Test Document

for

<IITKART>

Version <1.0>

Prepared by

Group #: Team 9 **Group Name: Matrix**

Akshunya Vijayvargiya	200092	akshunyavijayvargiya@gmail.com
Jasjot Singh	200468	jasjotb02@gmail.com
Kevalkumar Solanki	200991	kevalsolanki49049@gmail.com
Kumar Arpit	200532	kumararpit987@gmail.com
Kushal Gehlot	200541	kushal9427@gmail.com
Mayank	200568	mayankmayank67936@gmail.com
Samarpreet Singh	200848	singh.samar23@gmail.com
Saurav Kumar	200906	sauravsagu@gmail.com
Subodh Kumar	201007	9703subodh@gmail.com
Vaidik Sharma	201079	underscoresharma@gmail.com

Course: CS253

Mentor TA: Swastik Maiti

Date: 4th April 2022

Contents II

Revisions II

1 Introduction 1

2 Unit Testing 2

3 Integration Testing 3

4 System Testing 4

5 Conclusion 5

Appendix A - Group Log 6

1 Introduction

We've tested several functionalities that were implemented in our code. For some of the parts, automation testing was used. The main strategy we followed was top-down approach i.e. for example, we firstly checked the login feature which was the starting page of our app; after that, we tested home, buy and sell features and so on.

Testing was conducted in parallel with the implementation for each feature. For verification as well, we've tested again after a major chunk of the code was completed.

The group members who worked on testing are - Akshunya Vijayvargiya, Kushal Gehlot, Samarpreet Singh, Jasjot Singh and Vaidik Sharma.

Different types of Structural coverage like statement, decision, etc. coverages were used for different sections of testing according to requirements.

For the automation testing part, we had used an inbuilt django module named django.test.

2 Unit Testing

1. test[Register_new_User]

Firstly, the Sign Up button was clicked on the login page, which redirected to the Registration form. Correct entries to specific fields were provided and submitted.

Unit Details: function registerPage

Test Owner: Samarpreet Singh & Akshunya Vijayvargiya

Test Date: 03/24/2022

Test Results:

1. A user with username=”samarpreetsingh”, password=”uthinkuknowme” and email=”samar@samar.com” was created. The test was successful, since the user details were present in the django admin page where the complete user database is there.
2. A user with username=”samar”, password=”samarsamar” and email=”samar@abcd.com” was created. The test was unsuccessful, since the password was very similar to the username.
3. A user with username=”samar”, password=”samar@12345” and email=”samar@ab” was created. The test was unsuccessful, since the email address is invalid without “.com”. For other similar cases as well, this persists. So, we need a valid id.
4. After completing 1, if we create a user with username=”samarpreetsingh”, password=”dummyuser” and email=”samar23@samar.com”, the test would be unsuccessful since the username of each user is unique.
5. In above cases, we assumed to enter password and re-enter password to have same input. However, taking one of them to be “believethat” and other to be “believethis” doesn’t lead to successful registration.

Structural Coverage: In the function ‘registerPage’, only the else branch is accessed during testing. Our test cases have complete Decision/Condition coverage for 3rd if condition of the function ‘registerPage’. Test case 3 covers one of ‘AUTH_PASSWORDS_VALIDATORS’ for password validation.

2. test[login_user]

Firstly, we entered the username and password, and then clicked on the login button.

Unit Details: function loginPage

Test Owner: Samarpreet Singh

Test Date: 03/24/2022

Test Results: Since we’ve already carried out test 1, we consider point 1 user from that test. If we give its username and password as inputs, the test would be successful. However, if we consider the user from point 2, we won’t be able to login (unsuccessful), since those details aren’t already present in the database.

Structural Coverage: Username,password and Login from loginPage have been covered, except the Signup button, which has been tested earlier in unit test 1, so no need to test the same again. Both if and else have been covered.
Additional Comments: After successfully logging in, the user will be redirected to another page which will also be tested.

3. test[Navbar]

These were the test cases-

1) Click on the Buy button.
2) Click on the Sell button.
3) Click on the Logout button.
4) Type 'watch' in the Search bar and click on the Search button.

Unit Details: navbar.html , functions search and searchMatch

Test Owner: Akshunya Vijayvargiya
Test Date: 03/24/2022
Test Results : The results for respective test cases were as follows-

1) Success: User gets redirected to '/buy', the buy page.
2) Success: User gets redirected to '/Sell', the sell page.
3) Success: User gets logged out, and then gets redirected to the login page.
4) Success: User gets redirected to '/search', the search page. The functions 'search' & 'searchMatch' would filter the products containing the string 'watch' in their names or descriptions and then display the same.

Structural Coverage: All the buttons in file 'navbar.html' are tested, that is test cases cover complete statement coverage for this file. Complete Statement coverage in 'search' function and the function 'searchMatch' has complete Statement, Branch and Decision coverage but not complete Condition coverage.

