Online Auction

System Design Document

V1.0

02.12.2018

Berkay YILMAZ
Tolga GÜLDÜTUNA
Feyzullah Berkay DANIŞ
Tekin EVRİM

Prepared for
SE301 Software Engineering

IŞIK UNIVERSITY
COMPUTER SCIENCE AND ENGINEERING

Table of Contents

SYSTEM DESIGN DOCUMENT

1. Introduction

Online Auction is a auction application that is developed for android platform. It designed to become collection of components that communicate flexibly with each other since there is no complex tasks in the application to do.It designed with React using Javascript to write to code and translate the software to an android output. The application itself is a simple ccmposition of familiar websites of ebay, gittigidiyor , letgo etc. Since the auction applications is not used too much in our contry we also get information to develop our application by looking universal applications or websites that is doing the same task. We wanted to use layered architecture on our project also our design goals are as follows;

The system should be easy to use by under the means of usability also it must support multiple users to supply needed performance to users. OA also need to be reliable since it has to do with auctions. The app must be secure to be under the terms of reliability.It also needs to be supportable by many versions of the android the reach more users.

1.1. Purpose of the System

The main purpose of the applicaion is to provide registered users a platform that they can easily and fastly their stuff. OA also a good opportunity to bring the auctioning style shoppining in to the local area and make people’s items worth as much as they do to prevent them from being scammed on or prevent from stuck with unselling the item.

1.2. Design Goals

Usability

Since the end-user will be using the system while anytime in day even in work , it is essential for the system to be intuitive and easy to use.

Multiple Users

The application should support tasks that are performed by multiple users in concert, supplying each with the necessary information at the appropriate time.

Reliability

OA should be provide all of its users a secure experience. Unregistered visitor should be able to see the current auctions and stuff. Yet, users need to be registered user to do bidding and selling procession. Logging to the application should be provided with unique e-mails and password that are appropriate for password criteria. The membership information should be private and should not be shared with anyone without the will of the user.

Performance

OA should be responsive in different versions of Android OS and it should be able to scale correctly in different versions of Android OS and devices. It should be running in more than one mobile device . OA is going to be dynamic content, so there should not be complicated queries in back end to not decrease performance.

Understandability

OA must be easy to understand for users to all ages.Bidding or Selling functions must be most 2-3 clicks events.

Supportability

OA should be managed by admin. The application should be supported on different android versions and be independent by hardware mostly. In the react native part system should be able to open for new implementations easily.

Implementation

There are no constraints on the hardware platform. There are no constraints imposed by the maintenance team. There are no constraints imposed by the testing team. The design methodology is obtained as Agile Development Approach. System runs with query. JavaScript language is used in this system. Query based statements are handled by React Native. JavaScript, ………..

Legal

The Online Auction does not use any license or licensed work. It is a student project.

1.3. Definitions, Acronyms, and Abbreviations

SDD : System Design Document
OA : Online Auction
React : **React Native** is a JavaScript framework for writing real, natively rendering mobile applications for iOS and Android
Admin: System actor that administrates the system.

Visitor: System actor that is not a registered user of the system.

Subsystem: Collection of classes, associations, operations and events closely related to each other.
HTTP : The HyperText Transfer Protocol

1.4. References

www.gittigidiyor.com
www.ebay.com
www.tr.letgo.com/tr
www.liveauctineers.com

2. Current Software Architecture

There are some websites and applications that are related to our application . These applications are to provide auction bidding and selling platform between registered users. The applications and websites mentioned are;

www.gittigidiyor.com

www.ebay.com

www.tr.letgo.com/tr

www.liveauctineers.com

The common applications are big and focusing on a bigger area. But our application is only focusing on auctioning in our local area. Since there is no application that is serving only this service in our local area this is a good opportunity for our application. The common problems also in letgo there will be here too. We need to prevent users to be scammed in an auction process. Since we are not responsible for any payment and we are just a platform that helps registered users to sell their stuff with auctioning , the money transections will be in the users responsibility. Besides from that local people are not used to use auctioning to be a part of their lives. But this could initially turn into a opportunity.

3. Proposed Software Architecture

In our application we want to make sure that people can make offer or sell items easily. The interface will be user friendly, simple and easy to understand. The system has got three actors admin, registered user, visitor. The data about actors and auctions are stored in database. The data about actors and auctions are taken from database using queries. Users of the system can register to the system for free, they can search and see the items in the system but to be able to make offer or sell, they need to register to the system first. Admin and registered users will have profiles in the system and they will be able to edit profile, see their history of auctions etc.

