Bilkent University

Department of Computer Engineering

Senior Design Project

Project short-name: BidIt

Final Report

Efe Eroğlu, Muhammet Kamil Gök, Ahmet Serdar Gürbüz, Rumeysa Özaydın, Hasan Yıldırım

Supervisor: Uğur Doğrusöz
Jury Members: Çiğdem Gündüz-Demir and Can Alkan

Final Report
April 30, 2020

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfillment of the requirements of the Senior Design Project course CS491/2.

1.Introduction

Auction is a process of buying or selling products based on biddings. It is a good alternative for fixed-price selling mechanisms. The word comes from Latin word “augere” which means “to increase” [1]. One of the most common forms of auction is open ascending price auction where participants bid openly and every bid must be higher than the previous bid. The other form is the sealed-bid first-price auction where bidders submit their bids privately in sealed envelopes and the highest bidder wins [1]. The popular goods taking place in auctions are antiques and rare collectibles.

When it comes to online auctions, some conditions mentioned above change. The time is limited and the highest bidder at the end of the time buys the product. Also, nowadays, auctions do not have to be about antiques or rare objects. People can sell their second-handed products to get rid of them which makes it reasonable to have a marketplace in order to meet the demands of these people.

Online shopping is in demand these days. Either it is a brand new product or a second-hand product. We aim to introduce a bidding system for these online shopping platforms. Our platform, BidIt, will be an easy to use auction platform where users can buy and sell products online.

This report explains the proposed system in terms of requirements, constraints, high & low level design, social impact and user manual in detail.

2. Requirements

2.1 Functional Requirements

These following functions will be available for the users.

- Users can sell their products by indicating their caption, description and images. They will also initialize the start price. The expiration time of an advertisement will be indicated by the user and it can be at most 15 days.
- Users can bid on the auction to be eligible to acquire the item.
- Users can rate and comment on sellers.
- Users can monitor other seller's reviews.
- Users can search for an item by name or category.
- Users can add an auction to their favourites list.
- Users can browse the marketplace for the currently open sales.
- BidIt will notify buyers when someone out-bid their bid. Also, sellers will be notified when someone bids their product.
- Users will be notified when they win or lose the auction that they have bidded.
- Users can see the previously sold products of any user.
- Users will have an in-app wallet that they can deposit money via transferring money from their bank account to Bidit’s bank account.
- Users cannot bid more than their wallets’ balance.
- Sellers will receive the payment once the buyers approve the proper delivery of the product to protect buyers and sellers.
- Users will be able to withdraw money from their wallet as Bidit will transfer the amount to their bank account using their IBAN provided.

- BidIt will do the authentication of the users provided their email addresses and passwords.
- Users can update their account information.

2.2 Non-Functional Requirements

In the following subsections, the non-functional requirements are divided into subsections as reliability, usability, extensibility, compatibility and security.

2.2.1 Reliability

- The application must not randomly crash. It must be able to recover quickly from any errors that occur during application usage.
- The auctions should always be available for users, where the other users are able to interact with the releasing.
- There should not be any kind of data loss in case of any kind of error.

2.2.2 Usability

- The BidIt application should have a simple user-friendly interface that will allow all types of users to be able to easily adapt.

2.2.3 Extendibility

- Design of the application should be written following the Object Oriented Programming paradigms in order to add new functionalities easily.
- The storage in the system should be designed in a way that it can be extended in the necessity by upgrading the plan of the cloud systems.

2.2.4 Compatibility

- The application will be cross-platform and it will be adaptable for Android and iOS.

2.2.5 Security

- Personal data of the user such as name, surname, and phone number must be securely protected by the application.

- The money transaction between wallets must be atomic.
- Passwords are stored as hash instead of explicit value.

2.3 Pseudo Requirements

2.3.1 Language Constraints

- Our application will support Turkish language.

2.3.2 Economic Constraints

- The webpage is on the Github domain which is free to use.
- Github will be used for Version Control and code sharing which is free to use.
- Heroku is a platform as a service, where backend will be present and have a REST API.
- Free APIs will be used for purchasing service.
- Open-source libraries will be used for both frontend and backend.

2.3.3 Implementation Constraints

- The application will be working on mobile operating systems which are Android and IOS.
- 3-Tier Architecture model will be used as an application structure.
- React Native will be used in order to provide cross-platform application development.
- Version control will be sustained via Git and the source code will be hosted on Github.
- Object Oriented Design principles will be adapted in the design steps and appropriate design patterns will be used as necessary.

2.3.4 Social Constraints

- The application can be used by anyone who has a smartphone and an email.

2.3.5 Ethical Constraints

- The personal information of users will not be shared.
- Passwords will be encrypted.
- Code of Ethics will be followed.
- Any external software or library used will be properly referenced.

2.3.6 Timeline Constraints

Development of our product will be parallel with the following schedule [3],

- Project Specifications: **Monday, October 12, 2020**
- Analysis Report: **Monday, November 9, 2020**
- High-Level Design Report: **Monday, December 21, 2020**
- Low-Level Design Report: **Monday, February 8, 2021**
- Final Report: **Friday, April 30, 2021**
- Final Presentation: **Monday, May 3, 2021**

2.3.7 Sustainability Constraints

- We will closely follow and consider user feedback in order to improve the product.
- We enhance the product following the feedback from our first presentation.
- Bugs will be followed and updated from the initial release until the deadline.

2.3.8 Professional and Ethical Constraints

Code of Ethics proposed by National Society of Professional Engineers [3] will be followed during the implementation,

- Since the users have to sign up in order to sell or bid the products, we need to keep their usernames and passwords. In order to manage this in a safe manner, we are going to store the hash values of the passwords in our database. Also, the personal information of the users such as phone number, name, and surname will not be shared by other users or third parties without getting permission.

- The camera or storage permission will be needed when a buyer is uploading or taking pictures of the products. Before accessing the phone’s camera and storage, required permissions will be taken from users.

3. Final Architecture and Design Details

3.1 Overview

The system decomposition of BidIt aims to reflect the structures of its systems and subsystems clearly. The subsystem decomposition will show how our subsystems interact with each other. It is important to identify subsystems correctly before starting the implementation in order to get fewer errors and fewer revisions. 3-Tier Architecture is used for the subsystem decomposition and will be described in detail. Also, the mapping of the hardware and the software is explained. In addition, management of the persistent data, access and security, and boundary conditions are explained.

3.2 Subsystem Decomposition

BidIt has a classical 3-tier architecture style. It consists of Client Layer, Application

Layer, Data Layer. The decomposition of subsystems in each layer is illustrated in Figure 1.

