Assignment 2

300471606
Kamonchanok Suban Na Ayudtaya

React Native Reflection

React Native has been the framework I have used as this is my first experience in developing a mobile application. I found that React Native offers various benefits to developers. It is relatively concise, well documented and I found that the expanded functionality that React Native adds to the developer experience is well implemented. I think that this framework is best suited to those beginning their mobile development journey as there are a lot of components already created by Reat-native which reduces the difficulty of developing on the language and framework. Additionally, with emulator apps like Expo the process of setting up development is easy to implement and does not require the developer to have an in-depth understanding of ios or android setup requirements.

Fast Refresh: A comparable experience to mobile development usingReact Native I have had in the past is website development in HTML and CSS. HTML shares similar syntax and layout in terms of container use, components, navigation etc. which made the process of familiarisation much easier. However, a distinction I identified was that React Native did make iterating on designs a far more seamless experience than development with HTML, where each re-compilation takes more time. Compared to other languages or frameworks, with less developed ecosystems, React may significantly increase the pace of development as less tooling must be created from scratch.

Importing external components: In my application I made use of external components that would have consumed a significant amount of resources to create from scratch individually. For example "react-native-progress"
Import allowed me to implement an animated progress bar, "@react-native-community/segmented-control" allowed me to create a segmented value bar, and import DatePicker from "react-native-datepicker" allowed me to create a date input within a single component line. This allowed for development to proceed at a more efficient pace as I was able to reuse code that did not need to be custom to my application.
However, when I was experimenting with different imports I found that some introduced some complex issues with dependencies that were difficult to investigate due to limited documentation.
This resulted in restarting application development or removing unusable dependencies which was time consuming. Luckily expo made creating a new application very easy as well as ‘depcheck’ for removing used dependencies- another benefit of the react ecosystem.

Debugging: I found that debugging react was very difficult and time consuming - given that I was not able to get breakpoints to return any output I was limited to logging values to the console.
This may have been a result of my limited experience with the language and tooling; however, even with the help of expo, I felt like I was continuously debugging from error messages alone. When comparing this experience to other languages I have developed in I felt that React-Native was unintuitive and clucky in this aspect.
The current debugging system with expo runs on the computer and then sends the components to the phone or emulator. The latency from sending so much between the

devices makes the debugging process slower and I found that I often had to reload the app for expo to recognise certain fixes and changes.
In my opinion, adding the option to run the code on the target device when debugging and exposing the debugger from a WebSocket would result in a better debugging experience.

***Emulators**:* I felt that my development was limited to the expo go application due to my device storage limitations. Given that Apple's device emulator requires 40GB of free space to download and the android emulator also required additional space on top of this I was not able to use these emulators to observe my application's behaviour on other mobile specs.
This hindered the flexibility and usability of my application - I felt that the resource consumption of mobile development was a weakness in the react ecosystem and would have preferred if an online emulator was available for manual testing.

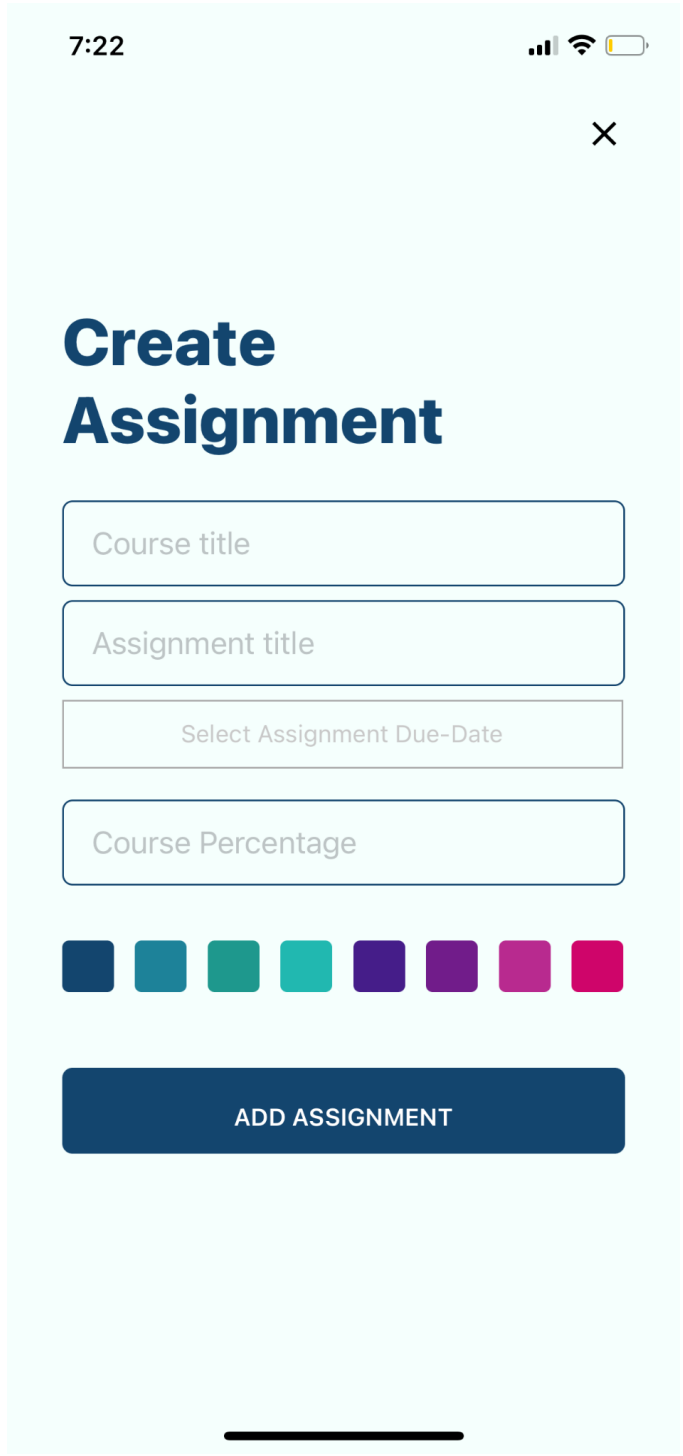
Component Position: React native does not provide the grid system found in React web nor the ability to view component outlines found in Figma. I felt that Layout options and visualisation, in general, was relatively limited within the framework. Given that layouts and grids are a fairly universally used attribute, the framework would be significantly improved by providing them by default. Additionally, it would be good to provide the option in Expo, if not the framework, to view component outlines as I spent a bit of time making the view component border visible to accurately position the content of the screen.
There are a few cases similar to this where seemingly fundamental or widely used features are not provided, but more complex functions are.

