**Agile Project**

**Submitted by:** **Bhavana Padol**

**Batch: 10th Aug 2024**

**Online Shopping Application**

The telephone agent uses an order registry and customer catalog to obtain access to an order & a customer respectively.

The order registry uses an order number as a qualifier to select particular order instance. A customer catalog uses customer name and phone number as a qualifier to select particular customer.

The attributes of an order are the order numbers and time when it is placed. The order consists of many items.

An item has item number, a quantity, unit price. It also has reference to catalog item which represents listing.

When an order is cancelled or committed, it cancels or commits each of its items first.

When an order’s total price method is invoked, the order calls the total price method of each of items and returns the sum.

**Valid Considerations:**

Inclusion: Customer provides his/her details online before placing an order.

**Agile Documents –**

Document 1: Definition of Done

Document 2- Product Vision

Document 3: User stories

Document 4: Agile PO Experience

Document 5: Product and sprint backlog and product and sprint burndown charts

Document 6: Sprint meetings

**Document 1: Definition of Done**

The **Definition of Done (DoD)** ensures that a user story or task is completed according to project standards and quality requirements. The following checklist must be fulfilled before considering any feature or functionality as "Done":

1. **Produced Code for Presumed Functionalities:**
* The development team has written and committed code implementing the intended functionalities as described in the user stories.
1. **Assumptions of User Story Met:**
* The developed feature aligns with the business requirements and satisfies all conditions outlined in the user story.
1. **Project Builds Without Errors:**
* The application compiles successfully without build-time errors, ensuring that the integration of new code does not break the system.
1. **Unit Tests Written and Passing:**
* Unit tests are implemented to validate the functionality of the code, and all tests pass without failures.
1. **Project Deployed on the Test Environment Identical to Production Platform:**
* The application is deployed in a test environment that mimics the production setup to ensure stability and reliability.
1. **Tests on Devices/Browsers Listed in the Project Assumptions Passed:**
* The application has been tested on all required devices and browsers, ensuring cross-compatibility.
1. **Feature Ok-ed by UX Designer:**
* The UX designer has reviewed and approved the feature, confirming that the user interface and experience meet design expectations.
1. **QA Performed & Issues Resolved:**
* Quality Assurance (QA) testing is completed, and all identified issues have been logged, addressed, and verified.
1. **Feature is Tested Against Acceptance Criteria:**
* The feature has been tested to confirm compliance with the acceptance criteria defined in the user story.
1. **Feature Ok-ed by Product Owner:**
* The Product Owner has reviewed and approved the feature as per business requirements.
1. **Refactoring Completed:**
* Any necessary code improvements have been implemented to enhance maintainability, readability, and efficiency.
1. **Any Configuration or Build Changes Documented:**
* Any configuration updates, environment settings, or build-related changes are properly documented.

1. **Documentation Updated:**
* All relevant documents, including user guides, API documentation, and technical notes, are updated.
1. **Peer Code Review Performed:**
* The code has been reviewed by peers, ensuring adherence to coding standards, best practices, and security guidelines.

**Document 2- Product Vision**

|  |  |
| --- | --- |
| **Scrum Project Name**: | Online Shopping Application |
| **Venue:** | [To be decided] |
| **Date:** | **Start time:** | **End Time:** | **Duration:** |
| **Client:** | Mr. ABC |
| **Stakeholder List:** | * Business Owner (Client)
* Customers (End Users)
* Telephone Agents
* Scrum Team (Product Owner, Scrum Master, Developers, QA Testers)
* Database Administrators
* UI/UX Designers
* Network & Security Team
* Customer Support Team
 |
| **Scrum Team** |
| **Scrum Master:** | Mr. XYZ |
| **Product owner:** | Mr. PQR |
| **Scrum Developer 1:** | Mr. LMN  |
| **Scrum Developer 2:** | Mr. POL |
| **Scrum Developer 3:** | Mr. KST |
| **Scrum Developer 4:** | Mr. MSP |
| **Scrum Developer 5:** | Mr. MNO |

|  |
| --- |
| **Vision:** To develop an Online Shopping Platform that improves order tracking, customer management, and automated price calculations, allowing businesses to manage customer orders efficiently and providing customers with a seamless shopping experience. |
| **Target Group*** **Customers** looking for an easy-to-use online shopping system.
* **Telephone Agents** who need a structured and automated order processing system.
* **Business Owners** aiming to scale and optimize order management.
 | **Needs*** **Customers:** Require an intuitive platform for placing, tracking, and managing orders without relying on telephone agents.
* **Agents:** Need fast and structured order retrieval using order numbers and a customer catalog based on names and phone numbers.
* **Business Owners:** Seek an automated, scalable, and secure order management solution.
 | **Product*** **An Online Shopping Platform with:**
1. Customer Registration & Profile Management
2. Order Registry & Tracking System
3. Secure Customer Data Storage & Searchability
4. Automated Order Calculations & Modifications (Cancellation & Commitment)
5. Real-Time Data Access for Orders & Customers
 | **Value*** **For Customers:** Faster, more convenient shopping experience with order tracking.
* **For Businesses:** Increased efficiency, better order management, and reduced manual errors.
* **For Agents:** Easier access to order and customer details, minimizing workload.
* **Business Model:** Scalable platform capable of handling an increasing number of users, ensuring future growth and expansion.
 |