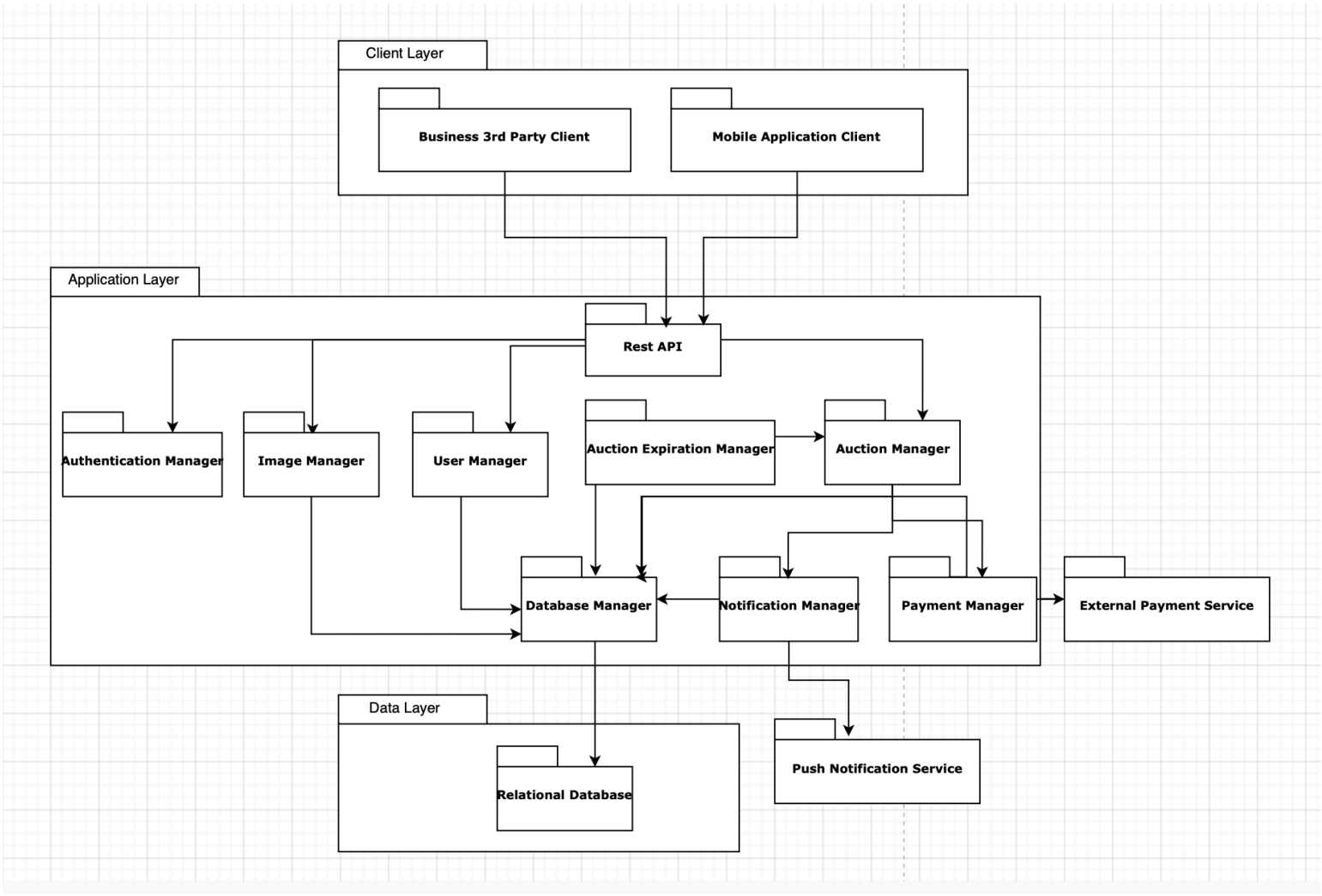

Figure 1. Subsystem Decomposition

3.3 Hardware/Software Mapping

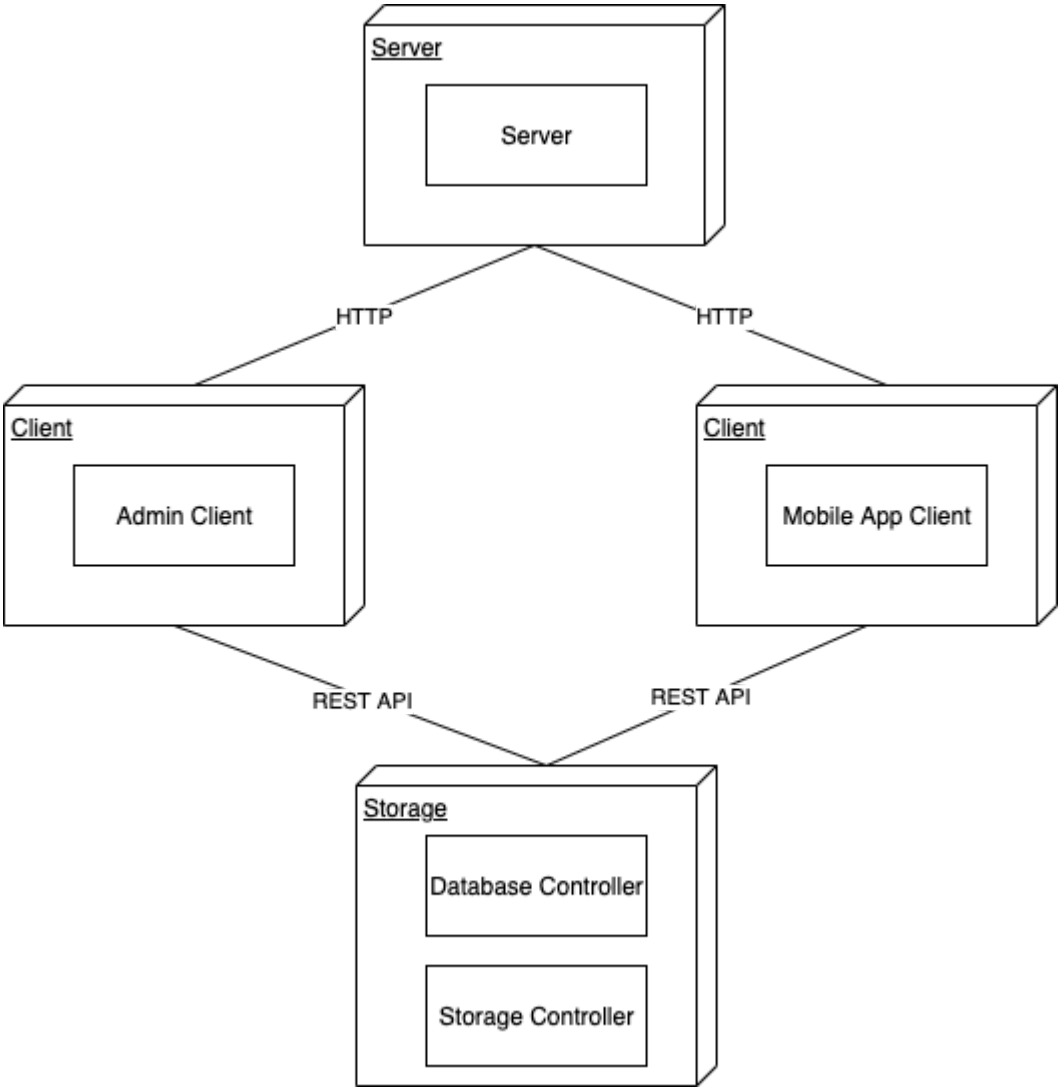

Figure 2. Hardware/Software Mapping

BidIt has two clients: business third party and mobile app user client. BidIt provides the service with HTTP requests from the server to show the app to users. The server side of the system performs the authorization according to the type of user who logs into the system.

