CMPE 352

Fundamentals of Software Engineering

bounswe2018group7

MILESTONE REPORT

AUTHOR: Dilruba Reyyan Kılıç

CONTENTS

1. Abstract

1.1 Introduction

This Milestone Report describes the project, the deliverables and the work accomplished by bounswe2018group7 members which are called team se7en, throughout this semester.

The project is hiStory I A Living History which is a crowdsourcing application for documenting stories of people about historical issues.
It lets users to share or explore memories and interact between others. Users share posts about old times and habits which can be annotated by other users so it creates a cumulative memory repository.

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 and diagraming part. Next semester we can start the code implementation.

1.2 Summary of the Project Status

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

- Met with group members, get to know each other & oriented.
- Created Personal Wiki Pages and started to write Personal Logs.
- Determined a documantation structure for both Wiki pages and Meeting Notes.
- Modified the read.me part.
- Customized and determined issues structure & labels.
- Studied git as a version manangement system.
- Prepared the Communication Plan.
- Determined and documented our Requirements in a structured manner.
- Created a Project Plan to see the upcomings and ongoings well, and see “when” and “by whome” the work will be done.
- Created User Stories, Scenarios and Mockups (both Web and Android).
- Created Class, Sequence and Use Case Diagrams.
- Created Test Cases.
- Designed and implemented a Twitter Project using Twitter api.

1.2 Changes Planned

- All the requirements will be reviewed according to W3C.
- Along the way, some of the class diagrams and requirements may be changed while the coding part.
- The communication and project plan will be reviewed if the team will have new members.
- Communication plan will be updated according to the free times of all members in the new semester.

2. Deliverables

2.1 List and Status of the Deliverables

DELIVERABLE	STATUS
READ.ME	DONE
WIKI HOME PAGE	DONE
PERSONAL WIKI PAGES	DONE
MEETING NOTES	DONE
REQUIREMENTS	DONE
COMMUNICATION PLAN	DONE
PROJECT PLAN	DONE
USER STORIES & PERSONAS	DONE
MOCK-UPS (WEB & ANDROID)	DONE
USE CASE DIAGRAMS	DONE
SEQUENCE DIAGRAMS	DONE
ACTIVITY DIAGRAMS	DONE
TWITTER API	DONE

2.1 Evaluation of the Status of Deliverables

In brief, we believe that the current status of deliverables is quite well satisfactory.
For the deliverables of the project plan, almost everything went as expected on schedule. We discussed every deliverable in our regular meetings and gave feedbacks. We determined weekly deadlines and valued the feedbacks.

Requirements
To have a clearly understanding what we're doing we first studied the examples of similar project's requirements, after that we have focused on our projects and we determined our project's requirements in different stages. We first preapared a Glossary. Then we classified our requirements as Functional and Non-Functional requirements and that we explained our requirements apparently with subtitles.

Project Plan
We crated a project plan with milestones, works assigned to certain assignees and deadlines. It is quite appropriate and we are up to date.

Scenarios
In the process of creating scenarios the most important thing was creating different scenarios so they can contain differents client types and can simulate ethe contents of the project as much as possible.

Mock-ups
For the deliverables of Android and Web applications, we prepared mockups and this helped us to think about our applications in a more concrete way. We tried to maintain similar user experiences in both Android and Web applications.

Class - Sequence - Use Case Diagrams
We first examined the previous projects examples and then documented a draft to be developed and drawn in figures.
All are explained and shown clearly with the arrows, subclasses or if-else structures.

Test Cases
W have tried to test our project on possible stiations and functions. They are clearly explained with the corresponding inputs and expected outputs.

Twitter API
We have implemented an Twitter API for our project by using Django Frame work and it is up and running succesfully.

3. Coding Work Done By Each Team Member

Only Code implemented in this semester is the Twitter API.

Dilruba Reyyan Kılıç	Displaying all the tweets (UI)
Cemal Burak Aygün	Creating new tweets (API & UI)
Enes Koşar	Retrieving all the tweets (API)
Ramazan Arslan	Post Including Only Urls(API & UI)
Neval Tüllük	
Ferhat Melih Dal	Searching for tweets (UI)
Serdar Ada	Searching for tweets (API)
Faik Emre Derin	

4. Requirements

These are the current requirements of our project.

Glossary (in alphabetical order)

- Account: An entity which uniquely represents a user in the system. It consists of a username, a real name, a profile picture, an email address and a password.
- Annotation: A note, comment or explanation attached to a specific part of a text or an image.
- Comment Section: A distinct area in a memory post that contains comments of people about the memory stated in the memory post.
- Homepage: The main web page of the system.
- Main Map: A world map which stores indicators of memory posts over related memory locations.
- Memory: Something that is remembered from the past.
- Memory Item: A material that is related to a memory.
- Memory Location: The location where a memory took place or to which a memory is related.
- Memory Media: A collection of images, videos and audios that are related to a memory.
- Memory Story: A text which contains some information and/or experience about a memory.
- Memory Time: The time during which a memory took place or to which a memory is related.
- Memory Title: A brief description about a memory.
- Memory Map: A map which displays the memory location in a memory post.
- Memory Post: A post that contains the memory items of a specific memory, a memory map, a comment section and tags.
- Profile: A user-specific page which contains account information of a user as well as the memory posts the user has created or liked.

- Recommendation: Displaying some memory posts, to a user on homepage, which the user might find interesting.
- Registration: Creating a unique account on the system for sign-in.
- Search Bar: A form which allows users to search for memory posts or users in the system.
- System: The application on which people can create, view, search, comment, like, dislike and annotate memory posts.
- Memory Tag: A keyword that categorizes a memory post.
- Username: A sequence of alphanumeric characters which uniquely identifies a user of the system.
- Visitor: A human being or a computer bot/program that visits the system (is online on the system).
- Guest: A visitor who is not signed-in to the system.
- User: A visitor who is signed-in to the system.
- Admin: A user who has the full control of the system.

---

1. Functional Requirements

1.1 User Requirements

1.1.1 Sign-In

1.1.1.1 Users shall be able to sign in using their username and password.
1.1.1.2 Users shall be able to sign in using their email address and password.
1.1.1.3 Users should be able to sign in using their Facebook account.

1.1.2 Sign-Out

1.1.2.1 Users shall be able to sign out from the system.

1.1.3 Profile

1.1.3.1 Users shall be able to view profiles of other users.
1.1.3.1.1 Users shall be able to see real name of the other users on profile.
1.1.3.1.2 Users shall be able to see username of the other users on profile.
1.1.3.1.3 Users shall be able to see profile picture of the other users on profile.
1.1.3.1.4 Users shall be able to see the memory posts created by the other users on profile.
1.1.3.2 Users shall be able to update their account information on their profile.
1.1.3.2.1 Users shall be able to update their username on their profile.
1.1.3.2.2 Users shall be able to update their real name on their profile.
1.1.3.2.3 Users shall be able to update their profile picture on their profile.
1.1.3.2.4 Users shall be able to update their email address on their profile.
1.1.3.2.5 Users shall be able to update their password on their profile.
1.1.3.3 Users shall be able to delete their profile/account.

1.1.4 Memory Post

1.1.4.1 Users shall be able to view memory posts.
1.1.4.2 Users shall be able to create memory posts.
1.1.4.2.1 Users shall be able to add a memory title to the memory post on creation.
1.1.4.2.2 Users shall be able to add a memory location to the memory post on creation.
1.1.4.2.3 Users shall be able to add a memory time to the memory post on creation.
1.1.4.2.4 Users shall be able to add a memory story to the memory post on creation.
1.1.4.2.5 Users shall be able to add memory media items to the memory post on creation.

1.1.4.2.6 Users shall be able to add tags to the memory post on creation.

1.1.4.3 Users shall be able to update their memory posts.

1.1.4.3.1 Users shall be able to update memory title of their memory posts.

1.1.4.3.2 Users shall be able to update memory location of their memory posts.

1.1.4.3.3 Users shall be able to update memory time of their memory posts.

1.1.4.3.4 Users shall be able to update memory story of their memory posts.

1.1.4.3.5 Users shall be able to update memory media of their memory posts.

1.1.4.3.6 Users shall be able to update tags of their memory posts.

1.1.4 Users shall be able to delete their memory posts.

1.1.4.1 Users shall be able to like memory posts.

1.1.4.2 Users shall be able to dislike memory posts.

1.1.5 Comment

1.1.5.1 Users shall be able to add comments to the comment section of a memory post.

1.1.5.2 Users shall be able to delete their comments.

1.1.5.3 Users shall be able to delete any comments in their memory posts.

1.1.6 Annotation

1.1.6.1 Users shall be able to view annotations.

1.1.6.2 Users shall be able to create annotations.

1.1.6.2.1 Users shall be able to annotate memory title.

1.1.6.2.2 Users shall be able to annotate memory story.

1.1.6.2.3 Users shall be able to annotate memory media (images only).

1.1.6.2.4 Users shall be able to annotate comments.

1.1.6.3 Users shall be able to update their annotations.

1.1.6.4 Users shall be able to delete their annotations.

1.1.6.5 Users shall be able to delete any annotations in their memory posts.

1.1.7 Search

1.1.7.1 Users shall be able to search for memory posts using search bar.

1.1.7.1.1 Users shall be able to search for memory posts with keywords.

1.1.7.1.2 Users shall be able to search for memory posts with tags.

1.1.7.2 Users shall be able to search for memory posts on main map.

1.1.7.3 Users shall be able to search for other users with keywords.

1.1.8 Admin

1.1.8.1 Admins shall have all the abilities users have.

1.1.8.2 Admins shall have following privileges:

1.1.8.2.1 Admins shall be able to ban any accounts in the system.

1.1.8.2.2 Admins shall be able to delete any accounts (profiles) in the system.

1.1.8.2.3 Admins shall be able to delete any memory posts in the system.

1.1.8.2.4 Admins shall be able to delete any annotations in the system.

1.1.8.2.5 Admins shall be able to delete any comments in the system.

1.2 Guest Requirements

1.2.1 Sign-Up

1.2.1.1 Guests shall be able to sign up by providing a username, a real name (optional), a profile picture (optional), an email address and a password.

1.2.1.2 Guests should be able to sign up via their Facebook account.

1.2.2 Memory Post

1.2.2.1 Guests shall be able to view all parts of a memory post.

1.2.3 Comment

1.2.3.1 Guests shall be able to view comments.

1.2.4 Annotation

1.2.4.1 Guests shall be able to view annotations.

1.3 System Requirements

1.3.1 Sign-Up

1.3.1.1 Guests shall be required to verify their email address on sign-up.

1.3.2 Sign-In

1.3.2.1 Users should be redirected to homepage after sign-in.

1.3.2.2 Users shall be blocked from sign-in for 10 minutes after 5 consecutive unsuccessful sign-in attempts.

1.3.2.2.1 An informative email should be sent to the users after the fifth unsuccessful sign-in attempt.

1.3.3 Sign-Out

1.3.3.1 Users shall be redirected to homepage after sign-out.

1.3.4 Account

1.3.4.1 Accounts shall have a username.

1.3.4.2 Accounts shall have a real name (optional).

1.3.4.3 Accounts shall have a profile picture (optional).

1.3.4.4 Accounts shall have an email address.

1.3.4.5 Accounts shall have a password.

1.3.5 Username

1.3.5.1 Usernames shall contain letters and digits only.

1.3.5.2 Usernames shall start with a letter.

1.3.5.3 Usernames shall contain at least 2 characters.

1.3.5.4 Usernames shall contain at most 32 characters.

1.3.6 Password

1.3.6.1 Passwords shall contain at least 8 characters.

1.3.6.2 Passwords shall contain at most 24 characters.

1.3.6.3 The system shall provide users with a password recovery mechanism.

1.3.7 Profile

1.3.7.1 The system shall provide users with a profile.

1.3.7.2 Profiles shall store users' account information.

1.3.7.3 Profiles shall store users' memory posts.

1.3.7.4 Profiles shall store the memory posts user has liked.

1.3.8 Homepage

1.3.8.1 Homepage shall contain main map.

1.3.8.2 Homepage shall contain most recently created memory posts.

1.3.8.2.1 The system shall allow users to see all the most recent memory posts.

1.3.8.2.2 The system shall allow guests to see the most recent 10 memory posts only.

1.3.8.3 Homepage shall contain recommended memory posts (for users only).

1.3.9 Memory Post

1.3.9.1 Memory posts shall have a created time attribute.

1.3.9.2 Memory posts shall have a created by attribute.

1.3.9.3 Memory posts shall have a like count attribute.

1.3.9.4 Memory posts shall have a dislike count attribute.

1.3.9.5 Memory posts shall have a comment count attribute.

1.3.9.6 Memory posts shall have a memory map.

1.3.9.7 Memory posts shall be required to have a memory title.

1.3.10 Memory Item

1.3.10.1 Memory Title

1.3.10.1.1 Memory Titles shall contain plain text only.

1.3.10.2 Memory Location

1.3.10.2.1 Memory Locations should be selected from a map.

1.3.10.2.2 Memory Locations should support points.

1.3.10.2.3 Memory Locations should support regions.

1.3.10.2.4 Memory Locations should support paths.

1.3.10.2.5 Memory Locations should support geolocation names.

1.3.10.3 Memory Time

1.3.10.3.1 Memory Times should be selected from a calendar.

1.3.10.3.2 Memory Times should support basic temporal units.

1.3.10.3.2.1 Basic temporal units should support month format.

1.3.10.3.2.2 Basic temporal units should support year format.

1.3.10.3.2.3 Basic temporal units should support decade format.

1.3.10.3.2.4 Basic temporal units should support day and month format.

1.3.10.3.2.5 Basic temporal units should support month and year format.

1.3.10.3.2.6 Basic temporal units should support day, month and year format.

1.3.10.3.3 Memory Times should support time intervals between basic time units.

1.3.10.4 Memory Story

1.3.10.4.1 Memory Stories shall contain plain text only.

1.3.10.5 Memory Media

1.3.10.5.1 Image files shall be supported.

1.3.10.5.2 Video files shall be supported.

1.3.10.5.3 Audio files shall be supported.

1.3.11 Memory Map

1.3.11.1 Memory Maps shall not be editable by anyone.

1.3.11.2 The system shall set the memory map of a memory post to the memory location of the memory post.

1.3.12 Main Map

1.3.12.1 Main Map shall not be editable by anyone.

1.3.12.2 Main Map shall contain indicator of memory posts in the system over related memory locations.

1.3.13 Annotation

1.3.13.1 Annotations shall be stored in the system.

1.3.13.2 Annotations should be tested for validity.

1.3.13.3 Annotations shall contain plain text.

1.3.13.4 Annotations should contain links.

1.3.13.5 Annotations should contain images.

1.3.13.6 Annotations should contain videos.

1.3.13.7 Annotations should contain audios.

1.3.14 Recommendation

1.3.14.1 The system shall recommend memory posts to users on homepage based on users' likes, dislikes, memory posts and tags.

