

BidiT

A Marketplace for unique and antique items

By Team Pistol

Developed By:

Gurvinder Singh

Kunwardeep Singh

Akhil Chib

Index

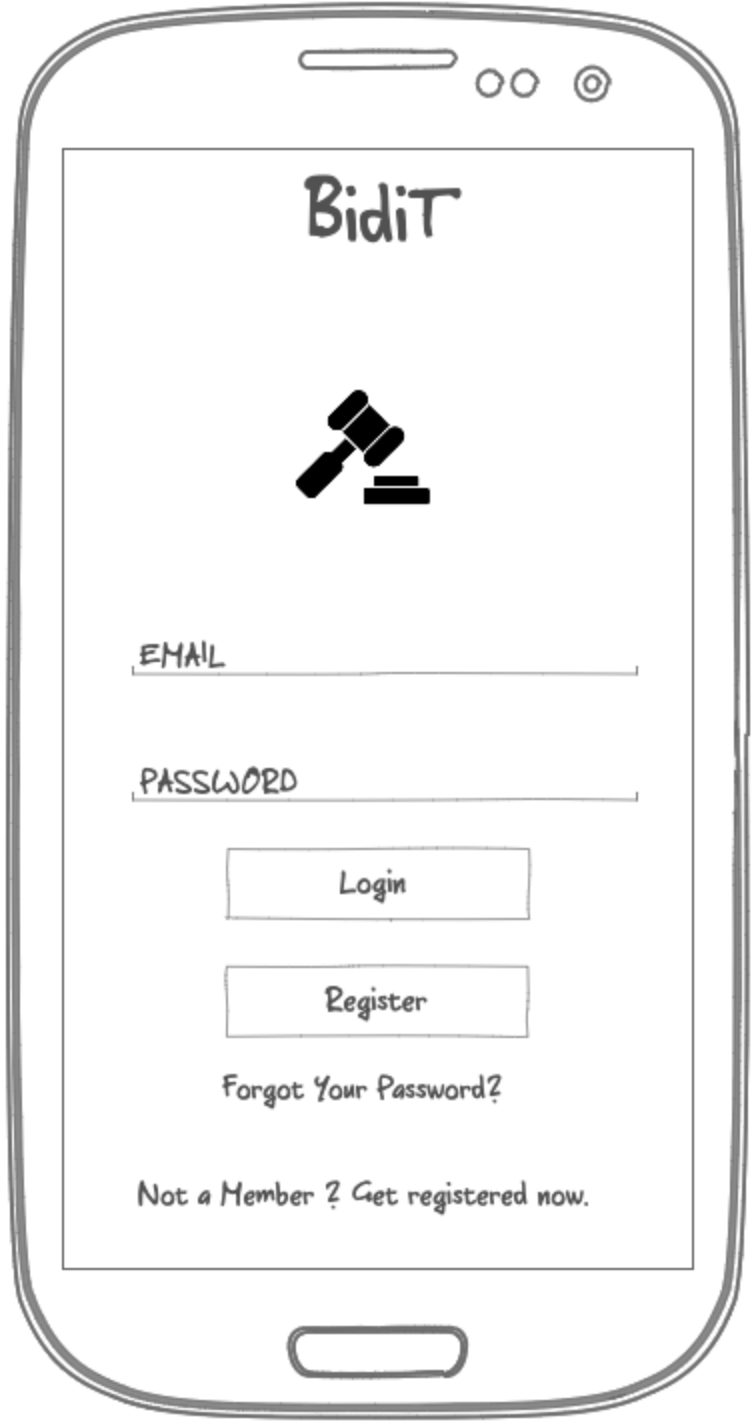
* Data flow Design for Buyer.
* Data flow Design for Seller
* Database Design.
* Test Cases
* Monetization Plan.
* Appendix A.

Code to push the bid to database.

