# Univus

Cross-University Social Timetabling

# Table of Contents

**Table of Contents**........ **2**
Introduction........ 3
    Background Research on Functionality........ 3
        University Context........ 4
        University Courses Attributes and Classification........ 4
        Technical Architecture Brief........ 5
    Completed Features........ 6
        Mandatory Features........ 6
        Extra Features........ 6
        Additional Incomplete Features........ 7
**Design**........ **8**
    Technical Design (Architecture) Examples........ 8
        Setup (Expo Go + Firebase Emulators)........ 12
    Database Design........ 13
    UI/UX - Comparable Systems........ 17
        X (Formerly Twitter)........ 18
        Threads........ 19
        Notion........ 20
        NILE........ 21
    UI/UX - Figma Wireframes........ 22
**Implementation**........ **23**
**Testing (+ Mandatory Feature Evidence + ALL DESIGNS)**........ **28**
    React Native (iOS)........ 28
    Testing Conclusions........ 68
**Conclusion**........ **68**
References Used........ 69

# Introduction

Brandon Haynes
22410811

"**Univus**" is a new mobile app being developed for the University of GraceTech, as requested by Vice Chancellor Kwame Yeboah mentioned in the brief. The app's main goal is to make it easy for students to mark their attendance in lectures and to let them interact with each other on social media, posting comments on different timelines (course, global). Using React Native with Expo and Firebase, "Univus" is designed to be a helpful tool for both students and the university.

A special feature of "Univus" is that it **works for many universities, not just one**. This is possible because of the advanced data aggregation done with Python. However, this brings challenges, like managing different user accounts for different universities, and making sure the app works well for everyone. (Lecturers, Students and Universities). The app needs a strong system to handle these different needs.

For the app's design, ideas are taken from apps like Twitter, Threads, and NILE, as well as some imagination. The aim is to make "Univus" easy to use for students, and  innovative to handle the management side of things for lecturers and universities. Handling how the app keeps track of user information is important too. Redux (global state management for React) will used for this to make sure the app runs smoothly.

Another important part is getting "Univus" ready for real users. This means making sure the app works well in real situations and not just in testing. (Dev Expo Go builds, and Production builds), such as using Firebase emulators locally for testing, and real world firebase for production. The app will have different versions for testing and for when it's actually used by people.

In short, "Univus" is being made to be a useful and nice-looking app for universities. It's being built to meet the specific needs of the University of GraceTech, but also to be flexible for other universities. The journey of making "Univus" involves solving different technical and design problems to make an app that is both useful and attractive to its users.

# Background Research on Functionality

## University Context

In UK universities, they all have something called a **UKPRN** (UK Register of Learning Providers) ID. This ID is important because it gives each university a unique number. This makes it easier to tell them apart and is really useful for apps like "Univus" that work with different universities. Uni owners can search based on this ID as well.

When setting up Univus so that universities can join nd signup, it's important to make sure that *only real* universities can sign up. This is done by using special email addresses that belong to the university staff, not students. These email addresses help prove that the university (the proposed person signing up) is real. To get these email addresses, information from a website ('https://register-api.officeforstudents.org.uk/api/Download/') was put together and aggregated using an advanced Python script. If some email addresses were missing, extra research was done to find them. This way, Univus makes sure that only real universities can use the app.

The UKPRN ID is also helpful because it connects to other things like courses. This is good for the app because it lets it link courses to the right universities. This helps make the app work better and be more useful for students and universities.

## University Courses Attributes and Classification

A collection of current courses from *every* university in the UK was obtained. This data was sourced from "https://www.hesa.ac.uk/support/tools-and-downloads/unistats." Each course in this dataset is identified by a unique KISCODE, which acts as an ID for the course. These courses are then connected to their respective universities using the UKPRN, the unique identifier for universities. This linkage here  is crucial as it enables clear identification of which university offers which courses.

To manage and utilise this very very large extensive dataset effectively, a custom Python script was developed. This script's primary function was to aggregate all the course data, initially in an Excel file format, into a more manageable JSON file. This JSON file is then used to upload both course and university data to Firebase through a Firebase Function.

In a strategic move to minimise the Read/Write costs associated with Firestore, the data architecture was designed to include a separate JSON file for each university. Each of these files contains the specific data for courses and university details, ensuring that only relevant data is accessed for each institution. This approach not only optimises Firestore usage but also enhances the efficiency and speed of data retrieval within the app.

## Technical Architecture Brief

Univus is built on the React Native framework, chosen for its ability to create mobile apps using JavaScript (looking back, typescript would have been more professional to use). What makes React Native special is its cross-platform capability, allowing a single codebase to function on both Android and iOS devices. This feature simplifies the development process and ensures a consistent user experience across different platforms. Complementing React Native, Expo is used as a really nice framework and platform for universal React-based apps. Expo brings tools and services that make it easier to build, deploy, nd update iOS, Android, and web apps quickly from the same codebase.

The application also heavily relies on the Firebase Suite for various functionalities. Firestore, a NoSQL database from Firebase, plays a very crucial role in storing and syncing data between users and the cloud. This allows real-time updates, which are essential for features like attendance tracking and social media interactions. Firebase Auth is another component used in Univus, providing a comprehensive identity solution. A custom, well-thought-out user signup flow was implemented using a combination of Realtime Database (storing temporary students until they sign up themselves, as well as to save on read and write costs), Firestore for storing users after signup, and Auth, for linking the two. It supports different forms of authentication, which is crucial for managing user authentication across different universities. The Realtime Database, another Firebase service, is similar to Firestore but caters more towards applications trying to save on read and write costs, which we do. Think about the amount of students there are in every single university, for example. This problem was solved to a good degree by using Firestore and Realtime DB as a hybrid. Storage in Firebase is used for holding user-generated content like images (profile pics), enhancing the social media aspect of the app. Firebase Functions, which are serverless, are used for backend operations like data aggregation and processing functions that have a complicated flow, like checking username validation server-side, ensuring efficient backend management.

Python plays an absolutely massive role in data aggregation for Univus. It's used for gathering university and course data from various sources and aggregating that data in to a format best for uploading to Firebase in a firebase function. Python is a versatile language, and its extensive library ecosystem makes it ideal for handling and processing large datasets. With past experience in crypto-related jobs, the skills involved to make this work were incredibly valuable.

Given the complexity of Univus, Redux is incorporated for global state management. Redux is valuable in large applications for maintaining consistency across different components and user interactions. It helps manage the app's state effectively, ensuring a smooth user experience.

## Completed Features

Every feature has been developed to an extremely high standard, even though the time spent to develop the features was incredibly limited (for an app of this scale handling multiple universities).

### Mandatory Features

Below I have highlighted the MANDATORY features that have to be completed as referenced in the assignment brief.

Admin Privileges (100%)

- ☑ Enrol a new student **(100%)**
- ☑ Enrol a student on a course **(100%)**
- ☑ Delete a new student **(100%)**
- ☑ Update student details **(100%)**
- ☑ Read student details **(100%)**

Student Privileges (100%)

- ☑ Update their profile details **(100%)**
- ☑ Post comments on the course timeline or global timeline **(100%)**

Other Assumed, Mandatory Privileges (based on as2 brief paragraph) (100%)

- ☑ Allow university owners to create courses **(100%)**
- ☑ Allow university owners to create modules **(100%)**
- ☑ Allow lecturers to create lessons **(100%)**
- ☑ Allow lecturers to manage attendance for lessons **(100%)**
- ☑ Allow students to register attendance without lecturers **(100%)**
- ☑ **Beautiful GUI / Design (100%)**

### Extra Features

Functionality appropriate for a *production-level* application!

Custom

- ☑ **QR Code Attendance** Functionality
- ☑ **Custom Website for Attendance** built with Next.js & Vercel (https://univus.co)
- ☑ **Dark Mode / Light Mode functionality (app & site)**
- ☑ **CROSS-UNIVERSITY TIMELINE for Social Media Posts**
- ☑ **3-Role system (Lecturers, Students, Universities)**
- ☑ **REAL-LIFE COURSES FROM REAL UNIVERSITIES**
- ☑ **Advanced Custom User Flow (with Firebase Functions)**
- ☑ **Expo Environment support (Firebase Emulators for Expo Go)**
- ☑ Custom-Made Logo

- [x] Timeline Posts Scroll-to-Refresh + Pagination
- [x] Advanced, Optimised Splash Screen to load data in background
- [x] Custom animations
- [x] **Global State Management for Scalability Purposes**
- [x] **Manual Student Attendance feature**
- [x] **Tested for Production Build (Expo)**
- [x] **Read/Write Firestore Costs optimised**

## Additional Incomplete Features

**Due to time constraints, these features could not be implemented in time.** They were planned from the very beginning.

**<u>Incomplete</u>**

- [ ] Dynamic, Fast-Expiring, User-Specific Attendance QR Codes to prevent attendance fraud. Decided to just generate one for time-sake.
- [ ] "Devices" screen for students, in which they can only register attendance on one device, until 30 days have passed, then they can change the device they register attendance on. This was to prevent students giving their phone to a friend in class to sign in for them (or login on another device). In combination with QR Codes, this would most likely 99% prevent fraudulent attendance marking.
- [ ] Replies on posts (Would have done a recursive firestore document structure)
- [ ] Likes on Posts (denormalised. A counter, and an array of users for seeing who liked the post, with pagination)
- [ ] Go to any profile via post (one line could have added this in, but couldn't be bothered as it was not a necessity at the time  in comparison to attendance features)
- [ ] Edit / Delete Modules
- [ ] Edit / Delete Lessons
- [ ] Haptic Feedback support
- [ ] NOTIFICATIONS :/
- [ ] CRON JOBS for deleting non-verified accounts.
- [ ] Production Security Rules - This is a huge weakness. Used Firebase Functions for most backend logic though.
- [ ] Trigger Functions for Updates
- [ ] Forgot Password

## Design

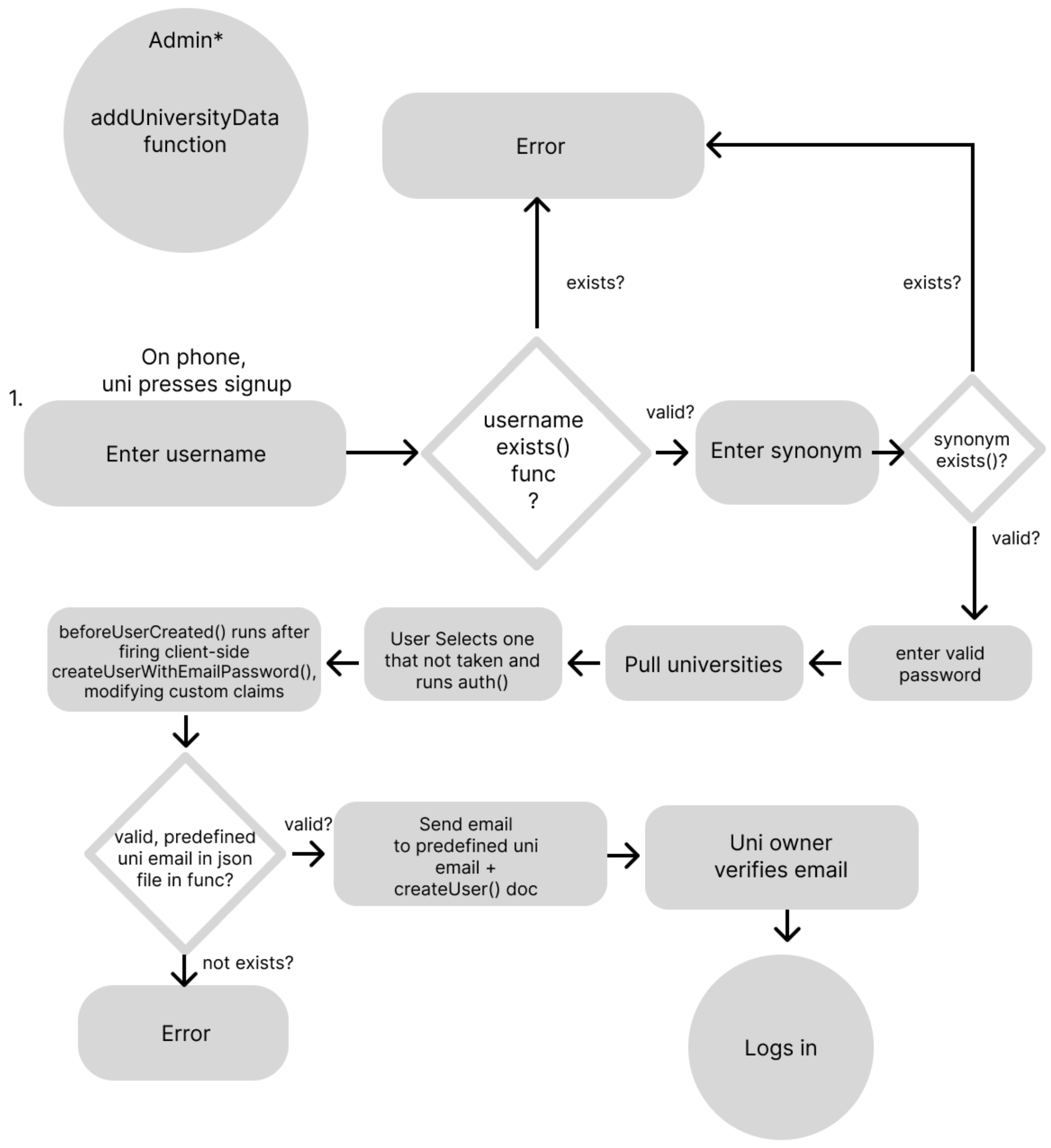
During the development of the application, the design, *especially* the UI/UX, turned out to be the toughest and most time-consuming part. The application has become one of the most detailed in terms of design. For ideas, there was a look at well-known social media websites and apps, as well as attendance apps like NILE, with a goal to make the mobile app (and additional website) just as easy to use. The technical architecture also had its challenges, at a much trickier difficulty due to the multi-university setup.

### Technical Design (Architecture) Examples

Below are some prime examples of the complicated architecture of the app, going through Firebase Authentication, Functions, Realtime Databaase, Firestore, Blocking Auth Functions, etc.

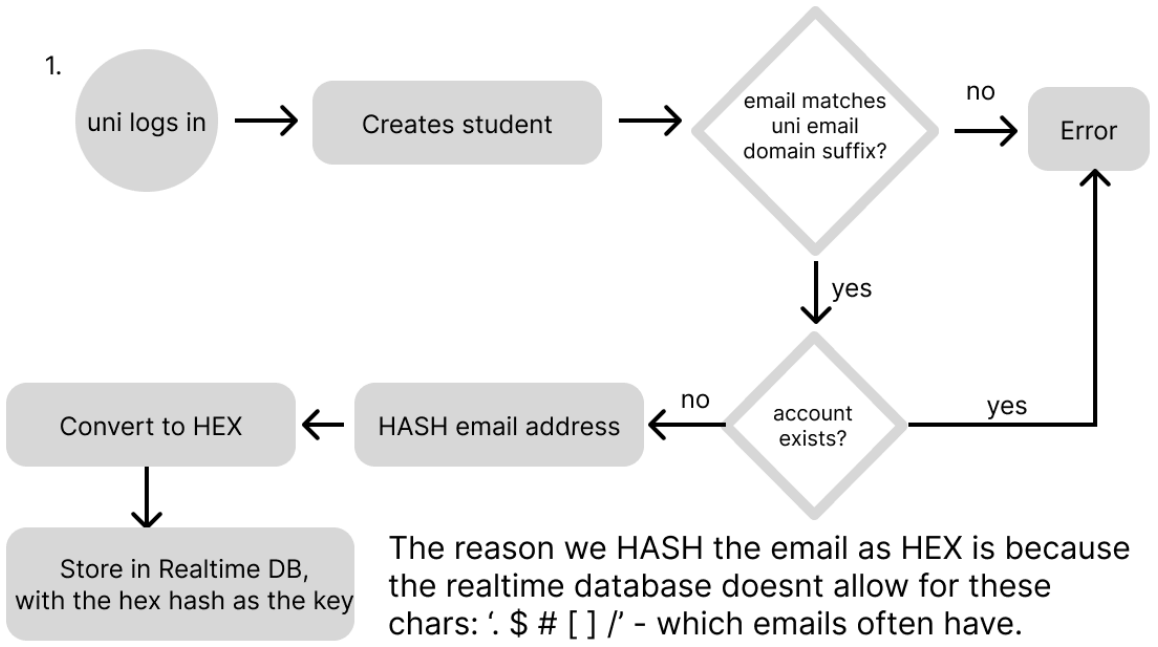
## Diagrams on Next Page

# Institution Custom Signup Flow

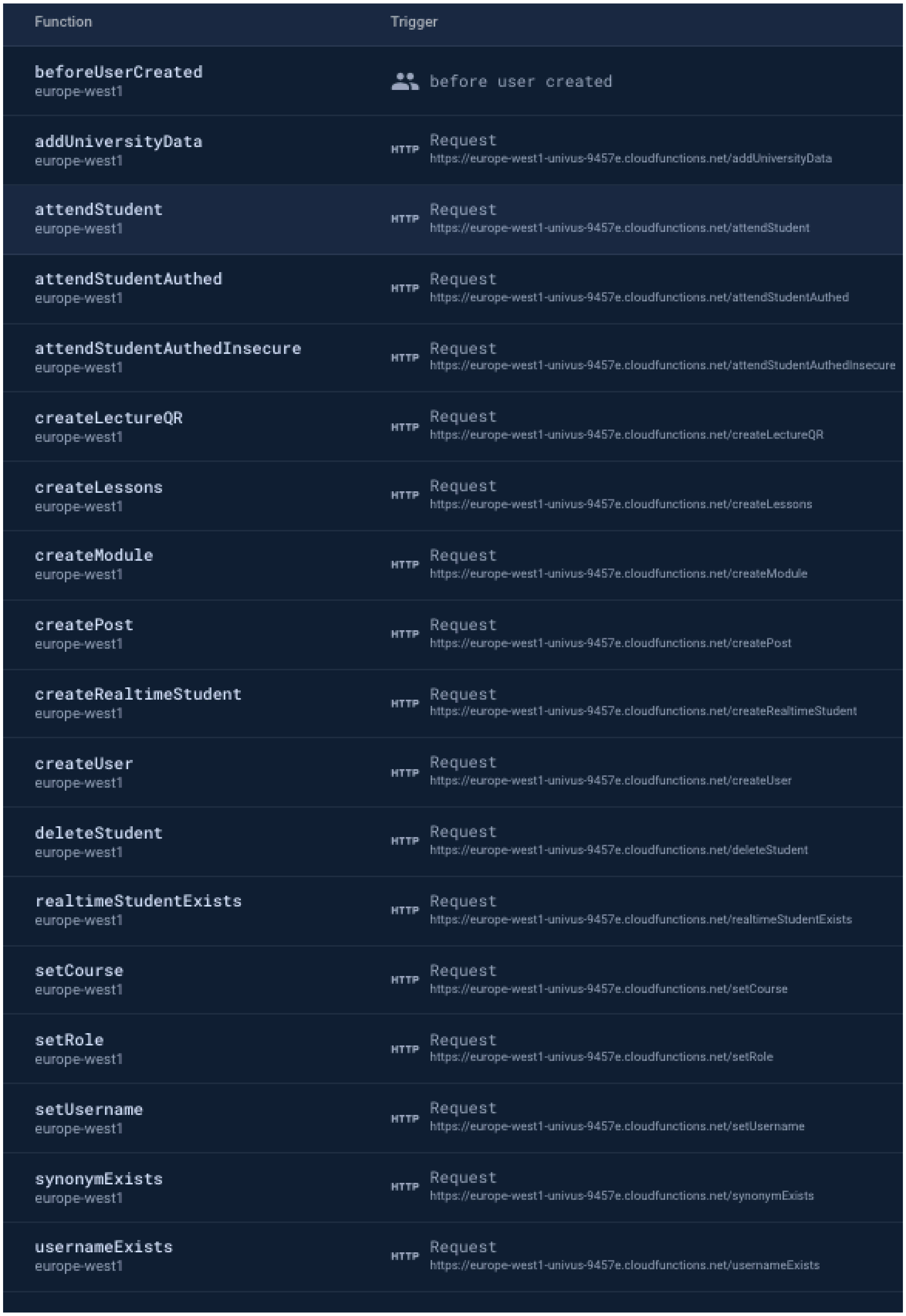

Admin*
addUniversityData function
Error
exists?
exists?
On phone, uni presses signup
1.
Enter username
username exists() func ?
valid?
Enter synonym
synonym exists()?
valid?
beforeUserCreated() runs after firing client-side createUserWithEmailPassword(), modifying custom claims
User Selects one that not taken and runs auth()
Pull universities
enter valid password
valid, predefined uni email in json file in func?
valid?
Send email to predefined uni email + createUser() doc
Uni owner verifies email
not exists?
Error
Logs in

