

Tutor Tech

COSC368 Assignment 1

Alex Hobson (ID 56391926)

Connor Hitchcock (ID 66119643)

Taran Jennison (ID 67420293)

Executive Summary

During 2020, education providers are being forced to deliver courses online due to the global pandemic. Because of this, Tutor Tech is looking to create a product for these education providers to facilitate communication between students looking for help with their coursework and tutors to answer their questions. Our team has worked together to create a primary preliminary design for a single-page web application. This uses different panels that can be swapped, changed, and hidden without impacting other components of the application.

This design document includes the identification of types of users interacting with our service, scenarios and tasks to be undertaken on our platform, and their prioritisation. This was used to create multiple preliminary design alternatives for all the service's functionality. These preliminary designs were discussed and combined to create a primary preliminary design, utilising the best aspects from the alternative preliminary designs.

Introduction

Tutor Tech is looking to create a product for education providers to facilitate communication between students needing help with coursework and tutors to answer their questions. This is relevant in 2020 where education providers are being forced to deliver classes and support online due to the global pandemic. Our team is designing a platform on behalf of Tutor Tech which will give tutors a dashboard to be able to reply to students' requests. This means that students can get help from tutors online, and tutors can respond in a platform that makes it easy to help multiple students at once.

There is a focus on simplicity for the design, rather than bombarding the user with many advanced features, as the user should be able to seamlessly communicate with a tutor without having to spend time learning how to use the user interface first.

The platform assumes that a user will sign in using their education provider credentials. This means that a user does not need to register an account with the system, and that the education provider will handle instances where the user has forgotten their password.

User identification

We have identified 3 groups of users. The first group of users are the tutors. Tutors are the most important group of users for this service as they are the primary drivers of the service during day to day operations. Tutors often have to work through a queue of students wanting help as there are more students than tutors, so the efficiency of tutors' ability to help students is paramount to the effectiveness of the system. Therefore, it is critical that the tutor can easily navigate the system in order to maximise their efficiency and reduce waiting times for students wanting help.

We have further split the tutors into the sub-categories of student tutors and senior tutors. The student tutors are fellow students, likely from higher course levels, that are tutoring for the given course, whereas senior tutors are tutors that are official members of the faculty for the course (typically having some authority within the course).

The second group of users are the instructors. Instructors are responsible for creating and managing the courses, but during day to day use of the service, an instructor functions as a tutor. As the instructors will only have to handle the course management parts of the system a handful of times for any given course (e.g. at the start of the semester), their role as an instructor is a lower priority than that of the tutors.

The third group of users are the students. Students are the users that are coming to the service to receive help/tutoring. Students are a high priority as they are the main user of the application. Tutors have a higher priority, however, as the time for a student to be connected with a tutor is entirely dependent on how quickly a tutor can answer other queries in the queue. We have sub categorized the students into two groups, students looking for clarification (asking a simple question) and students that require guidance (asking a question that requires explanation of theory). These two sub groups will both use the system in the same manner, but the system needs to be able to handle their differing needs.

Task identification

Many tasks have been identified for how users will interact with the system. These are assigned numbers below, where some of the functionality is applicable to multiple types of users. Explanation on these tasks is further explained under scenarios.

[1] Course Creation (applicable to instructors)

- [1.1] Creating a course
- [1.2] Assigning tutors to course

[2] Enrolling in a Course (applicable to students)

- [2.1] Self enrolment screen

[3] Asking a Tutor (applicable to students)

- [3.1] Submitting request for tutor
- [3.2] Messaging a Tutor
- [3.3] Screen sharing/video calling for detailed help

[4] Providing help to Student (applicable to tutors and instructors)

- [4.1] Viewing and accepting help requests
- [4.2] Responding to student
- [4.3] Elevate for senior tutor or instructor help
- [4.4] Communicating with other tutors
- [4.5] Defer student request to specific tutor

[5] Forum (applicable to all users)

- [5.1] Student posting a question to the forum
- [5.2] Tutor/Instructor posting an announcement to the class
- [5.3] Student/Tutor/Instructor replying to a question/post
- [5.4] Tutor/Instructor moderating the forum

[6] Miscellaneous

- [6.1] Modify/Manage course as an Instructor
- [6.2] Login

Scenarios

1. Course Creation

1.1 Creating a course: The instructor logs into the site and is presented with a home page of courses they run, with an option to create a new course. When the instructor opts to create a new course, they are presented with a form. This form requires details of the course (especially the course title), and has an optional enrolment password. The Instructor then assigns tutors to the course, with an option to flag them as senior tutors. The instructor then confirms the creation and it returned to their homepage.

1.2 Assigning tutors to course: If a course already exists, the instructor for that course will be able to see an edit button on that course in their list of courses. Upon clicking the edit button, the instructor is taken to a pre-filled version of the creation page. The instructor can then use the course creation page to change the tutors enrolled in the course and to modify their senior status, before saving the changes and being taken back to the homepage.

2. Enrolling in a Course

2.1 Self-enrolment: The student can browse all courses within their education provider in order to find one they want to enrol in. To enrol, the student selects the enrol button for that course and, if a password is required, the student is prompted to input the course password. Once the student has enrolled in the course they are then taken to their homepage where their enrolled courses (including the one they just enrolled in) are displayed.

3. Asking a Tutor

3.1 Submitting request for tutor: When a student is on the course dashboard, they can choose to submit a ticket. The student then can fill in a ticket form, indicating whether the ticket is for clarification or assistance, and filling a textbox with a description of their problem. Once filled in the student submits the ticket and is presented with a messaging page which has their description as the first message. The student is free to send more information with following messages before it has been assigned to a tutor.

3.2 Messaging a Tutor: Once the ticket is submitted, the student can use a message chat system to communicate with the tutor. The student and tutor are able to exchange text-based messages and images in this chat. At any time, the student or tutor can mark the chat as resolved, which will lock that chat and resolve the ticket (does not close the chat). Once a ticket has been resolved, the student can opt to re-open the ticket (if more issues arise) and this acts as a new ticket in queue (the queue preview will have the same starting message). If the user reopens the thread, its history is preserved (so the tutor can see what advice was already given), but the user is asked if they would like to change the ticket description before re-opening. The students may only have one active ticket at a time.

3.3 Screen sharing/video calling for detailed help: From the messaging screen the tutor is able to initiate a voice/video call, which will prompt the student to join. When the call is started, the call window will occupy the chat's area. Both users can share their screens simultaneously and can end the call at any time. Finishing/Ending the call does not consider a ticket as resolved.

4. Providing help to Student

4.1 Viewing and accepting help requests: When the tutor is viewing a course they are tutoring in, they have access to see all requests for help and a history of the (resolved) tickets for that course. This means tutors can see open requests for help that they are assigned to, all unanswered tickets/requests for help and all other open requests (belonging to other tutors). Closed tickets can be viewed in a separate list which allows the tutor to see previous responses to common questions. Unanswered requests should have a timer showing how long they have been active/waiting. The list of outstanding tickets should have some visual indication to separate requests for guidance from requests for clarification. The tutor can select a ticket and is able to claim it (to prevent multiple tutors from answering the same question).

4.2 Responding to a student: By default, requests are responded to via a direct messaging. Requests also have the option to start a call, close the ticket, defer to another tutor, or to flag it as requiring guidance rather than clarification. Tutors can still respond to multiple tickets at once and can mark tickets/conversations as resolved when they are finished.

4.3 Elevate for senior tutor or instructor help: A tutor has the option to refer a request to another tutor or instructor. This results in the other tutor receiving a notification and the item being shown in their queue as belonging to them. This is intended for instances where a tutor cannot answer a question, such as it being related to course administration.

4.4 Communicating with other tutors: There is a dedicated text-based chat that also allows sending images for tutors/instructors to message each other. This chat cannot be closed or seen by students.

5. Forum

5.1 Student posting a question to the forum: On the forum page, the student selects the 'new post' button. The student writes out their problem with associated media such as images and code. The student selects the post button which will add the post to the forum. The student can delete the post at any time after posting it.

5.2 Tutor/Instructor posting an announcement to the class: On the announcement forum page, the tutor/instructor selects the 'new post' button. This is the same as for a student creating a forum post, with the additional option of making the post an announcement.

5.3 Student/Tutor/Instructor replying to a question/post: On a forum thread page, a user can select 'reply' on any given post. The user can write a reply with associated media such as images and code. Once choosing to submit this post, it will be added to the forum and the user will immediately see their post on the forum.

[5.4] Tutor/Instructor moderating the forum: The tutor/instructor can lock a thread to prevent further replies.

Tutors/Instructors can also delete student posts and/or replies, requiring an additional confirmation step (such as a dialog box) before being deleted.

6. Miscellaneous

6.1 Modify/Manage course as an Instructor: The instructor has the ability to delete courses they created. On the instructor's homepage, beside all the available courses there will be a 'delete' button that the instructor can select. After an additional confirmation step, this will delete the course.

The instructor has the ability to edit details on their courses. After choosing to edit a post from the homepage showing all their courses, the instructor will be presented with an editable view of the course information and tutors.

6.2 Login: The user can access the site either directly through URL navigation or via links from relevant education provider's pages. If the user has previously logged in on that device and their session is still valid, the user is taken straight to their homepage. If the user is not yet logged in, they are confronted with a login page. Note, the user can only access the system if they are logged in. The user inputs their education provider's login details (username, password). If the login is valid then the user is taken to their home page. If the user does not know their username/password, a link to the tertiary provider's specific password reset processes can be visited.

Task Prioritisation

The goal of the outcome is to pair students with tutors to give students the help they require to complete their courses. Therefore, the most critical features relate to a student being able to ask a question, and a tutor being able to respond to that question as soon as possible.

In order to accomplish this, the most important set of features relate to a tutor's ability to be able to accept, respond and resolve help requests from students. Therefore, the highest priority tasks are that in section four of the task identification, where a tutor provides help to a student. This is the set of tasks with the highest frequency of use, and thus the most effort needs to prioritise them. This ensures the tutor does not waste time having to work out how to use the platform.

Also critical to the core functionality is the ability for students to ask questions in the first place. This is still critical, however, as a tutor may end up answering multiple students at the same time, the prioritisation of students' user experience is slightly lower.

The other sections of tasks relating to course creation, logging in and enrolling in a course are a medium priority. Although, these are features used less often, the absence of those features would make the service unusable.

The forum is the lowest priority as this is an augmentation of the core functionality. That functionality being tutors having the ability to answer the students' questions. However, it is a nice to have as it also encourages students helping each other, which may lower the tutor's workload.

Preliminary Design Alternatives

Taran's Initial Sketches

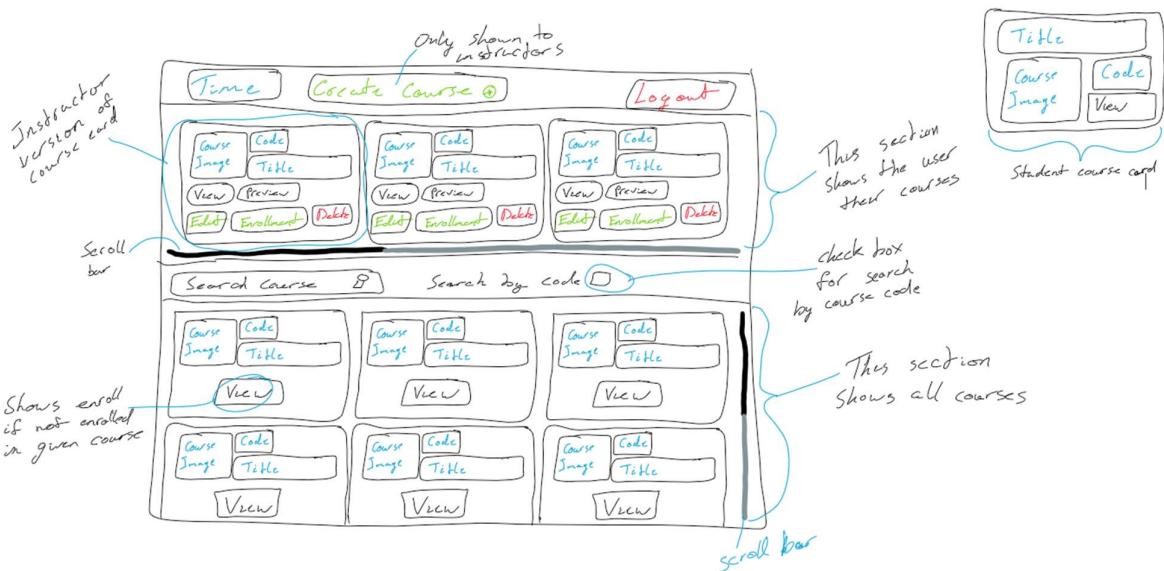


Figure 1: Taran's Homepage Sketch 2

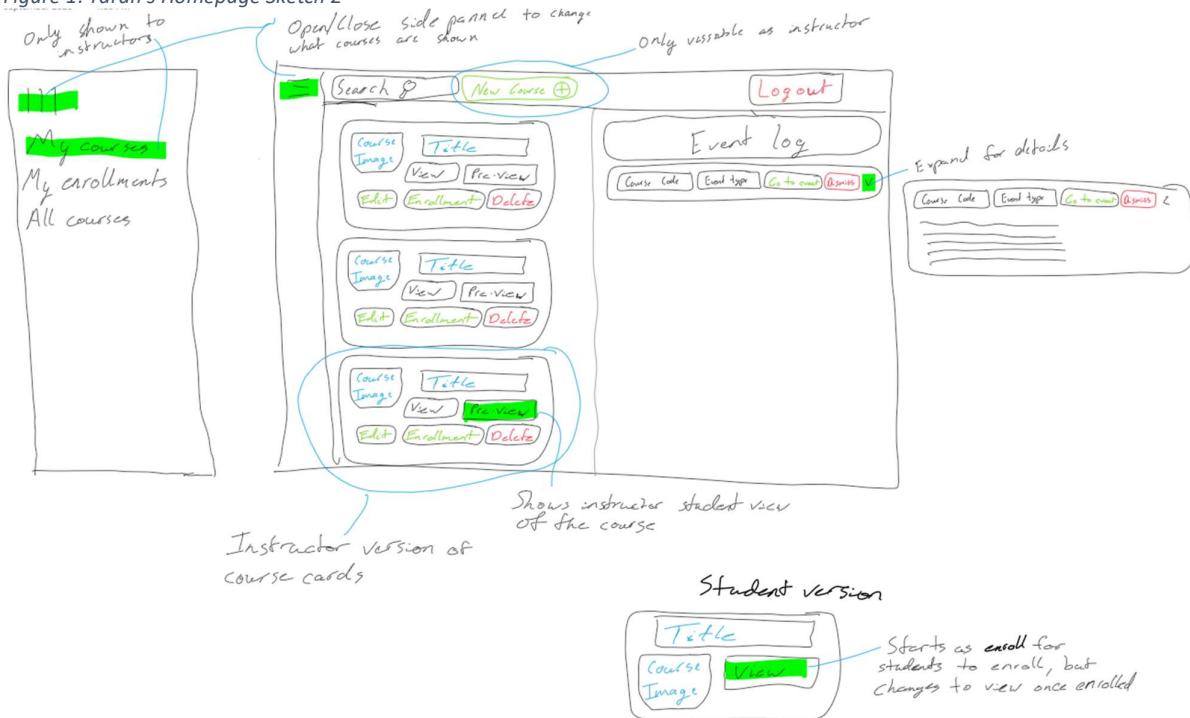


Figure 2: Taran's Homepage Sketch 1

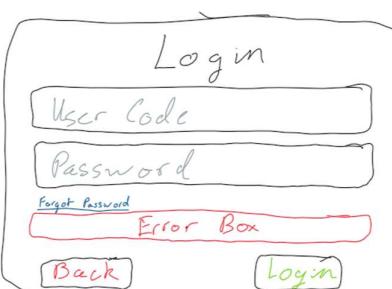
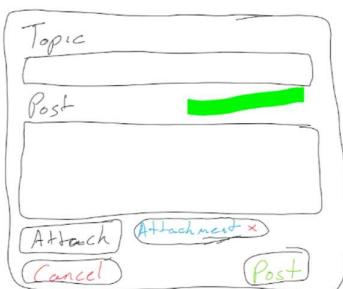


