CmpE 451
Fall 2019
Milestone 1

Group 4
Bahadır Hocamoğlu
Baturalp Yörük
Cemal Aytekin
Elif Çalışkan
Ege Başural
Emre Demircioğlu
Gürkan Demir
Levent Baş
Muhammed Bera Kaya
Taha Eyup Korkmaz

October 27, 2019

Contents

1 Executive Summary

1.1 Project Introduction

Our project Arken's aim is to establish an environment that users can socialize with each other, get information about any trading equipment, read articles and find out about economic events and also share their ideas about economic events and articles. There are three types of users in this project: Guest user-not registered-, Basic user and Trader.

Guest users can get information about any trading equipment, event or article. They can search users, trading equipment, events or articles. Basic users can make comments, chase events, create portfolios, rate articles, manually enter investments, make predictions about any trading equipment and follow users in addition to guest user's abilities. Their prediction rate for any equipment can be seen by other users. A user's profile can be private. Traders can make investments onsite with the IBAN number that they provide. Every user has profit/loss section and it shows the user's profit based on user's transactions with respect to the chosen trading equipment. Users can make annotations about events, trading equipment, articles etc.

Events have significance levels which represents their importance. Also events can have numeric values about the related trading equipment. Events can be filtered based on their significance level or country that it belongs. Articles can be rated, commented and shared by users.

Trading equipment should have functionalities such as: the previous close, percentage change with the percentage close, amount change with the previous close, day's range and moving averages.

The system provides semantic search and makes recommendations about articles or trading equipment based on their user history.

1.2 Work Done So Far

We have already designed our mockups, class diagram, sequence diagrams and use case diagram in the previous course, CmpE352. We also have analyzed the requirements then, so we only had to review all of those and update them wherever we see necessary. We also had to reiterate over our project plan and add milestone dates, rearrange tasks accordingly. Now, our diagrams, requirement analysis and project plan are up-to-date and we are referring them whenever we stuck somewhere during our project development.

For that milestone, we scheduled the implementations of login-signup for users, designing of our homepage, serving event contents to our users and viewing available trading equipments in our system. Those tasks were involved in all of our three teams: backend, frontend and mobile, and we separated them into subtasks and assigned to appropriate teams. Our frontend and mobile teams couldn't implement viewing trading equipments feature, due to some technical difficulties. They've overcome the issues related to those now. On the other hand, our backend team implemented all the features scheduled in the first milestone and helped the other teams as far as they could. Our project is in a good shape right now and things are going according to plan, with some small delays on some parts however.

1.3 Planned Changes

We may need do change our requirements and project plan in the future in order to make our app compliant with Personal Data Protection Law. We will discuss what can be done about that in the nearest meeting. We will also need to reschedule the missing features on frontend and mobile in the first milestone to some future date. We may also make our project plan a little bit more detailed before the next milestone, which is not a must, but that might help us assign subtasks more easily and continue our development process in a more organized manner.

2 List and status of deliverables

Deliverable	**Status**
Login & sign up	Done
Economic events	Done
Trading equipments	Partially done

Table 1: List and status of the deliverable in Mobile

Deliverable	**Status**
Login & sign up	Done
Economic events	Done
Trading equipments	Done
Database instance for production	Done
Database instance for development	Done
Online API documentation	Done
Deployment	Done

Table 2: List and status of the deliverable in Backend

Deliverable	**Status**
Login & sign up	Done
Economic events	Partially done
Trading equipments	Not done
Deployment	Done

Table 3: List and status of the deliverable in Frontend

3 Evaluation of the status of deliverables and its impact on project

Login & Sign up: The first deliverable was implementing login and sign up functionalities. We had already started implementing some functions in CmpE352. In this course we added email verification, signing via Google account and using map to find the location. First, backend group completed their work and informed other groups about the endpoints. Then, frontend and mobile groups implemented their parts. Since this was our first deliverable, unacquainted group members learned the environment and the feature was completed in an organized way. After the implementation, a user can sign up as a trader or basic user, log in to our platform. While signing, the user can use Google account and can choose his/her location with map.

