# **Handover Fragment**

Group: I3 Client: Dr. Hannes Hermann

Written by: Jia Son Pow (Jay)

# **Functional Requirements**

| Functional Requirements | Student(s) | Non-Functional | Occasionally Functional | Somewhat Functional | Mostly Functional | Fully Functional |
| --- | --- | --- | --- | --- | --- | --- |
| 1.1 | Isaac |  |  |  |  |  |
| 1.2 | Isaac |  |  |  |  |  |
| 1.3 | Filipe, Jay |  |  |  |  |  |
| 1.4 | Isaac |  |  |  |  |  |
| 2.1 | Filipe |  |  |  |  |  |
| 2.2 | Isaac |  |  |  |  |  |
| 2.3 | Filipe, Jay |  |  |  |  |  |
| 3.1 | Isaac |  |  |  |  |  |
| 3.2 | Jay |  |  |  |  |  |
| 3.3 | Filipe, Jay |  |  |  |  |  |
| 4.1 | Filipe |  |  |  |  |  |
| 4.2 | Filipe |  |  |  |  |  |
| 5.1 | Filipe |  |  |  |  |  |
| 5.2 | Filipe |  |  |  |  |  |
| 5.3 | Sanjay |  |  |  |  |  |
| 5.4 | Filipe |  |  |  |  |  |
| 5.5 | Sanjay |  |  |  |  |  |
| 5.6 | Filipe |  |  |  |  |  |
| 5.7 | Isaac |  |  |  |  |  |
| Functional Requirements | Student(s) | Non-Functional | Occasionally Functional | Somewhat Functional | Mostly Functional | Fully Functional |
| 6.1 | Isaac |  |  |  |  |  |
| 7.1 | Filipe |  |  |  |  |  |
| 8.1 | Sanjay |  |  |  |  |  |
| 9.1 | Sanjay |  |  |  |  |  |
| 9.2 | Sanjay |  |  |  |  |  |
| 10.1 | Sanjay |  |  |  |  |  |
| 10.2 | Sanjay |  |  |  |  |  |
| 11.1 | Isaac, Filipe |  |  |  |  |  |
| 11.2 | Isaac |  |  |  |  |  |
| 13.1 | Sanjay |  |  |  |  |  |
| 13.2 | Sanjay |  |  |  |  |  |
| 13.3 | Sanjay |  |  |  |  |  |
| 14.1 | All |  |  |  |  |  |
| 14.2 | All |  |  |  |  |  |
| 15.1 | Sanjay |  |  |  |  |  |
| 15.2 | Sanjay |  |  |  |  |  |
| 16.1 | Isaac |  |  |  |  |  |
| 16.2 | Isaac |  |  |  |  |  |
| 16.3 | Isaac |  |  |  |  |  |
| 17.1 | Isaac |  |  |  |  |  |
| 17.2 | Isaac |  |  |  |  |  |

# 

# **FR List**

**Signup**

1.1 - The backend system must be able to send emails

1.2 - The backend system must be able to ascertain what student ids require accounts based on the upcoming assessments

1.3 - The web page and backend must provide validation for the signup process (emails, names, student ids and passwords must be valid)

1.4 - Signup tokens related to the signup process must be securely stored

**Sign-In**

2.1 - The system must alert the user if a match is not found for the entered ID with an “Invalid ID” message.

2.2 - The system must warn the user when they attempt to log in using an account that hasn’t yet been activated and provide them with a link to begin the account activation process.

2.3 - The system must alert the user if their password is incorrect

**Password Reset**3.1 - The backend must be able to verify the user’s identity through the use of a user ID.

3.2 - The system must be able to alert the user if the password which has been entered is incorrect.

3.3 - The system must notify the user if the password they have entered does not meet the security requirements.

**Starting Assessment**4.1 - The system must be able workout what assessments the student can take at the current time

4.2 - The system must be able to present the user with plagiarism and banned manned material rules.

