# Testing

Testing aims at assuring the functionalities of the product fit our client's expectation and the product being delivered is of high quality.

Refer to the following pages for details of our test procedure, and test results:

- Testing Approach
- Back-End Unit / Integration Testing
- Front-End Unit Testing
- Acceptance Testing
- User Testing
- End-to-End Testing

# Testing Approach

## Purpose

Software testing is an important part of the Verification phase of our Quality Assurance plan.

This document outlines our approach for testing, and justifies our choice of strategy.

## Testing approach

We will follow an agile testing approach.

This primarily means that testing is performed by all developers, as part of their development work - unlike under traditional software development lifecycles, where testing would be a separate phase after the code is written and might be performed by separate team members. This helps us ensure we are deploying high-quality product at the end of each sprint. It ensures that all developers are responsible for code that is high-quality and stands up to tests.

Most importantly, it allows us to identify bugs and errors continuously throughout development, instead of waiting until the later stages of the project which may uncover unexpected issues and cause delays.

### Guidelines

1. All Code changes **should contain** a test in order to prove the correctness if the new code added. This includes adding tests to demonstrate correctness of new functionality or to verify a bugfix is working as expected.
2. To achieve an appropriate level of confidence in our system we should write tests where:
   a. **Correct Input is used** (Should reasonably encase the set of valid inputs)
      i. If your program takes in an array, write a test for the inputs of 0, 1, many values in array
      ii. Use Boundary Value Analysis & Equivalence Partitioning concepts to help
   b. **Incorrect Input is used** (Because programs *should* fail expectedly/gracefully)
3. Please aim to **at least** achieve **statement coverage** (if possible), that is your tests suite should execute each line at least once.

## Test types

A number of different test types are appropriate for the project and are described below. Further information on how these will be carried out is provided under ....

### Unit tests

Unit testing aims to evaluate individual pieces of code to ensure they function as intended. For this project, unit testing will be performed on each function in the code. They will be automated using appropriate testing frameworks for the front- and back-end, and can be run as part of continuous integration.

### Integration tests

Integration testing aims to evaluate that the different components of the software correctly interface together. The components of this software system include the database, back-end and front-end. Similar to unit tests, integration tests will be be automated. We will employ a bottom-up approach - that is, two components will be tested at a time, before testing the integration of all the components together.

### End-to-end tests & user acceptance testing

**End-to-end testing** replicate a user's behaviour to use our project in a realistic scenarios We will investigate the use of testing frameworks that can automate these tests, otherwise these can be performed manually (sparingly).

As user stories are completed during a sprint, the corresponding **user acceptance tests** will be performed manually to check if the software meets the requirements.

### Exploratory testing

Exploratory testing is a less structured approach where an indivdual tester, rather than formally designing test cases, simply uses the system and attempts to discover bugs on-the-fly, taking into consideration common weaknesses and potential faults. This practice exercises the judgement of the developer and can help discover errors that are not uncovered by automated testing. This will be performed towards the end of each sprint.

## Testing methodology

Automated test cases, in particular, will mainly be documented in the repository source code.

Manual test cases will be documented in Confluence. Tests are to be documented fully, including for each test case a description of the reason for the test case and choice of test input, and the expected output. When performing the testing, sufficient information must be captured from the input and output must be captured, such that the test is easily reproducible with exactly the same input.

Any failures in tests are to be documented before making fixes to the code.

When testing, the developer/tester should fix any straightforward, simple bugs. Any complicated or unsolvable bugs identified during testing are to be communicated to the team and logged on Jira.

## Further reading

- https://en.wikipedia.org/wiki/Agile_testing
- https://www.guru99.com/agile-testing-a-beginner-s-guide.html
- https://www.atlassian.com/continuous-delivery/software-testing/types-of-software-testing
- https://www.atlassian.com/continuous-delivery/software-testing/exploratory-testing

# Back-End Unit / Integration Testing

## Back-End Test Plan

### Goal:

Testing of the features and overall performance on the Back-End side. The elements to test are server and database.

### Start time:

After completion of the feature implementation and making a pull request to the **develop** branch.

### Post-condition:

Report to the back-end lead about the test results. Approve a pull request to the develop branch if the test is successful.

### Testing framework:

**Jest** framework is used for the unit testing of the software. The information on how to use the framework can be found here.

| Class | API | Method | Pre-Route | Route |
|---|---|---|---|---|
| game-controller.js | newGame | post | /play | /new |
| | joinGame | post | | /join/:gameID |
| | getPlayerStatus | get | | /wait/:gameID |
| | getGameStatus | get | | /:gameID/status/ |
| | getBackground | get | | /background/:gameID |
| | getRolesInfo | get | | /:gameID/roles/ |
| | showAllRoles | get | | /choose-role/:gameID |
| | chooseRoles | post | | /choose-role/:gameID |
| | fetchDecision | get | | /make-decision/:gameID |
| | makeDecision | post | | /make-decision/:gameID |
| | readyToDiscuss | post | | /:gameID/start-discussion/ |
| | getRoundDescription | get | | /:gameID/round/ |
| | getRoundStatus | get | | /:gameID/round-status |
| | getRoundOutcome | get | | /:gameID/round-outcome |
| option-controller.js | getOption | get | /options | / |
| | newAllOptions | post | | / |
| | dropOption | post | | /drop |
| question-controller.js | getQuestion | get | /questions | / |
| | newQuestion | post | | / |
| | dropQuestion | post | | /drop |
| roles.js | getRole | get | /roles | / |

| | | | | |
|---|---|---|---|---|
| | newRole | post | | / |
| | getRolesIntro | get | | /introduction |
| | | | | |

