

Robin: Final Milestone Report

Group 3

January 6, 2019

1. Final project assessment

Our project was creating a social platform for people who like to attend cultural activities. We started the planning phase in the beginning of the previous semester but since we didn't know what we are going to implement exactly and when are we going to do it, this planning was a little bit optimistic. We learned that most tasks require more time than predicted and act accordingly.

Since we could not comprehend the difficulty of the tasks beforehand, we are experienced hardships during implementation. Some requirements like semantic search could not be implemented because of this. In short, in order to meet the deadlines, sometimes we act out of project plan.

We formed three teams in implementation part; Backend, Android and Frontend. We used GitHub issues for communication. There were times that we could not manage to synchronize our work as different teams but we tried to solve them via issue processing. We learned that communication between and inside teams is very important in order to develop a software.

We tried to conform some protocols such as W3 Web Annotation Protocols, we realized that it is very helpful but needs time to even fully understand it let alone implement it.

One of the most important lessons that we learned was how to and how not to divide time for tasks. We experienced troublesome times because we were not very successful dividing our work.

2. A summary of coding work done by each team member (In tabular format)

Group Member	Member Team	Coding Work
Özge Dinçsoy	Frontend	Web navigation bar, Event page structure, view and style Add event page and style, Google maps setup, Comment view and style, Comment and event component structure, Add comment, Star rating and backend connection,
Recep Deniz Aksoy	Frontend	In addition to bug fixes and backend-frontend connections, implementation of the annotation protocol in frontend, connection of the annotation protocol to backend. Implementation of the view and functionality of the annotation in frontend. Functionality of the comments, backend connection of the comment feature.
Uğurcan Arıkan	Frontend	Changed the view of the create event page, added new fields such as photo addition and connected to backend, profile page backend connection, profile edit page backend connection, event page buttons and backend connections, edit event with addition of photos, follow and unfollow profiles, join and attend events, delete event for creator, follow and unfollow tags for users, add tags for events, create tags, home page event component modifications, settings page, privacy setting backend connection, guest user view, guest user notification bar, guest user navigation bar, web navigation bar finalized, added creator field to the event page, changed paths of event page and profile page to enable guest view
Çağdaş Tombul	Android	Simple user event layout for profile page, log in and sign up pages, warnings for users, use as guest pages, date and time pickers for create event page, forgot password page, event comments and ratings, relevant events and users on the profile page, edit profile pages, event related tags on the homepage, profile photos fetching, app design
Mustafa Şişik	Android	Create event page, search event and user functionalities, log in and sign up pages, event page design, profile page creation, event ratings, interested or attending functionalities, follow or unfollow a user, event photos fetching, app design
Onur Varkıvanç	Backend	Designing and updating models for events, ratings, tags and comments. Adding inter-model relationships between comments, ratings, images,

		events and users. Creating endpoints to for creating and deleting comments and users. Creating endpoints to edit existing comments. Implementing search for tags and comments. Implementing rating functionality and associated endpoints. Revamping rating system. Adding upvote/downvote system for comments. Setting up default profile picture directory. Reshaping server response to be more informative. Adding flagging system. Implementing messaging models, relations, serializers and views. Writing the documentation for backend endpoints. Adding google maps location coordinates.
Furkan Enes Yalçın	Backend	In addition to previous milestones, designing and updating views and urls. Implementing image upload system and models. Implementing annotation model.
Oğuzhan Yetimoğlu	Backend	Designing and updating serialization methods and models.

3. Requirements

1.Functional Requirements

1.1 System Requirements

1.1.1 Cultural Event Items

- 1.1.1.1 Cultural event items shall include theaters, musicals, exhibitions, museums and travelling internationally or domestically
- 1.1.1.2. The system shall provide performing artist information on the event page.
- 1.1.1.3 The system shall provide venue(location). The location shall be provided by Google Maps API.
- 1.1.1.4 The system shall provide event date.
- 1.1.1.5 The system shall provide event price.
- 1.1.1.6 The system shall provide additional information about the type of the event if necessary.
- 1.1.1.7 The system shall provide rating and comments of the event.
- 1.1.1.8 The system shall provide semantic tags.
 - 1.1.1.8.1 The tags can include genre.
 - 1.1.1.8.2 The tags can include artist country of origin.
 - 1.1.1.8.3 The tags can include any other piece of information based on user's choice.

1.1.2 Search

- 1.1.2.1 Search function shall have two options whether basic or advanced search.

- 1.1.2.2 Search function shall be based on "artist name", "location", "date", "price", "rating" or "semantic tags" that are related to given input by user.
- 1.1.2.3 Search function shall be used with more than one input.
- 1.1.2.4 The results of a search shall be limited by the given location.
- 1.1.2.5 The system shall support semantic search.

1.1.3 Recommendation

- 1.1.3.1 The system shall recommend events to users.
- 1.1.3.2 The recommendation system shall work based on user's event histories, users they followed and interests.

1.1.4 Annotation

- 1.1.4.1 All users shall be able to view annotations.
- 1.1.4.2 Registered users shall be able to create, edit and delete annotations.
- 1.1.4.3 Registered users shall be able to annotate an image.
 - 1.1.4.3.1 Users shall be able to select the fragments of the images for annotations.
- 1.1.4.4 Registered users shall be able to annotate event descriptions.
- 1.1.4.5 Registered users shall be able to include links in annotations.
 - 1.1.4.5.1 The links shall be able to refer to semantic resources for providing more detailed information about the story.
- 1.1.4.6 Registered users shall be able to annotate geographical references.
- 1.1.4.7 Registered users shall be able to include images in annotations.

1.1.5 Admin Panel

- 1.1.5.1 Admins shall be able to add, delete and edit content through admin panel.
- 1.1.5.2 Admins shall be able to have the right of banning users and events.

