Bilkent University

Department of Computer Engineering

Senior Design Project

Project short-name: BidIt

Low Level Design Report

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

Supervisor: Uğur Doğrusöz

February 8, 2021

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. 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 Object Design Trade-Offs

1.1.1 Functionality vs Usability

BidIt will favor usability because of its wide user base. Hence, the application should offer ease of use. The main functions of the app such as uploading an item and bidding for an item will be easily used. However, focusing on simple UI might cause a failure to highlight functionalities. Hence, implementing additional features is one of the challenges our group will try to overcome.

1.1.2 Scalability vs Performance

While users are making bids, the timing is important. Yet, as our app grows, scalability will affect the performance. It is known that planning for scalability is a safer and the easier choice

1.1.3 Reliability vs Rapid Development

To have a minimum viable product in our hands, we plan to start the development rapidly. Then, we will start prioritizing reliability and other functionalities.

1.2 Interface documentation guidelines

The following table will be used for explaining the interfaces and +/- signs indicate public/private attributes or methods:

Class	Class Name
Attributes	
+Attribute Name	Description
Methods	
-Method Signature	Description

1.3 Engineering standards

Unified Modelling Language(UML) is used to describe the classes and packages [2]. IEEE standards for citation are followed in the report [3].

1.4 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.

2.Packages & Class Interfaces

2.1 Client

Class	Login
Attributes	
-String username	Client userName
-String password	Client password
-String email	Client email
-String platformName	Client platform name
Methods	
+logIn(username,password)	User logins
+register(username,password,email)	User registers
+loginWith(platformName)	User logins with other platforms like google ,facebook..
+initPage()	Builds page

Class	HomeView
Attributes	
-String[] itemList	Show all items in home page
Methods	
+initPage()	Builds page

Class	Profile
Attributes	
-String userName	Client userName
-String email	Client email
-String address	Client address
-String profilePhotoURL	Client profile photo URL
-String[] items	Client items list
Methods	
+initPage()	Builds page

Class	UploadItem
Attributes	
-String itemName	Name of the item
-String[] itemPhotoURL	URL of the uploaded item photos
-String itemDescription	Description of item

Methods	
+addImage(User, item)	Adds the image into the apps image server and returns the image url
+changeImage(User, item,image)	Changes the specified image and returns the image url
+initPage()	Builds page

Class	AuctionView
Attributes	
-String itemName	Item name
-String itemPhotoURL	Item photo URL
Methods	
+initPage()	Builds page

2.2 Server

2.2.1 Controller

Class	UserController
Methods	
+changePassword(User, Password)	Changes the password of the user with the new password given as parameter.
+changeImage(User, Path)	Changes the profile picture of the user with the picture on the given path.

Class	PaymentController
Methods	
+choosePaymentOption()	Selects the payment option before processing the payment
+processPayment()	Processes the payment by executing transactions.

Class	AuctionController
Methods	
+newBid(User, Auction, price)	Makes a new bid on the requested auction by the given user.
+endAuction(Auction)	Ends the auction.
+isActive(Auction)	Checks whether the auction is still active or not.
+createAuction(User, Product)	Creates a new auction with the specific product.
+setExpirationTime(Auction)	Set expiration time for the auction.

Class	NotificationController
Methods	
+notifyViaEmail(User, content, time)	Sends notification to the user through email.

+notifyViaPushNotification(User, content, time)	Sends notification to the user through push notifications.

Class	DatabaseController
Methods	
+getAuctionById(auctionID)	Retrieves the auction data from the database.
+getUserById(userID)	Retrieves the user data from the database.
+findAuctionByKeyword(keyword)	Finds relevant auctions with the given keyword from the database.
+getBidHistoryOfAuction(auctionID)	Pulls the bid history of a specific auction from the database.
+getFavourites(userID)	Retrieves the favourite auctions of a specific from the database.

Class	ProductController
Methods	
+setDescription(description)	Sets the description of the product.
+setImage(imagePath)	Sets the image of the product.
+createProduct(title)	Creates the product object.

2.2.2 Model

Class	User
Attributes	
+Integer id	Unique ID of the user
+String name	Full name of the user
+String billingAddress	Full address of the user
+Wallet wallet	Bidit Wallet of the user
+String emailAddress	Unique email address of the user. Also needed as a login credential.
+String phoneNumber	Phone number of the user
+String passwordHash	Hashed password of the user instead of storing the password itself directly.
+String profileImage	Complete path of the profile image stored.
Methods	
+User()	The constructor of the class
+Getters and Setters	Gets and sets the attributes of the class

Class	Bid
Attributes	
+Integer id	Unique ID of the bid.
+User owner	User object representing the owner of the bid.

+Double value	Price of the bid.
+Date date	Date and time of the bid.
Methods	
+Bid()	The constructor of the class
+Getters and Setters	Gets and sets the attributes of the class

Class	Product
Attributes	
+Integer id	Unique ID of the product.
+String name	Product name/title.
+String imageURL	Complete path of the image stored.
+String description	Description of the product.
+User owner	Owner of the product.
Methods	
+Product()	The constructor of the class
+Getters and Setters	Gets and sets the attributes of the class

Class	Auction
Attributes	
+Integer id	Unique ID of the auction.
+Product product	The related product in the auction.

+Double startPrice	The initial price of the auction.
+Integer curPrice	Current price of the auction.
+Date endDate	Expiration time of the auction.
+Date startDate	Start time of the auction.
+Bid[] BiddingHistory	All bids made on the related auction.
+Boolean isActive	If the auction is on.
Methods	
+Auction()	The constructor of the class
+Getters and Setters	Gets and sets the attributes of the class

Class	Review
Attributes	
+Integer id	Unique ID of the review.
+Integer rating	Rating of the review out of 5.
+String title	Title of the review.
+String content	Content of the review.
+User author	Author of the review.
+User seller	The seller who is reviewed.
Methods	
+Review()	The constructor of the class
+Getters and Setters	Gets and sets the attributes of the class

Class	Wallet
Attributes	
+Integer id	Unique ID of the wallet.
+User owner	Owner of the wallet
+Double value	Current balance of the wallet
+Double[] transactions	Transaction history of the wallet.
Methods	
+Wallet()	The constructor of the class
+Getters and Setters	Gets and sets the attributes of the class

Class	Order
Attributes	
+Integer id	Unique ID of the order.
+Date orderDate	Time of the order placed.
+Date shippingDate	Time of the order shipped.
+String shippingAddress	The destination address of the order.
+String status	Current status of the order.
Methods	
+Bid()	The constructor of the class
+Getters and Setters	Gets and sets the attributes of the class

3.References

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

[2] “Unified Modeling Language,” http://www.uml.org/, Accessed: 2021-02-06.

[3] “IEEE Citation Guidelines,”

https://ieeedataport.org/sites/default/files/analysis/27/IEEE%20Citation%20Guidelines.pdf, Accessed: 2021-02-05.