National University of Computer and Emerging Sciences Chiniot-Faisalabad Campus



**metabase**

Software Quality Engineering

### Semester Project

### Phase 2

### BS (SE) 5A

## Team Name: zzwave-testing

## Team Member’s:

## 21F-9516 -> Suresh Kumar

## 21F-9519 -> Rai umer farooq

## 21f-9510 -> mian fahad akhtar

Table of Contents

[RISK ANALYSIS: - 2](#_Toc149859595)

[Security Risks: 2](#_Toc149859596)

[Data Integrity Risks: 2](#_Toc149859597)

[Performance Risks: 2](#_Toc149859598)

[Equivalence Class Partitioning (ECP): - 2](#_Toc149859599)

[Username and Password Inputs: 2](#_Toc149859600)

[Dashboard Widgets: 2](#_Toc149859601)

[Boundary Value Analysis (BVA): - 2](#_Toc149859602)

[Number of Dashboard Widgets: 2](#_Toc149859603)

[User Load Handling: 3](#_Toc149859604)

[TEST CASES: - 3](#_Toc149859605)

**APPLICATION: METABASE**

**MODULE: DASHBOARD MODULE**

# RISK ANALYSIS: -

## Security Risks:

Analyze the risk of unauthorized access to sensitive data displayed on the dashboard. Implement appropriate security measures such as encryption and access controls.

## Data Integrity Risks:

Assess the risk of data loss or corruption during sudden system failures. Develop strategies to ensure data integrity and continuity in case of unexpected outages.

## Performance Risks:

Evaluate the risk of the dashboard's performance degradation with an increasing number of widgets or simultaneous users. Implement performance testing to identify and mitigate potential bottlenecks.

# Equivalence Class Partitioning (ECP): -

## Username and Password Inputs:

ECP would categorize inputs into valid and invalid classes. Valid classes may include a typical username and password combination, while invalid classes may include empty fields or inputs not conforming to the username and password format.

## Dashboard Widgets:

ECP can be applied to various types of widgets. Valid classes may include text widgets, charts, and graphs, while invalid classes may include unsupported or corrupt widgets.

# Boundary Value Analysis (BVA): -

## Number of Dashboard Widgets:

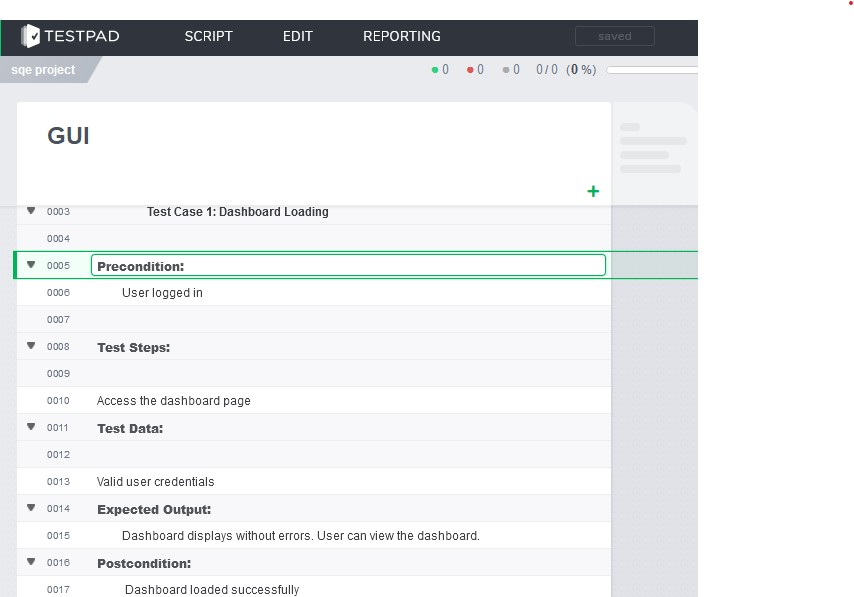
BVA would test the minimum and maximum numbers of widgets that the dashboard can handle. For example, testing the behavior of the dashboard with 0, 1, and 10+ widgets.

