QA Workshop-Day 4

BIT – 5th Semester (Professional Class)

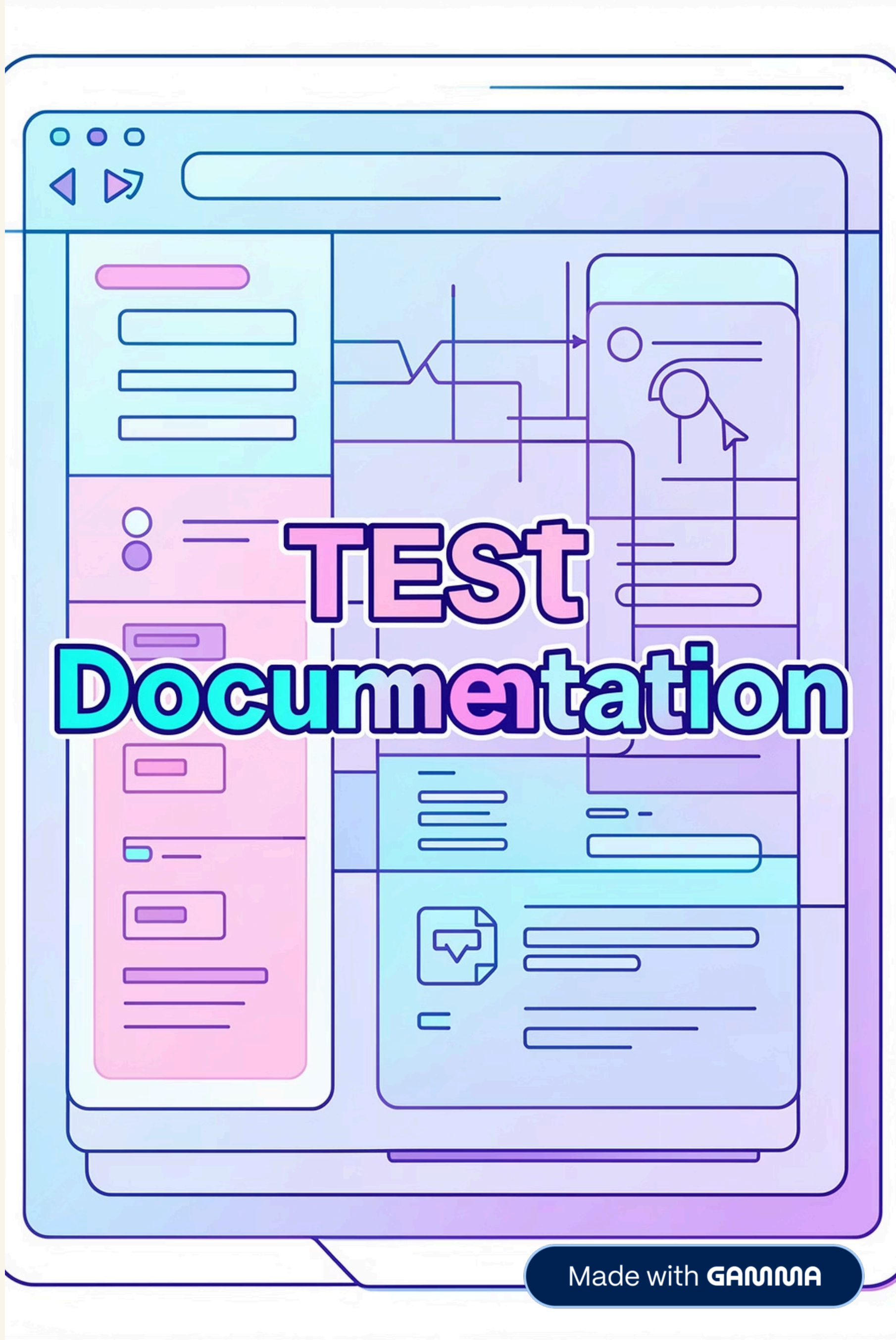
Facilitator: **Neha Rouniyar**

Test Documentation

Test documents help us plan testing properly, track issues clearly, and keep everything organized.