## Back-End Test Results

| Test suite | Test case - functionality | Sprint 3 Results (Pass/Fail) | Sprint 4 Results (Pass/Fail) |
|---|---|---|---|
| game_model.test.js | create & save game successfully | Pass | Pass |
| | insert game successfully | Pass | Pass |
| play.test.js | connect to server | Pass | Pass |
| | insert options | Pass | Pass |
| | create new game | Pass | Pass |
| | choose available role | Pass | Pass |
| | choose unavailable role | Pass | Pass |
| | make decision | Pass | Pass |
| | get role information | Pass | Pass |
| | get outcome | Pass | Pass |
| role_model.test.js | create & save role successfully | Pass | Pass |
| | insert role successfully | Pass | Pass |
| | create & save user successfully | Pass | Pass |
| | insert user successfully | Pass | Pass |

# Front-End Unit Testing

## Front-end test plan

### Goal

Testing of the features and overall performance on the Front-End side. The elements to test our interaction with Front-End elements, visual bugs and inconsistencies.

### Start time:

After completion of the feature implementation and making a pull request to the **develop** branch.

### Post-condition:

Report to the front-end lead about the test results. Approve a pull request to the develop branch if the test is successful.

### Testing framework:

**Jest** framework is used for the unit testing of the software. The information on how to use Jest can be found here.

### Why choose Jest?

1. Jest can use its unique snapshot test function to automatically test common frameworks such as React by comparing snapshot files generated by UI code. In addition, Jest's test cases are executed in parallel, and only the tests corresponding to the changed files are executed, which improves the test speed.

2. Easy to install and configure, very easy to use, almost zero-configuration, can be run directly by npm command installation

3. Jest has built-in test coverage tool Istanbul, which can be opened by command or configured in more detail in the package.json file. Running Istanbul In addition to the terminal display test coverage, a coverage directory will be produced under the project, with a report of test coverage, so that we can clearly see the test of the branch code.

4. Integrated assertion library, no need to introduce third-party assertion library, and very support React component testing.

### Further resources

https://jestjs.io/

`https://www.youtube.com/results?search_query=jest`

## Front-end unit testing plan

The following test cases are planned:

1. rendering the correct title of each page
2. display the correct cartoon background of the main page
3. check the button of each page is correct and displays ok

4. check the HTML render the right div class
5. check the HTML render the content
6. check the HTML render the correct router
7. check the HTML render the correct layout
8. check the HTML render the correct colour

Unit tests that are nice to have:

1. mimic the clickable button events
2. check the format of the returned data
3. checking system status

## Example:

**Home Page**

1. When the "New Game" button is pressed does it go to the "NewGame" page
2. When the "Join Game" button is pressed does it go to the "JoinGame" page
3. When the "Help" button is press does a instructions popup appear
4. Does clicking "Got It" on the instructions popup cause it to disappear

**New Game Page**

1. When the back arrow is pressed does it go to the home page
2. Is the Submit button only active when the name text box is filled
3. Is the round duration slider only active when the timed game checkbox is selected
4. When the submit button is pressed does it go to the "Waiting" page

**Join Game Page**

1. When the back arrow is pressed does it go to the home page
2. Is the Submit button only active when both the gameID and name field is filled
3. When the submit button is pressed does it go to the "Waiting" page

# Sprint 2 Testing

## Test plan

Test the following components of the front-end:

Home.vue
JoinGame.vue
NewGame.vue
ChooseYourRole.vue

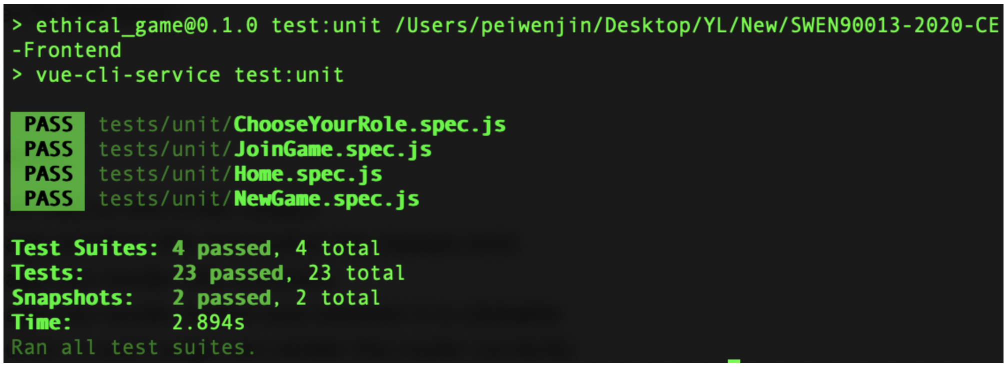
## Test results

A total of 23 tests were performed across 4 test suites:

Home.vue
JoinGame.vue
NewGame.vue
ChooseYourRole.vue

1.create snapshot files of the HTML output
2.check each page whether title and button list classes exist
3.check each page can render title correctly
4.check each page can render button and whether it is clickable
5.check each button in each page can render the router correctly
6.check if home page can render Help button content correctly

**All 23 tests** passed.



```
> ethical_game@0.1.0 test:unit /Users/peiwenjin/Desktop/YL/New/SWEN90013-2020-CE
-Frontend
> vue-cli-service test:unit

 PASS  tests/unit/ChooseYourRole.spec.js
 PASS  tests/unit/JoinGame.spec.js
 PASS  tests/unit/Home.spec.js
 PASS  tests/unit/NewGame.spec.js

Test Suites: 4 passed, 4 total
Tests:       23 passed, 23 total
Snapshots:   2 passed, 2 total
Time:        2.894s
Ran all test suites.
```

## Sprint 3 Testing

### Test plan

Background.vue

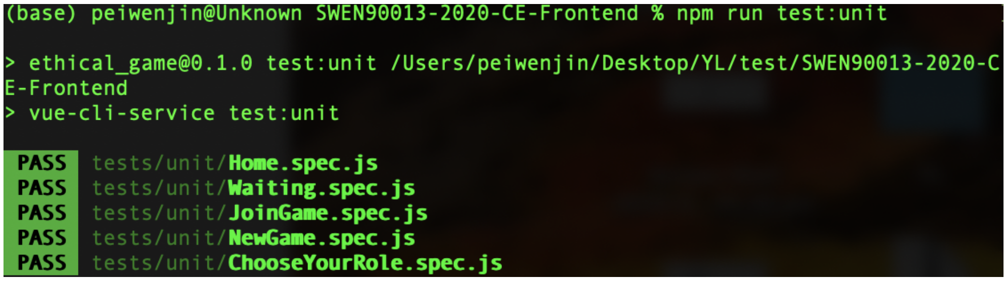
waiting.vue

ChooseYouRole.vue

ChapterBackground.vue

RoleBackground.vue

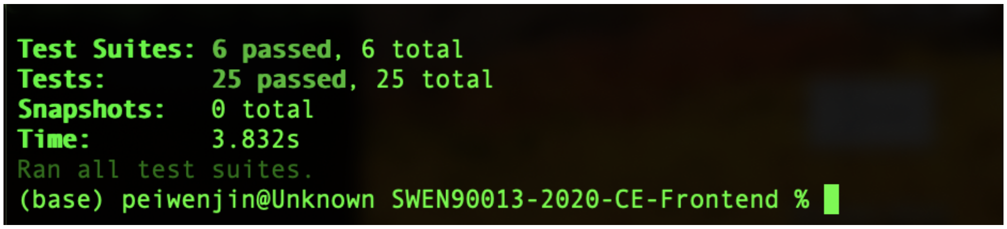
### Test results



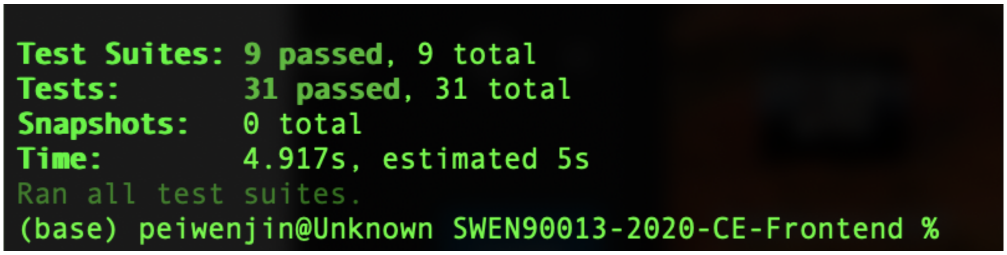
```
(base) peiwenjin@Unknown SWEN90013-2020-CE-Frontend % npm run test:unit

> ethical_game@0.1.0 test:unit /Users/peiwenjin/Desktop/YL/test/SWEN90013-2020-C
E-Frontend
> vue-cli-service test:unit

 PASS  tests/unit/Home.spec.js
 PASS  tests/unit/Waiting.spec.js
 PASS  tests/unit/JoinGame.spec.js
 PASS  tests/unit/NewGame.spec.js
 PASS  tests/unit/ChooseYourRole.spec.js
```



```
Test Suites: 6 passed, 6 total
Tests:       25 passed, 25 total
Snapshots:   0 total
Time:        3.832s
Ran all test suites.
(base) peiwenjin@Unknown SWEN90013-2020-CE-Frontend %
```



```
Test Suites: 9 passed, 9 total
Tests:       31 passed, 31 total
Snapshots:   0 total
Time:        4.917s, estimated 5s
Ran all test suites.
(base) peiwenjin@Unknown SWEN90013-2020-CE-Frontend %
```

