1. Title Page

Project #3. Implementation

Team 9:

Rakhman Asmatullayev	20180830
Hongxiao Yao	20196360
Umar Taufiqulhakim	20170896
Hyunseung Lim	20170548

Date: Dec 12, 2019

Table of Contents:

2. Implementation

GooStore as a flea market APP, it should be able to provide its users especially foreign students with convenient service and also flexible operation experience. To achieve this goal, our application GooStore is expected to implement users' registration and sign in with their own emails which is the 'Account Login and Creation Function' in our SRS. And also to satisfy some users' realistic needs, the Registration system also needs personal information modification function. What's more, as an application with the function of flea market, the most important part is its trading system. Hence our application supports users the 'Auction Function' which is the core of our APP, and when buyers select the item in the category list, the UI will jump to goods page, and this page will show all information about the goods which is selected by users. And then users can enter their own price to bid with other users and can get this goods after deadline. After the deadline, when buyer clicks the goods in their cart, the UI will jump to 'Payment Function' and notice the payment process. As I have mentioned few sentences former, one of the most outstanding features of our APP is that each user can be either buyer or seller, so there is a function for each user to create their goods for selling and upload it into our database, and then the goods can be shown in the APP and also can be chosen by other users. Other than that, according to the user's duality(buyer and seller), our APP is able to show the condition of their goods either under bidding or selling in their personal page which is called 'My Auction'. Last but not least, to clearer show the goods in our APP, the goods can be shown by their own category.

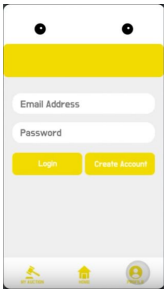
Compared with our SRS, we did not implement some functions in payment system, because of the limited time and resources, we have no idea how to create our own virtual account and as a result, there is no way to truly get money from buyer and send money to seller after buyer clicks the confirm button. And we also could not

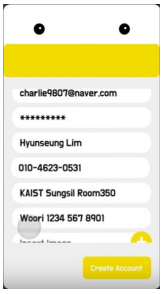
realize the search function. However, the other functions include the database have been already completed.

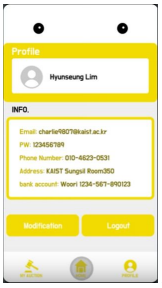
Implementation for our functional requirements in SRS:

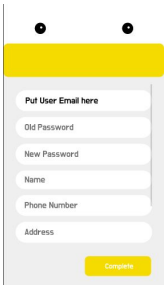
1. **Account Login and Creation Function**

   'The system requires the function that enables users to create their own accounts.' And 'The system requires the function that enables users to authenticate themselves when creating an account (Using email).' Users can click the 'Create Account' button to create their account with some information and also they can set their own images*(.Figure 1)*. And after users click 'Create Account' button, the users' information will be stored in our database and the page will jump to 'Profile'*(.Figure 2)*.The system requires the function that allows users to modify the information in their accounts*(.Figure 3).*





.*Figure 1*


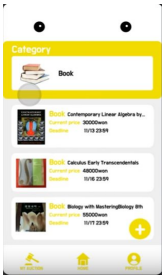
.*Figure 2*


.*Figure 3*

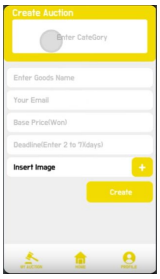
2. **Auction Function**

   'The system requires the function that allows sellers to auction off their good.'*(.Figure 4).* If seller wants to sell something, he or she can click the 'Add' button in *.Figure 4*. And then the page will jump to 'Add Goods' page, then 'The system requires the function that the seller to set reserve price and deadline for the auction.'*(. Figure 5).* After the seller enters all the information of goods including its image and clicks 'Create' button, the goods' information will be stored into our database and there will be some notices appear on the interface.

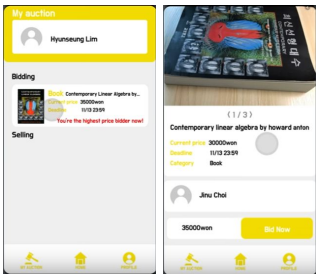
   For 'The system requires the function that enables buyers to search the ongoing auctions.' and 'The system requires the function that enables buyers and sellers to check current price.', we combined them into one function which is 'My Auction', in 'My Auction', the user can check their on selling goods' condition and the bidding they have participated*(. Figure 6)*. What's more, if they click the item in 'My Auction', the interface will jump to the corresponding goods page.



