Requirement Document

**Functional Requirements**

**System**

* System should be available in Windows, Mac and Linux. The priority is getting it working on Windows.
* The application must adhere to security standards set by organisations like OWASP.
  + The system must keep data secure.
  + The system must ensure that users only have access to information that they have access to.
  + The system must use encryption when sending data to and from the database.
* The application is primarily an offline application, not needing a constant connection to the internet.
* Data being sent to the database must be validated before being sent. This is to ensure data integrity.
* Data being downloaded from the database must be validated before being downloaded. This is to ensure data integrity.
* Files in the system must be stored in a database and be encrypted.
* APIs should be used for validation of data. Stick to DRY principles and don’t code stuff that has already been implemented before and has already been verified to be good.
  + The APIs used should be verified to ensure that they stick to security standards.
* The database that the application interfaces with should be in 3NF.
* There will be three databases for the application.
  + User – Stores information such as
    - First Name
    - Last Name
    - Email Address
    - Telephone Number
    - Type of user
    - Class code
  + Assessments – Stores information such as
    - Assessment Code
    - Class Code
    - Name of assessment
    - Type of assessment
    - Questions of assessment
    - Duration of assessment
    - Due date
  + Files – Stores information such as
    - Name of file
    - Type of file
    - Assessment code
    - User code
    - Class Code
* When the student is doing an assessment, the system should have access to their clipboard. The student will be notified of this.
* The application should have access to view the currently opened applications on the student system when they do their assessment. The student will be notified of this.
* The application should display X amount of posts on the dashboard per page that can be set by the user.

**General User**

* Users should be able to log into the system.
  + Users in the system have a username and password.
    - The username should be an email address.
    - The password should be at least 8 characters long, with at least 1 numeric value and at least 1 special character.
  + The system should remember those details so the user does not need to input it again.
  + Users have to verify their email address before they can use the application.
* The user should have the option to start the application on system start up or whenever they open the application.
* The application should connect to the internet to download information that has changed on the dashboard i.e. posts, assessments. It should do this:
  + Whenever starting up the program
  + On user request
* Users should be able to upload files into the system.
* Users should be able to download files from the system.
* Users should be able to edit details such as:
  + Address
  + Telephone Number
  + Name (With Admin Approval)

**Teacher**

* Teacher has a set of classes they teach, each denoted by a code.
* Teacher can upload posts for those classes.
* Teachers can delete posts made for those classes.
* Teachers can edit posts made for those classes.
* Teachers can set assessments of different types for those classes.
  + Assessment where the students have to upload a file.
  + Assessment where the students answer on the application.
  + Quizzes.
* Teachers can delete assessments.
* Teachers can edit assessments on things such as due dates, or descriptions.
* Teachers can grade assessments on the application.
* Teacher should be able to receive contact information on the students attending their class.

**Student**

* Students should be able to view posts for classes they enrol in.
* Students should be able to do assessments for classes they have enrolled in
  + For assessments where they have to upload a file, they should be able to upload a file to submit.
  + For assessments where they answer on the application, students should be able to answer on text boxes and submit.
  + For assessments that are quizzes, the students should be able to answer questions to submit.
* Assessments must have a countdown on them that is visible for the students to see.
* Assessments should give reminders on how much time is left by posting a notification on their system.
* Once the assessment has been submitted, students should not have the ability to edit their submission unless resubmission is allowed.
  + The time for resubmission is based on the how much time is left for the assessment.
* After the countdown hits zero and the student has access to the internet, the application should upload the assessment to the database.

**Admins**

* Admins should be able to delete records of user details.
* Admins should be able to accept or reject requests for detail changes sent to them.
* Admins should be able to monitor the database.

**Non-Functional Requirements**

* Users should not have access to pages that they shouldn’t have access to.
* Data sent to and from the databases should be encrypted.
* Data must be validated before being sent to and from the database.
* The application should be 500MB excluding any information downloaded onto the dashboard.
* The application should switch between pages in less than three seconds.
* The application should have very little downtime.
* Validation of data being downloaded should take up to five seconds.
* SQL queries should be efficient and should take up to three seconds to achieve
* The time taken for downloading and uploading files is dependent on the user’s internet connection speed.
* If downloading or uploading files is interrupted, the system should halt the transfer for security reasons.

ERD

A screenshot of a computer

Description automatically generated

Class Diagram

A diagram of a computer

Description automatically generated with medium confidence

Logging in state machine diagram

A diagram of a blue rectangular object with black text

Description automatically generated

Student Dashboard state machine diagram

A diagram of a student's work flow

Description automatically generated

Assessment taking Sequence Diagram

A screenshot of a computer

Description automatically generated