Executive Summary

Our work for this semester was divided into 2 levels:

1- Requirements and Design Issues(according to Requirements)

2- Creating API

1. Requirements and Design Issues

We started working by spending a few weeks on determination of our requirements to meet the demands of our customer in a reasonable timeframe. In time; of course, there have been some changes. However, we roughly completed the analysis and synthesis of the project requirements. In the light of the requirements, we created a project plan and divided the tasks into reasonable parts and assigned to the team members. Then, to interpret the project, we designed mock-ups for Android and Web applications of the project and tested them on some scenarios according to requirements and aimed to see how application could work.

One step ahead, to describe responsibilities of the system and to analyze and design the static view of an application, we worked on class diagrams. After that, in order to show the interactions between objects, users, servers and functions in the sequential order, we drew sequence diagrams.

Lastly; in the scope of requirements and design issues, we wanted to think in users' perspective. Thus, to find where our customers may get stuck and to get a fresh and unbiased perspective, we created user tests according to some assumptions and analyzed them as some consequences.

Until here, first milestone was reached on 2 April, 2018. (as planned)

2. Creating API

This semester, there were not any coding tasks for the project. Yet; next semester, because we will start with API part of the project, we worked on a simple Twitter API project according to some functionalities in order to practise and understand why we need an API and how it works.

Until here, second milestones was reached on 10 May, 2018. (as planned)

List and Status of Deliverables

1 - Project Plan

Deliverables	**Status**
Requirements Analysis	Done
Project Task - Timeline Plan	Done
User Stories	Done
Android & Web mockups	Done
User Scenarios	Done
Class Diagram	Done
Sequence Diagrams	Done
Use Cases	Done
Acceptance Tests	Done

2 - Android & Web

We haven't started the actual implementation of the project, but the list below contains major components of our Android and web applications.

Deliverables	**Status**
User Sign Up and Login	Mockup completed
Home Page	Mockup completed
User Profile	Mockup completed
User Settings	Mockup completed
Search Functionality	Mockup completed
Event Page	Mockup completed
Event Creation and Editing	Mockup completed
Private Messaging	Mockup completed
Guest Handling	Mockup completed

3 - Research & Practice Application

We did research about how we can use git effectively throughout the development process.

Also, we developed a sample Django web application which uses Twitter API to do few different things on Twitter.

Evaluation of the status of deliverables

For the deliverables of project plan, almost everything went as expected on schedule. We generally did two iterations for each deliverable, firstly our initial iteration and secondly after we got feedback. We made brain storms for every deliverable in our regular meetings. Also, we changed team leader every week to make better participation on our project for all of us. Team leaders managed and checked the evaluation of deliverables.

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.

A Summary of Coding Work

Group Member	Team	Contribution
Abdurrahman Dilmaç	Back-end	Had the knowledge of Django. Implemented front-end templates of API Homework. Implemented the main structure of API homework in back-end side. Implemented a main page of API homework in front-end side. Solved the conflicts in Github.
Mert Aközcan	Back-end	Had the knowledge of Rest-API. Started to learn Django. Implemented front-end pages of API homework. Implemented a API call in back-end side. Integrated the front-end with back-end of API homework.
Zeynep Işık	Android	Had the knowledge of Android. Started to research how to implement functionalities of project in Android. Implemented a API call in back-end side. Started to set up register functionalities.
Ömer Kırbıyık	Android	Started to learn Android. Created an Amazon EC2 instance and run.Updated server and setup tools like Django, virtual environment, and python to deploy back-end. Run Django application on the server. Started to set up admin page of application.
Atıf Emre Yüksel	Android	Started to learn Android. Started to research how to implement functionalities of project in Android. Implemented a API call in back-end side. Started to set up user login page.

Hilal Mente	Front-end	Started to learn React framework. Had the knowledge of HTML, CSS. Designed web mock-ups. Implemented a API call in back-end side.
Erkam Ağralı	Front-end	Started to learn React framework. Had the knowledge of HTML, CSS. Designed web mock-ups.
Fatih Maytalman	Front-end	Started to learn React framework. Had the knowledge of HTML, CSS. Designed Android mock-ups. Implemented a API call in back-end side.

Requirements

Glossary

Admin: A person who is responsible for the sustainability of the system.

Annotation: Explanatory content attached to an event compatible with W3C standards.

Comment: The response of users about activities of the other users or their profile, which can be voted by the others.

Community: A group of people sharing the same or similar interests and having communication or/and collaboration among themselves.

Database: An organised collection of data to store users' data safely.

Event: A cultural meeting that has a certain place and a specific time interval fields such as theater, cinema demonstration, opera, concert, book discussion, etc.

Event history: History of events which the user attends in the past, which is shown in his/her profile.

Follow: Users can follow the events attended by the specific user and his/her profile by this feature.

Google Maps API: Data gives you exact real-time information for locations, mapping, and navigation.

Guest: A person who is not registered and not having user privileges but viewing public pages.

Item: An article, an image or basically anything a user shares on the platform.

Password: Minimum 8 maximum 20 characters of secret sequence used to identify a user and grant the user with member user abilities.

Profile: Profile: A number of specific information about a user containing his/her interests and events s/he attends.

Recommendation: Suggesting a event liked by a user to other users or recommending users to follow.

Sign-up: When a new user registers by giving intended personal information by system.

System: The whole environment of Cultidate project project including Web, Mobile, server-side, databases, cold storages.

Tag: A descriptive word or phrase to define an item defined and can be searched better.

User: A person who has an account on the community.

Username: User specific unique characters to identify a user.

Vote: Rating system allowing users to vote and comment the other users or events positively or negatively.

1. Functional Requirements

1.1. User Requirements

1.1.1 Member User Requirements

- **1.1.1.1 Sign-Up**
    - 1.1.1.1.1 Users shall sign up by following one of the following procedures.
        - 1.1.1.1.1.1 Users shall provide a name, surname, username, password, e-mail address.
            - 1.1.1.1.1.1.1 Users shall verify their e-mail addresses by following the steps in the confirmation mail sent by the system.
            - 1.1.1.1.1.1.2 User emails and usernames shall be unique.
            - 1.1.1.1.1.1.3 Passwords shall be no shorter than 8 characters and no longer than 20 characters.
        - 1.1.1.1.1.2 Users shall sign up using their Facebook accounts.
    - **1.1.1.1.2 Account Verification**

        - 1.1.1.1.2.1 Users who wish to verify that their account belongs to an existing person with a corresponding profile information should be able to do so by following the procedure determined by the system.
- **1.1.1.2 Login**
    - 1.1.1.2.1 Users shall login using their username and password.
    - 1.1.1.2.2 Users shall login using their Facebook account.
    - **1.1.1.2.3 Resetting Password**
        - 1.1.1.2.3.1 In case of a user forget the account password, a password reset link is to be sent to the user's email address.
        - 1.1.1.2.3.2 A user who gets a reset link should be able to reset the password by going the unique link provided in the email.
- **1.1.1.3 Profile**
    - 1.1.1.3.1 Users shall have profile which includes but not limited to their information bio, interests.
    - 1.1.1.3.2 Users shall be able to edit their profiles.
- **1.1.1.4 Contribution**
    - 1.1.1.4.1 Users shall be able to create items such as events and add photos and edit performing artist, location, date, price information.
    - 1.1.1.4.2 Users shall be able to follow events.
    - 1.1.1.4.3 Users shall be able to share events.
    - 1.1.1.4.4 Users shall be able to comment on events and other users' profile.
    - 1.1.1.4.5 Users shall be able to mark their attendance status to the events.
- **1.1.1.5 Messaging**
    - 1.1.1.5.1 Users shall be able to send private messages to other users.
- **1.1.1.6 Tagging**
    - 1.1.1.6.1 Users shall be able to tag their items such as creating a tag or using an existing one.
- **1.1.1.7 Searching**
    - 1.1.1.7.1 Users should be able to search users and events semantically by means of the following services.
        - 1.1.1.7.1.1 They shall be able to make location-based filtering.
        - 1.1.1.7.1.2 They shall be able to make tags filtering.
- **1.1.1.8 Notifications**
    - 1.1.1.8.1 Users should be able to get notifications such as friends requests, event invites, private messages from other users, event changes.
- **1.1.1.9 Blocking**
    - 1.1.1.9.1 Users should be able to block certain users.

    - 1.1.1.9.2 A blocked user by another user shall not be able to see each other's profile and items.

1.1.2 Guest User Requirements

- 1.1.2.1 Guest users shall be able to only view events and actions done by members.
- 1.1.2.2 Guest users shall not be able to take any other action.

1.2. System Requirements

- ### 1.2.1 Content
    - 1.2.1.1 Users shall have freedom to create an item and an event.
    - 1.2.1.2 Users shall be able to delete their items and events.
    - 1.2.1.3 Users shall be able to edit their items and events.
    - 1.2.1.4 The system shall provide an enough information about artist, event date, and event price.
- ### 1.2.2 Recommendations
    - 1.2.2.1 Users shall be able to get recommendations for events based on their event histories.
- ### 1.2.3 Interface
    - 1.2.3.1 Project interface language shall be English.
    - 1.2.3.2 It shall be able to accept the inputs consisting of latin characters which are in ASCII table. The characters which do not exist in ASCII table shall not be accepted.
- ### 1.2.4 Annotations
    - 1.2.4.1 Users shall be able to annotate the event information and multimedia that may have been added to an event page with explanations, questions, and answers.
    - 1.2.4.2 System shall allow annotation including geographical references by using Google Map API.
    - 1.2.4.3 System shall allow text formatted annotation.
    - 1.2.4.4 System shall allow image formatted annotation.
    - 1.2.4.5 System shall allow links which may be semantic resources to provide more detailed information about the event.
- ### 1.2.5 Search

        - 1.2.5.1 System shall be able to perform semantic searches as specified by the users.
        - 1.2.5.2 The application should allow semantic search to be used.
        - 1.2.5.3 System shall protect and hide users' information among the users if one of them blocks the other.
        - 1.2.5.4 System should be able to show number of followers of a user while searching.
        - 1.2.5.5 System should be able to use the given location when user searches events in platform.
- **1.2.6 Database**
    - 1.2.6.1 System shall store data in the database.
    - 1.2.6.2 System shall update data in the database.
    - 1.2.6.3 System should make integrity check once in two weeks to prevent data loss.
- **1.2.7 Voting**
    - 1.2.7.1 System shall be able to store voting data coming from users.
    - 1.2.7.2 System shall be able to retrieve voting data coming from users.

2. Non-Functional Requirements

2.1 Security and Reliability

- 2.1.1 Data of users shall be protected and used according to LAW ON THE PROTECTION OF PERSONAL DATA (Kişisel Verilerin Korunması Kanunu)
- 2.1.2 SSL certificate shall be provided on all pages.
- 2.1.3 Users' email addresses shall not be reachable by other users or the public.
- 2.1.4 Users' passwords shall be stored with encryption in the database.
- 2.1.5 Users shall get notification emails when they change their password.
- 2.1.5 Captcha method shall be enabled while registering to prevent malicious bots.
- 2.1.6 System shall be protected against DDoS.
- 2.1.7 Weekly periodic backups shall be taken to ensure data is safe and sound.
- 2.1.8 In case of server failure and in the need, the system shall be restored with the latest backup.

2.2 Performance

- 2.2.1 The system should be able to respond to requests within 4 seconds.
- 2.2.2 100 requests per seconds should be responded.
- 2.2.3 The system shall be able to serve at least 20000 members.

2.3 Availability

- 2.3.1 Project shall be available on both Android and Web platform.
- 2.3.2 Compatibility with Google Chrome, Opera(Chromium based browsers), Firefox, Safari and Edge should be ensured.
- 2.3.3 The system should be guaranteed to be accessible always except for maintenance times.
- 2.3.4 System should be available 95% of the time.
- 2.3.5 In the case of a failure, system should recover in at most 1 hour.

2.4 Database

- 2.4.1 Data shall be held in a secure database.
- 2.4.2 Database hierarchy shall be constructed in such a way that efficiency and privacy of user data is protected.
- 2.4.3 Nothing shall be deleted from database.
- 2.4.4 All changes' logs shall be kept in database.

2.5 Annotations

- 2.5.1 The W3C Web Annotation Data Model shall be used for annotation.
- 2.5.2 The annotations shall be stored and retrieved based on Linked Data Platform.
- 2.5.3 JSON-LD format shall be used for annotation representation according to W3 documentation.
- 2.5.4 The annotations shall be tested by test team to ensure that they work correctly.
- 2.5.5 Suggestion of annotation should be allowed urls while creating an annotation.
- 2.5.6 Retrieving annotations shall be allowed for all users.
- 2.5.7 Creating annotations shall be allowed for only registered users.

- 2.5.8 Updating and deleting annotations shall be allowed only if the online user is the owner of the annotation.
- 2.5.9 Annotations can be flagged as spam (too much report can cause annotation to be removed).

2.6 Accessibility

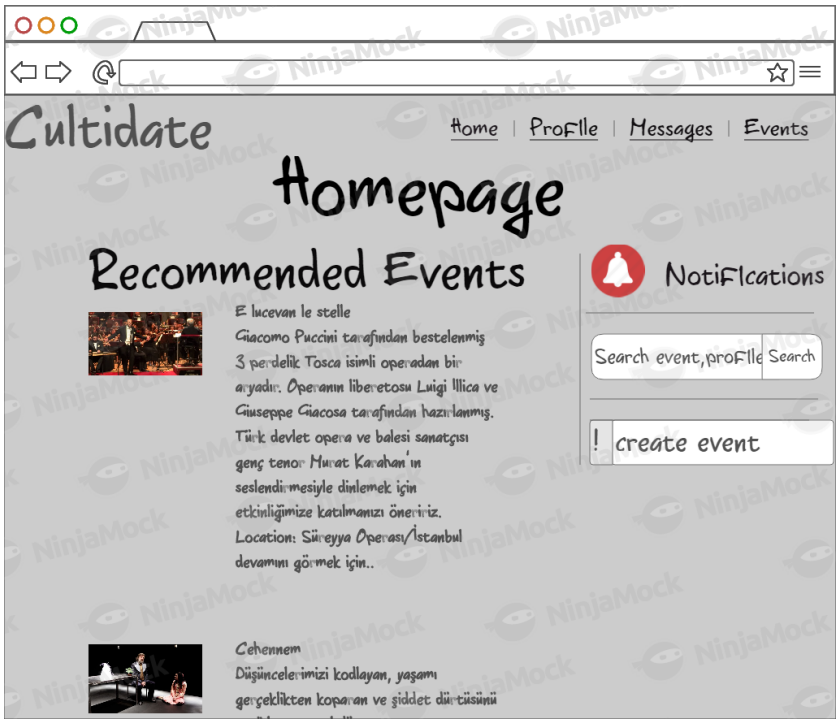
- 2.6.1 The system will be accessible by native Turkish and English users.
- 2.6.2 The system shall be accessible on Android version 6.0 and higher and web platforms.
- 2.6.3 The web platform shall be responsive to screen sizes with various sizes.
- 2.6.4 The system shall have different kinds of themes such as night, day and color-blindness themes.

