Group4-Milestone 1

1. Executive Summary

Introduction

Our project, Buyo, is a platform that aims to connect on-line vendors to their users. Guest users are free to search, filter or sort any product they desire according to the criteria such as product name, vendor name, vendor location, ratings and price. Guest users are free to sign up as a customer anytime they want by providing email, user name and a password. Customers have all the functionalities of guest users in addition to being able to make lists to keep track of items wish to buy, having a cart page which they can add items before checkout and an orders page which they are able to keep track of the items they have bought, cancel or return an item and comment and rate on the products they have bought. Guest users can also sign up as vendors if they provide at least one store location in addition to the email, user-name and password. Vendors are able to keep track of their orders, cancel them, make discounts on their products and read user comments on their products.

What's been done so far

Requirements: We have extracted the requirements from the project description, and then conducted a meeting with our customer to specify the details where needed.
Mock-ups/Scenarios: 3 user personalities who will use the system were defined and mock-ups of their behavior were created. We tried to balance the number of user types and platforms simulated.
User/Class/Sequence Diagrams: We created the user, sequence and class diagrams according to our requirements and then adjusted them according to the customers feedback.

Project Name/Logo: We have voted on and chosen a product name and logo that we felt best fits our branding.
Project Plan and RAM: We have planned the rest of the project activities, tasks, milestones, and resource allocation for the upcoming weeks and created an RAM for that.

What's next

In the upcoming weeks, we will create the skeleton of the platform including creating an Amazon server, creating skeletons for the API, web front-end and Android front-end , creating a database and the direct messaging protocol. After that we will be creating prototypes from the skeletons we've developed, writing test cases for our prototypes and documenting our API.

2. List and status of deliverables

Deliverable Name	Delivery date	Status
Communication plan	Feb 18, 2020	Delivered
Requirements	Mar 3, 2020	Delivered
Scenarios & mockups	Mar 3, 2020	Delivered
UML Software Designs	Mar 24, 2020	Delivered
Project Plan	Apr 21, 2020	Delivered
RAM	Apr 25, 2020	Delivered

3. Evaluation of deliverables

Communication Plan

We decided on our communication tools during the first meeting, we tried to choose the most functional tools so that our communication is simple but effective . One of the group members set up all the communication tools for use. Slack became our main communication tool thanks to its functionalities, we use its Github integration to keep ourselves up to date with our repository, we also created separate channels for topics such as meetings, help, specific tasks etc. We use Whatsapp only for urgent communication. We use Github issues to track our tasks and store our deliverables. Piazza is where we communicate with the instructors. We did not use Trello much because Github was enough for task tracking but for more complex tasks, it is available for us to use. We currently use Zoom for our weekly meetings.

Requirements

Requirements is the basis for our project so we took time to prepare it in detail. We split in groups to brainstorm different groups of requirements and then, other groups reviewed those requirements and did the necessary changes. Thanks to the instructors' feedbacks, we updated and refined it numerous times and we believe that final result is satisfactory as it is detailed and extensive. It is our reference for any project task.

Scenarios & Mockups

We prepared three scenarios and their corresponding mockups: customer user with web app, vendor user with web app and guest user with mobile app. In order to cover a range of functionalities, we used different users as well as different platforms. We also updated both scenarios and mockups according to the feedback so that the scenarios are more specific and target functionalities more.

UML Software Designs

We prepared class, use case and sequence diagrams as UML software design documents. We split into two groups, responsible for class and use case diagrams. Sequence diagrams were done by each group member.

Class diagrams: We first brainstormed classes, fields and their relations and wrote them in a Google Sheet during an online meeting, then a member drawed the diagram according to the information on the sheet. We communicated a lot and did a meticulous job to make sure the relations were logical and functional.

Use case diagrams: Use cases were added and fixed on the shared diagram over time where group member discussed and reviewed them. We made sure to cover a wide range of use cases.

Sequence diagrams: Each group member picked a use case and drawed their diagrams

We also updated these diagrams after the feedback from our instructors.

Project Plan & RAM

Different members of the group extracted the past and future work of the project. To extract the past work, meeting notes were used as they were very helpful to list the assignments and assignees. Future work is a tentative schedule but it gives us guidance for future tasks.

RAM was also created using meeting notes.

4. Evaluation of tools and processes

GitHub

GitHub is a repository hosting service for Git that also has a web based graphical interface. We mostly used the issues feature and wiki page of our Github repository.

With GitHub's issues feature tasks are assigned to each group member. By adding labels to the issues, we document the tasks. It is possible to indicate the priority, type, necessary effort to complete the task and other features of a task by adding various labels to the issue. Furthermore, with the labels it is possible to keep track of the status of an issue. Also, we can give feedback or discuss the given feedback from the instructors under an issue.

In our wiki page on GitHub, we document all the work that has been done so far and give in depth information about our project. We indicate the meeting notes, requirements, design documents, project plan of our project and other information related to the project by adding new pages to the wiki page.

Slack

We use slack as our communication channel. It is possible to open different channels for different topics. We made polls or shared files with the add-ins for other workplace tools of Slack. It is possible to use both web and mobile applications.

Trello

Trello is a collaboration tool that organizes the projects into boards. We make progress tracking with Trello by adding new cards for different tasks and assign them to corresponding group members. Every move made on the system can be seen by notifications. It is possible to use both web and mobile applications.

Lucidchart

Lucidchart is a cloud based platform for visual communication and cross platform collaboration. It is used for creating flowcharts, UML models, process maps, etc. In our project we used

Lucidchart to create sequence, use case and class diagrams. It is very easy to create diagrams even for a one who does not have any design experiences.

ProjectLibre

ProjectLibre desktop is an open-source project management software system. We create the project plan's Gantt chart using ProjectLibre. It has a user-friendly interface.

5. Summary of individual work

Name	Contributions
Emre Girgin	I have attended all of the weekly meetings so far from the beginning of the semester. I customized the issue labels and prepared a guide for it in the wiki. Then, we divided the requirements into three parts and I was a member of the subgroup categorizing the requirements with Olcayto and Çağrı. Afterward, Salih and I prepared the Vendor Mock-up. For the diagrams, I prepared a Sequence diagram on my own and Class diagram with Salih, Eylül, and Çağrı. I prepared 2 items of the Project Plan (Milestone1 and Milestone2). I researched API development and prepared a guideline for it from scratch. Lastly, prepared 3 simple issue templates. In addition to the tasks listed above, I contributed to the fixing of the requirements, mock-up, and class diagram according to

	feedback. Also, took the meeting notes of one of the meetings.
Berkay Alkan	Until now, I have attended most of the group meetings. I have created a page in our wiki about myself and added it to the opening page of our group's Github repository. I formalized the requirements and corrected some grammatical errors, added some requirements and made changes on requirements according to given feedback. I created a web site scenario and mockups of it for a customer user with Olcayto, then made changes on it according to given feedback. I have attended the customer meeting of our group. I created the 'Add Product to Cart' sequence diagram. I extracted the third week's work for the project plan and wrote on a Google Sheets document. Also, I have prepared one of the meetings agenda and took meeting notes.
Mehmet Erdinç Oğuz	I have attended most of the meetings so far. I made research about the Git and W3C Standards at the beginning of our project. I helped create issues and reviewed some of the issues my teammates created. I helped ask questions about the requirements to understand the requirements better. I

	participated in creating the Guest the Mockup for the mobile app. I prepared the meeting agenda and took meeting notes once. I helped the team make changes on the project when there is feedback. I helped create the use case diagram. I prepared the future work part of our Project Plan, which includes the part from now to the end of semester and the next semester related stuff. I helped choose the logo and the name of our project. And finally, I contributed to responding to feedback in general.
Eylül Yalçınkaya	I attended most of the weekly meetings, took notes and shared them with my teammates as well as put them up as meeting notes. I created my wiki page and added my favorite Github repo on our wiki. I also often did housekeeping on issues (creating issues, closing old issues, writing descriptions etc.). I prepared the communication plan and set up the tools for group use. I was one of the team members that extracted, reviewed and updated the requirements according to feedback. I came up with brand names and logos. I reviewed and gave feedback on our scenarios & mockups. I was one of the team members

	that created class diagrams, I also created a sequence diagram. I wrote the list and evaluation of deliverables for the Milestone I report.
M.Olcayto Türker	I attended most of the meeting so far. I have created a page in our wiki about myself and posted it on our group's Wiki page .Then,we categorized the requirements with respect to certain criteria with Emre and Çağrı. After that, we also finalized categorization of requirements according to feedbacks. Also, I created a customer user mockup scenario with Berkay. Then,we made the requiring changes on it according to feedbacks to our presentation. I contributed to user diagram and created login sequence diagram. After that, I changed our sequence diagrams according to feedback we got.
Burak Çuhadar	I attended all of the group meetings. I have created the initial version of the wiki page. I have added github profile pictures of the group members to the readme file and the group photo to the wiki page. I organized the extracted requirements by writing them in markdown properly. I have also corrected and added new requirements according to the feedback from the assistant. I have reviewed the scenarios created by other

	members and suggested improvements. I created a sequence diagram called “Customer returns a product”. To the use case diagram I added actors, sign-in and sign-up use cases, search and filter use cases, and several other use cases. I have created RAM on my own. To the project plan I added the tasks about UML diagrams.
Katariina Korolainen	I attended most meetings before leaving Turkey. So far I have added my personal page and favourite GitHub repository to the wiki, plus extracted requirements from the Project description, created Guest scenario & mockup with Mehmet, revised requirements on a couple of weeks, created Use case diagram with other assigned group members, made Customer Sign-up sequence diagram, extracted previous tasks from group meeting notes to Project Plan and made the Gantt chart in ProjectLibre.
Salih Kasım Benli	I have created my personal wiki page and added my favorite Github repo on our wiki in addition to writing my interest areas. I have joined most of the weekly meetings, taken notes and shared them with my teammates. I have opened and resolved several issues. As a part of a small group, I have extracted requirements from project description. I

	have created scenarios with a vendor and web application along with Emre. I have created the 5th meeting's agenda and took notes of it. I have also created issues regarding that. Then I have worked on fixing the issues about the scenarios we created. I was a member of the group which created class diagrams. I have also worked on extracting the past work from Github repository and Meeting Notes from the past. I have prepared the "Remove Product from Cart" sequence diagram. I have spent time helping my teammates to understand how an API works, how an application can be deployed to AWS since I had experience of doing them. I will be finalizing this document when every part gets ready by the assigned ones.
Koray Cetin	I've attended most of the meetings. I've created my wiki page and added my favourite repo to the wiki. I configured the trello board columns, created example trello cards and added an automation. I came up with a project name and logo suggestion. I've attended the first customer meetings and wrote my notes to the wiki. I've contributed to the use case diagram, created a sequence diagram for "Customer

	cancel order". I've also revised some of the requirements and made changes on them.
Çağrı Çiftçi	I could attend some of group meetings but I read all meeting notes in order to not miss anything about project and I talked some of my group member to take information about the meeting. I made some research about git and I analyze some well-written wiki-page. Then, we divided the requirements into three parts and I was categorizing the requirements with Olcayto and Emre. Then I fixed some of requirements according to the feedback. After that, we review all Mock-ups and scenarios with Eylül and Burak. For the diagrams, I prepared a Sequence diagram and we prepared Class diagram with Salih, Eylül, and Emre. Then, we made the requiring changes on Class diagram according to feedbacks. I prepared the meeting agenda and took meeting notes and created issues for that week once. And in general, I have spent time to understand how to implement backend and frontend.

6. Communication Plan

Where	Purpose	Frequency	Audience	Channel
Slack	Fast communication	When needed	Everyone	Online
Trello	Progress tracking	When needed	Everyone	Online
BM Lounge	Team meeting for weekly review	Every Thursday @9.00	Everyone	Face to face
Github	Progress tracking via issues and wiki	When needed	Everyone	Online
Google Hangouts	Joining weekly meeting via video conference	Every Thursday @9.00	Members who can not attend the meeting physically	Video conference
Piazza	Communicating with customer and instructor	When needed	Everyone	Online
Whatsapp	Urgent communication	When needed	Everyone	Online

7. Requirements

7.1 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 explain and discuss their shopping experience via this functionality.
- Customer: A logged-in user with the special privileges of creating wish-lists, 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.
- List: A page for customers to keep track of the 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", the 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.

