**AGILE DOCUMENTS**

**1: Definition of Done**

The definition of done (DoD) is when all conditions, or acceptance criteria, that a software product must satisfy are met and ready to be accepted by a user, customer or team. We must meet the definition of done to ensure quality. It lowers rework, by preventing user stories that don’t meet the definition from being promoted to higher level environments. It will prevent features that don’t meet the definition from being delivered to the customer or user.

**Checklist for DOD:**

**1.Produced code for presumed functionalities -** It means that code has been written for functionalities that were assumed to be necessary.

**2.Assumptions of User Story met -** A user story meets assumptions when the team has confirmed all identified assumptions. Assumptions are statements that a team believes to be true but still need to be confirmed

**3.Project builds without errors -** Project should be error free and the builds should be in updated version.

**4.Unit tests written and passing -** Unit tests are written to validate that a code unit is functioning properly. They should cover common scenarios and met the project standard.

**5.Project deployed on the test environment identical to the production platform -** So that the project can been tested without affecting the live environment and also bug can be identified and fixed.

**6.Tests on devices/browsers listed in the project assumptions passed -** Cross browser testing is conducted and testing conducted in different screen size so the testing will be complete

**7.Feature ok-ed by UX designer -** Design test across all the device is conducted and usability test is also conducted

**8.QA performed & issues resolved -** Quality assurance (QA) is a process that involves testing a product or service to ensure it meets customer expectations. QA issues are errors, defects, or inconsistencies that affect the product's quality. QA issues are documented and addressed.

**9.Feature is tested against acceptance criteria -** Acceptance criteria are a set of conditions that a software feature must meet to be considered complete and acceptable. Testing a feature against acceptance criteria verifies that the feature works. If any features are tested against acceptance criteria then deviations are documented and justified.

**10.Feature ok-ed by Product Owner -** Product Owner will review and approved the implemented feature and if Any feedback or changes in the feature are addressed.

**11.Refactoring completed -** Refactoring of code improves the standard and quality of the code and maintainability

**12.Any configuration or build changes documented -** Configuration changes are documented using a change control mechanism and configuration management documentation.

**13.Documentation updated -** README files are updated so that everyone can view the documents.

**14.Peer Code Review performed -** Peer code review are performed and documented and the feedback are been addressed and acknowledged.

**Document 2- Product Vision**



**Document 3: User stories**

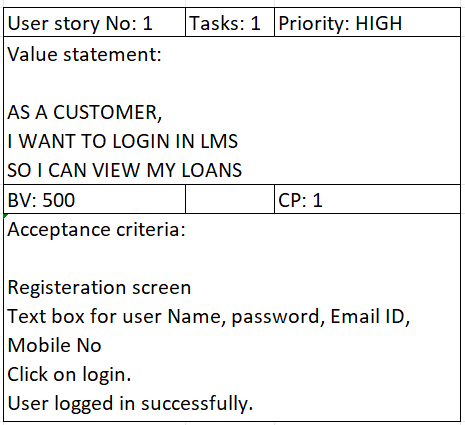
**User Stories** are short simple stories used to describe the functionality/requirements of the client from the user’s perspective. It shows the world on how the User’s side is and it simplifies the requirements description.

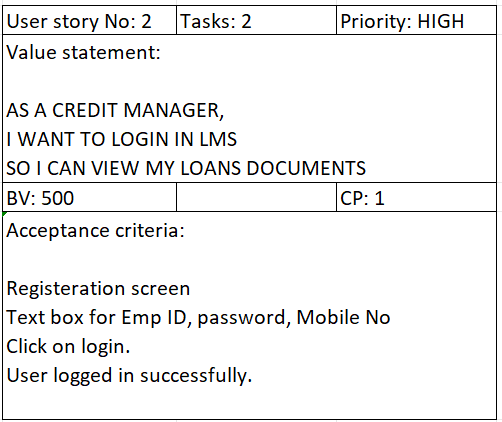
**Business Value (BV)**- BV is how important is this feature (User story) to the Business. It rated by the currency note technique.

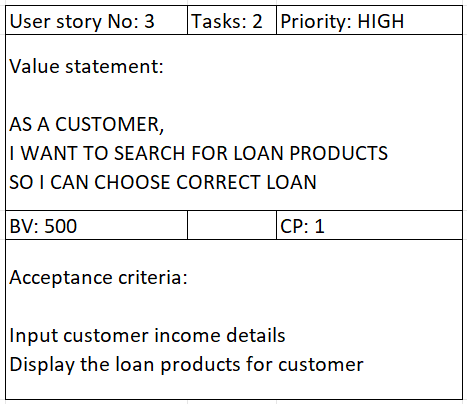
**Complexity Points (CP)**- Complexity points are the efforts used or the developers to write the code and the time spent for the requirement. It is rated as per the CP points known as Poker cards.

