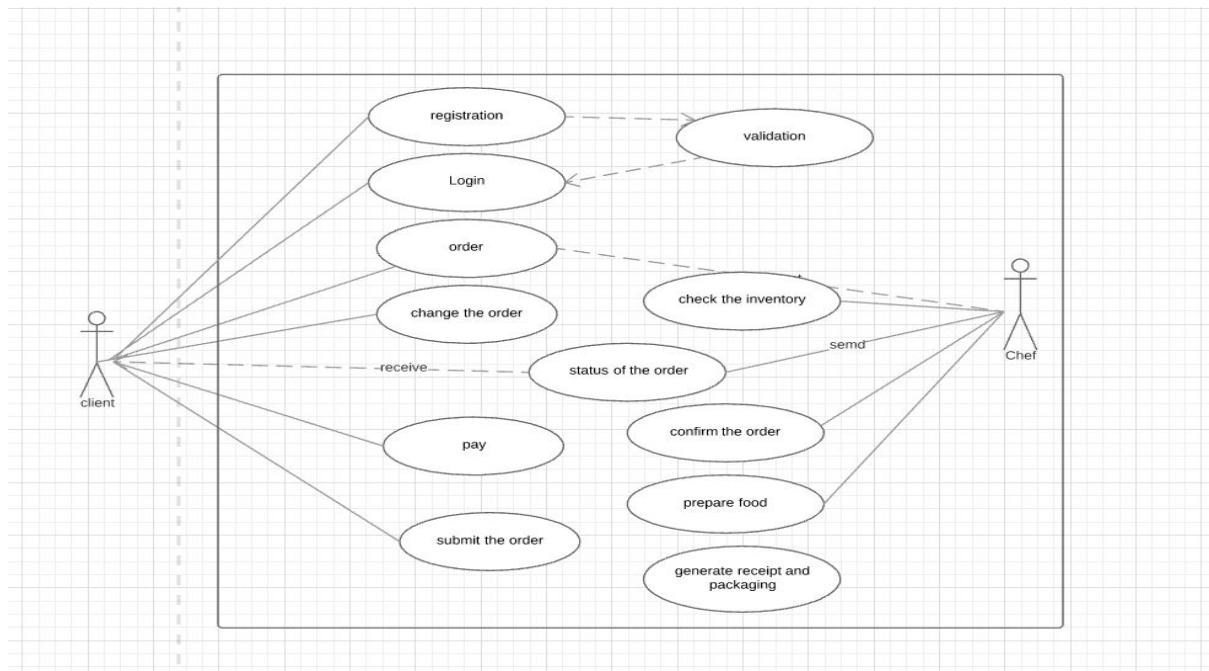
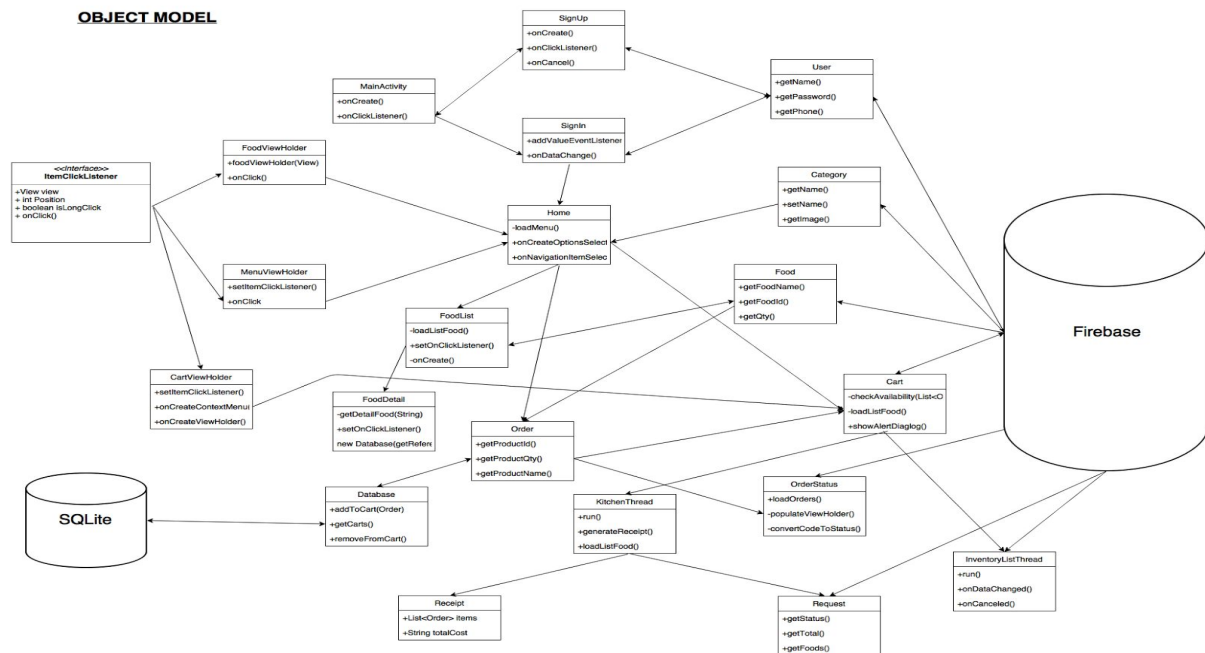