On the other hand both apps use REST API to get information from both database and storage controllers. The server side can be seen as a one shared database where every action takes place by the system. All the collection holds in the database for the further API calls, where images and data are stored in shared storage.

3.4 Persistent Data Management

Shopping applications keep the issue of data management in a very important place because most of the application consists of huge image data and this requires a very high storage space.

- **Client Data:** Individual clients unique data consist of first name, last name, ClientID, email, password, contact number, token, rating count, rating sum.
- **Auction Data:** Individual auction unique data consist of auctionID , sellerID , name, start date ,description, expire date , start price, buy now price, winning bid, bid history, winning bid client ID.
- **Comment Data:** Individual comment unique data consist of commentID, authorID, author name, sellerID, content, rating.
- **Favourite Data:** Individual favourite unique data consist of userID, favouriteID, auctionID.
- **Bid Data:** Individual bid unique data consist of userID, productID, price.
- **Image Data:** Individual image unique data consist of imageID, image location, image name, image content(in byte array format).
- **Wallet Data:** Individual in-app wallet data consist of walletID, userID, balance
- **Transaction Data:** Individual transaction unique data consist of transactionID, transactionStatus, amount, senderID, receiverID

3.5 Access Control and Security

BidIt will keep various information of users such as name, username, address etc. These data will be kept in a database and controlled with secure systems. Security will be important in our wallet system. Encryption techniques will be used to provide security of data. For the passwords, hashing will be used.

3.6 Global Software Control

The software control in BidIt is based on the interactions of users. They will be able to sell and buy a product and comment on the products and interact with users. The payment of the products will be handled by system security. The procedure of the auction can be explained in three main parts:

- Uploading the product to the system and setting the initial price and the duration
- Getting various bids until the end time
- The user with the highest bid at the end of the time will purchase the product.

3.7 Boundary Conditions

BidIt will have three boundary conditions including initialization, termination and failure. Backend REST API will return the response code of 2xx to notify the successful operations. In case of a system error, the server returns the response code of 5xx. In case of a client error, the server returns the response code of 4xx.

3.7.1 Initialization

The user must have the application. The internet connection must be established to initialize the application.

3.7.2 Failure

In case of wrong password or username, an authorization error will be encountered. If there is low internet connection, network failures may occur.

3.7.3 Termination

Users can log out from BidIt any time. Closing the app is not enough to log out, the log out button should be pressed.

3.8 Interface Subsystem

3.8.1 Business 3rd Party Client

Some API endpoints will be shared with business third party clients for commercial purposes such as analytics, automatization etc.

3.8.2 Mobile Application Client

The mobile application client is the main medium that is used by the end-users. In the application there is only one client, user where both seller and buyer can be the same user. The user can use the application by seeking through auctions, bidding the auctions, adding to favourites, filtering the auctions, viewing seller profiles, editing/viewing their profiles, selling their items by initializing a new auction. These are all of the main activities that users can achieve by using a mobile application client.

3.9 Application Logic Subsystem

3.9.1 REST API Gateway

The gateway server which follows the REST API conventions will be a single entry point to the whole system which handles the requests coming from the client-side and redirecting to the related subsystems. Authentication, authorization, encryption/decryption and session management will be supervised by the gateway server. The server will be stateless meaning that it will not keep any data of users, products and auctions.

3.9.2 User Manager

This service is responsible for management of the actions that are defined for end-users such as editing the user profile, creating and updating favourites, retrieving seller profiles. To fulfill these tasks, it will also communicate with the database manager and image service to update the database accordingly.

3.9.3 Image Manager

The image manager is responsible for management of the images where used in auctions and profiles. Image manager is authorized by REST API and connected to the data layer the postgre sql where both names and image are stored in the data layer.

3.9.4 Auction Expiration Manager

This subsystem will handle the execution of the auctions. This manager provides the execution of the auctions when the time is on to execute for the auction. The auction expiration manager is not reachable from the REST API where the execution basis on the natural habitat of the auctions.

3.9.5 Payment Manager

This subsystem will provide users the option to store credits in their account to purchase new auctions. Users will have credits stored in their profile which they use as a related amount that when they bid in auctions, the amount will be in the provision. When the user does not achieve to win the auction as get overbid, the amount will be no longer in provision and directly available in the profile again.

3.9.6 Auction Manager

This part is the most critical component of the whole application since it regulates the core functionality, which is the logic of the auction mechanism. The creation of new auctions and bidding requests will be organized and managed within this module by using predefined constraints and rules. Moreover, the subsystem will inform the “Notification Manager”

about the updates and the current state of the auction. On the other hand, the auction manager should be able to handle many concurrent auctions as close to real time as possible in a fair way. Lastly, the Auction Manager will update the database for each unique handled event through the Database Manager.

3.9.7 Notification Manager

The notification manager will send notifications via the push notification service to inform the user in a case where a new bid is placed, an auction ends. The manager uses the system to save and push the notifications to the database, where the system proceeds with getting the notification from the database whenever the notification is created from another user's interaction.

3.9.8 Database Manager

The Database Manager will act as a gateway, standing in front of the relational database and the object storage to process data retrieval/update requests that are coming from other subsystems.

3.10 Data Subsystem

3.10.1 Relational Database

The relational database provides a database to store the data that has been sent from the database manager. All the details of the stored data have already been discussed in section 3.4 .

4. Development/Implementation Details

In BidIt, where the application is a cross platform mobile application, we use the Android Studio IDE and Android emulator of the Android Studio. As a framework, we used React

Native and Spring boot. We made use of the Spring boot while using Java and React Native while using JavaScript.

The BidIt application is an online platform where the application should be software as a service. Hence we use Heroku as a platform as a service to maintain the sustainability of our application. We use the extension of Heroku Postgres to store our database.

We use two services for the application, Push Notification Service and External Payment Service. Push Notification Service is provided by expo where we use the notification extension of the expo. What is planned to be provided is YapiKredi developer API as an external payment service, where they provide us an API to transfer the money to the virtual wallet of the user and vice versa.

Furthermore, we have used many tutorials and discussions to solve the problems we have encountered. StackOverflow, Udemy, etc. are the ones that we have more frequently used.

5. Testing Details

We have always tested our backend by manual function testing. In each update in the backend, we challenge the code with some test by using Swagger, an online platform to describe the structure of the API. Swagger also analyzes HTTP headers request bodies, content type and responses. We have also used the IDE debugging tool to debug our code. We choose manual testing where it provides us an easier convention for realising our mistakes step by step.