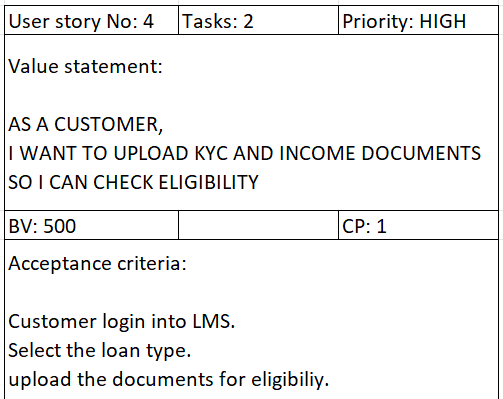
**Acceptance criteria-** While User stories aim at describing what exactly the user wants the system to do, the goal of acceptance criteria is to explain the conditions that a specific user story must satisfy.

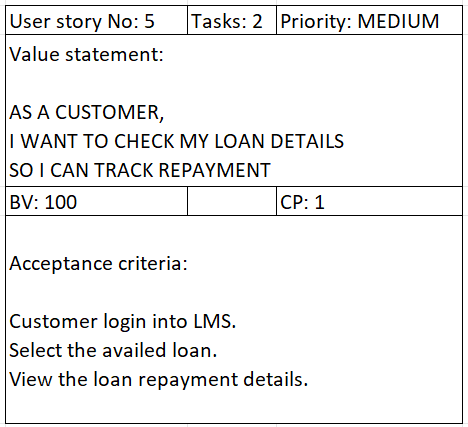
Acceptance criteria are the conditions that a software product must meet to be accepted by the user, a customer, or other systems. They are unique or each user story and define the feature’s behavior from the end user’s perspective and how we can achieve it.



