**Day 3**

**Date: 07-08-2024**

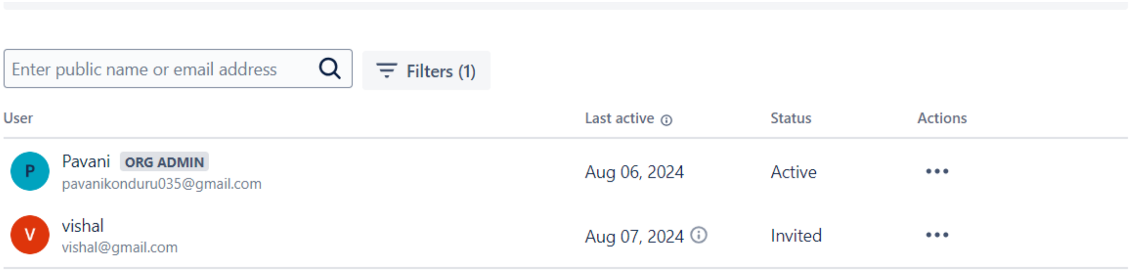
# Software Development Life Cycle and Agile Principles

## 1. What is meant by smoke testing?

Smoke testing, also called build verification testing or confidence testing, is a software testing method that is used to determine if a new software build is ready for the next testing phase. This testing method determines if the most crucial functions of a program work but does not delve into finer details.

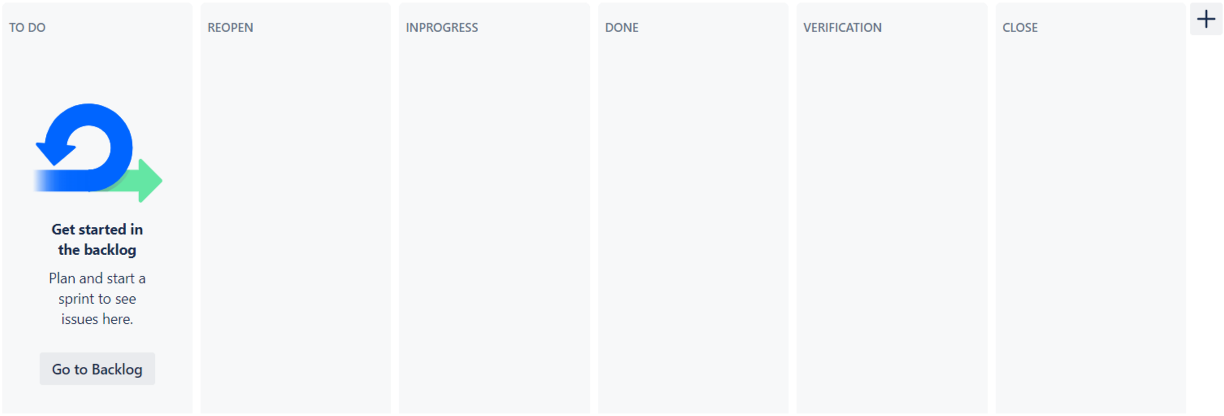
## 2. What is meant by sanity testing?

Sanity testing is used to validate the changes made to one or few specific parts of the application in order to make sure that the software is still functioning as expected after a small change or a bug fix.



| User | Last active | Status | Actions |
|---|---|---|---|
| Pavani ORG ADMIN<br>pavanikonduru035@gmail.com | Aug 06, 2024 | Active | ••• |
| vishal<br>vishal@gmail.com | Aug 07, 2024 | Invited | ••• |

**Image-1: Adding a another User into Jira account**



**Get started in the backlog**

Plan and start a sprint to see issues here.

**Image-2: create column in active sprint**

## What is Sprint?

A sprint is a fixed duration for achieving programming milestones.

A sprint in scrum is a fixed-length iteration during which a scrum team completes a set amount of work. These time-boxed periods typically last one month or less, creating a consistent rhythm for development and feedback.

