

# CMPE451 Final Milestone Report Group 4

## Group Members

Burak Çuhadar

Mehmet Erdinç Oğuz

Koray Çetin

Eylül Yalçınkaya

Emre Girgin

Berkay Alkan

M. Olcayto Türker

Çağrı Çiftçi

Alperen Bağ

Veli Can Ünal

Meriç Üngör

Berke Can Gürer

*Fall, 2020*

# Table of Contents

- [\*\*1. Final Project Assessment\*\*](#)
- [\*\*2. Deliverable Status and Evaluation\*\*](#)
  - [\*\*a. Backend\*\*](#)
  - [\*\*b. Mobile\*\*](#)
  - [\*\*c. Frontend\*\*](#)
- [\*\*3. Summary of Coding Work\*\*](#)
- [\*\*4. Unit Tests\*\*](#)
- [\*\*5. Glossary and Requirements\*\*](#)
- [\*\*6. Activity Stream Implementation & W3C Standard Compliance\*\*](#)
  - [\*\*a. Implementation\*\*](#)
  - [\*\*b. Standard Compliance\*\*](#)
- [\*\*7. API Documentation\*\*](#)
- [\*\*8. Design Documents\*\*](#)
  - [\*\*a. Class Diagram\*\*](#)
  - [\*\*b. Use Case Diagram\*\*](#)
  - [\*\*c. Sequence Diagrams\*\*](#)
- [\*\*9. Project Plan\*\*](#)
- [\*\*10. User Scenarios\*\*](#)
  - [\*\*a. Android Scenario\*\*](#)
  - [\*\*b. Frontend Scenario\*\*](#)
- [\*\*11. User Manual\*\*](#)
  - [\*\*a. Android\*\*](#)
  - [\*\*b. Frontend\*\*](#)
- [\*\*12. System Manual\*\*](#)
  - [\*\*a. Android\*\*](#)
  - [\*\*b. Backend\*\*](#)
  - [\*\*c. Frontend\*\*](#)
- [\*\*13. Assessment of the Final Customer Presentation\*\*](#)

# 1. Final Project Assessment

Buyo is an e-commerce platform where customers can find a variety of products from different vendors. The platform provides a functional interface for customers to search for and buy products as well as vendors to sell their products. Intuitive user interactions and global standards are important for the platform to be successful.

After Milestone 2, we focused on implementing the feature set of vendor and admin users as we were mostly done with customer users. Until the final milestone, we also implemented important features such as Google sign up and recommendation system. We fixed bugs and prepared our product for the final delivery.

During these two semesters, we believe to have gained important skills that will help us a lot in our professional lives. As for soft skills, we learned how to work as a team under pressure and time limitations. We got better at distributing tasks, having efficient meetings and using collaborative tools such as Github and Slack. We are content to have worked together without resentments or disputes within the team.

In terms of technical skills, we learned how to use Github for version control and collaborative coding which is a valuable skill for every computer engineer. We also got the chance to practice widely used frameworks such as W3C and activity streams. Some of the group members learned the technology they used without having prior knowledge whereas others found the opportunity to hone their existing skills. Overall, this course provided us with challenges and opportunities to grow.

# 2. Deliverable Status and Evaluation

## Backend

Notification Infrastructure	<b>DONE</b>
Socket Messaging and Listing Messages	<b>DONE</b>
Email Verification	<b>DONE</b>
Report Comment and Product	<b>DONE</b>
Admin Actions and Login	<b>DONE</b>
Forgot Password	<b>DONE</b>
Vendor Update, Get Products and Get Orders	<b>DONE</b>
Semantic Search	<b>PARTIALLY DONE</b>

Google Sign In	<b>DONE</b>
Update Order Status	<b>DONE</b>

- Notification Infrastructure: **Done**

Notification infrastructure works for product updates, order status changes and similar other updates. It doesn't work real time but this was because the logic behind the notification infrastructure was advanced and having it work real time was a hard challenge and may cause an overhead for our backend's performance. The user receives all necessary notification updates as we discussed with our customer and these updates are suitable with our requirements.

- Socket Messaging and Listing Messages: **Done**

Socket messaging is implemented using Socket.io in NodeJS, and works as expected while using messaging infrastructure. The implementation includes **redis** to handle two users sharing data in the same endpoint to create communication between message senders. This was a problem with NodeJS since it doesn't support memory sharing in the same endpoint so we decided to use **redis** to handle this problem in a clever way. Therefore, we have a working message platform for vendors and products.

- Email Verification: **Done**

Email verification is implemented so that users signed up for the first time should verify their emails to continue using the platform. This was handled using **nodemailer** library to send an email to the user to ask them verify their email address. We don't require users to verify their email address when they sign in with Google, since it didn't make sense when they already use their verified email address to login to the platform.

- Report Comment and Product: **Done**

Users can report a comment or product to the admin when they think the content of the reported comment/report is offensive/disturbing etc. We created separate tables for keeping these reports, and an admin can access these reported content and act according to the content.

- Admin Actions and Login: **Done**

We created an admin user to handle reports, ban users according to that, delete products and comments according to reports sent by users. Admin can update the content of the platform

to keep the platform disturbing-content free.

- Forgot Password: **Done**

Users can use the “**forgot password**” functionality when they forgot their password to reset their password. **Nodemailer** library is used to send an email to the user’s account and it includes a link where the user can change their password. This implementation works successfully, and we check if the user is verified when we send the password reset email.

- Vendor Update, Get Products and Get Orders: **Done**

Vendors can update the products they added to the platform, see their products either by themselves or by applying filters(similar to the filtering mechanism used in product filtering in getProducts logic) and see the ordered products of theirs. Vendors can change the stock of a product with given attributes(like size, color etc.), change the content of a product, add a new product with attributes. Vendors can see the products they uploaded to the platform and see their stock values. Finally, vendors can see the products that other customers ordered and make status changes on them, and also can see the total earnings they made with their orders.

- Semantic Search: **Partially Done**

We index the description, brand, vendor and the title of a product to use in our search using Mongoose. Our search makes a full text search in our database by looking at these fields, and returns the products it finds. Erdinc was planning to use ElasticJS for semantic search, but after trying Mongoose search, we figured that using an advanced semantic search method was an overhead for us since we didn’t have that much product and our products were created by us so we weren’t sure if the semantic search would make a big difference. However, semantic search can still be improved by assigning weights to those indexes and by having more indexes such as product attributes, which will let us return products when the user makes a search like “**red clothes**”.

- Google Sign In: **Done**

Google Sign In is implemented for customers in the backend since we only allow customers to use Google Sign In. Google sign in logic is implemented in a way that customers that signed in with Google for the first time are created as a new user. These customers can sign in with the same Google account to continue their session, or choose to sign up with the same email address to have a password as well. If they sign up with the same email address, customers can either use Google Sign In or generic login to enter the platform and continue their session connected with their email address. Also, when a customer uses Google Sign In, we assign their email address as verified since the email address is verified by Google.

- Update Order Status: **Done**

Status of an order can be updated to keep track of an order either on vendor's side or customer's side, handle **cancelled**, **returned** products behaviour and also to calculate the total earnings of a vendor. There are some restrictions while updating the status: A user(user is used for both customer and vendors) cannot cancel an order that is already shipped, cannot change the status of a product that is already cancelled. A customer cannot return a product that is not shipped yet. A vendor cannot return a product since it doesn't make sense, and the product's status cannot be changed if it's returned. Finally, each order's status is changed by an automated process so that it's assumed to be shipped in 1 day, delivered in 3 days and approved in 5 days. We decided to do that since we don't have real vendors and real shipping companies.

## Mobile

Vendor Profile Page	<b>DONE</b>
Vendor Account Information Page	<b>DONE</b>
Add Product	<b>PARTIALLY DONE</b>
Product List Page	<b>DONE</b>
Edit Product	<b>PARTIALLY DONE</b>
Messages	<b>PARTIALLY DONE</b>
Notifications	<b>DONE</b>
Image Search	<b>DONE</b>
Google Sign-in	<b>DONE</b>
Comments Page	<b>DONE</b>
Report Product/Comment Page	<b>DONE</b>
Vendor Orders Page	<b>DONE</b>

- Vendor Profile Page: **Done**

On this page, the vendor sees the buttons that are linked to his/her related pages. S/He can edit the profile information, add a new product, see the list of his/her products, all the orders, messages, notifications, change password, and logout. After the log-in, the user is directed to this page and can access it any time by clicking on the "Profile" tab.

- Vendor Account Information Page: **Done**

This is the page where the vendor user can edit his credentials. S/he can update his/her company, and website. The email is demonstrated but the vendor can not change it because we use it as the primary key.

- Add Product: **Partially Done**

This process consists of a set of pages. After the vendor user pressed on the "Add Product" button, s/he is directed to the Category Selection Page. After s/he selects the category of the product s/he is going to add, s/he is directed to the page where the details of the product are entered. On this page, the name, description, current and original price, brand, and the features of the product are wanted. After defining the name of the features and the available options (ie. for the RAM feature only the 8GB and 16GB options are available) are set, the vendor is directed to the page where the stock values are wanted. On this third page, all the combinations of the features are listed. (ie. 8GB RAM-500GB Disk size and 16GB RAM-500GB Disk size is two different combinations of the features) In other words, we compute the cartesian product of the possible features and want an integer from the vendor representing the stock value of that combination. (We accept 0 as the input.) The only missing part is the image upload part. The vendor can pick an image from his/her mobile device but its connection to the server is missing. Instead, the BUYO logo is used as the default for that product.

- Product List Page: **Done**

On this page, the products of the customer are listed. The vendor can edit, delete and see the details of her/his products. The product detail page is the same as the other type of users, except the like, add to cart, and report sections. We replaced like with delete and add to cart with the edit product button. Finally, we removed the report button from this page if the vendor owns this product. The vendor can still see the comments of his/her product.

- Edit Product: **Partially Done**

This page is exactly the same as the page where the vendor enters the attributes like name, description, and others. The only difference is that the fields of the product are filled with the original values so that the vendor has to update only the parts s/he wanted to change. However, we do not show a default stock value because it depends on the cartesian product of the features of the product. For example, if the vendor enters a new feature, one of the previous combinations is divided into the number of possible options of the new product. This is why we do not use the previous values of the stocks. Again, similar to the "add product" process, the vendor can not edit the image of the product using the mobile app. However, s/he can see the current image of it while editing as the default.

- Messages: **Partially Done**

The user can see all the conversations of him/her with the other users. Each conversation is separated from each other by the username of the other user and the last message sent to or received from there. After the user clicks on one of the conversations, the chat page is shown. On this page, the messages of the users are shown one after another based on the time they sent. The only missing part is that the messaging is not done in real-time. In order to see new messages coming from the other user, the user has to go back and re-enter the conversation.

- Notifications: **Done**

On this page, the notifications asserted for the user are listed. The user receives a notification if the status of one of his/her orders changes (ie. cancellation) or the price of the product s/he added to his/her wishlist changes.

- Image Search: **Done**

The users can search using the images on their phones. An image classification algorithm is embedded into the application. When the user uploads an image to the app, the predicted class of this image -by the classification model- is searched as the keyword. Then the user is directed to the search page.

- Google Sign-in: **Done**

The new users who want to use the BUYO Mobile app, can directly sign-in via their Google accounts and start the use of all the features without a need for verification.

- Comments Page: **Done**

The comments made by the customers who bought a product can make a comment to it. The comments made to a product are listed in the comments section in the product detail.

- Report Product/Comment Page: **Done**

All types of users can report a comment or product with supplying reasoning. The only exception is the vendor can not report his/her products. The report button exists in both product detail and comments pages.

- Vendor Orders Page : **Done**

On this page, the orders for the vendor are listed. The vendor can either approve or cancel the order. Also, the vendor can send a message to the customer using this interface. Also, by clicking on the extend icon, s/he can see the details of the order.

## Frontend

Messages	<b>DONE</b>
Vendor Products Page	<b>DONE</b>
Vendor Add/Edit/Delete Product	<b>DONE</b>
Vendor Profile Page	<b>DONE</b>
Vendor Orders	<b>DONE</b>
Notifications	<b>DONE</b>
Google Sign In/Up	<b>DONE</b>
Email Verification Page	<b>DONE</b>
Forgot Password Page	<b>DONE</b>
Report Product/Comment	<b>DONE</b>
Admin Page	<b>PARTIALLY DONE</b>

- Messages: **Done**

Our platform provides messaging between vendors and customers. It is achieved by the socket messaging endpoint provided by the backend team. Vendors can start a conversation by messaging the customer using the button displayed in the orders page. Customers can also start messaging the same way at their orders page. All the messaging history can be displayed at the messages page of the users. Users can also continue their conversation at that page. Note that there was a bug related to messaging which is fixed after the final milestone. The bug was that the messages were not displayed real time although the endpoint provides real time messaging. It was because of misunderstanding react states and javascript function closures. However that problem is now fixed.

- Vendor Products Page: **Done**

At this page vendors can see their products after they add a product. They can also delete and edit those products from this page.

- Vendor Add/Edit/Delete Product: **Done**

Vendors have the ability to add a new product, delete and edit added products. To add a product a vendor has to provide all the necessary information such as the category, name, attributes and stock information of the product. They can later edit those information.

- Vendor Profile Page: **Done**

This page is similar to the customer profile page (contains fields to display and change user profile information, and a component to change password), with the biggest difference being the Google Maps functionality. The user can edit its vendor location on the map.

- Vendor Orders: **Done**

This page is also similar to the customer orders page: It lists the vendor's orders, the order information for each order, and the products in each order. This page also displays the total earnings of the vendor. Vendors can cancel the delivery of any product, as long as the order has a status denoting that it's not delivered yet. They can also message the customer who made the order.

- Notifications: **Done**

Notifications for product discounts which are added to the wishlist and order status updates (for both customer and vendor) are displayed on this page, using the activity stream data sent by the backend. The notification cards on this page can be clicked on, in order to go to the relevant page (i.e. product page for a discount notification, orders page for a canceled order notification, etc.).

- Google Sign In/Up: **Done**

Our platform provides Google sign in and sign up functionality so that the users can start using our application as quickly as possible.

- Email Verification Page: **Done**

Email verification is an important security functionality of our platform. Upon sign up, an email is sent to the user by the backend, which contains a link specific to that user. When the user opens that link, a verification request is sent to the backend, and the user is verified. Unless the user goes through the email link, their account stays unverified.

- Forgot Password Page: **Done**

One of our requirements was that the users can change their passwords without having to sign in, by providing their email addresses. This is done via the reset password button (also known as the ‘forgot password’ button) found on the sign in page. It redirects the user to a page where they are asked to enter their email address. After the email address is submitted, a mail is sent to the user by the backend, which contains a link specific to the user. When the user opens that link, they are redirected to the ‘forgot password’ page, where they are asked to enter a new password (and confirm it by re-entering their new password). Upon the new password’s submission, the user’s password is correctly changed.

- Report Product/Comment: **Done**

One of the major responsibilities of the admins is to ensure that there aren’t any inconvenient products or comments. Users can report a product or comment by stating the problem as a simple text message.

- Admin Page: **Partially Done**

Admins can see the reported products and comments on their page. They can remove the reported comments and products or decide to ban the user who commented inconveniently or the vendor who added the inappropriate product. Note that when the admin chooses to remove a comment, the report is not removed. The report is removed when the user related to the report is banned. For the final milestone presentation we could not connect these functionalities to the backend. However we connected them after the milestone. Unfortunately we could not deploy the admin page.

### 3. Summary of Coding Work

Eylül Yalçınkaya (Frontend)	I implemented the frontend UI and backend connections for: <ul style="list-style-type: none"><li>- Vendor's all products page</li><li>- Vendor's add product page</li><li>- Vendor's update product page</li><li>- Vendor's delete product functionality</li></ul> I also contributed to bug fixes.
Meriç Üngör (Frontend)	Since the last customer meeting, I have dealt with the setup of the admin project, the implementation of reported products and reported comments and its deployment. Besides from the admin project, I implemented the google sign in button on the frontend, the different header we show to the vendor users and the switching logic with our normal header, and the backend connection for the customer addresses page. I have also meddled with various bug fixes to carry our project closer to a production ready stage.
Mehmet Erdinc Oguz (Backend)	I implemented the Google Signin logic in the backend. I receive the Token created by Google in Frontend or Android and store it in our database for the customers signed in using Google. I also added some logic to handle users signing up with their Google email address before/after they sign in using Google so that users won't have trouble continuing their account's session with different types of logins. Also, I implemented semantic search for our search endpoint which indexes the description, brand, title and vendor names for the products. This could have been improved more, but I realized that wasn't really useful for us since we created all our products by ourselves and not used real products, and creating semantic connections between products was harder than I expected. In

	<p>addition to that, I implemented the status update logic for both vendors and customers. Status of ordered products can be <b>Cancelled</b>, <b>Returned</b>, <b>Delivered at &lt;DATE&gt;</b>, <b>Pending</b>, <b>Approved</b>; related to that, a product's status can be changed according to its current status so that illogical status changes are not possible. For example, non-delivered products cannot be returned to the vendor, so trying this gives an error to the user. We also decided on having an automated delivery system, so when a user gets the status of their orders, if their status is not changed manually, order will be updated according to the time past after the order is made. For example, the status of an order made 1 day ago will be <b>Shipped</b> automatically by the backend.</p> <p>Also, I implemented getting orders and the total earnings information for the vendors. I updated the get orders method that is implemented for customers to minimize endpoints. Finally, I was involved in two peer programming sessions with our backend team to fix some crashes, the communication with other teams to make design decisions, and fixed some bugs related to the backend when I had the chance.</p>
Alperen Bağ (Android)	<ul style="list-style-type: none"> <li>- Notification page <ul style="list-style-type: none"> <li>- Compatible with the W3C activity stream standards</li> </ul> </li> <li>- Order page <ul style="list-style-type: none"> <li>- Status of orders</li> <li>- Approving or rejecting orders</li> <li>- Canceling or returning orders</li> <li>- Messaging the vendor of an ordered product or adding comment for an ordered product</li> </ul> </li> <li>- Bug fixes (Address loading bug)</li> </ul>

Koray Cetin (Backend)	<p>In this milestone, I've implemented notifications. Notifications isn't working real time but adding notifications in different events was challenging since it involved implementing the logic in several endpoints and implementing the get notifications endpoint. And wrote tests for the notifications. Also I implemented real-time messaging with socket.io. In nodejs, two different endpoints don't have to share the same resources. This means that when a user connects to a node, he/she may not be in the same process with the other user. In order to handle this communication, I've implemented a shared memory system between the nodes with redis. This also means running a redis container in our microservice architecture, and especially in AWS this was a tough implementation. Other than the socket request and responses, I've added get messages, get last messages, and send messages(to offer a non-real time option) too. And wrote tests for the messages. Besides, I've implemented users also liked and users also bought recommendation endpoints. These endpoints recommend the products based on users' interactions on our system. For example if user X likes product A, and user Y likes product B and C; we recommend C product to the X user. If there's no such case, we recommend the most selled(or bought) products. Also I've added user ids to all comment models and vendor ids to all vendor objects to make the ban logic easier. Moreover I was involved in two peer programming tasks and fixed several server crashes by tracking the logs and the database as soon as possible when they needed to be prioritized.</p>
Emre Girgin (Android)	<p>I've implemented the UI, logic, and backend connections of the following: Report Product, Report Comment, My Products page for the vendor user, Adding a new</p>

	product, editing, deleting an existing product.
Olcayto Türker (Backend)	<p>For the Final Milestone, I implemented sending verification email when signup, sending password reset email to users that forgot password, sending information email to banned users, forgot password endpoint, verify account endpoint, reporting a product and reporting a comment features. For verification email function, an email is being sent to a newly signed up user and the user activates her/his account by clicking the link in the email. Verification occurs by calling the verify account endpoint. For the forgot password function, an email is sent to the user to reset their password. For report operations, I created admin_service and required models, link admin service to appropriate ports. Additionally report product and report comment endpoints added to admin_service. Then I implemented unit tests for these functionalities using mocha. Also, we had lots of work together as pair programming as the backend team. In these sessions, we analyzed bugs and errors in the backend and solved them.</p>
Burak Çuhadar (Frontend)	<p>For this milestone I implemented the UI with backend connections for these functionalities: Real time messaging, report comment, report product, cancel order, return order. I also implemented small features like checking stock value at the product page, displaying user ratings at the comments section and adding an option to remove filtering. I also contributed to bug</p>

	fixes and reviewed code of my team members.
Berkay Alkan (Android)	<p>I implemented the UI, functionalities and API connection of</p> <ul style="list-style-type: none"> <li>- Vendor profile page</li> <li>- Vendor account information</li> <li>- Vendor change password</li> <li>- Logout button</li> <li>- Forgot password</li> <li>- Google sign in</li> </ul> <p>Also I made bug fixes.</p>
Veli Can Ünal (Backend)	<ul style="list-style-type: none"> <li>● Vendor : Add, Update, Delete Product Functions <ul style="list-style-type: none"> <li>- There was a bug related to MongoDB and Mongoose. We cannot update nested objects or add any nested objects for our stock information of products.</li> <li>- I needed to change product structure. ProductInfos is a string instead of an object. I also updated this attribute for all products in our database</li> <li>- I tried to applied this important type change to other parts of our code</li> </ul> </li> <li>● Admin : Get Reports: <ul style="list-style-type: none"> <li>- There are two type of reports customer report which is using for if customer reports another ones comment , vendor report if there is a problem with the product, customer reports the vendor</li> </ul> </li> <li>● Admin : Ban Customer: <ul style="list-style-type: none"> <li>- Admin can ban the customer by checking the comment report that comes from another customer</li> <li>- If admin bans customer, we are deleting the current comment, and current comment report as well</li> </ul> </li> </ul>

	<ul style="list-style-type: none"> <li>● Admin: Ban Vendor <ul style="list-style-type: none"> <li>- Admin can ban the customer by looking the product report that comes from a customer</li> <li>- If we ban the vendor, we are deleting all products of the vendor and current product report as well</li> </ul> </li> <li>● Maintenance for these endpoints and system <ul style="list-style-type: none"> <li>- There can be problems with these endpoints or any other development can affect them in a negative way, or other development teams can request extra change.</li> <li>It is all okay and part of the maintenance effort.</li> </ul> </li> </ul>
Çağrı Çiftçi(Android)	<p>UI and backend connections for these pages</p> <ul style="list-style-type: none"> <li>- Shortened Message History Page</li> <li>- Messaging page <ul style="list-style-type: none"> <li>- Shows whole chat history with chosen user</li> <li>- Sending message</li> </ul> </li> </ul> <p>Bug fixes and code reviews with the android team.</p> <p>Notes:</p> <p>For messaging we were planning to implement socket connection with the backend but in android I implemented socket manager(in the datamanager package in the project), but there was a bug that I could not fix so in android messaging is working but it is not in real time. To receive new messages, the user needs to reopen the page.</p>
Berke Can Gürer (Frontend)	I added the sign-out functionality to frontend. I added the login checks (checking if the user is verified & not banned). I implemented the frontend UI & backend connections for these pages:

	<ul style="list-style-type: none"> <li>- Vendor profile page</li> <li>- Vendor orders page</li> <li>- Notifications page (for both customer &amp; vendor)</li> <li>- Email verification page</li> <li>- Password reset ('forgot password') page</li> </ul> <p>I also contributed to bug fixes, reviewed merge requests of my team members, and continued to provide moral support!</p>
--	--

## 4. Unit Tests

Koray Cetin

Notifications: [c3605efab09c62122709eaaf5749063aa8f1e2fe](#)

Messages: [40a8cc0713d612fb139651ab532323a199bd075c](#)

Mehmet Erdinc Oguz

Account Service: [7e954a45c59a3dc8ff5f9a0b0eb477e5073b88f5](#)

Veli Can Ünal

Product Service Unit Tests :[d12c9602b98d94a8c4ea71bc796643fdb2869425](#)

M.Olcayto Türker

Admin Service Unit Tests: [7f7816d2c17f98f0c042f5724440cc040c84841e](#)

Account Service Unit Tests:[4522e0f8dfa2ec415f02c54d46a1c121d5e1cd0](#)

## 5. Glossary and Requirements

### Glossary

- Campaign: Campaigns include discounts for buying in bulk (discount on a single product is not included) such as but not limited to: buying 3 items but paying for 2, discount of 10% for orders more than 100 TL, free shipment.
- Cart: A functionality for customer users that enables them to save the products they want to buy. The customers can then order the products saved in the cart to buy them all at once.

- Combinative search: Searching mechanism that allows users to filter the results of a search by multiple attributes. For example, a user searching for 'hat' can filter the results by criteria A and B, and then sort by Y.
- Comment: Text displayed on product pages written by customers. Customers can explain and discuss their shopping experience via this functionality.
- Customer: A logged-in user with the special privileges of creating wishlists, ordering products and tracking orders.
- Guest user: A non-logged in user with the ability to search, display and read comments on products.
- GDPR: Abbreviation(General Data Protection Regulation) for the regulation in European Union law on data protection and privacy in the European Union and the European Economic Area. Its primary goal is to give control to individuals over their personal data.
- KVKK: Abbreviation(Kişisel Verilerin Korunması Kanunu) for the law introduced by Republic of Turkey to regulate how personal data should be respected, used and processed by individuals and organizations to protect basic rights and liberties of citizens.
- Wishlist: A page for customers to keep track of the products they are interested in.
- Collection A subcategory of wishlist where the customers can add their products they are interested in.
- Notification: Message sent by the platform to notify users of an event (i.e. item price changes, direct messages, etc.)
- Order processing stage: Time-frame between a customer ordering product and them receiving it.
- Orders Page: A special page for customers and vendors. Customers can track their orders and interact with their orders on this page (e.g. cancel an order, message a vendor, etc.). Vendors can also track the orders made for their products and cancel the orders in case of a problem (such as stock issues).
- Platform: The software product as a whole.
- Product: Any item that is for sale on the platform.
- Product Page: A page for details about the product. Price of the product, comments and rating on the product made by customers, and details of the product are displayed on this page.
- PWA: A website that looks and behaves as if it is a mobile app. PWAs are built to take advantage of native mobile device features, without requiring the end user to visit an app store, make a purchase and download software locally.
- Semantic Search: Searching mechanism where the results are generated by considering searcher's intent, context of the query and the relationships between words. A semantic search engine tries to understand the meaning of the query to improve search accuracy. For example in this project's context when a customer searches for "clothes for summer", search engine should understand what is meant by the query and generate appropriate results.
- User: People that interact with the platform.
- Vendor: Class of users that are offering products in exchange for money.

# **Requirements**

## **1. Functional Requirements**

### **1.1 User Requirements**

#### **1.1.1 User Types**

##### **1.1.1.1 Customer**

1.1.1.1.1 Customers shall be able to search for products.

1.1.1.1.2 Customers shall be able to filter searched products based on average customer review, brand, category, vendor, price range and available filters for that category.

1.1.1.1.3 Customers shall be able to sort searched products based on best-selling, latest arrival time, price, average customer review, number of comments, and rating.

1.1.1.1.4 Customers shall have a Wishlist and Collection.

1.1.1.1.4.1 Customers shall be able to save collections, to keep track of items they wish to buy and to purchase the same items multiple times.

1.1.1.1.5 Customers shall have a Cart page.

1.1.1.1.5.1 Customers shall be able to add products to their cart.

1.1.1.1.5.2 Customers shall be able to remove products from their cart.

1.1.1.1.5.3 Customers shall be able to purchase all items from their cart at once.

1.1.1.1.6 Customers shall have an Orders page.

1.1.1.1.6.1 Customers shall be able to see all their purchased items in the Orders page.

1.1.1.1.6.2 Customers shall be able to cancel their active orders in the Orders page before they have been shipped.

1.1.1.1.6.3 After the items have been delivered, customers shall be able to return the delivered orders in the Orders page.

1.1.1.1.6.4 Customers shall be able to see total amount of money they have spent so far in the Orders page. The total differs based on the actions the customer has taken, such as returning an item or cancelling an order.

1.1.1.1.6.5 Customers shall be able to file a complaint about the order to inform the vendor and the admin.

1.1.1.1.7 Customers shall be able to receive notifications.

1.1.1.1.7.1 Customers shall be notified when an order is delivered.

1.1.1.1.7.2 Customers shall be able to choose to be notified if the price of a chosen product changes.

1.1.1.1.7.3 Customers shall be able to choose to be notified if a product that is out of stock is available again.

1.1.1.1.8 Customers shall be able to comment on the products that they have purchased, after the products have been delivered.

1.1.1.1.9 Customers shall be able to rate the products that they have purchased, after the products have been delivered.

1.1.1.1.10 Customers shall be able to read user comments about products.

1.1.1.1.11 Customers shall give their names and address information before ordering their product.

1.1.1.1.12 Customers shall be able to edit their address during the ordering process or at their profile page.

### **1.1.1.2 Vendor**

1.1.1.2.1 Vendors shall be able to keep track of their orders via the orders page.

1.1.1.2.2 Vendors shall specify at least one store location through Google Maps.

1.1.1.2.3 Vendors shall be able to cancel an order which is in the order processing stage.

1.1.1.2.4 Vendors shall be able to specify a reason while cancelling an order.

1.1.1.2.5 Vendors shall be notified if an order is canceled.

1.1.1.2.6 Vendors shall be able to make a discount on a product for a certain time span.

1.1.1.2.7 Vendors shall be able to create campaigns for certain products for a certain time span.

1.1.1.2.8 Vendors shall not be able to make purchases using their vendor account.

1.1.1.2.9 Vendors should specify their stock information while adding a product to the platform.

1.1.1.2.10 Vendors shall be able to read user comments about products.

### **1.1.1.3 Guest User**

1.1.1.3.1 Guests shall be able to search for products.

1.1.1.3.2 Guests shall be able to view the prices.

1.1.1.3.3 Guests shall be able to read user comments about products.

1.1.1.3.4 Guests shall be required to sign up during their purchase process.

#### **1.1.1.4 Admin**

1.1.1.4.1 Admins shall be able to ban customers or vendors.

1.1.1.4.2 Admins shall be able to remove malicious comments.

#### **1.1.2 Social Interactions**

1.1.2.1 Customer users shall send message to vendors who are seller of the products they buy.

1.1.2.2 Vendors shall be able to communicate with platform admins about a certain product/order.

#### **1.1.3 Sign up**

1.1.3.1 Customer and vendor users shall provide an e-mail, an username and a password to sign up.

1.1.3.2 Vendors shall provide their location to sign up.

1.1.3.3 E-mail address and user type pairs shall be unique. (i.e. there can be a customer and a vendor accounts using the same e-mail).

1.1.3.4 User passwords should be in the form of alphanumeric and at least 6 characters long.

1.1.3.5 Google account should be used for signing up/in.

1.1.3.6 During registration, the email address should be verified with a confirmation mail.

1.1.3.7 All user types shall accept a KVKK agreement to signup.

1.1.3.8 Users shall be able to change their passwords without signing in by providing their e-mail addresses.

### **1.2 System Requirements**

## **1.2.1 Search**

### **1.2.1.1 Searching Mechanism**

1.2.1.1.1 Searching Mechanism shall utilize product names, features, vendor names, vendor location, customer reviews and ratings.

1.2.1.1.2 Searching Mechanism shall include semantic search (search for similar products).

1.2.1.1.3 Searching Mechanism shall include combinative search.

### **1.2.1.2 Filtering Mechanism**

1.2.1.2.1 Products shall be filtered by average customer review.

1.2.1.2.2 Products shall be filtered by brand.

1.2.1.2.3 Products shall be filtered by vendor.

1.2.1.2.4 Products shall be filtered by price range.

1.2.1.2.5 Products shall be filtered by product specified features (size, color, technical features).

### **1.2.1.3 Sorting Mechanism**

1.2.1.3.1 System shall sort products by popularity (bestsellers).

1.2.1.3.2 System shall sort products by the time of release (new arrivals).

1.2.1.3.3 System shall sort products by price.

1.2.1.3.4 System shall sort products by customer review.

1.2.1.3.5 System shall sort products by number of comments.

1.2.1.3.6 System shall sort products by rating.

## **1.2.2 Interactions**

1.2.2.1 Customers shall be able to comment on the products that they have purchased, after the products have been delivered.

1.2.2.2 Customers shall be able to rate the products that they have purchased, after the products have been delivered.

1.2.2.3 After making an order, customers shall be able to message the vendor(s) of the item(s) directly.

1.2.2.3 Vendors shall be able to message admins, e.g. to cancel an order.

1.2.2.4 Customer users shall send message to vendors who are seller of the products they buy.

1.2.2.5 The names of the customers shall be visible only with their initials on top of the comments they make for privacy. ex: M\*\*\*\* U\*\*\*\*

### **1.2.3 Interface**

1.2.3.1 System shall have categories for discovering new products.

### **1.2.4 Product**

1.2.4.1 When the same product is sold by different vendors, product data shall be distinguished.

### **1.2.5 Order Process**

#### **1.2.5.1 Customer's Orders Page**

1.2.5.1.1 Orders shall be exactly in one of those states, such as Pending, Being Prepared, Being Delivered, Pending for Customer Approval, Completed, Cancelled, Refund.