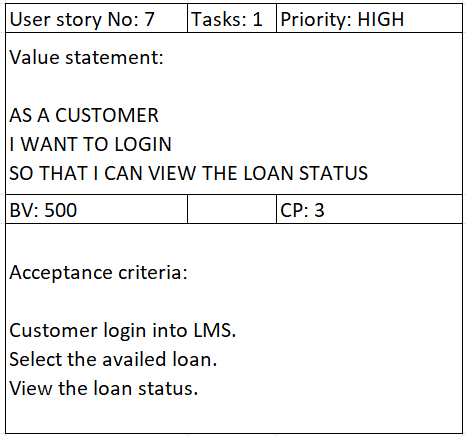


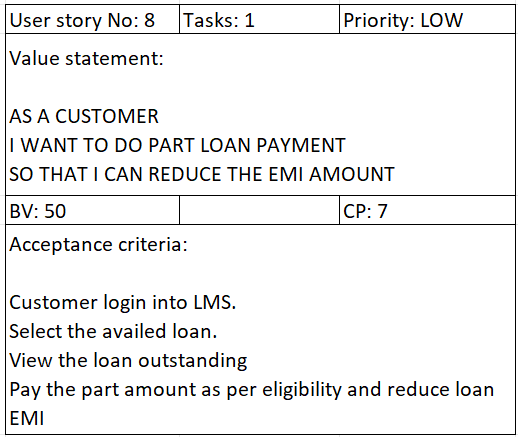


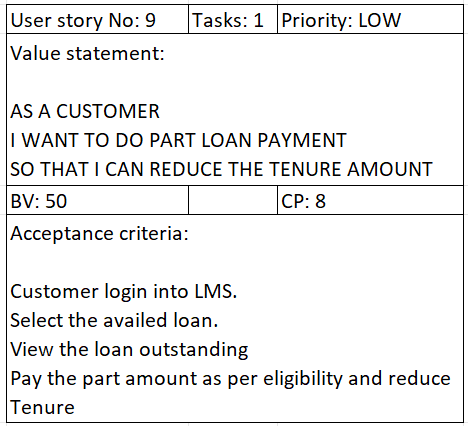


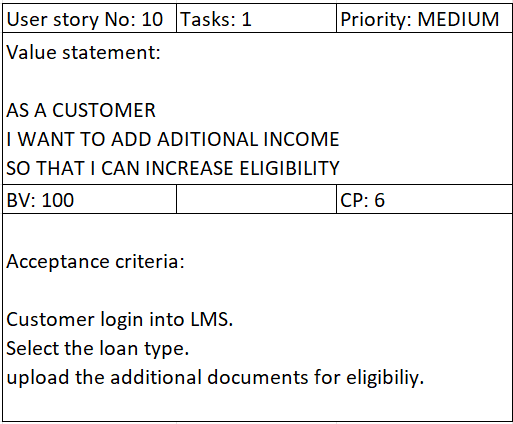


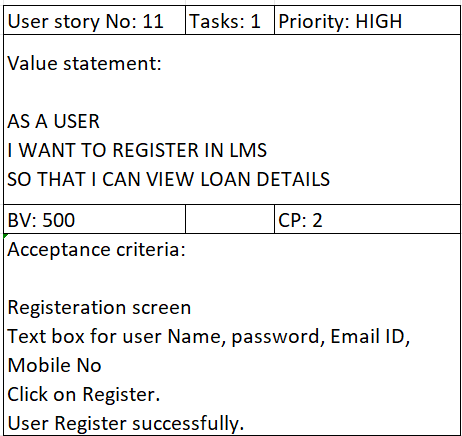


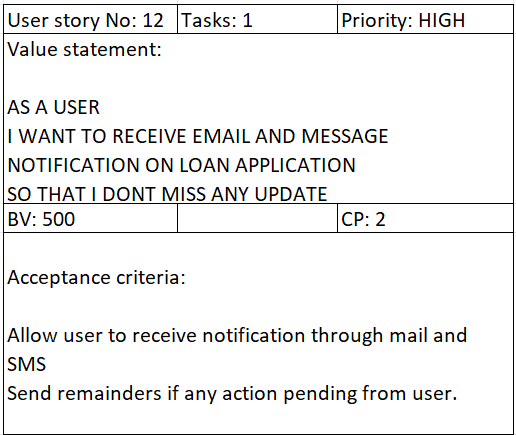


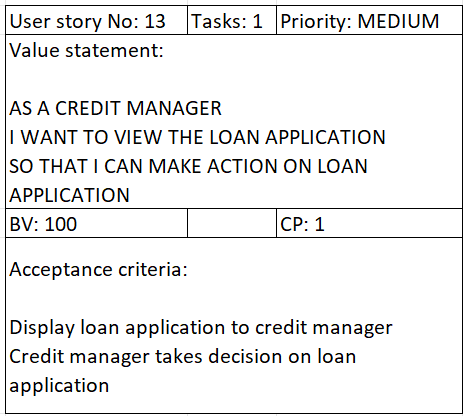


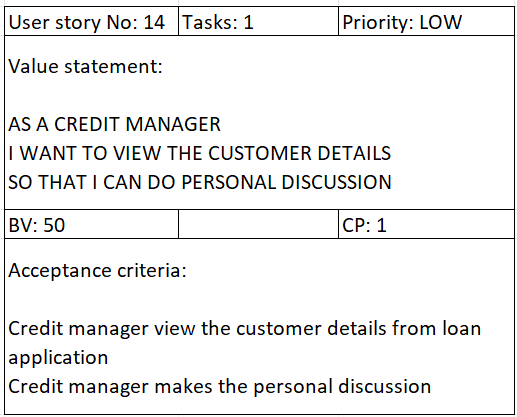


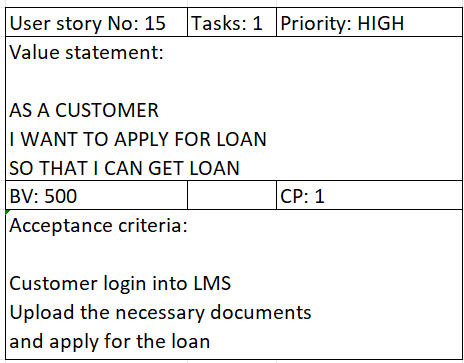


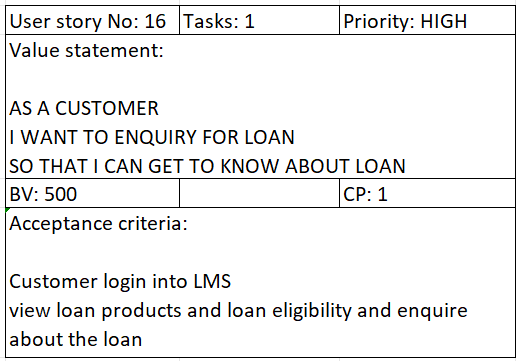


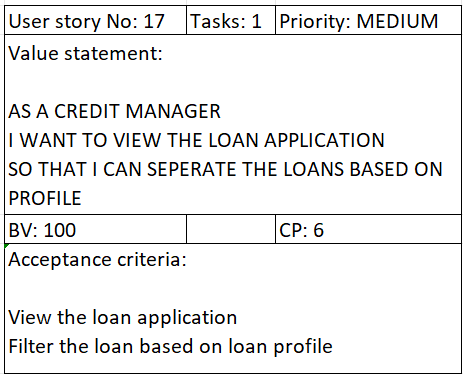


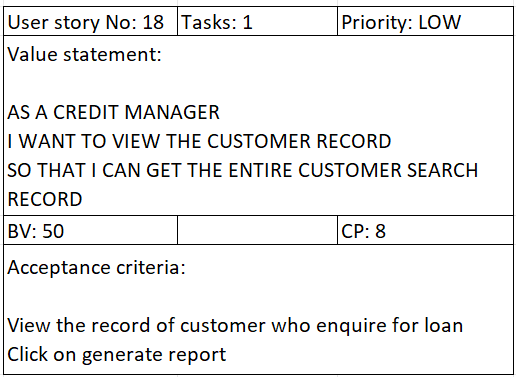


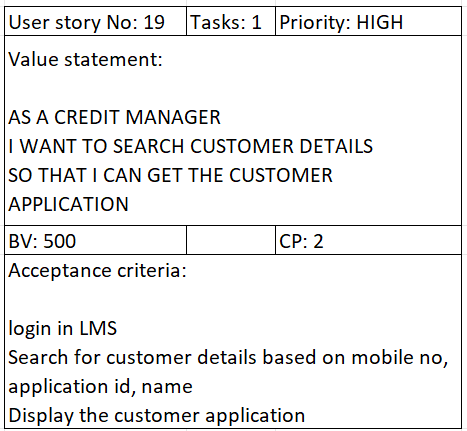


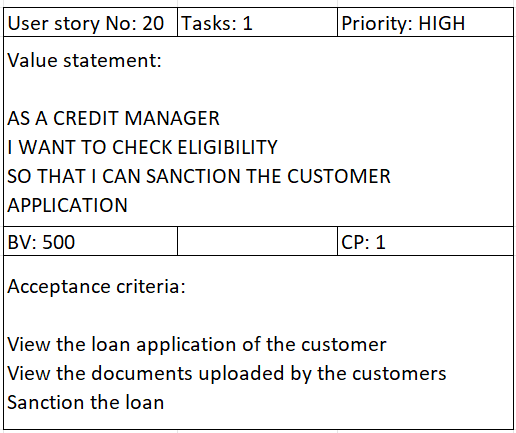


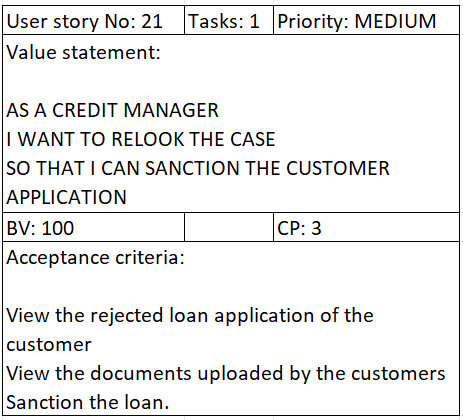


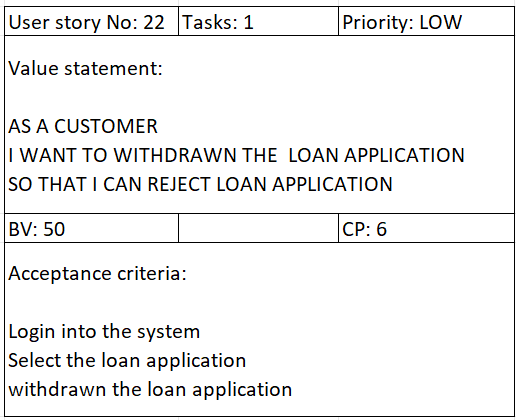


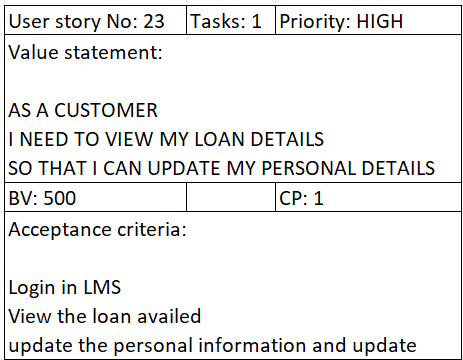




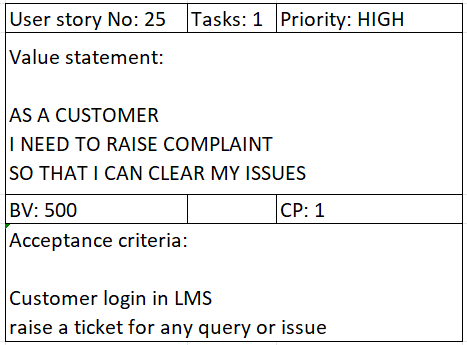


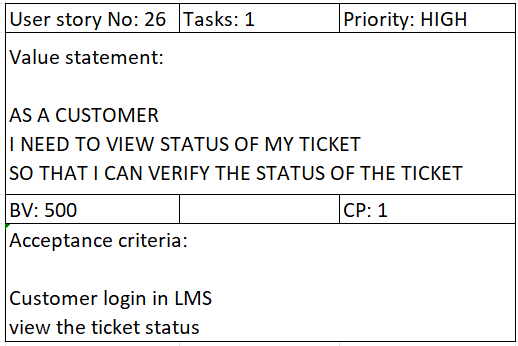


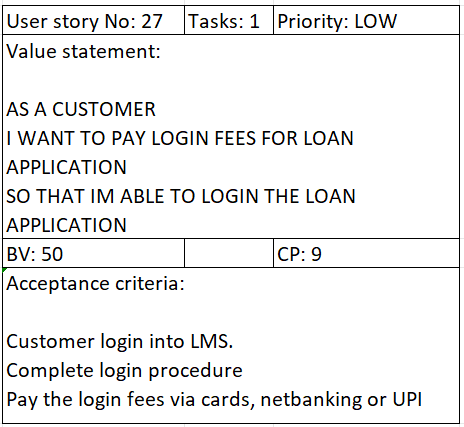


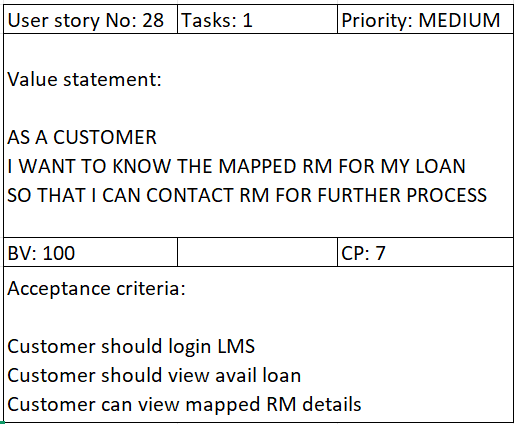


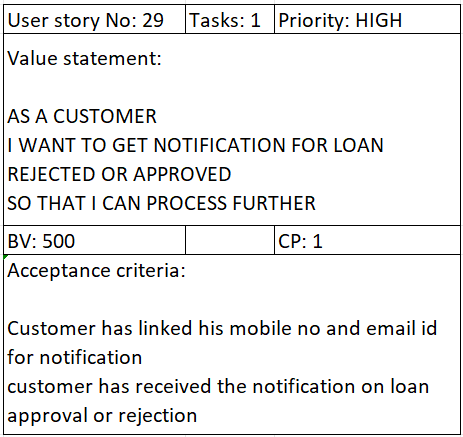


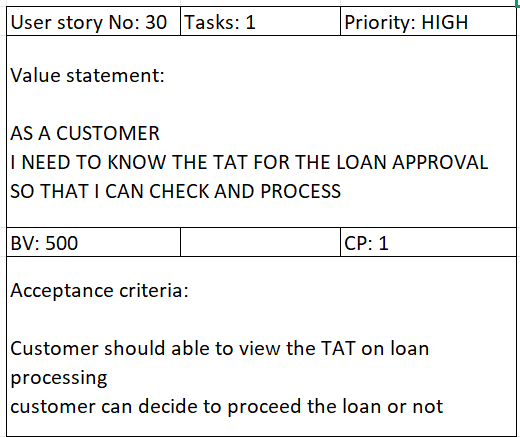


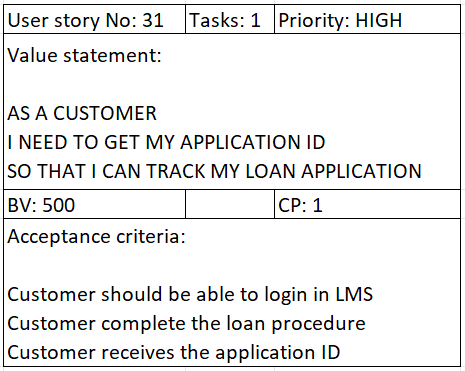


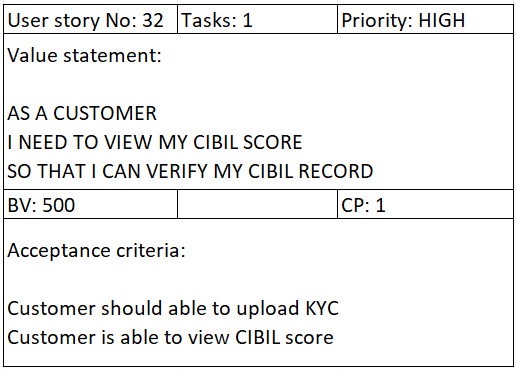


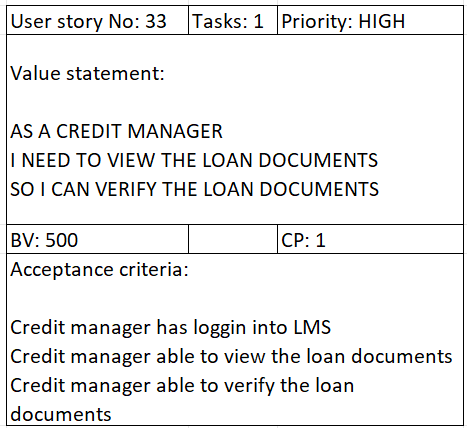


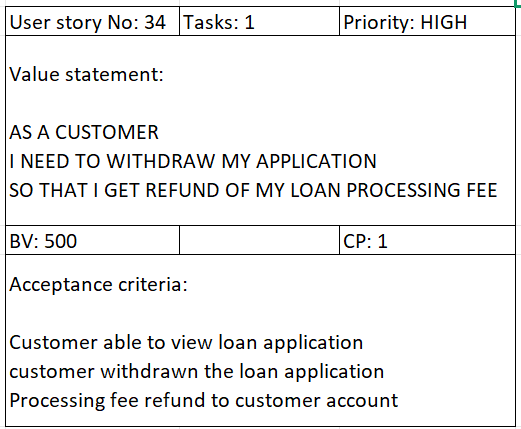


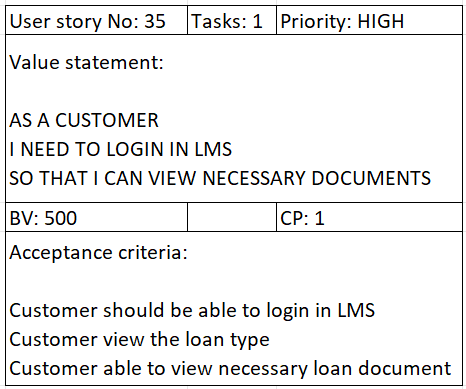