A sprint is a fixed time period where teams complete work from their product backlog. Sprints are usually one, two, or four weeks long. At the end of the sprint, a team will typically have built and implemented a working product increment.

## What is scrum?

Scrum is a Agile framework for managing complex Software project work.

Scrum is a framework for getting work done within agile. Scrum uses all the core principles of agile to define methods to facilitate a project.

## What is a Daily Scrum?

Daily scrums are quick meetings held each day at the same time for members of the product development team working on a particular sprint. The team collectively reviews the progress made toward achieving the Sprint Goal.

## Sprint Ceremonies:

--> Sprint Planning: A meeting to plan the upcoming sprint and prioritize the backlog.

** When practicing scrum, the sprint planning meeting is held at the beginning of the sprint and is where teams identify what can be delivered in the sprint and how that work will be achieved. At the end of the planning meeting, every scrum member needs to be clear on what can be delivered in the sprint and how the increment can be delivered.

--> Daily Scrum: A daily meeting where team members discuss what they are accomplished, what they are working on, and any blockers.

** The daily stand-up – daily scrum – is a short, 15-minute (or less) daily meeting to discuss progress and identify blockers. Attendees are urged to participate while standing to help keep the meeting short.

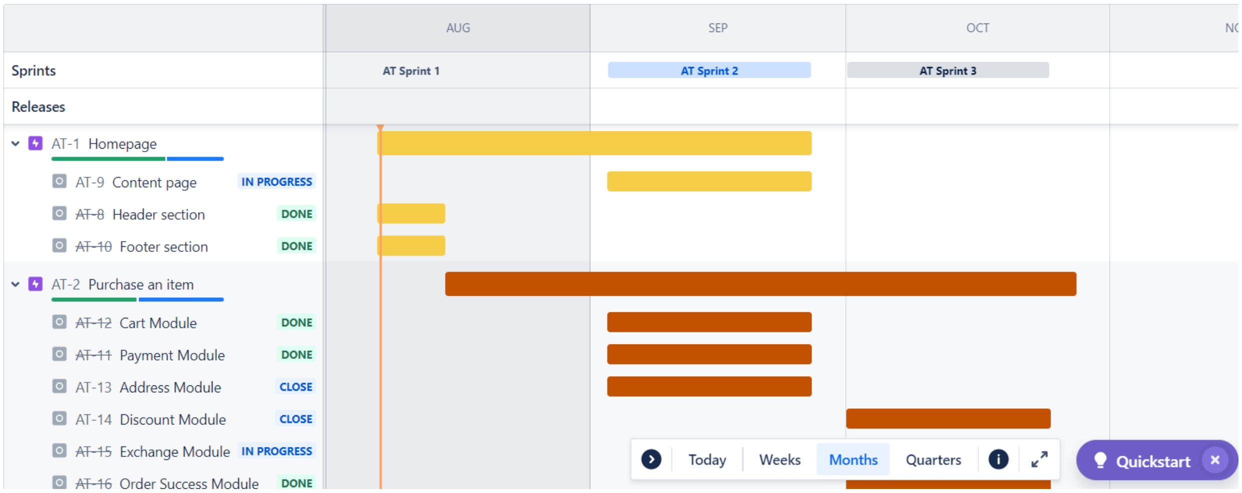
--> Sprint Review: A review of the completed work during the sprint and demo to stakeholders.

** The sprint review, also called an iteration review, is where the scrum team meets to reveal what was accomplished during the sprint. A development team shows which backlog items are “Done” to stakeholders and teammates, who can then give feedback.

--> Sprint Retrospective: A meeting to reflect on the previous sprint and identify areas for improvement.

