**Software Development Lifecycle**

**Authors**

**Components**

* **AboutSectionHome** (.scss, .tsx)– Returns the front end of the About Section in the home page.  **(Author: Keerthana)**
* **CardGrid** (.scss, .tsx)– Generate a grid of CardMini in the Search/Browse section **(Author: Jun Sheng)**
* **CardGridEmpty** (.scss, .tsx)– Returns a “No Result Message” when user queries for an anime that does not match any of the animes in the api **(Author: Hill Seah)**
* **CardGridLoading** (.scss, .tsx)– Displays the “Loading” Screen with rotating logo when user queries for an anime **(Author: Hill Seah)**
* **CardGridOffline** (.scss, .tsx)– Returns a “Offline - 404” message when user tries to query for an anime but user is not connected to the web **(Author: Hill Seah)**
* **CardMini** (.scss, .tsx) – Pulls information about an anime from the api into a “mini card” or poster form for users to click on **(Author: Jun Sheng)**
* **FeaturedSectionHome** (.scss, .tsx) – Returns the front end of the Featured Anime section in the home page consisting of “Action”, “Drama”, and “Psychological” rows with fixed CardMini of animes. **(Author: Jun Sheng)**
* **Footer** (.scss, .tsx) - navigates to the “home”, “Algorithm” and, “Survey” page and displays “WeebDepo ©{year} by EntertainmentAccomplished” at the bottom of the page. **(Author: Keerthana)**
* **HeaderHome** (.scss, .tsx)  - Returns the HomePage with the featured anime section. **(Author: Cooper)**
* **HeaderSearch** (.scss, .tsx) - Returns the Search page with fixed Animes and a Filter option that helps users to restrict their search according to preferred genres etc. **(Author: Cooper)**
* **Navbar** (.scss, .tsx) - helps navigate to the “home” , “Algorithm” and, “Survey” page **(Author: Cooper)**
* **PageLoading** (.scss, .tsx) - Displays “Loading” with a rotating logo as the user goes to a specific page. **(Author: Keerthana)**
* **PaginationBar** (.scss, .tsx) - navigates between the different pages with paging buttons ( single arrow and double arrow) **(Author: Bhat Sachin)**
* **RecommendationTab** (.scss, .tsx) - Returns the recommended page showing Anime related to a particular anime **(Author: Bhat Sachin)**
* **SurveyQuestion** (.scss, .tsx) - Returns the survey questions and recommends an anime **(Author: Bhat Sachin)**

**Application**

* **App** ( .tsx) **(Author: Keerthana)**
* **customTypes** ( .ts) **(Author: Keerthana)**
* **process.config** ( .ts) **(Author: Keerthana)**

**Pages**

* **Entry** (.scss, .tsx) **(Author: Bhat Sachin)**
* **Home** (.scss, .tsx) **(Author: Bhat Sachin)**
* **Search** (.scss, .tsx) **(Author: Bhat Sachin)**
* **Survey** (.scss, .tsx) **(Author: Bhat Sachin)**
* **Login** (.py, .html, .css) **(Author: Cooper)**
* **SignUp** (.py, .html, .css) **(Author: Cooper)**
* **Authenticated** (.py, .html, .css) **(Author:** **Cooper)**

**Styles**

* **\_mediaqueries** (.scss) **(Author: Bhat Sachin)**
* **\_mixins** (.scss) **(Author: Bhat Sachin)**
* **\_variables** (.scss) **(Author: Bhat Sachin)**
* **App** (.scss) **(Author: Bhat Sachin)**

**Hooks**

* **useOnScreen** (.js) **(Author: Hill Seah)**
* **\_variables** (.js) **(Author: Hill Seah)**

**Others**

* **index**(.js) **(Author: Hill Seah)**
* **react-app-env**(.js) **(Author: Hill Seah)**
* **Service-worker** (.ts) (**Author: JunSheng)**
* **serviceWorkerRegistration** (.ts) (**Author: JunSheng)**

Diagram

Description automatically generated

Re-visit the documentation from previous lab sessions to ensure that it is complete and sufficient for delivery to the client. In particular, document the traceability from requirements to design, then to implementation and test cases.

The author(s) for each work product must be clearly indicated.

**Requirements Specification**

*Functional Requirements*

1. **System Features**

*This section is organised by Use Cases (Functional Requirements)*

1. **Login**
   1. **Description and Priority**

The user must already have a registered account with WeebDepo. Upon entering the website, the user will be redirected to the login page. The user must enter their email and password. If the entered inputs are valid, the user will be sent an authentication email. Else, an error message will appear depending on the error type. The user must click the link provided in the email to be successfully logged in. After the user clicks the link, the user will be redirected to the Homepage.

