SOFTWARE TEST PLAN

Recommender System

SUBMITTED BY :
IIM2016003
ITM2016004
IRM2016002
ISM2016006

TABLE OF CONTENTS

1. INTRODUCTION

This document will address the different standards that will apply to the unit, integration and system testing of the specified application.

2. OBJECTIVE

Objective of Test plan is to define the various Testing strategies and testing tools used for complete Testing life cycle of this project.

3. SCOPE

The document mainly targets the Front end testing and validating data in report output as per Requirements Specifications provided by Users.

4. TESTING PROCESS OVERVIEW

Test Process :

- Understanding of requirements from the requirements specifications.
- Preparing test cases based on the requirement specifications. This will cover all scenarios for requirements.
- Reviewing test cases.
- Creating test data based on Test scenarios and Test cases.
- Executing Test Cases
- Update test result (Actual Result, Pass/Fail) in test case document.

5. TESTING STRATEGY

Software testing is a process of running with intent of finding errors in software. Software testing assures the quality of software and represents final review of other phases of software like specification, design, code generation etc.

5.1 GUI Testing

(i). POSITION OF IMAGES AND TEXT ON SCREEN -

Test Id - TESTG01
Assumptions - Options for text fields
Purpose - To check that the data being input is mandatory fields and is in correct format.
Steps -
1. Verify that there should be a warning if the mandatory fields are not filled.
2. Verify that width of data entered does not exceed allowed width.
Expected Outcomes - Positioning of images should be correct
Actual Outcomes - images and text displayed correctly
Status – Pass

(i). Proper Rating as per selection-
Test Id - TESTG02
Assumptions – Rating is given by the user.
Purpose - To check that stars are highlighted as per intended rating.
Steps -
1. Verify that the clicking on stars highlights them as per expected rating.
Expected Outcomes – Proper rating should be highlighted
Actual Outcomes - images and text displayed correctly
Status - Pass

5.2 Unit Testing

Unit testing is a dynamic method for verification, where program is actually compiled and executed. Unit testing is performed in parallel with the coding phase. Unit testing tests units or modules not the whole software.

Each of the modules was tested as separate units. In each module all the functionalities were tested in isolation.

1. Test Module

(i). ITEM BASED COLLABORATIVE FILTERING -
Test Id - TESTU01
Assumptions – Movie is clicked upon by the user.
Purpose - To check if appropriate movies are shown as per movie selection.
Steps -
1. Verify that the movies displayed resemble in description or timeline.
2. To check viable resemblance in genre.

Expected Outcome – Appropriate movies are shown.
Actual Outcome – Resemblance in movies recommended.
Status - Pass.

5.3 Functional Testing

(i). VIEW PROFILE-
Test Id - TESTU03
Assumptions - Option for viewing one's profile.
Purpose - A logged in user is able to view his watch list.
Steps -
1. Verify that the user sees his watch list upfront.
2. Verify that the ratings of the movies in watch list is shown.
Expected Outcome - User can view his watch list and ratings.
Actual Outcome - The user is able to view his watch list and their respective ratings.
Status - Pass.

(ii). UNRATED MOVIES ARE ALLOWED TO BE RATED -
Test Id - TESTU04
Assumptions - .The user has a rating to give.
Purpose - To check that the user is able to provide ratings for unrated movies.
Steps -
1. Verify that there should be an option for rating unrated movies.
2. Verify that user's needed rating is given.
Expected Outcomes - The ratings are properly fed.
Actual Outcome – The new ratings are fed.
Status - Pass.

(iii). CUSTOM NUMBER (k) OF COLLABORATIVE FILTERED ITEMS
Test Id - TESTU05
Assumptions - .The user provides the number of items he/she wants to view.
Purpose - To check that the user is shown k items.
Steps -
1. Verify that if k number of items are asked for k items are shown.
2. Verify that user's request is met.
Expected Outcomes - The k items are only shown.
Actual Outcome – The k valid items are shown.
Status - Pass.

5.4 Integration Testing

In integration testing a system consisting of different modules is tested for problems arising from component interaction. Integration testing should be developed from the system specification. Firstly, a minimum configuration must be integrated and tested.

In our project we have done integration testing in a bottom up fashion i.e. in this project I have started construction and testing with atomic modules. After unit testing the modules are integrated one by one and then tested the system for problems like following arising from component interaction.

(i) RATING FED IN FRONTEND IS RECEIVED IN THE DATASET -
Test Id - TESTI01
Assumptions – A suitable rating is provided by the user.
Purpose - To check that once the rating is fed the same is updated in the dataset.
Steps -
1. Verify that the user is always providing valid ratings.
3. Verify that the uploaded dataset file is being stored in our database.
Expected Outcome – The correct rating is fed and received.
Actual Outcome - The expected rating is fed and seen in the database.
Status – Pass.