Components: I felt that the components imported from react native were easy to use and customisable. I liked how there were many ways to adjust the visualisations of each component and there was built-in support for animating gestures which would have been time consuming to program manually. For this reason, I felt that the framework was well equipped to assist developers who were starting out in mobile development as it removed the complexity of coding common components.

As the assignment progressed, I felt that my understanding of the language, framework and tools involved developed significantly.
Most problems I encountered had been documented online, and once I better understood the syntax and functions I was able to understand the solutions provided by others, the development process became much smoother and more efficient.
However, I still do not enjoy npm or yarn - in my experience I felt that other languages like python have achieved better results with a less complex packaging systems, i.e. Pip.
I found that the biggest weakness in the framework was its overreliance on packages due to the complexity of package management and instability of the application's that could be generated from a single dependency breaking.
Overall, I would use React-native again but I would be selective about its use as I felt that it was not extremely compatible with mobile applications that require a significant amount of external components but efficient for more simple applications.

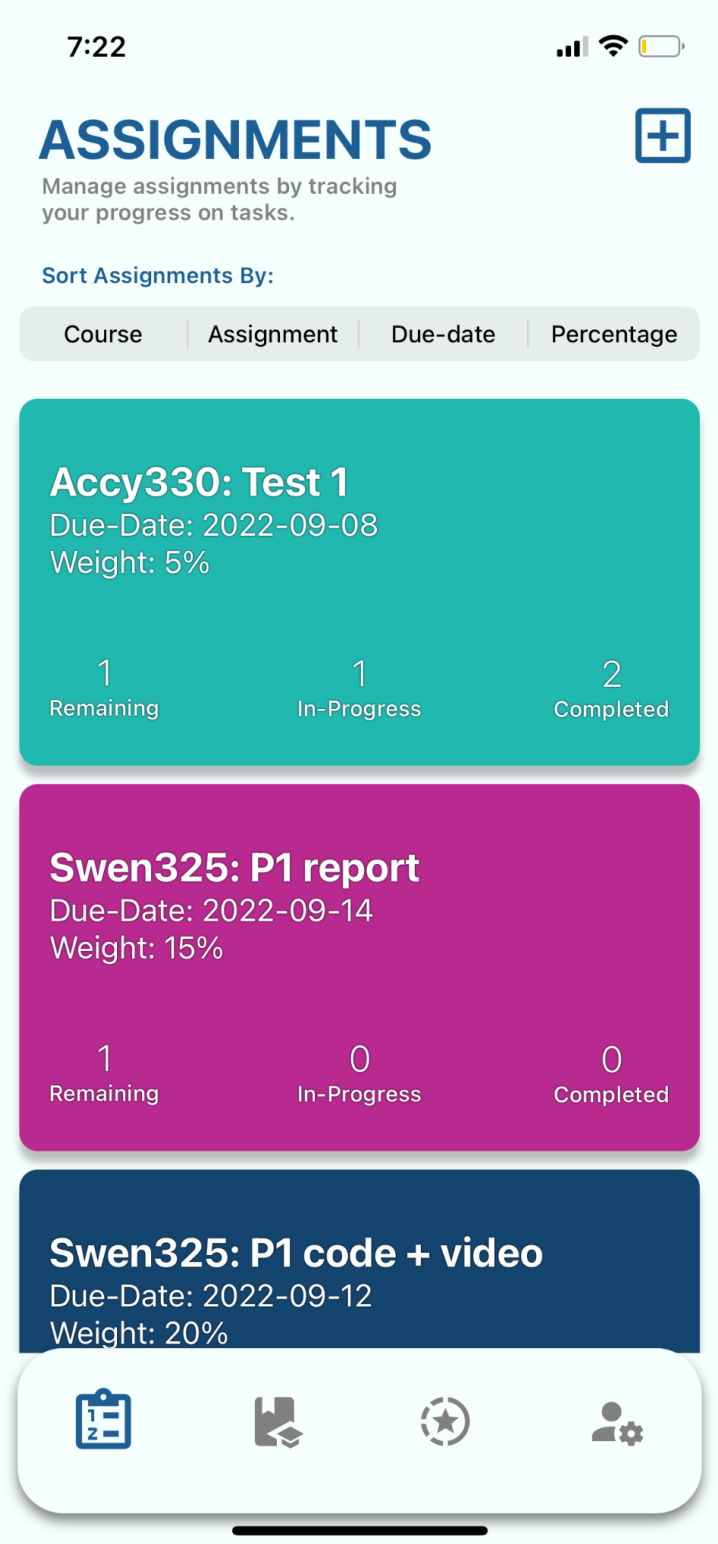
Appendix

Title: AddAssignmentModal
Function: User's are able to add assignments to their upcoming assignment list to manage their workload. Record upcoming assignment - users enter necessary fields to produce an assignment object that is converted to a document and added to the relevant collection under the user's ID in firestore.



