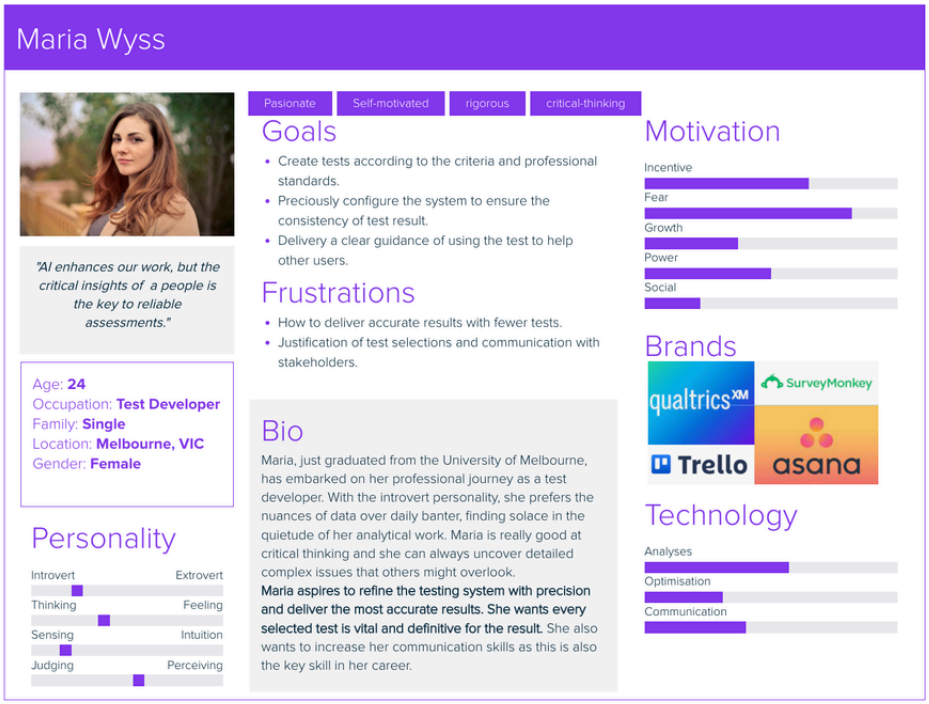
Personas

Three different personas of ***test developer*** has been created in this page. They all have different personalities, backgrounds and pursuits.



User stories:

1. As Maria, I want to **be able to create tests that meet professional standards** so that I can **do my job independently and provide reliable evaluations** to ensure that **each test is necessary and conclusive**.
2. As Maria, I want to **be able to accurately configure the system to ensure consistent test results** so that I can **complete and validate tests within an iteration cycle without exceeding the intended scope of work**.
3. As Maria, I want to **provide clear guidelines for using tests to help other users understand how to use the testing system effectively** so they can provide value.
4. As Maria, I want to **deliver accurate results by reducing the number of tests** so I can **maximise the amount of information I can get out of a limited number of tests while keeping them testable and manageable**.

Use Case: Streamlined Test Creation and Configuration

Title: Efficient Test Management

Actor: Test Developer (Maria)

Preconditions:

- Maria is logged into the test management system.
- Maria has received the necessary training on the system and test standards.
- The system is operational and has the latest updates installed.

Main Flow:

1. Maria selects the option to create a new test from the system dashboard.
2. The system presents a template that adheres to professional standards.

3. Maria inputs the test parameters and criteria, ensuring they align with professional standards.
4. The system validates the input against predefined standards and provides feedback.
5. Maria adjusts the test parameters based on the system's feedback until the test meets the standards.
6. Maria saves the test configuration, and the system schedules it for execution.
7. The system executes the test and collects results.
8. Maria reviews the results to ensure they are consistent and reliable.
9. Maria approves the final test configuration and documentation.

Alternate Flows:

- If the system identifies discrepancies in the test parameters, it prompts Maria for corrections.
- If Maria identifies inconsistencies in the test results, she initiates a reconfiguration process.

Postconditions:

- The system has a new test that meets professional standards.
- The test results are consistent and can be validated within an iteration cycle.
- Documentation is available for other users to understand the test setup and results.

Use case: Effective Communication of test Usage

Title: Clear Testing Guidelines Communication

Actor: Test Developer (Maria)

Preconditions:

- Maria has a set of tests that have been validated and are ready for use.
- Other users have access to the testing system and require guidance.

Main Flow:

1. Maria identifies the key points that users need to understand about the tests.
2. Maria creates a guideline document using a system-provided template.
3. The document includes instructions on test execution, interpretation of results, and troubleshooting steps.
4. Maria publishes the guidelines on the system's knowledge base.
5. Users access the guidelines and apply them to effectively use the testing system.
6. Maria receives feedback from users and updates the guidelines as necessary.

Alternate Flows:

- If users encounter issues not covered by the guidelines, Maria updates the document to include new information.

Postconditions:

- Users are informed about how to use the tests effectively.
- The testing system is used consistently by all users, providing value.