Economic Events: We had a similar functionality in CmpE352. In this course, we changed the design in frontend and implemented the mobile part. This part is partially done since in frontend, the events are not clickable. But this will be fixed in a short amount of time. After completing, the user can see economic events, their importance values and the country they belong. The event page is the home page and after logging in or going on as guest user, this page is shown. After clicking on any event, the user can see specific values such as source, actual value, forecast.

Trading Equipment: This is partially implemented in mobile part and not implemented in frontend. In mobile the user can only see listed currency pairs and their current values, but when the implementation is finished, user will be able to view the details of each currency pairs and their weekly, monthly, yearly values and graphs. If the user is not a guest, he can access to follow, comment and make prediction functionalities both in mobile and frontend.

4 Summary of coding work done

Member Name	**Work Done**
Bahadır Hocamoğlu	Initiated our backend repository. Added endpoints about user profiles. Added session-based authentication along with login and signup. Designed several database models and implemented middlewares to make accessing them easier. Added pagination to events endpoint. Created our AWS instance, deployed our backend and frontend continuously, deployed our development and production databases on AWS with Docker. Added a few middlewares to prevent code-duplication and ensure code reusability.
Baturalp Yörük	Since I was not familiar with android before, I worked on it to catch up as soon as possible and started with relatively small tasks. Firstly, I have done the password validation part by adding another password editbox. Also, I implemented the forgot password part. To do that, I created a new fragment for the page of it and added a button to the login part. At the forgot password page, I send user's email with POST method to the backend and user receives an email.
Cemal Aytekin	Initiated our git branch structure. Added Google Authentication along login and signup. Added logout endpoint and implemented logout system. Created verification mail template. Added time interval function to get events data periodically from 3rd party API. Added time interval function to get trading equipment data periodically from 3rd party API. Organized index.js file.
Elif Çalışkan	Initialized android branch and created the draft of login and signup pages. Added Google sign up functionality. Created navigation graph for the part after signing in. Changed the design of event list. Added a menu which consists of events trading equipment and log out under base fragment. Added map to google sign up fragment. Added password show icon to log in and sign up pages. Added log out functionality using Shared Preferences.

Ege Başural	Initialized React, React-router, react-redux for frontend application. Created the basic code structure. Created http service to make proper requests to the server. Configured redux infrastructure. Connected responses to an interceptor to be able to print out errors as an error message in the screen. Created header, containing static component, user component and guest component. Configured background of the application. Created sign in modal, connected it to database. Created Google sign in API Key from Google, implemented it to the project. Created sign up page. Implemented Google Maps location picker to Sign Up page. Created another sign up page for first log in of Google Sign In.
Emre Demircioğlu	Implemented initial signup and login data models to fix retrofit callback which is connects our app to API. Implemented listing events and trading equipment and event details pages including their fragments, data models and API connections. Implemented recyclerview, adapter and viewholder for listing events and trading equipment. Added first version of navigation property that helps switching between fragments. Fixed events fetching according to pagination. Added functionality of changing location via click on google maps. Worked as a member of android team.
Gürkan Demir	Implemented login, signup functionalities with other members in Backend team. Add validation and verification functions for password, IBAN, TCKN, email etc. Implemented verify account functionality. Created new endpoint for forget password, reset password with their email structure. Implemented endpoint for events which return all events in database and also specific event in database with their extra fields. Implemented endpoint for trading equipments which return all currencies' last values and also historical value of given equipment. Implemented follow, unfollow functionality for users to chase needed trading equipments.

Levent Baş	I started learning about fragments and general workflow of Android development since I was relatively new to the area. I added Google Maps integration using a Maps activity so that the user can find his/her location when signing up via Google Maps API. I also parsed the latitude and longitude returned by the API and showed the corresponding city and country. I added private profile functionality while signing up using a switch. I parsed the error messages returned by our backend team in signup and login pages so that error messages show exactly what needs to be done by the user for the errors to be fixed.
Muhammed Bera Kaya	I did not know much about JavaScript and frontend development with ReactJS. Firstly , I started with learning about them. Then, I implemented the ”Events” page where all events are listed in a table with pagination by using the data returned by our API(”events” endpoint).I added a new route for this page.I added a menu item to our website header which links to this page.
Taha Eyup Korkmaz	I wasn't very familiar with Javascript and ReactJS, so I asked from Ege to give us a quick take on what's going on. After that I started studying it by myself and I took the basic task of implementing the Profile Page. The user credentials was provided from our backend's user endpoint. I used the button at Header since Ege created it before and I made it connect to a new page that has the Profile information which changes if the user is Trader or not. Now I'm adding the Profit/Loss and My Portfolio segments with the follow button.