3.1. Overview

The system will have a layered architecture and there will be 3 layers interface , application and storage. Users will be interacting with the system using the interface layer, application layer will have all the functions of the system in it and storage layer will have all the information about the users and auctions

Storage layer will have the registeredUserData, CurrentAuction and OldAuction subsystems that has got the information about users and auctions.

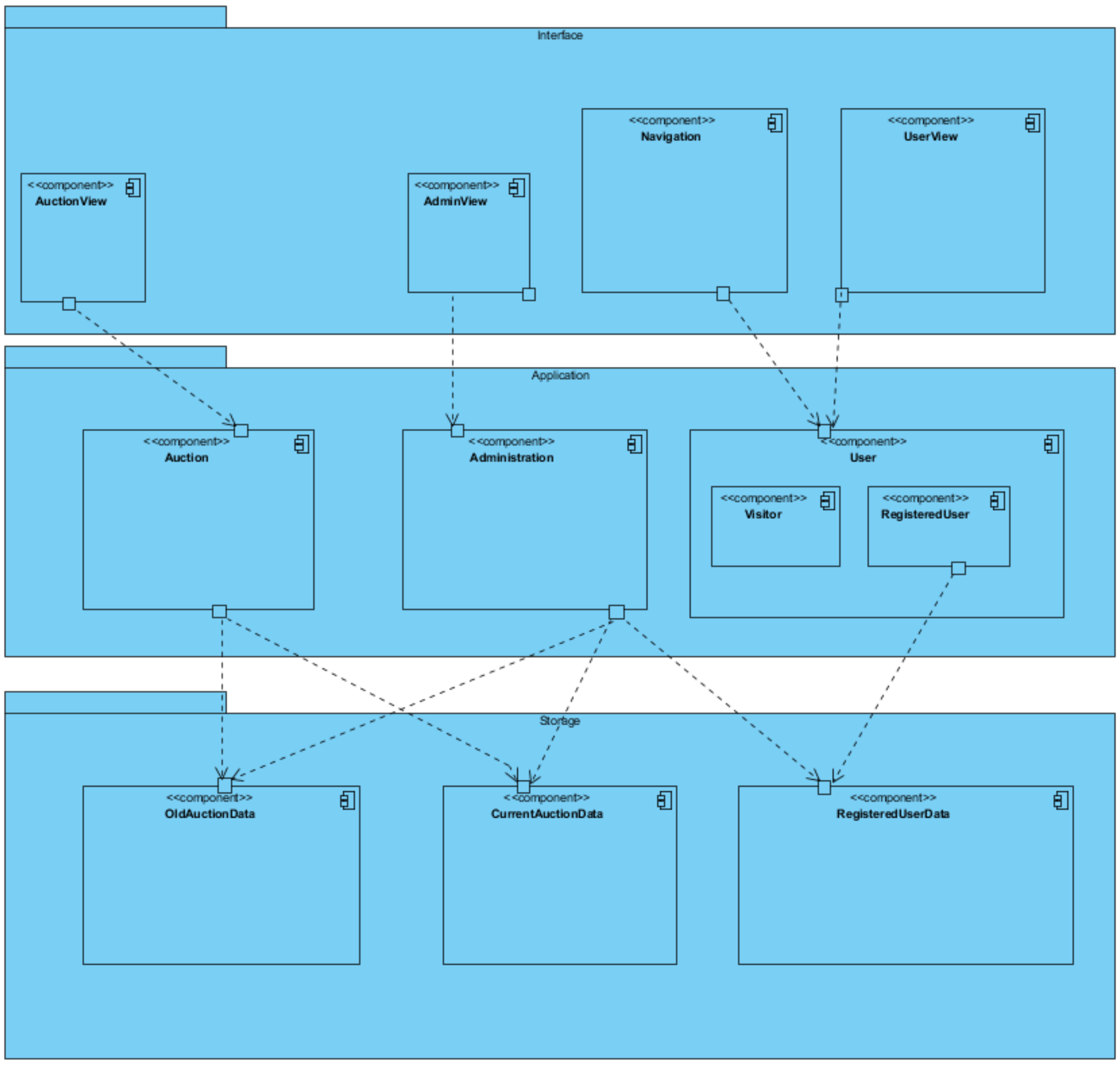
Application layer will have the auction, admin, user(visitor and registered user) subsystems.