Priority: High

1. **Stimulus/Response Sequences**
2. User launches WeebDepo website
3. User brought to login page
4. **Functional Requirements**

REQ-1: Webpage must provide an error message when the user has inputted incorrect details

REQ-2: Webpage must send verification email when the user has inputted correct details

REQ-3: Webpage must redirect the user to homepage when the user clicks verification link

1. **Sign Up**
   1. **Description and Priority**

First-time users will click the ‘Sign Up’ button. The website will redirect the user to the Sign Up page. The user is required to enter their email and password twice to confirm their registration. If the entered inputs are valid, the user will be sent an authentication email. Else, an error message will appear depending on the error type. The user must click the link provided in the email to be successfully create the account. After the user clicks the link, the account will be created and the user will be redirected to the Homepage.

Priority: High

1. **Stimulus/Response Sequences**
2. User clicks the ‘Sign Up’ button on the login page
3. User brought to Register page
4. **Functional Requirements**

REQ-1: Webpage must provide an error message when the user has inputted incorrect details

REQ-2: Webpage  must send verification email when the user has inputted correct details

REQ-3: Webpage must redirect the user to homepage when the user clicks verification link

1. **Search**
   1. **Description and Priority**

The user can search for an anime by inputting a search string into the search bar and clicking the filters.

Priority: Medium

1. **Stimulus/Response Sequences**
2. User clicks the search bar
3. User types in search string to search for animes
4. If search string is invalid, webpage will notify the user “No search results found.” under the search bar
5. Else, a list of animes will appear under the search bar
6. User can choose to add filters to the search to reduce the number of search results
7. **Functional Requirements**

REQ-1: Webpage must allow users to search for animes

REQ-1.1: Webpage must allow users to search for animes by name

REQ-1.1.1: Webpage must limit 60 characters for search string

REQ-1.1.2: Webpage must require user to type 3 characters and above in search string

REQ-1.2: Webpage must allow users to search for animes by filters

REQ-1.3: Webpage must display error message when no search results are found

REQ-1.4: Webpage must display anime list below the search bar

1. **Recommend**
   1. **Description and Priority**

The user can get animes recommended to them by answering survey questions.

Priority: High

1. **Stimulus/Response Sequences**
2. User clicks the ‘Algorithm’ button on the homepage
3. User selects their answer to the questions
4. User brought to description page of an anime
5. **Functional Requirements**

REQ-1: App must display questions and answers

REQ-1.1: App must display the next question if the user has selected an answer

REQ-2: App must generate an anime based on the answers to the questions

REQ-3: App must redirect the user to an anime after user has answered all questions

1. **Featured**
   1. **Description and Priority**

The user can view featured animes on the homepage

Priority: Medium

1. **Stimulus/Response Sequences**
2. User scrolls down to ‘Featured’
3. User selects an anime on the homepage
4. **Functional Requirements**

REQ-1: App must display list of animes under the ‘Featured’ header

REQ-2: App must redirect the user to the anime description page when the user clicks one of the animes displayed

*Non-Functional Requirements*

**1. Usability**

1.1 Help messages.

1.1.1 The help message on the homepage provides an introduction to the web app's functionalities

1.1.1.1 The first clickable word “search engine” shall direct a user to the search engine.

1.1.1.2 The second clickable word “algorithm” shall direct a user to the algorithm page

1.1.1.2.1 The algorithm page shall have a list of questions for the user to aid in the selection of a recommended anime based on a user’s answered interests

1.1.2 The help messages on navigating through each anime, shall be displayed when the user clicks each anime.

1.1.2.1 The anime shall contain prequel and sequel buttons that aid in then search for prequels and sequels

1.2 Free settings.

1.2.1 The user shall use this application with a login account

1.3.1.1 Recommendation shall be on and displayed under featured anime when the user has logged in

1.2.2 The user shall fill up personal information or leave blank after signing in.

1.2.2.1 The user shall fill up avatar or leave blank.

1.2.2.2 The user shall fill up nickname or leave blank.

1.2.2.3 The user shall fill up personal preference or leave blank.

1.2.2.4 The user shall choose gender or leave unchosen.

1.2.3 The application shall allow the user to delete the account.

1.2.3.1 The account information associated with the account shall be deleted in the database.

1.2.3.2 The frequently searched animes and genres associated with the account shall be deleted from the database.

1.2.3.3 The frequently searched genres associated with the account shall be deleted from the database.

1.3 This application is free of change.

**2.**    **Reliability**

2.1APIs shall be implemented.

