# Chapter 8: Testing

## 8.1 Chapter Overview

In this chapter the author will be skimming through the testing process that has been involved during the period of implementing the proposed thesis. The author will cover the information about the objectives and goals, functional testing, non-functional testing, and integration testing.

## 8.2 Objectives and Goals of Testing

The main objective and goal of the testing process is to make sure that the build software by the author is based on the proposed requirement to validate the following these.

The following testing goals that the author must achieve are as follows:

* Ensuring that the recommendation model is working according and the content-based filtering technique.
* Validating if the code written by the author as followed the best practices.
* Making sure that the errors are being handled properly, so that it wont crash the system to the end user.
* The system should satisfy the MoSCoW technique pattern mandatory "Must have" and "Should have" of the functional requirements.
* Another goal is to make sure the non-functional requirement is appropriately identified and satisfied the system.

## 8.3 Testing Criteria

The test plan for the Salary recommendation system can be categorized in 2 different categories.

* Functional Testing
* Non-Functional Testing

For each of the prototype's functional and non-functional needs, test cases were written independently to test the application. Hence this test cases will help to determine if the implemented system is functioning as intended.

## 8.4 Model Testing

According to the proposed system by the author, the Salary recommendation system was done in a hybrid approach. The model that was built by the author was a Random Forest Classifier, the other was a most common technique in recommendation system which is the Content-Based filtering approach. Moreover, other classification models were also used by the author to compare what would be the best fit model for the proposed approach.

|  |  |
| --- | --- |
| Model | Testing approach |
| Random Forest Classifier | Accuracy, Precision, Recall, F1 score |
| Support Vector Machine | Accuracy |
| Decision Tree Classifier | Accuracy |
| k Neighbors Classifier | Accuracy |
| Content Based filtering | Cosine Similarities |

### 8.4.1 Models accuracy

Chart

Description automatically generatedThere were different models that were used by the author to make the recommendation system and each model that has been built had given different results as depicted in below diagram.

As the above diagram were the model accuracy results, which was used to get the best possible salary for the relevant user input given to the model.

|  |  |
| --- | --- |
| Model | Accuracy |
| Random Forest Classifier | 56.5% |
| Support Vector Machine | 5.3% |
| Decision Tree Classifier | 56.4% |
| k Neighbors Classifier | 16.5% |

### 8.4.2 F1 Score

To evaluate the performance of the model, the F1 score was used. It is a metric that takes int account both precision and recall. The model which was most suitable was the Random Forest Classifier which returned an F1 score of 0.55. This indicates that the model has moderate performance in predicting the target variables.

### 8.4.3 Precision

According to calculations, the precision score, which expresses the percentage of accurate positive predictions among all positive predictions generated by the model, was 0.73. This indicates that 73% of the time, the model's predictions of positive situations were accurate. In applications where false positives are expensive, a high precision score is preferred. The accuracy score in this investigation shows that the model is successful at locating positive cases with a relatively low percentage of false positives.

