CMPE-451 Milestone 1 Report
Group 5

Members
Kayacan Vesek
Emre Hoşer
Volkan Bulca
M.Zeynep Çayırçimen
Mısra Yavuz
Sertay Akpınar
İsmet Sarı
Yaşar Selçuk Çalışkan
Muhammed Halas
Yusuf Yüksel
Algı Kanar
Ramiz Dündar

Contents

1 Executive Summary

1.1 Project Status

This year, new members joined to our group. Before starting implementing the ecommerce platform, we arranged a meeting and explained them what we have done in our project so far on CmpE352 course. In our first group meeting, we determined our development teams: frontend, backend and mobile (Android). Also, although we have finalized our project requirements and project plan during CmpE352 course, we decided to go through and update them. We have given some Customer milestones and we set some internal milestones for each team, and make some updates on project plan accordingly.

Until our first customer milestone, we have implemented the user register and login functionalities in both backend and android. There is also a password change functionality on backend as well. Backend team has also implemented the feature for users to change their personal information. The design of the homepage, search bar and category bar, login and register pages are done by the frontend team. User profile page is also implemented, from which the users can see their personal information and change their password.

1.2 Moving Forward

The first job that we will do is implementing the missing or inadequate features of the first customer milestone. This includes adding Google login feature and implementing verification email send functionality. We will do some research on KVKK and GDPR terms and policies and implement that on our user register page as soon as possible. On frontend side, we will separate the customer and vendor profile pages and put information specific to those user types. Also, pages for personal information change functionality will be added. Android team will implement the profile pages. Those should be added to the project plan, therefore a small update on project plan is needed.

After we are done with the quick changes relating to the first milestone, we will immediately start working on the tasks that we have determined to finish until second customer milestone. The jobs that are planned to be done until the second customer milestone can be found in our updated project plan on our GitHub page.

2 List and Status Deliverables

Deliverable	**Status**
Login & Sign up	Done
Home Page	Done
Profile Page	Not Done

Table 1: List and status of the deliverable in Mobile

Deliverable	**Status**
Login & Sign up	Done
User Profile	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 Back-end

Deliverable	**Status**
Login & Sign up	Done
Home Page	Done
Profile Page	Partially Done
Deployment	Done

Table 3: List and status of the deliverable in Front-end

3 Evaluation of the Status of Deliverables

Login & Signup:

All teams implemented their login and signup functionalities. Frontend and mobile team firstly started to implement the pages without any functionality. After backend team developed the user endpoints, frontend and mobile team established the connection to backend via requests. Vendor signup is also completed.

Home page:

Since we haven't developed the product component yet, frontend and mobile teams only worked on a visual home page with mock data. Only functionality of the home page for now is to have a page to redirect the customer after login, and also a page to redirect from to go to the profile page. Still, we believe that, with any website development, seeing a home page as what it would be like in the end is crucial, even if the page is static for now.

User profile:

Frontend team implemented a profile page for customers, and is working on the vendor profile page. Profile page includes menu items that only one of them works for now, which is to change password option. This option redirects to the change password page, which is connected to backend and works just fine as presented during the customer milestone presentation. Mobile team hasn't developed a profile page yet. Backend team developed all the user profile endpoints, which are not fully included in frontend and mobile. These endpoints include changing basic user informations such as name, surname, address and etc.

Deployment:

Backend and frontend platforms are dockerized and deployed on AWS EC2 instances. They are on separate instances for now, to prevent any crushes that would lead to no working deployed app at all. CI/CD pipeline has not been completed yet, but we are working on it.

4 A Summary of Coding Work Done by Each Team Member

Member Name	Contributions
Kayacan Vesek	- I implemented navigation between pages, and give functions to buttons. - I created the missing boxes in register xml. - I created a RestApiService class to send request to backend service. - I created the methods and classes that send and take response from backend with Emre Hoşer. - I have created pull request #159 (navigation between pages) #161 (Register Page Last Version) #162 (Merge to master).
Emre Hoşer	- I initialized the Android project according to MVP design architecture. - I have created the mobile-init branch and work on this branch during the development. - I created Base classes and interfaces could be used in project. - I generated some design template, buttons and main colors. - I implemented Login and Register .xml file. - I implemented the Login fragment and Login activity classes. - I created the methods and classes that send and take response from backend with Kayacan Vesek. - I added the some of picture, description on home page with other teammates.
Volkan Bulca	- I have implemented the /api/auth/register endpoint of API. - I have implemented the view and the serializer of the user registration. - I have not created an additional branch, but I worked under the common backend branch and pushed my commits there. - I reviewed the pull requests #120 and #124. - I worked together with İsmet during the deployment of the frontend to EC2. - I worked together with İsmet during the Google authentication; however, we failed to make it work properly, so we postponed it.
M.Zeynep Çayırçimen	- I have implemented the home page with footer and navigation bar with Mısra and Algı. - I merged branch 'profile-page' into password-change-page - I reviewed pull requests #116, #122, #141 - I created pull requests #128, #156 - I have implemented user profile page with dynamic name + surname using local storage and fetch data from backend with Algı and Mısra
Mısra Yavuz	- We worked on the home page together with Zeynep and Algı, because we thought the visual looks (colors and UI components) should not be a decision of one person, rather decided together. - I worked on the profile page with Zeynep, implemented the backend connection and dynamic username update specifically. - I implemented the change password page and the connection to backend to update the password. - My commits were mostly from password-change-page and profile-page branches. - I reviewed pull requests #116, #122, #130, #141, #151, #156.

Member Name	Contributions
Sertay Akpınar	- I have implemented everything but recycler view in the homepage. (.xml file) - I have assisted to Yaşar while giving the final look to the home page. - I have created the mobile-sertay branch.(worked in this branch during the development). - I have created issues #139 and #163. - I reviewed the pull request #160 and #159 merged it to the mobile-init branch. - I presented the mobile scenario to the customer.
İsmet Sarı	- I have implemented both the views and serializers of the password change. - I have made contribution to initiate the first server which has no functionality. - I have deployed both frontend and backend to EC2 instances. - I have written Dockerfile and dockerignore files of frontend and backend. - I have not created a branch, I just pushed my commits to our common branch called backend and merged it to master branch with pull request number #155. - I have reviewed three pull requests: #148, #124, #120 - I got keys for Facebook and Google authentications then added these keys to Django-superuser.
Yaşar Selçuk Çalışkan	- I have implemented the homepage .xml file. - I have implemented the recyclerView that takes place in the homepage. - I have implemented HomepageActivity.kt file and created random products. - I have created the Product class to represent a basic product in the app. - I have implemented the ProductAdapter.kt file to be able to inflate the recyclerView with products. - I have created the mobile-yasar branch and worked in this branch during the development. - I have created and merged the pull request #160.
Muhammed Halas	- I helped the coding of the initial server. - I helped the bug-fixing of the register feature. - I reviewed pull requests.

