Design Document for <<Bid it All>>

Group <2_NS_1>

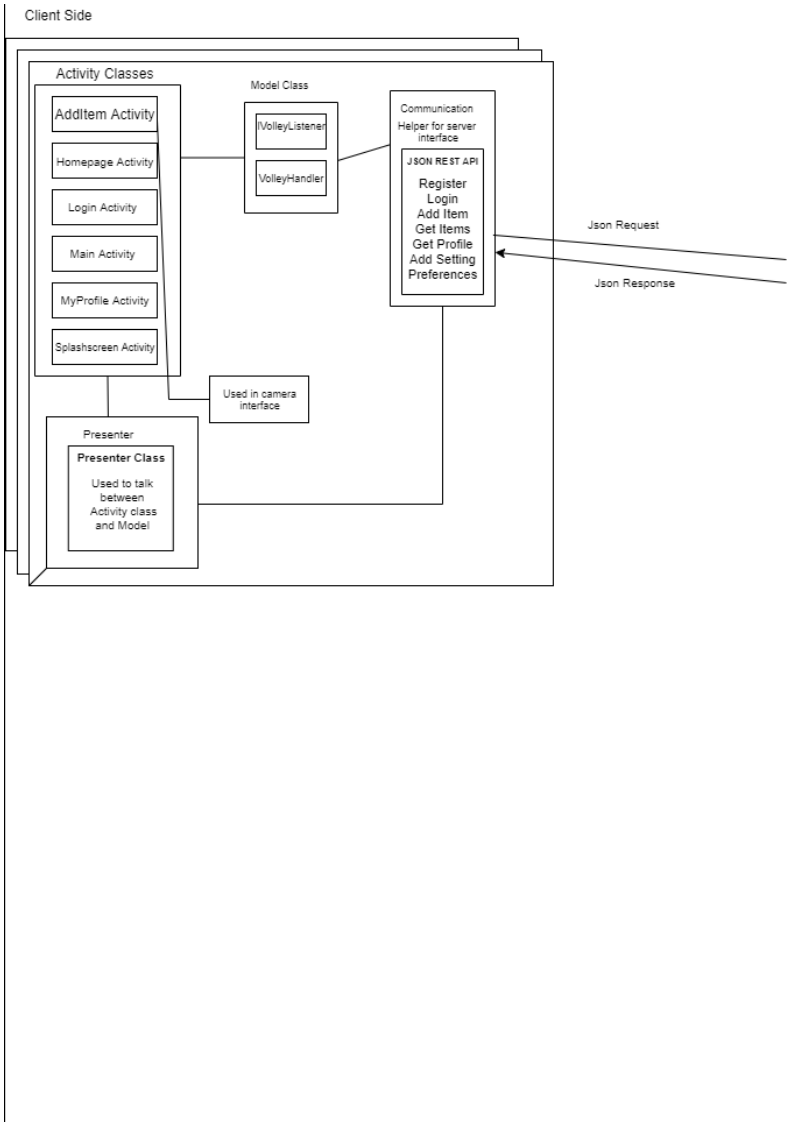
Ishan Banerjee: %25

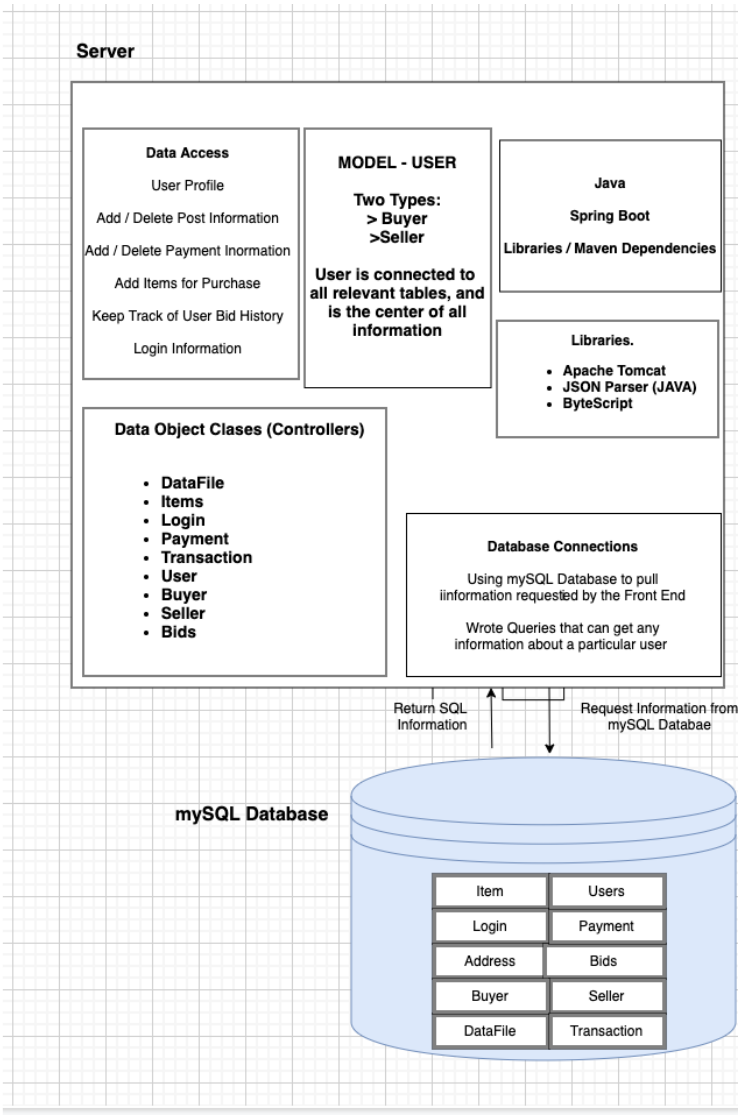
Aditi Kulkarni : % 25

Aashish Komaragiri: % 25

Anirudha Kyathsandra: % 25

BLOCK DIAGRAM





Design Description

Android View:

Our application's front-end consists solely of Android clients; thus, we utilize a combination of XML and Java classes to control the communication and population of our application's graphical user interface. Since Bid it all is solely dependent on user interface and data retrieving, the application's code written in frontend deals with retrieving ad name, text description, numeric price and image. Other view, provides users with their profile information, settings page for their preferences and a page to add items.

Android Model:

Our app has its own designated server to do tasks and uses MySQL for data storage.We use the Volley library to pass in JSON requests from the Android client to our server. There are two classes under model currently but more might be added according to future needs. By using model class, we are standardizing how transfer of data will occur along the client. It will also be used to transfer data between classes.

Android Presenter:

The presenter class is mainly to fetch data from model class and apply UI logic into showing what needs to be displayed. For now, we just have one class to handle requests but more will be added soon.

Server Design Description

Model - USER:

The User table is the main database table, that is connected with all other tables mentioned in the Back End side database. Using one-to-one and one-to-many relationships, the user table is able to link with other tables and retrieve data about a certain User in many different aspects. This is a very nice feature to have, as it makes everything so much easier for the Frontend, as they just need to pass in a UserID, or user information, and the backend can communicate any information needed by the user easily.

Data Access:

The Data Access portion of the block diagram mentions all the types of information the user can have access to, and the back end is able to send to the Front end. When we receive a request from the Front End, we evaluate

the JSON object, and determine which controller to use the appropriate request for. This makes the layering organized and efficient, and allows the certain requests being mapped to follow a fixed process.

Below are brief descriptions of the type of data received

- User Profile - retrieve any importation information about the profile of the user such as first name, last name, address, email or phone number.
- Add / Delete Post - retrieve and store information about a new post on database, or delete a post and remove record on database
- Add / Delete Payment Information - retrieve and store information about a new payment information on database, or delete a payment information and remove record on database