1.2 User Requirements

1.2.1 Registered Users

1.2.1.1 Registration and Login

- 1.2.1.1.1 Users shall be able to sign up by providing necessary personal information.
- 1.2.1.1.2 Users shall be able to sign up by linking their Facebook account.
- 1.2.1.1.3 Registered users shall be able to log in using the method they signed up.
- 1.2.1.1.4 Users shall be able to reset their passwords via a secure e-mail.
- 1.2.1.1.5 Users shall be able to verify their profiles.

1.2.1.2 Profiles

- 1.2.1.2.1 Users shall be able to customize their profiles by changing/adding profile photos, and including a short bio and/or personal details.
- 1.2.1.2.2 Users shall be able to see the profiles of other users.
- 1.2.1.2.3 Users shall be able to block other users.
- 1.2.1.2.4 Users shall be able to follow other users.

- 1.2.1.2.5 Users shall be able to rate and/or comment on other users.
- 1.2.1.2.6 User profiles should contain a short list of events which the user has attended.

1.2.1.3 Messaging

- 1.2.1.3.1 Users shall be able to message other users.
- 1.2.1.3.2 Users shall be able to decide if they want to receive messages from strangers.

1.2.1.4 Recommendations

- 1.2.1.4.1 Users shall be able to watch certain event types to get notifications.
- 1.2.1.4.2 Users shall be able to block certain the event types to avoid spamming.
- 1.2.1.4.3 Users should be able to get notifications about events that the users they follow will attend.

1.2.1.5 Tags

- 1.2.1.5.1 Users shall be able to classify events using tags.
- 1.2.1.5.2 Profiles should include tags that the user is interested in.
- 1.2.1.5.3 Users shall be able to classify events by their locations.

1.2.1.6 Searching

- 1.2.1.6.1 Users shall be able to search events by name.
- 1.2.1.6.2 Users shall be able to search events by tags.
- 1.2.1.6.3 Users shall be able to search for other users.
- 1.2.1.6.4 Users shall be able to search and filter events by location.
- 1.2.1.6.5 Users shall be able to filter their search by ratings.
- 1.2.1.6.6 Users shall be able to search for events by applying date restrictions.
- 1.2.1.6.7 Users shall be able to search to find annotations.
- 1.2.1.6.8 Users shall be able to search with parameters that include time intervals.

1.2.1.7 Events and Event Pages

- 1.2.1.7.1 Users shall be able to create event pages for both upcoming and past events.
- 1.2.1.7.2 Users shall be able to mark upcoming events as "going" and past events as "attended".
- 1.2.1.7.3 Users shall be able to annotate event information and multimedia added to an event page with explanations, questions, answers, location and links.
- 1.2.1.7.4 Users shall be able to see highly voted, nearby upcoming events in their personal homepage labelled with the tags they are watching.
- 1.2.1.7.5 Users shall be able to rate reflecting their experience on cultural event pages.
- 1.2.1.7.6 Users shall be able to put comments reflecting their experience on cultural event pages.
- 1.2.1.7.7 Users shall be able to share cultural events.
- 1.2.1.7.8 Users shall be able to follow cultural events.

- 1.2.1.7.9 Users shall be able to see how many people will attend or attended an event.

1.2.1.8 User Interface

- 1.2.1.8.1 Users should be able to change the website's colour scheme.

1.2.1.9 Comments

- 1.2.1.9.1 Users shall be able to vote comments by upvoting or downvoting them.

1.2.2 Guest Users

- 1.2.2.1 Guests shall be able to see event pages created by users.
 - 1.2.2.2 Guests shall be able to see comments and ratings of users.
 - 1.2.2.3 Guests should see a sign up dialogue on the homepage.
-

2. Non-Functional Requirements

2.1 Availability and Accessibility

- 2.1.1 The system should work on any web browser.
- 2.1.2 The system should be available %99 of the time.
- 2.1.3 If a failure happens, system should be back up within 1 hour.
- 2.1.4 The application shall be provided for both web and mobile platforms.
- 2.1.5 At least one of the colour schemes should be suitable for colour-blind people.

2.2 Performance and Response time

- 2.2.1 The system shall work with up to a thousand users without crashing.
- 2.2.2 The system shall respond to all search requests in 1 seconds.

2.3 Reliability

- 2.3.1 The system shall back up its database every 1st and 15th day of the month to prevent user related data loss.
- 2.3.2 The system shall encrypt all user passwords before storing them in the database.

2.4 Security

- 2.4.1 The system should use HTTPS, so that the traffic between browser (or app) and the web server is encrypted.
- 2.4.2 System should deny requests if excessive amount of requests has been made.
- 2.4.3 System should be secure against SQL injections.
- 2.4.4 System should be secure against cross site scripting.

- 2.4.5 System should be secure against file upload vulnerabilities.
- 2.4.6 System should not create weak session IDs so that session hijacking is impossible.

2.5 Portability

- 2.5.1 The application shall be compatible with various Android devices and web browsers.

2.6 Annotations

- 2.6.1 Web Annotation Protocol shall be used to store annotations.
- 2.6.2 The annotations shall be tested to assure validity.
- 2.6.3 JSON-LD shall be used for annotation representation according to W3C documentation.

