Bugs/Incomplete parts of the project:

Change User button does not work in Message Page.

Change Profile picture button not yet implemented on profile page. The submit button next to select class list is not implemented.

Add File, Press Enter to Send, and Add Appointment buttons are not yet implemented.

Search Button in the Navigation bar is not yet implemented.

Share Button and Save Button on a Feed Item is not Implemented yet, they will just redirect you to the home page.

Login and Logout system is not implemented yet, any username and password will redirect you to the homepage of the website.

Message Components: These components are used in the Message Page of the website.

Contact: It shows a picture and a username of a user, anyone who is involved in this conversation is a Contact. it was used in the 'Recent Contact'.

Message: This is a single message in the conversation. It contains a picture, username, timestamp and the content.

Message Box: This is the message box containing all the conversations. It shows a list of "Message"s.

Message Editor: The text box is where the user. It contains a "Send", "Press Enter to Send", "Add File" , and "Add Appointment" button. If you check "Press Enter to Send" button, then you can access "Send" button by pressing Enter. The send button will generate a "Message" with the texts in the textbox as contents. Then the "Message" generated is push to the message box.

Recent Contact: It contains all the "Contacts" in this message page. It is a list of "Contacts" that is involved in this conversation.

Other React Components:

Academic Detail: This is the structure for what our Academic Page looks like. It contains a "Post Update Entry" for user to enter their request. A list of "Feed Item"s below the "Feed".

Category box: This is box containing a picture and the class category. In our websites, there are six categories so there are six category boxes. After clicking it,it will redirect the user to the Academic Detail Page that is relevant to the subject. In this Mock-Up , they all redirect you to the same Academic Detail Page.

Configuration: This is used in the Configuration Page. It contains textboxes for the user to change their Username, Email, and Password.

Contents: This contains the content of the “Feed Item”.

Feed: This contains a list of “Feed Items”.

Feed Item: These are posts by different users. It contains a category, summary, and “Content”.

Navigation Bar: This is the navigation bar for every single web page we have. Clicking the

Post Update Entry: The user can enter the summary and question of their post here. Then choose either academic or service. Then after pressing Post, the Post Update Entry will change a “Feed Item” and added to the top of the “Feed”.

Profile: This is the structure of the Profile Page of the website. It contains a picture of the user, first name, last name, academic level, academic institutions, favorite quote and area of interest. The user can change some of these elements by entering information in the text box and click the save button. If user entered information into textbox and clicks cancel button then the text inside textbox will be set back to original. There is a multiselect box with list of classes.

Schedule: This is the component of the Schedule Page. It contains a list of “Schedule Box”s.

Login: This is the component for the Login Page of the website. When the user clicks logout from the “Navigation Bar”, then the user will be redirected to this page.

Schedule Box: It is a single box with appointment details. It has informations such as the name of the person that user had appointment with, date of the post, the contents of the service, start time and end time of the service. It is used in the Schedule Page of the website. On service page, the numbers of “Schedule Box” should be equal to the number of appointments that the user has.

Utility: This is a utility class for developers to use if they want to change numbers into actually readable timestamps.

Team Contribution:

Thien: Made the components for the message page. Wrote server functions to retrieve from database messages, list of users that sent messages, and user profile pics. This page shows contacts on the left sidebar. Retrieves list of users participating in the message, and uses the userid to get the profile pic and name and display that in contacts panel. Implemented the send message function. When send message, message box displays the new message with timestamp, user name, and profile picture.

JuCong: Did the academic page. This is the page show all the requests that relate to the topic that user chose at the academic page. User can post a new request via the entry that on top of the feed. If user try to post a new request, it must chose one of the categories below and image is optional. If user click on the image icon, it will load a new feed with image in it after hit submit. Title, request descriptions and category are required. For category picking, user only need to click on the either of the button on the down-right corner. If any of this is missing, the post function will not tricker. Also users are allowed to “delete” feeds that they don’t want to see by clicking the cross at the right upper corner. If cross is click, the reference pointer in the date of user’s feed will be remove and will not shown in user’s feed anymore. View count will increase whenever this feed is loaded and click function is similar to the one we implement in workshop 4 and 5.

Xin: Did schedule part. refactored schedule page into a react component. Created schedule in the database and built the connection between the database and react components. Created Schedulebox.js and schedule.js. Wrote server functions for schedules. Implemented retrieving data from database. Implemented deleting schedule function(There is a bug to fix, you need to refresh the page manually). Will add “Make an appointment” function later which is user can manually add appointment to schedule.(Asked Prof. Richard, and he said it’s fine for us to do this tiny piece function next time)

Timothy:
Initially created post request interface as part of academic detail. Later,it was refactored into an individual react component to enable it to be used in service detail later This post entry is now functional.I also refactored the service category page into react components and essentially made the service button of the navbar functional.I refactored service detail into a react component.We have chosen to not make this functional now. I also created a sign in to be made fully functional when we get authentication.Hence as at now, the logout button links to the sign in page and the sign in button redirects to the academic category

Karen: Helped edit the database. Created the profile page. Function gets userData and use that user data to display correct information in corresponding field i.e. the education level in the Academic Level field. Can edit the information in text field for certain fields. If you click the save key, then the textfields will update. If you click cancel then any changes in the textfield will display and the field will display original fields.

JianYi: Created the initial template for the database based on the ER diagram. Changed the static UI for config.js into React. Tested all the functions of the website and documented any buttons that does not work / not yet implemented yet. Wrote the descriptions for the components.