7.2 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, vendor, and price range.
- 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 Lists.
  - 1.1.1.1.4.1 Customers shall be able to save lists, 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.
        - 1.1.1.1.7 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.
    - 1.1.1.2 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.

1.2 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.

- 1.2.3 Interface
    - 1.2.3.1 System shall have categories for discovering new products.
- 1.2.4 Product
    - 1.2.4.1 Customers shall be able to create private lists that they can name, edit, delete, add products to, e.g. to keep track of the items' prices.
    - 1.2.4.2 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.
    - 1.2.5.2 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 chosen 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.

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 GPDR 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.

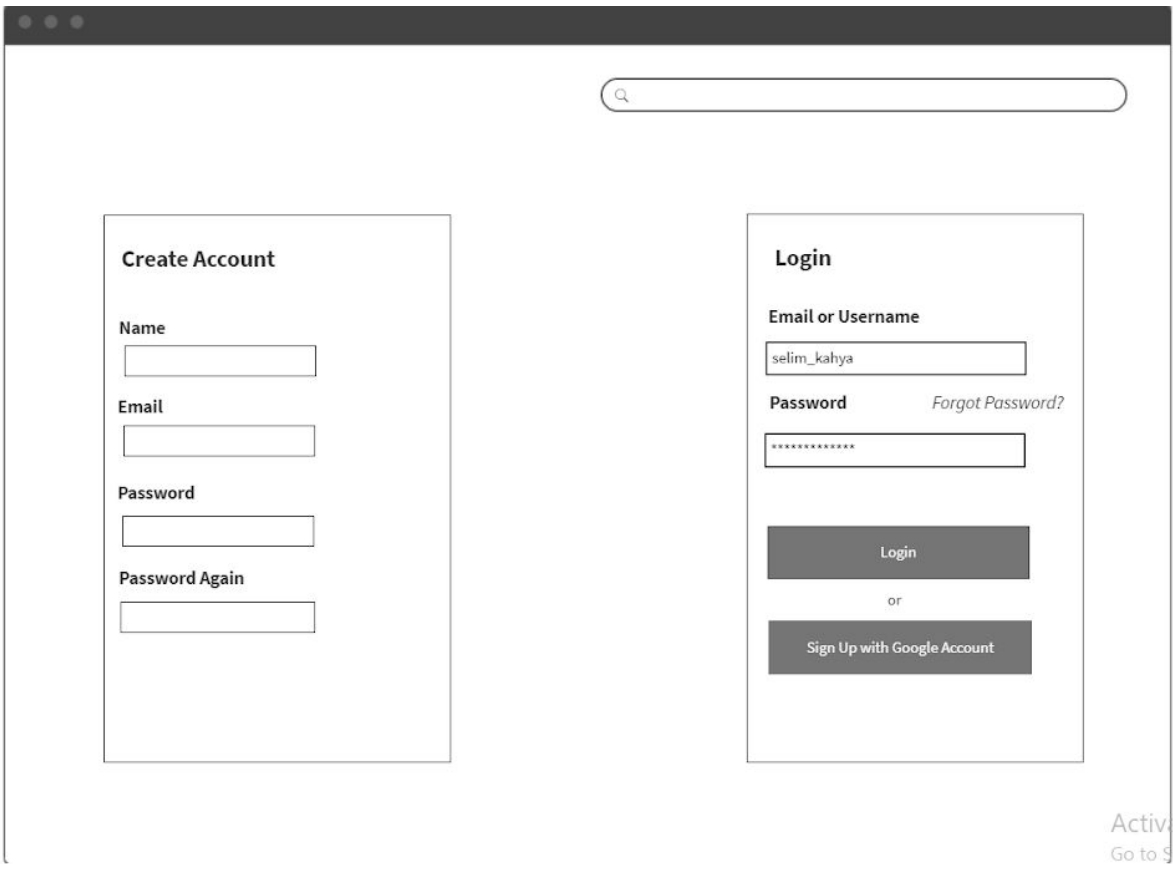
2.5 Performance

- 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.

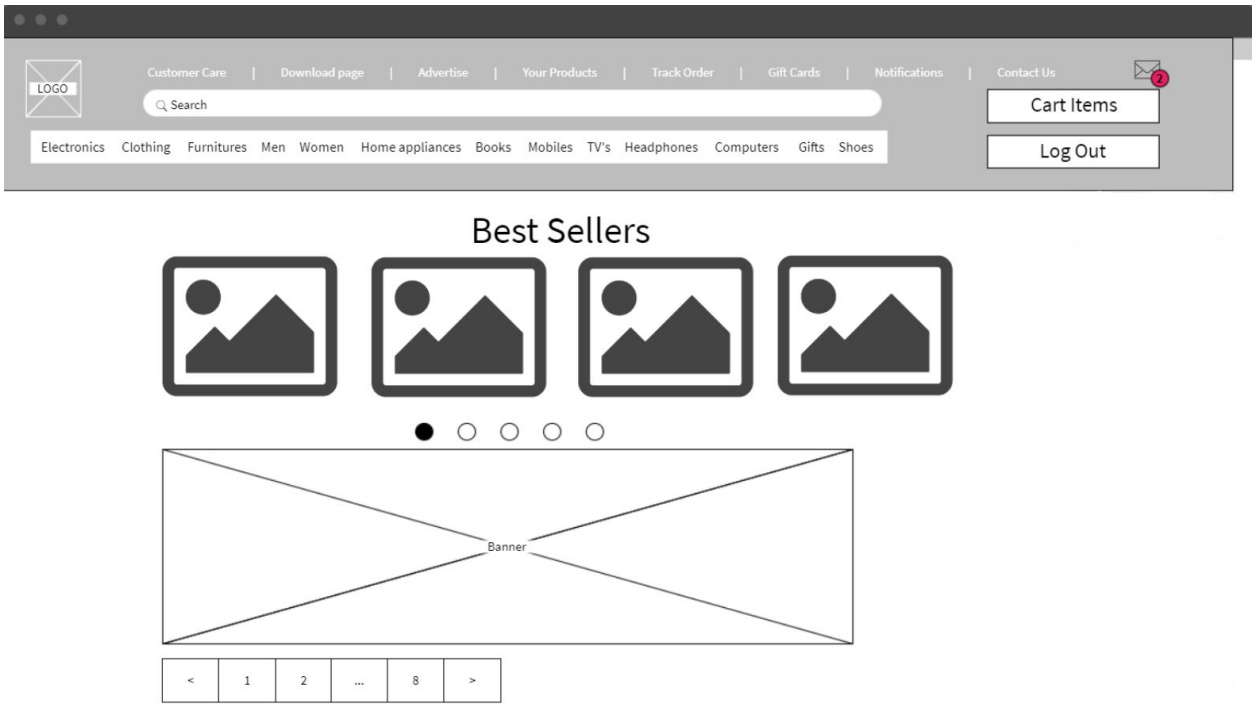
8. Mock-ups

8.1 Customer

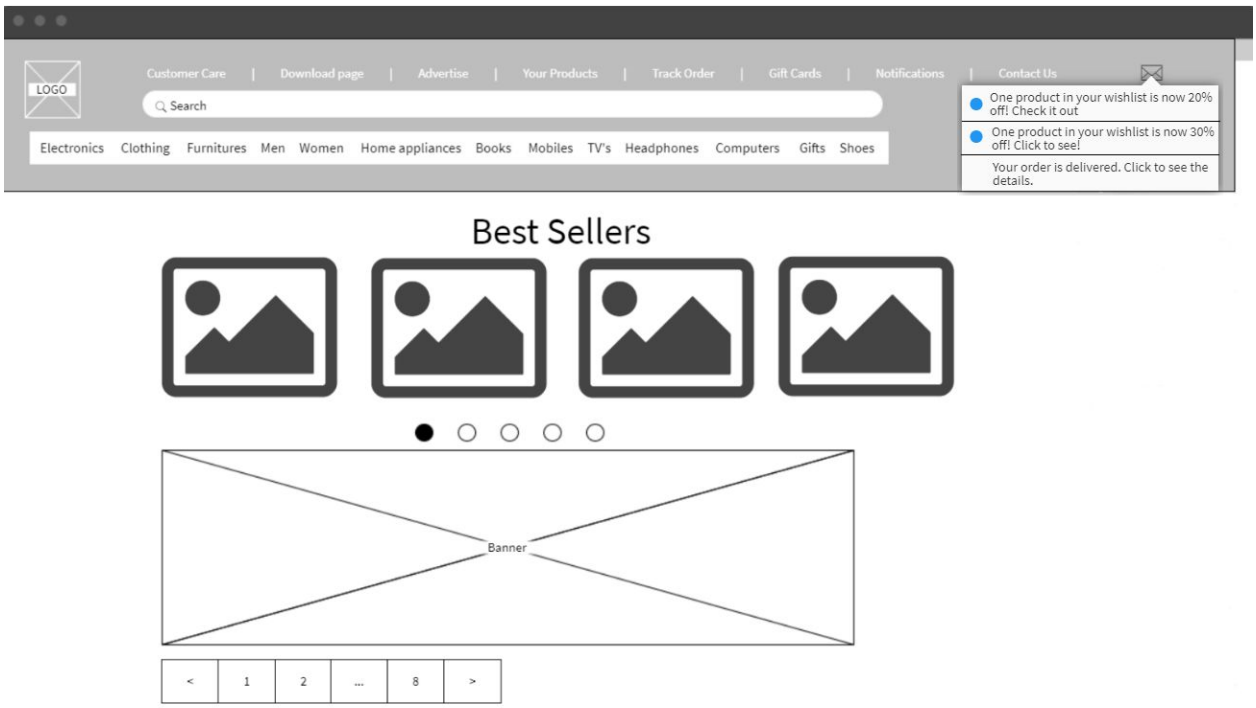
1. Selim opens the main page and logs in.



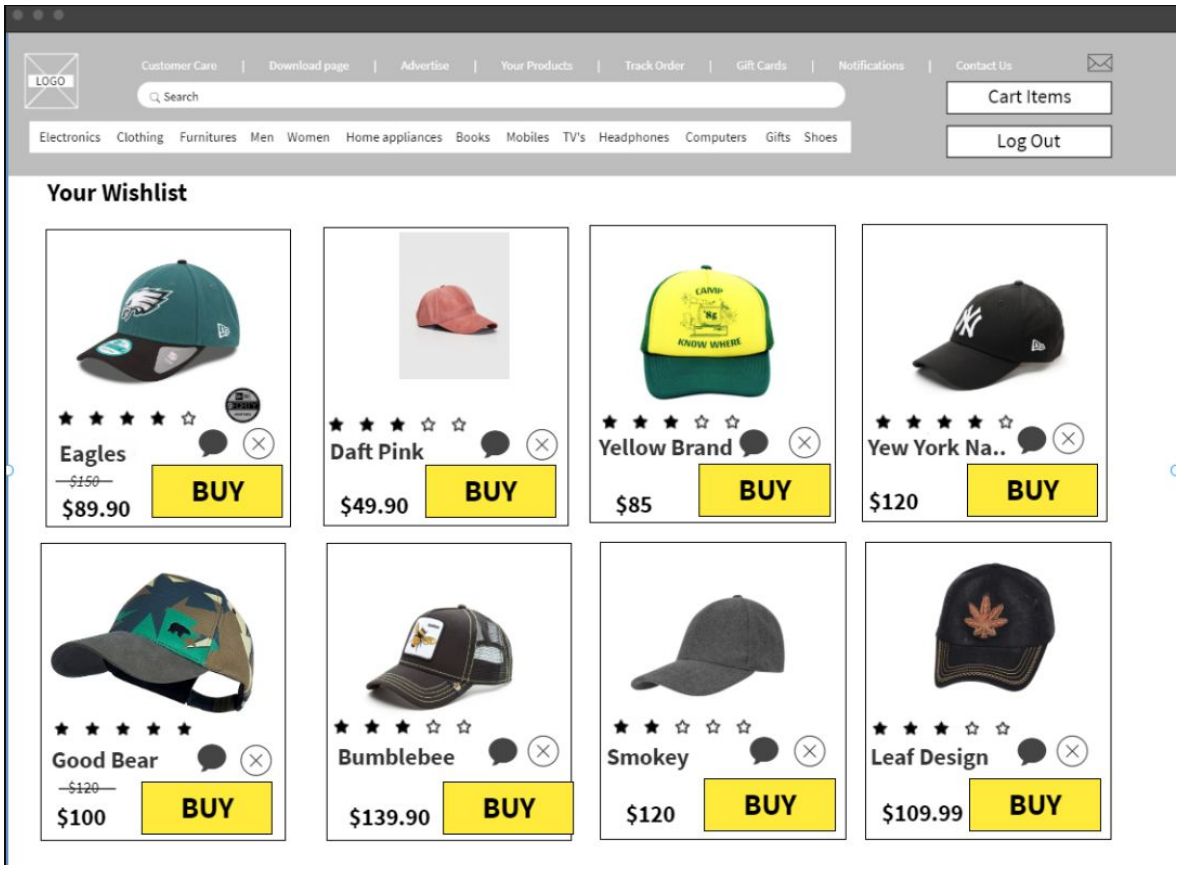
2. Selim realizes the notifications.