# Student Custom Enrolment Flow

(Not a student signing up)



The reason we HASH the email as HEX is because the realtime database doesnt allow for these chars: ‘. $ # [ ] /’ - which emails often have.
We hash because we perform lookups using raw email, hash on backend (firebase function), then lookup.



| Function | Trigger |
|---|---|
| beforeUserCreated<br>europe-west1 | before user created |
| addUniversityData<br>europe-west1 | HTTP Request<br>https://europe-west1-univus-9457e.cloudfunctions.net/addUniversityData |
| attendStudent<br>europe-west1 | HTTP Request<br>https://europe-west1-univus-9457e.cloudfunctions.net/attendStudent |
| attendStudentAuthed<br>europe-west1 | HTTP Request<br>https://europe-west1-univus-9457e.cloudfunctions.net/attendStudentAuthed |
| attendStudentAuthedInsecure<br>europe-west1 | HTTP Request<br>https://europe-west1-univus-9457e.cloudfunctions.net/attendStudentAuthedInsecure |
| createLectureQR<br>europe-west1 | HTTP Request<br>https://europe-west1-univus-9457e.cloudfunctions.net/createLectureQR |
| createLessons<br>europe-west1 | HTTP Request<br>https://europe-west1-univus-9457e.cloudfunctions.net/createLessons |
| createModule<br>europe-west1 | HTTP Request<br>https://europe-west1-univus-9457e.cloudfunctions.net/createModule |
| createPost<br>europe-west1 | HTTP Request<br>https://europe-west1-univus-9457e.cloudfunctions.net/createPost |
| createRealtimeStudent<br>europe-west1 | HTTP Request<br>https://europe-west1-univus-9457e.cloudfunctions.net/createRealtimeStudent |
| createUser<br>europe-west1 | HTTP Request<br>https://europe-west1-univus-9457e.cloudfunctions.net/createUser |
| deleteStudent<br>europe-west1 | HTTP Request<br>https://europe-west1-univus-9457e.cloudfunctions.net/deleteStudent |
| realtimeStudentExists<br>europe-west1 | HTTP Request<br>https://europe-west1-univus-9457e.cloudfunctions.net/realtimeStudentExists |
| setCourse<br>europe-west1 | HTTP Request<br>https://europe-west1-univus-9457e.cloudfunctions.net/setCourse |
| setRole<br>europe-west1 | HTTP Request<br>https://europe-west1-univus-9457e.cloudfunctions.net/setRole |
| setUsername<br>europe-west1 | HTTP Request<br>https://europe-west1-univus-9457e.cloudfunctions.net/setUsername |
| synonymExists<br>europe-west1 | HTTP Request<br>https://europe-west1-univus-9457e.cloudfunctions.net/synonymExists |
| usernameExists<br>europe-west1 | HTTP Request<br>https://europe-west1-univus-9457e.cloudfunctions.net/usernameExists |

## Setup (Expo Go + Firebase Emulators)

Install Expo: Run npm install -g expo-cli in your terminal.
Install Firebase Emulators: Run npm install -g firebase-tools and then firebase init emulators in your terminal.

‘cd  in to ‘univus/univus’

IF USING FIREBASE EMULATORS, RUN EMULATORS WITH ‘firebase emulators:start’. After, modify the firebaseConfig.js (in univus/univus directory) to connect to the correct ports, etc. THEN, CHANGE ‘if (!__DEV__)’ TO ‘if (__DEV__) ’, and save.

IF YOU ARE NOT USING EMULATORS (YOU’RE USING ACTUAL, REAL, LIVE, FIREBASE) - then don’t touch anything if you are running with Expo Go.

Run ‘npm expo start’.

Open on a device.

Enjoy.

–

( If you are creating a production build of the app, be sure to change ‘if (!__DEV__)’ TO ‘if (__DEV__) ’, in firebaseConfig.js )

—-

## Database Design

The database design involves denormalization, or duplicating data across collections and documents, to reduce read and write costs. It was very tricky to get this right! This approach is evident in the structure of the posts, users, and lectures, etc.
The reason both Firestore & Realtime Database were used is to lower read/write costs when universities enrol students. Since every university is allowed to signup, think about how many students that is! Students (users) only get created in Firestore properly IF the student signs up themselves.

One weakness with the denormalisation approach is that, due to time constrains, there was just not enough time to implement proper trigger functions to update the correct documents (CASCADE).

Below are the attributes for each document in a collection.

/universities collection:

| Attribute | Type |
|---|---|
| email | String |
| emailDomain | String |
| legalName | String |
| ukprn | String |
| courses | Map{} |
| synonym | String |
| synonymUppercase | String |

/universities/courses (it’s a map, not a subcollection):

| Attribute | Type |
|---|---|
| courseId | String |
| title | String |

/users

| Attribute | Type |
|---|---|
| courseID | String |
| courseTitle | String |
| createdAt | Timestamp |

| | |
|---|---|
| email | String |
| emailHash | String |
| emailVerified | Boolean |
| fullName | String |
| role | String |
| synonym | String |
| synonymUppercase | String |
| universityDomain | String |
| universityId | String |
| universityLegalName | String |
| updatedAt | Timestamp |
| userId | String |
| username | String |
| usernameUppercase | String |

/posts

| Attribute | Type |
|---|---|
| courseId | String |
| courseTitle | String |
| id | String |
| likes | Number |
| photoURL | String |
| postText | String |
| replyTo | Null/String |
| synonym | String |
| timeline | String |
| timestamp | Number |
| universityId | String |

| | |
|---|---|
| universityLegalName | String |
| uppercaseUsername | String |
| userId | String |
| username | String |

/modules

| Attribute | Type |
|---|---|
| courses | Array |
| createdAt | Number |
| createdBy | String |
| id | String |
| lessons | Array |
| name | String |
| universityId | String |

/lectures

| Attribute | Type |
|---|---|
| duration | Number |
| endDate | Timestamp |
| lectureId | String |
| lecturerId | String |
| moduleId | String |
| moduleName | String |
| room | String |
| startDate | Timestamp |
| studentIds | Array |
| students | Map |

| | |
|---|---|
| uniqueId | String |
| universityId | String |

/lectureSecrets

| Attribute | Type |
|---|---|
| attendanceLectureCode | String |
| attendanceLectureQR | String |
| lectureId | String |
| lecurerId | String |
| secret | String |
| uniqueId | String |
| universityId | String |

/users (REALTIME DATABASE)

| Attribute | Type |
|---|---|
| course | String |
| createdBy | String |
| email | String |
| emailHash | String |
| fullName | String |
| role | String |
| synonym | String |
| universityDomain | String |
| universityId | String |
| universityName | String |
| uppercaseEmail | String |
| uppercaseSynonym | String |
| userId | String |

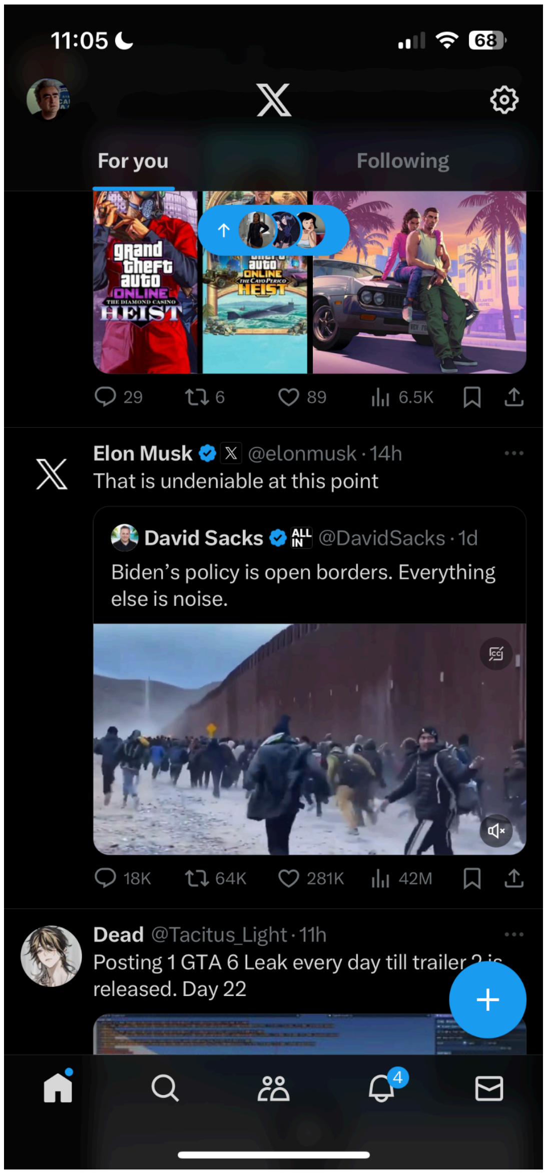
## UI/UX - Comparable Systems

For designing Univus, some well-known platforms were used as models. The app's Posts, Home Screen, and Drawer Designs were inspired by X, which used to be Twitter. A special feature from X that was really useful was switching between dark and light mode. This makes the app easier to use in different lights and more comfortable for users.

More design ideas came from Threads, which is like Facebook's version of Twitter/X. Threads was mainly used for ideas on how to make the profile screen look good and easy to use. Also, NILE (Northampton Integrated Learning Environment) helped shape the design of the "Univus" attendance website (https://univus.co). To make the website look great and work well, NextUI's tools and styles were used, and it was built with Next.js. For the Welcome Screen in the Univus App, the Welcome Screen from Notion was used as a guide. This was to make sure new users find the app welcoming and easy to understand.

For the logo design, DALL-E was used to generate logos based on prompts given, remove the background using photoshop and clean up the logo by modifying and creating an SVG to ensure the logo can be used across everywhere.

## X (Formerly Twitter)


11:05

For you

Following

29 6 89 6.5K

Elon Musk @elonmusk · 14h

That is undeniable at this point

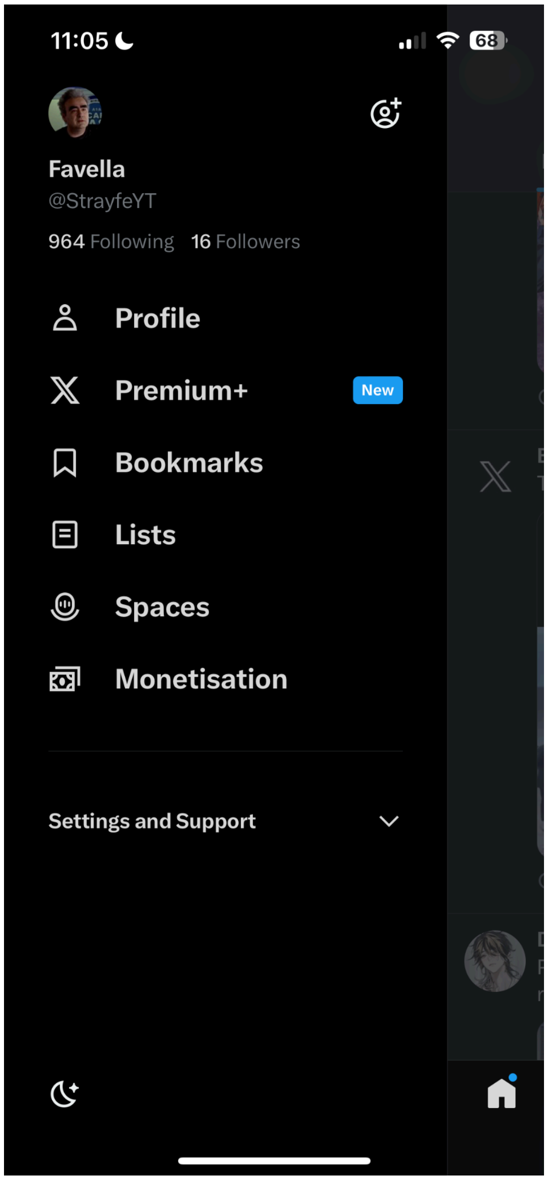
David Sacks @DavidSacks · 1d

Biden's policy is open borders. Everything else is noise.

18K 64K 281K 42M

Dead @Tacitus_Light · 11h

Posting 1 GTA 6 Leak every day till trailer 2 is released. Day 22


11:05

Favella

@StrayfeYT

964 Following 16 Followers

Profile

Premium+ New

Bookmarks

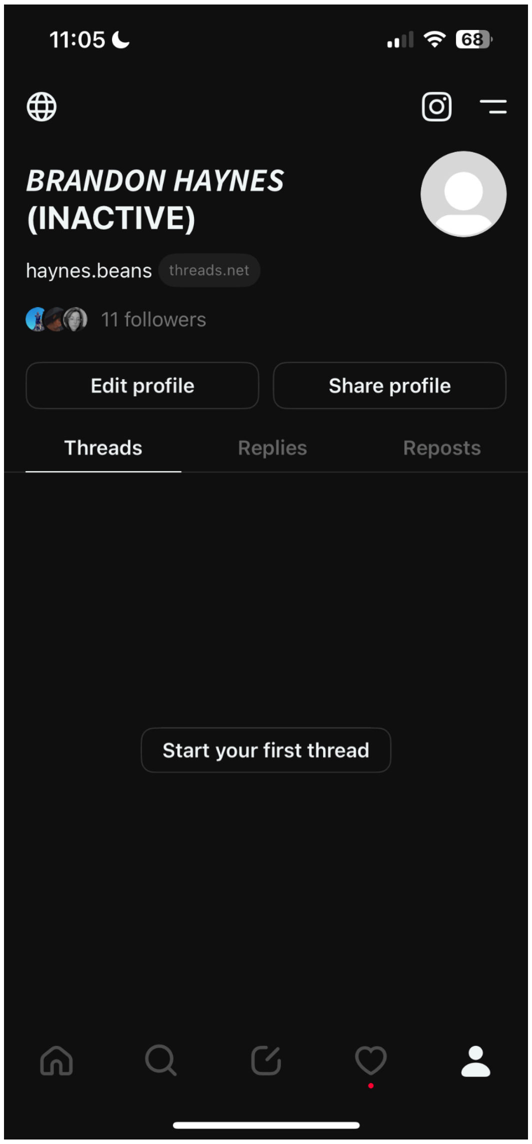
Lists

Spaces

Monetisation

Settings and Support

## Threads

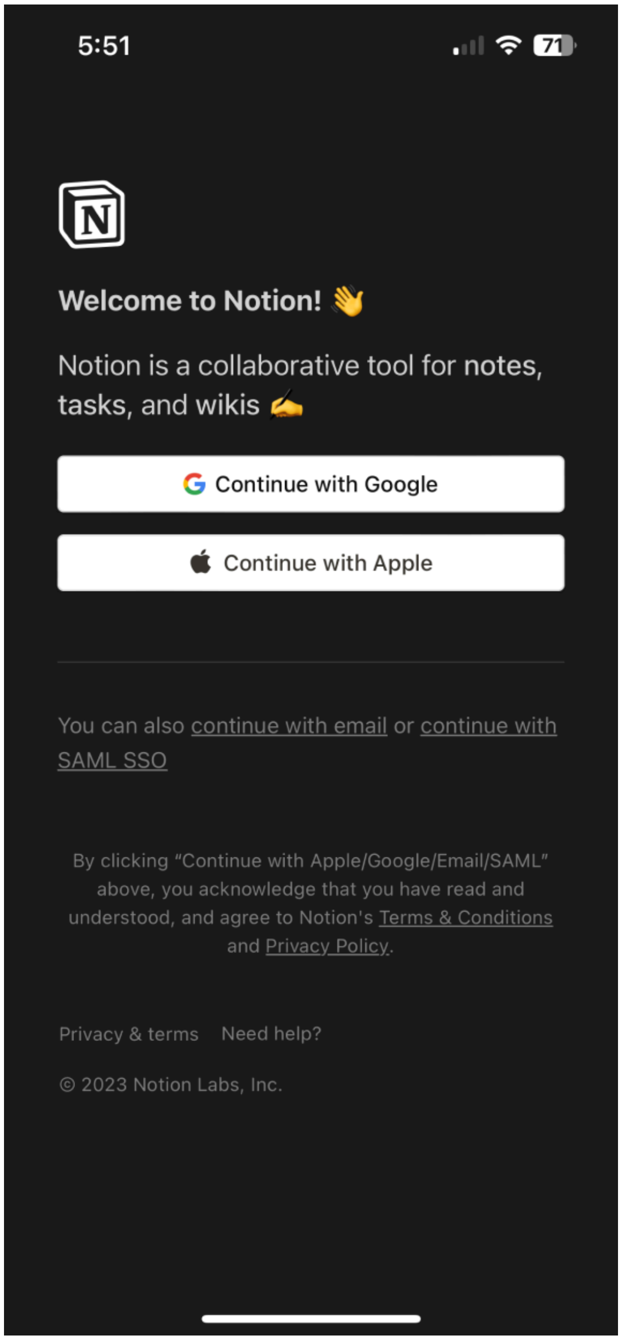

11:05

BRANDON HAYNES (INACTIVE)