Added test cases for the following 6 test suites for:
ChooseYourRole.spec.js
Background.spec.js
Waiting.spec.js
RoleBackground.spec.js
ChapterBackground.spec.js
IndividualBackground.spec.js
Mainly test: each page renders(class, title, content); each page router whether correctly; mock fake data to test whether backend can return correct data and format to frontend(ChooseYourRole.spec.js;Background.spec.js)

## Sprint 4 Testing

### Test plan

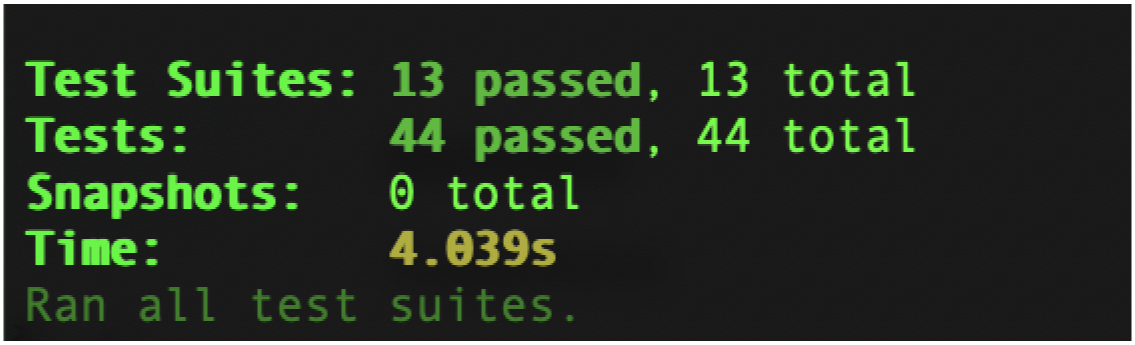
Outcome.vue

Reflection.vue

Ready.vue

Discussion.vue

### Test results



```
Test Suites: 13 passed, 13 total
Tests:       44 passed, 44 total
Snapshots:   0 total
Time:        4.039s
Ran all test suites.
```

Mainly test: each page renders(class, title, content); each page router whether correctly;

## Unit Test Results

### Key

| Sprint | Colour |
|---|---|
| 2 | 🟥 |
| 3 | 🟦 |
| 4 | 🟨 |

### Test results

38 test cases passed as across Sprints 2 and 3.

| ID | Test Goal | Test Suite Path | Test Result | Code Status |
|---|---|---|---|---|
| 1 | check whether title and button list classes exist | Home.spec.js | pass | uploaded |
| 2 | check if home page can render first title correctly | Home.spec.js | pass | uploaded |
| 3 | check if home page can render second title correctly | Home.spec.js | pass | uploaded |
| 4 | check if home page can render New Game button and it is clickable | Home.spec.js | pass | uploaded |
| 5 | check if home page can render Join Game button and it is clickable | Home.spec.js | pass | uploaded |
| 6 | check if home page can render Help button | Home.spec.js | pass | uploaded |