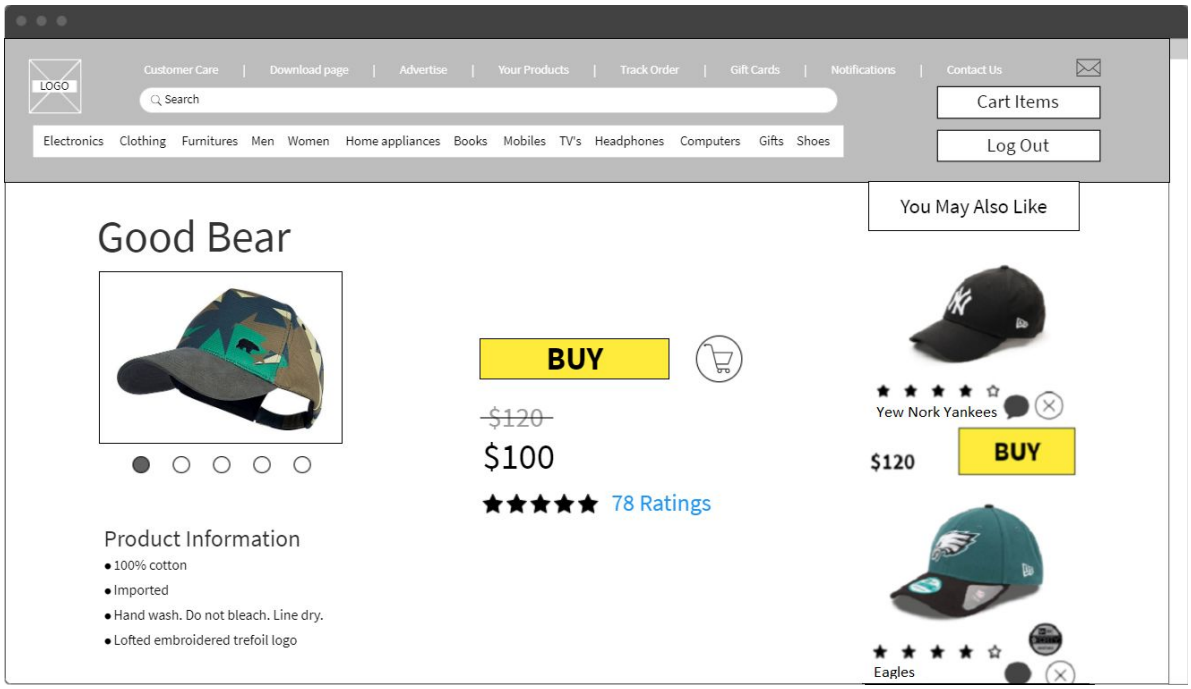
3. Selim opens the notifications.



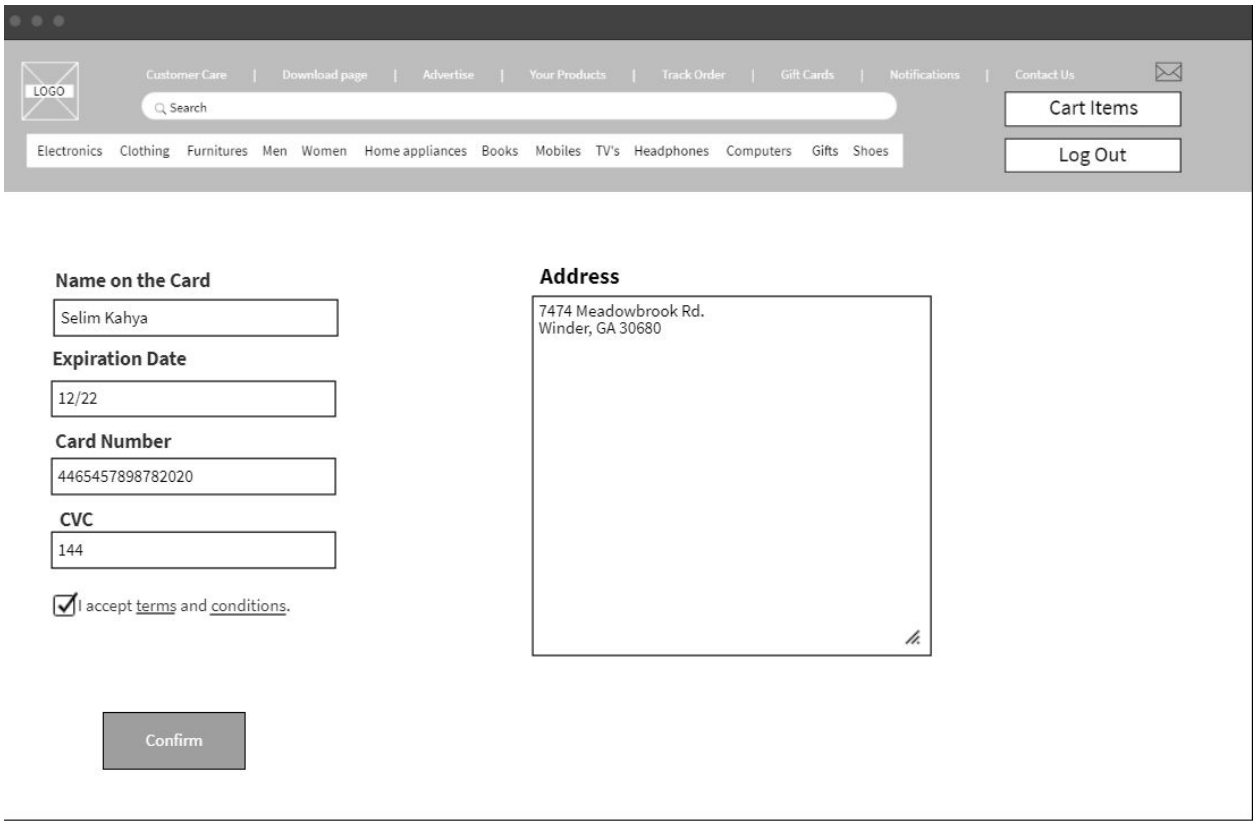
4. Selim clicks to notification and his wishlist is opened.



5. Product Page

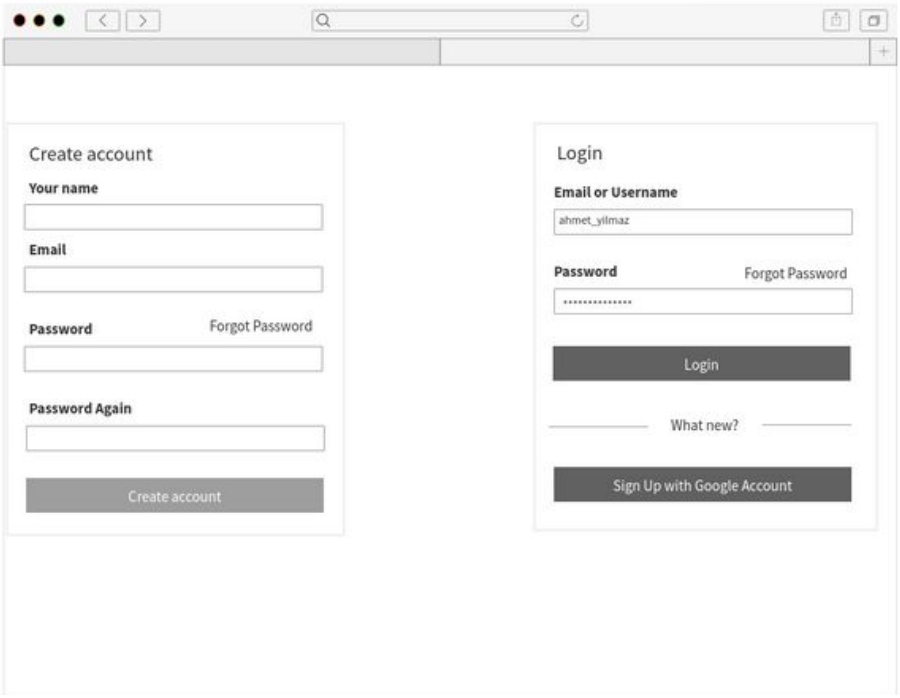


6. Selim writes his card information and address to complete his purchase.

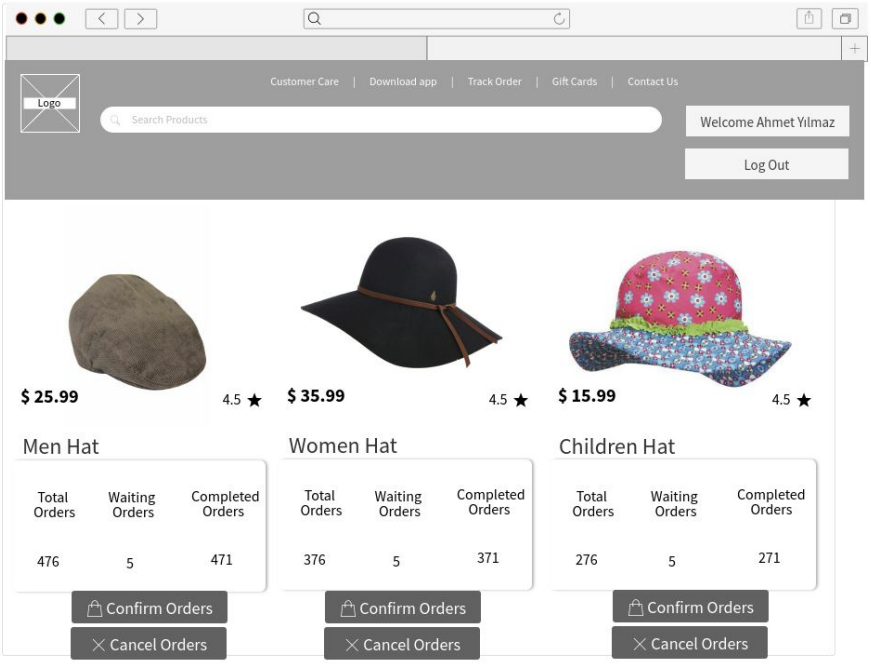


8.2 Vendor

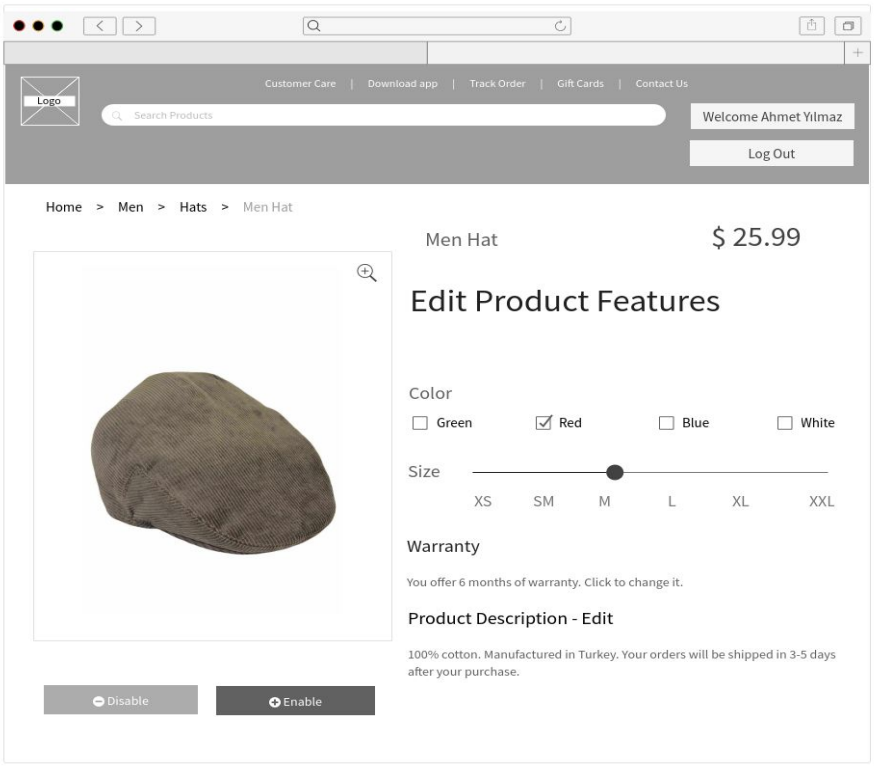
1. Ahmet logs in to the platform.