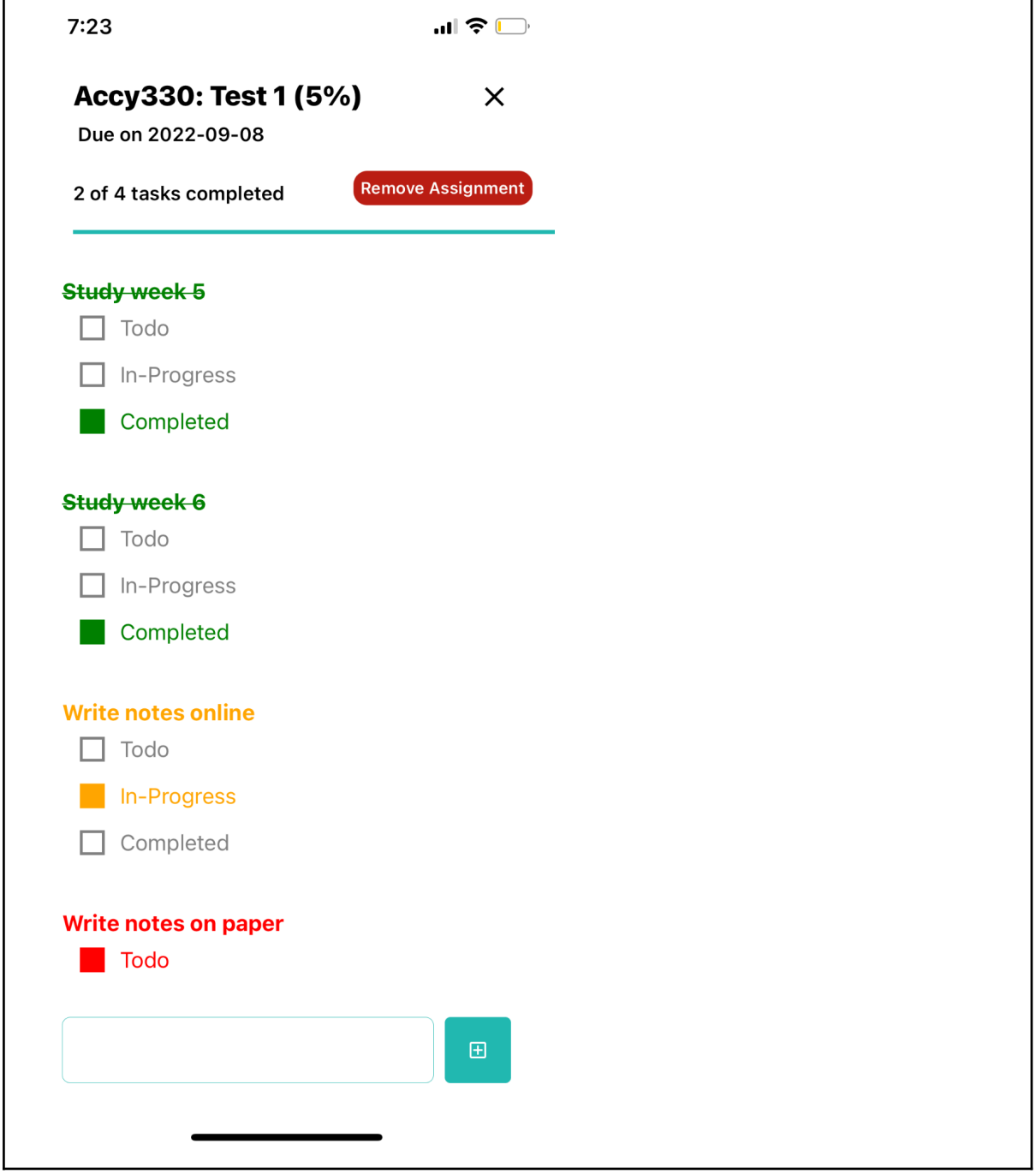
Title: MainAssignmentBoardScreen

Function:
Users are able to reorder assignments, view upcoming assignments, break assignments down into achievable tasks and view the summary of their progress.