1.3.14.2 The system should recommend memory posts to users on homepage based on users' current location.

1.3.15 Comment

1.3.15.1 Comments shall contain plain text.

1.3.15.2 Comments should contain links.

1.3.15.3 Comments shall not be editable.

1.3.16 Tag

1.3.16.1 Tags shall contain alphanumeric characters only.

1.3.17 Search

1.3.17.1 When users search for memory posts with keywords, the system shall retrieve memory posts which contain specified keywords in memory title, memory story or memory location.

1.3.17.2 When users search for memory posts with tags, the system shall retrieve memory posts which have specified tags.

1.3.17.3 When users search for other users with keywords, the system shall retrieve accounts (profiles) which contain specified keywords in username or real name.

2. Non-functional Requirements

2.1 Accessibility

2.1.1 The language of the system shall be English.

2.2 Availability

2.2.1 The system shall have a web application.

2.2.1.1 3 most recent main versions of Mozilla Firefox shall be supported.

2.2.1.2 3 most recent main versions of Google Chrome shall be supported.

2.2.1.3 3 most recent main versions of Opera shall be supported.

2.2.2 The system shall have an Android application.

2.2.2.1 3 most recent main versions of Android shall be supported.

2.3 Security

2.3.1 The system shall use SSL protocol.

2.3.2 Users' passwords shall be kept encrypted in the database.

2.3.3 Users' email addresses should be hidden to other users on their profile.

2.4 Performance

2.4.1 The system should respond to a request within at most 3 seconds.

2.4.2 The system should be able to support at most 1000 concurrent visitors.

2.5 Annotation

2.5.1 The W3C Web Annotation Data Model shall be used.

2.5.2 JSON-LD should be used for annotation representation as described in the W3C documentation.

2.6 Map

2.6.1 Google Maps services shall be used for Main Map.

2.6.2 Google Maps services shall be used for Memory Map.

5. User Stories

Admin

Persona

David Anderson (from UK. Age :23, Single.)

He is a master's student in Boğaziçi University Computer Engineering department, lives in Istanbul. He loves walking at Bebek coast around midnight.

He speaks Turkish as secondary language fluently and he is pre-intermediate level Spanish and French speaker.

He is a team player who can be a part of big software projects and he is interested in back-end development.
He wants to build an IT company in a few years, so working with "hiStory" he aims to gain some experience and money.
He connects to the platform everyday between 5PM and 7PM.

User Stories
- As an admin, David can start the system maintenance so that there will be minimal data loss on a server crash.
- As an admin, David can delete some memory posts so that he can prevent spamming.
- As an admin, David can delete some comments so that he can prevent irrelevant comments.
- As an admin, David can delete some users (profiles) so that he can ban users who use offensive language.

Acceptance Criteria
- System shall provide deleting memory items.
- System shall provide deleting comments.
- System shall provide deleting users.
- Registered User

Member

Persona
Muzaffer YILMAZ is a retired man born in Sivas, but currently lives in İstanbul. Since he is not going to Sivas very often, he likes to tag every detail that reminds him his hometown, Sivas. Like all the old people, he likes to live in his memories and enjoys thinking about the past.
He has two kids and three grandkids. He is a great dog lover, knows lots of things about raising dogs and taking care of them. He has lots of photos of the old times of İstanbul which belongs to his teenagehood and older photos of Sivas from his childhood.
Since he has a lot of free time, he likes to spend a lot of time online, or outside, taking photos and exploring new places and areas.

User Stories
- As a user, he wants to share his old photos and their stories in a memory post and tag them into the map in his profile, so that he can share the world he grew up once with other people, especially younger like his grandkids and their friends.
- As a user, he wants to search for the past tags of the places and events he has been to when he was younger, with names and tags. So, he can enjoy re-living his memories through hiStory app.
- As a user, he wants to see his recommended places on the map` and make a daily exploring route so that he can see hidden new places and the things which might be better for his interests.

Acceptance Criteria
- He wants no unrelated posts in the application.

- He wants his profile to be secure.
- He wants his personal information not to be used by a third party without his consent.
- He wants to post new items.
- He wants to comment, like, annotate new items.

Guest

Persona
Natalya Siminova from Russia. Age :27, Single.
She is a Yoga trainer, lives in St.Petersburg. She is interested in history and historical places of other countries, especially with the gothic architecture.
She speaks English and a little bit French. She is an amateur tennis player. She owns a 2 years old Pug, Gulf. She is cooperative, likes to be a part of the share of information. She likes to help abandoned animals. She also works voluntarily in PETA. She searches for different historical places in the platform everyday.

User Stories
- As a guest, she wants to have an idea about memory posts posted near her, so that she can see the main map.
- As a guest, she wants to have an idea about what the memory posts look like, so that she can the selected memory posts in the main page.
- As a guest, she can sign up and become a registered user so that she will be able to share and like posts, make comments and create her own profile.
- Acceptance Criteria
- She wants to see the popular memory posts.
- She wants to access some of the content without registering.

6. Scenarios & Mock-Ups

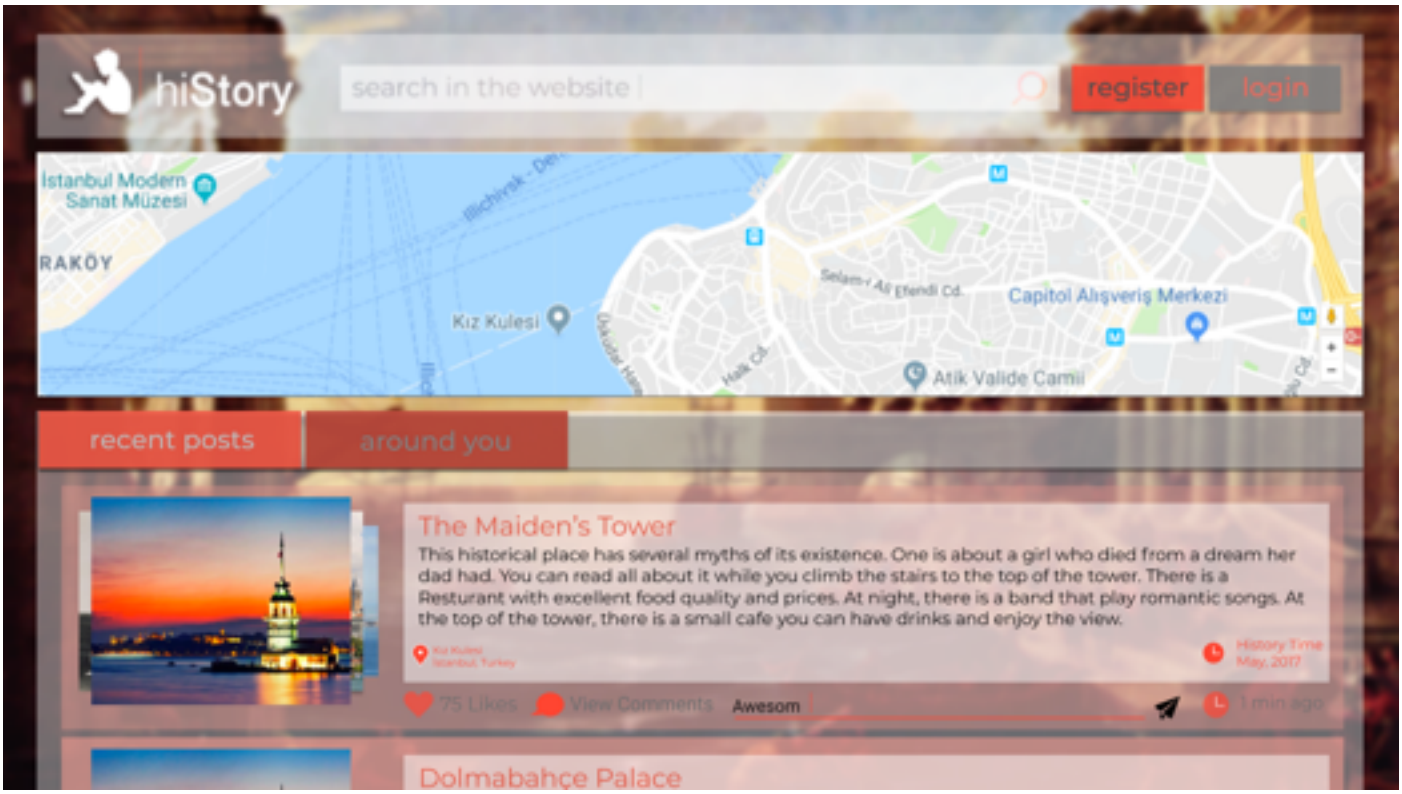
5.1 Web Scenarios

Guest Scenario

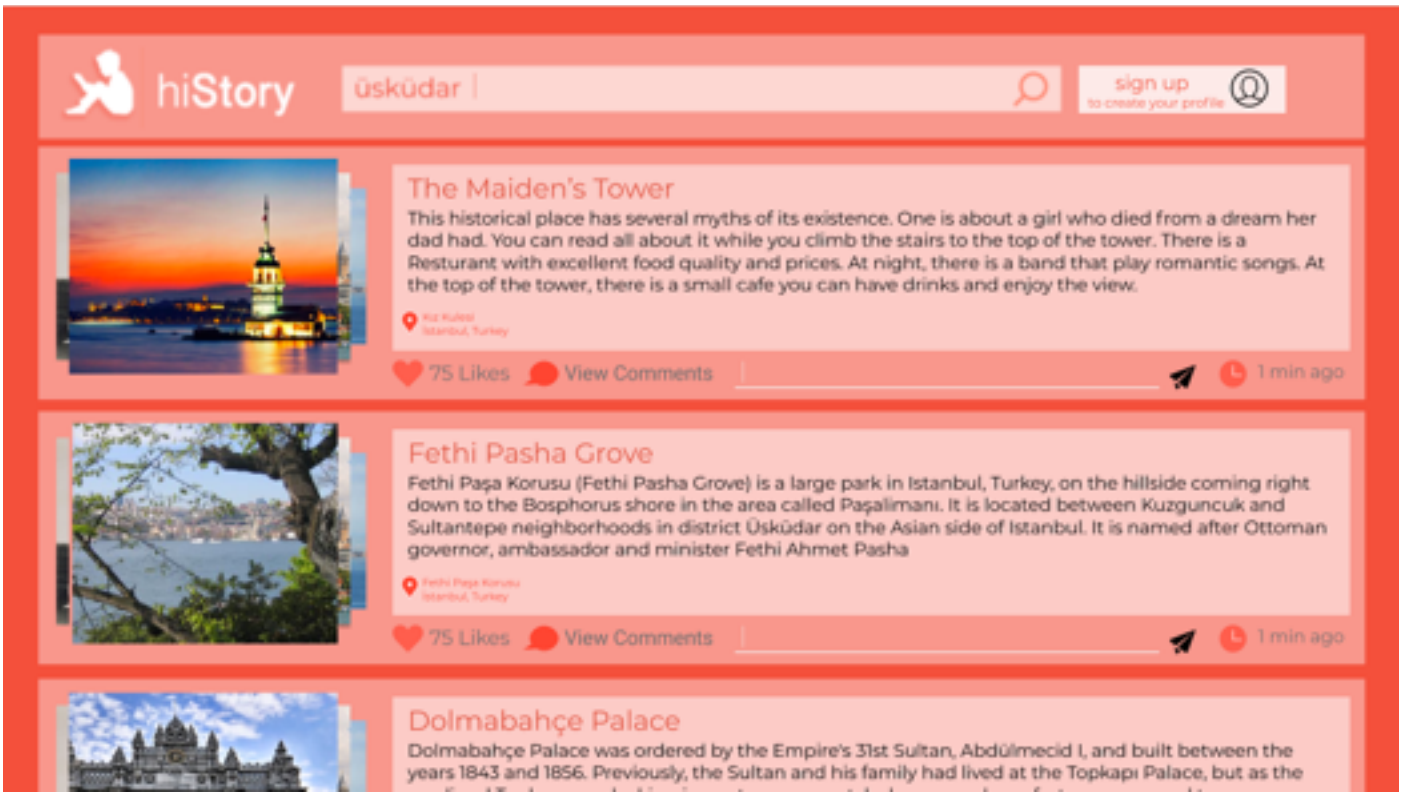
Meet Natalya Siminova. She is a yoga trainer from Russia. She is best friends with Zeynep, who is one of the Natalya's trainees. One day, after the yoga training hour, Natalya and Zeynep have a chat at a cafe. Zeynep mentions about her hometown, Üsküdar. She tells Natalya that Üsküdar was so nice once but nowadays there are construction sites everywhere and the neighbourhood is too crowded. Natalya becomes curious about the hometown of her best friend.