Mockups

Sign-up

Login

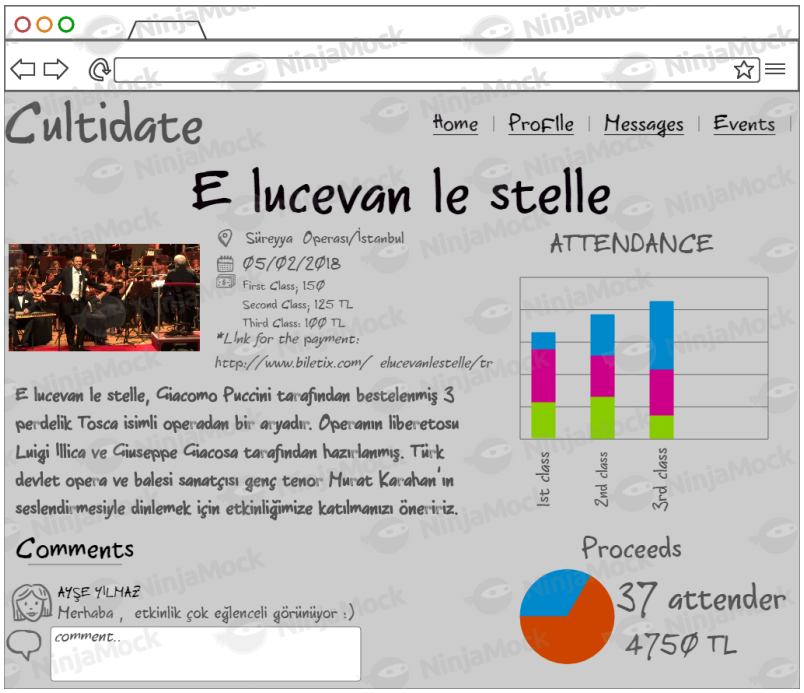
Homepage



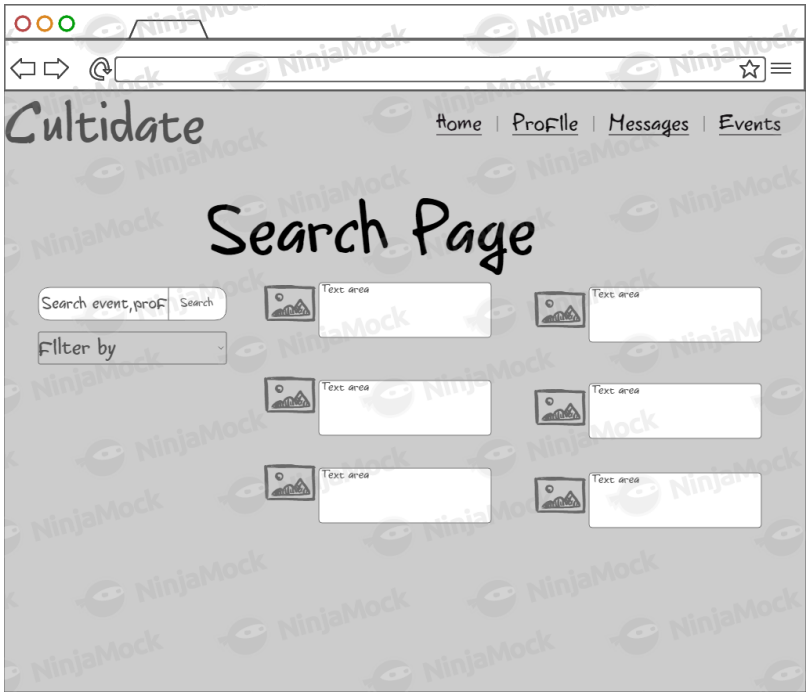
User Profile

Settings

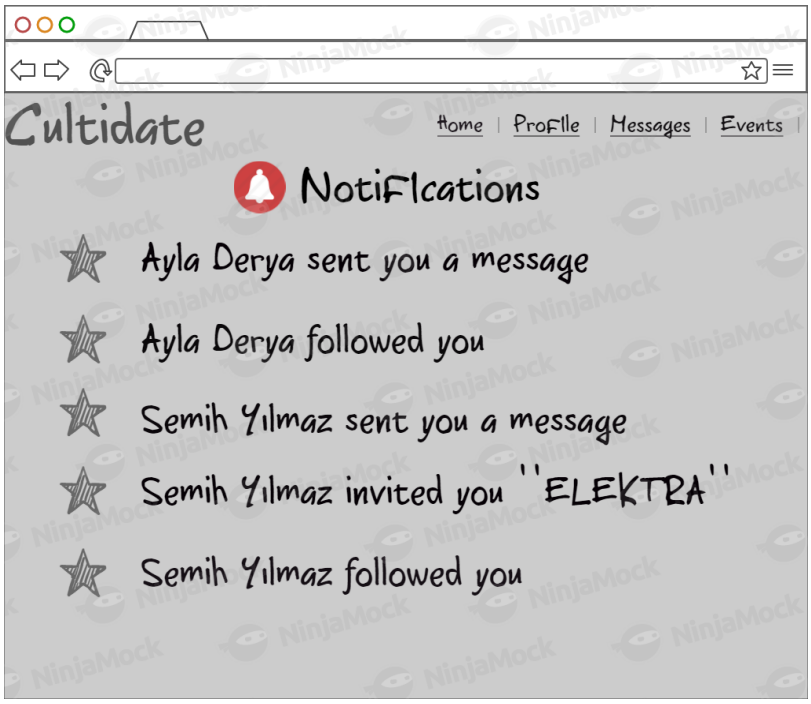
Event Page



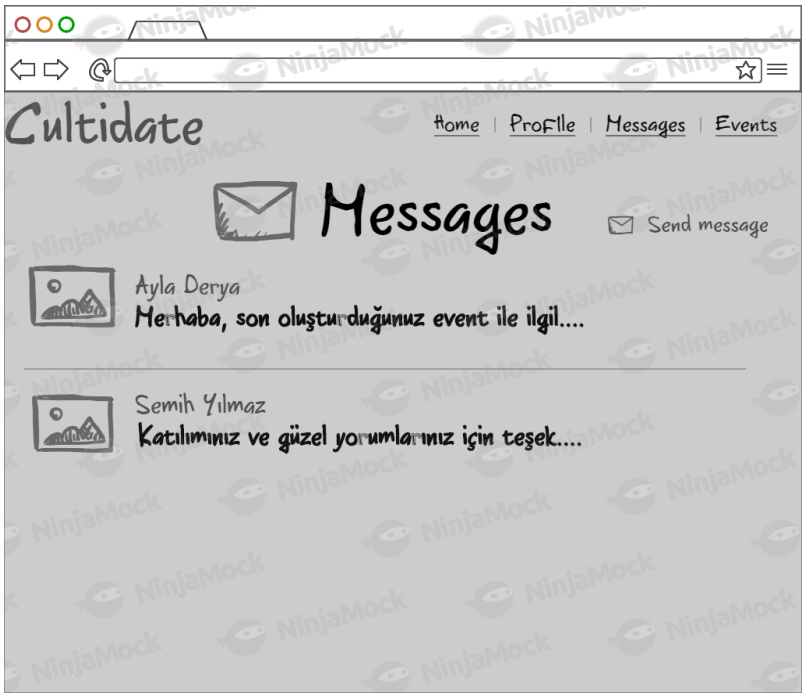
Search Page



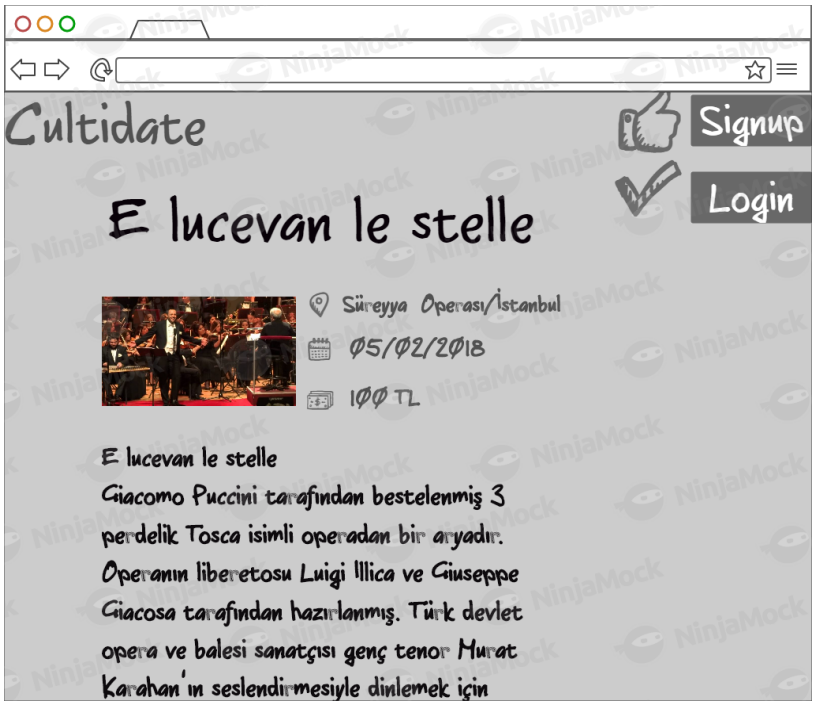
Notifications



Messages



Guest User



Create Event

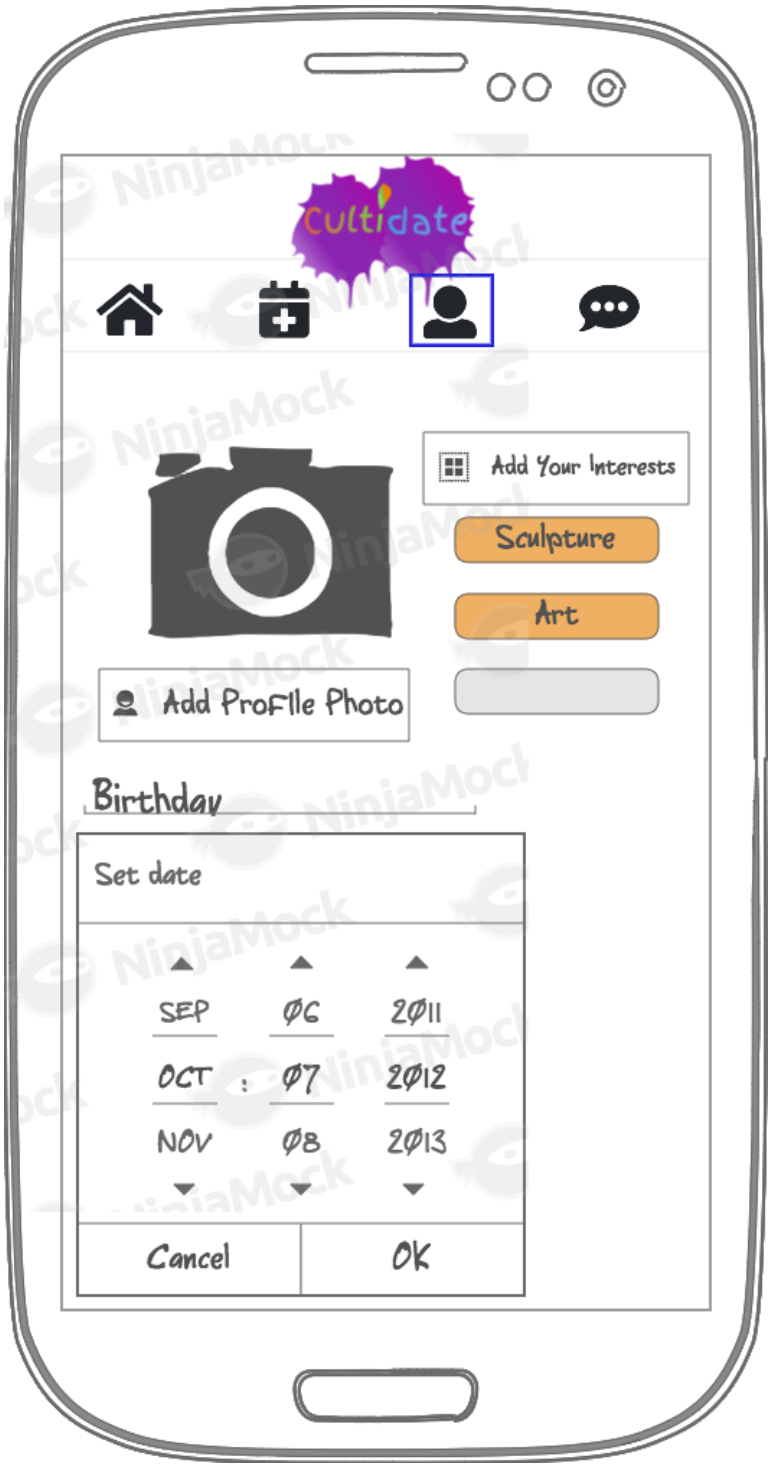
Android Device Mock-up

Signup

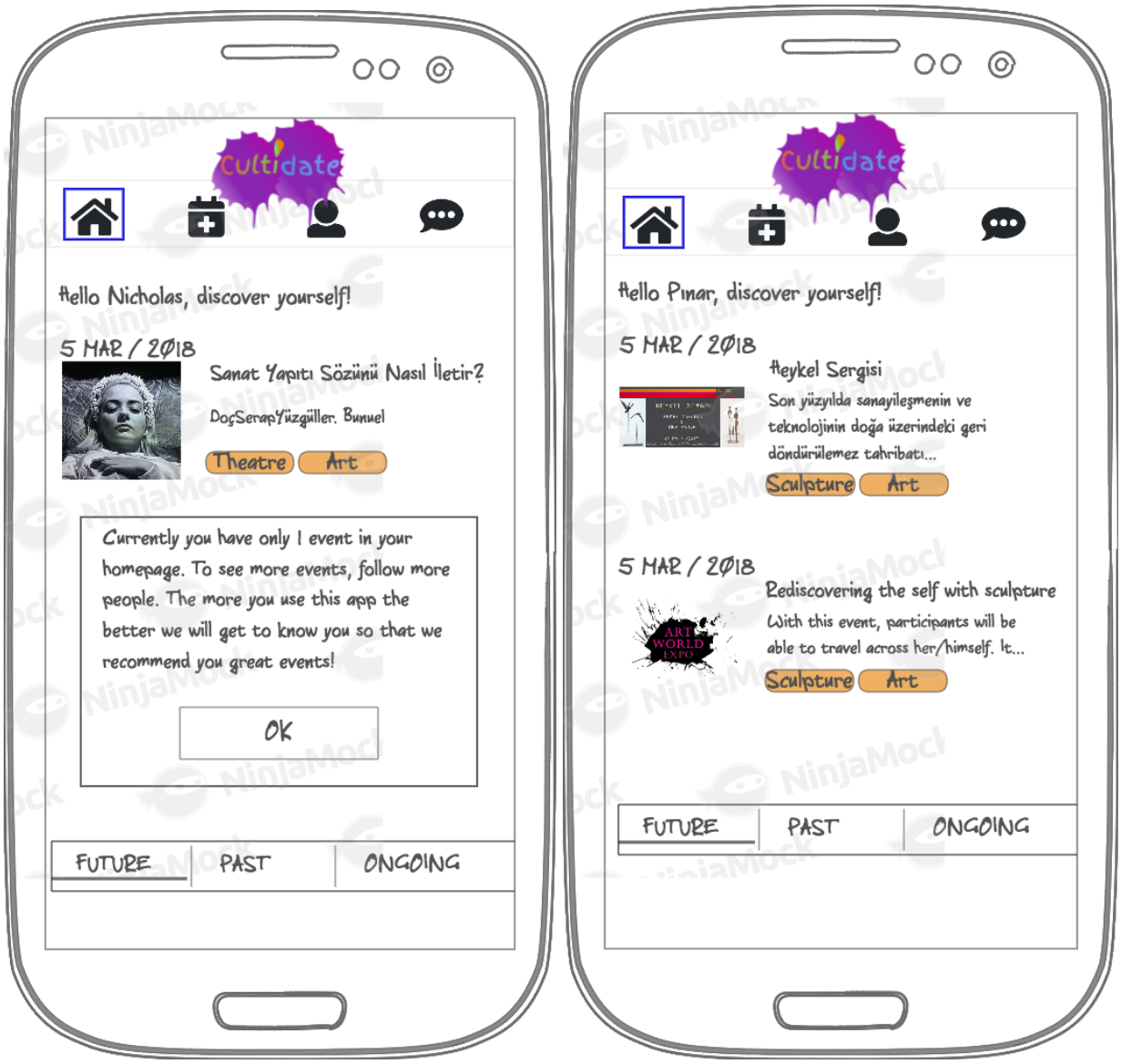
Login


Cultidate
Login
email
password
Signup
Login
Log in with Facebook

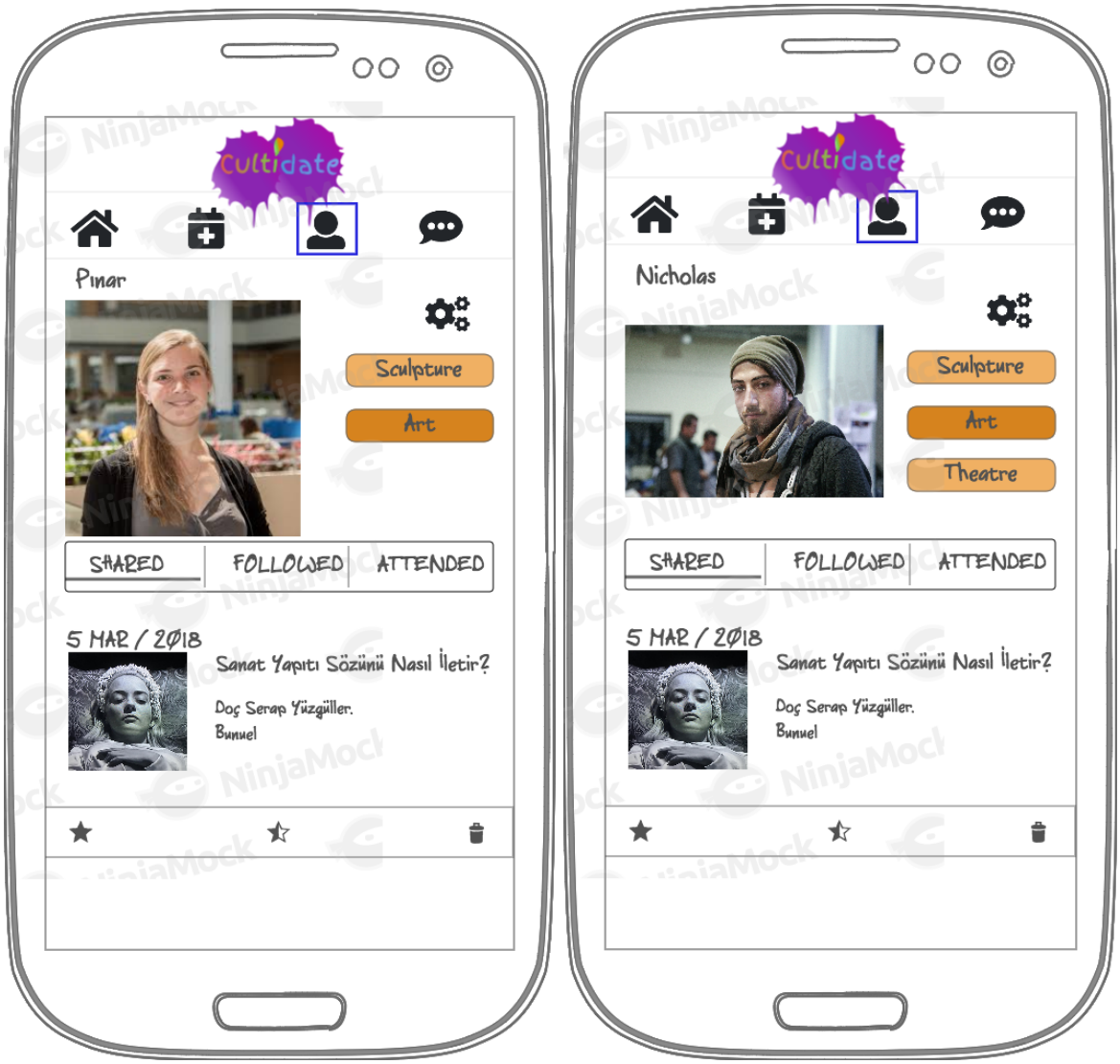
Profile Create - Edit



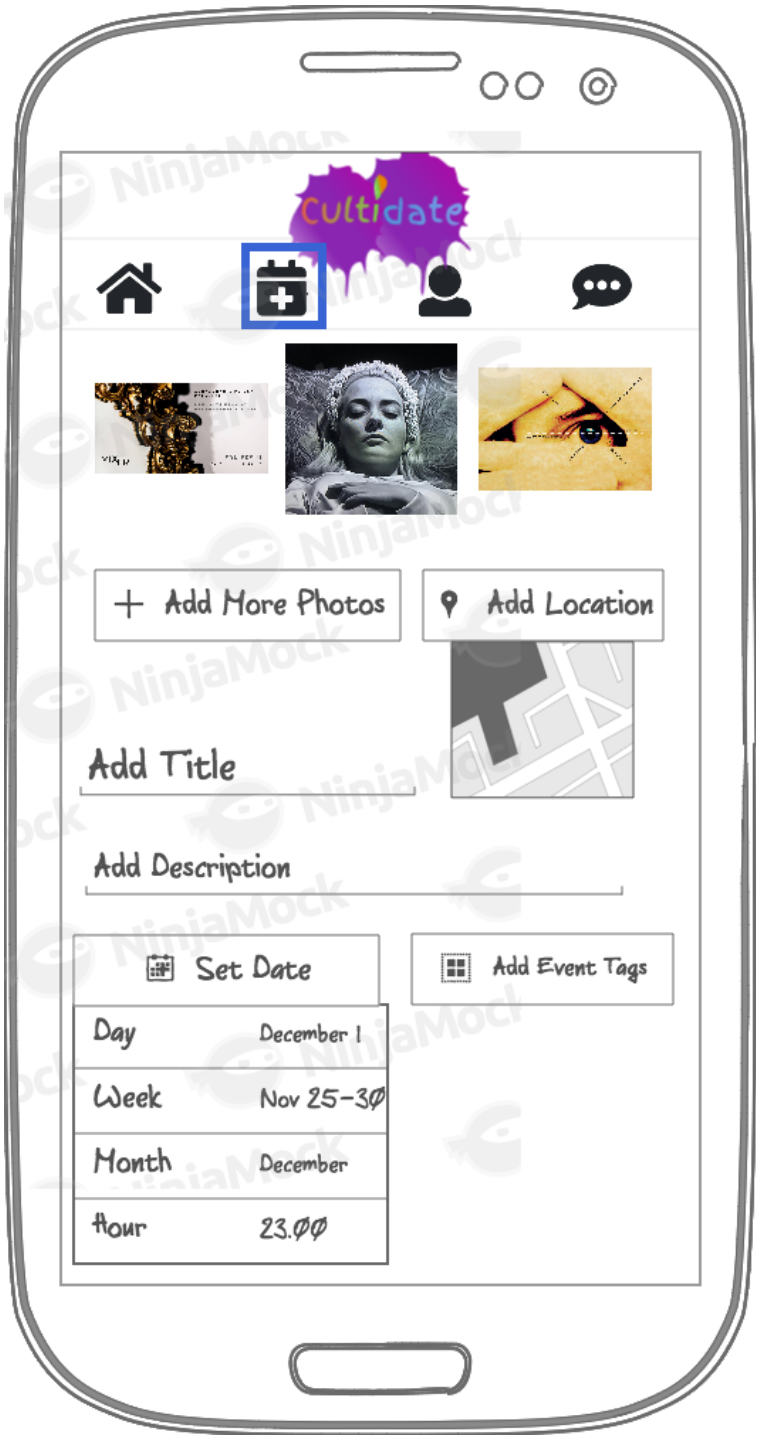
Home Page


cultidate
Hello Nicholas, discover yourself!
5 MAR / 2018
Sanat Yapıtı Sözünü Nasıl İletir?
DoçSerapYüzgüller. Bunuel
Theatre
Art
Currently you have only 1 event in your homepage. To see more events, follow more people. The more you use this app the better we will get to know you so that we recommend you great events!
OK
FUTURE
PAST
ONGOING
cultidate
Hello Pınar, discover yourself!
5 MAR / 2018
Heykel Sergisi
Son yüzyılda sanayileşmenin ve teknolojinin doğa üzerindeki geri döndürülemez tahribatı...
Sculpture
Art
5 MAR / 2018
Rediscovering the self with sculpture
With this event, participants will be able to travel across her/himself. It...
ART WORLD EXPO
Sculpture
Art
FUTURE
PAST
ONGOING

Profile


cultidate
Pınar
Sculpture
Art
SHARED
FOLLOWED
ATTENDED
5 MAR / 2018
Sanat Yapıtı Sözünü Nasıl İletir?
Doç Serap Yüzgüller.
Bunuel
cultidate
Nicholas
Sculpture
Art
Theatre
SHARED
FOLLOWED
ATTENDED
5 MAR / 2018
Sanat Yapıtı Sözünü Nasıl İletir?
Doç Serap Yüzgüller.
Bunuel