**Document 3: User stories**

**User Story 1**: Customer Registration

|  |  |  |
| --- | --- | --- |
| **User story No:** 1 | **Tasks:** 2 | **Priority:** High |
| **Value statement:** As a new user,I want to register on the platformSo that I can place orders and track my purchases |
| **BV:** 1000 | **CP:** 8 |
| **Acceptance criteria:**1. The system should allow users to enter their name, phone number, and address.
2. Users should receive a confirmation email/SMS upon successful registration.
 |

**User Story 2:** Customer Login

|  |  |  |
| --- | --- | --- |
| **User story No:** 2 | **Tasks:** 2 | **Priority:** High |
| **Value statement:** As a registered user,I want to log in to my accountSo that I can place and track my orders**.** |
| **BV:** 1000 | **CP:** 5 |
| **Acceptance criteria:** 1. Users must log in using their phone number and password.
2. Incorrect credentials should prompt an error message.
 |

**User Story 3:** Order Placement

|  |  |  |
| --- | --- | --- |
| **User story No:** 3 | **Tasks:** 3 | **Priority:** High |
| **Value statement:** Asa customer,I want to place an order onlineSo that I can buy products conveniently. |
| **BV:** 1000 | **CP:** 13 |
| **Acceptance criteria:**1. Customers can select multiple items with quantity and unit price.
2. The system should generate a unique order number.
3. Users should receive an order confirmation message.
 |

**User Story 4:** Order Cancellation

|  |  |  |
| --- | --- | --- |
| **User story No:** 4 | **Tasks:** 2 | **Priority:** Medium |
| **Value statement:** As a customer,I want to cancel my orderSo that I can modify or remove unnecessary items. |
| **BV:**   | **CP:** 8 |
| **Acceptance criteria:**1. Users should be able to cancel an order before it is processed.
2. Users should receive a cancellation confirmation.
 |

**User Story 5:** Order Tracking

|  |  |  |
| --- | --- | --- |
| **User story No:** 5 | **Tasks:** 3  | **Priority:** High |
| **Value statement:** As a customer,I want to track my order statusSo that I can know when to expect delivery. |
| **BV:** 1000 | **CP:** 13 |
| **Acceptance criteria:**1. Users should see order status updates (Processing, Shipped, Delivered).
2. Customers should receive email/SMS notifications for status changes.
3. Orders should have a unique tracking number.
 |

**User Story 6:** Customer Catalog for Agents

|  |  |  |
| --- | --- | --- |
| **User story No:** 6 | **Tasks:** 2 | **Priority:** Medium |
| **Value statement:** As a telephone agent,I want to search for customer detailsSo that I can assist them with orders. |
| **BV:** 500 | **CP:** 8 |
| **Acceptance criteria:**1. Agents should be able to search customers using name and phone number.
2. The system should display customer details and order history.
 |

**User Story 7:** Order Registry for Agents

|  |  |  |
| --- | --- | --- |
| **User story No:** 7 | **Tasks:** 2 | **Priority:** Medium |
| **Value statement:** As a telephone agent,I want to retrieve orders using order numbersSo that I can assist customers with their purchases. |
| **BV:** 500 | **CP:** 8 |
| **Acceptance criteria:** 1. Agents should enter an order number to view order details.
2. The system should display itemized order details, total price, and status.
 |

**User Story 8:** Total Price Calculation

|  |  |  |
| --- | --- | --- |
| **User story No:** 8 | **Tasks:** 2 | **Priority:** High |
| **Value statement:** As a customer,I want to see the total price of my orderSo that I can review my purchase before checkout. |
| **BV:** 1000 | **CP:** 5 |
| **Acceptance criteria:**1. The system should sum up the unit price \* quantity for all items in an order.
2. The total price should update dynamically when items are added or removed.
 |