Then, she opens her favourite browser on her laptop and visits hiStory website hoping to find some memories and stories of people living in Üsküdar.

She is redirected to the homepage where she can see the most recently published memory posts and some recommended memory posts related to the her current location.



To find memory posts about Üsküdar, she types "uskudar" in search bar while on homepage and clicks "Search" button. In Search Results page, she sees descriptions of all the memory posts in the system related to Üsküdar. She finds "The Maiden's Tower" interesting and clicks the arrow icon next to the description of that memory post to see some details.



In Memory Post page (of "The Maiden's Tower"), she sees a story about The Maiden's Tower. She also sees a comment section where people share their opinions and knowledge about The Maiden's Tower.

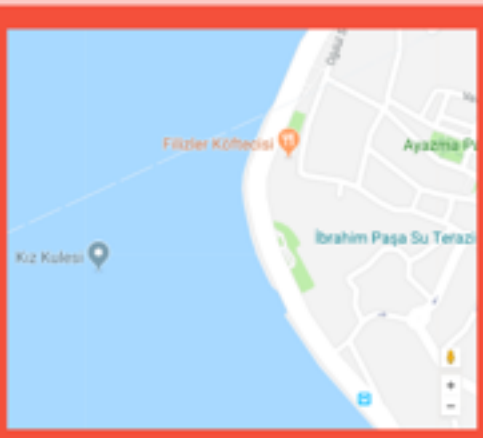
hiStory

üsküdar

The Maiden's Tower (Turkish: Kız Kulesi), also known as Leander's Tower (Tower of Leandros) since the medieval Byzantine period, is a tower lying on a small islet located at the southern entrance of the Bosphorus strait 200 m (220 yd) from the coast of Üsküdar in Istanbul, Turkey.

History Time
11th century

Kız Kulesi
Üsküdar,İstanbul



After the naval victory at Cyzicus, the ancient Athenian general Alcibiades possibly built a custom station for ships coming from the Black Sea on a small rock in front of Chrysopolis (today's Üsküdar). In 1110 Byzantine Emperor Alexius Comnenus built a wooden tower protected by a stone wall. From the tower an iron chain stretched across to another tower erected on the European shore, at the quarter of Mangana in Constantinople. The islet was then connected to the Asiatic shore through a defense wall, whose underwater remains are still visible. During the Ottoman conquest of Constantinople (Istanbul) in 1453, the tower held a Byzantine garrison commanded by the Venetian Gabriele Trevisano. Subsequently, the structure was used as a watchtower by the Ottoman Turks during the reign of Sultan Mehmed the Conqueror.

Ebru_86 | 15 comments | 75 Likes | **Post Time** 5 May 2016

maidens tower | tower | üsküdar

you need to sing in to comment this memory post

Comments

Hulusi_Amca — 7 May 2016 11:23 AM

People used to swim there, now there is a restaurant!.. :'(

Sofia.london — 8 May 2016 9:53 PM

It's an old tower in the sea with Romantic and Drama Stories that you can see the paintings of them. In the tower you can find a restaurant and lovely cafe

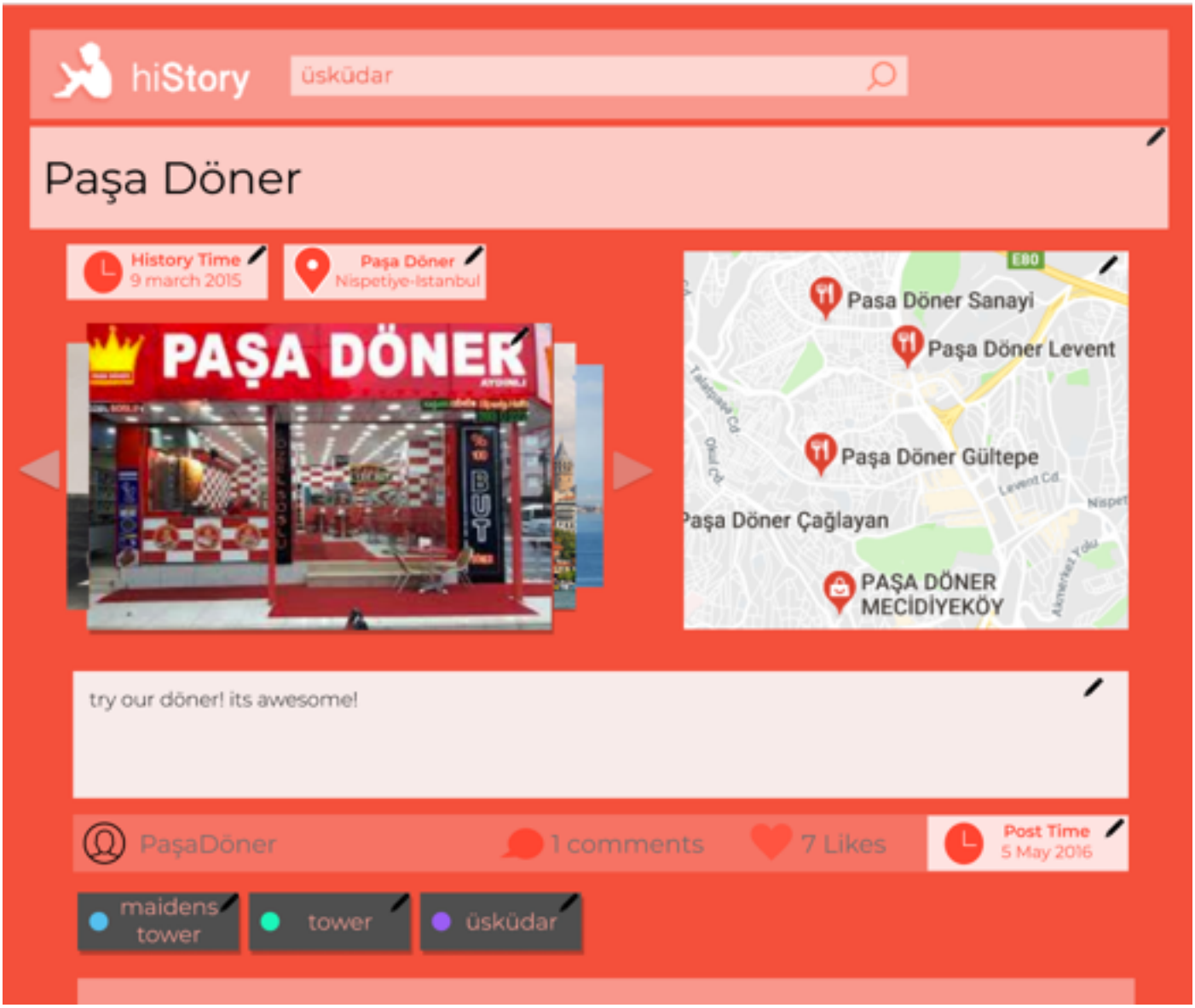
Gilberto.Md — 17 July 2016 2:37 PM

Enjoyed dinner here with friends. The setting and the bar at the top are spectacular and afford 360 views along the Bosphorus. The water taxi and service was well planned and efficient but the food was a bit like the catering at a ballroom as in mass produced and lacked any feeling...[show more]

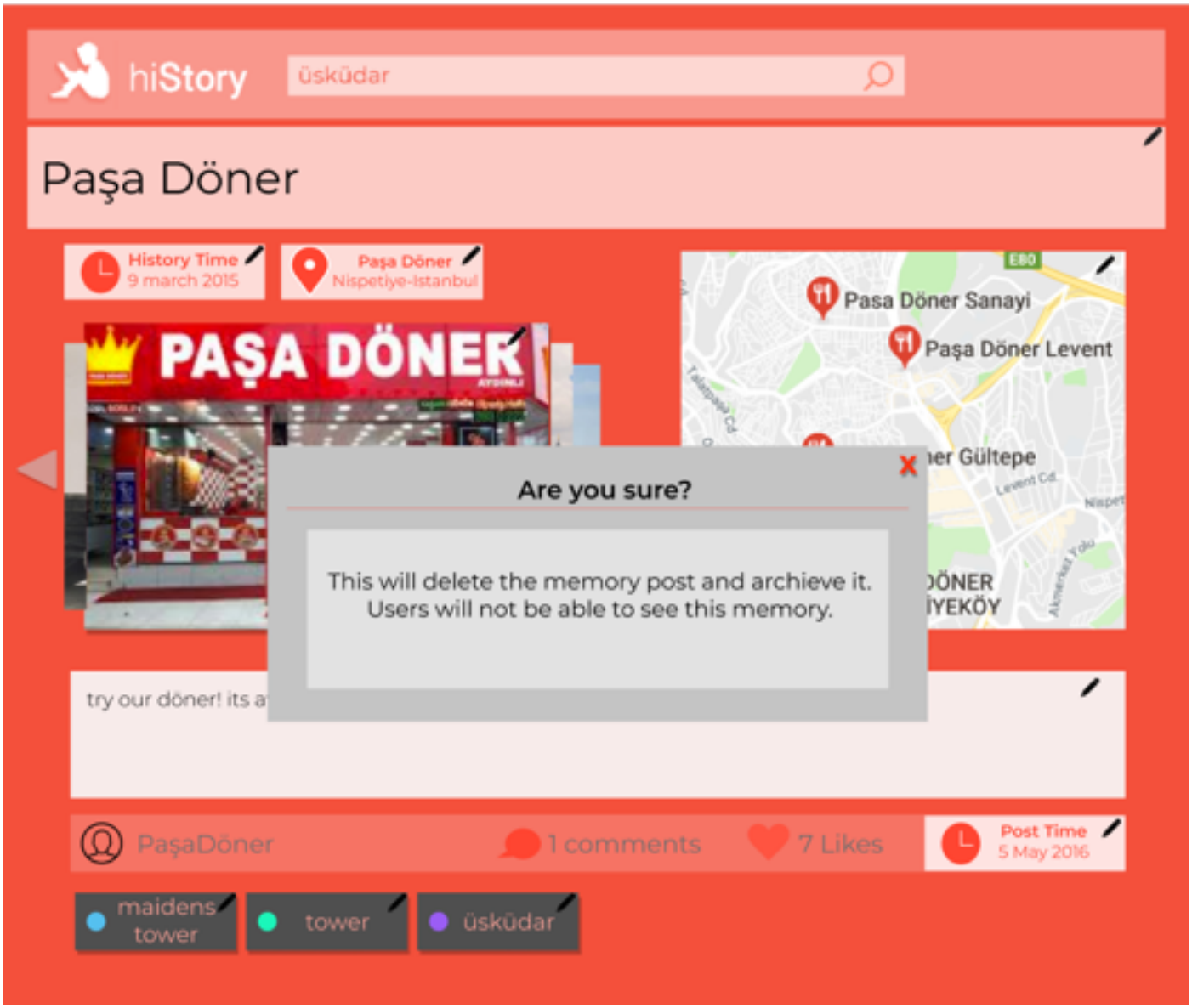
Admin Scenario

This is David Anderson. He has a passion to keep history living in the web so that anyone can access and see how people lived in the past. This is why he is part of the hiStory team and an admin of the system. He likes to check out the posts and learn from them just as anybody.

One of the concerns he has is that people can post items to the highly used hiStory website out of commercial interest or that people can include any other content than what is approved by the hiStory project manifesto. Thats why he keeps looking at posts and looks to see if there are any posts that should be deleted. While trying to see what shop was there before "Paşa Döner", he comes across a post that he thinks carries a commercial interest. He deletes the content. The content is sent to archive. And is no longer open to non-admin users.



He is asked if he is sure about his decision.

