CMPE 451
Fall 2018

Milestone Report #1

Group #1

Ece Ata

Ahmet Yasin Alp

Akın İlerle

Deniz Etkar

Buse Ece Güldiken

Emre Bilgili

Utku Bozdoğan

Ziya Akyüz

Berke Can Gürer

Contents

0. Links of Retro

1. Executive Summary

2. List and Status of Deliverables

3. Evaluation of the Status of Deliverables

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

5. Requirements Analysis

6. Design

7. Project Plan

8. Issues will be inspected online and will not be in the delivered report.

9. Code

10. Evaluation of Tools and Managing the Project

11. CONCLUSION & ASSESSMENT

0. Links of Retro

Web: http://memoryretro.tk:3000

Android: (Not in Play Store)

1. Executive Summary

This is Milestone Report #1. We will be providing a description of our tasks, completed and incomplete and how they align with our project plan. We will give a description of our implementation and our individual work.

2. List and Status of Deliverables

- **Requirements:** Glossary and functionalities of the project.
    - **Status:** Completed.

- **Project Plan:** Plan of the project and how the task divided between group members for the entire 2 semesters of the course.
    - **Status:** Completed.

2.1. Backend

- **Backend&Database Design and Deployment:** Design of the backend and the database. Deployment of the backend on AWS servers.
    - **Status:** Completed

- **Login & Register:** Creation of User class, login endpoint and register endpoint
    - **Status:** Completed

- **Posting a Memory:** Creation of Memory class, and post a memory endpoint
    - **Status:** Completed

- **Memory Page:** Creation of an endpoint for returning details of a memory.
    - **Status:** Completed

2.2. Frontend

- **Login & Register:** Login and register pages were designed. Both login and register pages were connected to our server. Users can create an account and access the system.
    - **Status:** Completed.

- **Posting a Memory:** Add Memory page was designed. Member can post a memory.
    - **Status:** Basic part is completed. File upload and location selection will be implemented later.

- **Show Memory:** Page created. Members can see created memories.
    - **Status:** A simple show memory page completed.

2.3. Android

- **Login & Register:** Login and register pages are connected to our server. User can create an account and access the system.
    - **Status:** Completed.

- **Posting a Memory:** Member can post a memory.
    - **Status:** Basic part is completed. File upload and location selection will be implemented later.

3. Evaluation of the Status of Deliverables

The deliverables promised in our Project Plan for Milestone #1 are:

1. Requirements:

Almost finalized our requirements according to needs of our customers. We will change some details after we take some feedback from customers. We will remove the story feature inside the memory and add multiple location support and use only one time interval within a memory. So we need to plan and change the implementation of the memory part again before second milestone.

2. Project Plan:

   We are exactly at the right point of our project plan.

3. Deploying Project:
4. Features:
    a. Register and Login Pages:
    b. Posting a Memory Page:

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

Member	Team	Work
Ece Ata	Frontend	React and JavaScript learned. Frontend design is completed. Login, register and memory pages are implemented by using react. Loading our project on a domain.
Ahmet Yasin Alp	Android	Android memory posting page(design and implementation)
Akın İlerle	Backend	Developed the backend together with Buse. Used pair programming.
Deniz Etkar	Android	Android actionbar, memory feed and memory view implementation and design.
Buse Ece Güldiken	Backend	Developed the backend together with Akin. Used pair programming.
Emre Bilgili	Android	Android login and register UI implement, additional small functionalities.
Utku Bozdoğan	Frontend	Frontend-backend connections, login-register validation.
Ziya Akyüz	Frontend	Learned how to write React.
Berke Can Gürer	-	-

5. Requirements Analysis

Time: 23.10.2018

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.*
- *User: A person who anonymously viewing and searching memories, but can't do any editing.*
- *Member: A person who has an account and logged in, 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 specific part of text or rectangular area of a photo which is relevant to the content.*
- *Username: A specific handle that is unique for each registered user.*
- *Password: A specific sequence of characters that will allow members to access the system.*
- *Story: 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).*
- *Memory: Memory is the way users share a post in our app/website. It can contain multiple stories.*
- *Profile Page: A page that is unique to each user where members can see the memories shared by the owner of that profile page.*
- *Location: Location can be a region or can be a path.*

1.Functional Requirements

1.1. User Requirements

- *1.1.1. Users*
  - *1.1.1.1. Users should not be able to like or comment on current memories.*
  - *1.1.1.2. Users should not be able to do annotations on the memories.*
- *1.1.2. Login/Logout and Register*
  - *1.1.2.1. Register*
    - *1.1.2.1.1. Users shall be able to register using email verification system.*
    - *1.1.2.1.2. Registration form contains mandatory fields which are username, password and email address.*