Today, we start an important lesson: learning how to :

- Write Complete Test Plans
- Make Detailed Test Cases
- Report and Track Bugs Well



DAY 4 OVERVIEW

Getting Good at Test Documents

01

What Test Documents Are

Learn the basic ideas and why good test notes are important.

02

How to Write Test Plans and Test Steps

Learn to write clear test plans and exact steps that help you test well.

03

How to Report Bugs Simply

Find out how to write about problems clearly and quickly so they can be fixed fast.

04

Practice Activities

Use what you learned with hands-on practice and real-life examples.

What is a Test Plan?

A Test Plan basically explains **what we will test**, **how we will test**, and **what risks** we may face. It's the key guide that helps your team understand what to test and how to do it.

What to Test & Why

It clearly says **which things we will test** and **which things we won't test**, so everyone knows the plan.

Shows How to Test

It explains **how the testing will be done** step by step, from start to finish.

Handles Problems & Time

It helps us understand **possible issues** and make plans so the testing can finish **on time.**

TEST PLAN ESSENTIALS

Key Components of a Comprehensive Test Plan

Scope & Objectives

What features are covered? What are the testing goals? Define boundaries and desired outcomes.

Test Approach & Strategy

How will testing be performed? Which techniques and methodologies will be used?

Test Environment

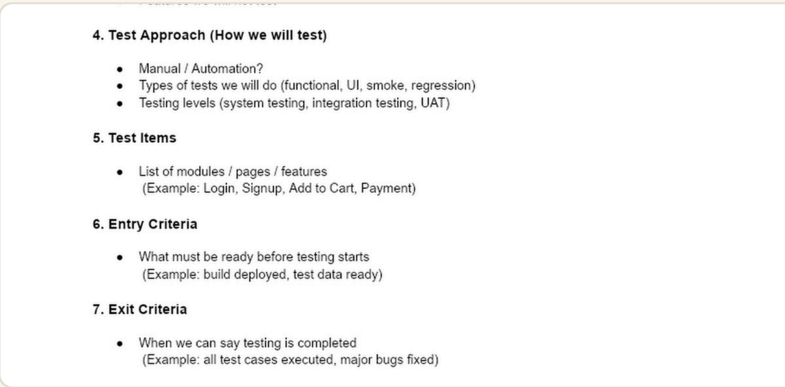
Details about hardware, software, and network configurations needed for testing.

Roles & Responsibilities

Who does what? Clearly assign duties to team members involved in the testing process.

Risks & Mitigation

Identify potential roadblocks and outline plans to minimize their impact.



4. Test Approach (How we will test)

- Manual / Automation?
- Types of tests we will do (functional, UI, smoke, regression)
- Testing levels (system testing, integration testing, UAT)

5. Test Items

- List of modules / pages / features
  (Example: Login, Signup, Add to Cart, Payment)

6. Entry Criteria

- What must be ready before testing starts
  (Example: build deployed, test data ready)

7. Exit Criteria

- When we can say testing is completed
  (Example: all test cases executed, major bugs fixed)

Google Docs

Test Plan Template

Test Plan Template 1. Introduction What the project is about Why we are testing it What this test plan will cover 2. Objective What we want to achieve (Example:...

Made with GAMMA

Crafting Effective Test Cases

Test cases are detailed, step-by-step instructions designed to verify specific functionalities. They ensure thorough coverage and consistent testing.

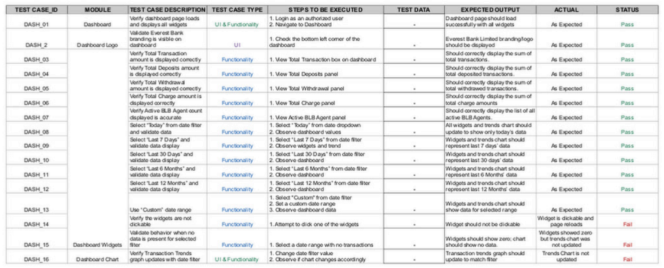
Break down features into precise steps

Cover both positive (expected) and negative (error) scenarios

Ensure repeatability and clarity for all testers

Below is the necessary field for effective test cases :

- Test Case ID
- Module , Sub - Module
- Test Case Description
- Test Case type (BE/FE/UI)
- Steps to be Executed
- Test Data
- Expected Output
- Actual Output
- Status (Pass/Fail)