## 8.5 Benchmarking

## 8.6 Functional Testing

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test case | FR ID | User Action | Expected Result | Actual Result | Result Status |
| 1 | FR01 | Users can enter salary information details. | All the inputs should be filled else the error message should be shown. | Error message has been shown to the end user. | Passed |
| 2 | FR02 | User inputs data to get the salary recommendation data based.  on the parameters that the user has pooled into the system. | Show the recommendation data for the user input passed. | The most predictable data has been shown to the user | Passed |
| 3 | FR03 | Saving the recommended data based on the user input into the database. | Run on another thread and save the data to the database. | Saved the data to the database by using another thread. | Passed |
| 4 | FR04 | Saving the salary information provided by the information user to the database. | Saving the data to the database. | Saved the data to the database. | Passed |
| 5 | FR05 | Recommending relevant salary data for the user’s entered inputs. | Show result from the model and content-based filtering | Displaying the result relevant to the user input from model and content-based filtering. | Passed |
| 6 | FR06 | The system allows the user to give ratings to the recommended salary data. | Show the edit button for the rating to the user to give a score for the salary that is displayed. | Show the edit button for the rating to the user to give a score for the salary that is displayed. | Passed |
| 7 | FR07 | The system shows error messages for inappropriate use of the system. | Display error messages | Display error messages | Passed |
| 8 | FR11 | Adjusting the relevant salary recommended salary to the latest inflation rate. | Update the salary while saving to the current inflation if the year payment is before the current year. | Update the salary while saving to the current inflation if the year payment is before the current year. | Failed |
| 9 | FR12 | Update the database after the User has given the score. | Updating the rating column for the relevant salary. | Updating the rating column for the relevant salary. | Passed |
| 10 | FR13 | Sort the content-based filtering according to the rating given to the salary | Sorting according to the rating given to the salary. | Sorting according to the rating given to the salary. | Passed |

## 8.7 Module and Integration Testing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Module | Input | Expected Output | Actual Output | Status |
| Input New Salary Information for the system. | Do validation if all the inputs have been entered. | Display error message saying to fill all the important inputs. | Displayed error message saying to fill all the important inputs. | Passed |
| Input fields are entered with the datatype expected. | Display error message informing to enter correct input value. | Displayed error message informing to enter correct input value. | Passed |
| Save the Data after clicking on the Submit button | Display success message and save the data to the database. | Displayed success message and save the data to the database. | Passed |
| Get recommendation for the relevant User inputs. | Do input validation | Display error message | Displayed error message | Passed |
| Input data to get salary recommendation | Show recommended salary for relevant input. | Showed recommended salary for relevant input. |
| Save the recommended data to the database. | Saving to the database with the recommended data. | Saved data to the database |
| Update Score for the Salary | Once clicking on the save button the score should be updated to the relevant salary amount. | Show success message that the rating is updated. | Showed the relevant success message. | Passed |
| Once click on the cancel button the score shouldn’t be updated. | Show success message that the rating is cancelled to updated. | Showed the relevant message. |

## 8.8 Non-Functional Testing

There are few non-functional requirements identified by the author for the proposed system, and which has been depicted in the SRS chapter. According to this chapter the non-functional requirements will be discussed in detail. The accuracy testing of the model is explained under the Model accuracy.

### 8.8.1 Performance Testing

The model performance testing was done by using the F1 score, precision and it was explained in the above chapter. As a result, in this topic the author will be doing showing the performance between the integration to get the recommended salary for the given user input, but however the performance testing was done in local, which could differ if the deployment has done to the cloud.

Graphical user interface, application, table, Excel

Description automatically generated

The diagram depicts the time scale to get the salary recommendation for the user input that has been passed to the backend server.

Graphical user interface

Description automatically generated with low confidence

The above diagram shows the amount of time that has been taken whilst requesting the backend server to get the recommendation.

### 8.8.3 Security Testing

The security of the system is maintained by not collection information about the user profile, on who is entering the salary and the salary information the user inputs to the system doesn’t save the information of the company that an individual is working on, therefore there has been no violation of collecting personal information that could be used for false interpretation in the future.

### 8.8.4 Usability Testing

The ease with which users can interact and navigate through a user interface is a key indicator of an application's usability. This system was developed with simplicity as a top priority to improve its general use. The screen shot of the application could be referred in the implementation chapter.

### 8.8.5 Maintainability Testing

The backend, frontend as well as the model implementation has been properly modularized so that it would be very much easy to understand and skim through how to rebuild the system above for the future scope of the proposed project.

## 8.9 Limitations of testing process

The limitation of the testing process could be the Non-functional testing. Since the system has not been deployed to cloud, the application will be running in the local, therefore it will show a better performance due to only one user could be able use the application at a time, therefore the responses and all other testing will be very high due to low scalability.

## 8.10 Chapter Summary

## 