Use Case: Optimisation of Test Quantity

Title: Maximising Test Efficacy

Actor: Test Developer (Maria)

Preconditions:

- Maria has access to historical test data and performance metrics.
- The system is capable of handling complex test configurations.

Main Flow:

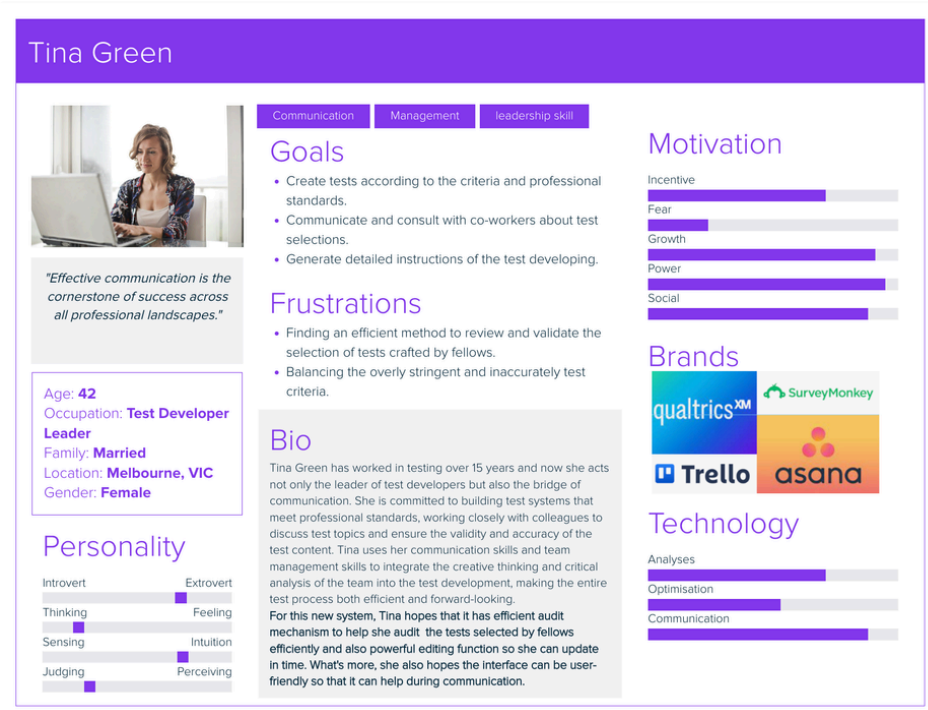
1. Maria analyzes historical test data to identify patterns and redundancies.
2. Maria selects a subset of tests that provide the most information.
3. The system simulates the reduced test set to predict outcomes.
4. Maria reviews the simulation results to ensure accuracy and coverage.
5. Maria finalises the optimised test set and updates the system configuration.
6. The system executes the optimised tests and collects results.
7. Maria evaluates the effectiveness of the reduced test set.

Alternate Flows:

- If the simulation results are unsatisfactory, Maria revises the test selection.

Postconditions:

- The system delivers accurate results with a reduced number of tests.
- The tests remain manageable and within the scope of an iteration cycle.



User Stories:

- As Tina, I want to create tests that adhere to professional standards and criteria so that I can ensure the quality and reliability of the test systems.
- As Tina, I want to communicate and consult with my co-workers about test selections so that we can collectively ensure the validity and accuracy of the tests.
- As Tina, I want to generate detailed instructions for test development so that my team can build tests that are both efficient and meet professional benchmarks.

Use Cases: Test Development Standardisation

Actor: Test Developer Leader (Tina)

Preconditions:

- Tina has access to the test development platform.
- The platform is pre-loaded with professional standards and criteria.

Main Flow:

1. Tina logs into the test development platform.
2. She selects the option to create a new test.
3. Tina inputs the required test criteria, aligning with professional standards.
4. The platform validates the criteria against the standards.
5. Tina reviews and finalizes the test setup.
6. The platform saves the test for execution and review by peers.

Alternate Flows:

- If the criteria do not meet the standards, Tina revises them until they align.
- If a peer suggests changes, Tina evaluates and incorporates the feedback as necessary.

Postconditions:

- A new test is created that meets the agreed professional standards.
- Peers have clear instructions for executing and reviewing the test.

User case: Collaborative Test Selection and Review

Actor: Test Developer Leader (Tina)

Preconditions:

- Tina's team has drafted a selection of tests.
- The test development platform has an efficient audit mechanism.

Main Flow:

1. Tina initiates the test review process on the platform.
2. She examines the tests selected by her team members.
3. Tina uses the audit mechanism to assess the tests' adherence to standards.
4. She provides feedback and suggestions for improvement.
5. The team revises the tests based on Tina's input.
6. Tina approves the final selection of tests.

Alternate Flows:

- If a test does not meet the standards, Tina can request a re-evaluation or discard it.
- If there are disagreements, Tina facilitates a discussion to reach a consensus.

Postconditions:

- The selected tests are validated and ready for deployment.
- The team has a clear understanding of the test selection rationale.

User cases: Enhancing Team Communication and Test Instruction Clarity

Actor: Test Developer Leader (Tina)

Preconditions:

- Tina needs to disseminate test development instructions to her team.
- The platform has a user-friendly interface for communication.