TEST CASE_ID	MODULE	TEST CASE DESCRIPTION	TEST CASE TYPE	STEPS TO BE EXECUTED	TEST DATA	EXPECTED OUTPUT	ACTUAL	STATUS
DASH_01	Dashboard	Verify dashboard page loads and displays all widgets	UI & Functionality	1. Login as an authorized user 2. Navigate to Dashboard	-	Dashboard page should load successfully with all widgets	As Expected	Pass
DASH_2	Dashboard Logo	Validate Everest Bank branding is visible on dashboard	UI	1. Check the bottom left corner of the dashboard	-	Everest Bank Limited branding/logo should be displayed	As Expected	Pass
DASH_03	Dashboard Widgets	Verify Total Transaction amount is displayed correctly	Functionality	1. View Total Transaction box on dashboard	-	Should correctly display the sum of total transactions.	As Expected	Pass
DASH_04		Verify Total Deposits amount is displayed correctly	Functionality	1. View Total Deposits panel	-	Should correctly display the sum of total deposited transactions.	As Expected	Pass
DASH_05		Verify Total Withdrawal amount is displayed correctly	Functionality	1. View Total Withdrawal panel	-	Should correctly display the sum of total withdrawed transactions.	As Expected	Pass
DASH_06		Verify Total Charge amount is displayed correctly	Functionality	1. View Total Charge panel	-	Should correctly display the sum of total charge amounts	As Expected	Pass
DASH_07		Verify Active BLB Agent count displayed is accurate	Functionality	1. View Active BLB Agent panel	-	Should correctly display the list of all active BLB Agents.	As Expected	Pass
DASH_08		Select "Today" from date filter and validate data	Functionality	1. Select "Today" from date dropdown 2. Observe dashboard values	-	All widgets and trends chart should update to show only today's data	As Expected	Pass
DASH_09		Select "Last 7 Days" and validate data display	Functionality	1. Select "Last 7 Days" from date filter 2. Observe widgets and trend	-	Widgets and trends chart should represent last 7 days' data	As Expected	Pass
DASH_10		Select "Last 30 Days" and validate data display	Functionality	1. Select "Last 30 Days" from date filter 2. Observe dashboard	-	Widgets and trends chart should represent last 30 days' data	As Expected	Pass
DASH_11		Select "Last 6 Months" and validate data display	Functionality	1. Select "Last 6 Months" from date filter 2. Observe dashboard	-	Widgets and trends chart should represent last 6 Months' data	As Expected	Pass
DASH_12		Select "Last 12 Months" and validate data display	Functionality	1. Select "Last 12 Months" from date filter 2. Observe dashboard	-	Widgets and trends chart should represent last 12 Months' data	As Expected	Pass
DASH_13		Use "Custom" date range	Functionality	1. Select "Custom" from date filter 2. Set a custom date range 3. Observe dashboard data	-	Widgets and trends chart should show data for selected range	As Expected	Pass
DASH_14		Verify the widgets are not clickable	Functionality	1. Attempt to click one of the widgets	-	Widget should not be clickable	Widget is clickable and page reloads	Fail
DASH_15		Validate behavior when no data is present for selected filter	Functionality	1. Select a date range with no transactions	-	Widgets should show zero; chart should show no data.	Widgets showed zero but trends chart was not updated	Fail
DASH_16	Dashboard Chart	Verify Transaction Trends graph updates with date filter	UI & Functionality	1. Change date filter value 2. Observe if chart changes accordingly	-	Transaction trends graph should update to match filter	Trends Chart is not updated	Fail

Google Docs

Test Case – QA Workshop

TEST CASE IN ACTION

Example: Valid Login Scenario

This test case verifies that a user can successfully log in with correct credentials, ensuring the primary access path functions as expected.

1

Test Case: Login with Valid Credentials

- **ID:** TC_LOGIN_001
- **Priority:** High
- **Precondition:** User account exists

2

Steps to Execute

1. Open the application's login page
2. Enter a valid email address into the 'Username' field
3. Enter the correct password into the 'Password' field
4. Click the "Login" button

3

Expected Outcome

The user should be successfully redirected to the main dashboard or home page, and a "Welcome, [Username]!" message should be displayed.

