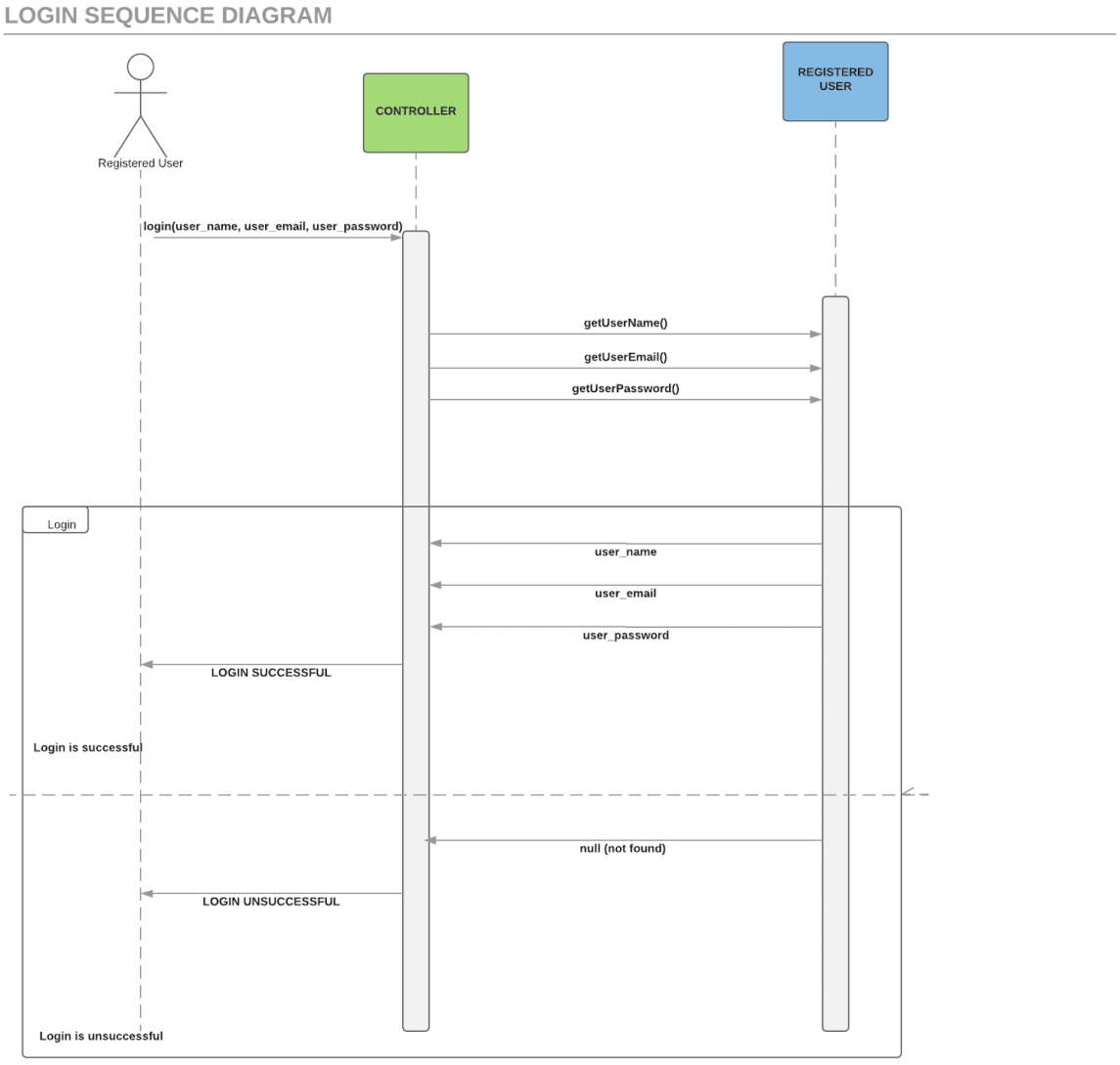
Registered User

Preconditions:

User must have an active account and is logged out.

Postconditions:

User is now logged in his/her account.



