



Bilkent University

Department of Computer Engineering

Senior Design Project

Project short-name: Bidlt

High Level Design Report

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

Supervisor: Uđur Doğrusöz

Jury Members: Çiđdem Gündüz-Demir and Can Alkan

December 27, 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	3
1.1 Purpose of the System	3
1.2 Design Goals	4
1.2.1 Usability	4
1.2.2 Robustness	4
1.2.3 Extendibility	4
1.2.4 Security	4
1.2.5 Maintainability	4
1.3 Definitions, Acronyms, and Abbreviations	5
1.4 Overview	5
2.Current Software Architecture	6
3.Proposed Software Architecture	6
3.1 Overview	6
3.2 Subsystem Decomposition	6
3.3 Hardware/Software Mapping	8
3.4 Persistent Data Management	9
3.5 Access Control and Security	9
3.6 Global Software Control	9
3.7 Boundary Conditions	10
3.7.1 Initialization	10
3.7.2 Failure	10
3.7.3 Termination	10
4. Subsystem Services	10
4.1 Client Layer	10
4.1.1 Admin Client	10
4.1.2 Mobile Application Client	11
4.2 Application Layer	11
4.2.1 REST API Gateway	11
4.2.2 User Manager	11
4.2.3 Search Engine	11
4.2.4 Product Manager	12
4.2.5 Payment Manager	12
4.2.6 Auction Manager	12
4.2.7 Notification Manager	12
4.2.8 Database Manager	13
4.3 Storage Layer	13
4.3.1 Relational Database	13
4.3.2 Storage Bucket	13
5. New Knowledge Acquired and Learning Strategies Used	13
6. References	14

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. As long as there are at least two bidders, the auction continues. 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 very 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 details of the proposed system in terms of System Overview, Requirements, Constraints, and Similar Products to further explain the purpose and the innovativeness of the project.

1.1 Purpose of the System

BidIt is an application that is designed to make online auctions easier. It will also be a good alternative to regular second-hand shopping. The audience for online shopping is gradually growing these days, and bidding will be an interesting and maybe cheaper way of buying things. BidIt aims to cover the needs of customers and provide them an easy to use auction platform.

1.2 Design Goals

1.2.1 Usability

The target audience is everyone regarding their age. All users should be able to understand how to perform their intended action without reading any manual. The user interface is planned to be user-friendly and guide users and newcomers throughout the process.

1.2.2 Robustness

The application must not randomly crash. It must be able to recover quickly from any errors that occur during application usage. Also, it will be designed to take care of any common errors that could occur.

1.2.3 Extensibility

The design of the application will be done in a way that the system is divided into proper sub-systems with low coupled and highly cohesive classes and packages. By doing so, our code will be more understandable and extendable. Also, the storage in the system will be designed in a way that it can be extended when needed.

1.2.4 Security

BidIt will ask for permission to access the camera when users want to upload a picture of the product. The personal data of the user such as name, surname, phone number, and address will be protected by the application to maintain security. Also, money transactions will be covered with an API. In addition, no user data will be shared with any third parties or saved without consent.

1.2.5 Maintainability

To address maintainability, BidIt needs modularity. The design will be done in a way that the system is divided into proper sub-systems and when one of these subsystems undergoes any changes or updates, other subsystems should not be affected.

1.3 Definitions, Acronyms, and Abbreviations

API: Application programming interface is an interface for the interaction between two systems.

UI: User Interface

Client: Part of the system that the user interacts

Server: Parts of the system that handles accessing a resource.

3NF: A design approach for database schemas to reduce duplication of data, avoid data anomalies, and simply manage the data.

BCNF: It is a normal form used for normalization for databases to manage databases properly which is a stronger version of 3NF.

