Software Testing Plan

Music Recommender

By

Abhimanyu Olla 16BCE0928

Sidarth Wadhwa 17BCE0168

Shraveen BS

Introduction

The purpose of the project is to provide a music recommendation system that works based on both collaborative and content-based filtering techniques, drawing on the suggestions arrived at by both these techniques and stitching them together to provide an optimum result. The results will be used to provide users with suggestions for songs that they might be interested in.

Test Items

- Database quality
- Collaborative filtering performance
- Content-based filtering performance
- Recommendations provided
- Evaluation quality

Features to be Tested

- The recognitions capability
- User adaptability
- Response time
- Uninterrupted usability

Features Not to be Tested

- We will not be
- The database featured are flexible and will vary largely depending upon the application and thus, will not be tested extensively.
- The actual execution speeds will depend upon a large number of variables and hence, cannot be ascertained in a testing environment.

Approach

V-model SDLC is being used. In this case, the development and testing of the product can be done simultaneously. This model will save time and defects can be found at an earlier stage in the development and testing process.

Pass/Fail Criteria

The prototype will be passed if it is able to successfully extrapolate the user's potential likings based on their listening history as well as the and mark the corresponding student's attendance in the respective database.

The prototype will be failed if it is unable to perform the above mentioned task within an acceptable amount of time.

Suspension Criteria

- User's listening history is outside the purview of the model.
- If the users location data is unavailable for collaborative filtering .
- If model's functionality is not being used appropriately.

Test Deliverables

The tests will be conducted in house and the resultant quality control certificate shall serve as an assurance of quality and dependability of the app, when it is used in the correct manner under specified conditions.

Testing Tasks

Testing environment must ready prior to the testing task. Test summary needs to be prepared.

- At least 50 different users must be tested.
- The application must be tested on Ubuntu, Windows, Android and MacOS.

Environmental Needs

The test will be conducted in an optimised environment with low test case dilution. The database will be considered optimised and ready to receive data as soon as the scanner scans it. The wireless speeds will be subject to change within the margins of average corporate wireless speeds.

Responsibilities

Test Plan must be prepared by the software developer and the testing must be done by the software documenters .

Staffing and Training Needs

The listeners will be provided with a basic tutorial about the usage of the app. Most of these features will be available in predefined functions that drive these processes. The users will not require any special skills that could not be expected of an average listener.

Schedule

- Collaborative filtering -> 10 man-hours.
- Content based filtering -> 10 man-hours
- System testing ->15 man-hours
- Test summary -> less than 3 man-hours

Risks and Contingencies

The wireless connections might be prone to attacks and protection against these will be incumbent upon the client. Private connections are recommended to help prevent such attacks.

It will be critical for the connection format to be convenient and accessible to all clients. If there are any interruptions in the network, resultant drops of data frames cannot be chalked up to a problem in the software.

Approvals

The software developer and the project managers must approve the project after testing. Finally, the faculty responsible for providing this opportunity must approve of the project as a genuine and lucid project venture.