. *Figure 4*

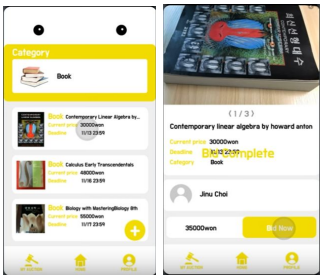


. *Figure 5*



. *Figure 6*

And for the last requirement of Auction System, 'The system requires the function that enables buyers to bid for an auction.' was implemented by the processed as follow: 1. user chooses one item in category list. 2. the interface jumps to corresponding goods' page. 3. enter his or her expected price and click 'Bid Now' button.*(.Figure 7)* And then the price will be uploaded into database as the goods' current price;



. *Figure 7*

3. **Payment System**

As I have mentioned above, because of some realistic factors, the requirements 'The system requires the function that provides virtual accounts that can be deposited by buyer.' and 'The system requires the function that send money to buyer or seller, depending on whether they confirm or some other situations.' could not be implemented, but we have designed their corresponding interfaces and left some ports for them*(.Figure 8)* , and if it is possible for us to further work on it, these defects will be fixed one day.

Then we go back to the payment processes, requirement 'The system requires the function that enables buyers and sellers to set good delivery methods.' is realized in the second part of payment page, once buyer click 'accept' button, his or her money will be sent to our virtual bank account, and then the second step will be available for seller to confirm buyer's some information and choose the delivery method ,and after they send out the goods the third step is valid for buyer to confirm whether they have received the goods and make sure it is in good condition.

Finally, once buyer clicks 'accept' button, money will be sent to seller from our platform, and also if buyer click 'reject' button, the money will be returned to buyer and this deal is automatically closed. Furthermore, the final requirement 'The system requires the function that let both buyer and seller confirm that it is a successful transaction of goods.' is also confirmed by these three steps.

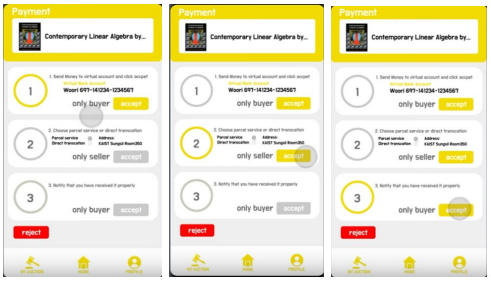


Figure 8

4. **System Address:** https://github.com/Rainhunter13/CS350-Project.git
5. **Demo Video:** **https://www.youtube.com/watch?v=gNNLazrU6Vo**

3. Unit Testing

A. Generated Test Case

R.F.1.1 There is a function that enables user to create an account.

Test case	Expected result	Output
Clicking “create account” button in profile page	Redirected to registration page	True
Filling “email” form with wrong format (i.e no @email.com)	Format error message show up, and user creation failed	True
Filling “password” form with less than 6 characters	Weak password error message show up, and user creation failed	True
Filling “phone” form with characters other than number	Format error message show up, and user creation failed	True
Filling each form with appropriate input and click “create button”	User is created, added into firebase user, and their info added into firebase database	True

R.F.1.3 There is a function that enables user to login and logout of their accounts.

Test case	Expected result	Output
Leaving one of the forms or both field empty and click the “login” button	Empty form error message show up, and login failed	True
Filling “email” form with non-existing email and clicking the “login” button	Invalid user email or password error message show up, and login failed	True
Filling “password” form with unmatching password and clicking the “login” button	Invalid user email or password error message show up, and login failed	True
Filling “email” and “password” forms with	Login success, and profile page showed	True

existing user email and password	up, along with their information	

R.F.1.2 There is a function that enables user to modify their account information.

Test case	Expected result	Output
Clicking “modification” button in profile page	Redirect user to modification page	True
Filling only the field user want to modify and click the “complete” button	Modification success and the user data in database is updated	True
Leaving all fields empty and click the “complete” button	Message notifying to fill at least one field show up and modification failed	True
Filling only “new password” field and click the “complete” button	Message notifying to fill the old password field show up and modification failed	True
Filling “old password” field with the wrong password and click the “complete” button	Message notifying that the old password is wrong show up and modification failed	False (1)
Filling “old password” field correctly and “new password field”, and click the complete button	Modification success, and the user data in firebase user is updated	True

R.F.2.1 There is a function that enables the seller to upload the good they want to sell.