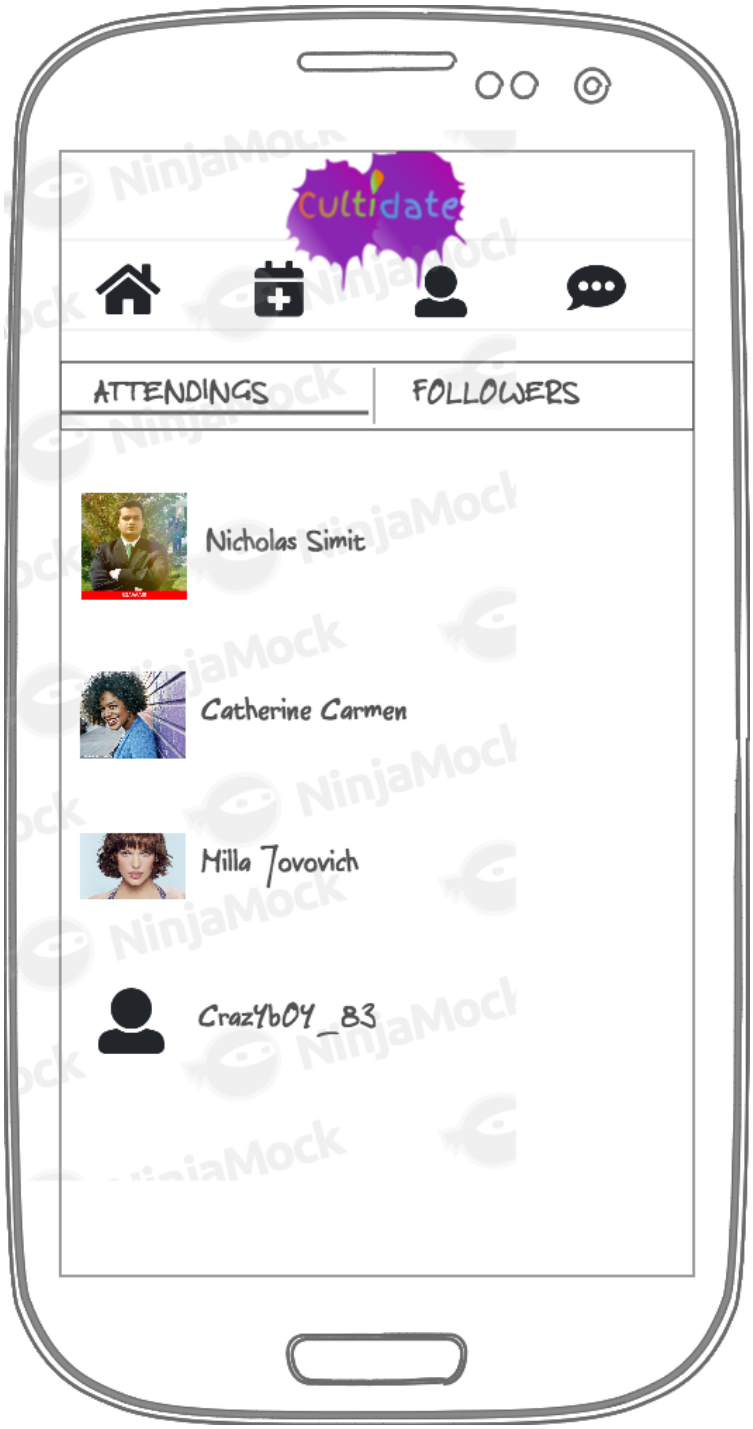
Create Event



Event


Cultidate
26 JULY
HEYKEL SERGİSİ
PEMBE TÜZÜNER
&
TİNA VARON
GALERİ ALAÇATI
Sculpture
Art
23 MAR,2018 19.45
JOIN!
FOLLOW
SHARE
+ 97 Attending
Description
Son yüzyılda sanayileşmenin ve teknolojinin doğa üzerindeki geri döndürülemez tahribatı ve bu tahribata insanın tepkisiz, umursamaz tavrı, çarpık kentleşme, doğal hayatın yok oluşu ve bütün bu hayata dair olumsuzlukların yine insan üzerindeki etkileri, heykeltraş Nicholas'ın eserlerinin üretim sürecinde işlediği temalar.

Attendings - Followings

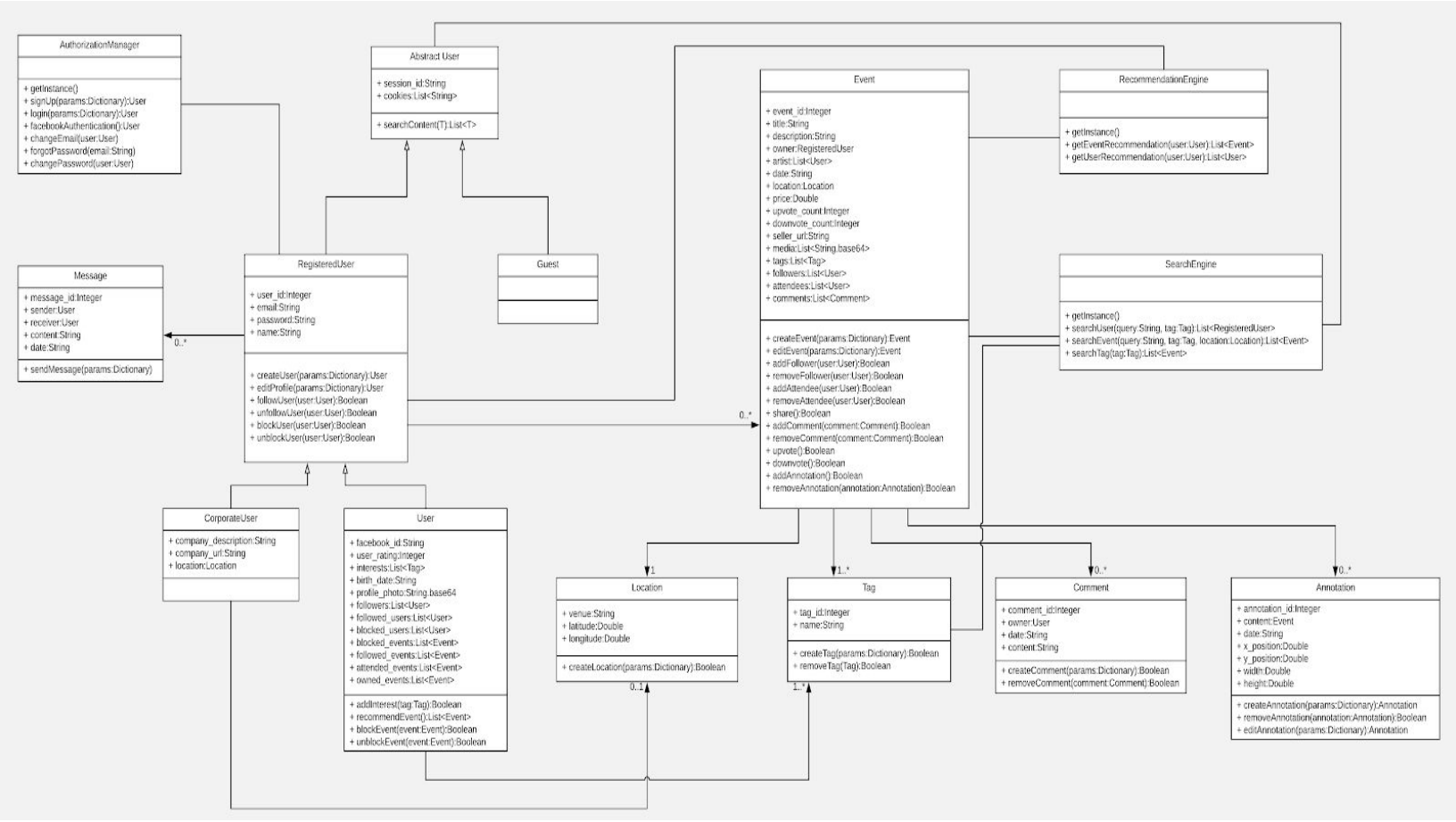


Message List - Message


cultidate
Tamer
Happy Birthday to my Niko!
Catherine
Your event was awesome!

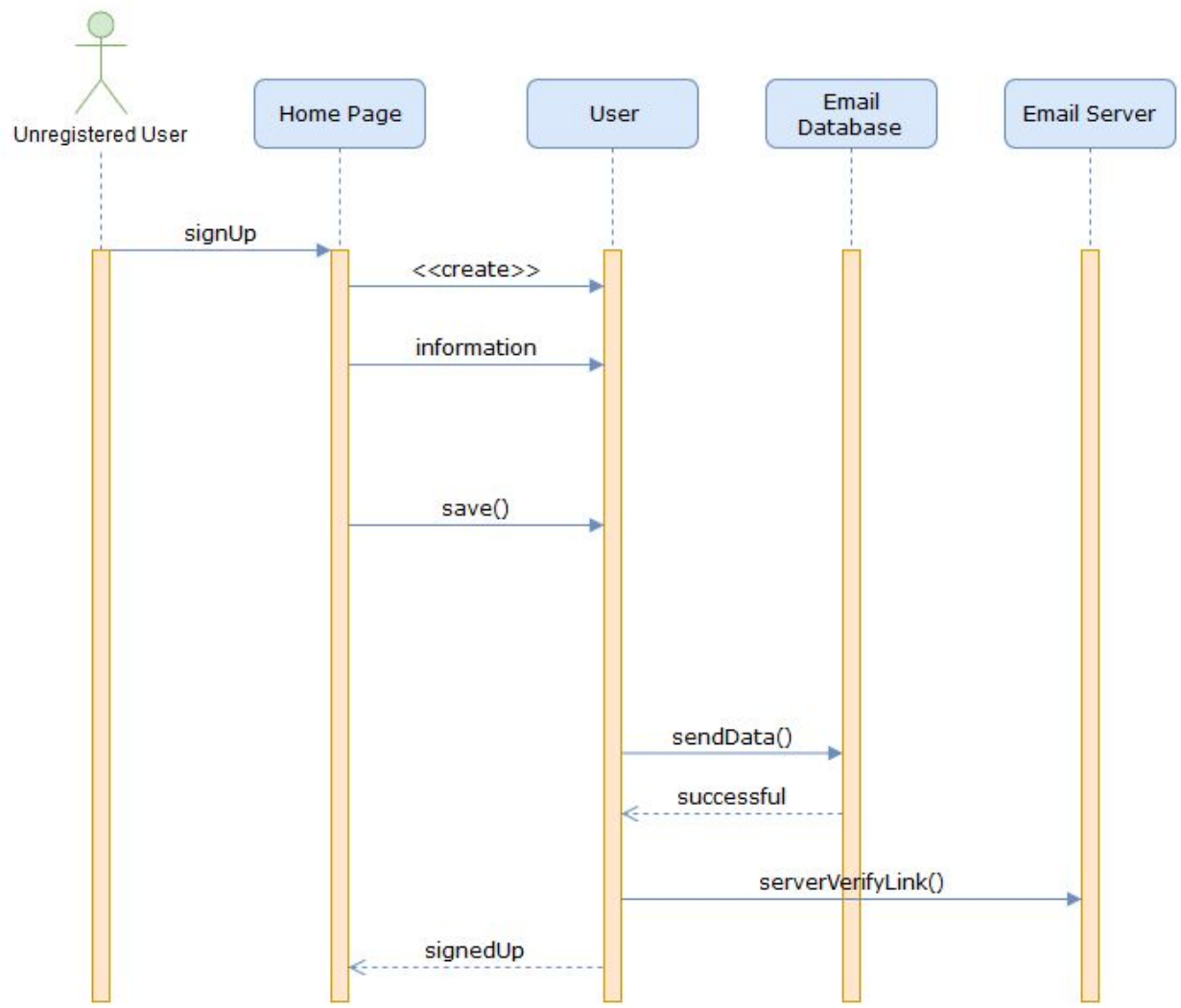

Cultidate
Nicholas
Hmm. Why not? This app is great!
Catherine
Thanks for coming, Pınar!

Software Design

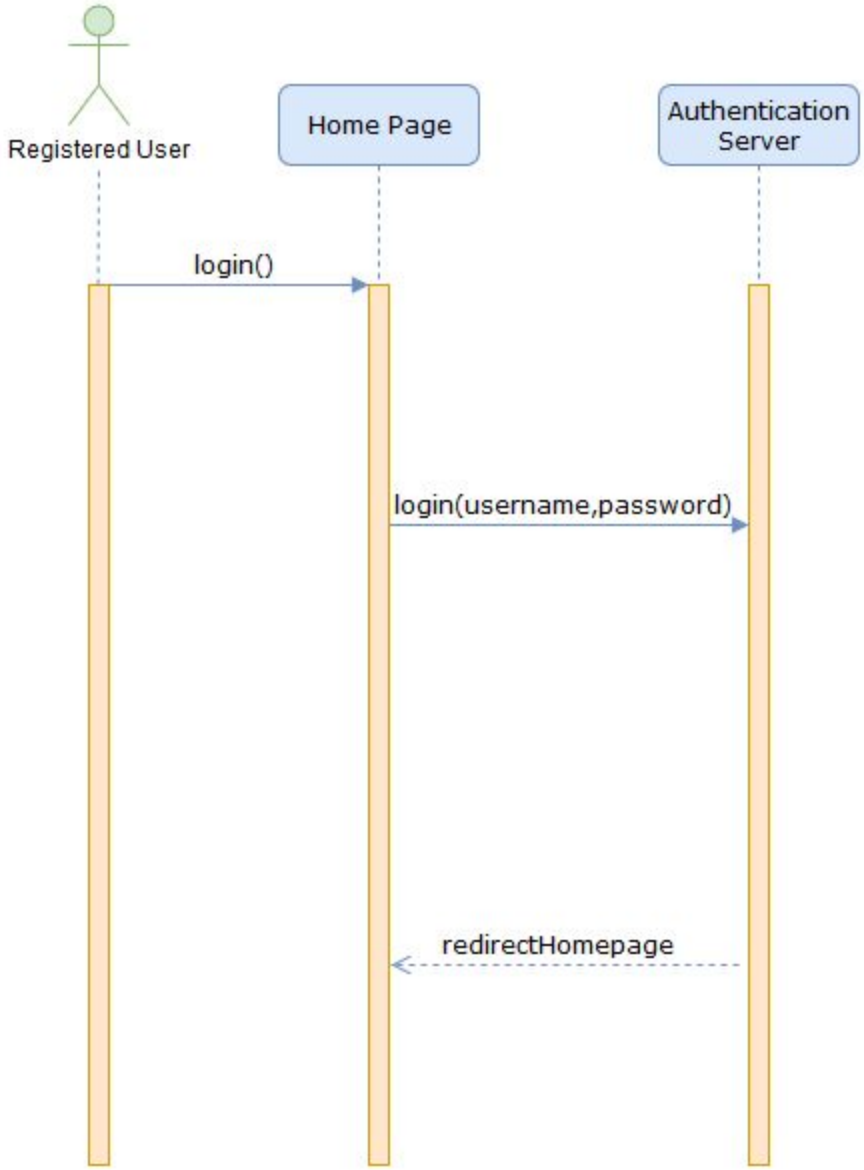


(For the high resolution image, please visit: https://raw.githubusercontent.com/bounswe/bounswe2018group6/master/wiki_assets/class_diagram.png)