HANDLING ERRORS

Example: Invalid Login Scenario

Testing negative scenarios is crucial. This test case ensures the system properly handles incorrect inputs and provides appropriate feedback, protecting against unauthorized access.

Test Case: Login with Invalid Password

- **ID:** TC_LOGIN_002
- **Priority:** Medium
- **Precondition:** User account exists

Steps to Execute

1. Navigate to the application's login page
2. Enter a valid email address
3. Enter an incorrect or expired password
4. Click the "Login" button

Expected Outcome

An error message such as "Invalid credentials" or "Incorrect password" should be prominently displayed, and the user should remain on the login page.

Bug Reporting: Essential Elements

Effective bug reports are concise, clear, and actionable. They provide developers with all the necessary information to quickly identify, reproduce, and fix defects.

- **Steps to Reproduce:** Precise, numbered actions that lead to the bug.
- **Expected vs. Actual Result:** What should have happened versus what actually occurred.
- **Severity & Priority:** How critical is the bug (e.g., Blocker, Major)? How urgent is the fix (e.g., High, Low)?
- **Environment Details:** Browser, OS, device, version numbers where the bug was found.
- **Attachments:** Screenshots, video recordings, or console logs to provide visual evidence.

BUG REPORTING BEST PRACTICES

Good vs. Bad Bug Reports

Good Bug Report

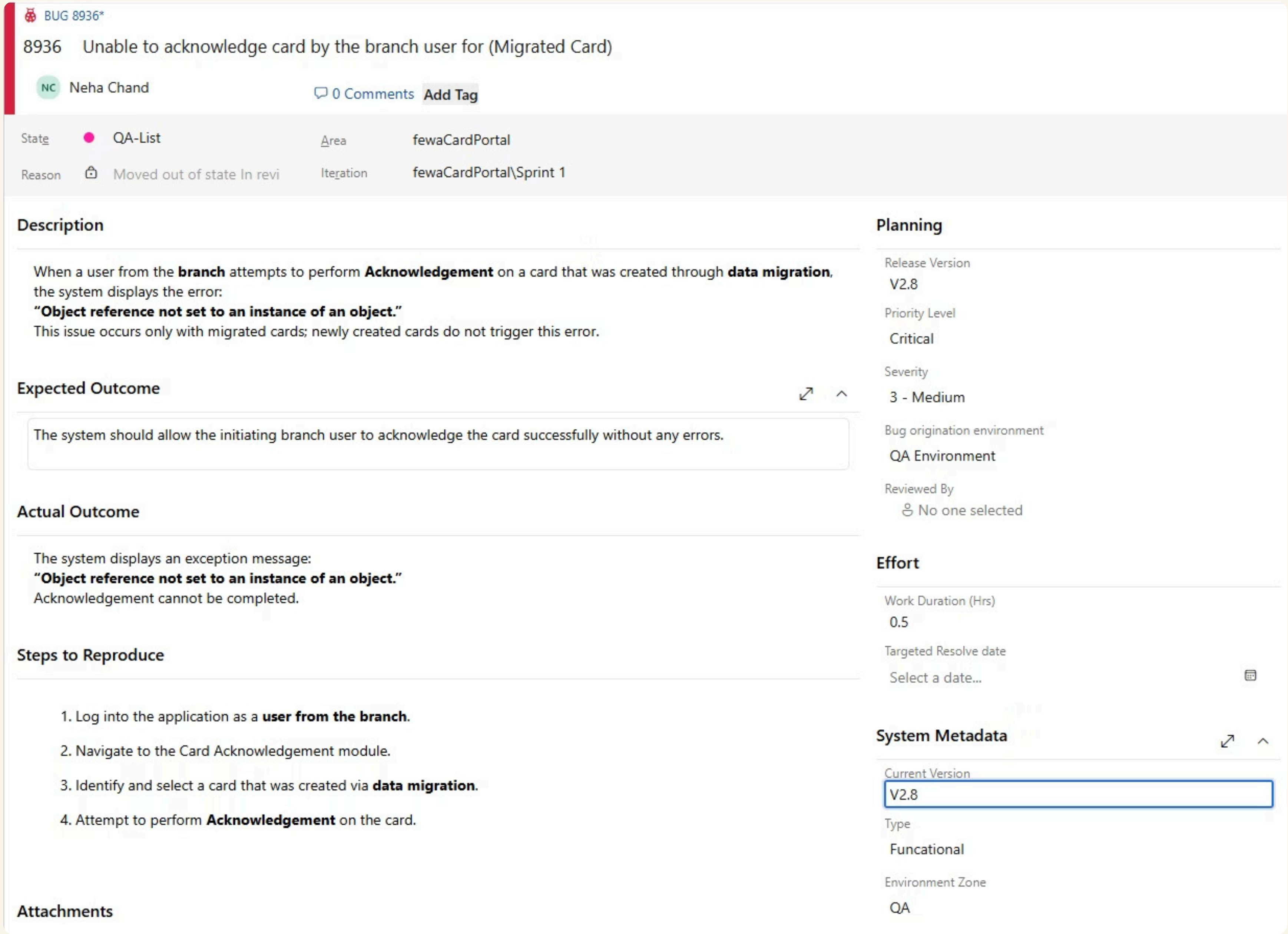
Title: User cannot log in with valid credentials on Chrome