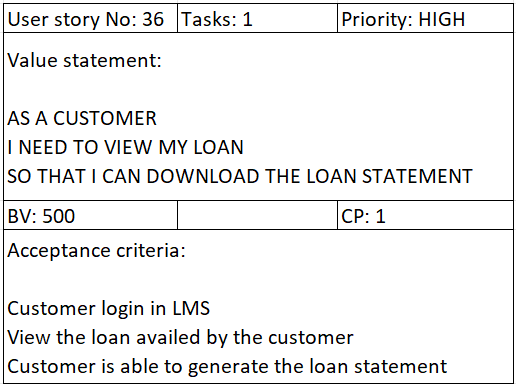


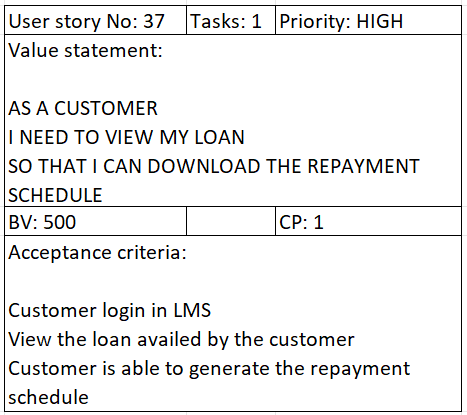


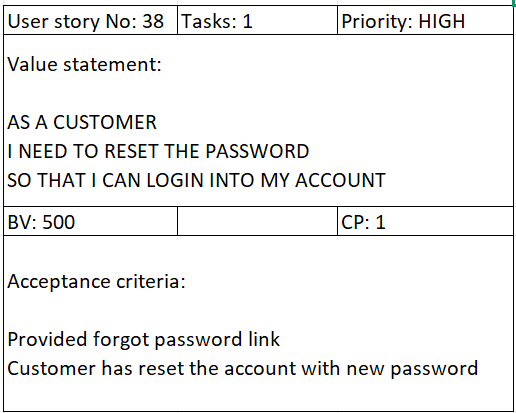


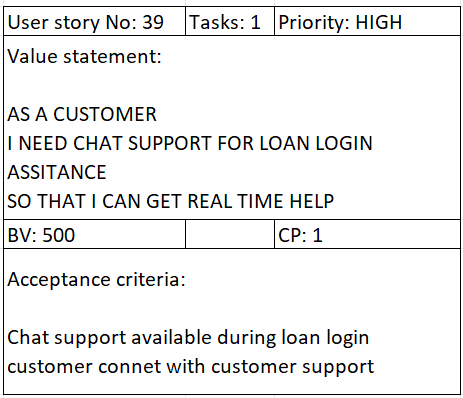


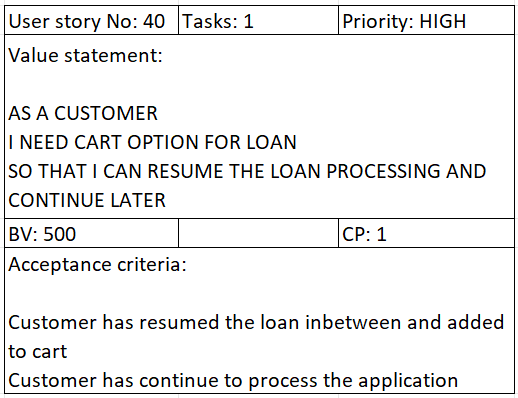












**Document 4: Agile PO Experience**

The Product Owner has a vision of the product keeping the domain/industry experience and the market need.

❖ Following are the responsibilities of PO in a project

➢ **Market Analysis -** Market analysis is a detailed assessment of your business's target market and the competitive landscape within a specific industry. This analysis lets you project the success you can expect when you introduce your brand and its products to consumers within the market.

➢ **Enterprise Analysis -**Enterprise Analysis is the identification of business opportunities, development and maintenance of a business architecture, and the determination of optimum project investment. The most significant result that can come from effective enterprise analysis would be a business meeting its strategic goals

➢ **Product Vision and Road map -** “Product Vision specifies the What and Why of the product, while Product Strategy elaborates how to realize the vision with a specific approach, and provides a road map showing a timeline for executing the strategy.

