**Group 46: Project Acceptance Tests**

**Objectives**

This document outlines a collection of tests for the Virtual Customer Service conversational AI system. 3 Tests are outlined in detail below, all of which if realized will confirm that the project has been a success and achieved its desired functionality. Note that this document may require adjustment as the client changes their standards for what an acceptably functional project requires.

**Document References**

* Problem statement/Project Description
* Scope of Work
* Risk Register
* Set of User stories

**Test Summary**

The tests described focus on verifying the functionality of the conversational AI system as outlined by the client’s problem statement and further descriptions of the project’s aims via an initial meeting with our team. Data ingestion, error detection and user interaction are the functions tested and outlined. It is worth noting also that integration with the client’s existing solution (their current LIMS) may be a factor.

**Testing Strategy**

The 3 subsystems to be tested include the Data ingestion module, Error Detection module and the User interface.

The ingestion of data in the appropriate format/s and the ability of the AI to detect the errors in formatting of the given documents will be tested via rigorous automated unit testing where the system will need to handle increasingly complicated scenarios.

The User interface will be tested through a staging environment that closely resembles the real production environment, through this it can be verified that our client and the project’s intended users find the UI to be suitably intuitive and visually appealing.

**Test A: Data Ingestion**

This initial test focuses on the ingestion of the .csv files into the conversational AI system. It ensures that the user’s data submission is parsed and read appropriately for future use. The system requires flexibility and robustness in its data ingestion.

**Test Specification**

The test will use a variety of .csv files as well as non .csv files and will score the system based on if it throws the correct errors and/or correctly parses the given document/s.

**Test Description**

* Means of Control: .csv files and non .csv files are entered through an automated testing script
* Data
  + Input Data: Diverse range of .csv files and some non .csv files
  + Input Commands: Commands to upload, ‘submit file’
  + Output Data: Parsed data structures and error messages, a score measuring the systems parsing accuracy
  + System Messages: Confirmation or Error messages i.e. ‘incorrect file type given’
* Procedures: Script automates input commands and compares them to a predefined expected result, files being either successfully and correctly parsed or correctly omitted increases the score for this test.

**Test Analysis Report**

The comparison of expected versus actual outputs serves as the primary data measure. The system should undoubtedly pass this type of testing with 100% accuracy by its final version. During production if any errors are found the test case should notify the developer verbosely such that it can be addressed. Issues like the user attempting to parse a non .csv file are most notable.

**Test B****: Error Detection**

This test evaluates the systems ability to detect errors in the .csv files once they have been parsed and ingested.

**Test Specification**

The test will use a variety of .csv all with a wide range of formatting errors and will score the system based on if it identifies these formatting errors sufficiently

**Test Description**

* Means of Control: Data is entered manually to create the error ridden .csv files, the .csv files are automatically received by the system once the test case begins
* Data
  + Input Data: .csv file/s with formatting errors such as Whitespace added, incorrect headers, Wrong lines and missing sample ID’s
  + Input Commands: ‘Begin test’
  + Output Data: A list of errors successfully identified, a score for the test case measuring the systems ability to detect errors.
  + System Messages: ‘Test successfully completed’, ‘Failure closing’
* Procedures: The test procedure records system responses and scores it’s error detecting ability

**Test Analysis Report**

The test report should identify and list all the formatting or other errors it successfully identified within the .csv file/s as well as providing a list of all the test case errors it failed to find. This test should achieve complete success with no unidentified errors from a wide range of given cases by the end of production.

**Test C: User Interface**

This test focuses on the usability and visual appeal of the systems UI, most importantly the end user should be able to intuitively use the system with no prior knowledge of computing.

**Test Specification**

Test users are presented with the conversational AI and asked to use it to upload a variety of .csv files through it in a staging environment. At the end of the experience the user will provide scores ranging from 0-10 about their opinion of the systems usability and visual appeal. The user is then asked if they would be happy to use our system on a regular basis, ‘yes/no’ and asked to provide any additional feedback on their experience.

**Test Description**

* Means of Control: Given .csv files are uploaded manually by the user via the system, The users feedback/scores on the experience are recorded manually by a supervising team member
* Data
  + Input Data: Test users’ name, test .csv files
  + Input Commands: \*
  + Output Data: Users’ feedback and scores
  + System Messages: \*
* Procedures: The test procedure simulates how the system will be used by typical users in a live final production environment.

**Test Analysis Report**

The final test report will be a list of test users with their scores out of 10 on usability and visual appeal and yes/no answer if they would be happy to use the product in the future. The score on usability is most important and the results of this test should satisfy the client in the systems final rendition, a score of 7/10 or higher for usability is an appropriate target. Any additional user feedback provided can also be a very valuable and should be noted and suggestions potentially adopted.

**Test Materials**

* Wide range of .csv files containing a myriad of different errors such as ‘Whitespace added, incorrect headers, Wrong lines and missing sample ID’s’
* Range of files that are not .csv files to test incompatibility
* Some .csv files that do not contain errors
* Sufficient computing ability to run tests A and B
* A range of diverse test subjects with no prior knowledge of the project
* Space and appropriate documentation to store the results of past and future tests