Figure 3: Login Panel Sketch 3 Figure 4: Taran's New Post Sketch 1

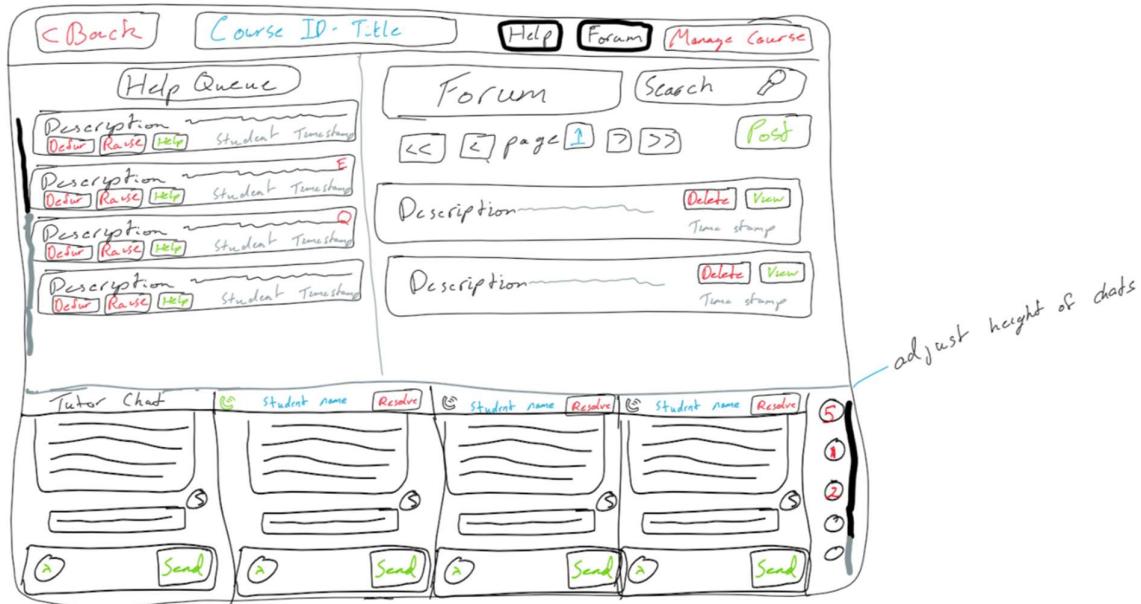


Figure 5: Taran's Tutor Homepage Sketch

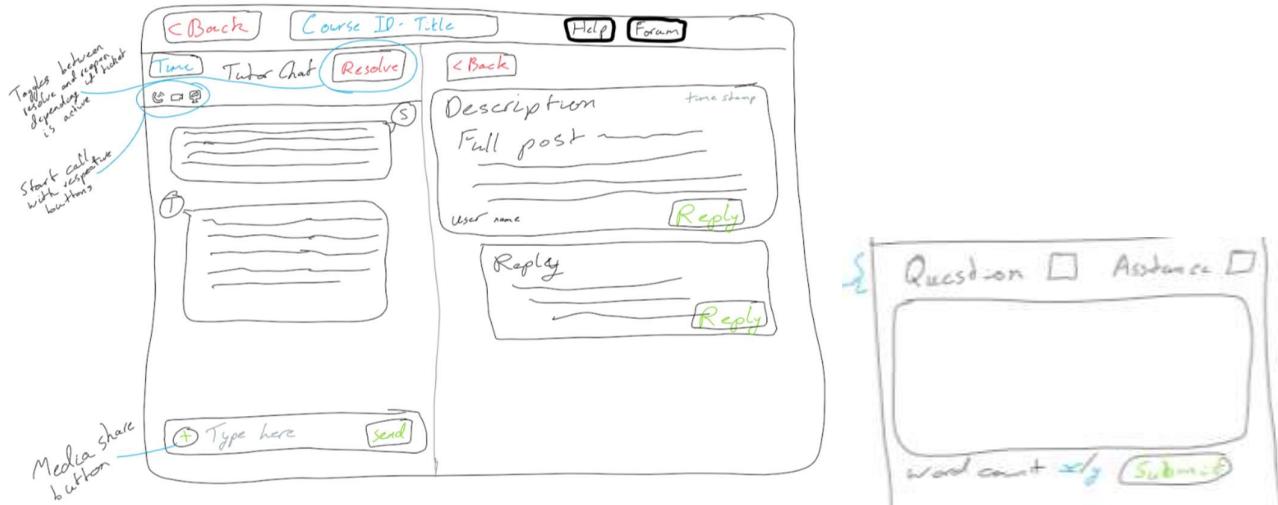


Figure 6: Taran's Student Homepage Sketch

Figure 7: Taran's Ticket Submission Panel (Student Course Dashboard)

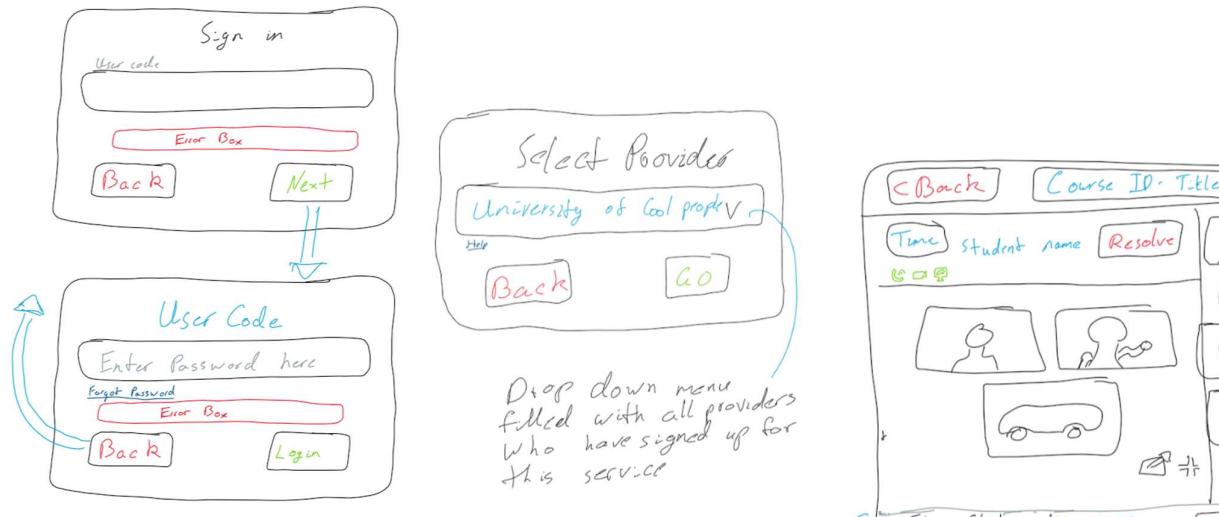


Figure 8: Taran's Login Panel Sketch 2 Figure 9: Taran's Forgot Password Sketch Figure 10: Taran's Video Call Sketch (Tutor Course Dashboard)

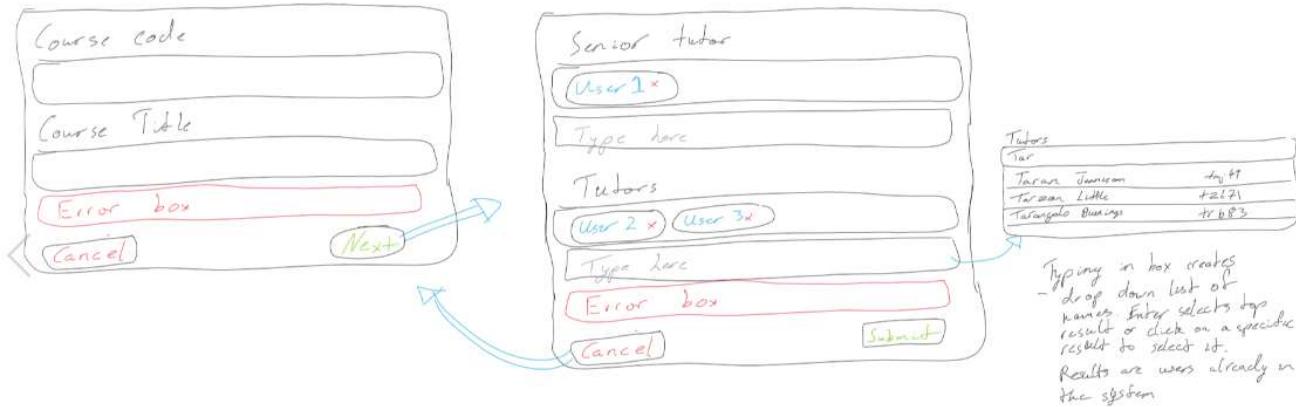


Figure 11: Course Creation Sketch

New Post Sketch (Figure 4): The green highlighted area was identified as a location for a word count for the post text field. Once an attachment is added, it is indicated as a chip to show the user that their attachment has been added while also providing them a clear way of removing it. Pros: The attachment chip gives clear communication to the user.

Login Panel Sketch 3 (Figure 3): The forgot password link can connect to the education provider's database system for password recovery (or their webpage about how to do that). Pros: The simple design reflects that all users understand what a login is and do not need superfluous information. The forgot password link leaves open the opportunity for either internal or external forgot page/mechanisms. Cons: The small login element will mean there is large amounts of white space.

Login Panel Sketch 2 (Figure 8): Separating the username and password allows for verification of each separately and leaves the ability to have the password component to be handled by the education provider if they so choose. Pros: Separation of concerns, flexibility of password implementation to be internal or external. Cons: Needless click through just to move through login. The small login element will mean there is large amounts of white space.

Homepage Sketch 2 (Figure 1): Top scroll pane allows for quick access to the user's current courses with the bottom pane providing the easy ability to search for new courses. The top header bar also provides space for extra functionalities for the logout button and the create course button (if the user is an instructor). Pros: Easy access to both actively enrolled courses as well as all other courses. The use of course cards allows different cards to be shown and modified for different user types without having to modify the page. Cons: The large number of courses displayed can easily clutter the interface.

Homepage Sketch 3 (Figure 2): A horizontal split of the page provides one section for quick access to currently enrolled courses and another section for an event log. The idea behind the event log is to show what has been happening in your enrolled courses. This could store information about recent forum posts or information about active tickets. A fold out menu on the left can allow for changing the contents of the course panel between different selection sets (my courses or courses, etc). Pros: The event log provides a nice preview of the current courses, which could be very helpful for the tutors and instructors. Cons: The folding menu has very few contents so this would be largely white space.

Forgot Password Sketch (Figure 9): There is a simple dropdown selector to select between all education providers that have signed up for the Tutor Tech system. Pros: Locking to a dropdown prevents invalid inputs. Cons: Selecting the education provider that is responsible for your account may be redundant as that information could be gained from the account user login/name.

Course Dashboard (Figure 5 and Figure 6): The idea behind this design is to create a base layout that can easily be modified and have components swapped without effecting the entire page. For example, the user can navigate around the forum panel, whilst dealing with an active ticket on the other half of the page. This design also allows for integration of new components/systems and the use of alternative designed panels, should the need arise.

Forum (Figure 5): This forum panel is rather generic. Each post shows a brief description of the initial post and further details and replies can be seen upon clicking through. The posts are paginated and navigation between panels can sit above the page of posts. The tutors and instructors are also shown the delete button to allow them to delete any forum posts should the need arise. Pros: Pagination makes navigating the forum less overwhelming. Hiding details of each post allows users to navigate the forum without getting stalled by extra, unnecessary details. Cons: Having pagination creates a mess of buttons for navigation between pages and this creates a lot of clutter for the forum.

Help Queue Sketch (Figure 5): The queue is a list of cards where each card displays a summary of each ticket and shows any respective tags that ticket has. The tickets also have buttons for the actions a tutor can take on each ticket, this allows them to easily take an action on a ticket without having to open it. Pros: Displaying a summary of each ticket and its tags allow for tutors to have a quick understanding of a ticket before picking it up. Cons: Having three buttons for each ticket card clutters the user interface.

Ticket Chat, Tutor Version (Figure 5): The tutors can have multiple chats open at a time across the bottom of the page. There is also space for a notification bar at the end of the chat panel to any arbitrary limits on the number of chats that each tutor can have active. Each chat also has buttons for starting a call and resolving the ticket. There is also a dedicated tutor chat for tutors to be able to communicate with each other. Pros: No limits on number of chats. Easily manage numerous chats at one time without having to constantly jump between different pages/windows or rearrange the interface. Cons: Having multiple chats open at once reduces the space that each chat has.

Ticket Chat, Student Version (Figure 6): After the ticket is opened, the ticker side of the interface is converted into a simple chat window. Pros: Opening the chat locks the user out of creating more tickets until the active chat is resolved.

