Introduction to Software Engineering

# Software Testing

The student team is required to complete the **Software Testing** documentation for the assigned course project, following the attached template.


Software Engineering Department
Faculty of Information and Technology
University of Science

# Table of Contents

**Objectives** 1

**1 Member Contribution Assessment** 2

**2 Test plan** 3

**3 Test cases** 4

3.1 List of test cases 4

3.2 Test case specifications 4

3.2.1 Test case 1 4

# Software Testing

## Objectives

This document focus on the following topics:

- ✔ Completing the Software Testing document with the following sections:
  - Test Plan
  - Test Cases
- ✔ Understanding the Software Testing document.

# 1 Member Contribution Assessment

| ID | Name | Contribution (%) | Signature |
|---|---|---|---|
| 23127006 | Trần Nguyễn Khải Luân | 20% | |
| 23127113 | Nguyễn Trần Phú Quý | 20% | |
| 23127144 | Đinh Đại Vũ | 20% | |
| 23127179 | Nguyễn Bảo Duy | 20% | |
| 23127189 | Trần Trọng Hiếu | 20% | |

# 2 Test plan

**Important Note on Scope Adjustment:** Based on the re-evaluation of project timeline and resources, consistent with the Data Design in PA2, the team has decided to remove the **Authentication, Social Network, and Content Moderation** modules. The project now focuses entirely on the **Analytics & Team Builder** core features. Consequently, testing objects related to User Accounts and Posting have been omitted.

## 2.1 Testing strategy

- **Black-box Testing:** Focuses on Functional Testing at the GUI Tier (React) to ensure business requirements (Team Builder, Top Comps viewing, Player Search) function as described in the SRS without inspecting the internal code structure.
- **Integration Testing:** Verifies the data communication between the Frontend, the Backend API Services, and the Database. This ensures JSON responses are correctly parsed and displayed on the client side.
- **Validation Testing:** Ensures that data constraints and logic defined in the design are enforced. For example: A unit must have a maximum of 3 items; Team synergy must update immediately upon unit placement.

## 2.2 Object to be tested

### 2.2.1 Functional Objects

- **Data & Stats Services:**
  - Fetch Data: Retrieve match statistics and rankings (U001, U002).
  - Search: Query player information and match history via Riot API Proxy (U003).
- **Core Business Logic:**

- Synergy Calculation: Compute real-time synergy scores based on units on the board (U004).
- Validation Services: Validate team compositions (Max items per unit, valid unit placement) (U004).

#### 2.2.2 Document/User Interface Objects (Documents/UI)

- **Main Information Pages:**
  - Top Comps Page: Ranking table showing win/pick rate and trends.
  - Game Stats Pages: Detailed pages for champions, traits, and items.
  - Player Profile: Interface displaying player rank, LP, and match history.
- **Interactive User Interfaces:**
  - Team Builder UI: Complex drag-and-drop interface for building teams, equipping items, and viewing active traits.

#### 2.2.3 System Non-functional Attributes

- **Performance:** System response time for heavy data pages (Top Comps).
- **Responsiveness:** Layout adaptability on Mobile and Desktop screens.

# 3 Test cases

## 3.1 List of test cases

| Seq | Test Case Name | Target Feature | Description |
|---|---|---|---|
| TC01 | View Top Comps - Sorting | Top Comps Page | Verify sorting by Win Rate, Pick Rate, and Avg Place. |
| TC02 | View Top Comps - Sorting and Filtering | Top Comps Page | Verify filtering data by specific Ranks (e.g., Emerald+) and Patch versions. |

| | | | |
|---|---|---|---|
| TC03 | View Unit Details | Unit Stats Page | Verify the display of unit costs, stats, and recommended items. |
| TC04 | Item Filtering | Item Stats Page | Verify filtering items by type (Craftable, Radiant, Artifact). |
| TC05 | Search Player - Valid | Player Search | Verify finding an existing player by the correct Riot ID. |
| TC06 | Search Player - Invalid | Player Search | Verify error handling when searching for a non-existent player. |
| TC07 | Team Builder - Drag & Drop | Team Builder | Verify placing units from the list onto the Hex Grid. |
| TC08 | Team Builder - Traits | Team Builder | Verify that Trait counts update correctly when units are added/removed. |
| TC09 | Team Builder - Items | Team Builder | Verify equipping items to units and checking the "Max 3 items" rule. |
| TC10 | Team Builder - Stars | Team Builder | Verify toggling unit star levels (1★ → 2★ → 3★) via hovering and clicking. |
| TC11 | Share Team Code | Team Builder | Verify generating a text string/link representing the current board state. |

| | | | |
|---|---|---|---|
| TC12 | Clear Board | Team Builder | Verify the "Reset" button clears all units and items. |
| TC13 | Performance Load | System-wide | Verify that the initial data load takes less than 5 seconds. |
| TC14 | Mobile Responsiveness | UI Layout | Verify that the Team Builder grid is usable on mobile screen sizes. |