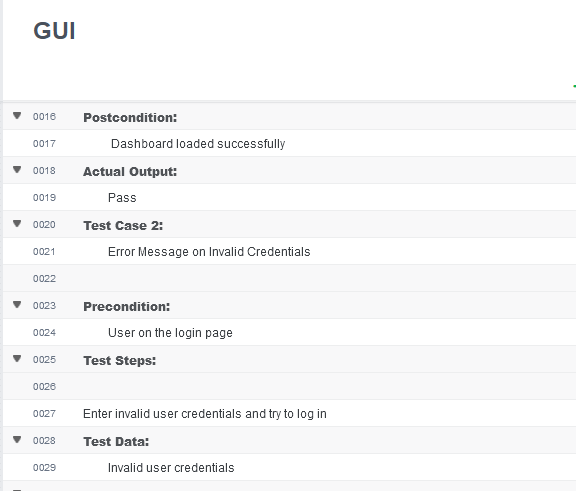
## User Load Handling:

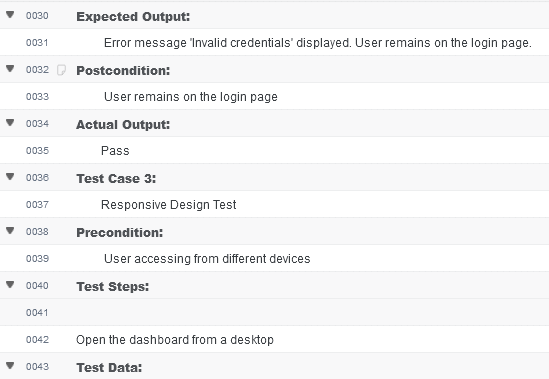
BVA can be used to determine the system's behavior under minimum and maximum user loads. For instance, testing the dashboard's response with 1, 100, and 1000 simultaneous users.