1.4 Overview

BidIt aims to be an easy-to-use auction platform where users can sell their products. It will be a mobile application that is cross-platform adaptable for both Android and iOS. We will have two different use cases for users which give them an option to buy and sell. For selling items, users will define their items by writing a caption and adding images. For categorization, the best-suited category will be suggested by using the image and the description of the item but the user can manually change the category specification later. Users can select the time for their advert by determining the publish time and the length of the advert. Users are not allowed to remove the product from the marketplace after it got a bid or bought. Users will define an initial start price where other users can bid to buy this item and the highest bidder at the end of the auction can buy the item. Also, users can optionally define a buy now price where other users can buy the item instantaneously. When publishing an item, the platform will suggest optional pricing for the item from the description. This platform will automatically take the payment from the auction winner, and after the consolation of both buyers and sellers for the transaction, the seller will receive the payment. Buyers can review the sellers. Users can not bid on their own adverts in any manner. Users will get notifications about the auction dates, prices, etc. about their interested items and subscriptions. When a user is out-bid, the user will be informed with a notification instantly.

We will use the Client-Server model for the structure of the application. React Native will be used to provide a cross-platform application.

2.Current Software Architecture

There are numerous softwares that focus on e-commerce such as Amazon[2], eBay [3], Alibaba [4], etc. Although there is no such software that has the support of online live auction or price estimation. There are several applications for the used cars and items estimators although they do not support mobile platforms. Above that, all of these e-commerce companies use your personal data and force sellers to either pay a selling percentile or seller subscription fee. Hence these softwares have accessibility issues for the non-usual sellers.

3.Proposed Software Architecture

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.