1.2.5.1.2 For both active or completed cases, there shall be product information, order date, product price, cargo information, and delivery date.

1.2.5.1.3 There shall be options for canceling active orders that are not at the Being Delivered stage or returning completed ones.

1.2.5.1.4 For each case (active, canceled, completed, returned), the total amount of money spent shall be available. In terms of canceled and returned products belonging to a shopping basket, users shall see the amount of money they get back when they check their basket.

1.2.5.1.5 There shall be only one shipment option, which is BUYO Express.

1.2.5.1.6 Customers shall pay with Turkish Lira.

#### **1.2.5.2 Vendor's Orders Page**

1.2.5.2.1 Orders shall be exactly in one of those states, such as Pending, Being Prepared, Being Delivered, Completed, Cancelled, Refund.

1.2.5.2.2 For both active or completed cases, there shall be product information, order date, product price, cargo information, and delivery date.

1.2.5.2.3 Vendors shall be able to communicate with platform admins about a certain product/order.

1.2.5.2.4 Vendors shall be able to cancel an order (e.g. if out of stock) during order processing stage.

1.2.5.2.5 Vendors shall be able to see all orders and the total earnings.

1.2.5.2.6 Vendors shall be able to disable and re-enable new orders for a product (e.g. if out of stock).

## **1.2.6 Notifications**

1.2.6.1 Customers shall be able to choose notified if the price of a wishlisted product changes.

1.2.6.2 Customers shall be able to choose notified if a product that is out of stock is available again.

1.2.6.3 Customers shall be notified when the order is delivered.

1.2.6.4 Vendors shall be notified if an order is canceled.

## **1.2.7 Recommendation**

1.2.7.1 The platform shall recommend certain products to the users based on their interactions on the platform.

## **1.2.8 Payment**

1.2.8.1 When a customer makes a purchase, the platform shall keep the money for three days. When the user approves that they got the product, money shall be transferred to the vendor.

1.2.8.2 If the cargo has not arrived to the customer yet, the customer shall extend the period for approval of the product.

1.2.8.3 System shall not accept the payment if the customer does not approve an e-commerce shopping agreement.

1.2.8.4 Customers shall approve an KVKK agreement after writing the payment information.

## **1.2.9 Wishlist**

1.2.9.1 Customers shall be able to create private collections that they can name, edit, delete, add products to, e.g. to keep track of the items' prices.

1.2.9.2 Customers shall create empty collections before adding any products.

1.2.9.3 Products that gets removed from a wishlist, are also removed from all the collections.

1.2.9.4 Products that gets removed from a collection, does not get removed from the wishlist.

1.2.9.5 Wishlist includes all the products from all the collections.

## **2. Nonfunctional Requirements**

### **2.1 Availability**

2.1.1 There shall be a web application and native Android applications.

2.1.2 The mobile app should not be a PWA or hybrid application like Angular & Ionic combination or Cordova.

2.1.3 Platform shall be deployable to a configurable server, Amazon EC2 or Digital Ocean.

2.1.4 Platform shall be [Dockerized](#).

2.1.5 Open-source software with appropriate use permissions shall be used, as long as it is properly attributed and documented, and unless otherwise specified.

2.1.6 System shall support Turkish and English characters.

2.1.7 Direct messaging feature implemented between different user types shall be in real-time.

2.1.8 Users shall be able to access the platform via a web browser or Android mobile device.

### **2.2 Standards**

2.2.1 System shall support [W3C Activity Streams protocol](#) so that the activities on the platform are expressed as a stream – including the actions taken by vendors e.g. adding a new product, discounts.

2.2.2 System shall follow [W3C standards](#) and the standards of the used packages.

### **2.3 Privacy**

2.3.1 System shall follow rules defined by [GDPR](#) and [KVKK](#).

2.3.2 The personal information, contact information, copyrighted contents, license issues and everything related to these paradigms should be respected and considered.

## 2.4 Security

2.4.1 User passwords should be encrypted before stored in the database.

2.4.2 Users should be sent a verification mail when they want to change their passwords. Until the users click the link in the mail, their passwords should not be changed.

## 2.5 Performance

2.5.1 The maximum response time of the system should be 2 seconds.

2.5.2. The system should be able to respond up to 1000 users simultaneously.

# 6. Activity Stream Implementation & W3C Standard Compliance

## Implementation

Activity stream in BUYO works based on users' own interactions. Since BUYO is an e-commerce service, users shouldn't be bothered with other users' interactions as long as they do concern them. Thus the activity stream has three types of notifications.

- Vendors are notified when customer cancels or returns an order
- Customers are notified when vendor cancels an order
- Customers are notified when the price of a product he/she like is dropped.

In the backend, we added notifications to the databases when these events occurred. And when a user wants to see their activity stream, clients request to get notifications and display them.

## Standard Compliance

To fit our activity stream to W3C compliance, we followed some specific class declarations. We used Collection class in the W3C vocabulary to store our notifications list.

```
class Collection {  
    "@context": "https://www.w3.org/ns/activitystreams",  
    "summary": "Sally's notes",  
    "type": "Collection",  
    "totalItems": 2,  
    "items"(Array(Object)): [  
    ]
```

```

    "type": "Note",
    "name": "A Simple Note"
},
{
    "type": "Note",
    "name": "Another Simple Note"
}
]
}

```

And in items list, we store Activity instances to represent a notification

```

class Activity extends Object {
    "@context": "https://www.w3.org/ns/activitystreams",
    "type": "Activity",
    "summary": "Sally did something to a note",
    "actor": {
        "type": "Person",
        "name": "Sally"
    },
    "name": "activity_name",
    "summary": "Sally did something to a note",
    "target": "target_id"
}

```

And Activity class has an Actor instance.

```

{
    "@context": "https://www.w3.org/ns/activitystreams",
    "type": "Person",
    "name": "Sally Smith"
}

```

Here is an example notification response.

```
{
    "status": {
        "code": 200,
        "message": "OK"
    },
    "data": {
        "notifications": [
            {
                "@context": "https://www.w3.org/ns/activitystreams",
                "summary": "BUYO Notifications",
                "type": "Collection",
                "totalItems": 2,
                "items": [
                    {
                        "type": "Update",
                        "name": "Discount",
                        "startTime": "2014-12-31T23:00:00-08:00",
                        "summary": "<vendor>'s <productName> price dropped from <originalPrice> to <price>",

```

```

"actor": {
    "type": "Organization",
    "name": "<vendor_name>",
    "id": "<vendor_id>"
},
"target": "<product_id>"}
{
    "type": "Update",
    "name": "Cancel Order",
    "startTime": "2014-12-31T23:00:00-08:00",
    "summary": "Your order <order_id> is cancelled by the customer!",
    "actor": {
        "type": "Person",
        "name": "<customer_email>",
        "id": "<customer_id>"
    },
    "target": "<order_id>"}
}
]
}
}

```

## 7. API Documentation

**Base URL:** <http://localhost:8080/>

**Base URL on Amazon server:** <http://3.141.25.245:8080/>

### What is this app

This is a running backend of an e-commerce product which serves web and android applications.

### How to run the app

Once you are inside the backend/product\_service folder, install docker and run

docker-compose build docker-compose up

Web app is available on the base URL, you can query the app with any request tool.

### Endpoints

All of the endpoints returns responses in JSON format.

### Google Signin

Allows users to signin/signup to the platform using Google Signin.

Endpoint: /google-signin

Method: POST

Authorization: None

Parameters:

email=google\_email@gmail.com&name=erdincoguz&token=HASd879ahs(RECEIVED\_TO\_KEN\_FROM\_GOOGLE)

Response: ` { "status": { "code": 200, "message": "OK" }, "data": { "userId": 10 } }

## Delete Comment

Deletes a comment from a product.

Endpoint: /comment

Method: DELETE

Authorization: None

Parameters: id=5fcf71186e8db70012a8e2f3

Response: { "status": { "code": 200, "message": "OK" } }

## Report Comment

Reports a comment with a message.

Endpoint: /report/comment

Method: POST

Authorization: None

Parameters: commentId=600c42d45b75b5001906d833&message=mymessage

Response: { "status": { "code": 200, "message": "OK" , "commentReportId":"5fcf71186e8db70012a8e2f3"} }

## Report Product

Reports a product with a message.

Endpoint: /report/product

Method: POST

Authorization: None

Parameters: productId=600c42d45b75b5001906d833&message=mymessage

Response: { "status": { "code": 200, "message": "OK" , "productReportId":"5fcf71186e8db70012a8e2f3"}}

## Verify Account

Verifies an account with given id and user type. Endpoint: /account/verify

Method: GET

Authorization: None

Parameters: userType=customer&id=5ffb0ac4d94c150012bf7d3d

Response: { "status": { "code": 200, "message": "OK"}}

## Forgot Password

Gets user email and send an email to that user for changing his/her password. Endpoint: /account/forgotPassword

Method: POST

Authorization: None

Parameters: email=molcaytoturker@gmail.com

Response: { "status": { "code": 200, "message": "OK"}}

## Add Address to Customer User

Adds an address to a customer. Note: addressTitle must be unique.

Endpoint: /account/address

Method: POST

Authorization: None

Parameters: id=123&address={"addressTitle": "home", "city": "istanbul", "province": "sariyer", "street": "elm", "address": "dksdssd"}

Response: { "status": { "code": 200, "message": "OK" }}

## Update Address of Customer User

Updates address of a customer with given addressTitle.

Endpoint: /account/address

Method: PATCH

Authorization: None

Parameters: id=123&address={"addressTitle": "home", "city": "denizli", "province": "merkez", "street": "elm", "address": "dksdssd"}

Response: { "status": { "code": 200, "message": "OK" }}

## Delete Address of Customer User

Deletes address of a customer with given addressTitle.

Endpoint: /account/address

Method: DELETE

Authorization: None

Parameters: id=123&address={"addressTitle": "home"}

Response: { "status": { "code": 200, "message": "OK" }}

## Add Comment

Adds a comment to a product.

Endpoint: /comment

Method: POST

Authorization: None

Parameters: userId=123&productId=123&comment=asdadsdsadas&rating=4.23

Response: { "status": { "code": 200, "message": "OK" }, "data": {"commentId": 324} }

## Sort products

Sorts products according to rating, name and price in ascending or descending order. sortingFactor parameter can be rating, name and price. sortingType can be ascending or descending.

Endpoint: /products

Method: POST

Authorization: None

Parameters:

categories=["Kids"]&sortingFactor=rating&subcategory=T-shirt&size=XS&brand=Adidas&color=Blue&sortingType=descending

Response: { status: { code: 200, message: "OK", }, data: { products: [] } }

## Get Categories

Returns an inspirational quote from a famous, historical person.

Endpoint: /categories

Method: GET

Authorization: None

Parameters: None

Response: { status: { success: true, code: 200 }, data: { categories: [ { name: "Woman", path: "Woman", subcategories: [ { name: "Shoes", path: "Woman/Shoes", subcategories: [ ] } ] } ] }

## Get Products of a Category

Gets the products of a category

Endpoint: /products

Method: GET

Authorization: None

Parameters(Query):?categories=["Phones"] Warning: "filterCriterias" shows that which attributes can we use the current category. Response:

```
{
  "status": {
    "code": 200,
    "message": "OK"
  },
  "data": {
    "products": {
      "productList": [
        {
          "category": [
            "Electronics",
            "Phones"
          ],
          "productInfos": [
            {
              "attributes": [
                {
                  "name": "screenSize",
                  "value": "5.5 inch"
                }
              ]
            }
          ]
        }
      ]
    }
  }
}
```

```

        "value": "5.5 Inch"
    },
    {
        "name": "RAM",
        "value": "4 GB"
    },
    {
        "name": "diskSize",
        "value": "64 GB"
    }
],
"stockValue": 21
},
{
    "attributes": [
        {
            "name": "screenSize",
            "value": "5.5 Inch"
        },
        {
            "name": "RAM",
            "value": "4 GB"
        },
        {
            "name": "diskSize",
            "value": "128 GB"
        }
    ],
    "stockValue": 18
}
],
"description": "Uncover the Ultimate Design. Beauty Beyond the Visual",
"name": "Samsung S8",
"price": 2800,
"originalPrice": 3000,
"imageUrl":
"https://elasticbeanstalk-us-east-2-334058266782.s3.amazonaws.com/images/16089907472046",
"rating": 4,
"brand": "Samsung",
"vendor": {
    "name": "JohnsShop",
    "rating": 4.23,
    "id": "3129281hasda91892312"
},
"id": "5fe86281a17abd0012d601e6"
},
{
    "category": [
        "Electronics",
        "Phones"
    ],
    "productInfos": [
        {
            "attributes": [
                {

```

```

        "name": "screenSize",
        "value": "5.5 Inch"
    },
    {
        "name": "RAM",
        "value": "8 GB"
    },
    {
        "name": "diskSize",
        "value": "64 GB"
    }
],
"stockValue": 54
},
{
    "attributes": [
        {
            "name": "screenSize",
            "value": "5.5 Inch"
        },
        {
            "name": "RAM",
            "value": "8 GB"
        },
        {
            "name": "diskSize",
            "value": "128 GB"
        }
],
"stockValue": 74
}
],
"description": "Uncover the Ultimate Design. Beauty Beyond the Visual",
"name": "Samsung S9",
"price": 4000,
"originalPrice": 4200,
"imageUrl":
"https://elasticbeanstalk-us-east-2-334058266782.s3.amazonaws.com/images/16089907472057",
"rating": 4.4,
"brand": "Samsung",
"vendor": {
    "name": "Ahmet",
    "rating": 3.22,
    "id": "3129281hasda91892312"
},
"id": "5fe86281a17abd0012d601e7"
},
{
    "category": [
        "Electronics",
        "Phones"
    ],
    "productInfos": [
        {
            "attributes": [

```

```

        {
          "name": "screenSize",
          "value": "6 Inch"
        },
        {
          "name": "RAM",
          "value": "8 GB"
        },
        {
          "name": "diskSize",
          "value": "64 GB"
        }
      ],
      "stockValue": 95
    },
    {
      "attributes": [
        {
          "name": "screenSize",
          "value": "6 Inch"
        },
        {
          "name": "RAM",
          "value": "8 GB"
        },
        {
          "name": "diskSize",
          "value": "128 GB"
        }
      ],
      "stockValue": 95
    }
  ],
  "description": "Uncover the Ultimate Design. Beauty Beyond the Visual",
  "name": "Samsung S10",
  "price": 4500,
  "originalPrice": 5000,
  "imageUrl":
  "https://elasticbeanstalk-us-east-2-334058266782.s3.amazonaws.com/images/16089907472078",
  "rating": 4.4,
  "brand": "Samsung",
  "vendor": {
    "name": "AyseTeyze",
    "rating": 3.21,
    "id": "3129281hasda91892312"
  },
  "id": "5fe86281a17abd0012d601e8"
},
{
  "category": [
    "Electronics",
    "Phones"
  ],
  "productInfos": [
    {

```

```
"attributes": [
    {
        "name": "screenSize",
        "value": "5 Inch"
    },
    {
        "name": "RAM",
        "value": "4 GB"
    },
    {
        "name": "diskSize",
        "value": "64 GB"
    }
],
"stockValue": 47
},
{
    "attributes": [
        {
            "name": "screenSize",
            "value": "5 Inch"
        },
        {
            "name": "RAM",
            "value": "4 GB"
        },
        {
            "name": "diskSize",
            "value": "128 GB"
        }
],
"stockValue": 94
}
],
"description": "Uncover the Ultimate Design. Beauty Beyond the Visual",
"name": "Samsung S7",
"price": 4500,
"originalPrice": 5000,
"imageUrl":
"https://elasticbeanstalk-us-east-2-334058266782.s3.amazonaws.com/images/16089907472099",
"rating": 3.5,
"brand": "Samsung",
"vendor": {
    "name": "JohnsShop",
    "rating": 4.23,
    "id": "3129281hasda91892312"
},
"id": "5fe86281a17abd0012d601e9"
},
{
    "category": [
        "Electronics",
        "Phones"
    ],
    "productInfos": [
```

```
{
  "attributes": [
    {
      "name": "screenSize",
      "value": "5 Inch"
    },
    {
      "name": "RAM",
      "value": "8 GB"
    },
    {
      "name": "diskSize",
      "value": "64 GB"
    }
  ],
  "stockValue": 39
},
{
  "attributes": [
    {
      "name": "screenSize",
      "value": "5 Inch"
    },
    {
      "name": "RAM",
      "value": "8 GB"
    },
    {
      "name": "diskSize",
      "value": "128 GB"
    }
  ],
  "stockValue": 23
}
],
"description": "Uncover the Ultimate Design. Beauty Beyond the Visual",
"name": "Iphone 8",
"price": 4700,
"originalPrice": 5000,
"imageUrl":
"https://elasticbeanstalk-us-east-2-334058266782.s3.amazonaws.com/images/160899074721010",
"rating": 4.8,
"brand": "Apple",
"vendor": {
  "name": "Pablos",
  "rating": 2.43,
  "id": "3129281hasda91892312"
},
"id": "5fe86281a17abd0012d601ea"
},
{
  "category": [
    "Electronics",
    "Phones"
  ],
}
```

```

"productInfos": [
    {
        "attributes": [
            {
                "name": "screenSize",
                "value": "5 Inch"
            },
            {
                "name": "RAM",
                "value": "4 GB"
            },
            {
                "name": "diskSize",
                "value": "64 GB"
            }
        ],
        "stockValue": 42
    },
    {
        "attributes": [
            {
                "name": "screenSize",
                "value": "5 Inch"
            },
            {
                "name": "RAM",
                "value": "4 GB"
            },
            {
                "name": "diskSize",
                "value": "128 GB"
            }
        ],
        "stockValue": 45
    }
],
"description": "Uncover the Ultimate Design. Beauty Beyond the Visual",
"name": "Iphone 6",
"price": 5200,
"originalPrice": 5500,
"imageUrl": "https://elasticbeanstalk-us-east-2-334058266782.s3.amazonaws.com/images/160899074721211",
"rating": 4,
"brand": "Apple",
"vendor": {
    "name": "Pablos",
    "rating": 2.43,
    "id": "3129281hasda91892312"
},
"id": "5fe86281a17abd0012d601eb"
},
],
"filterCriterias": [
{
    "name": "screenSize",

```

```

    "displayName": "Screen Size",
    "possibleValues": [
        "5.5 Inch",
        "6 Inch",
        "5 Inch"
    ]
},
{
    "name": "RAM",
    "displayName": "RAM",
    "possibleValues": [
        "4 GB",
        "8 GB"
    ]
},
{
    "name": "diskSize",
    "displayName": "Disk Size",
    "possibleValues": [
        "64 GB",
        "128 GB"
    ]
}
}
}

```

## Get Products with Keyword

Returns the products whose title is the keyword.

Endpoint: /products

Method: GET

Authorization: None

Parameters(Query): search=cocuk

Response: { status: { code: 200, message: "OK", }, data: { products: [ { category: ["Cocuk", "Erkek Cocuk", "Ic Giyim"], sizes: null, colors: ["Red", "Blue", "White", "Purple", "Orange", "Black"], name: "Erkek Cocuk Pamuklu Atlet 2'li", id: 10001, imageUrl: "https://img-lcwaikiki.mncdn.com/mnresize/230/-/pim/productimages/20201/4041406/l\_20201-0w1007z4-jyx\_a.jpg", rating: 0.78, price: 22.99, originalPrice: 22.99, brand: "Adidas", stockValue: { Red: 17, Blue: 69, White: 72, Purple: 57, Orange: 71, Black: 44, }, vendor: { name: "Ahmet", rating: 3.22, id: "5823324823csdsadf237" }, { category: ["Cocuk", "Kiz Cocuk", "Kazak"], sizes: null, colors: ["Red", "Blue", "White", "Purple", "Orange", "Black", "Grey", "Green"], name: "Kiz Cocuk Kalin Triko Kazak", id: 10002, imageUrl: "https://img-lcwaikiki.mncdn.com/mnresize/230/-/productimages/20192/4/3870373/l\_20192-9wp531z4-gzn\_a.jpg", rating: 4.92, price: 19.99, originalPrice: 39.99, brand: "Nike", stockValue: { Red: 82, Blue: 37, White: 35, Purple: 15, Orange: 53, Black: 26, Grey: 49, Green: 17, }, vendor: { name: "Ahmet", rating: 3.22, id: "312873128jcds723e2" } } ] } }

## Get Product Detail

Returns the detailed information of a product.

Endpoint: /product

Method: GET

Authorization: None

Parameters(Query): id=5fe86281a17abd0012d601eb

Response:

```
{  
    "status": {  
        "code": 200,  
        "message": "OK"  
    },  
    "data": {  
        "result": {  
            "category": [  
                "Electronics",  
                "Phones"  
            ],  
            "productInfos": [  
                {  
                    "attributes": [  
                        {  
                            "name": "screenSize",  
                            "value": "5.5 Inch"  
                        },  
                        {  
                            "name": "RAM",  
                            "value": "4 GB"  
                        },  
                        {  
                            "name": "diskSize",  
                            "value": "64 GB"  
                        }  
                    ],  
                    "stockValue": 21  
                },  
                {  
                    "attributes": [  
                        {  
                            "name": "screenSize",  
                            "value": "5.5 Inch"  
                        },  
                        {  
                            "name": "RAM",  
                            "value": "4 GB"  
                        }  
                    ]  
                }  
            ]  
        }  
    }  
}
```

```
        "name": "diskSize",
        "value": "128 GB"
    },
],
"stockValue": 18
}
],
"description": "Uncover the Ultimate Design. Beauty Beyond the Visual",
"name": "Samsung S8",
"price": 2800,
"originalPrice": 3000,
"imageUrl":
"https://elasticbeanstalk-us-east-2-334058266782.s3.amazonaws.com/images/16089907472046",
"rating": 4,
"brand": "Samsung",
"vendor": {
    "name": "Pablos",
    "rating": 2.43,
    "id": "11323123asjdac8a123"
},
"id": "5ff37169542dbb0012339131",
"comments": [
    {
        "id": "5fe8af602e7b430019d8004c",
        "rating": 4.23,
        "text": "niceee",
        "owner": {
            "username": "Koray Cetin",
            "email": "koray@gmail.com",
            "id": "3129281hasda91892312"
        }
    }
],
"filterCriterias": [
    {
        "name": "screenSize",
        "displayName": "Screen Size",
        "possibleValues": [
            "5.5 Inch"
        ]
    },
    {
        "name": "RAM",
        "displayName": "RAM",
        "possibleValues": [
            "4 GB"
        ]
    },
    {
        "name": "diskSize",
        "displayName": "Disk Size",
        "possibleValues": [
            "64 GB",
            "128 GB"
        ]
    }
]
```

```
        }
    ]
}
}
```

## Like / Dislike

Triggers the like or dislike event from a user to a product.

Endpoint: /like

Method: POST

Authorization: None

Parameters(Query): customerId=12341&productId=10003

Response: { "status": { "code": 200, "message": "OK" } }

## Get Wishlist

Gets the wishlist of a user.

Endpoint: /wishlist

Method: GET

Authorization: None

Parameters: customerId=12341

Response: { "status": { "code": 200, "message": "OK" }, "data": { "products": [ { "category": [ "Cilt", "Yüz Bakımı", "Yüz Maskesi" ], "sizes": null, "colors": null, "name": "Saf Kil Detoks Maskesi 50 ml", "id": 12312, "imageUrl": "https://img-watsons.mncdn.com/Content/Images/Thumbs/0320616\_saf-kil-detoks-maskesi-50-ml.png", "rating": 1.2, "price": 69.95, "originalPrice": 69.95, "brand": "Nike", "stockValue": { "self": 45 }, "vendor": { "name": "AyseTeyze", "rating": 3.21, "id": "3129281hasda91892312" } } ] } }

## Get Customer User Information

Gets a customer user's information.

Endpoint: /account

Method: GET

Authorization: None

Parameters(Query): id=1024&userType=customer

Response: {"status": {"code": 200, "message": "OK"}, "data": {"result": {"address": [], "email": "customer@mail.com", "id": 1024, "password": "1234", "gender": "male"}}}

## Get Vendor List

Gets a vendor user's information.

Endpoint: /vendor/vendorlist

Method: GET

Authorization: None

Warning: This endpoint added for development purposes. It makes easy to check vendorId. If you want to create a feature by using vendor list please communicate with backend team

Response:

```
{"status": {"code": 200, "message": "OK"}, "data": {"result": [{"_id": "600d56a63bf84a001266eda4", "name": "Johns Shop", "email": "john@gmail.com", "longitude": "41.0082° N", "address": "Rumelihisari mah. Cami sk. No:123/32", "latitude": "28.9784° E", "rating": 4.23, "website": "www.johns.com"}, {"_id": "600d56a63bf84a001266eda5", "name": "Ahmet", "email": "ahmet@gmail.com", "longitude": "41.1082° N", "address": "Rumelihisari mah. 6. sk. No:123/32", "latitude": "28.9284° E", "rating": 3.22, "website": "www.ahmets.com"}, {"_id": "600d56a63bf84a001266eda6", "name": "AyseTeyze", "email": "ayse.teyze@gmail.com", "longitude": "41.3082° N", "address": "Rumelihisari mah. 11. sk. No:123/32", "latitude": "28.9484° E", "rating": 3.21, "website": "www.ayseteyzeshop.com"}, {"_id": "600d56a63bf84a001266eda7", "name": "Pablos", "email": "pablos@gmail.com", "longitude": "41.6082° N", "address": "Rumelihisari mah. 14. sk. No:123/32", "latitude": "28.9184° E", "rating": 2.43, "website": "www.pablos.com"}]}}
```

## Get Vendor User Information

Gets a vendor user's information.

Endpoint: /account

Method: GET

Authorization: None

Parameters(Query): id=1024&userType=vendor

Response: {"status": {"code": 200, "message": "OK"}, "data": {"result": {"longitude": "long", "latitude": "lang", "email": "customer@mail.com", "id": 1024, "password": "1234", "company": "vendor's company", "website": "www.vendor.com", }}}}

## Login

Performs the login event of a user.

Endpoint: /login

Method: POST

Authorization: None

Parameters(Query): userType=customer&email=mark@zucker.org&password=1234

Response: { "status": { "code": 200, "message": "OK" }, "data": { "userId": 10 } }

## Vendor Sign-up

Performs the sign-up event for a vendor user.

Endpoint: /signup

Method: POST

Authorization: API key from .env file

\*\*Parameters(Query):\*\*userType=vendor&email=mark@zucker.org&password=1234&longitude=long&latitude=lang&website=website.com&company=company&name=Mark Zuckerberg

Response: { "status": { "code": 200, "message": "OK" }, "data": { "userId": 10 } }

## Customer Sign-up

Performs the sign-up event for a customer user.

Endpoint: /signup

Method: POST

Authorization: API key from .env file

Parameters(Query): userType=customer&email=mark@zucker.org&password=1234

Response: { "status": { "code": 200, "message": "OK" }, "data": { "userId": 10 } }

## Reset the Mock Database

This endpoint is created for internal purposes, it can be used when the database is needed to initialized or reset.

Endpoint: /db/init

Method: POST

Authorization: API key from .env file

Parameters: None

Response: { "status": { "code": 200, "message": "OK" } }

## Product Info

ProductInfo is an object where the attributes of a product and the quantity of it is defined. Its schema is as follows:

```
'{"attributes": [Attribute], "quantity": Number}'
```

where Attribute's schema is:

```
'{"name": <ATT_NAME>, "value": <ATT_VALUE>}'
```

ATT\_NAME: Name of the attribute the product have. E.g. *size*, *color*, *material*, *diskSize*...

ATT\_VALUE: The value for the attribute specified. E.g. *red*, *Large*, *L*, *wood*, *256GB*...

Below endpoints use the above structure for productInfo objects.

## Add Product to Cart

Add a product to customer's cart. Product Info can be created as below:

```
PRODUCT_INFO = {"attributes": [{"name": "size", "value": "L"}, {"name": "color", "value": "Blue"}],  
"quantity": 1}
```

Endpoint: /cart

Method: POST

Authorization: None

Parameters(Query):

```
customerId=1234567890abc&productId=0987654321abc&productInfo=<PRODUCT_INFO>
```

Response: { "status": { "code": 200, "message": "OK" } }

## Remove Product from Cart

Remove a product from customer's cart. Same endpoint as above is used, except the "quantity" field in ProductInfo should be missing when deletion is made. created as below:

```
PRODUCT_INFO = {"attributes": [{"name": "size", "value": "L"}, {"name": "color", "value": "Blue"}]}
```

Endpoint: /cart

Method: POST

Authorization: None

Parameters(Query):

customerId=1234567890abc&productId=0987654321abc&productInfo=<PRODUCT\_INFO>

Response: { "status": { "code": 200, "message": "OK" } }

## Get Products in the Customer Cart

Gets all products in a customer's cart. Product Info should be created as below:

Endpoint: /cart

Method: GET

Authorization: None

Parameters(Query): customerId=1234567890abc

Response: { "status": { "code": 200, "message": "OK" }, data: { products } }

## Empty the Cart

Remove all products in a customer's cart.

Endpoint: /cart

Method: DELETE

Authorization: None

Parameters(Query): customerId=1234567890abc

Response: { "status": { "code": 200, "message": "OK" } }

## Checkout Order

Create an order from the products in the cart.

Endpoint: /order

Method: POST

Authorization: None

Parameters(Query): customerId=1234567890abc&creditCard={"name": "712837123", "number": 1234666666667777, "expirationMonth": 10, "expirationYear": 2024, "cvc": 999}&address=ADDLATER

Response: { "status": { "code": 200, "message": "OK" }, "data": { cartId: 7821478372, orderedProducts: orderedProducts, unavailableProducts: unavailableProducts, customerId: params.customerId, } }

## Get Orders

Get the orders of a vendor, or customer.

Endpoint: /order

Method: GET

Authorization: None

Parameters(Query): id=721837123&userType=customer

Response for customer: { "status": { "code": 200, "message": "OK" }, "data": { orders: orders } }

Response for vendor: { "status": { "code": 200, "message": "OK" }, "data": { orders: orders, totalEarnings: totalEarnings } }

## Update Order Status

Update the status of the given order.

Available Status for client to use:

- For vendor:
  - Approved(automatically approved in 5 days)
  - Shipped(automatically shipped in 1 day)
  - Cancelled(if not shipped)
- For customer:
  - Approved(automatically approved in 5 days)
  - Returned(if the product is delivered)
  - Cancelled(if not shipped)
- For both(experimental):
  - Allows any other string, but not sure about its use case.

Endpoint: /order

Method: PATCH

Authorization: None

Parameters(Query):

userType=customer&userId=721837123&status=Approved&orderId=6as7dasdh

Response: { "status": { "code": 200, "message": "Order status is successfully changed." } }

## Update OrderProduct Status

Update the status of the given product in an order.

Available Status for client to use:

- For vendor:
  - Approved(automatically approved in 5 days)
  - Shipped(automatically shipped in 1 day)
  - Cancelled(if not shipped)
- For customer:
  - Approved(automatically approved in 5 days)
  - Returned(if the product is delivered)
  - Cancelled(if not shipped)
- For both(experimental):
  - Allows any other string, but not sure about its use case.

Endpoint: /order

Method: PATCH

Authorization: None

Parameters(Query):

userType=customer&userId=721837123&status=Approved&orderId=6as7dasdh&productId=6gashgdash26178867

Response: { "status": { "code": 200, "message": "Product status is successfully changed." } }

## Filter Data

Filters the data according to given criterias.

Endpoint: /products

Method: GET

Authorization: None

Parameters(Query): /products?brand=LC Waikiki&color=Green&categories=["Bag"]

Warning: "filterCriterias" shows that which attributes can we use the current category.