Table 4: Summary of work done by each team member

5 Requirements

5.1 Glossary

- **Application:** The whole project as seen by guests, members and users.
- **Article:** Text written by user about trading equipment.
- **Basic User:** Signed up user who can't perform investments inside the application, can be chosen during sign up.

- **Comment:** A body of text, image or links shared in response to an article.
- **Database:** An organized collection of data.
- **Day's Range:** The range in which a trading equipment traded. For example, a day's range that reads "48.50-51.25" means that the lowest price the trading equipment reached that day was $48.50/share and the highest price was $51.25.
- **Economic Event:** Events about economy that occurred and posted inside the application, will have a numeric result.
- **Followed User:** A user who accepted other users following request.
- **Following User:** A user who's following request has been accepted by other user.
- **Guest:** Unregistered person.
- **Moving Average:** A succession of averages derived from successive segments (typically of constant size and overlapping) of a series of values.
- **Profile Page:** A page where user can see information about user and additionally change the information if it's their own profile page.
- **Prediction Rate:** A rate showed on user's profile page, calculated by their past predictions.
- **Previous Close:** What the price of a trading equipment was when the market closed on the previous trading day.
- **Semantic Search:** A prediction system to improve search accuracy by understanding the searcher's intent and the contextual meaning of terms as they appear in the searchable dataspace.
- **Significance Level:** A rating system that describes the importance of an event.
- **System:** The whole project, including design and functionalities.
- **Trading Equipment:** Indices, stocks, ETFs, commodities, currencies, funds, bonds and cryptocurrencies.
- **Trading User:** Signed up user who can make investments inside the application, can be chosen during sign up.
- **User:** Registered basic user or trading user.

5.2 Functional Requirements

5.2.1 User Requirements

5.2.1.1 Guests

- **5.2.1.1.1** Guests shall be able to search economic events, articles and trading equipment.
- **5.2.1.1.2** Guests shall view the price of trading equipment.
- **5.2.1.1.3** Guests shall read comments about trading equipment.
- **5.2.1.1.4 Registration**
  - **5.2.1.1.4.1** User shall be able to choose between basic user and trading user.
  - **5.2.1.1.4.2** Basic user shall provide name, surname, email and password.
  - **5.2.1.1.4.3** Trading user shall additionally provide IBAN and TC identification number.
  - **5.2.1.1.4.4** User shall provide location with Google Maps.
  - **5.2.1.1.4.5** User shall validate account via e-mail.
  - **5.2.1.1.4.6** Users shall be able to register via their Google accounts.

5.2.1.2 Registered Users

5.2.1.2.1 Login

- **5.2.1.2.1.1** User shall login via email and password provided upon registration.
- **5.2.1.2.1.2** User shall login via Google account.
- **5.2.1.2.1.3** Users shall be able to reset their passwords if they forget their passwords by clicking ”Forget password?” button.
- **5.2.1.2.1.4** Users shall be able to logout.

5.2.1.2.2 User Follow System

- **5.2.1.2.2.1** User shall be able to follow other users.
- **5.2.1.2.2.2** User shall be able to send following request to other users who have private profile.
- **5.2.1.2.2.3** Users shall be able to accept or reject following requests.
- **5.2.1.2.2.4** User shall be followed user upon accepting following request.

5.2.1.2.3 Trading Equipment

- **5.2.1.2.3.1** User shall be able to follow trading equipment.
- **5.2.1.2.3.2** User shall be able to set alerts for certain levels of trading equipment.
- **5.2.1.2.3.3** User shall be able to comment on trading equipment.
- **5.2.1.2.3.4** Trading user shall be able to invest in trading equipment.

