Software Implementation and Testing Document

**For**

**Group <3>**

Version 3.0

**Authors**:

Luke Baker

# Programming Languages (5 points)

Currently the main webpage is designed using HTML and CSS. The backend is developed with Python using the Flask API as well as SQLAlchemy for the database components.

As of increment two the main components of the projects of the project are the several webpages designed, the Calendar Interactable, and the backend code. For the webpages we are currently using HTML, CSS, and some JavaScript. The calendar interactable is also using the previous three languages. The backend is done entirely in Python.

As of increment three and the conclusion of the project the main components are the several webpages designed, the Calendar Interactable, the Database code using Flask and SQLAlchehmy. The webpages use HTML, CSS, and the calendar specifically uses a lot of JavaScript. The backend and database as mentioned use a great deal of Python specifically Flask and SQLAlchemy. Flask and FlaskForms are used to display information upon signing in and loading specific pages as well as getting information from the various forms and then Alchemy is used to check the database or enter new information into the database.

# Platforms, APIs, Databases, and other technologies used (5 points)

As previously mentioned the main API in use is the Flask API for a lot of the backend features. It was intended to use the Canvas API and there is some code about this is the CanvasExtractor.py file. There is a database being used using the SQLAlchemy Python library.

# Execution-based Functional Testing (10 points)

Because project is in very early stages minimal testing has been performed. Main webpage can be viewed by opening the file MainPage.html using a web browser and this has been tested. Further, Arturo tested the template Flask webpage files and they have been successful. Once done merging Main Page with the Flask code will conduct more testing.

Have tested Flask interactions with webpages, through executing commands to launch webpages and no issues. We have also tested webpages features such as changing to other pages and how they interact with Flask and the only issue so far is with the sign-in page. This page uses JavaScript which is not playing well with Flask. The Sign-In page in testing will display an error message if the email/password are incorrect but will not move to the main page when the correct testing information is entered. This bug will need to be fixed.

We have also tested some features of the Calendar Interactable, it is displaying properly on the Main Page, but as for current bugs it is not able to change months at the moment and is obscuring sections of the webpage.

Through iteration three a lot more testing went into the project as we finally implemented the database and the ability to add events to the calendar.

For webpage design and layout the majority of testing was just through viewing the webpages and trying every link available across all the various pages. This was to ensure that all links worked properly and their backend features worked as intended, such as when selecting the settings page the users current email address was displayed.

For Flask features again this was mostly tested through viewing the webpages. We would test entering invalid items into the forms across the various pages to test how Flask would respond and if the correct error messages were displayed when invalid options were attempted.

For testing the database features this was tested primarily through trying to enter incorrect options into the create account, sign in, and user settings forms. For example on the create account page we made sure that all settings must be correct and that both passwords needed to match, we would test this by entering invalid emails and through mismatching passwords to see if the website would act correctly and display correct errors. Along with this we tested the database and how it would handle if an email address already in use would be entered, this caused errors and so I coded an additional error checking form to ensure that if the email entered was already in use it would not send an error flag and not accept the input, showing that testing was successful in finding issues. Along with this because we added the functionality to change user settings via the settings page, this gave the ability to do more testing. Such as changing passwords and seeing if the old passwords would be accepted on sign in or changing emails and seeing if old emails work! Since all of these did not lead to errors and worked as intended it indicated that our forms and database interactions were working as intended!

Finally, for the Calendar testing, I did not work on the Calendar that was done primarly by Andrew so he would know best, but he tested adding various events to days and removing them to see if issues would occur. He mentioned that the biggest issue he was having was being able to add multiple events to the same day and it seems this bug was not able to be resolved by the end of increment three

# Execution-based Non-Functional Testing (10 points)

No testing of nonfunctional requirements as most of the nonfunctional features have not yet been implemented.

There has been no testing of non-functional requirements yet as all the current non-functional requirements involve the hosting or database both of which are not yet developed.

We had very minimal non-functional requirements so not a lot of testing went into this area. For secure passwords we made sure to encrypt passwords before they were entered into the database so minimal testing went into this area. For speed and reliability we ran the webpage several times and there seemed to be no indication of slow or unreliable interactions with the database or loading speeds so those non-functional requirements seem to be working!

# Non-Execution-based Testing (10 points)

For non-execution based testing the largest test we used was code inspection as we went over code to try and find issues many times and to see if certain interactions were accounted for. Besides this there was not a great deal of non-execution based testing.