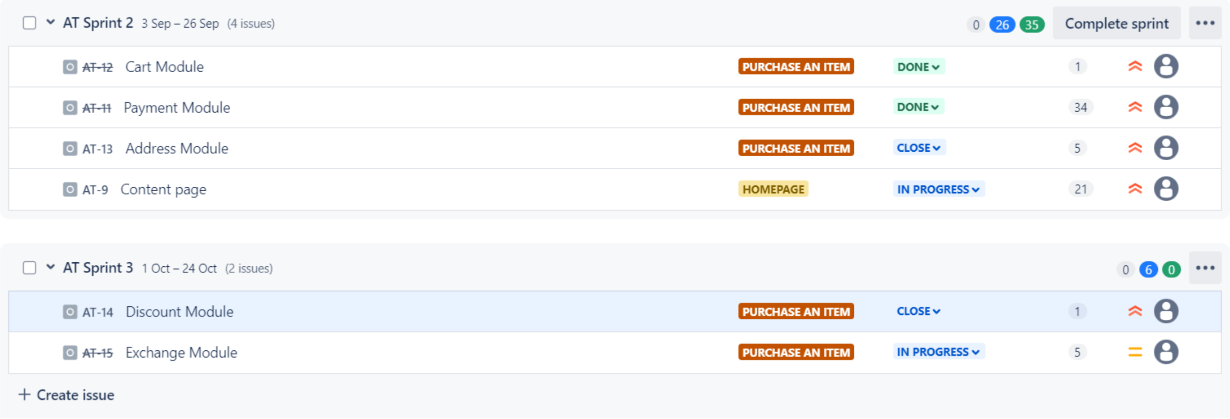
** A sprint retrospective is a meeting to review what was successful during the sprint and what can be improved upon. Agile teams can specifically review team dynamics, processes, and tools, then create plans to improve the way the team works.

--> Backlog Refinement: A meeting to review and prioritize the product backlog.



| | | |
|---|---|---|
| AT-1 | Homepage | |
| AT-9 | Content page | IN PROGRESS |
| ~~AT-8~~ | Header section | DONE |
| ~~AT-10~~ | Footer section | DONE |
| AT-2 | Purchase an item | |
| ~~AT-12~~ | Cart Module | DONE |
| ~~AT-11~~ | Payment Module | DONE |
| AT-13 | Address Module | CLOSE |
| AT-14 | Discount Module | CLOSE |
| ~~AT-15~~ | Exchange Module | IN PROGRESS |
| ~~AT-16~~ | Order Success Module | DONE |

**Image-3: set Sprint**



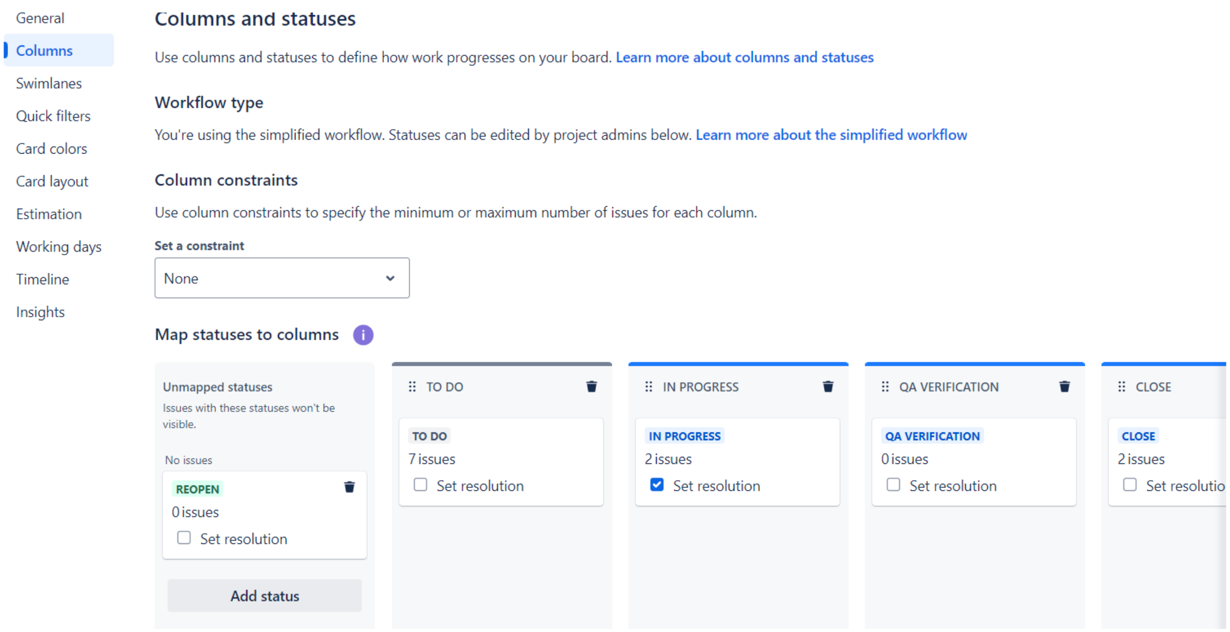
□ AT Sprint 2 3 Sep – 26 Sep (4 issues) 0 26 35