5.2.1.2.4 Profile

- **5.2.1.2.4.1** User shall have a profile page.
- **5.2.1.2.4.2** Profile page shall have the prediction rate of user.
- **5.2.1.2.4.3** User shall be able to choose to be public user or private user.
- **5.2.1.2.4.4** Private users profile page content other than prediction rate shall be seen only by following users.
- **5.2.1.2.4.5** Public users profile page content shall be able to be seen by all other users.
- **5.2.1.2.4.6** Users shall see and edit their personal information in profile page.
- **5.2.1.2.4.7** Users shall view their old actions on their profile page.
- **5.2.1.2.4.8** Users shall be able to reach their own followers and followings list in their profile page.

5.2.1.2.5 Articles

- **5.2.1.2.5.1** User shall be able to share ideas about trading equipment as articles.
- **5.2.1.2.5.2** User shall be able to comment on articles.
- **5.2.1.2.5.3** User shall be able to rate articles.

5.2.1.2.6 Portfolio

- **5.2.1.2.6.1** User shall have at least one portfolio.
- **5.2.1.2.6.2** User shall be able to have different portfolios.
- **5.2.1.2.6.3** User shall be able to rename portfolio.
- **5.2.1.2.6.4** Users shall be able to add or remove trading equipments from/to portfolio.

- **5.2.1.2.6.5** User shall be able to share portfolio in profile page.
- **5.2.1.2.6.6** Other users shall be able to follow other user's portfolio.
- **5.2.1.2.6.7** Users shall be able to create/delete their portfolios.

5.2.1.2.7 My Investments Page

- **5.2.1.2.7.1** Trading users shall have my investments page.
- **5.2.1.2.7.2** Basic users shall not have my investments page.
- **5.2.1.2.7.3** Trading users shall be able to invest on trading equipment in my investments page.
- **5.2.1.2.7.4** Trading users shall be able to create a buy order for a trading equipment for a specified rate in my investments page.
- **5.2.1.2.7.5** Trading users shall be able to set stop/loss limits on trading equipment in my investments page.

5.2.1.2.8 Profit/Loss Section

- **5.2.1.2.8.1** Users shall have a profit/loss section.
- **5.2.1.2.8.2** Profit/loss section shall be private to each user.
- **5.2.1.2.8.3** User shall be able to see profit/loss in terms of currency chosen by user.
- **5.2.1.2.8.4** Users shall be able to manually enter investments to see calculated profit/loss.
- **5.2.1.2.8.5** Profit/loss section should include investments made in the platform by a trading user when calculating profit/loss.

5.2.1.2.9 Events

- **5.2.1.2.9.1** Users shall be able to see events fetched from third parties.
- **5.2.1.2.9.2** Users shall be able to comment on events.
- **5.2.1.2.9.3** Users shall be able to filter events by its currency, significance level, location etc.
- **5.2.1.2.9.4** Users shall be able to set an alarm to a specific event

5.2.2 System Requirements

5.2.2.1 Trading Equipment

- **5.2.2.1.1** System shall provide following functionalities for a trading equipment:
  - **5.2.2.1.1.1** Previous close
  - **5.2.2.1.1.2** Percentage change compared with previous close
  - **5.2.2.1.1.3** Amount change compared with previous close
  - **5.2.2.1.1.4** Day's range
  - **5.2.2.1.1.5** Moving averages

5.2.2.2 Recommendation & Notification

- **5.2.2.2.1** System shall recommend users, portfolios, articles and trading equipment to users based on their history in the system.
- **5.2.2.2.2** System shall send notification to users according to their alerts.
- **5.2.2.2.3** System shall notify trading users according to a transaction which is made because of stop/loss limit.

5.2.2.3 Search

- **5.2.2.3.1** System shall provide searching for users, trading equipment and economic events.
- **5.2.2.3.2** System shall support semantic search.
- **5.2.2.3.3** System shall sort search results by their significance level and location.
- **5.2.2.3.4** System shall also sort users according to their prediction success rate.

5.2.2.4 Events

- **5.2.2.4.1** System shall have events page which contains economic events.
- **5.2.2.4.2** Events shall be able to have different significance levels
- **5.2.2.4.3** Events shall have numerical results.

5.2.2.5 User Authentication

- **5.2.2.5.1** A confirmation link shall be sent to the users' email address for authentication.
- **5.2.2.5.2** After clicking the confirmation link, users shall be registered to their account and logged in automatically.