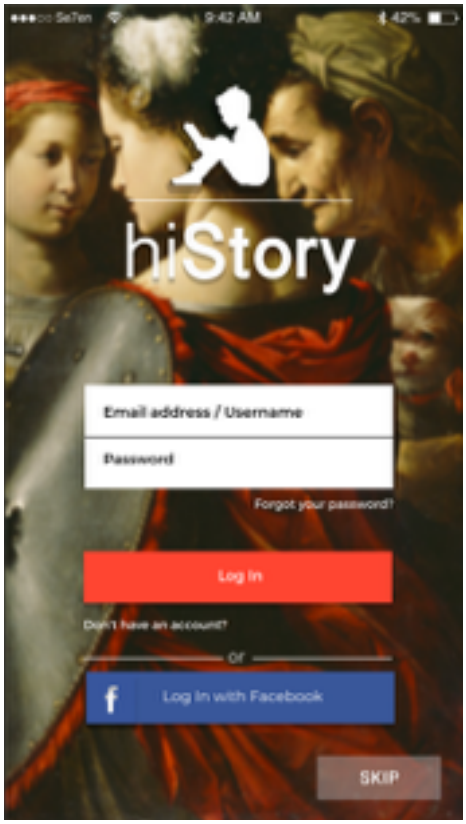


On one of his other visits to the website, he comes across this very new post about how old the Hisarüstü Barber Shop is. When David scrolls through the comments, he realizes that one of the comments is very aggressive and includes slang word.

He decides to delete the comment.

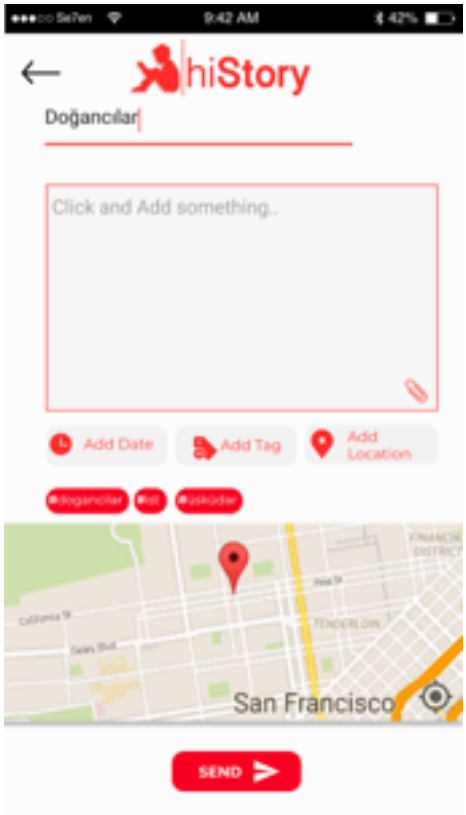
5.1 Android Scenarios

User Scenario



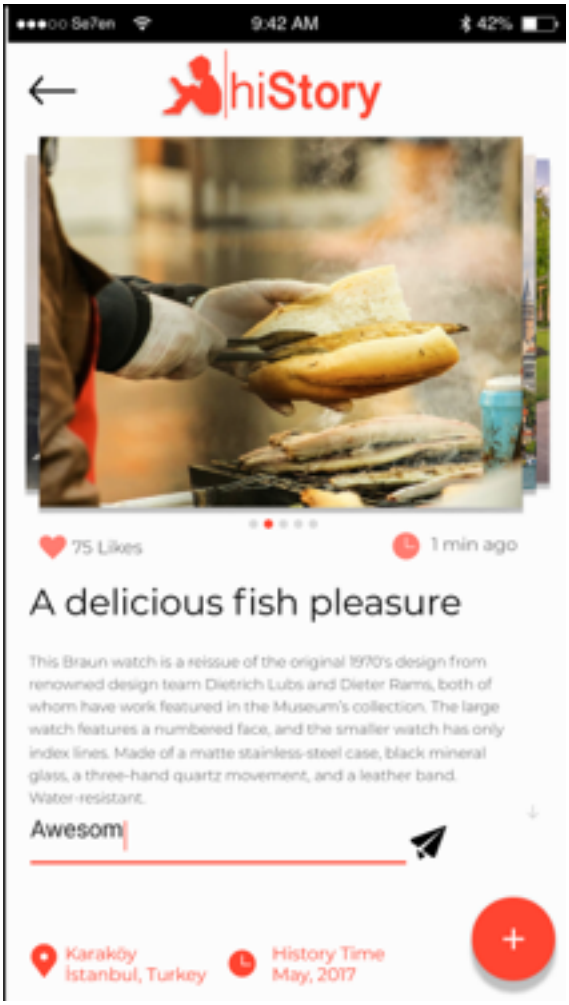
Meet Muzaffer YILMAZ. He is a retired man born in Sivas, but currently lives in İstanbul. He loves to write everything he's been traveling around. One day, While Muzaffer was having breakfast with his grandchildren, Elif, his youngest granddaughter, tells his grandfather that she wants to go outside. Muzaffer can not stand too much and takes his grandkids out and travels out. They wander in Üsküdar to Doğancılar. After a tiring wanderer, Muzaffer opens the hiStory application to write information about the places he travels.

After logging into the HiStory application from his mobile phone,



He opens the Create Memory Post screen, He writes briefly about his impressions of the Doğancılar. He also add some labels and add the current day before he send post. Muzaffer really like to post that wandered. The job that gives pleasure to him.

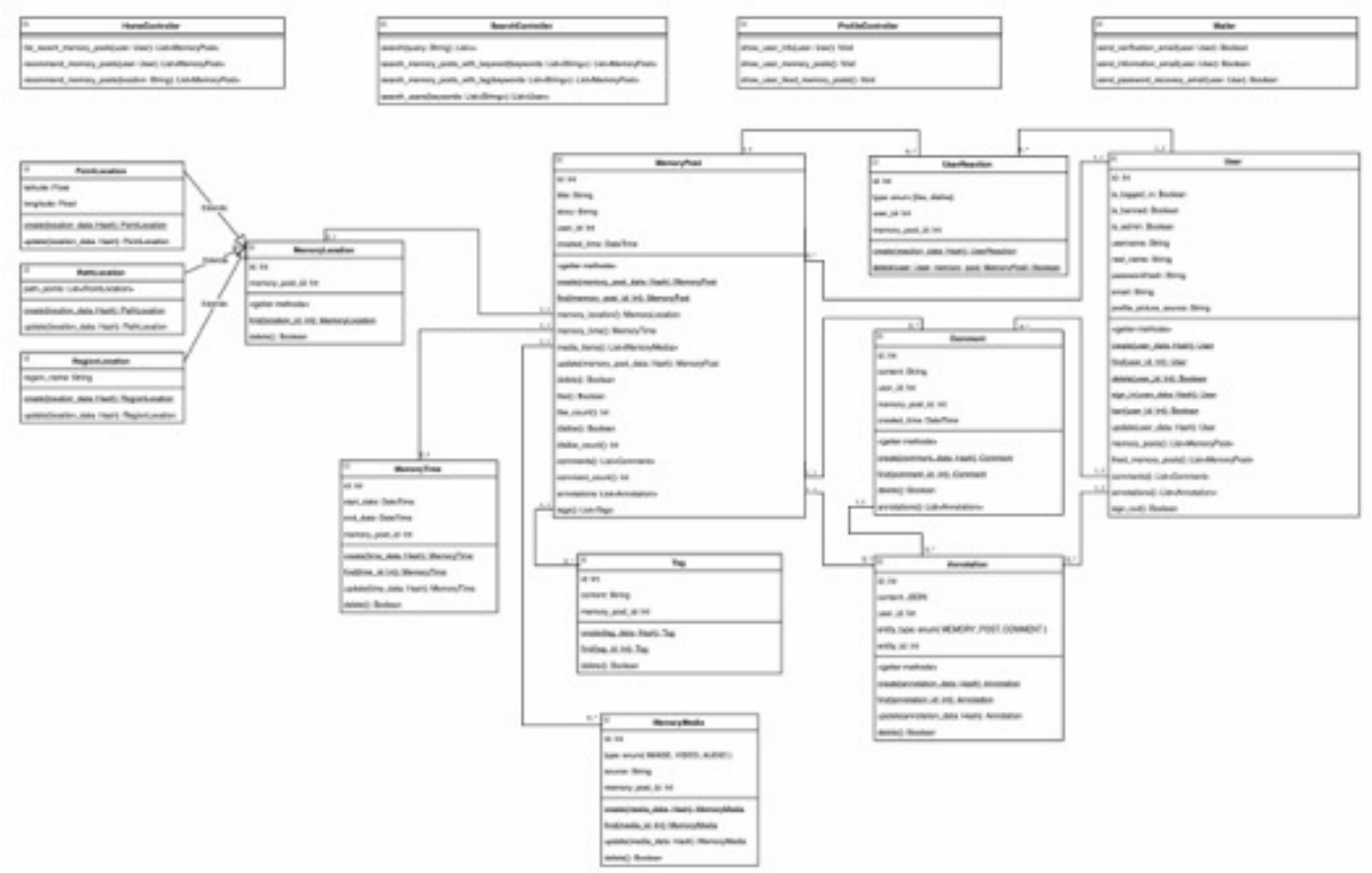
After sending the post, He demand to find recent post about Turkey. Therefore, He click the Search screen and examine the recent posts.



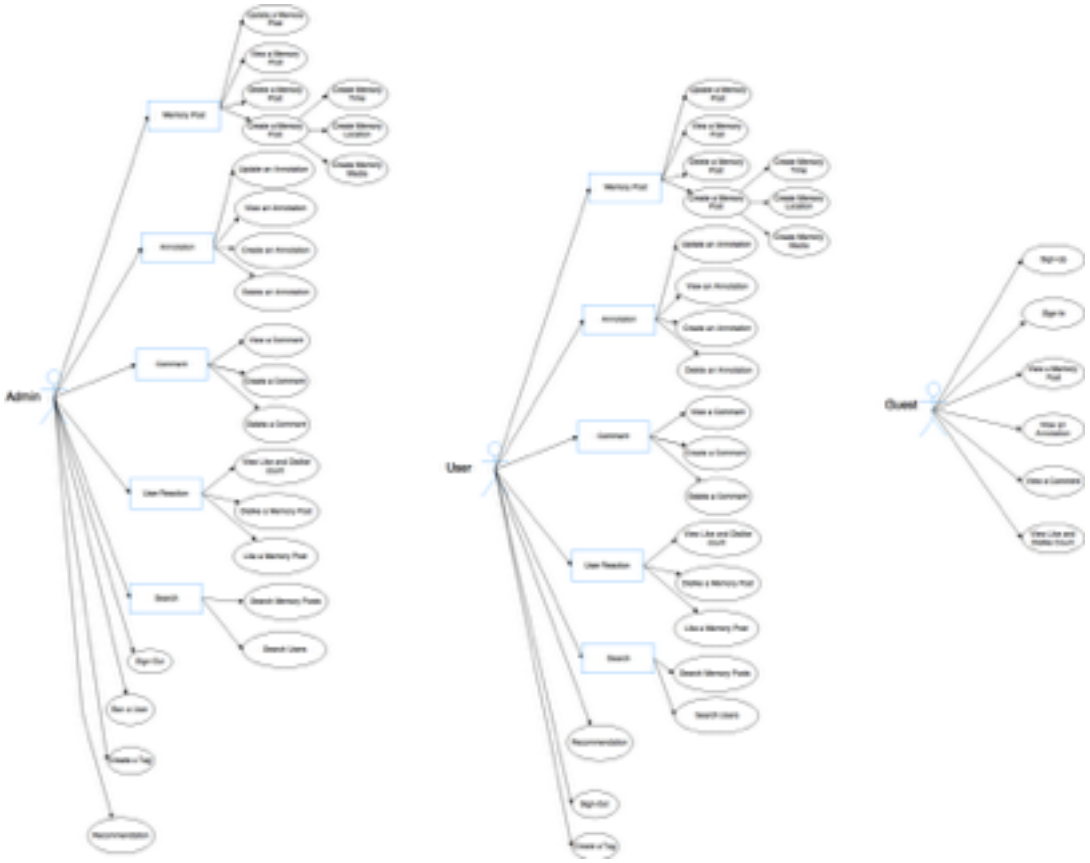
As he scrolling through the page, he suddenly caught the attention of a post which saved in Karaköy about "delicious fish pleasure". This story captures his attention and now he thinks of the times that he had delicious fish in Karaköy. He decides to appreciate this post by commenting “awesome"

7. Software Design

Class Diagram



Use Case Diagram



#1 Make a Comment

Actor: User

Preconditions: * There is a Memory Post to be commented.

Actions: * Clicks the comment section on a memory post. * Writes some text on the comment section. * Clicks `post comment` button.

#2 Delete a Comment

Actor: User

Preconditions: * There is a Memory Post with a comment. * The user owns the Memory Post or the Comment. Otherwise the user must be an Admin.

Actions: * Clicks on the `edit comment` button next to the related Comment on a Memory Post. * Selects `delete comment`.