Call, Screenshare, and Video Call (Figure 10): When a call is started, for the tutor, the queue panel is replaced with a call panel. However, for the student, the top of the chat panel is simply replaced with the call panel. The call panel shows a thumbnail for each user in the call and a thumbnail for any shared screens. Clicking on these thumbnails would expand the clicked view to the entire call panel. There are also buttons on the top of the panel to toggle audio (mute), video and screen share as well as an additional resolve button to resolve/close a ticket. At the bottom of the call panel, there are buttons to full screen the call interface as well as pull the call out into a pop-out window. Pros: Provides plenty of options during the call. Keeping the call in the dedicated panel segment allows the user to maintain access to the other features of the app while in a call.

Forum Post Details Sketch (Figure 6): Opening a full individual post for the forum will display the start of the thread at the top with all replies following down the page in chronological order for each level of reply, It is then staggered in from the margin to reflect the level of information the reply falls on. Each post and reply have a dedicated button for replying to that post or reply. Pro: Commonly used forum design mean that many users will be familiar with how it works. Using indentation provides a simple indication of what post/reply belongs to. Cons: Indentations may not always provide a clear indication as to where what a reply is in respect to.

Ticket Submission (Figure 7): Upon opening the course page, the left panel defaults to a ticket submission form with a ticket history below this (the ticket history is not drawn but is indicated for where the highlight is). The student is given the option to flat their ticket as a question or assistance, and there is a small subtle word count label under the text box. Pros: As submitting tickets are a main function of this website, showing the user with this option straight away maybe helpful.

Course Creation (Figure 11): Tutors can be added by typing into the box. This drops down an auto suggestion list of uses as the instructor types the name. The instructor can then add a user via arrow navigation and pressing enter, or by clicking on the user. This adds a chip to the tutor selection/entry box that indicates that the user will be added as a tutor and provides the option to remove the user in an intuitive way. Pros: As submitting tickets are a main function of this website, presenting the user with this option straight way may be very helpful.