| | | | | |
|---|---|---|---|---|
| 7 | check if home page can render Help button content correctly | Home.spec.js | pass | uploaded |
| 8 | check if home page can render Help Got it button and check whether it is clickable | Home.spec.js | pass | uploaded |
| 9 | check whether title and form classes exists | JoinGame.spec.js | pass | uploaded |
| 10 | check if JoinGame page can render main title correctly | JoinGame.spec.js | pass | uploaded |
| 11 | check if JoinGame page can render submit button correctly | JoinGame.spec.js | pass | uploaded |
| 12 | check if JoinGame page can display a left arrow button and it is clickable | JoinGame.spec.js | pass | uploaded |
| 13 | check whether title and form classes exists | NewGame.spec.js | pass | uploaded |
| 14 | check if NewGame page can render main title correctly | NewGame.spec.js | pass | uploaded |
| 15 | check if NewGame page can render submit button correctly | NewGame.spec.js | pass | uploaded |
| 16 | check if NewGame page can display a left arrow button and it is clickable | NewGame.spec.js | pass | uploaded |
| 17 | check whether title and button list classes exists | ChooseYourRole.spec.js | pass | uploaded |
| 18 | check if ChooseYourRole page can render main title correctly | ChooseYourRole.spec.js | pass | uploaded |
| 19 | check if ChooseYourRole page can render Boeing Executive button correctly | ChooseYourRole.spec.js | pass | has been removed because of updating code |
| 20 | check if ChooseYourRole page can render Aeronautical Engineer button correctly | ChooseYourRole.spec.js | pass | has been removed because of updating code |
| 21 | check if ChooseYourRole page can render Software Developer button correctly | ChooseYourRole.spec.js | pass | has been removed because of updating code |
| 22 | check if ChooseYourRole page can render Pilot button correctly | ChooseYourRole.spec.js | pass | has been removed because of updating code |
| 23 | check if ChooseYourRole page can render FAA Official button correctly | ChooseYourRole.spec.js | pass | has been removed because of updating code |
| 24 | check whether ChooseYourRole page title render correctly and make sure its class exists | ChooseYourRole.spec.js | pass | uploaded |
| 25 | check ChooseYourRole page can get role object and check the backend role name and its description are correctly | ChooseYourRole.spec.js | pass | uploaded |
| 26 | check Next button going to the chapterBackground page | ChooseYourRole.spec.js | pass | uploaded |
| 27 | check whether Background page title render correctly and make sure its class exists | BackGround.spec.js | pass | uploaded |
| 28 | check Background page can get correct data from backend | BackGround.spec.js | pass | uploaded |
| 29 | check Next button going to the Choose Your role page | BackGround.spec.js | pass | uploaded |
| 30 | check whether ChapterBackground page title render correctly and make sure its class exists | ChapterBackground.spec.js | pass | uploaded |
| 31 | check Next button going to the IndividualBackground page | ChapterBackground.spec.js | pass | uploaded |
| 32 | check whether IndividualBackground page title render correctly and make sure its class exists | IndividualBackground.spec.js | pass | uploaded |
| 33 | check Next button going to the Discussion page | IndividualBackground.spec.js | pass | uploaded |
| 34 | check whether RoleBackground page title render correctly and make sure its class exists | RoleBackground.spec.js | pass | uploaded |
| 35 | check Next button going to the ChapterBackground page | RoleBackground.spec.js | pass | uploaded |
| 36 | check whether Waiting page title render correctly and make sure its class exists | Waiting.spec.js | pass | uploaded |
| 37 | check whether Waiting page can render correctly guide to end users | Waiting.spec.js | pass | uploaded |
| 38 | check Next button going to the background page | Waiting.spec.js | pass | uploaded |
| 39 | check if menu button gp to IndividualBackground page | Discussion.spec.js | pass | uploaded |
| 40 | check if discussion information can render correctly | Discussion.spec.js | pass | uploaded |
| 41 | check whether Discussion Content class show correctly | Discussion.spec.js | pass | uploaded |
| 42 | check whether title and class of Discussion page shows correctly | Discussion.spec.js | pass | uploaded |
| 43 | check whether title of drawer of the game menu content show correctly | Outcome.spec.js | pass | uploaded |
| 44 | check whether GameOutcome Text class show correctly | Outcome.spec.js | pass | uploaded |

| | | | | |
|---|---|---|---|---|
| 45 | check whether title of drawer of the game menu content show correctly2 | Outcome.spec.js | pass | uploaded |
| 46 | check if RoleOutcome Text class can show correctly | Outcome.spec.js | pass | uploaded |
| 47 | check if Outcome page router works well | Outcome.spec.js | pass | uploaded |
| 48 | check whether title and content exists | Ready.spec.js | pass | uploaded |
| 49 | check whether title and content exists | Ready.spec.js | pass | uploaded |
| 50 | check whether title and class of reflection page shows correctly | Reflection.spec.js | pass | uploaded |
| 51 | check whether All Decisions Table class show correctly | Reflection.spec.js | pass | uploaded |
| 52 | check if exit game button works well | Reflection.spec.js | pass | uploaded |

# Acceptance Testing

## Test summary

- **As at Sprint 3:**
  - 15/15 acceptance tests passed
  - 0 tests failed
  - 6 tests not applicable (functionality not yet implemented). Will be tested in a later sprint.
- **As at Sprint 4:**
  - 19/19 acceptance tests passed
  - 0 tests failed