Member Name	Contributions
Algı Kanar	- I have helped during implementation of home page dealing with footer and navbar. - I have helped user profile page fetching data from backend. - I additionally made changes in home page navbar's log in and sign up button with their appropriate links and created related render components. - I made changes in the home page listing to show name and price along with giving necessary images. - I made the commits through profile page branch. - Merged issue #151. (login connect backend) - I reviewed the issue #156.
Ramiz Dündar	- Initialized project and created basic folder structure. - Created login, user sign up and vendor sign up pages. - Made validation system for these 3 pages. - Connected them to backend and created related async error messages. - Assisted in frontend deploy. - Dockerized frontend with Ismet. - Reviewed and merged all pull requests in frontend other than mine. - Reviewed one backend pull request (#148). - Created always a new branch for a feature and then created pull request. Note that some branches may be deleted by reviewer after merge operation. - Presented frontend scenario in milestone 1.
Yusuf Yüksel	- I initialized project and created folder structure. - I created MongoDB via MongoDB Atlas and made connection between backend and MongoDB. - I created User, Customer, Vendor models and implemented create_user_account() function to create all types of users easily. - I implemented /api/users and /api/users/{id} user endpoints of API with their corresponding views and serializers. - I implemented /api/auth/login, /api/auth/logout, /api/auth/user_info and /api/auth/profile_update authentication endpoints of API with their corresponding views and serializers. - I integrate swagger and redoc with /api/swagger and /api/redoc endpoints to API to document it. - I created our common backend branch, pushed commits there. - I created pull requests #120, #124, #148. - I reviewed pull request #151, #155.

5 Deployment, Dockerizing, and CI/CD processes

We have accomplished to deploy both the front-end and the back-end platforms on the Amazon EC2 instances. First, we installed the docker on our local machine and tried to write a docker file. For the front-end platform, we first import the minimal nodejs v15.2.1 image built-on Alpine-Linux distribution by using FROM command of docker. We established a working directory named 'frontapp' and copy all the files and directories into this working directory. RUN and CMD commands of docker will be executed in this working directory so we install all the dependencies here and run the app by listening port 3000. For the back-end part we used the same idea. We used again Alpine-Linux distribution by using python:3.9-alphine. This time we say docker that there is no need to cache gcc and Alpine-Linux packages like musl-dev or python-dev. We set the working directory named 'bupazarapp' and copied all the things in the directory that the Dockerfile exists. We write some commands that makes the docker image we created install all the dependencies written in the requirements.txt file and migrate all the migration after making them. After all we run the app by listening this time port 8000. The docker images were perfectly run in our local machine so we pushed these two images to Dockerhub after setting some unique tags.

After dockerizing part we created EC2 instances and they have run perfectly. However when we tried to run front-end docker image in the same EC2 instance it did not work since front-end was listening port 3000 and it did not match with the EC2 instance. Once we realize our mistake we decided to create an other instance to deploy our front-end platform. We could update the first EC2 instance but this way looked more secure since if one of the two instance crushes then the other one can continue to run. The two docker image were already in the Dockerhub we just set the port and run the image in the two EC2 image.

CI/CD have not completed yet but it is in progress and will be released soon.

6 Requirements and Changes

6.1 Changes

Here are the changes we have done to project requirements in this semester:

- **1.1.1.1.4.** Guest user shall be able to sign up while the purchase process.
- **1.1.2.1.1.** Customers shall be able to rate the products they bought, out of 5.
- **1.1.2.1.2.** Customers shall be able to rate the vendors whom they bought a product from, out of 10.
- **1.1.2.2.1.** Users shall be able to comment on the products that they have already bought.
- **1.1.2.2.2.** Users shall be able to choose to anonymise their comment prior to posting.
- **1.1.2.2.3** Admins shall be able to delete inappropriate comments.
- **1.2.1.1.4.** System shall verify vendors when they sign up. They have to add their national id number.

6.2 Requirements

Glossary

- **Amazon-EC2:** by allowing users to rent virtual computers on which to run their own computer applications.
- **Admin User:** A person who is responsible for system sustainability and management in general
- **Guest User:** A person who does not have an account and have restricted access to the application
- **Vendor:** A person who supply products to customers.
- **Search:** A tool to help the users find the relevant contents for given input words in the application
- **Server:** something that shares data or resources among multiple clients or performing computation for a client
- **Sign In:** Entering to the application by providing correct email and password
- **Sign Up:** Creating an account to be a member of application
- **User:** A person that interacts with the application
- **Docker:** It is a tool designed to make it easier to create, deploy, and run applications by using containers.
- **Public Profile:** A profile type which shall be visible to all users and guests.
- **Private Profile:** A profile type which shall be visible to limited users and guests.

1. **Functional Requirements**

   - **1.1. User Requirements**

     - **1.1.1. User Basics**

       - **1.1.1.1. Sign Up**

         - **1.1.1.1.1.** Guests shall be able to sign up as a customer by providing their name, surname, e-mail address, choosing a password and a user name.
         - **1.1.1.1.2.** Guests should be able to sign up with their Google or Facebook account.
         - **1.1.1.1.3.** Guests shall be able to sign up as a vendor by providing their name, surname, e-mail address, choosing a password, a user name and their location (at least one).
         - **1.1.1.1.4.** Guest user shall be able to sign up while the purchase process.

       - **1.1.1.2. Sign In**

         - **1.1.1.2.1.** Users shall be able to sign in with their e-mail or user name and password.
         - **1.1.1.2.2.** There shall be a "forgotten password" button in case the user forgets the password.
         - **1.1.1.2.3.** Users should be able to sign in with their Google or Facebook account.