Connor's Sketches

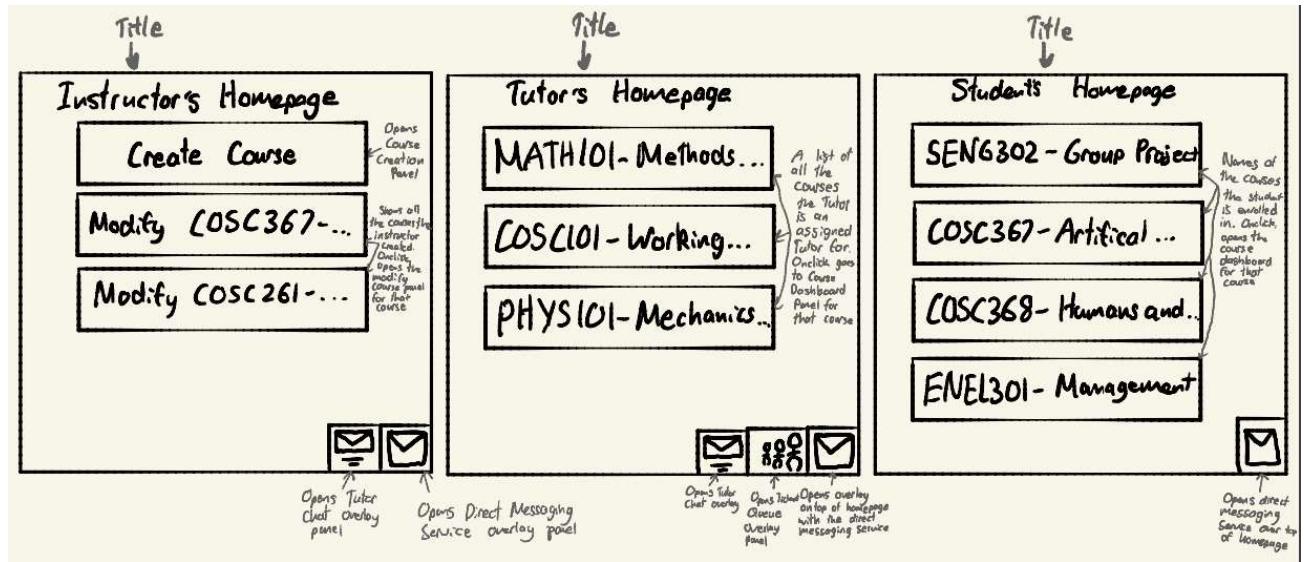


Figure 12: Connor's Homepage Sketch 1

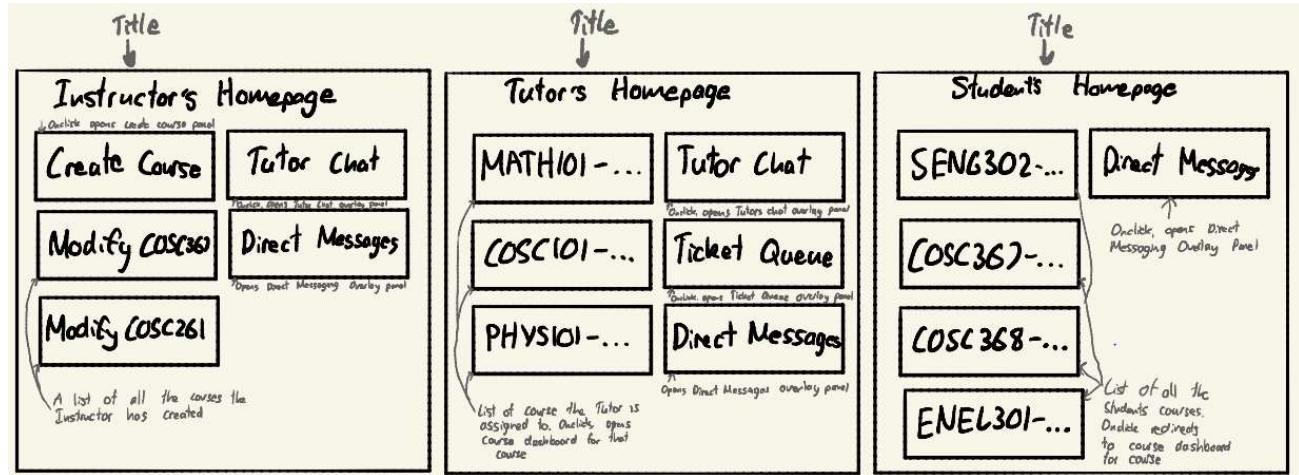


Figure 13: Connor's Homepage Sketch 2

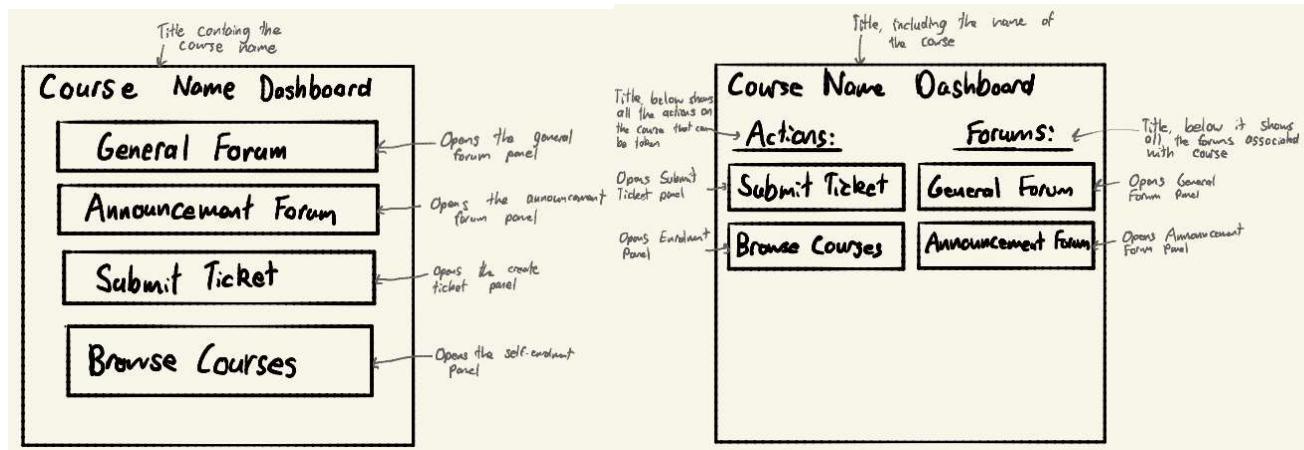


Figure 14: Connor's Course Dashboard Sketch 1

Figure 15: Connor's Course Dashboard Sketch 2

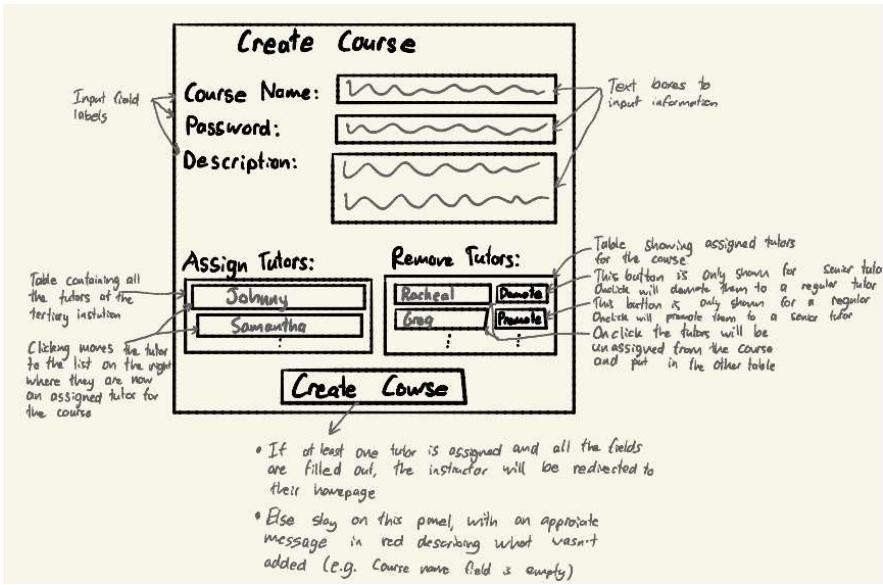


Figure 16: Connor's Course Creation Sketch 1

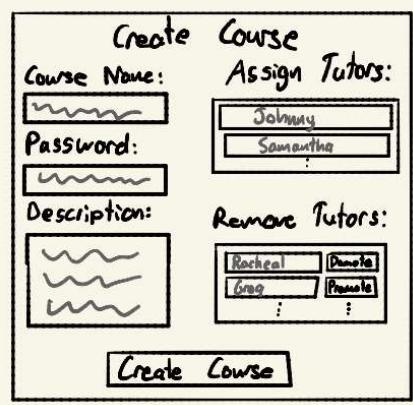


Figure 17: Connor's Course Creation Sketch 2

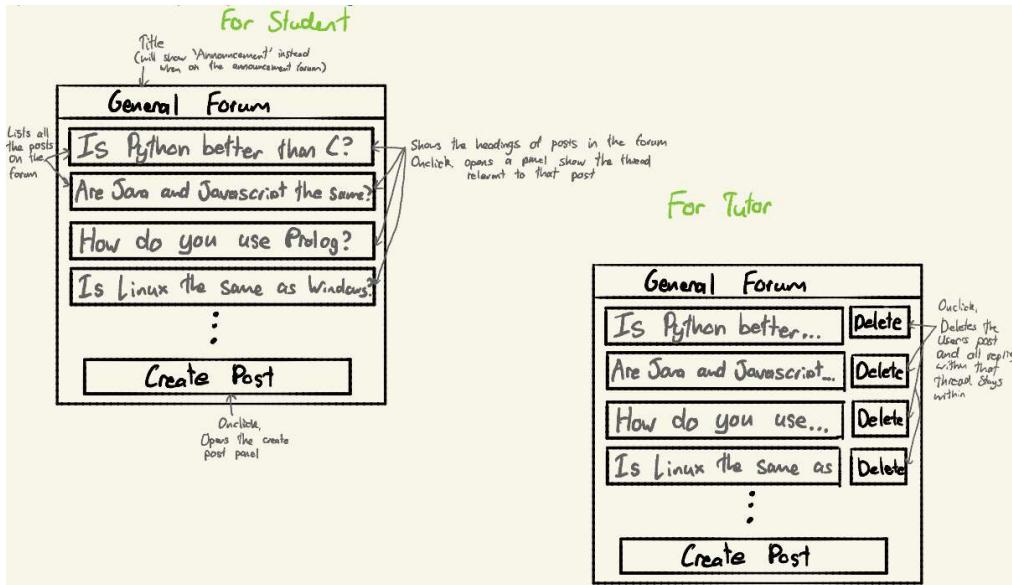


Figure 18: Connor's Forum Panel Sketch

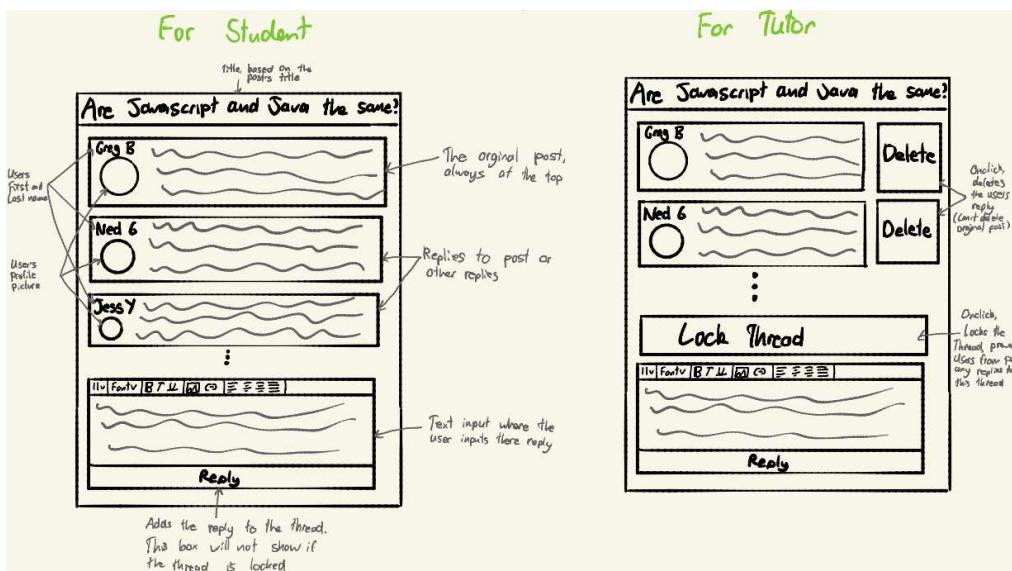


Figure 19: Connor's Thread and Replies Panel Sketch

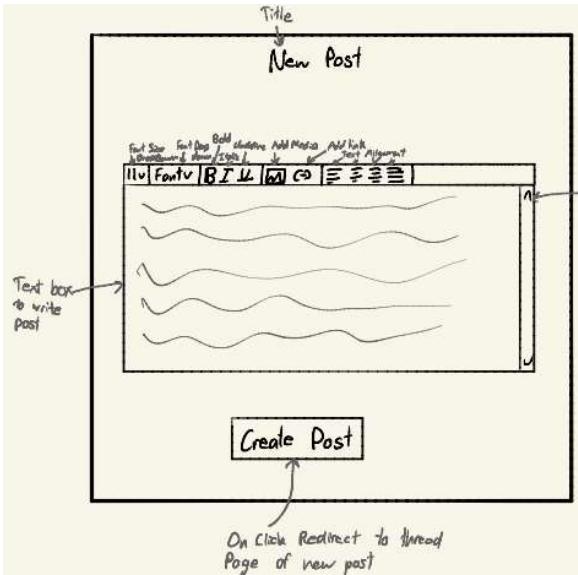


Figure 20: Connor's New Post Panel Sketch

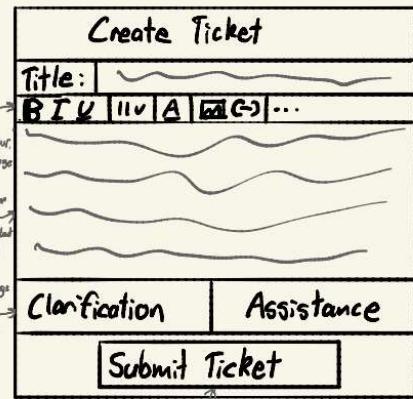


Figure 21: Connor's Ticket Creation Panel Sketch

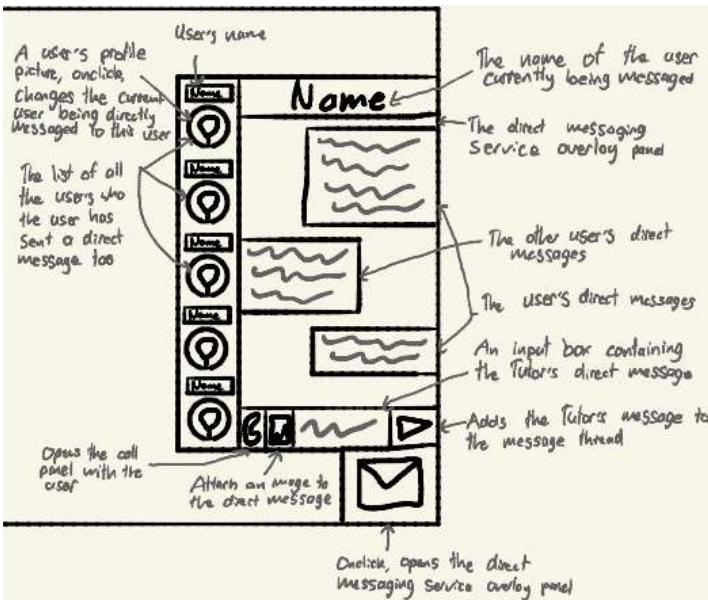


Figure 22: Connor's Direct Message Service Panel (Ticket Chat) Sketch 1

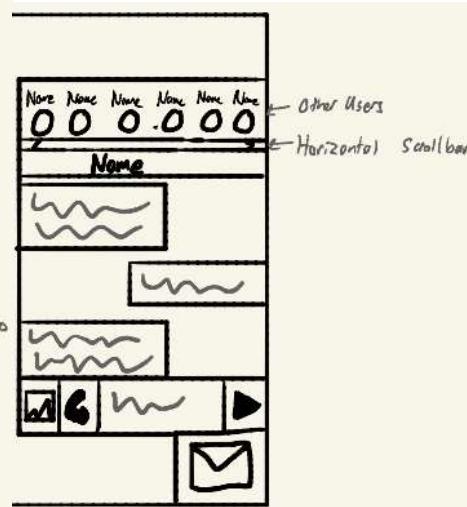


Figure 23: Connor's Direct Messaging Service Panel Sketch 2

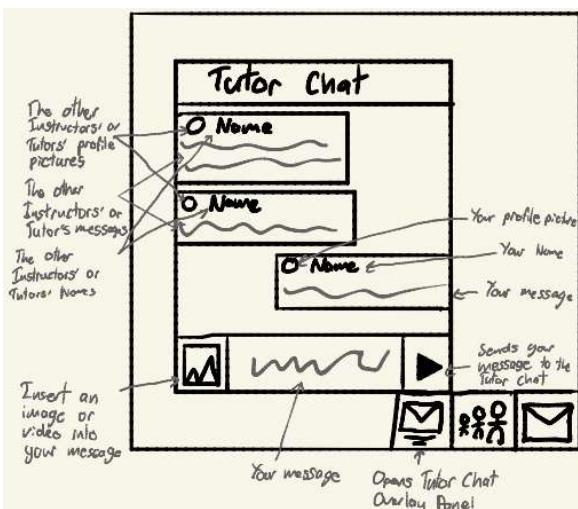
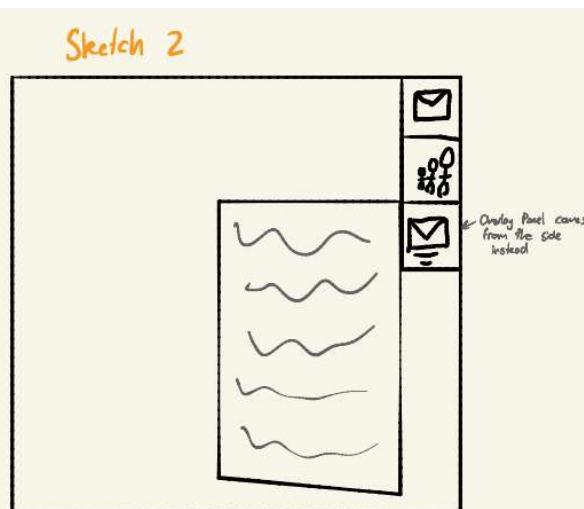


Figure 24: Connor's Tutor Chat Panel Sketch



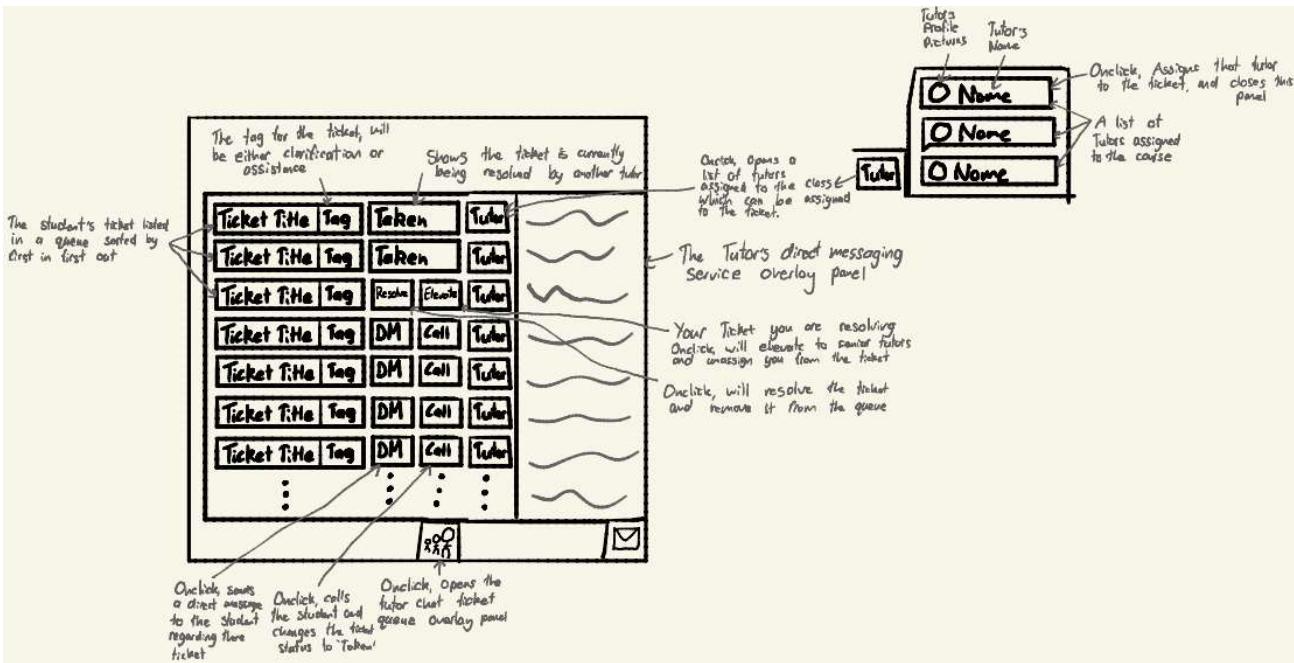


Figure 25: Ticket Queue Panel Sketch

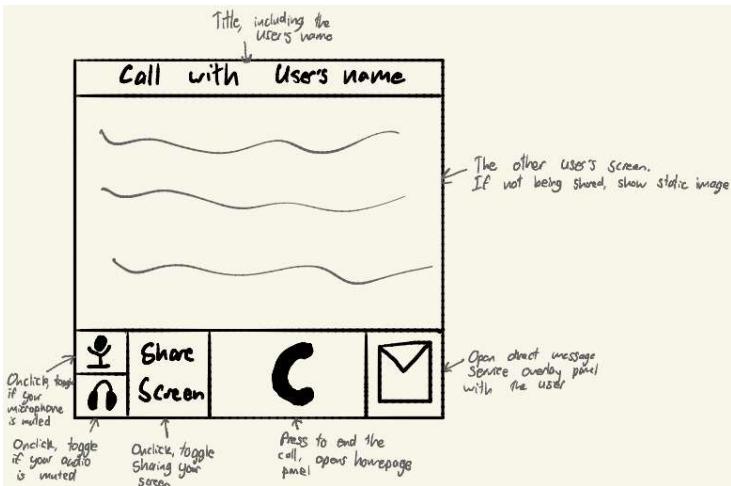


Figure 26: Connor's Call and Screenshare Panel Sketch

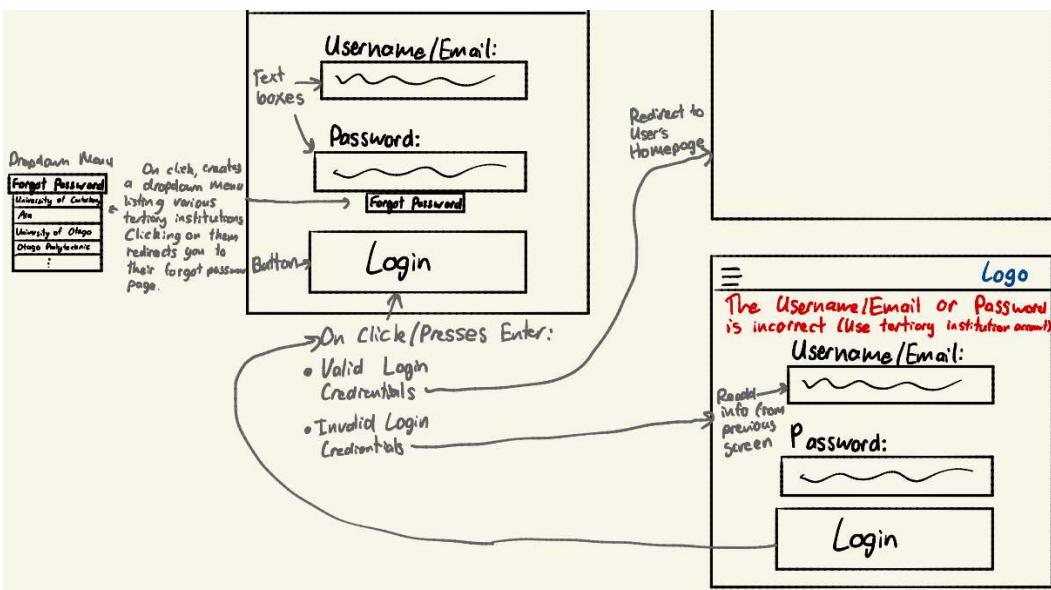


Figure 27: Connor's Login Panel Sketch

The main idea was to build an application with the ability to function within a small window, so the user may easily multitask with other applications at the same time. However, this created the disadvantage of tasks often requiring extra steps of navigation before reaching the desired functionality.

Homepage Sketch 1 (Figure 12): There are different homepages for each type of user as they have different use cases. The instructor can create and modify courses. Therefore, there homepage includes a button to open the create course panel and a selection of buttons to modify their courses. This design assumes tutors cannot enrol in courses, and therefore, will only show a selection of buttons representing the courses they are a tutor for. The students are shown a selection of buttons representing the courses they are enrolled in. Clicking on the course buttons for the student or tutor will open the course dashboard panel for that course. There is a block in the bottom right that will be present on all the screens containing expanding functional overlays, such as the chat overlay. For students, this only includes the direct messaging service or ticket chat, while for the instructor they also have access to the tutor chat, and finally tutors have access to both of these, plus the ticket queue. Pros: Simple interface with a small amount of self-explanatory information that most users should easily understand. Cons: If you have a particular task in mind, it will likely be locked behind multiple stages of navigation though the interface.

Homepage Sketch 2 (Figure 13): The functionality of this homepage is the same as the first sketch, except the permanent block in the bottom right has been changed into a list of buttons that are along the right of the screen. Pros: The interface is easier to understand with words explaining a buttons function. Cons: The user loses the multitasking potential of the overlay block

Course Dashboard Sketch 1 (Figure 14): This design assumes that the general forum and announcement forum are separate forums. The course dashboard shows the name of the course you are in with a heading at the top and then has four buttons. The first two will open the announcement forum or general forum for that course, the third button opens the submit ticket panel and the fourth button opens the course enrolment panel. Of note the third and fourth buttons are only available to the students. Pros: The interface is very simple as there will be a maximum of four actions any user can take on the course dashboard. Cons: By separating information and functionality across multiple pages, the complexity of the system is increased

Course Dashboard Sketch 2 (Figure 15): The functionality of this design is the same as the first sketch, however you now have titles separating the actions and forums into distinct categories. Pros: Clear communication of each element. Cons: Given that there are only four buttons shown for any user, there is little need to separate these segments. Another issue is that Tutors will have no actions as they cannot submit tickets or browse courses.