**Sitting Assessment**5.1 - The system must be able to display an accurate assessment timer

5.2 - The user must be able to ask question to a UC/Invigilator via the student client

5.3 - The system must be able to present Notification from the UC

5.4 - The system must be able to notify students when they have limited time left

5.5 - The system must provide a mechanism for saving a students work

5.6 - The system must provide the ability to end an assessment early

5.7 - The system must be able to sync the student work folder to the backend server

**Changing Computer during an Assessment**6.1 - The system must be able to retrieve the most recently updated test data from the database

**Test-time Alteration during the Test**7.1 - The system must be able adjust the duration of a students assessment

**UC Send Alerts to Students**8.1 - The system must store the messages sent to and from for auditing purposes

8.2 - The system must be able to send and retrieve alerts

**UC/Invigilator View Student Messages**9.1 - The system must be able to receive messages

9.2 - The system must be able to mark the messages as resolved

**Create Incident Notes During Test (UC)**10.1 - The system must store the messages sent to and from for auditing purposes

10.2 - The system must be able to send and receive messages

**Ending an Assessment**11.1 - The system must be able to sync the work folder with the backend

**Viewing Work Folder after an Assessment**   
11.2 - The system must be able to restrict access to the work folder

**Uploading an Assessment**13.1 - The system must be able to interpret a list of student ids to student users

13.2 - The system must allow for assessment documents to be uploaded

13.3 - The system must be able to save and validate assessment information

**Downloading Test Files**14.1 - The computer that the web application is being accessed on must be connected to the internet.

14.2 - The browser that the web application is running on must be installed correctly.

**View/Edit Upcoming Assessment**15.1 - The system must be able to update assessment information

15.2 - The system must be able to upload assessment documents

**Health Monitoring**16.1 - Provide observability to usage of

* + CPU
  + Threading
  + Network usage

16.2 - Must be able to send emails, when conditions based on observability metrics are met. E.g when ram usage > 90%

16.3 - Display Observability metrics as graphs

**Performance Monitoring**17.1 - Provide observability to usage of

* + Rsync processes
    - Num processes
    - Mean duration
    - Max duration
  + API usage

17.2 -Must be able to send emails, when conditions based on observability metrics are met. E.g when rsync num process > 100

# 

# **Non-Functional Requirements**

| Non- Functional Requirements | Student(s) | Not Observed | Occasionally Observed | Somewhat Observed | Mostly Observed | Fully Observed |
| --- | --- | --- | --- | --- | --- | --- |
| 1.1 | Sanjay,  Isaac |  |  |  |  |  |
| 2.1 | Jay, Filipe |  |  |  |  |  |
| 2.2 | Jay, Filipe, Isaac |  |  |  |  |  |
| 2.3 | Jay, Filipe, Isaac |  |  |  |  |  |
| 2.4 |  |  |  |  |  |  |
| 2.5 | Filipe |  |  |  |  |  |
| 3.1 | Filipe, Sanjay |  |  |  |  |  |
| 3.2 | Sanjay, Isaac |  |  |  |  |  |
| 3.3 | Sanjay,  Isaac |  |  |  |  |  |
| 3.4 |  |  |  |  |  |  |
| 4.1 | Filipe, |  |  |  |  |  |
| 4.2 | Sanjay, Isaac |  |  |  |  |  |
| 4.3 | Filipe, Isaac |  |  |  |  |  |
| 4.4 |  |  |  |  |  |  |
| 5.1 |  |  |  |  |  |  |
| 5.2 |  |  |  |  |  |  |
| 6.1 | Filipe, Isaac |  |  |  |  |  |
| 6.2 | Sanjay, Isaac |  |  |  |  |  |
| 7.1 | Filipe |  |  |  |  |  |
| 7.2 | Isaac |  |  |  |  |  |
| Non- Functional Requirements | Student(s) | Not Observed | Occasionally Observed | Somewhat Observed | Mostly Observed | Fully Observed |
| 7.3 | Isaac |  |  |  |  |  |
| 7.4 | Sanjay, Isaac |  |  |  |  |  |
| 8.1 | Filipe, Sanjay |  |  |  |  |  |
| 9.1 | Sanjay, Isaac |  |  |  |  |  |
| 10.1 |  |  |  |  |  |  |
| 11.1 | Isaac |  |  |  |  |  |
| 12.1 | Isaac |  |  |  |  |  |
| 13.1 |  |  |  |  |  |  |
| 14.1 |  |  |  |  |  |  |
| 15.1 | Isaac |  |  |  |  |  |
| 15.2 | Isaac |  |  |  |  |  |
| 15.3 |  |  |  |  |  |  |
| 15.4 |  |  |  |  |  |  |
| 15.5 |  |  |  |  |  |  |
| 15.6 |  |  |  |  |  |  |
| 15.7 |  |  |  |  |  |  |
| 15.8 | Filipe, Isaac |  |  |  |  |  |
| 15.9 |  |  |  |  |  |  |
| 15.10 |  |  |  |  |  |  |
| 15.11 | Isaac |  |  |  |  |  |
| 16.1 | Filipe, Sanjay,  Jay,  Isaac |  |  |  |  |  |
| 17.1 | Filipe, Sanjay,  Jay,  Isaac |  |  |  |  |  |
| 18.1 | Filipe,  Isaac |  |  |  |  |  |
| 19.1 | Filipe, Isaac |  |  |  |  |  |
| 20.1 | Filipe, Isaac |  |  |  |  |  |