| | | | | |
|---|---|---|---|---|
| ~~AT-12~~ | Cart Module | PURCHASE AN ITEM | DONE | 1 |
| ~~AT-11~~ | Payment Module | PURCHASE AN ITEM | DONE | 34 |
| AT-13 | Address Module | PURCHASE AN ITEM | CLOSE | 5 |
| AT-9 | Content page | HOMEPAGE | IN PROGRESS | 21 |

□ AT Sprint 3 1 Oct – 24 Oct (2 issues) 0 6 0

| | | | | |
|---|---|---|---|---|
| AT-14 | Discount Module | PURCHASE AN ITEM | CLOSE | 1 |
| ~~AT-15~~ | Exchange Module | PURCHASE AN ITEM | IN PROGRESS | 5 |

**Image-4: Adding Backlog into sprint**



General
Columns
Swimlanes
Quick filters
Card colors
Card layout
Estimation
Working days
Timeline
Insights

Columns and statuses

Use columns and statuses to define how work progresses on your board. Learn more about columns and statuses

Workflow type

You're using the simplified workflow. Statuses can be edited by project admins below. Learn more about the simplified workflow

Column constraints

Use column constraints to specify the minimum or maximum number of issues for each column.

Set a constraint

None

Map statuses to columns

Unmapped statuses
Issues with these statuses won't be visible.
No issues
REOPEN
0 issues
□ Set resolution
Add status