R.F.3.1 There is a function that enables sellers to set the price for the good and appropriate deadline.

Test case	Expected result	Output
Clicking “plus” button in category page	Redirect user to add goods page	True
Leaving at least one of the form blanks and click the “create” button	Message notifying to fill all the box show up and goods creation failed	True
Filling the “deadline” form with non number characters	Message notifying to fill the field with number only show up and goods creation failed	True
Filling the “deadline” form with number less than two or more than seven	Message notifying to fill the deadline again show up and goods creation failed	True
Filling the “price” form with non number characters	Message notifying to fill the field with number only show up and goods creation failed	True
Not attaching an image and click the	Message notifying to attach an image show	True

“create” button	up and goods creation failed	
Filling all the filled correctly, attached an image, and click the “create” button	Goods creation success, and redirected to the main page	True

R.F.2.3 There is a function that enables buyer to see the detailed information about the specific item.

R.F.3.2 There is a function that enables buyers to bid the price for the good.

Test case	Expected result	Output
Clicking a category in the main page	Redirect user to the specific category page	True
Clicking an item displayed in the category page	Redirect user to the item page, and correctly display the item information	True
Enter a bid price that is higher than the current bid price and click “bid now” button	Bidding success, and the price of item increase in both display and database	True
Leaving the bid price field empty and click “bid now” button	Message notifying user to enter the bid price show up and bidding failed	True
Enter non number character in bid price and click “bid now” button	Message notifying user to enter bid price in number show up and bidding failed	True
Enter a lower bid price and click “bid now” button	Message notifying user to enter higher bid show up and bidding failed	True

Additional test cases for myAuction page, where it displays the list of items user sell and bid

Test case	Expected result	Output
Clicking “myAuction” button from any pages	Redirect user to the myAuction page	True
Clicking an item in the list	Redirect user to the item page, and correctly display the item information	False (2)
Creation of the myAuction page	Display the list of items the user sells and bids	False (3)

B. Testing Result and Analysis

For the testing, we made test cases based on the functional requirements written in SDD. For each functional requirements test, we made a few test cases in order to examine how well the application works. As shown in the tables above, most of the test cases' results came out as expected. However, some of the outputs did not come out as expected, which are:

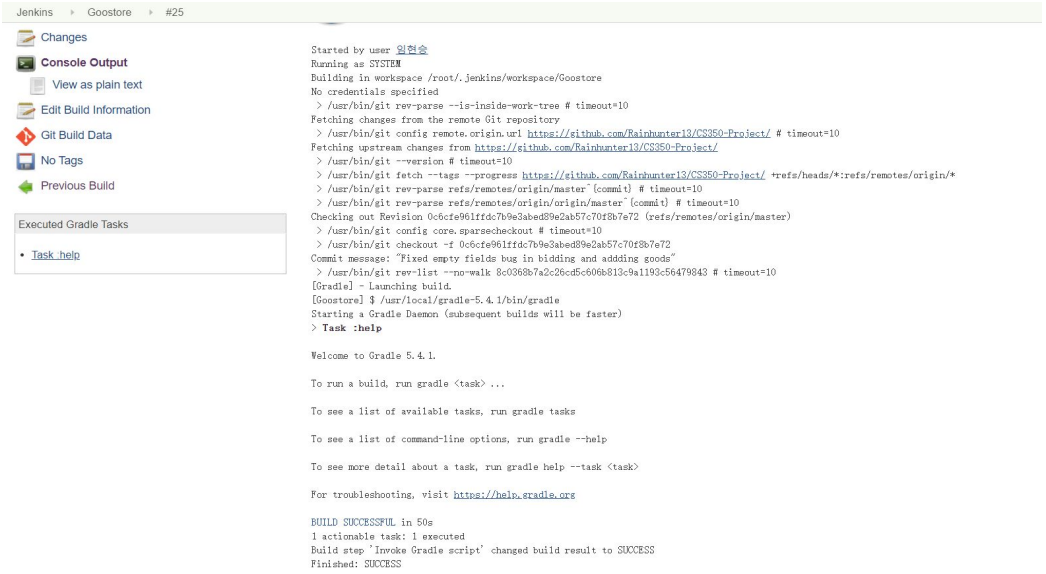
(1) During user info modification, if we put the wrong old password, the message will show that it is wrong, but the app will still proceed and act as if the modification was successful. This is because there is no flag to indicate that the password verification fail. We've tried to fix this, but up until this was written, our effort was to no avail.
(2) If user click one of the items displayed on the list , the user will be directed to the goods page. However, in some cases the image of the good is not properly displayed. This might be due to time taken to load the image from the firebase database.
(3) Once the myAuction page has been created, it should show the list of all items the user bid or sells, However, if the page was open for the first, some items may not be displayed. If we open another page first, and then come back to the myAuction page, it will be displayed properly. The source of this problem is not clear for us, but it was most likely due to the loading from database.