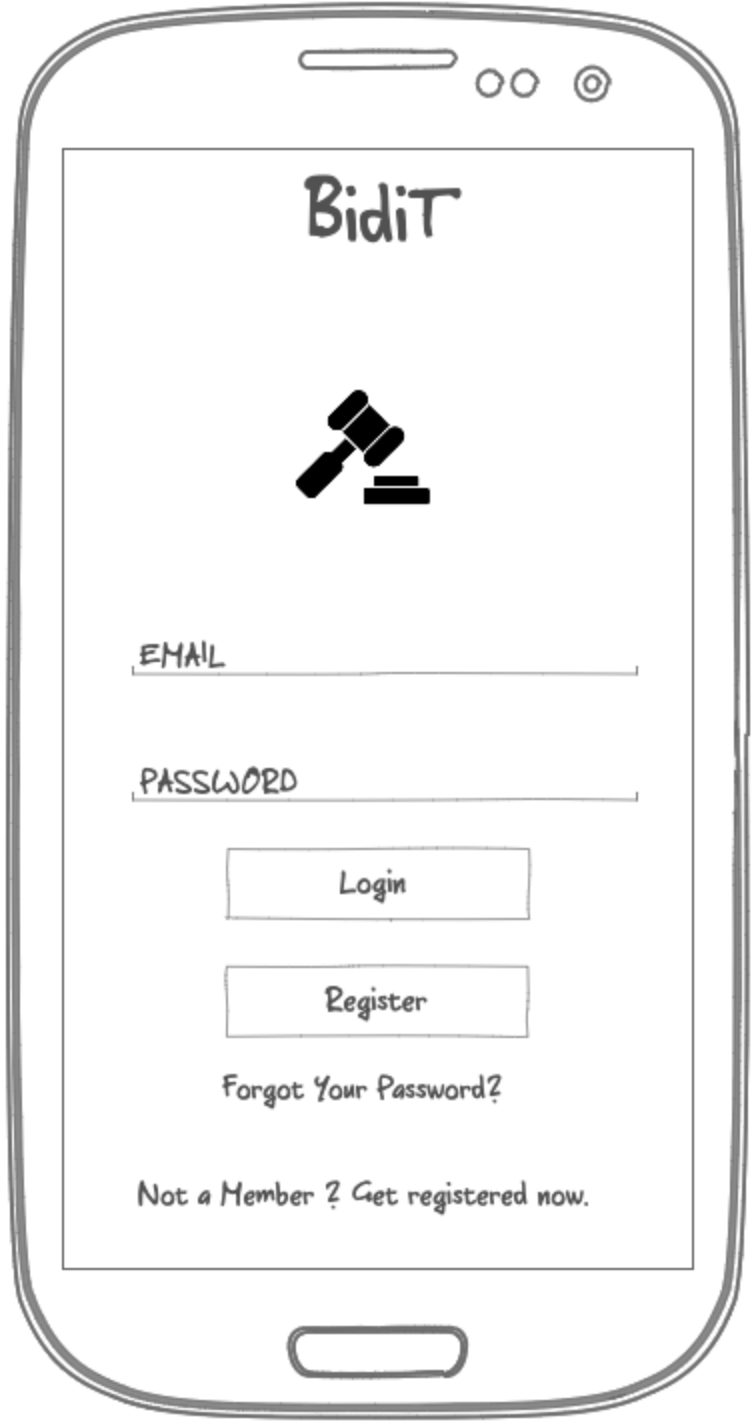
* Appendix B.

