**Test Spec**

**1.** **Introduction**

This project aims to create a decision support system that will help any decision-maker make decisions. Our team has approached a client working in a firm for digital authentication and financial crime. Out client is the product owner, meaning he is in charge of determining product features and planning out the schedule for executing these features. He is in charge of assigning the right resources and capital in the correct places.

Our client has to decide which of the features presented are the best options. This is done by taking into consideration factors like cost, benefit and effort of each option, while also making sure of the ‘mutually exclusive’ and ‘precedes’ conditionals. The main problem that our client is facing is that his work is mainly manual, and very time consuming. Therefore, we are aiming to automate most of his work and make it easier for him to navigate between the options using VBA on excel.

**2. TEST FEATURE IDENTIFIER**

|  |  |
| --- | --- |
| **Feature to be tested** | **Identifier** |
| Data Entry from File | A1 |
| Manual Data Entry | A2 |
| Chart Data | B1 |
| Statistical Analysis | B2 |
| Output Sorted Calculations | C1 |
| Edit Erroneous Data | D1 |

**3.** **Assumptions/Constraints/Risks**

3.1 Assumptions

* The client will manually insert the feature data and insert additional features through insert data user form.

3.2 Constraints

* Time Constraint: We have four months to make our project
* Quality Constraint: We are only using Excel VBA programming
* Scope Constraint: The buttons available to the user cannot be accessed from the ribbon in the excel workbook.

3.3 Risks

|  |  |  |  |
| --- | --- | --- | --- |
| **Risks** | **Priority (H, M ,L)** | **Category** | **How to overcome it?** |
| Misinterpreting the instructions | M | Instructions | Take notes and focus on the instructor’s words |
| No Error Handling | L | Data Entry | Try to add us much code to handle as much errors as possible |
| Deadline management | H | Instructions | Have the deadline time and date written on Calendar and turn on reminder |
| Not meeting the clients expectations | H | Product Satisfaction | Record the meeting with the client and take notes during the meeting to avoid missing any client expectations |

**4.0** **Test Case Summary**

The product was tested in the presence of the members. All the requirements were tested against the Requirements Traceability Matrix. A sample demo was given to the user with all the product requirements mentioned in the functional requirements.

**5.0** **Test Case-To-Requirements Traceability Matrix**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement Number** | **Description** | **Outcome (Pass or Fail)** | **Verification Method** | **User Satisfaction/Review** |
| 1. | Automation of the decision making process | (Pass; observation) | Most of the DSS features works with least errors | 2 indications: User Satisfaction and whether the user wanted it or not. |
| 2. | The precedes are highlighted out |  | Colored cells for the features that has preceding feature |  |
| 3. | Removal of the mutually exclusive features |  | The mutually exclusive items are sorted out to the end of the column |  |
| 4. | User form to input data deals with all the errors and pastes the data | Pass; error message pops up if there wrong data is inserted into the textboxes. | Data is sorted into the relevant columns and error message in case of invalid data input |  |
| 5. | Separate sheet for different databases | Pass; the input userform creates a new datasheet if ‘yes’ is selected for the imported data | New sheet appears with the newly imported/entered data |  |
| 6. | Sorting Buttons according to the statistical data | Pass; the data were sorted according to the output results | The change in the order of the data in the raw data sheet |  |
| 7. | Delete button to remove erroneous data | Pass; entire row with the specified feature number is removed | Removal of a row with the erroneous data |  |

**6.0** **Test Case Details**

**6.1 Test Identifier**

Test number: 1

Prototype Version: 1

Date: November 15th 2020

**6.1.1 Test Objective**

The product works in a user-friendly manner and deals with errors effectively. The test spec will be used to test items repeatedly based on the same criteria.

**6.1.2 Inter case Dependencies**

1. For testing the feature of importing a data requires an existing database
2. The User also has the Micro-Enabled Excel.
3. The data entered has to save for the Excel to maintain the archive

**6.1.3 Test Items**

|  |  |
| --- | --- |
| **ID** | **Functional Requirement Description** |
| R.1 | Input User Form: Importing a Data |
| R.2 | Manual Data Input |
| R.3 | Creating a new sheet |
| R.3.1 | Manually inputted data goes incrementally to the previous entered data or below the last feature from the set |
| R.4 | Comparison of the features |
| R.5 | Calculation of the raw data into according to the statistical methods |
| R.6 | The user can view a chart that shows him how well the project is going to do |
| R.7 | The user can view ranks for individual features and combined if one must precede another. |
| R.8 | User can see alternate decisions that can be made based on the two calculation methods |
| R.9 | User can edit a feature row by pressing edit feature button |
| R.10 | User can delete a feature row by pressing delete feature button |

**6.1.4 Prerequisite Conditions**

Software Conditions:

1. Existing Microsoft Excel within the OS
2. Excel is Macro-Enabled

**6.1.5 Input Specifications**

Table 1: Test Input Rows and Columns (text.xlsx)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Feature #** | **Cost** | **Benefit** | **Effort (days)** | **Precedes** | **Mutually Exclusive** |
| **1** | **6** | **4** | **8** | **3** |  |
| **2** | **3** | **5** | **4** |  | **6** |
| **3** | **1** | **1** | **1** |  |  |
| **4** | **8** | **14** | **8** |  |  |
| **5** | **9** | **10** | **10** | **6** |  |
| **6** | **2** | **4** | **3** |  |  |
| **7** | **9** | **2** | **9** |  | **8** |
| **8** | **4** | **4** | **5** |  |  |
| **9** | **2** | **4** | **2** |  |  |

**Column Description**

* Column 1 - The decision or feature ID, which is a unique identifier that identifies the respective features in the client’s own excel workbook
* Column 2 - The cost of developing a feature with respect to its number
* Column 3 - The benefit/ profit in terms of dollars that the client earns for developing a feature
* Column 4 - The number days that are spent on developing a particular feature by the team
* Column 5 -The feature has to implemented before the feature number mentioned in the precedes column
* Column 6 - The Feature is mutually exclusive of the feature number mentioned in this column. The feature will not be implemented if its counterpart is implemented.

**Source of Input:** The input is to be manually entered by the user or to be imported from another excel worksheet (happens automatically in the program)

**Error Handling:** We are aware of the lack of error handling tests in the program, but this task is very complex as the user will mostly import 500 rows of data from another workbook rather than manually adding it. We are still figuring out an alert mechanism which tells the user which locations or range of location the data error is in.

**6.1.6 Expected Test Results**

Table 2: Expected Output Rows and Columns

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Feature #** | **Cost** | **Benefit** | **Effort (days)** | **Precedes** | **Mutually Exclusive** |
| **9** | **2** | **4** | **2** |  |  |
| **4** | **8** | **14** | **8** |  |  |
| **5** | **9** | **10** | **10** | **6** |  |
| **6** | **2** | **4** | **3** |  |  |
| **8** | **4** | **4** | **5** |  |  |
| **1** | **6** | **4** | **8** | **3** |  |
| **3** | **1** | **1** | **1** |  |  |
| **2** | **3** | **5** | **4** |  | **6** |
| **7** | **9** | **2** | **9** |  | **8** |

**6.1.7 Pass/Fail Criteria**

1. The input data user form works if importing a data from an existing database is copied into the DSS file on either a new sheet or an existing sheet. If a new sheet is created and the files are copied then the input user form completes the pass criteria.
2. When the user types a feature number upon clicking the edit feature button, the button will select the column A (with the feature numbers) and will activate the cell with the specified feature number. IF the specified feature number is activated then the button clears this criterion.
3. When the user types a feature number after clicking the Delete Feature row button, it will remove the entire row for the specified feature number. The button clears the pass criteria if an entire row is removed and the feature number gets completely removed from the data.
4. Upon opening the output user form the user gets to select which sheet they wish to analyze. IF statistical statements are displayed on the first page then the output user form page 1 clears the pass criteria.
5. Page 2 of the output user form has to show graphs for each sheet that the user wishes to view. If a chart is presented for a sheet selected by the user then the output user form page 2 clears the pass criteria.
6. The sorting buttons:
   1. Sort by feature number: Successful if the features are sorted in ascending order in the raw data excel sheet.
   2. Sort by Profit per day: Successful if the column for the profit per day is sorted in the descending order.
   3. Benefit per cost per day: Successful if the column for benefit per cost per day is sorted in the descending order.

**6.1.8 Test Procedure**

Order of test procedure: Open ‘FinanceBot’ excel document

1. Input data user form pops up
2. Select a file on the userform (select the method to input data)
3. Fill in the column text boxes in the form. This will select data from the column letter/number that you enter in the text box from the pre-existing file. The output will be in the order that the details are listed in the form. For example, feature number is the first, so if the user enters “B” in the text box next to feature number, then the program will select the data from column B in the pre-existing file and enter it into column A.
4. Click on the Edit Feature button to edit a specific feature number row.
5. Click on the Delete feature row button to delete unnecessary/mistyped feature row
6. Select the buttons to rank the features (benefit per cost or benefit per cost per day) to help determine the most efficient and/or effective.
7. Click on the button for the output form.
8. User can select different graphs and chart to display their data, for a more visual representation of their statistics based on the ranking features in this DSS (benefit per cost and benefit per cost per day)

**Reference:**

We chose this format after we went through and understood the format properly. The test spec is supposed to show how we tested our program for errors, and basically helps for quality management. So we chose the best format according to us.

Test Spec Sample. (2020). Retrieved 15 November 2020, from <https://www.cms.gov/Research-Statistics-Data-and-Systems/CMS-Information-Technology/XLC/Downloads/TestCaseSpecification.docx>

Parashchenko, R. (2020). Software testing test design templates. Retrieved 15 November 2020, from <https://strongqa.com/qa-portal/testing-docs-templates/test-design>