User subsystem will have the functions of register, login, searching an item on the system, editing the user profile and sending or reading messages. User subsystem keeps some of the functions for both visitor and registered user such as searching an item but some of the functions are specific to registered user and visitor separately. Visitor susbsytem has got the registration function. Registered user subsystem will keep the functions of login, editing or freezing profile, messaging.

Auction subsystem handles the auctions created by registered user, subsystem will have the functions of giving an offer to auction and winning the auction.

Admin subsystem will have the functions of managing an auction such as approving or rejecting the auction, managing the user profile such as editing their information or freezing account and managing the messages

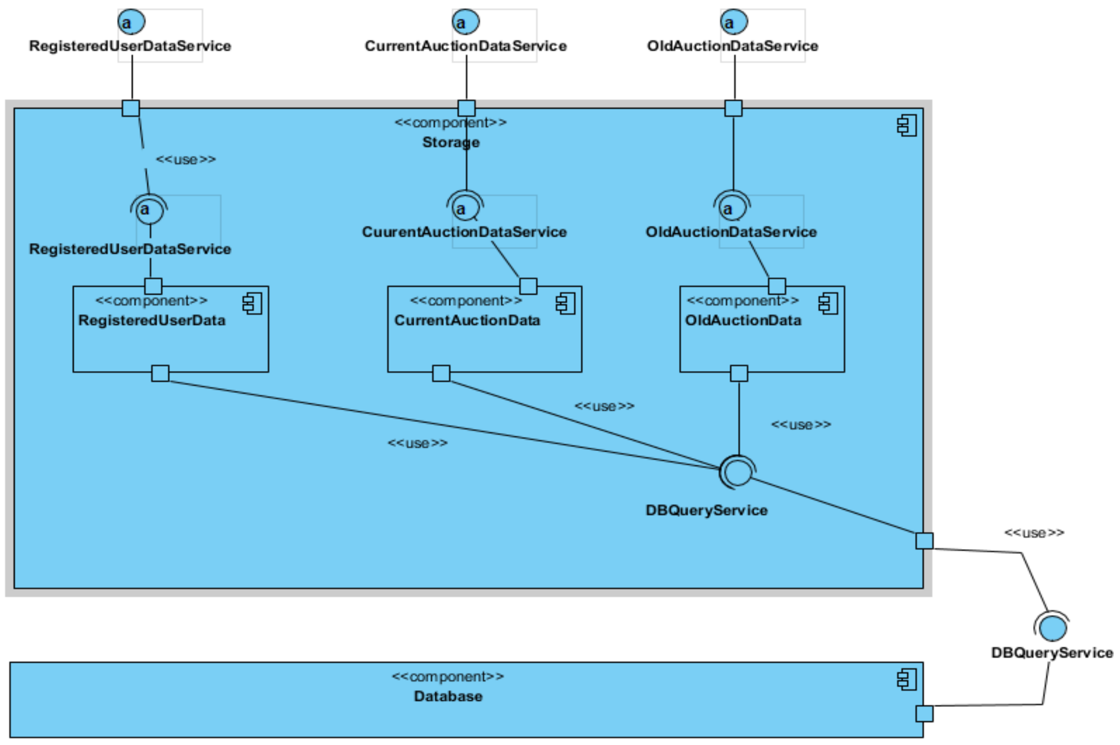
3.2. System Decomposition



Storage layer holds the RegisteredUserData, OldAuctionData and CurrentAuctionData subsystems. RegisteredUserData holds the information about the OA Applications’s users which is RegisteredUsers and the Admin.