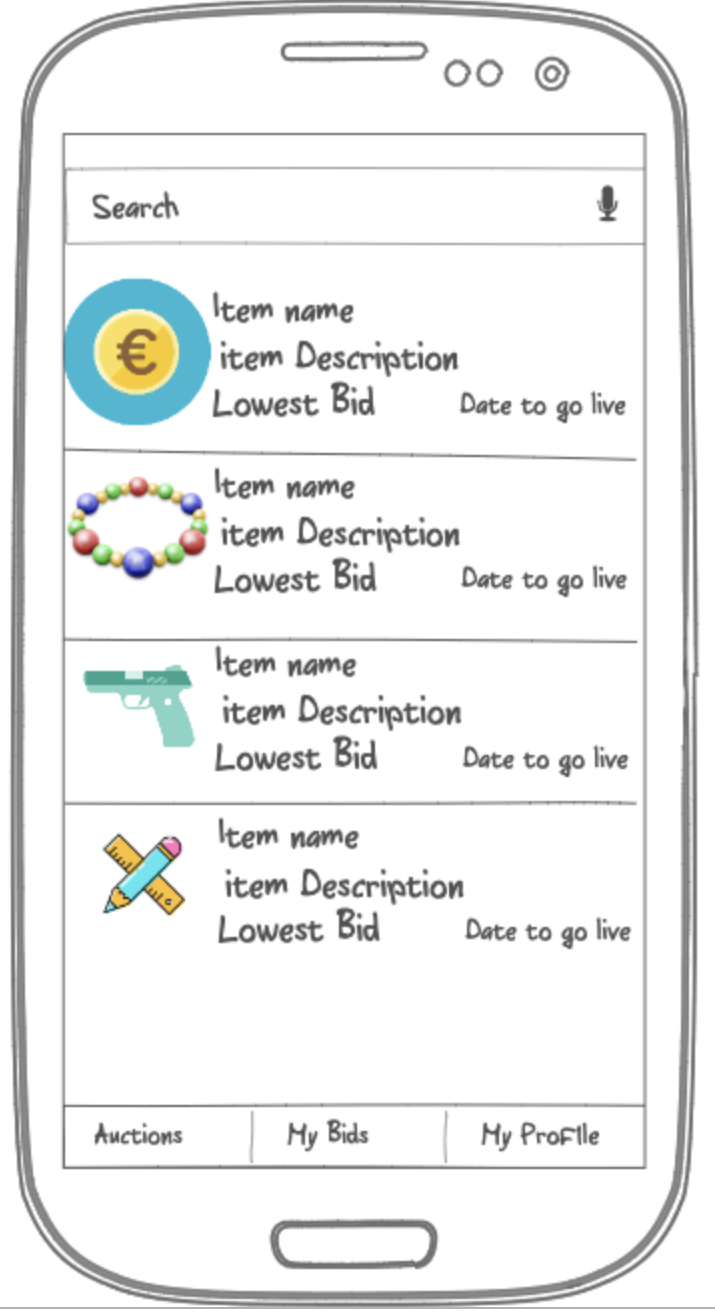
Library used to validate visa card details.