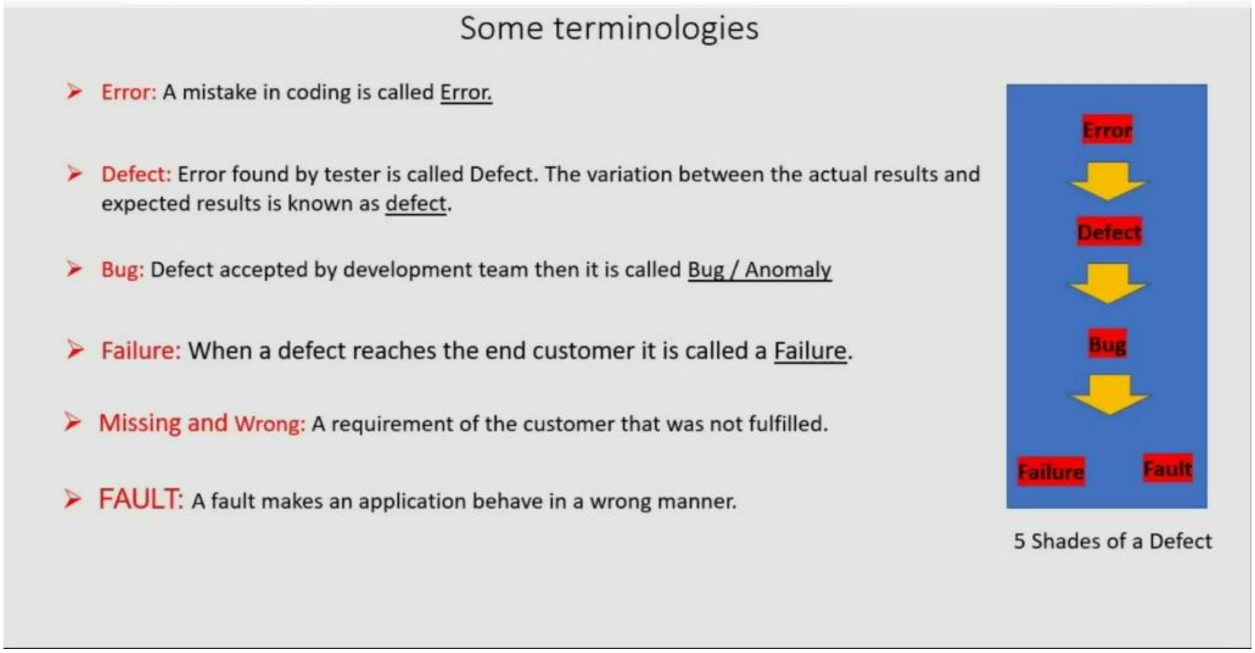
TO DO
TO DO
7 issues
□ Set resolution

IN PROGRESS
IN PROGRESS
2 issues
☑ Set resolution

QA VERIFICATION
QA VERIFICATION
0 issues
□ Set resolution

CLOSE
CLOSE
2 issues
□ Set resolutio

**Image-5: Creating a status Column**



Some terminologies

- Error: A mistake in coding is called Error.
- Defect: Error found by tester is called Defect. The variation between the actual results and expected results is known as defect.
- Bug: Defect accepted by development team then it is called Bug / Anomaly
- Failure: When a defect reaches the end customer it is called a Failure.
- Missing and Wrong: A requirement of the customer that was not fulfilled.
- FAULT: A fault makes an application behave in a wrong manner.

5 Shades of a Defect

## Part of defect:

1. description
2. priority
3. seviority
4. author
5. environment
6. date

7. steps to reproduce

8. Actual Result

9. Expected Result

10 Attachment

11. which story it is blocking

## ASSIGNEMENT:

User is not able to remove the items from the cart

User is not able to login using valid login credentials

Prerequisite:

Login Credentials

Application URL

Step to repro

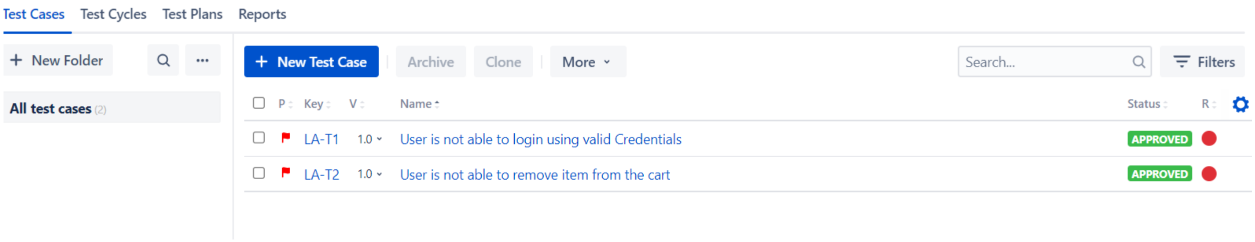
1. Open home page of the Application
2. User see the Username and password text box
3. Provide Username and password in the respective mode
4. User select the login button observe the results