        - *1.1.2.1.3. Users shall be able to use authenticate with their Facebook accounts while registering.*
        - *1.1.2.1.4. Users shall be able to reset their password by using a link that is sent to their verified email.*
    - *1.1.2.2. Login/Logout*
        - *1.1.2.2.1 Members shall be able to login with email address/username and password.*
        - *1.1.2.2.2 Members who already logged in shall be able to log out.*
        - *1.1.2.2.3 Members shall be able to change his/her password and email.*
- *1.1.3. Memory Features*
    - *1.1.3.1. Members should be able to create and share memories.*
    - *1.1.3.2. Memories consist of stories.*
    - *1.1.3.3. Members shall be able to add location information to stories.*
        - *1.1.3.3.1. Location shall be able to have map support.*
    - *1.1.3.4. Members shall be able to add time information to stories.*
        - *1.1.3.4.1. Time information shall be able to contain the date or the time interval of the story.*
        - *1.1.3.4.2. Time information shall be able to contain the date that the memory is added to the system.*
    - *1.1.3.5. Members shall be able to add photos to stories.*
    - *1.1.3.6. Members shall be able to add audio to stories.*
    - *1.1.3.7. Members shall be able to add video to stories.*
    - *1.1.3.8. Media are added to stories as attachments.*
    - *1.1.3.9. Members 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 times of the memory.*
    - *1.1.4.5. Users shall be able to search memories based on tags.*
- *1.1.5. Browse*
    - *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 a memory location in homepage.*
    - *1.1.5.3. Users can go to a specific location and see all the memories there on the map.*
- *1.1.6. Profile*
    - *1.1.6.1. Profiles are visible to members.*

        - *1.1.6.2. Members' own profile*
            - *1.1.6.2.1. Members shall be able to see a list of memories that he/she liked.*
            - *1.1.6.2.2. Members shall be able to see their posted memories in their profiles.*
            - *1.1.6.2.3. Members should be able to edit his/her memories in their profiles.*
            - *1.1.6.2.4. Members should be able to delete his/her memories in their profiles.*
            - *1.1.6.2.5. Members should be able to edit his/her bio in their profiles.*
            - *1.1.6.2.6. Members should be able to add birthday, gender information(unknown if it is not set), places that they lived to their profiles.*
        - *1.1.6.3. Other members' profiles*
            - *1.1.6.3.1. Members shall be able to see the memories other members posted in their profiles.*
- *1.1.7 Feedback*
    - *1.1.7.1. Members shall be able to leave a comment to a memory.*
    - *1.1.7.2. Members shall be able to like a memory.*
    - *1.1.7.2. Members shall be able to leave an annotation.*
- *1.1.8 Annotations*
    - *1.1.8.1. Members shall be able to add annotation to images in a memory by choosing a rectangular area of the image.*
    - *1.1.8.2. Members shall be able to add annotation to a specific part of a text in a story.*
    - *1.1.8.3. Members shall be able to add annotation to a specific part of a comment in a memory.*