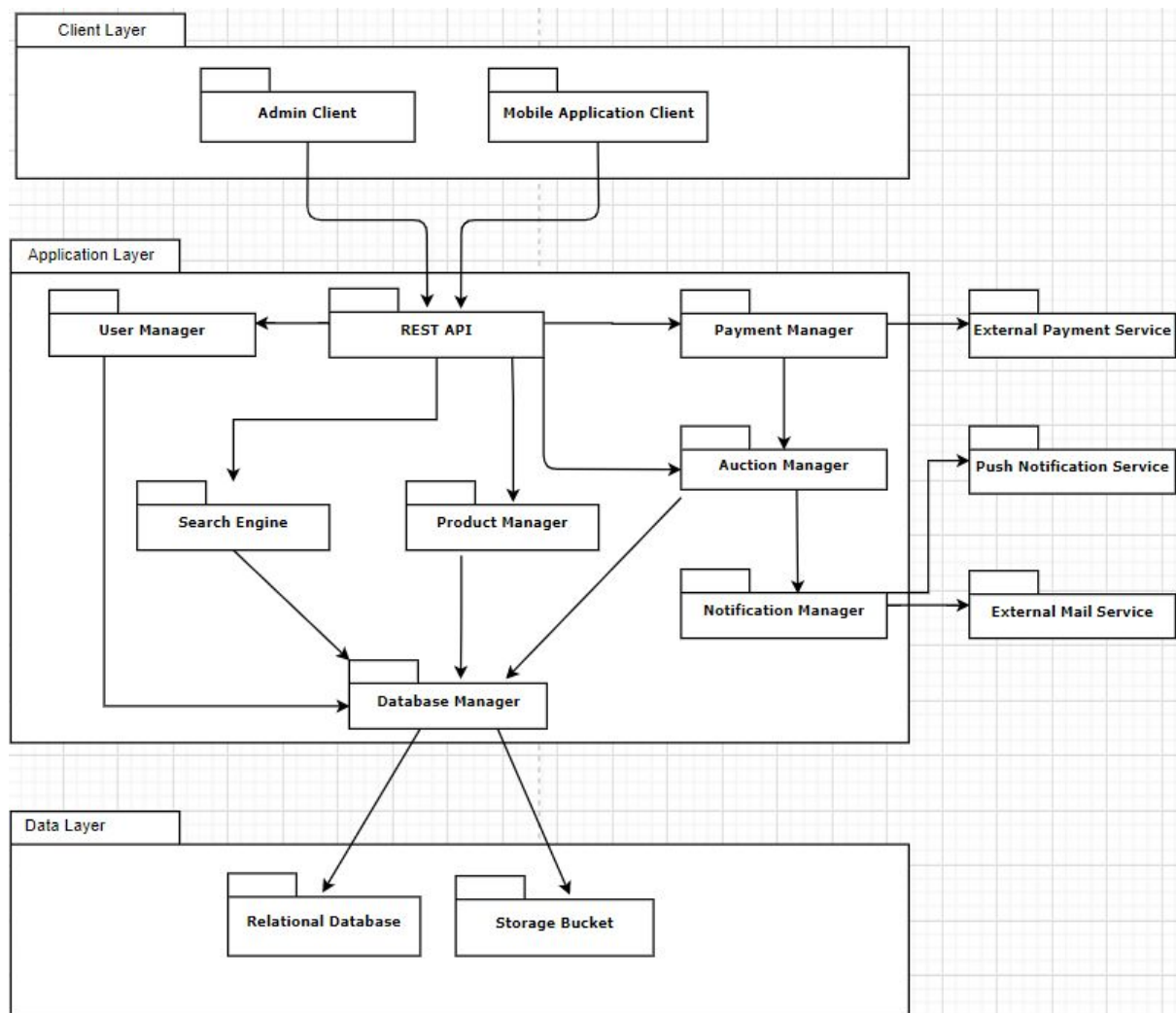


Figure 1. Subsystem Decomposition

3.3 Hardware/Software Mapping

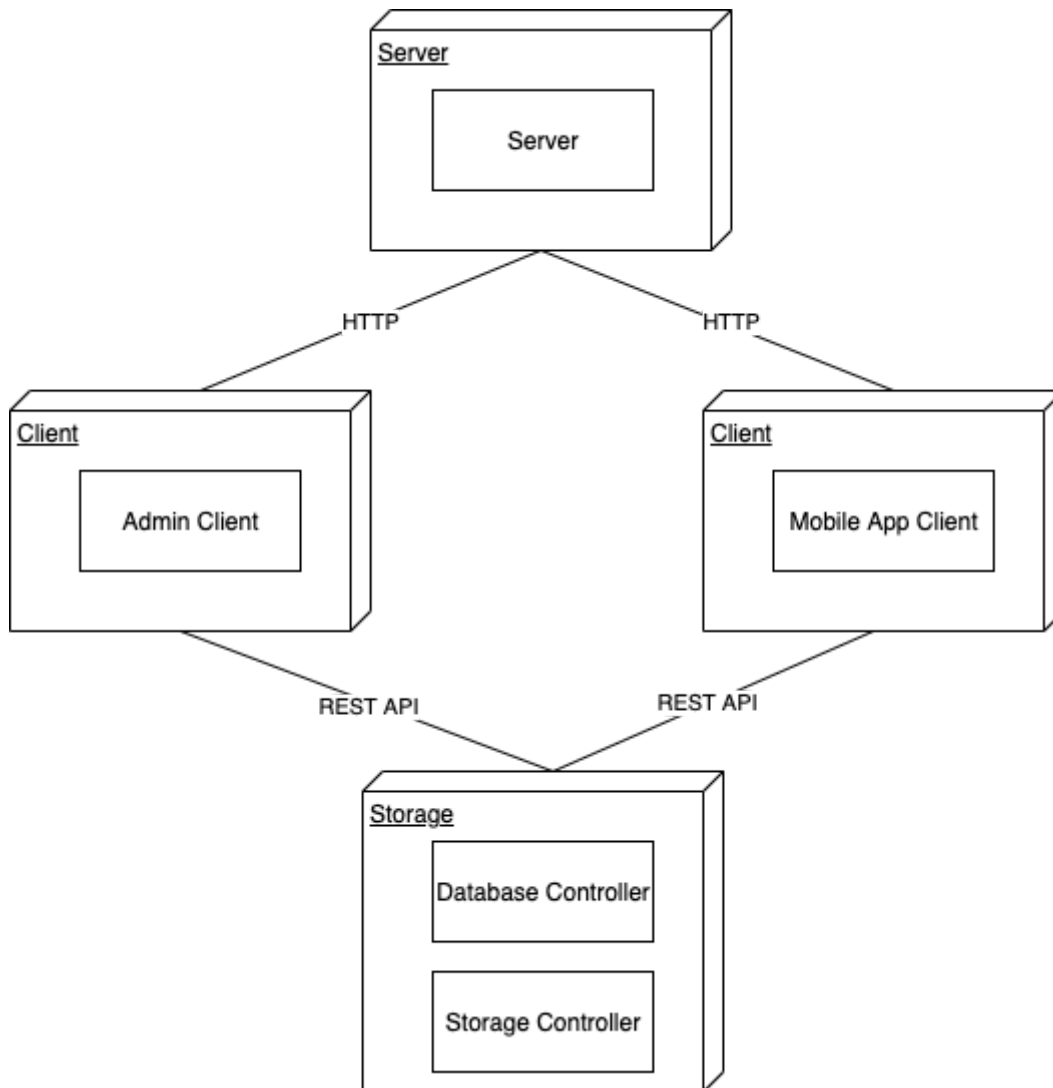


Figure 2. Hardware/Software Mapping

BidIt has two clients: admin and mobile app user client. Administrator client for web application admin for management of app. Also mobile app for mobile app user clients. 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 divided into two, one of which holds and presents user and application data. Another is responsible for the storage and presentation

of the photos uploaded to the system. In storage controller Amazon S3 bucket connection taking place.

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 username, password, ClientID, email, name, surname, address.
- **Product Data:** Individual Product unique data consist of productID , sellerID , name , imageURL ,description, expire date , start price, buy now price, winning bid,bid history.

Another service provider used to store the photos, thus ensuring optimum efficiency in database usage.

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 end date
- Getting various bids until the end time
- The highest bid at the end of the time will buy the product.

3.7 Boundary Conditions

BidIt will have three boundary conditions including initialization, termination and failure.

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.

4. Subsystem Services

4.1 Client Layer

4.1.1 Admin Client

Admin clients will be able to supervise auctions and users. Also, it sustains general maintenance of the user interactions.

4.1.2 Mobile Application Client

The mobile application client is the main medium that is used by the end-users.

4.2 Application Layer

4.2.1 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.

4.2.2 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 wishlist, retrieving ratings. To fulfill these tasks, it will also communicate with the database manager to update the database accordingly.

4.2.3 Search Engine

The search engine will enable the user to search products, auctions based on tags and sellers. The user can also customize the search with filters and sort options. The search engine will regularly get updates from the database to construct and maintain an indexing data structure to optimize the search operation. To elaborate, the data structure will be like a hashmap where keys are the tags and values are the related products.

4.2.4 Product Manager

This subsystem will handle the creation and editing of the products. Since the products will have varying attributes such as images, descriptions, categorization levels and stock information, it requires a subsystem for its own sake.