Actual: User is not able to login

Expected: User should be able to login using given valid credentials

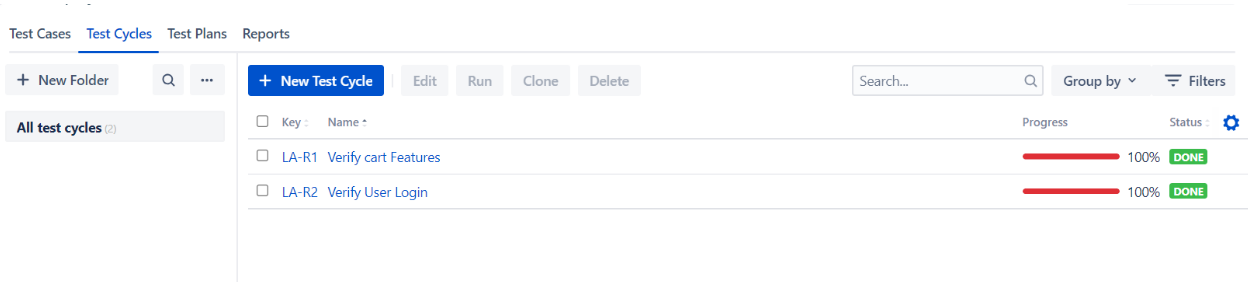
ref:

Screenshot attached



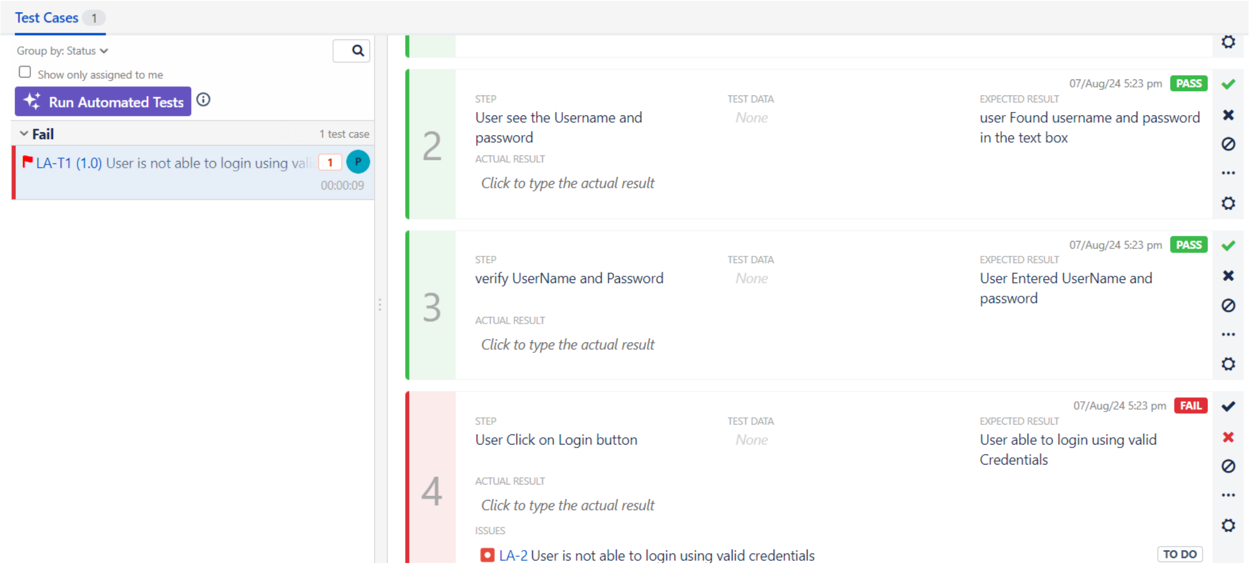
| ☐ | P | Key | V | Name | Status | R |
|---|---|---|---|---|---|---|
| ☐ | 🟥 | LA-T1 | 1.0 | User is not able to login using valid Credentials | APPROVED | ● |
| ☐ | 🟥 | LA-T2 | 1.0 | User is not able to remove item from the cart | APPROVED | ● |

**Fig-1: Creating Test Cases**



| ☐ | Key | Name | Progress | Status |
|---|---|---|---|---|
| ☐ | LA-R1 | Verify cart Features | 100% | DONE |
| ☐ | LA-R2 | Verify User Login | 100% | DONE |

**Fig-2: Created Test Cycles and add test cases to it**



Test Cases 1

Group by: Status

□ Show only assigned to me

Run Automated Tests

Fail — 1 test case

LA-T1 (1.0) User is not able to login using val 1 — 00:00:09

2 — STEP: User see the Username and password — TEST DATA: None — EXPECTED RESULT: user Found username and password in the text box — 07/Aug/24 5:23 pm PASS