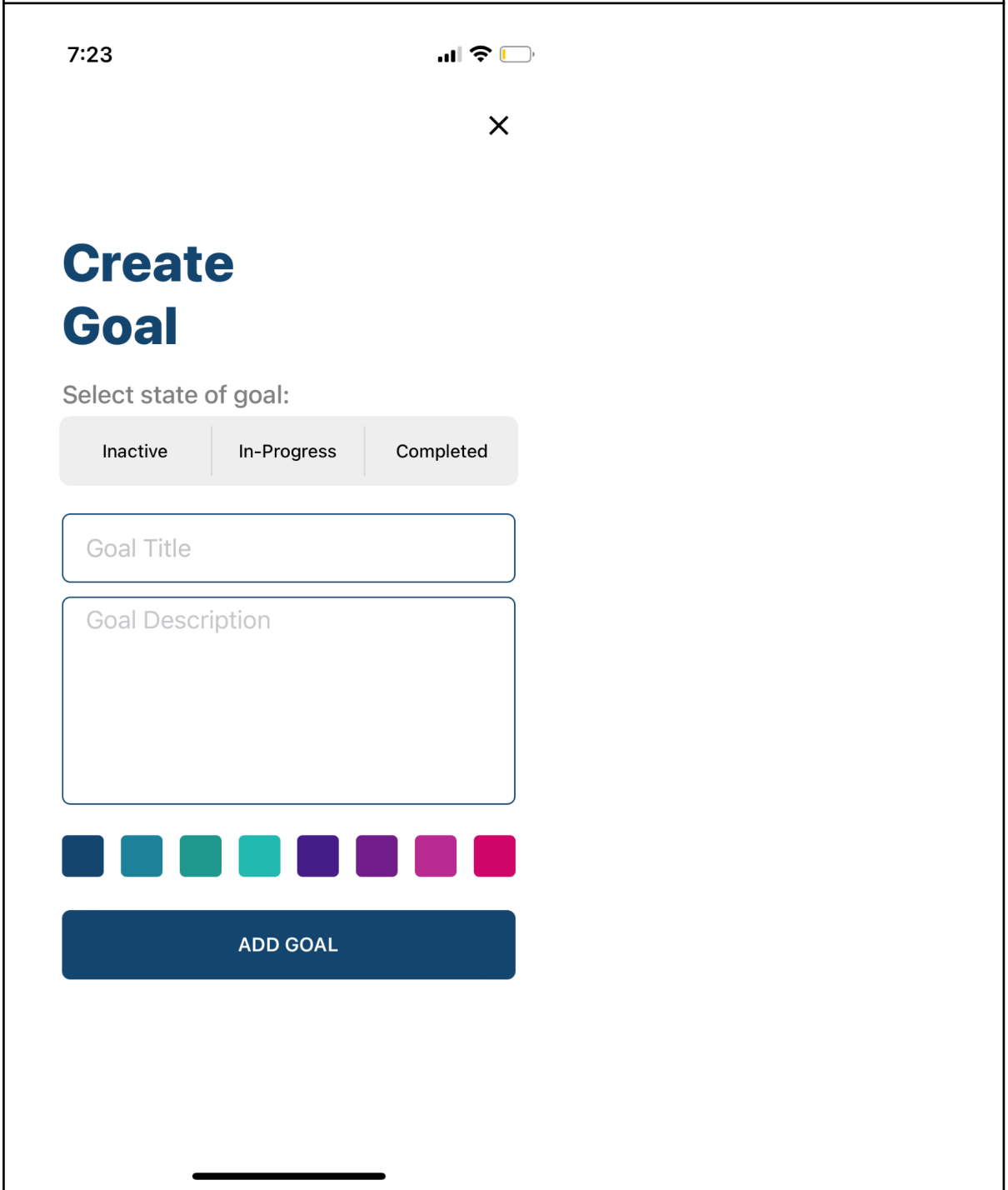
Title: TaskModal

Function:
Users are able to view existing tasks, add tasks, remove tasks, delete an assignment and change the state of a task within an assignment.
Add task - generates task object and add to assignment document in firestore
Toggle task state - update task object held in assignment document in firestore
Delete task - removes task object from assignment document in firestore
Delete Assignment - removes assignment document from the user's collection on firestore.



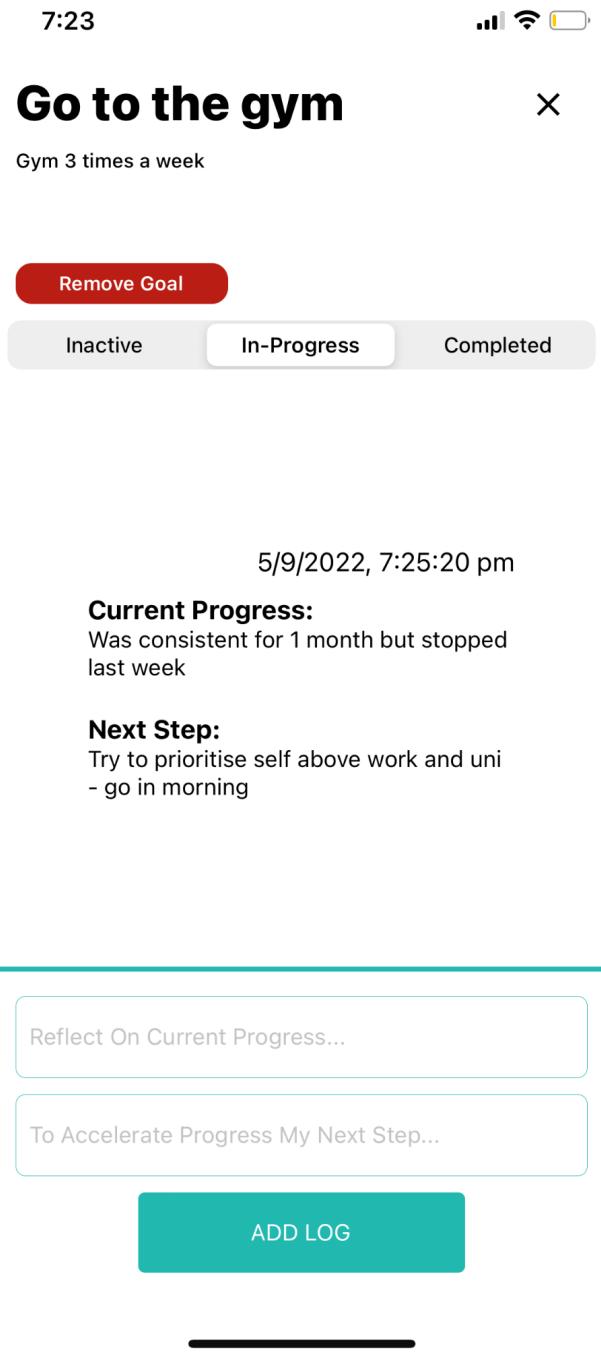
Title: AddGoalModal

Function:
User's are able to add goals to their goals list to document their engagement with their goals.
Record goals - users enter necessary fields to produce a goal object that is converted to a document and added to the relevant collection under the user's ID in firestore.