4. test[Sell_Form]

We click on the Sell link or Sell Products link on the homepage, which redirects users to the Sell form. User fills in the details of the product to be sold including name, image, price and description of the item.

Unit Details: class Sell_Create (CreateView)

Test Owner: Kushal Gehlot
Test Date: 03/31/2022
Test Results : User successfully submits the product and redirects to the homepage of the website. The product details get stored in the database which was confirmed by the django admin page.
Structural Coverage: Since we have created a class based view for this form, every field in it has been tested.

Additional Comments: If the user selects incorrect user id then also the form gets submitted and gets displayed on the buy page.

5. test[Delete_Product]

User clicks on the delete buttons and gets redirected to the confirmation page of the delete product. If we select NO, then it is directed to the home page again. Otherwise the product was deleted.

Unit Details: class Sell_Delete (DeleteView)

Test Owner: Saurav Kumar

Test Date: 03/26/2022

Test Results: Product was deleted from the database as confirmed from admin side and was removed from the display of the website.

Structural Coverage: We have used DeleteView, a type of class based view for adding this option. Both the choices available on this page have been covered.

6. test[Update_Product]

User clicks on the update button and gets redirected to the sell form which already has details of current product displayed. User changes the respective details and submit the form.

Unit Details: class Sell_Update (UpdateView)

Test Owner: Kushal Gehlot

Test Date: 03/27/2022

Test Results: The product was updated successfully and the new details were updated in the database by changing all the fields.

Structural Coverage: We have used UpdateView, another type of class based view for adding this function. All the fields of item in test have been updated.

7. test[send_message]

This is a part of the chat feature. It allows interested users to send messages to the owner of the product for enquiring about the same. For testing it, first a message was typed and submitted to the database by clicking the send button or enter (on keyboard). The message appears in the chat room interface. This message is specific to a user and room id.

Unit Details: class Message, function send

Test Owner: Samarpreet Singh

Test Date: 03/28/2022

Test Results: A message “I’m interested in knowing details regarding battery life” was typed for an item named “mobile” with the owner's username being “abc” by user “xyz”. Then upon pressing enter, the exact message appears in the chat room, hence the test is successful.

Structural Coverage: The function ‘send’ doesn’t have any branching, hence it has been covered completely.

Additional Comments: If we just hit enter without typing any message, then a blank message is submitted to the chat room.

8. test[search_bar]

DIfferent inputs were given to the search bar and the available products for that search were displayed. We can search any typical feature of the product which will be displayed if the description of any object contains that feature.

Test cases:

1) Badminton:

   If we search for this, we will get all the badminton racquets available for purchase.

2) Now, in continuation to this if we need a badminton of particular brand say ‘yonex badminton’

   Result : It showed badminton which had yonex in description.

Unit Details: Functions ‘search’ & ‘searchMatch’

Test Owner: Kushal Gehlot & Akshunya Vijayvargiya.

Test Date: 04/03/2022

Test Results: Search results are displayed in the same way as buy pages along with an interesting option for chat.

Structural Coverage: Complete statement coverage in ‘search’ function and the function ‘searchMatch’ has complete Statement, Branch and Decision/Condition coverage.

9. test[receive_message]

This is a follow up to test 7. Upon sending the message, the chat feature makes sure that message is received to the owner/user. This functionality achieves this by showing the entire chat history.

Unit Details: class Message, function getMessages

Test Owner: Samarpreet Singh

Test Date: 04/01/2022

Test Results: After conducting the test given in test[message_sent], “abc” replies “Battery life is approx 10 hours”. The message appears in the chat room which is common to all users. The given chat room is unique to the product.

Structural Coverage: The function ‘getMessages’ doesn’t have any branching, hence it has been covered completely.

10. test[chat_from_detail_page]

This is regarding the redirection from detail page to chat page. Detail page refers to the page which contains complete information regarding the product.

Unit Details: class Sell_Details(DetailView), sell_detail.html

Test Owner: Samarpreet Singh

Test Date: 2nd April, 2022

Test Results: The 'interested' button on this page should open the chat room, however it doesn't function. Hence, the test for link to chat is unsuccessful.

Structural Coverage: Since the test case failed, we couldn't check the functioning initially.

Additional Comments: We have modified the code to fulfil the requirement. It is working now.

3 Integration Testing

1. User Authentication

The user registration, along with login and logout functionality was tested. First, the user registers and creates a new account. Then, they are directed to the login page for entering the website.

Module Details: User_Authenticate
Test Owner: Jasjot Singh
Test Date: [04/01/2022] - [04/02/2022]
Test Results: The account is created when the user registers, and the same is added to the database. This data is then fetched from the database when the user tries to login and the verification process happens. They also have the option to logout at any time and that happens successfully.

2. Selling a product and buy page updation

The user has to fill a form for putting a product for sale on the website by entering its details such as name, price, information, and image. After the product is uploaded on the website, the same gets reflected in the database along with the seller name. The same product is then also visible on the general buy page for everyone. Also, the user can see their own product and can make further changes or modifications.