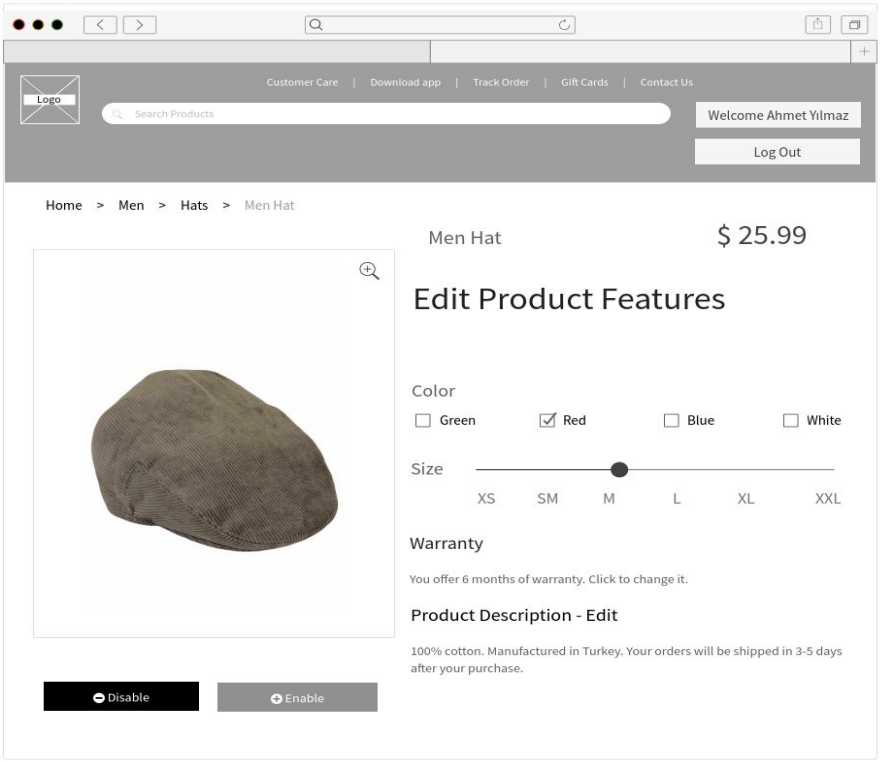
2. Ahmet is redirected to his order's page.