ACTUAL RESULT: *Click to type the actual result*

3 — STEP: verify UserName and Password — TEST DATA: None — EXPECTED RESULT: User Entered UserName and password — 07/Aug/24 5:23 pm PASS

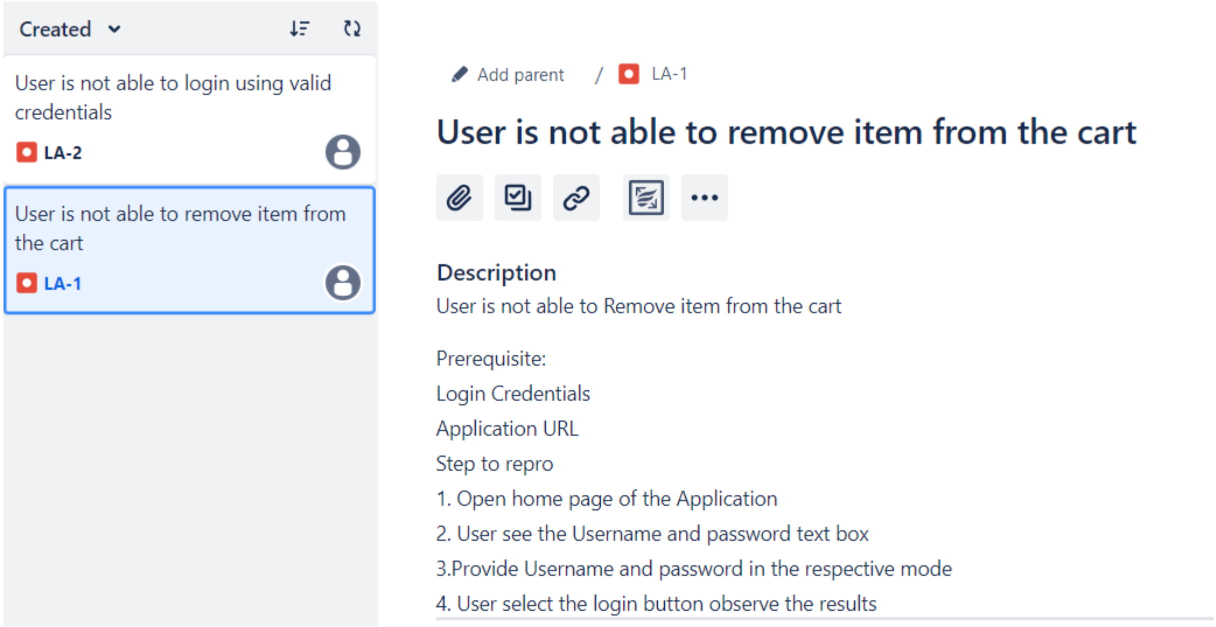
ACTUAL RESULT: *Click to type the actual result*

4 — STEP: User Click on Login button — TEST DATA: None — EXPECTED RESULT: User able to login using valid Credentials — 07/Aug/24 5:23 pm FAIL

ACTUAL RESULT: *Click to type the actual result*

ISSUES: LA-2 User is not able to login using valid credentials — TO DO

**<u>Fig-3:</u> Creating an issue**



Created

User is not able to login using valid credentials — LA-2

User is not able to remove item from the cart — LA-1

Add parent / LA-1

User is not able to remove item from the cart

Description

User is not able to Remove item from the cart

Prerequisite:

Login Credentials

Application URL

Step to repro

1. Open home page of the Application
2. User see the Username and password text box
3.Provide Username and password in the respective mode
4. User select the login button observe the results

**Fig-4**

## <u>Assignement-2:</u>

### SDLC Models for Engineering Projects

1. Waterfall Model

**Overview:** The Waterfall model is a linear and sequential approach where each phase must be completed before the next one begins. It's often depicted as a cascade of phases: Requirements, Design, Implementation, Testing, Deployment, and Maintenance.