➢ **Managing Product Features**

* Market research & user research.
* Idea management.
* Technical specifications.
* Road mapping.
* Prioritization.
* Product development.
* MVP release and customer feedback collection and iterations.

➢ **Managing Product Backlog**

* Review the backlog before the iteration planning meeting to ensure that the prioritization of tasks is correct and that the previous feedback has been implemented.
* When the backlog gets larger, you must categorize the backlog items into short-term and long-term.
* Delete items that aren’t required and keep the ones that aren’t ready to be off the backlog.
* Do not add tasks unless planned.
* Make prioritization of the backlog tasks a priority

➢ **Managing Overall Iteration Progress**

An iteration manager manages the work done by one team within an iteration, although there might be some coordination with other teams. An iteration manager helps to optimize the efficiency of the team. People on software development teams tend to like to focus on their tasks.

* Set iteration goals. The future iteration's goal or a new feature may come from a user request, from the customer, or a priority order/list.
* Estimate user stories.
* Establish capacity.
* Planning steps – internal priority.
* From stories to tasks.

❖ From this project I have learned how to handle sprint meetings such as

**Sprint planning meeting -** Which requirements to be undertaken and can be delivered withing the scrum time framework

**Daily scrum meeting -** This usually lasts for less than 10 min where each team member informs what they did and what they are working on.

**Sprint review meeting -** Finished work is shown to the stakeholders, customers and or the users who provide the feedback. Based on the response to the finished work, the product owner will add, remove or adjust prioritizes of product backlog items.(Scrum master, Product owner, development team)

**Sprint retrospective meeting -** Entire team gathers in a room and discusses the burndown chart and what went right and what went wrong

**Backlog refinement meeting-** During this meeting, the product owner and the Scrum team look at the remaining product backlog items. The product owner may re-prioritize product backlog items. Scrum team members then plan a new sprint based on which item is at the top of the list.

❖ Also, User stories creation and what things will be included in user stories such as

**1.User Story**

As a [type of user],

I want [an action or feature]

So that [benefit or value].

For example:

As a website visitor,

I want to be able to reset my password,

So that I can regain access to my account.

**2.Tasks**

List of specific tasks required to complete the user story. These are the individual steps or actions that need to be taken to implement the feature.

For example:

Implement "Forgot Password" button on login page

Create a secure password reset mechanism

Send confirmation email to the user

**3.Priority**

Indicates the importance or urgency of the user story in the context of the overall project.

Priorities are usually categorized as high, medium, or low.

For example:

High priority: Critical for the next release

Medium priority: Important but can wait for the next iteration

Low priority: Nice to have, can be considered later

**4.Acceptance Criteria**

A set of conditions or criteria that must be met for the user story to be considered complete. These criteria help ensure that the implemented feature meets the user's expectations.