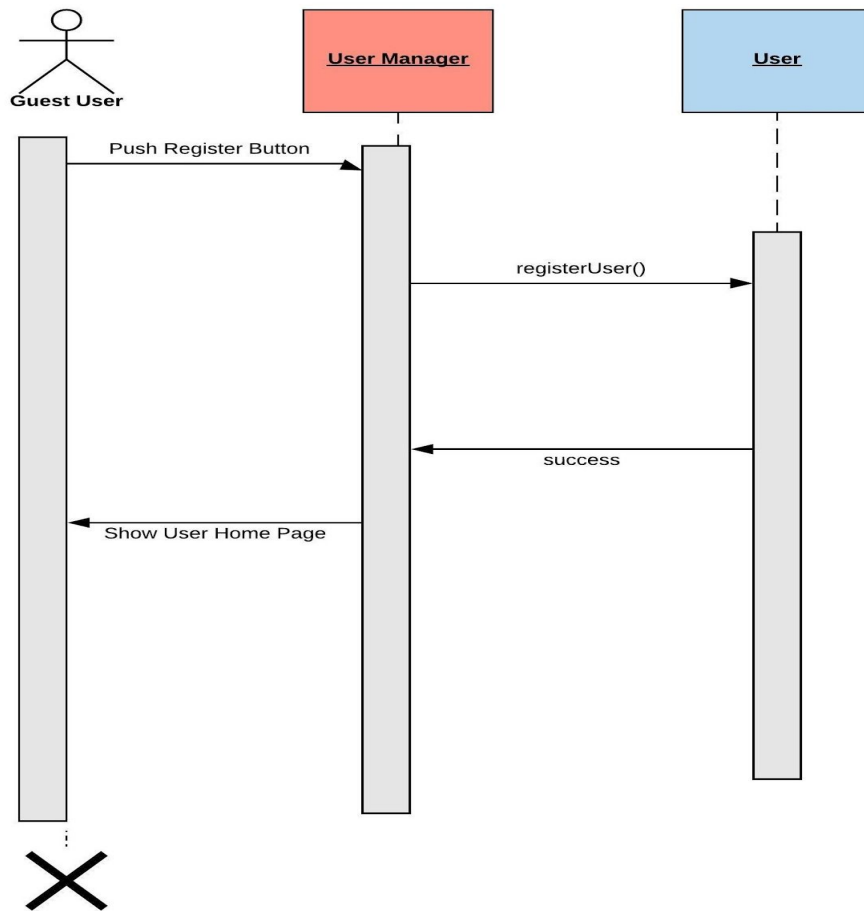
2.7 Documentation / Coding Style

- 2.7.1 Documentation and coding style should be consistent with the standard of the language used. For example, if Python will be used for back-end, PEP-8 style guide should be followed.