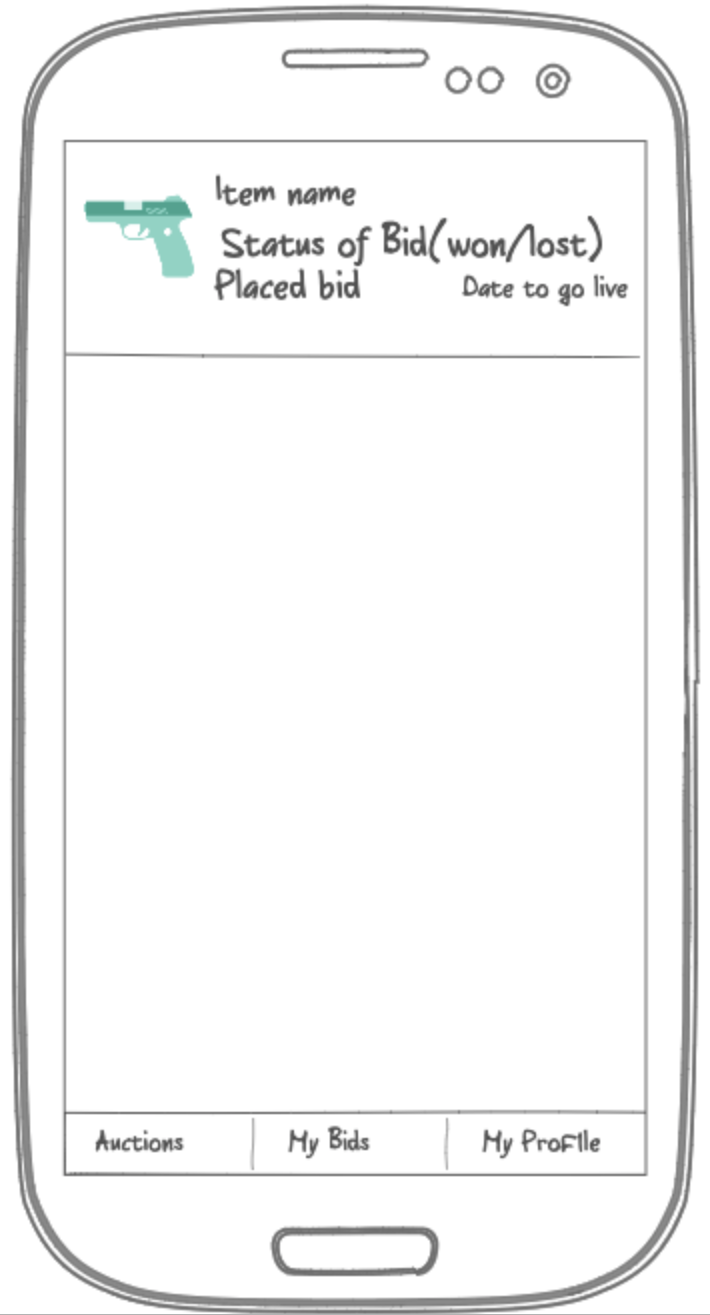
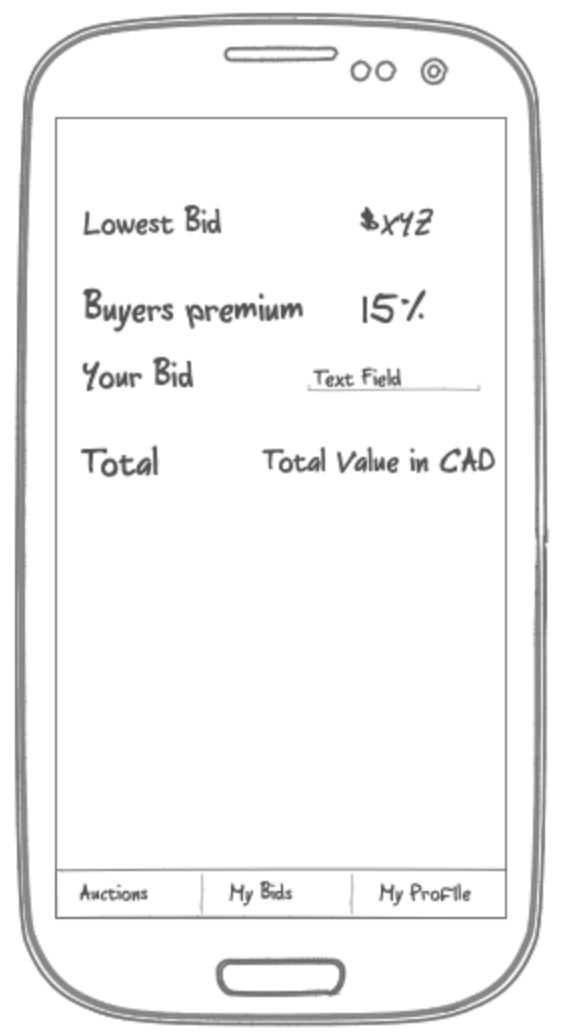
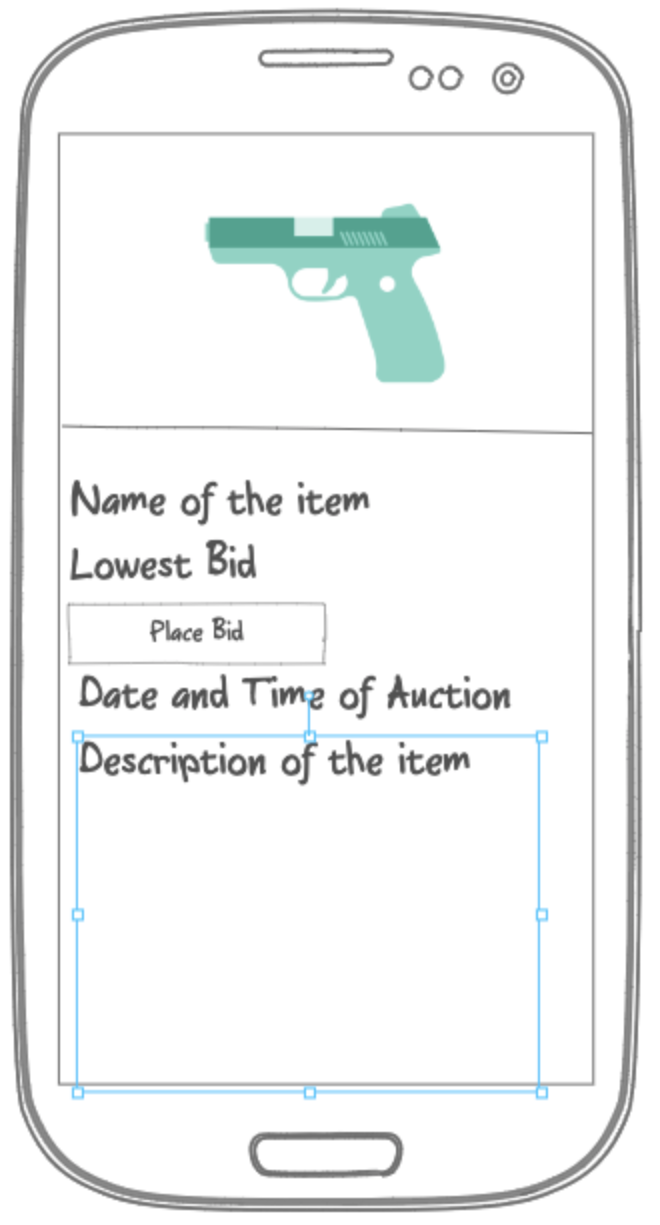
