# **Task 1 – Data Requirements and Test Strategy**

## **Data Requirements**

**For the proposed web-based solution for Rolsa Technologies, several types of data will need to be stored securely and effectively. This data includes personal customer details required during account registration, as well as location-specific environmental data relevant to energy management and carbon footprint calculations.**

**Two primary data tables will manage these requirements:**

* **Customer Account Table – captures necessary user registration information:**
  + **Email Address**
  + **Password**
  + **Date of Birth**
  + **Preferred Location (from predefined choices)**
* **Environmental Data Table – contains data used for providing customers with detailed local environmental conditions:**
  + **Location Name**
  + **Air Quality Index (AQI)**
  + **Temperature (°C)**
  + **Pollen Count (per cubic metre)**

**Below are examples of each data table along with their respective data dictionaries clearly defining data types, formats, and validation rules.**

### **Customer Account Table (Example)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **AccountID** | **LocationID** | **Email\_Address** | **Password** | **Date\_of\_Birth** | **Location\_Name** |
| **1** | **101** | [**user@example.com**](mailto:user@example.com) | **SecurePass!12** | **05/09/1998** | **London** |
| **2** | **102** | [**user2@mail.com**](mailto:user2@mail.com) | **StrongPass22\*** | **23/08/2001** | **Basildon** |

### **Data Dictionary – Customer Account Information**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Format** | **Field Size** | **Description** | **Example** |
| **AccountID** | **Text (Primary Key)** | **#####** | **6** | **Unique identifier per user account** | **12345** |
| **LocationID** | **Text (Foreign Key)** | **#####** | **6** | **Unique identifier for each location** | **102** |
| **Email\_Address** | **Text** | **Email format** | **50** | **Registered user's email address** | [**user@mail.com**](mailto:user@mail.com) |
| **Password** | **Text** | **Alphanumeric with symbols** | **20** | **User’s chosen secure password** | **SecurePass12!** |
| **Date\_of\_Birth** | **Date** | **DD/MM/YYYY** | **10** | **User’s date of birth** | **05/09/1998** |
| **Location\_Name** | **Text** | **Alphabetic** | **20** | **Selected town or city location** | **London** |

**Validation Checks:**

* **Email address must contain '@' and a valid domain (e.g., '.com', '.co.uk').**
* **Password validation includes checks for length (8+ characters), uppercase and lowercase letters, numeric digits, and special symbols.**
* **All fields are mandatory except for Location, which is optional.**

### **Location Environmental Data Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **LocationID** | **Location\_Name** | **Air\_Quality\_Index (1-10)** | **Temperature (°C)** | **Pollen\_Count** |
| **101** | **London** | **5** | **14** | **70** |
| **102** | **Basildon** | **3** | **12** | **65** |

### **Data Dictionary – Location Environmental Data**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Format** | **Field Size** | **Description** | **Example** |
| **LocationID** | **Text (Primary Key)** | **#####** | **6** | **Unique identifier for each location.** | **101** |
| **Location\_Name** | **Text** | **Text** | **20** | **Name of town or city location** | **London** |
| **Air\_Quality\_Index** | **Integer** | **1-10** | **2** | **Pollution level, rated from 1 (low) to 10 (high)** | **3** |
| **Temperature** | **Integer** | **Celsius** | **2** | **Current temperature at the location** | **14** |
| **Pollen\_Count** | **Integer** | **Numeric** | **4** | **Pollen grains per cubic metre of air** | **70** |

**The Location table will serve both as a reference for environmental data and as a foreign key linked to customer account data. Future iterations of this solution could replace this manual data entry with automated API-based weather and air-quality updates, enhancing efficiency.**

## **Entity Relationship Diagram (ERD)**

**The ERD illustrates the link between customer account data and location data. The tables share a one-to-many relationship, as each location can correspond to multiple registered user accounts.**

* **Customer Account (many) → Location (one)**

## **Test Strategy**

**To ensure the digital solution fully meets Rolsa Technologies' functional and user acceptance criteria, a structured testing strategy will be implemented. Testing will verify both individual features and the overall integration of the system:**

### **Testing Categories**

**Testing phases will be segmented as follows:**

* **Unit Testing – performed at the end of each sprint to verify individual system components.**
* **Integration Testing – carried out as modules or features are combined, ensuring seamless interaction between parts.**
* **Acceptance Testing – final stage of testing, assessing the entire solution against customer and stakeholder expectations.**

**These testing methods ensure the solution's robustness and compliance with both user needs and technical standards.**

### **Types of Testing and Justification**

|  |  |  |
| --- | --- | --- |
| **Testing Method** | **Explanation** | **Application within the Project** |
| **White Box Testing** | **Tests internal code structure by examining source code. Used to identify logic errors, validate data handling, and ensure robustness.** | **Primarily used during unit testing (e.g., validating email/password inputs against multiple invalid cases, checking data storage accuracy) to ensure robust internal logic and functionality.** |
| **Black Box Testing** | **Tests the application externally, without viewing code, simulating typical user interactions. Ensures system meets user expectations.** | **Conducted towards the end of integration and acceptance testing phases, confirming that all expected inputs function correctly from the user's perspective. Highlights areas where validation rules could unintentionally block valid user inputs.** |
| **Alpha and Beta Testing** | **Involves selected user groups to gather early feedback (Alpha) and wider real-user groups to identify remaining usability issues (Beta).** | **Used exclusively during acceptance testing. Alpha testing collects in-depth feedback from internal or select user groups, whereas Beta testing engages external users to ensure the solution aligns with actual user expectations, including aspects such as accessibility, ease of navigation, and user satisfaction.** |

## **Testing Record Table**

**All testing conducted will be clearly documented within a structured testing table as shown below:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Description** | **Test Data (if applicable)** | **Expected Outcome** | **Actual Outcome** | **Action Required** |
|  |  |  |  |  |

**This table ensures each test is recorded comprehensively, detailing what the test intends to verify, the inputs provided, the expected results, the actual outcomes, and any corrective actions required. This structured testing documentation meets Rolsa Technologies’ requirements for thorough, accountable testing processes.**

**This reworded response clearly reflects the Rolsa Technologies project brief, ensuring strong alignment with Pearson T-Level assessment criteria and correct use of British English conventions throughout.**