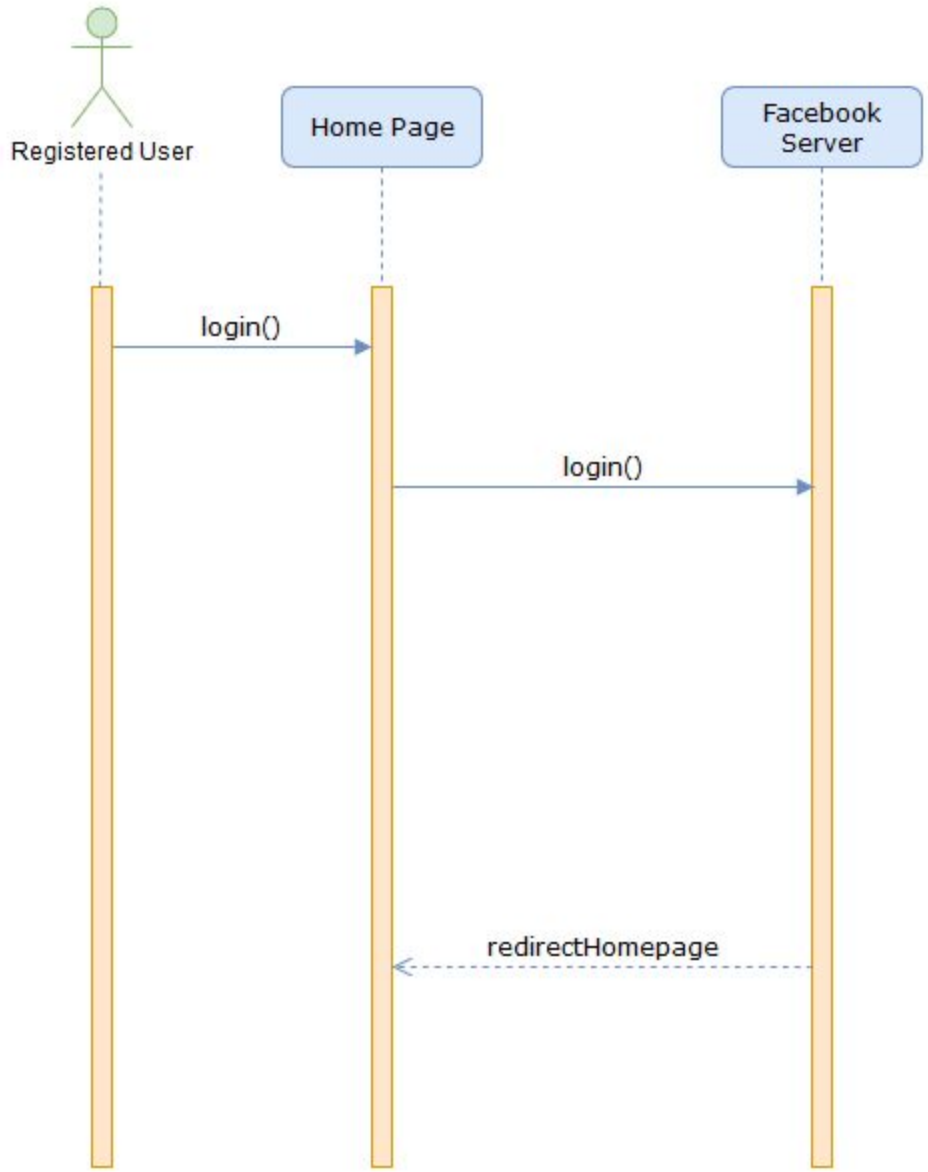
Sign Up


Unregistered User
Home Page
User
Email Database
Email Server
signUp
<<create>>
information
save()
sendData()
successful
serverVerifyLink()
signedUp

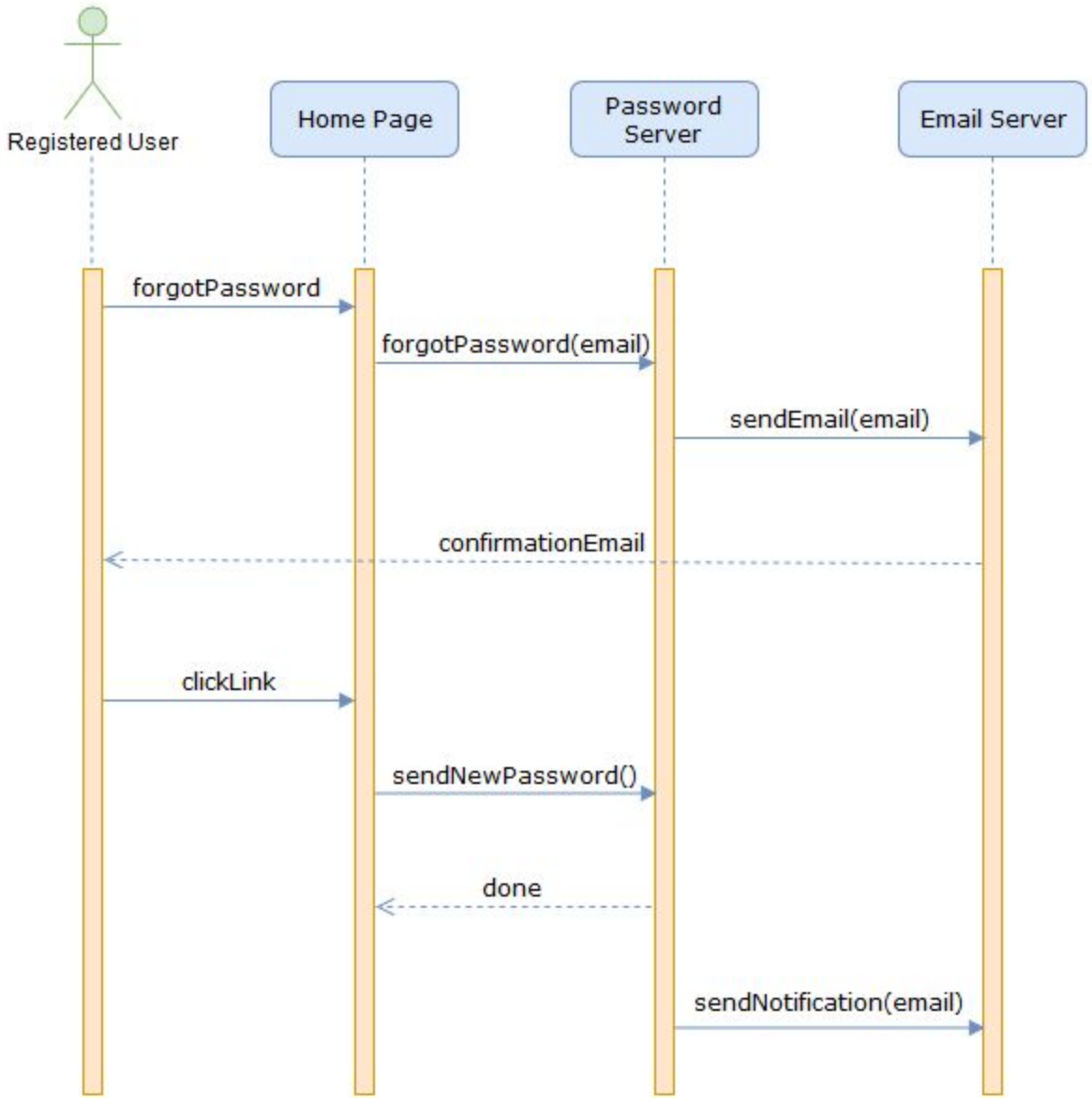
Login with Email



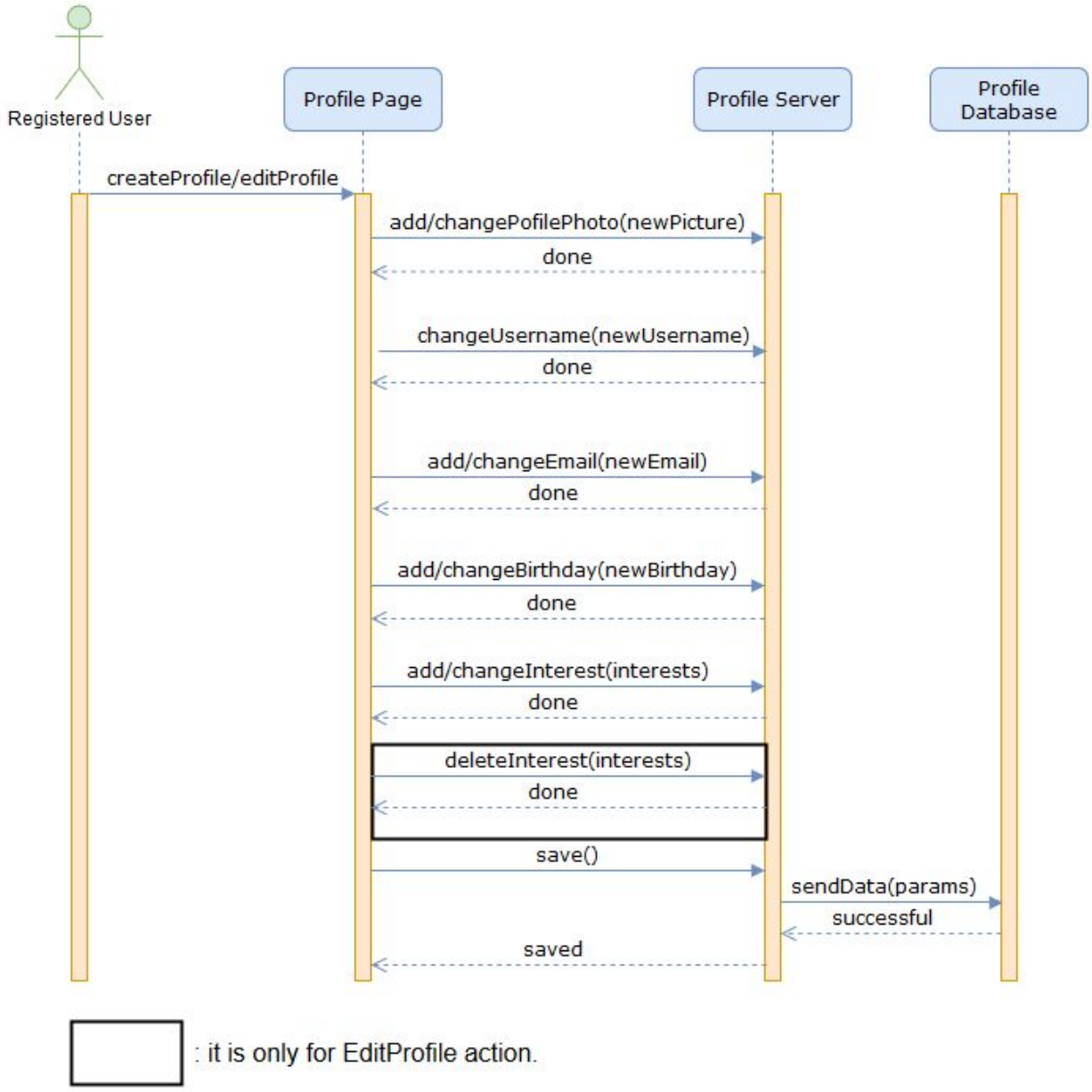
Login with Facebook



Forgot Password

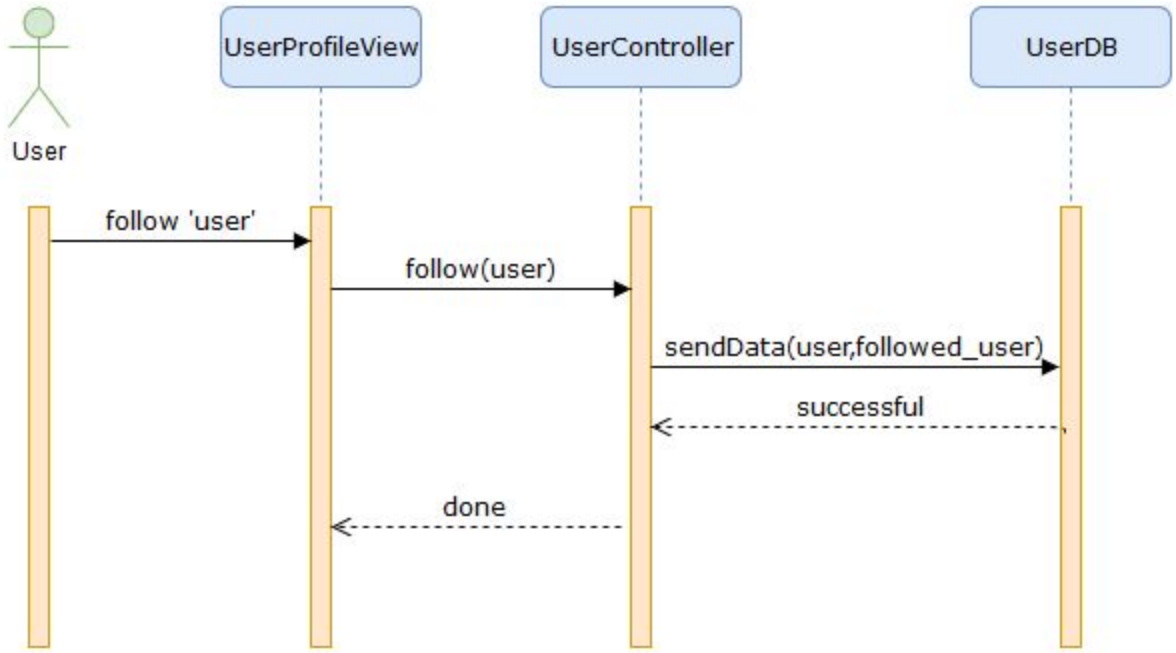


Create/Edit Profile

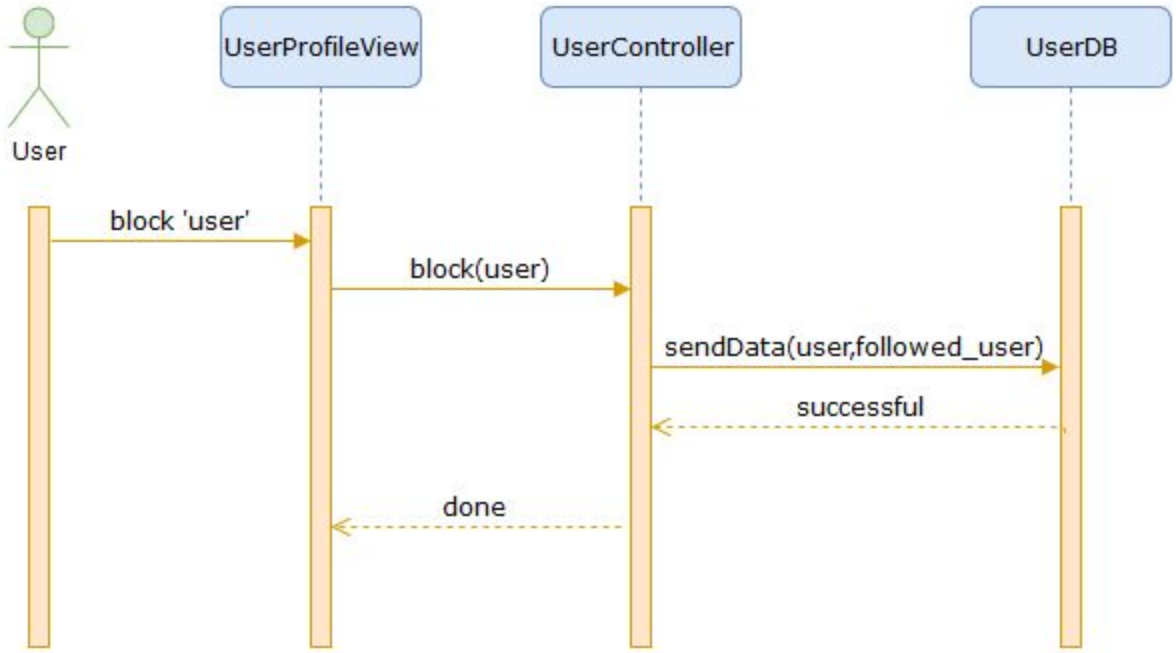


: it is only for EditProfile action.