#3 Create a Memory Post

Actor: User

Preconditions: * The user is not banned.

Actions: * Clicks on `Create New Memory Post` button. * Gives the desired parameters such as Title, Story, Time, Location, Media.

#4 Edit a Memory Post

Actor: User

Preconditions: * The user owns the Memory Post.

Actions: * Clicks the `Edit Post` button on the Memory Post. * Changes or deletes the desired fields. * Clicks `Save Changes` button.

#5 Search

Actor: User or Guest

Preconditions: * The user is on the system.

Actions: * Clicks the `Search` section on the Home Page. * Writes some keywords, tags or usernames

on the search section. * Clicks the `search` button.

#6 Annotate

Actor: User

Preconditions: * There is a Memory Post to be Annotated.

Actions: * Clicks the `Annotate` button on a memory post. * Selects the desired point or area. * Writes the text for annotation. * Clicks the `Done` button.

#7 Like a Memory Post

Actor: User

Preconditions: * There is a Memory Post to be liked.

Actions: * Clicks the `Like` button on a memory post.

#8 Ban a User

Actor: Admin

Preconditions: * The user is assigned as an Admin.

Actions: * Clicks on the User to be banned. * Selects `Actions` button and selects `Ban this User`.

#9 Sign In

Actor: Visitor

Preconditions: * The visitor is on the System. * The visitor has an account that is registered in the database.

Actions: * Clicks the `Sign In` button. * Fills the `E-mail` and `Password` sections. * Clicks `Sign In` again.

#10 Sign Up

Actor: Visitor

Preconditions: * The visitor is on the System.

Actions: * Clicks the `Sign Up` button. * Fills the needed `Username` , `E-mail` and `Password` sections. *

Clicks `Create Account`.

#11 Recommendation

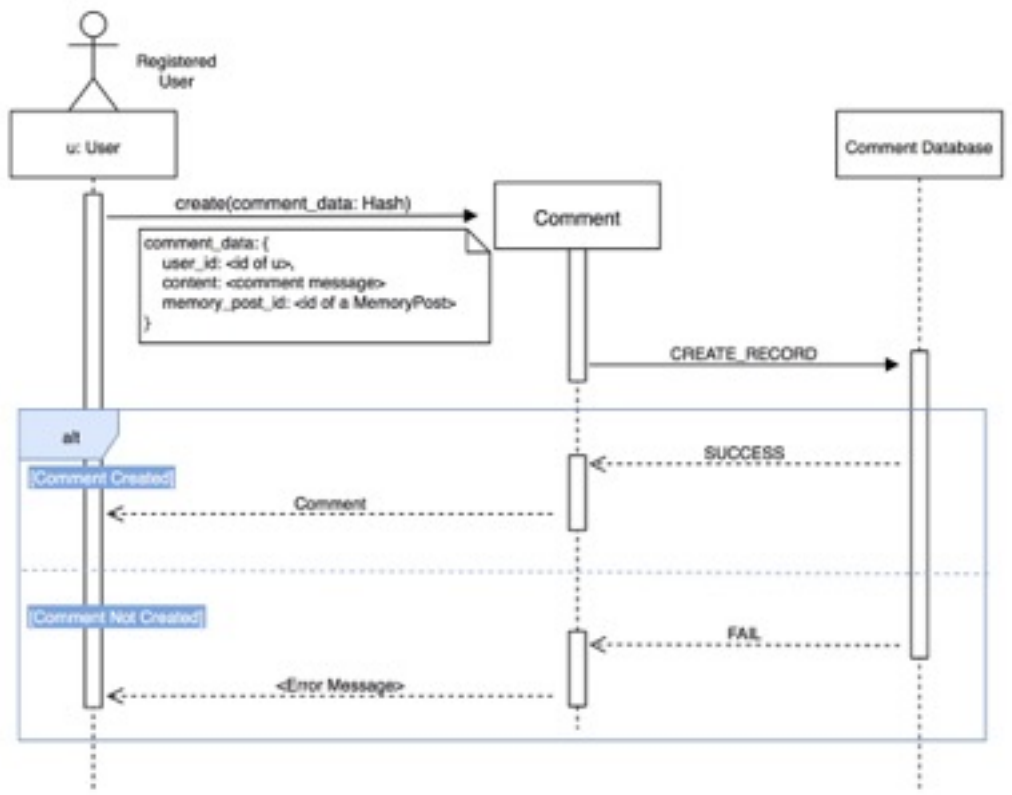
Actor: User

Preconditions: * The user is logged in the System.

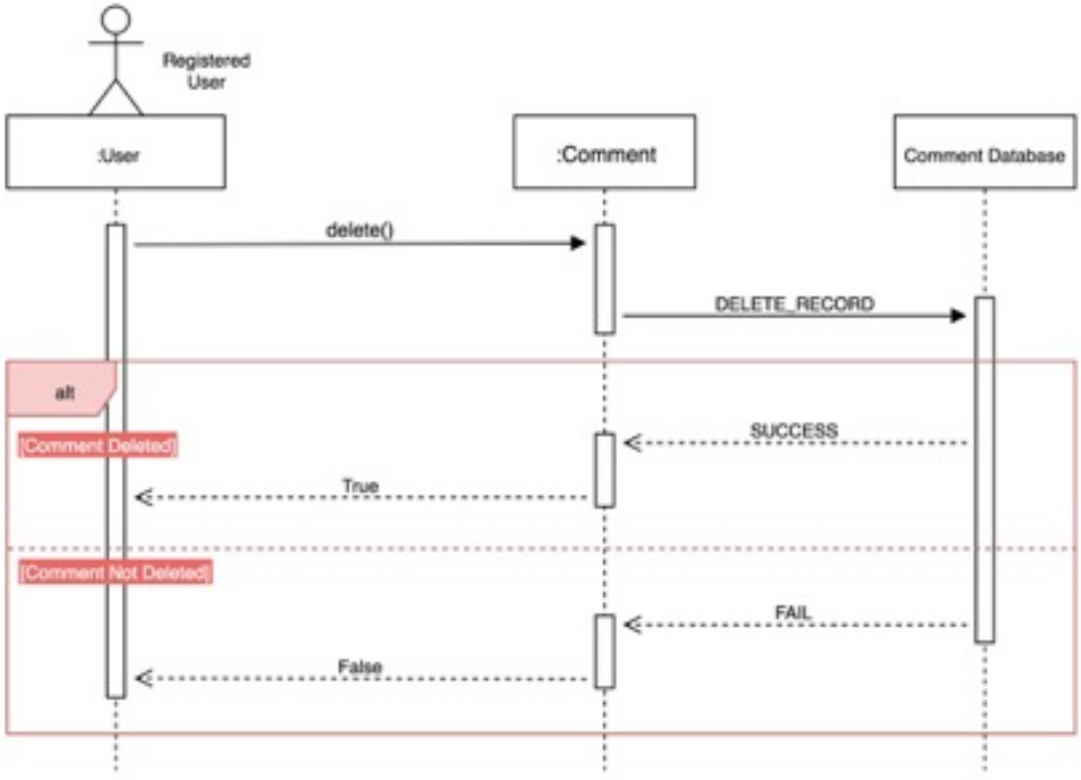
Actions: * Visits the `Home Page`.