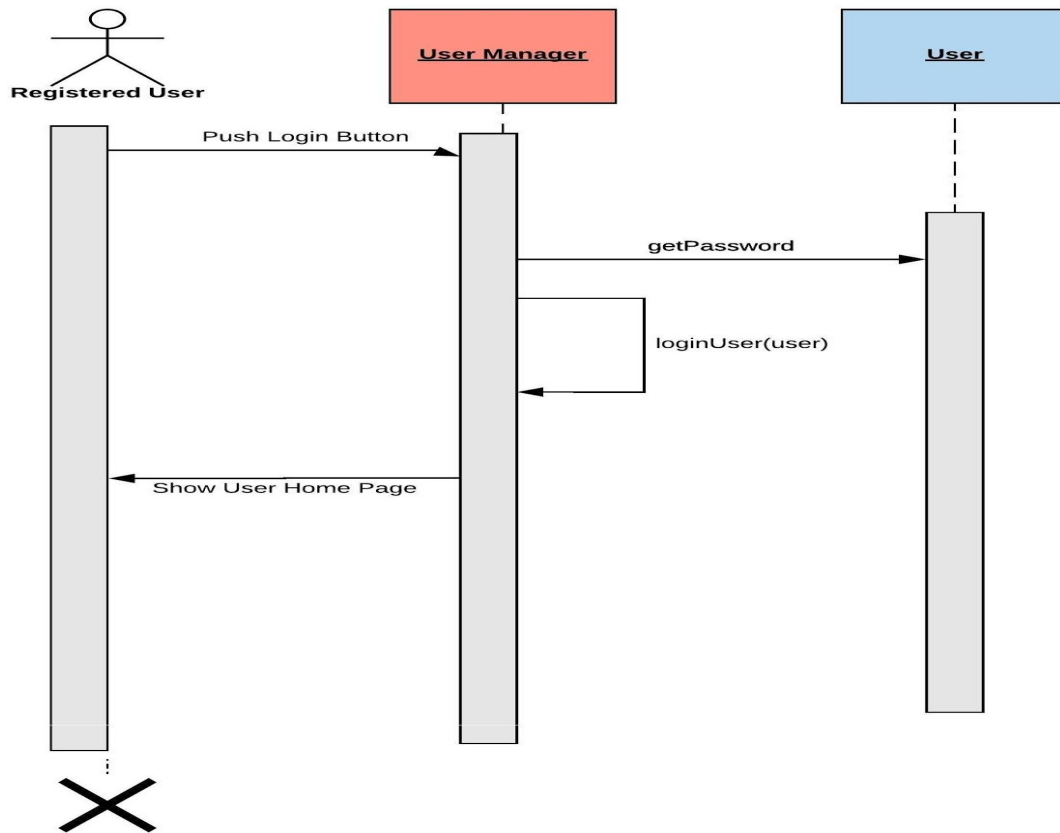
4. All design documents

- Sequence Diagrams

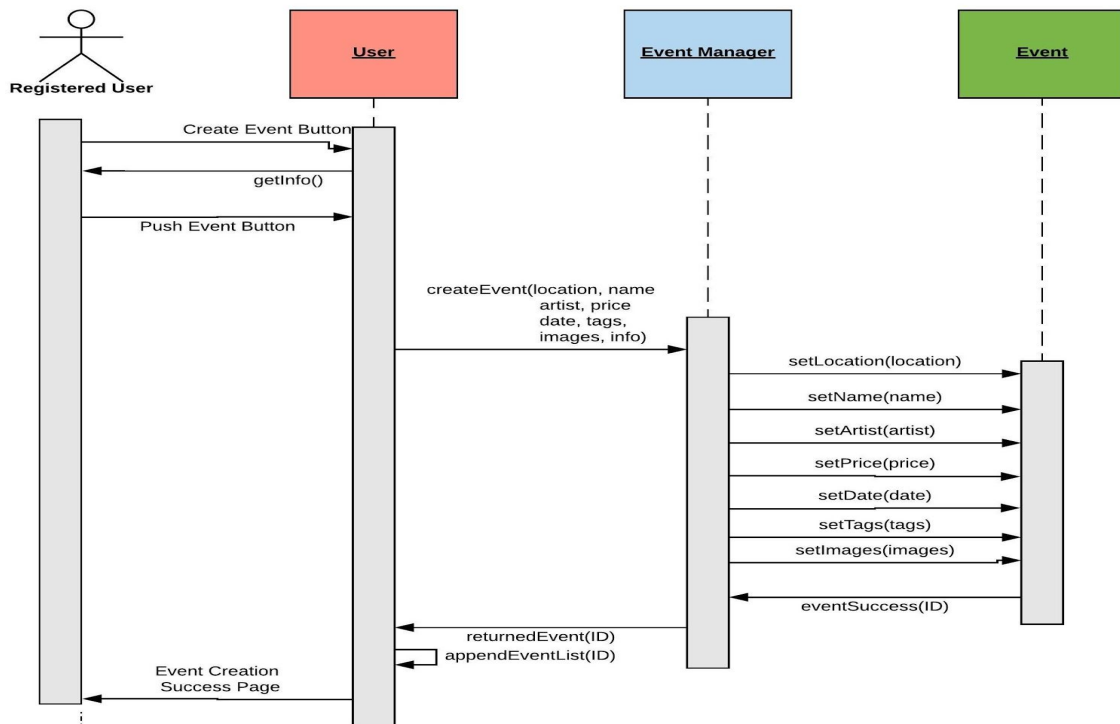
1. Register



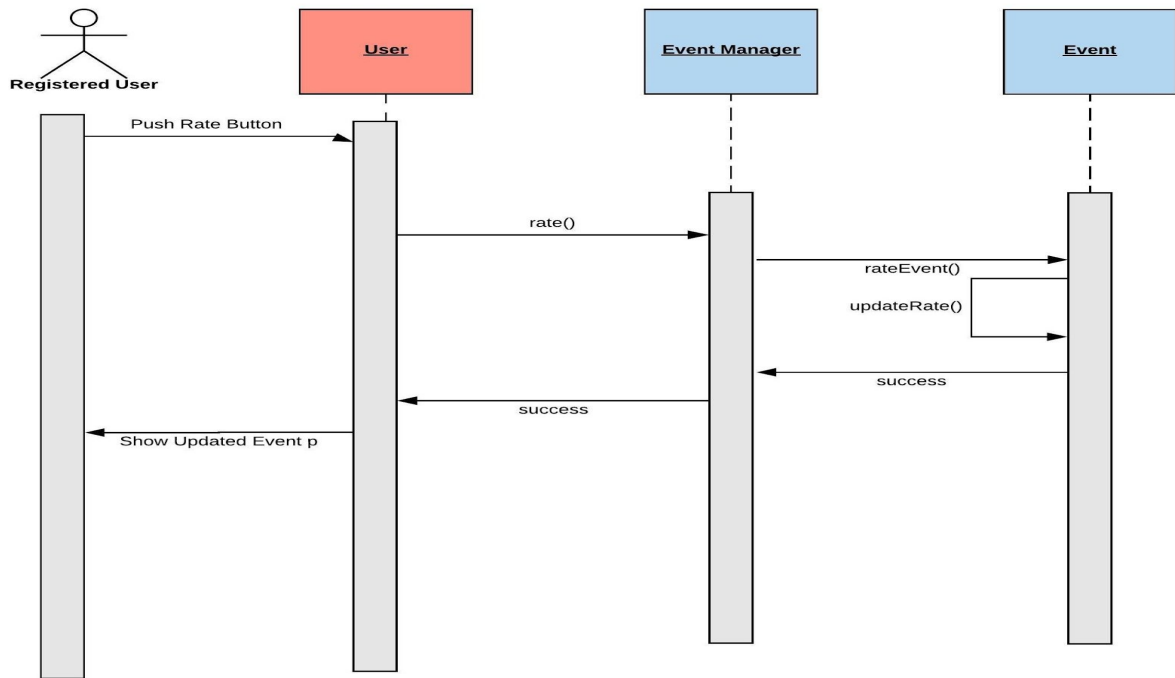
2. Login



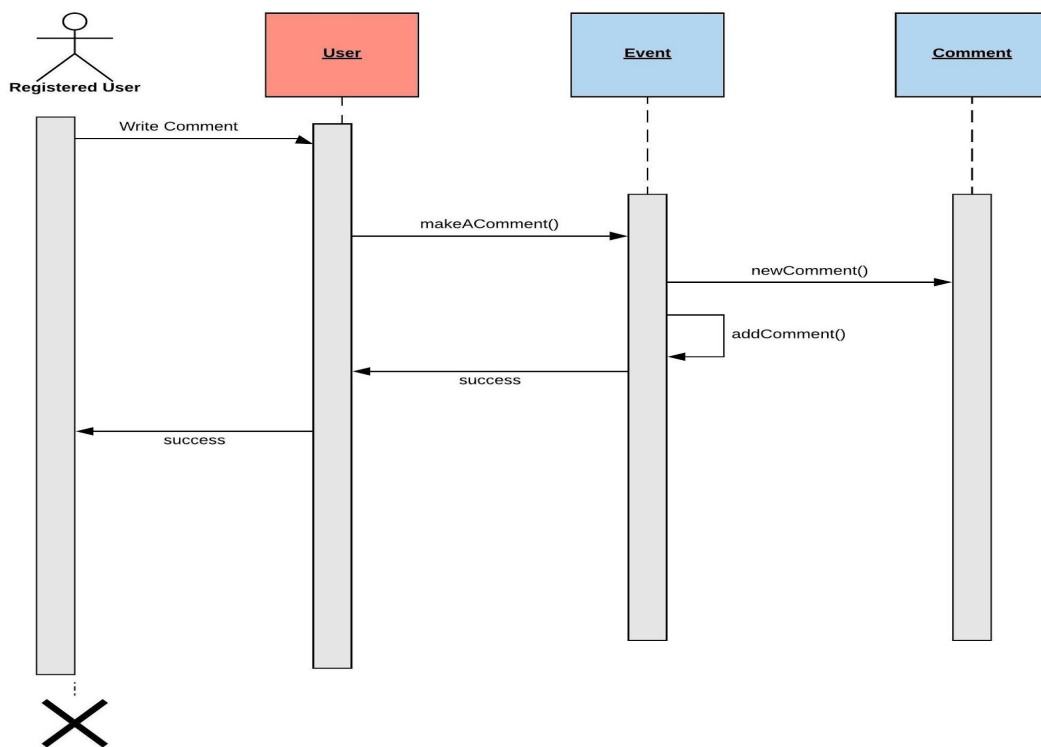
3. Event Creation



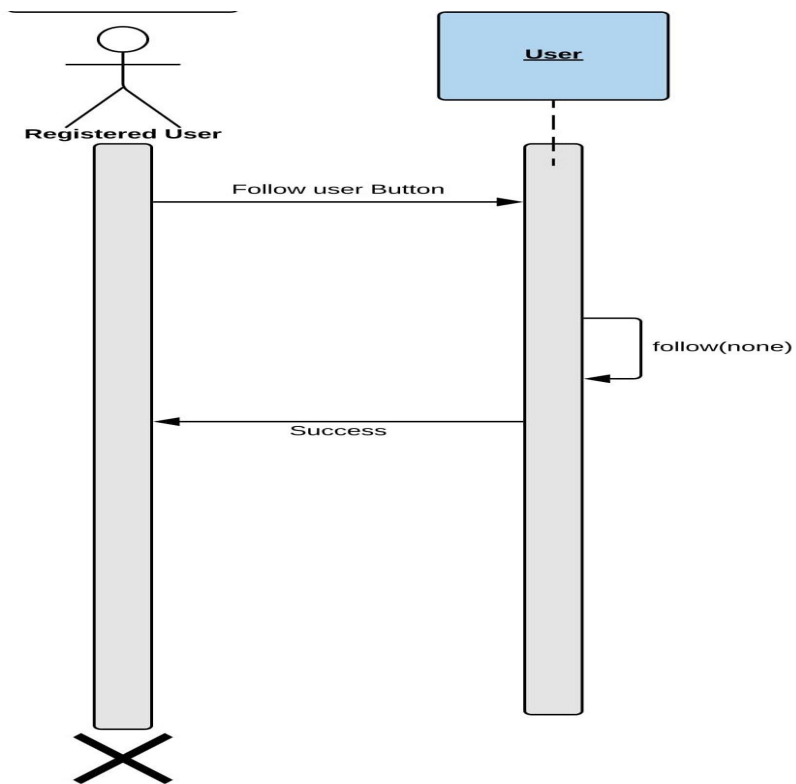
4. Rate an Event



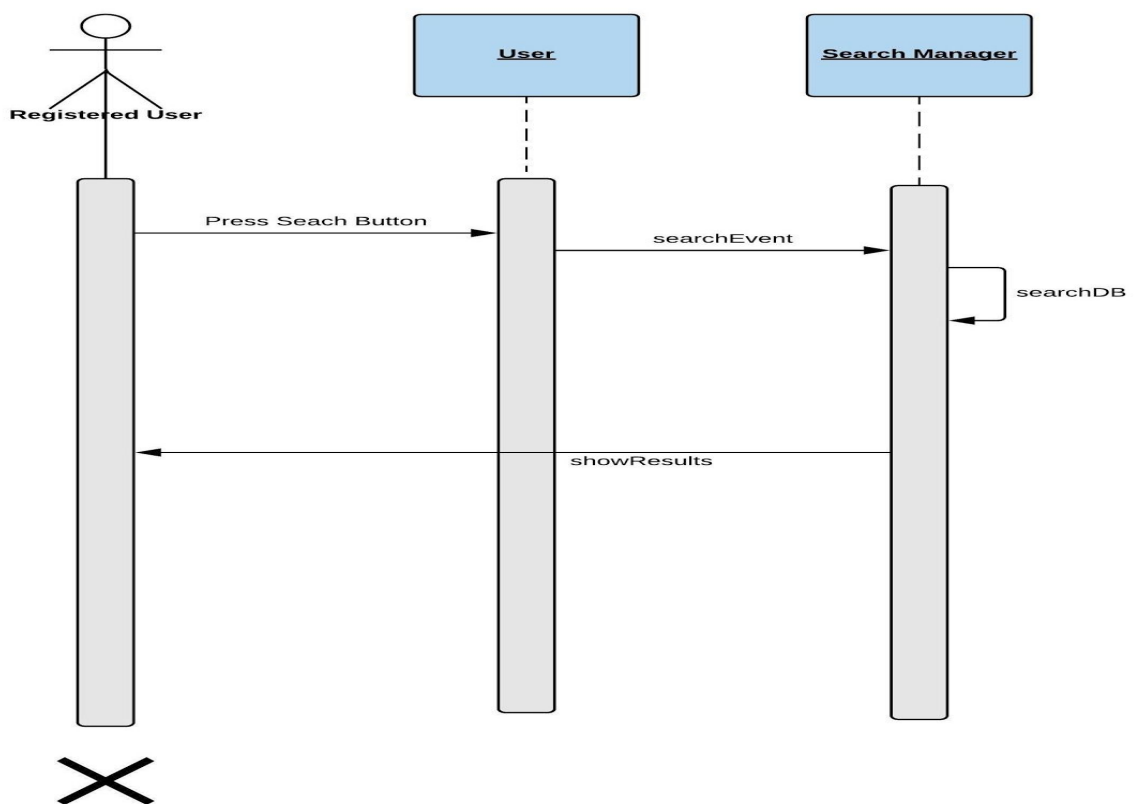
5. Comment on an Event



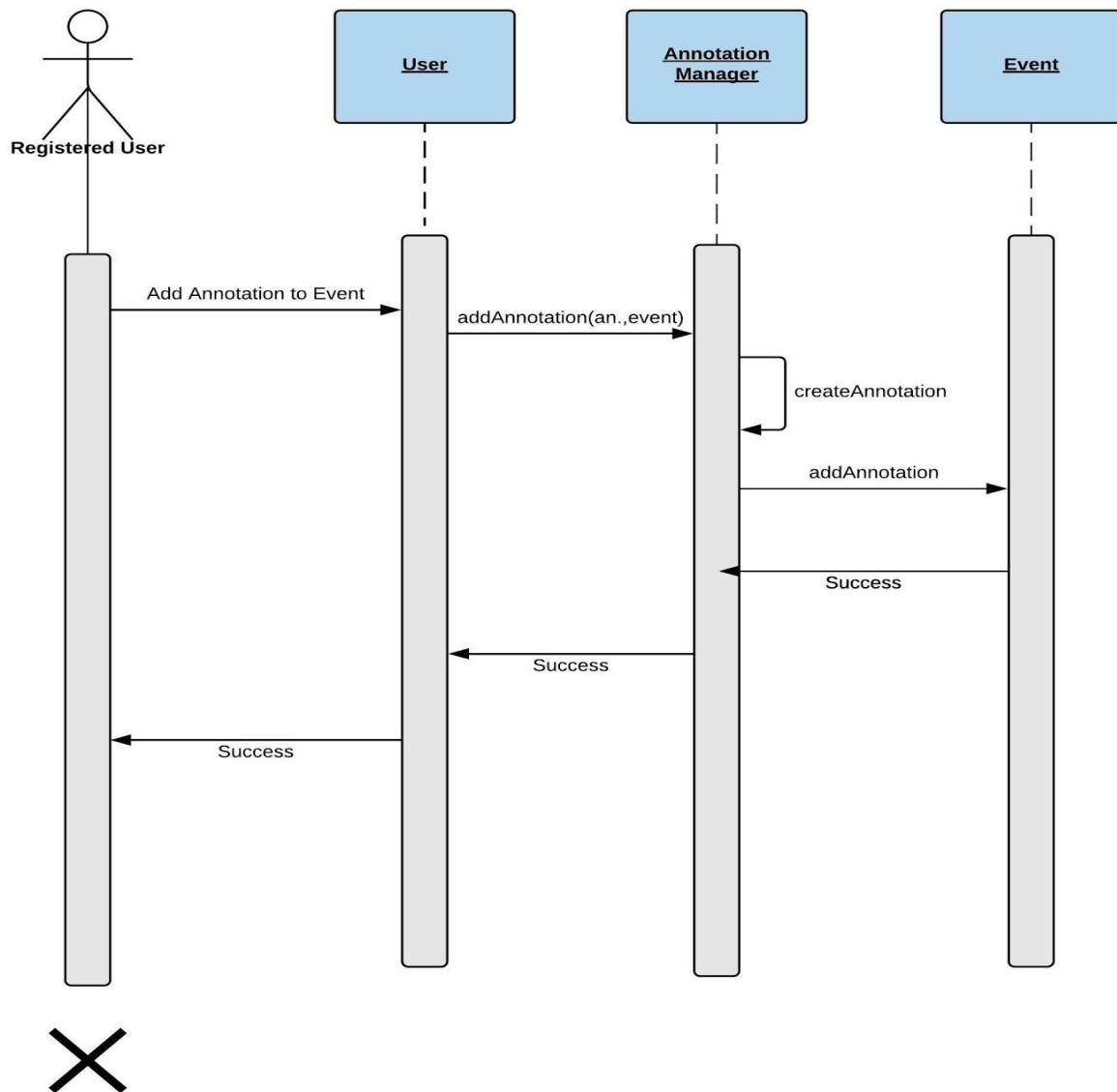
6. Following a User



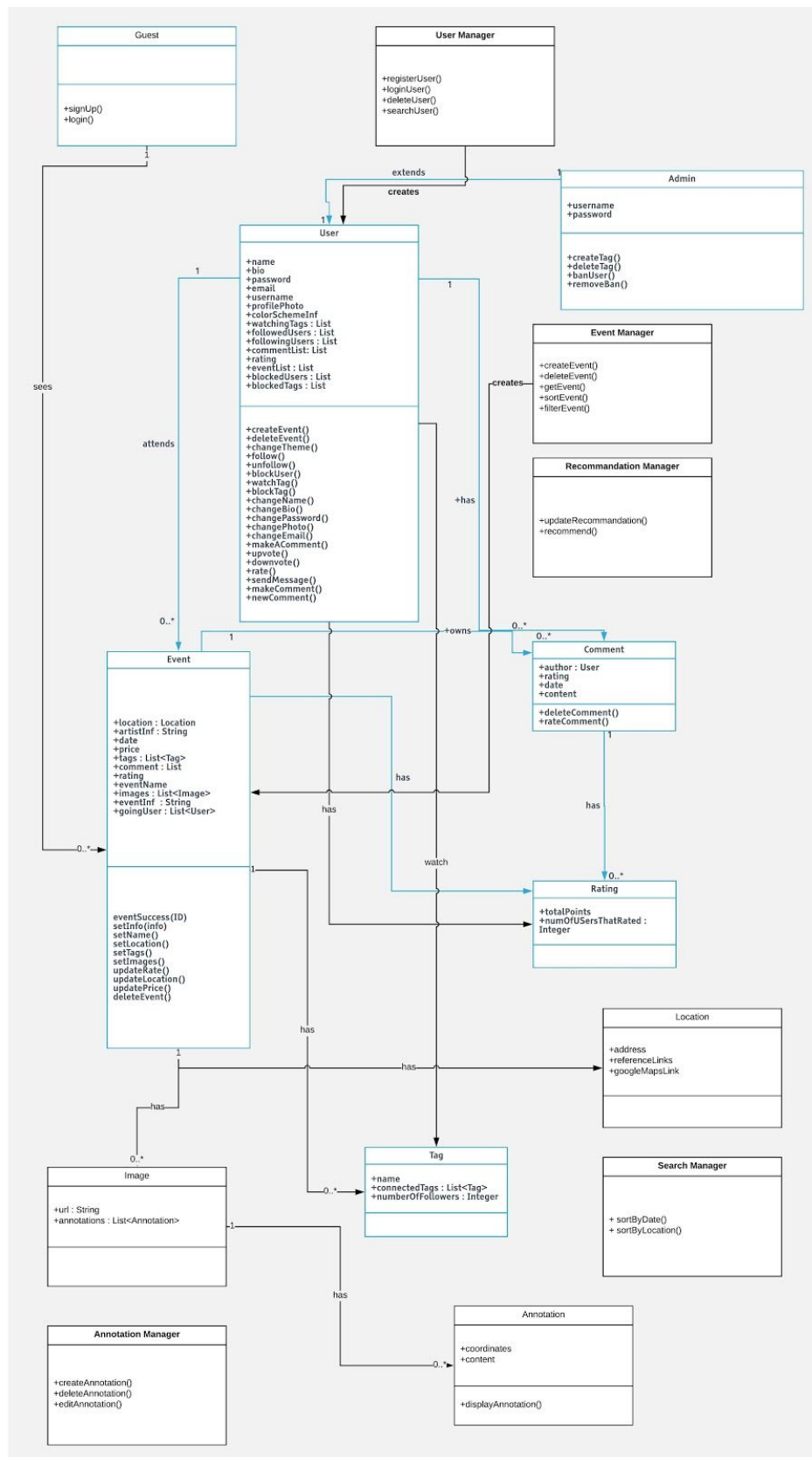
7. Search for an Event



8. Annotation



- Class Diagram



- Test Cases

Test Case 1**Title:** Editing Profile Page**Description:** To change profile page information**Related Requirement:** 1.2.1.2.1**Preconditions:**

- User must be logged in.
- User must be on his own profile page.

Steps:

Step	Test Condition	Test Data	Expected Result	Actual Result	Status(P/F)	Notes
1	User clicks edit field	Edit Button in Profile Page	Editing page is opened			
2	Change the fields on edit page (name field to "folks")	Submit button in edit profile page	Values on profile page is changed			

Post Condition: Profile page information is changed.**Test Case 2****Title:** Commenting on an Event**Description:** To add a comment to an event**Related Requirement:** 1.2.1.7.6**Preconditions:**

- User must be logged in.