haynes.beans threads.net

11 followers

Notion



5:51

71

Welcome to Notion! 👋

Notion is a collaborative tool for notes, tasks, and wikis ✍️

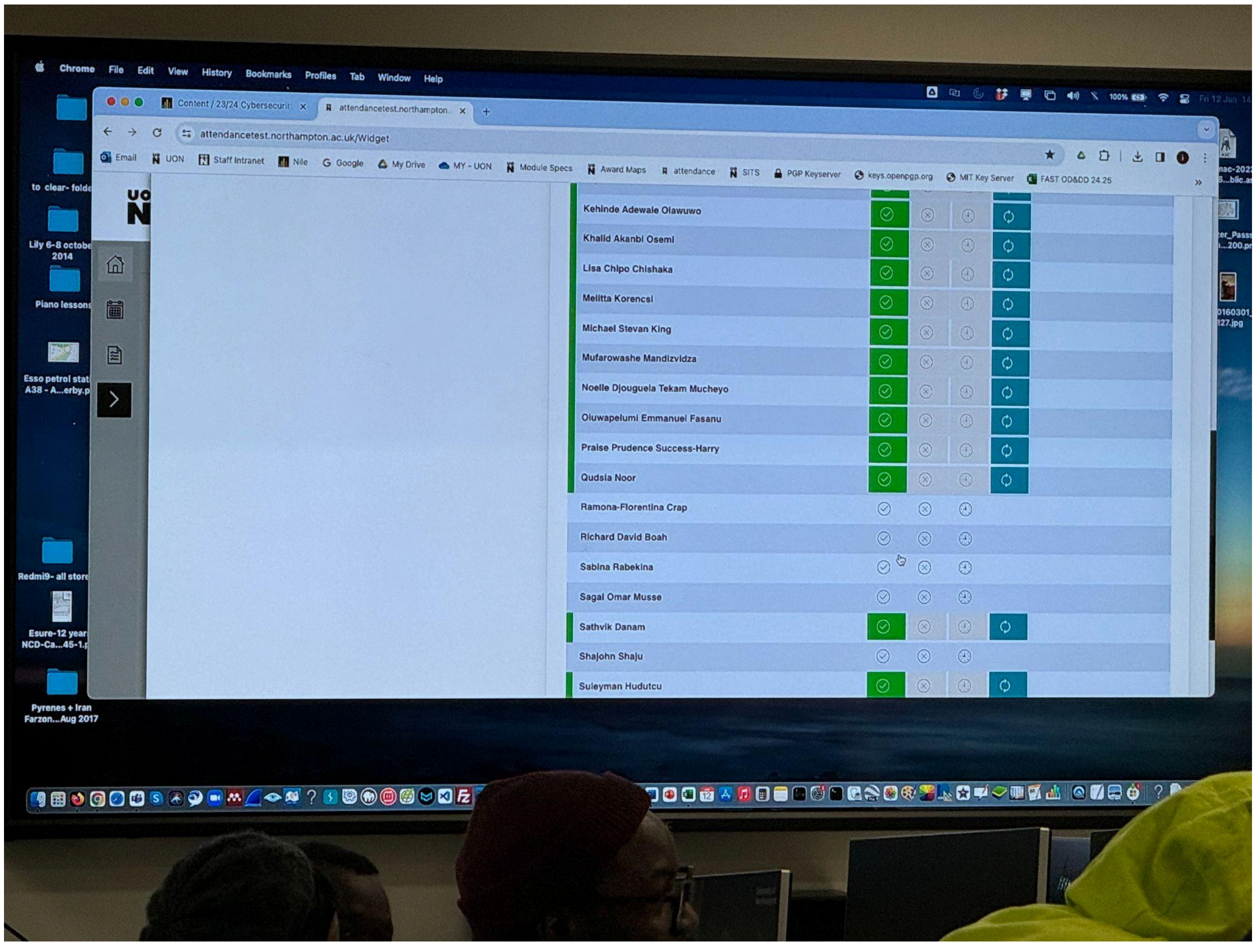
You can also continue with email or continue with SAML SSO

By clicking "Continue with Apple/Google/Email/SAML" above, you acknowledge that you have read and understood, and agree to Notion's Terms & Conditions and Privacy Policy.

Privacy & terms Need help?

© 2023 Notion Labs, Inc.

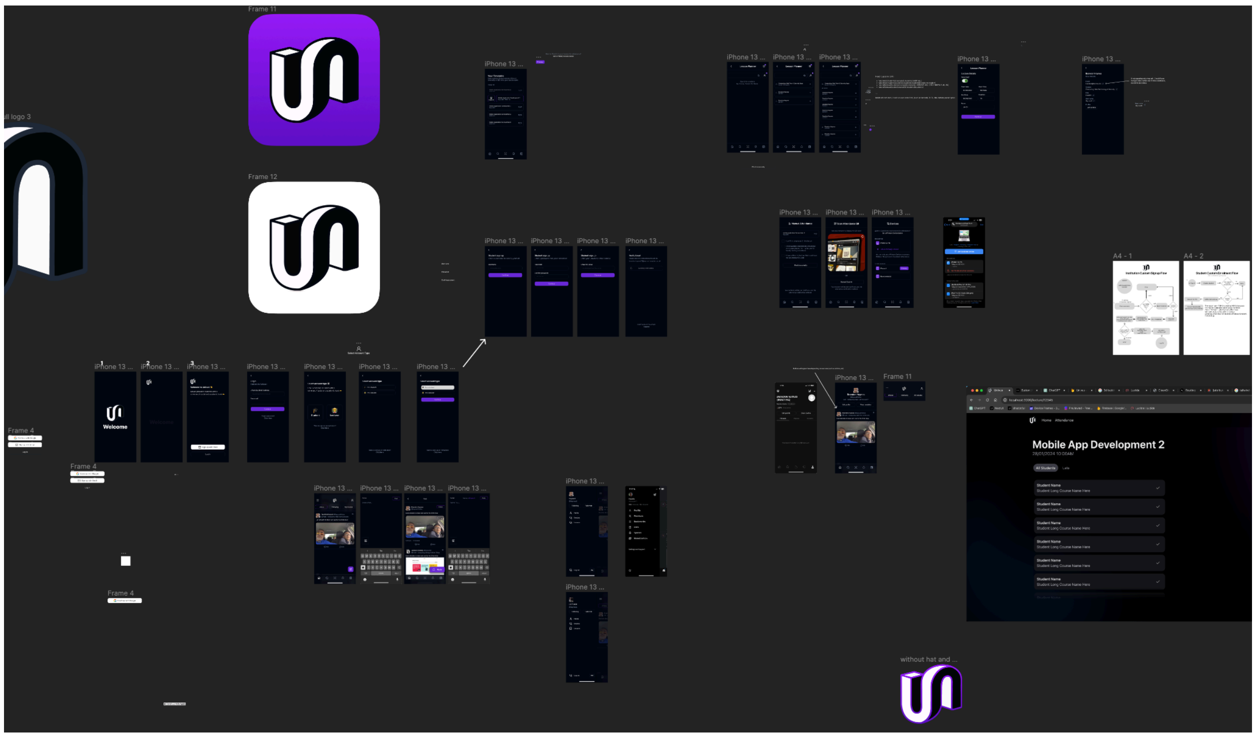
NILE


Chrome File Edit View History Bookmarks Profiles Tab Window Help
Content / 23/24 Cybersecurit
attendancetest.northampton
attendancetest.northampton.ac.uk/Widget
Email UON Staff Intranet Nile Google My Drive MY - UON Module Specs Award Maps attendance SITS PGP Keyserver keys.openpgp.org MIT Key Server FAST OD&DD 24.25

| | | | | |
|---|---|---|---|---|
| Kehinde Adewale Olawuwo | 🟩 | ⊗ | ◷ | ↻ |
| Khalid Akanbi Osemi | 🟩 | ⊗ | ◷ | ↻ |
| Lisa Chipo Chishaka | 🟩 | ⊗ | ◷ | ↻ |
| Melitta Korencsi | 🟩 | ⊗ | ◷ | ↻ |
| Michael Stevan King | 🟩 | ⊗ | ◷ | ↻ |
| Mufarowashe Mandizvidza | 🟩 | ⊗ | ◷ | ↻ |
| Noelle Djouguela Tekam Mucheyo | 🟩 | ⊗ | ◷ | ↻ |
| Oluwapelumi Emmanuel Fasanu | 🟩 | ⊗ | ◷ | ↻ |
| Praise Prudence Success-Harry | 🟩 | ⊗ | ◷ | ↻ |
| Qudsia Noor | 🟩 | ⊗ | ◷ | ↻ |
| Ramona-Florentina Crap | ✓ | ⊗ | ◷ | |
| Richard David Boah | ✓ | ⊗ | ◷ | |
| Sabina Rabekina | ✓ | ⊗ | ◷ | |
| Sagal Omar Musse | ✓ | ⊗ | ◷ | |
| Sathvik Danam | 🟩 | ⊗ | ◷ | ↻ |
| Shajohn Shaju | ✓ | ⊗ | ◷ | |
| Suleyman Hudutcu | 🟩 | ⊗ | ◷ | ↻ |

## UI/UX - Figma Wireframes

Figma was *really* important in making Univus. It was used to make detailed wireframes before starting the actual coding. Figma is great because it lets designers try out different designs easily. With Figma, it was possible to play around with how the app looks, like changing layouts and colours, without needing to code anything yet. This method is very useful because it saves time and makes sure the final design is good for users. It's common to use Figma like this in app development, to make sure everything looks right before building it.
**There is a high quality figma link here:**
**https://www.figma.com/file/kkasMCifgxoxKMLZa9C0nl/AS2-Mobile-%7C-22410811?type=design&node-id=0%3A1&mode=design&t=yajumbLIxUGGzJKG-1**



# Implementation

**Root.js, Redux Store & Slices:**
In the application, Redux was used to manage the global state, an important part of keeping track of user information and app behaviour. Redux basically works by using a central store where the app's state is stored. This store is divided into *'slices'*, each handling a different part of the state. For user authentication, a 'userSlice' was created. Actions and reducers within this slice were used to update the user's login status. Based on the information stored in the userSlice, the app could then conditionally render the correct screens. Root.js plays a crucial role by using Redux slices to manage various states, such as loading screens and user authentication. The appInitSlice is particularly important as it handles the loading of fonts. When the app starts, Root.js uses this slice to check if the fonts are loaded. Once loaded, it updates the state, which is tracked in Redux.

The user's login status is managed by another slice, which keeps track of whether a user is logged in or not. This information is used to decide which screens to display. If the user is logged in, they see one set of screens, and if not, they see a different set. This is done by checking the auth status stored in the userSlice.

Overall, Root.js, with the work of (annoyingly tedious to setup) Redux slices like appInitSlice and the user authentication slice, efficiently manages the app's initial loading screen, font loading, and user authentication states. This setup ensures that the right screens are displayed based on the user's login status, enhancing the user experience.

**Navigation.js:**
Navigation.js plays a crucial role in guiding the user through  various screens and functionalities. It uses *createNativeStackNavigator* from React Navigation to handle the navigation stack, ensuring a smooth transition between screens.

The Redux store is ysed to determine the user's auth status. Based on this status, different navigation stacks are presented. If the user is not logged in (user === null), the AuthStack is shown, guiding them through the authentication process. Once the user is authenticated (user !== null), and if their email is verified (user.verified), they are directed to the DrawerGroup, which is the main app interface. If the user is not verified, the VerifyEmailStack is shown to prompt them for email verification.

Several specific screens like CalendarSelectScreen, ManageStudentsAddSelectScreen, and others are integrated into the navigation stack too for more specific use cases, ensuring the right screens are shown where the user shouldnt see drawer or bottom tabs for example. Each screen is configured with custom navigation options, such as modal presentation and custom headers. These headers are styled using the useTheme hook, ensuring consistency with the app's dark mode and light mode.

Navigation.js effectively manages the flow of the application pretty much.

**firebaseConfig.js:**
In the firebaseConfig.js file, the application's connection to Firebase is configured to adapt based on the development or production environment. The Firebase application is initialised with specific settings provided in firebaseConfig, which include keys and identifiers for Firebase services. (Didn't put in env file since this can be public as per Firebase documentation) This initial setup is cruc ial for the application to connect correctly to the project's  Firebase services.

The environment in which the app is running is determined using Expo's Constants. If the app runs in Expo Go, it's recognized as being in a development environment. In this case, the script connects the app to Firebase emulators for services like Auth, Firestore, Functions, Storage, and Database. These emulators are essential for development, allowing for testing of Firebase functionalities without impacting the production database. The connection to these emulators is configured based on the runtime environment, defaulting to "localhost" if no specific setting is found.

For Firebase services such as Firestore, Functions, Storage, and Realtime Database, they are initialised using the Firebase app instance, enabling various functionalities within the app. This flexible setup in firebaseConfig.js ensures that you can efficiently work in a development environment using Firebase emulators and then seamlessly transition to real Firebase services for production builds.

**App.js:**
In App.js, the app is set up with important parts for managing state and themes. It uses the Redux Provider with the app's store for handling the app's state everywhere. The ThemeProvider makes sure the app looks consistent and can be used from anywhere. Inside these, the Root component takes care of the main navigation and screens. Also, the splash screen is set to stay until everything is ready. This makes App.js a key part of putting the app together, combining state management, theming, and navigation.

**WelcomeScreen.js:**
In the WelcomeScreen of the app, animations are used to make the screen lively and engaging. When the screen loads, animations are set up for the logo, welcome text, description text, and buttons at the bottom. These animations include fading in, making each

component appear in a smooth, staggered manner. This is done using Animated from react-native, with each element having its own animation controlled by Animated.Value. The animations start together, but each element appears one after the other, adding a nice effect to the welcome experience. This screen not only welcomes users but also showcases the app's interactive and user-friendly design. Pretty nice :)

**AttendanceQRScreen:**
In the AttendanceQRScreen,  a student can sign in to a lesson by scanning a QR code. When the screen opens, it asks for camera permission to scan QR codes. The QR code scanned is a special secret (hashed, but completely random) code. After scanning, the app checks if the code starts with 'lesson_'. If it does, the code is sent to a Firebase function to see if it's a valid code for attendance.

The app uses the camera to scan the QR code. If the code is right, a message says the attendance is recorded. If not, it says the attendance is not recorded. The screen uses a timer to avoid scanning the same code many times quickly. This helps prevent too many requests to the server. The screen is simple and tells the student to ask the lecturer for the QR code to scan for attendance.

**ManageLecturesAddLectureCreateLessons (+ other ManageLecture screens):**
In the ManageLecturesAddLectureCreateLessons screen, lecturers can create lessons for their courses. This screen includes options to set the start date and time, duration, and room for the lecture. There's also a feature to make the lecture recurring weekly, with the ability to choose how many weeks it repeats.

The screen uses the editLecture state from Redux to keep track of the lecture details. This state helps manage information like the start date, module, and students for the lecture. By using this state, information can be easily shared and managed across different screens in the app. When the lecture details are set, a Firebase function is called to create the lesson with the specified details. This function updates the information in the database, completing the lecture creation process. The screen is designed to be user-friendly, ensuring lecturers can easily set up their lectures.

**AuthStack**
The AuthStack in the app is a group of screens that help users sign up or log in. It starts with a welcome screen. Then, if a user wants to sign up, they go through different screens to give their information. This includes choosing the type of account, setting up a username, password, and email. There are also screens to verify the email and pick a university. There's a screen for users who already have an account to log in.
The app uses a navigatorr to move between these screens easily. This means users can go back to a previous screen without losing any information they've already entered. The look of these screens matches the app's overall style, so everything looks consistent.
In short, the AuthStack is an important part of the app. It makes sure users can sign up or log in smoothly.

**SignUpInstitutionSelect:**
In the SignUpInstitutionSelect screen, users can pick their university when signing up. When the screen opens, it loads a list of universities from the database. While loading, the screen shows placeholders where the university list will appear. This is done using animations and skeleton screens to make the loading look nice. Once the list is loaded, users can search for their university. They can type the name or the UKPRN (a unique number for each university) in the search bar. When they find their university, they can select it. After selecting, the app takes them to the next screen to continue signing up. This is specifically for institutions only. General students won’t be using this to sign up.

**CreatePostScreen**
The CreatePostScreen in the app allows users to craft and share their messages. On this screen, a text input field is provided where users can type out their post.
At the top of the screen, there's a customizable header component, CustomPostHeader, which displays the selected timeline for the post. This could be a global timeline, specific to a university, or related to a course, or, as an extra feature, the Univus timeline, which any student, from any university can see. Users have the flexibility to choose the appropriate timeline for their message.

The app includes a validation feature: if a user attempts to post a message that's too short (less than three characters), a toast notification appears, indicating the requirement for a longer post. Once the user composes a suitable message, they can submit it. The message is processed by a Firebase function, which is responsible for posting it to the selected timeline.

**EditProfileScreen**
The EditProfileScreen in the app lets users update their profile, mainly their profile picture. When users visit this screen, they can change their profile pic. The screen starts by loading the user's current profile image from the database. If there's no image, it shows a default one.

Users can tap on the profile image to pick a new one from their phone's gallery. The app uses the Image Picker to let users choose an image. Once selected, the image is uploaded to Firebase storage. Then, the app updates the user's profile in Firebase with the new image URL.

**Firebase Functions**

In the app, Firebase functions play a crucial role in managing backend operations. These functions are separate pieces of code that run on Firebase servers. They are triggered by specific events or requests and handle various tasks, like updating databases, validating data, or processing user actions.
Data Addition and Validation: Some functions are designed to add data to Firebase's Firestore database. For example, they can take a list of universities from a file and upload it to the database. These functions include checks to prevent duplicate entries and ensure that only authorised requests are processed.
Other functions handle user-related tasks. They can create new user records in Firebase's Realtime Database, ensuring that user data like email, name, and role are valid and correctly formatted. They also manage user-specific actions, like setting a username, checking if it's already in use, and updating it in the database.
These functions are essential for maintaining the security and integrity of the app's data. They ensure that only valid and authed information is added or modified in the databases. This is crucial in maintaining a trustworthy and reliable app environment.
By running on Firebase servers, these functions execute automatically in response to specific triggers, making the backend processes efficient and scalable. They can handle requests without needing a user interface, making the app more responsive and faster.