- **1.1.1.3. Profile**
  - **1.1.1.3.1.** Vendor and customer users shall have a profile page.
  - **1.1.1.3.2.** Users shall be able to set their profile to be public or private.
  - **1.1.1.3.3.** Vendor's profile shall contain vendor's rating, types of the products that vendor sell, products' prices and vendor's contact info.
  - **1.1.1.3.4.** Customer's profile shall contain user's address, user's info (age, sex etc.), the saved credit cards, user's previous orders and user's assessments about the vendors and the products.
- **1.1.1.4. User Types**
  - **1.1.1.4.1.** Guest: A user who is using the platform but has not signed up yet.
  - **1.1.1.4.2.** Customer: A user who is able to use all the functionality of the system other than selling products.
  - **1.1.1.4.3.** Vendor: A user who is able to sell products on the platform, in addition to the features that customers have.
  - **1.1.1.4.4.** Admin: An administrative user who is able to ban users and manage the whole system processes.

- **1.1.2. User Interactions**
  - **1.1.2.1. Rating**
    - **1.1.2.1.1.** Customers shall be able to rate the products they bought, out of 5.
    - **1.1.2.1.2.** Customers shall be able to rate the vendors whom they bought a product from, out of 10.
  - **1.1.2.2. Commenting on Products**
    - **1.1.2.2.1.** Users shall be able to comment on the products that they have already bought.
    - **1.1.2.2.2.** Users shall be able to choose to anonymize their comment prior to posting.
  - **1.1.2.3. Search**
    - **1.1.2.3.1.** Users shall be able to search for products and vendors. See 1.1.5 for more info.
  - **1.1.2.4. Communication**
    - **1.1.2.4.1.** Users shall be able to communicate with vendors through direct messaging.
  - **1.1.2.5. Carts**
    - **1.1.2.5.1.** Users shall be able to add products to their cart.

        - **1.1.2.5.2.** Users shall be able to remove products from their cart.

    - **1.1.2.6. Lists**

        - **1.1.2.6.1.** Customers shall be able to create private lists.
        - **1.1.2.6.2.** Customers shall be able to add products to their lists.
        - **1.1.2.6.3.** Customers shall be able to delete products from their lists.
        - **1.1.2.6.4.** Customers shall be able to delete their lists.

    - **1.1.2.7. Orders**

        - **1.1.2.7.1.** Customers shall be able to make and cancel orders and follow their orders. See 1.1.3 and 1.1.4 for more info.

- **1.1.3. Vendor Specific Interactions**

    - **1.1.3.1. Adding New Product**

        - **1.1.3.1.1.** Vendors shall be able to add a new product to the platform with the necessary information of products(price, condition, category, etc.)
        - **1.1.3.1.2.** Vendors shall be able to sell as many products as they want.
        - **1.1.3.1.3.** Vendors shall be able to mention about the stock of a product.

    - **1.1.3.2. Communication**

        - **1.1.3.2.1.** Vendors shall be able to communicate with admins about orders.
        - **1.1.3.2.2.** Vendors shall be able to communicate with customers that text themselves through direct message.

    - **1.1.3.3. Product Sold by Different Vendors**

        - **1.1.3.3.1.** Vendors shall be able to sell the same product with another vendors.

    - **1.1.3.4. Orders**

        - **1.1.3.4.1.** Vendors shall be able to follow all processes about their ordered products as the customers.
        - **1.1.3.4.2.** Vendors shall be able to cancel an order during the order processing stage.

    - **1.1.3.5. Recommendation System**

        - **1.1.3.5.1.** Customers shall be able to be recommended based on their interactions on the platform.

- **1.1.4. Customers Specific Interactions**

    - **1.1.4.1. Customers' Lists**

      - **1.1.4.1.1.** Customers shall be able to create their own lists and carts.
      - **1.1.4.1.2.** Customers shall be able to add as many products as they want to their baskets.

    - **1.1.4.2. Communication**

      - **1.1.4.2.1.** Customers shall be able to communicate with vendors through direct messaging.

    - **1.1.4.3. Orders**

      - **1.1.4.3.1.** Customers shall be able to follow their orders via the orders page.
      - **1.1.4.3.2.** Customers shall be able to see their active and delivered orders with sufficient information about the orders.
      - **1.1.4.3.3.** Customers shall be able to cancel their active orders.
      - **1.1.4.3.4.** Customers shall be able to return their delivered orders.

    - **1.1.4.4. Notifications**

      - **1.1.4.4.1.** Customers shall be able to be notified about changes in products that are in their lists or favorites.
      - **1.1.4.4.2.** Customers shall be able to set alarm for a certain price and choose to be notified if the price of product goes below the chosen price.

    - **1.1.4.5. Recommendation System**

      - **1.1.4.5.1.** Customers shall be able to be recommended based on their interactions on the platform.

  - **1.1.5. Searching/Listing**

    - **1.1.5.1. Search Bar**

      - **1.1.5.1.1.** Users shall be able to search for both product pages and vendor profiles using the search bar. Search results should also include semantic results, similar vendors and similar products.

    - **1.1.5.2. Filter**

      - **1.1.5.2.1.** Users shall be able to filter products based on brand, vendor, price range, rating, and discount rate.

    - **1.1.5.3. Sorting**

      - **1.1.5.3.1.** Users shall be able to sort products based on bestsellers, newest arrivals, price, number of customer reviews, rating, and number of comments.

- **1.2. System Requirements**

- **1.2.1. Security**
  - **1.2.1.1. Sign Up**
    - **1.2.1.1.1.** System shall send a verification email when a customer signs up.
    - **1.2.1.1.2.** System shall allow the passwords to be at least 8 characters which must include at least one uppercase, one lowercase character and a number
    - **1.2.1.1.3.** System shall ask customer to enter their password twice while signing up.
    - **1.2.1.1.4.** System shall verificate vendors when they sign up.
  - **1.2.1.2. Sign In**
    - **1.2.1.2.1.** System shall allow customers to enter wrong password only 3 consecutive times. After that, system shall block the account and send an email to customer for giving information about the trial.
    - **1.2.1.2.2.** System shall remind customers to change their passwords regularly.
  - **1.2.1.3. Payment**
    - **1.2.1.3.1.** System shall ask all credit card information before any transaction if no credit card information is given or no card has already been saved. The information shall include proper card number with 16 characters, expiration date and CVV (3 character security code).
- **1.2.2. Performance**
  - **1.2.2.1.** The system shall be able to respond to requests within 10 ms in general. Maximum response time should not exceed 1s.
  - **1.2.2.2.** The system shall cache frequently accessed contents to deliver faster and reduce response time.