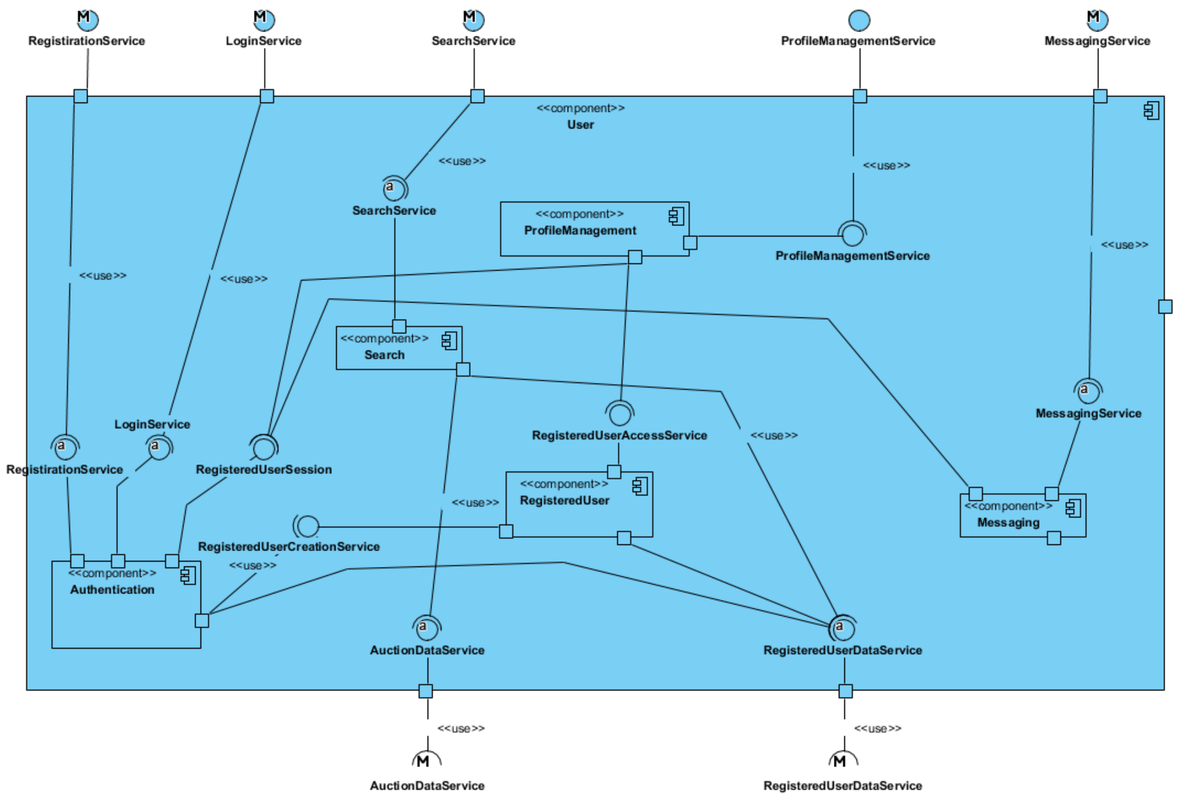
Application layer holds Auction,Administration and User subsystems which has 2 types of as visitor and RegisteredUser.In terms of registeredUser , user subsystem is about common user functionalities like messaging , profile management, search and log-in/log-out services. Auction subsystem is the subsystem that manages the main auctioning services. Administration subsystem is the subsystem that manages the all admin privilages like approving or rejecting an auction proposal or even freezing an user’s account.User subsystem is the subsystem that manages all the common services that an user needs.

Finally in the Interface layer holds AuctionView,AdminView,Navigation and a UserView. AuctionView is the subsystem that controls the giving and winning auction services.AdminView is essential for admin to manage common services that includes users. Navigation is the primary for services like login search and register.Last of all UserView has services that giving users to manage simple actions.