Create Memory Sequence Diagram

Actor:

Registered User

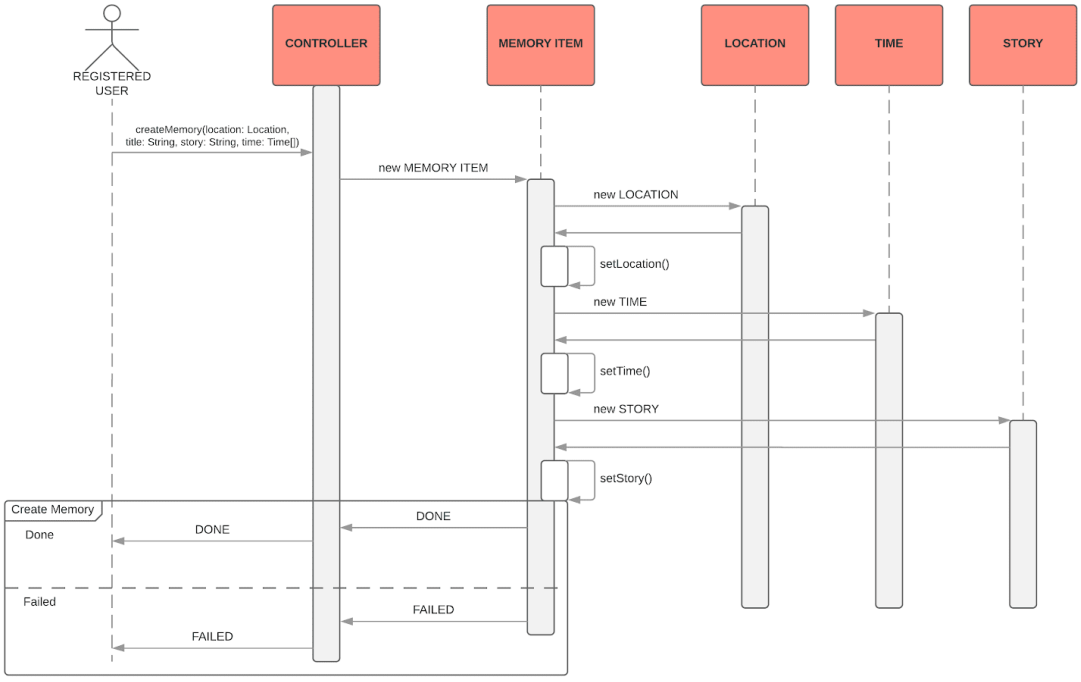
Preconditions:

User must have an active account and is logged in.

Postconditions:

User has now created a new memory.

CREATE MEMORY SEQUENCE DIAGRAM



Add Annotation Sequence Diagram

Actor:

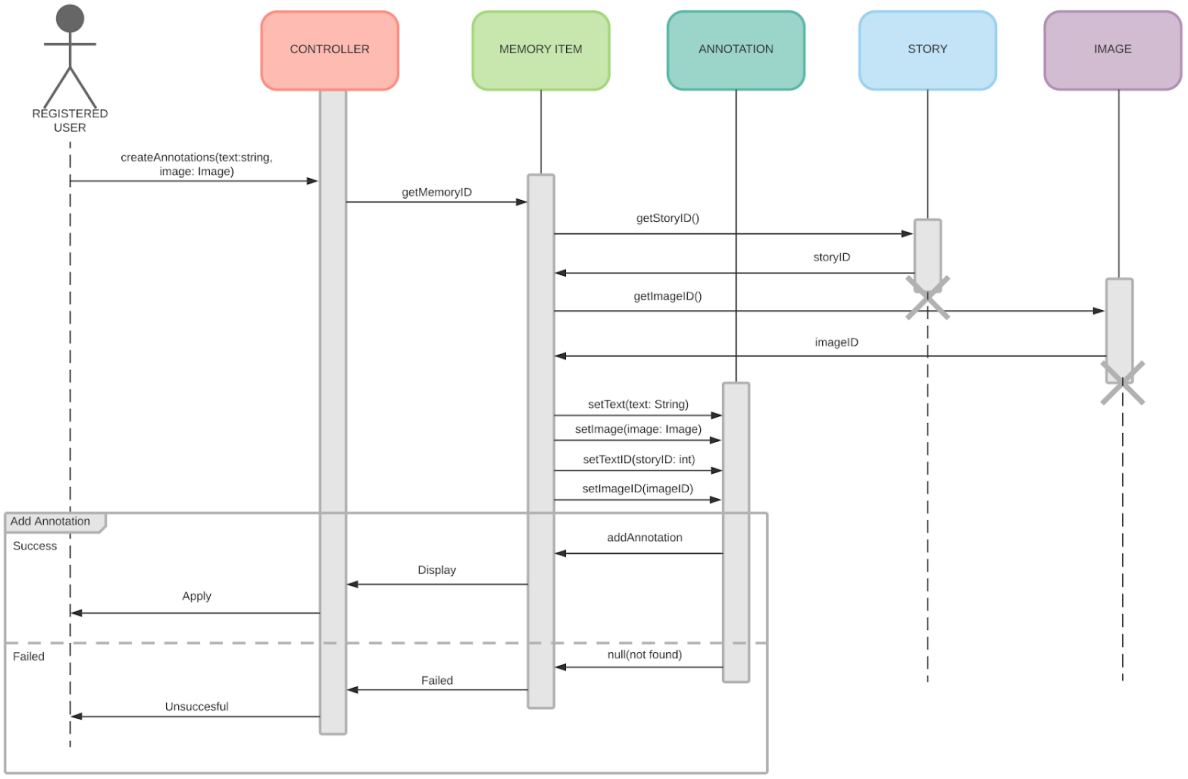
Registered User

Preconditions:

User must have an active account and find a memory item.

Postconditions:

User is now added annotation in someone's memory item.



Edit Comment Sequence Diagram

Actor:

Registered User

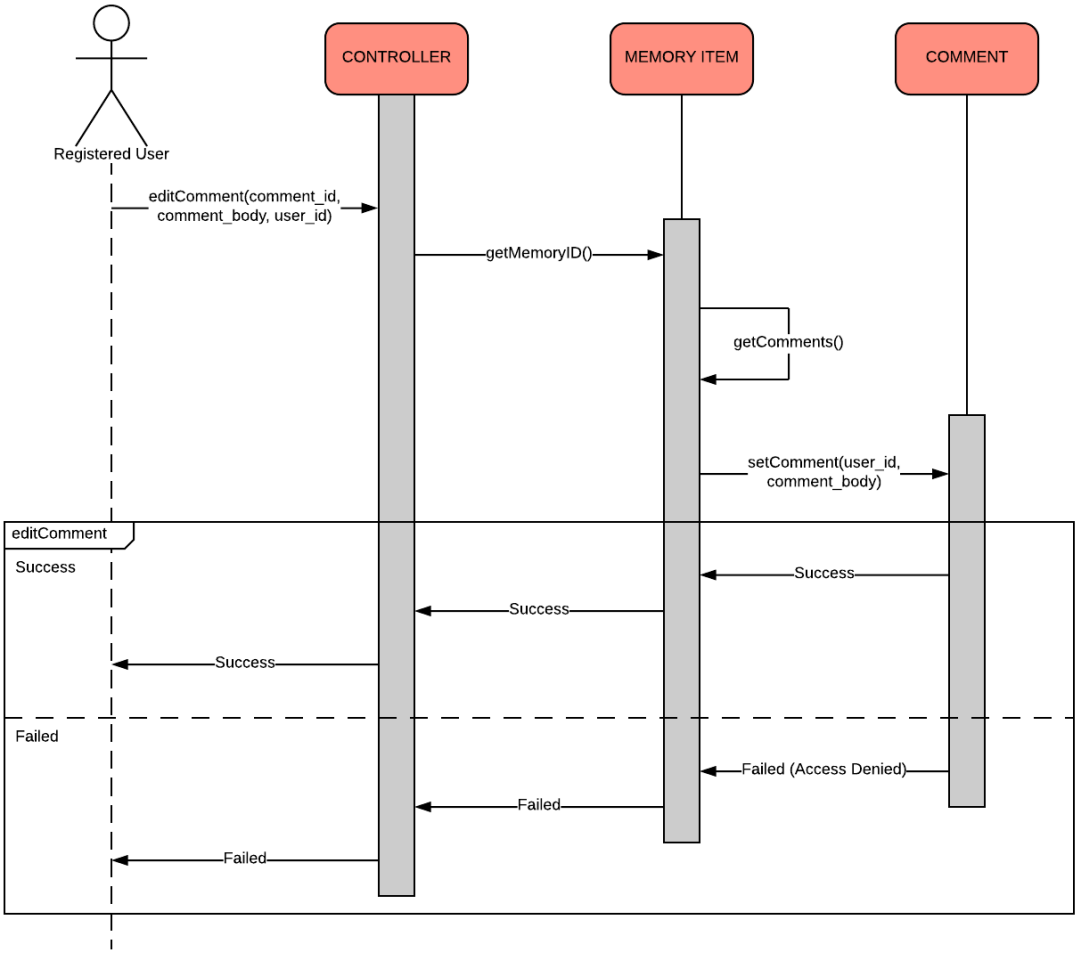
Preconditions:

User must have an active account and should have access to edit comment.