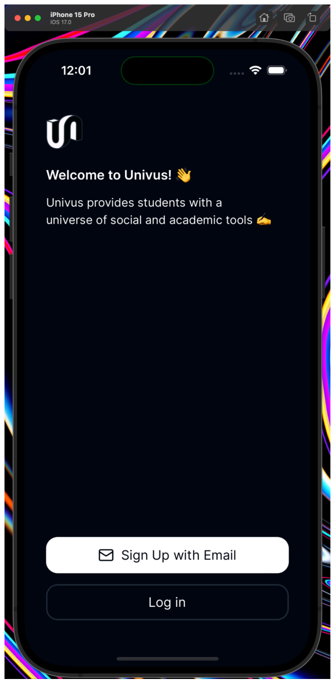
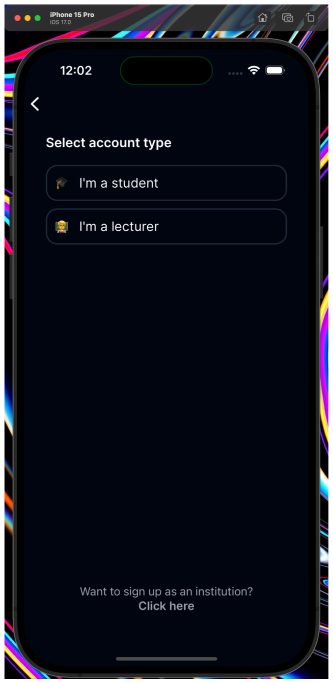
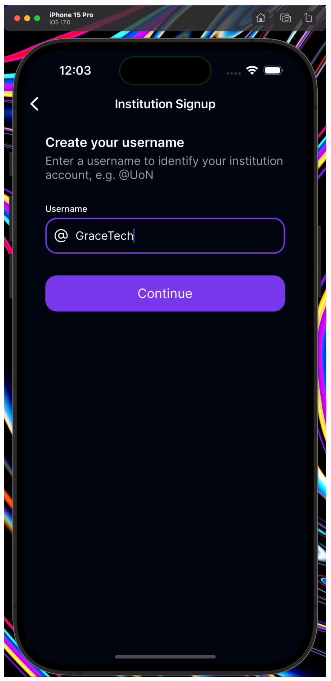
**And more…**
There is a lot more code than these files, but they all share a lot of the similarities of the files mentioned previously. For example, adding, updating, reading or deleting data from Firebase via Functions, just with different collections and documents, loading strategies and the general stuff you see in most screen files, such as using the re-usable components made to make the design nice, etc.

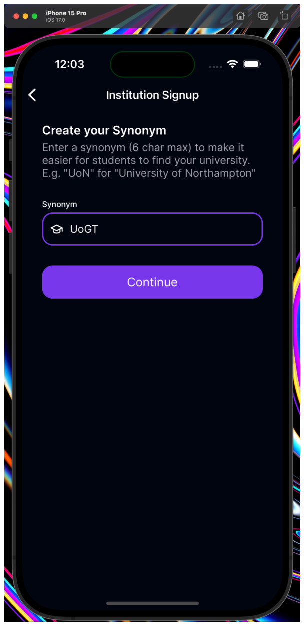
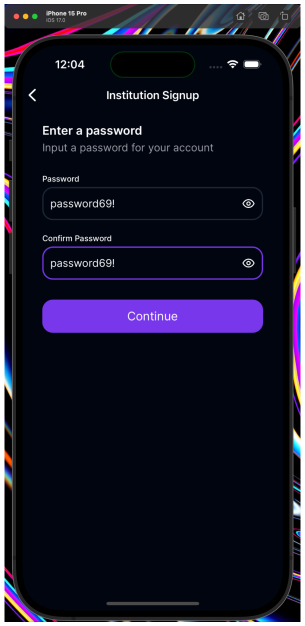
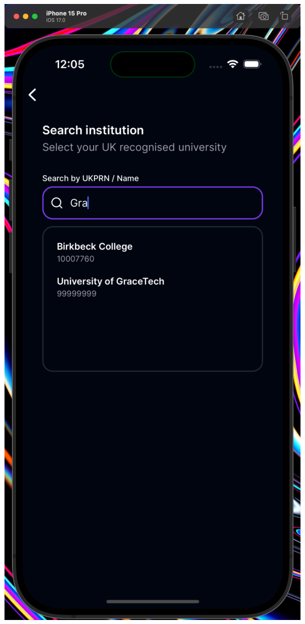
# Testing (+ Mandatory Feature Evidence + ALL DESIGNS)

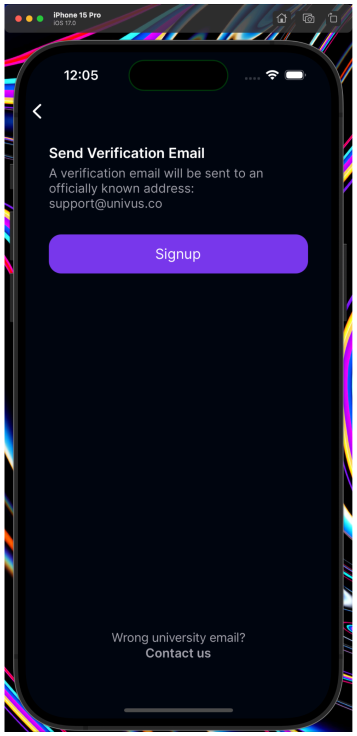
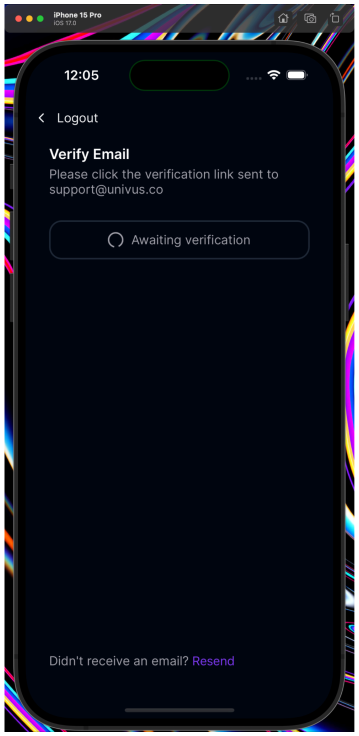
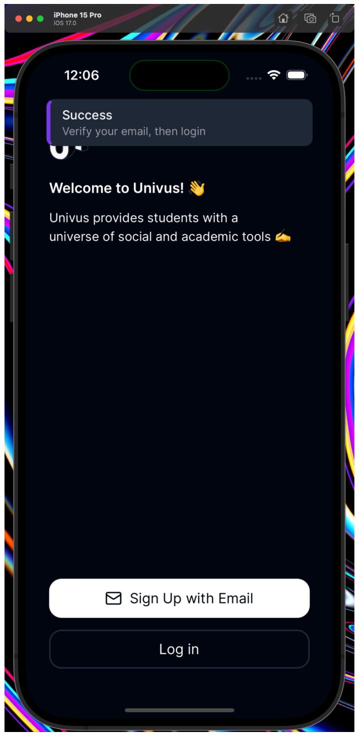
## React Native (iOS)

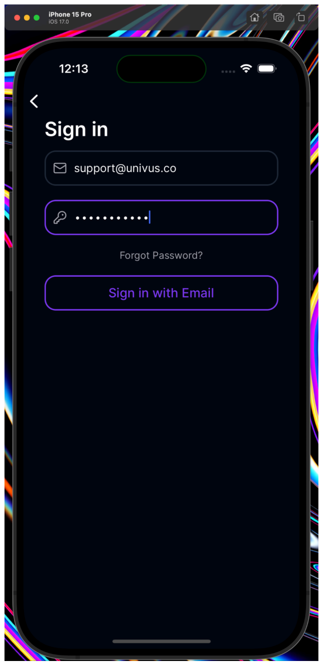
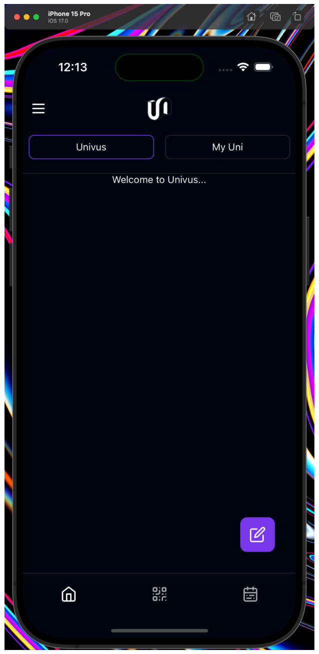
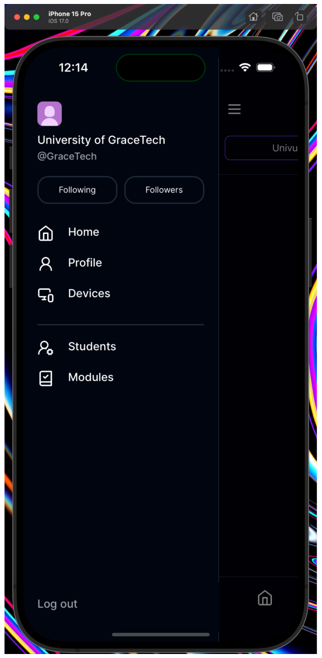
(NOTE: ANDROID WORKS FOR ALL SCREENS.)
( I mentioned there was a ‘few bugs’ in Android, but after the video, I fixed it! Android works perfectly!! :) )

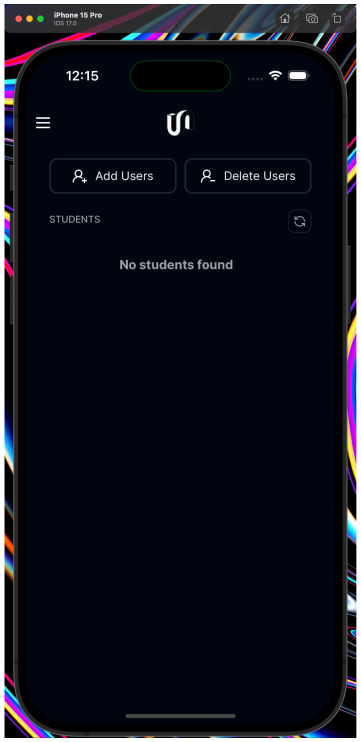
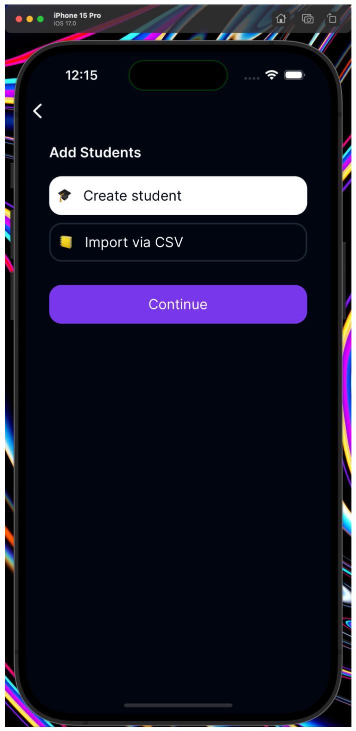
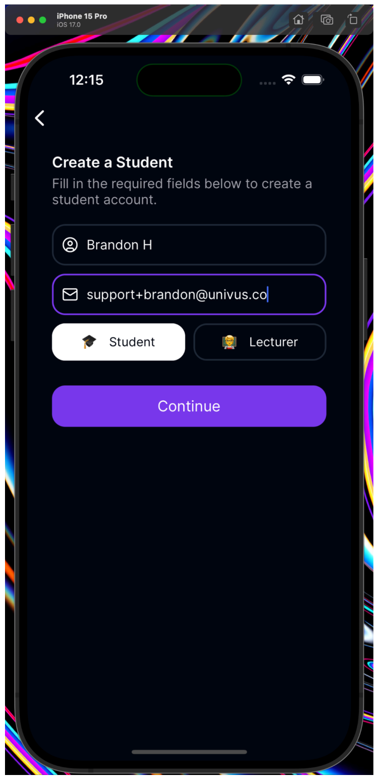
| Test | Input / Action | Expected Result | Actual Result | Success /Fail? | Screenshots |
|---|---|---|---|---|---|

| | | | | | |
|---|---|---|---|---|---|
| Institution Signup | @GraceTech<br>UoGT<br>password69!<br><br>University of GraceTech<br><br>support@univus.co | Username and Synonym should be checked successfully to ensure they dont exist already,<br><br>Password validation should be done client side to ensure match to confirm password,<br><br>"GraceTe" search on university should show GraceTech University,<br><br>support@univus.co (widely known institution email) should appear,<br><br>A verification email should be sent to the user, and return to welcome screen, indicating verification is required. | As expected | Success |  12:01 Welcome to Univus! 👋 Univus provides students with a universe of social and academic tools 🫴 Sign Up with Email Log in  12:02 Select account type 🎓 I'm a student 🧑‍🏫 I'm a lecturer Want to sign up as an institution? Click here  12:03 Institution Signup Create your username Enter a username to identify your institution account, e.g. @UoN Username @ GraceTech Continue |

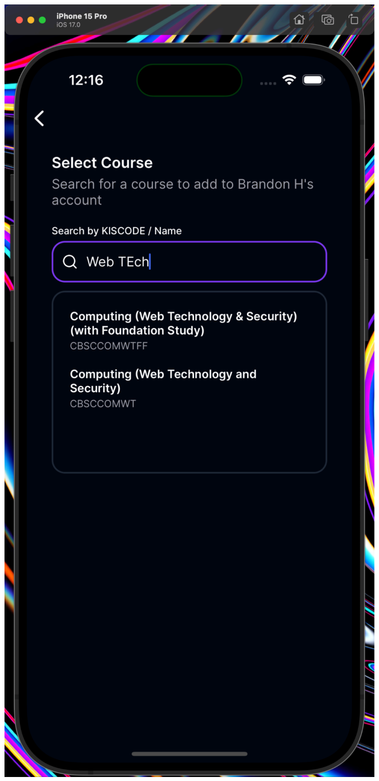
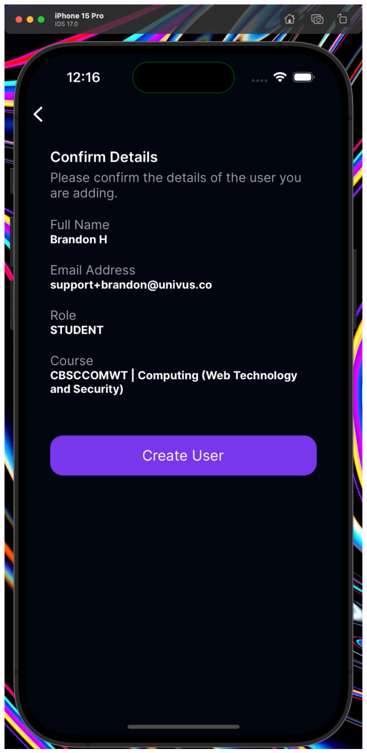
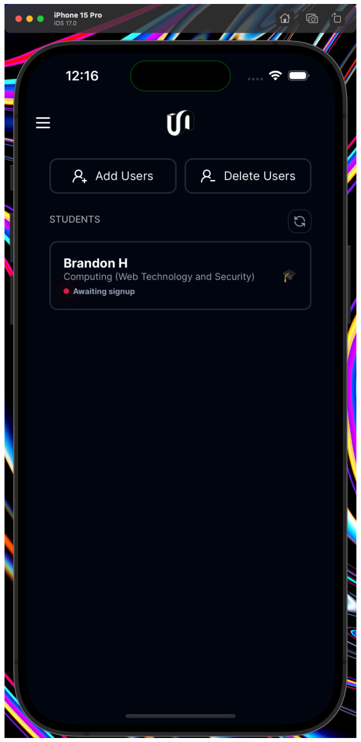
| | | | | | |
|---|---|---|---|---|---|
| | | | | |  iPhone 15 Pro iOS 17.0<br>12:03<br>Institution Signup<br>**Create your Synonym**<br>Enter a synonym (6 char max) to make it easier for students to find your university. E.g. "UoN" for "University of Northampton"<br>Synonym<br>UoGT<br>Continue<br> iPhone 15 Pro iOS 17.0<br>12:04<br>Institution Signup<br>**Enter a password**<br>Input a password for your account<br>Password<br>password69!<br>Confirm Password<br>password69!<br>Continue<br> iPhone 15 Pro iOS 17.0<br>12:05<br>**Search institution**<br>Select your UK recognised university<br>Search by UKPRN / Name<br>Gra<br>Birkbeck College<br>10007760<br>University of GraceTech<br>99999999 |

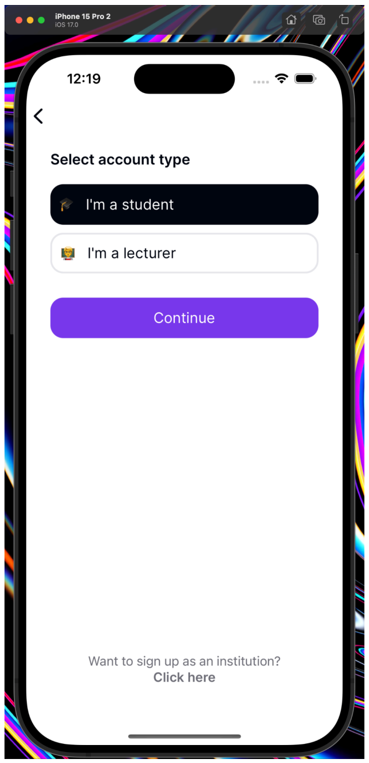
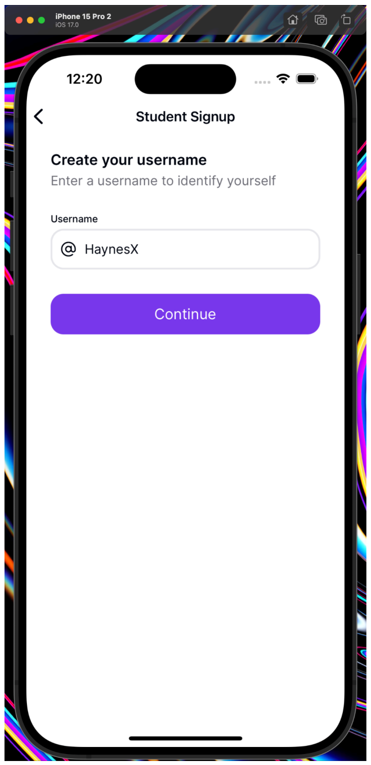
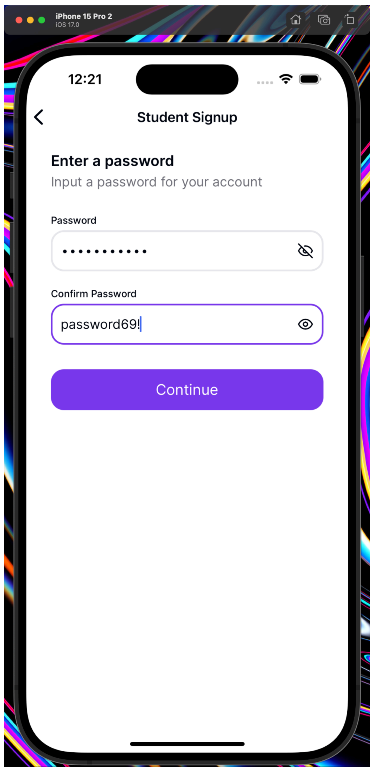
| | | | | | |
|---|---|---|---|---|---|
| | | | | | <br>iPhone 15 Pro<br>iOS 17.0<br>12:05<br>**Send Verification Email**<br>A verification email will be sent to an officially known address: support@univus.co<br>Signup<br>Wrong university email?<br>**Contact us**<br><br>iPhone 15 Pro<br>iOS 17.0<br>12:05<br>Logout<br>**Verify Email**<br>Please click the verification link sent to support@univus.co<br>Awaiting verification<br>Didn't receive an email? Resend<br><br>iPhone 15 Pro<br>iOS 17.0<br>12:06<br>Success<br>Verify your email, then login<br>**Welcome to Univus! 👋**<br>Univus provides students with a universe of social and academic tools ✍️<br>Sign Up with Email<br>Log in |