- User must be on an event page that he will comment.

Steps:

Step	Test Condition	Test Data	Expected Result	Actual Result	Status(P/F)	Notes
1	User clicks type event box ("nice event")	Comment box in event page	Comment box is filled with comment text			
2	Clicks add comment button	Submit comment button in event page	A new comment is appearing below the event			

Post Condition: A new comment is added with the adder's name and adding time.

Test Case 3**Title:** Creating an Event**Description:** To add a new event to the app**Related Requirement:** 1.2.1.7.1**Preconditions:**

-User must be logged in.

Steps:

Step	Test Condition	Test Data	Expected Result	Actual Result	Status(P/F)	Notes
1	User clicks add event button on homepage	Add event button on homepage	Directing to create event page			
2	User fills the fields ("incesaz concert")	Input boxes on add event page	Seeing entered inputs			
3	User clicks submit button	Submit event button on create event page	The new event is added			

Post Condition: The new event is added and can be shown in homepage.**Test Case 4****Title:** Deleting an Event**Description:** To check if the only one who can delete the event is the creator.**Related Requirement:** 1.1.5.1**Preconditions:**

-User must be logged in.

-User should have an event to delete.

Steps:

Step	Test Condition	Test Data	Expected Result	Actual Result	Status(P/F)	Notes
1	User clicks delete event button on event page("incesaz concert")	Delete Event button on event page	The event is deleted and redirected to homepage			

Post Condition: Redirection to homepage without deleted event.**Test Case 5****Title:** Adding Photos to an Event**Description:** To check if the only one who can add new photos to the event is the creator.**Related Requirement:****Preconditions:**

-User must be logged in.

-User should have an event.

Steps:

Step	Test Condition	Test Data	Expected Result	Actual Result	Status(P/F)	Notes
1	User clicks choose a file field and marks the photos that s/he adds.	Upload Image	The images are selected			
2	User clicks add photo button on event page	Add Photo Button	The new photos are shown below the event			

Post Condition: The new photos are shown below the event.**Test Case 6****Title:** Registration**Description:** To check input validation and successful account creation.**Related Requirement:** 1.2.1.1.1.**Preconditions:**

-User should be a guest user.

Steps:

Step	Test Condition	Test Data	Expected Result	Actual Result	Status(P/F)	Notes
1	User enters invalid email (kdreyyan)	Email Text Box	Invalid Email Error			
2	User fills the required fields	Sign Up button	Confirmation mail is sent to email address			
3	User clicks the confirmation link	Confirmation Link	New user is added to database			

Post Condition: New user is registered.

Test Case 7**Title:** Login**Description:** To check valid email while login and successful login.**Related Requirement:** 1.2.1.1.3.**Preconditions:**

-User should be a guest user.

Steps:

Step	Test Condition	Test Data	Expected Result	Actual Result	Status(P/F)	Notes
1	User enters email without domain (mahmutkaraca)	Email Text Box	Red Border on Text box			
2	User enters valid email and password and clicks sign in button(**@**)	Sign In Button	Login to Robin and redirecting to homepage			

Post Condition: Login to Robin.**Test Case 8****Title:** Search Events & Users**Description:** To find event/user by searching relevant words in search bar.**Related Requirements:** 1.2.1.6.1.-2.-3.**Preconditions:**

-User must be logged in.

Steps:

Step	Test Condition	Test Data	Expected Result	Actual Result	Status(P/F)	Notes
1	User clicks search text field and types words relevant to the wanted user/event("art", "Turkish")	Search Bar	Written text is shown on search textbox			
2	User clicks search button on homepage after filling the search box	Search Button	The event/user that the current user wants comes to the page			

Post Condition: User views the searched user/event.