- **Description:** User is unable to log into the application using valid credentials when using Chrome browser.
- **Steps:**
  - 1. Open Chrome.
  - 2. Go to login page.
  - 3. Enter 'test@example.com' and 'password123'.
  - 4. Click Login.
- **Expected:** User logs in successfully.
- **Actual:** Error message "Login failed" is displayed, and user remains on login page.
- **Severity:** Blocker **Priority:** High
- **Screenshot:** attached.

Bad Bug Report

Title: Login button broken

- **Description:** Can't log in. Button doesn't work.
- **Steps:** Tried to log in.
- **Expected:** Log in.
- **Actual:** No log in.
- **Severity:** Don't know **Priority:** ASAP
- **Screenshot:** None.



BUG 8936*

8936 Unable to acknowledge card by the branch user for (Migrated Card)

NC Neha Chand

0 Comments Add Tag

State ● QA-List

Reason Moved out of state In revi

Area fewaCardPortal

Iteration fewaCardPortal\Sprint 1

Description

When a user from the **branch** attempts to perform **Acknowledgement** on a card that was created through **data migration**, the system displays the error:
"Object reference not set to an instance of an object."
This issue occurs only with migrated cards; newly created cards do not trigger this error.

Expected Outcome

The system should allow the initiating branch user to acknowledge the card successfully without any errors.

Actual Outcome

The system displays an exception message:
"Object reference not set to an instance of an object."
Acknowledgement cannot be completed.

Steps to Reproduce

1. Log into the application as a **user from the branch**.
2. Navigate to the Card Acknowledgement module.
3. Identify and select a card that was created via **data migration**.
4. Attempt to perform **Acknowledgement** on the card.

