**USER ACCEPTANCE TESTING (UAT)**

|  |  |
| --- | --- |
| **Date** | 20th June 2025 |
| **Team ID** | LTVIP2025TMID29987 |
| **Project Name** | CRM Application for Jewelry Management – (Developer) |
| **Maximum Marks** | *(To be filled by evaluator)* |

**1. Project Overview**

|  |  |
| --- | --- |
| Parameter | Description |
| Project Name | CRM for Jewel Management |
| Project Description | A Salesforce-based CRM application to automate customer, item, billing, and order management for a jewelry business. |
| Project Version | v1.0 |
| Testing Period | 19 June 2025 – 20 June 2025 |

**2. Testing Scope**

**Functionalities to be Tested:**

* Customer record creation and editing
* Order placement and association with item
* Automated billing and payment tracking
* Trigger-based email notification
* Role-based access (Admin, Worker, Goldsmith)
* Dashboard and report visualization

**User Stories to be Verified:**

* USN-01: As a user, I can register a customer and view order history
* USN-02: As an admin, I receive billing confirmation via email
* USN-03: As a worker, I can view item records but not billing details
* USN-04: As a goldsmith, I can access item details and pending orders

**3. Testing Environment**

|  |  |
| --- | --- |
| **Parameter** | **Value** |
| **URL/Platform** | [https://login.salesforce.com](https://login.salesforce.com/) |
| **Testing Org** | Salesforce Developer Edition (Custom CRM Build) |
| **Login Credentials** | Username: **ketanadinesh.22.it402@agentforce.com**  Password: **Dinesh@151** |

**4. Test Cases Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Test Scenario | Test Steps | Expected Result | Actual Result | Pass/Fail |
| TC-001 | Create a new customer record | 1. Login  2. Navigate to Jewel Customer  3. Click "New" and fill details | Customer record saved successfully | Record created | Pass |
| TC-002 | Place an order for an item | 1. Select Customer  2. Select Item  3. Create Customer Order | Order record linked to item and customer | Works as expected | Pass |
| TC-003 | Generate Billing and auto-calculate payment | 1. Create Billing record  2. Enter Paying Amount | Paid\_Amount\_\_c auto-updated | Auto-updated on save | Pass |
| TC-004 | Trigger email after billing | 1. Save billing record with Paid\_Amount\_\_c  2. Check email | Confirmation email sent to customer | Email received | Pass |
| TC-005 | Validation rule on payment | 1. Enter Paid Amount > Total Amount | Error message shown | Error validation triggered | Pass |

**5. Bug Tracking Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Bug ID | Bug Description | Steps to Reproduce | Severity | Status | Additional Feedback |
| BG-001 | Email not triggered on 1st attempt | 1. Create billing2. Save record | Medium | Resolved | Retry through Flow fixed the issue |
| BG-002 | Paid\_Amount field not refreshing | 1. Update Paying\_Amount2. Check Paid\_Amount | Low | Closed | Trigger fixed after Apex debug |
| BG-003 | Validation Rule ignored in some cases | 1. Save Paid > Total manually | High | Fixed | Rule adjusted to handle blank fields |

**6. Feedback & Observations**

* The CRM flow is smooth and intuitive.
* Email delivery is consistent post-billing.
* Reports provide clear insight into customer and billing records.
* The trigger and validation setup reduce manual errors.
* Performance is stable under normal data load.

**Notes & Best Practices**

* Cover both positive and negative test cases for each major function.
* Track bugs thoroughly with severity and current status.
* Screenshot each tested step to validate success/failure.
* Gather feedback from users with different roles (Admin, Worker, Goldsmith).
* Ensure all features match user stories before deployment.