For example:

User receives a password reset email

Clicking the reset link takes the user to a secure page

User can successfully set a new password

**5.Business Value**

Describes the value that the user story brings to the business or end-user. It helps prioritize features based on their impact.

For example:

High business value: Increases user engagement and satisfaction

Medium business value: Adds a useful feature but not critical

Low business value: Nice-to-have, minimal impact on users

**6.Complexity Points**

An estimation of the complexity or difficulty of implementing the user story. Complexity points are often used in Agile methodologies for capacity planning and sprint commitments.

For example:

1-3 points: Low complexity, straight forward implementation

5-8 points: Medium complexity, may require some effort

13+ points: High complexity, may involve significant challenges

❖ In Scrum, a product owner serves as the liaison between multiple areas of an organization. This person communicates with business stakeholders and collaborates closely with Scrum teams to keep all areas of the business informed on a project's development.

❖ The product owner develops a vision of a product's function and operation, which in turn allows this Scrum team member to define product features and break those features into product backlog items.

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

**Product backlog:** A product backlog is a list of the new features, changes to existing features, bug fixes, infrastructure changes, or other activities that a team may deliver in order to achieve a specific outcome.

It should be cheap and fast to add a product backlog item to the product backlog, and it should be equally as easy to remove a product backlog item that does not result in direct progress to achieving the desired outcome or enable progress toward the outcome.

Product backlog items take a variety of formats, with user stories being the most common. The team using the product backlog determines the format they chose to use and looks to the backlog items as reminders of the aspects of a solution they may work on.



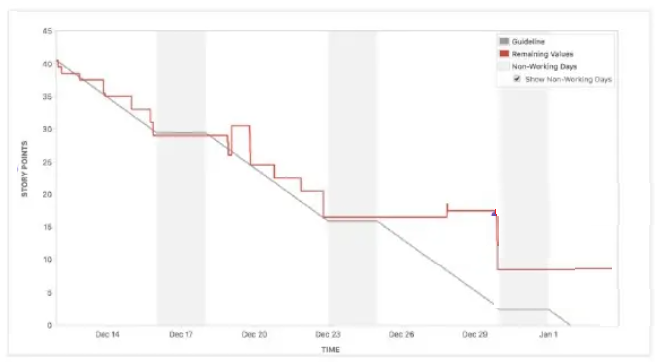
**Sprint backlog:** The Sprint Backlog is composed of the Sprint Goal (why), the set of Product Backlog items selected for the Sprint (what), as well as an actionable plan for delivering the Increment(how).

The Sprint Backlog is a plan by and for the Developers. It is a highly visible, real-time picture of the work that the Developers plan to accomplish during the Sprint in order to achieve the Sprint Goal. Consequently, the Sprint Backlog is updated throughout the Sprint as more is learned. It should have enough detail that they can inspect their progress in the Daily Scrum.



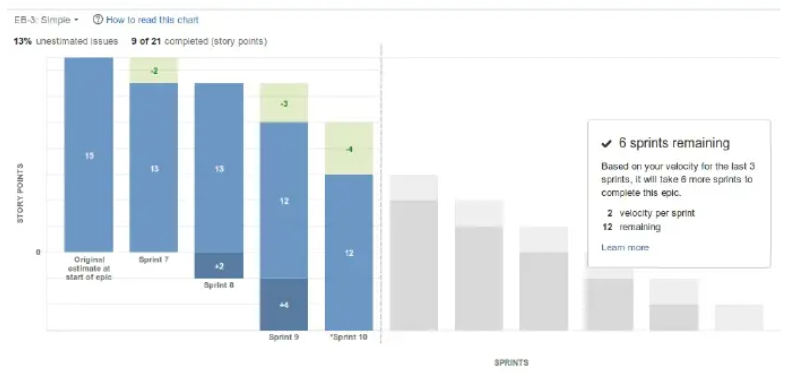
**Product burndown Chart**

A product burndown chart shows how much work remains for the entire project, whereas a sprint burndown chart shows how much work remains in a specific iteration. A product burndown chart collects a larger amount of data.



**Sprint burndown Chart**

It can be used to track the total work remaining in the sprint, and to project the likelihood of achieving the sprint goal. By tracking the remaining work throughout the sprint, a team can manage its progress, and respond to trends accordingly.



**Document 6 : Sprint meetings**

**Sprint Planning meeting**



**Agenda Topics**



**Other Information**



**Sprint review meeting**





**Meeting Type 3: Sprint retrospective meeting**





**Daily Stand-up meeting**