Postconditions:

User is now edited the comment.

EDIT COMMENT SEQUENCE DIAGRAM



Delete Image Sequence Diagram

Actor:

Registered User

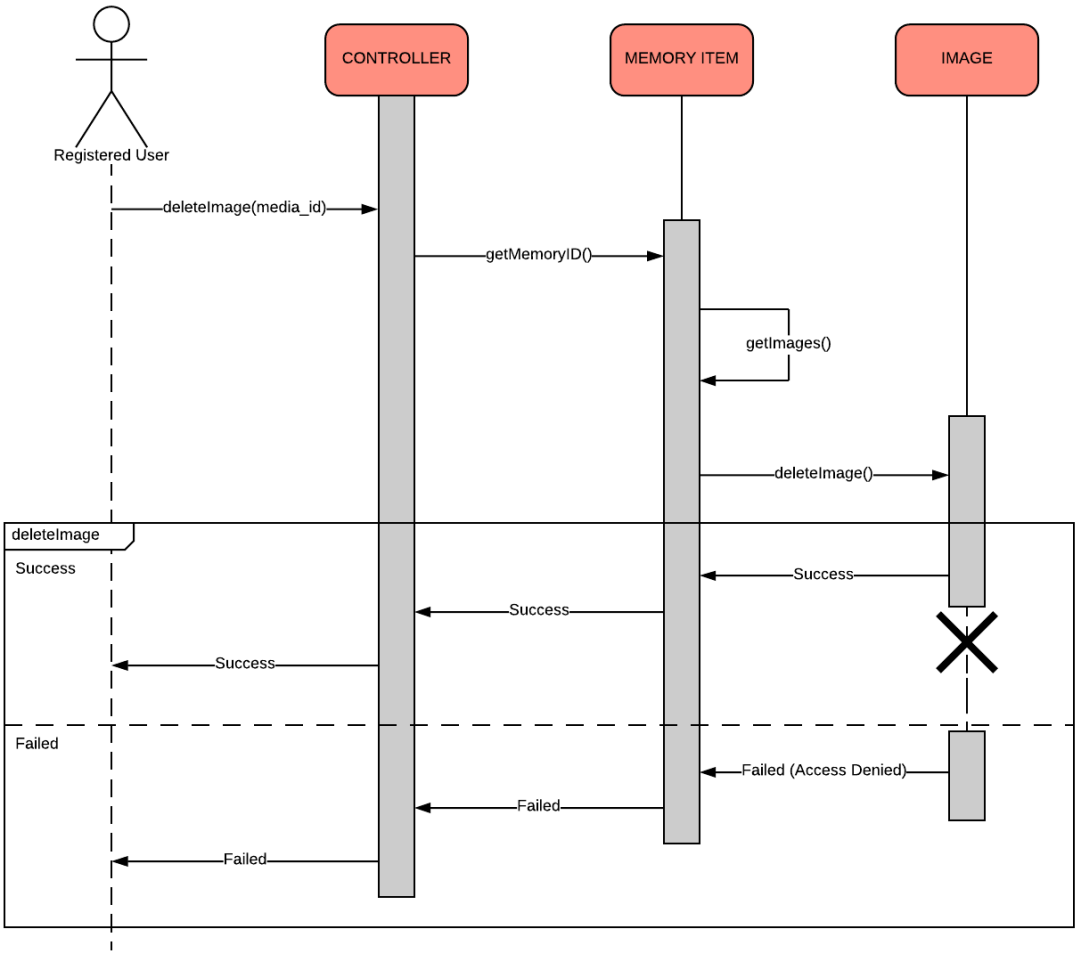
Preconditions:

User must have an active account and should have access to delete image.

Postconditions:

User is now deleted the image from the memory.