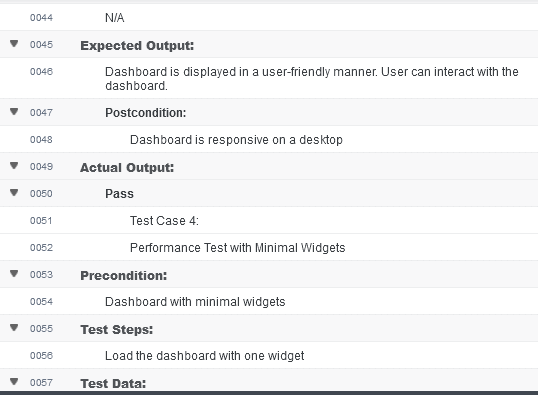
# TEST CASES: -

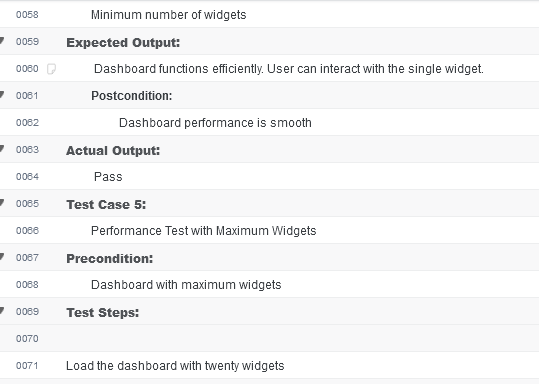
**GUI: -**

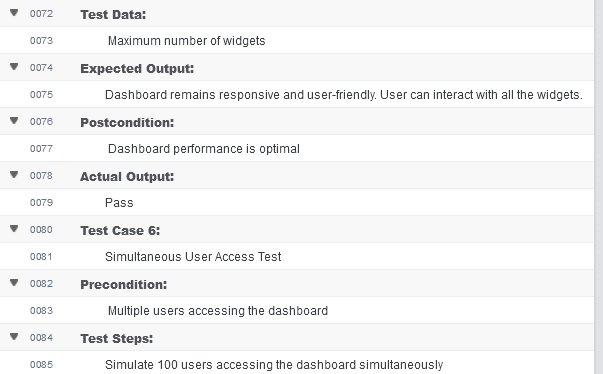


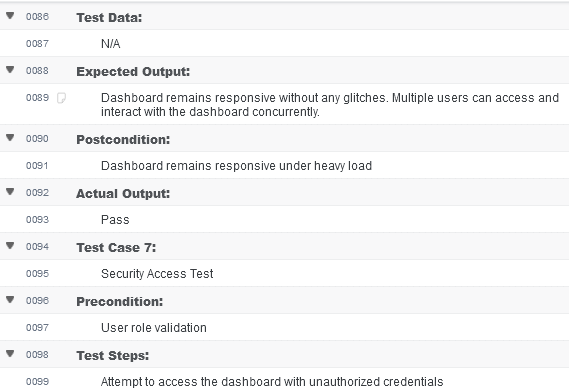


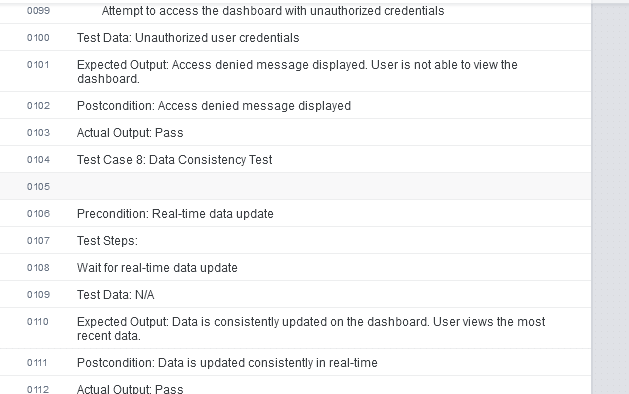


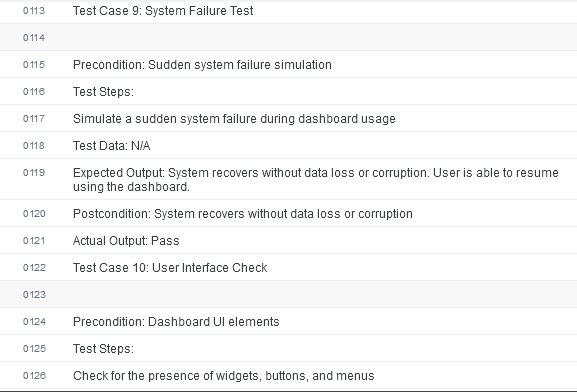


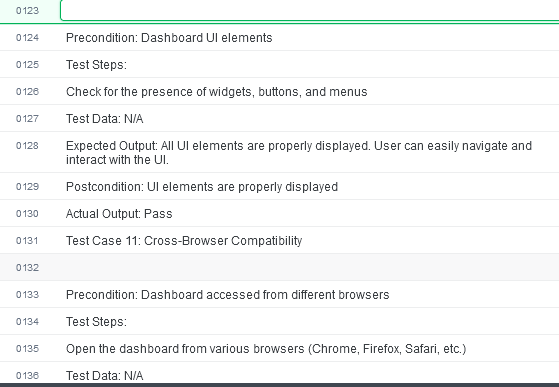


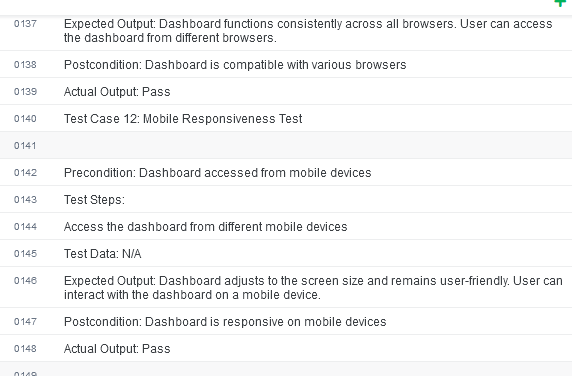












**API: -**