4.2.5 Payment Manager

The REST API server will redirect the payment request with payment credentials to the payment manager which will communicate with the external payment service. If payment is fulfilled successfully, it will inform the auction manager to update and terminate the auction.

The user will be informed about the payment status as well.

4.2.6 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.

4.2.7 Notification Manager

The notification manager will send notifications via the push notification service and the external mail service to inform the user in a case where a new bid is placed, an auction ended, payment failed/achieved. The push notification service should be able to deliver the notifications even if the client app is not on the focus or the phone is locked.

4.2.8 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.

4.3 Storage Layer

4.3.1 Relational Database

In order to store the relevant data about auctions, users, products and user activities, a relational database management system will be used by considering the needs of the application. The main reason for preferring a relational database is the necessity of ACID properties in the application. The tables in the database will be created in 3NF or BCNF forms to reduce the repetition and redundancy.

4.3.2 Storage Bucket

The various static files such as images, pdf files, videos will be stored in and retrieved via an object storage system. The storage system consists of buckets which are organized based on the file types. The orchestration of the buckets will be managed in this subsystem.

5. New Knowledge Acquired and Learning Strategies Used

First of all, React Native is going to be used and it is important for each member to be familiar with it by watching tutorials and some hands-on experience. Also, we aim to develop our leadership skills by letting each group member be the leader in some phase of the implementation. Each group member is obligated to learn the subjects that are related to their project in the workflow. After a period of learning, we believe we are ready to go to the next stage of the Senior Design Project which is implementation.

6. References

- [1] V. Krishna, *Auction theory*. Amsterdam: Elsevier, 2010.
- [2] *Amazon*. [Online]. Available: <https://www.amazon.com/>. [Accessed: 25-Dec-2020].
- [3] *eBay*. [Online]. Available: <https://www.ebay.com/>. [Accessed: 21-Nov-2020].
- [4] *Alibaba*. [Online]. Available: <https://www.alibaba.com/>. [Accessed: 25-Dec-2020].