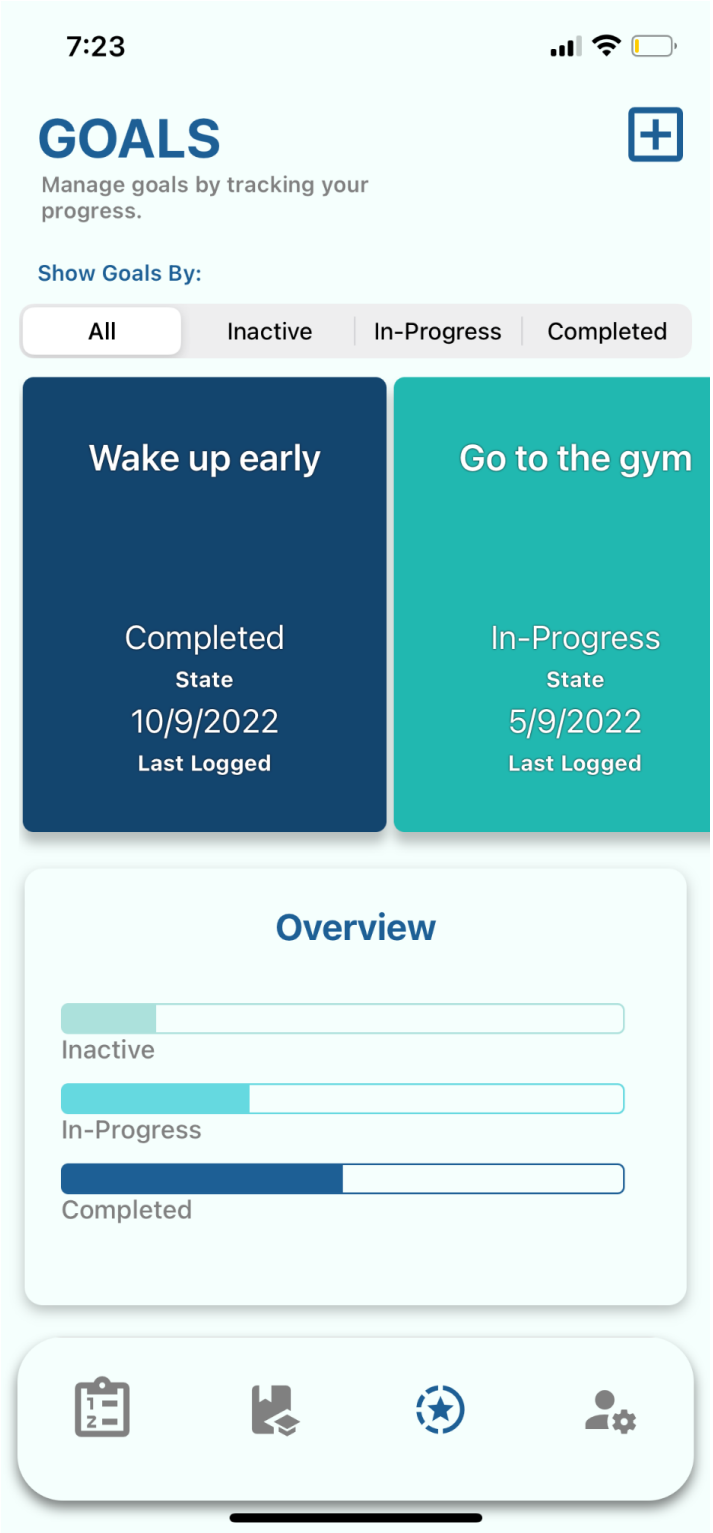
Title: LogModal

Function:
Users are able to view previous progress logs, add logs, remove logs, set goal state and delete a goal.
Add log - generates log object and add to goal document in firestore
Delete log - removes log object from goal document in firestore
Delete Goal - removes goal document from the user's collection on firestore.
Set Goal State - update goal document from the user's collection on firestore.