- Add Items For Purchase - update user items that need to be purchases on database
- Bid History - save old and new bid information that user has made
- Login Information - save, update or delete login information for a given user

Database Connections:

SQL- One-One - One:Many:

_____Developed an organized, database model that allows the user to link with all tables relevant to the user. This allows tables that are made to be well-structured, and strict to the guidelines for the given table.

The tables implemented for this project are Item, Login, Address, Buyer, User, Buyer, Seller, DataFile, Payment Bids, and Transaction.

The Database used for this project is mySQL. Once the Controller sends a request to the database, asking for certain information pertaining to some user, the database retrieves the given information and returns it back to the controller. Then a JSON object is created, and sent back to the front end with the requested data that was wanted.

Data Object Classes:

When a user from the frontend sends a request to the backend, the controller is responsible for mapping the request(in the form of a JSON object) to the information needed by the request. Once it maps the information needed, it gets it from the database, and sends it back to the frontend. This is one of the most important features of the program because it deals with communication between the frontend and backend. There are various controllers that take care of what kind of information needs to be sent back to the frontend. Some of the components of the controller are:

DataFiles: Stores files into the MySQL database.

Items: Stores and returns information about the items that are for sale. Some of the information it stores includes item price, item name, item id, images of the item, posted date, end date, etc. So once a user sets this information while creating the ad, this information is sent to the backend via the controller and when another user "requests" for that information, the controller maps it to the corresponding table and returns that information back to the user.

Login/Registration information: Stores and returns login information. Basic items such as username, an encrypted password, login info such as phone number/email address are stored here.

Payment: This transports payment information. This controller mainly stores the JSON from the frontend which sends payment information such as credit card number, other transaction type of information.

Transaction: This maps transaction information such as who is the seller, who is the buyer (seller and buyer ID), price, item that was sold, etc.

User: This controls the basic user information such as user ID, username, things common to buyer and seller tables.

Buyer: This maps information about the buyer-> buyer ID, name, address, etc.

Seller: This stores information about the seller->seller ID, name, address, etc.

Bids: Gives information about current bids, who is the buyer, who is the seller, bidding price, etc.

Bid it All ERR Diagram