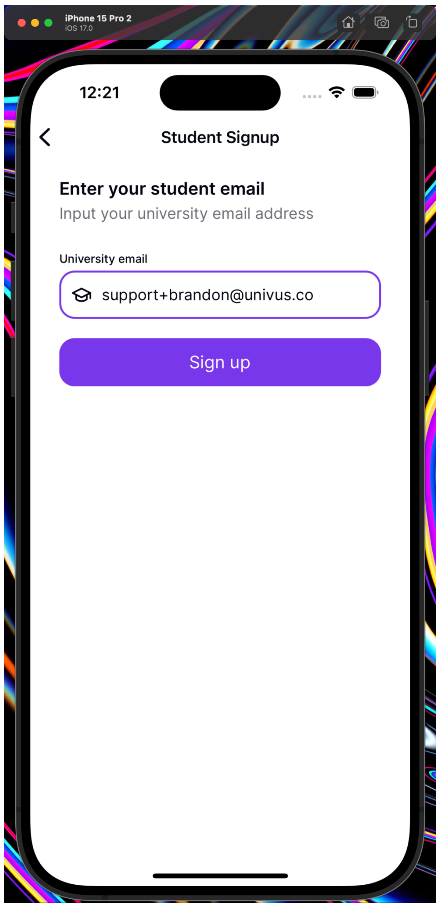
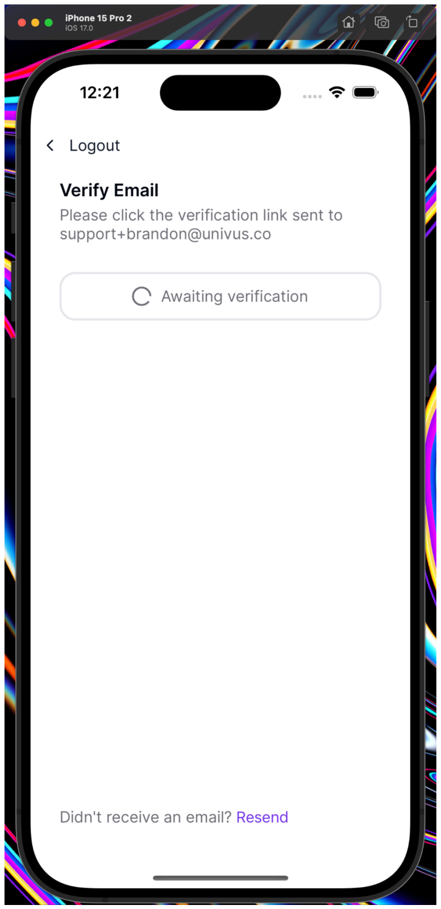
| | | | | | |
|---|---|---|---|---|---|
| Institution Login after Email Verification | support@univus.co<br>password69! | It should login and navigate to Main Stack (Home Screen) and show Students and Modules buttons inside of the Drawer | As expected | Success |  12:13<br>Sign in<br>support@univus.co<br>Forgot Password?<br>Sign in with Email  12:14<br>University of GraceTech<br>@GraceTech<br>Following<br>Followers<br>Home<br>Profile<br>Devices<br>Students<br>Modules<br>Log out |

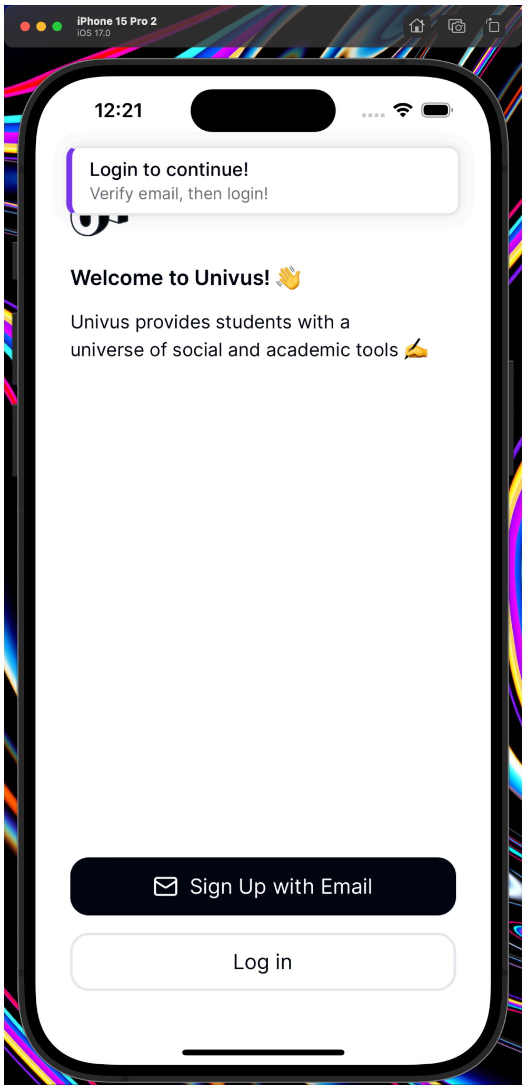
| | | | | | |
|---|---|---|---|---|---|
| **(MANDATORY**<br><br>Enrol a Student<br><br>+<br><br>Enrol a student on a course | Brandon H<br>support+brandon@univus.co<br><br>Student<br><br>Web Tech & Security Course | Create a temporary student in the realtime database (to save on read/write costs from firestore) with the data entered and show a screen with the user, with an indicator for them to manually sign up | As expected | Success |    |

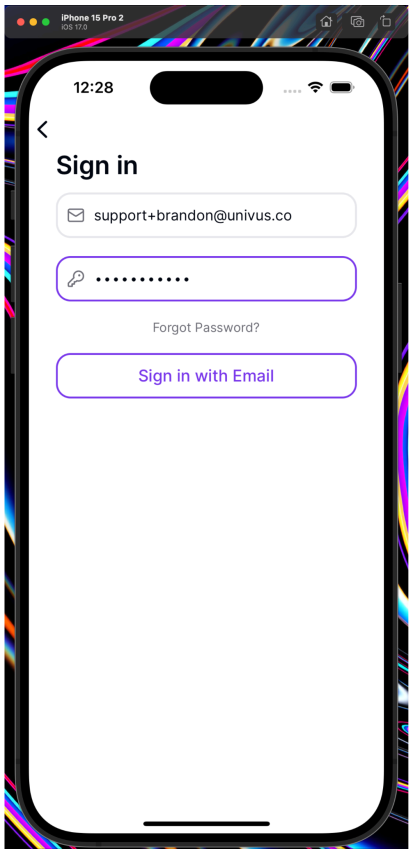
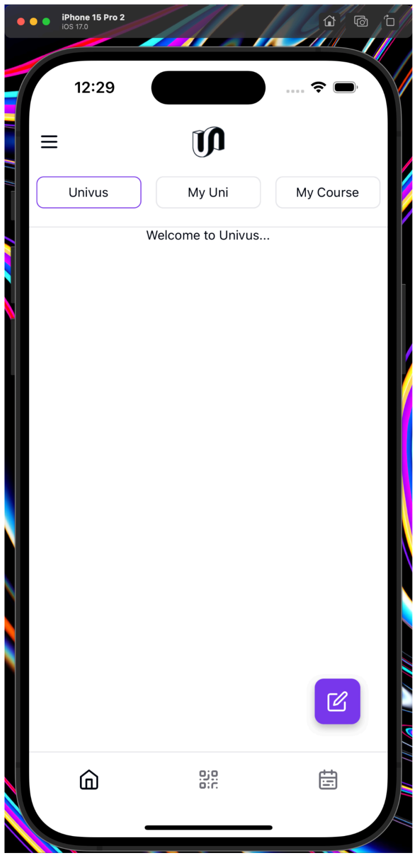
12:15
Create a Student
Fill in the required fields below to create a student account.
Brandon H
support+brandon@univus.co
Student
Lecturer
Continue

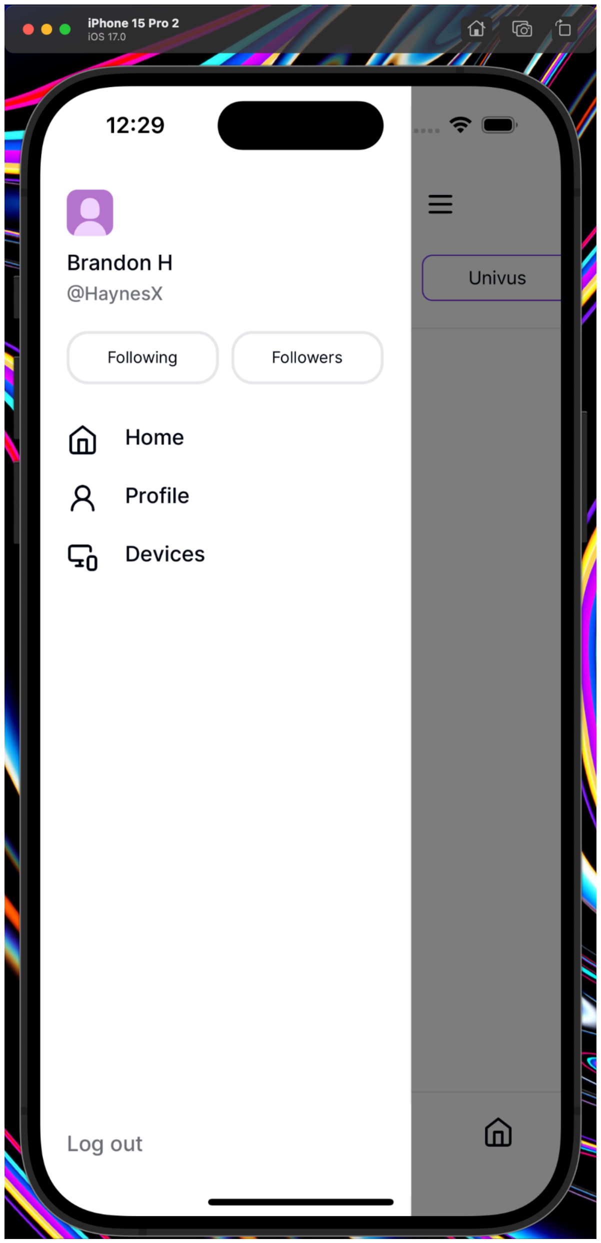
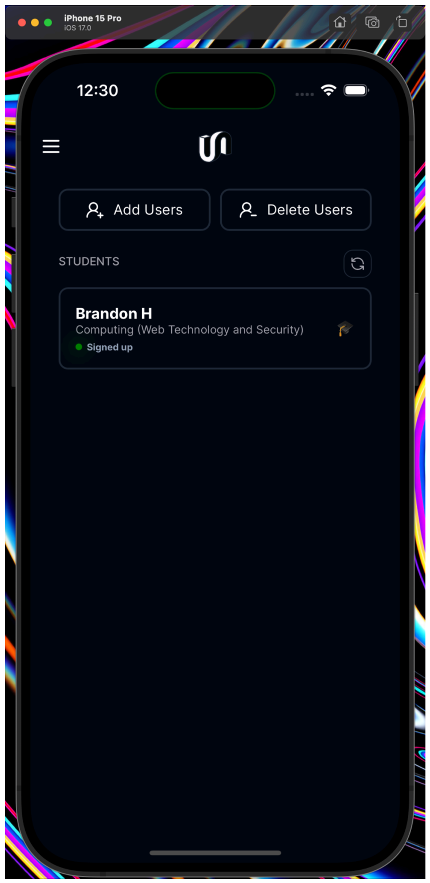
| | | | | | |
|---|---|---|---|---|---|
| | | | | | <br>iPhone 15 Pro<br>iOS 17.0<br>12:16<br>**Select Course**<br>Search for a course to add to Brandon H's account<br>Search by KISCODE / Name<br>Web TEch<br>**Computing (Web Technology & Security) (with Foundation Study)**<br>CBSCCOMWTFF<br>**Computing (Web Technology and Security)**<br>CBSCCOMWT<br><br>iPhone 15 Pro<br>iOS 17.0<br>12:16<br>**Confirm Details**<br>Please confirm the details of the user you are adding.<br>Full Name<br>**Brandon H**<br>Email Address<br>**support+brandon@univus.co**<br>Role<br>**STUDENT**<br>Course<br>**CBSCCOMWT \| Computing (Web Technology and Security)**<br>Create User<br><br>iPhone 15 Pro<br>iOS 17.0<br>12:16<br>Add Users<br>Delete Users<br>STUDENTS<br>**Brandon H**<br>Computing (Web Technology and Security)<br>● Awaiting signup |

| | | | | | |
|---|---|---|---|---|---|
| Student signup after enrolled from admin | Student<br><br>@HaynesX<br><br>support+brandon@univus.co<br><br>password69! | Should create student (user in firestore, and in auth, with custom claims) and send verification email to user. | As expected | Success |  12:19<br>**Select account type**<br>I'm a student<br>I'm a lecturer<br>Continue<br>Want to sign up as an institution? **Click here**  12:20<br>**Student Signup**<br>**Create your username**<br>Enter a username to identify yourself<br>Username: @ HaynesX<br>Continue  12:21<br>**Student Signup**<br>**Enter a password**<br>Input a password for your account<br>Password: •••••••••••<br>Confirm Password: password69!<br>Continue |

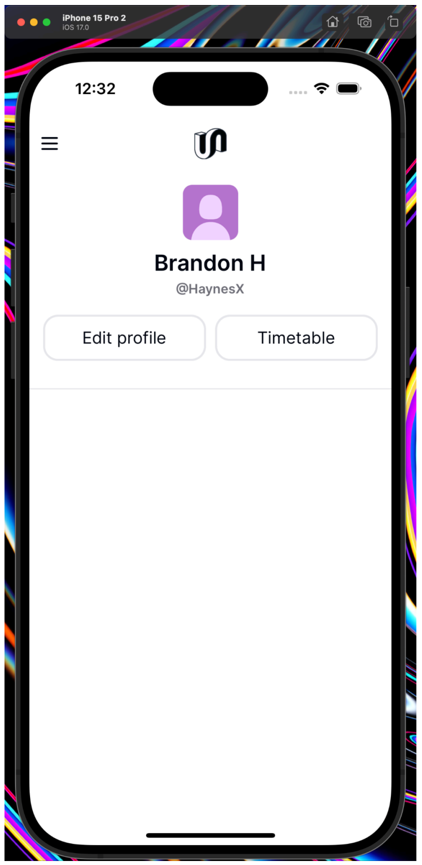
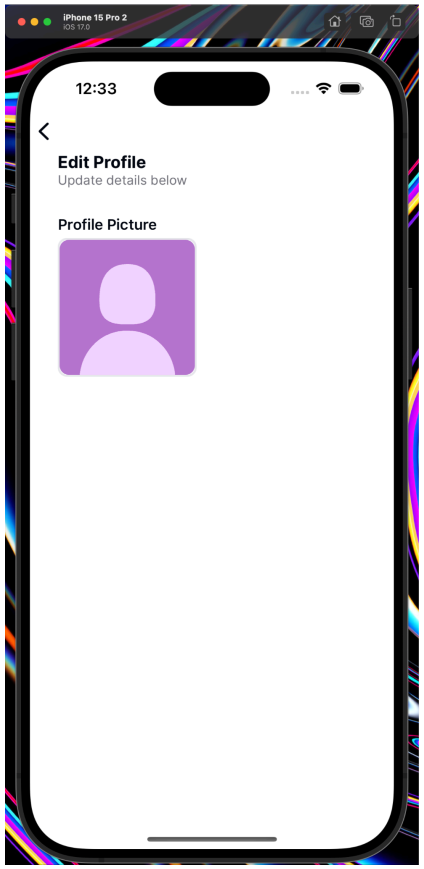
| | | | | | |
|---|---|---|---|---|---|
| | | | | |  iPhone 15 Pro 2 iOS 17.0 12:21 **Student Signup** **Enter your student email** Input your university email address University email support+brandon@univus.co Sign up  iPhone 15 Pro 2 iOS 17.0 12:21 Logout **Verify Email** Please click the verification link sent to support+brandon@univus.co Awaiting verification Didn't receive an email? Resend |

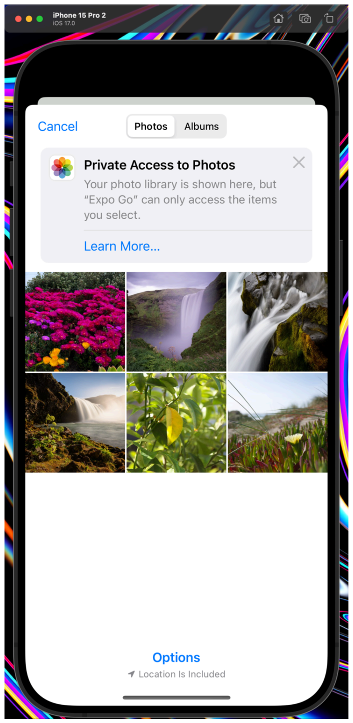
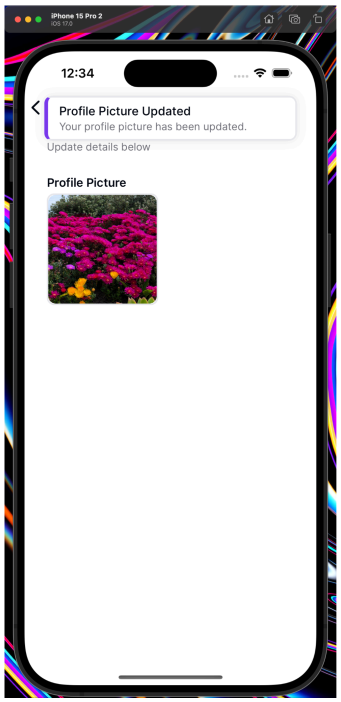
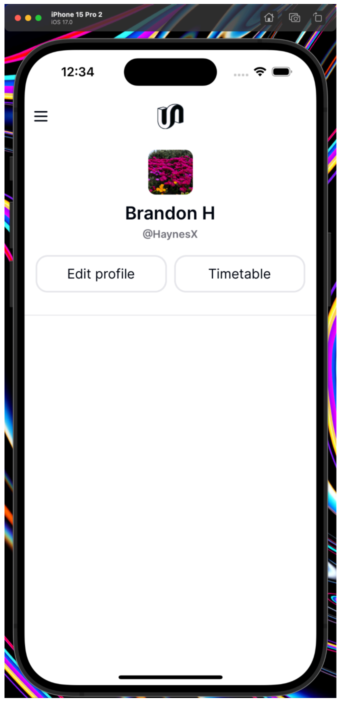
| | | | | | |
|---|---|---|---|---|---|
| | | | | |  iPhone 15 Pro 2 iOS 17.0 12:21 Login to continue! Verify email, then login! Welcome to Univus! 👋 Univus provides students with a universe of social and academic tools ✍️ |