Response:

```
{  
  "status": {  
    "code": 200,  
    "message": "Success"  
  }  
}
```

```
"message": "OK"
},
"data": {
  "products": {
    "productList": [
      {
        "category": [
          "Men Clothing",
          "Bag",
          "Backpack"
        ],
        "productInfos": [
          {
            "attributes": [
              {
                "name": "size",
                "value": "20L"
              },
              {
                "name": "color",
                "value": "Green"
              }
            ],
            "stockValue": 68
          }
        ],
        "description": "Men casual 20L backpack. It's smart and very useful.",
        "name": "Men Casual Grey Backpack",
        "price": 89.99,
        "originalPrice": 111.99,
        "imageUrl":
          "https://elasticbeanstalk-us-east-2-334058266782.s3.amazonaws.com/images/16089146597058",
        "rating": 3.97,
        "brand": "LC Waikiki",
        "vendor": {
          "name": "Ahmet",
          "rating": 3.22,
          "id": "3129281hasda91892312"
        },
        "id": "5fe86281a17abd0012d601bb"
      }
    ],
    "filterCriterias": [
      {
        "name": "size",
        "displayName": "Size",
        "possibleValues": [
          "70L",
          "20L"
        ]
      },
      {
        "name": "color",
        "displayName": "Color",
        "possibleValues": [

```

```

        "Green"
    ]
}
}
}
}
```

## Sort Data

Update the status of the given order.

Endpoint: /products

Method: GET

Authorization: None

Parameters(Query): ?sortingFactor=name&sortingType=ascending&categories=["Phones"]

\*\* ! Warning:\*\* "sortingType" can be "ascending" or "descending". "sortingFactor" can be "name", "rating", "originalPrice" & "price" Response:

```
{
  "status": {
    "code": 200,
    "message": "OK"
  },
  "data": {
    "products": {
      "productList": [
        {
          "category": [
            "Electronics",
            "Phones"
          ],
          "productInfos": [
            {
              "attributes": [
                {
                  "name": "screenSize",
                  "value": "5 Inch"
                },
                {
                  "name": "RAM",
                  "value": "4 GB"
                },
                {
                  "name": "diskSize",
                  "value": "64 GB"
                }
              ],
              "name": "Phone A"
            }
          ],
          "name": "Phone A"
        }
      ],
      "name": "Electronics"
    }
  }
}
```

```
        "stockValue": 42
    },
    {
        "attributes": [
            {
                "name": "screenSize",
                "value": "5 Inch"
            },
            {
                "name": "RAM",
                "value": "4 GB"
            },
            {
                "name": "diskSize",
                "value": "128 GB"
            }
        ],
        "stockValue": 45
    }
],
"description": "Uncover the Ultimate Design. Beauty Beyond the Visual",
"name": "Iphone 6",
"price": 5200,
"originalPrice": 5500,
"imageUrl":
"https://elasticbeanstalk-us-east-2-334058266782.s3.amazonaws.com/images/160899074721211",
"rating": 4,
"brand": "Apple",
"vendor": {
    "name": "Pablos",
    "rating": 2.43,
    "id": "3129281hasda91892312"
},
"id": "5fe86281a17abd0012d601eb"
},
{
    "category": [
        "Electronics",
        "Phones"
    ],
    "productInfos": [
        {
            "attributes": [
                {
                    "name": "screenSize",
                    "value": "5 Inch"
                },
                {
                    "name": "RAM",
                    "value": "8 GB"
                },
                {
                    "name": "diskSize",
                    "value": "64 GB"
                }
            ]
        }
    ]
}
```

```
        ],
        "stockValue": 39
    },
    {
        "attributes": [
            {
                "name": "screenSize",
                "value": "5 Inch"
            },
            {
                "name": "RAM",
                "value": "8 GB"
            },
            {
                "name": "diskSize",
                "value": "128 GB"
            }
        ],
        "stockValue": 23
    }
],
"description": "Uncover the Ultimate Design. Beauty Beyond the Visual",
"name": "Iphone 8",
"price": 4700,
"originalPrice": 5000,
"imageUrl":
"https://elasticbeanstalk-us-east-2-334058266782.s3.amazonaws.com/images/160899074721010",
"rating": 4.8,
"brand": "Apple",
"vendor": {
    "name": "Pablos",
    "rating": 2.43,
    "id": "3129281hasda91892312"
},
"id": "5fe86281a17abd0012d601ea"
},
{
    "category": [
        "Electronics",
        "Phones"
    ],
    "productInfos": [
        {
            "attributes": [
                {
                    "name": "screenSize",
                    "value": "6 Inch"
                },
                {
                    "name": "RAM",
                    "value": "8 GB"
                },
                {
                    "name": "diskSize",
                    "value": "64 GB"
                }
            ]
        }
    ]
}
```

```

        }
    ],
    "stockValue": 95
},
{
    "attributes": [
        {
            "name": "screenSize",
            "value": "6 Inch"
        },
        {
            "name": "RAM",
            "value": "8 GB"
        },
        {
            "name": "diskSize",
            "value": "128 GB"
        }
    ],
    "stockValue": 95
}
],
"description": "Uncover the Ultimate Design. Beauty Beyond the Visual",
"name": "Samsung S10",
"price": 4500,
"originalPrice": 5000,
"imageUrl":
"https://elasticbeanstalk-us-east-2-334058266782.s3.amazonaws.com/images/16089907472078",
"rating": 4.4,
"brand": "Samsung",
"vendor": {
    "name": "AyseTeyze",
    "rating": 3.21,
    "id": "3129281hasda91892312"
},
"id": "5fe86281a17abd0012d601e8"
},
{
    "category": [
        "Electronics",
        "Phones"
    ],
    "productInfos": [
        {
            "attributes": [
                {
                    "name": "screenSize",
                    "value": "5 Inch"
                },
                {
                    "name": "RAM",
                    "value": "4 GB"
                },
                {
                    "name": "diskSize",
                    "value": "64 GB"
                }
            ]
        }
    ]
}

```

```
        "value": "64 GB"
    }
],
"stockValue": 47
},
{
"attributes": [
{
"name": "screenSize",
"value": "5 Inch"
},
{
"name": "RAM",
"value": "4 GB"
},
{
"name": "diskSize",
"value": "128 GB"
}
],
"stockValue": 94
}
],
"description": "Uncover the Ultimate Design. Beauty Beyond the Visual",
"name": "Samsung S7",
"price": 4500,
"originalPrice": 5000,
"imageUrl":
"https://elasticbeanstalk-us-east-2-334058266782.s3.amazonaws.com/images/16089907472099",
"rating": 3.5,
"brand": "Samsung",
"vendor": {
"name": "JohnsShop",
"rating": 4.23,
"id": "3129281hasda91892312"
},
"id": "5fe86281a17abd0012d601e9"
},
{
"category": [
"Electronics",
"Phones"
],
"productInfos": [
{
"attributes": [
{
"name": "screenSize",
"value": "5.5 Inch"
},
{
"name": "RAM",
"value": "4 GB"
}
]
}
```

```
        "name": "diskSize",
        "value": "64 GB"
    },
],
"stockValue": 21
},
{
"attributes": [
{
    "name": "screenSize",
    "value": "5.5 Inch"
},
{
    "name": "RAM",
    "value": "4 GB"
},
{
    "name": "diskSize",
    "value": "128 GB"
}
],
"stockValue": 18
}
],
"description": "Uncover the Ultimate Design. Beauty Beyond the Visual",
"name": "Samsung S8",
"price": 2800,
"originalPrice": 3000,
"imageUrl":
"https://elasticbeanstalk-us-east-2-334058266782.s3.amazonaws.com/images/16089907472046",
"rating": 4,
"brand": "Samsung",
"vendor": {
    "name": "JohnsShop",
    "rating": 4.23,
    "id": "3129281hasda91892312"
},
"id": "5fe86281a17abd0012d601e6"
},
{
"category": [
"Electronics",
"Phones"
],
"productInfos": [
{
"attributes": [
{
    "name": "screenSize",
    "value": "5.5 Inch"
},
{
    "name": "RAM",
    "value": "8 GB"
}
],
```

```

        {
            "name": "diskSize",
            "value": "64 GB"
        }
    ],
    "stockValue": 54
},
{
    "attributes": [
        {
            "name": "screenSize",
            "value": "5.5 Inch"
        },
        {
            "name": "RAM",
            "value": "8 GB"
        },
        {
            "name": "diskSize",
            "value": "128 GB"
        }
    ],
    "stockValue": 74
}
],
"description": "Uncover the Ultimate Design. Beauty Beyond the Visual",
"name": "Samsung S9",
"price": 4000,
"originalPrice": 4200,
"imageUrl":
"https://elasticbeanstalk-us-east-2-334058266782.s3.amazonaws.com/images/16089907472057",
"rating": 4.4,
"brand": "Samsung",
"vendor": {
    "name": "Ahmet",
    "rating": 3.22,
    "id": "3129281hasda91892312"
},
"id": "5fe86281a17abd0012d601e7"
}
],
"filterCriterias": [
{
    "name": "screenSize",
    "displayName": "Screen Size",
    "possibleValues": [
        "5.5 Inch",
        "6 Inch",
        "5 Inch"
    ]
},
{
    "name": "RAM",
    "displayName": "RAM",
    "possibleValues": [

```

```
        "4 GB",
        "8 GB"
    ],
},
{
    "name": "diskSize",
    "displayName": "Disk Size",
    "possibleValues": [
        "64 GB",
        "128 GB"
    ]
}
]
}
```

## Change Password

Changes password of a user.

Endpoint: /account-change-password

Method: POST

Authorization: None

Parameters: userType=customer&id=5fe79b54500d4000191358c5&password=3244892

Response: { "status": { "code": 200, "message": "OK" } }

## Update Profile Info

Updates information of a user.

Endpoint: /account

Method: POST

Authorization: None

Parameters:

userType=customer&email=jim.gmail@hotmail.com&id=5fe79b54500d4000191358c5&name=Jim&surname=Morrison&gender=male&phoneNumber=3244892

Response: { "status": { "code": 200, "message": "OK" } }

## Get Last Messages

Endpoint: /messages/last

Method: GET

Authorization: None

Parameters: userType=customer&id=5fff448a04ba0e00144ec688

Response:

```
{  
  "status": {  
    "code": 200,  
    "message": "OK"  
  },  
  "data": {  
    "lastMessages": [  
      {  
        "user": {  
          "name": "Koray",  
          "userType": "customer",  
          "id": "5fff449804ba0e00144ec689"  
        },  
        "lastMessage": "Test message",  
        "date": "2021-01-13T19:18:00.345Z"  
      }  
    ]  
  }  
}
```

## Get Messages

Endpoint: /messages

Method: GET

Authorization: None

Parameters:

id=5fff448a04ba0e00144ec688&userType=customer&withId=5fff449804ba0e00144ec689&withType=customer

Response:

```
{  
  "status": {  
    "code": 200,  
    "message": "OK"  
  },  
  "data": {  
    "messages": [  
      {  
        "message": "Test message",  
        "date": "2021-01-13T19:18:00.345Z",  
        "id": "5fff4768773c3e001488eb75",  
        "user": {  
          "name": "Koray",  
          "userType": "customer",  
          "id": "5fff449804ba0e00144ec689"  
        }  
      }  
    ]  
  }  
}
```

```

    "user": {
      "id": "5fff448a04ba0e00144ec688",
      "name": "Koray",
      "userType": "customer"
    }
  },
  {
    "message": "Test message",
    "date": "2021-01-13T19:17:28.002Z",
    "id": "5fff4748773c3e001488eb74",
    "user": {
      "id": "5fff449804ba0e00144ec689",
      "name": "Olcay",
      "userType": "customer"
    }
  },
  {
    "message": "Test message",
    "date": "2021-01-13T19:16:31.660Z",
    "id": "5fff470fd3e781001467add6",
    "user": {
      "id": "5fff449804ba0e00144ec689",
      "name": "Olcay",
      "userType": "customer"
    }
  },
  {
    "message": "Test message",
    "date": "2021-01-13T19:15:13.527Z",
    "id": "5fff46c1ac554e0013ce59e6",
    "user": {
      "id": "5fff449804ba0e00144ec689",
      "name": "Koray",
      "userType": "customer"
    }
  },
  {
    "message": "Test message",
    "date": "2021-01-13T19:14:07.595Z",
    "id": "5fff467fa0a8ce00141ec450",
    "user": {
      "id": "5fff449804ba0e00144ec689",
      "name": "Koray",
      "userType": "customer"
    }
  }
]
}

```

## Socket Messaging

### Server URL

<http://3.141.25.245:5003/>

Server gets user id with this event.

Event name: discover

Parameters: (payload, responseFn)

Payload: { "userType": "customer" | "vendor" | "admin", "id": "5ff37169542dbb00123390fd" }

Response function: function (response) { // handle response}

Response: { "status": { "code": 200, "message": "OK" } }

Send message

Event name: message

Parameters: (payload, responseFn)

Payload: { "userType": "customer" | "vendor" | "admin", "id": "5ff37169542dbb00123390fd", "withType": "customer" | "vendor" | "admin", "withId": "5ff37169542dbb00123390fd", "message": "Some message" }

Response function: function (response) { // handle response}

Response: { "status": { "code": 200, "message": "OK" } }

## Vendor : Get Products

<http://3.141.25.245:5003/>

Endpoint: /vendor/products

Method: GET

Authorization: None

Parameters: 600bdf7f496a03001ce0e071 Body: {"categories": ["Earphone"]} (You can filter by using other attributes as well) \*\* Example request:

<http://3.141.25.245:5003/vendor/products/600bdf7f496a03001ce0e071>. (This id might not exist in deployed one.)

```
{
  "status": {
    "code": 200,
    "message": "OK"
  },
  "data": {
    "result": {
      "productList": [
        {
          "name": "Earphone A",
          "category": "Earphone",
          "price": 100
        },
        {
          "name": "Earphone B",
          "category": "Earphone",
          "price": 120
        }
      ]
    }
  }
}
```

```
"category": [
    "Electronics",
    "Earphone"
],
"sizes": [],
"colors": [],
"description": "Powerful sound & unique enjoyment!",
"name": "Sony Earphone XS-II",
"price": 400,
"originalPrice": 600,
"imageUrl": "https://elasticbeanstalk-us-east-2-334058266782.s3.amazonaws.com/images/160899074722117",
"rating": 4.3,
"brand": "Sony",
"productInfos": [
    {
        "attributes": [
            {
                "name": "noiseCancelling",
                "value": "No"
            }
        ],
        "stockValue": 38
    }
],
"vendor": {
    "name": "Pablos",
    "rating": 2.43,
    "id": "3129281hasda91892312"
},
{
    "category": [
        "Electronics",
        "Earphone"
    ],
    "sizes": [],
    "colors": [],
    "description": "Powerful sound & unique enjoyment!",
    "name": "Philips Earphone P-I",
    "price": 400,
    "originalPrice": 400,
    "imageUrl": "https://elasticbeanstalk-us-east-2-334058266782.s3.amazonaws.com/images/160899074722318",
    "rating": 4.1,
    "brand": "Philips",
    "productInfos": [
        {
            "attributes": [
                {
                    "name": "noiseCancelling",
                    "value": "No"
                }
            ],
        }
    ]
},
```

```

        "stockValue": 31
    }
],
"vendor": {
    "name": "Pablos",
    "rating": 2.43,
    "id": "3129281hasda91892312"
},
"id": "600bdf7f496a03001ce0e0b1"
}
],
"filterCriterias": [
{
    "name": "noiseCancelling",
    "displayName": "Noise Cancelling",
    "possibleValues": [
        "No"
    ]
}
]
}
}
}

```

## Vendor : Update Products

<http://3.141.25.245:5003/>

Endpoint: /vendor/products

Method: PATCH

Authorization: None

Parameters: 600bdf7f496a03001ce0e071 Body: {"name": "Iphone 12"} (You can change other first nested attributes as well) \*\* Example request:

<http://3.141.25.245:5003/vendor/products/600bdf7f496a03001ce0e071>. (This product id might not exist in deployed one.) Body Parameter1:

```
{
    "attributes": [
        {
            "name": "size",
            "value": "42"
        },
        {
            "name": "color",
            "value": "Black"
        }
    ],
    "stockValue": 1000000
}
```

```
}
```

Body Parameter2 - AS an alternative:

```
{
    "name": "Buyo Best Product"
}
```

Body Parameter3 - As an alternative:

```
{
    "price": 99999
}
```

Response:

```
{
    "status": {
        "code": 200,
        "message": "OK"
    },
    "data": {
        "result": {
            "category": [
                "Electronics",
                "Computers"
            ],
            "sizes": [],
            "colors": [],
            "_id": "600bdf7f496a03001ce0e0a2",
            "description": "Engineered for long-lasting performance. Comfortable confidence.",
            "name": "Macbook Pro",
            "price": 6000,
            "originalPrice": 6200,
            "imageUrl": "",
            "rating": 4.8,
            "brand": "Macbook",
            "productInfos": [
                {
                    "attributes": [
                        {
                            "name": "screenSize",
                            "value": "13 Inch"
                        },
                        {
                            "name": "RAM",
                            "value": "8 GB"
                        },
                        {
                            "name": "diskSize",
                            "value": "256 GB"
                        }
                    ]
                }
            ]
        }
    }
}
```

```
        }
    ],
    "stockValue": 53
},
{
    "attributes": [
        {
            "name": "screenSize",
            "value": "13 Inch"
        },
        {
            "name": "RAM",
            "value": "8 GB"
        },
        {
            "name": "diskSize",
            "value": "512 GB"
        }
    ],
    "stockValue": 29
},
{
    "attributes": [
        {
            "name": "screenSize",
            "value": "13 Inch"
        },
        {
            "name": "RAM",
            "value": "16 GB"
        },
        {
            "name": "diskSize",
            "value": "256 GB"
        }
    ],
    "stockValue": 92
},
{
    "attributes": [
        {
            "name": "screenSize",
            "value": "13 Inch"
        },
        {
            "name": "RAM",
            "value": "16 GB"
        },
        {
            "name": "diskSize",
            "value": "512 GB"
        }
    ],
    "stockValue": 46
}
```

```

        ],
        "vendorId": "600bdf7f496a03001ce0e06e"
    }
}

```

## Vendor : WHOLE Products : It takes whole product for updating product information

<http://3.141.25.245:5003/>

Endpoint: /vendor/wholeproducts

Method: PATCH

Authorization: None

Parameters: 600bdf7f496a03001ce0e071

\*\* Example request:

<http://3.141.25.245:5003/vendor/wholeproducts/600bdf7f496a03001ce0e071>. (This product id might not exist in deployed one.) Body Parameter:

```
{"category":["Women Clothing","987987"],"description":"Nice orange blouse. It's a causal and comfortable product you'll use in daily life. 987987 lo lo lo ","name":"Women Orange Casual Blouse olsun 987 987 ","price":987987,"originalPrice":180.12,"imageUrl":"https://elasticbeanstalk-us-east-2-334058266782.s3.amazonaws.com/images/16089146596972","rating":4.86,"brand":"Coco Chanel","id":"600de877ce666a001911ea13","productInfos":[{"attributes":[{"name":"size","value":"XS"}, {"name":"color","value":"Orange"}],"stockValue":987987}, {"attributes":[{"name":"size","value":"S987987"}, {"name":"color","value":"Orange"}],"stock4Value":987987}],"vendor":{"name":"JohnsShop","rating":4.23}}
```

Response:

```
{
  "status": {
    "code": 200,
    "message": "OK"
  },
  "data": {
    "result": {
      "category": [
        "Women Clothing",
        "987987"
      ],
      "description": "Nice orange blouse. It's a causal and comfortable product you'll use in daily life. 987987 lo lo lo ",
      "name": "Women Orange Casual Blouse olsun 987 987 ",
      "price": 987987,
      "originalPrice": 180.12,
      "imageUrl": "https://elasticbeanstalk-us-east-2-334058266782.s3.amazonaws.com/images/16089146596972"
    }
  }
}
```

```

  "imageUrl": "https://elasticbeanstalk-us-east-2-334058266782.s3.amazonaws.com/images/16089146596972",
  "rating": 4.86,
  "brand": "Coco Chanel",
  "id": "600de877ce666a001911ea13",
  "productInfos": [
    {
      "attributes": [
        {
          "name": "size",
          "value": "XS"
        },
        {
          "name": "color",
          "value": "Orange"
        }
      ],
      "stockValue": 987987
    },
    {
      "attributes": [
        {
          "name": "size",
          "value": "S987987"
        },
        {
          "name": "color",
          "value": "Orange"
        }
      ],
      "stock4Value": 987987
    }
  ],
  "vendor": {
    "name": "JohnsShop",
    "rating": 4.23
  }
}
}

```

## Vendor : Add Products

<http://3.141.25.245:5003/>

Endpoint: /vendor/products

Method: POST

Authorization: None

Parameters: 600bdf7f496a03001ce0e071 (vendorId) Body: It should be product list. You should add this exact product array to the body.

```
[{
    "category": [
        "Electronics",
        "Computers"
    ],
    "description": "Engineered for long-lasting performance. Comfortable confidence.",
    "name": "Macbook Pro",
    "price": 6000,
    "originalPrice": 6200,
    "imageUrl": "",
    "rating": 4.8,
    "brand": "Macbook",
    "productInfos": [
        {
            "attributes": [
                {
                    "name": "screenSize",
                    "value": "13 Inch"
                },
                {
                    "name": "RAM",
                    "value": "8 GB"
                },
                {
                    "name": "diskSize",
                    "value": "256 GB"
                }
            ],
            "stockValue": 53
        },
        {
            "attributes": [
                {
                    "name": "screenSize",
                    "value": "13 Inch"
                },
                {
                    "name": "RAM",
                    "value": "8 GB"
                },
                {
                    "name": "diskSize",
                    "value": "512 GB"
                }
            ],
            "stockValue": 29
        },
        {
            "attributes": [
                {
                    "name": "screenSize",
                    "value": "13 Inch"
                },
                {
                    "name": "RAM",
                    "value": "8 GB"
                },
                {
                    "name": "diskSize",
                    "value": "512 GB"
                }
            ],
            "stockValue": 29
        }
    ]
}]
```

```

        "value": "13 Inch"
    },
    {
        "name": "RAM",
        "value": "16 GB"
    },
    {
        "name": "diskSize",
        "value": "256 GB"
    }
],
"stockValue": 92
},
{
    "attributes": [
        {
            "name": "screenSize",
            "value": "13 Inch"
        },
        {
            "name": "RAM",
            "value": "16 GB"
        },
        {
            "name": "diskSize",
            "value": "512 GB"
        }
],
"stockValue": 46
}
],
"vendorId": "600bdf7f496a03001ce0e06e"
}]

```

(You can change other first nested attributes as well) \*\* Example request:

<http://3.141.25.245:5003/vendor/products/600bdf7f496a03001ce0e071> (This vendor id might not exist in deployed one.) \*\* Response:

{"idList" :["1","2"] (Ids of added products)}

## Vendor : Delete Products

<http://3.141.25.245:5003/>

Endpoint: /vendor/product

Method: POST

Authorization: None

Parameters: 600bdf7f496a03001ce0e071 (vendorId) Body: It should be product list. You should add this exact product array to the body.

```
{  
    "productId": "t3f243434g3"  
}
```

(You can change other first nested attributes as well) \*\* Example request:  
<http://3.141.25.245:5003/vendor/product/600bdf7f496a03001ce0e071> (This vendor id might not exist in deployed one.)

```
{  
    "status": {  
        "code": 200,  
        "message": "The product is deleted successfully"  
    }  
}  
  
{  
    "status": {  
        "code": 400,  
        "message": "Please check your product information. It has never existed or been already deleted"  
    }  
}
```

## Upload photo:

<http://3.141.25.245:8080/>

Endpoint: /file

Method: POST

Authorization: None

Request body form-data:

photoName(unused): file object Response:

```
{  
    status: { code: 200, message: "Success" },  
    data: {  
        urls: ["https://192.83.u9128ejjasidklasmdsa.com/tr"],  
    },  
}
```

## Recommendations:

<http://3.141.25.245:8080/>

Endpoint: /products/recommendation

Method: GET

Authorization: None

Request Params:

userId: String,

type: alsoPurchased | alsoLiked

Response:

```
{
  "status": {
    "code": 200,
    "message": "OK"
  },
  "data": {
    "productList": [
      {
        "category": [
          "Electronics",
          "Earphone"
        ],
        "productInfos": [
          {
            "attributes": [
              {
                "name": "noiseCancelling",
                "value": "No"
              }
            ],
            "stockValue": 31
          }
        ],
        "description": "Powerful sound & unique enjoyment!",
        "name": "Philips Earphone P-I",
        "price": 400,
        "originalPrice": 400,
        "imageUrl":
        "https://elasticbeanstalk-us-east-2-334058266782.s3.amazonaws.com/images/160899074722318",
        "rating": 4.1,
        "brand": "Philips",
        "vendor": {
          "name": "AyseTeyze",
          "rating": 1.471203743810447,
          "id": "5ff37169542dbb00123390fc"
        },
        "id": "5ff37169542dbb001233913d"
      }
    ]
  }
}
```

```
{  
  "category": [  
    "Electronics",  
    "Computers"  
  ],  
  "productInfos": [  
    {  
      "attributes": [  
        {  
          "name": "screenSize",  
          "value": "15 Inch"  
        },  
        {  
          "name": "RAM",  
          "value": "4 GB"  
        },  
        {  
          "name": "diskSize",  
          "value": "256 GB"  
        }  
      ],  
      "stockValue": 36  
    },  
    {  
      "attributes": [  
        {  
          "name": "screenSize",  
          "value": "15 Inch"  
        },  
        {  
          "name": "RAM",  
          "value": "4 GB"  
        },  
        {  
          "name": "diskSize",  
          "value": "512 GB"  
        }  
      ],  
      "stockValue": 28  
    },  
    {  
      "attributes": [  
        {  
          "name": "screenSize",  
          "value": "15 Inch"  
        },  
        {  
          "name": "RAM",  
          "value": "4 GB"  
        },  
        {  
          "name": "diskSize",  
          "value": "1024 GB"  
        }  
      ],  
    }  
  ]  
}
```

```
        "stockValue": 26
    },
    {
        "attributes": [
            {
                "name": "screenSize",
                "value": "15 Inch"
            },
            {
                "name": "RAM",
                "value": "8 GB"
            },
            {
                "name": "diskSize",
                "value": "256 GB"
            }
        ],
        "stockValue": 33
    },
    {
        "attributes": [
            {
                "name": "screenSize",
                "value": "15 Inch"
            },
            {
                "name": "RAM",
                "value": "8 GB"
            },
            {
                "name": "diskSize",
                "value": "512 GB"
            }
        ],
        "stockValue": 74
    },
    {
        "attributes": [
            {
                "name": "screenSize",
                "value": "15 Inch"
            },
            {
                "name": "RAM",
                "value": "8 GB"
            },
            {
                "name": "diskSize",
                "value": "1024 GB"
            }
        ],
        "stockValue": 96
    },
    {
        "attributes": [
```

```
{  
    "name": "screenSize",  
    "value": "15 Inch"  
},  
{  
    "name": "RAM",  
    "value": "16 GB"  
},  
{  
    "name": "diskSize",  
    "value": "256 GB"  
}  
],  
"stockValue": 43  
},  
{  
    "attributes": [  
        {  
            "name": "screenSize",  
            "value": "15 Inch"  
        },  
        {  
            "name": "RAM",  
            "value": "16 GB"  
        },  
        {  
            "name": "diskSize",  
            "value": "512 GB"  
        }  
],  
"stockValue": 100  
},  
{  
    "attributes": [  
        {  
            "name": "screenSize",  
            "value": "15 Inch"  
        },  
        {  
            "name": "RAM",  
            "value": "16 GB"  
        },  
        {  
            "name": "diskSize",  
            "value": "1024 GB"  
        }  
],  
"stockValue": 5  
}  
],  
"description": "Engineered for long-lasting performance. Comfortable confidence.",  
"name": "HP Pavilion Intel Pentium",  
"price": 3800,  
"originalPrice": 4500,
```

"imageUrl":  
"https://elasticbeanstalk-us-east-2-334058266782.s3.amazonaws.com/images/16089907472025",  
"rating": 3.8,  
"brand": "HP",  
"vendor": {  
    "name": "AyseTeyze",  
    "rating": 1.471203743810447,  
    "id": "5ff37169542dbb00123390fc"  
},  
"id": "5ff37169542dbb001233912f"  
},  
{  
    "category": [  
        "Men",  
        "Shoes",  
        "Sneakers"  
    ],  
    "productInfos": [  
        {  
            "attributes": [  
                {  
                    "name": "size",  
                    "value": "40"  
                },  
                {  
                    "name": "color",  
                    "value": "Red"  
                }  
            ],  
            "stockValue": 1  
        },  
        {  
            "attributes": [  
                {  
                    "name": "size",  
                    "value": "41"  
                },  
                {  
                    "name": "color",  
                    "value": "Red"  
                }  
            ],  
            "stockValue": 86  
        },  
        {  
            "attributes": [  
                {  
                    "name": "size",  
                    "value": "42"  
                },  
                {  
                    "name": "color",  
                    "value": "Red"  
                }  
            ],  
            "stockValue": 100  
        }  
    ]  
}

```
        "stockValue": 59
    },
    {
        "attributes": [
            {
                "name": "size",
                "value": "43"
            },
            {
                "name": "color",
                "value": "Red"
            }
        ],
        "stockValue": 3
    },
    {
        "attributes": [
            {
                "name": "size",
                "value": "44"
            },
            {
                "name": "color",
                "value": "Red"
            }
        ],
        "stockValue": 87
    },
    {
        "attributes": [
            {
                "name": "size",
                "value": "45"
            },
            {
                "name": "color",
                "value": "Red"
            }
        ],
        "stockValue": 55
    }
],
"description": "Men colorful sneakers. You won't feel your shoes and make many miles with them.",
"name": "Men Black and Red Sneakers",
"price": 229.99,
"originalPrice": 339.99,
"imageUrl":
"https://elasticbeanstalk-us-east-2-334058266782.s3.amazonaws.com/images/160891465971013",
"rating": 4.03,
"brand": "Nike",
"vendor": {
    "name": "Pablos",
    "rating": 2.43,
    "id": "5ff37169542dbb00123390fd"
},
```

```
"id": "5ff37169542dbb001233910b"
},
{
  "category": [
    "Cosmetics",
    "Liquid Soap"
  ],
  "productInfos": [
    {
      "attributes": [
        {
          "name": "aroma",
          "value": "Rose"
        }
      ],
      "stockValue": 7
    },
    {
      "attributes": [
        {
          "name": "aroma",
          "value": "Vanilla"
        }
      ],
      "stockValue": 47
    },
    {
      "attributes": [
        {
          "name": "aroma",
          "value": "Strawberry"
        }
      ],
      "stockValue": 45
    }
  ],
  "description": "Quality liquid soap. 99.9% antibacterial.",
  "name": "Clean Liquid Soap",
  "price": 200.65,
  "originalPrice": 49.9,
  "imageUrl":
  "https://elasticbeanstalk-us-east-2-334058266782.s3.amazonaws.com/images/16089789244670",
  "rating": 3.6,
  "aromas": [
    "Rose",
    "Vanilla",
    "Strawberry"
  ],
  "brand": "Clean",
  "vendor": {
    "name": "AyseTeyze",
    "rating": 1.471203743810447,
    "id": "5ff37169542dbb00123390fc"
  },
  "id": "5ff37169542dbb0012339115"
```

```
},
{
  "category": [
    "Women Clothing",
    "Dress"
  ],
  "productInfos": [
    {
      "attributes": [
        {
          "name": "size",
          "value": "XS"
        },
        {
          "name": "color",
          "value": "Red"
        }
      ],
      "stockValue": 83
    },
    {
      "attributes": [
        {
          "name": "size",
          "value": "M"
        },
        {
          "name": "color",
          "value": "Red"
        }
      ],
      "stockValue": 62
    },
    {
      "attributes": [
        {
          "name": "size",
          "value": "L"
        },
        {
          "name": "color",
          "value": "Red"
        }
      ],
      "stockValue": 23
    }
  ],
  "description": "Women shiny red dress, makes you the most attractive person in the party.",
  "name": "Women Red Dress",
  "price": 112.99,
  "originalPrice": 142.99,
  "imageUrl":
  "https://elasticbeanstalk-us-east-2-334058266782.s3.amazonaws.com/images/160891465970811",
  "rating": 4,
  "brand": "Inci",
```

```

    "vendor": {
      "name": "Pablos",
      "rating": 2.43,
      "id": "5ff37169542dbb00123390fd"
    },
    "id": "5ff37169542dbb0012339109"
  }
}

```

## Get Notifications

Endpoint: /notifications

Method: GET

Authorization: None

Parameters: userType=customer&userId=5fe79b54500d4000191358c5

Response:

```
{
  "status": {
    "code": 200,
    "message": "OK"
  },
  "data": {
    "notifications" : {
      "@context": "https://www.w3.org/ns/activitystreams",
      "summary": "BUYO Notifications",
      "type": "Collection",
      "totalItems": 2,
      "items": [
        {
          "type": "Update",
          "name": "Discount",
          "startTime": "2014-12-31T23:00:00-08:00",
          "summary": "<vendor>'s <productName> price dropped from <originalPrice> to <price>",
          "actor": {
            "type": "Organization",
            "name": "<vendor_name>",
            "id": "<vendor_id>"
          },
          "target": "<product_id>",
        },
        {
          "type": "Update",
          "name": "Cancel Order",
          "startTime": "2014-12-31T23:00:00-08:00",
          "summary": "Your order <order_id> is cancelled by the customer!",
          "actor": {