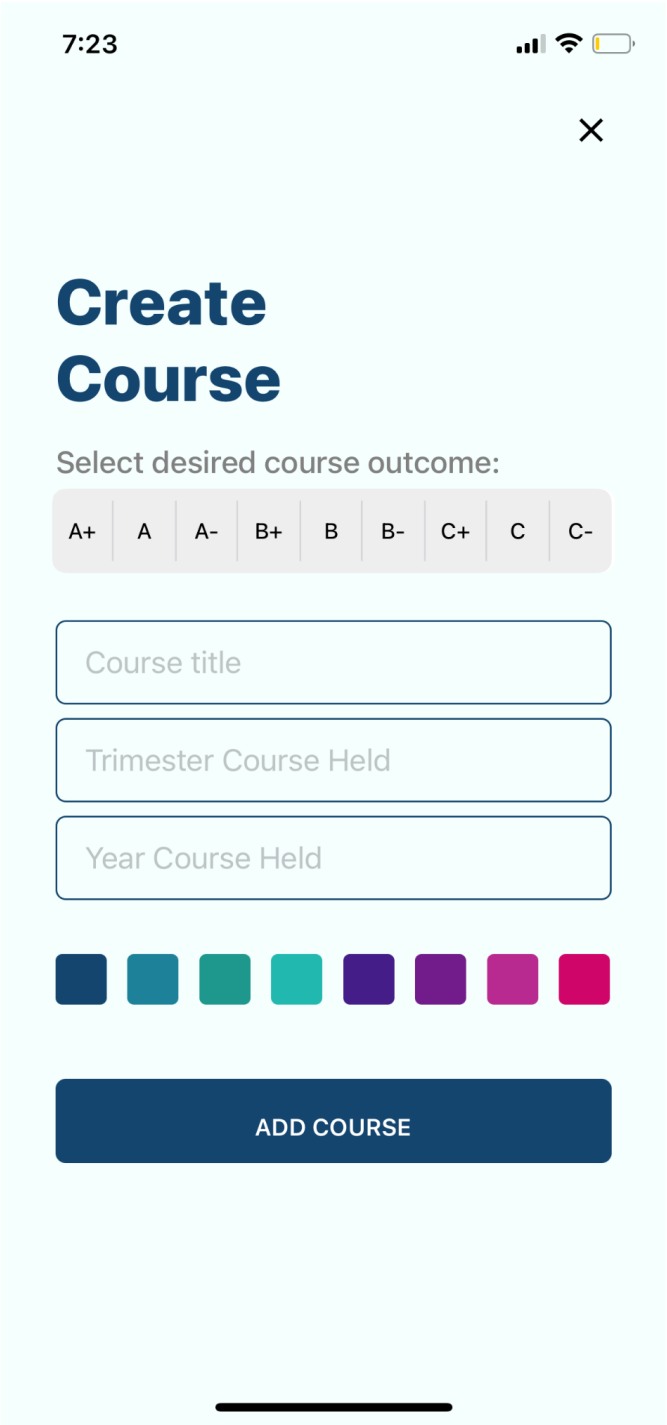
Title: MainGoalScreen

Function:
Users are able to filter goals, view goals, log their progress and view the summary of their engagement with each goal.



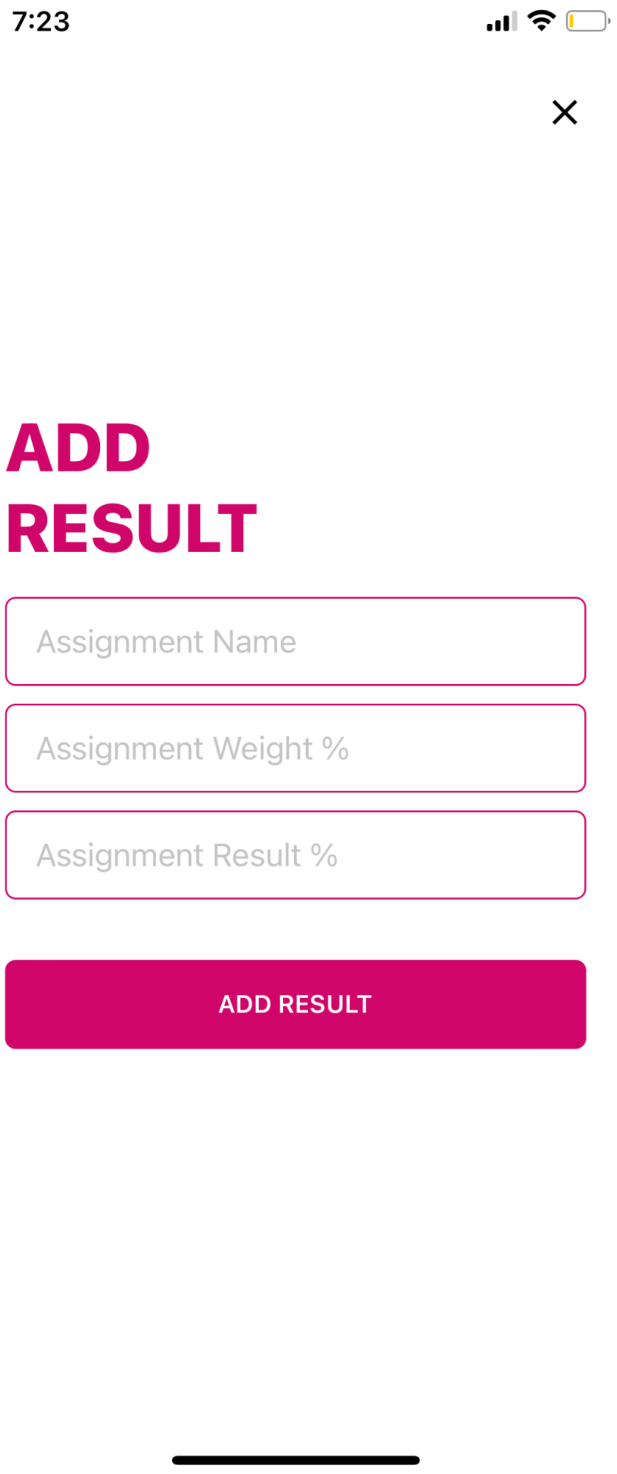
Title: AddCourseModal

Function:
User's are able to add courses to their course list to document their academic outcomes.
Record course - users enter necessary fields to produce a course object that is converted to a document and added to the relevant collection under the user's ID in firestore.