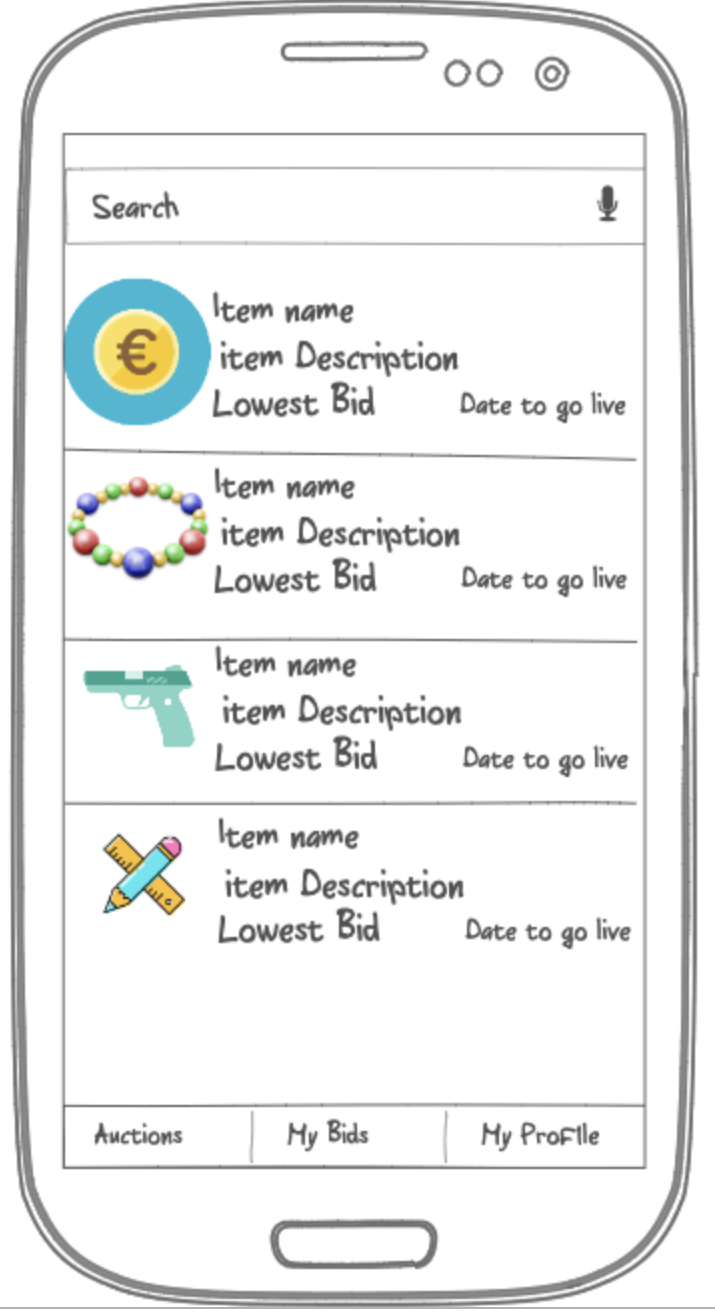
Control Flow of the Application for Seller