1.2. System Requirements

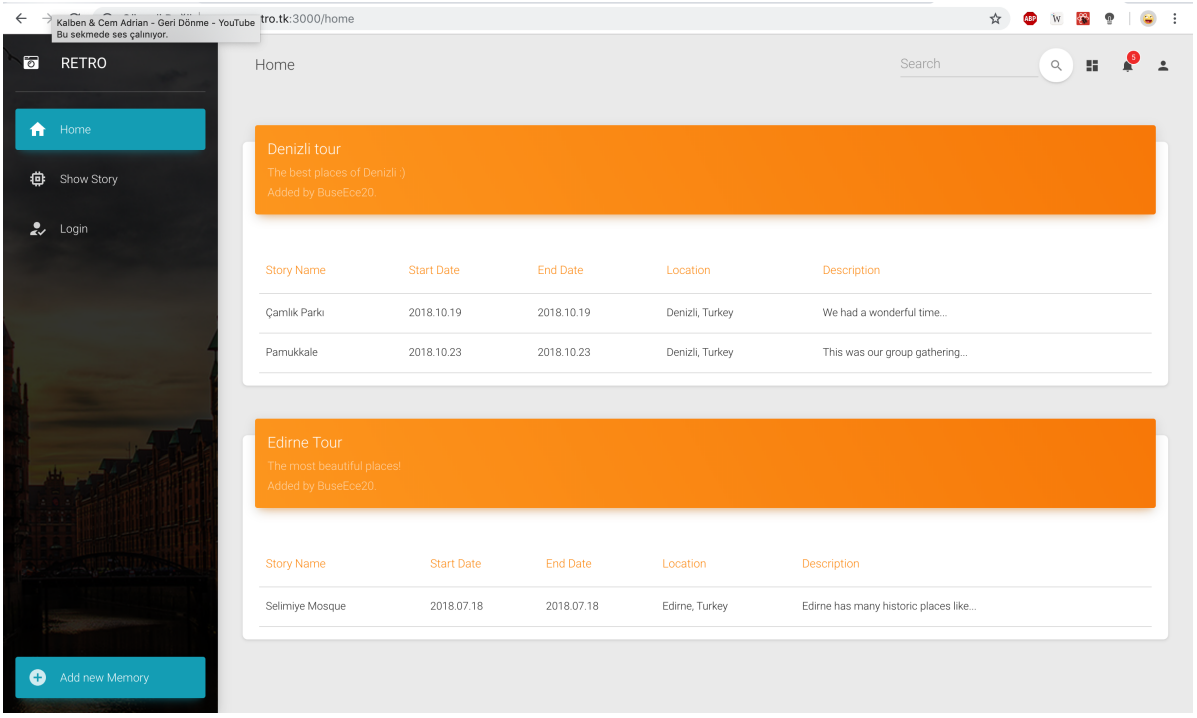
- *1.2.1. Recommendation*
    - *1.2.1.1. System shall recommend memories.*
        - *1.2.1.1.1. Recommendations shall be based on likes.*
        - *1.2.1.1.2. Recommendations shall be based on location.*
        - *1.2.1.1.3. Recommendations shall be based on tags.*
        - *1.2.1.1.4. Recommendations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

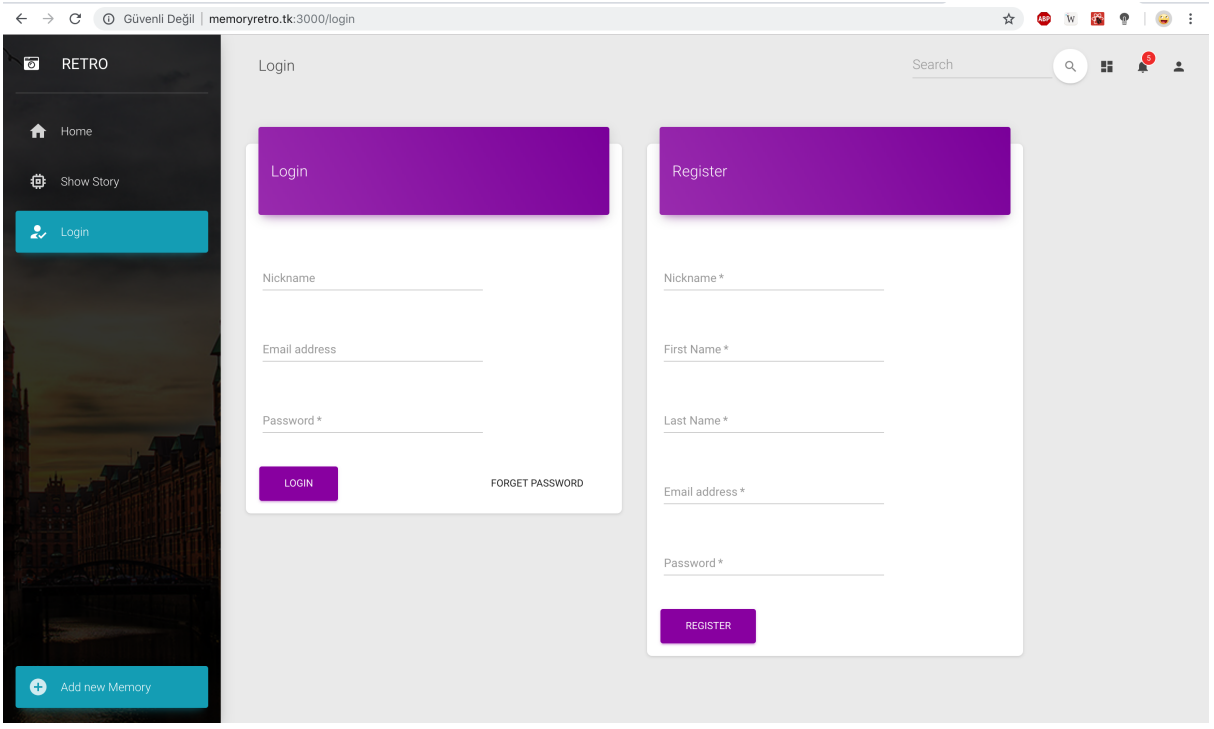
- *2.1. Security*
  - *2.1.1. Database system shall be protected.*
  - *2.1.2. Privacy of user data shall be protected.*
  - *2.1.3. Passwords shall be at least 8 characters long and contain at least one number and one letter.*
  - *2.1.4. Passwords shall be stored in hashed format.*
- *2.2 Availability*
  - *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 annotations must be compliant with The W3C Web Annotation Data Model.*
- *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.*

