Coursework 5047 – Functional and quality requirements:

* **Activation of New Society.** Once an application of a new society is approved by the union, the society leader will receive a notification on its account to activate the society. The activation implies that the society is active so that subscription to the society is enabled and information about the society will appear in the union’s website.
* **Update Society Registration Data**. The system should enable the society leader to update its registration data, especially to hand over the society leader to another union member.
* **Management of Society Membership**. The system should enable the society leader to set the option whether joining to the society will be approved manually or automatically.
* **Communication with Society Members**. The system should provide a facility that enables the society leader to view information about its members and communicate with them through group broadcast and individual messages.
* **Termination of the society.** The system should enable the society leader to submit a request of termination of the society. Once a request of termination the society is submitted, it will become inactive so that no new subscriptions will be allowed. When the society termination request is approved, the society will be removed from the system, and all its members will be unsubscribed from the society.

## Quality Requirements for managing a society:

“A mobile application for society leaders to manage and operate societies”

* Security and Privacy protection
* When the society is approved. The activation notification should only be accessible or visible for the society leader who had requested it.
* If society is handed over. It should have all previous owner’s data and information removed.
* Society owner should be able to set the society up to be whitelist only on members joining to avoid any people flooding the society.
* Members should have the option to opt out of society communications. Made clear that they opt in when they join but can change their minds.
* Once the society has been terminated. It should also unattached itself from any of the data of the students themselves. If the student’s data on the database includes ids for the societies they are in, remove it from the list.
* Only the owner of the society can request the deletion of the society
* To request deletion, it must require a security question. Otherwise, direct contact with the approval team.
* Performance
* The creator to be notified swiftly upon creation of the society so they can start to manage it.
* Registration data is done instantly, making sure that access can be handed over as soon as approved/requested.
* Any changes to the details of the society and information of events. Available as soon as possible for all members subscribed.
* Members can join instantly or as close as possible upon signing up or paying for their membership.
* Once members are removed, they have access removed as soon as possible.
* The application loads the joining page within 3s.
* Loading joined members data should not take too long. Have data visible quickly for the society owner.
* Notifications are received to all members that are opted in swiftly.
* Society is terminated and data has been removed from the databases once approved instantly. The society should no longer be accessible by members as soon as it is removed.
* Reliability
* Management system should always be online, 99% uptime.
* Applications should be managed swiftly. Potentially viewed automatically and filtered to allow manager to work through worklist quicker.
* Once data has been updated. It should not affect uptime. Make sure that the society mangers are unable to take down the system by adding incorrect or dangerous data. No accidental or malicious attacks.
* Do not allow the society to be owned by no one. If handed over to another person, do not let that account deactivate and do not let the society be handed over to an empty account.
* Society should always have certain information linked. Contact information or description of the society.
* Access to members should always be available to the owner. Updates about if they have joined or left should be clear to the owner.
* Notification to the owner if an individual message has bounced back as they will need to know if it has been received. Read receipts from the account.
* All relevant data should be removed. If another person would like to create the same society later. They should not be able to pull across any old data that overlaps.
* Scalability
* Increase database capacity, allow for the creation of more societies and increase the capacity of how many members can join each one.
* Having a payment system for societies that require budgets through memberships. Seeing which members have paid.
* Having multiple societies owned by one owner could be expanded. But concerns could be raised in reliability if the owner is spread too thin.
* Making sure that the database can hold the required data but also able to expand to hold more information such as email addresses or even resident addresses.
* Ability to link similar societies together to help new members finding them.
* Option to temporarily time members out if they must be removed but not fully deleted data wise.
* Batch sending out emails to members or even have emails sent out to promote the society to potential members who have agreed over fresher’s week.
* AI assistant to give help to people with questions to any directed questions to assist with the owner’s workload.
* Potentially have a termination period. Have data put to one side for 7-30 days in case it has been accidently deleted or if anyone does appear to want to take over the society once it has been removed.