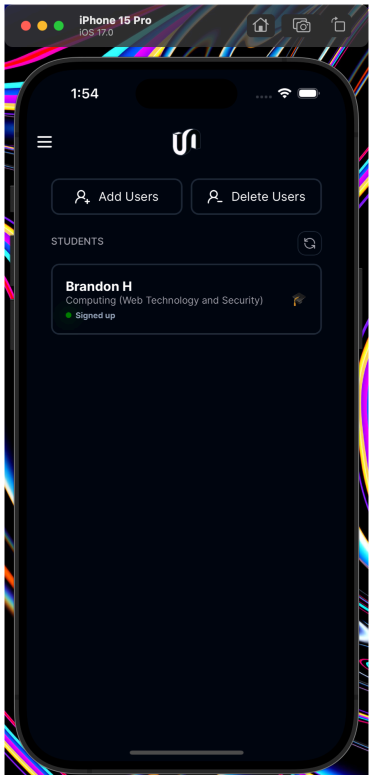
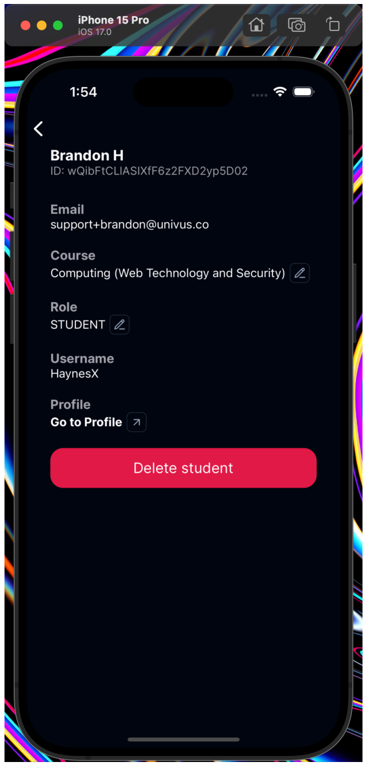
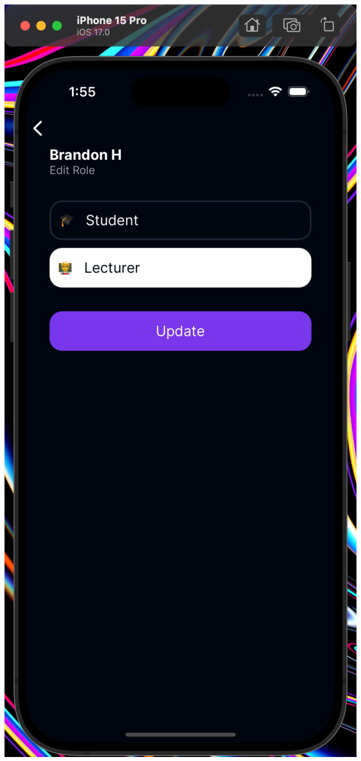
|  |  |  |  |  |  |
|---|---|---|---|---|---|
| Login as Student after verifying email | support+brandon@univus.co<br><br>password69! | Should login the user, show all timeline (course, global and super global, AKA, course, my uni, and univus) and have no admin related stuff in the drawer.<br><br>On admin screen, should show they are signed up. | As expected | Success |  12:28 Sign in support+brandon@univus.co ••••••••••• Forgot Password? Sign in with Email  12:29 Univus My Uni My Course Welcome to Univus... |

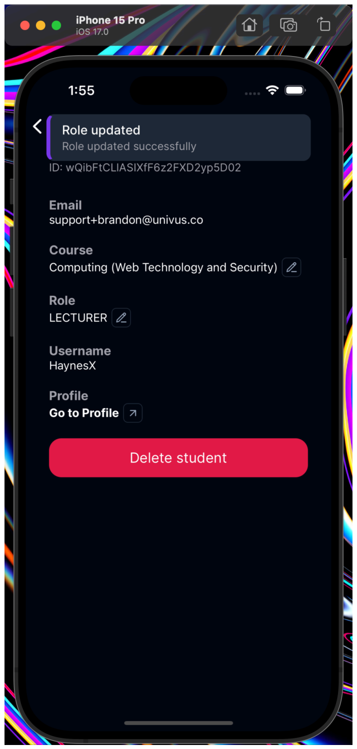
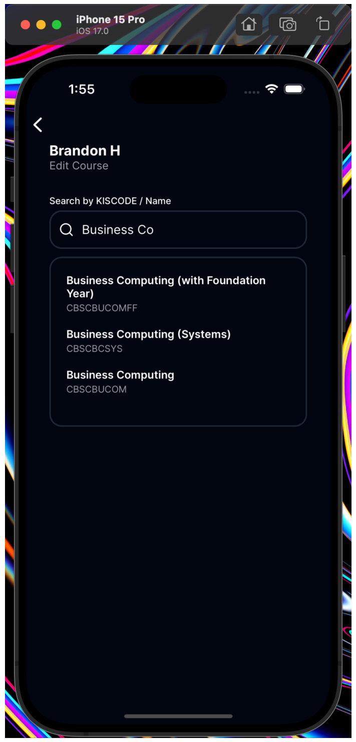
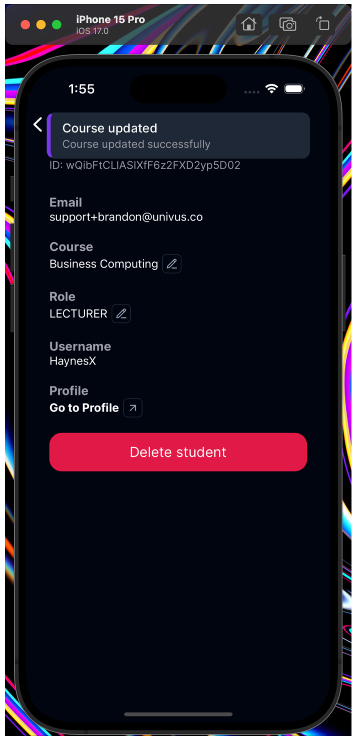
| | | | | | |
|---|---|---|---|---|---|
| | | | | |   |

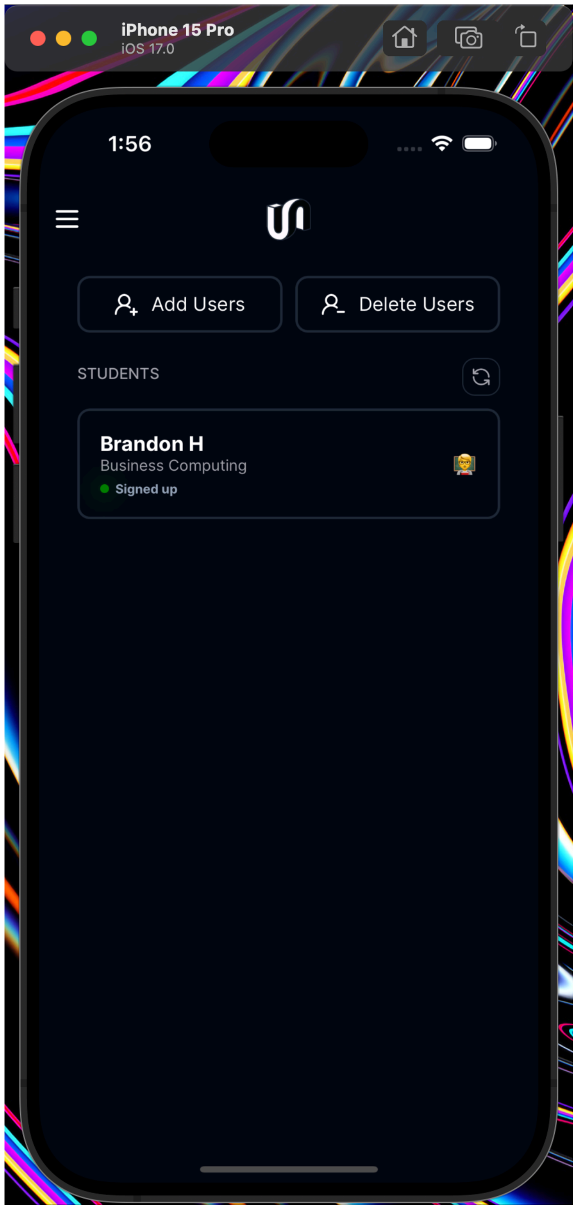
iPhone 15 Pro 2
iOS 17.0
12:29
Brandon H
@HaynesX
Following
Followers
Home
Profile
Devices
Univus
Log out
iPhone 15 Pro
iOS 17.0
12:30
Add Users
Delete Users
STUDENTS
Brandon H
Computing (Web Technology and Security)
Signed up

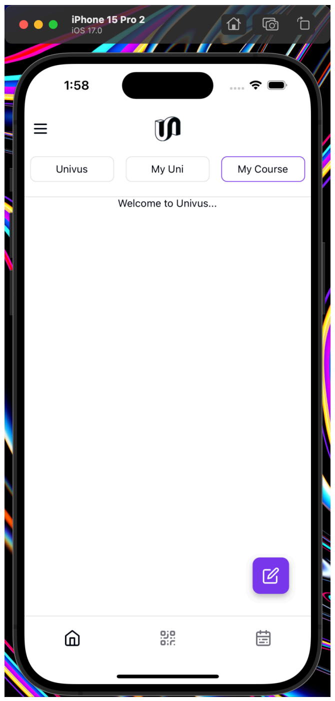
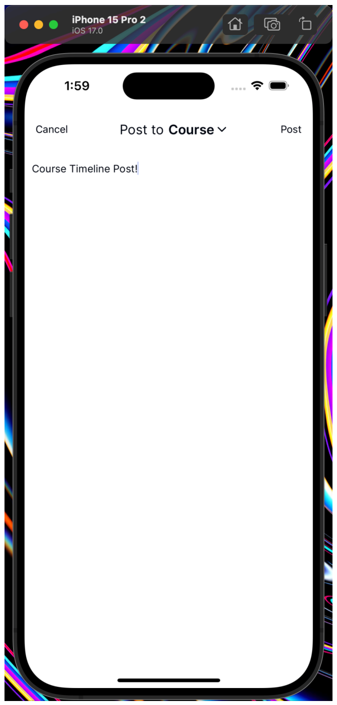
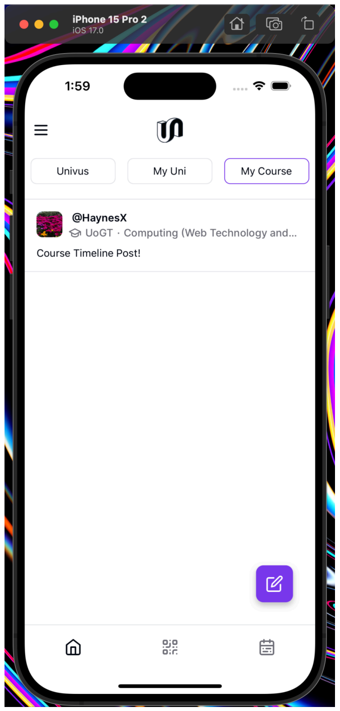
| | | | | | |
|---|---|---|---|---|---|
| **(MANDATORY)** Student can update their profile details | New profile picture | Should upload to Firebase Storage and change user profile picture on user doc and firebase auth token | As expected | success |   iPhone 15 Pro 2<br>iOS 17.0<br>12:32<br>Brandon H<br>@HaynesX<br>Edit profile<br>Timetable   iPhone 15 Pro 2<br>iOS 17.0<br>12:33<br>Edit Profile<br>Update details below<br>Profile Picture |

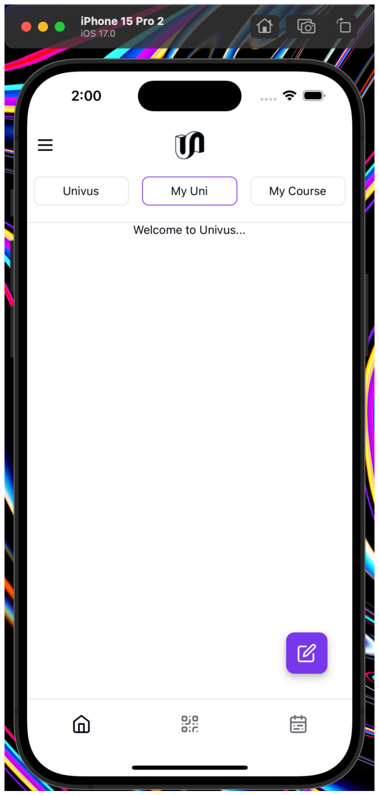
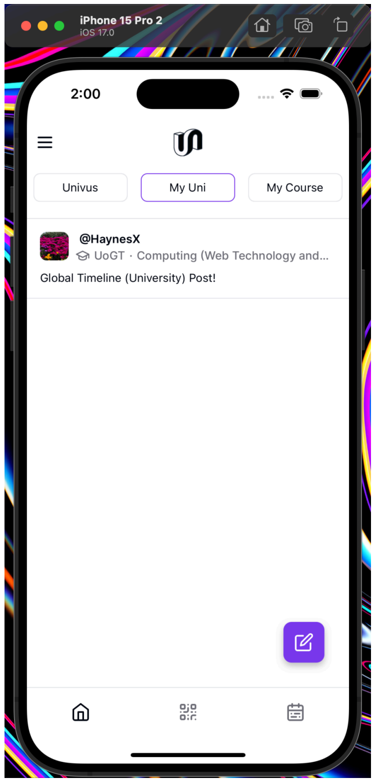
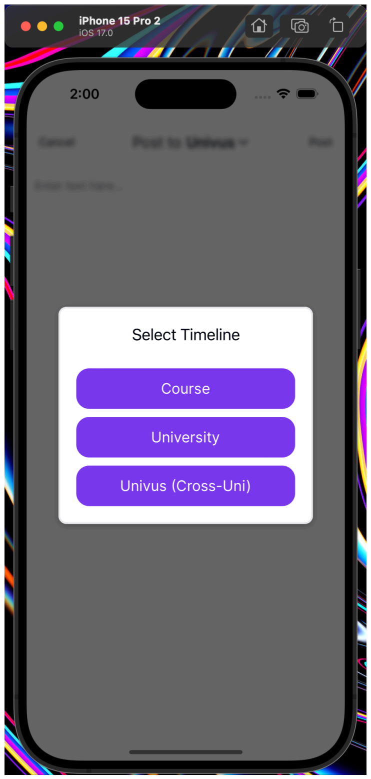
| | | | | | |
|---|---|---|---|---|---|
| | | | | |  iPhone 15 Pro 2<br>iOS 17.0<br>Cancel<br>Photos<br>Albums<br>Learn More...<br>Options<br>Location Is IncludedPrivate Access to Photos<br>Your photo library is shown here, but "Expo Go" can only access the items you select.<br> Profile Picture Updated<br>Your profile picture has been updated.<br>Update details below<br>Profile Picture<br> iPhone 15 Pro 2<br>iOS 17.0<br>12:34<br>Edit profile<br>TimetableBrandon H<br>@HaynesX |

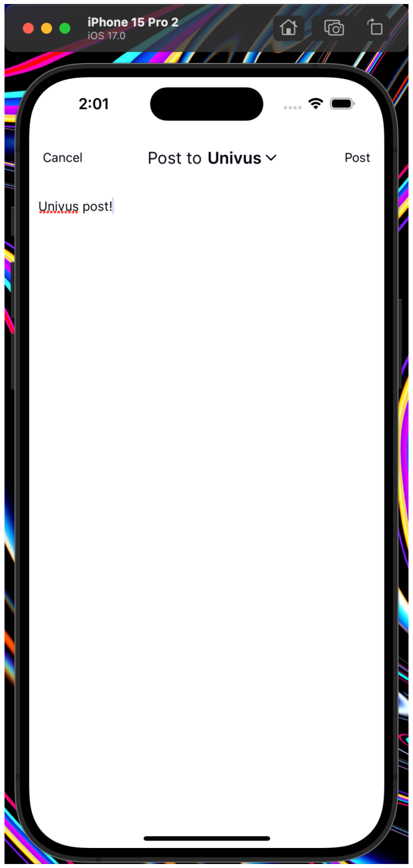
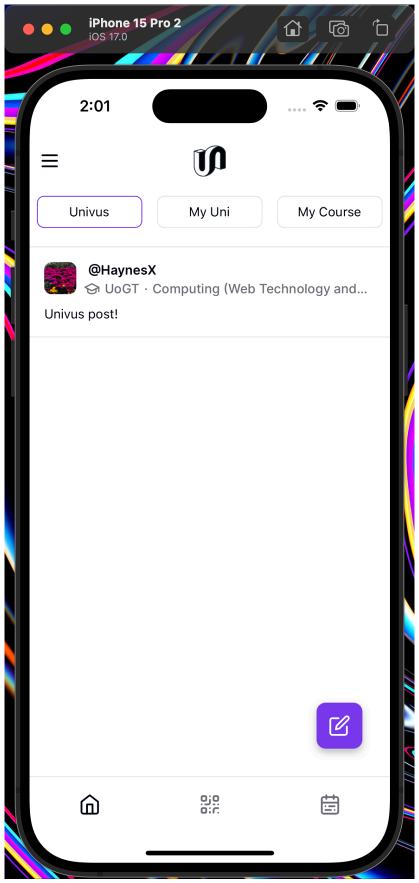
| | | | | | |
|---|---|---|---|---|---|
| **(MANDATORY)**<br><br>**Admin can update student details**<br><br>**Admin can read student details** | STUDENT to LECTURER<br><br>WEB TECH to BUSINESS COMPUTING<br><br>And can view | Change details and refresh screen with new details | As expected | success |  iPhone 15 Pro iOS 17.0<br>1:54<br>Add Users · Delete Users<br>STUDENTS<br>Brandon H<br>Computing (Web Technology and Security)<br>Signed up<br> iPhone 15 Pro iOS 17.0<br>1:54<br>Brandon H<br>ID: wQibFtCLIASIXfF6z2FXD2yp5D02<br>Email<br>support+brandon@univus.co<br>Course<br>Computing (Web Technology and Security)<br>Role<br>STUDENT<br>Username<br>HaynesX<br>Profile<br>Go to Profile<br>Delete student<br> iPhone 15 Pro iOS 17.0<br>1:55<br>Brandon H<br>Edit Role<br>Student<br>Lecturer<br>Update |