Course Creation Sketch 1 (Figure 16): The top half of the page is an input section to input information about the course. The course name and password have one-line input areas due to their simple nature. The description is an expandable text area as it is expected to be a sentence to a paragraph in length. Below this is the assigning tutor section. On the left-hand side is a list of tutors from your education provider that can be assigned to the course, and similarly on the right is a list of tutors currently assigned to the course. By clicking a tutor on the left-hand column, you will move them to the right-side column, thus assigning them to the course, with removal of tutors being the same process in reverse. Beside every assigned tutor name is a promote button to promote a regular to tutor to a senior tutor, or a demote button to demote a senior tutor to a regular tutor. Pros: The assigned tutor system shows a minimal interface for all the requirements of assigning a tutor. The layout is similar to the industry standard fill out form layout and thus most people will be familiar with this interface from using it on other platforms. Cons: The description input area does not take up all the available horizontal area. The extra space would be valuable for type paragraph long descriptions for the course. The assigning tutor system uses a small amount of vertical space, and therefore only a few tutors can be seen at a given time.

Course Creation Sketch 2 (Figure 17): This design has the same functionality as the first sketch; however, the assign tutor system is now on the right.

Pros: More vertical real estate for assigning and removing tutors. Cons: You have less horizontal real estate for filling out the information about the course.

Forum Panel Sketch (Figure 18): The forum panel shows a selection of buttons containing the title of each thread on in that course's forum and clicking on said title to open up the respective thread panel. At the bottom is a create post button that opens the post creation panel. The tutor has an extra delete button beside every thread, which

allows them to remove the entire thread. Pros: The layout is simple to understand, as there are only a few elements being shown. Cons: Users cannot see the replies to each thread from the forum page.

Forum Thread and Replies Sketch (Figure 19): This design assumes the user cannot reply to individual replies, only the original post. The panel has a heading at the top showing the title of the thread. Below that is the thread showing a list of cards, with the original at the top and all of the users replies in chronological order following that. Each card shows the user's profile picture and name on the left and their reply on the right. At the bottom is a text area where the user can write out their reply, they also have the ability to bold or italic text and import media through the toolbar or potentially through markdown. At the bottom is a reply button that when clicked (given the input area is not empty) will add that reply to the bottom of the thread. The tutors have two extra features being the ability to delete individual replies with the delete button beside their replies and the ability to lock threads, which will prevent other users from further replying to that thread. This button is located about the reply input area. Pros: The layout is simple and easy to understand as all of the replies are shown in chronological order. Cons: You cannot individually reply to other users replies.

New Post Sketch (Figure 20): In the middle is an input area, where you will put the information about your new forum post. There is a toolbar at the top to allow you to make text bold or italic and other functions such as importing media. However, this toolbar could be removed in place of markdown to achieve the same feature set. At the bottom is a create post button, which will add that post to the list of threads on the course's forum. Pros: Simple interface that primarily shows the user the input area so they can focus on writing their post. Cons: This interface is simple enough that the new post panel could be built straight into the forum panel to reduce the number of steps it would take to create a forum post.

Ticket Creation Sketch (Figure 21): At the top is an input line where the user would input the title of the ticket. Below that is a text area for the ticket description, this also has the ability to import images and bold and italic text through the toolbar or through markdown. Below that is a button that will specify whether the ticket is for clarification or assistance. This differentiation is important as clarification would normally be resolved through one or two messages with the tutor, while assistance would normally require media to be attached with a paragraph explaining the problem, and a video call may be needed as well. At the bottom is a submit button that when clicked, given the input area is not empty and the ticket has been flagged by for clarification or assistance, will create a ticket and add it to the tutor's ticket queue. Pros: The interface is simple and easy to follow. Cons: Given the simplicity of creating a ticket, it could be built into the course dashboard panel to save an extra step to submit a ticket.

Direct Messaging Service or Ticket Chat Sketches (Figure 22 and Figure 23): The direct messaging service resides in a permanent block in the bottom right-hand-side of the screen, that will show up on every page. By clicking the mail icon it will toggle open the service's overlay panel. On the left-hand-side of the panel is a list of all the students the user has contacted, their profile picture is shown with their name right above it. You can click any student's profile picture to open up the chat you have with them, where the student you are currently chatting with will show up at the top of the contact's list. On the right-hand-side is the messaging panel, where at the top is the name of the person you are currently talking to, below that is all the messages you have had with that person, with your messages on the right side and the other users on the left. At the bottom is a text input line where you will type your message, on the right is the send button, and on the left is an image icon where the user can import images and on the left of that is a call button where tutors can engage a call with the student. Pros: You can access your direct messages at any time from the permanent block in the bottom right. Since the direct messages in an overlay you can continue doing other tasks while messaging other people. Cons: You can only have one instance of the direct messaging service open at once. If the service took up the entire screen, then the user could see more messages at once. The permanent block in the bottom right takes up screen real estate. The second sketch of this service is the same thing, except the permanent block is on the top right instead.

Tutor Chat Sketches (Figure 24): The tutor chat panel is similar in functionality to the ticket chat panel, however there is no contact list, and only the instructor and course tutors are in the chat. Where the user's chats are on the right side and other's messages are on the left side. Pros: Simple interface and service that allows tutors to bring up problems with the instructor and other tutors. You can view this chat at any time, whilst also being able to do other actions on the current page. Cons: If the tutor chat panel took up the entire window, users could read more messages.

Ticket Queue Sketch (Figure 25): The ticket queue resides in the permanent block in the bottom right. When it's clicked the ticket queue overlay panel is toggled open. Only tutors can access this ticket queue panel. The ticket queue panels show a list of all the tickets currently in the queue. All ticket types will show the title of the ticket on the left-side, followed by the clarification/assistance tag, and on the right side will be a tutor button that will open a pop-out window. This window will show a list of tutors that the ticket can be differed too. The first type of ticket is the taken tickets shown at the top, which are tickets the other tutors are currently resolving. The second type below that is your tickets, which have two extra buttons, the first resolves the ticket and removes it from the queue and the second is elevated the ticket to a senior tutor. Pros: Having a pop-out panel to defer the ticket to other tutors is useful as a tutor may know another tutor that is familiar with a particular type of problem and can defer said ticket to them as they will be able to solve it faster.

Call and Screenshare Sketch (Figure 26): At the top is a heading showing the student you are in a call with. This design assumes you can only be in a one-on-one call. Below that is the main section where you will either see the other users screen, or a static image. At the bottom is a toolbar which includes a microphone icon to mute yourself, a headphones icon to mute the other user, a share screen button to share your screen with the other user (both screens can be shared at once, but you cannot see your own), a call icon which will end the call (both users can end the call at any time) and on the very right is the direct messaging service button. Pros: The sharing screen function takes up most of the screen. Cons: In this scenario the ability to mute the other person is not necessary as if you are in a call you should be wanting to hear the other user at all times, if not then they shouldn't be in a call. The ticket queue layout is in a logical arrangement that most users could easily follow. You can access the ticket queue at any time. Cons: This layout looks cluttered with some many buttons be shown. If the ticket panel used the entire screen, users can see more tickets and information.

Login Sketch (Figure 27): The login information is directly connected to the education providers database, and therefore the information they use there will work here too. The user enters their username or email in the input line and the password below that. If the user forgets their password, there is a forgot password button which opens a dropdown showing a list of education providers in their country. Once they click on their education provider, they will be redirected to their institutions forgot password help page. At the bottom is a login button, when clicked, given the login credentials are valid, will open their homepage panel. Pros: The user does not have to create a new account to use the system. Cons: The education providers must have their own database with an account for every student.