Control Flow of the Application for Buyer





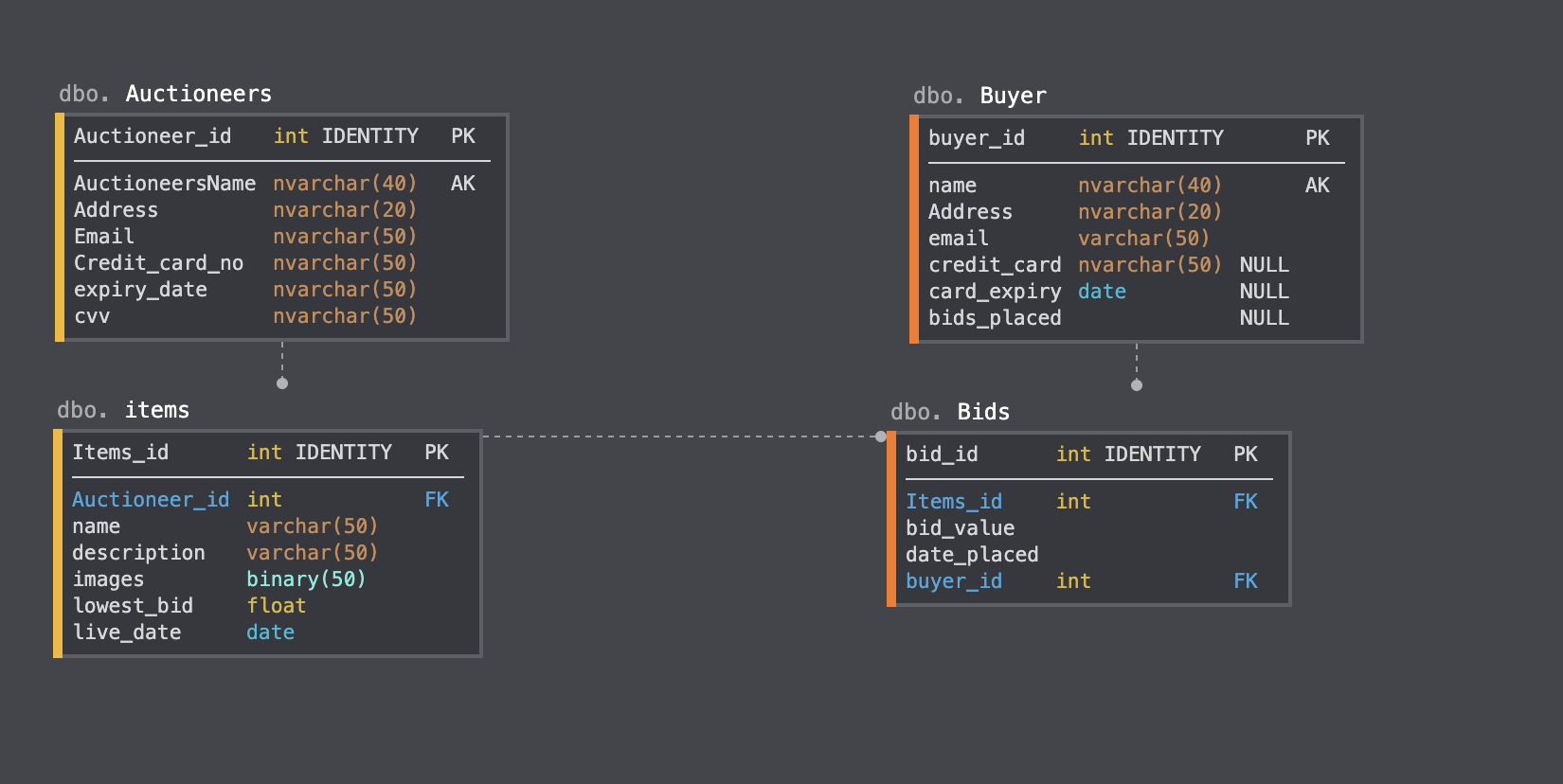
Database Design

The data will be saved in various tables , the schema for which would be shown in coming pages.

**Auctioneer**: The data of the people who want to sell their items will be saved in here, the entities in this table are required as the authentication for the seller is necessary, including credit card details.