6. Design

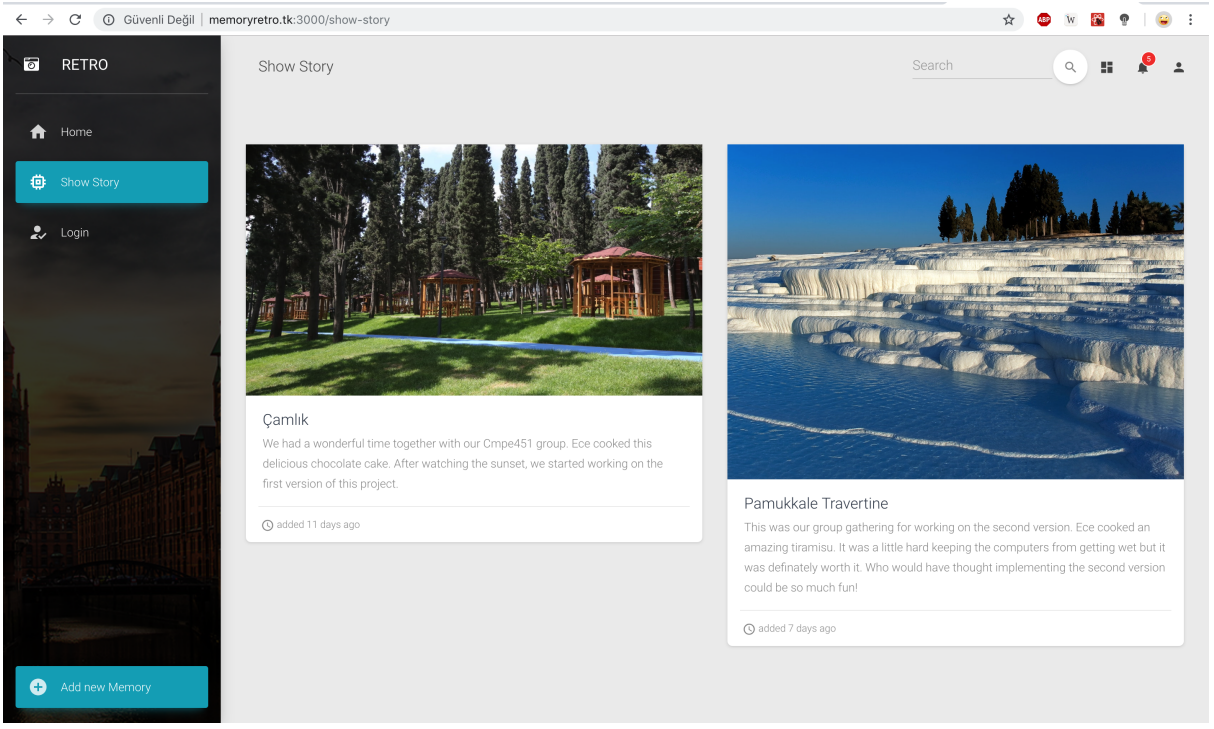
- Web:
  - Home page of our application:



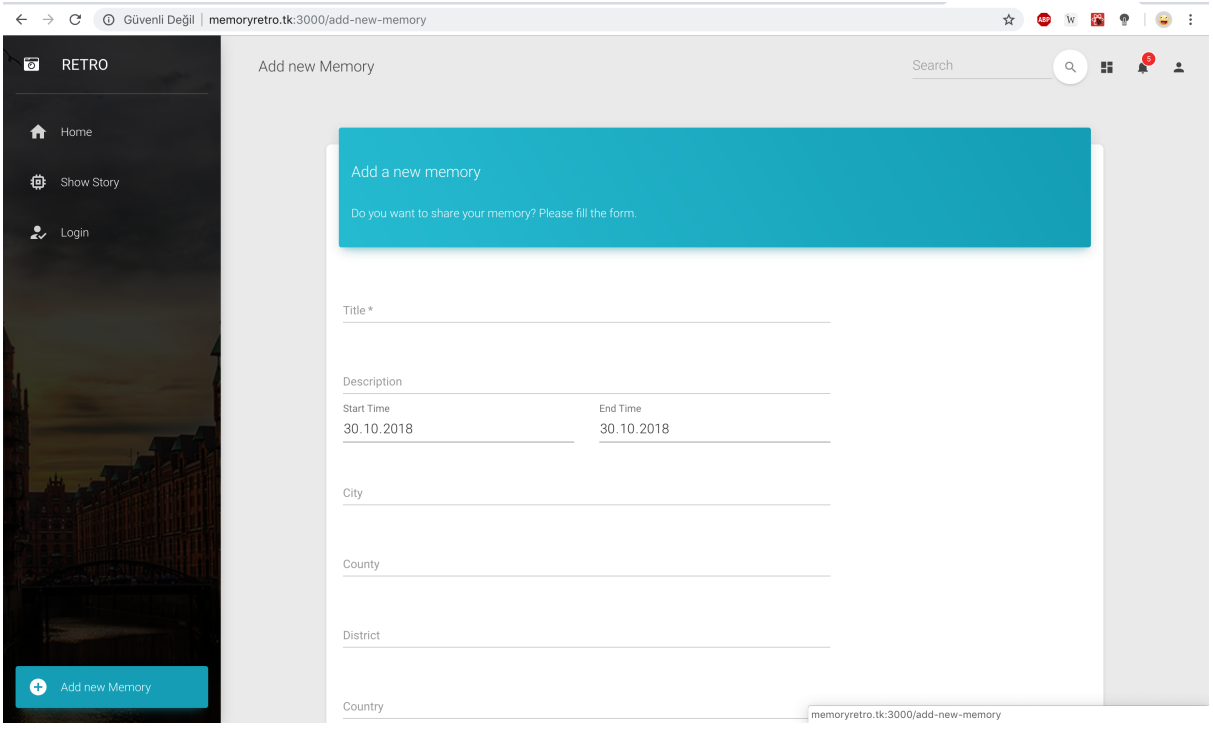
- Login and register page of our application:



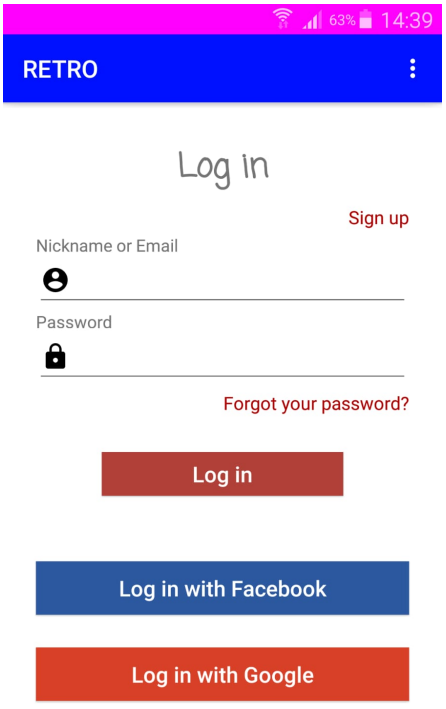
- Show memory page of our application:



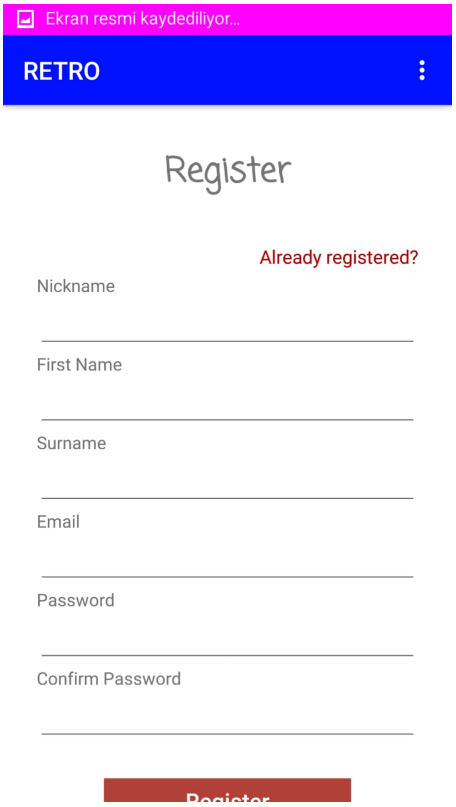
- Add memory page of our application:

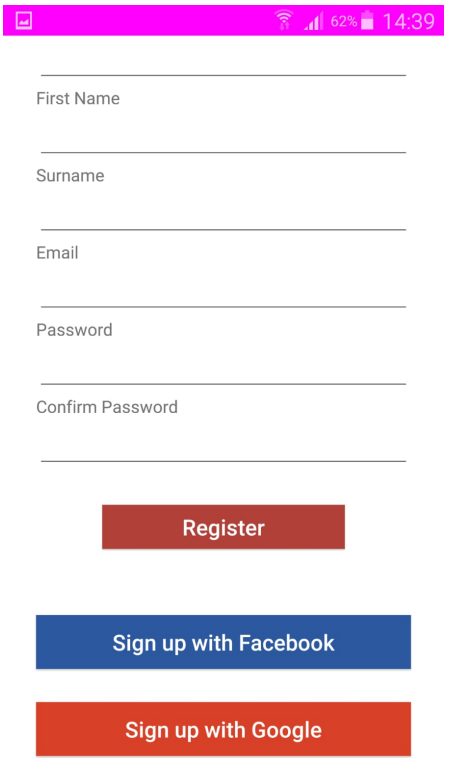


- Android:
  - Login Page:

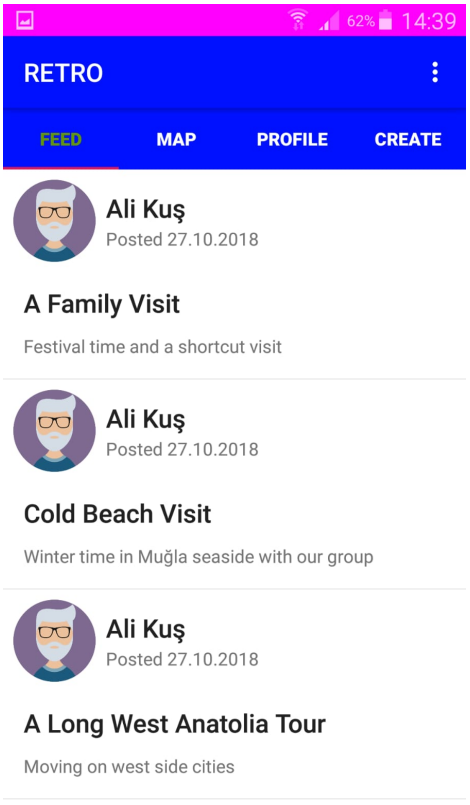


- Register Page:

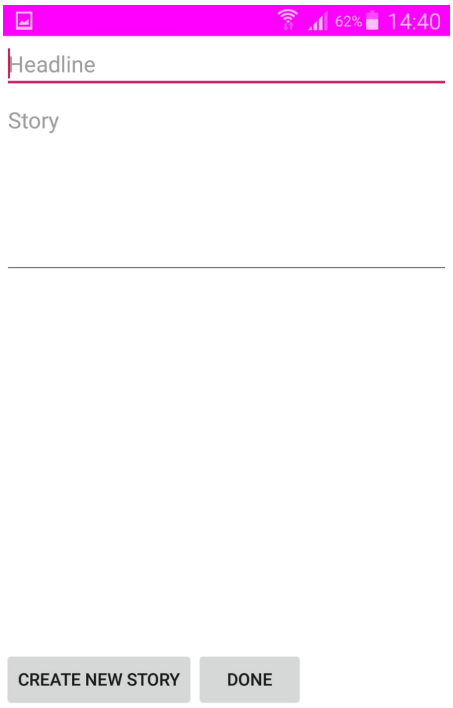


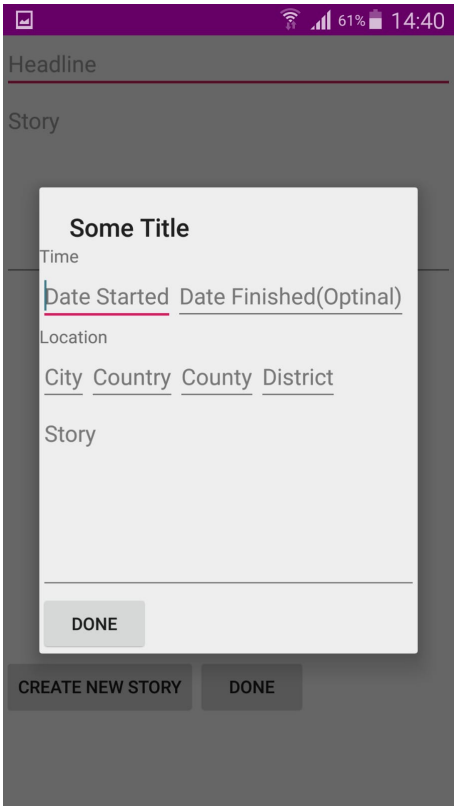


- Home Page:

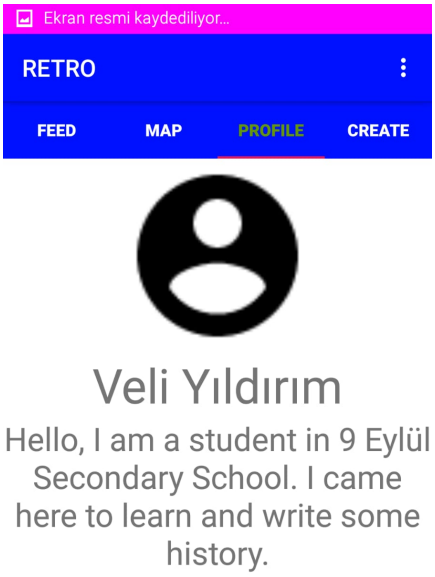


- Add Memory Page:





- Profile Page:



7. Project Plan

Time: 23.10.2018

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?

Deliverables:

- Project Plan

Status	Task Name	Duration	Start	Finish	Assigned To
Done	Task Assignment	5d	09/20/18	09/24/18	Akın İlerle, Ahmet Alp, Deniz Etkar, Ece Ata, Buse Güldiken, Emre Bilgili, Ziya Akyüz, Utku Bozdoğan
	Learning	76d	09/26/18	12/10/18	
	Android Learning	76d	09/26/18	12/10/18	Ahmet Alp, Deniz Etkar, Emre Bilgili
	Backend Learning	76d	09/26/18	12/10/18	Akın İlerle, Buse Güldiken
	Frontend Learning	76d	09/26/18	12/10/18	Ece Ata, Utku Bozdoğan, Ziya Akyüz
	Milestone #6	1d	12/10/18	12/10/18	
	Implementation	75d	10/03/18	12/16/18	
	Backend	75d	10/03/18	12/16/18	
Done	Backend and database design	5d	10/03/18	10/07/18	Akın İlerle, Buse Güldiken
Done	Register and login	5d	10/03/18	10/07/18	Akın İlerle, Buse Güldiken
Done	Connect backend to frontend and android	7d	10/08/18	10/14/18	Akın İlerle, Buse Güldiken
Done	Memory page	7d	10/15/18	10/21/18	Akın İlerle, Buse Güldiken
Done	Posting a memory	7d	10/22/18	10/28/18	Akın İlerle, Buse Güldiken
Done	Milestone #4	1d	10/30/18	10/30/18	
	Memory feed	7d	10/29/18	11/04/18	Akın İlerle, Buse Güldiken
	Search	7d	11/05/18	11/11/18	Akın İlerle, Buse Güldiken
	Profil page	7d	11/12/18	11/18/18	Akın İlerle, Buse Güldiken
	Editing profil page	7d	11/19/18	11/25/18	Akın İlerle, Buse Güldiken
	Editing user info	7d	11/26/18	12/02/18	Akın İlerle, Buse Güldiken
	Milestone #5	1d	12/04/18	12/04/18	
	Editing memories	7d	12/03/18	12/09/18	Akın İlerle, Buse Güldiken
	Commenting memories	7d	12/10/18	12/16/18	Akın İlerle, Buse Güldiken
	Annotation	48d	10/30/18	12/16/18	Akın İlerle, Buse Güldiken
	Finding a name and domain	75d	10/03/18	12/16/18	Akın İlerle, Buse Güldiken
	Setting up to domain	7d	12/10/18	12/16/18	Akın İlerle, Buse Güldiken
	Frontend	75d	10/03/18	12/16/18	
Done	Register and login	5d	10/03/18	10/07/18	Ece Ata, Utku Bozdoğan, Ziya Akyüz
Done	Connect frontend to backend	7d	10/08/18	10/14/18	Ece Ata, Utku Bozdoğan, Ziya Akyüz
Done	Memory page	7d	10/15/18	10/21/18	Ece Ata, Utku Bozdoğan, Ziya Akyüz
Done	Posting a memory	7d	10/22/18	10/28/18	Ece Ata, Utku Bozdoğan, Ziya Akyüz
Done	Milestone #4	1d	10/30/18	10/30/18	
	Memory feed	7d	10/29/18	11/04/18	Ece Ata, Utku Bozdoğan, Ziya Akyüz
	Search	7d	11/05/18	11/11/18	Ece Ata, Utku Bozdoğan, Ziya Akyüz
	Profil page	7d	11/12/18	11/18/18	Ece Ata, Utku Bozdoğan, Ziya Akyüz
	Editing profil page	7d	11/19/18	11/25/18	Ece Ata, Utku Bozdoğan, Ziya Akyüz
	Editing user info	7d	11/26/18	12/02/18	Ece Ata, Utku Bozdoğan, Ziya Akyüz
	Milestone #5	1d	12/04/18	12/04/18	
	Editing memories	7d	12/03/18	12/09/18	Ece Ata, Utku Bozdoğan, Ziya Akyüz
	Commenting memories	7d	12/10/18	12/16/18	Ece Ata, Utku Bozdoğan, Ziya Akyüz
	Annotation	48d	10/30/18	12/16/18	Ece Ata, Utku Bozdoğan, Ziya Akyüz
	Finding a name and domain	75d	10/03/18	12/16/18	Ece Ata, Utku Bozdoğan, Ziya Akyüz
	Android	75d	10/03/18	12/16/18	
Done	Register and login	5d	10/03/18	10/07/18	Ahmet Alp, Deniz Etkar, Emre Bilgili
Done	Connect android to backend	7d	10/08/18	10/14/18	Ahmet Alp, Deniz Etkar, Emre Bilgili
Done	Memory page	7d	10/15/18	10/21/18	Ahmet Alp, Deniz Etkar, Emre Bilgili
Done	Posting a memory	7d	10/22/18	10/28/18	Ahmet Alp, Deniz Etkar, Emre Bilgili
Done	Milestone #4	1d	10/30/18	10/30/18	
	Memory feed	7d	10/29/18	11/04/18	Ahmet Alp, Deniz Etkar, Emre Bilgili
	Search	7d	11/05/18	11/11/18	Ahmet Alp, Deniz Etkar, Emre Bilgili
	Profil page	7d	11/12/18	11/18/18	Ahmet Alp, Deniz Etkar, Emre Bilgili
	Editing profil page	7d	11/19/18	11/25/18	Ahmet Alp, Deniz Etkar, Emre Bilgili
	Editing user info	7d	11/26/18	12/02/18	Ahmet Alp, Deniz Etkar, Emre Bilgili
	Milestone #5	1d	12/04/18	12/04/18	
	Editing memories	7d	12/03/18	12/09/18	Ahmet Alp, Deniz Etkar, Emre Bilgili
	Commenting memories	7d	12/10/18	12/16/18	Ahmet Alp, Deniz Etkar, Emre Bilgili
	Finding a name and domain	75d	10/03/18	12/16/18	Ahmet Alp, Deniz Etkar, Emre Bilgili