Alex's Sketches

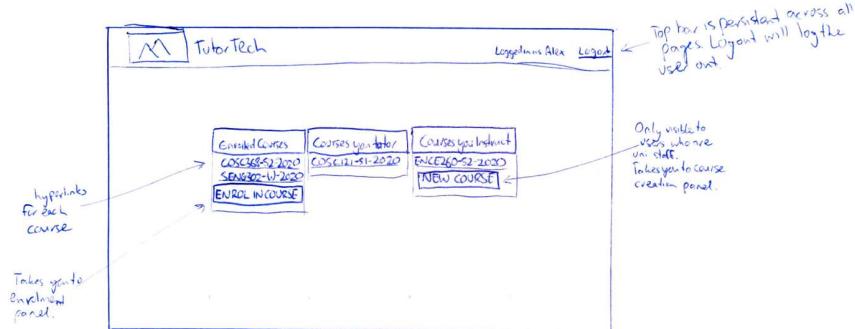


Figure 28: Alex's Homepage Sketch 1

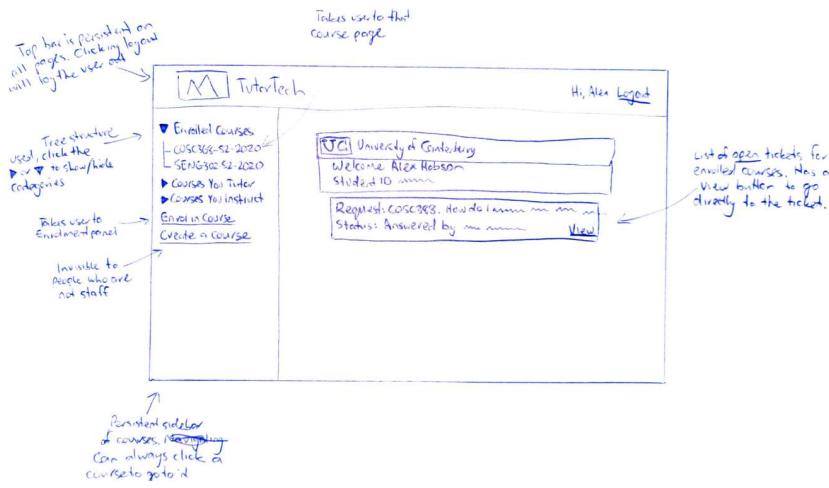


Figure 29: Alex's Homepage Sketch 2

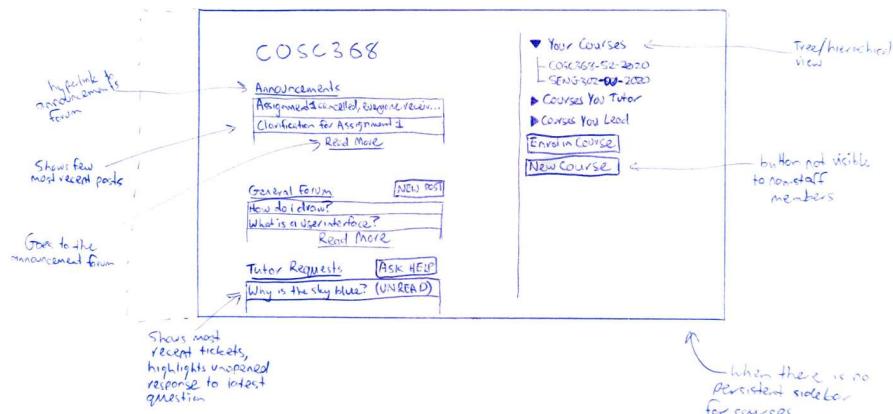


Figure 30: Alex's Course Dashboard Sketch 1

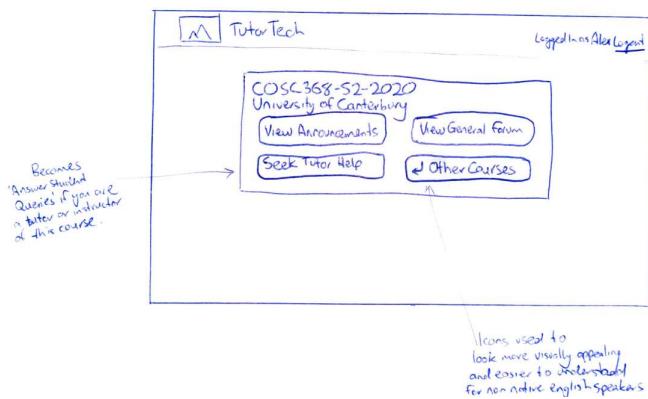


Figure 31: Alex's Course Dashboard Sketch 2

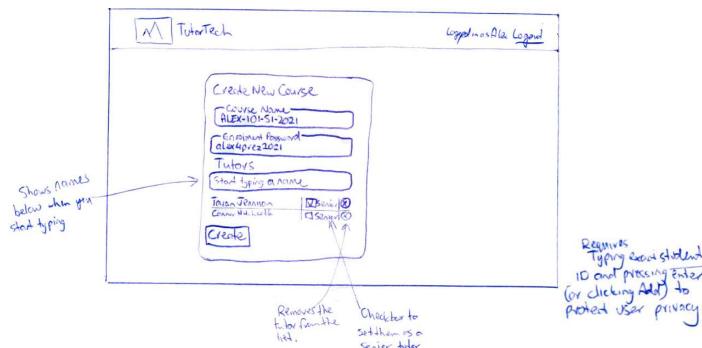


Figure 32: Alex's Course Creation Panel Sketch 1

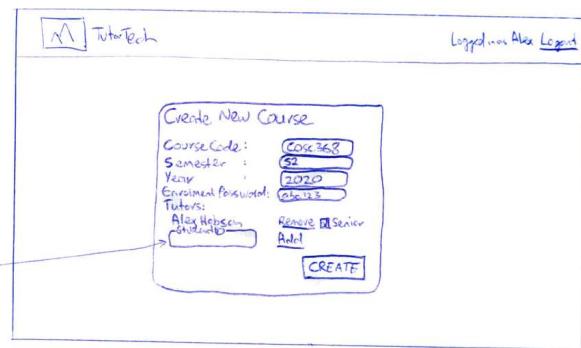


Figure 33: Alex's Course Creation Panel Sketch 2

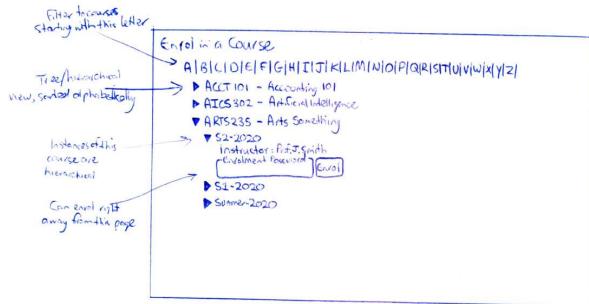


Figure 34: Alex's Enrolment Panel Sketch 1

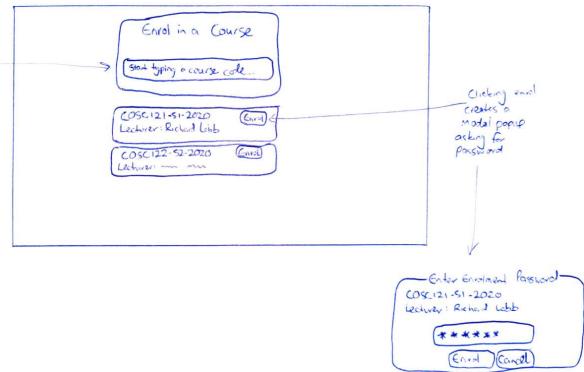


Figure 35: Alex's Enrolment Panel Sketch 2

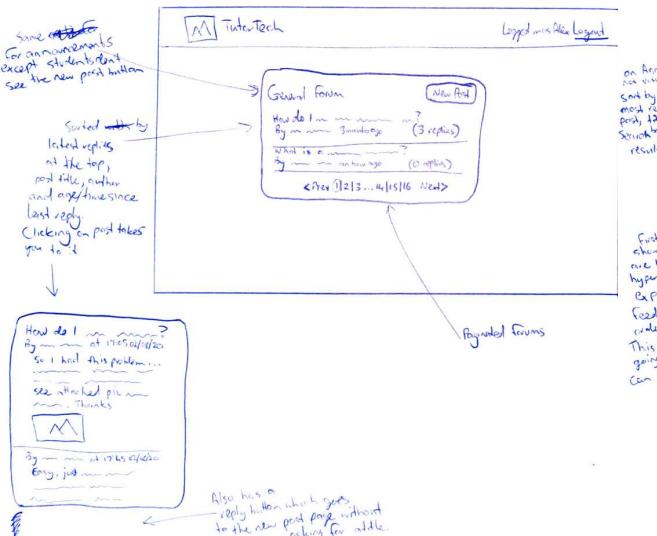


Figure 36: Alex's Forum Panel Sketch 1

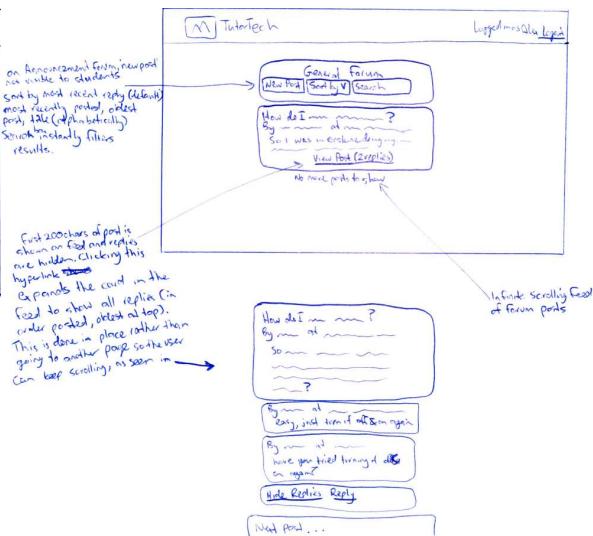


Figure 37: Alex's Forum Panel Sketch 2

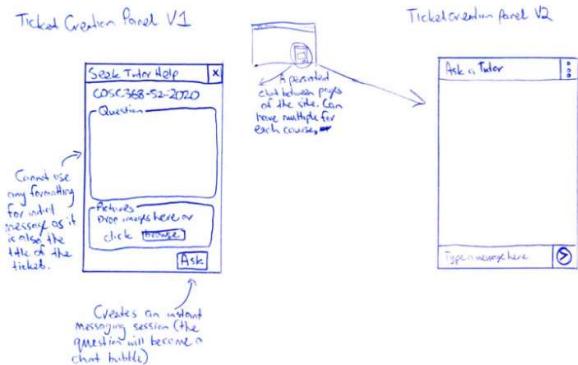


Figure 38: Alex's Ticket Creation Panel Sketch

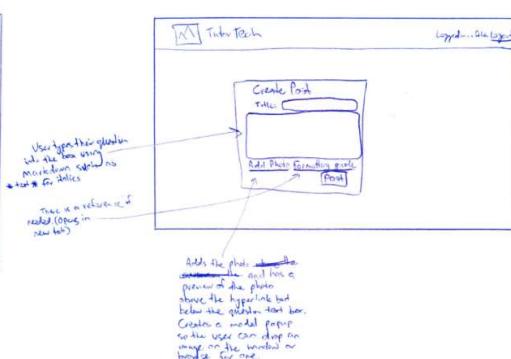


Figure 39: Alex's New Forum Post Panel Sketch 1

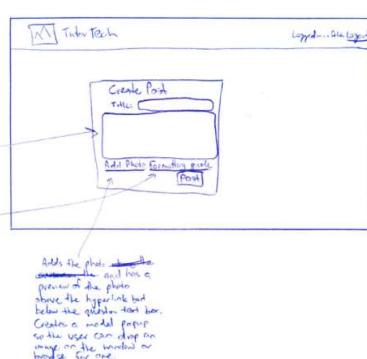
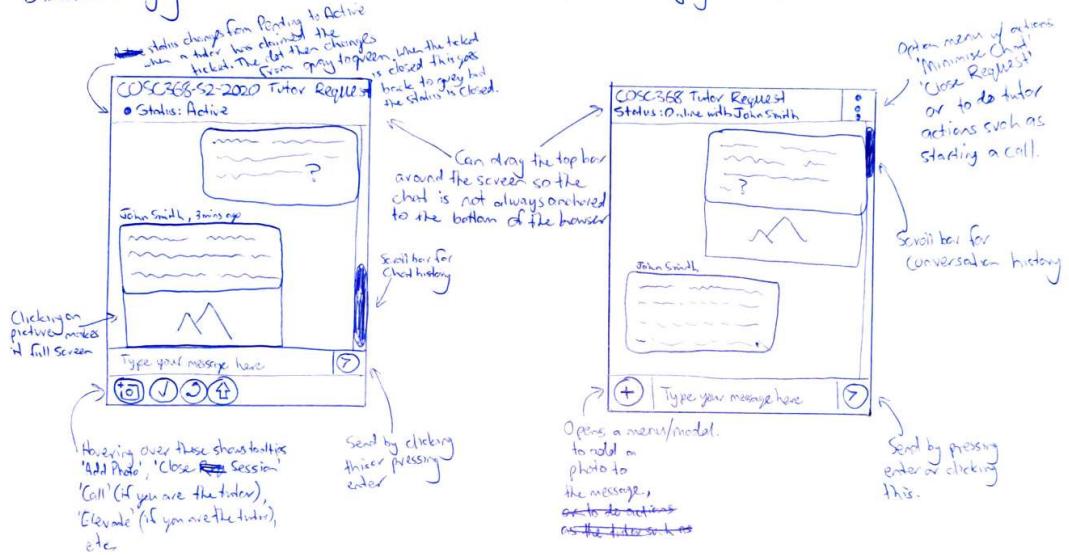


Figure 40: Alex's New Forum Post Panel Sketch 2

Direct Messaging Interface V1



Direct Messaging Interface V2

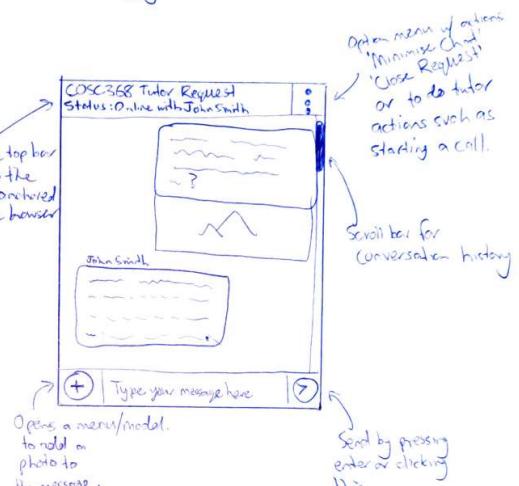


Figure 41: Alex's Direct Messaging Interface Panel (Ticket Chat) Sketch

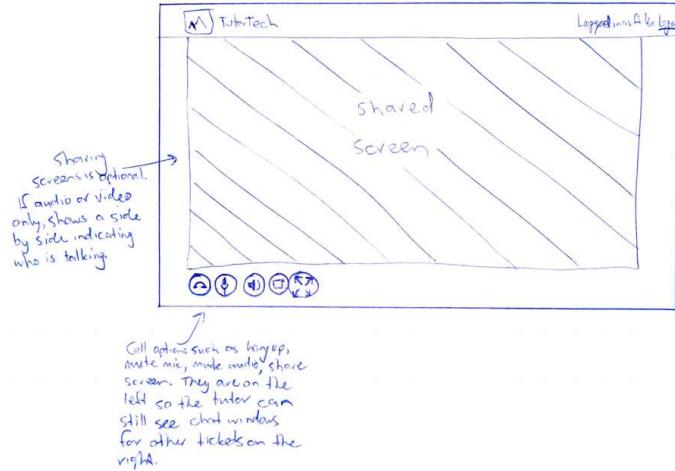


Figure 42: Alex's Call and Screenshare Panel Sketch 1

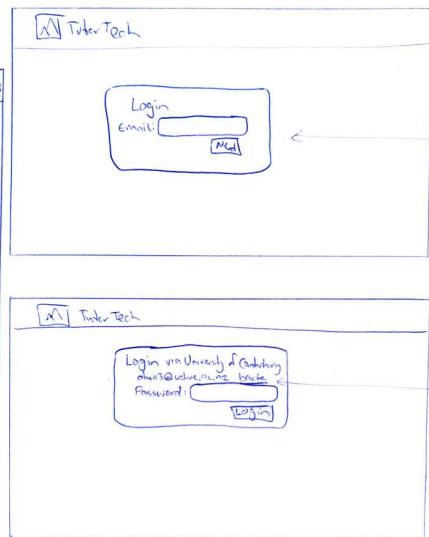


Figure 43: Alex's Login Panel Sketch

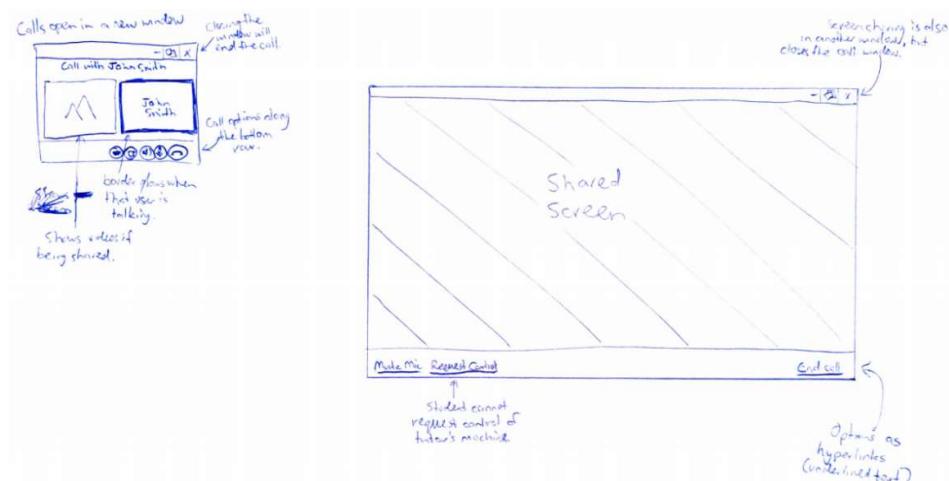


Figure 44: Alex's Call and Screenshare Panel Sketch 2

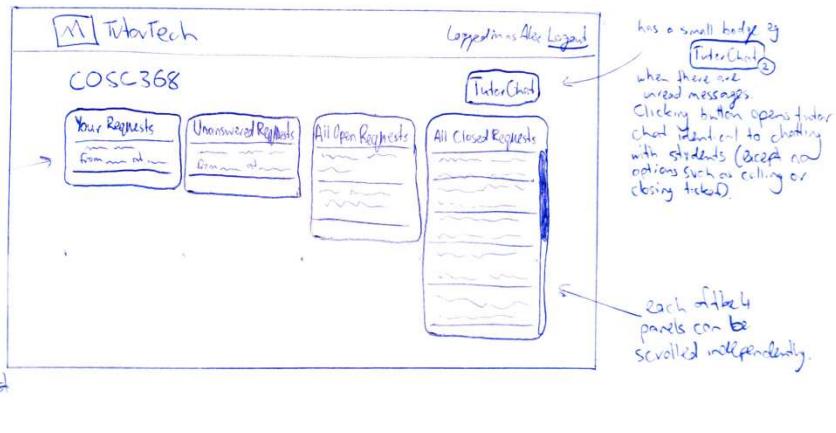


Figure 45: Alex's Ticket Queue Panel Sketch 1

Queue Panel V2

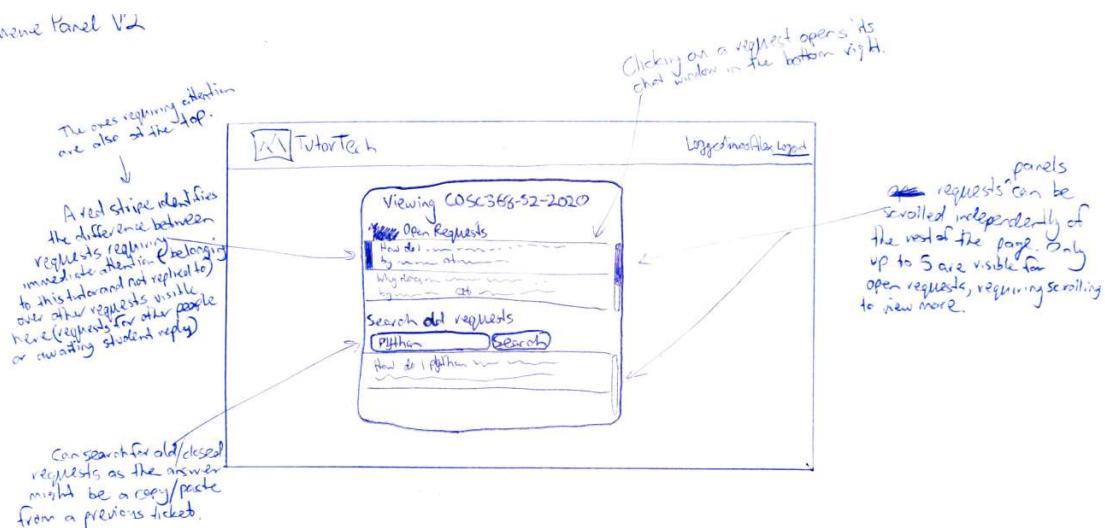


Figure 46: Alex's Ticket Queue Panel Sketch 2

Homepage Sketch 1 (Figure 28): The minimalist homepage has up to three panels based on whether you tutor any classes or have the instructor role within your university. The three panels “Enrolled Courses”, “Courses you Tutor” and “Courses you Instruct” appear or disappear based on these conditions. This keeps the information on the homepage to a minimum, however this means the majority of the people (students) only see one panel in the centre, which leaves a lot of whitespace. Pros: Shows all of the courses in separate panels making it clear which courses you are a tutor, student or instructor in. Cons: Either there will be empty cards when a person is not a tutor in any course, instructor in a course or enrolled in one; or a lot of whitespace around it.

