Test Plan and Test Cases

User Stories

You need to be logged in to view this page

Log in to Confluence

Functional Testing

US1: As a user I want to select my study type before I start planning my course (Undergraduate, Graduate…)

- [ ] **TC1:** Verify that the user can successfully select their study type and save the information, so that the system provides the correct course options for planning.

Test Case for User Story 1
Test Type: Functional
Exection Type: Manual
Objective: Ensure that the user can successfully select their study type and save the information, so that the system provides the correct course options for planning.
Setup: 1. User has access to the welcome page. 2. User is logged out or accessing the site for the first time.
Pre-Conditions: 1. User should not have a study type previously selected. 2. System is functioning correctly, and study types are available for selection.
Notes: Ensure that the system provides appropriate feedback if a study type is not selected before proceeding to course planning.
Steps: 1. Log in to the system (if applicable). 2. Navigate to the welcome page. 3. Select a study type (e.g., Undergraduate, Graduate). 4. Confirm the selection. 5. Verify that the correct course options are displayed based on the selected study type. 6. Ensure the selection is saved, and no further changes are required before proceeding to course planning.
Time Constraint: Minimum: 5 min Maximum: 10 min

US2: As a user I want to log in to the system using the unimelb account.

- [ ] **TC2:** Verify that the user can log in using their Unimelb account.

Test Case for User Story 2
Test Type: Functional

Execution Type: Manual
Objective: Ensure the user can log in using their Unimelb account and save study plan progress, with the saved data remaining available upon the next login.
Setup: 1. User has a valid UniMelb account. 2. The system supports authentication via UniMelb credentials. 3. Study plans can be created and saved.
Pre-Conditions: 1. User is logged out or accessing the system for the first time. 2. User has started creating a study plan (optional).
Notes: Ensure the study plan progress is saved and remains valid upon logout and subsequent login.
Steps: 1. Log in to the system using valid UniMelb credentials. 2. Access the study plan section and begin creating a study plan. 3. Add or modify courses to the plan. 4. Save the progress of the current study plan. 5. Log out of the system. 6. Log back in using the same UniMelb credentials. 7. Verify that the previously saved study plan is correctly restored and can be edited further.
Time Constraint: Minimum: 10 min Maximum: 15 min

US3: As a user I want to be informed of any important information before proceeding on planning my course

Test Case for User Story 3
Test Type: Functional
Execution Type: Manual
Objective: Ensure the user is informed of any important information before planning their course.
Setup: 1. User is on the course planning page. 2. The system displays information before proceeding with course planning.
Pre-Conditions: 1. User has not started course planning. 2. Important notifications are available in the system.
Steps: 1. Nevigate to the course planner.

2. Review any notifications that are displayed.
3. Verify that the user is informed of all necessary details before proceeding.
4. Check that the notifications contain accurate and relevant information.

Time Constraint:
Minimum: 2 min
Maximum: 5 min

US4: As a user I want to filter and search for subjects

Test Case for User Story 4
Test Type: Functional
Execution Type: Manual
Objective: Verify that the user can filter and search for subjects in the course planner.
Setup: 1. User has access to the course planner. 2. User is on the course search/filter page.
Pre-Conditions: 1. The course planner is populated with subjects. 2. Search and filter options are available.
Steps: 1. Log in to the system. 2. Navigate to the course planner. 3. Filter subjects based on criteria (e.g., subject level, field of study). 4. Verify that the system returns correct results based on selected filters. 5. Check that users can clear and change filters easily.
Time Constraint: Minimum: 5 min Maximum: 8 min

US5: As a user I want to check if my selection of subjects break the course rules

Test Case for User Story 5
Test Type: Functional
Execution Type: Manual
Objective: Verify that the system checks for course rule violations when subjects are selected.
Setup: 1. User is on the course planner page. 2. The system has course rules implemented
Pre-Conditions:

1. User has selected subjects for planning.
2. Course rules for subject combinations are established.

Steps:

1. Log in to the system.
2. Select subjects for the study plan.
3. Submit the selection.
4. Verify that the system flags any rule violations.
5. Ensure that the system provides clear feedback on how to correct the issue.

Time Constraint:
Minimum: 5 min
Maximum: 10 min

US6: As a user I want to be able to add changes to my course plan when planning

Test Case for User Story 6
Test Type: Functional
Execution Type: Manual
Objective: Ensure that users can modify their study plan by adding or removing subjects.
Setup: 1. User is logged in with access to the course planner. 2. A study plan has been created.
Pre-Conditions: 1. Study plan must already exist. 2. The system allows for modifications to existing plans.
Steps: 1. Log in to the system. 2. Access the course planner and the existing study plan. 3. Add or remove subjects from the study plan. 4. Save the changes and verify they are reflected correctly. 5. Log out and log back in to check if the changes persist.
Time Constraint: Minimum: 5 min Maximum: 10 min