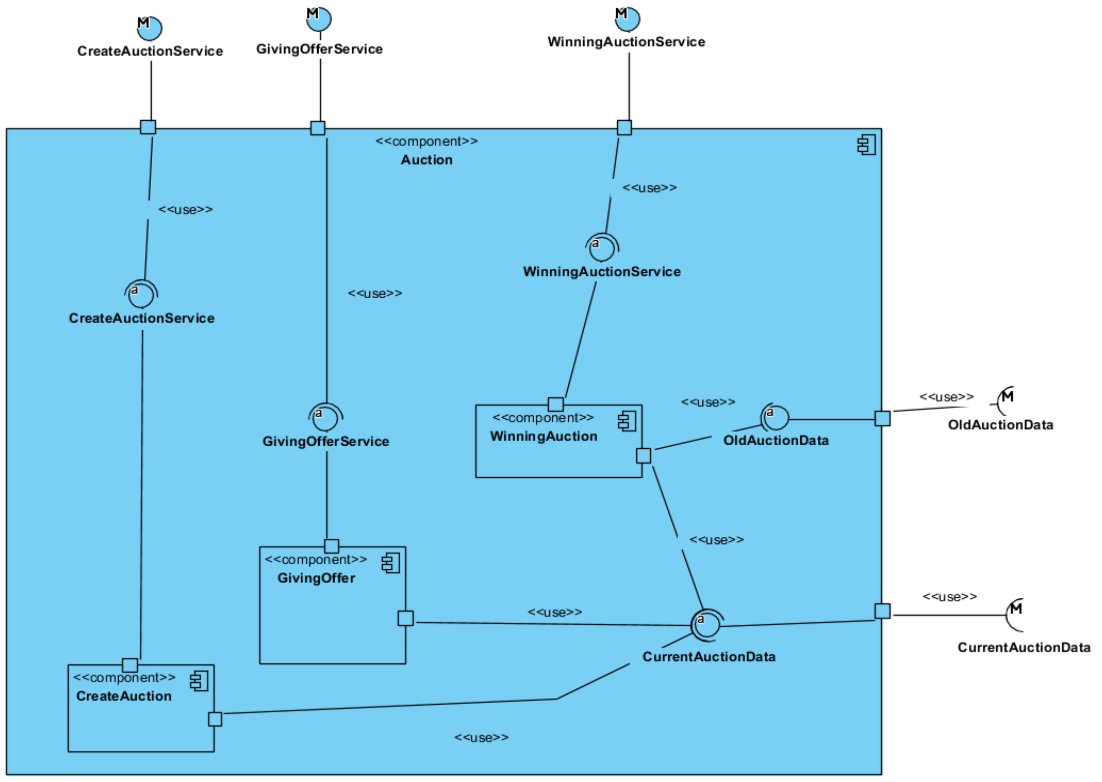


The ‘Data’ subsystems provide an interface to the Database for a more ‘human readable’ access to its data. In the Storage subsystem we are controlling all necessary data and connects to the Database system via DBQueryService. It holds data’s like Current and Old Auction Datas and also RegisteredUserData.

<Online Auction>

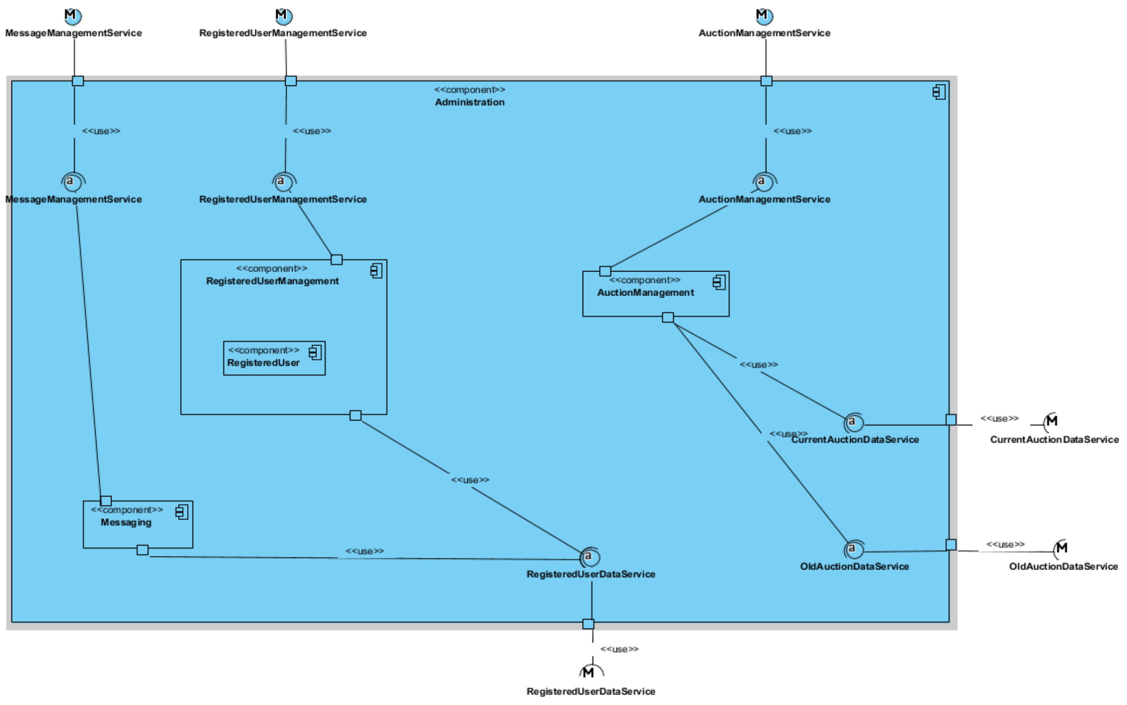


The User subsystem provides registeredUsers their self-made functions.These are like follows; Registration , Login – profile management , messaging with other users or admin, Viewing ongoing auctions which is neither the registereduser’s own auction or participating auction and finally It also provides the search functionality for system.

