Robin: Milestone 2 Report
Group 3
December 6, 2018

1 Executive Summary

This section summarizes the current status of the project by the date of 6th of December and gives an insight about the future development plans.

1.1 Summary of Project Status

According to Milestone 4, the main parts of the project to be complete by the date of 4th of December were

- Event page for android and web frontend,
- Admin Panel page for android and web frontend,
- Profile page for android and web frontend,
- Homeage for android and web frontend,
- And Backend endpoints of above pages.

As of 4th of December, we have completed most of these tasks successfully. We created several endpoints for events, profiles, tags etc. We have managed to connect most of them to Frontend and Android. We implemented a basic search function for events. Now our users can create and delete events. In frontend, we created some new pages and enhanced existing ones. We had some issues while connecting backend to frontend and android. Because of these problems, not all of implemented functions are distributed to all parts of the project equally, but we are confident that we will manage do this in the upcoming days.

1.2 Changes from Milestone 2

1.3 Changes Planned

For the Milestone 5, the main tasks and additions planned for our project include:

- Implementing Annotation functions, at least partly,
- Enhancing search functions,
- Implementing following and blocking of tags and users,
- Adding comments,
- Making homepage more unique for each user by using following tags data,
- Implementing recommendation,
- Defining the color scheme for Android and making it more similar with frontend.
- Increasing documentation for our code.

2 List and Status of Deliverables

✓Web: Navigation bar (frontend) -> Completed.
✓Web: Event Create/Delete (frontend) -> Completed.
✓Web: Search Page (frontend) -> Completed.
✓Web: Homepage (backend) -> Completed. Event model is available as an end point.
✓Android: Homepage -> Completed
✓Android: Navigation Bar -> Completed
✓Android: Search Page -> Completed
✓Web: Sign up and Login Page (backend) -> Completed.
✓Android: Sign up and Login Page -> Completed.
✓Android: Guest User Page-> Completed.
✓Web: Admin Panel (backend) -> Completed.
✓Web: Event Creating/Editing (backend) -> Completed.
✓Web: User Creating/Editing (backend) -> Completed.
✓Web: Searching (backend) -> Completed.
✓Web: Tagging (backend) -> Completed.
✓Web: Rating (backend) -> Completed.

In frontend part, we implemented post and get requests to talk with the backend, used axios for doing so. We added functionalities such as:

1. Event Creation
2. Event Deletion
3. Event search
4. Interested/Going to an event
5. Rating of the events

In addition to the milestone deliverables, we also implemented search page and comment structure in our design.

In android, enhancements of the login and sign up pages were made like correcting the warnings when there was a typo or making the user aware of the wrong inputs in email and password areas. We also created functional pages for guest users where they could see and search events and reach the registration page easily with a button existing on the page. When a registered user entered the system, thanks to the new improvement, s/he can see the events that the backend has. And also, a template for creating an event was created but without function. These were all leading us to complete the project day by day. Now new challenges are ahead.

In backend, we made get and edit methods for event and user models. We add a couple of modules for django-filters and django-cors-headers. We implemented search function by using django-filters. We are also going to use it for filtering several models when we need. To sort events in homepage, a comparative method is implemented.

3 Coding Done

Coding Done by Members		
Team	Group Member	Work Done
Backend	Furkan Enes Yalçın	Designing and implementing endpoints for events, designing the template of serializers for other models. Fixing, CORS issue and changing token expiration interval. Implementing search and homepage for event.
	Oğuzhan Yetimoğlu	Designing and updating serialization methods and models for users.
	Onur Varkıvanç	Designing and updating models for events, tags and comments. Adding inter-model relationships. 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.
Frontend	Özge Dinçsoy	Event page view, event page style, add event page, add event page style, google maps setup, comment view and style, add comment, star rating
	Recep Deniz Aksoy	Back-end = Front-end connections(axios), data management and Search page and functionality, redirections.
	Uğurcan Arıkan	Navigation bar, different registration errors, profile page, profile edit page, private profile page, backend connection of the profile page, login or register notification bar for guest users, create event page style changes and error messages, create event success page
Android	Çağdaş Tombul	Guest user page, design, login and sign up pages' warnings
	Mustafa Şişik	Homepage, event-user searching template, event creation page, profile page arrangement

4 Requirements

For Milestone-2, we tried to implement the rest of the specifications that we determined before. In other words, we followed our own milestones. At the time when we had customer meeting, we were almost up-to-date considering our project plan. Actually, at the end of the project, we are going to implement annotation. Therefore, the actions that we must take should follow a path to this aim. After the first milestone, which is, after the login to the system, the first thing that a user should do is to see existing events and create new events. Moreover, the user also should be able to see and manage his/her profile page. Actually the way we think implementing the project corresponds to a journey which a user possibly experience while using the applications. Therefore, the requirements that we tried to satisfy are listed below as fully and half completed.

These items are the ones that the connection between backend and frontend/android are not made. The buttons or information existing in both platforms for all items are explicitly stated, however they are not functioning.