**User Story 9:** Secure Customer Data Storage

|  |  |  |
| --- | --- | --- |
| **User story No:** 9 | **Tasks:** 3 | **Priority:** High |
| **Value statement:** As a system administrator,I want to ensure customer data is stored securelySo that I can protect sensitive information. |
| **BV:** 1000 | **CP:** 20 |
| **Acceptance criteria:** 1. Customer details should be encrypted before storage.
2. Unauthorized users should not access customer information.
3. The system should comply with data protection regulations.
 |

**User Story 10:** Customer Notifications

|  |  |  |
| --- | --- | --- |
| **User story No:**10 | **Tasks:** 2 | **Priority:** Low |
| **Value statement:** As a customer,I want to receive notifications about my ordersSo that I stay informed about order status updates. |
| **BV:** 100 | **CP:** 3 |
| **Acceptance criteria:**1. Customers should receive notifications for order confirmation, dispatch, and delivery.
2. Messages should be clear and include relevant order details.
 |

**Document 4: Agile PO Experience**

**Agile Product Owner (PO) Experience**

**1. Role and Responsibilities of a Product Owner**

The Product Owner (PO) in Agile plays a crucial role in ensuring the development team delivers maximum value. Key responsibilities include:

* Defining and managing the product backlog.
* Prioritizing user stories based on Business Value (BV) and Complexity Points (CP).
* Collaborating with stakeholders, developers, and Scrum teams.
* Ensuring user stories are well-defined with clear acceptance criteria.
* Participating in sprint planning, reviews, and retrospectives.
* Making decisions on product increments and releases.
* Aligning product development with business goals and customer needs.

**2. Product Backlog Management**

The PO maintains a structured backlog by:

* Creating user stories with clear goals and value.
* Assigning Business Value (BV) using Scrum currency denominations (₹1000, ₹500, ₹100, ₹50, ₹20, ₹10).
* Assigning Complexity Points (CP) using Planning Poker (1, 2, 3, 5, 8, 13, 20, 40, 100, BIG).
* Conducting backlog refinement sessions with stakeholders and developers.
* Removing outdated or irrelevant backlog items.

**3. Sprint Planning and Execution**

During sprint planning:

* The PO collaborates with the Scrum team to select user stories based on priority and capacity.
* Provides clarifications on user stories to developers.
* Ensures that each story meets the "Definition of Ready" before execution.
* Assists developers in breaking down complex stories into smaller tasks.

**4. User Story Definition and Acceptance Criteria**

Each user story follows the format:
**As a** [user], **I want to** [goal], **so that** [benefit].

* User stories are assigned BV and CP values.
* Acceptance criteria are clearly defined to guide development and testing.

**5. Stakeholder Collaboration**

* Conducts regular meetings with stakeholders to gather requirements and feedback.
* Engages in discussions with developers to refine requirements.
* Aligns product vision with market trends and customer expectations.

**6. Sprint Review and Retrospective**

* Reviews completed user stories and gathers feedback.
* Ensures sprint goals are met.
* Identifies areas for improvement in the development process.
* Works with the team to enhance future sprints based on feedback.

**7. Agile Metrics and Reporting**

* Tracks team velocity using sprint burndown charts.
* Monitors backlog health and ensures timely refinement.
* Analyzes completed vs. planned story points.

**8. Continuous Improvement and Scaling Agile**

* Encourages innovation and iterative improvements.
* Participates in Agile scaling frameworks like SAFe for enterprise-level coordination.
* Conducts workshops and training sessions for Agile best practices.

**9. Challenges and Solutions**

**Common Challenges:**

* Conflicting stakeholder priorities.
* Unclear or frequently changing requirements.
* Limited development capacity.
* Technical debt and backlog overload.

**Solutions:**

* Prioritization techniques like MoSCoW (Must have, Should have, Could have, Won't have).
* Conducting regular backlog grooming sessions.
* Aligning with stakeholders through transparent communication.
* Ensuring technical debt is addressed in each sprint.

**Document 5: Product and sprint backlog and product and sprint burndown charts**

**Product Backlog:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **User Story ID** | **User Story** | **Tasks** | **Priority** | **BV** | **CP** | **Sprint** |
| 1. | Customer Registration | 2 | High | ₹ 1,000 | 8 | - |
| 2. | Customer Login | 2 | High | ₹ 1,000 | 5 | - |
| 3. | Order Placement | 3 | High | ₹ 1,000 | 13 | - |
| 4. | Order Cancellation | 2 | Medium | ₹ 500 | 8 | - |
| 5. | Order Tracking | 3 | High | ₹ 1,000 | 13 | - |
| 6. | Customer Catalog for Agents | 2 | Medium | ₹ 500 | 8 | - |
| 7. | Order Registry for Agents | 2 | Medium | ₹ 500 | 8 | - |
| 8. | Total Price Calculation | 1 | High | ₹ 1,000 | 5 | - |
| 9. | Secure Customer Data Storage | 3 | High | ₹ 1,000 | 20 | - |
| 10. | Customer Notifications | 2 | Low | ₹ 100 | 3 | - |

