Team Members:
Tyler Dionne (tdionne2021@my.fit.edu), Kendall Kelly (kelly2021@my.fit.edu), Braden Corkum (corkumb2013@my.fit.edu)

Project Advisor:
Sneha Sudhakaran, ssudhakaran@fit.edu

Project Title:
Automated FIT Transcript Evaluator

Client:
Sneha Sudhakaran

Dates of Meetings with the Client for Developing this Plan:
Once a week every two weeks

Requirements Document

1. Functional Requirements

1.1 User Accounts

Description: The system will allow users to create user accounts. These accounts will only have access to the files they input and their output file.

Behavior:
- To create an account the user must provide an email and password.
- Users can login with these credentials, if they give the wrong email or password an error message will occur.
- Users are able to input a .txt file with their transfer credits and a .txt file with a link to the FIT catalog that they want to compare their transcript with. They are also able to access the output file that includes the evaluation results.

1.2 AdminAccounts

Description: This is a separate account from the user accounts. There is only one admin login. This admin account will have access to the site source code and backend processing.

Behavior: Admin login with special credentials, if credentials are incorrect an error message will occur and access will be denied. When access is granted the admin is able to access sensitive information about/on the website as well as backend processing.

1.3 File Upload

Description: There are two main types of uploads; transcript and desired catalog. These files will be compared and evaluated in order to determine if there are equivalent classes to the transfer credits at FIT.

Behavior: Two input files that if in the proper format the file will be accepted, if not an error message will be displayed.
- The transcript file will be a .txt file that will include all of the transfer credits they would like to have evaluated.
- The FIT catalog file will be a .txt file that only includes the link to the desired catalog on the FIT website.

1.4 Automated Credit Evaluation

Description: The web app should be able to automatically evaluate transfer credits against the credits in the desired FIT catalog.

Behavior:
- The system processes the uploaded transcript and selected catalog.
- It matches courses from the transcript with FIT courses based on titles, descriptions, and credit hours.
- The system generates a detailed evaluation report, which will be the output file.

1.5 Results Output

Description: After the evaluation an output file with the evaluation results will be provided and stored on the system.

Behavior: A summary of the accepted credits and not accepted credits will be included in the output file. If accepted, the equivalent FIT course will be included in the output as well. Users should be able to view, save, and download/print the report.

2. Non-Functional Requirements

2.1 Security

Description: Security measures will be put in place to prevent unauthorized access and protect sensitive data.

Behavior:
- All data transmission is encrypted using protocols such as HTTPS.
- We will implement protections against common web vulnerabilities such as SQL injection, XSS, etc.

2.2 Performance

Description: The credit evaluations should be processed quickly and efficiently.

Behavior: The average evaluation time should be completed within 30 seconds.

2.3 Scalability

Description: The application should be able to accommodate user growth and an increase in data.

Behavior: The database design should support large data sets and the architecture needs to allow for scaling of resources.

2.4 Reliability

Description: The web application needs to be reliable and available at most times.

Behavior: We are aiming for at least 85% uptime during the periods of time where the website is at peak usage. Backups will need to be performed regularly in order to avoid loss of any data.

2.5 Usability

Description: Our web app needs to be user friendly.

Behavior: The UI should be designed to be easy to use by making sure navigation and instructions are clear. Helpful error messages need to be provided to guide users.

3. Interface Requirements

3.1 Web Interface

Description: The web app must be responsive and accessible from most desktops and/or laptops.

Behavior: Must work on most desktops and laptops as well as being able to work on most popular browsers such as chrome and safari.

3.2 File Upload Interface

Description: Needs to have an intuitive interface like a button for the file uploads.

Behavior: Should have two upload buttons, one for the transcript file and one for the catalog file. There should also be a progress bar for the uploads and clear messages should be shown whether the upload was successful or not.

3.4 Admin Interface

Description: There will be a separate interface for admin users that will have access to more sensitive information.

Behavior: The admin interface will provide access to backend processes and the website source code. It will also give access to manage user accounts and monitoring of the performance of the application.