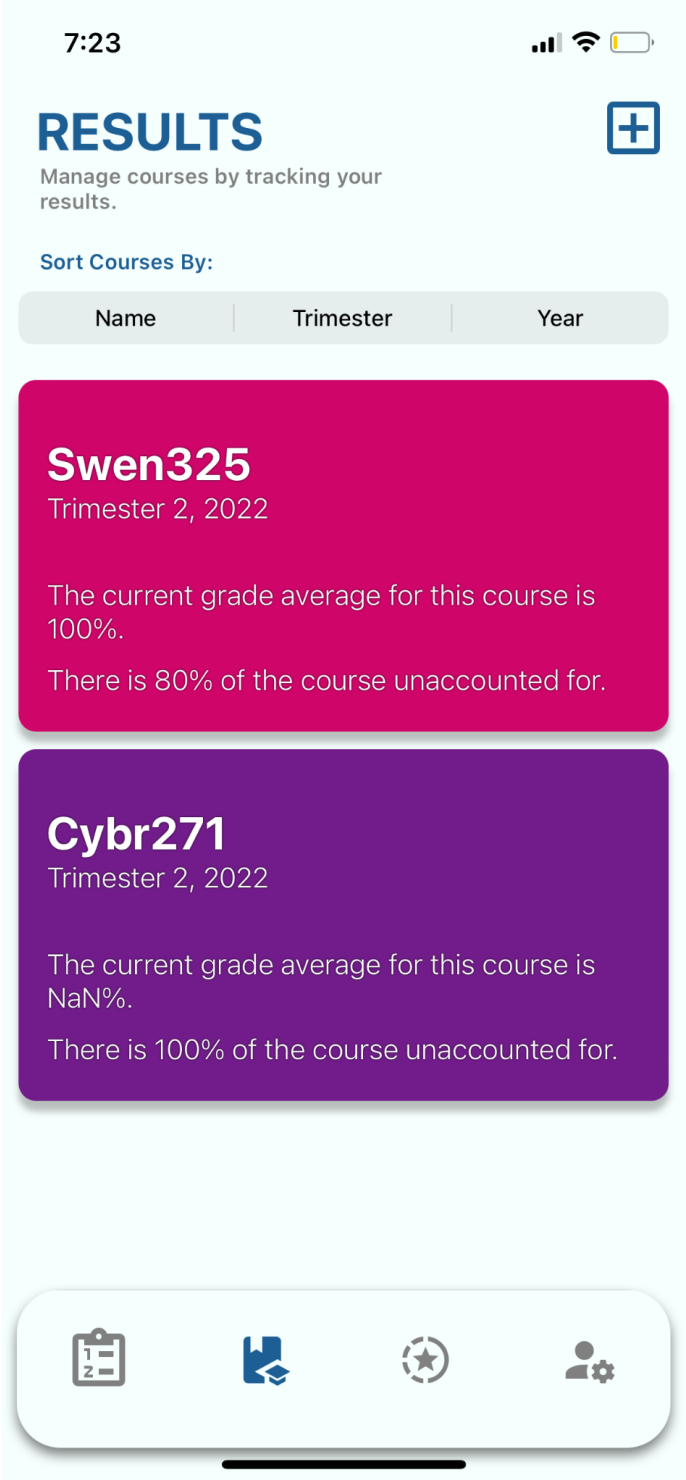
Title: AddResultModal

Function:
User's are able to add results from assignments to their courses to document their grade and what they need to achieve in remaining assignments to reach their desired grade outcome.
Add result - generates result object and add to course document in firestore.