3. Ahmet can disable a product which is out of stock.

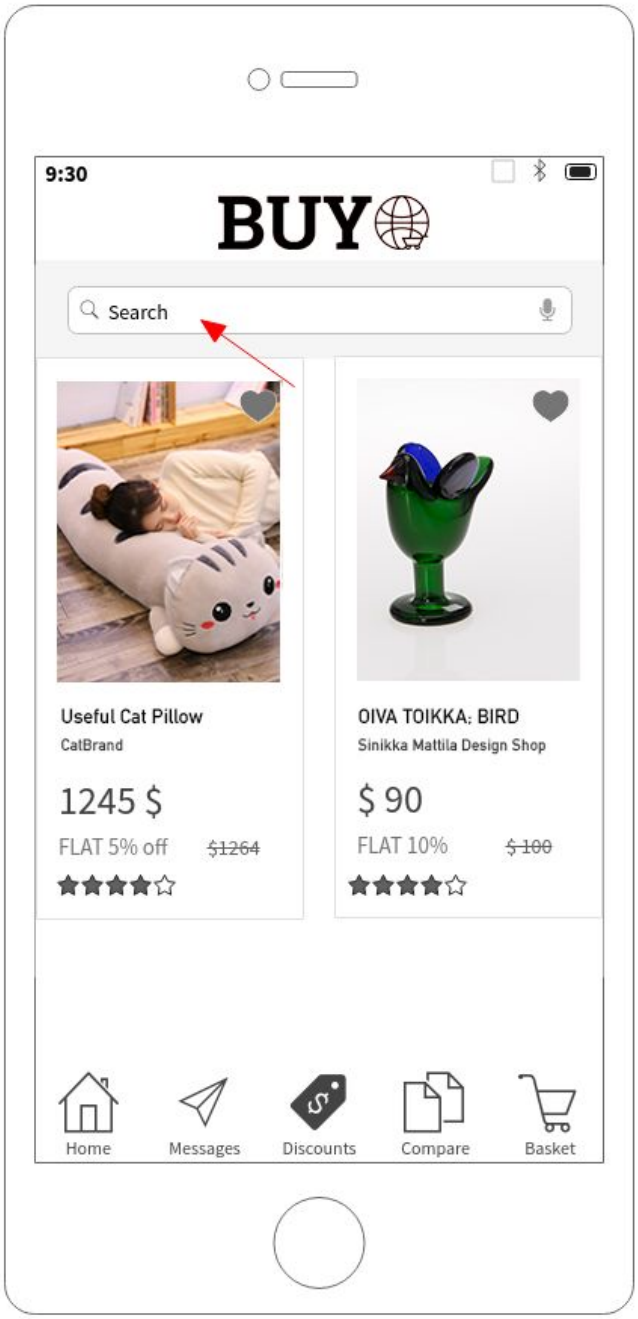


4. Ahmet can re-enable a product.

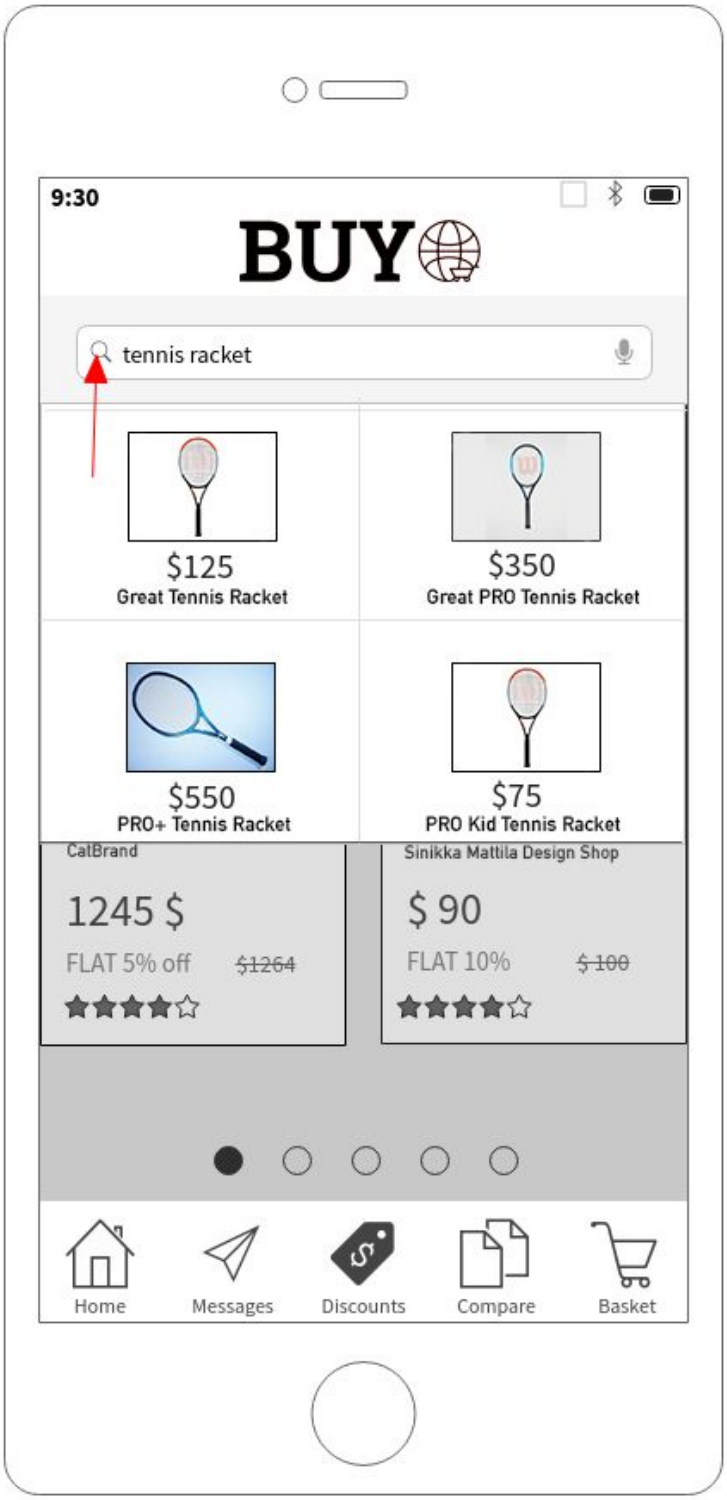


8.3 Guest

1. Sasha opens the e-commerce app on her phone.

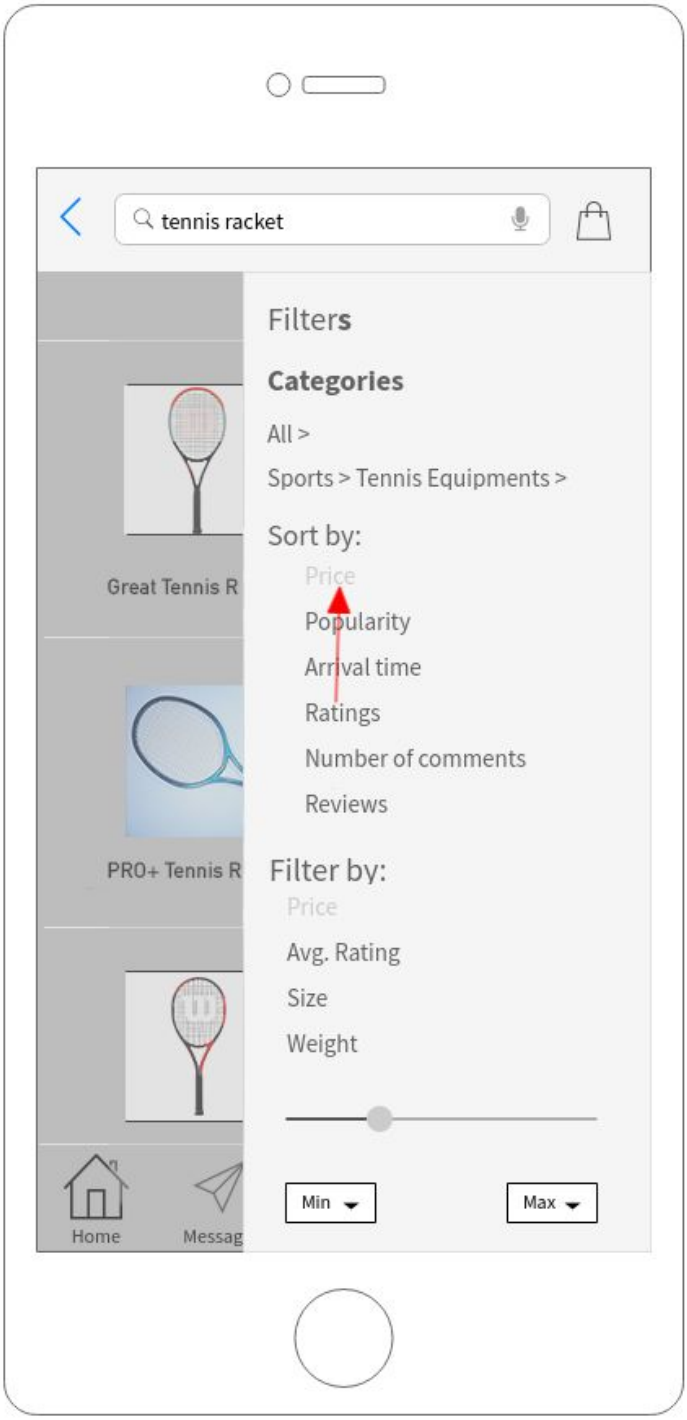


2. Sasha searches for the "tennis racket". (1.2.1.1. Search)

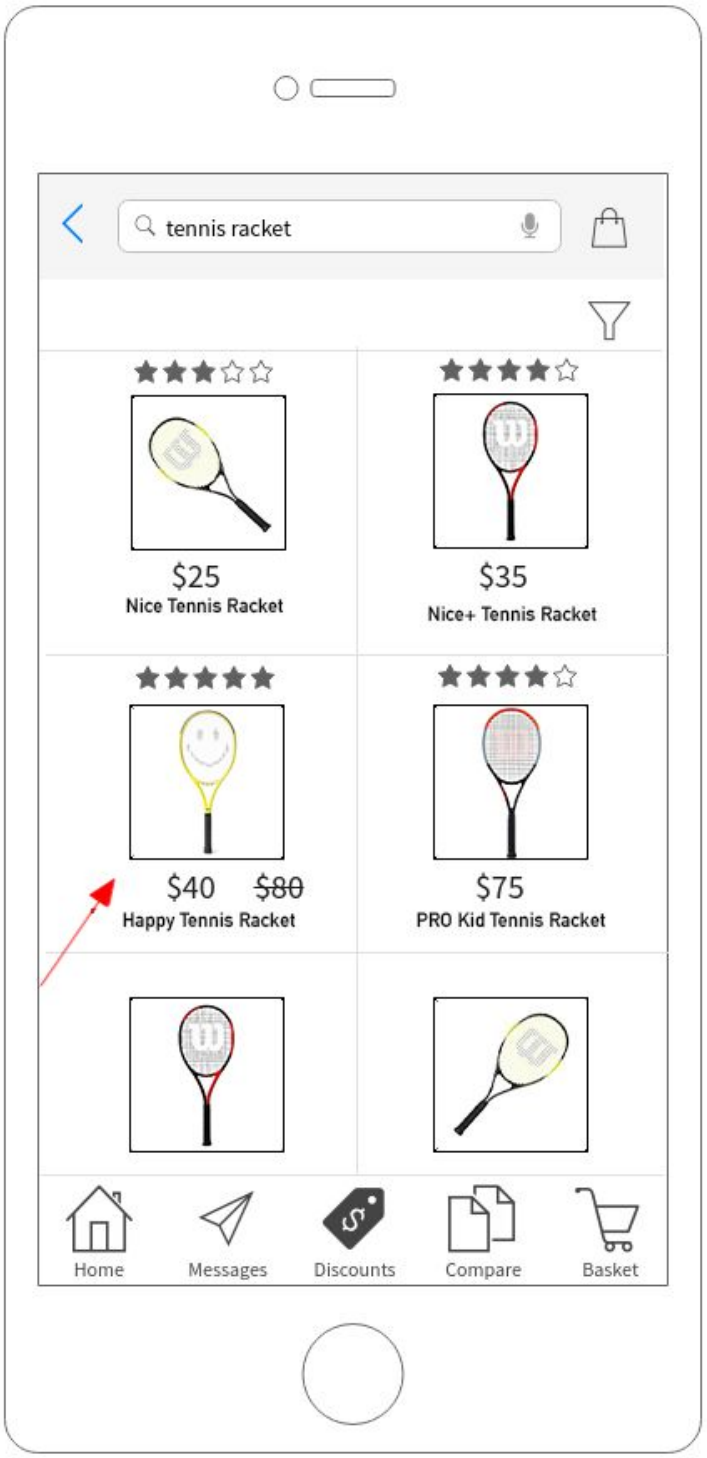


3. Sasha gets the results.

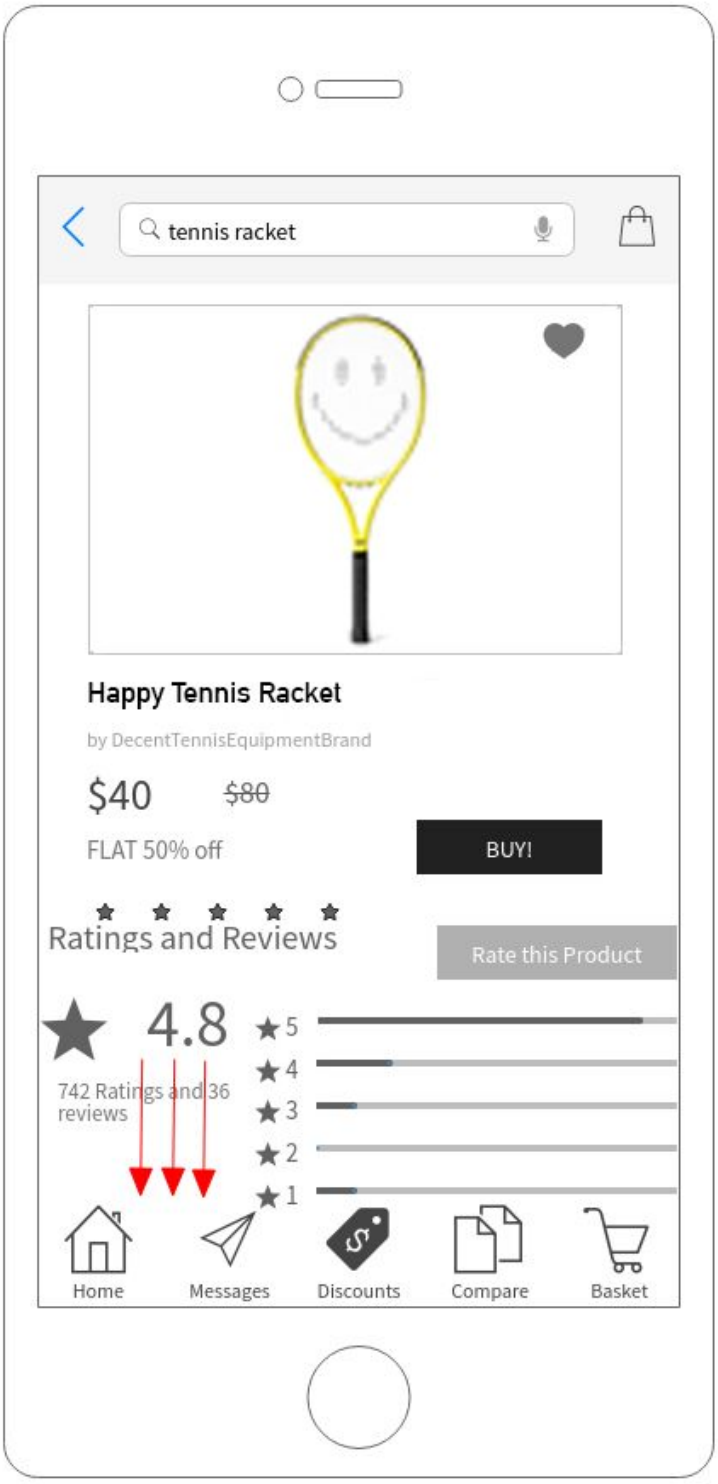
4. Sasha found the best-selling tennis rackets expensive and decides sort the product list. (1.2.1.3.1 Sorting by popularity is considered default)



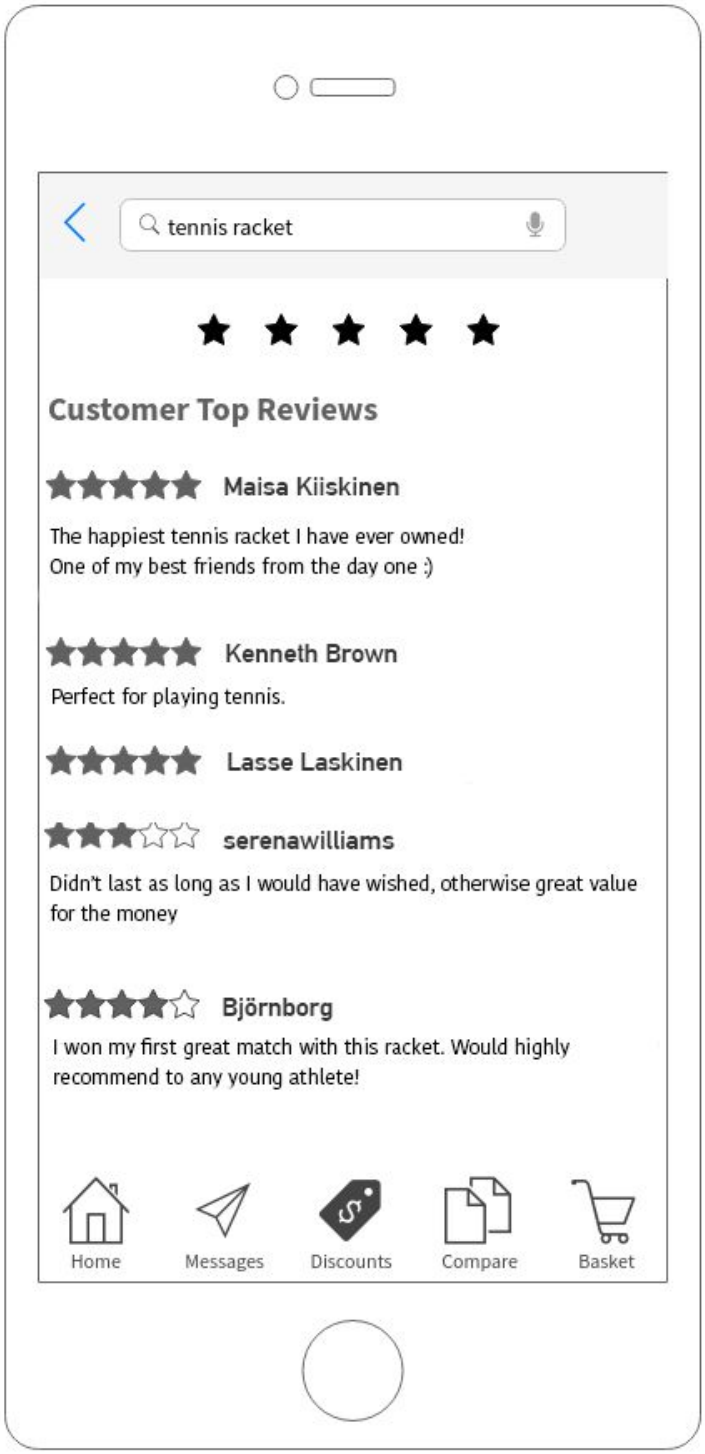
5. Sasha sorts the results in ascending order by price. (1.2.1.3.3 Sort by price)