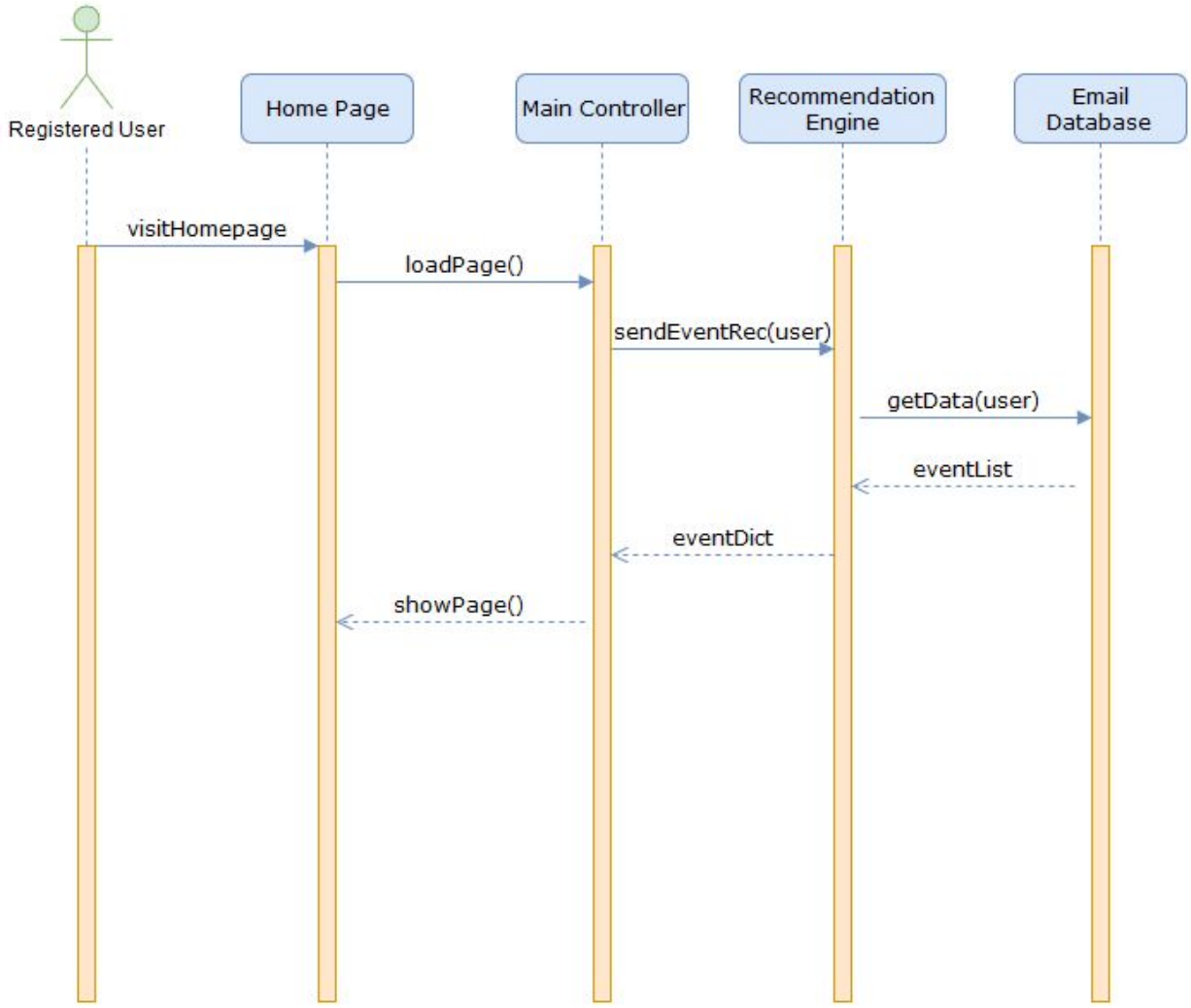
Follow User


User
UserProfileView
UserController
UserDB
follow 'user'
follow(user)
sendData(user,followed_user)
successful
done

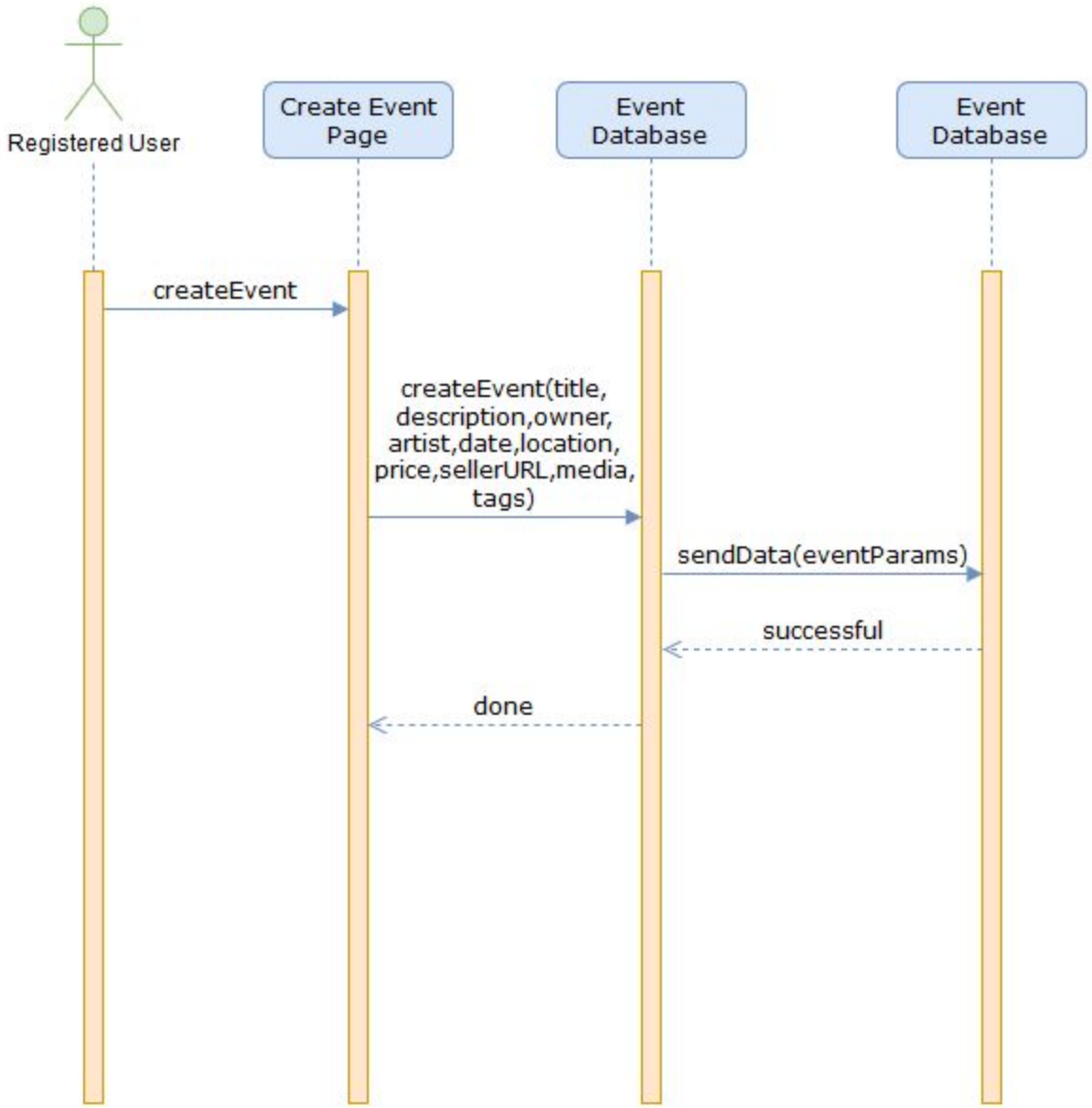
Block User


User
UserProfileView
UserController
UserDB
block 'user'
block(user)
sendData(user,followed_user)
successful
done