Figure 2: A part of project plan

8. Issues will be inspected online and will not be in the delivered report

Our group understood some features differently from how it was asked to be implemented by our customers. We will edit this feature immediately.

Register and forget password features need email validation. We will add this feature before the second milestone.

9. Coding Work

We have three different development branches for android, backend and frontend. This allows us to only pull relevant commits.

9.1 Backend: On the backend branch we have decided to use feature branches and pull requests but since we mostly used pair programming implementing this part, we did not use feature branches and pull requests on some commits. Documentation of our APIs is in our swagger page.

9.2 Frontend: On the frontend, we used branches for different functionalities. We had our initial branch, in which we put the file structure and a very simple login/register page. On different branches we implemented memory and improved our design. We use pull requests and make sure we have our code reviewed by another member, who approves and merges the request.

9.3 Android: On the Android, we did not prefer use different branches for development. Android Studio usually overrides some code parts in the other source files when a developer creates or updates any page or code part. So, we divided the tasks and implemented them one by one to avoid from unresolvable merge problems.

10. 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.

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

11. CONCLUSION & ASSESSMENT

Since a large software project is a collaborative effort of many people, it was vital that we had information regarding the status of our own subgroup and of other subgroups. We were constantly in touch, always communicating about what is ready, where there is an error and so on. We used Github effectively, no member was allowed to push directly to the master branch and inside our groups, we ruled that our tasks be developed with branches. We did not allow merges without review and approval of another member for safety.

We also had to make time for developing this project, we all take multiple demanding courses apart from this one, we need to do other projects and we have midterms and other responsibilities. But for this project we had to attend meetings, do our part in implementation and do all of these in a timely manner because we had to show our progress to our customer each week. It is a challenging task, and we have managed it thus far. In our eyes, this is a gain.