US7: As a user I want to go directly to enrolment after planning the course

Test Case for User Story 7
Test Type: Functional
Execution Type: Manual

Objective: Ensure that users can proceed directly to enrollment after planning their course.
Setup: User is logged in and has completed their study plan.
Pre-Conditions: 1. The study plan has been completed. 2. Enrollment options are available.
Steps: 1. Log in to the system. 2. Access the course planner and review the completed study plan. 3. Select the option to proceed to enrollment. 4. Verify that the system redirects the user to the enrollment page. 5. Ensure that all selected subjects are correctly transferred to the enrollment system.
Time Constraint: Minimum: 5 min Maximum: 12 min

US8: As a user I want to view prerequisites for each subject

Test Case for User Story 8
Test Type: Functional
Execution Type: Manual
Objective: Ensure that users can view prerequisites for each subject before selecting it.
Setup: 1. User is logged in and accessing the course planner. 2. The system has prerequisite information for subjects.
Pre-Conditions: 1. Subjects have predefined prerequisites stored in the system. 2. The user has selected a subject.
Steps: 1. Log in to the system. 2. Select a subject in the course planner. 3. Verify that the system displays prerequisites for the selected subject. 4. Ensure that prerequisites are clear and accurate.
Time Constraint: Minimum: 3 min Maximum: 7 min

US9: As a user I want to receive suggestions based on my previous choices

Test Case for User Story 9

Test Type: Functional
Execution Type: Manual
Objective: Verify that users can receive suggestions based on their previous choices.
Setup: 1. User has logged in and previously selected subjects. 2. The system can suggest additional courses.
Pre-Conditions: 1. The user has selected subjects. 2. The system can suggest relevant subjects based on previous selections.
Steps: 1. Log in to the system. 2. Navigate to the course suggestions section. 3. Verify that the system suggests subjects related to previous choices. 4. Ensure that suggestions are relevant and logical.
Time Constraint: Minimum: 3 min Maximum: 5 min

US10: As a user I want to export my course plan as a PDF

Test Case for User Story 10
Test Type: Functional
Execution Type: Manual
Objective: Ensure that users can export their course plan as a PDF.
Setup: 1. User is logged in and has completed their study plan. 2. The system supports PDF export.
Pre-Conditions: 1. The study plan is complete. 2. PDF export functionality is available.
Steps: 1. Log in to the system. 2. Access the course planner and complete the study plan. 3. Select the option to export as a PDF. 4. Verify that the system generates the PDF correctly. 5. Ensure that the PDF contains all the required course information.

Time Constraint:
Minimum: 2 min
Maximum: 5 min

US11: As a user I want to compare different course plans side by side

Test Case for User Story 11
Test Type: Functional
Execution Type: Manual
Objective: Verify that users can compare different course plans side by side.
Setup: 1. User is logged in with multiple course plans. 2. The system allows comparison of plans.
Pre-Conditions: 1. User has created more than one course plan. 2. The system can display these plans simultaneously.
Steps: 1. Log in to the system. 2. Access the course planner with multiple plans. 3. Select the comparison option. 4. Verify that the system displays the plans side by side. 5. Ensure that all relevant course details are clearly visible for comparison.
Time Constraint: Minimum: 3 min Maximum: 7 min

US12: As a user I want to manage the list of available courses

Test Case for User Story 12
Test Type: Functional
Execution Type: Manual
Objective: Ensure that users can manage the list of available courses and view other students' lesson plans and reviews.
Setup: 1. User is logged in and can access the course management system. 2. The system stores course plans and reviews from other students.
Pre-Conditions: 1. The course management system contains other students' plans and reviews. 2. The user has access to the relevant section.
Steps:

1. Log in to the system.
2. Access the course management section.
3. Search for available courses and view other students' plans and reviews.
4. Verify that all course and review information is displayed correctly.

Time Constraint:
Minimum: 4 min
Maximum: 8 min

US13: As a user I want to receive regular reminders and updates about course selections

Test Case for User Story 13
Test Type: Functional
Execution Type: Manual
Objective: Ensure that users receive regular reminders and updates about their course selections.
Setup: 1. User has logged in and selected courses. 2. The system can send notifications.
Pre-Conditions: 1. User has logged in and selected courses. 2. The system can send notifications.
Steps: 1. Log in to the system. 2. Verify that the user receives updates about course selection status. 3. Ensure that reminders and updates are timely and accurate.
Time Constraint: Minimum: 2 min Maximum: 5 min

US14: As a administrators I want to importing existing course planning templates

Test Case for User Story 14
Test Type: Functional
Execution Type: Manual
Objective: Verify that administrators can import existing course planning templates for student use.
Setup: 1. Administrator is logged in with appropriate privileges. 2. The system contains course templates.
Pre-Conditions:

1. Administrator is logged in with appropriate privileges. 2. The system contains course templates.
Steps: 1. Log in as an administrator. 2. Navigate to the course template section. 3. Select a template to import. 4. Verify that the system imports the template correctly and makes it available to students.
Time Constraint: Minimum: 3 min Maximum: 6 min