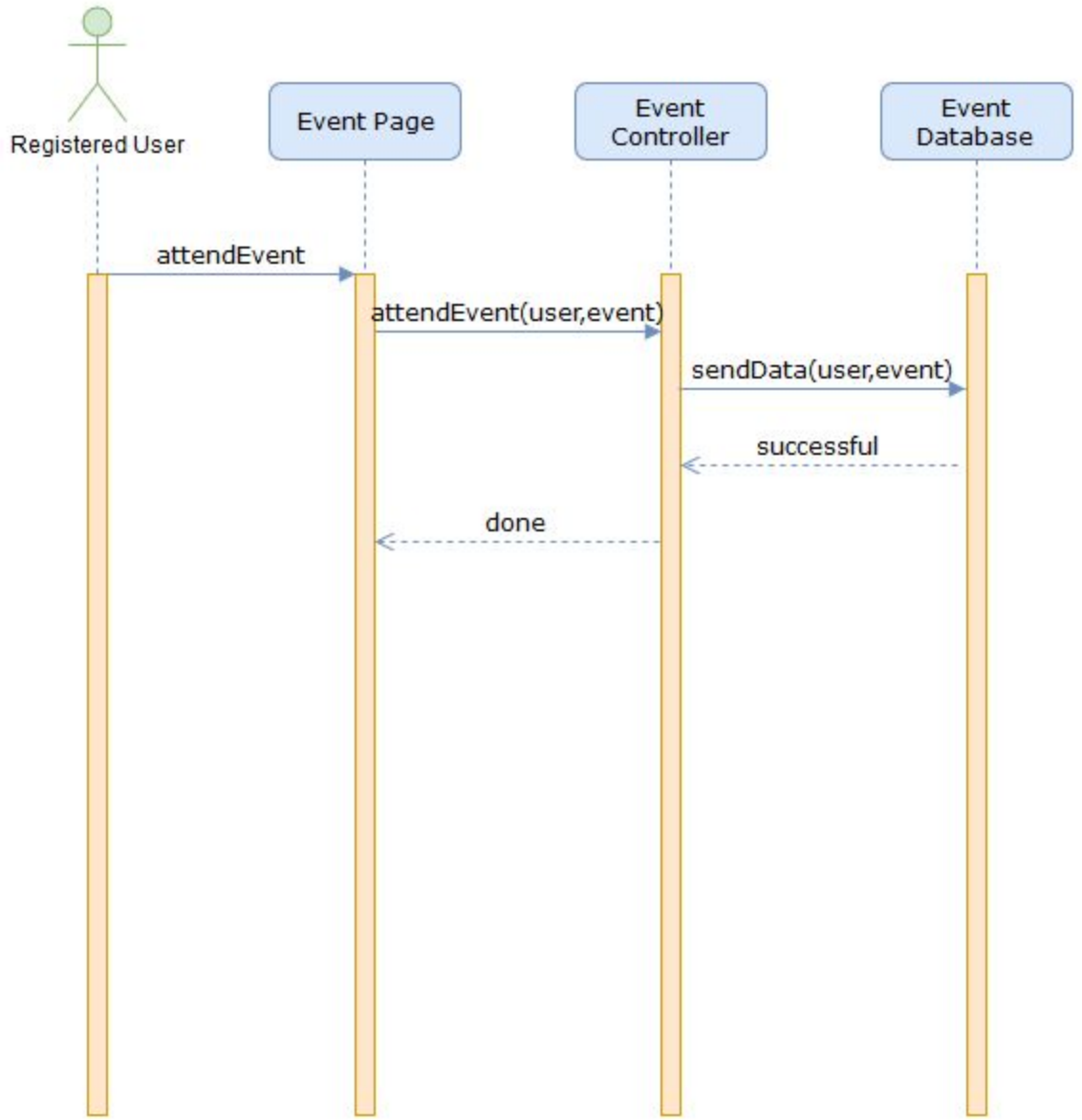
Recommend Event



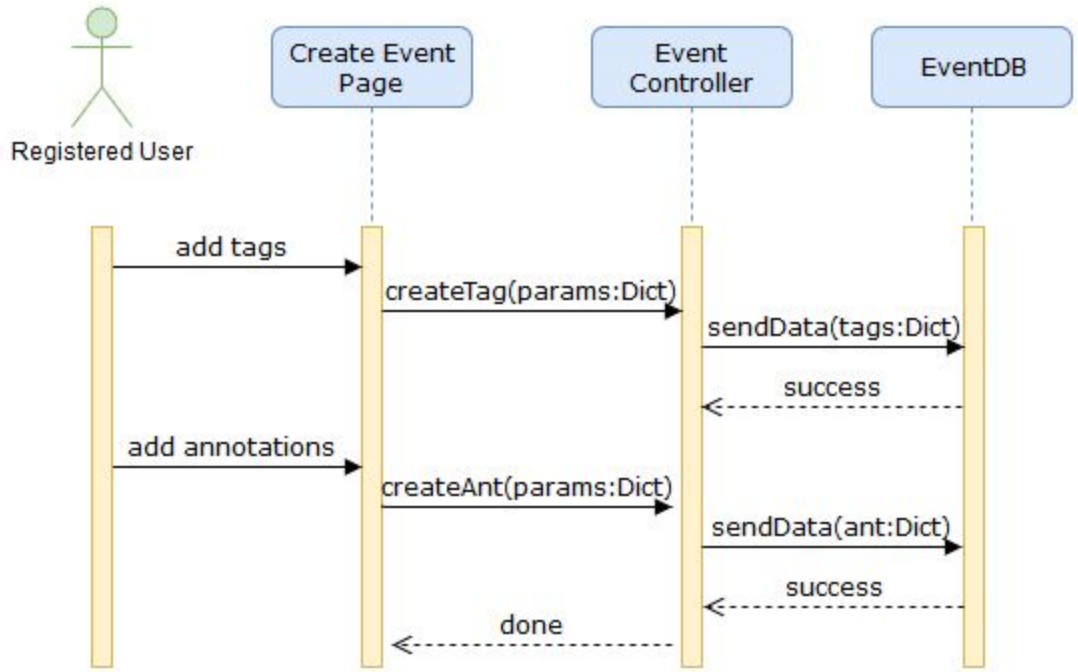
Create Event



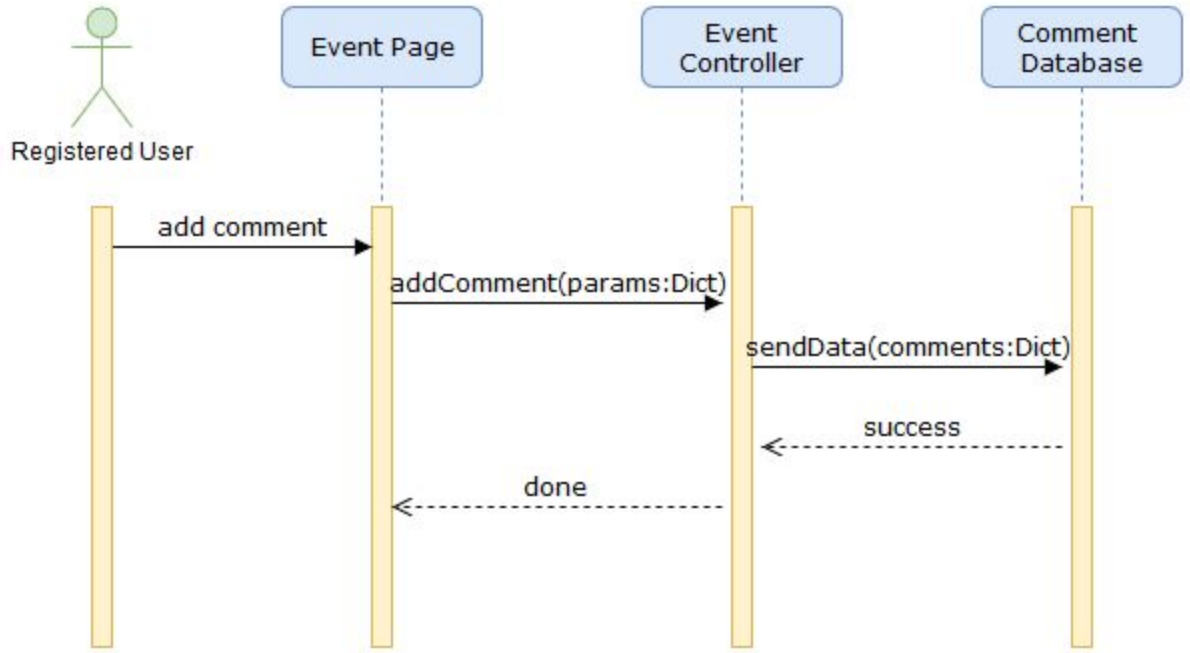
Attend Event



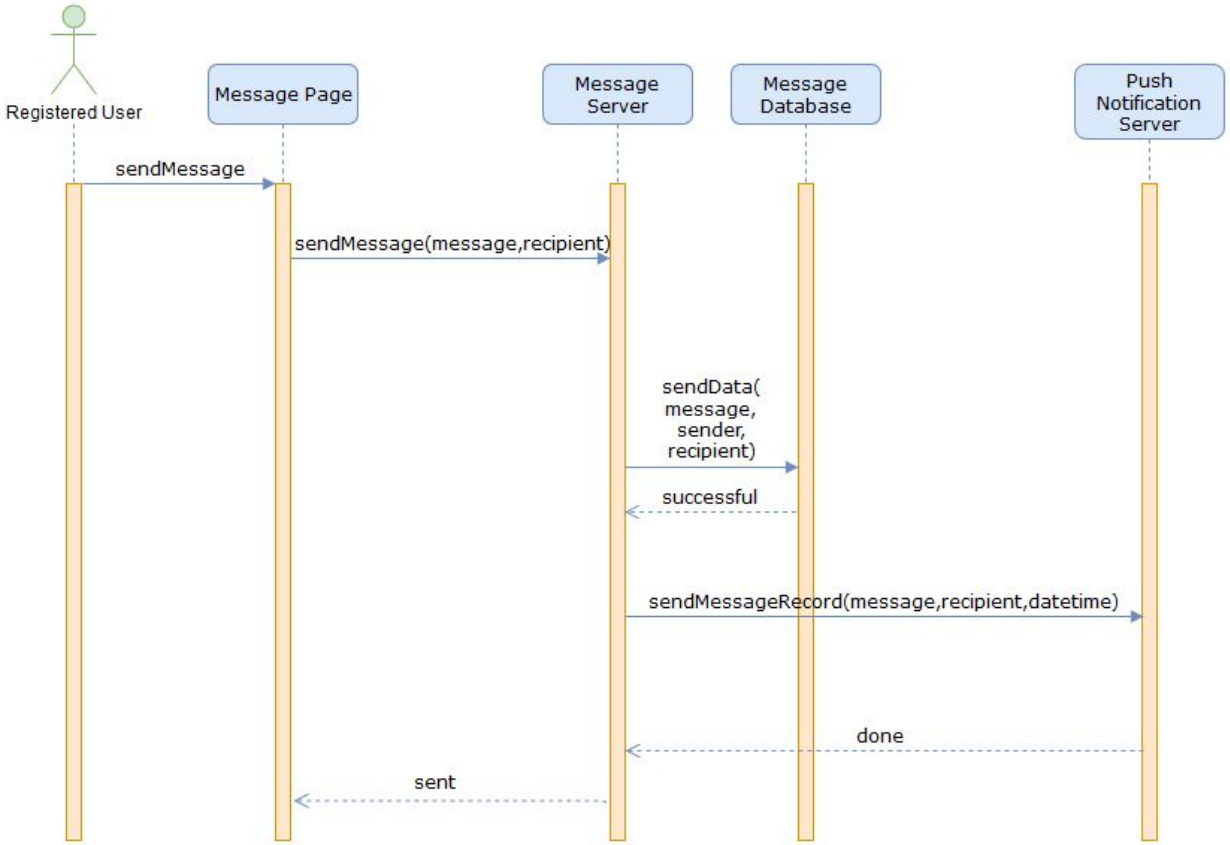
Add Tag/Annotation



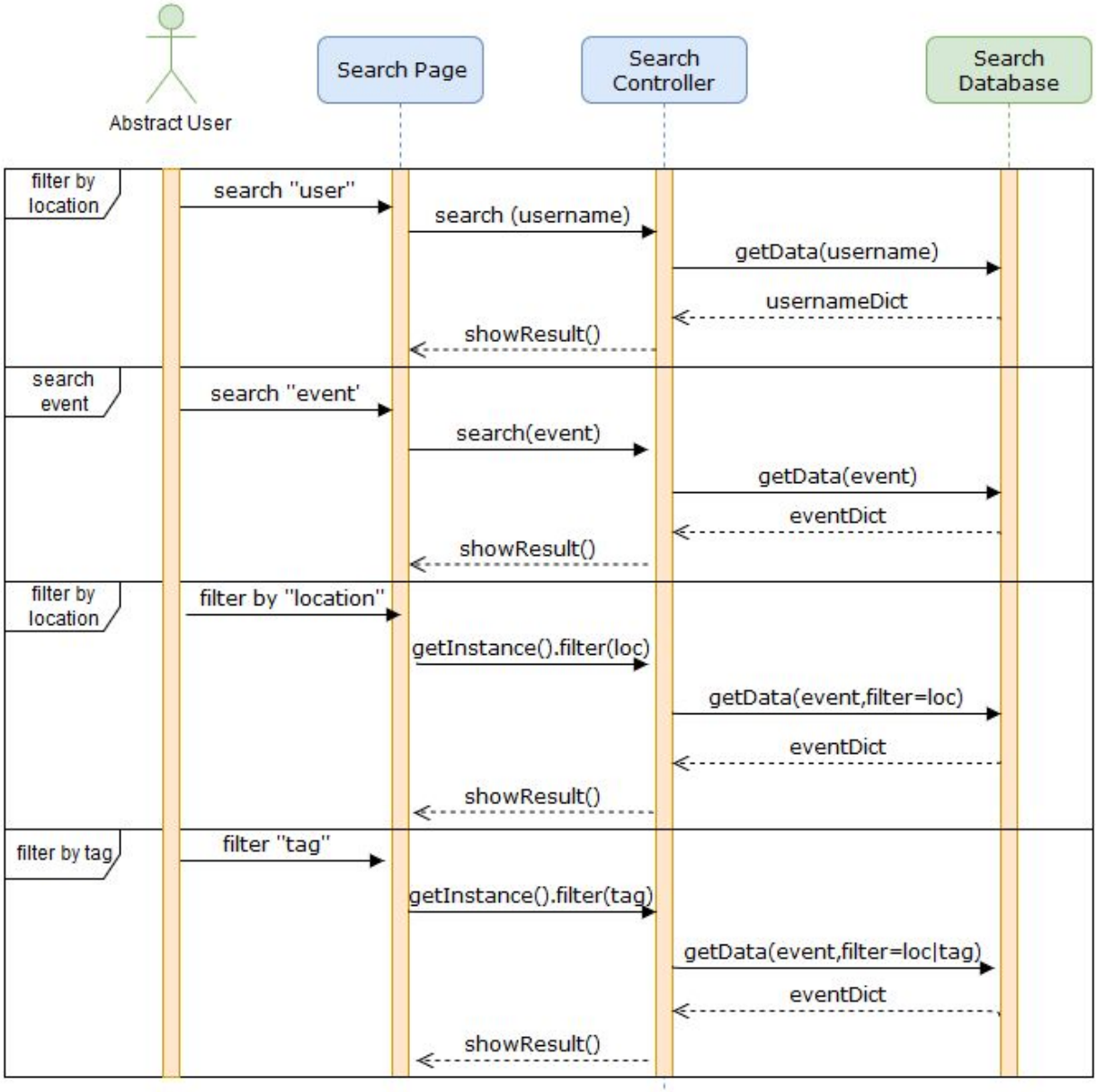
Add Comment


Registered User
Event Page
Event Controller
Comment Database
add comment
addComment(params:Dict)
sendData(comments:Dict)
success
done

Messaging


Registered User
Message Page
Message Server
Message Database
Push Notification Server
sendMessage
sendMessage(message,recipient)
sendData(message, sender, recipient)
successful
sendMessageRecord(message,recipient,datetime)
done
sent

Search with Filtering by Location and Tag



User Tests

Project Name: Cultidate **Group 6**

CHANGE PASSWORD

Information

Test Case ID: TC1

Description: Testing if a user can change his/her own password

Assumptions

- User is registered and log in.
- User is at the his/her own profile page or homepage.
- User must know his/her previous password that is “crazyroot352”.
- User must know his/her email which is toor@gmail.com .

Step Number	Step Description	Input	Expected Result	Actual Result
1	User clicks the Profile button at the top right of the screen	Profile button	His/her own profile page will be shown	-
2	User clicks the Settings button and selects the change password button	Settings button and Change password button	Change password page will be shown	-
3	User enters his/her old password	old password: lazyroot352		-
4	User enters his/her new password	new password: nopassword18		-
5	User enters his/her new password twice	new password again: nopassword18		-

6	User fails to change his/her password	Change button	The old password is not correct!	-
7	User enters his/her old password	old password: crazyroot352		-
8	User enters his/her new password	new password: nopassword18		-
9	User enters his/her new password twice	new password again: nopassword18		-
10	User completes the process of changing password	Change button	The password has been updated successfully.	-

Consequences

- User has a new password.
- User cannot enter the system with his/her old password.

Project Name: Cultidate **Group 6**

FORGOT PASSWORD

Information

Test Case ID: TC2

Description: Testing if a user can take new password due to forgotten one.

Assumptions

- User must be in the home page.
- User must know his/her email which is toor@gmail.com .

Step Number	Step Description	Input	Expected Result	Actual Result
1	User clicks the forgotten password button	Forgotten password button	Forgotten password page will be shown	-
2	User enters his/her own email and clicks continue button	email: root@gmail.com	The system will respond with the error message containing that this email is not found	-
3	User enters his/her own email and clicks continue button again	email: toor@gmai.com	Create new password page will be shown	-
4	User enters his/her new password	new password: cra1		-
5	User enters his/her new password twice for checking	new password: cra1		-
6	User clicks the reset password button but fails.	Reset password button	Password must contain 8-20 characters with at least one number.	-

7	User enters his/her new password	new password: crazyboy18		-
8	User enters his/her new password twice for checking	new password: crazyboy18		
9	User clicks the reset password button and completes the process of taking new password	Reset password buttton	Your password is updated successfully.	

Consequences

- User has a new password.

Project Name: Cultidate **Group 6**

FOLLOW USER

Information

Test Case ID: TC3

Description: Tests whether user is available to follow any other user.

Assumptions

- User must be logged in.
- Both user(following and being followed) should not be on each other's block list.
- Profile page of the account that will be followed should be accessible via username or by searching.

Step Number	Step Description	Input	Expected Result	Actual Result
1	User opens profile page of whom will be followed.	-	Profile page should be visible with a follow button on it.	-
2	User clicks/tab the follow button.	Follow button	The system changes status on follow button to another color and puts a tick on it.	-
				-
				-
				-
				-
				-

Consequences

- Desired profile is followed by the user.