## 3.2 Test case specifications

### *3.2.1 Test case 1*

| ***Test case*** | **TC01 - View Top Comps - Data Display** |
|---|---|
| **Related Use case** | U001 - View Top Team Comps |
| **Context** | 1. Backend API is running and serving meta data.<br>2. The user navigates to the "Top Comps" tab. |
| **Input Data** | Action: Load the page (Default view). |
| **Expected Output** | 1. A list of team compositions is displayed.<br>2. Each row shows: Comp Name (e.g., "Aatrox, Galio..."), Core Units, Play Rate, and Average Place.<br>3. The list is populated, not empty. |
| **Test steps** | 1. Navigate to /top-comps.<br>2. Observe the data grid.<br>3. Verify the first entry is the top comp name and champions which belong to that comp. |

| | |
|---|---|
| **Actual Output** | As expected. The table displays compositions with the stats: Play Rate 0.1, Place 3.87. |
| **Result** | **Passed.** |

### *3.2.2 Test case 2*

| ***Test case*** | **TC02 - View Unit Stats - Tier Grouping** |
|---|---|
| **Related Use case** | U002 - View Game Stats |
| **Context** | The user navigates to the "Units" tab. |
| **Input Data** | Action: Scroll through the unit grid. |
| **Expected Output** | 1. Champions are grouped visually by Cost/Tier.<br><br>2. Each unit card displays: Portrait, Name, Avg Place, and Top 4 Rate.<br><br>3. Background colors distinguish different Tiers (e.g., Brown for Tier 1, Green for Tier 2). |
| **Test steps** | 1. Click on "UNITS" in the navigation bar.<br><br>2. Verify "Ezreal" appears in "Tier 1".<br><br>3. Verify "Vi" appears in "Tier 2". |
| **Actual Output** | As expected. Units are correctly categorized by Tier with visible statistics. |
| **Result** | **Passed.** |

### 3.2.3 Test case 3

| ***Test case*** | **TC03 - View Item Stats - Data Integrity** |
|---|---|
| **Related Use case** | U002 - View Game Stats |
| **Context** | The user navigates to the "Items" tab. |

| | |
|---|---|
| **Input Data** | Action: Observe the Item Table. |
| **Expected Output** | 1. A list of items is displayed with columns: Item Name, Play Rate, Place, Top 4, Win, and Top Users.<br><br>2. "Top Users" column displays champion icons using that item. |
| **Test steps** | 1. Click on "ITEMS" in the navigation bar.<br><br>2. Check the row for "Spear of Shojin".<br><br>3. Verify stats are numeric (ex: Play rate 1.00). |
| **Actual Output** | As expected. Item list loads correctly with all statistical columns and top user icons. |
| **Result** | **Passed.** |

### 3.2.4 Test case 4

| ***Test case*** | **TC04 - View Trait Stats** |
|---|---|
| **Related Use case** | U002 - View Game Stats |
| **Context** | The user navigates to the "Traits" tab. |
| **Input Data** | Action: Observe the Trait Table. |
| **Expected Output** | 1. A list of traits is displayed.<br><br>2. Columns include: Trait Name, Play Rate, Place, Top 4, Win.<br><br>3. Trait icons are visible next to the names. |
| **Test steps** | 1. Click on "TRAITS" in the navigation bar.<br><br>2. Verify "Star Guardian" is listed.<br><br>3. Check if Play Rate/Win Rate data is populated. |

| | |
|---|---|
| **Actual Output** | As expected. Trait statistics are displayed clearly. |
| **Result** | **Passed.** |

### 3.2.5 Test case 5

| ***Test case*** | **TC05 - Search Player - Valid** |
|---|---|
| **Related Use case** | U003 - Player Information Lookup. |
| **Context** | The user navigates to the "Players" tab. |
| **Input Data** | 1. Data: Enter a valid Riot ID (GameName + TagLine). Example: "Luke#TNKL" or "BuhDuy#VN".<br><br>2. Action: Press the "Enter" key or click the Search icon. |
| **Expected Output** | 1. The system navigates to the Player Profile Page.<br><br>2. The profile header displays the correct Name, Tag, and current Rank Tier.<br><br>3. A list of recent match history is loaded below the header. |
| **Test steps** | 1. Click on the "PLAYER" in the navigation bar.<br><br>2. Type "Luke#TNKL".<br><br>3. Click the Search button.<br><br>4. Verify that the URL changes to /profile/Luke-TNKL and profile data loads. |
| **Actual Output** | The feature is not implemented yet. |
| **Result** | **Not run.** |