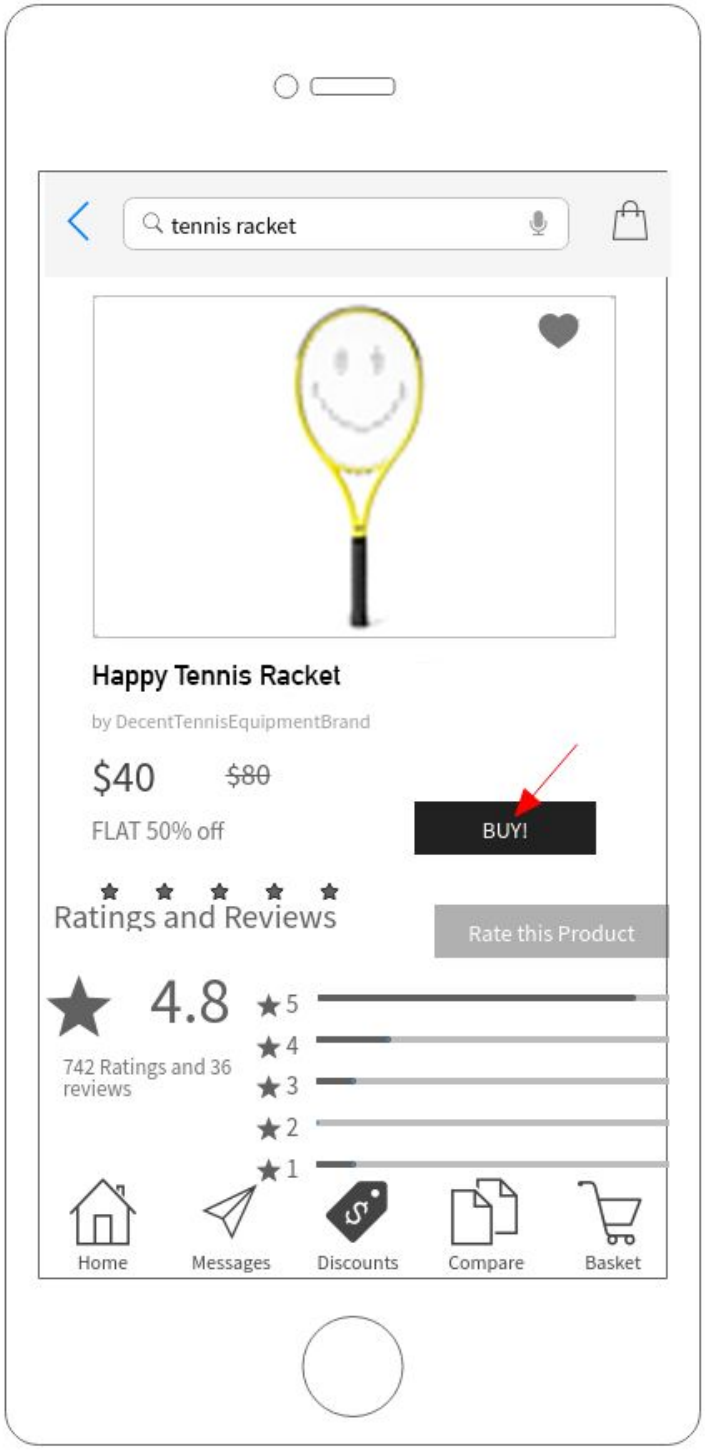
6. Sasha chooses the racket with the best ratings from the list, also considering the price. (1.2.2.2 Rating)



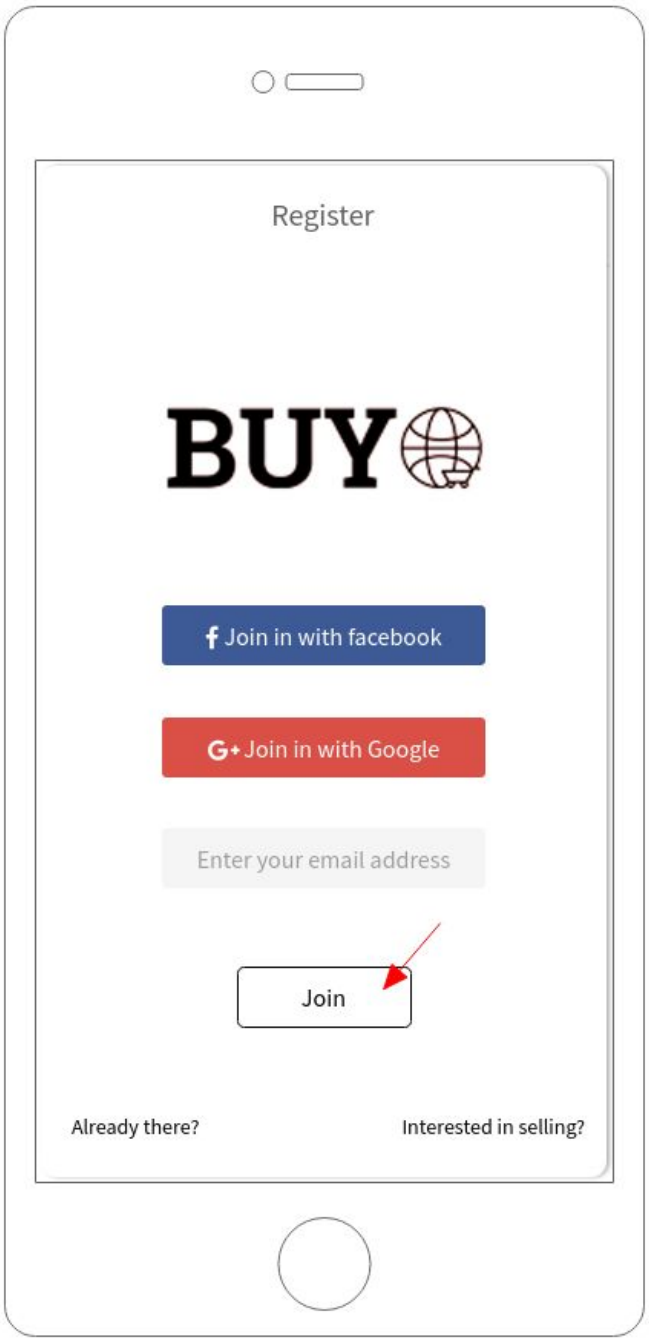
7. Sasha reads the comments about that racket. (1.2.2.1 Comments)



8. Sasha decides to buy the racket.



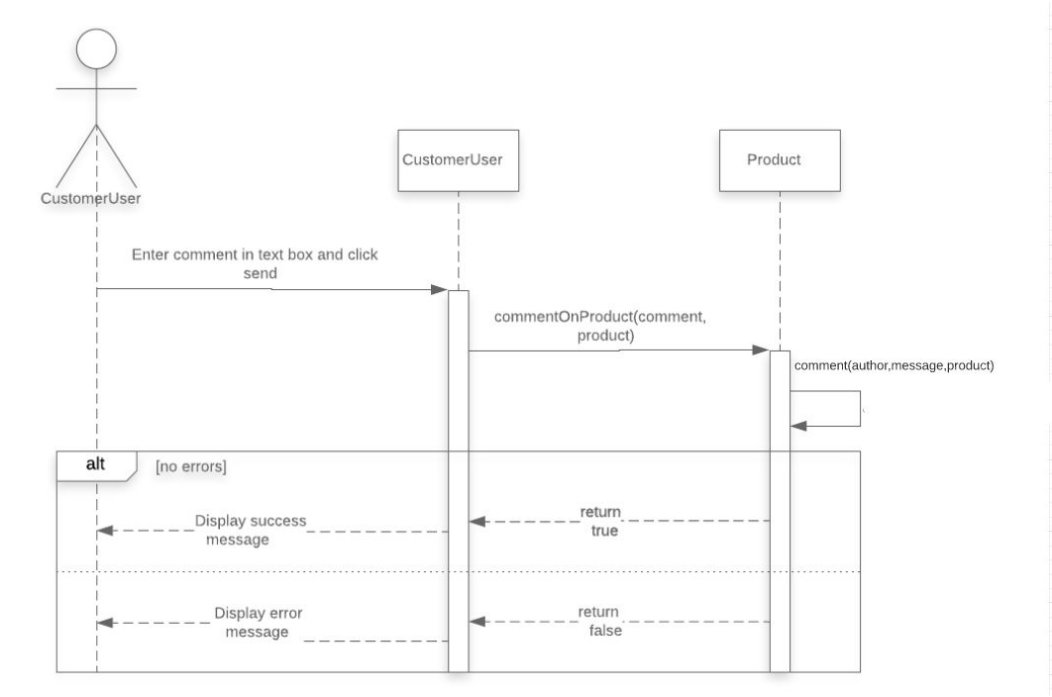
9. The platform responds with a message saying that she should create an account or log in if she already has an account. (1.1.3 Sign up)