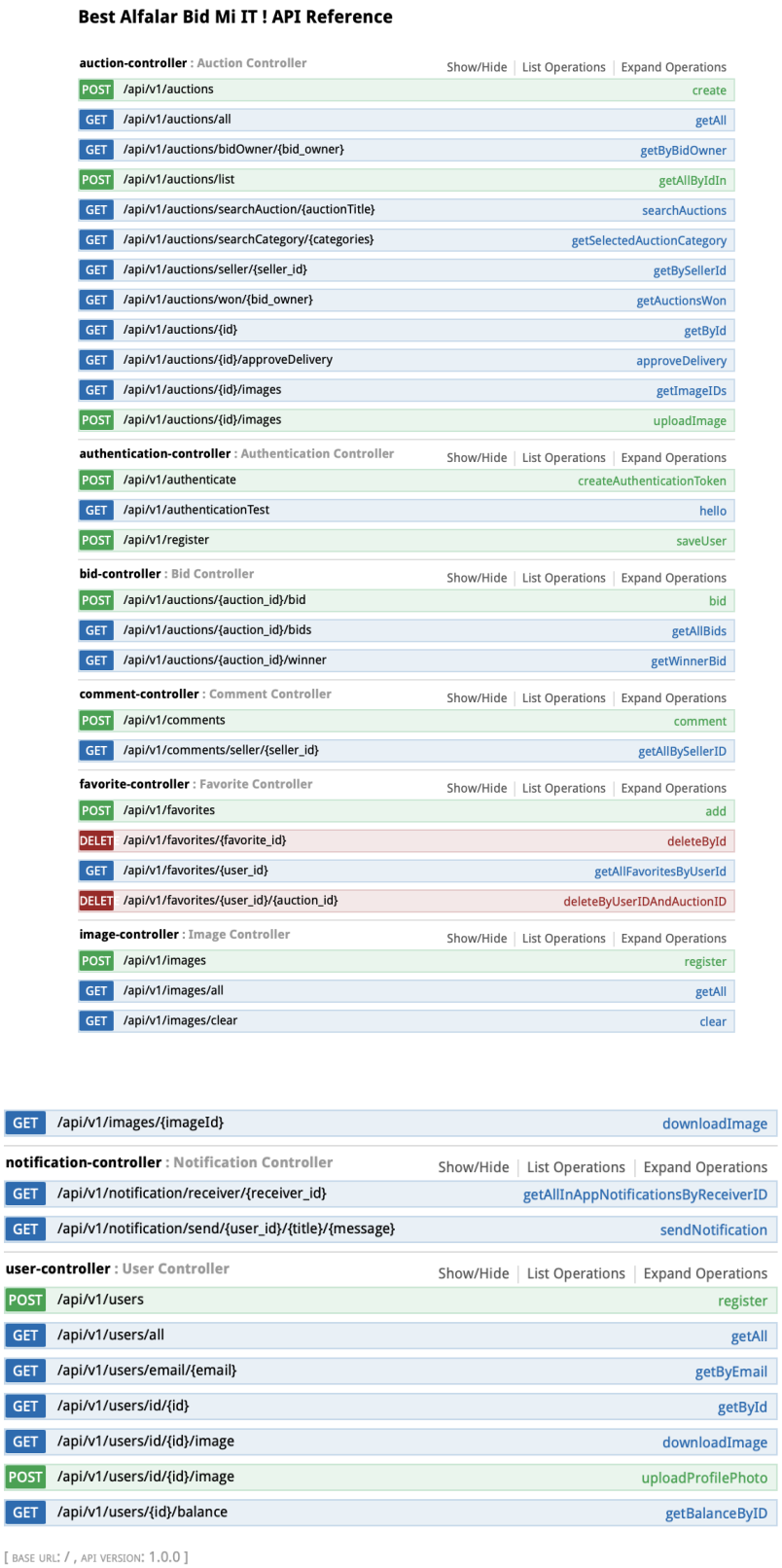

Figure 3. Example of Swagger

For the user interface, we have tested the code whenever a change is made in the code. After coding, we monitor the changes in the emulator to further debugging or changes. We have also checked the integration of the mobile client usability after finishing integrating the REST API to the UI, where we challenge all the usability of the user.

6. Maintenance Plan and Details

We plan our project as a SAAS, where the application is basically a cloud application. Hence we need to use online platforms to make sure our project is stable and running continuously. We also use the PostgreSQL extension of Heroku for the database. Therefore the connection limits, data limits are strict for free users as this is instantaneous although it can be scaled up easily when there is an increase in the amount of active users. We have designed and implemented the project for several months, thus there would not be any huge fatal errors that force us to create a downfall in the application. Therefore, the usual maintenance might occur once in a month for a not common hours, where the upgrade in the plans and minor errors might be checked.

For the upcoming stage of the project, we are aiming to launch the application for the Google Play Store and App Store. After getting the reflection for the application, we may consider to make some improvements casually to obtain a final product.

7. Other Project Elements

7.1. Consideration of Various Factors

7.1.1. Public Health

BidIt is an online application, where the screen time of users might increase due to the amount of the competitiveness in the auctions. Although, we have implemented the notification manager to aim this screen time to get drop where users do not constantly in necessity to check the application.

7.1.2. Public Welfare

Auctions are aiming to increase the amount of sales for the used products, where this is advantageous for the community to increase the possibility to access products cheaper where this will benefit the public economic standing.

7.1.3. Global Factors

BidIt is mostly designed for Turkey local, where the application should be assessed again to change the language of the application and availability payment service.

7.1.4. Cultural Factors

BidIt is a useful application for solving cultural solidarity as well, where the application might be extended to buy aid for the person in need for cheaper than the retail prices. BidIt will increase the amount of people that will be aided in the same amount of money.

7.1.5. Social Factors

BidIt will increase the interaction of the community and strengthen the social bonds. It will also increase the savings of the people which can be used in further solidarity.

7.1.6. Environmental Factors

BidIt orient the users for used items instead of brand new, where it would decrease the production line which would be extremely beneficial for the environment.

7.1.7. Economic Factors

BidIt will increase the purchasing power of people, where economically it will be beneficial for both sellers and buyers.

In the table below, we indicate the significance of these factors from 0 to 10 which higher points reflect higher significance.

	Effect	Effect Level
Public Health	2	Increase the screen time but modeled to be minimum
Public Welfare	5	Decrease the product and spare some cash
Global Factors	1	App language is Turkish
Cultural Factors	4	Improve the chance of solidarity
Social Factors	4	Interaction in the community
Environmental Factors	6	Decrease in the production line
Economic Factors	10	Cheaper options for purchasing and gaining extra cash for spare items

Table 1. Factors that can effect analysis and design

7.2. Ethics and Professional Responsibilities

BidIt is a social platform for increasing the effectiveness of spare items by selling to gain some extra cash and acquiring items that you are looking forward to less than the retail price. This will increase the useability of the items and decrease the mass production problems in the environment. For achieving the fair auction environment, each user should give a debt of more or equal than the bid price which will eliminate the risk of the bad

intended users. Also all the personal information in the database stored as hash values to secure the information.