2. **Non-Functional Requirements**
   - **2.1. Protocol**
     - **2.1.1.** The system shall meet the standards written in the W3C protocol.
     - **2.1.2.** The system shall follow the W3C Activity Streams protocol.
   - **2.2. Ethical Issues**
     - **2.2.1.** When users sign up they must accept Privacy Policy.
     - **2.2.2.** When users sign up they must accept Terms of Use.
     - **2.2.3.** User data shall be processed according to the rules specified by GDPR and KVKK.
   - **2.3. Deployment**
     - **2.3.1. Deployment on a Server**

    - ∗ **2.3.1.1.** The system shall be deployed on Amazon EC2 server

  - **2.3.2. Docker**

    - ∗ **2.3.2.1.** The system shall has docker technology to ease the development and deployment processes.

- **2.4. Availability**

  - **2.4.1. Web Access**

    - ∗ **2.4.1.1.** The system shall have a Web application that supports Chrome browser that supports all versions since 2011.
    - ∗ **2.4.1.2.** The system shall have a Web application that supports Firefox browser that supports latest version.
    - ∗ **2.4.1.3.** The system shall have a Web application that supports Safari browser that supports all versions since 2014.
    - ∗ **2.4.1.4.** The system shall have a Web application that supports Opera browser that supports all versions since 2016.

  - **2.4.2. Mobile Access**

    - ∗ **2.4.2.1.** The system shall be compatible with Android 5.1 or higher version.

- **2.5. Accessibility**

  - **2.5.1.** The system shall maintain itself every Monday between 3.00 am and 3.05 am regularly.
  - **2.5.2.** Every user shall receive an alert message 1 hour before the maintenance of the system starts.

7 API Documentation

Swagger url: bupazar API

7.1 Authentication

Functionality	URL	Method	Parameters	Responses
Sign up	/api/auth/register	POST	email*, username*, first_name*, last_name*, password*, address is_customer, is_vendor	email, username, first_name, last_name, is_customer, is_vendor, is_active, is staff, address, auth_token
Login	/api/auth/login	POST	email*, password*	id, email, username, first_name, last_name, is_customer, is_vendor, is_active, is_staff, address, auth_token
Logout	/api/auth/logout	POST	auth_token (inside headers)	success
Password Change	/api/auth/password_change	POST	auth_token (inside headers) current_password*, new_password*	success
User Information	/api/auth/user_info	POST	auth_token (inside headers)	id, email, username, first_name, last_name, is_customer, is_vendor, is_active, is_staff, address
User Profile Update	/api/auth/profile_update	POST	auth_token (inside headers) email, username, first_name, last_name, address	success