Sequence Diagram

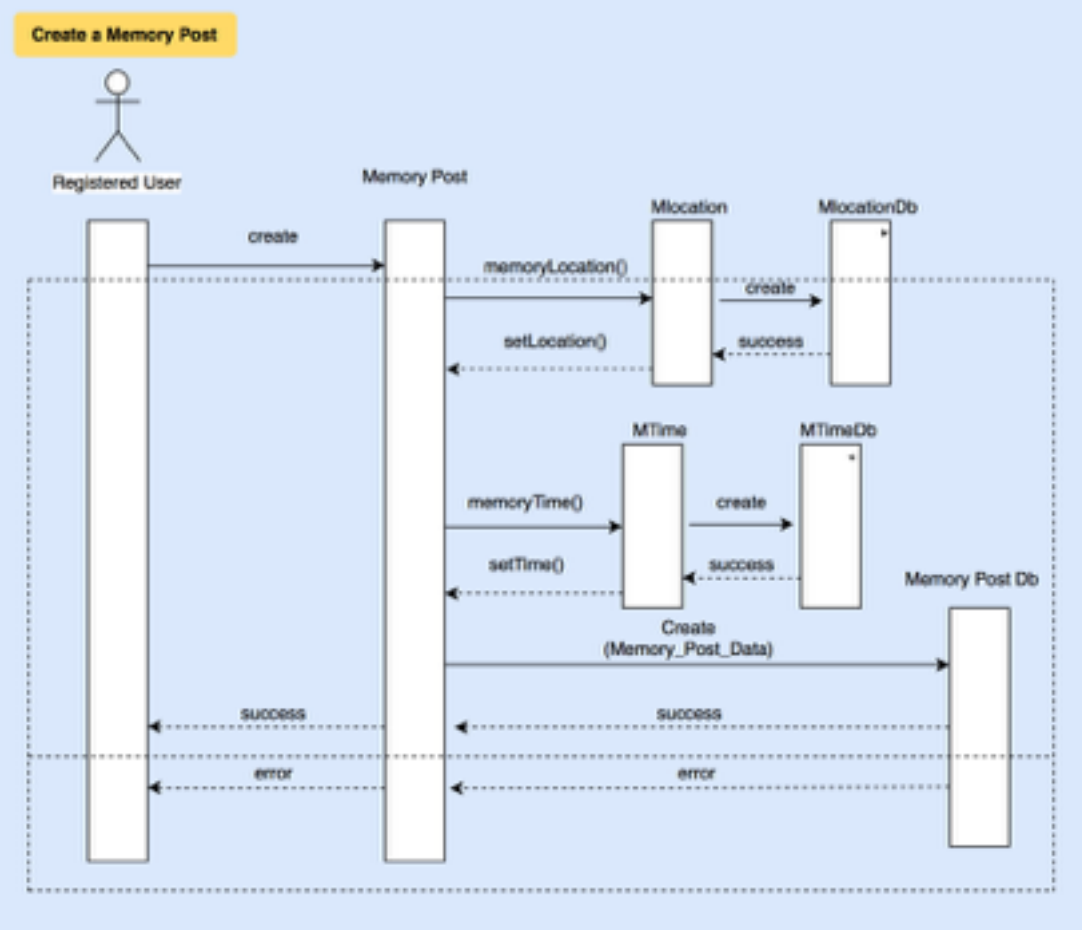
Make a Comment



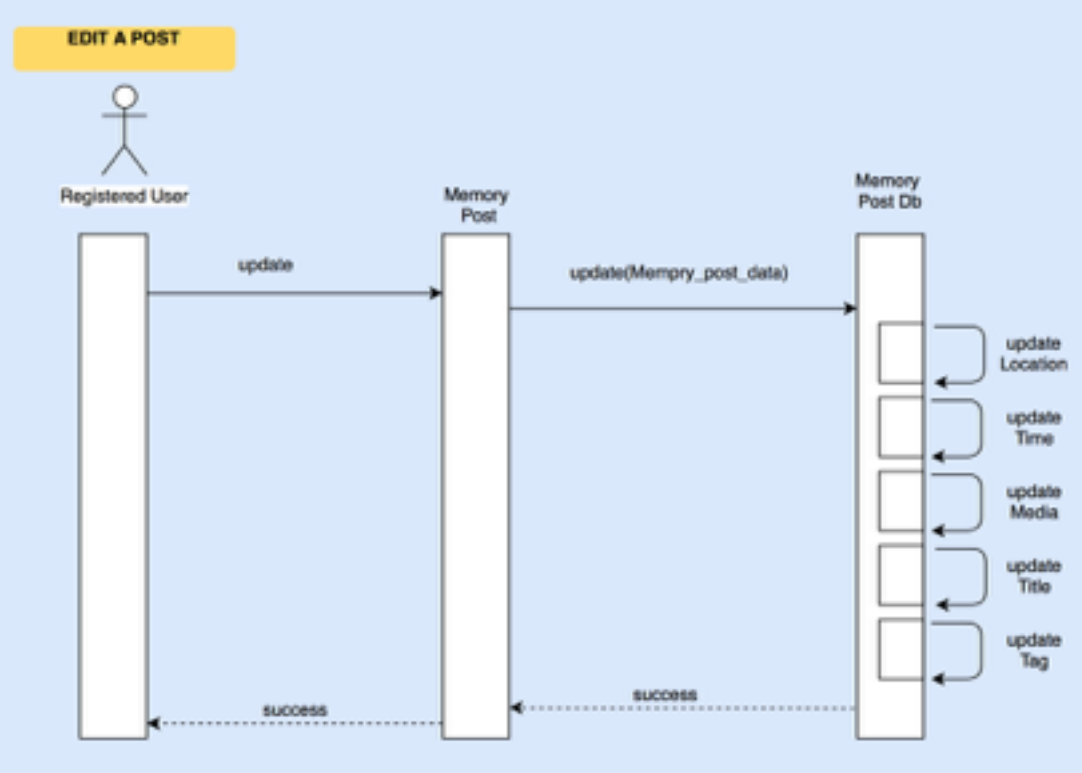
Delete a Comment



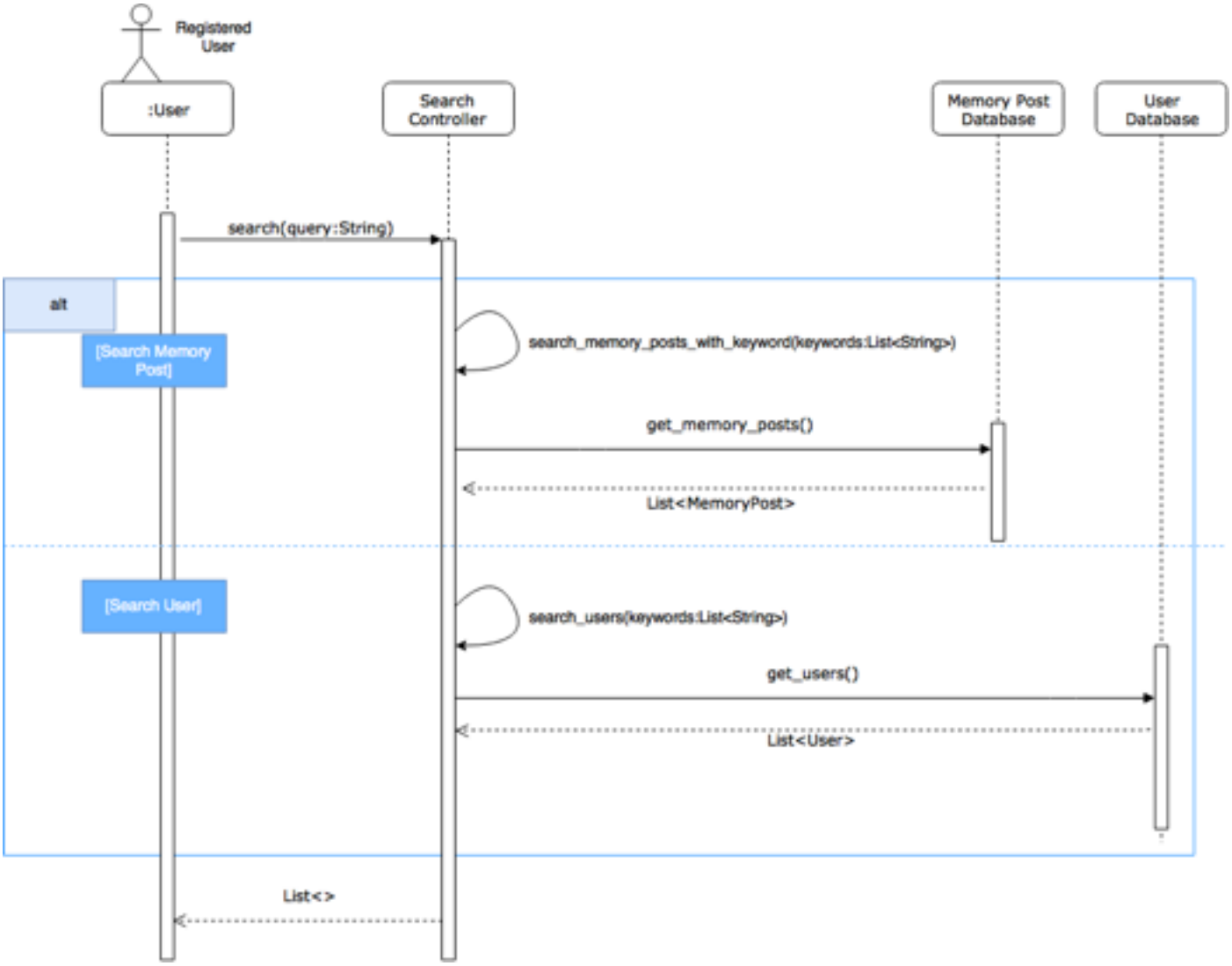
Create a Memory Post



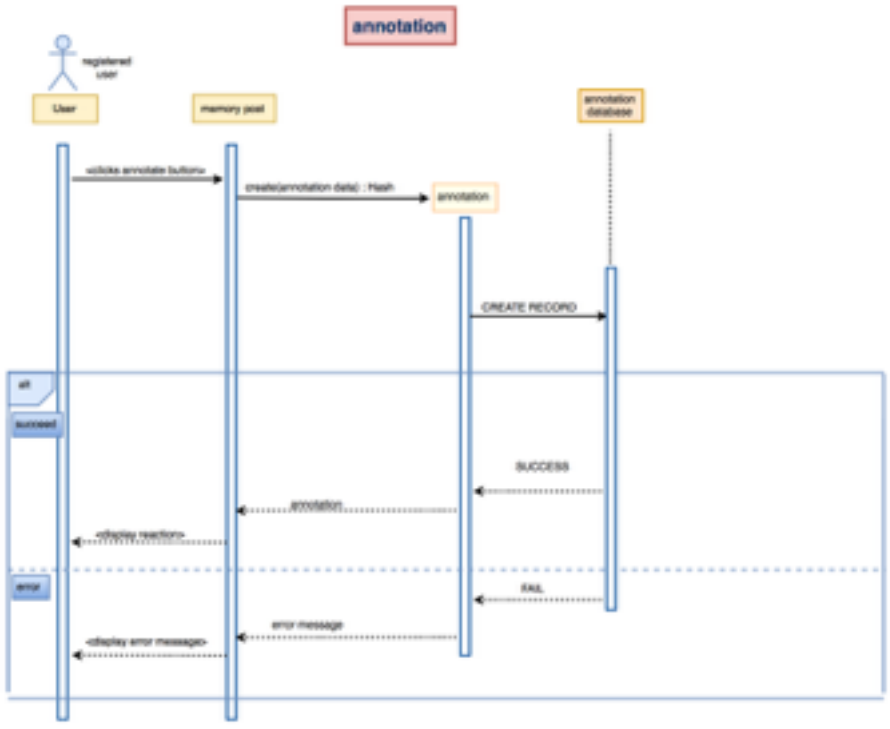
Edit a Memory Post



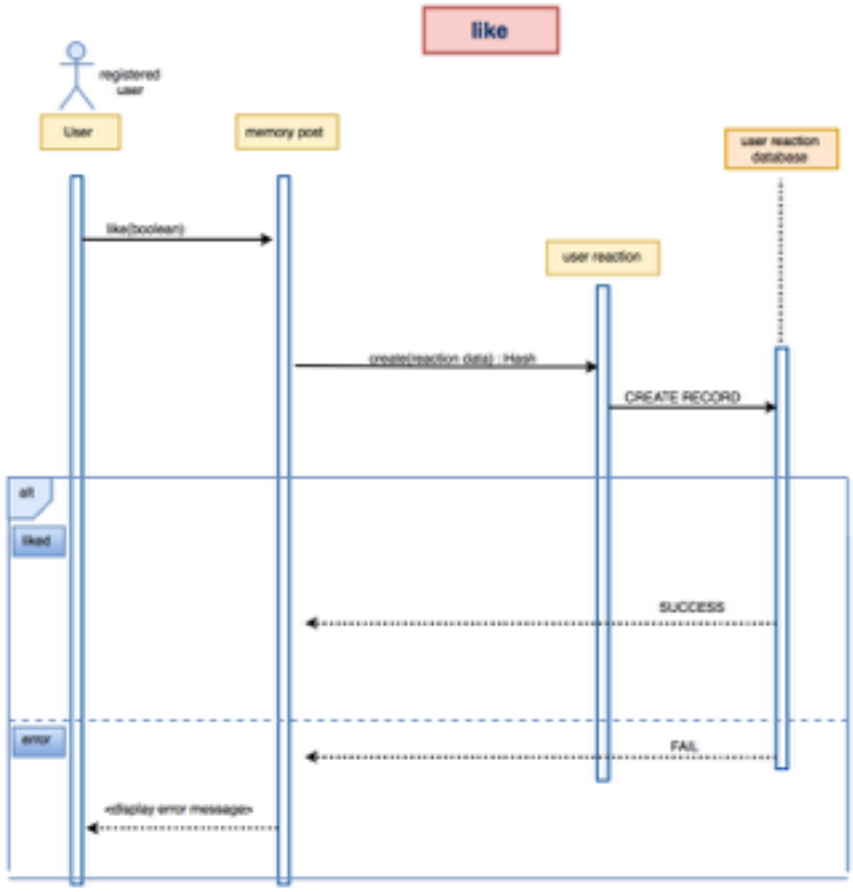
Search



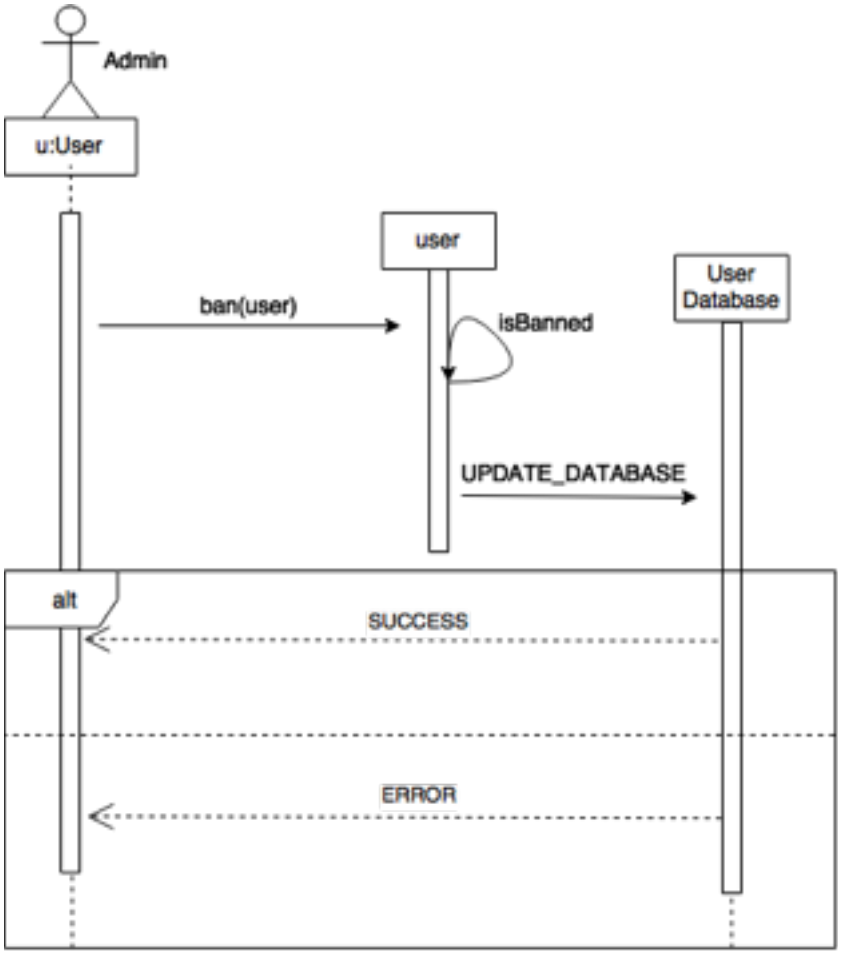
Annotate



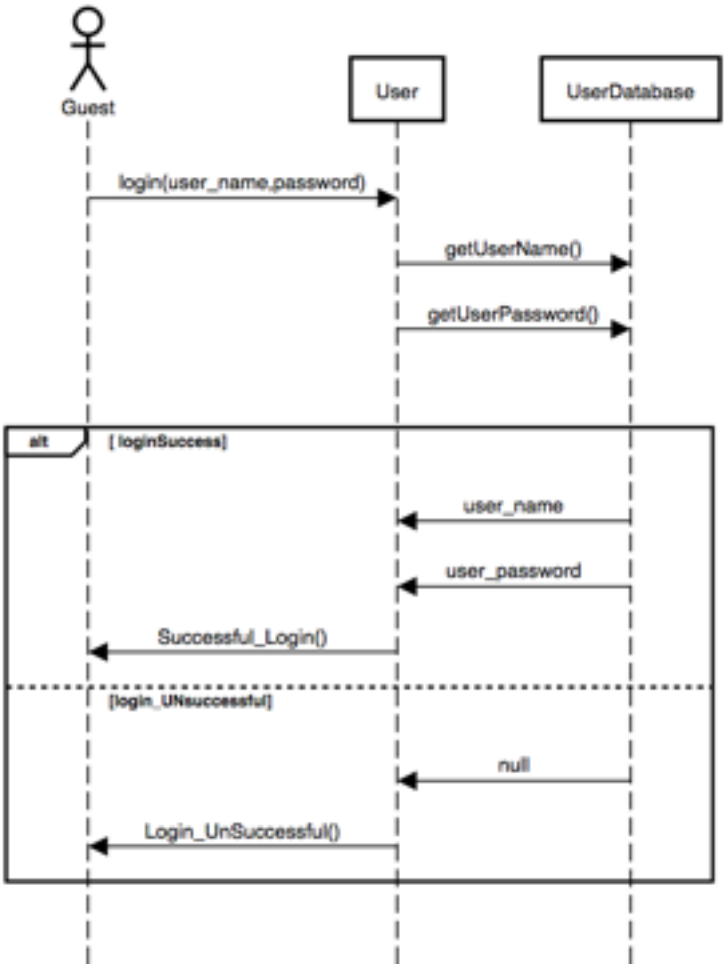
Like a Memory Post



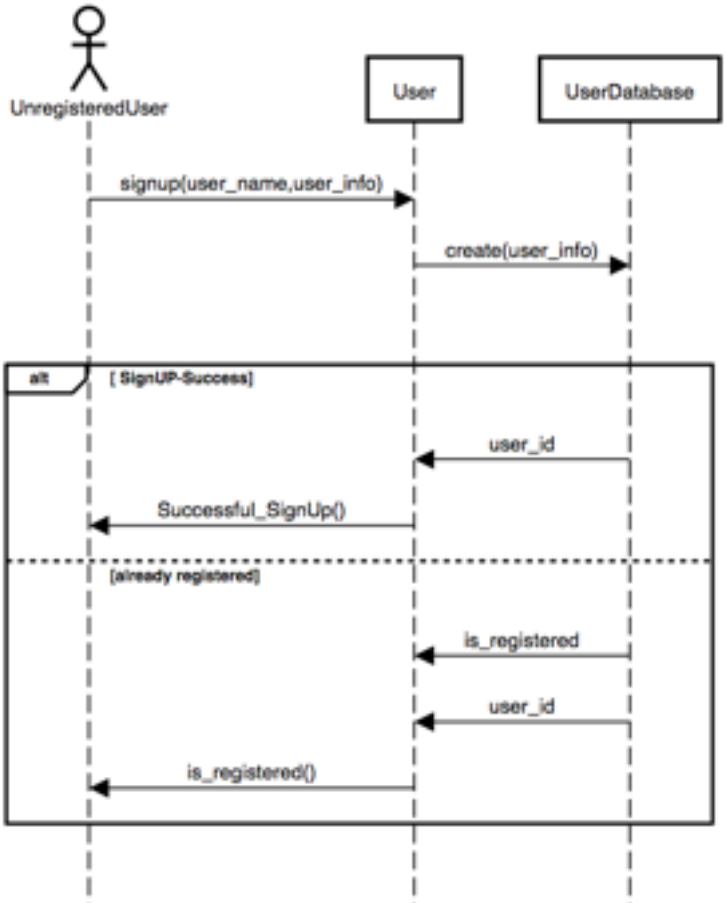
Ban a User



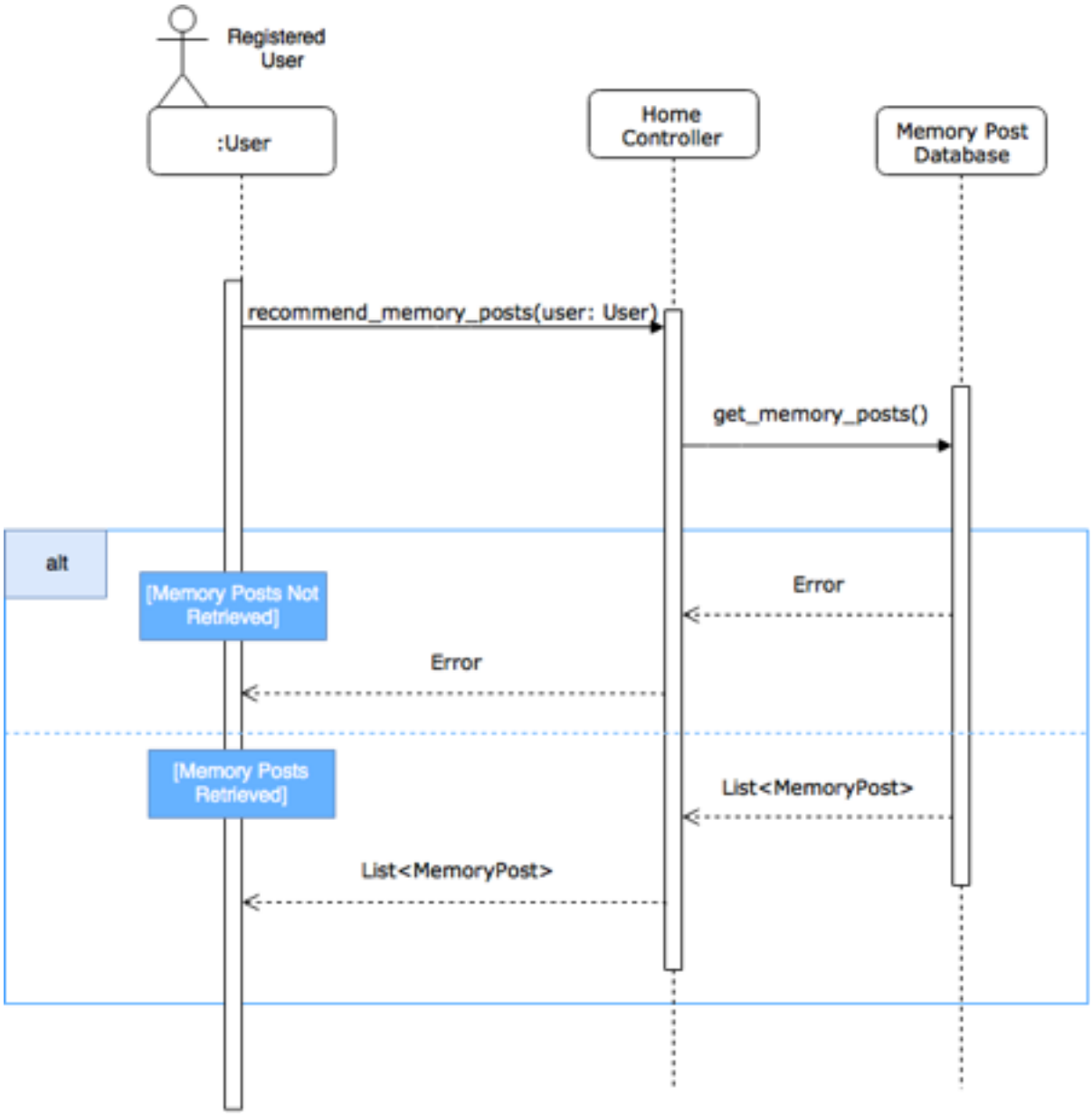
Sign In



Sign Up



Recommendation



7. Test Cases

Test Case 1

Test ID: 1.1

Title: Memory Post Creation

Designed By: Cemal Burak Aygün

Related Requirements: 1.1.4.2

Pre-Conditions: 1. User is signed-in to the system. 2. User is currently on homepage.

STEPS	ACTIONS	INPUT DATA	EXPECTED OUTPUT	ACTUAL OUTPUT	RESULT
1	Click "New Memory Post" button on homepage header.	-	Memory Post creation page is displayed.	?	?
2	Write a title in "Memory Title" text input.	The Maiden's Tower	-	?	?
3	Write a story in "Memory Story" text input.	According to a legend, an emperor has the tower built and places his daughter there to prevent a prophecy which tells that she will be killed by a venomous snake on her 18th birthday. On the 18th birthday, the emperor brings a basket of exotic fruits to his daughter. An asp which hides among the fruits bits the princess and kills her.	-	?	?
4	Find and select the location of The Maiden's Tower in map.	-	-	?	?
5	Select a time (interval) from calender widget.	1700 - 1800	-	?	?
6	Choose an image file from computer.	maidens_tower.png	-	?	?
7	Write comma separated tags in "Tags" text input.	maidens,tower,uskudar	-	?	?

STEPS	ACTIONS	INPUT DATA	EXPECTED OUTPUT	ACTUAL OUTPUT	RESULT
8	Click "Create Memory Post" button.	-	A success message is displayed and user is redirected to the newly created Memory Post page.	?	?

Post-Conditions: 1. "maidens", "tower" and "uskudar" tags are saved to database. 2. Memory Location is saved to database. 3. Memory Time is saved to database. 4. Memory Media is saved to database. 5. Memory Post is saved to database.

Test ID: 1.2

Title: Memory Post Creation without Memory Title and Memory Time

Designed By: Cemal Burak Aygün

Related Requirements: 1.1.4.2, 1.3.9.7

Pre-Conditions: 1. User is signed-in to the system. 2. User is currently on homepage.

STEPS	ACTIONS	INPUT DATA	EXPECTED OUTPUT	ACTUAL OUTPUT	RESULT
1	Click "New Memory Post" button on homepage header.	-	Memory Post creation page is displayed.	?	?
2	Write a story in "Memory Story" text input.	According to a legend, an emperor has the tower built and places his daughter there to prevent a prophecy which tells that she will be killed by a venomous snake on her 18th birthday. On the 18th birthday, the emperor brings a basket of exotic fruits to his daughter. An asp which hides among the fruits bits the princess and kills her.	-	?	?
3	Find and select the location of The Maiden's Tower in map.	-	-	?	?
4	Choose an image file from computer.	maidens_tower.png	-	?	?

STEPS	ACTIONS	INPUT DATA	EXPECTED OUTPUT	ACTUAL OUTPUT	RESULT
5	Write comma separated tags in "Tags" text input.	maidens,tower,uskudar	-	?	?
6	Click "Create Memory Post" button.	-	An error message (saying that the title cannot be empty) is displayed.	?	?
7	Write a title in "Memory Title" text input.	The Maiden's Tower	-	?	?