Addition to this, we will also notify the users about the upcoming legal constraints to continue to use the application environment.

7.3. Judgements and Impacts to Various Contexts

The numbers in the table below are out of 10 where 0 is the bare minimum and 10 is the max score.

	Impact Level	Impact
Impact in Global Context	1	Language and payment is a barrier
Impact in Economic Context	9	Great option for retail prices
Impact in Environmental Context	6	Good for decreasing the mass production
Impact in Social Context	3	Mediocre for increasing the unity of the community

Table 2: Impacts of this judgement

7.4. Teamwork and Peer Contribution

Both semesters in Bilkent University was occured in the pandemic lockdown hence we only met up virtually. We meet up regularly once a week and try to peer code. We also double check the implementations by other group members.

- Ahmet Serdar Gürbüz:

    1. Established the organization of the project using GitHub issues and branching.
    2. Backend and database deployment to Heroku.
    3. Auction, Wallet, Bidding logic, Favorite, automatic Auction expiration, in-app notification implementation with their API endpoints.
    4. Constructing the fundamental backend structure.
    5. Frontend implementation support
    6. Frontend and backend testing
    7. Logo and color design.
    8. Writing reports

- Rumeysa Ozaydin:

    1. Constructing the fundamental frontend structure.
    2. Screens and UI components implementation
    3. Updating project website.
    4. Writing reports.

- Efe Eroğlu:

    1. Push Notification, Auction Category, Search Auction, Image Service, Global Exception Handler implementation
    2. Installing Swagger, testing the backend
    3. Constructing the backend structure
    4. Writing reports.

- Muhammet Kamil Gok:

1. Frontend implementation support.
2. Wallet, Transaction, Authentication, in-app Notification implementation.
3. Backend implementation support.
4. Writing reports.

- Hasan Yildirim:

1. Notification manager and configure the service.
2. Writing reports.

7.5. Project Plan Observed and Objectives Met

The project met most of the exceptions, only minor changes from the analysis report have been made. The biggest changes we made are that the project is not bilingual, where only Turkish language is available at the moment. Second one is the search engine is not detailed as we discussed previously where it does not return the related auctions that it waits for the exact wording.

7.6. New Knowledge Acquired and Learning Strategies Used

While doing the project, we have used Github efficiently for having version control, peer reviewing, etc. The platform seems familiar for all of the members since we started the project. We have already need to learn a lot about deployment of the project, spring boot framework where the application is not locally running.

Because of having the UI components, we try to reach our limits for having a user friendly and neat looking interface, which concludes that we need to practice on React Native.

Lastly, we try to finish the product fully till the demo day like in the industry. It teaches us the obligation, necessity and leadership skills.

8. Conclusion and Future Work

To sum up, we try to fill the missing application that we taught can be useful for many of us. It is important to evaluate the unused products, and we try to emphasize on that.

For the future, we are eager to publicly release the project where all the functional requirements are met with the application. Language improvements, universal payment service and legal regularization could be made to shift the application to the product.

9. Users Manual

In the below we are giving the UI of the application with short descriptions and usage.