2.1.1 Anilist GraphQL API shall be used to retrieve anime details: synopsis, genres, trailers, format, scores, episodes, released date, and animation studio

2.2 The system shall handle exceptions.

2.2.1 Reminding message shall display when the user signs up with an invalid email address.

2.2.2 Reminding message shall display when the user signs up with an email address that has been registered.

2.2.3 Reminding message shall display when the user inputs the wrong verification code.

2.2.4 Reminding message shall display when the user signs in with an incorrect password.

2.2.5 Reminding message shall be displayed when the user provides nickname with blank or over 50 characters.

2.2.6 Reminding message shall be displayed when the user searches for an anime that does not exist.

2.2.7 Reminding message shall be displayed when the user clicks on pages that have no anime in them

2.3 The application must not leak user information.

              2.3.1 User’s email address shall not be seen or used by others

2.3.2 User information shall be deleted if the user account is deleted.

2.4   Restart shall be acceptable when failure happens

**3.**       **Supportability**

3.1 The application shall be extended in the future.

3.1.1 New user interface design and format shall be adopted.

3.1.2 New recommendation algorithm shall be adopted.

3.1.3 New features shall be adopted.

3.1.3.1 Users shall add animes to their watchlist.

3.1.3.2 Recommended animes will be generated based on the users’ watchlist.

3.1.3.3 Users shall write reviews on animes they have watched.

3.1.3.4 Reviews on each anime shall be displayed under the anime’s information.

  3.1.4 New API shall be added.

3.1.4.1 Translating API shall be implemented to translate into multiple languages.

3.1.4.2 Links API shall be implemented to fetch website links to watch a specific anime under the anime’s information.

3.2   Entertainment Accomplished team shall maintain the system.

3.3  The application shall get a mobile version, both IOS and Android

**Design**

Use case diagram, and other diagrams

**Use Case Model ( Use Case Diagram +  Use Case Description)**Diagram

Description automatically generated

**Use Case Description 1**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | **Case\_1\_GTH** | | |
| **Use Case Name:** | **Go to homepage** | | |
| **Created By:** | **Seah Jun Sheng** | **Last Updated By:** | **Seah Jun Sheng** |
| **Date Created:** | **6/9/22** | **Date Last Updated:** | **26/10/22** |

|  |  |
| --- | --- |
| **Actor:** | **User (Participating)** |
| **Description:** | **Brings user to the homepage from whichever page the user is on** |
| **Preconditions:** |  |
| **Postconditions:** | **Homepage is displayed** |
| **Priority:** | **2** |
| **Frequency of Use:** | **Whenever user wants to go back to homepage** |
| **Flow of Events:** | 1. **User selects the “Home” option** 2. **System displays the homepage** |
| **Alternative Flows:** | **NIL** |
| **Exceptions:** | **NIL** |
| **Includes:** | **NIL** |
| **Special Requirements:** | **NIL** |
| **Assumptions:** | **NIL** |

**Use Case Description 2**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | **Case\_2\_GTRP1** | | |
| **Use Case Name:** | **Get a Recommendation** | | |
| **Created By:** | **Seah Jun Sheng** | **Last Updated By:** | **Seah Jun Sheng** |
| **Date Created:** | **6/9/22** | **Date Last Updated:** | **26/10/22** |

|  |  |
| --- | --- |
| **Actor:** | **User (Participating)** |
| **Description:** | **Recommends user animes depending on which anime the user is currently interested in** |
| **Preconditions:** | **-** |
| **Postconditions:** | **“Recommended” subpage is displayed** |
| **Priority:** | **2** |
| **Frequency of Use:** | **Whenever the user wants to go to the “recommended” subpage** |
| **Flow of Events:** | 1. **User clicks “browse” or “search” icon** 2. **System will show the user a list of animes (search page)** 3. **User clicks on an anime card he is interested in** 4. **System will show “more info” and “related” depending on the anime poster clicked** 5. **User clicks on “related”** 6. **“Recommended” subpage is displayed** |
| **Alternative Flows:** | **Nil** |
| **Exceptions:** | **Nil** |
| **Includes:** | **NIL** |
| **Special Requirements:** | **Nil** |
| **Assumptions:** | **Nil** |