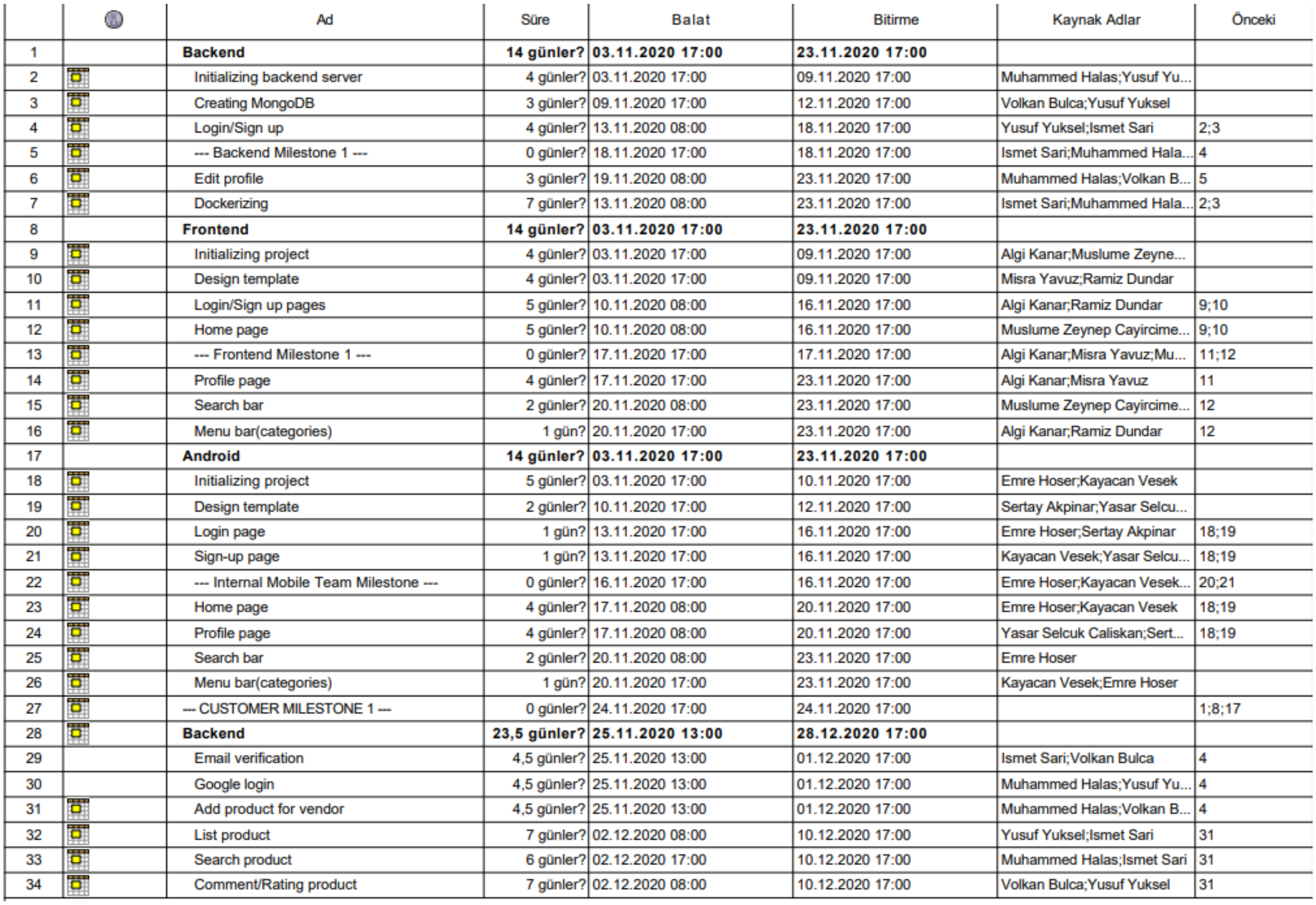
Table 4: Authentication Endpoints

7.2 Users

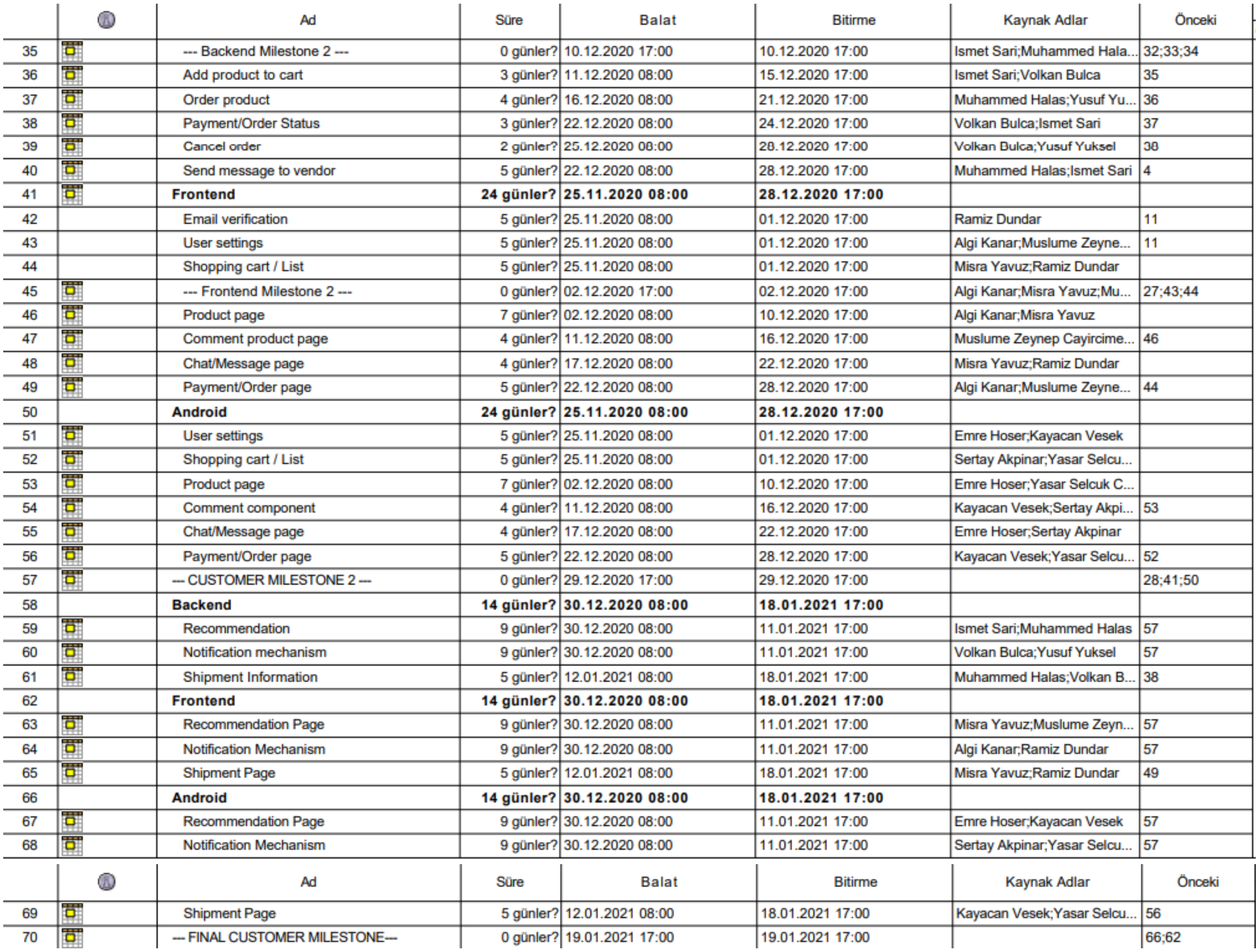
Functionality	URL	Method	Parameters	Responses
List all users	/api/users	GET	-	id, email, username, first_name, last_name, is_vendor, is_customer, is_active, is_staff, address
Retrieve user details by user id	/api/users/{id}	GET	-	id, email, username, first_name, last_name, is_vendor, is_customer, is_active, is_staff, address