```

```
        "type": "Person",
        "name": "<customer_email>",
        "id": "<customer_id>"
    },
    "target": "<order_id>",
}
]
}
}
}
```

## Admin : Vendor Change Status

Endpoint: /admin/vendor/changeStatus

Method: POST

Authorization: None

Parameters: /600db67dc0283100194f7ed8

\*\* Example Request \*\* :

<http://3.141.25.245:8080/admin/vendor/changeStatus/600db67dc0283100194f7ed8> Body

Parameter: Status can be "verified", "not-verified", "banned" Warning: If users status is already "banned" or "verified", our function will returned "Please check your parameters"

```
{"status": "verified", "productId": "1wqeqwe"}
```

Response:

```
{
  "status": {
    "code": 200,
    "message": "The vendor status is changed"
  }
}
```

## Admin : Customer Change Status

Endpoint: /admin/customer/changeStatus

Method: POST

Authorization: None

Parameters: /600db67dc0283100194f7ed8

\*\* Example Request \*\* :

<http://3.141.25.245:8080/admin/customer/changeStatus/600db67dc0283100194f7ed8> Body

Parameter : Status can be "verified", "not-verified", "banned" Warning: If users status is already "banned" or "verified", our function will returned "Please check your parameters"

```
{"status" : "banned", "commentId" : "23werewfwf"}
```

Response:

```
{
  "status": {
    "code": 200,
    "message": "The customer status is changed"
  }
}
```

## Admin : Get Comment Reports

Endpoint: /admin/report/product

Method: GET

Authorization: None

Parameters: No Parameter

\*\* Example Request \*\* : <http://3.141.25.245:8080/admin/report/product>

```
{"status" : "banned"}
```

Response:

```
{
  "status": {
    "code": 200,
    "message": "OK"
  },
  "data": {
    "productReport": [
      {
        "message": "\"This is an old product. Do better job!\"",
        "productDetails": {
          "category": [
            "Women Clothing",
            "Shoes",
            "Sneakers"
          ],
          "sizes": [],
          "colors": [],
          "_id": "600de877ce666a001911ea25",
          "description": "Women adidas shoes. It's most classic sneakers of adidas and proved itself since 2001."
        }
      }
    ]
}
```

```

    "name": "Women Adidas Black and White Sneakers",
    "price": 253.99,
    "originalPrice": 353.99,
    "imageUrl":
"https://elasticbeanstalk-us-east-2-334058266782.s3.amazonaws.com/images/160891465972020",
    "rating": 4.92,
    "brand": "Adidas",
    "productInfos":
[{"attributes": [{"name": "size", "value": "38"}, {"name": "color", "value": "Black"}], "stockValue": 1}, {"attributes": [{"name": "size", "value": "39"}, {"name": "color", "value": "Black"}], "stockValue": 3}, {"attributes": [{"name": "size", "value": "40"}, {"name": "color", "value": "Black"}], "stockValue": 6}], "vendorId": "600de877ce666a001911ea0d"
}
]
}
}
}

```

## Admin : Get Comment Reports

Endpoint: /admin/report/comment

Method: GET

Authorization: None

Parameters: No Parameter

\*\* Example Request \*\* : <http://3.141.25.245:8080/admin/report/comment>

```
{"status" : "banned"}
```

Response:

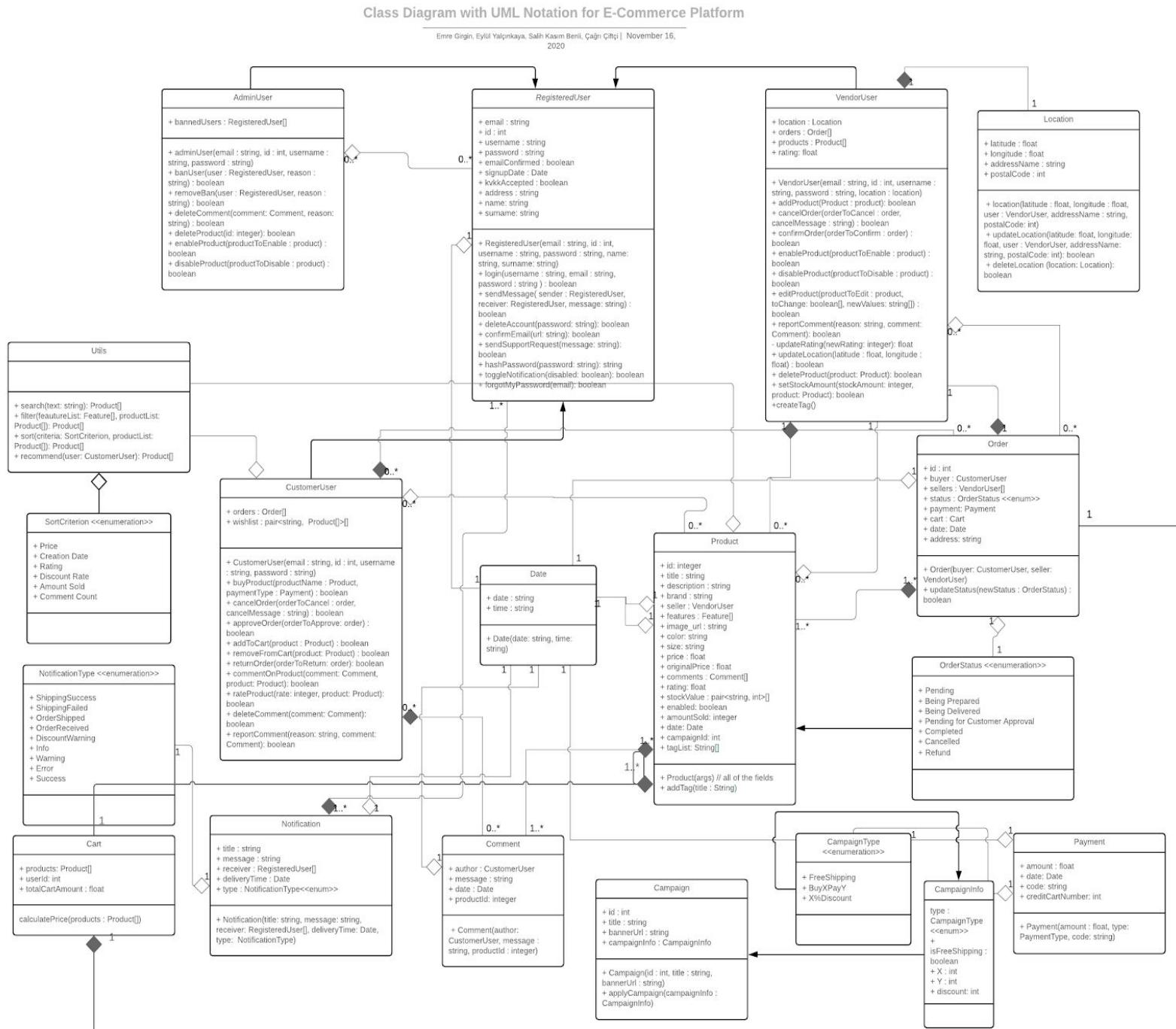
```
{
  "status": {
    "code": 200,
    "message": "OK"
  },
  "data": {
    "productReport": [
      {
        "message": "\"This is abusive comment\"",
        "commentDetails": {
          "userId": "600dasf3fe34t3tf01911ea0d",
          "productId": "600de877ce666a001911ea25",
          "vendorId": "600de877ce666a001911ea0d",
          "rating": 4.23,
          "text": "I hate kids."
        }
      }
    ]
  }
}
```

$$\left. \begin{array}{c} \\ \end{array} \right\}$$

## 8. Design Documents

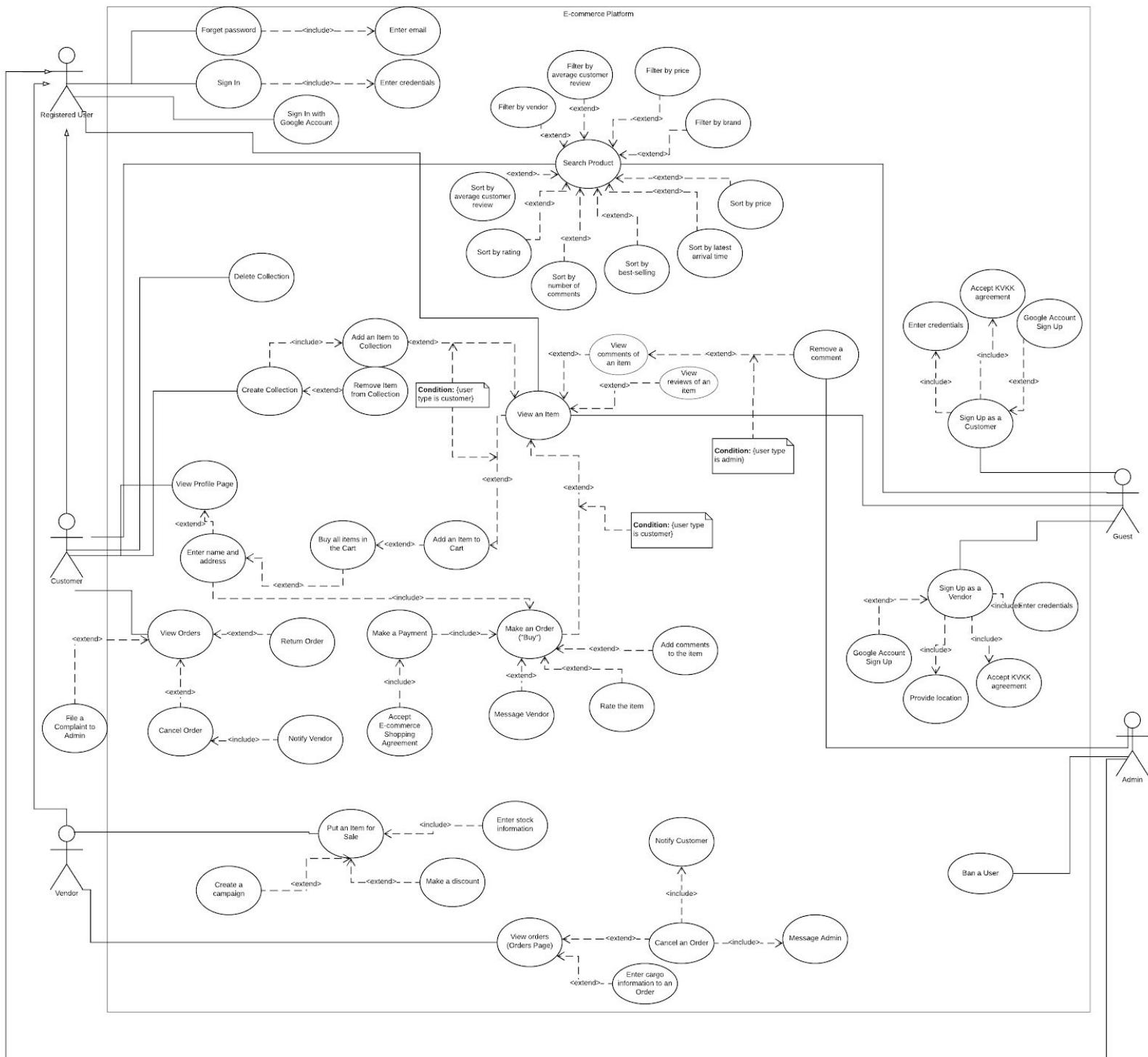
# Class Diagram

## Link



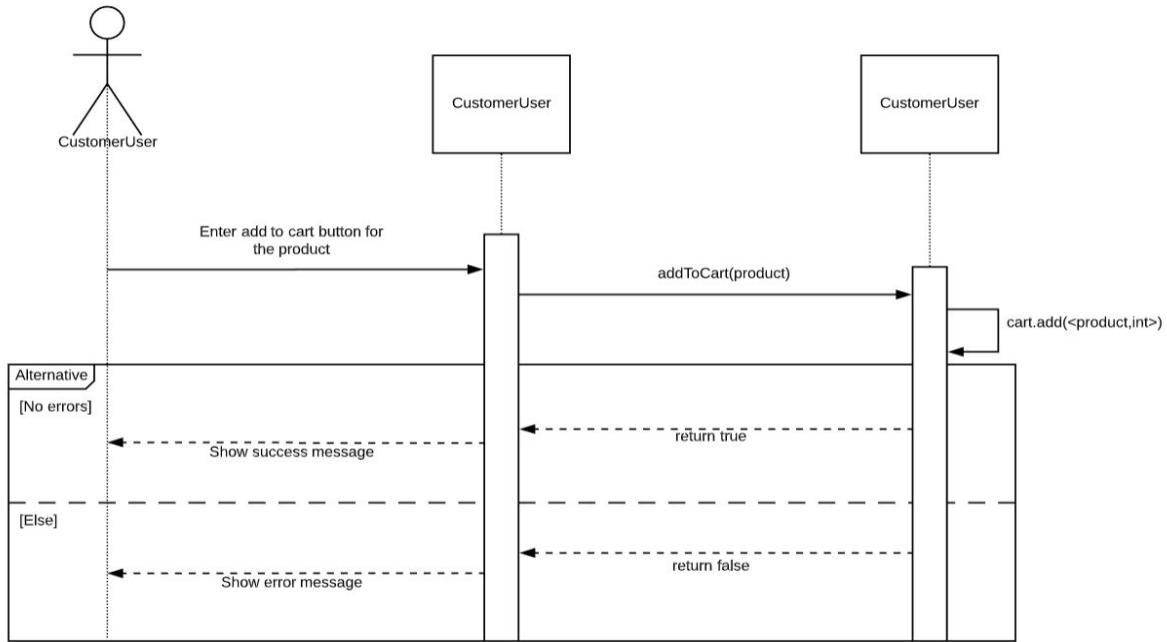
# Use Case Diagram

[Link](#)

