

Robin: Milestone 1 Report

Group 3

November 1, 2018

1 Executive Summary

This section summarizes the current status of the project by the date of 30th of October and gives an insight about the future development plans.

1.1 Summary of Project Status

According to Milestone 3, the main parts of the project to be complete by the date of 30th of October were

- Sign Up page for android and web frontend,
- Sign In page for android and web frontend,
- Login API,
- Sign up API,

As of 30th of October, we have completed all these tasks successfully. We have a working API for login which successfully checks the credentials - email and password - and for sign up which registers new users by their required inputs - email, full name and password - . In frontend and android, we also have adequate pages and designs for both sign in and sign up. We have followed the same design principles as of colours and css styles etc.

On top of our Milestone 3 goals, we have also managed to go further and accomplish some tasks from Milestone 4 as well. Those tasks include:

- Forgotpassword page for web frontend
- Forgotpasswordsuccess page for web frontend
- Registersuccess page for web frontend
- Home page template for web frontend and android
- Event page template for web frontend and android
- Event components to be rendered in the web frontend and android
- Web admin panel for web backend

1.2 Changes Planned

For the Milestone 4, the main tasks and additions planned for our project include:

- Completing home page for web frontend and android with addition of a more complete navigation bar with search functionality,
- Completing event page with more details and a better design for web frontend and android,
- Adding profile page for android and web frontend,

- Having a more user-friendly admin panel for web backend,
- Increasing and ameliorating the documentation of our current codes for android, web frontend and backend,
- Adding API endpoints for home page, event page and profile page,
- Testing already completed API endpoints,
- Adding pull request template to github for a better github experience and for more code security

2 List and Status of Deliverables

- ✓ Web: Homepage (frontend) -> Completed and it shows list of events.
- ✓ Web: Homepage (backend) -> Completed. Event model is available as an end point.
- ✓ Android: Homepage -> Homepage shows events and has navigation bar but it does not connect with backend.
- ✓ Web: Sign up and Login Page (frontend) -> Completed.
- ✓ Web: Sign up and Login Page (backend) -> Completed.
- ✓ Android: Sign up and Login Page -> It works functional. Completed.

In frontend part, we first used material ui, beside react. It was not much useful in case of converting components, so we decided to use bootstrap. We have 4 main pages for milestone 1 as homepage, login, signup, and event. Login and signup were functional, and they are connected to backend. In homepage, we have used reactstrap library for navigation bar but we will change it to basic bootstrap to write our code as simple as possible. In event page, we have used react-star-rating-component to vote event with stars.

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

In android, what was logical was to start with implementing sign up and login pages as it was so for the others. Moreover, it was a decision which we made as a group to make the codes compatible with each other. After login page, users should have been provided a homepage so that they can utilize the application as it should be. To show some events on the homepage was another thing that we thought it would be needed. After that we intuitively felt that the first part of the project for android is done and ready to be presented in the first milestone.

In backend, we first decided to make a legit login and registration with tokens and confirmations. The reason for this choice was to decide a request template for communicating between backend and frontend. For example we are now using JWT tokens at the headers of the requests for authorization and continue to implement other functions, endpoints with this convention. Now our priority is creating database models and determining how to access these data.

3 Coding Done

Coding Done by Members		
Team	Group Member	Work Done
Backend	Furkan Enes Yalçın	Registration, login, logout, and email verifying
	Oğuzhan Yetimoğlu	Serialization
	Onur Varkıvanç	Database models
Frontend	Özge Dinçsoy	Web navigation bar, event page structure and comment component
	Recep Deniz Aksoy	Event components in homepage, token management
	Uğurcan Arıkan	Web login, signup, forgot password, signup success, forgot password success, and redirection pages, backend requests, logo design
Android	Çağdaş Tombul	Android signup, android design
	Mustafa Şişik	Android home page, event page, login, android design

4 Requirements

For Milestone-1, we tried to implement as much as we could. Also we needed to follow our own milestones. At the time when we had customer meeting, we were up-to-date considering our project plan. We were implementing event items even if they were not ready exactly since we had time for their deadlines. The parts that we implemented were not random. For the initialization of the project, the first step should have been the connection of frontend, backend, and android and activate the system as a whole. Since a user is going to use the application starting with the signup and login pages, starting the implementation with these pages looked quite logical. After the creation of these, the connection took part and the system was up to use. All these pages were implemented with the help of the requirements. Every piece of code was written in the light of them. The requirements that we fulfilled or half-fulfilled for the customer meeting are presented below:

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

- 1.1.1.4 The system shall provide event date.
- 1.1.1.5 The system shall provide event price.
- 1.1.1.7 The system shall provide rating and comments of the event.
- 1.2.1.1.2 Users shall be able to sign up by linking their Facebook account.
- 1.2.1.1.4 Users shall be able to reset their passwords via a secure e-mail.
- 1.2.1.7.1 Users shall be able to create event pages for both upcoming and past events.

This part is done correctly and works as it should do. They function without any problem.

- **1.2.1.1.1** Users shall be able to sign up by providing necessary personal information.
- **1.2.1.1.3** Registered users shall be able to log in using the method they signed up.
- **1.2.1.1.5** Users shall be able to verify their profiles.
- **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.
- **2.4.3** System should be secure against SQL injections.

5 Design

5.1 Frontend

For the web frontend design, we have decided to change the mock-up design of sign in and sign up pages after receiving the feedback about its excessive similarity to the Facebook login and sign up page. The biggest change was to separate those 2 functionalities in different pages, although they resided in the same page in our past mock-ups, which resulted in having a different sign in and sign up pages. The other big decision taken after the customer's feedback was to make those pages as simple and efficient as possible. As a result of those decisions, we have a simple yet elegant page design both for login and sign up pages, with sign up and login forms in #f7f7f7 colour and no other material.

Another big design change was the new logo design, which replaced our old logo with a much elegant and simple black and white robin logo, which resided at the left upper corner of the screen.

As part of our frontend design, we have decided to check validity of the inputs in frontend as well, such as password being 8 to 20 characters long and email provided by the guest for the registration being valid. In case those inputs provided were not entirely valid, the submit button - login button for the login page and sign up button for registration page - would be disabled and less bright. When those inputs were corrected and all the inputs are valid - as well as user agreed to our terms and conditions -, then the submit button would be enabled and also be a little more bright.

Finally, we have also designed the error message to be displayed in case user entered wrong credentials to be red, little and simple as well.

For the home page and event page, we have decided to have a fixed navigation bar at the top of the page having search input and some links.

5.2 Android

We selected a base color for android and backend as gray 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. After that we also created a login and register page and designed them considering its mobility of use for android application. regarding email and facebook login respectively. It is all done for now.

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 item	24/10/2018	31/10/2018
Implementing admin panel	31/10/2018	7/11/2018
Milestone	7/11/2018	
Implementing searching, recommendation, rating	7/11/2018	14/11/2018
Implementing W3 annotations	14/11/2018	21/11/2018
Design and implementation improvements	21/11/2018	16/12/2018
Milestone - Project Finalized	17/12/2018	
Testing	17/12/2018	23/12/2018
Debugging	24/12/2018	31/12/2018
Presentation	31/12/2018	

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 created a new branch called “backend” and separated our work from other teams in order to work cleanly. Since we worked on different fragments of code we didn’t encounter any conflicts and with every change, first we tested locally(if it is possible) then pushed it.

7.2 Frontend

We created a branch called ua-frontend as our main branch. For each big feature we created a different branch and when the feature is done we merge this branch with ua-frontend. We also created several branches for working alone and then merge our branches with ua-frontend.

7.3 Android

We created a new branch for android named “android”. We started to work here together. But since we are editing the same files our codes conflicted, unfortunately. We resolved conflicts but we could not make it in time for the presentation.

8 Evaluation of Tools and Management

8.1 Backend:

For backend we've used Django and Django REST Framework. Django REST Framework has a great feature that allows us to explore API endpoints via web browser and it enables us to write serializers for our data models. These features suits us well.

We used rest-auth library and JWT for authentication. As Backend Team we all used VS Code while editing and pipenv for package management.

We are using a Digital Ocean Droplet for deployment. Since we are using pipenv, with just two commands we can update our deployed product.

8.2 Frontend:

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: to 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.

8.3 Android:

For Android we are using Android Studio as ide which is very common for android development. Android Studio allows multiple options and has many useful features for development.

When developing android project we used sdk version 28 of android. There are also some external libraries in our implementation.

- 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**)