- **1.2.1.7.8** Users shall be able to follow cultural events. (backend is ready but not connected to both platforms)
- **1.1.1.7** The system shall provide rating and comments of the event. (android-backend connection is not done)
- **1.2.1.1.2** Users shall be able to sign up by linking their Facebook account. (it is not functional for both platforms)
- **1.2.1.7.1** Users shall be able to create event pages for both upcoming and past events. (android-backend connection is not done)
- **1.2.1.7.5** Users shall be able to rate reflecting their experience on cultural event pages. (android-backend connection is not done)
- **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. (some of these features are done in frontend)
- **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. (some of these features are done in frontend)
- **1.2.1.7.6** Users shall be able to put comments reflecting their experience on cultural event pages. (backend is ready but not connected to both platforms)

This part is done correctly and works as it should do. They function without any problem. Also the platforms where the requirements are done are written explicitly.

- **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. (both platforms)
- **1.1.1.4** The system shall provide event date. (frontend connected to backend)
- **1.1.1.5** The system shall provide event price. (both platforms)
- **1.1.1.7** The system shall provide rating and comments of the event. (frontend connected to backend)
- **1.1.2.3** Search function shall be used with more than one input. (both platforms)
- **1.1.5.1** Admins shall be able to add, delete, and edit content through admin panel.
- **1.2.1.1.1** Users shall be able to sign up by providing necessary personal information. (both platforms)
- **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. (frontend connected to backend)
- **1.2.1.2.6** User profiles should contain a short list of events which the user has attended. (frontend connected to backend)
- **1.2.1.6.1** Users shall be able to search events by name. (frontend connected to backend)
- **1.2.1.6.3** Users shall be able to search for other users. (frontend connected to backend)
- **1.2.1.6.4** Users shall be able to search and filter events by location. (frontend connected to backend)
- **1.2.1.7.1** Users shall be able to create event pages for both upcoming and past events. (frontend connected to backend)
- **1.2.1.7.5** Users shall be able to rate reflecting their experience on cultural event pages. (frontend connected to backend)

- **1.2.2.1** Guests shall be able to see event pages created by users. (android connected to backend)
- **1.2.2.3** Guests should see a sign up dialogue on the homepage. (android connected to backend)
- **2.1.4** The application shall be provided for both web and mobile platforms.
- **2.3.2** The system shall encrypt all user passwords before storing them in the database.
- **1.2.1.1.5** Users shall be able to verify their profiles.
- **2.4.3** System should be secure against SQL injections.

5 Design

5.1 Frontend

In addition to our design in the previous milestone, we have designed profile page and event pages and also the other pages which are related to them such as profile edit page or create event page etc.

There weren't a specified profile page design in our mockups for the web frontend so we have designed it from scratch and we have decided to have a profile card on the left hand side of the profile page which will summarise the user profile with its name, location and birthday infos and also profile photo. We then decided to have 5 tabs which will be about, events, followers, following and photos with their functionalities being pretty straightforward.

For a user to edit his/her own profile, we have decided to add an edit tab in addition to those. We have also designed a private profile page which will only show the name and the profile photo of the user in case he/she decides to make his/her profile. In addition to profile page, we have also added an event page to our website.

Unlike the profile page, we already had a design mockup and we stayed loyal to this mockup. Our event page consists of event name, photos, event info, price, date, time, comments, map creator of the event and join and interested buttons for the users to be able to join or get notified about the event.

Also we have added create event page for users to be able to create events. It asks users for the necessary information about the event such as name and info etc. and displays error messages if there are any.

After the event is created, it redirects the user to the create event success page which summarises the info user just entered. As we did not have and create event success page, it was also designed from scratch.

We have also designed a search results page which will show the results of the user's latest search and list them. We already had a search results page mockup and we followed its example.

We have also added a login or register to view more bar in order to redirect unregistered users to register to our website in order to view more info and use all the functionalities of the website.

5.2 Android

We selected a base color for android and backend as grape and created a logo implies Robin. We arranged them so that they look in harmony with each other. Of course, it was not like the one imaged in the mock-ups because we even changed the logo of the application. After that, android and backend worked on different sides. As android team firstly we have created a simple event design for event list in homepage, and also a new event page design with xml. When creating design we use material design.

In our android application there are 2 pages for user control, Login Page, and Register Page and there are also 3 base page in Mainactivity such as homepage, profile page, and my events page(this will be considered later) and some extra pages also such as event page, create event page, edit profile page.

In login page we have 3 main login functionality, email-password, as guest and Facebook login. If user wants to register with email, he or she can go to register page with clicking register text.

In register page we have 4 textfield to get user data, username, email, and two password field. If user registered successfully, he or she informed with “verification email sends” text as Toast dialog, else informed with error message. There is also a login text in bottom of this page. If user click here user is redirected to login page back.

MainActivity has different design from login page, and register page, It has a toolbar on top of page including a title, a subtitle, and a menu icon, At bottom side there is navigation bar also which includes 3 icon for logged in user. In this activity we have 3 base page connected with bottom navigation icon clicks, homepage, profile page, myevents page(will be considered later.). These pages reside between bottom navigation and toolbar at top.