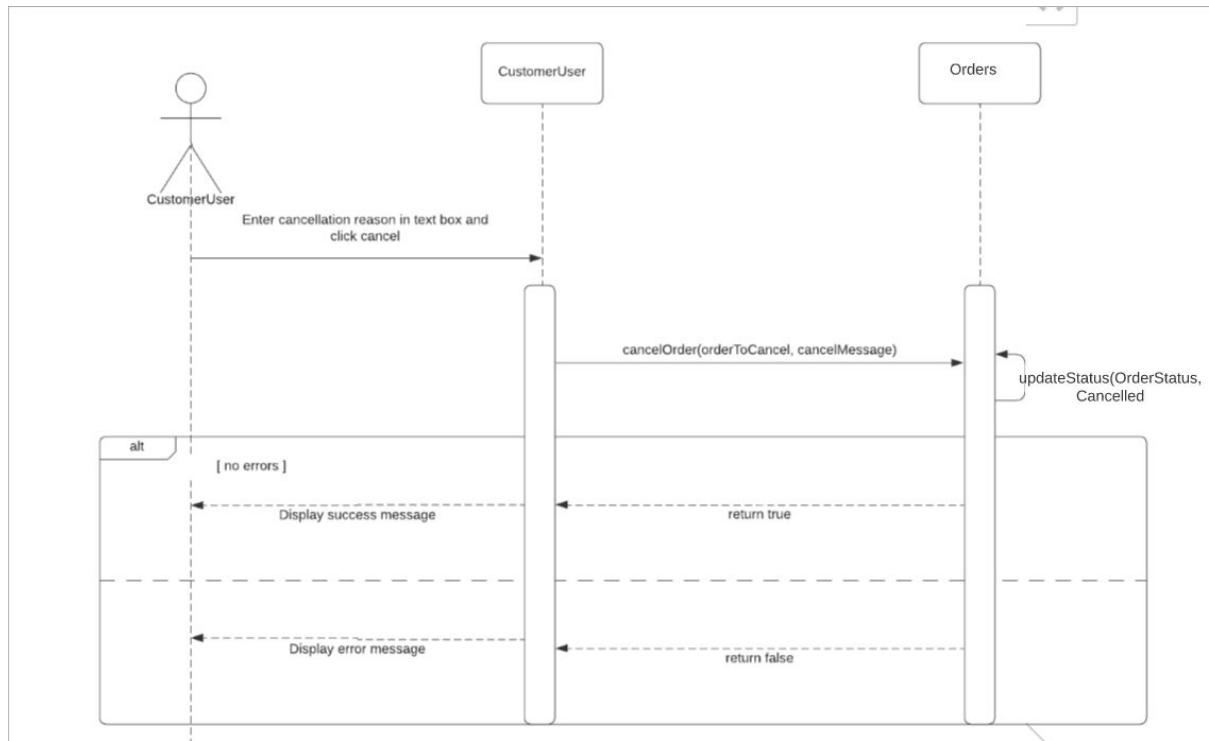


# Sequence Diagrams

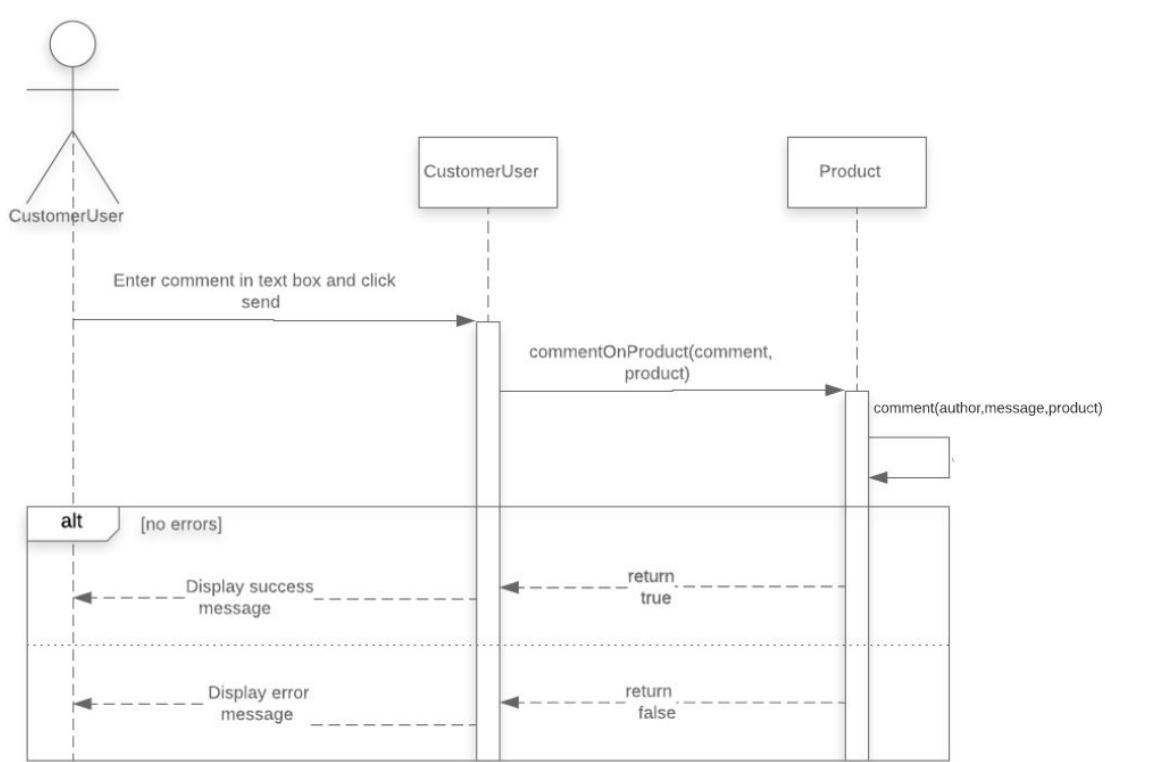
## Add Product to Cart



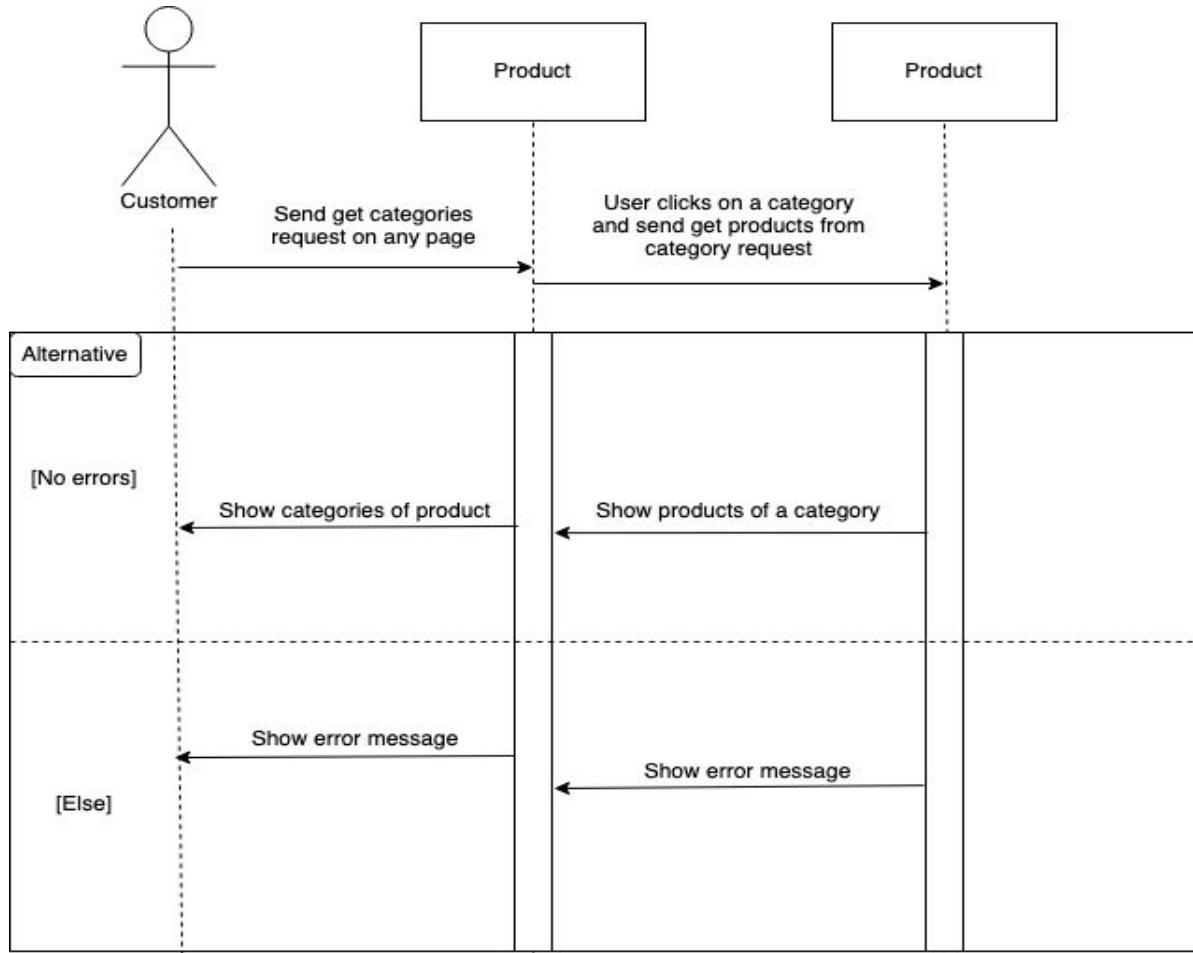
## Cancel Order



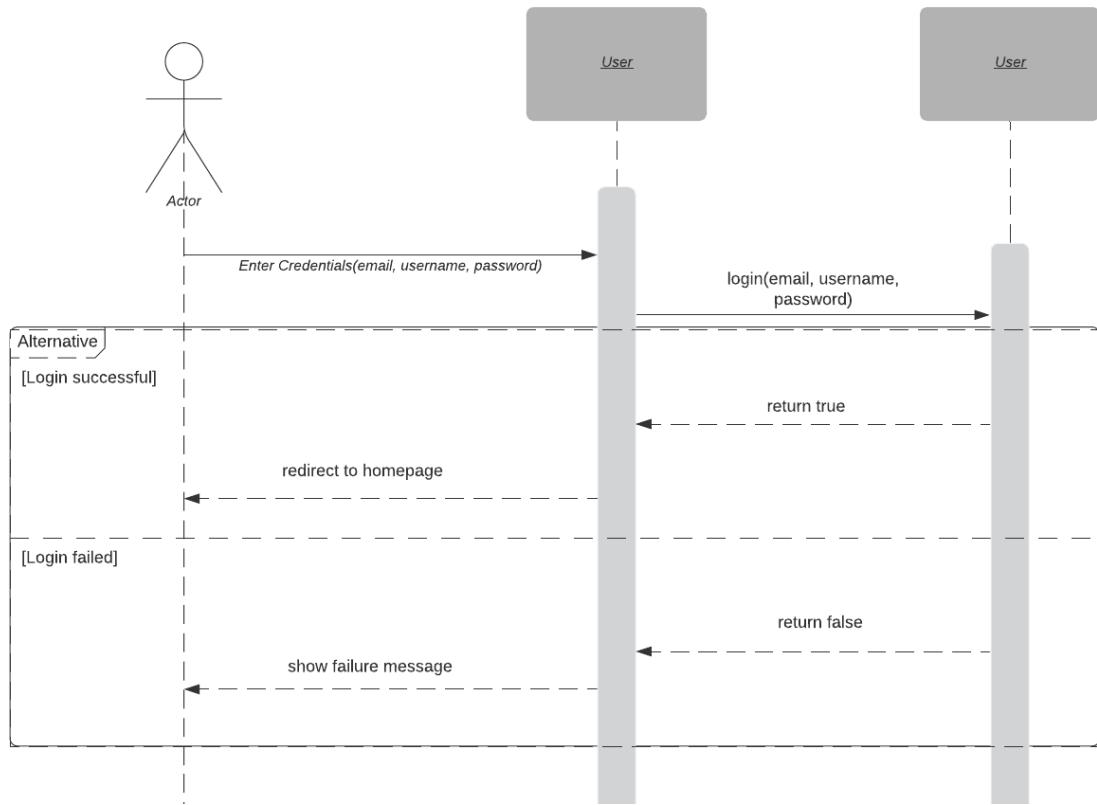
## Comment



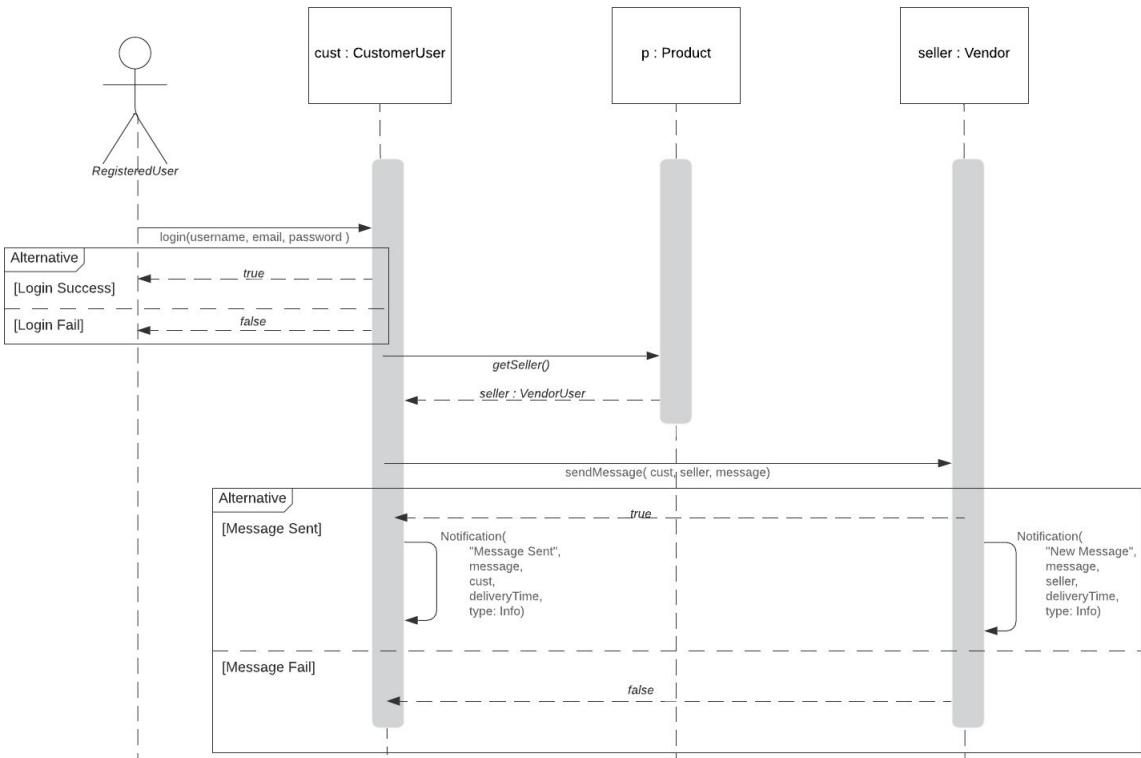
## Display Products of a Category



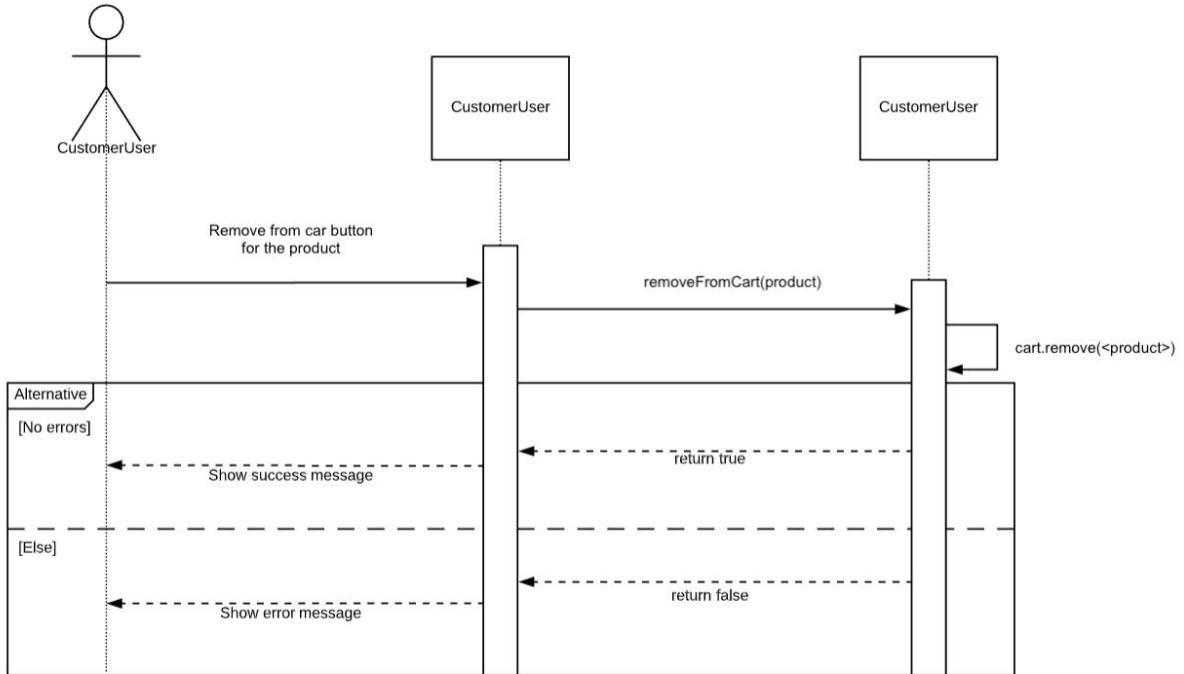
## Login



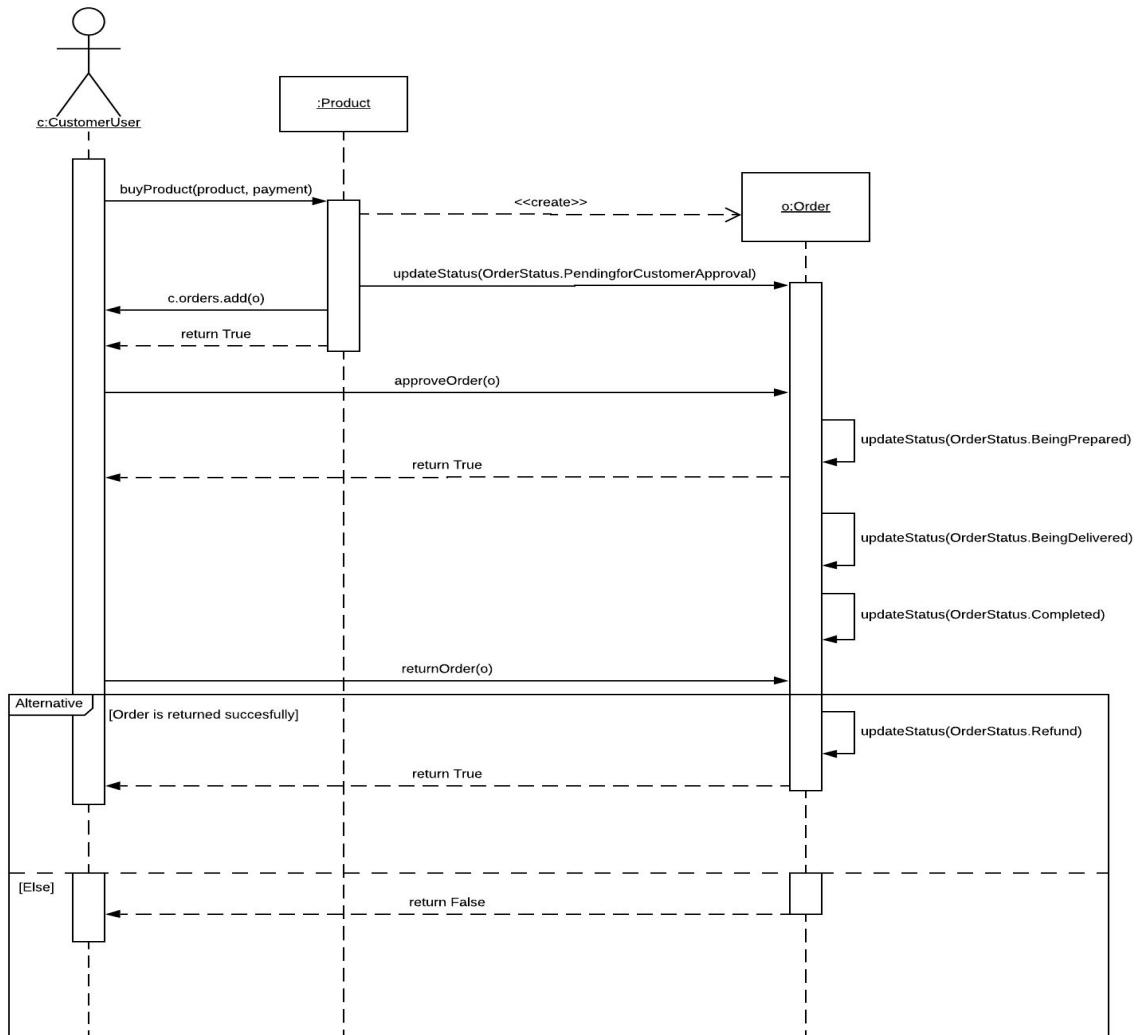
## Message



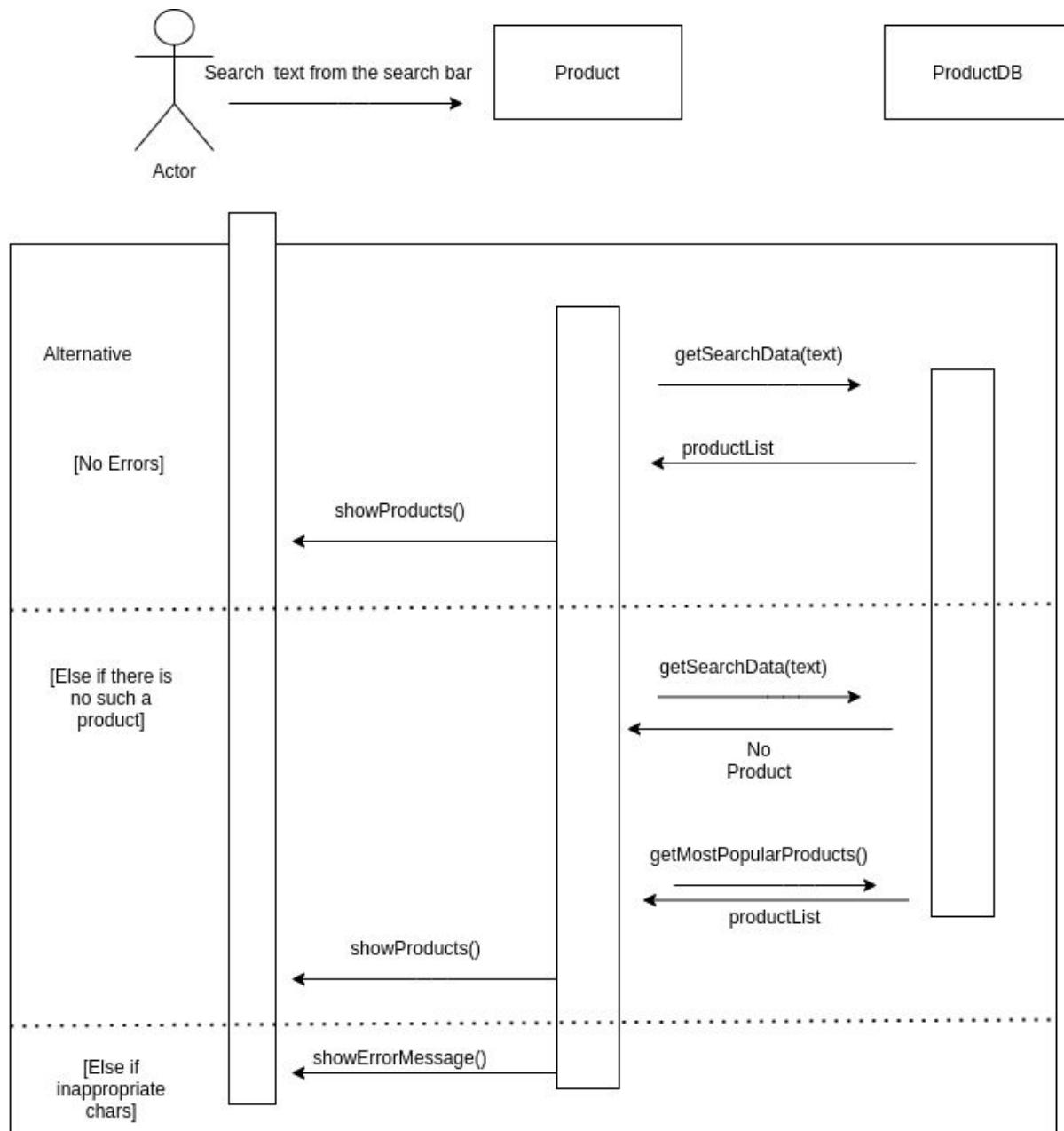
## Remove Product From the Cart



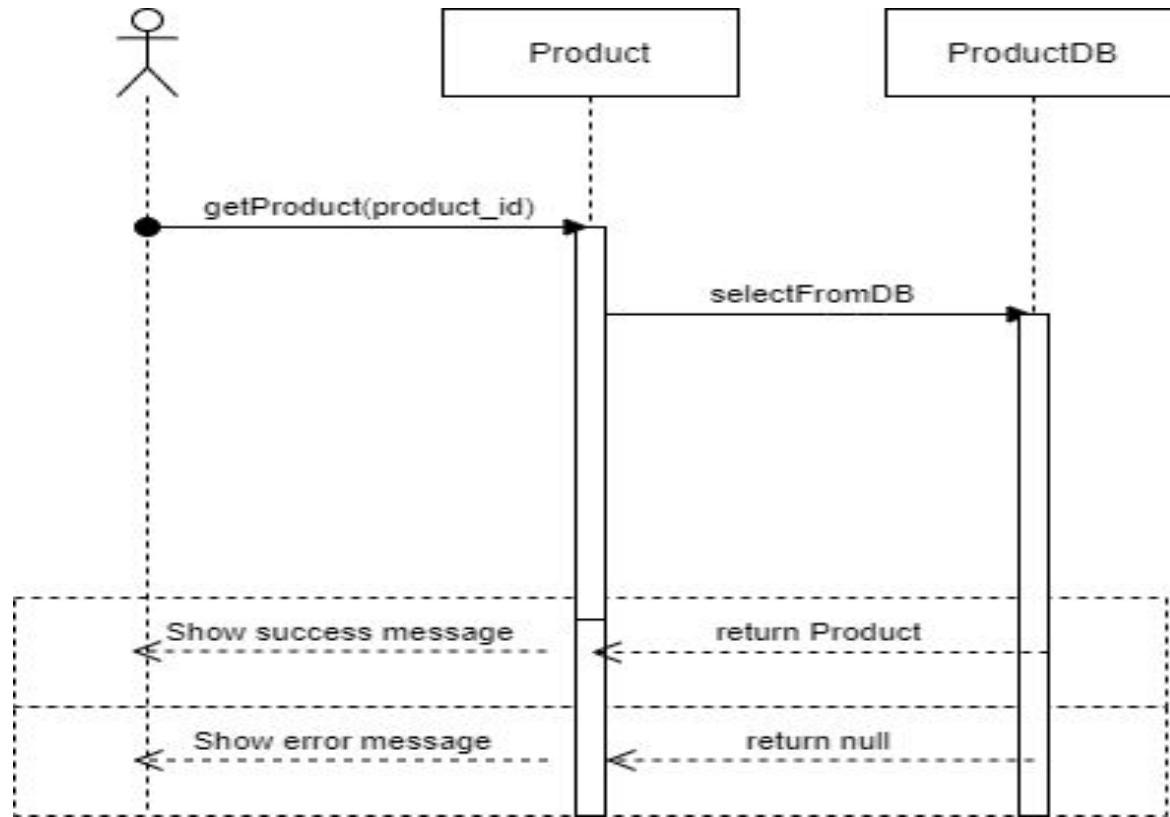
## Return Product



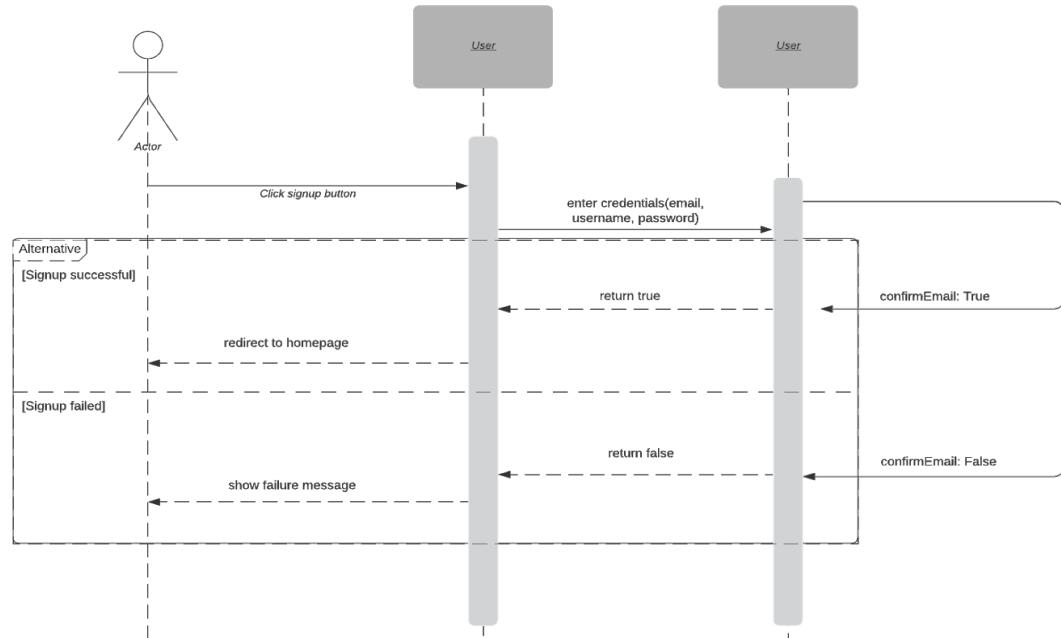
# Search



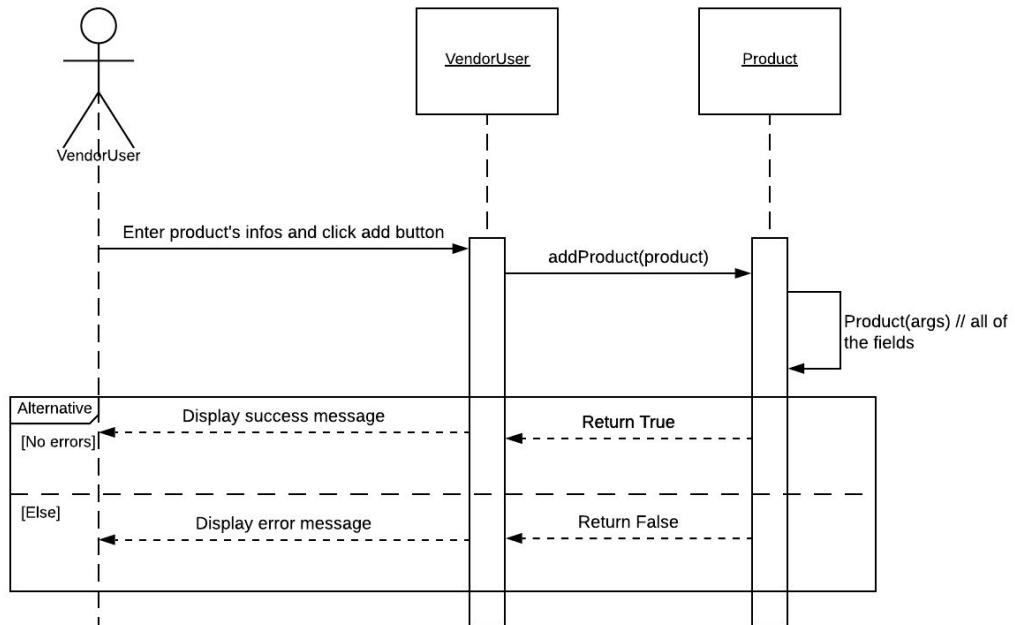
## Show Product



## Sign Up



## Vendor Adding Product



# 9. Project Plan

1	<input type="checkbox"/> W2: Requirements	7 günler	18.02.2020 08:00	24.02.2020 17:00		
2	Configuring Trello board	7 günler	18.02.2020 08:00	24.02.2020 17:00	Koray Cetin	
3	Researching W3C	7 günler	18.02.2020 08:00	24.02.2020 17:00	Mehmet Erdinç Oguz	
4	Extracting requirements from project description	4 günler	18.02.2020 09:00	22.02.2020 09:00	Eylül Yalcinkaya;Katarina Korolainen;Salih Kasim Benli	
5	Categorizing requirements	1 gün	22.02.2020 09:00	23.02.2020 09:00	4 Çağrı Çiftçi;Emre Girgin;Ölcayto Türker	
6	Formalizing requirements & glossary	1,5 günler	23.02.2020 09:00	24.02.2020 14:00	4;5 Berkay Alkan;Berke Özdemir;Burak Cuhadar	
7	Researching licensing	7 günler	18.02.2020 08:00	24.02.2020 17:00	Çağrı Çiftçi;Mehmet Erdinç Oguz	
8	<input type="checkbox"/> W3: Scenarios & Mockups	14 günler	25.02.2020 08:00	09.03.2020 17:00		
9	Fixing Requirements	7 günler	25.02.2020 08:00	02.03.2020 17:00	6 Burak Cuhadar;Çağrı Çiftçi	
10	Fixing Communication plan	7 günler	25.02.2020 08:00	02.03.2020 17:00	Eylül Yalcinkaya	
11	Web Scenario & Mockup for Customer	4 günler	27.02.2020 08:00	01.03.2020 17:00	Berkay Alkan;Ölcayto Türker	
12	Web Scenario & Mockup for Vendor	4 günler	27.02.2020 08:00	01.03.2020 17:00	Emre Girgin;Salih Kasim Benli	
13	Mobile Scenario & Mockup for Guest	4 günler	27.02.2020 08:00	01.03.2020 17:00	Katarina Korolainen;Mehmet Erdinç Oguz	
14	Preparing for Customer meeting	6 günler	27.02.2020 08:00	03.03.2020 17:00	Berke Özdemir;Koray Cetin	
15	Brainstorming project name & logo	12 günler	27.02.2020 08:00	09.03.2020 17:00	Everyone	
16	<input type="checkbox"/> W4: Finalizing Requirements	7 günler	03.03.2020 08:00	09.03.2020 17:00		
17	Reviewing scenarios & mock-ups	5 günler	03.03.2020 08:00	07.03.2020 17:00	11;12... Burak Cuhadar;Eylül Yalcinkaya	
18	Customer meeting notes & feedback	2 günler	04.03.2020 08:00	05.03.2020 17:00	14 Koray Cetin	
19	Fixing scenarios & mock-ups	2 günler	08.03.2020 08:00	09.03.2020 17:00	17 Berkay Alkan;Emre Girgin;Katarina Korolainen;Mehmet Erdinç Oguz;Ölcayto Türker;Salih Kasim Benli	
20	Finalizing Requirements	3 günler	06.03.2020 08:00	08.03.2020 17:00	9;18 Everyone	
21	<input type="checkbox"/> W5-6: UML Diagrams	15 günler	10.03.2020 08:00	24.03.2020 17:00		
22	Deciding project name & logo	7 günler	10.03.2020 08:00	16.03.2020 17:00	15 Everyone	
23	Glass diagram	12 günler	10.03.2020 08:00	21.03.2020 17:00	20 Çağrı Çiftçi;Emre Girgin;Eylül Yalcinkaya;Salih Kasim Benli	
24	Use case diagram	12 günler	10.03.2020 08:00	21.03.2020 17:00	20 Berkay Alkan;Berke Özdemir;Burak Cuhadar;Katarina Korolainen;Koray Cetin;Mehmet Erdinç Oguz;Ölcayto Türker;Salih Kasim Benli	
25	Sequence diagrams	3 günler	22.03.2020 08:00	24.03.2020 17:00	23;24 Everyone	
27	Fixing UML diagrams based on feedback	14 günler	31.03.2020 07:00	13.04.2020 17:00	23;24... Everyone	
28	<input type="checkbox"/> W10-11: Project Plan	15 günler	14.04.2020 08:00	28.04.2020 17:00		
29	Extracting past work to Project plan	6 günler	14.04.2020 08:00	19.04.2020 17:00	Berkay Alkan;Burak Cuhadar;Eylül Yalcinkaya;Salih Kasim Benli	
30	Milestones	5 günler	14.04.2020 08:00	18.04.2020 17:00	Emre Girgin	
31	Listing future work to Project plan	5 günler	14.04.2020 08:00	18.04.2020 17:00	Mehmet Erdinç Oguz	
32	Project plan (Gantt diagram)	1 gün	20.04.2020 08:00	20.04.2020 17:00	29;30... Katarina Korolainen	
33	Updating & fixing Project Plan	1 gün	27.04.2020 08:00	27.04.2020 17:00	32 Katarina Korolainen	
34	RAM	15 günler	14.04.2020 08:00	28.04.2020 17:00		
35	<input type="checkbox"/> W12	7 günler	28.04.2020 08:00	04.05.2020 17:00		
36	Executive Summary	4 günler	28.04.2020 08:00	01.05.2020 17:00		
37	List & status of deliverables	4 günler	28.04.2020 08:00	01.05.2020 17:00		
38	Evaluating the status of the deliverables	4 günler	28.04.2020 08:00	01.05.2020 17:00		
39	Evaluating tools & processes	4 günler	28.04.2020 08:00	01.05.2020 17:00		
40	Summary of work done by each member	4 günler	30.04.2020 08:00	03.05.2020 17:00	34	
41	Communication plan	4 günler	30.04.2020 08:00	03.05.2020 17:00	10	
42	Requirements	4 günler	30.04.2020 08:00	03.05.2020 17:00	20	
43	Scenarios & Mockups	4 günler	30.04.2020 08:00	03.05.2020 17:00	19	
44	UML Diagrams	4 günler	30.04.2020 08:00	03.05.2020 17:00	27	
45	Project Plan	4 günler	30.04.2020 08:00	03.05.2020 17:00	33	
46	RAM	4 günler	30.04.2020 08:00	03.05.2020 17:00		
47	Milestone 1	0 günler	04.05.2020 17:00	04.05.2020 17:00	36;37...	
48	<input type="checkbox"/> W13: API Assignment	14 günler	02.05.2020 08:00	15.05.2020 17:00		
49	Finding APIs	3 günler	02.05.2020 08:00	04.05.2020 17:00	Everyone	
50	Creating RESTful API	9 günler	05.05.2020 08:00	13.05.2020 17:00	49	Everyone
51	Unit testing	9 günler	05.05.2020 08:00	13.05.2020 17:00	Everyone	
52	Pull requests & reviews	11 günler	05.05.2020 08:00	15.05.2020 17:00	Everyone	
53	<input type="checkbox"/> W14	6 günler?	12.05.2020 07:00	18.05.2020 08:00		
54	Executive Summary	4 günler	12.05.2020 07:00	15.05.2020 17:00		
55	List & status of deliverables	4 günler	12.05.2020 07:00	15.05.2020 17:00		
56	Evaluating the status of the deliverables	4 günler	12.05.2020 07:00	15.05.2020 17:00		
57	Evaluating tools & processes	4 günler	12.05.2020 08:00	15.05.2020 17:00		
58	Summary of work done by each member	4 günler	14.05.2020 07:00	17.05.2020 17:00		
59	API Documentation	4 günler	14.05.2020 08:00	17.05.2020 17:00		
60	Milestone 2	0 günler?	18.05.2020 07:00	18.05.2020 08:00	54;55...	
61	BREAK	100 günler?	18.05.2020 09:00	05.10.2020 09:00		
62	<input type="checkbox"/> SECOND SEMESTER	65 günler?	28.10.2020 08:00	26.01.2021 17:00		
63	Update the requirements	3 günler?	28.10.2020 08:00	30.10.2020 17:00		
64	Extract features from the requirements	5 günler?	30.10.2020 17:00	06.11.2020 17:00		
65	Discuss possible new requirements	8 günler?	28.10.2020 08:00	06.11.2020 17:00		
66	Team distribution	8 günler?	28.10.2020 08:00	06.11.2020 17:00		
67	Create the design of the product	8 günler?	28.10.2020 08:00	06.11.2020 17:00		

68		<input type="checkbox"/> Backend	10 günler?	03.11.2020 08:00	16.11.2020 17:00
69		Research: Docker	4 günler?	03.11.2020 08:00	06.11.2020 17:00
70		Research: AWS	4 günler?	03.11.2020 08:00	06.11.2020 17:00
71		Get some mock data	4 günler?	03.11.2020 08:00	06.11.2020 17:00
72		Deployment	3 günler?	12.11.2020 08:00	16.11.2020 17:00
73		Integrate Jenkins	3 günler?	12.11.2020 08:00	16.11.2020 17:00
74		Integrate Docker	3 günler?	12.11.2020 08:00	16.11.2020 17:00
75		Update Class Diagrams	3 günler?	12.11.2020 08:00	16.11.2020 17:00
76		Update Sequence Diagrams	3 günler?	12.11.2020 08:00	16.11.2020 17:00
77		Initialize DB	3 günler?	12.11.2020 08:00	16.11.2020 17:00
78		Add some endpoints for testing	3 günler?	12.11.2020 08:00	16.11.2020 17:00
79		Connect endpoints with database	3 günler?	12.11.2020 08:00	16.11.2020 17:00
80		Research: Kubernetes	3 günler?	12.11.2020 08:00	16.11.2020 17:00
81		; <input type="checkbox"/> Backend	6 günler?	17.11.2020 08:00	24.11.2020 17:00
82		Homepage Recommendation Endpoint	5 günler?	17.11.2020 08:00	23.11.2020 17:00
83		Login/SignUp as Vendor	4 günler?	17.11.2020 08:00	20.11.2020 17:00
84		Login/SignUp as User	2 günler?	17.11.2020 08:00	18.11.2020 17:00
85		Forgot My Password	1 gün?	17.11.2020 08:00	17.11.2020 17:00
86		Category/Product Endpoint	6 günler?	17.11.2020 08:00	24.11.2020 17:00
87		Wishlist for User	2 günler?	17.11.2020 08:00	18.11.2020 17:00
88		Search	6 günler?	17.11.2020 08:00	24.11.2020 17:00
89		Vendor Data	6 günler?	17.11.2020 08:00	24.11.2020 17:00
90		Shopping Cart	2 günler?	17.11.2020 08:00	18.11.2020 17:00
91		Frontend	17 günler?	03.11.2020 08:00	25.11.2020 17:00
92		React and Redux research	7 günler?	03.11.2020 08:00	11.11.2020 17:00
93		Project Setup	1 gün?	10.11.2020 08:00	10.11.2020 17:00
94		Integrate Redux	1 gün?	11.11.2020 08:00	11.11.2020 17:00
95		Homepage UI	2 günler?	12.11.2020 08:00	13.11.2020 17:00
96		Navbar UI	3 günler?	14.11.2020 08:00	18.11.2020 17:00
97		Login/Signup Page UI	7 günler?	10.11.2020 08:00	18.11.2020 17:00
98		Category Products Page UI	7 günler?	10.11.2020 08:00	18.11.2020 17:00
99		Product Details Page UI	7 günler?	10.11.2020 08:00	18.11.2020 17:00
100		Deployment	3 günler?	17.11.2020 17:00	20.11.2020 17:00
101		Setup CI/CD	2 günler?	20.11.2020 08:00	23.11.2020 17:00
102		Best Sellers, On Sale, New Releases, etc. UI	3 günler?	22.11.2020 08:00	25.11.2020 17:00
103		Wishlist	6 günler?	17.11.2020 17:00	25.11.2020 17:00
104		Android	15 günler?	03.11.2020 08:00	23.11.2020 17:00
105		Android core topics research	7 günler?	03.11.2020 08:00	11.11.2020 17:00
106		Project Setup	3 günler?	10.11.2020 08:00	12.11.2020 17:00
107		Homepage UI-Logic	10 günler?	10.11.2020 08:00	23.11.2020 17:00
108		Categories Page UI-Logic	10 günler?	10.11.2020 08:00	23.11.2020 17:00
109		Wishlist Page UI-Logic	10 günler?	10.11.2020 08:00	23.11.2020 17:00
110		Product List Page UI-Logic	10 günler?	10.11.2020 08:00	23.11.2020 17:00
111		Log-in/Sign-up UI-Logic	10 günler?	10.11.2020 08:00	23.11.2020 17:00
112		Final review and test	1 gün?	23.11.2020 08:00	23.11.2020 17:00
113		Milestone 1	0 günler?	24.11.2020 08:00	24.11.2020 08:00
114		Backend	17 günler?	24.11.2020 08:00	16.12.2020 17:00
115		Vendor Orders	5 günler?	24.11.2020 08:00	30.11.2020 17:00
116		Most Purchased, Most Liked, Discounted Product	4 günler?	30.11.2020 17:00	04.12.2020 17:00
117		Filtering Mechanism	4 günler?	04.12.2020 17:00	10.12.2020 17:00
118		Update Shopping Cart	4 günler?	10.12.2020 17:00	16.12.2020 17:00

119			<input type="checkbox"/> <b>Frontend</b>	17 günler?	<b>24.11.2020 08:00</b>	<b>16.12.2020 17:00</b>
120			Connect hardcoded functionality to backend	7 günler	24.11.2020 08:00	02.12.2020 17:00
121			Set up Persist/Rehydrate for React	4 günler?	24.11.2020 11:00	30.11.2020 11:00
122			Product Details Page UI	3 günler	01.12.2020 08:00	03.12.2020 17:00
123			Dockerization and Deployment	3 günler	24.11.2020 14:00	27.11.2020 14:00
124			Cart Page UI	3 günler	30.11.2020 08:00	02.12.2020 17:00
125			Checkout Page UI	3 günler	24.11.2020 12:00	27.11.2020 13:00
126			Sign in, Sign up as vendor	4 günler	30.11.2020 09:00	04.12.2020 09:00
127			Orders Page	3 günler?	07.12.2020 08:00	09.12.2020 17:00
128			Profile Page	2 günler?	07.12.2020 09:00	09.12.2020 09:00
129			CI/CD	5 günler?	07.12.2020 09:00	14.12.2020 09:00
130			Google Maps for vendor signup	1 gün?	15.12.2020 08:00	15.12.2020 17:00
131			Comment	2 günler?	14.12.2020 08:00	15.12.2020 17:00
132			lint Setup	1 gün?	16.12.2020 08:00	16.12.2020 17:00
133			<input type="checkbox"/> <b>Android</b>	17 günler?	<b>24.11.2020 08:00</b>	<b>16.12.2020 17:00</b>
134			Connect hardcoded functionality to backend	7 günler?	24.11.2020 08:00	02.12.2020 17:00
135			Shopping Cart Page UI-Logic	8 günler?	28.11.2020 08:00	09.12.2020 17:00
136			Profile Page UI-Logic	8 günler?	28.11.2020 08:00	09.12.2020 17:00
137			Vendor Page UI	5 günler?	28.11.2020 08:00	04.12.2020 17:00
138			Payment Page UI	5 günler?	28.11.2020 08:00	04.12.2020 17:00
139			Orders Page UI-Logic	8 günler?	28.11.2020 08:00	09.12.2020 17:00
140			Filter and Sort Logic	5 günler?	09.12.2020 08:00	15.12.2020 17:00
141			Final review and test	1 gün?	16.12.2020 08:00	16.12.2020 17:00
142			Milestone 2	0 günler	17.12.2020 08:00	17.12.2020 08:00
143			<input type="checkbox"/> <b>Backend</b>	8 günler	<b>17.12.2020 08:00</b>	<b>28.12.2020 17:00</b>
144			Payment	3 günler	17.12.2020 08:00	21.12.2020 17:00
145			Orders Information	4 günler	21.12.2020 17:00	25.12.2020 17:00
146			Profile Information for Users	2 günler	23.12.2020 17:00	25.12.2020 17:00
147			Update Profile for Users	1 gün	25.12.2020 17:00	28.12.2020 17:00
148			<input type="checkbox"/> <b>Frontend</b>	8 günler?	<b>17.12.2020 08:00</b>	<b>28.12.2020 17:00</b>
149			Admin Page Setup	2 günler?	17.12.2020 08:00	18.12.2020 17:00
150			Customer Addresses	2 günler?	18.12.2020 08:00	21.12.2020 17:00
151			Filter Sort UI	3 günler?	17.12.2020 08:00	21.12.2020 17:00
152			Project Plan Update	1 gün?	22.12.2020 08:00	22.12.2020 17:00
153			Fixing dummy data inconsistencies	3 günler?	22.12.2020 08:00	24.12.2020 17:00
154			Backend Connections on only UI implemented p	5 günler?	22.12.2020 08:00	28.12.2020 17:00
155			Vendor Add Products	3 günler?	17.12.2020 09:00	22.12.2020 09:00
156			Vendor See Products	3 günler?	22.12.2020 08:00	24.12.2020 17:00
157			Final bug fixes for second customer meeting	5 günler?	22.12.2020 08:00	28.12.2020 17:00
158			<input type="checkbox"/> <b>Android</b>	30 günler?	<b>17.11.2020 08:00</b>	<b>28.12.2020 17:00</b>
159			Old milestones review	3 günler?	17.11.2020 08:00	19.11.2020 17:00
160			Payment Logic	5 günler?	19.11.2020 08:00	25.11.2020 17:00
161			Vendor Logic	6 günler?	19.12.2020 08:00	28.12.2020 17:00
162			Complaint Logic	5 günler?	19.11.2020 08:00	25.11.2020 17:00
163			Final review and test	1 gün?	28.12.2020 08:00	28.12.2020 17:00
164			Customer Scenarios Preparation	1 gün?	25.12.2020 08:00	25.12.2020 17:00
165			Customer Meeting Presentation Prep	1 gün?	27.12.2020 08:00	28.12.2020 17:00
166			Legal Documents Preparation	10 günler?	21.12.2020 08:00	01.01.2021 17:00
167			Milestone 3	0 günler	29.12.2020 17:00	29.12.2020 17:00

168			Backend	19 günler	28.12.2020 17:00	22.01.2021 17:00	Koray Cetin;Olcayto Türker;Veli Can Unal;Mehmet Erdinç Oguz
169			Email Verification	7 günler	28.12.2020 17:00	06.01.2021 17:00	Olcayto Türker
170			Report Comment	4 günler	06.01.2021 17:00	12.01.2021 17:00	Olcayto Türker
171			Report Product	4 günler	12.01.2021 17:00	18.01.2021 17:00	Olcayto Türker
172			Forgot Password	4 günler	18.01.2021 17:00	22.01.2021 17:00	Olcayto Türker
173			Admin Login	3 günler	28.12.2020 17:00	31.12.2020 17:00	Veli Can Unal
174			Admin Actions	10 günler	31.12.2020 17:00	14.01.2021 17:00	Veli Can Unal
175			Listing Messages	2 günler	28.12.2020 17:00	30.12.2020 17:00	Koray Cetin
176			Socket Messaging	7 günler	30.12.2020 17:00	08.01.2021 17:00	Koray Cetin
177			Vendor Update Products	4 günler	14.01.2021 17:00	20.01.2021 17:00	Veli Can Unal
178			Vendor Get Products	2 günler	19.01.2021 17:00	21.01.2021 17:00	Veli Can Unal
179			Vendor Get Orders	3 günler	28.12.2020 17:00	31.12.2020 17:00	Mehmet Erdinç Oguz
180			Update Order Status	7 günler	31.12.2020 17:00	11.01.2021 17:00	Mehmet Erdinç Oguz
181			Semantic Search	4 günler	11.01.2021 17:00	15.01.2021 17:00	Mehmet Erdinç Oguz
182			Google Signin	5 günler	15.01.2021 17:00	22.01.2021 17:00	Mehmet Erdinç Oguz
183			Notification Infrastructure	10 günler	08.01.2021 17:00	22.01.2021 17:00	Koray Cetin

184			Frontend	17 günler?	02.01.2021 08:00	26.01.2021 17:00	
185			Admin Reported Comments Page	2 günler	02.01.2021 08:00	05.01.2021 17:00	
186			Admin Reported Products Page	2 günler	06.01.2021 08:00	07.01.2021 17:00	
187			Report Comment	1 gün?	04.01.2021 09:00	05.01.2021 09:00	
188			Report Product	1 gün?	06.01.2021 08:00	06.01.2021 17:00	
189			Sign in with google	2 günler?	11.01.2021 09:00	13.01.2021 09:00	
190			Vendor Profile Information	2 günler	11.01.2021 09:00	13.01.2021 09:00	
191			Vendor Orders Page	2 günler?	11.01.2021 08:00	12.01.2021 17:00	
192			Vendor Notifications	3 günler?	14.01.2021 08:00	18.01.2021 17:00	
193			Customer Notification	3 günler?	14.01.2021 08:00	18.01.2021 17:00	
194			Vendor Messages	3 günler?	18.01.2021 09:00	21.01.2021 09:00	
195			Customer messages	3 günler?	18.01.2021 08:00	20.01.2021 17:00	
196			Admin panel backend connections	3 günler?	18.01.2021 08:00	20.01.2021 17:00	
197			Admin deployment	2 günler?	20.01.2021 08:00	21.01.2021 17:00	
198			Bug fixes	5 günler	20.01.2021 08:00	26.01.2021 17:00	
199			Android	13 günler?	06.01.2021 08:00	22.01.2021 17:00	
200			Vendor Logic	4 günler?	06.01.2021 08:00	11.01.2021 17:00	
201			Sign in with google	4 günler?	06.01.2021 08:00	11.01.2021 17:00	
202			Notification	10 günler?	11.01.2021 08:00	22.01.2021 17:00	
203			Messaging	10 günler?	11.01.2021 08:00	22.01.2021 17:00	
204			Report Mechanism	10 günler?	11.01.2021 08:00	22.01.2021 17:00	
205			Cancel, return order	10 günler?	11.01.2021 08:00	22.01.2021 17:00	
206			Final Milestone	0 günler	26.01.2021 08:00	26.01.2021 08:00	

## 10. User Scenarios

### Android Scenario

- Section 1

**Persona:** Alperen is a senior Computer Engineering student at Bogazici University. He realized he started to spend his entire life on his desk at home and decided to invest in a more comfortable desk. Then he saw an advertisement for Buyo and decided to try it out by downloading the application to his phone.