## Detailed test table and results

| **Test Pass /Fail<br>(Sprint 3)** | **Test Pass /Fail<br>(Sprint 4)** | **User Story ID** | **User story** | **Acceptance Criteria ID** | **Given** | **When** | **Then** | **Acceptance Test ID** | **Test*** | **Expected Result** |
|---|---|---|---|---|---|---|---|---|---|---|
| ✓ | ✓ | US_01 | As a student, I want to have an option to view help for the game, so that I can get familiar with the features of the game, such as what actions I will take, how the UI works and how gameplay will work. | AC_01 | The student confirms that he/she has already connected to the game and can see the main page of the game. | He/She clicks "Help" button on the main page. | He/She can enter the help part which can help he/she to familiar with the game | AT_01_01 | 1. User Clicks "Help" button on the main page | Help Page content is displayed. |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| ✓ | ✓ | US_02 | As a student /teacher, I want to be able to join a particular game session of my choosing, so that I can join a game with any other users in it. | AC_02 | The student /teacher confirms that he/she has already joined the game by inputting their name and game ID(if they choose to join the game). | He/She clicks the next arrow button in "start a new game" or "join a game" page | He/She would enter the "waiting" page and wait for other online users to join with he /she and play together. | AT_02_01 | **Requires 2 devices**<br>- On Device 1<br>1. User Clicks "New Game"<br>2. User Enters a String with a length greater than 0 in "Your Name" Field and clicks next arrow<br>3. User retains Game ID for device 2<br>- On Device 2<br>1. User Clicks "Join Game"<br>2. User Enters Game ID (Given from Device 1) in Game ID Field<br>3. User enters a string with a length greater than 0 in "Your Name" Field<br>4. User clicks next arrow | Device 2 is placed on the waiting page with<br>1. User's name displayed<br>2. Game ID displayed |
| ✓ | ✓ | | | | | | | AT_02_02 | 1. User Clicks "Join Game"<br>2. User Enters **valid** Game ID<br>3. User enters **a blank** string in "Your Name" field | The Submit button is disabled |
| ✓ | ✓ | | | | | | | AT_02_03 | 1. User Clicks "Join Game"<br>2. User Enters **Invalid** Game ID<br>a. Empty String<br>3. The user enters a **valid** string (null or not null) in "Your Name" Field | The Submit button is disabled, and a message informs the user to enter a Game ID |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| ✓ | ✓ | | | | | | | AT_02_04 | 1. User Clicks "Join Game"<br>2. User Enters **Invalid** Game ID<br>a. Random String that is not linked to any game<br>3. The user enters a **valid** string (null or not null) in "Your Name" Field<br>4. User clicks Submit | A prompt is shown indicating that the game id specified is incorrect. |
| ✓ | ✓ | US_03 | As a student /teacher, I want to be able to initialise a game session, so that I can create a session for other users to join and play. | AC_03 | The student /teacher has started the application and can see the main page of the game. | He/She clicks the "New Game" button. | He/She is assigned an "ID" for other players to join. | AT_03_01 | 1. User clicks "New Game"<br>2. User does not enter a name | The Submit button is disabled |
| ✓ | ✓ | | | | | | | AT_03_02 | 1. User Clicks "New Game"<br>2. User Enters a **valid** string (length greater than 0) in "Your Name" Field<br>3. User Clicks Next Arrow | Waiting Page is shown with<br>1. User's name displayed<br>2. Game ID displayed |
| ✓ | ✓ | US_04 | As a student, I want to have an option to choose which character i play in the game, so that I can more easily play unique characters, or characters I'd prefer over others. | AC_04 | The student confirms that he/she has already started the game and see the game background and roles introduction. | He/She enters "choose your role "page. | He/She could choose the wanted character if it is his/her turn. | AT_04_01 | Requires multiple devices<br>- A device to create a game<br>- 4 other devices to fill all the roles<br>Assuming that a game is created<br>1. User clicks "Join Game"<br>2. User enters a **valid** string (length greater than 0) in "Your Name" Field<br>3. User enters a **valid** Game ID<br>4. User clicks next arrow<br>5. All the user slots are filled (enough players join a game)<br>6. User clicks "Start Game" button<br>7. The user jumps into the "Background" page and clicks next arrow<br>8. The user chooses a role in "choose your role page" that **has not** previously been selected | The Role that the user selected is highlighted, and all other role selection buttons are disabled |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| ✓ | ✓ | | | | | | | AT_04_02 | *Assuming the actions of AT_06 having completed* ***step 8***<br>1. The user chooses a role in "choose your role page" that **has** simultaneously been selected by another user | One user will be allocated the role, and the other won't be |
| ✓ | ✓ | US_05 | As a student , I want to have access to a detailed backstory for a character, as well as their character in the company before the game start, so that I can empathise with and adopt the perspective of this character more easily. | AC_05 | The student has already seen the "background" page | He/She clicks "next button" to see the role introduction page. | He/She could see a brief introduction to each role in the game. | AT_05_01 | *Assuming that the user is in a waiting room and the game is ready to start*<br>1. User joins the game or creates the game successfully<br>2. The user jumps into the "Background" page and clicks next arrow<br>3. The user jumps into the "Choose Your Role" page | The "Role" page is shown successfully with descriptions for each Role shown |
| ✓ | ✓ | US_06 | As a student, I want to be introduced to some game background information before game start, so that I can understand the general game. | AC_06 | The student has already started the game when other users are ready for the game | He/She clicks "next button" to see the "background" page. | He/She will see an informative context to introduce the background of the game. | AT_06_01 | *Assuming that the user is in a waiting room and the game is ready to start*<br>1. User joins the game or creates the game successfully<br>2. The user jumps into the "Background" page | The "background" page is shown successfully by following the order. |
| ✓ | ✓ | US_07 | As a student, I want to receive a set of goals outlining what objectives my character needs to achieve by game end, so that I have some guiding direction in terms of how I approach key decisions in-game. | AC_07 | The student has finished choosing his /her preferred role and other user finished choosing their role already | He/She clicks "start game" button on "choose your role" page. | He/She will see a clear set of goals for his/her own chosen character, with quantitative objectives in order to achieve each. | AT_07_01 | *Assuming that the user has been assigned a role and other users have finished the same process either.*<br>1. User clicks enter hits the Next button and see the special role aim description | A backstory and set of goals are shown. |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| ✓ | ✓ | US_08 | As a student, I want to receive prompts where necessary to facilitate discussion or personal decision-making, so that I am clearly aware at all times of exactly what is required of me at that time during the game. | AC_08 | The student has started the game/or(and) ready to enter to the beginning of each chapter | He/She clicks "next" button and enter the "chapter" page and clicks "next" arrow enter the "your information" page | He/She will sequence to see related prompts which help to facilitate discussion or personal decision-making. | AT_08_01 | *Assuming that the user has been assigned a role and clicked start game and also ready to enter to the beginning of each chapter.*<br>1. User clicks next arrow to enter into "chapter" page<br>2. Chapter Story is displayed, user clicks next arrow<br>3. "Your Information" page is displayed, user clicks next arrow | A prompt is displayed indicating that users need to discuss the decision as a group |
| ✓ | ✓ | US_09 | As a student, I want to be presented with an intuitive and clear ui for any decision-making, so that I can clearly understand all of the options that are available, as well as the current game situation. | AC_09 | For each round, the student knows the story plot and make sure he/she knows their personal player prompts. | He/She clicks "next" arrow and enter "group discussion" page | He/She enter the discussion page and see the question and optional decisions in a user-friendly UI. | AT_09_01 | *Assuming that the user has been assigned a role and clicked start game and also finished the prompts to support to make decisions.*<br>1. User clicks next arrow and enters to "Group Decision" page | A list of choices are displayed for the decision that needs to be made |
| ✓ | ✓ | US_10 | As a student, I want to be able to frequently discuss decisions with other characters in the game, so that I can communicate decisions with classmates, and negotiate for better outcomes related to my goals. | AC_10 | The student has started a game as any character and finish the prompts at the beginning of each round | He/She has enter "Group Discussion" page | He/She has a prompt for discussion | AT_10_01 | *Assuming that the user has been assigned a role and clicked start game.*<br>1. User clicks enter chapter button and the chapter description<br>2. User clicks next arrow see "your information"<br>3. User clicks next arrow and jumps into "Group Discussion" page | The user is presented with information and a decision, in order to prompt the group for discussion |
| | ✓ | US_12 | As a student, I want to be able to see all chosen decisions made for all of my personal and group decisions, so that I can get some insight into what other group members decided, and recall any forgotten decisions made earlier in the game. | AC_12 | Student ends the game and also finish the conclusion and outcome. | He/She clicks "reflection" button on "outcome" page | He/She will see a list of all decisions made throughout the game concerning that character. | AT_12_01 | *Assuming that the user has finished all rounds and decisions*<br>1. User clicks "end game" on the last of the game round page<br>2. User clicks next arrow on the conclusion page<br>3. User clicks next "reflection" on the outcome page | A list of all decisions made throughout the game concerning that character will be displayed on "reflection" page. |