**Use Case Description 3**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | **Case\_3\_GTTP** | | |
| **Use Case Name:** | **Go to “Trending” Page** | | |
| **Created By:** | **Loke Kah Hou** | **Last Updated By:** | **Seah Jun Sheng** |
| **Date Created:** | **6/9/2022** | **Date Last Updated:** | **26/10/2022** |

|  |  |
| --- | --- |
| **Actor:** | **User (Participating)** |
| **Description:** | **Brings User to the “Trending” page** |
| **Preconditions:** | **-** |
| **Postconditions:** | **“Trending” page is displayed by the System** |
| **Priority:** | **2** |
| **Frequency of Use:** | **When the User decides to go to the “Trending” page** |
| **Flow of Events:** | 1. **User clicks “browse” or “search” icon** 2. **System will display the search page** 3. **User clicks on filter icon** 4. **User clicks on “Trending”** 5. **System displays “Trending” page consisting of all the anime cards of those which are trending** |
| **Alternative Flows:** | **NIL** |
| **Exceptions:** | **NIL** |
| **Includes:** | **NIL** |
| **Special Requirements:** | **NIL** |
| **Assumptions:** | **NIL** |

**Use Case Description 4**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | **Case\_4\_SM** | | |
| **Use Case Name:** | **Surprise Me** | | |
| **Created By:** | **Bhat Sachin** | **Last Updated By:** | **Seah Jun Sheng** |
| **Date Created:** | **8/9/2022** | **Date Last Updated:** | **26/10/2022** |

|  |  |
| --- | --- |
| **Actor:** | **User (participating, receiving)** |
| **Description:** | **Surprises the user with a random anime title.** |
| **Preconditions:** | **NIL** |
| **Postconditions:** | **The user is shown a random anime title.** |
| **Priority:** | **3** |
| **Frequency of Use:** | **When the user wants a surprise title.** |
| **Flow of Events:** | 1. **The user is on the homepage currently.** 2. **The user clicks on “Surprise Me!”** 3. **System will ask the user questions about his anime preference and other history** 4. **System will direct user to a description page of a random anime depending on the answers given to (3)** |
| **Alternative Flows:** | **NIL** |
| **Exceptions:** | **NIL** |
| **Includes:** | **NIL** |
| **Special Requirements:** | **NIL** |
| **Assumptions:** | **NIL** |

**Use Case Description 5**

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case ID:** | **Case\_5\_GTDP** | | |
| **Use Case Name:** | **Go to Description Page** | | |
| **Created By:** | **Bhat Sachin** | **Last Updated By:** | **Seah Jun Sheng** |
| **Date Created:** | **8/9/2022** | **Date Last Updated:** | **26/10/2022** |

|  |  |
| --- | --- |
| **Actor:** | **User (participating, receiving)** |
| **Description:** | **Go to the description page of an anime.** |
| **Preconditions:** | **nil** |
| **Postconditions:** | **The user is directed to the description page of an anime.** |
| **Priority:** | **3** |
| **Frequency of Use:** | **Whenever the user wants to find out more about an anime** |
| **Flow of Events:** | 1. **In the homepage, user can click on an anime poster he is interested in** 2. **System will display the description page of the anime the user clicked on** |
| **Alternative Flows:** | 1. **In the search page, trending page, or recommended page, user can click on an anime card he is interested in** 2. **System will show “more info” and “related” depending on the anime poster clicked** 3. **User clicks on “more info”** 4. **System will display the description page of the anime the user clicked on** |
| **Exceptions:** | **NIL** |
| **Includes:** | **NIL** |
| **Special Requirements:** | **NIL** |
| **Assumptions:** | **NIL** |

**Class Diagram**

Diagram

Description automatically generated

**Sequence Diagrams**

*Access Featured Anime*

Chart, diagram

Description automatically generated

*Get Recommended*

Diagram

Description automatically generated

*Search Anime*

Diagram

Description automatically generated

*Surprise Me*

Chart, box and whisker chart

Description automatically generated

*Choosing Filters*

Diagram, box and whisker chart

Description automatically generated

*Click Anime Search Result*

**State machine diagram**Diagram

Description automatically generated

Diagram

Description automatically generated

**Layered Architecture**

Timeline

Description automatically generated

**UI Mockups**

**Sign up/Login pageGraphical user interface

Description automatically generated**

**Graphical user interface

Description automatically generated**

**Homepage**

A picture containing text, different, various, bunch

Description automatically generated

**Trending page**Graphical user interface, website

Description automatically generated

**Code**

* + App skeleton file attached in the same folder “Lab 4”

**Testing**

* + Black and white box testing files are attached in the same folder “Lab 4”