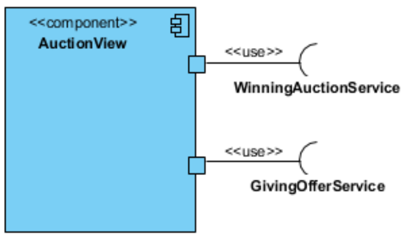


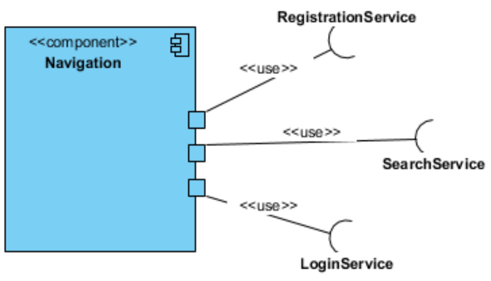
<Online Auction>

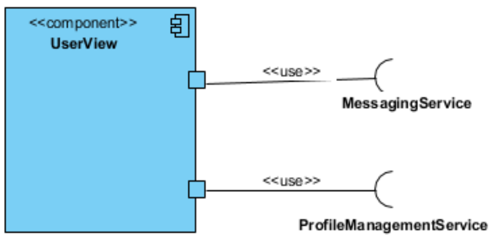
The Auction Subsystem is about giving offer to an auctions and also winning an auction by a registeredUser. It handles auction creation(created auctions send for approval to the admin), giving an offer to an auction or winning an auction and also handles the storage management and the interface with the storage.

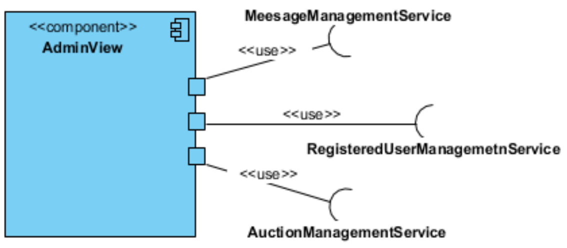


The Administration Subsystem handles Admin’s functionalities, such as management of Registereduser(freezig their Accounts etc.), Auction information(which is starting as currentAuctionData and after times up transferred into OldAuctionData), approval or/and rejection of auction creations, viewing received messages from RegisteredUsers and sending them messages.





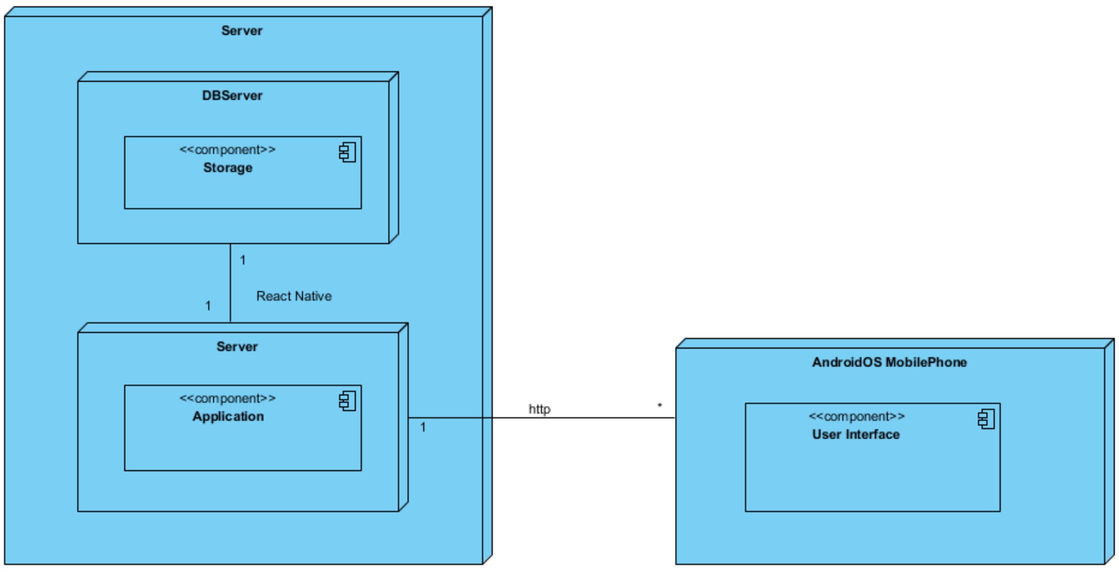




Our system is a mobile application that runs on AndroidOS and in the presentation layer it holds the which users can interact with.It holds 3 View subsystem for each application layer subsystem. It also holds the Navigation subsystem which holds the pages like homepage, register and login pages as well.