|  |  |  |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|---|
| n/a | n/a | US_14 | As a student, I want to be prompted for confirmation whenever i make a decision, so that I can be sure that I did not make the wrong decision. | AC_14 | The student has already finished talking with other users and ready to answer the question in each round | He/She selects an option of a question | He/She is presented with a confirmation dialogue before their decision is submitted. | AT_14_01 | **N/A** | A message will be popped up to notice whether the user to continue his/her choice or cancel |
| n/a | n/a | US_15 | As a teacher, I want for the decision-making process to be relatively complex, in that there should be no clear-cut 'correct' decision, so that I can feel confident that students who play won't follow the same line of decision-making, and will enjoy unique experiences depending upon their own decision-making habits. | AC_15 | The student is in a decision-making phase of the game. | He/She makes a decision. | He/She possesses some level of uncertainty as to some consequences of that decision in the game context. |  | **N/A**<br>**Determining whether the choices in the game present ethical grey areas is of the personal opinion of the users playing the game** |  |
| n/a | n/a | US_16 | As a teacher, I want for the game to present asymmetric information and distribute asymmetric power, based upon the characters for each player, so that The game more closely mimics the reality of a working industry environment, which should improve student engagement and outcomes. | AC_16_01 | Students have been assigned characters and are at the start of a new round. | Students click "next" on the first context page. | Each student is presented with additional information which is representative of how relevant the decision being made is to the character. |  | **N/A**<br>**Determining whether the information that each user receives for their role is representative of how relevant the decision being made is to the character, is of the personal opinion of the users playing the game.** |  |
|  | ✓ |  |  | AC_16_02 | Students have been assigned characters and have made a decision for around. | All students have voted for that round. | The final group decision made for the round is weighted based on each student's character. | AT_16_02_01 | *Assuming that the user has been assigned a role and clicked start game.*<br>1. User clicks enter chapter button<br>2. Chapter Story is displayed, user clicks next arrow<br>3. "Your Information" page is displayed, user clicks next arrow<br>4. "Group Decision" page is displayed, user clicks "Enter Choice"<br>5. user selects choice and wait for other users finish their chooses<br>6. the game shows what decision the group has taken | The final group decision will also be displayed for each round |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | ✓ | US_17 | As a student, I want to receive concrete outcomes at the conclusion of the game related to my goals, as well as the plot, all based on decisions made throughout the game, so that I can clearly see whether or not I achieved my character goals or not, and also reflect upon whether personal success, if attained, came at any cost to plot outcomes. | AC_17 | The student has already ended the game and read the conclusion on the next page. | He/She clicks next arrow on the "conclusion" page | He/She is presented with the game-outcomes with additional details relevant to the student's character. | AT_17 | *Assuming that the user has finished all rounds and decisions*<br>1. User clicks "end of game" on the last of the game round page<br>2. User sees a conclusion for the game and an outcome for their character | The outcome will be displayed whether or not user achieved character goals or not, and it also reflect upon whether personal success, if attained, came at any cost to plot outcomes. |
| | ✓ | US_18 | As a student, I want to receive some feedback in the form of hints after the game is completed and results presented, so that I can get some takeaways about different ways to approach my decision-making for future playthroughs. | AC_18 | The student has reached the game-ending page and get the conclusion and outcome. | He/She clicks "reflection" button on "outcome" page. | He/She is presented with a reflection page with feedback. | AT_18 | *Assuming that the user has finished all rounds and decisions*<br>1. User clicks "end of game" on the last of the game round page<br>2. User clicks next arrow on the conclusion page<br>3. User sees the reflection page with all the choices made during the game by all players shown | Reflection information - a list of all decisions made - will be displayed in order to prompt discussion and reflection. |
| | n/a - this user story and functionality is no longer in scope | ~~US_19~~ | ~~As a teacher, I want to limit the discussion time for each decision in game, so that I can make students put more focus on the contradictive information and force them to make decisions, which will improve their critical thinking of whether their decisions are right to do.~~ | ~~AC_19~~ | ~~For each round, the student knows the story plot, make sure he /she knows their personal player prompts and is ready to be engaged in the discussion.~~ | ~~He/She clicks the "Ready to Discuss" button at the end of the personal character prompts page.~~ | ~~The timer can start counting down until the end of the specified time~~ | ~~AT_19~~ | ~~*Assuming that the user has been assigned a role and clicked start game.*~~<br>1. ~~User clicks enter chapter button and the chapter description~~<br>2. ~~User clicks next arrow see "your information"~~<br>3. ~~User clicks next arrow and jumps into "Group Discussion" page~~<br>4. ~~User clicks "voice" button to start talking if it is his/her turn~~<br>5. ~~User only can speak in a limited time~~ | ~~The timer will start counting down until the end of the specified time~~ |