DELETE IMAGE SEQUENCE DIAGRAM



12. Testing

Time: 31.03.2018

Objectives:

- Design how our system should perform.
- Figure out specific input/output relationships of our use cases.
- Itemize each step with realistic input and expected output.

Deliverables:

- Test Cases

TestCase ID: EDIT_MEMORY

- *Test Priority: High*
- *Test Scenario: User wants to change the story in the memory.*
- *Test Purpose: To see if editing story in a memory works.*
- *Designed by: Deniz Etkar*
- *Design date: 29/03/2018*
- *Executed by:*
- *Execution date:*
- *Related Requirements: 1.1.6.1.2.*
- *Preconditions: User is logged in to the account.*

Step	***Test Step***	***Test Data***	***Expected Data***	***Actual Output***
Step #1	*Start editing a memory.*	*User clicks edit button of their memory.*	*The edit memory page is opened.*	
Step #2	*Type edited story.*	*User makes changes to story text box.*	*If the story is non-empty then the memory is ready to be edited.*	
Step #3	*Finish editing the memory.*	*User clicks save changes button of the edit page.*	*Story of the memory has been changed to the edited story in step 2.*	

TestCase ID: CREATE_MEMORY

- *Test Priority: High*
- *Test Scenario: User wants to create a new memory.*
- *Test Purpose: To create a memory.*
- *Designed by: Ece Ata*
- *Design date: 30/03/2018*
- *Executed by: -*
- *Execution date: -*
- *Related Requirements: 1.1.2.2. and 1.1.3.*

- *Preconditions: User is logged in to the account.*

Step	***Test Step***	***Test Data***	***Expected Data***	***Actual Output***
Step #1	*Start creating a memory*	*Users click the add memory button*	*New memory page is opened*	
Step #2	*Add any class of memory*	*Users add all instance of a memory*	*All instance are added*	
Step #3	*Finish creating a memory*	*Users click done button*	*New memory shows at the main page*	

- *Postcondition: New memory show to all users.*

TestCase ID: LIKE

- *Test Priority: Medium*
- *Test Scenario: User wants to like a memory.*
- *Test Purpose: To like a memory.*
- *Designed by: Ece Ata*
- *Design date: 30/03/2018*
- *Executed by: -*
- *Execution date: -*
- *Related Requirements: 1.1.2.6.*
- *Preconditions: User is logged in to the account.*

Step	***Test Step***	***Test Data***	***Expected Data***	***Actual Output***
Step #1	*Like a memory*	*Users click the like memory button*	*Name of the liker is shown at memory*	

- *Postcondition: Liker is shown at the bottom of a memory.*

TestCase ID: CHANGE_PASSWORD

- *Test Priority: HIGH*
- *Test Scenario: User wants to change his/her password.*
- *Test Purpose: Changing password*
- *Designed by: Cemal Aytekin*
- *Design date: 31.03.2018*
- *Executed by: -*
- *Execution date: -*
- *Related Requirements: 1.1.2.4.*
- *Preconditions: User must be logged in to the account.*

Step	***Test Step***	***Test Data***	***Expected Data***	***Actual Output***
Step #1	*User wants to see profile settings*	*"Profile Settings" button*	*Profile Options page will be shown*	*---*
Step #2	*User wants to change his/her password*	*"Change password" button*	*Change password page is shown*	*---*
Step #3	*User will enter his/her previous password*	*previous password = "<previous_password>"*	*---*	*---*
Step #4	*User will enter his/her new password twice*	*new password = "<new_password>"*	*---*	*---*

Step #5	*User will complete the process*	*"Change" button*	*The password has been updated*	*---*

- *Postcondition: Password has been updated*

TestCase ID: ADD_ANNOTATION

- *Test Priority: MEDIUM*
- *Test Scenario: User wants to add an annotation*
- *Test Purpose: Adding annotation*
- *Designed by: Cemal Aytekin*
- *Design date: 31.03.2018*
- *Executed by: -*
- *Execution date: -*
- *Related Requirements: 1.1.2.9. and 1.1.7.*
- *Preconditions: User muss be logged in to the account.*

Step	***Test Step***	***Test Data***	***Expected Data***	***Actual Output***
Step #1	*User wants to add annotation to an area*	*Everywhere on an image or a text area*	*---*	*---*
Step #2	*User type a text or add an image*	*"Annotate" Button*	*Annotation will be on the screen*	*---*