Design:



Sign in and **Sign up**. User can sign up a new account or sign in as an exist account. All user account data will be stored in **firebase**.



Catagory, food detail. Divide the food in three categories: **Burgers, Chicken and sides**. When click on each one, the **food list** will show on the creen. Click on every food item, user can see all **details** and **add it to chart**. All these information will be **synchronized on the firebase**.



Define **kitchen thread** and running in Cart.java, the thread keep managing the synchronized **order list**. Define another thread: **ReadInventory thread**, responsible for updating local **inventory list** from data source of Firebase every hour. Once the Place Order button clicked, update the Inventory list immediately



Create **CheckAvailability** method to check order availability by comparing the **availability FLAG** of the foods between local **inventory list** and order.



After “place order” button clicked, call the Check Availability method and **collect unavailable food information** for later use



Send notification and options(**partial order or cancel order**) to user



According to feedback from user, if this order is partial, add to head of order list; If it is complete order, add to the end of **order list**



In **kitchen thread**, food be cooked and prepared by the **order list's sequence** between 180-300 secs



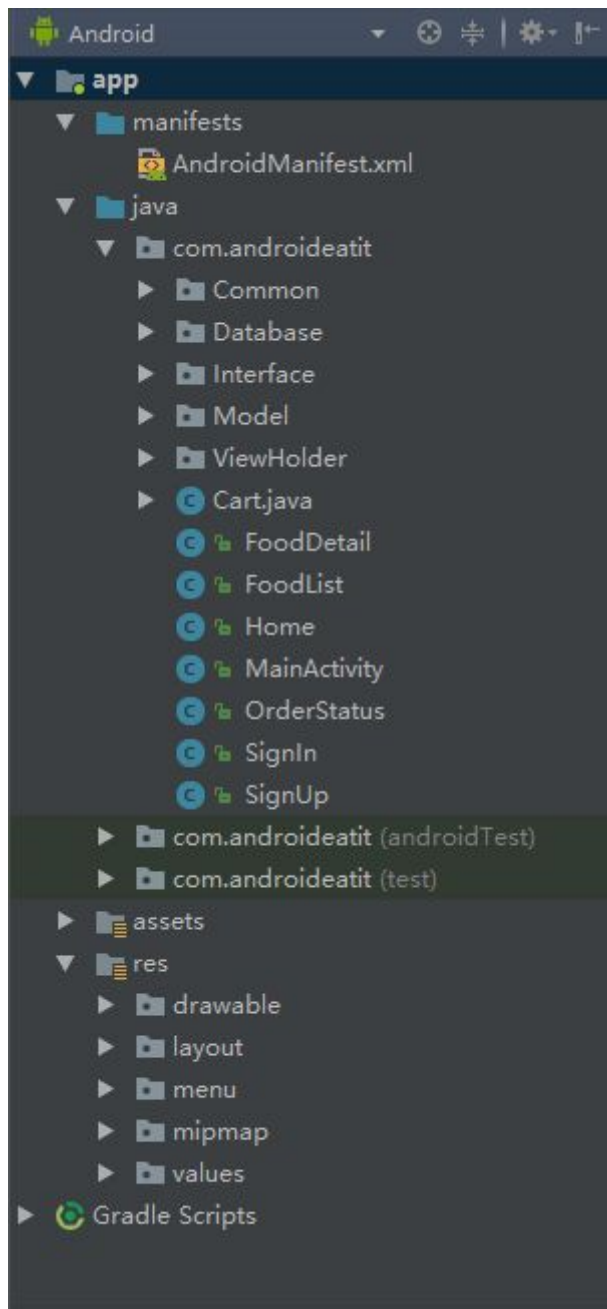
When the head order in order list cooked complete(180-300 secs), cook next order, and **show the finished order in order status activity** for cashier



Generate receipt, send "**Pick up**" notification to User, and change the order status from "placed" to "preparing" to "packaged" to "Food Ready"

Code:

Project Structure:



Test:

We test each part of code and add new features, make sure local project running well, and push to our Github team project. If there were conflicts, we solve it and merge our code together