Homepage Sketch 2 (Figure 29): This homepage shows your enrolled courses in a panel on the left side, where each of the three sections “Enrolled Courses”, “Courses you Tutor” and “Courses you Instruct” can be collapsed. ‘Enrolled courses’ is expanded by default since the majority of the users (students) will only be interested in this section. The rest of the space on the right represents a dashboard which shows aggregated tutor requests from all your enrolled classes (if any), with a button to visit this request. Pros: Courses section is only as wide as it needs to be, meaning lots of room for other information. Also, a quick interface to check on requests for all classes. Cons: Showing more than the minimum necessary amount of information in order to navigate to a class, especially considering that a student normally only studies for one class at a time.

Course Dashboard Sketch 1 (Figure 30): The course dashboard pictured above shows a summary of the three most recent posts in the announcements and general forum. ‘Tutor requests’ is also depicted as a forum in order to make the user interface more consistent between tutor requests and the forums. On the right is a panel that can be used to navigate between courses, however the final design may have a permanent sidebar on the left with courses. In this case, the right sidebar would be removed. Pros: Showing a summary of the three most recent forum posts in each forum means the user interface doesn’t get too crowded. There is quick access to create a new post without

having to visit the forum. Tutor requests is treated as another forum, so it's easy to switch between using the two without having to learn how to use differing user interfaces. Cons: Treating messaging a tutor in real time as a forum post gives a sense of permanence which may discourage a student from seeking help.

Course Dashboard Sketch 2 (Figure 31): In this course dashboard, a minimal amount of information is shown in order for a user to navigate the site without being overwhelmed by many buttons and widgets. This makes the site easier to navigate but gives it a 'installation wizard' feel where there may be many clicks necessary in order to do a simple task. The button "Seek Tutor Help" changes to "Answer Student Queries" if you are a tutor or instructor in the course instead, which will navigate to the dashboard for answering queries instead of allowing the user to submit a help request. Pros: Only showing the minimal amount of information necessary to complete a task. Cons: Adds a lot of extra navigation steps. Lots of whitespace since there are only four buttons on the whole page.

Course Creation Sketch 1 (Figure 32): This course creation panel allows the instructor to assign tutors from creation. By typing the name, a popup shows people matching that name allowing the course creator to assign tutors with instant feedback. Once adding a user, there is a checkbox to toggle senior tutor status, and a delete button in case there was an incorrect selection. Pros: Shows only the information necessary. Typing a name might be a privacy risk as you can search for all students by name. Might not be clear that a course does not require an enrolment password. Cannot differentiate users with same first and last name.

Course Creation Sketch 2 (Figure 33): This course creation panel has three separate fields for a course title: course code, semester and year. This makes it easier for other parts of the UI to differentiate different occurrences of a course. Tutors are entered in by their student ID, which eliminates any ambiguity by people with the same first and last name, while also respecting privacy as the search box cannot be used to lookup every student's name at a given university. Pros: Privacy is enhanced by requiring a tutor to enter the student ID in. Splitting the course title into course code, semester and year makes it easier for sorting in other UI components. Cons: Instructor might not have quick access to student IDs of tutors and have to use University of Canterbury's multi-million-dollar enrolment application.

Enrolment Sketch 1 (Figure 34): This design for the enrolment panel is navigable without the keyboard up until the student needs to enter the enrolment password. By filtering by letter, then by paper, then by occurrence, enrolling in a course should take exactly 4 clicks. Pros: Quick effective way of sorting courses without having to reach for the keyboard. The user interface is not cluttered since most information is hidden until you expand a course. Cons: Adds many steps and lots of visual searching to navigation which could be solved by typing a few letters.

Enrolment Sketch 2 (Figure 35): This design for the enrolment page has a search box which instantly filters results. The results have one button which is to enrol, which creates a modal asking for the password and a button to enrol or cancel. Pros: Has the efficiency of typing a course code plus the minimal user interface of just having a single button to enrol, triggering a modal asking for the password (if necessary).

Forum Sketch 1 (Figure 36): The forum is paginated and has one button: new post. By clicking on a post, it will take you to a page with the full forum post (as pictured above) and all the replies. The cursor will turn to a hand indicating that you can click on it when you hover over a post. The number of replies a post has is also visible on the general forum, along with the time since the most recent reply. Cons: Unclear how to get back to course dashboard as there is a lack of button to get back. It may be unclear that the user has to use browser navigation (the back button).

Forum Sketch 2 (Figure 37): This forum page treats the forum as a feed with infinite scrolling. By clicking on a post, replies are loaded in-line below the post, indicated by being narrower than the post. Replies to replies are also narrower than the reply, forming a hierarchy. Pros: Being able to customise sort type in ways such as latest or oldest replies first, posts with latest replies, etc. AND having a search box; means there are advanced tools for someone who knows what post they are looking for. Cons: Infinite scrolling makes it difficult to go back and forth between posts.

Ticket Creation Sketch 1 (Figure 38): A small overlay for a chat appears which lets you type the initial question and add images. The length of the first question is capped as this forms the title that tutors see, however after clicking Ask the student can type further messages before the request has been attended by a tutor. The overlay which is

common between these 2 designs is inspired by live chats on websites, which often occupy the bottom right corner of the screen and are positioned on top of the rest of the page. Pros: Clear how to add a picture.

Ticket Creation Sketch 2 (Figure 38): Version 2 of the ticket creation panel also sits in the bottom right over the rest of the page. Instead of having a separate interface for creating the initial message, the student is thrown straight into a chat window where they can ask a question. This makes a seamless experience that may take some of the anxiety away of having to phrase a question just right before submitting it to a tutor. Pros: Straight to the point that you are entering a chat with a tutor.

New Forum Post Sketch 1 (Figure 39): This create forum post page has a rich text editor, a place to upload images (inspired by Learn/Moodle) and a title of a post. The cancel button takes the user back to the forum, where tutors may go back to the announcements forum if they were attempting to post in announcements for example. Cons: Advanced formatting such as headers are probably not needed for posting a question, which clutters the UI with lots of buttons that would not be used by most users.

New Forum Post Sketch 2 (Figure 40): Minimalistic create forum post page with only two elements: title and post body. Adding a photo requires clicking the hyperlink which opens a modal allowing the user to drop an image into the browser window, or to click browse to open their file explorer. Advanced users can learn how to use basic formatting with markdown. Markdown is quite commonly used by programmers so would be familiar to a subset of users who are more likely to need some of the advanced features (such as code syntax highlighting), however clicking "Formatting guide" would open a guide in a new tab. Pros: There is some basic formatting such as syntax highlighting for code for advanced users who know markdown. The same users who know markdown are the people who would make use of this feature since it's common in other applications such as GitLab and discord. Cons: Title label looks out of place being the only text box that needs a label. Lots of whitespace.

Direct Messaging Interface or Ticket Chat Sketch 1 (Figure 41): On top of your average instant messaging platform, this direct messaging interface shows the status of the request, as in, it shows whether the request is being actively attended by a tutor. There are buttons for tutor actions such as starting a call or elevating the request to a senior tutor, which will be invisible to students. This means students will only see the add photo button and a button to close the request if the student believes the request has been properly answered. Pros: Shows a colour indication of the status of the ticket. Cons: Takes up a whole row under the input box, where students would only see two icons.

Direct Messaging Interface or Ticket Chat Sketch 2 (Figure 42): This version of the direct messaging interface hides operations such as closing the request or elevating the request (as a tutor) behind a menu with three dots in the top right of the panel. This means the panel has more space to show more of the chat history without bombarding the user with many icons. This also has the advantage of being able to show text for each menu item rather than an icon, which gives more information about the action the user is about to do. Pros: Clear user interface with few buttons to click, which means more space for the chat which is the most relevant part.

Call and Screenshare Sketch 1 (Figure 42): The call/screenshare takes up the full window. Call options are positioned on the left so when a tutor has several chats open, they do not obstruct the buttons. The advantage of making the call take up the full screen is that small text is easily readable when screen sharing, however not all calls are screen sharing ones. In this case, your own video and the other party's video are shown side-by-side at equal sizes. And in the event that a user does not have their video on, their name is shown, similar to in zoom.

Call and Screenshare Sketch 2 (Figure 44): In this version of the call/screensharing interface, a call opens in a new browser window which is sized small by default. This makes it easy to move the call's interface around, so it isn't obstructing other chat windows, or so it is visible while screensharing (while hiding sensitive details about other user's chats if the tutor is the one sharing their screen). Pros: Resizable new window makes it easy to read text when someone is screensharing. Cons: Window might get minimised which means there's little indication that your microphone is switched on and actively in a call.

Login Sketch (Figure 43): This login page shows a two-step process where a user initially enters their tertiary provider's email address. This means that universities requiring a user sign in with their single sign-on can be accommodated for, and would eliminate the need for a forgot password option since this would be handled by the university (and realistically, this should be handled by the university too if our website uses their credentials). Pros:

Asking for email address means the name of the university can be obtained for stage 2, which is to redirect to the university's single sign-on page. University's sign-on page is unlikely to have a similar design.

Ticket Queue Sketch (Figure 45): The interface for the tutor to accept and answer requests is visible above. This shows four panels with different categories of requests, where more important ones to a tutor would generally be found on the left. Requests with unread/unanswered messages are highlighted so the tutor knows which ones need responding to. "Your Requests" shows open/active requests belonging to the tutor who is signed in. "Unanswered Requests" shows requests that have not been claimed by a tutor. "All Open Requests" shows open requests that are being answered by other tutors, and "All Closed Requests" shows previous requests. This is useful for seeing answers to similar questions that have been answered in the past. Finally, there is a tutor chat button which has a badge showing the number of unread messages. This button is always visible while viewing a course's page as the tutor chat is designed for tutors to be able to communicate with each other. Pros: Categorised in priority order of the four categories. Cons: Easy to ignore tutor chat if your attention is always on the left side. Quarter width of the screen might not be enough to read the description of a request effectively.

Ticket Queue Sketch (Figure 46): This arrangement of tutor requests shows all open requests at the top, and all closed ones at the bottom. A stripe at the left of open requests gives colour coding to requests in cases such as ones belonging to the tutor or unclaimed requests. Requests belonging to a tutor that have an unread message are also a different colour (the brightest, indicating the highest priority). Old requests can be searched to efficiently find old questions about a given topic. Pros: Colour coding makes it obvious at a glance which requests need attention. Searching through old requests makes it easy to copy/paste responses from old requests. Cons: Not making very good use of horizontal space, while there are vertical scroll bars for sections. Tutor chat is just treated as another chat in open requests with a different colour, making it easy to get lost.

Primary Preliminary Design

Course Dashboard (Student)

The preliminary design in figure 47 shows the student view of the course dashboard for a class. The modular design allows the size of the help and forum panel to change by dragging the slider in the middle left or right. On the left is the panel for asking help from a tutor and the right is the forum. At the top of the page are options to toggle the visibility of the ‘Help’ and ‘Forum’ sections. This causes the panel to collapse so the other one takes up the full width. It is not possible to hide both help and forum at the same time.

Forum

The forum (pictured in figure 47) highlights announcements with a light blue background so it is clear which posts are announcements. Users may also filter to only see announcements with the “Hide general posts” checkbox to the right of the search bar.

The forum is paginated with the page numbers at the top of the screen. There is also a search (which updates after pressing enter or clicking the magnifying glass icon), allowing users to efficiently find posts they are looking for.

The view button on each of the forum posts pictured will navigate the forum panel to that forum post. This will show all replies to the post and allow the user to reply on posts that aren’t announcements. See the view forum post section for more information.

Ask a Tutor

The ‘ask a tutor’ panel is visible for students on the course dashboard (figure 47) by default when they navigate to a course’s dashboard. This encourages a user to ask for help while they are on the course dashboard as the input field is visible without having to click anything.

The initial message for a ticket is limited to 200 words as this becomes the title of the ticket that a tutor sees (and as part of the previous tutor queries below the ask a tutor box). Once the student sends the initial question, the panel becomes an instant messaging chat where the student can give more information or send images to further clarify their question, even before it has been answered by a tutor. See ticket chat (student) for more information.

There is a radio button to mark a ticket as ‘question’ or ‘assistance’. ‘Question’ is intended for a request that has a definite answer without requiring further explanation, whereas assistance requires explanation of theory related to the problem. Once the ticket has been created, a tutor can change its status from question to assistance in case the situation changes (such as a string of questions eventually leading to the need to explain some of the theory). The intention behind this feature is that tutors can answer multiple questions in parallel while offering assistance to one person. The system does not enforce this, rather it serves as an indicator for tutors on the complexity of the ticket (however it is perfectly acceptable for a tutor to attend to two ‘assistance’ tickets in parallel).

Previous Tutor Queries

There is also a section labelled “Previous Tutor Queries” within this section. This is a reference for students to see chat logs for previous tickets they created. Unlike tutors, students can only see their own closed tickets in order to respect the privacy of other students. Clicking on “View query” by each closed ticket will show the conversation the same as an active ticket chat (as a student), however the box to type a reply is replaced with a link to reopen the ticket (unless the student already has an active ticket).

The screenshot shows the COSC368 course dashboard. On the left, the 'Ask a Tutor' section has a text input field containing 'I'm stuck on lab 9 So far I tri'. Below it are two radio buttons: 'Question' (selected) and 'Assistance'. A 'Submit' button is at the bottom right. In the center, the 'Forum' section displays a navigation bar with arrows and page numbers (1, 2, 3, ..., 7, 8, 9, >). A search bar and a 'New Post' button are also present. Below the navigation is an 'ANNOUNCEMENT' about assignment cancellation. The main content area shows a post from 'Andrew Cockburn' about high-fidelity designs, with a 'View' button. Another post from 'Alex Hobson' is also visible.

Figure 47: Student view showing course dashboard containing ‘Forum’ and ‘Ask a tutor’ sections

View Forum Post

The right-hand panel of figure 48 shows the preliminary concept for viewing and replying to a forum thread. This concept takes the ideas of indenting replies from one concept and matches it with the dedicated reply button from another. This gives the user a clean visual indication of each reply on a post and allows them to reply directly to other replies. Opting to a post or reply opens a footer at the bottom of the page to allow the user to reply whilst still being able to view the details of the thread.