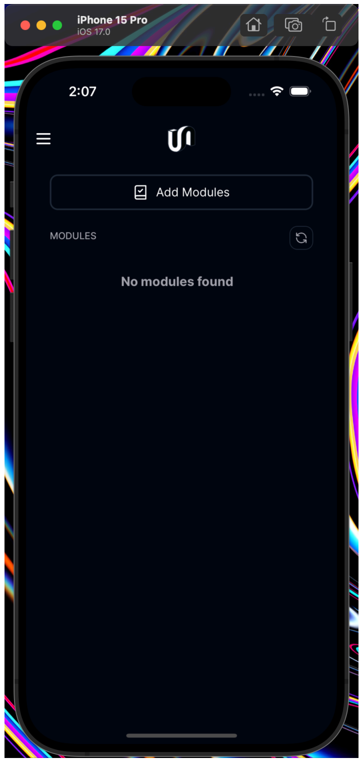
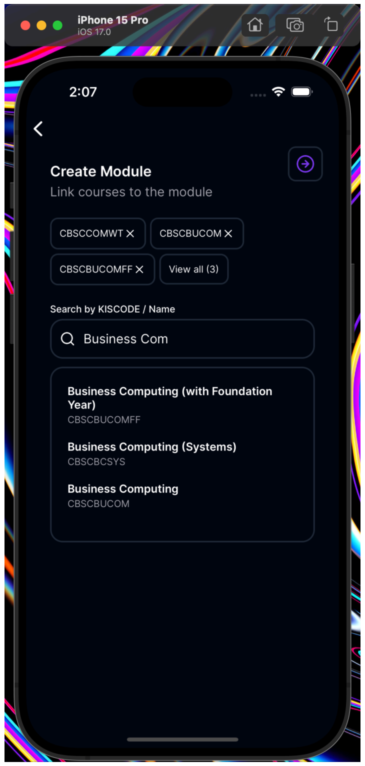
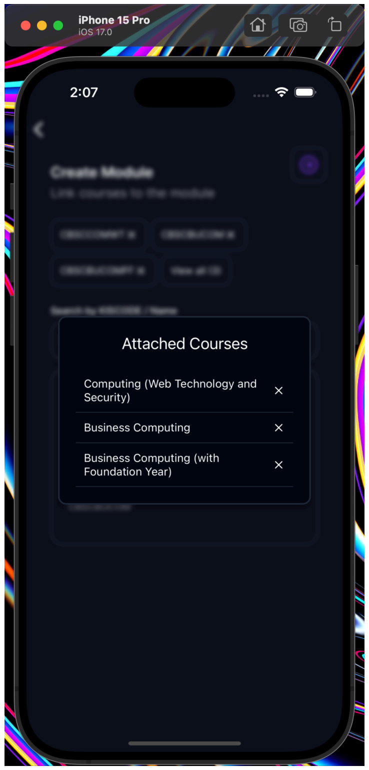
| | | | | | |
|---|---|---|---|---|---|
| | | | | |  iPhone 15 Pro<br>iOS 17.0<br>1:55<br>Role updated<br>Role updated successfully<br>ID: wQibFtCLIASIXfF6z2FXD2yp5D02<br>**Email**<br>support+brandon@univus.co<br>**Course**<br>Computing (Web Technology and Security)<br>**Role**<br>LECTURER<br>**Username**<br>HaynesX<br>**Profile**<br>**Go to Profile**<br>Delete student<br> iPhone 15 Pro<br>iOS 17.0<br>1:55<br>**Brandon H**<br>Edit Course<br>Search by KISCODE / Name<br>Business Co<br>**Business Computing (with Foundation Year)**<br>CBSCBUCOMFF<br>**Business Computing (Systems)**<br>CBSCBCSYS<br>**Business Computing**<br>CBSCBUCOM<br> iPhone 15 Pro<br>iOS 17.0<br>1:55<br>Course updated<br>Course updated successfully<br>ID: wQibFtCLIASIXfF6z2FXD2yp5D02<br>**Email**<br>support+brandon@univus.co<br>**Course**<br>Business Computing<br>**Role**<br>LECTURER<br>**Username**<br>HaynesX<br>**Profile**<br>**Go to Profile**<br>Delete student |

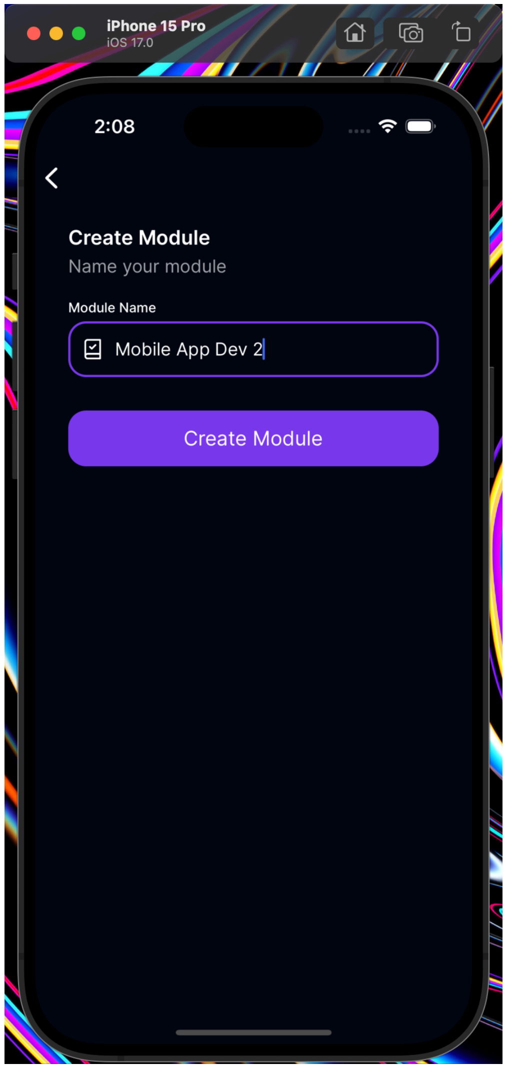
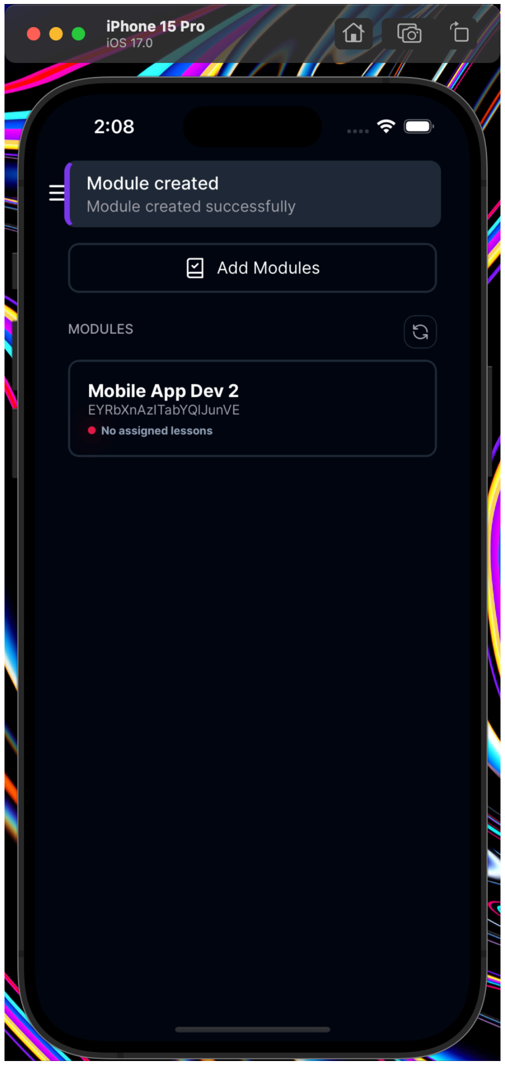
|  |  |  |  |  |  |
|---|---|---|---|---|---|
|  |  |  |  |  |  iPhone 15 Pro iOS 17.0 1:56 Add Users Delete Users STUDENTS Brandon H Business Computing Signed up |

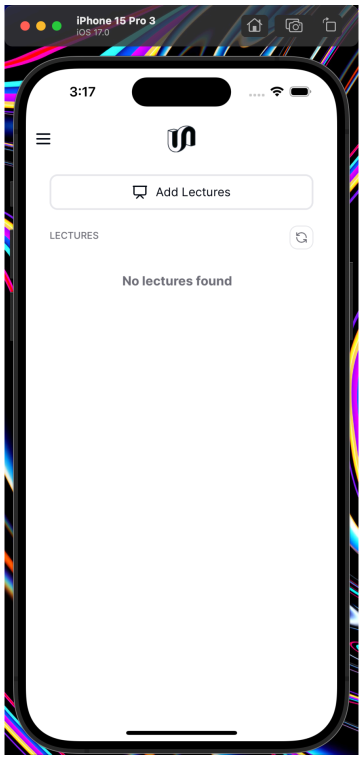
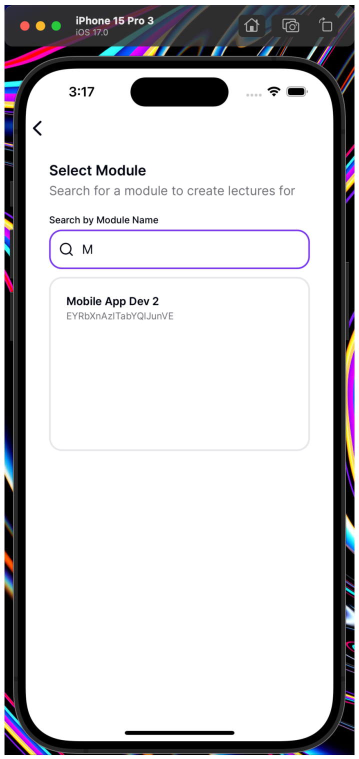
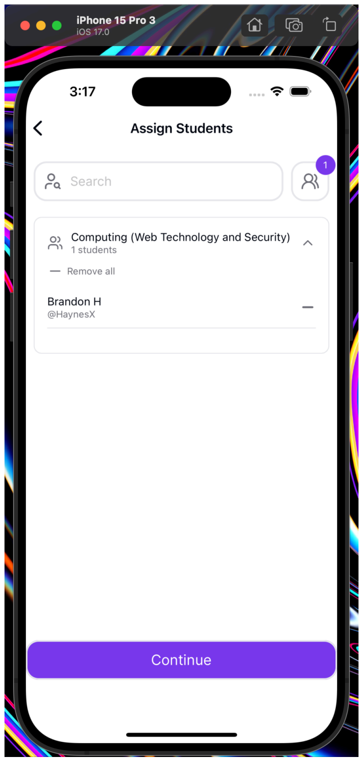
| | | | | | |
|---|---|---|---|---|---|
| **(MANDATORY)**<br><br>**Student can post on global and course timeline**<br><br>**Student can also post on cross-uni timeline (extra feature)** | Course: “Course timeline post:<br><br>Global: “Global Timeline (University) Post”<br><br>—--<br><br>Univus (Cross-Uni): “Univus post!” | Should post on all timelines and refresh automatically | As expected | Sucess |  iPhone 15 Pro 2<br>iOS 17.0<br>1:58<br>Univus<br>My Uni<br>My CourseWelcome to Univus...<br> iPhone 15 Pro 2<br>iOS 17.0<br>1:59<br>Cancel<br>Post to Course<br>PostCourse Timeline Post!<br> iPhone 15 Pro 2<br>iOS 17.0<br>1:59<br>Univus<br>My Uni<br>My Course@HaynesX<br>UoGT · Computing (Web Technology and...<br>Course Timeline Post! |

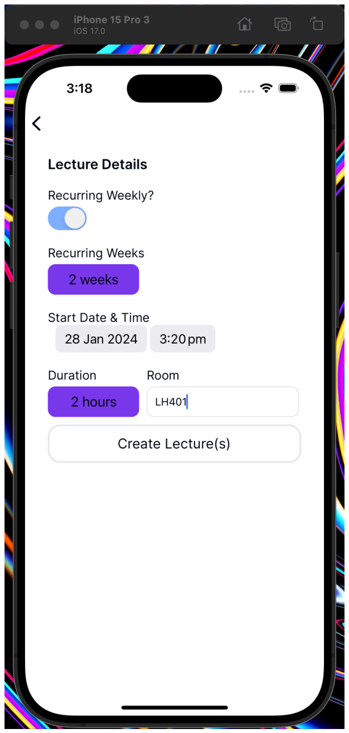
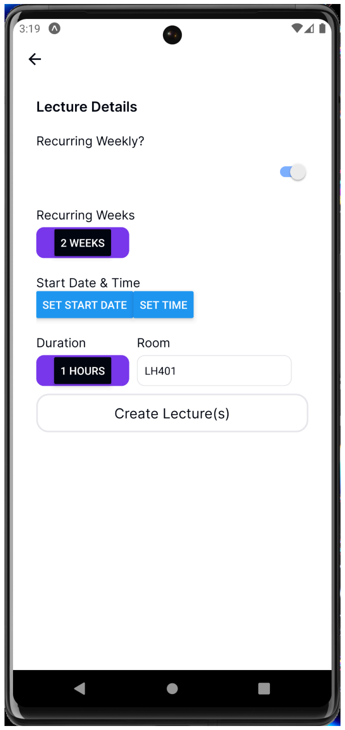
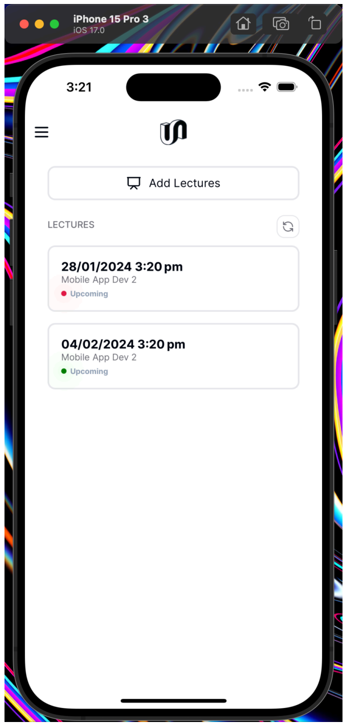
| | | | | | |
|---|---|---|---|---|---|
| | | | | |  iPhone 15 Pro 2<br>iOS 17.0<br>2:00<br>Univus<br>My Uni<br>My Course Welcome to Univus...  iPhone 15 Pro 2<br>iOS 17.0<br>2:00<br>Univus<br>My Uni<br>My Course @HaynesX UoGT · Computing (Web Technology and... Global Timeline (University) Post!  iPhone 15 Pro 2<br>iOS 17.0<br>2:00<br>Select Timeline<br>Course<br>University<br>Univus (Cross-Uni) |

|  |  |  |  |  |  |
|---|---|---|---|---|---|
|  |  |  |  |  |   iPhone 15 Pro 2<br>iOS 17.0<br>2:01<br>Cancel<br>Post to Univus<br>Post<br>Univus post!   iPhone 15 Pro 2<br>iOS 17.0<br>2:01<br>Univus<br>My Uni<br>My Course<br>@HaynesX<br>UoGT · Computing (Web Technology and...<br>Univus post! |

| | | | | | |
|---|---|---|---|---|---|
| Admin - Create a Module | Link Courses:<br><br>- Computing Web Tech & Security<br><br>- Business Computing<br><br>- Business Computing (with Foundation Year)<br><br>—<br><br>Name of Module: Mobile App Dev 2 | Link multiple courses, as well as able to show modal in 'View all', and create module and show on modules screen with refresh. | As expected | Success |   iPhone 15 Pro<br>iOS 17.0<br>2:07<br>Add Modules<br>MODULES<br>No modules found   iPhone 15 Pro<br>iOS 17.0<br>2:07<br>Create Module<br>Link courses to the module<br>CBSCCOMWT<br>CBSCBUCOM<br>CBSCBUCOMFF<br>View all (3)<br>Search by KISCODE / Name<br>Business Com<br>Business Computing (with Foundation Year)<br>CBSCBUCOMFF<br>Business Computing (Systems)<br>CBSCBCSYS<br>Business Computing<br>CBSCBUCOM   iPhone 15 Pro<br>iOS 17.0<br>2:07<br>Attached Courses<br>Computing (Web Technology and Security)<br>Business Computing<br>Business Computing (with Foundation Year) |

| | | | | | |
|---|---|---|---|---|---|
| | | | | |  iPhone 15 Pro<br>iOS 17.0<br>2:08<br>Create Module<br>Name your module<br>Module Name<br>Mobile App Dev 2<br>  iPhone 15 Pro<br>iOS 17.0<br>2:08<br>Module created<br>Module created successfully<br> <br>MODULES<br>Mobile App Dev 2<br>EYRbXnAzlTabYQlJunVE<br>● No assigned lessons |

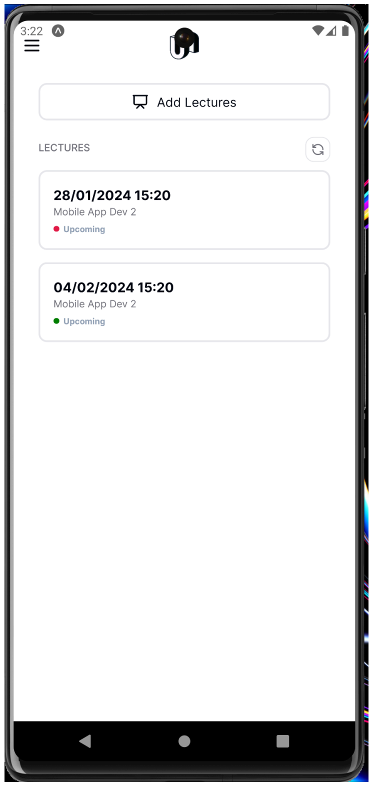
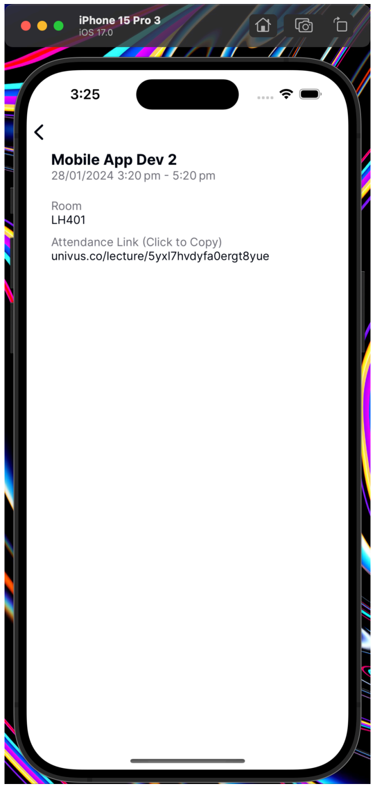
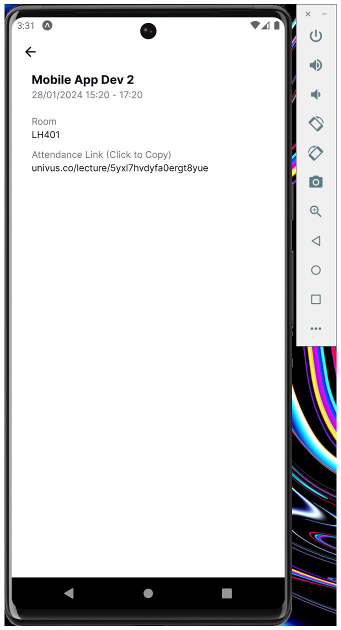
| | | | | | |
|---|---|---|---|---|---|
| Create Lessons Recurring<br><br>**(NOTE: IN THE VIDEO, I MENTIONED ANDROID HAD SOME BUGS - THIS WAS IT. I FIXED IT. IT WAS A DATE TIME ISSUE)**<br><br>**:)** | 28/01/2024 3:20PM<br>2 weeks (Recurring)<br><br>Brandon H Student added which was in Web Tech & Security<br><br>Mobile App Dev2 Module | Add 2 lessons, each with their own lecture link/code | As expected | Success | <br>iPhone 15 Pro 3<br>iOS 17.0<br>3:17<br>Add Lectures<br>LECTURES<br>No lectures found<br><br>iPhone 15 Pro 3<br>iOS 17.0<br>3:17<br>Select Module<br>Search for a module to create lectures for<br>Search by Module Name<br>M<br>Mobile App Dev 2<br>EYRbXnAzITabYQIJunVE<br><br>iPhone 15 Pro 3<br>iOS 17.0<br>3:17<br>Assign Students<br>Search<br>1<br>Computing (Web Technology and Security)<br>1 students<br>Remove all<br>Brandon H<br>@HaynesX<br>Continue |