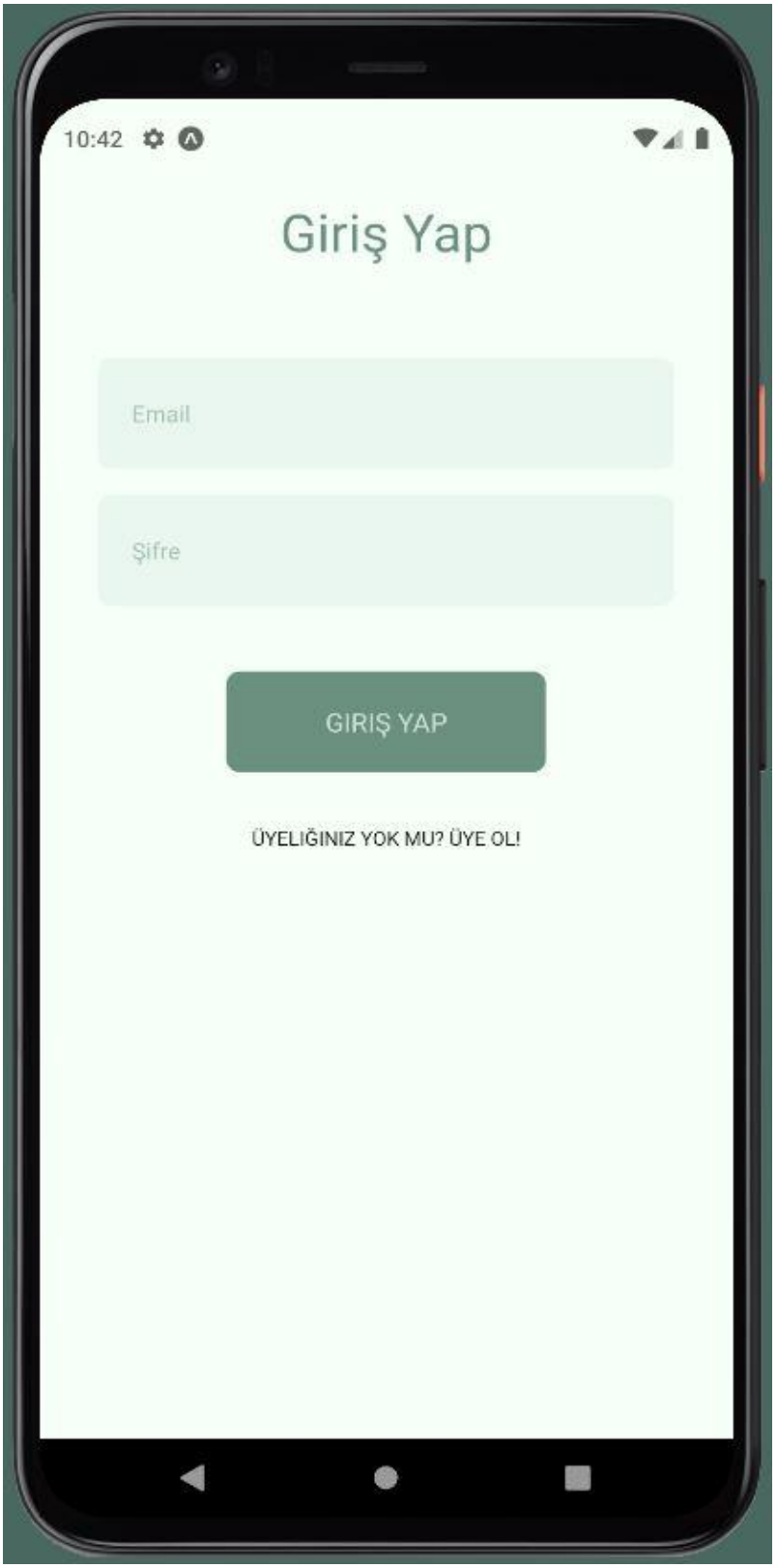

Figure 4 :Login Screen

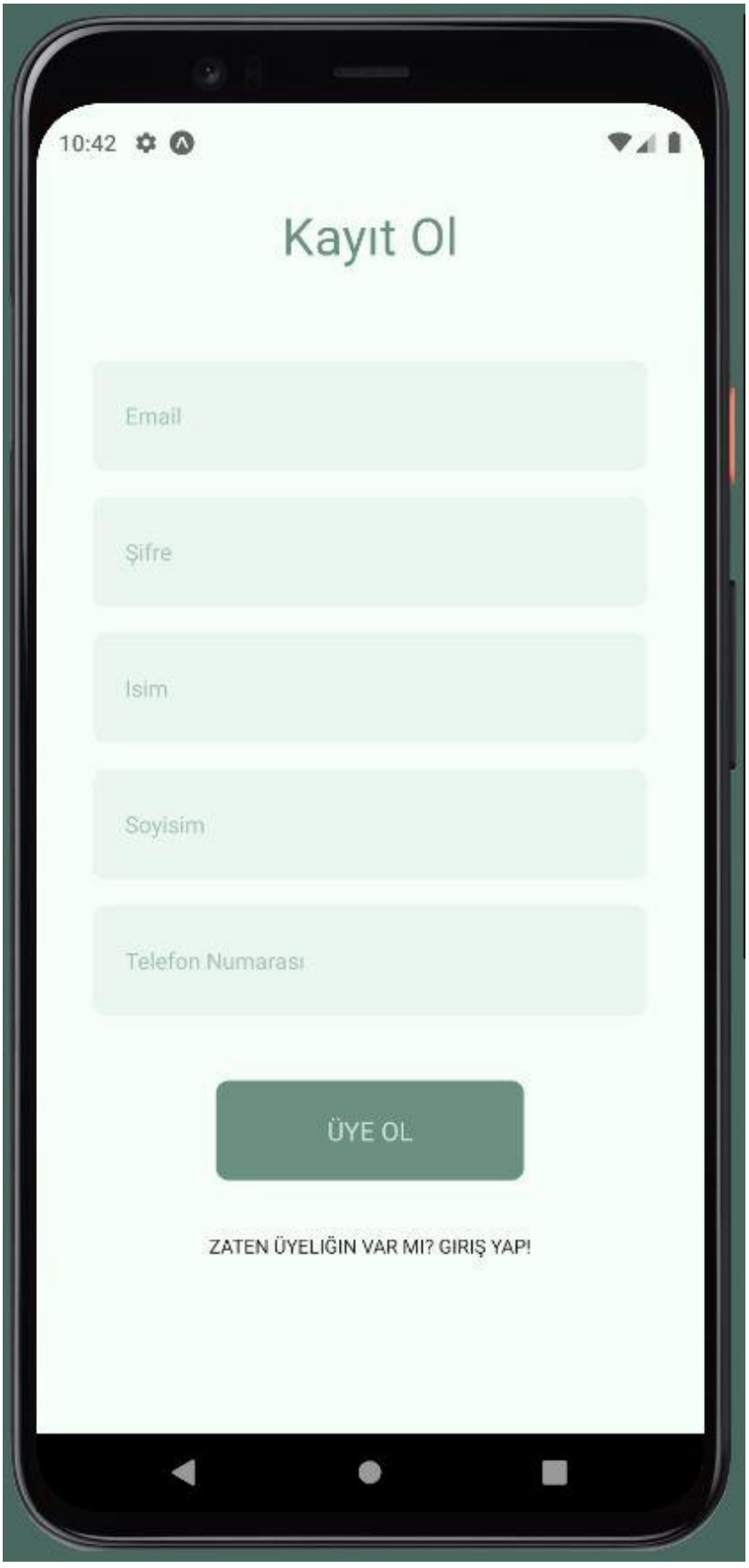

Figure 5: Sign In Screen

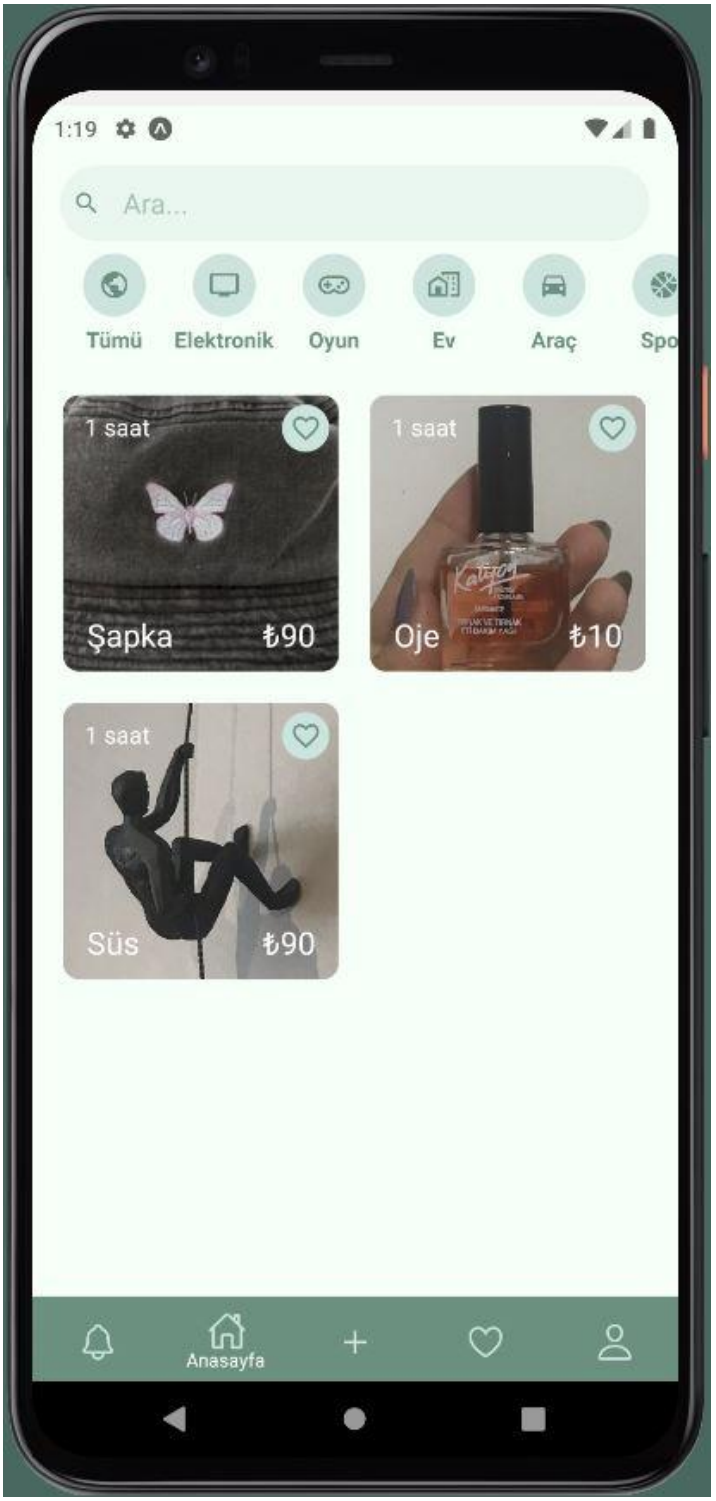

Figure 6: Home Screen

Through this interface, users can see the active auctions and make customized search via search bar and category buttons. They can add auctions to their favorites list by clicking the heart button. Also, users can navigate to the auction detail page by clicking the image of the auction.