This screenshot shows the same dashboard as Figure 47, but the right-hand panel is focused on a specific forum post. The post is titled 'How to make high fidelity designs for assignment 1?' and was asked by 'Alex Hobson' on 11/09/2020. The post content is: 'How to make high fidelity designs for assignment 1? Hi everyone, I'm trying to make high fidelity designs for assignment 1 but I'm not sure what tools are available to create them with A recommendation would be sweet.' Below the post, a reply is being composed by 'Connor Hitchcock' on 11/09/2020: 'I'm doing mine with pencil and paper (and a ruler) It makes it so easy to annotate things since all you gotta do is draw an arrow and write something.' At the bottom of the reply, there are 'Delete' and 'Reply' buttons. A new reply is being typed by 'John Smith (Tutor)' on 12/09/2020: 'I strongly recommend you make high fidelity sketches on the computer. Last year for my assignment my team used mockups. We ran into an issue where we couldn't share it with the group members without paying for the premium package, so I asked the university to provide a license for mockups since it's course-related. Email me at smj345@ucilive.ac.nz with your google account email address and I will add you to the university license'.

Figure 48: Viewing a forum post. The logged in user is also replying to one of the posts on the forum.

Tutor View (Tutor)

The high-fidelity preliminary design for the tutor view (figure 49) shows what a tutor will see when they visit a course they tutor from the homepage. When a tutor visits a course, they will see the forum identical to how a student sees it. Tutors are able to moderate the forum; however the tutor would have to be viewing a forum post for these options to be available. The chat view can be resized vertically for all chats, allowing the tutor to adjust the screen when necessary.

Help Queue

The use of a small colour band on the left of the ticket in the help queue (figure 49) indicates the status of the ticket. All actions other than accepting the ticket have been removed from the ticket card to reduce UI clutter. Instead they are located within the ticket chat's options menu (the 3 dots in each chat). Clicking on any item in the queue in any location other than the accept button will add it to the chat pane on the bottom of the screen without accepting it. This means a tutor can see all active tickets in real time, even when assigned to other tutors. The queue history can be accessed via a button at the top right of the queue panel, which toggles between the live queue and a historic view of closed tickets. This allows a tutor to view previous threads with other students. The ticket preview contains an indication of the type of ticket, as well as an active timer for how long the ticket has been alive (or how long it has been unclaimed for).

The screenshot displays the 'COSC368: Humans and Computers' course page. At the top, there are tabs for 'Chat', 'Queue', and 'Forum'. The 'Queue' tab is selected, showing a 'Help Queue' panel on the left and a 'Forum' panel on the right. The 'Help Queue' panel lists five student requests:

- Does question 6 require working? (Request by Thelittlewiseowl, Type: Question, Time active: 3:37)
- Please help me, my code is broken for question 10 (Request by Mawntee, Type: Assistance, Wait time: 4:35)
- Can you please clarify the requirements for section 3 of the assignment (Request by Revelate_Z, Type: Assistance, Wait time: 5:48)
- What is the airspeed velocity of an unladen swallow?

Each request card has an 'Accept' button. The 'Forum' panel shows an announcement about assignment cancellation and several student messages. A sidebar on the left indicates 'No Chat Open' and provides options for managing offline users. The bottom of the screen features a large chat pane with multiple conversations and a message input field.

Figure 49: Tutor view showing help queue and chat view in addition to the forum

Tutor Chat

In Figures 49 and 50, the bottom half of the screen is occupied by the chat pane. This pane is only visible to tutors and instructors, as students can only have one open chat for one class, which occupies the left pane.

There are several chat slots within this pane, where the number dynamically changes based on the screen resolution, in order to maintain readability. The basic functionality of a chat is to function as instant messaging between a student and tutor. The layout is inspired from already existing instant messaging applications, so the majority of users will be familiar with basic usage around this feature.

In addition to instant messaging functionality, the tutor can send images to the student by clicking the plus button to the left of the input field for typing the message. This will open the built-in file browser that every web browser has, allowing the user to choose a picture from their files to be sent to the student. A preview of the image is shown and is only sent when the tutor sends the message they were already typing. This is to prevent accidentally sending the

wrong image, and means that an image is sent along with a message (see the far-right chat in figure 49 with Alex Hobson for an example of how messages with images appear).

All chats have a status indicator, with a small dot of colour indicating whether the other party is online or not. This also gives a short bit of info, as pictured in figure 50, with messages such as “Status: Closed” or “Status: In a call with Alex Hobson.”

Collapsed Chats

On the far right of the chat pane are bubbles representing all collapsed chats. This allows for a tutor to have more chats open than their screen would allow for, and instead have them collapsed on the side (so they can be quickly revisited). Clicking on one of the bubbles opens the chat in the first available slot (starting from the right side). In order to hide one of the already open chats (such as to free that slot up to open another chat), the tutor can click the three dots in the top right of the individual chat and click ‘Hide Chat’ as one of the options visible in figure 49. The blue “TC” on the collapsed chats on the far right represents the tutor chat. The tutor chat is a chat that only tutors can see, which has a bubble visually different from chats with students and is always at the top. The tutor-only chat cannot be closed, marked as guidance, host a call or be elevated to a senior tutor; so a tutor would only see the options to hide the chat. The bubble for the tutor chat will always be visible at the top of the collapsed chats section as this is a permanent chat between tutors. A badge showing how many unread messages in each collapsed chat is visible, indicating to the tutor which chats require attention (including the tutor chat).

Student/Tutor Call

In Figure 50, the top left panel shows the preliminary concept for the call system. This call panel is the same for both tutors and students, but is shown here from the tutor's point of view. The initial concept included a resolve button instead of an end call button, which was changed in the final design as the terminology is clearer.

In each call there is a thumbnail for each user as well as an additional thumbnail for any video feeds in the call (screen shares or webcam). Clicking a thumbnail in the call will maximize the thumbnail to the whole panel and hide the others. The top left of the call panel includes a selection of toggle buttons for muting yourself, screen sharing and video, which can all be toggled during the call. In the bottom right of the call panel there is a button for full screening the call panel, and a button for popping out the call panel into another dedicated window. These two options allow the user to view any video feeds it contains in greater detail.

Figure 50: Tutor view of a course, showing an in-progress call (occupying the help queue) and the chat panel

Ticket chat (student)

The student's ticket chat is refined to be clean and cohesive. Because the user is locked into the ticket until it is closed, the user can only send messages or resolve the ticket (with the close ticket button shown in Figure 51).

The screenshot shows a split-screen interface. On the left, a 'Help chat' window is open under the course title 'COSC368: Humans and Computers'. It displays a conversation between a student ('Wizzled__') and an administrator ('Alex Hobson'). The student asks about a quadratic denominator, and the administrator responds that they got help from a friend. A message box at the bottom indicates that 'Wizzled__ is typing...'. On the right, a 'Forum' section is visible, showing a list of posts. The first post is an announcement about the cancellation of assignment 1, followed by a question about high-fidelity designs.

Figure 51: Student view of ticket chat

Course creation/editing

The preliminary design for course creation/editing (Figure 52) takes two different inspirations from the initial concept. The top half of the form splits the course information into course code/year for sorting purposes. The bottom half takes the auto suggest and chip concept from a different initial concept. This results in a clean but detailed form that is intuitive for instructors to complete.

The screenshot shows a split-screen interface. On the left, a 'Courses' dashboard lists several courses: 'COSC367 Artificial Intelligence', 'COSC368 Humans and Computers', 'SENG302 Software Engineering Group Project', and 'ENEL301 Engineering Management and Economics'. Each course entry has 'Edit', 'Enrollment', and 'Delete' buttons, along with 'View' and 'View as Student' links. On the right, a 'Course Creation' panel is open. It contains fields for 'Course code' (with a placeholder 'Course ID code'), 'Year' (placeholder 'Year of course'), 'Course Title' (placeholder 'Title of course'), and 'Password' (placeholder 'Enrollment password'). Below these are sections for 'Senior tutors' (listing 'Dr_Nryac' with an 'X' button) and 'Tutors' (listing 'Phoenixin' with an 'X' button, 'Noah Powers', and 'Noxn'). At the bottom are 'Cancel' and 'Create' buttons.

Figure 52: Course dashboard and course creation panel (instructor view)

New forum post

Carrying through the simplicity from the initial concepts, the final preliminary design for creating forum posts (pictured in Figure 53) uses a hyperlink for adding attachments. We have also made the choice to use markdown styling instead of having a rich text editor, as most posts will not need styling. This declutters the UI, while still supporting basic formatting for advanced users. A link to learn how to use this formatting is provided so people can learn.

The screenshot shows a split-screen interface. On the left, under 'Ask a Tutor', there is a text input field containing 'I'm stuck on lab 9 So far I tri' with a character count of 9/200 Words. Below it are two radio buttons: 'Question' (selected) and 'Assistance'. A 'Submit' button is at the bottom right. On the right, under 'New Forum Post', there is a 'Title:' field with 'What proportion of the test will be Prolog?' and a 'Post:' field containing 'I understand that we will be covering the past five weeks worth of lecture material, and that Prolog only covers two of them. However, Prolog seems to be a very important part of the project and I was wondering if that would be reflected in how much there will be covered in the test'. Below the post field are 'Add image' and 'Markdown guide' links. At the bottom are 'Delete' and 'Post' buttons. The top navigation bar includes 'Toggle visibility', 'Help', 'Forum', and user information 'Logged in as Alex Hobson Logout'.

Figure 53: New Forum Post page (student view)

Login

The primary preliminary concept for logging in has taken inspiration from the simplest of our initial concepts, where our system only takes the user's login email, then hands the full login authentication off the education provider's systems after using the email to identify which education provider the user belongs to. To counter the white space of a small text form on a full page, a neutral image is used in the background (as visible in Figure 54).

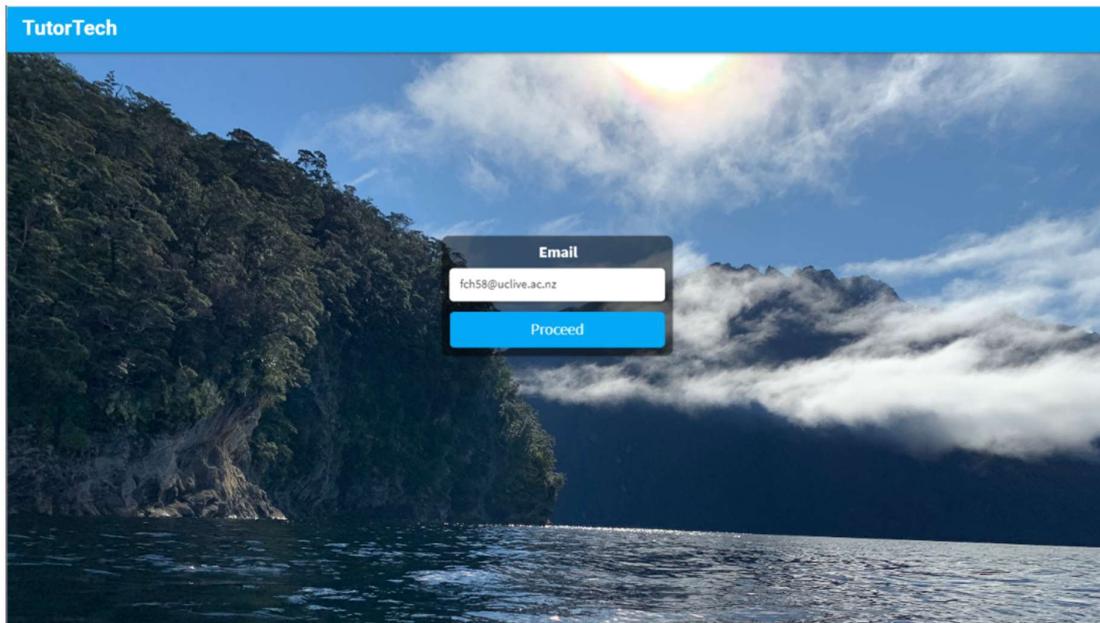


Figure 54: Login page

Homepage

The preliminary concept for the home page is based off the partnering design for course dashboard from the initial concepts, that utilize the half page split, with the modification of removing the hamburger menu sidebar in favour of making greater use of the header. This gives the system a consistent layout throughout, while providing a space for an event log. The event log can allow a user to have a quick overview of any information that may be relevant to them, which may be invaluable to tutors and help them ensure all students receive the help they need.

The courses panel provides the user with the ability to navigate through different categories of courses via buttons along the header. The panel will default to showing only the courses the user is enrolled in for a student, courses an instructor manages, or courses that a tutor is enrolled in or teaches. Any user can navigate between these categories with the buttons at the top of the page.

Search New Course My Courses My Enrollments All Courses Logged in as Connor Hitchcock Logout

Courses

COSC367 Artificial Intelligence Edit Enrollment Delete
View View as Student

COSC368 Humans and Computers Edit Enrollment Delete
View View as Student

SENG302 Software Engineering Group Project Edit Enrollment Delete
View View as Student

ENEL301 Engineering Management and Economics Edit Enrollment Delete
View View as Student

Event Log

COSC367 Created Forum Post 16:32 Fri Go to Event Dismiss

What proportion of the test will be prolog?
I understand that we will be covering the past five weeks worth of lecture material, and that Prolog only covers two of them. However, Prolog seems to be a very important part of the project and I was wondering if that would be reflected in how much there will be covered in the test.
By Connor Hitchcock at 16:32 11/09/2020

COSC368 Announcement 12:09 Thu Go to Event Dismiss

SENG302 New Reply 20:12 Wed Go to Event Dismiss

SENG302 Course Dropped 13:49 Wed Go to Event Dismiss

COSC362 Course Enrolled 08:42 Tue Go to Event Dismiss

Figure 55: Student & tutor view of homepage

Conclusion

The preliminary design comes together as a system based on single page design ideas where each course is fully handled on a single page. This is done through the subdivision of the page into different panels that can be swapped, changed and hidden without impacting the other components. This design provides a layout that enables and encourages multitasking throughout the interface as it allows students continued access to the forums while they have a ticket open. Tutors also have this ability, while also providing the ability to manage multiple tickets at once. These features combine to allow all users to make use of the full suite of features of the system at the same time, even while actively interacting with a ticket.

Contribution Statement

Alex Hobson 33⅓%

Connor Hitchcock 33⅓%

Taran Jennison 33⅓%