- *Postcondition: Annotations on the specified area has been updated.*

TestCase ID: EDIT BIO

- *Test Priority: Medium*
- *Test Scenario: User wants to edit her/his bio.*
- *Test Purpose: Editing user bio.*
- *Designed by: Mr. Keleş*
- *Design date: 31.03.2018*

- *Executed by: -*
- *Execution date: TBD*
- *Related Requirements: 1.1.6.1.4.*
- *Preconditions: User must have a registered account and logged in.*

Step	***Test Step***	***Test Data***	***Expected Data***	***Actual Output***
Step #1	*User opens profile page*	*User clicks profile page button*	*Profile page opens up*	*---*
Step #2	*User wants to open edit bio page*	*User clicks on edit bio*	*Edit bio page opens up*	*---*
Step #3	*User wants to change/edit his/her bio*	*User makes change in bio textbox and clicks save*	*User profile page with the new edited bio is displayed*	*---*

- *Postcondition: User bio is changed*

TestCase ID: SEARCH_BY_FILTERS

- *Test Priority: Medium*
- *Test Scenario: User wants to search by filtering.*
- *Test Purpose: To see if a user can search by filters*
- *Designed by: Hatice Melike Ecevit*
- *Design date: 31.03.2018*
- *Executed by:-*
- *Execution date:-*
- *Related Requirements: 1.1.4.*
- *Preconditions: ---*

Step	***Test Step***	***Test Data***	***Expected Data***	***Actual Output***
Step #1	*User wants to search with a filter*	*Search area*	*Cursor will be on the search area*	*---*
Step #2	*User starts to type some filters*	*Keywords that has been written by the user*	*-*	*---*
Step #3	*User selects the filter type that he/she wants to filter by*	*Selected filter*	*Filtered timeline will be shown to user*	*---*

- *Postcondition: In the filtered memory page, memories will be sorted in terms of popularity.*

TestCase ID: LOG_OUT

- *Test Priority: Low*
- *Test Scenario: User wants to log out from the application*
- *Test Purpose: To let user log out from the app.*
- *Designed by: Hatice Melike Ecevit*
- *Design date: 31.03.2018*
- *Executed by:-*
- *Execution date:-*
- *Related Requirements: 1.1.2.2.*
- *Preconditions: User needs to be logged in.*

Step	***Test Step***	***Test Data***	***Expected Data***	***Actual Output***
Step #1	*User wants to log out from the app*	*Log out button*	*-*	*-*

Step #2	*User presses the log out button*	*Log out button*	*User will log out from the app*	*-*

- *Postcondition: ---*

TestCase ID: SIGN_UP

- *Test Priority: High*
- *Test Scenario: User wants to sign up*
- *Test Purpose: To see if users are be able to sign up*
- *Designed by: Ahmet Yasin Alp*
- *Design date: 31.03.2018*
- *Executed by: N/A*
- *Execution date: N/A*
- *Related Requirements: 1.1.1.1*
- *Preconditions: N/A*

Step	***Test Step***	***Test Data***	***Expected Data***	***Actual Output***
Step #1	*User presses to sign up button*	*Sign up button*	*Sign up page will open*	*-*
Step #2	*User fill sign up form*	*Username, password and e-mail address*	*-*	*-*
Step #3	*User presses to register button*	*Register button*	*Confirmation massage will be shown and user will be redirected to main page*	*-*

- *Postcondition: User will be able to sign with username and password*

TestCase ID: SIGN_IN

- *Test Priority: High*
- *Test Scenario: User wants to sign in*
- *Test Purpose: To see if users are able to sign in*
- *Designed by: Ahmet Yasin Alp*
- *Design date: 31.03.2018*
- *Executed by: N/A*
- *Execution date: N/A*
- *Related Requirements: 1.1.2*
- *Preconditions: User must be registered before the sign in*

Step	***Test Step***	***Test Data***	***Expected Data***	***Actual Output***
Step #1	*User presses to sign in button*	*Sign in button*	*Sign in page will open*	*-*
Step #2	*User fill username and password fields*	*Username and password*	*-*	*-*
Step #3	*User presses to sign in button*	*Sign in button*	*User will be redirected to main page*	*-*

