Software Requirement Specification (SRS)

for

**Moxi-Review Data Table**

Prepared by:

*Liang Jit Wei*

*Return Legacy Sdn. Bhd*

*3rd May 2023*

**Table of Contents:**

1. **Introduction**
   1. Purpose
   2. Intended Audience
   3. Product Scope
2. **Overall Description**
   1. User Needs
   2. Assumptions and Dependencies
3. **System Features and Requirements**
   1. Functional Requirements
   2. System Features

**1. Introduction**

**1.1 Purpose**

The purpose of this online application is to build an application to document and tabulate data regarding the results of surveys given to a selection of customers. The data is not only stored for viewing and statistical purposes, but the response will also contain a pair of images sent by each customer/reviewer, which will need to be approved manually in the application.

**1.2 Intended Audience**

The application’s intended audience and user base would be the personnel from both the Marketing Department and Customer Service Department of Return Legacy Sdn. Bhd.

**2. Overall Description**

**2.1 User Needs**

The users of the system require a clear and plain channel to view, compare and tabulate the data gathered from surveys given to customers.

The Marketing Department staff are mainly concerned with the technical parts of the data, like Member Code, Member ID, and Review Submission Date, which will be placed and displayed on the table in the main page.

The Customer Service Department, on the other hand, will take charge of the customer’s personal thoughts and feedback in the survey. This will be controlled by a button appended to each customer’s “row” in the table, which will then open a separate tab that will list out all the survey’s questions and that particular customer’s answers.

**2.2 Assumptions & Dependencies**

First off, the

Furthermore, the soundness and validity of the API commands and URL’s coded and prepared off-site., and the latter is because the API commands have to be manually written and created separate of this app, and the user has no control or ability to change/fix unresponsive or failing API commands.

This widget also operates under the assumption that the user knows the member’s Code and can find them based off of that alone. The widget’s current state does not allow

**3. System Functions & Requirements**

**3.1 System Features**

The application’s main body will have a table displaying the data on each customer’s survey. This will include:

i) Approval Status v) Customer Contact No.  
 ii) Member ID vi) Customer Email  
iii) Member Code vii) Review Submission Date  
iv) Product Redemption Date viii) Photo Status

The rows will be generated dynamically using a v-for loop, with data taken from the database through an API call. Each row will also be given an ‘ID’ attribute based on the customer’s data during the creation loop, giving each row a defining key, which will be used later.

At the top of the table, there will also be two options to filter the results. One of them being the option to filter by **Approval Status** (*All, Approved, Pending, Rejected*), and the other to filter by **Product Redemption Date** (*All, 15th May, 22nd May, 29th May*).

At the front of each row containing customer review data, will be a button, also dynamically generated. Using the ‘ID’ attribute given to each row earlier, each button will append that unique number to the back of an API call to retrieve the data on the target customer’s response to the survey questions. The application will then open up a separate browser tab where that data will be displayed, consisting of:

i) Customer Details  
 ii) Images of the customer before using the product, and 7 days after  
iii) A series of 10 conditions to best describe the experience of the product, which can be ticked independently of each other.  
iv) A series of 11 questions that can be answered with a choice of 1, 2 3; 1 being no changes experienced, and 2 being mild improvements, and 3 being major positive changes.  
 v) 2 sets of manually typed-out reviews

The answers of (ii), (iii), (iv) and (v) will be placed in accordion components that will shrink upon first viewing to both conserve space on the page and to reduce clutter. Each accordion can be opened by clicking, and multiple accordions can be opened simultaneously.

Finally, at the bottom of each page containing the customer’s review response, there will be a slider with 3 options: Approved, Pending and Rejected. These will be controlled by the staff from the Customer Service Department, and whatever changes made here will be reflected in the ‘Approval Status’ data column in the main page.

*(It should be noted that the main table is strictly* ***view-only****. While the Cust. Service staff can change the approval status from the secondary page, the main table’s values are immutable.)*