In homepage, we have search bar on top of page and also a list below search bar. At opening of the page we can see homepage events in that list. If we want to search for events and user we can type in search bar and click search icon in keyboard. Event or user search can be choosen with a switch button next to search bar. if it disabled we can search events, else after click on search icon we can see users in that list.

Profile page has a profile icon on top-left of page and name right of this image. Under the image we can see a tab bar which include 3 text buttons in order of Events, Following and Follower . We can see user events by clicking events button, and see following and followers by clicking related buttons. These info listed below tabbar.

Event page is a detailed page for event redirected after clicking event in the event list in homepage. This page has a banner image for event, artist name, price, ratingbar, event title, description, There is also an image gallery below all items mentioned above. After this image gallery we have a google map showing event location. In addition to this there is a list containing user comments below google map part.

My events page(Will be considered later)

6 Project Plan

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	
Recomendation	05/12//2018	12/12/2018
Implementing W3 annotations	12/12/2018	19/12/2018
Design and implementation improvements	19/12/2018	25/12/2018
Milestone - Project Finalized	25/12/2018	
Testing	17/12/2018	23/12/2018
Debugging	24/12/2018	31/12/2018
Presentation	02/01/2019	

7 Code Structure

We are working in the folder app and within it all teams have a folder, these are backend, frontend and android. We worked differently as different teams in different branches. Then merged our works in a new branch called dev. We opened issues for features and implemented them in respective branches.

7.1 Backend

We wrote our code on “backend” branch. Since there were many models that needed implementing for this milestone (like users, events, comments, etc.), we have divided the work among ourselves easily. Sometimes when the job at hand was big, a further division was made when writing the database model, its serializers, its views and its endpoints were written by a collaboration of different members. After completing the task at hand, new code is always tested on localhost before being pushed into the repository. There were no conflicts since mostly we worked on separate models. After all components were tested and merged in the repository, new changes are pulled to the server and deployed. And finally, after deployment, a documentation was added so that it would be easier to connect android and frontend components to the endpoints we created.

7.2 Frontend

We created new branch for each work. When we completed the task in the new branch, we make a pull request and add our team members (Deniz, Özge, Uğurcan) as reviewers. If they check and apply the pr, we merge the branch to dev. Dev is our protected branch and direct action to dev branch is restricted.

7.3 Android

By creating a new branch “android2”, we (Çağdaş and Mustafa) can able to work after merging all codes to dev branch. In this branch we are working together in different parts of application in different time intervals, because of this we are working on the same branch (“android2”).

8 Evaluation of Tools and Management

8.1 Backend:

Two of the few new modules that we’re using are django-filters and django-cors-headers. django-cors-headers is used for fixing the CORS issue that is the backend part was not accepting some requests from backend because of their headers, we think.

We used django-filters for implementing search function. We are also going to use it for filtering several models when we need.

We made our JWT tokens expire in a week rather than in a 5 mins in order to make our Android app functional.

We are still using a Digital Ocean Droplet for deployment but since we are deploying backend and frontend servers on the same droplet, we had to increase its RAM etc.

8.2 Frontend:

Since we did not add any additional libraries to frontend except for google maps and star rating libraries we only built upon existing libraries and functions.. For frontend, we are using Visual Studio Code for our ide. We are also using React as our framework. For the installation of the dependencies and package management, we are using npm. And for the rendering of our pages, we are using bootstrap for better responsiveness. Some other important dependencies are as follows:

- bootstrap : main library used for the render part of our pages. We are using bootstrap4.
- jquery : for some functionalities used in bootstrap components, such as dropdown.
- axios : for the request to the backend endpoints.
- react-router-dom: tp handle routing of our website, as our website is multipage and react was first intended for single page sites.
- js-cookie : to handle authorisation by putting token to the cookies at login.
- font-awesome: for some of the fonts and icons that do not exist in the current bootstrap.
- reactstrap : used for some of the components it contains such as card, button etc.
- react-google-maps : to use google maps in our app.
- react-location-picker : to mark locations and show locations.
- react-star-rating-component : to rate events, comments, and users.

8.3 Android:

For Android we are using Android Studio as ide which is a very common code editor for android development. Android Studio provide us with multiple options and has many useful features for development. When developing android project we are using sdk version 28 of android. There are also some external libraries in our implementation listed below.

- Bottom Navigation: It is a library for bottom navigation, it is simple and has some extra feature regarding standard library of android.
  (**com.aurelhubert:ahbottomnavigation:2.1.0**)

- Glide Library: This library is image loading framework. It is very fast and has advance image caching feature.
  (**com.github.bumptech.glide:glide:4.7.1**)

- Circle Imageview: This makes imageview circular. It is just used for design.
- Volley Library: is a HTTP library that makes networking for Android apps easier and, most importantly, faster.
  (**com.mcxiaoke.volley:library:1.0.19**)

- Google Map: is a map tool, for display event location for our application (**com.google.android.gms:play-services-maps:16.0.0**)
-
- Circle Image View: is a library for make imageview as circle, we use it in some listing fields.
  (**de.hdodenhof:circleimageview:2.2.0**)

- CardView: This is a material design library to show items like cards, we are using this tool to show our events and listing element as cards
  (**com.android.support:cardview-v7:28.0.0**)