- *Postcondition: User will be able to see his/her main page*

TestCase ID: DISPLAY RECOMMENDED MEMORY

- *Test Priority: Medium*
- *Test Scenario: User wants to see recommended memories*
- *Test Purpose: In order to check whether recommendation system works or not*
- *Designed by: Volkan YILMAZ*
- *Design date: 01.04.2018*
- *Executed by: N/A*
- *Execution date: N/A*
- *Related Requirements: 1.2.1.*

- *Preconditions: User must have a registered account and logged in.*

Step	***Test Step***	***Test Data***	***Expected Data***	***Actual Output***
Step #1	*User presses lists button in Nav Menu*	*Lists button in Nav Menu*	*Flowing page opens up and some recommended memories are loaded and displayed*	*---*
Step #2	*User swipes up on the screen*	*Swiping up action*	*New recommended memories are loaded and displayed*	*---*

- *Postcondition: In the flowing memories page, memories will be sorted in terms of recommendation algorithm*

TestCase ID: COMMENT

- *Test Priority: High*
- *Test Scenario: User comments on an existing story*
- *Test Purpose: To ensure that commenting functionality works*
- *Designed by: Akın İlerle*
- *Design date: 01/04/2018*
- *Executed by: -*
- *Execution date: -*
- *Related Requirements: 1.1.2.6.*
- *Preconditions: User is logged in.*

Step	***Test Step***	***Test Data***	***Expected Data***	***Actual Output***
Step #1	*Open a memory*	*Click or tap on a memory in the main feed*	*Detailed memory page is opened*	
Step #2	*Go to the comments section*	*Scroll to the comments section*	*Comments on the memory is shown*	
Step #3	*Start creating a comment*	*Click or tap on new comment box.*	*Keyboard appears (on mobile)*	
Step #4	*Write a comment*	*Type a comment*	*The comment is shown on the screen*	
Step #5	*Finish writing the comment*	*Click or tap on send button*	*A confirmation appears and the comment is now listed with other comments*	

- *Postcondition: The database is updated.*

13. Software Development Infrastructure

In the twitter API project, I have coded our twitter bot “trendhist” which get first 10 trend topics and post a tweet including them by using API. Firstly, I signed up for a twitter account and gave the name trendhist. I created an application and got credentials for this account. Then I used those tokens and keys to login twitter account API. In the body part of the code, I sent query to get first 10 trend topics. After getting topics, I merged them and post the tweet which includes 10 topic by updating status. Finally another tweet is posted which announces our trend topic web site.

Cemal Aytekin

In our api assignment, I have coded a script for our twitter bot trendhist to retweet and tweet on fridays about follow fridays. I used our account’s credentials to login twitter account API. Then in a while loop, I fetched 100 tweets that have “#followfridays’ in them using tweepy. After that I made our bot tweet the first one using the api. I also wrote some test cases to our web server in backend using spring boot.

Hatice Melike Ecevit

We assigned a Twitter Api Project in this semester. I have coded frontend part of our web page that shows trend topics of twitter. Me and Samed have written Html, Css and Js to create this page. This page includes Worldwide and Turkey parts. Also, there are datepicker button so users can be able to see past trend topics.

Ece Ata

I assigned to Twitter Api Project’s backend. I create a bot that gets trending topics and tweets every night and stores on database. I use python and libraries for MongoDb and Twitter Api.

Ahmet Yasin Alp

Our web server has 2 endpoints: getTrendingTopics and getTweets. First endpoint takes 2 arguments: date and preference of worldwide or in Turkey. It queries the MongoDB supplied by mLab and returns top trending topics of the day in given reigion(TR or worldwide). The second endpoint takes the trendingTopicId as argument and returns its most popular tweets in HTML format. Our API documentation can be seen in following link: http://54.147.221.99:8080/swagger-ui.html . The backend is deployed in AWS.

Akın İlerle

In this project I helped with the UI of our trending topic history program. The UI part was written in plain HTML, CSS and JS. After that I deployed the UI and bot parts of the project to my VPS. Database and API servers are deployed to another server so I had to create a proxy to that server through my server using Nginx Proxy Pass. Finally, bot scripts has to run periodically so I added con jobs to my server that runs these scripts every day (one of the script is running weekly and others are running daily).

Halil Samed Çıldır

In the front end part I use javascript to manage the connection of backend and frontend. With my colleague Oncel Keles, we created several functions to adjust data type, get trending topics and populate those topics and load them afterwards.