Test Case 9

Title: Adding Tags

Description: To tag event while a registered user is creating it.

Related Requirement: 1.2.1.5.1.

Preconditions:

-User must be logged in.

-User must be on event creation page.

Steps:

Step	Test Condition	Test Data	Expected Result	Actual Result	Status(P/F)	Notes
1	User enters non-existing tag("trekking")	Tag Creation	The non-existing tag is added to the tags			
2	User enters an existing tag("art")	Tag Addition to An Event	The event is created with tags			

Post Condition: The tag/new tag is added to cultural event item.

Test Case 10

Title: View Event Adder

Description: To check whether the link besides the created by field is working or not.

Related Requirement: 1.2.1.2.2.

Preconditions:

-User must be logged in.

-User must be on an event page.

Steps:

Step	Test Condition	Test Data	Expected Result	Actual Result	Status(P/F)	Notes
1	User clicks link named with event created besides created by field.	Created by	Opens creator user profile page			

Post Condition: Redirecting to event adder profile page.

Test Case 11

Title: Rating on an Event

Description: To check if rating component works.

Related Requirement: 1.2.1.7.5.

Preconditions:

-User must be logged in.

-User must view an event.

Steps:

Step	Test Condition	Test Data	Expected Result	Actual Result	Status(P/F)	Notes
1	User clicks the number of stars according to his/her rate	Star-Rating	New rating is seen above rating component.			

Post Condition: Rating of the event is changed according to the new rating.

Test Case 12

Title: Follow/Unfollow Another User

Description: To check follow & unfollow a user.

Related Requirement: 1.2.1.2.4.

Preconditions:

-User must be logged in.

-User must view another user's profile page

Steps:

Step	Test Condition	Test Data	Expected Result	Actual Result	Status(P/F)	Notes
1	User clicks follow button on the other user's profile page	Follow Button	The follow button is changed to following			
2	User clicks unfollow button on a followed user's profile page.	Unfollow Button	The following button is changed to follow			

Post Condition: The other user is followed/unfollowed by the current user.

Test Case 13

Title: Join/Mark As Interested Event

Description: To check whether an event can be marked as join or mark as interested or not.

Related Requirement: 1.2.1.7.2.

Preconditions:

- User must be logged in.
- User must be viewing a cultural event item.

Steps:

Step	Test Condition	Test Data	Expected Result	Actual Result	Status(P/F)	Notes
1	User clicks join button on "Contemporary Art Discussions" event.	Join Button	The user is added to attending users			
2	User clicks join button on "Rio Carnival" event.	Mark as Interested Button	The event is labeled as interested			

Post Condition: The event is labeled as interested/going.

Test Case 14

Title: Annotate an Event

Description: To check image annotation and annotation viewing.

Related Requirement: 1.1.4.3.

Preconditions:

- User must be logged in.
- User must be the creator of the event.

Steps:

Step	Test Condition	Test Data	Expected Result	Actual Result	Status(P/F)	Notes
1	User draws a rectangle above the photo that s/he wants to annotate below the event page.	Annotation Coordinates	The rectangle area is selected			
2	User enters annotation text	Annotate	The annotation is added			
3	User sees the annotation text when s/he drags the rectangle	Annotation	Annotated text appears			

Post Condition: The annotation is added to the photo.

5. User & system manuals

5.1. Frontend

5.1.1. System Manual

Requirements

Node.js

npm

Google Chrome

Setup

- 1- Clone the project at your local from project repository
- 2- Inside the cloned repository, change directory to app/frontend/robin by using the command "cd app/frontend/robin "
- 3- Run the command "npm install" in order to install the dependencies

Run Locally

- 1- Still inside the directory app/frontend/robin, run the command "npm start"

2- Open Google Chrome

3- Visit localhost:3000, in case it is not automatically opened

Build and Deploy

1. npm install

2. npm run build

3. If you wish to start an additional frontend server at 0.0.0.0:5000:

a. nohup serve -s build &

4. else you should use build files and serve them via another server.

5.1.2 User Manual

1- Open Google Chrome

2- Visit "<http://139.59.128.92:5000/>"

5.2. Backend

Requirements

Python3.6.7

Pipenv

Setup

1. Clone project to local

2. Install Python3

3. Install pipenv

4. Install requirements (\$ pipenv install --dev)

Run Locally

5. Synchronizes the database (\$ pipenv run python manage.py migrate)

6. Run server on the local machine (\$ pipenv run python manage.py runserver)

7. By default, the server runs on port 8000 on the IP address 127.0.0.1.

Build and Deploy

8. (\$ nohup pipenv run python manage.py runserver 0.0.0.0:8080&)

5.3. Android

5.3.1. System Manual

Requirements

Android 4.0.3+

User permissions

Setup

To setup the project:

- Download the given APK file to your phone
- Open the file to install the app

5.3.1. System Manual

Requirements

Android 4.0.3+

User permissions

Internet connection

Usage

Open the application regularly

6. Project Plan (simplified version in a report-friendly format)

Project Plan		
Task	Start Date	End Date
Researching technologies and environment setup	01/05/2018	04/10/2018
Learning technologies	02/10/2018	09/10/2018
Implementing homepage	10/10/2018	17/10/2018
Implementing signup and login	17/10/2018	24/10/2018
Milestone	24/10/2018	
Implementing event page,comment component	24/10/2018	7/11/2018
Implementing admin panel	7/11/2018	21/11/2018
Implementing searching, rating	21/11/2018	04/12/2018
Milestone	04/12/2018	
Recommendation and tags	20/12/2018	02/01/2019
Implementing W3 annotations	20/12/2018	02/01/2019
Design and implementation improvements	20/12/2018	02/01/2019
Milestone - Project Finalized	02/01/2019	
Testing	25/12/2018	02/01/2019
Debugging	25/12/2018	02/01/2019
Presentation	02/01/2019	