**Sprint Backlog:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **User Story ID** | **User Story** | **Tasks** | **Owner** | **Status** | **Estimated Effort** |
| 1. | Customer Registration | 2 | Developer A | In Progress | 8 |
| 2. | Customer Login | 2 | Developer B | Not Started | 5 |
| 3. | Order Placement | 3 | Developer C | Not Started | 13 |
| 5. | Order Tracking | 3 | Developer D | Not Started | 13 |
| 8. | Total Price Calculation | 1 | Developer E | Not Started | 5 |

* Sprint capacity is considered while selecting stories.
* High-priority features are taken first.
* Medium/Low-priority stories are deferred to future sprints.

**Product Burndown Chart:**

**Sprint Burndown Chart:**



**Document 6: Sprint meetings**

**Meeting Type 1: Sprint Planning meeting**

|  |  |
| --- | --- |
| **Date** | [To be filled] |
| **Time** | [To be filled] |
| **Location** | [To be filled] |
| **Prepared By** | Business Analyst |
| **Attendees** | Product Owner, Scrum Master, Development Team, QA Team |

**Agenda Topics**

|  |  |  |
| --- | --- | --- |
| **Topic** | **Presenter** | **Time Allotted** |
| Sprint Goals Review | Product Owner | 15 mins |
| Backlog Prioritization | Scrum Master | 20 mins |
| Task Breakdown | Development Team | 30 mins |
| QA & Testing Strategy | QA Lead | 15 mins |
| Sprint Capacity Planning | Scrum Master | 15 mins |

**Other Information**

|  |  |
| --- | --- |
| Observers | [To be filled] |
| Resources | Jira, Balsamiq, Visio |
| Special Notes | Ensure all dependencies are resolved before the sprint begins. |

**Meeting Type 2: Sprint review meeting**

|  |  |
| --- | --- |
| **Date** | [To be filled] |
| **Time** | [To be filled] |
| **Location** | [To be filled] |
| **Prepared By** |  |
| **Attendees** |  |

**Sprint Status:**

|  |  |  |  |
| --- | --- | --- | --- |
| Sprint status | Things to demo | Quick updates | What’s next |
| Sprint 1 Completed | User Registration, Login, Order Placement | Customer registration and login functionalities are working. Minor UI improvements suggested. | Next sprint will focus on Order Tracking and Secure Data Storage |

**Meeting Type 3: Sprint Retrospective Meeting**

|  |  |
| --- | --- |
| **Date** | [To be filled] |
| **Time** | [To be filled] |
| **Location** | [To be filled] |
| **Prepared By** |  |
| **Attendees** |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Agenda** | **What Went Well** | **What Didn’t Go Well** | **Questions** | **Reference** |
| User Login System | Implemented Secure Authentication | Delayed due to API issues | How to improve API response time? | Backend Documentation |
| Product Page UI | Optimized for mobile devices | Design inconsistencies | Should we test on multiple devices? | UI Guidelines |
| Payment Module | Successful Transactions | Some payment failures | How to handle failed payments better? | Payment API Docs |

**Meeting Type 4: Daily Stand-up Meeting**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Question** | **Name/Role** | **Monday** | **Tuesday** | **Wednesday** | **Thursday** | **Friday** | **Saturday** | **Sunday** |
| **What did you do yesterday?** | Developer 1 | Worked on Login API | Fixed UI issues | Reviewed code | Integrated with DB | Tested login | N/A | N/A |
|  | Developer 2 | Setup Product DB | Created Product Filters | Updated UI | Implemented Search | Performance optimization | N/A | N/A |
|  | Developer 3 | Integrated Payment API | Debugged payment issues | Tested transactions | Fixed checkout errors | Deployed fixes | N/A | N/A |
| **What will you do today?** | Developer 1 | Debug session handling | Improve password recovery | API performance tuning | Review security logs | Finalize auth testing | N/A | N/A |
|  | Developer 2 | Optimize queries | Implement sorting features | Finalize category filters | Test UI responsiveness | Deploy changes | N/A | N/A |
|  | Developer 3 | Test refund system | Enhance payment logging | Ensure PCI compliance | Final testing | Submit for review | N/A | N/A |
| **What is blocking your progress?** | Developer 1 | API response delays | N/A | Server issues | N/A | N/A | N/A | N/A |
|  | Developer 2 | UI inconsistencies | Awaiting design approval | N/A | N/A | N/A | N/A | N/A |
|  | Developer 3 | Payment provider downtime | N/A | N/A | N/A | N/A | N/A | N/A |