Project Name: Cultidate **Group 6**

BLOCK USER

Information

Test Case ID: TC4

Description: Tests whether user can block any other user in case of inconvenience.

Assumptions

- User must be logged in.
- Profile page of the account that will be followed should be accessible via username or by searching.

Step Number	Step Description	Input	Expected Result	Actual Result
1	User opens profile page of whom will be followed.		Profile page should be visible with a ***block*** button on it.	-
2	User clicks/tab the block button.	Block button	A prompt asking for confirmation should emerge.	-
3	User clicks/tab on ***"Yes"***	***"Yes"*** button	Desired user is blocked and he/she cannot contact with blocking user anymore.	-
				-
				-
				-
				-

Consequences

- User blocks the account that is desired.
- They cannot contact each other.
- Profile pages are visible to each other in restricted form.

Project Name: Cultidate **Group 6**

SQL INJECTION

Information

Test Case ID: TC5

Description: Checks if the system is vulnerable to SQL Injection attacks.

Assumptions

- User has bad intentions.
- User has access to all public pages.
- User is navigated to login page.

Step Number	Step Description	Input	Expected Result	Actual Result
1	User enters an injection code to make username always true	" or ""="		-
2	User enters an injection code to make password always true	" or ""="		-
3	User clicks log-in		An error message will show up.	-
4	User comes back to login page			-
5	User enters a correct email	sqlking1996@me.com		-
6	User enters an injection code to make password always true	105 OR 1=1		-

7	User clicks log-in		An error message will show up.	-
8	User comes back to step 6.			
9	User enters an injection code	105; DROP TABLE Event		
10	User clicks log-in		An error message will show up.	
11	User comes back to step 6			
12	User enters an injection code to harm the DB	\"; DROP TABLE Events; --		
13	User clicks log-in		An error message will show up.	
14	User comes back to step 6			
15	User enters an injection code to add a user	X'; INSERT INTO Users ('email','passwd','login_id','full_name') VALUES ('xDxDLeo@me.com','oscarRocks','Leogardo’,'Leogardo DiGatrio');--		
16	User clicks log-in		An error message will show up.	

Consequences

- The database will not be affected.
- The user will not be able to inject and code to the DB.

Project Name: Cultidate **Group 6**

ADD ANNOTATION

Information

Test Case ID: TC6

Description: User adds an annotation to an image.

Assumptions

- User has logged in.

Step Number	Step Description	Input	Expected Result	Actual Result
1	User clicks add new annotation link		Annotation popup shows.	-
2	User fills the information needed except content.	content: '', x_position: 10, y_position: 50		-
3	User clicks "add annotation" button		An error message will show up saying that content is empty	-
4	User fills the information needed but not choose a coordinate	content: 'green footprints', x_position: null, y_position: null		-

5	User clicks “add annotation” button		En error message will show up saying that position on the image is not selected	-

Consequences

- User has successfully added an annotation

Project Name: Cultidate **Group 6**

DELETE AN EVENT

Information

Test Case ID: TC7

Description: User deletes the event that s/he creates

Assumptions

- User must be registered.
- User must be logged in.
- User must be the owner of the event that is wanted to delete.

Step Number	Step Description	Input	Expected Result	Actual Result
1	User goes to the event's profile page.		Profile page of the event.	-
2	User clicks to the "Delete" button in the event's profile page.		"Are you sure?" message shows up.	-
3	User clicks to the "Yes" button in the message.		Event is deleted and application directs the user to the homepage.	-
				-

Consequences

- User successfully deletes the event.

Project Name: Cultidate **Group 6**

FOLLOW-ATTEND AN EVENT

Information

Test Case ID: TC8

Description: User follows or attends an event.

Assumptions

- User must be registered.
- User must be logged in.
- Event that is searched exists.

Step Number	Step Description	Input	Expected Result	Actual Result
1	User searches for an event.	Name, location or tags for an event	A list of events related to keywords.	-
2	User selects to the event that s/he searches.	-	Selected event profile page.	-
3	User clicks to the "Follow" or “Attend” button displayed on event’s profile page.	-	“Attend” or “Follow” box changes to “Attended” or “Followed” respectively.	-

Consequences

- User successfully follows or attends an event.

Project Name: Cultidate **Group 6**

Facebook Login

Information

Test Case ID: TC9

Description: User logs in with Facebook

Assumptions

- User must be registered to the system.
- Webpage or the mobile application must be opened by user.
- User must be signed in Facebook.

Step Number	Step Description	Input	Expected Result	Actual Result
1	User opens webpage or mobile phone application.	URL	Webpage or application is opened.	-
2	User clicks Login With Facebook button.	Unconfirmed/deficient facebook account	There is no Facebook account to login the system.	-
3	User clicks Login With Facebook button.	User's Facebook Account	Not shown pop-up.	-
4	User clicks Login With Facebook button.	User's Facebook Account	User logs in webpage or application.	-
				-

Consequences

- User successfully logs in the system.

Project Name: Cultidate **Group 6**

Event Comments

Information

Test Case ID: TC 10

Description: User comments on an event

Assumptions

- User must be registered.
- User must be logged in.
- Event must be exist

Step Number	Step Description	Input	Expected Result	Actual Result
1	User searches the event which the topic he/she wants to comment on	Name, location or tags for an event	A list of events related to keywords.	-
2	User finds the event	-	Text of event is shown	-
3	User writes the comment on the text box	1020 characters	The comment must be 1 - 1000 character!	-
4	User writes the comment on the text box	1-1000 character	The comment is added successfully.	-
				-

Consequences

- User successfully comments on an event.
- The comment will be saved on event page.

Project Name: Cultidate **Group 6**

REGISTER WITH EMAIL

Information

Test Case ID: TC11

Description: Testing if a user can register to the system by providing his/her email address and password

Assumptions

- User must be in the Sign Up page.
- User must have a valid email address.
- User must have a secure connection with the system.

Step Number	Step Description	Input	Expected Result	Actual Result
1	User enters an invalid email address	nicholas.sgmail.com	Email is invalid.	-
2	User enters a valid email address	nicholas.s@gmail.com	Email is acceptable.	-
3	User enters an invalid password	1234567890	Password should contain uppercase, lowercase letters and at least one special character.	-
4	User enters an invalid password	Nico*	Password should contain at least 8 and at most 20 characters.	-
5	User enters a valid password	Nico1234*	Password is acceptable.	-
6	User enters a different	nico1234*	Passwords don’t match.	-

	password to confirmation field			
7	User enters the same password to confirmation field	Nico1234*	Passwords match, sign up button is clickable.	-
8	User clicks sign up button	Click sign up button	Email is already taken, please use another email or go to login page.	-
9	User enters another email	nicholas.s@outlook.com	Email is acceptable.	-
10	User clicks sign up button	Click sign up button	Sign up is successful, user is redirected to profile.	-

Consequences

- User is signed up to the system with a unique email address and a secure password.

Project Name: Cultidate **Group 6**

LOGIN WITH EMAIL

Information

Test Case ID: TC12

Description: Testing if a user can login to the system with his/her email and password

Assumptions

- User must be in the Login page.
- User must be registered to the system.
- User must have a secure connection with the system.

Step Number	Step Description	Input	Expected Result	Actual Result
1	User enters his/her email address	pinar.96@gmail.com		-
2	User enters his/her password	Pinar*	Login button is clickable.	-
3	User clicks login button	Click login button	Cannot find an account registered with this email address.	-
4	User enters another email address	pinar.1996@gmail.com	Login button is clickable.	-
5	User clicks login button	Click login button	Password is not correct, try another password or go to forgot password page.	-
6	User enters another	Pinar2018*	Login button is	-

	password		clickable.	
7	User clicks login button	Click login button	Login is successful, user is redirected to homepage.	-

Consequences

- User is logged in to the system.

Project Name: Cultidate **Group 6**

DELETE USER

Information

Test Case ID: TC13

Description: Testing a user can delete himself/herself from the system

Assumptions

- User must be registered.
- User must be logged in.
- User must know his/her password that is “crazyroot352”.

Step Number	Step Description	Input	Expected Result	Actual Result
1	User goes to the settings page.		Settings page of the user.	-
2	User clicks to the “Delete” button in the event’s profile page.		Re-authentication pop-up shows.	-
3	User enters his/her password	password:“crazyroot351”		-
4	User fails to type his/her password	Delete button	The password is not correct!	-
5	User enter his/her password	password:“crazyroot352”	The password is not correct!	-
6	User completes the process of deletion	Delete button	The user has been deleted successfully.	-
				-

Consequences

- User removes his/her from the system.
- User can not see events and other contents of the system anymore.

Project Name: Cultidate **Group 6**

MESSAGE

Information

Test Case ID: TC14

Description: Testing a user can send and receive message correctly

Assumptions

- All users must be registered and User1 is logged in
- User2 is blocked by User1

Step Number	Step Description	Input	Expected Result	Actual Result
1	User2 sends message to User1	“Will you come tomorrow?”	“You can’t message to User1” failure message	-
2	User1 sends message to User3	“Will you come tomorrow?”	The message is received by User3	-
3	User3 sends message to User2	“Yes, of course……”, this message has length more than 1000	“Your message is too long” failure message	-
				-

Consequences

- User2 can’t send message to User1 because he/she is blocked.
- User1 can send message to User3.
- Users can’t send message more than 1000 characters.

Project Name: Cultidate **Group 6**

Create Event

Information

Test Case ID: TC15

Description: Test if a user can successfully create an event

Assumptions

- User is registered and is in an open session.
- User has successfully navigated to their profile page.

Step #	Step Description	Input	Expected System Response	Actual System Response
1	User clicks on the “Create Event” button	“Create Event” button	Navigate the user to the “Create Event” page	-
2	User fills in the information fields.	‘Title: Figaro Goes Jazz‘ ‘Description: Best of the Best‘ ‘Artist: D. G. Boymann‘ ‘Date: 07.03.2018‘ ‘Location: Albert Long Hall‘ ‘Price: 40‘ ‘Seller URL: https://klasikmuzik.boun.edu.tr/‘ ‘Media: media.jpeg‘ ‘Tags: music, concert‘		-
3	User clicks on the		User receives a	-

	“Create” button		message saying: “Event successfully created!”	

Consequences

- An event is created in the system with given information fields

Project Name: Cultidate **Group 6**

Edit Event

Information

Test Case ID: TC16

Description: Test if a user can successfully edit an event

Assumptions

- User is registered and is in an open session.
- User has successfully navigated to the event page.
- User is the owner of the event they want to edit.
- The event has ID: 248.

Step #	Step Description	Input	Expected System Response	Actual System Response
1	User clicks on the “Edit” button.	“Edit” button	Navigate the user to the “Edit Event” page for the event with the ID 248.	-
2	User fills in one or more relevant information fields.	‘Title: ‘ ‘Description: ‘ ‘Artist: ‘ ‘Date: ‘ ‘Location: ‘ ‘Price: 7‘ ‘Seller URL: ‘ ‘Media: ‘ ‘Tags: ‘		-
3	User clicks on the “Save” button.		User receives a message saying: “Saved changes!”	-

Consequences

- Event with the ID 248 is updated by the system according to the specifications of the user.

Project Plan

Since the plan is a huge paper on a pdf, please visit:
https://github.com/bounswe/bounswe2018group6/blob/master/wiki_assets/Cultidate_project_plan.pdf

Software Development Infrastructure

For the HW8, we worked on Twitter API. It was an exercise to understand how APIs work, analyze and serve data, as well as how they are developed and deployed. Moreover it was a good opportunity to practice with GitHub's environment deeply.

Initially, our team held a meeting, get some insight on Twitter API together and discussed which framework and technologies will be used and how these works will be shared. For the backend framework, Django was selected; since it's extensible, scalable, based on Python, backed by a great community, and easy-to-manage.

Our plan was to create a base using Django and allow everyone to develop their API endpoints and API web interfaces individually. Since Django has some parts that shouldn't be publicly revealed, we excluded those parts using .gitignore.

For every API endpoint that we provide, there exists two features: An API call returning JSON response and a corresponding frontend HTML page showing the API call with visual interface. Every member of the team followed guided URL structure and template for implementing their features.

Once the base part of Django was set up, every member of the team created their branches to implement and test their part. Sometimes we made some changes on the master branch and requested our members to merge it to their branches to get the latest version of it.

During development, testing was crucial and we performed tests on our local system using Postman and local browser. The same tests were applied just after deployment too.

After collecting Pull Requests on GitHub, it was very beneficial to read and review what the others wrote and request changes if necessary. After completing this, we did a meeting to merge Pull Requests. Before the meeting, everyone was told to merge the master branch to

their individual branches to solve conflicts beforehand and keep the merging process painless. Since we kept the must routine, it was easy to merge.

When our project was ready to deploy, we created a virtual private server, created a virtual environment, installed the requirements, pulled the repository, tested and hence completed the deployment. If you want to see the project please go to: https://github.com/bounswe/bounswe2018group6/tree/master/hw_twitterapi whereas if you want to give a try, visit the address written at this link.

Evaluation of Tools and Managing the Project

Purpose	Technology	Evaluation
Back End	PyCharm	A Python IDE which was used to write the code in the Twitter API app. It has all the features one would expect from a proper IDE.
Back End	Django	High level python web framework which we have used in Twitter app. A very convenient and easy-to-manage tool.
Front End	Webstorm	An IDE for JS and compiled-to-JS languages, HTML and CSS. It is among the more popular IDEs used in its respective field, and our team is quite comfortable with it.
Front End	Bootstrap	Popular and very useful CSS library for easy design of visual web interfaces.
Front End	Html5	Current active major version of the most popular markup language. Used with ease in the Twitter API apps front end.
Android	Android Studio	Official IDE to develop native

		Android apps.
Versioning System	Git	Main SCM system used by the team for the entire project. Its distributed and branching nature is what makes it convenient to use.
Travis	Continuous Integration	In order to make integration process painless and keep and track builds of the projects alive.
Database	PostgreSQL	The most convenient relational database for using with Django.
Task Queue	Celery	Since Django has no built-in feature to run timed tasks, leading Python library Celery is selected to do this.
Diagram	LucidChart	Chart-topping diagram drawing application. Used for drawing Class Diagrams and Sequence Diagrams. One of the convenient tools for its respective application used in our project.