Volkan YILMAZ

In the project, my task was to implement javascript to the frontend to establish communication between frontend and backend with Volkan Yılmaz. We implemented some functions to retrieve trending topics and tweets, and load them to the web site afterwards.

Öncel Keleş

14. Evaluation of Tools and Managing the Project

- **GitHub**: GitHub is a place to share code with friends, co-workers, classmates, and complete strangers. With the collaborative features of GitHub.com, desktop and mobile apps, the development platform helps individuals and teams to write better and faster code. Users can host and review code, manage projects, and build software alongside millions of other developers. GitHub brings teams together to work through problems, move ideas forward, and learn from each other along the way. On GitHub, lightweight code review tools are built into every pull request. Teams can create review processes that improve the quality of code and fit neatly into the workflow. Project managers and developers can coordinate, track, and update their work in one place, so projects stay transparent and on schedule. Users can fine tune the process as the team changes, update tools as technologies shift, and find new ways to work better. The possibility of collaborating with any free software project in the world (they are all hosted here) is very rewarding.You can talk to the Software creators, report mistakes and learn different solutions. It is very cool because it manages to unite the entire community of developers around the world in an easy, useful and totally altruistic way. There are even huge companies, like Facebook for example, that have opened their code because they are interested in receiving feedback from Github community in order to improve it. I also really like the graphical interface to control the pull requests system because it is very detailed. Github is the tool that best fits with Git. The ability to plan tasks and synchronize it with external tools such as Trello is

great. Github also allows you to check, in a very visual way, the exact changes that have been made in each line of code throughout the life of the project. This makes your life much easier.

- **Slack**: We use slack for communication. It is useful since it provides channels. Each team member can focus their tasks thanks to channel system. It also In this manner Slack is pretty good to use. Additionally, Slack integrates with tools like Trello, GitHub, Dropbox, Mailchimp, and dozens of others, so we can have a centralized event feed of our project right alongside our chat. This is tremendously useful for keeping context with our discussions.

- **Smartsheet**: We use Smartsheet to manage and maintain our Project plan. It is a software as a service application for collaboration and work management. It is used to assign tasks, track project progress, manage calendars, share documents and manage other work. It was extremely powerful and familiar to spreadsheets power users. Bot it requires lots of learning. Also it was not free.

- **Doodle**: Doodle is an online scheduling tool that can be used quickly and easily to find a date and time to meet with multiple people. It provides choosing days and times that you are considering for the event. We use it to schedule our meeting dates or determine other things such as team-name.

- **Proto.io**: Proto.io is a powerful web application for creating fully interactive, high-fidelity mobile application prototypes. It allows designers to create mobile app screens, apply screen transitions and define touch events for each design element on the screen. This creates a high quality prototype, which will work, and feel like a real app. Additional device events such as screen orientation are also supported. Prototypes can be viewed and experienced interactively within the browser but more importantly can be launched in the actual mobile device i.e. iPhone, iPad or equivalent, creating a breath-taking user experience. We used Proto.io to create our mockups for both desktop and mobile platforms. At the beginning it was a little bit hard to use but later it became easier. The best side was the Proto.io prototype editor offers an intuitive drag-and-drop user interface.

- **Lucidchart**: Lucidchart is a web-based diagramming tool that makes drawing diagrams fast and easy. Quickly draw flowcharts, org charts, wireframes, UML, mind maps and more. Work together with an unlimited number of others to create diagrams in real time, with changes synced instantaneously, great for team collaboration. We use Lucidchart to create our class diagrams and sequence diagrams.

- **Python**: Python is an interpreted high-level programming language for general-purpose programming. We used Python in our Twitter API application. It is so easy to code.

- **HTML**: Hypertext Markup Language (HTML) is the standard markup language for creating web pages and web applications. We use HTML in our Twitter API application for front-end.

- **CSS**: Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language like HTML. We used CSS in our Twitter API application for front-end.

- **JavaScript**: JavaScript is a high-level, interpreted programming language. It is a language which is also characterized as dynamic, weakly typed, prototype-based and multi-paradigm. . We used JS to make front-end of our Twitter API Application more dynamic.

15. CONCLUSION & ASSESSMENT

We learned a lot this semester about software project management and are looking forward to get into the coding phase of our “The Living History” project. The most important lesson to be taken from this semester was to realize how critical it is to be able to track everyone’s progress since a large software project is a collaborative effort of many people.