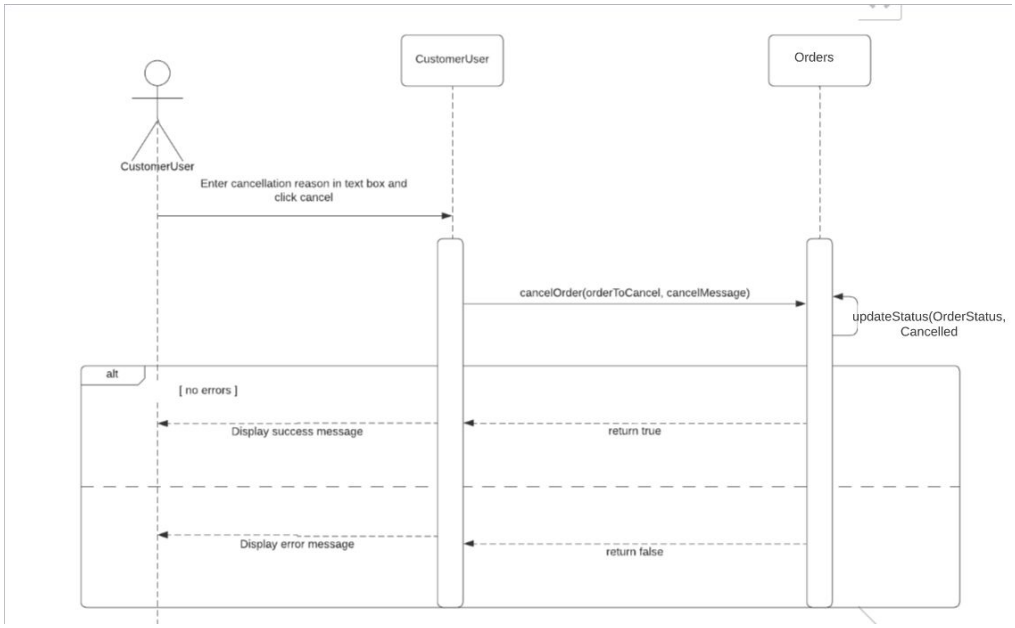
9. Design Documents

9.1 Sequence diagrams

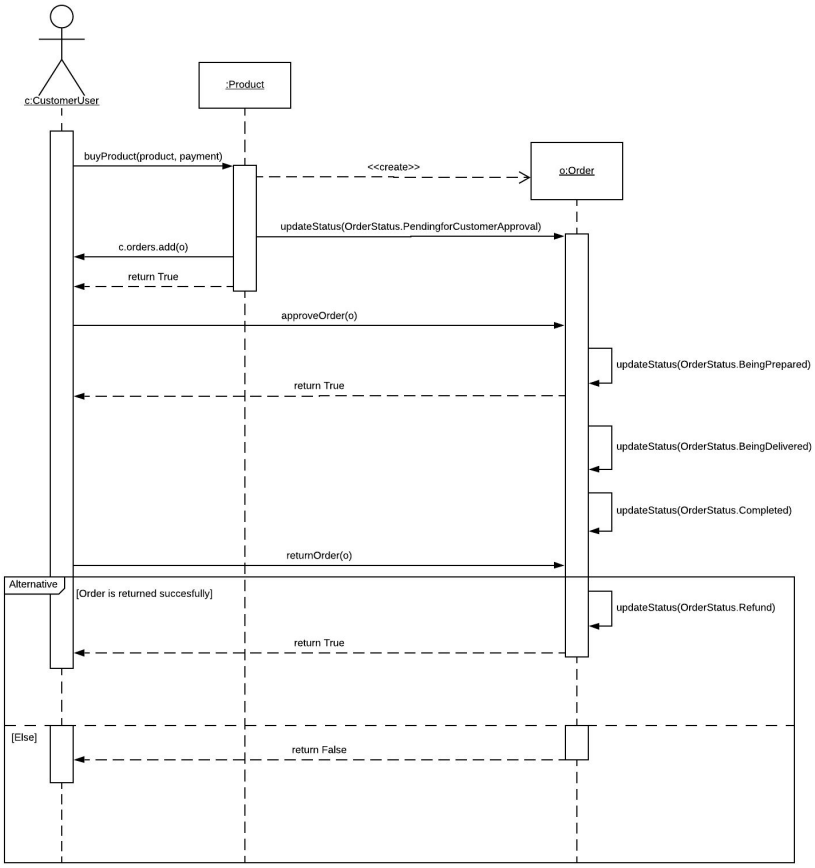
Comment



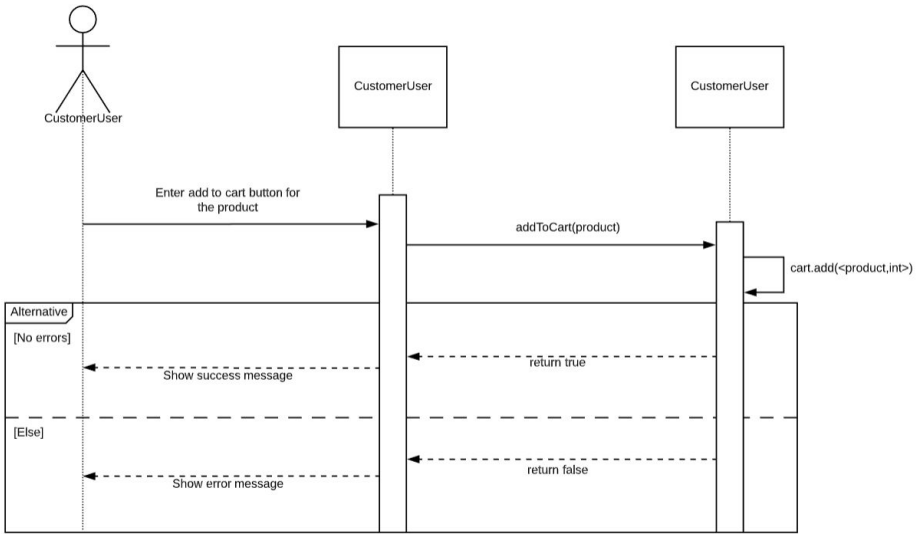
Cancel Order



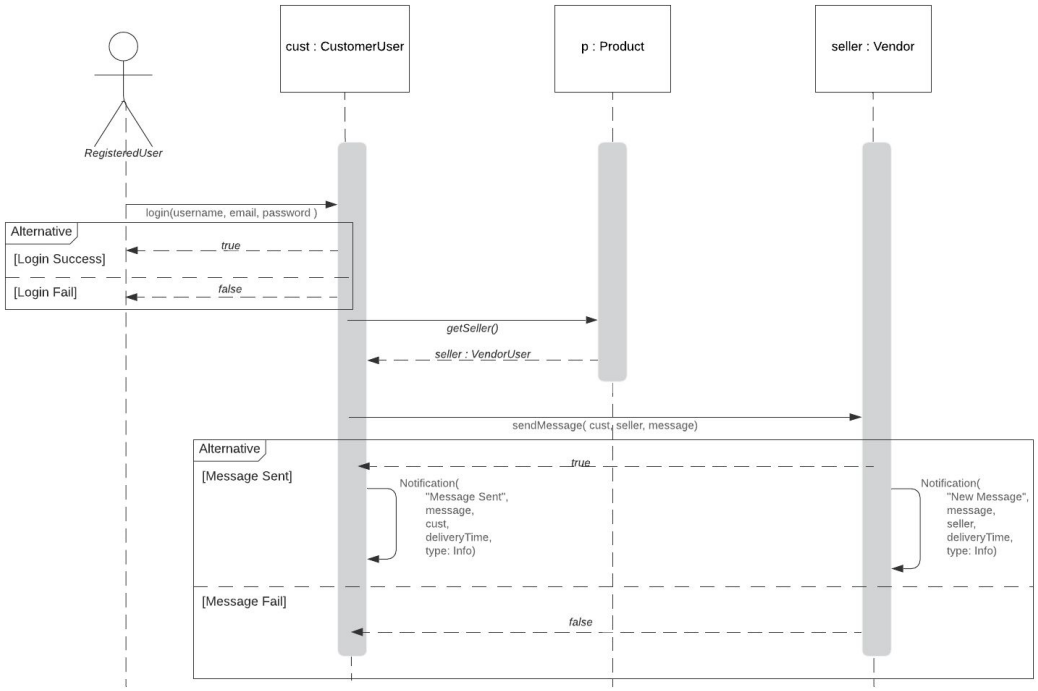
Return Product



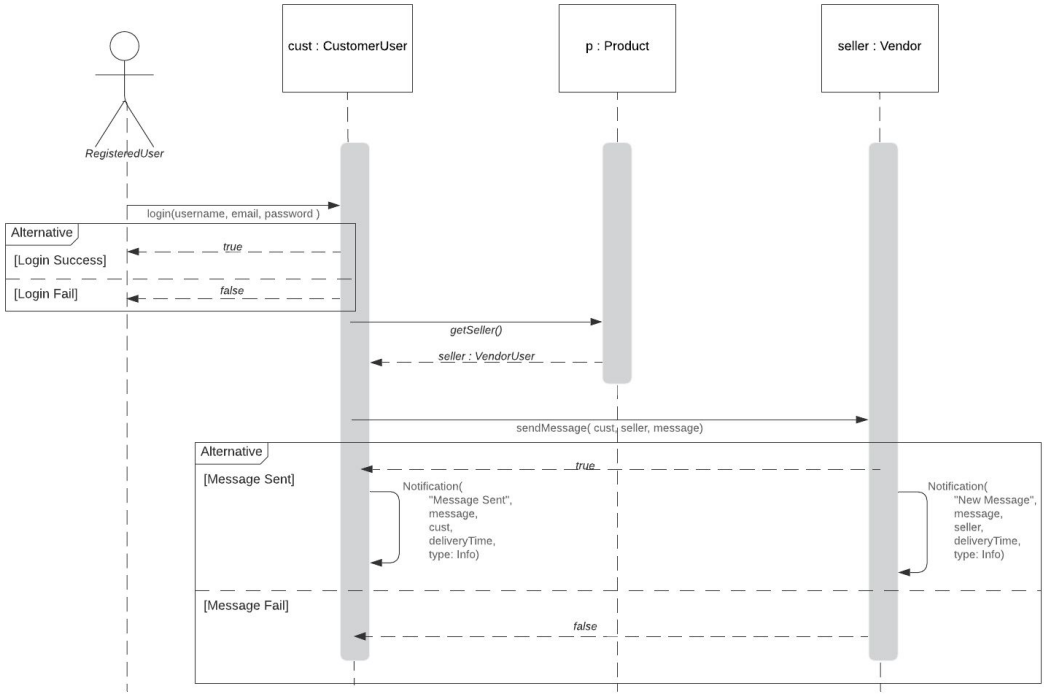
Add Product to Cart



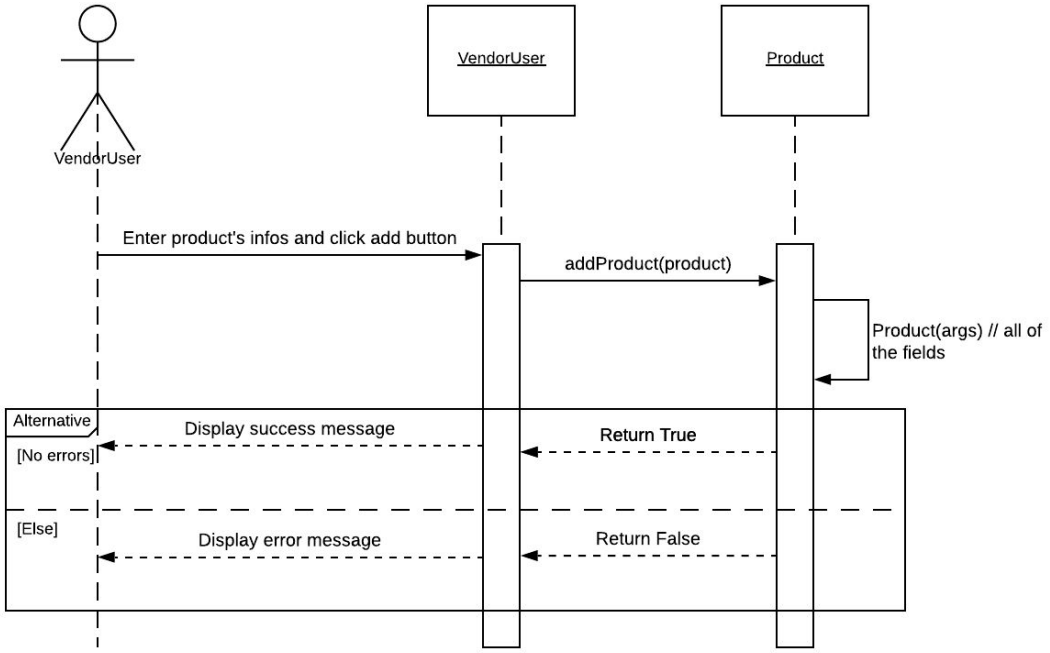
Message



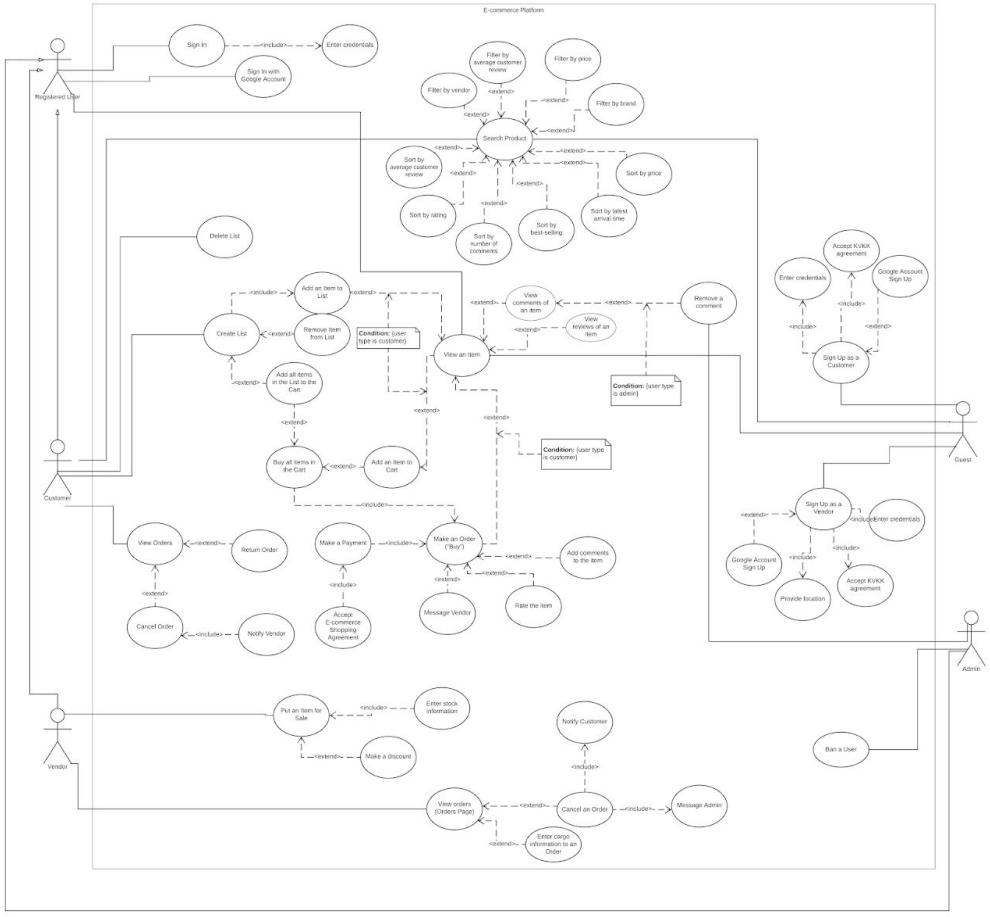
Remove Product from Cart



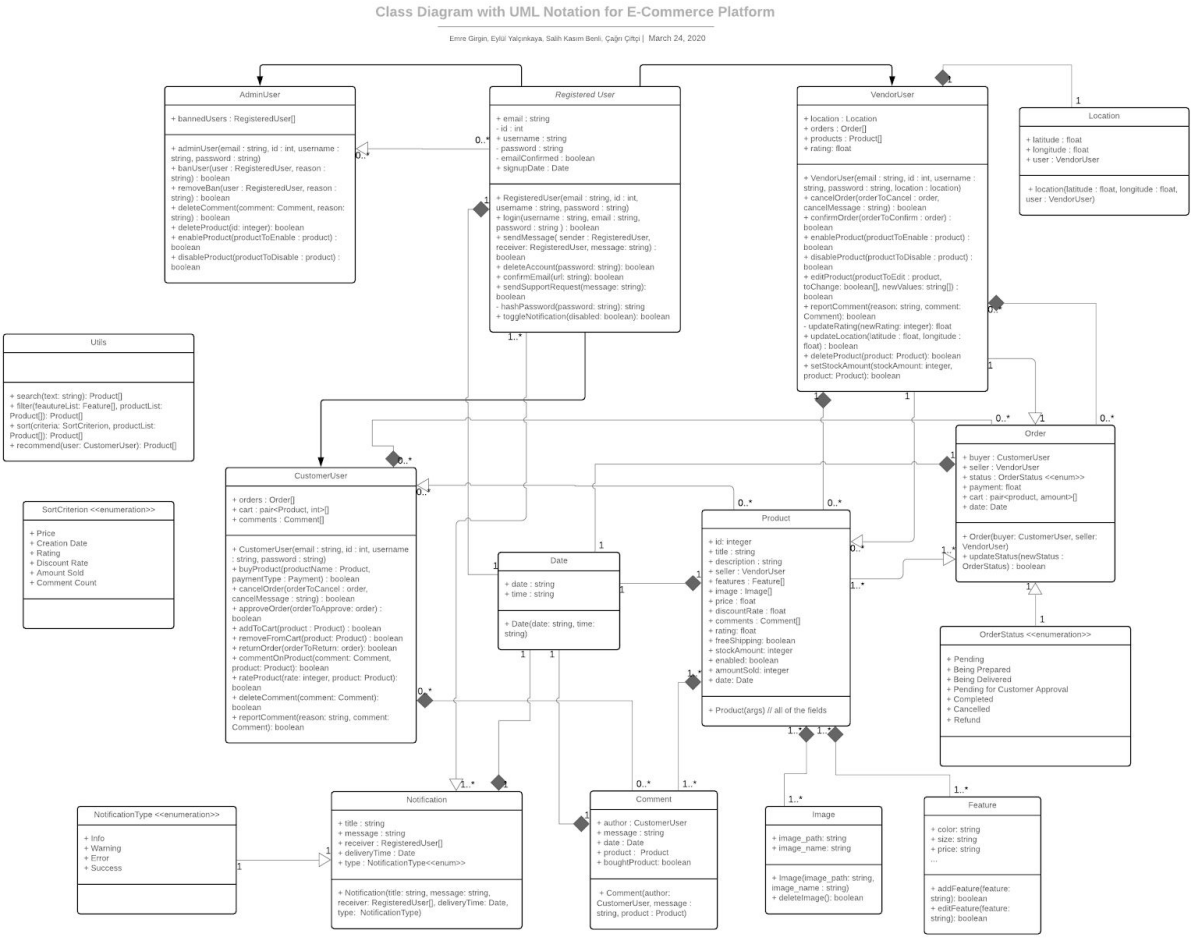
Vendor Adding Product



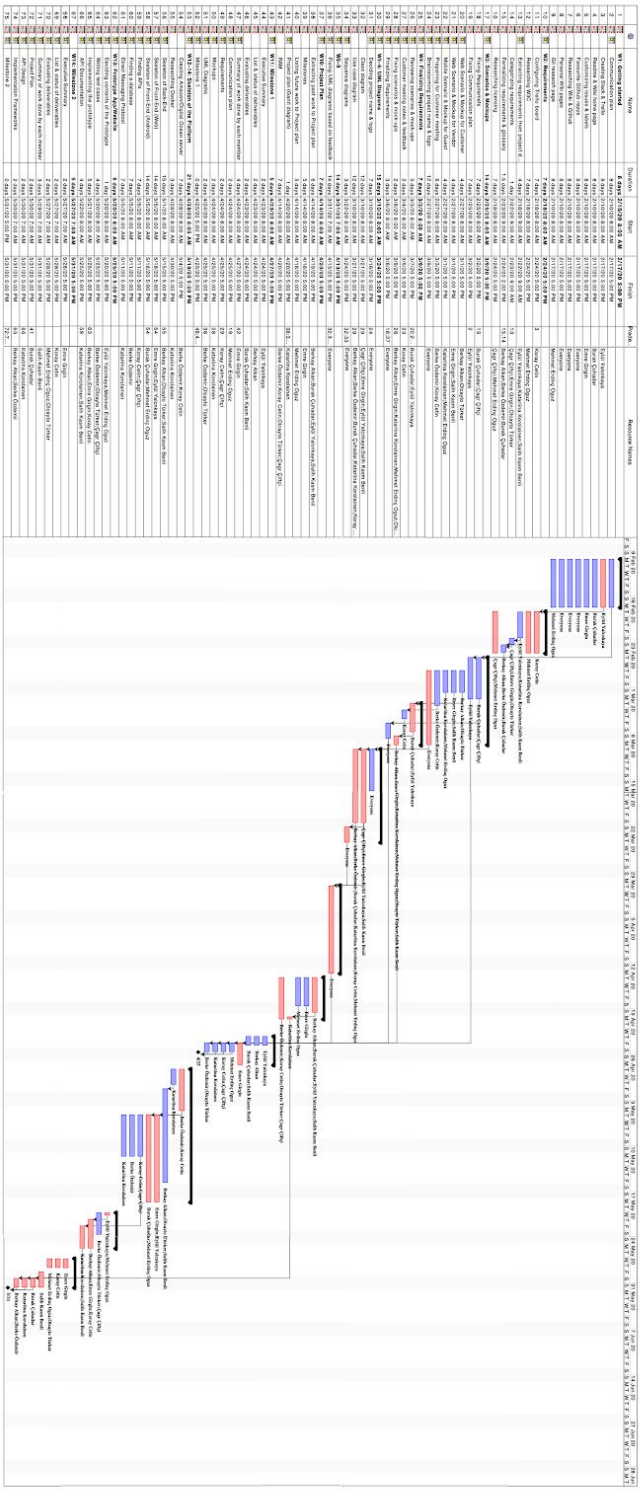
9.2 Use case diagram



9.3 Class diagram



10. Project Plan & RAM



Approval – approves the item as complete **L**ead – Final responsibility **S**econdary – Lead person's backup, else C and R **C**ontributor – physically contributes to item **R**eviewer – Only reviews **N**one	Burak Çuhadar	Mehmet Erdinç Oğuz	Koray Çetin	Eylül Yalçınkaya	Emre Girgin	Salih Kasım Benli	Berke Özdemir	Katariina Korolainen	Berkay Alkan	Çağrı Çiftçi	M. Olcayto Türker
W1: Getting started											
Creating Slack & Trello	N	A	S	L	R	N	N	N	N	N	N
Readme & Wiki home page	L	C	A	N	N	R	N	C	N	N	N
Customizing issues & labels	C	N	N	A	L	N	N	N	C	N	R
Favourite Github repos	C	C	C	C	C	C	C	C	C	C	C
Researching Wiki & Github	N	L	N	N	N	N	N	N	N	N	N
Personal Wiki pages	C	C	C	C	C	C	C	C	C	C	C
Git research page	N	L	N	N	N	N	N	N	N	N	N
W2: Requirements											
Configuring Trello board	N	N	L	N	S	R	N	N	N	N	N
Researching W3C	N	L	N	N	N	N	R	N	N	N	N
Extracting requirements from project description	R	A	R	C	R	C	R	L	R	R	R
Categorizing requirements	R	R	R	R	L	R	R	R	R	C	L
Formalizing requirements & glossary	C	R	R	R	R	R	L	R	C	R	R
Researching licensing	R	L	R	R	R	R	R	R	R	C	R
W3: Scenarios & Mockups											
Fixing Requiremets	L	R	R	R	R	R	R	R	R	C	R
Fixing Communication plan	N	N	N	L	N	N	N	N	N	N	N
Web Scenario & Mockup for Customer	R	R	R	R	R	R	R	R	L	R	C
Web Scenario & Mockup for Vendor	R	R	R	R	C	L	R	R	R	R	R
Mobile Scenario & Mockup for Guest	R	L	R	R	R	R	R	C	R	R	R
Prepairing for Customer meeting	N	N	L	N	N	N	C	N	N	N	N
Brainstorming project name & logo	C	C	C	C	C	C	C	C	C	C	C
W4: Finalizing Requirements											