**Preconditions:** Alperen has already downloaded the app.

**Goals:**

- Alperen wants to sign-up to BUYO.
- Alperen wants to buy a desk.
- Alperen wants to check the comments of the products he thinks to buy.

**Actions:**

1. Alperen sign-ups to BUYO.
2. He checks his email to verify his account
3. He signs-in to BUYO.
4. He switches to the Homepage tab.
5. He clicks on a product.
6. He goes to the comments page of that product.
7. He saw a malicious comment.
8. He reports the malicious comment.
9. He adds the product to his wishlist.
10. He turns back to the Homepage.
11. He clicks on another product.
12. He adds this product to his cart.
13. He opens his cart page.
14. He adds a new address.
15. He provides the payment details.
16. He updates the profile information.
17. He pays for the order after approving the Distance Sale Conditions.
18. He checks his orders page.
19. He logs-out.

**Acceptance Criteria:**

**1.1.3.1** Customer and vendor users shall provide an e-mail, an username and a password to sign up.

**1.1.3.6** During registration, the email address should be verified with a confirmation mail.

**1.1.1.1.1** Customers shall be able to search for products.

**1.1.1.1.10** Customers shall be able to read user comments about products.

**1.1.1.1.4.1** Customers shall be able to save collections, to keep track of items they wish to buy and to purchase the same items multiple times.

**1.1.1.1.5.1** Customers shall be able to add products to their cart.

**1.1.1.1.11** Customers shall give their names and address information before ordering their product.

**1.2.8.4** Customers shall approve an KVKK agreement after writing the payment information.

**1.1.1.1.6.1** Customers shall be able to see all their purchased items in the Orders page.

- **Section 2**

**Persona:** Mehmet is a local desk manufacturer. He recently decided to take his business online so entire Turkey can reach his products. At the same time, Mehmet logs-in to the BUYO as a vendor to see if there is a new order for his products.

**Preconditions:** Mehmet already has a vendor account with two products.

**Goals:**

- Mehmet approves his orders.
- Mehmet sends messages to his customers.
- Mehmet changes the price of one of his products.

**Actions:**

1. Mehmet logs-in to BUYO
2. He checks his orders.
3. He sees a new order from Alperen.
4. He approves the order.
5. He sends a message to Alperen.
6. He goes to his products.
7. He goes to the edit page of one of his products.
8. He changes the price of the product.
9. He logs-out.

**Acceptance Criteria:**

**1.1.1.2.1** Vendors shall be able to keep track of their orders via the orders page.

**1.2.2.3** After making an order, customers shall be able to message the vendor(s) of the item(s) directly.

**1.1.1.2.6** Vendors shall be able to make a discount on a product for a certain time span.

- **Section 3**

**Persona:** Alperen from section 1

**Preconditions:** Alperen already has a customer account and ordered a product. Also, he added another product to his wishlist.

**Goals:**

- Alperen wants to check his orders.
- Alperen wants to check his messages.
- Alperen wants to check his notifications

**Actions:**

1. Alperen logs-into the platform.

2. He checks his notifications.
3. He saw the price of the product he liked decreased.
4. He goes to his wishlist and removes it.
5. He goes to his messages.
6. He saw the message from vendor Mehmet.
7. He writes back a message.
8. He goes to the Homepage.
9. He clicks on the “Image Search” button.
10. He picks a photo from his gallery.
11. He searches for the object in the photo.

#### **Acceptance Criteria:**

- 1.2.6.1** Customers shall be able to choose notified if the price of a wishlisted product changes.
- 1.2.9.1** Customers shall be able to create private collections that they can name, edit, delete, add products to, e.g. to keep track of the items' prices.
- 1.1.2.1** Customer users shall send messages to vendors who are sellers of the products they buy.

## **Frontend Scenario**

Our frontend scenario was divided into multiple sections, each featuring its own persona.

- **Section 1**

**Persona:** Orhan is the dodgy owner of a phone dealership on the verge of bankruptcy. He has shifted his business to BUYO as a last ditch effort. He sees BUYO as his only hope to revitalize his dying business.

**Preconditions:** Orhan has a computer with an internet connection & web browser. He has a vendor account on BUYO. He already has some products (phones), and has already made some sales. He has active (pending) orders as well.

#### **Goals:**

- Orhan wants to update his portfolio by deleting an old product and adding a new one.
- He wants to figure out if he's still almost bankrupt, by viewing his total earnings on Buyo so far.
- He wants to try an old marketing scheme: He wants to change the stock of a product, and claim that somebody stole some phones last night. (He's dishonest like that...)

#### **Actions:**

1. Orhan signs in as a vendor.
2. He deletes an old phone from his products.
3. He adds a new phone to his products.
4. He checks his total earnings from his orders page.
5. He updates the name and stock of a product.
6. He checks that the orders are also successfully updated, by checking the updated name.
7. He sees that there is an order containing a product whose product description was entered incorrectly (i.e. he is shipping a different product than described).
8. He fixes that mistake by canceling the order.
9. He notifies the customer via the message functionality, and explains why he's canceling the order.

### **Acceptance Criteria:**

- 1.1.1.2.9** Vendors should specify their stock information while adding a product to the platform.
- 1.1.1.2.1** Vendors shall be able to keep track of their orders via the orders page.
- 1.2.5.2.5** Vendors shall be able to see all orders and the total earnings.
- 1.2.5.2.1** Orders shall be exactly in one of those states, such as Pending, Being Prepared, Being Delivered, Completed, Cancelled, Refund.
- 1.2.5.2.2** For both active or completed cases, there shall be product information, order date, product price, cargo information, and delivery date.
- 1.1.1.2.3** Vendors shall be able to cancel an order which is in the order processing stage.
- 1.1.1.2.4** Vendors shall be able to specify a reason while cancelling an order.

- **Section 2**

**Persona:** Burak is a senior Computer Engineering student who hasn't slept for 2 nights trying to prepare for a presentation, and is feeling a bit grumpy because of that. His phone was broken during this week's finals, and he has ordered another one. He is anxiously waiting for his new phone's arrival.

**Preconditions:** Burak has a computer with an internet connection & web browser. He has a customer account on BUYO. He has made an order before (containing a phone product), and is waiting for the arrival of his order.

**Goals:** Burak just wants his finals to be over. More pertaining to our scenario, he wants to check the status of his phone order.

### **Actions:**

1. Burak signs in with Google.
2. He checks his notifications, and sees that his order has been canceled.
3. He looks at his messages to see why the order has been canceled.
4. He messages the vendor back in an accusatory tone. (He's understandably mad at this point.)

5. He reports the product he had ordered due to the misleading description.

#### **Acceptance Criteria:**

**1.1.3.5** Google account should be used for signing up/in.

**1.1.1.1.7** Customers shall be able to receive notifications.

**1.2.2.3** After making an order, customers shall be able to message the vendor(s) of the item(s) directly.

- **Section 3**

**Persona:** Meriç is a veteran web developer. After helping the initial development of BUYO, he decided to stick around as an admin. He has a reputation for being too trigger-happy with the bans.

**Preconditions:** Meriç has a computer with an internet connection & web browser. He has an admin account on BUYO. He has at least one reported comment (from the Android scenario) and one reported product (from the previous section) on his admin panel.

**Goals:** Meriç wants to sign in, and deal with any reported comments or products.

#### **Actions:**

1. Meriç signs in as admin.
2. He checks reported comments, and sees the comment reported by Alperen.
3. He removes the reported comment.
4. He checks reported products, and sees the product reported by Burak.
5. He realizes that the customer is right, and bans the owner of the reported product (Orhan).

#### **Acceptance Criteria:**

**1.1.1.4.2** Admins shall be able to remove malicious comments.

**1.1.1.4.1** Admins shall be able to ban customers or vendors.

- **Section 4**

**Persona:** Orhan is the same dodgy vendor from the first section.

**Preconditions:** Orhan still has a computer with an internet connection & web browser, and he still has a vendor account on BUYO. But this time around, his account has been banned by an admin.

**Goals:** Orhan wants to sign in and do his routine business on BUYO.

#### **Actions:**

1. Orhan tries to sign in, but he can't (because he's banned).
2. He sees an alert that asks him to contact support.

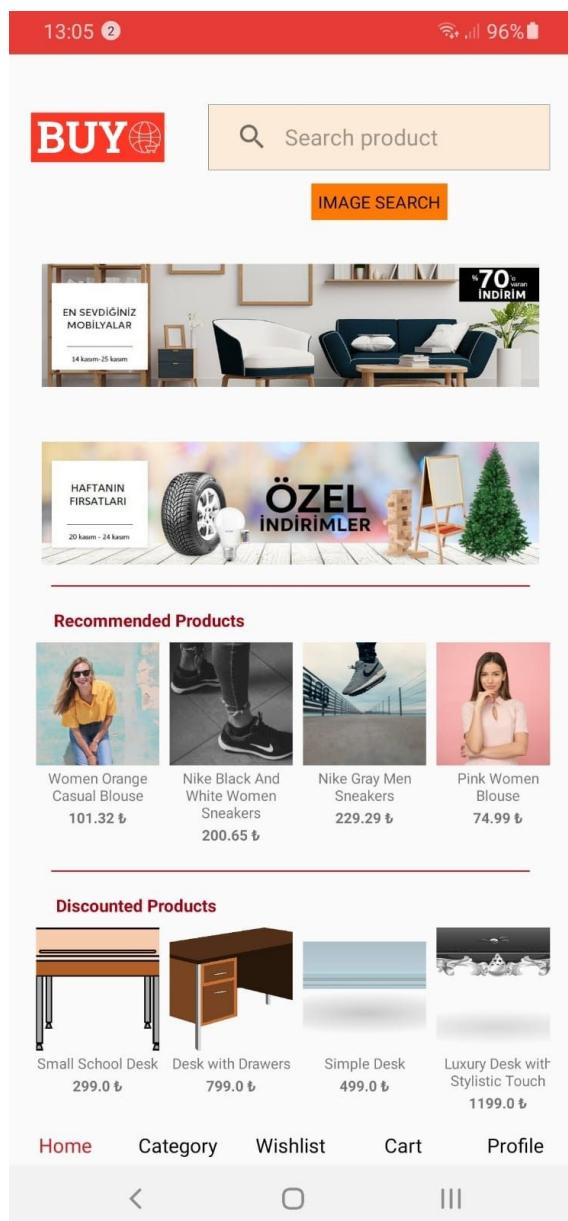
## Acceptance Criteria:

The only acceptance criteria here is that the banned user should not be able to sign in.

# 11. User Manual

## Android

### 1- Home Page



The home page is the first page the user will see when they open the application. In this page, **image search** and **semantic search** can be done via the search bar and the image search button at the top. The user can see current **campaigns** as banner images. In addition to these, they can see **recommended** and **discounted** products in the middle of the page. While the discounted products are the same for all users, recommended products are personalized when the user is logged-in. If not, they can be thought of best-sellers.

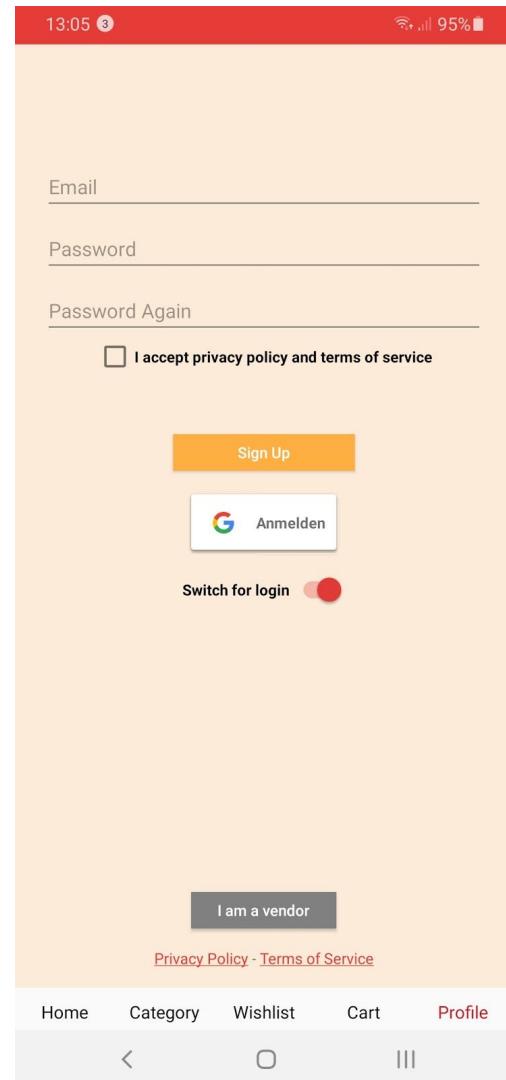
Another important component is the navigation tab that is at the bottom of the screen. The user can change the current tab via this navigator. Each tab is specialized for some group of features which will be explained throughout the manual.

## 2- Customer Sign-Up Page

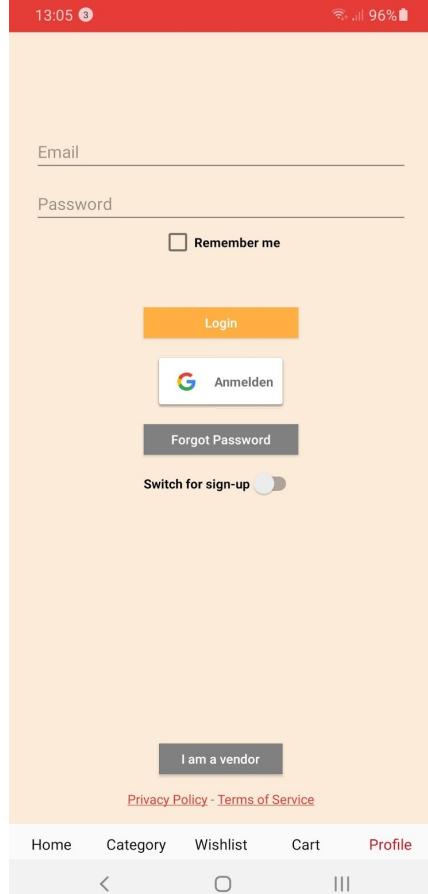
Customers can **sign-up** on this page by entering necessary information. Alternatively, they can benefit from the **Google sign-in**. They should accept the privacy policy and terms of service before signing-in.

The users can go to the vendor sign-up page by clicking the “**I am vendor**” button, and to the customer login page by clicking the “**Switch for login**” switch.

Finally, the users can read the privacy policy and the terms of service by clicking the links at the bottom.



A screenshot of a mobile application's sign-up page. At the top, there are fields for 'Email', 'Password', and 'Password Again'. Below these is a checkbox for accepting the 'privacy policy and terms of service'. A large orange 'Sign Up' button is centered. To the right, there is a Google sign-in button labeled 'Anmelden'. A toggle switch labeled 'Switch for login' is turned off. At the bottom, there is a 'I am a vendor' button and links for 'Privacy Policy - Terms of Service'.



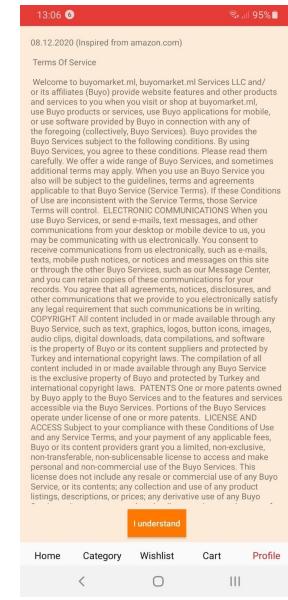
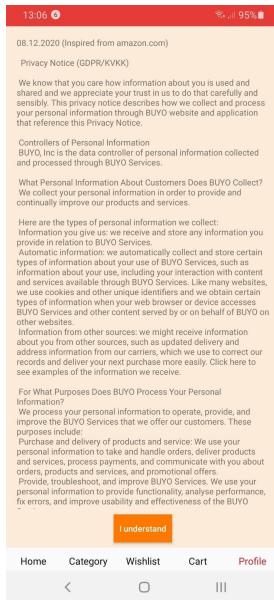
A screenshot of a mobile application's sign-up page. It features fields for 'Email' and 'Password', with a 'Remember me' checkbox below them. An orange 'Login' button is positioned above a Google sign-in button labeled 'Anmelden'. A 'Forgot Password' link is also present. A toggle switch labeled 'Switch for sign-up' is turned off. At the bottom, there is a 'I am a vendor' button and links for 'Privacy Policy - Terms of Service'.

## 3- Customer Login Page

The customers can login by giving their emails and passwords. If they don't want to login always, they can select the **remember me**.

Google sign-in is also possible on this page. In addition to this, users can reset their passwords by clicking the “**Forgot Password**” button. For other functionalities, you can look at the “Customer Sign-Up Page” section.

## 4- Legal Documents

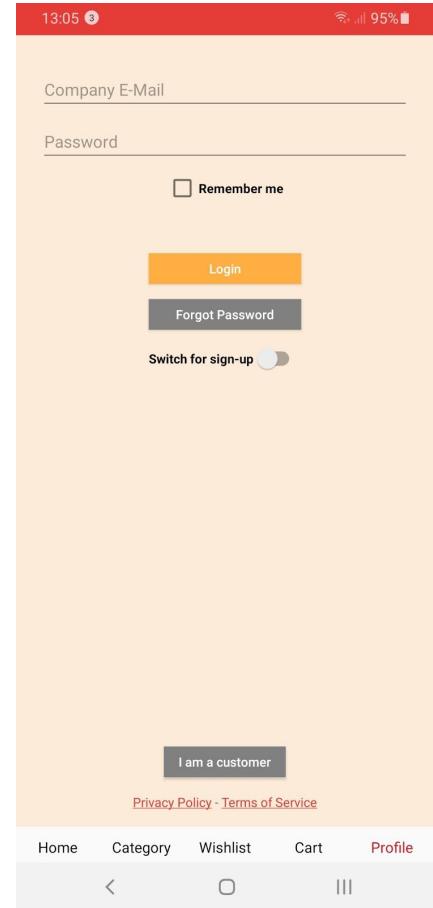
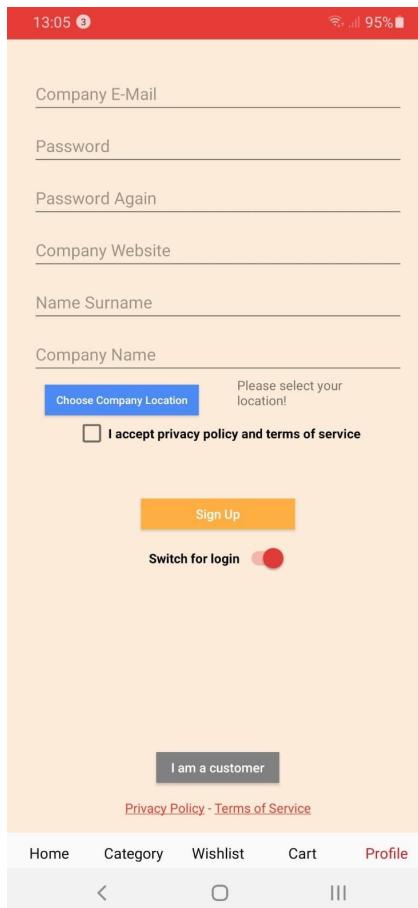


Privacy Notice(GDPR/KVKK)

Terms of Service

The user can read the legal documents they need to accept during the sign-up process. They can go back by clicking the “I understand” button.

## 5- Vendor Login and Sign-Up Pages



On the sign-up page, vendors should give their credentials. One important thing is that vendor's e-mail and company website must have the same domain name for the sake of validation. For example, if the company website is "www.mycompany.com", then the email should be "xyz@mycompany.com".

Vendors must also select the company location via **Google Maps**. They can select the location by clicking the "Select" button after specifying the correct location. The selected address will be shown next to the "**Choose Company Location**" button.

Vendors can login using the login page. The functionality is very similar to the customer login, except that the user is a vendor now.

## 6- Order Pages



The users, both vendor and customer, can see their orders and their status on this page. Each item is an ordered product and contains some information about the order such as the customer ID, the vendor name, order date, order number, details about the product and the status of the order. Users can click on the triangle button to see some details about the ordered product, or they can click on the product itself to go to the product detail page for more information about the product.

Each item has two buttons in it. The functionalities of these buttons change according to the status of the order. While vendors can approve or reject an order, they can message the customer, customers can cancel an order if it has not shipped yet, return order if 15 days have not passed yet, or message the vendor anytime.

## 7- Notification Page

On this page, the user can see the notification about some events which are discount and order cancellation for the customers, and for the vendors, just order cancellation.

If the notification is for discount, the user can go to the related product by clicking on the notification.

If the notification is for order cancellation, the user can go to the related order by clicking on the notification.



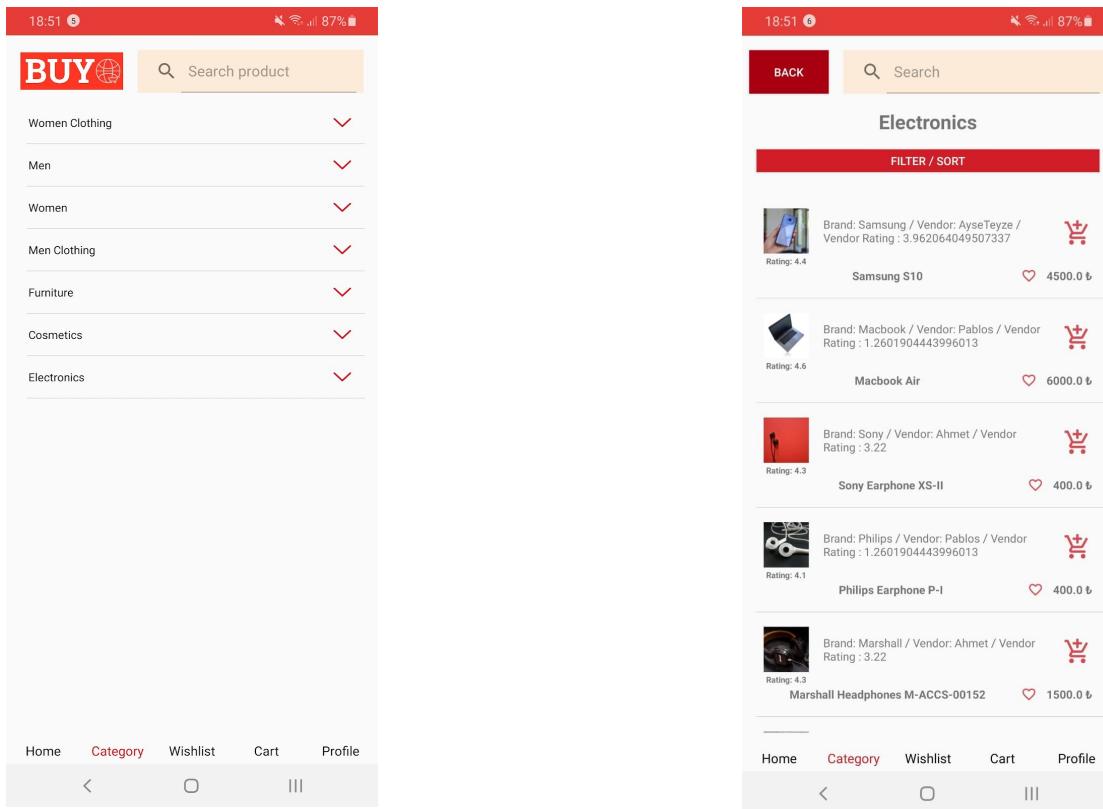
## 8- Search Product

All users (including the guest user) can search for a product. The search can be conducted in two ways, search by keyword and search by category.

In order to search by keyword, tap on the homepage tab on the navigation bar and click on the search bar on top of the page. Enter a keyword and related products will be listed on the page where the user is directed to.

In order to search by category, tap on the category tab on the navigation bar. Then select one of the listed categories or expand the list by clicking on the downward arrow of any category name. After clicking on one of the category names, the user is directed to the related product list page.

The customers can directly add the product to their cart or wishlist if they want to.



## 9- Filter&Sort

After listing the product by search (by either keyword or category) the user can filter or sort the products listed. By clicking on the "Filter & Sort" button, the user is directed to

the filtering page. The user can select an option for each criterion or left them empty. In this case, empty attributes mean to fetch all.

Moreover, the user can apply a sorting criterion to the filter s/he did.

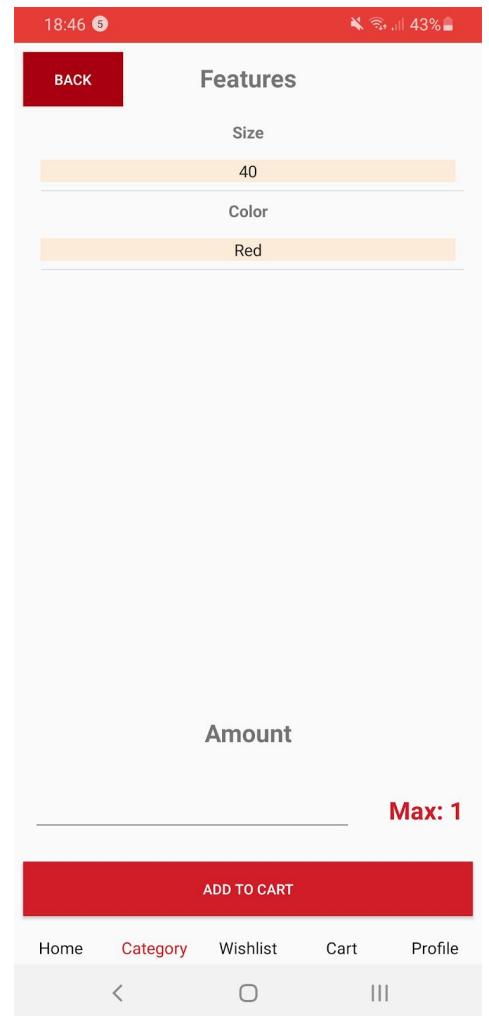
The screenshot shows a mobile application's filter and sort interface. At the top, there is a red header bar with the time '18:46' and battery level '43%'. Below the header, the word 'Filter' is displayed. Under 'Filter', there are two sections: 'Size' with the value '44' and 'Color' with the value 'Black'. To the right of the filter section is a 'Sort' section. The 'Sort' section includes a 'Rating' button, an 'Ascending' button, and a large red 'APPLY' button. Below these buttons are navigation links: 'Home', 'Category', 'Wishlist', 'Cart', and 'Profile'. At the bottom of the screen are three grey navigation icons: a back arrow, a square, and a three-line menu icon.

## 10- Product Details

The details of any product can be seen in a lot of ways. Some of the products are listed in the recommendation and discount section of the homepage or the users can search for a product. After clicking on the product, the user is directed to the product detail page of the product.

On this page, the image, name, vendor, discount if it has, brand, stock values for each combination, and the description are demonstrated. The customers can add the product to their cart or wishlist. Also, the comments made to this product by the customer who purchased it, are listed in the comments section. Moreover, the users can report the product itself or a comment made to it.

The screenshot shows a mobile application's product detail page for a Samsung S10. At the top, there is a red header bar with the time '18:52' and battery level '87%'. Below the header, there are three buttons: 'BACK', 'REPORT', and 'COMMENTS'. The main content area features a large image of a blue Samsung S10 smartphone held in a hand. Below the image, the text 'Samsung S10' and 'AyseTeyze' is displayed. There is a rating section with five stars and a heart icon. A 'DISCOUNT' message states: 'Buy this product for 4500.0 instead of 5000.0.' Below the discount, the text 'Brand: Samsung' and 'Available options: (stocks)' is shown, followed by two options: 'SCREENSIZE:6 INCH & RAM:8 GB & DISKSIZE:64 GB (95)' and 'SCREENSIZE:6 INCH & RAM:8 GB & DISKSIZE:128 GB (95)'. A small note at the bottom says 'Uncover the Ultimate Design. Beauty Beyond the Visual'. At the bottom of the page, the price '4500.0 ₺' is displayed next to a red 'ADD TO CART!' button. At the very bottom, there is a footer with navigation links: 'Home', 'Category', 'Wishlist', 'Cart', and 'Profile', along with three grey navigation icons: a back arrow, a square, and a three-line menu icon.



## 11- Add to Cart (Customer)

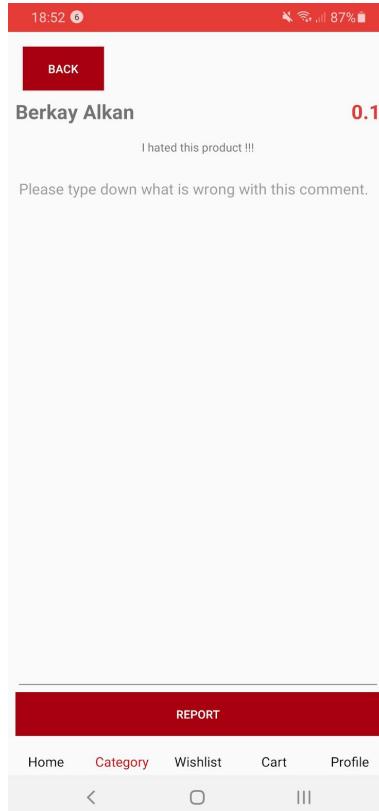
In order to add a product to the cart, go to the details of the product. Then click on the "Add to Cart" button. The user is directed to the details of his/her order. The user is wanted to select an option for each attribute of the product and the current stock amount is shown on this page also. After entering the amount s/he wants, then the order can be completed. The user can see the product in the cart tab in the navigation bar.