5.3 Nonfunctional Requirements

5.3.1 Security and Reliability

- **5.3.1.1** User data shall be processed according to 'Law on the Protection of Personal Data'.
- **5.3.1.2** User's password shall be stored with encryption in the database.
- **5.3.1.3** User shall get notification emails when changing their password.
- **5.3.1.4** Weekly backups shall be taken in order to ensure data is safe and sound.
- **5.3.1.5** In any case of server failure or any other need, the system shall be restored with any backup.

5.3.2 Performance

- **5.3.2.1** The system should be able to respond to requests within 3 seconds.
- **5.3.2.2** At least 150 requests per second should be responded.
- **5.3.2.3** The system shall be able to serve at least 100000 members.

5.3.3 Availability

- **5.3.3.1** The system shall be accessible on both native Android and web platform.
- **5.3.3.2** The system shall support English language
- **5.3.3.3** The system should be available 99% of the time.
- **5.3.3.4** In the case of failure, the system should recover in at most 30 minutes.
- **5.3.3.5** The system shall support Turkish characters.

5.3.4 Annotations

- **5.3.4.1** The annotations shall be designed according to W3C web annotation data model.
- **5.3.4.2** The annotations shall be tested by the test team to ensure that they work correctly.

5.3.5 Database

- **5.3.5.1** User data shall be held in a secure database.
- **5.3.5.2** Database hierarchy shall be well-constructed so that it shall be efficient and protect the user's privacy.
- **5.3.5.3** All changes shall be logged in a database.

6 API Documentation

The domain of our API is api.dev.arkenstone.ml.

6.1 Authentication

Functionality	**URI**	**Method**	**Parameters**
Sign up	/auth/signup	POST	name, surname, email, password, location
Login	/auth/login	POST	email, password
Verify email with token	/auth/verify	GET	token
Forget password	/auth/forget-password	POST	email
Reset password	/auth/reset-password	POST	token, password
Logout	/auth/logout	GET	–

Table 5: Authentication Endpoints

6.2 User Profiles

Functionality	**URI**	**Method**	**Parameters**
Retrieve profile details by user ID	/profile/:userid	GET	–

Table 6: User Profile Endpoints

6.3 Events

Functionality	**URI**	**Method**	**Parameters**
List events with pagination and filtering	/events	GET	page(default: 1), limit(events per page, default: 10), importance(optional), country(optional)
Retrieve event details with calendar ID	/events/:calendarID	GET	–

Table 7: Event Endpoints

6.4 Trading Equipments

Functionality	**URI**	**Method**	**Parameters**
List trading equipments last values	/trading-equipments	GET	–
Retrieve given currencies' historical data	/trading-equipments/:currency	GET	–
Follow trading equipments	/trading-equipments/follow	POST	tEq
Unfollow trading equipments	/trading-equipments/unfollow	POST	tEq

Table 8: Trading Equipments Endpoints

7 Project Plan

Team	**Members**
Backend	Bahadır Hocamoğlu, Cemal Aytekin, Gürkan Demir
Frontend	Ege Başural, Muhammed Bera Kaya, Taha Korkmaz
Mobile	Baturalp Yörük, Elif Çalışkan, Emre Demircioğlu, Levent Baş

Table 9: Members of Each Subgroup in Our Team

Task	**Time Reserved**	**Start**	**Due Date**	**Assigned to**
Login-signup @backend	5 days	30/09/2019	04/10/2019	Backend team
Homepage & events @backend	5 days	07/10/2019	11/10/2019	Backend team
Viewing trading equipments @backend	5 days	14/10/2019	18/10/2019	Backend team
Login-signup @frontend	5 days	30/09/2019	04/10/2019	Frontend team
Homepage & events @frontend	5 days	07/10/2019	11/10/2019	Frontend team
Viewing trading equipments @frontend	5 days	14/10/2019	18/10/2019	Frontend team
Login-signup @mobile	5 days	30/09/2019	04/10/2019	Mobile team
Homepage & events @mobile	5 days	07/10/2019	11/10/2019	Mobile team
Viewing trading equipments @mobile	5 days	14/10/2019	18/10/2019	Mobile team
Deployment	1 day	21/10/2019	–	Bahadır Hocamoğlu