Main Flow:

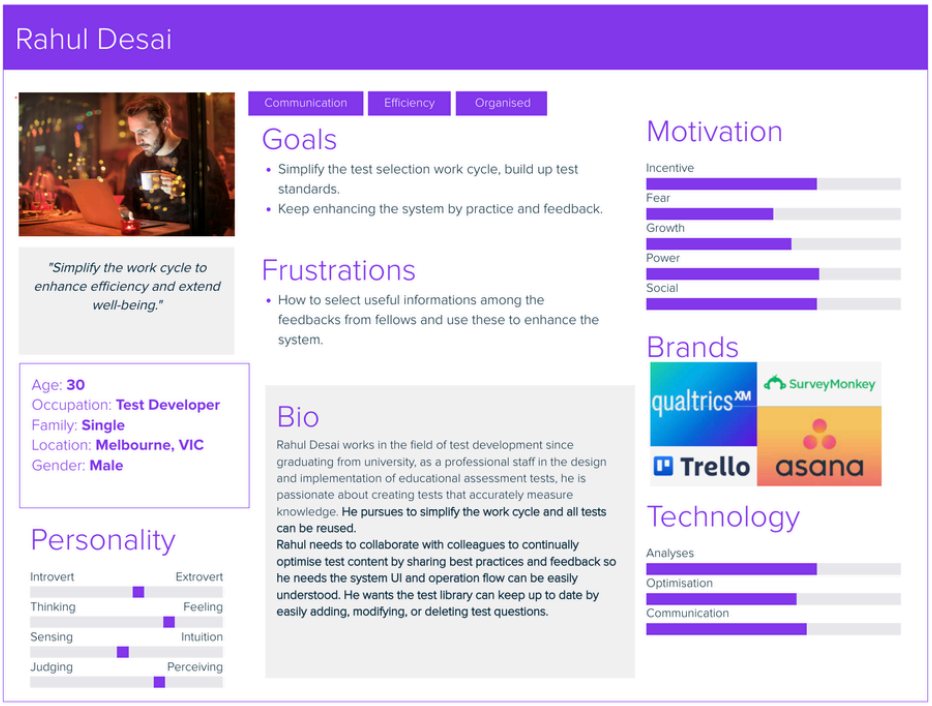
1. Tina drafts detailed test development instructions.
2. She uploads the instructions to the platform.
3. The team accesses the instructions and provides feedback.
4. Tina revises the instructions based on the team's input.
5. The finalized instructions are distributed to the team.
6. The team uses the instructions to develop tests.

Alternate Flows:

- If the instructions are unclear, Tina holds a meeting to clarify.
- If there are updates to the test criteria, Tina promptly revises the instructions.

Postconditions:

- The team has a clear set of instructions for test development.
- Communication between Tina and her team is streamlined and effective.



User Stories:

- As Rahul, I want to **streamline the test selection process** so that I can **build up test standards and simplify the work cycle**.
- As Rahul, I want to **incorporate feedback from colleagues into the system** so that we can continually **enhance the test content**.
- As Rahul, I want the **user interface and operation flow of the system to be intuitive so that collaboration and sharing of best practices are facilitated**.
- As Rahul, I want to **keep the test library up-to-date with the ability to easily add, modify, or delete test questions** so that **the tests remain relevant and effective**.

Use Cases: Streamlining Test Selection

Title: Simplified Test Selection Work Cycle

Actor: Test Developer (Rahul)

Preconditions:

- Rahul is logged into the test management system.
- The system is pre-loaded with a comprehensive library of test questions.

Main Flow:

1. Rahul accesses the test selection module within the system.
2. The system presents an organized view of test questions, categorized by subject and difficulty.
3. Rahul selects the desired test questions, guided by the system's recommendations based on established standards.
4. The system updates the test library with Rahul's selections and provides a preview.
5. Rahul reviews and confirms the selections, finalising the test setup.

Alternate Flows:

- If Rahul requires additional questions, he uses the system's search and filter functions to find suitable additions.
- If Rahul decides to modify a question, he accesses the editing tool within the system to make changes.

Postconditions:

- The test standards are updated and the work cycle is simplified.
- The test library reflects Rahul's current selections, ready for use.

Use Case: Feedback Incorporation

Title: Continuous Test Enhancement through Feedback

Actor: Test Developer (Rahul)

Preconditions:

- Rahul has received feedback from colleagues on test content.
- The system is capable of tracking changes and suggestions.

Main Flow:

1. Rahul opens the feedback module in the system.
2. The system displays recent feedback and suggestions from colleagues.
3. Rahul evaluates the feedback and decides on the appropriate actions.
4. The system implements Rahul's decisions, updating the test content accordingly.
5. Rahul verifies the updates and approves the final changes.

Alternate Flows:

- If Rahul disagrees with certain feedback, he initiates a discussion with colleagues through the system's communication tools.

Postconditions:

- The test content is optimised based on collaborative feedback.
- The system's test library is enhanced, promoting efficiency and well-being.