8	Click "Create Memory Post" button.	-	A success message is displayed and user is redirected to the newly created Memory Post page.	?	?

Post-Conditions: 1. "maidens", "tower" and "uskudar" tags are saved to database. 2. Memory Location is saved to database. 3. Memory Media is saved to database. 4. Memory Post is saved to database.

Test Case 2

Test ID: 2

Title: Comment Creation

Designed By: Cemal Burak Aygün

Related Requirements: 1.1.5.1, 1.3.15.1, 1.3.15.2

Pre-Conditions: 1. User is signed-in to the system. 2. User is currently on a Memory Post page.

STEPS	ACTIONS	INPUT DATA	EXPECTED OUTPUT	ACTUAL OUTPUT	RESULT
1	Click "Add Comment" button on comment section.	-	An error message (saying that comment cannot be empty) is displayed.	?	?
2	Write a comment in "Comment" text input.	I miss the old days. You can see photos from old days of İstanbul here: https://in.pinterest.com/marisacm/istanbul-old-days	-	?	?
3	Click "Add Comment" button on comment section.	-	Comment appears at the top of the comment section. Link "https://in.pinterest.com/marisacm/istanbul-old-days" is clickable.	?	?

Post-Conditions: 1. Comment is saved to database. 2. "Comment Count" information on Memory Post page is updated (increased by 1).

Test Case 3

Test ID: 3.1

Title: Search memory posts with tag

Designed by: Serdar Ada

Related Requirements: 1.1.7.1.2 , 1.3.17.2

Pre-Conditions: 1. User is signed-in to the system. 2. User is currently on homepage.

STEPS	ACTIONS	INPUT DATA	EXPECTED OUTPUT	ACTUAL OUTPUT	RESULT
1	Write a tag in "Search Bar"	#Üsküdar	-	?	?
2	Click "Search" button on homepage header.	-	Search page will be displayed. Memory posts which have the specified tag will be shown at that page	?	?

Post-Conditions:

Test ID: 3.2

Title: Search memory posts with keyword

Designed by: Serdar Ada

Related Requirements: 1.1.7.1.1 , 1.3.17.1

Pre-Conditions: 1. User is signed-in to the system. 2. User is currently on homepage.

STEPS	ACTIONS	INPUT DATA	EXPECTED OUTPUT	ACTUAL OUTPUT	RESULT
1	Write keywords in "Search Bar"	Maiden's Tower	-	?	?
2	Click "Search" button on homepage header.	-	Search page will be displayed. Memory posts which contan the specified keywords at their `title` or `story` will be shown at that page	?	?

Post-Conditions:

Test ID: 3.3

Title: Search users

Designed by: Serdar Ada

Related Requirements: 1.1.7.3 , 1.3.17.3

Pre-Conditions: 1. User is signed-in to the system. 2. User is currently on homepage.

STEPS	ACTIONS	INPUT DATA	EXPECTED OUTPUT	ACTUAL OUTPUT	RESULT
1	Write keywords in "Search Bar"	@muzafferyılmaz	-	?	?
2	Click "Search" button on homepage header.	-	Search page will be displayed. User profile which contain the specified keywords at their `username` or `real name` will be shown at that page	?	?

Post-Conditions:

-

Test Case 4

Test ID: 4.1

Title: Ban A User Scenario (Successful)

Designed by: Enes Koşar

Related Requirements: 1.1.8.2.1

Pre-Conditions: 1. A user who has behaved inappropriately.

STEPS	ACTIONS	INPUT DATA	EXPECTED OUTPUT	ACTUAL OUTPUT	RESULT
1	Click the username of the user to go to profile of the user.	-	The profile of the user must be displayed.	?	?
2	Click the "Three-Dot" button to see the options.	-	The options list must be listed.	?	?
3	Click the "Ban User" button.	-	A pop-up message must have been shown informing that the user is banned successfully.	?	?