Non-functinal testing

Performance Testing

- **Scenario:** During peak registration periods, the course planner is accessed simultaneously by hundreds of students.
- **Test Case:**
  - Simulate 100, 500, and 1000 concurrent users logging into the system and attempting to access the course planning features.
  - Measure response times for the following actions:
    - Logging into the system
    - Selecting a course and viewing its details
    - Applying changes to the course plan and saving them
  - **Expected Outcome:** The system should respond within 3 seconds for each action, even with 1000 users accessing the system simultaneously.
  - **Monitoring Tools:** Use tools like Apache JMeter or LoadRunner to simulate the user load and gather performance metrics.
- **Follow-up Actions:** If the response time exceeds 3 seconds, identify performance bottlenecks (e.g., database query delays or server capacity issues) and take steps to optimize system performance.

Security Testing

- **Scenario:** Ensuring that only authorized users can access sensitive information in the course planner.
- **Test Case:**
  - Attempt multiple login attempts with incorrect credentials to check if the system locks the account after a set number of attempts (e.g., 5).
  - Perform SQL injection tests by entering malicious code into input fields like login and search boxes.
  - Test for XSS vulnerabilities by inputting scripts that could potentially execute on other users' browsers.
  - **Expected Outcome:** The system should:
    - Lock the user account after 5 failed login attempts and notify the user via email.
    - Detect and reject any SQL injection or XSS attempts, displaying a secure error message without exposing sensitive information.
  - **Monitoring Tools:** Use tools like OWASP ZAP or Burp Suite to automate and detect vulnerabilities.
- **Follow-up Actions:** If vulnerabilities are detected, update the codebase to sanitize inputs and implement security patches to prevent such issues.

Compatibility Testing

- **Scenario:** Ensuring the course planner works seamlessly across different devices and browsers.
- **Test Case:**
  - Access the course planner using:
    - Different browsers: Chrome (latest version), Firefox, Safari, and Microsoft Edge.
    - Different devices: Windows desktop, MacBook, iPhone, and Android tablet.
  - Verify the following:
    - The layout is consistent (no misalignment or missing elements).
    - All interactive elements (buttons, drop-downs, forms) function as intended.
    - No functionality is lost when using mobile or tablet versions (e.g., the planner must be responsive and adjust to screen size).
  - **Expected Outcome:** The course planner should display consistently and all functionalities should be accessible across all tested platforms and devices.
- **Monitoring Tools:** Use browser developer tools and cross-browser testing services like BrowserStack or CrossBrowserTesting.
- **Follow-up Actions:** If issues are found (e.g., buttons not functioning on Safari), adjust CSS styles or JavaScript code to ensure cross-browser compatibility.

Strategy for Addressing Potential Issues

Defect Tracking and Resolution Plan:

- **Scenario:** A defect is identified where the manual eligibility check feature fails to load correctly.
- **Action Steps:**
  - Log the defect in a tracking tool like JIRA with details such as:
    - Defect ID: DEF-101
    - Severity: Critical
    - Description: "Manual eligibility check feature not loading; users receive a blank screen upon clicking the button."
    - Assigned To: Backend Development Team
    - Deadline: 24 hours
  - **Testing After Fix:** Once fixed, perform a targeted test on the eligibility check feature to ensure it works as expected. Then conduct regression testing on related features like the course selection and save functions to verify that they remain unaffected.
  - **Expected Outcome:** The eligibility check should display correctly, and other features should function without any issues post-fix.
- **Follow-up Actions:** Document the cause of the defect and the solution implemented to prevent similar issues in the future.

Regression Testing After Fixes:

- **Scenario:** Fixes were made to resolve the long loading times when applying course changes.
- **Action Steps:**
  - After deploying the fix, execute a suite of tests covering all core functionalities, such as:
    - Logging in
    - Selecting and saving courses
    - Checking eligibility
    - Accessing course comparison tools
  - Use automated testing tools like Selenium to streamline the process and ensure comprehensive coverage.

    - **Expected Outcome:** All functionalities should work without errors, and loading times should be reduced to meet performance benchmarks.
- **Follow-up Actions:** If any new issues arise during regression testing, they are logged and resolved promptly.

Documentation and Continuous Improvement:

- **Scenario:** During testing, a recurring issue with loading times was identified.
- **Action Steps:**
    - Document the issue in a testing report with the following:
        - Description: "Loading times exceed acceptable limits when multiple users access the system simultaneously."
        - Steps Taken: "Optimized database queries, added caching mechanisms, and increased server capacity."
        - Tools Used: "LoadRunner and New Relic for performance monitoring."
    - Schedule monthly reviews to update the testing document and ensure it reflects any new issues and solutions. This will provide the team with a clear reference for future testing phases.
- **Expected Outcome:** Over time, fewer recurring issues should be identified, and the team should have a solid framework for addressing similar problems efficiently.