## 12- Report

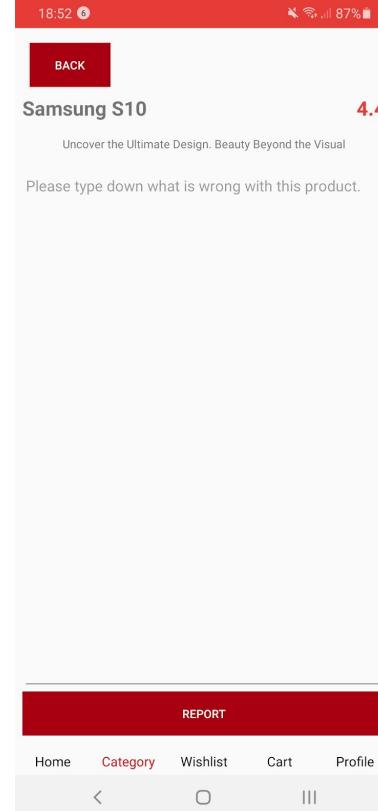
Any user can report a comment or a product. The only exception is that a vendor can not report his/her own products.

In order to report a product, go to details of the product. Then click on the report button seen above the screen. The user is going to be directed to the report page. On this page, the user can see the details of the product s/he wants to report. Provide reasoning below and click on the Report button. The report is sent.

In order to report a comment, go to the page of the product in which the comment the user wants to report is made. Then click on the comments button. Go to the product that is wanted to be reported and click on the report just above it. S/he is going to be directed to the report page. The user can see the comment there. Provide reasoning below and click on the Report button. The report is sent.



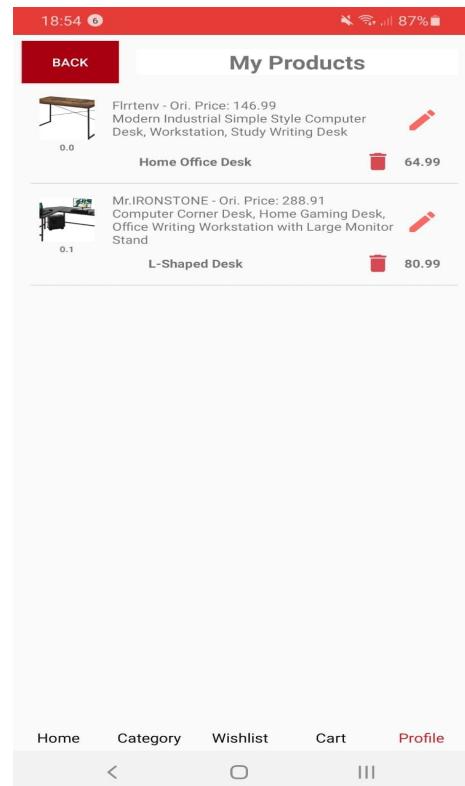
Report Comment



Report Product

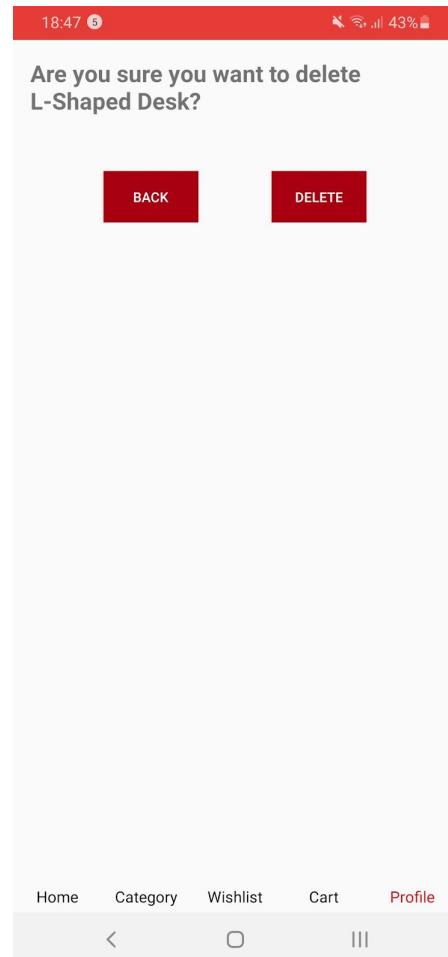
### 13- My Products (Vendor)

After logged-in as a vendor user, click on the Profile tab. Then click on my products page. The vendor can see the list of products s/he has. Click on the trash icon to delete and edit icon to edit the product.



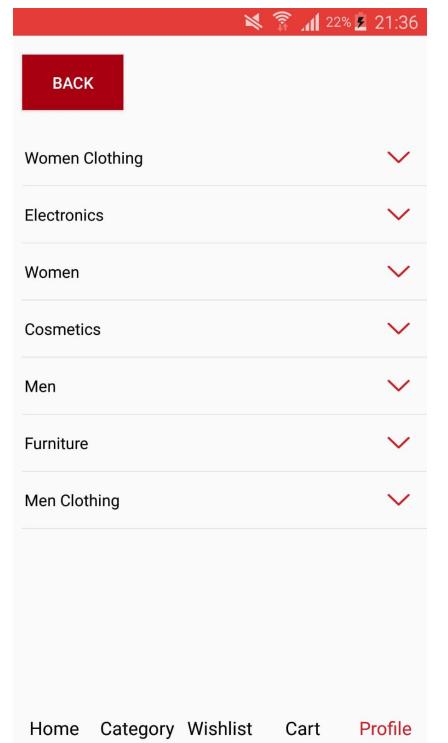
## 14- Delete Product (Vendor)

After logged-in as a vendor user, click on the Profile tab. Then click on my products page. The user can see the list of products s/he has. Then find the product that is wanted to be deleted. Click on the trash button next to its name. The user is going to be directed to the delete page. S/he can cancel the deletion process by clicking on the cancel button. In order to delete the product click on the delete button. The user will be directed to the page where s/he sees the remaining products. The product is deleted.



## 15- Add & Edit Product

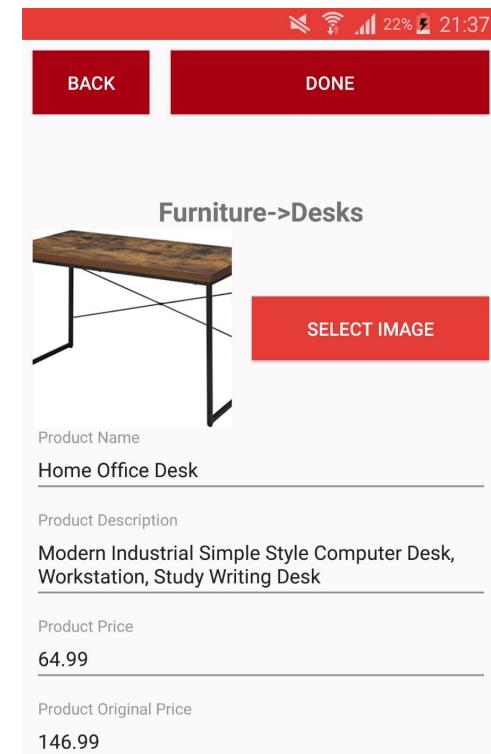
After signing-in as the vendor, by navigating to the profile tab and then the "add product" button, the vendor is going to be directed to a page where s/he can select the category of the new product s/he wants to add. This page is exactly the same as the one in the categories tab. The vendor can either directly select a category by clicking on the name of the category or navigates through to the sub-categories of one of them by clicking on the downward arrow next to it.



After selecting the category the vendor is directed to the page where the details of the product are entered. The vendor needs to provide the name, description, price, original price, and brand of the product. Also, for the features of the product, the vendor should enter the number of attributes that the product has. (ie. RAM, DiskSize, Screen Size are the attributes of a computer) After entering the number of the attributes, the vendor has to provide a name for each attribute and the possible options of each attribute. The possible options are needed to be separated by the dash (-). The image of the product is the logo of the BUYO by default.

After providing the details of the product, the vendor clicks on the DONE button and proceeds to the page where s/he enters the stock values for each attribute combination. After providing the stock values (zero is accepted as a stock value), the vendor clicks on the Done button, and the new product is added and can be seen on my products page.

Editing the product has a similar interface.



Product Description  
Modern Industrial Simple Style Computer Desk, Workstation, Study Writing Desk

Product Price  
64.99

Product Original Price  
146.99

Product Brand  
Flrrtenv

Product Number of Attributes  
2

Color  
Please split your options with '-'  
Black-Wood

Size  
Please split your options with '-'  
31Inch-39Inch

Black-31Inch      2

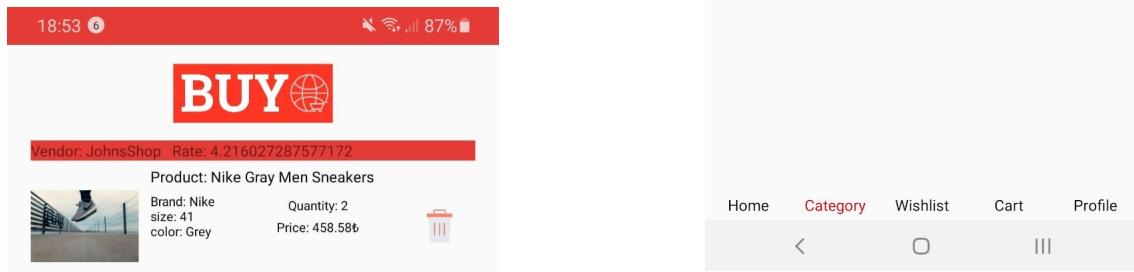
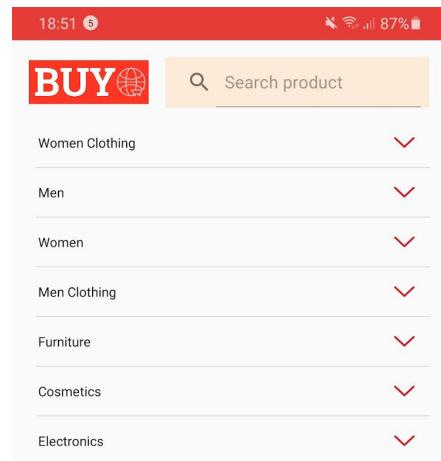
Black-39Inch      3

Wood-31Inch      5

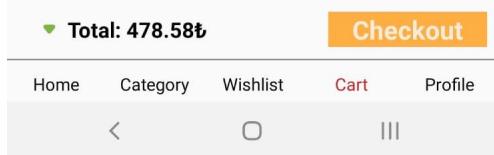
Wood-39Inch      7

## 16- Categories

You can see all main categories in the categories tab which is the second tab in the navigation bar. You can see the subcategories of a category by clicking the dropdown icon. You can also see the products in a category by clicking the category item. It directs you to product list page.



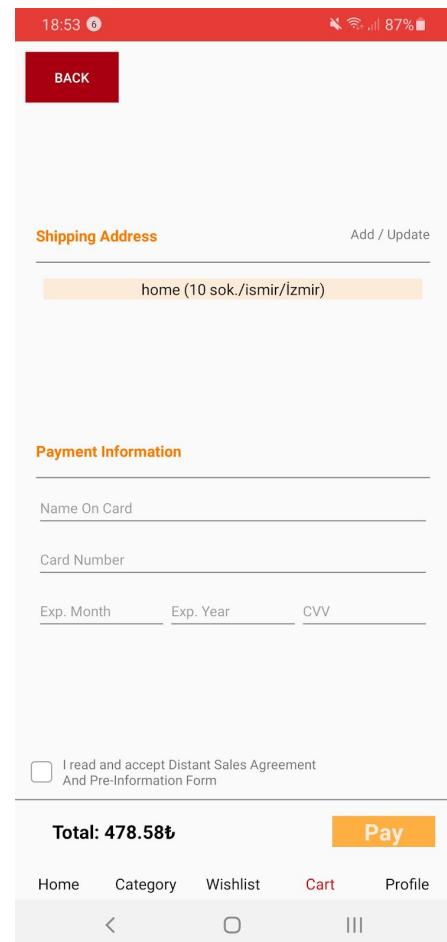
## 17- Cart



Cart page is reachable using the cart tab in the navigation bar. Customers can view the products they added to their cart. They can see the quantity and features they chose as well as the cart total. They can also remove the product from their cart using the trash icon. Clicking on the little green dropdown icon next to 'Total' price text will open the detailed version of cart price. Clicking the checkout button will proceed to payment(checkout page).

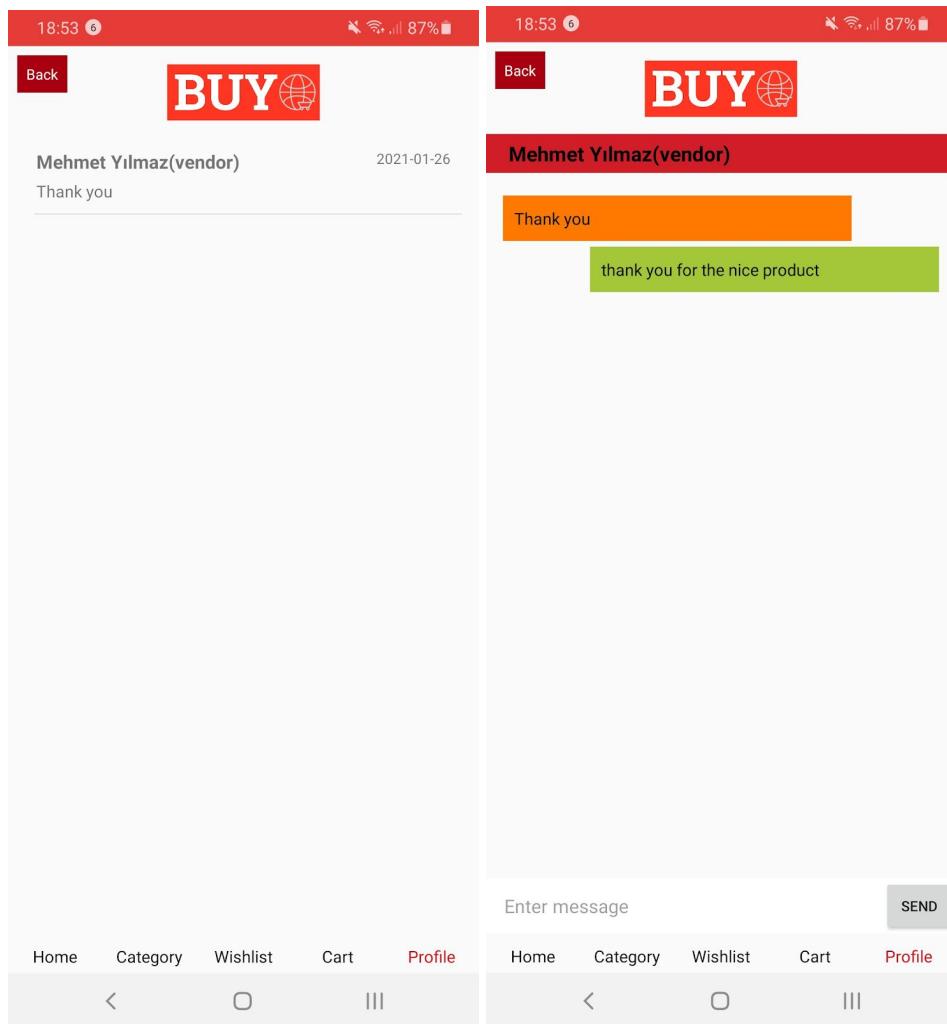
## 18- Checkout

In the checkout page, you need to fill three information to order your products in the cart. First one is your profile information. You should enter your profile information only once. To enter your profile information, you should go to the profile info page, enter your information then save them by clicking the save button. If you have done this already, just keep filling the information in the checkout page. Second one is the shipping address. If you have already entered one or more addresses, you can select one of them from dropdown by clicking to shortened address info. By clicking the ‘Add/Update’ button, you can add or update an address. Third one is the payment information. Our platform does not save your card information. You should enter your credit card information on every checkout. Last step, you should read ‘Distant Sales Agreement And Pre-Information Form’. You can see the form by clicking the form name which is at the end of the checkout page. You need to accept it by clicking the checkbox.



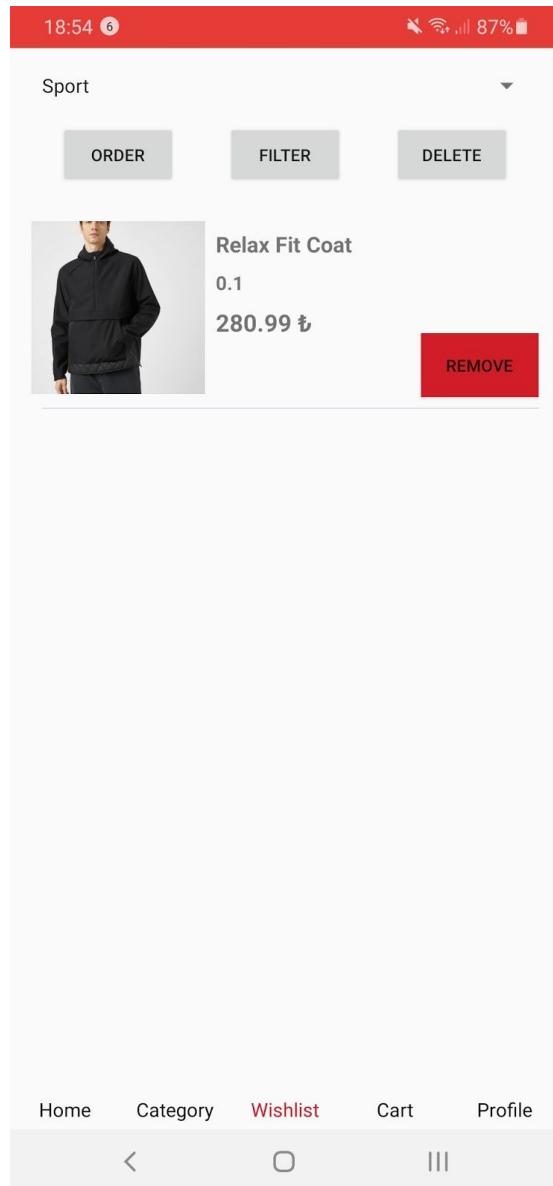
## 19- Messaging

In the profile tab, clicking my messages button will direct you to the messages page (Left image in the below) which shows a shortened version of your message history. In this page you can see the names of the users who they interacted with via messaging and the last message with that user. By clicking on a user, you will be directed to the live chat page(right image in the below). Your messaging history is displayed on this page. The users can then send a message by typing a message to the input field at the bottom and then pressing ‘SEND’ button found on the bottom right. The messaging is real time which means that the messages sent and received are displayed as soon as the messages are sent or received without refreshing the page.

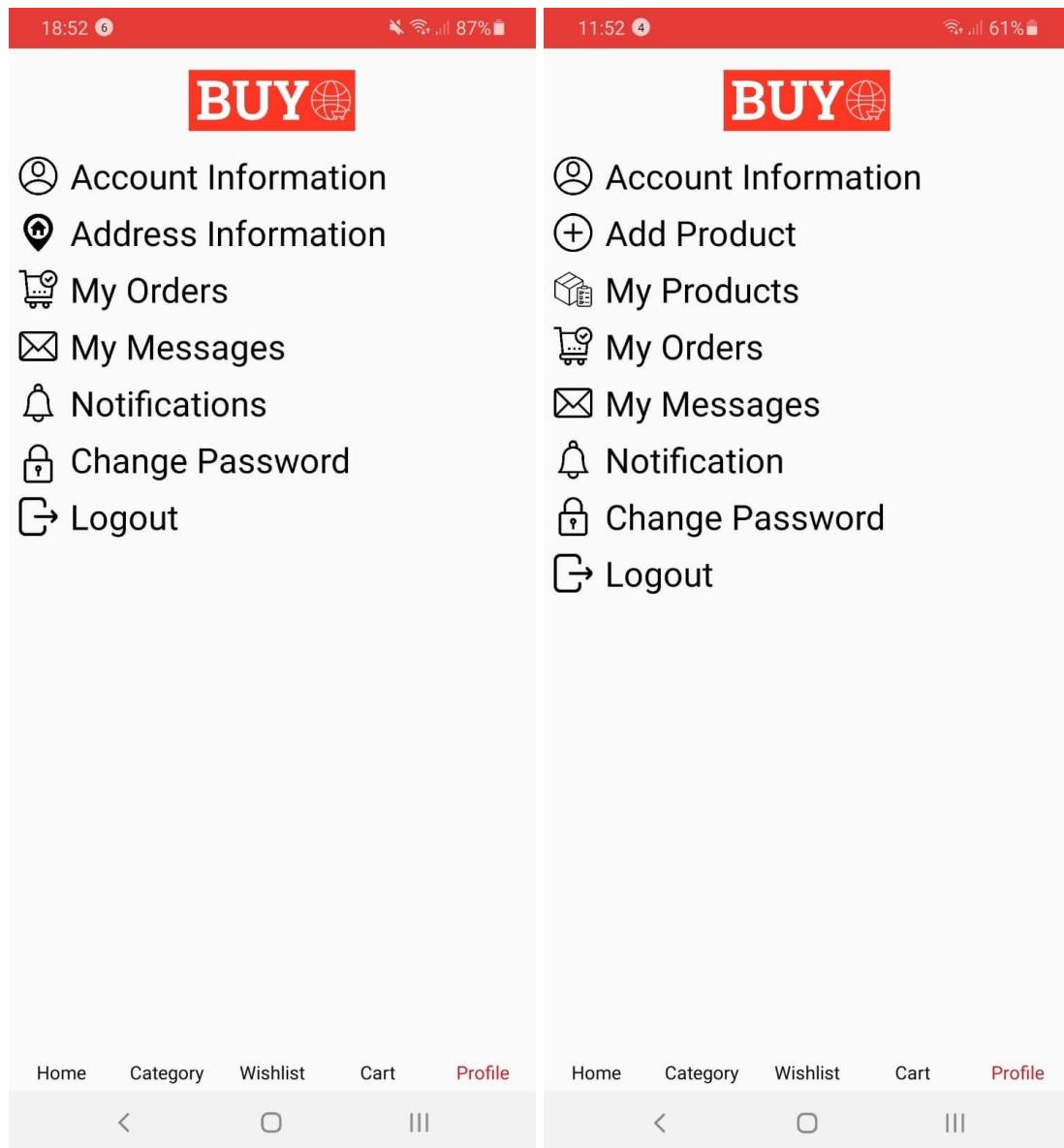


## 20-Wishlist Page

After customers add products to their wishlist, they can see the products that they liked from the wishlist tab. They can remove the product from their wishlist by clicking the remove button or click to product itself to see the product detail page.



## 21-Profile Page for Customer and Vendor

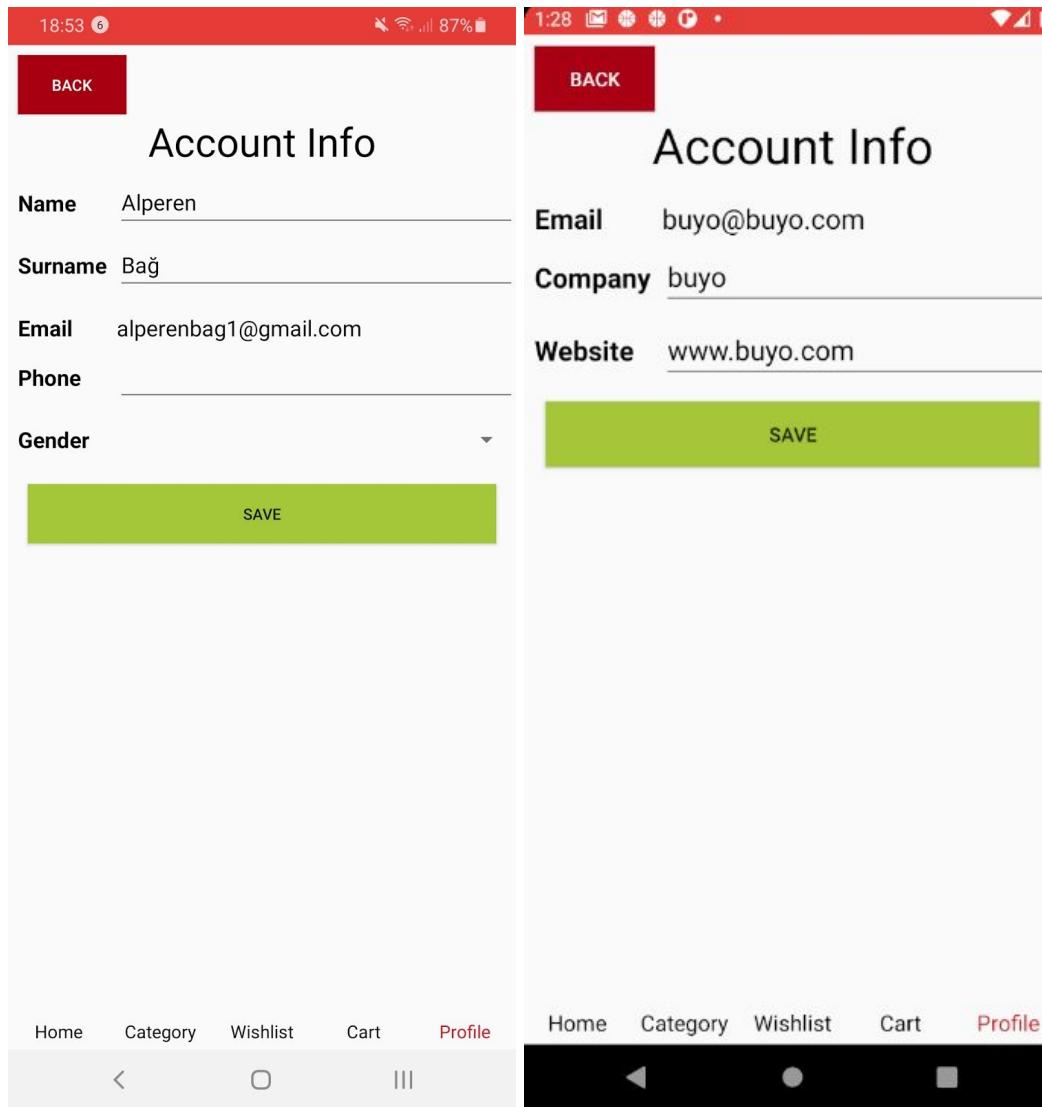


Profile pages for customer and vendor are shown respectively.

Customers can access their account information, address information, their orders, their messages, notifications and change password page. Also they can log out from the app.

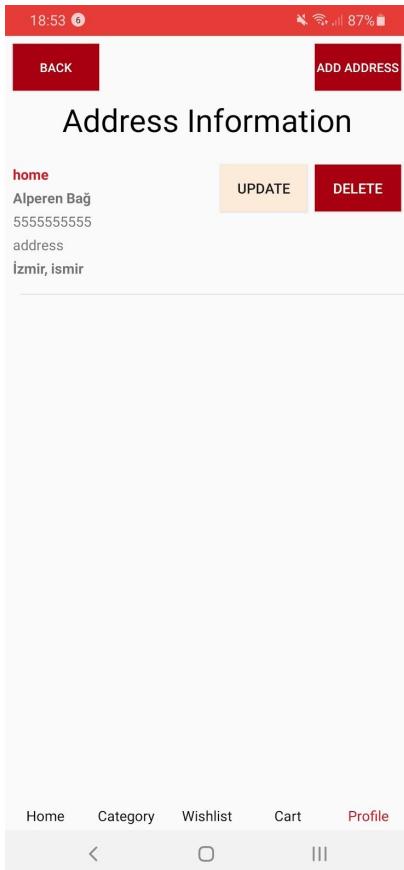
Vendors can access their account information, add product page, their products, given orders, their messages, notifications and change password page. Also they can log out from the app.

## 22-Account Information for Customer and Vendor



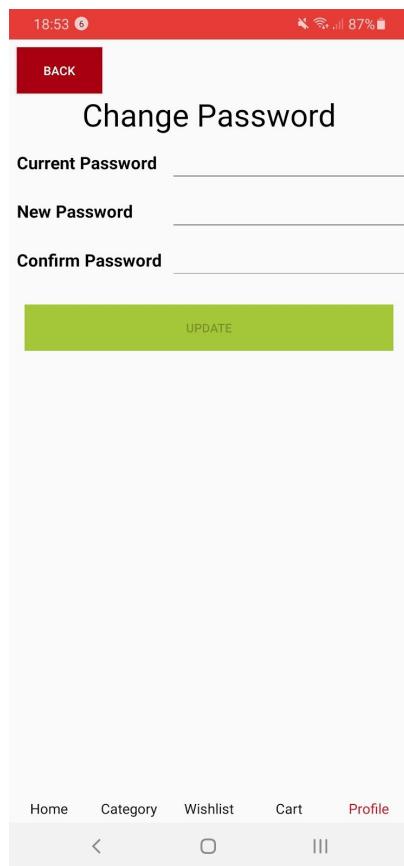
Account information pages for customer and vendor are shown respectively.  
Customers can view their email address and update their name, surname, phone and gender.

Vendors can see their email address and update their company name and website.



## 23-Address Information

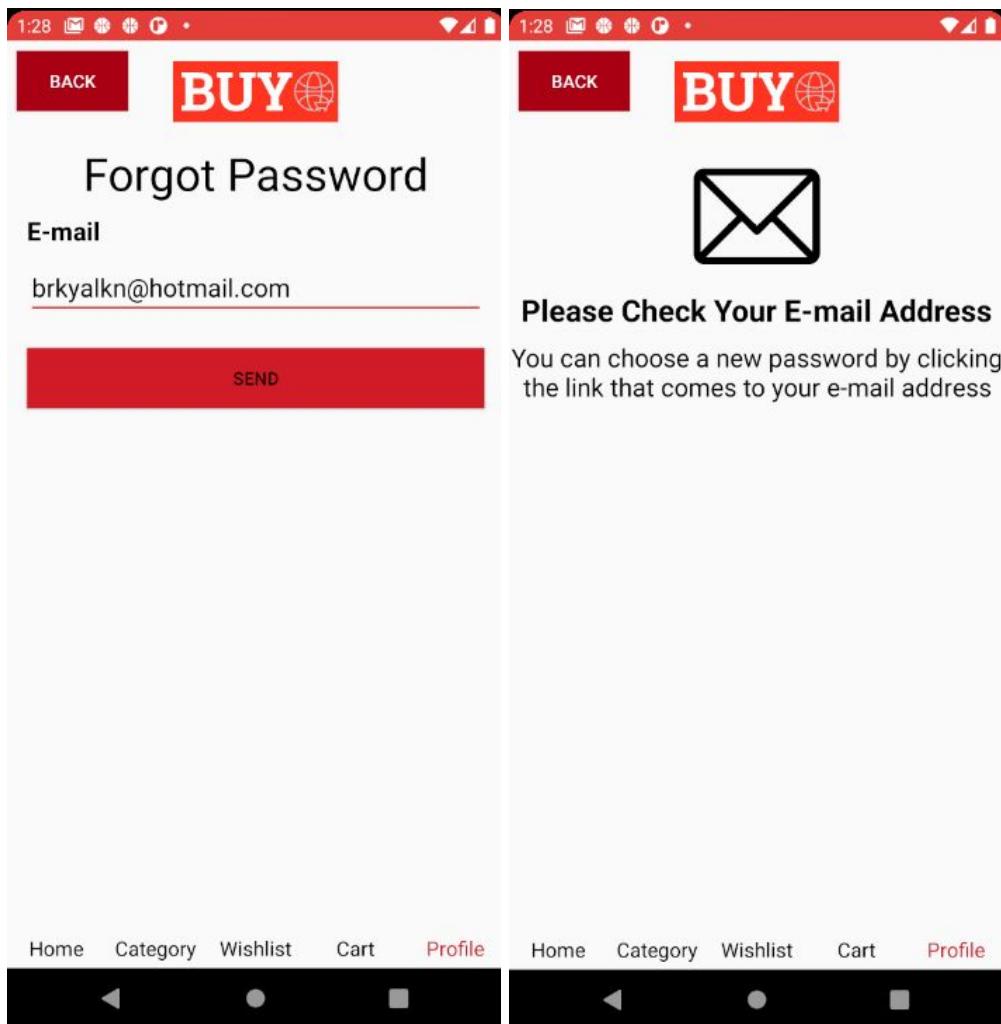
Customers can see their saved addresses. Address title, address name, phone number, address, city and province fields is shown. Customers can update or delete the address. Also they can add a new address by clicking add address button.