3.3. Hardware Software Mapping

The software for our application uses 3 layered architectural style which allows flexibility while we are deploying our application. Bur for the other purposes , we will have 2-node design where we are holding our application and storage layers in our one node which is in our server and the presentation layer on a mobile device running on Android OS. While this provides great ease in deploying the system, it also creates a bottleneck, that is, the Server node itself. For high traffic situations, it might become overloaded with requests but for our initial small scale deployment it should be fine.

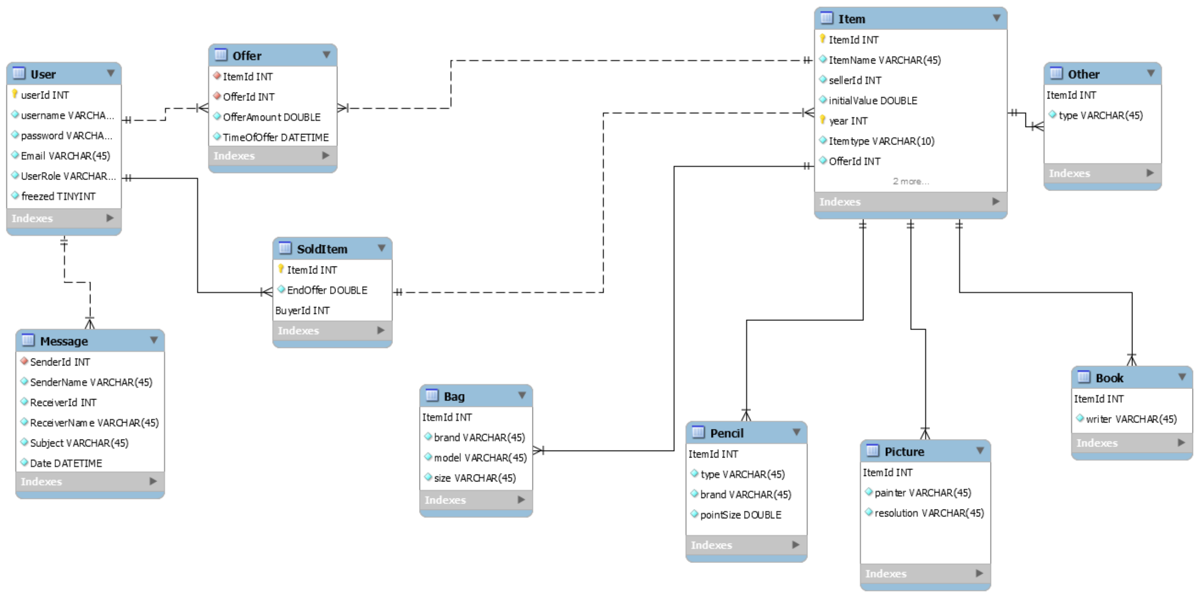


3.4. Persistent Data Management

The system needs persistent storage of its users, their information, offers, items and their relations with each other And these data must be accessed repeatedly - many times by multiple users - and each request should be returned as fast as possible. For these reasons, using a Relational Database Management System is a must.

While the initial userbase may be small, the system storage infrastructure should be scalable for a growing userbase. Data security is also another concern.

For these reasons, we will be using MySQL as our Database Management solution which covers all of our needs and more. Also, it’s Community Edition - which we use, is open source under the GNU General Public License, helping us reduce costs. It’ll also create an access queue for the data it holds, which allows for reliability in concurrent access scenarios.



The data scheme for the database is mainly made up of its users, their offers and the auction created by registered user’s. The ‘user’ table holds information needed from each and every user, password (hashed) etc. The ‘registered user’ tables hold additional for each user type’s needed information. This might be an registered user “About Us” text or a auction’s offers. The Admin user resides in neither of these two tables and is the sole entry of ‘user’ table.

‘Item table’ included subtables of the item types.Their are bag, pencil, book, bag and other items.

Finally, there is the ‘messages’ table that hold the messages received by each user .As for encapsulation, the Storage Layer ‘glued’ atop the Database will provide query-independent access to data for the higher layers.