Table 5: User Endpoints

8 Project Plan



	ⓘ	Ad	Süre	Balat	Bitirme	Kaynak Adlar	Önceki
1		**Backend**	**14 günler?**	**03.11.2020 17:00**	**23.11.2020 17:00**		
2	📅	Initializing backend server	4 günler?	03.11.2020 17:00	09.11.2020 17:00	Muhammed Halas;Yusuf Yu...	
3	📅	Creating MongoDB	3 günler?	09.11.2020 17:00	12.11.2020 17:00	Volkan Bulca;Yusuf Yuksel	
4	📅	Login/Sign up	4 günler?	13.11.2020 08:00	18.11.2020 17:00	Yusuf Yuksel;Ismet Sari	2;3
5	📅	--- Backend Milestone 1 ---	0 günler?	18.11.2020 17:00	18.11.2020 17:00	Ismet Sari;Muhammed Hala...	4
6	📅	Edit profile	3 günler?	19.11.2020 08:00	23.11.2020 17:00	Muhammed Halas;Volkan B...	5
7	📅	Dockerizing	7 günler?	13.11.2020 08:00	23.11.2020 17:00	Ismet Sari;Muhammed Hala...	2;3
8		**Frontend**	**14 günler?**	**03.11.2020 17:00**	**23.11.2020 17:00**		
9	📅	Initializing project	4 günler?	03.11.2020 17:00	09.11.2020 17:00	Algi Kanar;Muslume Zeyne...	
10	📅	Design template	4 günler?	03.11.2020 17:00	09.11.2020 17:00	Misra Yavuz;Ramiz Dundar	
11	📅	Login/Sign up pages	5 günler?	10.11.2020 08:00	16.11.2020 17:00	Algi Kanar;Ramiz Dundar	9;10
12	📅	Home page	5 günler?	10.11.2020 08:00	16.11.2020 17:00	Muslume Zeynep Cayircime...	9;10
13	📅	--- Frontend Milestone 1 ---	0 günler?	17.11.2020 17:00	17.11.2020 17:00	Algi Kanar;Misra Yavuz;Mu...	11;12
14	📅	Profile page	4 günler?	17.11.2020 17:00	23.11.2020 17:00	Algi Kanar;Misra Yavuz	11
15	📅	Search bar	2 günler?	20.11.2020 08:00	23.11.2020 17:00	Muslume Zeynep Cayircime...	12
16	📅	Menu bar(categories)	1 gün?	20.11.2020 17:00	23.11.2020 17:00	Algi Kanar;Ramiz Dundar	12
17		**Android**	**14 günler?**	**03.11.2020 17:00**	**23.11.2020 17:00**		
18	📅	Initializing project	5 günler?	03.11.2020 17:00	10.11.2020 17:00	Emre Hoser;Kayacan Vesek	
19	📅	Design template	2 günler?	10.11.2020 17:00	12.11.2020 17:00	Sertay Akpinar;Yasar Selcu...	
20	📅	Login page	1 gün?	13.11.2020 17:00	16.11.2020 17:00	Emre Hoser;Sertay Akpinar	18;19
21	📅	Sign-up page	1 gün?	13.11.2020 17:00	16.11.2020 17:00	Kayacan Vesek;Yasar Selcu...	18;19
22	📅	--- Internal Mobile Team Milestone ---	0 günler?	16.11.2020 17:00	16.11.2020 17:00	Emre Hoser;Kayacan Vesek...	20;21
23	📅	Home page	4 günler?	17.11.2020 08:00	20.11.2020 17:00	Emre Hoser;Kayacan Vesek	18;19
24	📅	Profile page	4 günler?	17.11.2020 08:00	20.11.2020 17:00	Yasar Selcuk Caliskan;Sert...	18;19
25	📅	Search bar	2 günler?	20.11.2020 08:00	23.11.2020 17:00	Emre Hoser	
26	📅	Menu bar(categories)	1 gün?	20.11.2020 17:00	23.11.2020 17:00	Kayacan Vesek;Emre Hoser	
27	📅	— CUSTOMER MILESTONE 1 —	0 günler?	24.11.2020 17:00	24.11.2020 17:00		1;8;17
28	📅	**Backend**	**23,5 günler?**	**25.11.2020 13:00**	**28.12.2020 17:00**		
29		Email verification	4,5 günler?	25.11.2020 13:00	01.12.2020 17:00	Ismet Sari;Volkan Bulca	4
30		Google login	4,5 günler?	25.11.2020 13:00	01.12.2020 17:00	Muhammed Halas;Yusuf Yu...	4
31	📅	Add product for vendor	4,5 günler?	25.11.2020 13:00	01.12.2020 17:00	Muhammed Halas;Volkan B...	4
32	📅	List product	7 günler?	02.12.2020 08:00	10.12.2020 17:00	Yusuf Yuksel;Ismet Sari	31
33	📅	Search product	6 günler?	02.12.2020 17:00	10.12.2020 17:00	Muhammed Halas;Ismet Sari	31
34	📅	Comment/Rating product	7 günler?	02.12.2020 08:00	10.12.2020 17:00	Volkan Bulca;Yusuf Yuksel	31



	ⓘ	Ad	Süre	Balat	Bitirme	Kaynak Adlar	Önceki
35	📅	--- Backend Milestone 2 ---	0 günler?	10.12.2020 17:00	10.12.2020 17:00	Ismet Sari;Muhammed Hala...	32;33;34
36	📅	Add product to cart	3 günler?	11.12.2020 08:00	15.12.2020 17:00	Ismet Sari;Volkan Bulca	35
37	📅	Order product	4 günler?	16.12.2020 08:00	21.12.2020 17:00	Muhammed Halas;Yusuf Yu...	36
38	📅	Payment/Order Status	3 günler?	22.12.2020 08:00	24.12.2020 17:00	Volkan Bulca;Ismet Sari	37
39	📅	Cancel order	2 günler?	25.12.2020 08:00	28.12.2020 17:00	Volkan Bulca;Yusuf Yuksel	38
40	📅	Send message to vendor	5 günler?	22.12.2020 08:00	28.12.2020 17:00	Muhammed Halas;Ismet Sari	4
41	📅	**Frontend**	**24 günler?**	**25.11.2020 08:00**	**28.12.2020 17:00**		
42		Email verification	5 günler?	25.11.2020 08:00	01.12.2020 17:00	Ramiz Dundar	11
43		User settings	5 günler?	25.11.2020 08:00	01.12.2020 17:00	Algi Kanar;Muslume Zeyne...	11
44		Shopping cart / List	5 günler?	25.11.2020 08:00	01.12.2020 17:00	Misra Yavuz;Ramiz Dundar	
45	📅	--- Frontend Milestone 2 ---	0 günler?	02.12.2020 17:00	02.12.2020 17:00	Algi Kanar;Misra Yavuz;Mu...	27;43;44
46	📅	Product page	7 günler?	02.12.2020 08:00	10.12.2020 17:00	Algi Kanar;Misra Yavuz	
47	📅	Comment product page	4 günler?	11.12.2020 08:00	16.12.2020 17:00	Muslume Zeynep Cayircime...	46
48	📅	Chat/Message page	4 günler?	17.12.2020 08:00	22.12.2020 17:00	Misra Yavuz;Ramiz Dundar	
49	📅	Payment/Order page	5 günler?	22.12.2020 08:00	28.12.2020 17:00	Algi Kanar;Muslume Zeyne...	44
50		**Android**	**24 günler?**	**25.11.2020 08:00**	**28.12.2020 17:00**		
51	📅	User settings	5 günler?	25.11.2020 08:00	01.12.2020 17:00	Emre Hoser;Kayacan Vesek	
52	📅	Shopping cart / List	5 günler?	25.11.2020 08:00	01.12.2020 17:00	Sertay Akpinar;Yasar Selcu...	
53	📅	Product page	7 günler?	02.12.2020 08:00	10.12.2020 17:00	Emre Hoser;Yasar Selcuk C...	
54	📅	Comment component	4 günler?	11.12.2020 08:00	16.12.2020 17:00	Kayacan Vesek;Sertay Akpi...	53
55	📅	Chat/Message page	4 günler?	17.12.2020 08:00	22.12.2020 17:00	Emre Hoser;Sertay Akpinar	
56	📅	Payment/Order page	5 günler?	22.12.2020 08:00	28.12.2020 17:00	Kayacan Vesek;Yasar Selcu...	52
57	📅	--- CUSTOMER MILESTONE 2 ---	0 günler?	29.12.2020 17:00	29.12.2020 17:00		28;41;50
58		**Backend**	**14 günler?**	**30.12.2020 08:00**	**18.01.2021 17:00**		
59	📅	Recommendation	9 günler?	30.12.2020 08:00	11.01.2021 17:00	Ismet Sari;Muhammed Halas	57
60	📅	Notification mechanism	9 günler?	30.12.2020 08:00	11.01.2021 17:00	Volkan Bulca;Yusuf Yuksel	57
61	📅	Shipment Information	5 günler?	12.01.2021 08:00	18.01.2021 17:00	Muhammed Halas;Volkan B...	38
62		**Frontend**	**14 günler?**	**30.12.2020 08:00**	**18.01.2021 17:00**		
63	📅	Recommendation Page	9 günler?	30.12.2020 08:00	11.01.2021 17:00	Misra Yavuz;Muslume Zeyn...	57
64	📅	Notification Mechanism	9 günler?	30.12.2020 08:00	11.01.2021 17:00	Algi Kanar;Ramiz Dundar	57
65	📅	Shipment Page	5 günler?	12.01.2021 08:00	18.01.2021 17:00	Misra Yavuz;Ramiz Dundar	49
66		**Android**	**14 günler?**	**30.12.2020 08:00**	**18.01.2021 17:00**		
67	📅	Recommendation Page	9 günler?	30.12.2020 08:00	11.01.2021 17:00	Emre Hoser;Kayacan Vesek	57
68	📅	Notification Mechanism	9 günler?	30.12.2020 08:00	11.01.2021 17:00	Sertay Akpinar;Yasar Selcu...	57
69	📅	Shipment Page	5 günler?	12.01.2021 08:00	18.01.2021 17:00	Kayacan Vesek;Yasar Selcu...	56
70	📅	--- FINAL CUSTOMER MILESTONE---	0 günler?	19.01.2021 17:00	19.01.2021 17:00		66;62