Figure 7: Auction Detail Screen

Throughout this interface, users can see the details of the auction bid for it. Also, this page changes according to the user. For the owner of the auction, there is no “Teklif Ver” button. Instead, they have a button to end the auction.

Figure 8: User Detail

Throughout this interface, users can make comments about sellers and see the previous comments.Also, they can see all the auctions of this seller by clicking “Ilanları Gör”.

Figure 9: User Auctions Screen

Throughout this interface, users can see all the auctions of a seller. Also, they can navigate to the comments page of the seller by clicking “Yorumlari Gör”.

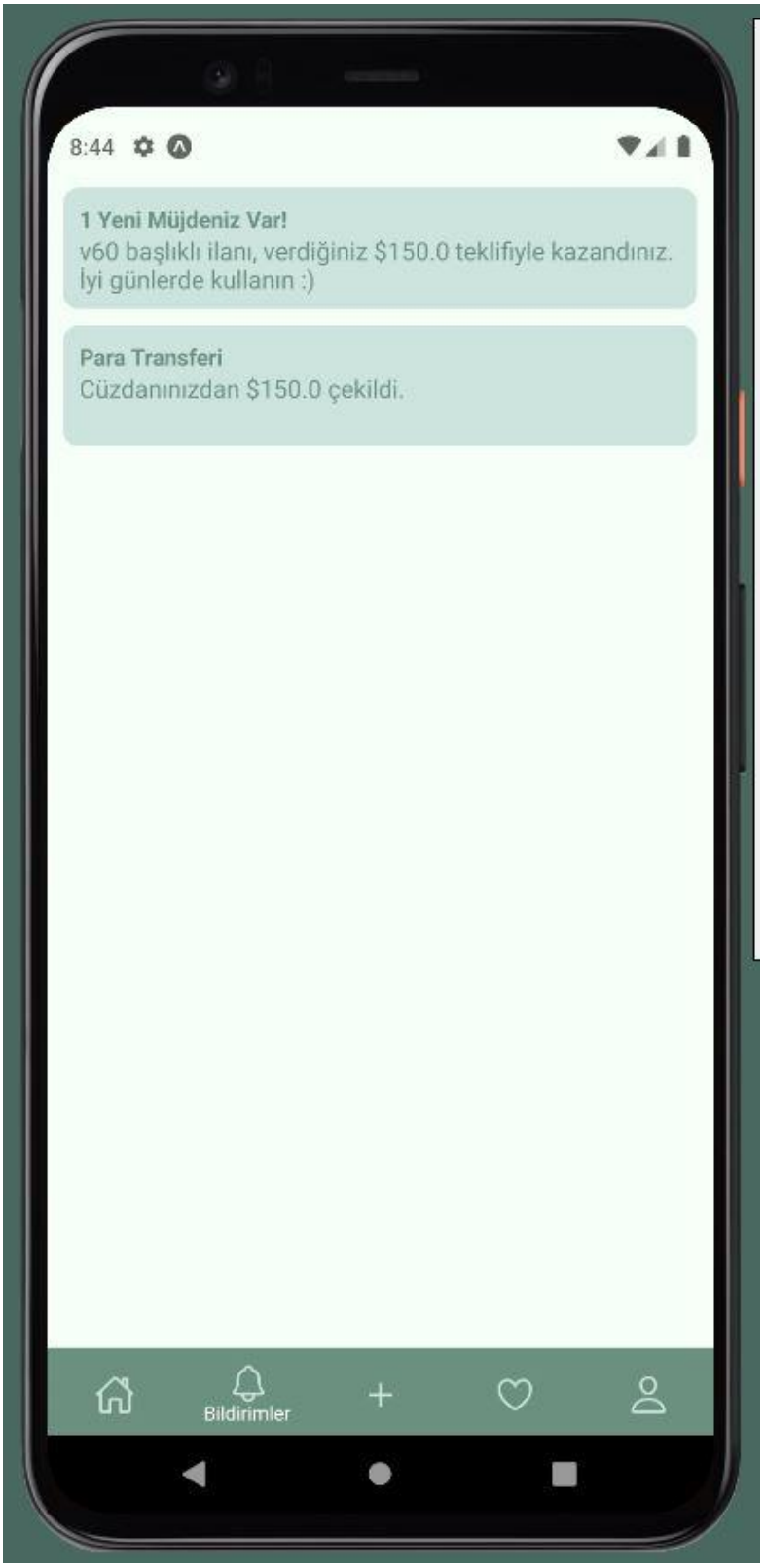

Figure 10 : Notification Screen

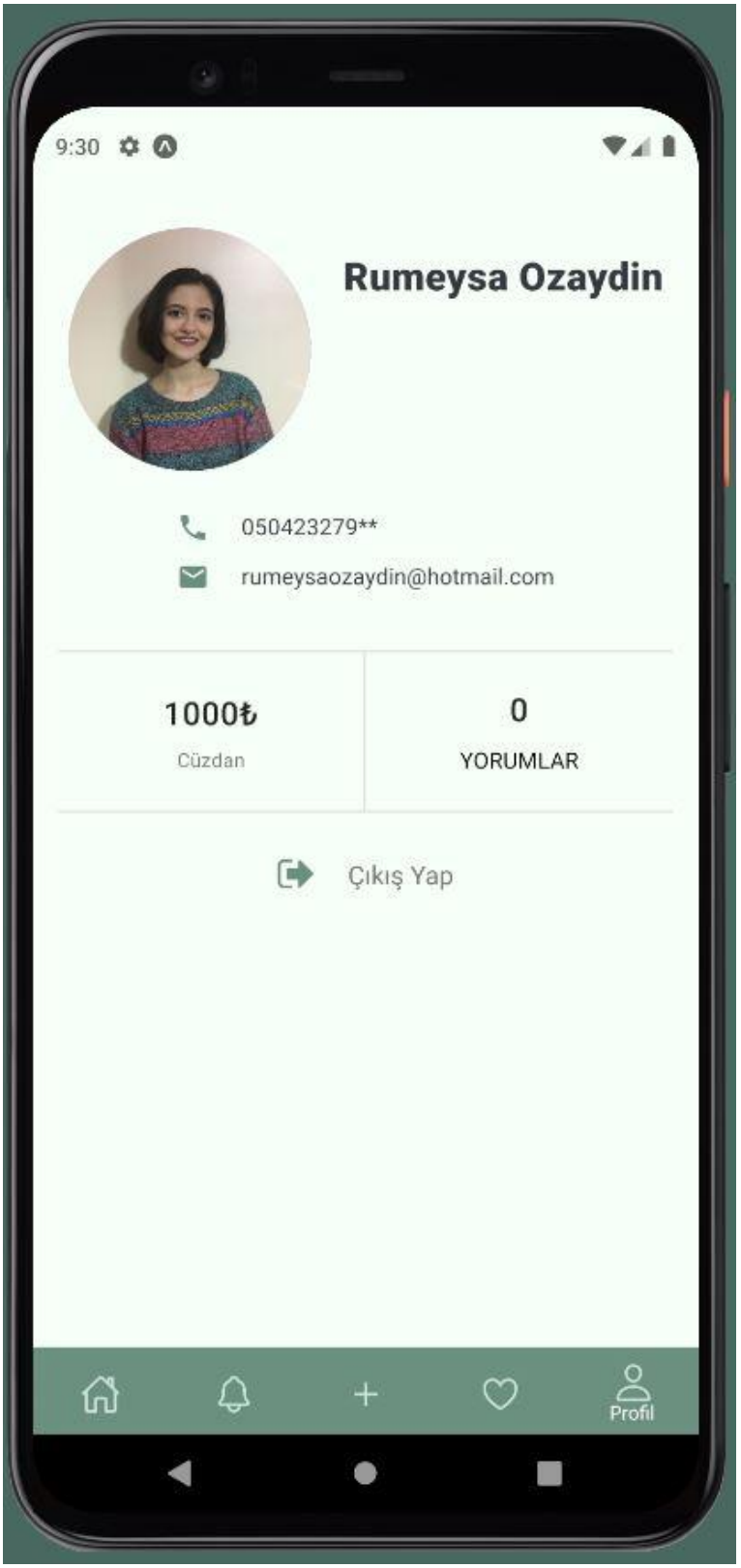

Figure 11: Profile Screen