### Advantages:

- **Simplicity**: Easy to understand and manage due to its sequential nature.
- **Structured Documentation**: Each phase has distinct deliverables and documentation, making tracking progress straightforward.
- **Clear Milestones:** Well-defined stages and deliverables provide clarity on project status and requirements.
  Disadvantages:
- **Inflexibility**: Changes are difficult and costly once a phase is completed.
- **Late Testing**: Testing only occurs after the implementation phase, potentially leading to the discovery of fundamental issues late in the project.
- **Assumption of Stability**: Assumes requirements are well understood and unlikely to change, which may not be true in dynamic environments.
  Applicability:
- **Best Suited For**: Projects with well-understood requirements and low likelihood of changes, such as regulatory or compliance-driven projects.
- **Less Suitable For**: Projects where requirements are expected to evolve or are not fully understood at the start.

## 2. Agile Model

**Overview**: Agile is an iterative and incremental approach that emphasizes flexibility, collaboration, and customer feedback. It involves frequent iterations (sprints) and regular reviews to adapt to changes.

### Advantages:

- **Flexibility**: Easily accommodates changes and evolving requirements.
- **Customer Collaboration**: Continuous feedback from stakeholders helps align the project with customer needs.
- **Early Delivery**: Incremental releases allow for early and frequent delivery of functional software.

### Disadvantages:

- **Scope Creep**: Frequent changes can lead to scope creep if not managed properly.
- **Requires Active Stakeholder Involvement**: Success relies heavily on constant communication and feedback from stakeholders.
- **Less Emphasis on Documentation:** May result in less comprehensive documentation compared to traditional methods.
  Applicability:
- **Best Suited For**: Projects with evolving requirements, complex or innovative solutions, and where customer feedback is crucial.

- **Less Suitable For:** Projects with fixed requirements and where thorough documentation and structured phases are required.

### 3. Spiral Model

**Overview**: The Spiral model combines iterative development with systematic aspects of the Waterfall model. It involves repetitive cycles (spirals) of planning, risk analysis, engineering, and evaluation.

#### Advantages:

- **Risk Management**: Focuses on risk assessment and management at each iteration, allowing for early detection and mitigation of issues.
- **Iterative Development**: Facilitates incremental development and continuous improvement.
- **Flexibility**: Allows for changes based on ongoing feedback and evolving requirements.

#### Disadvantages:

- **Complexity**: Can be complex to manage due to its iterative nature and multiple phases.
- **Cost**: Can be more expensive due to its iterative approach and focus on risk management.
- **Requires Expertise**: Effective implementation requires experienced project managers and engineers.

#### Applicability:

- **Best Suited For**: Large, complex projects with significant risks and uncertainties, such as advanced engineering and research projects.
- **Less Suitable For**: Smaller projects or projects with well-defined, stable requirements.

### 4. V-Model (Validation and Verification Model)

**Overview**: The V-Model is an extension of the Waterfall model that emphasizes validation and verification processes. Each development phase has a corresponding testing phase, creating a V-shaped diagram.

#### Advantages:

- **Emphasis on Testing**: Each phase of development is matched with a corresponding phase of testing, ensuring issues are identified and addressed early.
- **Clear Structure**: Provides a clear and structured approach with well-defined stages.
- Early Validation: Ensures that requirements are validated early in the development process.

## Disadvantages:

- **Inflexibility**: Like Waterfall, changes can be difficult and costly once a phase is completed.
- **Late Feedback**: Testing occurs after each development phase, potentially leading to late discovery of issues.
- **Assumes Fixed Requirements**: Assumes that requirements are stable, which may not be the case in dynamic environments.

## Applicability:

- **Best Suited For**: Projects requiring rigorous testing and validation, such as safety-critical systems in aerospace, automotive, and medical fields.
- **Less Suitable For**: Projects with high uncertainty or where requirements are expected to evolve frequently.