Attachments

Planning

Release Version
V2.8

Priority Level
Critical

Severity
3 - Medium

Bug origination environment
QA Environment

Reviewed By
No one selected

Effort

Work Duration (Hrs)
0.5

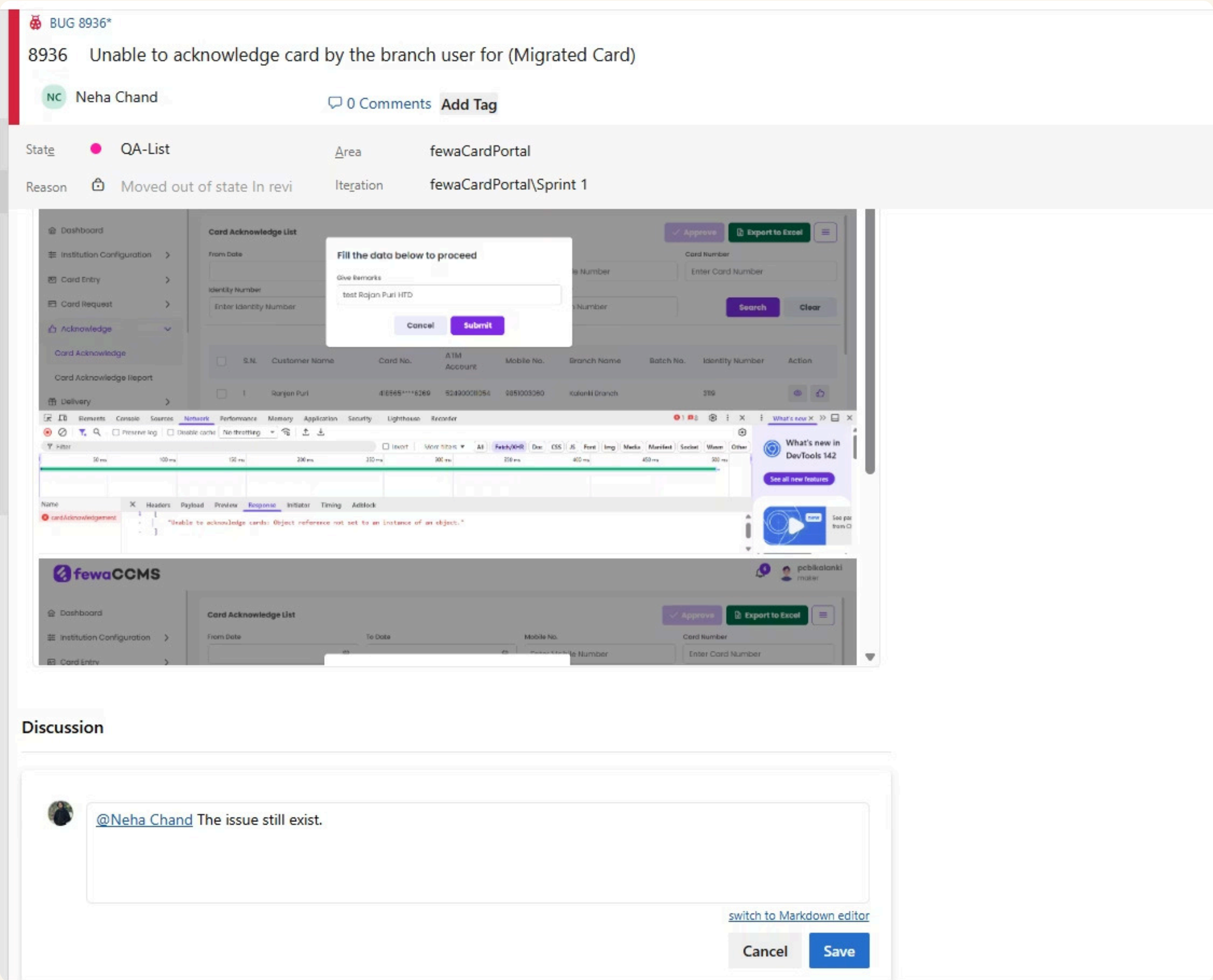
Targeted Resolve date
Select a date...

System Metadata

Current Version
V2.8

Type
Funcational

Environment Zone
QA



BUG 8936*

8936 Unable to acknowledge card by the branch user for (Migrated Card)

NC Neha Chand

0 Comments Add Tag

State ● QA-List

Area fewaCardPortal

Reason Moved out of state In revi

Iteration fewaCardPortal\Sprint 1

Card Acknowledge List

Fill the data below to proceed

Give Remarks: test Rajan Puri HTD

	S.N.	Customer Name	Card No.	ATM Account	Mobile No.	Branch Name	Batch No.	Identity Number	Action
□	1	Ranjan Puri	418565****6269	524900001054	9851003060	Kalanki Branch		319	