During testing, we often encounter NumberFormatException due to invalid test case, such as empty fields or wrong format. We also found some NullPointerException, usually due to wrong variable name or the variable wasn't initialization. Since both exceptions can cause crashing, creating the test cases and do the testing was a crucial part of this implementation.

However, this testing was only done to functionalities that are already implemented, such as login function and auction function, so functions that hasn't been implemented such as payment function wasn't tested.

4. CI Environment & Static Analysis

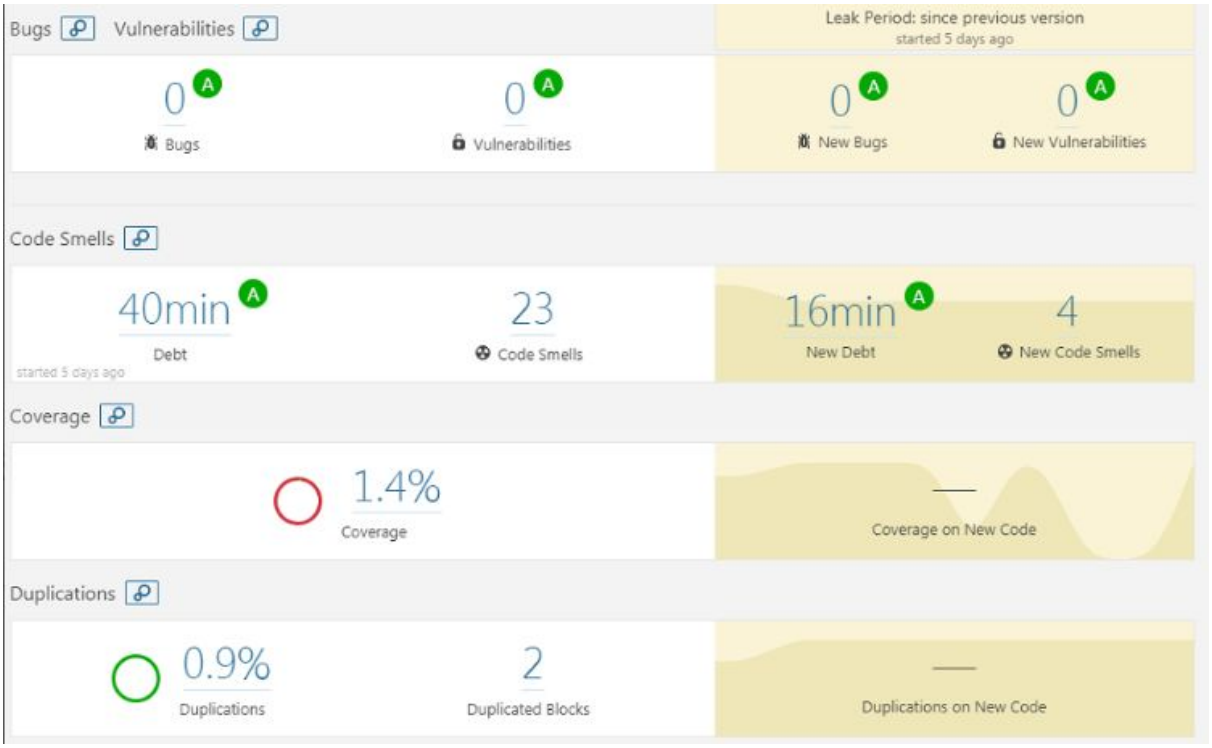
A. CI Environment



We used the gradle and SonarQube modules in our project.

B. Static Analysis & Improvement

We use SonarQube program for static analysis of our app.



Actually, Android Studio has quite good inbuild static analyzer, so every time we committed program analyses code for bugs and warnings. So generally, we were seeing there, rather than using SonarQube constantly.

5. Acknowledgement

This document was done by our team in this partition:

Rakhman Asmatullayev: Static Analysis
Hongxiao Yao: Implementation, CI
Hyunseung Lim: Helped everyone with his part
Umar Taufiqulhakim: Unit Testing