### 3.2.6 Test case 6

| ***Test case*** | **TC06 - Search Player - Invalid** |
|---|---|

| | |
|---|---|
| **Related Use case** | U003 - Player Information Lookup. |
| **Context** | The user navigates to the "Players" tab. |
| **Input Data** | 1. Data: Enter a non-existent Riot ID. Example: "ThisUserDoesNotExist#12345".<br><br>2. Action: Press the "Enter" key or click the Search icon. |
| **Expected Output** | A visible error message appears stating: "Player not found". |
| **Test steps** | 1. Click on the Search Bar.<br><br>2. Type "ThisUserDoesNotExist#12345".<br><br>3. Press Enter.<br><br>4. Observe the UI response. |
| **Actual Output** | The feature is not implemented yet. |
| **Result** | **Not run.** |

### 3.2.7 Test case 7

| ***Test case*** | **TC07 - Team Builder - Drag & Drop** |
|---|---|
| **Related Use case** | U004 - Team Builder Create & Test Teams |
| **Context** | 1. The user is accessing the Team Builder page.<br><br>2. The Unit Selector list (bottom panel) is successfully loaded. |
| **Input Data** | Action: Click and hold the unit "Jinx" from the Unit Selector.<br><br>Action: Drag the unit to the coordinate [Row 2, Column 3] on the Hex Grid.<br><br>Action: Release the mouse button (Drop). |

| | |
|---|---|
| **Expected Output** | 1. The unit "Jinx" remains fixed on the specific hex tile where it was dropped.<br><br>2. The unit's model/icon is clearly visible on the board. |
| **Test steps** | 1. Navigate to the /team-builder page.<br><br>2. Locate "Jinx" in the unit list.<br><br>3. Perform drag-and-drop action onto the board.<br><br>4. Observe the board state. |
| **Actual Output** | The feature is not implemented yet. |
| **Result** | **Not Run.** |

### 3.2.8 Test case 8

| ***Test case*** | **TC08 - Team Builder - Traits** |
|---|---|
| **Related Use case** | U004 - Team Builder Create & Test Teams |
| **Context** | 1. The user is on the Team Builder page.<br><br>2. The board is currently empty. |
| **Input Data** | Action 1: Drag and drop "Garen" (Bastion trait) onto the board.<br><br>Action 2: Drag and drop "Leona" (Bastion trait) onto the board.<br><br>Action 3: Drag "Leona" off the board to remove her. |
| **Expected Output** | 1. After Action 1: The "Active Traits" sidebar shows "Bastion: 1/2" (Inactive/Grey).<br><br>2. After Action 2: The sidebar updates to "Bastion: 2/2" (Active/Bronze color), indicating the synergy is active .<br><br>3. After Action 3: The sidebar reverts to "Bastion: 1/2". |

| | |
|---|---|
| **Test steps** | 1. Navigate to /team-builder.<br>2. Drag unit "Garen" to the board.<br>3. Observe Trait Sidebar.<br>4. Drag unit "Leona" to the board.<br>5. Observe Trait Sidebar update.<br>6. Remove "Leona" and observe the update. |
| **Actual Output** | The feature is not implemented yet. |
| **Result** | **Not Run.** |

### 3.2.9 Test case 9

| ***Test case*** | **TC09 - Team Builder - Items** |
|---|---|
| **Related Use case** | U004 - Team Builder Create & Test Teams |
| **Context** | 1. The user is on the Team Builder page.<br>2. A unit (e.g., "Jinx") is already placed on the board. |
| **Input Data** | Action 1: Drag "Infinity Edge" from Item Selector to Jinx.<br>Action 2: Drag "Blue Buff" to Jinx.<br>Action 3: Drag "Giant Slayer" to Jinx.<br>Action 4: Attempt to drag "Warmog's Armor" (4th item) to Jinx. |
| **Expected Output** | 1. After Actions 1-3: The item icons appear attached to the Jinx unit model on the board.<br>2. After Action 4: The system rejects the 4th item (The item snaps back to the selector or a visual error indicator appears). The unit remains with only the first 3 items . |
| **Test steps** | 1. Drag "Jinx" to the board. |

| | |
|---|---|
| | 2. Drag 3 different items onto Jinx one by one.<br>3. Verify all 3 are equipped.<br>4. Try to drag a 4th item onto Jinx.<br>5. Verify the 4th item is not equipped. |
| **Actual Output** | The feature is not implemented yet. |
| **Result** | **Not Run.** |