Reviewing scenarios & mock-ups	C	N	N	C	N	N	N	N	N	N	N
Customer meeting notes & feedback	N	N	L	N	N	N	N	N	N	N	N
Fixing scenarios & mock-ups	N	C	N	N	C	C	N	N	C	N	C
Finalizing Requirements	C	C	C	C	C	C	C	C	C	C	C
W5–6: UML Diagrams											
Deciding project name & logo	C	C	C	C	C	C	C	C	C	C	C
Class diagram	R	R	R	C	C	C	R	R	R	C	R
Use case diagram	C	C	C	R	R	R	C	C	C	R	R
Sequence diagrams	C	R	C	C	C	C	C	C	C	C	C
W8–9											
Fixing UML diagrams based on feedback	C	C	S	C	C	C	R	C	A	C	L
W10: Project Plan											
Extracting past work to Project plan	C	N	N	C	N	C	N	N	C	N	N
Milestones	N	N	N	N	L	N	N	N	N	N	N
Listing future work to Project plan	N	L	N	N	N	N	N	N	N	N	N
Project plan (Gantt diagram)	N	N	N	N	N	N	N	L	N	N	N
RAM	L	N	N	N	N	N	N	N	N	N	N
W11: Milestone 1											
Executive Summary	R	R	A	L	R	R	S	R	R	R	R
List & status of deliverables	R	R	R	R	S	R	R	R	L	R	A
Evaluating deliverables	C	R	R	A	R	C	R	R	R	R	R
Summary of work done by each member	C	C	C	C	C	C	C	C	C	C	C
Communication plan	N	L	N	N	N	N	N	N	N	N	N
Requirements	N	N	C	N	N	N	N	N	N	C	N
Mockups	N	N	N	N	N	N	N	L	N	N	N
UML Diagrams	N	N	N	N	A	N	C	N	N	N	C
Milestone 1											
W12–14: Skeleton of the Platform											
Creating Amazon/Digital Ocean server	N	N	C	N	N	R	C	N	A	N	N
Researching Docker	R	R	R	R	R	R	R	L	R	R	R
Skeleton of Back-End	N	N	N	N	N	C	N	N	C	N	C
Skeleton of Front-End (Web)	N	N	N	C	C	N	N	N	N	N	N
Skeleton of Front-End (Android)	C	C	N	N	N	N	N	N	N	N	N
Finding APIs	N	N	C	N	N	N	N	N	N	C	N
Finding a database	N	N	N	N	N	N	L	N	N	N	N
Direct Messaging Protocol	N	N	N	N	N	N	N	C	N	N	N
W13: Prototype App/Website											
Deciding contents of the Prototype	N	N	C	C	N	N	N	N	N	N	N
Writing test cases	R	N	A	N	N	N	C	N	N	C	C
Implementing the prototype	N	R	C	N	C	N	N	A	C	N	N
API Documentation	N	N	N	N	N	C	N	C	N	N	N
W14: Milestone 2											
Executive Summary	N	N	N	N	L	N	N	N	N	N	N
List & status of deliverables	N	N	L	N	N	A	N	N	N	N	R
Evaluating deliverables	N	C	N	A	N	R	N	N	N	N	C
Summary of work done by each member	C	C	C	C	C	C	C	C	C	C	C
Project Plan	L	N	N	N	N	R	N	N	N	N	A
API Design	N	N	N	R	N	R	N	L	N	A	N
Implementation Frameworks	R	R	A	N	N	N	C	N	C	N	N
Milestone 2											