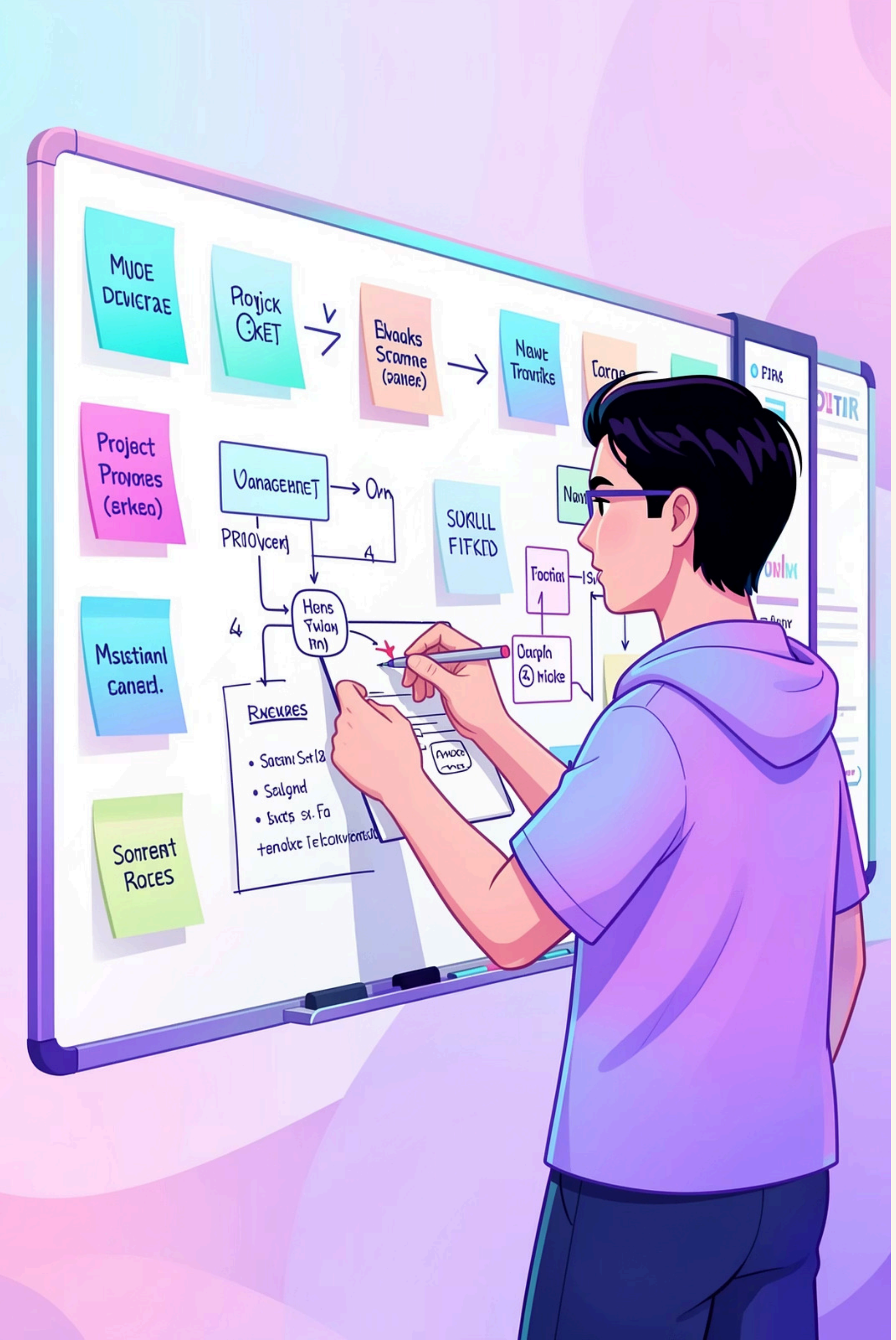
```
[
  "Unable to acknowledge cards: Object reference not set to an instance of an object."
]
```

Discussion

@Neha Chand The issue still exist.

switch to Markdown editor

Cancel Save



Today's Lessons:

01

Test Documentation Basics

02

Writing Test Plans

03

Crafting Test Cases

04

Bug Reporting Best Practices

Activity

🚗 buggy.justtestit.org

Buggy Cars Rating

1

Activity 1: Mini Test Plan Draft

Create a **1-page Test Plan** for the **Buggy Cars Rating** website(above given link).

Cover key sections:

- Scope (what parts of the site you will test)
- Out of Scope
- Test Approach (functional, UI, validation)
- Test Environment
- Risks & Assumptions
- Test Deliverables

2

Activity 2: Login Page – Test Cases

Prepare **5 clear test cases** for the **Login page**, including:

- Valid login
- Invalid username/password
- Empty fields
- Field validation checks
- UI elements (button state, error message display)

3

Activity 3: Discussion & Review

Work with your classmates to:

- Present your test plans and test cases
- Share bugs found during the login flow
- Discuss challenges in documenting test steps
- Refine your work based on peer feedback

Questions & Discussion