### 3.2.10 Test case 10

| ***Test case*** | **TC10 - Team Builder - Stars** |
|---|---|
| **Related Use case** | U004 - Team Builder Create & Test Teams |
| **Context** | 1. The user is on the Team Builder page.<br>2. A unit (Jinx) is placed on the board (Default: 1-star). |
| **Input Data** | Action 1: Drag a unit "Jinx" and drop on the board.<br>Action 2: Right click on that unit. |
| **Expected Output** | 1. After Action 1: Star icons appear and allow user to choose star level for Jinx<br>2. After Action 2: Jinx toggles to 2-star or 3-star status |
| **Test steps** | 1. Place a unit on the board.<br>2. Right click on that unit.<br>3. Observe the star level of that unit.<br>4. Right click on that unit again.<br>5. Observe the star level of that unit. |
| **Actual Output** | The feature is not implemented yet. |

| | |
|---|---|
| **Result** | **Not Run.** |

### 3.2.11 Test case 11

| ***Test case*** | **TC11 - Share Team Code** |
|---|---|
| **Related Use case** | U004 - Team Builder Create & Test Teams |
| **Context** | 1. The user is on the Team Builder page.<br><br>2. The board is not empty (at least 1 unit is placed). |
| **Input Data** | Action: Click on the "Copy Code" button in the control panel. |
| **Expected Output** | 1. A notification appears: "Team Code copied to clipboard!".<br><br>2. The system generates a json string representing the current units and items.<br><br>3. The user's clipboard contains this generated string. |
| **Test steps** | 1. Place units "Jinx" and "Vi" onto the board.<br><br>2. Click the "Copy Code" button.<br><br>3. Open a Notepad or Text Editor.<br><br>4. Press Ctrl + V (Paste).<br><br>5. Verify that a code string appears in a text string (or code) in the text editor. |
| **Actual Output** | The feature is not implemented yet. |
| **Result** | **Not run.** |

### 3.2.12 Test case 12

| ***Test case*** | **TC12 - Clear Board** |
|---|---|

| | |
|---|---|
| **Related Use case** | U004 - Team Builder Create & Test Teams |
| **Context** | 1. The user is on the Team Builder page.<br>2. The board contains multiple units and items. |
| **Input Data** | Action: Click on the "Clear Board" button. |
| **Expected Output** | 1. All units are immediately removed from the Hex Grid.<br>2. All equipped items are removed.<br>3. The "Active Traits" sidebar resets to empty.<br>4. The board returns to the initial blank state. |
| **Test steps** | 1. Drag random units to fill the board.<br>2. Verify the board is full.<br>3. Click the "Clear Board" button.<br>4. Observe the board and the traits sidebar. |
| **Actual Output** | The feature is not implemented yet. |
| **Result** | **Not run.** |

### 3.2.13 Test case 13

| ***Test case*** | **TC13 - Performance Load** |
|---|---|
| **Related Use case** | Non-functional Requirement (NR1 - Performance) |
| **Context** | 1. The application is deployed on a standard server (or Localhost).<br>2. Internet connection is stable. |
| **Input Data** | Action: Perform a "Refresh" (Ctrl + F5) on the "Top Comps" page. |

| | |
|---|---|
| **Expected Output** | 1. The main content (ranking table) is visible and interactive.<br><br>2. The total load time is less than 5 seconds. |
| **Test steps** | 1. Open Google Chrome DevTools (F12).<br><br>2. Go to the "Network" tab.<br><br>3. Navigate to /top-comps.<br><br>4. Observe the "Load" time at the bottom of the DevTools panel. |
| **Actual Output** | As expected. The page content fully loaded in 1.5 ~ 2.5 seconds after 5 trials. The worst attempt is 3.3 seconds. |
| **Result** | **Passed.** |

### 3.2.14 Test case 14

| ***Test case*** | **TC14 - Mobile Responsiveness** |
|---|---|
| **Related Use case** | Non-functional Requirement (NR2 - Usability) |
| **Context** | 1. The user is accessing the site via a mobile device (or Mobile Simulator).<br><br>2. Screen width is set to 912px (iPhone 14 Pro Max). |
| **Input Data** | Action: Navigate to the "Top Comps" page and scroll through the list. |
| **Expected Output** | 1. The ranking table/list adapts to the screen width (Responsive Design).<br><br>2. Text and Unit Icons are legible and not overlapping. |
| **Test steps** | 1. Open Chrome DevTools (F12).<br><br>2. Toggle the "Device Toolbar" (Ctrl + Shift + M).<br><br>3. Select "iPhone 14 Pro Max". |

| | |
|---|---|
| | 4. Reload the page and observe the layout.<br>3. Select "iPad Air".<br>4. Reload the page and observe the layout. |
| **Actual Output** | Layout Issue. The table behaves correctly on Laptop/Tablet but overflows horizontally on mobile screens. |
| **Result** | **Failed.** |