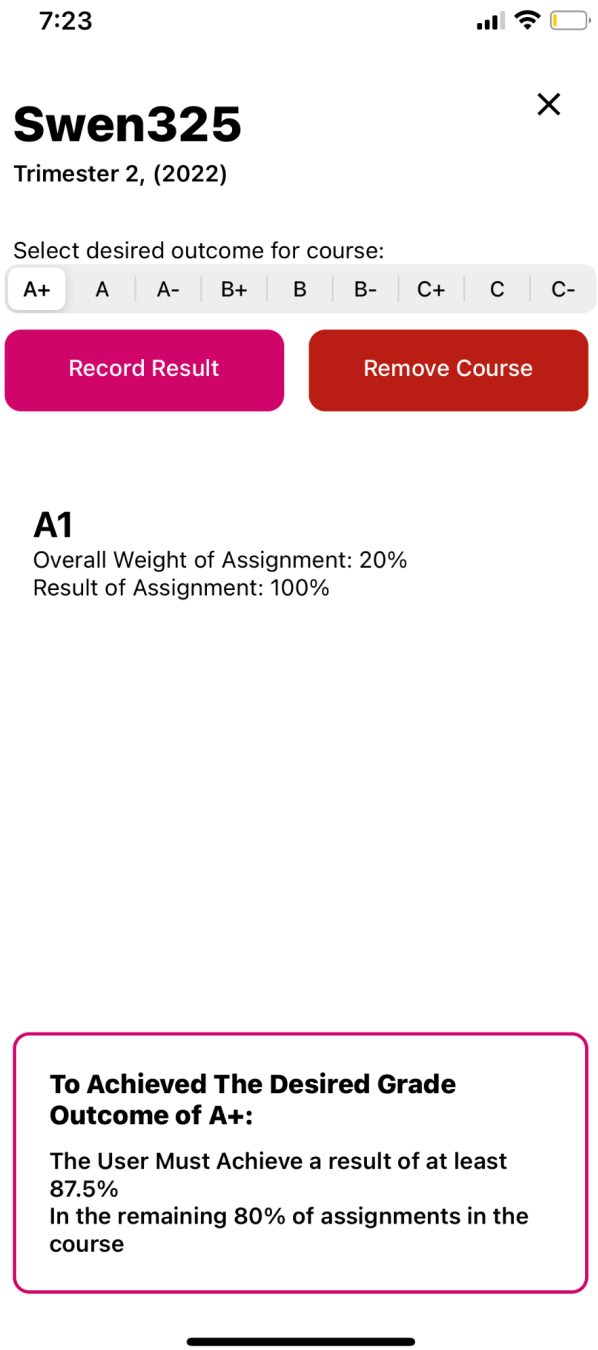
Title: MainCourseScreen

Function:
Users are able to reorder courses, keep track of course progress, add courses and plan how to achieve their desired outcome in their remaining assignments.



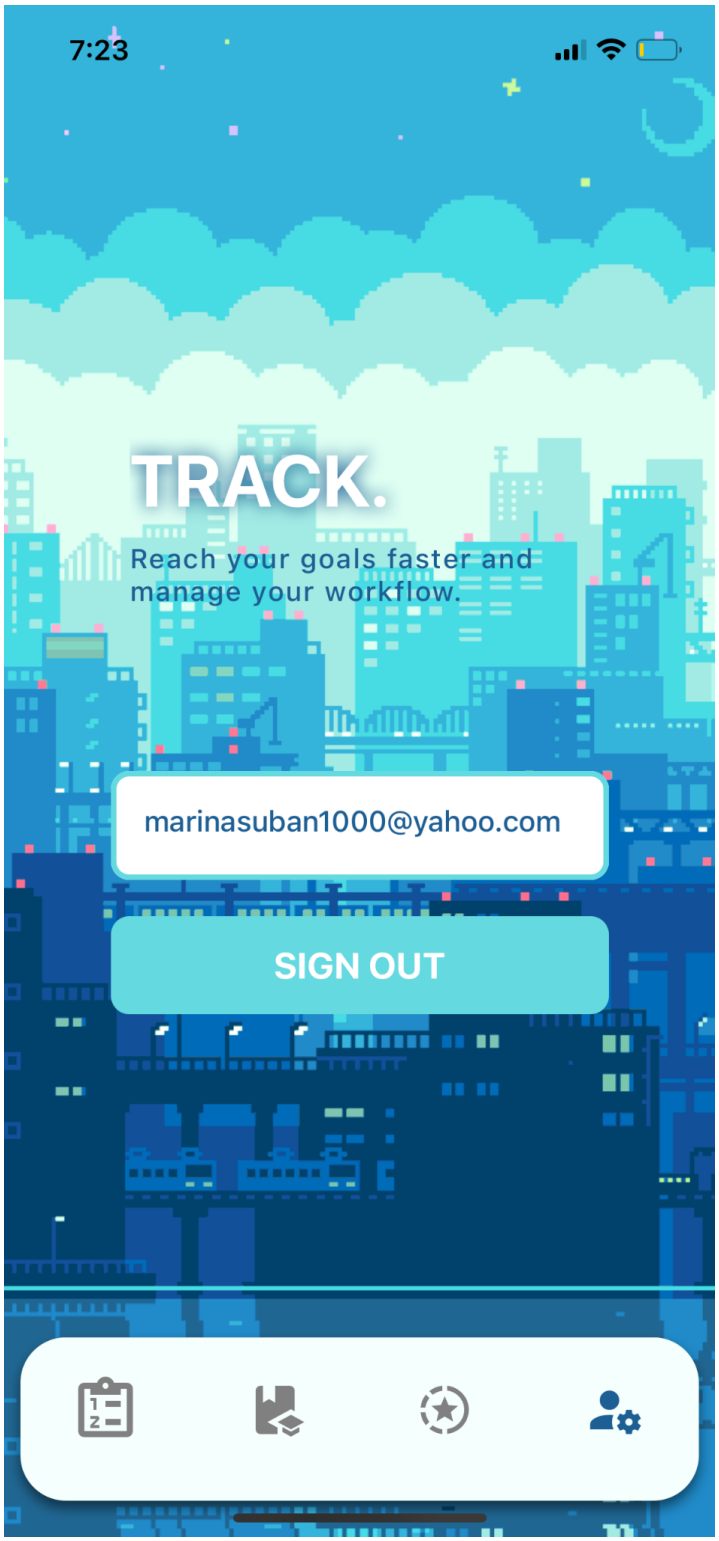
Title: ResultModal

Function:
Users are able to view previous results, add results, remove results, set desired grade outcomes and delete a course.
Delete results - removes log object from goal document in firestore
Delete course - removes course document from the user's collection on firestore.
Set desired course outcome - update course document from the user's collection on firestore.



Title: AccountScreen

Function:
Users are able to logout from this screen and are returned back to the OnBoarding screen. They are disconnected from the database.



Title: OnBoardingScreen

Function:
Users are able to register or login from this screen
Register - user enters necessary fields (input warning if not valid) - creates user associated to the input entered by user and adds a document with the userId to Firestore's "user" collection.
Login - input authenticated by firebase authentication, if user found proceed into the application and connect to relevant firestore collection otherwise input warning (invalid user).