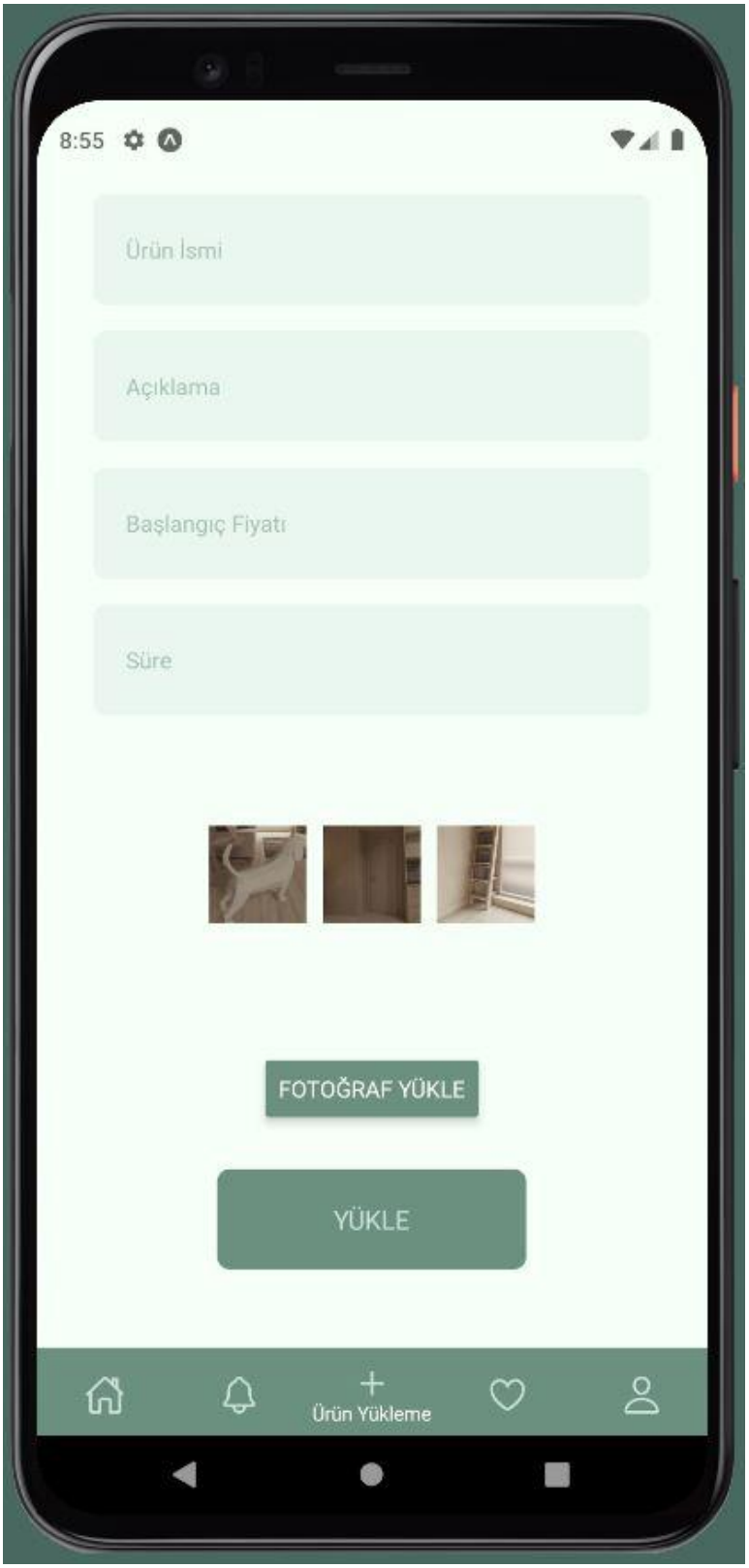

Figure 12: Auction Upload Page

Throughout this interface, users can upload new auctions. If successful, they will be directed to the auction detail page.

Figure 13 : Auction Lists Screen

Throughout this interface, users can see their favorites list, their own auctions, the auctions they have bidded and the auctions they won. They can navigate though this lists by clicking their name on the top of the page.

6. References

[1] V. Krishna, *Auction theory*. Amsterdam: Elsevier, 2010.

[2] “Code of Ethics,” Code of Ethics | National Society of Professional Engineers. [Online]. Available: https://www.nspe.org/resources/ethics/code-ethics. [Accessed: 10-Oct-2019].

[3] *CS491-2 Senior Design Project I-II*. [Online]. Available: http://www.cs.bilkent.edu.tr/~cs4912/current/. [Accessed: 11-Oct-2020].