Implementation Discussion

We will be running the backend using Python with the Flask framework alongside JWT and Flask-SQLAlchemy. Flask allows us to quickly and easily set up multiple web endpoints. JWT will provide access control for the Flutter mobile app; specifically, the authorization tokens will be created on the Flutter app and sent in the URL request to Flask.

After the endpoint has verified that the JWT is valid, it can make a request to the SQL database using SQLAlchemy, extract the relevant data, and return it via a REST API. The entire server will be run within a Docker container so it can be easily deployed onto a VM or VPS. The frontend will be developed entirely in Flutter, utilizing persistent storage to keep the user logged in using the shared storage system. When the app opens, a JWT will be created for the session, allowing the app to download the list of users' societies and check for new notifications.

We also explored alternatives for this stack. We considered using Firebase, which would remove the need for a dedicated server and SQL database by utilizing Google's serverless backend and built-in authentication. We also looked at React Native as a frontend alternative due to its ease of use and extensive set of libraries to help optimize development; however, the major advantage of Flutter is that all our members already know Dart. We will be using git as our version control system, in VS Code as all our members have experience with this working method – see our repository at https://github.com/Team-4D2/Coursework-1/

Testing Discussion

We will test the non-functional requirements using our tools as defined above – writing tests in Dart and making use of VS Code's development tools.

Non-functional requirements 1, 2, 3 and 5 will be tested by timing the speed of the system, passing if they do not exceed the time limit

Requirement 4 will be tested by simulating logging in and then closing the application, then reopening it and – if the user is still logged in the test would pass.

Requirement 6 will be tested by trying to upload an image over the max file size, if it fails the test passes.

Requirement 7 will be tested by trying to mutate a post while unauthenticated, if the system does not allow the mutation the test passes.

Requirement 8 will be tested by trying to execute XSS/SQL injection attacks if the attacks fail the test passes

The tests will be automatically run via Flutter unit tests; other testing methods would be harder to integrate into the system as we are already using Flutter. As it's the front end of the system, the other components can be tested with it too.

Critical Analysis

This semester we have identified a key gap in the university experience for students - one which we believe our application can fill. By designing our application with our research informed requirements, we believe we are equipped to develop our design. This process was however not straightforward, and we had to overcome various setbacks – complications challenging our teamwork skills, offering us ample opportunity to learn and achieve.

An overshadowing aspect of our teamwork this semester has been the split which occurred, noted in the progress tracker (see Appendix). This split took place roughly halfway through the semester – dramatically adjusting the timeline of completion of the 5 chapters of this work.

Pre-Split

We began tracking our progress in a simple word document, akin to the progress monitoring system advised for our Networks and Databases coursework from last year – this featured a weekly attendance tick box, and a space for notes on the session.

As we decided upon using GitHub to track our overall progress, the initial progress system began to fall away, with little need for notes reflected in our commits – and equally we noticed our former team members often making concessions regarding their attendance, marking themselves as present for instance when they missed most of the session.

Teamwork became difficult after the first week, with our project pitch being the only aspect where we had full proactive participation of all members. Our intent to rotate leadership failed, as the mixed attendance often meant we spent a significant amount of time ensuring our former team members were on the same page, and understood the weekly activities from the module. Equally, the lack of consistent communication exacerbated the matter, as task delegation became frivolous.

As a result of the lack of engagement in the project during this period, there simply was not any conflict; our former teammates generally had little disagreement with many of our suggestions. Any disagreements which did arise were generally discussed, and we came to a consensus on how to resolve it, for instance questions about the suitability of an interview question.

Transitory Period (The Split)

In this period, we worked together to assess the long-term impact of continuing in this manner and emboldened by our Module Lead we pursued the split. This was the teamwork we engaged in during this period – knowing our previous attempts to cultivate further collaboration and team spirit was met with little regard.

Our progress monitoring evolved here, as an Excel sheet was developed – highlighting the disparity in engagement between two groups of members, ourselves, with 100% attendance, and the others with engagement rates ranging from just under 75% to just over 50%, which was what served as evidence of the conflict which lay at the heart of our group – disengagement v engagement.

Post-Split

The above overriding conflict ultimately was resolved, enabling consistent and coherent communication, which allowed for us to excel with our teamwork, as we carefully delegated tasks to ensure we could meet the fast-approaching deadline. To help keep us on track Nickyyl stepped up to become the group's leader and updating the task tracking sheet. This was such that we planned our meetings around the tasks more carefully – ensuring we gave special attention for instance to our core architecture design and system requirements – key components of this first item of our project.

Any minor disagreements we did have were given time to be considered, reviewing the specification, provided samples and our own research to ensure we made the best decisions for the project. We continued using Excel sheet to track our attendance, and our new GitHub repository for our progress.