Testing document S3 IPS

### Unit testing

* The initial first step in the testing process.
* Used to test the internal code of the application.
* Should test both positive behaviour and failure of an component.
* Should be independent of one another

Advantages:

* Make testing of code simpler
* Time and cost efficient
* Low maintenance
* Can be performed anytime
* Quickly identify faulty code changes

Disadvantage:

* Does not expose integration errors

### Integration testing

* Done after unit testing
* Used to test the interaction between components
* Requires external sources (database, hardware etc)

Advantages:

* Visibility of the integration structure
* Uncover bugs when different modules interact that are not visible with unit testing (mocking)

Disadvantages:

* Can be slow to run
* Complex
* Expensive to maintain

### Integration tests

#### Login

* The user should be navigated to the home page after having provided valid login information and clicking the Login button
* The user should see a warning message if incorrect login information is provided

#### Register

* User should be able to register a new account after the correct information is provided and clicking the register button.
* The user should see a warning if the provided username is already contained within the database.
* The user should see a warning if the provided email address is already contained within the database.

#### User manager

* User sees an access denied page when trying to access this page from a non admin account
* User sees an overview of all the user accounts that are stored

#### Products

* The user can navigate to the page of a specific product via it’s ID and see the details belonging to this product.
* The user is shown a 404 page not found error when trying to navigate to a non-existing id

#### Categories

* The user can navigate to the page of a specific Category and see the details and products belonging to this genre
* The user is shown a 404 page not found error when trying to navigate to a non-existing id

### Unit testing front end

#### Login

* Email field input should be valid
* Password field input should be valid
* Login button is enabled when all fields have received input

#### Register

* Email, Username, Password and Password confirmation field input should be valid
* Password confirmation field input should match password field input exactly
* Register button is enabled when all fields have received input

#### User manager

* An access denied image is rendered when not providing an logged in user
* An access denied image is rendered when providing an account with non admin rights
* The table containing all the provided users is rendered
* The provided data matches the data in the table

#### Products

#### Overview

* No data provided, A “no products found” messages should be shown
* The provided list of games gets rendered. The title, image, url link are correctly rendered

#### Detail

* Not providing a product to the state will render a page not found component
* Provide a product to the state. The correct information should be rendered in the appropriate divs

#### Categories

#### Overview

* No data provided, A “no category found” messages should be shown
* The provided list of genres gets rendered. The names and url links are rendered

#### Detail

* Not providing a category to the state will render a page not found component
* Provide a category with no products to the state. Check name of genre is shown and “There are no products for this category” message should be rendered.
* Provide a category with products to the state. Check name of category is shown and the names/image/url links to the products are rendered.

### Unit testing back end