# **NFR List**

**User Interface**

1.1 - Colour Scheme

**Student Client**

2.1 - Dashboard

2.2 - Future and Past Assessments

2.3 - Display Assessments

2.4 - Download Past Assessments

2.5 - Current Assessment

**Tab Bars**

3.1 - Navigation Tab Bar

3.2 - Notifications

3.3 - Questions

3.4 - Settings

**UC Client**

4.1 - Dashboard

4.2 - Create Assessment

4.3 - Current Assessment

4.4 - Student Profile

**Hardware Interfaces**

5.1 - Compatibility with non-desktop devices

5.2 - Interface provided to computers in lab

**Software Interfaces**

6.1 - Interfaces provided to the OS used by the computers in lab

6.2 - Software provides forms for test creation and scheduling

**Communications Interfaces**

7.1 - UC client provides communication interface to web browsers on computers

7.2 - UC client provides communication interface to network server used by invigilators

7.3 - Student client provides communication interface to network server used by students to store their submissions

7.4 - Student client provides communication interface to query messaging platform used to alert invigilators.

**Persistent Error Messages**

8.1 Error messages remain displayed until window has been closed

**Help Alert Response Time**

9.1 - Alert message sent to UC within 10 seconds after the help request is sent by the student.

**Ignore Alert Response Time**

10.1 - Alert message sent to student within 10 seconds after a help request is ignored by the UC.

**Setup Efficiency**

11.1 - A user should be able to log in and set up their assessment session within 5 mouse clicks.

**Folder/Server Backup**

12.1 - The folder should be synchronised with the server every 5 minutes.

**Coding Standard**

13.1 - The code adheres to Curtin University’s coding standard.

**Safety Requirements**

14.1 - Data stored in folders provided to the students are backed up regularly and it meets the requirement set in 12.1.

**Security Requirements**

15.1 - Users verified via ID login through the initial application view.

15.2 - Students receive signup emails with a signup link to create accounts.

15.3 - Student accounts cannot view assessments prior to the start of the test.

15.4 - Strong password policy

15.5 - Quick password recovery

15.6 - Failed password attempts recorded

15.7 - Only UC accounts allowed to access student work folder once the assessment is completed

15.8 - Student accounts cannot manipulate test timer apparatus manually

15.9 - All work folders secured during and after test

15.10 - Application code obfuscation

15.11 - Database secured against SQL injections

**Robustness**

16.1 - Highly modular software where additional extensions can be integrated smoothly

**Maintainability**

17.1 - Written with clean code to future proof against functional obsolescence

**Availability**

18.1 - Assessment material available to students once the session time has commenced and not before

**Usability**

19.1 - Ensures that minimal time is spent setting up the application

**Documentation**

20.1 - Project includes adequate documentation for students and admins for installation, use, and source code building.