# User Testing

## Background

The goal of user testing to test the deployed product with representative users. By observing the users complete tasks with our program, and also directly asking for feedback, can help provide insights as to the usability and design of our software, and help us ensure it meets its functional and non-functional goals.

## Usability testing - 28/10/20

User testing was performed on the Sprint 3 deployed release, with 4 students/graduates (all from outside the development team/subject), plus 1 member of the development team (Rainer), using a mix of mobile phones and laptops to use the web application. The users were asked to create/join a game and then play through the sample rounds.

The following insights/feedback were identified, and addressed as indicated below:

| Feedback | Response | Addressed in Sprint 4 |
|---|---|---|
| Decision page - the interface for the user to select their preferred option is unnecessarily complex. The different options are shown in a numbered list with their descriptions, but the buttons (labelled "Option 1", "Option 2" etc.) to enter their choice are shown in a group underneath, separately from the list. | This will be improved by combining this into a radio button interface - this will be easier to use as the input element is presented together with the description for the option. | ✓ |
| Application at times appears to hang when loading/processing certain actions (no indication to show that it is loading or processing) | This occurs when retrieving information from the back-end throughout the game (and is particularly noticeable, when the first request is made and the back-end needs to start up). This will be addressed by showing an animated 'loading' icon on all elements that need to obtain information from the back-end. | ✓ |
| The white boxes that show content sometimes look like input boxes | This is because some boxes are blank or show very short filler content. Better, more realistic game content will be loaded in the game during Sprint 4, and then it will be much clearer that it is not an input box.<br>Additionally, this is also partly due to the delay in retrieving the content from back-end, which makes boxes appear momentarily empty - this will be fixed by showing a loading icon until the content is ready. | ✓ |
| A horizontal line appears across the top of every page, for no apparent reason | This is a known issue relating to the styling of the nav bar, which will be fixed | ✓ |
| Cannot submit forms by pressing 'Enter' key - have to click on the Submit button | This will be fixed in Sprint 4. | ✓ |

Overall, the application functioned well during testing, and was intuitive enough for the users to play the game with minimal instruction. No significant issues were noted.

# End-to-End Testing

## Background

End-to-end tests aim to replicate user behaviour to perform tasks in the deployed software.

## Tests performed

For the final, sprint 4 release, we performed the following end-to-end tests. All tests were performed by running five instances of the game, to replicate five users playing the game on their own devices.

| Workflow | Test Passed |
|---|---|
| 1. User creates a game, and four other users join | ✓ |
| 2. Five users each choose a unique role | ✓ |
| 3. Read through some initial background information (general game context, plus role-specific info) at the start of the game | ✓ |
| 4. Read through relevant information for each round (general information, plus role-specfic info) | ✓ |
| 5. Be presented with a question to discuss with the group | ✓ |
| 6. Enter a preferred option for the decision | ✓ |
| 7. Receive the result of the round after receiving the option from all five users | ✓ |
| 8. View the in-game menu in-game, to see a list of other players | ✓ |
| 9. Exit the game from the in-game menu | ✓ |