## 24-Change Password

Both customers and vendors can change their password using their current password.

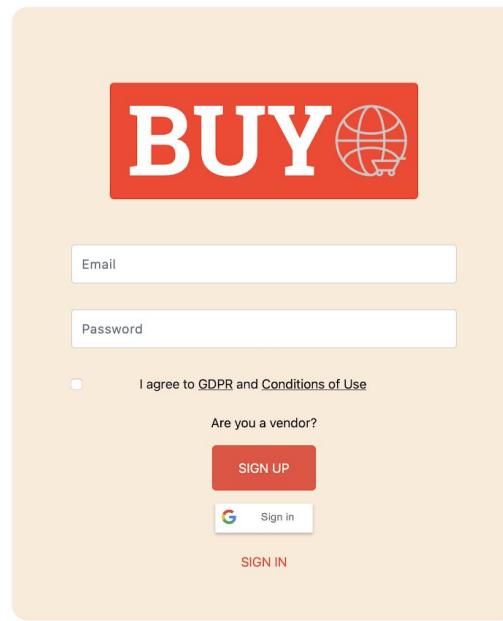
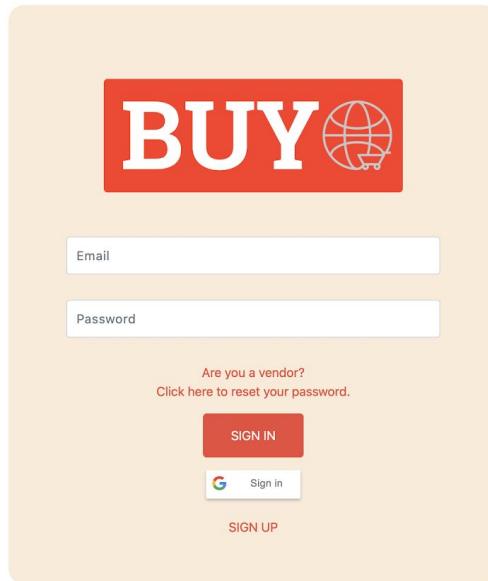
## 25-Forgot Password



Both customer and vendor can use forgot password functionality. After the user enters her email address and clicks the send button, an email is sent to her which includes the steps for resetting the password.

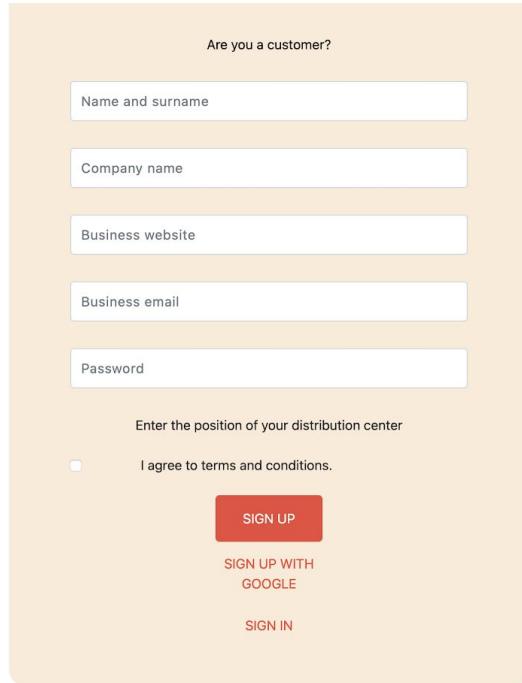
# Frontend

1- Sign in/up for customers



Customers can sign in/up using Google sign in/up or enter their email and password. A verification email will be sent to verify the email during sign up. Customers also have to accept GDPR and Conditions of Use to be able to sign up.  
Vendors use the same interface for signing in.

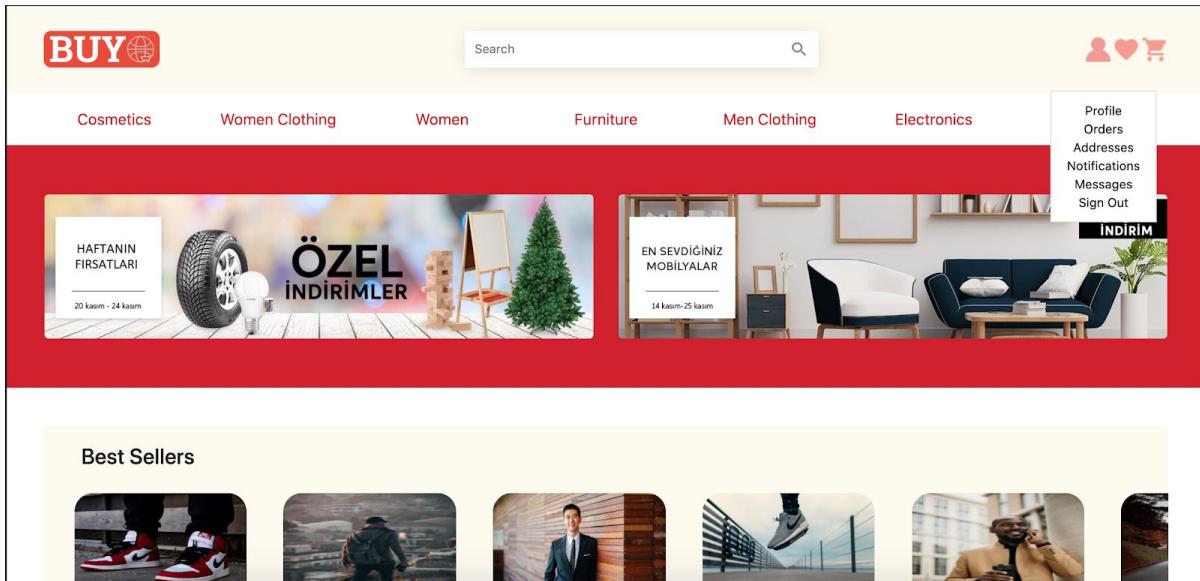
## 2- Sign in/up for vendors



The screenshot shows a sign-up form for vendors. At the top, it asks 'Are you a customer?' with a radio button for 'No'. Below are five input fields: 'Name and surname', 'Company name', 'Business website', 'Business email', and 'Password'. A placeholder text 'Enter the position of your distribution center' is present above a checkbox labeled 'I agree to terms and conditions.' A large red 'SIGN UP' button is centered below the checkbox. To its right are links for 'SIGN UP WITH GOOGLE' and 'SIGN IN'.

Vendors have to provide information visible in the screenshot above to be able sign up. They also have to provide their location via Google Maps and accept terms and conditions.

## 3- Homepage



When the user first goes to our url or after signing in, they are greeted with our homepage. The homepage consists of our header which is also present in every customer page, a promotions section, a recommendations section and a best sellers section.

On our header there is a Buyo icon which which redirects the user to this homepage from anywhere in the website, there is a search bar which redirects the user to a page with the resulting products of your search query when you click the search icon, a wishlist icon which will redirect to the wishlist page, a cart icon which will redirect to the cart page, and a profile

icon. When the user clicks on the profile icon, a popup menu opens underneath with all the pages relevant to the users profile. From that menu, the user can navigate to profile information page, orders page, addresses page, notification page, messages page, or just sign out. If the user is not signed in, that menu will just show two options as sign in and sign up which will redirect to corresponding pages. The header also hosts the categories bar. When you click on one of the category, its subcategories will appear. When you click on one of the subcategories, you will be redirected to the products page consisting of products in that subcategory.

Below the header is our promotions tab. The user can see currently available promotions, however it is not yet interactable in this version of our platform. Below those we have Best Sellers and Recommendations sections. The user sees a list of best seller products or personally recommended products and when the user clicks on one of the products, she is redirected to its product details page

#### 4- Profile information for customers

The screenshot shows a profile information page. At the top, there's a navigation bar with categories: Furniture, Women Clothing, Men Clothing, Cosmetics, Women, Electronics, and Men. On the right side of the header are icons for user profile, heart, and shopping cart. The main content area has a title 'Profile Information'. It contains input fields for Name (Deniz), Surname (Yilmaz), Email (deniz@gmail.com), Phone number, and Gender (Female). There's also a 'Change password' section with fields for Current password, New password, and New password (again). A red 'Update your password' button is located at the bottom right of this section. A red 'Update your profile' button is at the bottom left.

Customers can change their profile information and change their password using their current password. They can update their name, surname, phone number and gender.

#### 5- Addresses Page for Customers:

The screenshot shows an 'Addresses' page. At the top, there's a navigation bar with categories: Cosmetics, Women Clothing, Men, Electronics, Men Clothing, Women, and Furniture. On the right side of the header are icons for user profile, heart, and shopping cart. The main content area has a title 'My Addresses'. It displays two address entries in a table format. The first entry, 'Ev', has the address: Meric 05348297556, Sakarya Apartman no:4 Besiktas/Istanbul. The second entry, 'Is', has the address: Meric 05392837644, Akasya Avm is konaklari daire Akasya/Istanbul. Each entry has edit and delete icons. A button 'Add New Address' is located at the bottom left.

This page behaves exactly the same as the addresses section on the checkout page. For more information, see the checkout page manual.

## 6- Cart for customers

The screenshot shows a 'Cart' page with two items:

- Bumba Deluxe Orthopaedic Bed** (Vendor: JohnsShop)  
Quantity: 1  
Price: 4000 (Original) / 2000 (Discounted)  
Features: Bumba, Yes
- Women Casual Blue Jeans** (Vendor: AyseTeyze)  
Quantity: 1  
Price: 119.99 (Original) / 89.99 (Discounted)  
Size: S  
Color: Blue

On the right, there is a 'Cart Total' summary:

4119.99
2089.99
<b>Checkout</b>

Customers can view the products they added to their cart. They can see the quantity and features they chose as well as the cart total. They can also remove the product from their cart using the trash icon. Clicking on the checkout button will proceed to payment. Cart page is reachable using the cart icon on the header.

## 7- Checkout

The screenshot shows a 'Checkout' page with three main sections:

- Shipping Address**: Shows two address options: 'Ev' (Home) and 'Is' (Office).
  - 'Ev': Meric, 05348297556, Sakarya Apartman no:4, Besiktas/Istanbul
  - 'Is': Meric, 05392837644, Akasya Avm is konakları daire, Akasya/Istanbul
- Payment Information**: Fields for Name On Card, Card Number, Exp. Month, Exp. Year, and CVV. A 'BUY!' button is at the bottom.
- Summary**:

Product Total	111.99
Discount	-22
Total	<b>89.99</b>

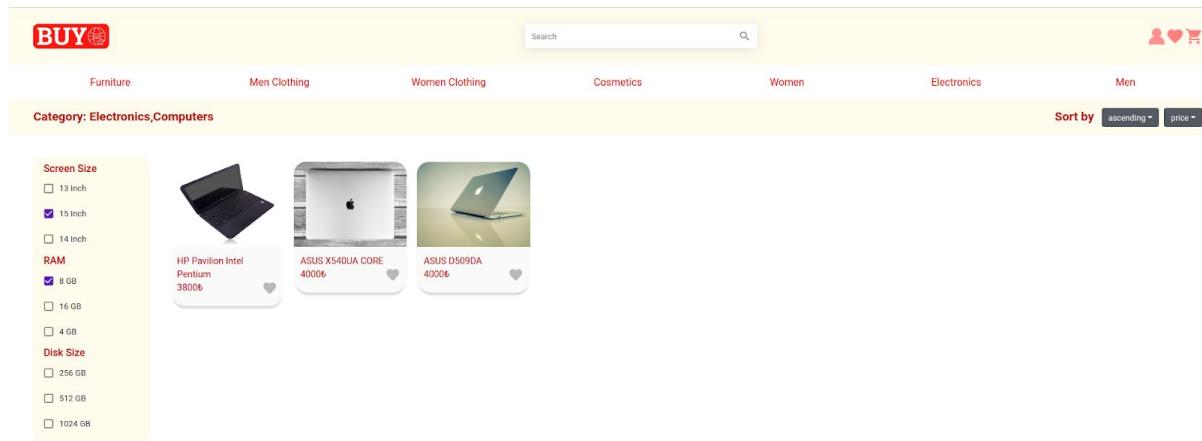
A checkbox at the top right states: "I have read and accepted the Preliminary Information Form and the Distance Sales Contract".

We have four sections on the checkout page. In the shipping address, the user can select any of her addresses by clicking the desired one. If she would like to add a new address, she can click the 'Add New Address' button and filling the form that pops up. By clicking the pen icon on top of one of the addresses, the user can edit that address. The same form pops up with the information already filled. The user can then change any of the desired

information besides the title and click submit to apply the changes. The user can also delete any of the addresses by clicking on the trash icon on top of the address.

On the top right section, the user has to check to checkbox claiming she has read and agrees with our legal documents before completing the payment. Just below that, there is a summary section where the user can see the total amount she is going to pay and the discount she got. On the bottom left, the user has to fill in her payment information and click buy to complete the payment. If everything is successful, the user sees a popup stating that and redirected to homepage.

## 8 - Product List Page



Users can see the products of a specific category here by choosing the category from the header. This page also displays the products of the search result. Users can enter their query at the search bar found in the middle of the header. On the left the possible filter options are listed. Users can choose at most one filtering criteria for each attribute listed. They can also remove the filter by clicking again. If a user is logged in, she can also add a product to her wish list by clicking the like button(button with the heart symbol). Sorting functionality is also provided. Users can sort the products by rating, name or price in ascending or descending order. Clicking on a product redirect the user to the page of that product which is explained next.

## 9 - Product Page

The screenshot shows a product page for a men's brown coat. At the top, there are navigation links for Men Clothing, Furniture, Women Clothing, Women, Cosmetics, Electronics, and Men. A search bar is also present. The main content area features a large image of a man wearing the coat. Below the image, the product name is 'Men Smart Brown Coat' by 'Boss'. It has a rating of 3.3 from 5 stars. The price is listed as 'Price: 449.99€'. There are dropdown menus for 'Choose size:' (XS) and 'Choose color:' (Brown), and a 'Quantity:' input field set to 1. A red 'Add to Cart' button is located at the bottom right. To the right of the product details, there is a sidebar with a 'Report Product' button. The sidebar also shows a seller rating of 4.5 from 5 stars and a 'Report Product' button. A 'Comments' section follows, showing a single comment from a user named 'b\*\*\*\*\*' with the message 'Cooool!' and a 'Report Comment' button. The page has a light yellow background with some shadows.

Users can see the information about a product on this page. Moreover they can interact with our platform by reporting the product, reporting a comment, adding the product to their wish list or adding the product to the cart. Before adding the product to the cart the user can choose different options for the product such as its size, color, material etc.

## 10 - Wishlist Page

The screenshot shows a wishlist page. At the top, there are navigation links for Women, Electronics, Men Clothing, Cosmetics, Women Clothing, Furniture, and Men. A search bar is also present. The main content area shows a 'Wishlist' heading. Below it, there are three items listed in a grid: 1. Men Black and Red Sneakers, 2. Hermes Men Perfume, and 3. Lenovo Thinkpad. Each item has a small image, the product name, the original price, the current price, and a red heart icon indicating it's in the wishlist. The page has a light yellow background with some shadows.

This page displays the products which are added to the wish list of a user. Users have to be logged in to see their wishlist. Users can access the products listed here by clicking on them. If the price of a product added to the wishlist changes, the user is informed via notifications page.

## 11 - Customer Orders Page

The screenshot shows a user interface for a shopping website. At the top, there's a navigation bar with categories: Furniture, Men Clothing, Women Clothing, Electronics, Cosmetics, Men, and Women. A search bar is also at the top. Below the navigation, a section titled 'Orders' is highlighted in yellow. It lists three recent orders:

Order No.	Date	Shipping
KKDT0M4FJ9R8M	26.01.2021 12:35:26	9.99€
KKDVOQJ2HBQKB8	26.01.2021 13:51:34	9.99€
KKDVRXMQHLVEO	26.01.2021 13:52:40	9.99€
KKIDH0F030R6P	29.01.2021 17:19:08	9.99€

For the second order (KKDVOQJ2HBQKB8), there's a detailed view showing a product image of a man in a brown coat, product details (Men Smart Brown Coat Boss 449.99€), size (XS Brown), quantity (1), vendor (AysTezye), and status (Shipped). A red 'Message Vendor' button is visible next to the vendor information.

Here the users can list all of their orders. By clicking on an order more detailed information is displayed. Users have the ability to cancel the order if its status is ‘Pending’ or to return the order if its status is ‘Delivered’. They can also give feedback to a product here if the status of the product is delivered. Lastly, they can message the vendor to request information about their order by the ‘Message Vendor’ button.

### Give Feedback

Rate this product: ★★★★★

We are glad that you liked this product! Please share with us your comments to inform everyone.

Comment

The product was good as I expected!

CANCEL GIVE FEEDBACK

### Message Vendor

Send your message to the vendor:

Your Message

CANCEL SEND MESSAGE

## 12 - Messages Page

Buy

Search

Women Clothing Cosmetics Furniture Men Clothing Men Electronics

**Messages**

Burak Cuhadar: Hi, the product description was wrong. Therefore I cancelled your order.  
30.01.2021 18:03:44

Orhan Yıldırım: But this was an important order for me. How can you do this so unexpectedly?  
30.01.2021 18:04:17

Veli Ünal: You are right being angry at us. We are deeply sorry for the inconvenience.  
30.01.2021 18:05:02

Burak Cuhadar: Please make sure that this kind of problem does not occur again.  
30.01.2021 18:06:08

Orhan Yıldırım: Of course we will! Thanks for your understanding.  
30.01.2021 18:06:33

Type Something

On the left panel of this page the users can see the names of the users who they interacted with via messaging. By clicking on a user their messaging history is displayed on the right. The users can then send a message by typing a message to the input field at the bottom and then pressing enter or the right arrow button found on the bottom right. The messaging is real time which means that the messages sent and received are displayed as soon as the messages are sent or received without refreshing the page.

## 13 - Notifications page

Buy

Search

Women Clothing Cosmetics Women Electronics Furniture Men Clothing

**Notifications**

1/31/2021, 9:59:07 PM  
Osman Gürgül's iPhone 7 price dropped from 10000 to 8000

1/31/2021, 9:58:22 PM  
Your order KKL6NMR20LOZ is cancelled by the vendor

This page displays notifications pushed to the user, due to activity such as a wishlisted product being discounted, or an order being canceled. Each notification card includes the activity's timestamp as well as a short informative message. The users can click on a notification to go to a relevant page. For example, clicking on a discount notification redirects to the discounted product's page. Clicking on a canceled order redirects to the user's orders page.

## 14 - Vendor profile page

The screenshot shows the vendor profile page. At the top, there is a navigation bar with links: BUY, Products, Orders, Profile, Add Product, Notifications, Messages, and Sign Out. Below the navigation bar, the title "Profile Information" is displayed. On the left, there is a form for updating basic profile information: First Name (Osman), Last Name (Gürgil), Company Name (Osman Holding), Website (www.osmanholding.com.tr), and Email (osman@osmanholding.com). On the right, there is a "Change password" section with fields for Current password, New password, and New password (again), and a "Update your password" button. Below the profile information, there is a map of Istanbul showing the location of the distribution center. A red pin marks the location near the European Side. At the bottom of the map is an "Update your profile" button.

This page displays the profile of vendor users, and allows them to update their profile information. Vendor users can edit their first name, last name, company name, and website. They can also change the location of their distribution center by clicking on the map, and change their account password.

## 15 - Vendor add product

### Add a new product

#### 1. Choose a category

Chosen category: Women,Dress

Women Clothing

Men Clothing

Cosmetics

Furniture

Women

Trousers

Dress

Coat

Electronics

Men

Next

To add a product, the vendor first chooses a category.

#### 2. Enter product information and criterion

Name

Floral dress

Brand

H&M

Description

Beautiful flowy dress

Price

125.99

Discounted price

109.99

Enter the original price if there is no discount.

Enter product criteria and press enter (size, color, material etc.)

size

color

Enter product criteria

Next

Then, adds product information and criterion.

### 3. Enter possible values for criterion separated by ,

For ex. Size: S,M,L

size

S,M

color

red,yellow

Next

### 4. Enter stock information

size

S

S

M

M

color

red

yellow

red

yellow

Stock

5

6

7

8

Next

Then, enters the possible values and stock information.

### 5. Upload an image of your product

Choose File photo-1496747611176-843222e1e57c.jpeg

Upload

### 6. Submit your product

Submit

Then upload an image and submit the product.

## 16 - Vendor all products page

The screenshot shows a vendor's dashboard with a navigation bar at the top. The navigation bar includes a logo labeled 'BUY' with a globe icon, followed by links for 'Products', 'Orders', 'Profile', 'Add Product', 'Notifications', 'Messages', and 'Sign Out'. Below the navigation bar are two product cards. The first card features a photo of a person wearing a floral dress, with the text 'Floral dress' and 'H&M' below it. It shows two price options: '125.99₺' and '109.99₺'. The second card features a photo of a person wearing a red winter coat, with the text 'Winter Coat' and 'Pull&Bear' below it. It also shows two price options: '349.99₺' and '348₺'. Each card has an 'Edit' button and a 'Delete' button.

Vendors can see all of their products on this page with their price information. They can use the delete button to delete the product. Edit button redirects to update product page.

## 17 - Vendor update product page

The screenshot shows a form titled '4. Enter stock information'. It starts with a section titled 'Current stock information:' displaying the current stock levels for different size and color combinations. Below this is a table where vendors can update the stock. The table has columns for 'size', 'color', and 'Stock'. For each combination, there is a dropdown menu for 'size' and 'color', and an input field for 'Stock'. The table rows correspond to the current stock information shown above. A 'Next' button is located at the bottom right of the form.

size	color	Stock
S	red	<input type="text"/>
S	yellow	<input type="text"/>
M	red	<input type="text"/>
M	yellow	<input type="text"/>

Update product page has the same interface as the add product page except for the stock information part. In this part, vendors can see their current stock information and update it. Other fields are the same as add products, except that the form is loaded with current information.

## 18 - Vendor orders page

**Orders**

Order No: KKL16NMR20LOZ	Date: 1/31/2021, 9:54:22 PM	Shipping: 9.99฿ ▾											
Total Earnings: 5100.00฿													
<p>On this page, vendor users can see a list of all of their orders, as well as their total earnings.</p> <p>Clicking on an order displays more detailed information, as seen below.</p> <table border="1"><tr><td>Order No: KKL16NMR20LOZ</td><td>Date: 1/31/2021, 9:54:22 PM</td><td>Shipping: 9.99฿ ^</td></tr><tr><td></td><td>Samsung Galaxy 5 Samsung 5000฿</td><td>white Quantity: 1 Status: Pending</td><td><a href="#">Message Customer</a> <a href="#">Cancel Order</a></td></tr><tr><td></td><td>Nokia 3310 Nokia 100฿</td><td>blue Quantity: 1 Status: Pending</td><td><a href="#">Message Customer</a> <a href="#">Cancel Order</a></td></tr></table>			Order No: KKL16NMR20LOZ	Date: 1/31/2021, 9:54:22 PM	Shipping: 9.99฿ ^		Samsung Galaxy 5 Samsung 5000฿	white Quantity: 1 Status: Pending	<a href="#">Message Customer</a> <a href="#">Cancel Order</a>		Nokia 3310 Nokia 100฿	blue Quantity: 1 Status: Pending	<a href="#">Message Customer</a> <a href="#">Cancel Order</a>
Order No: KKL16NMR20LOZ	Date: 1/31/2021, 9:54:22 PM	Shipping: 9.99฿ ^											
	Samsung Galaxy 5 Samsung 5000฿	white Quantity: 1 Status: Pending	<a href="#">Message Customer</a> <a href="#">Cancel Order</a>										
	Nokia 3310 Nokia 100฿	blue Quantity: 1 Status: Pending	<a href="#">Message Customer</a> <a href="#">Cancel Order</a>										

Vendors have the ability to cancel the delivery of any product with a ‘Pending’ status, by clicking the ‘Cancel Order’ button next to it. They can also click the ‘Message Customer’ button to message the customer who made the order, to give any relevant information about their order.

## 19 - Admin login

Username:

Password:

**Login**

The login page for the admin portal is quite simple. The admin just needs to enter its credentials and click login. And if they are right, she will be redirected to the admin portal.

## 20 - Admin - Reported Products Page

Reported Products

Report Message: Bad skate	Report Message: Inlkrm
 <p>Name: Men Casual Blue T-shirt Description: 100% Natural. 100% Nature Friendly. It will definitely look good on you. You won't regret buying this. Attributes: size color Attribute Values: XS Blue S Original Price: 99.99   Price: 79.99 Brand: Kappa</p> <p><a href="#">Remove Product</a> <a href="#">Ban Vendor</a></p>	 <p>Name: Sony Headphones XC-II Description: Powerful sound &amp; unique enjoyment! Attributes: noiseCancelling Attribute Values: No Original Price: 1700   Price: 1500 Brand: Sony</p> <p><a href="#">Remove Product</a> <a href="#">Ban Vendor</a></p>

The admin can see all reported products listed in this page. All the necessary information in the products box. After that if the admin wishes to remove the products from the database, she can by clicking the remove product button. Also the admin can ban the owner of the product from the system as well by clicking on the ban vendor button.

## 21 - Admin - Reported Comments

Reported Comments

Report Message: I dont like it	Report Message: I JUST WANT TO REPORT	Report Message: I dont like this comment
<p>Comment: I hated this product !!!</p> <p><a href="#">Remove Comment</a> <a href="#">Ban User</a></p>	<p>Comment:</p> <p><a href="#">Remove Comment</a> <a href="#">Ban User</a></p>	<p>Comment: What a stupid product!</p> <p><a href="#">Remove Comment</a> <a href="#">Ban User</a></p>

Similar to the reported products page, admin can see all the reports about comments on this page. She can see the comment itself and the reason it was reported and also remove the comment from the system or ban the user who made the comment entirely by clicking on the appropriate buttons.

## 12. System Manual

### Android

- You need to have an Android development environment that supports Kotlin. You need Android 5.0 and above (API Level 21). Suggested IDE is Android Studio 3.0 and up.
- When you have a suitable IDE, clone the project from Github Repository, open the source code in IDE. Then, Install a virtual device or connect an Android phone.
- No additional setup necessary to build and run BUYO app. All operations are common to usual Android project operations.

### Backend

#### Requirements

- Docker
- Redis

#### Deployment

1. Clone the repository from this url:  
**git clone https://github.com/bounswe/bounswe2020group4.git**
2. Go to backend directory by using this command  
**cd ./bounswe2020group4/backend**
3. It is very important to update nginx file with your current IP address.

Check your ip address with this command: ifconfig

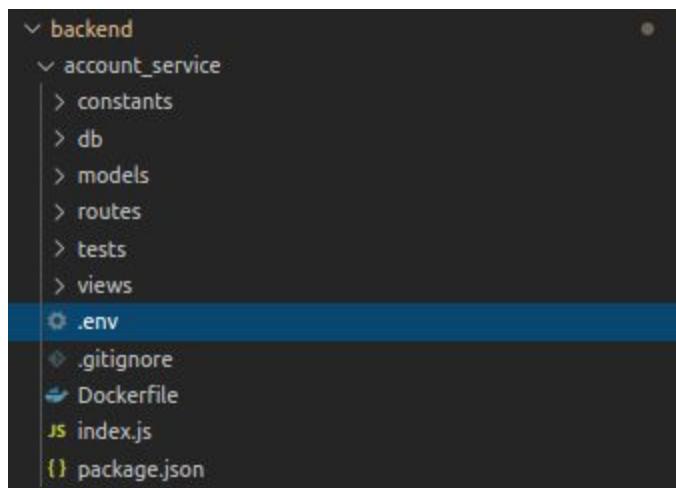
```
wlp2s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.1.193 netmask 255.255.255.0 broadcast 192.168.1.255
        inet6 fe80::65f0:3d64:855f:3c11 prefixlen 64 scopeid 0x20<link>
          ether ac:ed:5c:5d:19:89 txqueuelen 1000 (Ethernet)
            RX packets 519708 bytes 488412181 (488.4 MB)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 305034 bytes 124907375 (124.9 MB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

- Type the value (192.168.1.193 for our case) instead of new one for every service:

```
backend > ⚙ nginx.conf

    You, seconds ago | 4 authors (koraycetin and others)
1 user nginx;
2
3 events {
4     worker_connections 1000;
5 }
6
7 http {
8     server {
9         listen 8080;
10        location /product {
11            proxy_pass http://192.168.1.193:5000; // New one
12        }
13        location /categories {
14            proxy_pass http://3.141.25.245:5000; // Current one
15        }
16        location /wishlist {
17            proxy_pass http://3.141.25.245:5000;
18        }
19        location /cart {
20            proxy_pass http://3.141.25.245:5000;
21        }
22        location /db {
23            proxy_pass http://3.141.25.245:5001;
24        }
25        location /google-signin {
26            proxy_pass http://3.141.25.245:5002;
27        }
28        location /login {
29            proxy_pass http://3.141.25.245:5002;
30        }
31    }
32}
```

- Make sure that you added the .env files in each service



#### .env file :

PORT = 9876

MONGO\_URL =

"mongodb+srv://buoyo-dev:buoyoapp123@buoyo-app.aycr0.mongodb.net/buoyo-app?retryWrites=true&w=majority"

```
TEST_MONGO_URL =  
"mongodb+srv://buyo-dev:buyoapp123@buyo-app.aycr0.mongodb.net/test-db?retryWrites=true&w=majority"
```

**Warning :** For the **interaction service**, please add below to .env file as well:  
**REDIS\_URL** = "redis://cache"

6. Make sure that there is no package-lock.json in the services.

7. Install redis

- on Mac

**brew install redis**

- on Ubuntu

**sudo apt install redis-server**

8. Run redis

- On Mac

**brew services start redis**

- On Ubuntu

**redis-server**

9. Build docker containers

**sudo docker-compose build**

10. Make docker containers up and running

**sudo docker-compose up**

## Running Tests

1. Visit the root file of backend
2. Give execution permission to run\_tests.sh  
**chmod +x run\_tests.sh**
3. Run tests with the bash file  
**./run\_tests.sh**

## Frontend

Running the application in local development mode:

1. Pull the repository to your local machine

2. Inside the repository, run ‘npm install’ to install necessary npm packages that is not stored on the git repository.
3. Then run ‘npm start’ to run the react application in development mode. This will serve the application on port 3000 (or some other port if it is being used) where you can access it from any web browser by going to the url localhost:3000

Deploying the application to ‘<http://ec2-3-17-180-100.us-east-2.compute.amazonaws.com>’:

A CI/CD workflow is implemented using github actions. So when you make a commit or merge a pull request to the master branch, our github actions workflow will automatically do the necessary steps to deploy the application on our server.

Deploying the application to another machine:

- Pull the repository to your desired machine.
- Make sure there are no images named webimage or running containers remaining.
- Inside the repository, build the docker image with command: ‘docker build -t webimage web’
- Run the docker image with command: ‘sudo docker run -p 80:80 -d webimage’
- The application is now live on your machines port 80

The same steps can be followed to deploy the admin project (with different image names and different port if you like)

Linting:

- We are using the package eslint for linting. Eslint rules are configured in .eslintrc.json
- To use it normally: ./node\_modules/.bin/eslint src
- To automatically fix most of the common cases: ./node\_modules/.bin/eslint src --fix

Running the Packaged Web Application on the Submission Delivery Package:

- Unzip web-build file
- Run ‘npm install -g serve’ (You may need npm for this. Also on some platforms, you may need to grant access to some folders)
- Then run ‘serve -s build’
- You can follow the same steps for the admin application

## **13. Assessment of the Final Customer Presentation**

The final presentation was overall good. We demonstrated our applications with realistics and detailed scenarios. Our customers understood how to use our applications and they were pleasant about the product as we observed.

After the demo, our customers had some questions about several features, and we answered them properly. Our communication was healthy and we were all on the same page during the meeting.

Unfortunately, we had some problems during the presentation. One of them occurred during the front-end demo. We could not show the correct message from the vendor to the customer during the scenario. The message was sent right after the vendor canceled the customer's order, via the 'message customer' button. The reason was that we had (mistakenly) made the order from a different customer's account (not the customer we were using during the demo); so the message was sent to a different customer. We solved this issue after the presentation. We explained the reason and showed the correct message being displayed in the correct customer account. We also sent a message from the customer to the vendor, and showed it as well, to showcase that the messaging system worked both ways without any problems.

In addition to the aforementioned issue, we had some small bugs in the android application of which we were aware before the presentation. We explained the bugs in the demo, we also mentioned them in our README file and in our GitHub repository.