Post-Conditions: 1. "isBanned" field of the user has been changed to TRUE in the database. 2. After being banned, the user cannot log in the system.

Test ID: 4.2.1

Title: Ban A User Scenario (Fail)

Designed by: Enes Koşar

Related Requirements: 1.1.8.2.1, 1.1.3.3

Pre-Conditions: 1. A user who has behaved inappropriately. 2. The user has deleted his/her profile immediately.

STEPS	ACTIONS	INPUT DATA	EXPECTED OUTPUT	ACTUAL OUTPUT	RESULT
1	Click the username of the user to go to profile of the user.	-	An error message saying that the user is not found.	?	?

Post-Conditions: 1. No changes in the system due to some inconsistencies.

Test ID: 4.2.2

Title: Ban A User Scenario (Fail)

Designed by: Enes Koşar

Related Requirements: 1.1.8.2.1, 1.1.3.3

Pre-Conditions: 1. A user who has behaved inappropriately. 2. The user database has been halted for an unknown reason.

STEPS	ACTIONS	INPUT DATA	EXPECTED OUTPUT	ACTUAL OUTPUT	RESULT
1	Click the username of the user to go to profile of the user.	-	The profile of the user must be displayed.	?	?
2	Click the "Three-Dot" button to see the options.	-	The options list must be listed.	?	?
3	Click the "Ban User" button.	-	An error message saying that there has been an error while accessing the database.	?	?

Post-Conditions: 1. No changes in the system due to some inconsistencies.

Test Case 5

Test ID: 5

Title: Delete a Comment

Designed By: Dilruba Reyyan Kılıç

Related Requirements: 1.1.5.2 , 1.3.9.5

Pre-Conditions: 1. User is signed-in to the system. 2. User is currently on a Memory Post page. 3. There exists a comment.

STEPS	ACTIONS	INPUT DATA	EXPECTED OUTPUT	ACTUAL OUTPUT	RESULT
1	User clicks on the "Delete Comment" button	-	The Comment is deleted from the viewed Memory Post	?	?

Post-Conditions: 1. Comment is removed from the Database. 2. Memory Post's comment count is be decreased by 1.

Test Case 6

Test ID: 6.1

Title: Edit Post (Successful)

Designed By: Dilruba Reyyan Kılıç

Related Requirements: 1.1.4.3.1

Pre-Conditions: 1. User is signed-in to the system. 2. User is currently on a Memory Post.

STEPS	ACTIONS	INPUT DATA	EXPECTED OUTPUT	ACTUAL OUTPUT	RESULT
1	Clicks "Edit Memory Post" button.	-	Memory Post edit page is displayed.	?	?
2	Clicks on the title field to change the input.	-	Cursor appears on the title field.	?	?
3	Inserts text to change the title.	The Galata Tower	The Galata Tower appears on the title field on the edit page.	?	?
3	Click "Save Edits" button.	-	The title of the Memory Post is changed to "The Galata Tower"	?	?

Post-Conditions: 1. Memory Title of the Memory Post is changed on the Database.

Test ID: 6.2

Title: Edit Post (Failed)

Designed By: Dilruba Reyyan Kılıç

Related Requirements: 1.1.4.3.1 , 1.3.9.7

Pre-Conditions: 1. User is signed-in to the system. 2. User is currently on a Memory Post.

STEPS	ACTIONS	INPUT DATA	EXPECTED OUTPUT	ACTUAL OUTPUT	RESULT
1	Clicks "Edit Memory Post" button.	-	Memory Post edit page is displayed.	?	?
2	Clicks on the title field to change the input.	-	Cursor appears on the title field.	?	?
3	Deletes the title.		The title field is empty.	?	?
4	Click "Save Edits" button.	-	An "A Memory Post must have a Title" Error appears.	?	?

Post-Conditions: 1. Memory Post is not changed.

Test Case 7

Test ID: 7.1

Title: Sign-Up (Success)

Designed By: Ferhat Melih Dal

Related Requirements: 1.2.1

Scenario: Guest wants to sign-up

Pre-Conditions: 1. This is a guest user. 2. The user has not signed-up before.

STEPS	ACTIONS	INPUT DATA	EXPECTED OUTPUT	ACTUAL OUTPUT	RESULT
1	Guest presses Sign-Up button	-	Sign-Up form appears	?	?
2	Guest fills the form with appropriate information	Melih Dal, crazy_cmpeboy_1863, email: melih.dal@boun.edu.tr, pswrd: ACollectionOfHistorySince_1966_ToThe_PresentDay#	-	?	?

STEPS	ACTIONS	INPUT DATA	EXPECTED OUTPUT	ACTUAL OUTPUT	RESULT
3	Guest adds a profile picture	-	Picture is accepted.	?	?
4	Guest presses the Sign_Up button	-	Guest is now a registered user. She is navigated to Homepage.	?	?

Post-Conditions: 1. Homepage is open with user info displayed 2. User can now login using username and password that was entered

---

Test ID: 7.2

Title: Sign-Up (FAIL)

Designed By: Ferhat Melih Dal

Related Requirements: 1.2.1

Scenario: A member wants to sign-up

Pre-Conditions: 1. This is a guest user. 2. The user has an account before, but still tries to sign-up

STEPS	ACTIONS	INPUT DATA	EXPECTED OUTPUT	ACTUAL OUTPUT	RESULT
1	Guest presses Sign-Up button	-	Sign-Up form appears	?	?
2	Guest fills the form with appropriate information	Melih Dal, crazy_cmpeboy_1863, melih.dal@boun.edu.tr, pswrd: ACollectionOfHistorySince_1966_ToThe_PresentDay#	-	?	?
3	Guest adds a profile picture	-	Picture is accepted.	?	?
4	Guest presses the Sign_Up button	-	There is an error message saying "You are already a member. Forgot you password?"	?	?

STEPS	ACTIONS	INPUT DATA	EXPECTED OUTPUT	ACTUAL OUTPUT	RESULT
5	Guest presses types her email address	melih.dal@boun.edu.tr	-	?	?
6	Guest presses "I forgot my password" button.	-	-	?	?

Post-Conditions: 1. An e-mail is sent to "melih.dal@boun.edu.tr", containing email address and reset password instructions

Test Case 8

Test ID: 8

Title: Sign-In

Designed By: Ferhat Melih Dal

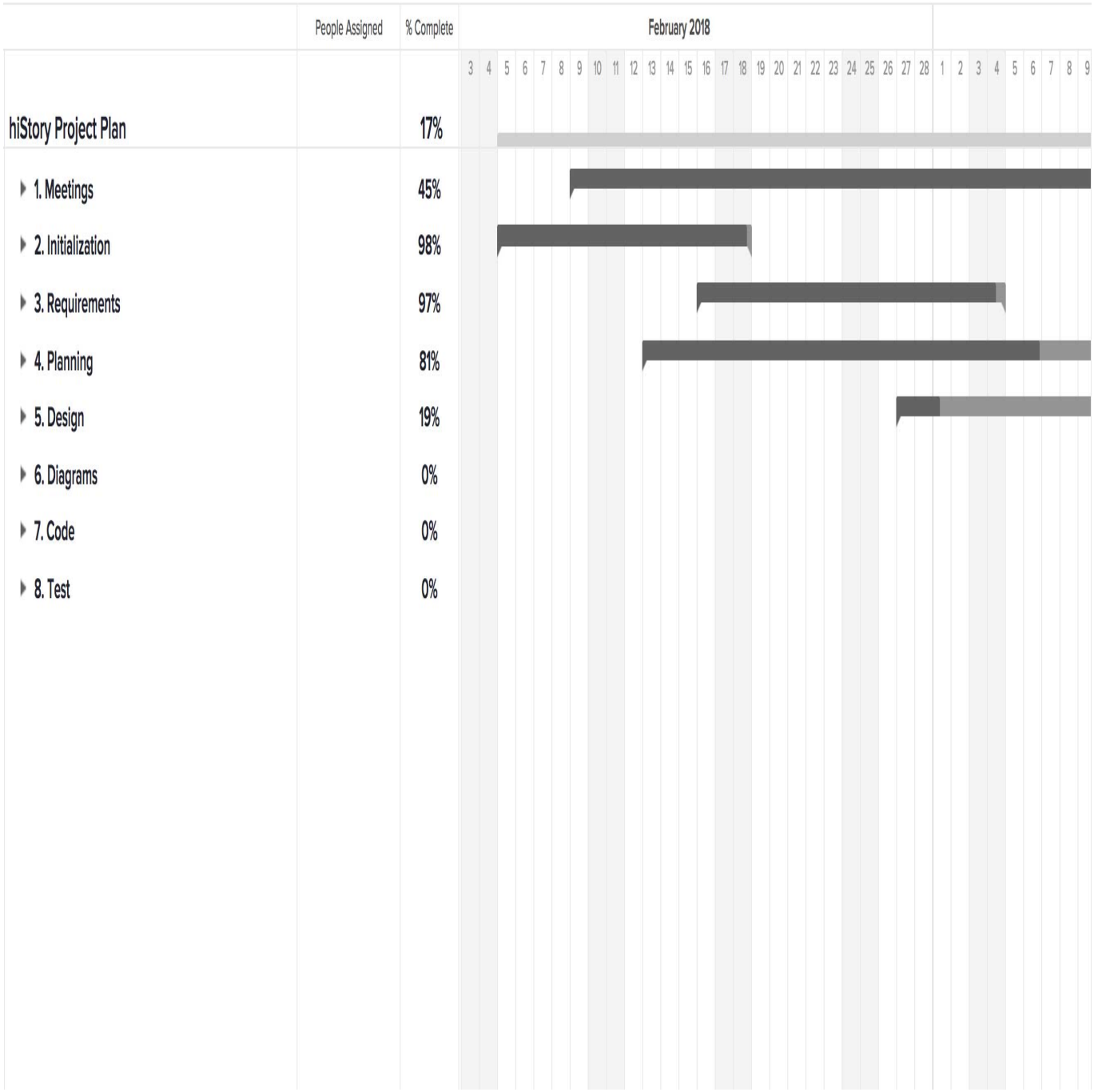
Related Requirements: 1.1.1

Pre-Conditions: 1. User is registered. 2. User has not sign-in yet

STEPS	ACTIONS	INPUT DATA	EXPECTED OUTPUT	ACTUAL OUTPUT	RESULT
1	User presses the Sign-In button	-	Sign-In page opens	?	?
2	User fills username and password info	usr_name: crazy_cmpeboy_1863, password: ACollectionOfHistorySince_1966_ToThe_PresentDay#	-	?	?
3	User presses to sign in button	-	User is navigated to the Homepage	?	?

Post-Conditions: 1. User specific content is retrieved to homepage

Reach the Project Plan



Milestones:

- Milestone 1 - Initialization
- Milestone 2 - Analysis
- Milestone 3 - Planning
- Milestone 4 - Design
- Milestone 5 - Implementation
- Milestone 6 - Testing

Milestone 1

- **Deadline:** 11.02.2018
- **Initialization:**
  - Meet up and get to know each other. Find a team name.
  - Initialize gitHub repository, customize Read.me and create wiki home page.
  - Create personal wiki pages in order to give fundamental access information.
  - Create a template for filling the meeting notes.
- **Deliverables:**
  - README.md
  - wiki Home page
  - Personal Wiki Pages
  - Meeting Notes

Milestone 2

- **Deadline:** 05.03.2018
- **Analysis:**
  - Examine the project description and elicitate Requirements.
  - Revise Requirements Draft and introduce Requirements Version 1.0
  - Make changes according to Customer Meetings and introduce Requirements 2.0
- **Deliverables:**
  - Requirements
  - Customer Meeting 1

Milestone 3

- **Deadline:** 11.03.2018
- **Planning:**
  - Define a Communication Plan in order to introduce the contributors and indicate the outlines of the team plan.
  - Create a High-Level Project Plan that outlines the major activities and milestones.
- **Deliverables:**
  - Communication Plan
  - Project Plan

Milestone 4

- **Deadline:** 23.03.2018
- **Design:**
  - Creating Personas, User Stories and Acceptance Criteria.
  - Creating Web & Android Mock-ups according to the scenarios.
  - Creating Use Case, Class, Activity and Sequence Diagrams.
- **Deliverables:**
  - User Stories & Personas
  - Web Mock-ups
  - Android Mock-ups
  - Use Case Diagrams
  - Class Diagrams
  - Sequence Diagrams

    - Activity Diagrams

  Milestone 5

- **Deadline:** 22.12.2018
- **Implementation:**
    - Create the Code Design of database and API.
    - The full-stack (back-end front-end) code implementation.
    - Review the implemented code.
- **Deliverables:**
    - Code Design
    - Web Service
    - Android Application

  Milestone 6

- **Deadline:** 05.01.2019
- **Testing:**
    - Testing the implemented code.
    - Getting experiences and feedbacks from the customers.
- **Deliverables:**
    - Unit Tests
    - Customer Reviews