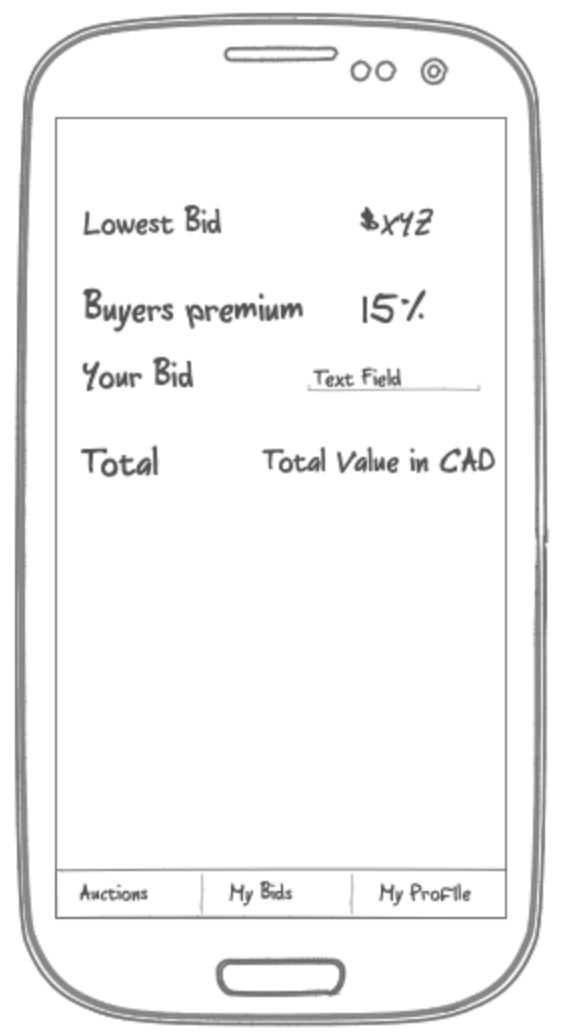
**Buyer** : The data for the potential buyers will be saved in this table, the columns for the table are listed in the design. The credit card details are not a compulsion when registering for buyer, it can be added later when the bid is placed.

**Items:** This table will save the data of all the posted items that are available for bidding, including the id of the person who posted it and the date when the bidding expires.

**Bids**: This table will include all the bids that are placed, it will include the id of the item on which the bid is placed, the id of the person that placed the bid and the date on which the bid was placed with the value of the bid.

Test Cases

* At first we will post some things on the application and distribute the application for buyer in friends and family and ask them to place bids.
* We will also ask some people to register as sellers so that we can verify the whole registration process for the seller.
* Like this we will test the whole process. Starting from the registration of the seller and posting of an item to be auctioned.
* For the registration of the buyer and placing of bids by various people and then winning of one bid and the deduction of the payment using the gateway and getting the money.
* Then we will be actually shipping the product from one place to another and to see if the whole process works from start to end.

Monetization Plan

* We are planning to earn money via taking a few percentage of the fee from the placed bid, the that is being placed includes the percentage that we are keeping so that the user does not get any surprises later.
* This fee also includes the fee that is required to process the shipping of the product.
* The product will be shipped using premium services and will be only handed to the person who placed the bid and a signed form will be required that the the person got the product in a good condition.
* The deliveries will be previously scheduled and if the buyer misses the delivery because of some reason then they can go and collect their item from the shippers office, the delivery would be only done once.

Appendix A

This is the code to push the bid to the database, spinner is our dropdown which shows the options of the bids and Bid is the class which is used to push row into the table.

public void saveBid(View view){

progressBar.setVisibility(View.*VISIBLE*);

String currentUserId = FirebaseAuth.*getInstance*().getCurrentUser().getEmail();

String text = spinner.getSelectedItem().toString();

DateFormat dateFormat = new SimpleDateFormat("MMM dd, yyyy hh:mm:ss a");

Date date = new Date();

String strDate = dateFormat.format(date).toString();

Bid bid = new Bid(sellerId,text,currentUserId,productId, strDate,imageUrl);

String keyid = mDatabasePlayers.push().getKey();

mDatabasePlayers.child(keyid).setValue(bid)

.addOnSuccessListener(new OnSuccessListener<Void>() {

@Override

public void onSuccess(Void aVoid) {

//Do what you need to do

Toast.*makeText*(Biding.this,"Bid placed!", Toast.*LENGTH\_SHORT*).show();

progressBar.setVisibility(View.*GONE*);

Intent intent = new Intent(Biding.this,BottomNavigation.class);

startActivity(intent);

}

});

}

AAAppendix B

We have used the library braintreepayments to validate the credit card information. It is paypal service which can be used to do visa transactions, PayPal transactions and apple pay transactions also.

Library used:

implementation 'com.braintreepayments:card-form:3.1.1'

Code Used:

final CardForm cardForm = findViewById(R.id.*card\_form*);

cardForm.cardRequired(true)

.expirationRequired(true)

.cvvRequired(true)

.postalCodeRequired(true)

.mobileNumberRequired(true)

.mobileNumberExplanation("SMS is required on this number")

.setup(registration.this);

cardForm.getCvvEditText().setInputType(InputType.*TYPE\_CLASS\_NUMBER* | InputType.*TYPE\_NUMBER\_VARIATION\_PASSWORD*);