Module Details: Sell_Buy_Update
Test Owner: Jasjot Singh
Test Date: [04/01/2022] - [04/02/2022]
Test Results: The user successfully uploads the product for sale and the same is reflected on the buy page as well as the homepage of the user for any update on it.

3. Deleting product

The user also has a choice to delete any product that they had earlier uploaded for sale. After the delete option is used, the product is removed from the database, the buy page and the homepage for the user

Module Details: Del_Product
Test Owner: Jasjot Singh
Test Date: [04/01/2022] - [04/02/2022]
Test Results: The user can successfully delete the product and the change is reflected in the database which contains product information. Also, the product icon is removed from the buy page catalog and the homepage.

4. Update product information

Even after the product has been uploaded, the user has the option to update its details from the homepage. They are redirected to another page for updating its information and the changes are then applied everywhere.

Module Details: Update_Prod_Info

Test Owner: Jasjot Singh

Test Date: [04/01/2022] - [04/02/2022]

Test Results: The user updates the product info and then the changes can be seen in the database as well as the buy page and the homepage.

5. Chat Room

The chat room allows an interested buyer and the seller to communicate via a chat interface. Upon arriving to the chat room, the message once sent, arrives to the database and is also visible to the seller and other people interested in the same product. The message is tied to a room and user ID. Also, it is made sure that the user receives the message via the chat history.

Module Details: Chat_Room

Test Owner: Jasjot Singh

Test Date: [04/01/2022] - [04/02/2022]

Test Results: The interested user clicks on the ‘interested’ button on the product page and is then redirected to the chat room. The message is typed in the box and when the user clicks the ‘send’ button, it is reflected in the chat history for every interested buyer and the seller to be seen and the same is also a part of the database.

4 System Testing

FUNCTIONAL REQUIREMENT TESTING

1. PRODUCT DISPLAY

All the items available to be sold must be shown to the users.
Test Owner: Vaidik Sharma
Test Date: 04/03/2022
Test Results: All the items uploaded by the seller were visible on the buy page of the website without any exception to all the users.

2. COMMENT/ CHAT

Buyer side must be able to send messages privately to the seller so that they can negotiate and come to a conclusion over the deal.
Test Owner: Vaidik Sharma
Test Date: 04/03/2022
Test Results: Seller and the interested users were able to text each other on the Chat. Other interested users are open to see the previous chats as comments so as to stay updated on the ongoing negotiations.
Additional Comments:Further update needed if chats are supposed to be end to end encrypted.

3. UPDATING PRODUCT INFORMATION

The Seller side must be able to update the details of that particular item which is to be sold after if needed.
Test Owner: Vaidik Sharma
Test Date: 04/03/2022
Test Results: The Seller is able to update his/her product's information at any time. And the updated information is instantly updated everywhere with no delay.
Additional Comments: Notification regarding updated information on interested items can be added.

4. LOGIN

Every user must be allowed to login only through his iitk.ac.in domain email id.
Test Owner: Vaidik Sharma
Test Date: 04/03/2022
Test Results: This is a failed requirement.Users are able to register using any valid mail id but not specifically IITK mail id.
Additional Comments: This field is left for future update.

HARDWARE REQUIREMENT TESTING

ACCESSIBILITY ON HARDWARE

Since the application must run over the internet, all the hardware shall be required to connect to the internet, which will be the hardware interface for the system. As for e.g. Wi-Fi, Ethernet Cross-Cable, etc. Any smartphone or PC with proper internet connection will serve the purpose.
Test Owner: Vaidik Sharma
Test Date: 04/03/2022
Test Results: The website was running smoothly on all hardwares including PC, Tablet, Phone(ios + android). The only requirement is a stable connection with the internet.

5 Conclusion

How Effective and exhaustive was the testing?

The testing was overall an effective procedure which involved manual as well as automated testing for unit tests. It was exhaustive as well as we've tried to completely cover functional coverage and the structural coverage in the testing procedure. Giving a figure, it was about 90% exhaustive in coverage.

Which components have not been tested adequately?

The chat room is one of the functionalities where we faced some difficulty in adequately testing.

What difficulties have you faced during testing?

The manual testing took a lot of time as we were trying to maintain a high coverage rate for all the different testing systems. We did switch eventually to automated testing for the unit test cases.

How could the testing process be improved?

The testing process could have been improved if we had tried to implement automated testing at an earlier stage which would have helped to get done the integration testing and the system testing at a faster pace and achieve exhaustive coverage.

Appendix A - Group Log

We all were at campus and discussed how we can test our document. Regular meetings and discussions were conducted in RM building and hostel rooms.

20th March 2022 - Some editing in implementation and work distribution for various components of testing and documentation.

25th March 2022 - List of manual testing cases was decided. Unit testing was started by various group members in different components of the software.

29th March 2022 - Automated testing classes were implemented for some cases by one of the group members.

2nd April 2022 - Unit and Integration testing were completed.

3rd April 2022 - System testing was completed.

4th April 2022- Documentation was completed.