9 User Scenarios

User scenarios are stories which designers create to show how users might act to achieve a goal in a system or environment. We presented two scenarios in milestone 1.

9.1 The First Scenario: Frontend Scenario - Furkan Kale

9.1.1 Demographics

- Boğaziçi University Graduate Student
- Currently pursuing MBA
- 24 years old

9.1.2 Goals

- He wants to give feedback to this friend Ramiz, who is part of developement team of bupazar.

9.1.3 Scenario

Furkan Kale, who is pursuing MBA in Boğaziçi University, told by his close friend Ramiz that he is developing a new e-commerce website with his friends. After talking about his thoughts and ideas about the bupazar, Furkan also mentions this website to his friend Umut, who will be reviewing the mobile app of the bupazar.

9.2 The Second Scenario: Mobile Scenario - Umut Cengiz

9.2.1 Demographics

- Boğaziçi University Senior Undergraduate Student
- Currently pursuing BSc in Economics
- 23 years old

9.2.2 Goals

- He will give a feedback to his friend Furkan about the android app of an e-commerce platform that Furkan's friends are developed.

9.2.3 Scenario

Umut Cengiz explained the first android version of the e-commerce application released by his close friend Furkan's friends. First, he logged in as a guest to review the home page of the app. After that, he decided to create a new account. While creating the account he tried to use a simple password to check the security issues of the app. Finally, he logged in with the his new account.

10 Code Structure and Group Process

10.1 Backend - app/backend

10.1.1 Folder Structure

All related code is in the app/backend directory. We created two subfolders inside that directory. One is api which is django app and other one is bupazar which is django project. We created migrations, models, serializers, templates, test, utils and views subfolders inside api to keep project structure simple as possible. Thus, we put new file to related folder easily. In order to install dependencies easily, we put all of the dependencies to requirements.txt

10.1.2 Branch Structure

We created a common branch is called backend for development. We keep backend branch up to date according to master branch. We don't push master directly. We create pull request when there are remarkable commits. Reviewers pull repository and test it locally. When they get expected results, they merge it to master. If bugs arise, we assign bug fixing to related persons.

10.1.3 Workflow

We hold meetings in discord to divide tasks. We create related issues about tasks. Later we communicate over discord and whatsapp. We generally share our screens over discord to communicate easily when we need help. If there is remarkable progress, we create pull request and reviewers review it. We iterate same process until internal milestone dates. When we have internal milestone, we generally talk about what we have done so far and what are the next steps. After customer milestones, we consider customer feedbacks and integrate them via same process.

10.2 Frontend - app/frontend

10.2.1 Folder Structure

All related code is in the app/frontend directory. We created subfolders inside src folder for tasks and implemented related components inside them, such as login folder. Also there is common folder for code/components used frequently.

10.2.2 Branch Structure

All tasks implemented in related feature based branches, after which, pull request is created. Then pull request is reviewed and merged by reviewer. All work is done this way and there is no common branch or direct push to master. Since merged branches may be deleted by reviewer, there may be less active/merged branches than we as a team actually used. Here is the the 2 example branches from the repository:

- login-connect-backend: Used for connecting existing UI to backend API's.
- profile-page: Used for creating profile page of the user.

10.2.3 Workflow

We hold meetings in order to confirm divison of tasks, after which related branches and issues are created. Later we inform each other via discord and whatsapp as we progress through tasks. When it's finished pull request is created and everyone other that assignees reviews the pull request. Then we iterate the same process over the new tasks until internal/customer milestones where we add additional changes over the project according to the in-team/customer feedback.

10.3 Mobile - app/mobile/BuPazar

10.3.1 Folder Structure

All related code can be found in the app/mobile/BuPazar directory. While the build files lies within this directory, we have implemented our components in the app/mobile/BuPazar/app/src/main/java/com/example/bupazar directory. Our models and pages can be found in the corresponding directories within this directory.