Table 10: Milestone 1

Task	**Time Reserved**	**Start**	**Due Date**	**Assigned to**
Search-comment-prediction @backend	10 days	23/10/2019	05/11/2019	Backend team
User profile @backend	5 days	06/11/2019	12/11/2019	Backend team
Portfolio @backend	5 days	13/11/2019	19/11/2019	Backend team
Follow system @backend	4 days	20/11/2019	25/11/2019	Backend team
Search-comment-prediction @frontend	10 days	23/10/2019	05/11/2019	Frontend team
User profile @frontend	5 days	06/11/2019	12/11/2019	Frontend team
Portfolio @frontend	5 days	13/11/2019	19/11/2019	Frontend team
Follow system @frontend	4 days	20/11/2019	25/11/2019	Frontend team
Search-comment-prediction @mobile	10 days	23/10/2019	05/11/2019	Mobile team
User profile @mobile	5 days	06/11/2019	12/11/2019	Mobile team
Portfolio @mobile	5 days	13/11/2019	19/11/2019	Mobile team
Follow system @mobile	4 days	20/11/2019	25/11/2019	Mobile team

Table 11: Milestone 2

Task	**Time Reserved**	**Start**	**Due Date**	**Assigned to**
Investment & profit loss @backend	5 days	27/11/2019	03/12/2019	Backend team
Notification & recommendation systems @backend	5 days	04/12/2019	10/12/2019	Backend team
Annotation @backend	5 days	11/12/2019	17/12/2019	Backend team
Investment & profit loss @frontend	5 days	27/11/2019	03/12/2019	Frontend team
Notification & recommendation systems @frontend	5 days	04/12/2019	10/12/2019	Frontend team
Annotation @frontend	5 days	11/12/2019	17/12/2019	Frontend team
Investment & profit loss @mobile	5 days	27/11/2019	03/12/2019	Mobile team
Notification & recommendation systems @mobile	5 days	04/12/2019	10/12/2019	Mobile team
Annotation @mobile	5 days	11/12/2019	17/12/2019	Mobile team

Table 12: Final Milestone

8 User Scenarios

A user persona is a fictional representation of your ideal customer. A persona is generally based on this user research and incorporates the needs, goals, and observed behavior patterns of your target audience. We presented 3 user scenarios in the first milestone, that cover all the features that have developed so far.

8.1 Story I: Berke Esmer - A Student of Economics

8.1.1 Demographics:

- Boğaziçi University Economics
- 23 Years old
- Works at Boni Technologies

8.1.2 Goals:

- He wants to check any economic events

8.1.3 Scenario:

- As an economics student, Berke wants to check any economic events. He has heard about our app. He wants to try our app. He signs up and verifies his email. Then he logs in to our website. After that, he checks out the new economic events. Finally, he logs out.

8.2 Story II: Berke Esmer - Professor in Economics

8.2.1 Demographics:

- Boğaziçi University Economics - Bachelor
- University of California - PhD
- 54 Years old
- Works at Boğaziçi University
- Registered to System as a Basic User

8.2.2 Goals:

- He wants to reset his password

8.2.3 Scenario:

- Berke is now an old economics professor. He forgot his password and wants to reset his password so that he can access his account again. He clicks the forgot password button. He writes his email and clicks reset password button. Then he checks his email and clicks the link sent. After that, he writes his new password and confirms it, then clicks reset password. Finally, he logs in and looks at his profile.

8.3 Story III: Emre Can - A Young Waiter

8.3.1 Demographics:

- Adana Vocational and Technical Anatolian High School
- 27 Years old
- Works at Mutfak Cafe
- Registered to System as a Trader User

8.3.2 Goals:

- He wants to make investments as a trader
- He wants to follow the events and make trades accordingly

8.3.3 Scenario:

- Emre is a young waiter who is unfortunately not rich. He has heard about our app. He wants to take look and see if this app helps him increase his income. He clicks the Go On As A Guest button. He looks at Economic Events page and clicks on one event. After that he thinks our app is very helpful for his goal and wants to register. He signs up via his Google Account. Now, he checks out Trading Equipment page.

9 Evaluation of Tools and Managing the Project

9.1 Mobile

The communication between team members is very active and we are using Slack to communicate. We are keeping track of the works by Github and helping each other via Slack or by having some meetings face-to-face. Furthermore we meet each week, go over what has been done so far and get each other's opinion in order to make our app user friendly and convenient.