3.5. Access Control and Security

	Administration	User	Auction

Admin	ApproveAuction()
	RejectAuction()
	FreezeUser()
	viewUserInformation()
	viewUsersList()
	viewNGOs()
	messageUser()
	viewMessages()
	sendMessages()
		login()
		search()
		viewItemInfo()
			ViewAuctionInfo()

Registered User		login()
		forgetPassword()
		search()
		UpdateProfileInformation()
		viewItemInfo()
		viewMessage()
		sendMessages()
			GiveOffer()
			CreateAuction()

User (Visitor)		register()
		search()
		viewItemInfo()	

Users must provide an at least 6 characters long password to register. The database shall hold the password information hashed. As the system is not designed to hold sensitive information for users, no other encryption is required.

3.6. Global Software Control

- Administration subsystem initiates OldAuctionData, CurrentAuctionData and RegisteredUserData subsystem to access/add/edit database data.

- User subsystem initiates RegisteredUserData subsystem to access/add/edit database data.

- Auction subsystem initiates CurrentAuctionData subsystem to access/add/edit database data.

- Navigation subsystem initiates Authentication subsystem for registration and login operations.
- Navigation subsystem initiates Search subsystem to search for running auctions.

- UserView subsystem initiates ProfileManagement subsystem to allow the user to interact with his/her profile information.

- UserView subsystem initiates the Messaging subsystem to allow the user to access his/her messages and/or message the Admin.

- AdministrationView subsystem initiates UserManagement, CampaignManagement or DonationManagement subsystems to manage each respectively.
- AdministrationView subsystem initiates UserManagement or CampaignManagement subsystem to approve user registrations or campaign creations respectively.
- AdministrationView subsystem initiates MessageManagement subsystem to view admin messages and allow the admin to message other users.

- CampaignView subsystem initiates CampaignEditor to create and edit NGO created campaigns.
- CampaignView subsystem initiates CampaignViewer for retrieving campaign information for viewing purposes.

The database queues each query which provide concurrency control for storage.

3.7. Boundary Conditions

Startup:
- First of all , admin is declared with react native.

Logging in:
- Username and/or password field are blank.
- Password is not 6 characters long.
- Password and username don’t match.
- Username is wrong or does not exist.
- Main Page does not appear after logging in.

Register:
- Username is already in use
- Password is not 6 character long
- Username and/or password and/or password again fields are blank.
- Password and Password again are not match

Profile Management:
- User can’t edit his/her information or the changes do not reflect.
- System crashes while editing profile information.

Starting Auction:
- Registered user send a request for a new auction without selecting a category.
- Registered user send request for a new auction without filling the empty blanks about the information of the item.

Giving Offer:

- Entered bid amount is lower than current bid.
- Bidding ends while making another bid.
- System crashing while bidding process.

4. Subsystem Services

4.1. Presentation Layer Subsystem Services

- Takes user input
- Passes user input to the Application layer
- Displays Application Layer results

4.2. User Subsystem Services

- *RegistrationService:* User registration
- *LoginService:* User login
- *UserSession:* Identifies the current logged in user using the system
- *ProfileManagementService:* Provides access to the logged user's own information. Ability to view and edit its various properties.
- *RegisteredUserCreationService:* Generator for an editable *RegisteredUser*object for the current session.
- *RegisteredUserAccessService:* Access functionalities for the current logged in user's session's Donator object.
- *MessagingService:* Access to current logged in user's messages and messaging functionality.
- *SearchService:* Search functionalities for *Registered Users* and their Auctions.

4.3. Offer Subsystem Services

- *GivingOfferService:* Creation of an offer to an auction.

4.4. Auction Subsystem Services

- *WinningAuctionService:* Winning of auction information.
- *AuctionEditorService:* Creation of auctions, editing of auctions information.

4.5. Administration Subsystem Services

- *UserManagmentService:* Access to system users and functionality to create and edit user information.
- *AuctionManagementService:* Access to created auctions and functionality to approve or rejected their information.
- *MessageManagementService:* Access to messages sent for Administration. Functionality to send messages to users.

4.6. Storage Layer Subsystem Services

- *UserDataService:* Interface service for Database access to User information.
- *OfferDataService:* Interface service for Database access to Offer information.
- *AuctionDataService:* Interface service for Database access to Auction information.
- *DBQueryService:* Database query entry service.

5. References

1. https://stackoverflow.com/questions/36385231/can-i-use-3-tiers-architecture-and-mvc-together-in-android
2. Bruegge B. & Dutoit A.H.. (2010). Object-Oriented Software Engineering Using UML, Patterns, and Java, Prentice Hall, 3rd ed.
3. “Gnu.org.” The GNU General Public License v3.0 - GNU Project - Free Software Foundation, www.gnu.org/licenses/gpl-3.0.en.html.