10.3.2 Branch Structure

After distributing tasks among the team members that we have pre-defined before starting to implement the project, we have coded our tasks in the branches that is created with our names in it. After implementing them, we have created the corresponding pull requests to merge these branches into the "mobile-register-2" branch which we have used as a a common branch during the development period for the milestone-1. It is then merged into the master branch.

10.3.3 Workflow

Before starting to implement the project, we have determined and later divided the tasks among us to be able to commence earlier and get familiar with the Kotlin programming language and Android environment. During the development process, we always have kept ourselves in the loop about where we are and what are the possible challenges that we see ahead. We held regular meetings to stay up-to-date and solve the problems encountered by any team member. Whenever a feature is completed, a corresponding pull request is created and team members are assigned as reviewers to review and merge the pull request.

11 Evaluation of Tools and Managing the Project

11.1 Managing the Project

We decided the backend, frontend and mobile teams before the classes started by arranging a meeting. We tried to ensure that the workload was balanced by giving equal numbers of people to each group. Although we divided the group into 3 subgroups, we decided to use discord to communicate. We created 3 separate channels for frontend, backend and mobile to keep the groups in touch.During the coding, instead of pushing the master, we created branches and continued from these branches. We opened a pull request for the completed branches, after they were approved, we deleted those branches. Each team used the predetermined prefix in order to avoid confusion while commiting and opening issues.

11.2 Frontend Part

11.2.1 Frontend Framework

As a frontend team we decided to use React JS framework. Since there was no one in the team who was familiar with React before, we had a little difficulty with coding with React but we adapted in a short time. We took advantage of React's component based structure and implement reusable codes.

11.2.2 IDE

We did not make any restriction on which IDE to use in the project, so we preferred 3 different IDEs to use as frontend team. These were IntelliJ, WebStorm and Vscode. Although we have used 3 different IDEs, we have not had any problems in the project so far because all 3 had git integration and they worked in harmony with each other.

11.2.3 UI Framework

We were between 2 React UI frameworks which are React Bootstrap and Material UI. We decided with the mobile team which one to use in terms of compatibility, and chose Material UI. The main factor in our decision was that there were larger components in Material UI framework.

11.3 Mobile Part

11.3.1 Mobile Framework

As a mobile team we decided to use Android framework and Kotlin language. Since there was no one in the team who was familiar with Android development before, we start with tutorials. At first, we had a hard time implementing Kotlin and Android application. Even if we could do what we wanted later, we could not find the most accurate and efficient way due to time constraints. We still have shortcomings, but we are getting used to it. We believe that we should learn about Kotlin and Android before writing more code.

11.3.2 IDE

In the IDE selection decision, when we did the research, we came to the conclusion that Android Studio was the best. We continued to develop our project on Android Studio. Some parts were easy to use due to its similarity to IntelliJ. On the other hand, there are features and shortcuts that we still cannot discover. We decided to learn these as well, so that we can develop our project more efficiently.

11.3.3 UI Framework

We are using the framework of the Android within the scope of the UI framework. We implement our UI file in .xml format. We had to use different kind of layout(linear, relative and constraint) to design our pages.

11.4 Backend Part

11.4.1 Backend Frameworks

As the backend team, we decided to use Django Rest Framework. Since there was no one in the team who was familiar with Django and Rest Framework before, we had a little difficulty with coding with Django but we adapted in a short time. We took advantage of Django's MVT architecture and ORM to implement reusable codes.

11.4.2 IDE

We used Visual Studio Code and Pycharm. Visual Studio Code is open source and free and we already accustomed to using it. It provides many functionalities that the other IDEs provide. We used pylint as coding structure and it also support this extension. Besides, there are super cool themes designed for the python implementation in VS code.

11.4.3 Database and Other Tools

We used MongoDB as database and MongoDB Atlas as cloud database service. We integrated it via djongo engine because djongo lets us use Django with MongoDB without changing the Django ORM.

We used Postman and Swagger to test our endpoint while implementing the endpoints.

12 Design Documents

We have implemented the design documents related to our project during the CmpE 352 course, and we uploaded the documents to our GitHub wiki page. The design documents can be found using the following links:

- **Use Case Diagram:** Click here to display our use case diagram.
- **Class Diagram:** Click here to display our class diagram.
- **Sequence Diagrams:** Click here to display our sequence diagrams.

13 Assessment of the Customer Presentation

First of all, for the direction of our project it can be told that it is going according to the schedule we updated at the beginning. This schedule update actually gave us an effective control through our project development. We presented log in and sign up, home page, profile page for web site and mobile application. The most prominent feature for the first milestone was the initial registering process and updating the registration information according to the updates of the user, as well as the interaction between home page, profile page and log in and sign up.

In the frontend presentation, our validation warnings at register page were praised. Also, the overall structure of our home page and profile page were found decent. Yet it was mentioned that after the logout, only the button change was not seemed satisfactory so we can add more pointing out indicators for a logout. Moreover, the only bug we have been pointed out was when we put the same email that has already been saved, the warning prompt says it is a username error. This will be fixed. Other than these, website presentation went smoothly. Taking the API response according to each individual, redirecting the user to profile page and showing the necessary account name in the profile page were executed well. For the future we need to focus on adding more functionalities to home page and profile page like product and vendor search functionality using the search bar and product pages that includes product information and add to cart/list functionalities. These were indicated in the second milestone.

For the mobile presentation, it was said that it is not logical to display logout button in the homepage of a guest entry. Also, in the mobile app when we enter a weak password, the prompt gave an error, which is 'Format is wrong." and this was not found adequate because it creates an ambiguity for the user whether what changes a user needs to do for a successful entry. This needs an alteration. The overall scenario was same with the web site and well executed. But it had been told that the design could be enhanced for the mobile application.

In the question and answer session, for both web site and mobile we had been faced with small number of questions and improvement advice. The first one was why it is easy to register as a vendor user. It is recommended to add some security features like asking tax number or Turkish identity number for vendors to prevent exploitation and it is said that we must have personal email verification for all users while registering.

When we look at the whole presentation, we can say that we showed what we offered for the Customer Milestone 1. The most difficult part was the communication between each team members and connection of the development teams. However, we managed to overcome this classic problem by effective time usage. The presentation could have gone bad if we had a major bug or a deadline issue, but since we completed our planned actions on time and without any errors, it went smoothly. In short, for the first milestone, the presentation was a success.