9.1.1 Android Studio

Android Studio is very easy to use since it has a preview property in layout and shape resources. Since it has a version control inside, it is more convenient to use Git in Android Studio rather than using terminal or another GUI.

9.1.2 Java/Kotlin

Java is a language which we are all were familiar with, from our freshman year. This fact was important for us to use it. Kotlin has some advantages compared to Java, so we are also using it in our project.

9.1.3 Gradle

Gradle is an important tool for android project. It is actually a build tool that collects all dependencies according to their versions that used in the project. It is very effective to use different android packages in the project.

9.1.4 Navigation - RecyclerView - Retrofit

Navigation is an android tool that helps fragment switching and parameter passing between fragments. Recyclerview is used in the listing pages. It helps customizing row view and data binding. Retrofit is used for API calls and json mapping to data models via callback methods.

9.2 Backend

The backend development in our project has been very smooth so far, actually. Our backend team was familiar with the technologies we used, as we used them in the previous course, CmpE352, as well. The backend channel in our Slack group was very active, always discussing any difficulties they faced and searching for solutions, helping each other by recommending various resources, etc.

9.2.1 NodeJS/ExpressJS

We have used ExpressJS framework in our backend, as it's very easy to learn and extensible, thanks to its minimalistic and unopinionated nature. We have added some small libraries for input sanitation, external API access, sending email, etc. and it was very easy to integrate them into our application.

9.2.2 MongoDB

MongoDB is a NoSQL library, using collections and JSON documents instead of tables and table entries. The documents aren't restricted to follow a certain predefined structure, which is the case for any SQL database, and that helps us develop and modify our database without worrying about data migrations and compatibility issues. We are using MongooseJS ORM library to access our database, which makes it even easier to work with MongoDB, as it provides lots of predefined schema types, validation and middleware options.

9.3 Frontend

The frontend team uses a Slack channel for team communication. The progress and the work done by team members are tracked via GitHub. The Slack channel was used actively in helping other team members and asking for help,since most of the team was not familiar with frontend development and needed help of other team members.

9.3.1 ReactJS

ReactJS is an open-source JavaScript library which is used for building user interfaces specifically for single page applications. ReactJS is chosen as frontend framework by the team, because of previous familiarity and it's ease of learning. It's basic structure of state, props and life cycle methods makes it easier to understand and start developing quickly. Component based system helps developers creating more readable code.

9.3.2 Webstorm

As frontend team, we didn't decide on a IDE to use. However, we are all using WebStorm as our frontend IDE. It is a JavaScript IDE, with git integration and version control functionalities. Main advantages of it includes it's conflict solver, which creates a great advantage in multi member projects.

9.3.3 Redux

Redux is a predictable state container for JavaScript apps, as it's documentation says. It is a framework that can also be connected to React, that makes as able to pass on data and actions between components of React, which is a lot of pain without frameworks like it.

9.3.4 Semantic UI React

We have chosen Semantic UI React between other UI frameworks, because of it's ease of use and simplicity. We didn't need complex working components for our project, so UI framework is mostly used for the design.

9.4 Managing the Project

9.4.1 Code Structure

As a whole team, we were very consistent throughout the project with how we implement new features and how we deal with bugs and issues. We created branches for each feature and assigned enough number of people to implement that feature. After writing out the code locally, we opened a pull request and assigned our team members to review the code we have written regarding the corresponding feature. If all of the assignees approved the pull request, we merged the branch into our development branch (android-dev, backend-dev, frontend-dev).

Whenever a bug was found after a pull request was approved and the corresponding branch was merged, we created another branch to fix the bug and went through the same steps as explained above.

9.4.2 Communication

Group communication was a determining factor for our success in delivering the expected results so far. Backend team, frontend team and mobile team all used necessary communication means to deal with issues and decide how they should be organized as a sub-team to deliver their own part. However, internal communication in sub-teams was not enough, all of the sub-teams were also in contact with other sub-teams and knew what the others were doing so that they were in synch as a whole group. Each sub-team had its own channel in Slack where the group members of related sub-teams were discussing their implementation details and each sub-team was allowed to monitor the communication between members of other sub-teams.