(i) INTEGRATION OF FLASK WITH SPARK-

Test Id - TESTI02
Assumptions – Valid data in RDD is provided.
Purpose - To check that the data in RDD is properly converted to JSON.
Steps -
1. Verify that the valid data is fed in RDD.
3. Verify that the JSON data is matching with input to RDD.
Expected Outcome – The correct data is received.
Actual Outcome - The data passed matches the final data in JSON.
Status - Pass.

5.5 Security Testing

(i).CONFIDENTIALITY OF DATA -

Test Id - TESTSE01
Assumptions - Use of an encryption mechanism
Purpose - prevent unauthorized access to sensitive data and protected data from harmful intruders.
Steps - Try accessing the channel through which the data flows to-and-fro the web browser and the server.
Expected Outcomes - Come up with solutions to mitigate security issues related to data confidentiality.

(ii). PASSWORD POLICY REVIEW / PASSWORD SECURITY TESTING -

Test Id - TESTSE02
Assumptions - Uses a password in encrypted format. Requires user authentication.
Purpose - To prevent intruders from cracking user passwords/ defeating a cipher or authentication mechanism. To prevent a password guessing attack.
Steps -
1. Use easy, non-complex passwords.
2. Use a brute force attack, dictionary attack to try to guess the password.
3. Retry attacks from different IP addresses (in case the website blocks the IP address for numerous password guessing attempts)
Expected Outcomes - Clarity about the strength of password policy review. Make necessary amends.

(iii). USER AUTHENTICATION AND SESSION MANAGEMENT-

Test Id - TESTSE03
Assumptions - Authentication required in order to verify the digital identity of the sender of a communication. Restricted access by authorized users (or API) only.
Purpose - To perform vulnerability scanning and detect issues regarding authentication and session management.
Steps -

1. Check whether user authentication credentials are protected when stored using hashing or encryption.
2. Check that the session IDs aren't exposed.
3. Perform session fixation attacks to check vulnerability against session management.
4. Check whether session IDs are rotated after successful login.
5. To ensure that passwords, session IDs and other credentials are sent over encrypted channel.

Expected Outcomes - Review of user authentication and session management.

(iv). CRYPTOGRAPHY -
Test Id - TESTSE04
Assumptions - Use of encryption mechanism.
Purpose - Validate the cryptographic mechanism used.
Steps -

1. Test for path traversal
2. Test for vertical access control problems (a.k.a. privilege escalation)
3. Test for horizontal access control problems (between two users at the same privilege level)
4. Test for missing authorization
5. Test for insecure direct object references

Expected Outcomes - Review of cryptography

5.6 System Testing

System testing means testing the system as a whole. All the modules/components are integrated in order to verify if the system works as expected or not.

System testing is done after integration testing. This plays an important role in delivering a high-quality product.

(i). LAUNCHING WEB APP -
Test Id - TESTSY01
Assumptions - The code for frontend and backend integrates well.
Purpose - To verify that the web app launches properly.
Steps -
1. Verify that the server has been started.
2. Verify that the web app is launched successfully and all the gui components of the page are visible to user.
Expected Outcome - The web app opens properly and user is able to see all the gui components.
Actual Outcome - The web app launches successfully when the server is operational.

Status - Pass.

(ii). USER REGISTER AND LOGIN -
Test Id - TESTSY02
Assumptions - The web app manages user session properly.
Purpose - To verify that the user is able to register and login properly .
Steps -
1. Verify that the server has been started.
2. Verify that the user is not able to login with wrong credentials.
3. Verify that the user is able to register if his email is existential and he sets his password which include at least 7 characters, an uppercase letter and a digit (0-9).
4. Verify that the information of the user are stored in a database.
Expected Outcome - The user can login and register so that he can use the utility of viewing his past test runs and use default dataset.
Actual Outcome - Login and register feature working successfully.
Status – Pass.

5.7 Most Critical Test Case

Validation of predicted rating
Test Id - TESTM01
Assumptions – The software has shown the predicted rating.
Purpose - To check that the data being predicted is relevant and upto the user's expecttion
Steps -
1. Open the already rted movie.
2. It will show the predicted rating.
3. Compare it with the rating the user has already given.
Expected Outcomes – the predicted and actual ratings are comparable
Actual Outcomes – the difference between the actual and predicted ratings is averagely within 0.8.
Status – Pass

6. TOOLS -

No tools were used because only manual testing was performed.