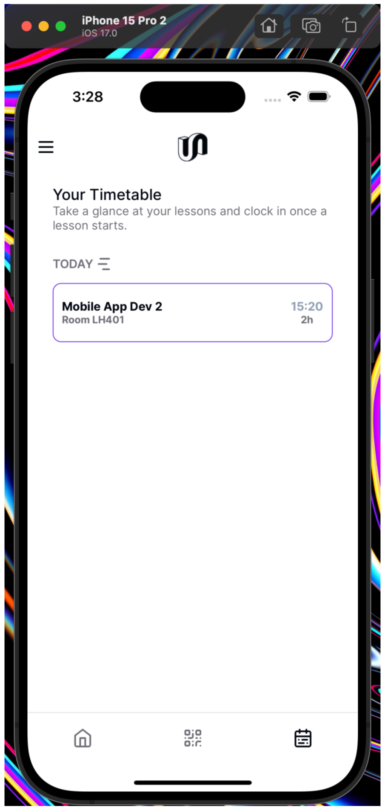
| | | | | | |
|---|---|---|---|---|---|
| | | | | |    |

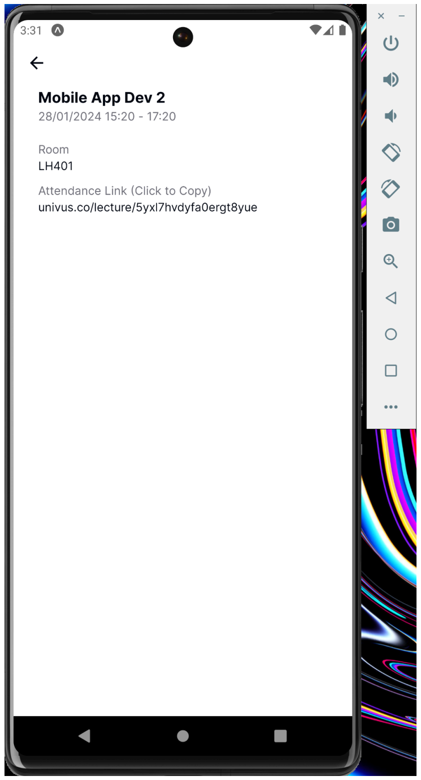
iPhone 15 Pro 3
iOS 17.0
3:18
Lecture Details
Recurring Weekly?
Recurring Weeks
2 weeks
Start Date & Time
28 Jan 2024
3:20 pm
Duration
Room
2 hours
LH401
Create Lecture(s)

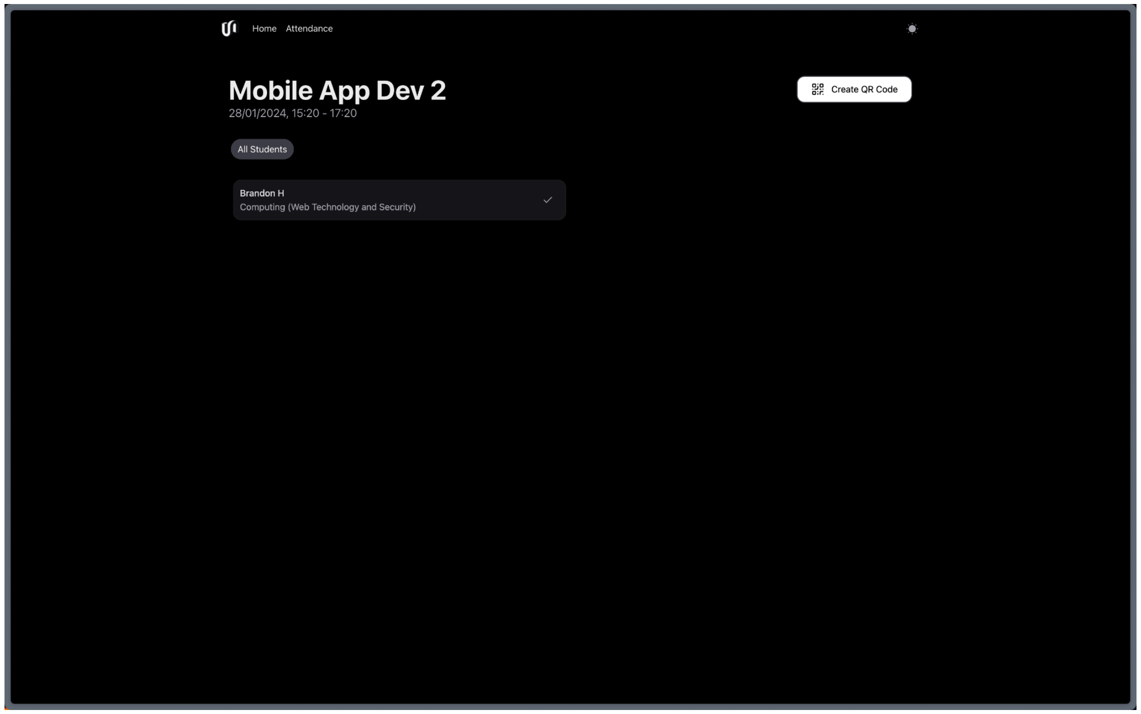
3:19
Lecture Details
Recurring Weekly?
Recurring Weeks
2 WEEKS
Start Date & Time
SET START DATE
SET TIME
Duration
Room
1 HOURS
LH401
Create Lecture(s)

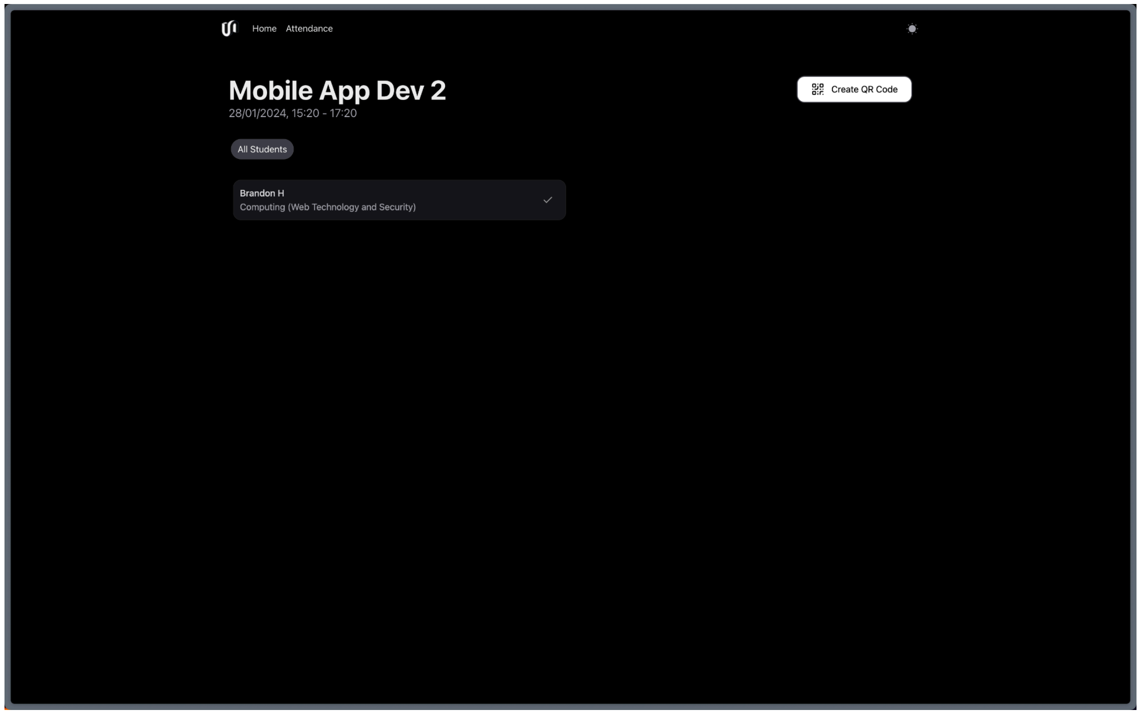
iPhone 15 Pro 3
iOS 17.0
3:21
Add Lectures
LECTURES
28/01/2024 3:20 pm
Mobile App Dev 2
Upcoming
04/02/2024 3:20 pm
Mobile App Dev 2
Upcoming

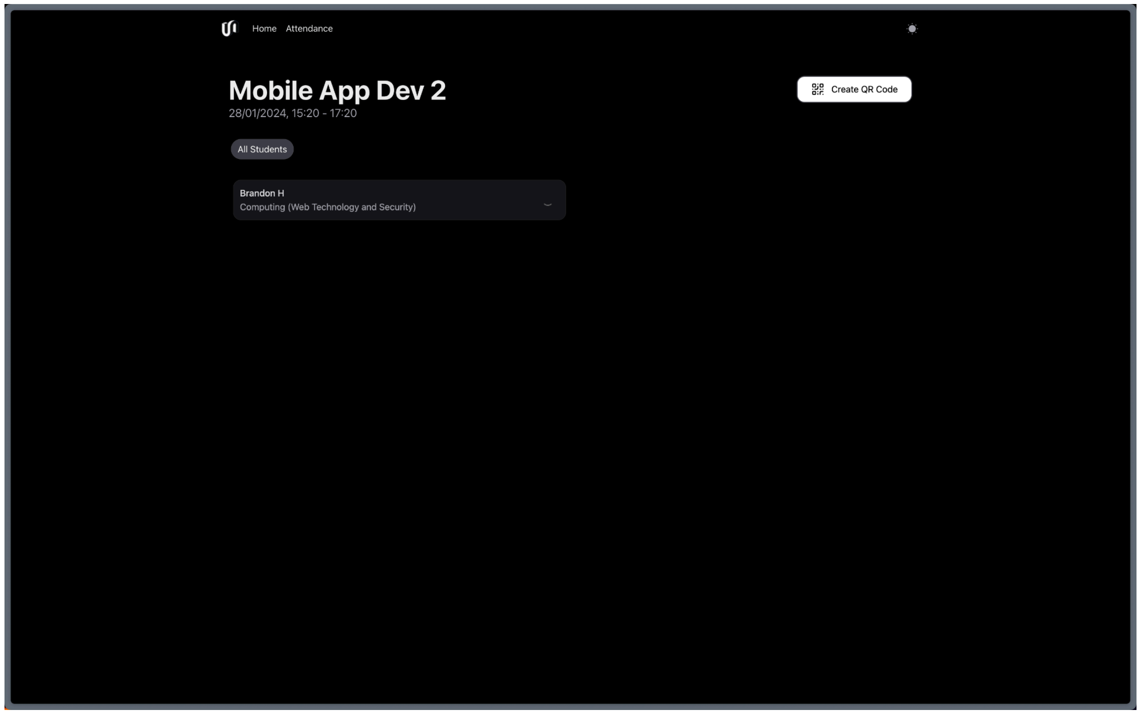
| | | | | | |
|---|---|---|---|---|---|
| | | | | |  3:22<br>Add Lectures<br>LECTURES<br>28/01/2024 15:20<br>Mobile App Dev 2<br>Upcoming<br>04/02/2024 15:20<br>Mobile App Dev 2<br>Upcoming<br> iPhone 15 Pro 3<br>iOS 17.0<br>3:25<br>Mobile App Dev 2<br>28/01/2024 3:20 pm - 5:20 pm<br>Room<br>LH401<br>Attendance Link (Click to Copy)<br>univus.co/lecture/5yxl7hvdyfa0ergt8yue<br> 3:31<br>Mobile App Dev 2<br>28/01/2024 15:20 - 17:20<br>Room<br>LH401<br>Attendance Link (Click to Copy)<br>univus.co/lecture/5yxl7hvdyfa0ergt8yue |

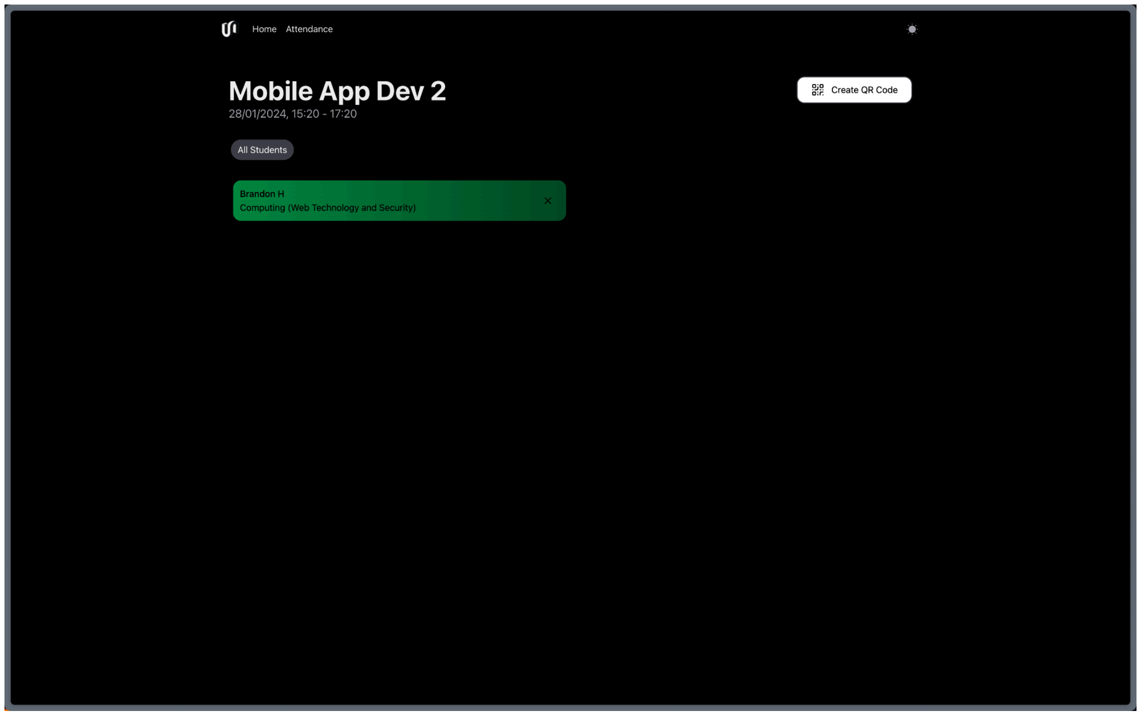
| | | | | | |
|---|---|---|---|---|---|
| View Lessons as Student | | View Lesson made earlier in test, highlighted purple border to indicate it is currently lesson time, which it was | As expected | success |  iPhone 15 Pro 2 iOS 17.0 3:28 Your Timetable Take a glance at your lessons and clock in once a lesson starts. TODAY Mobile App Dev 2 Room LH401 15:20 2h |

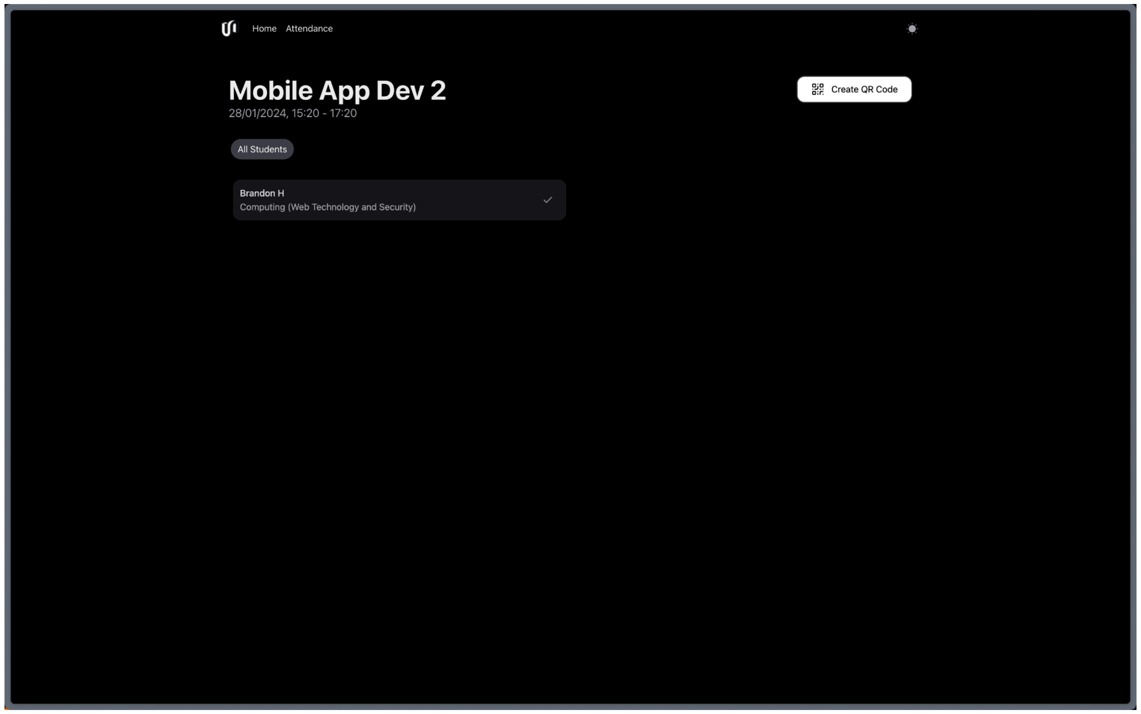
|  |  |  |  |  |  |
|---|---|---|---|---|---|
| View Attendance of Students<br><br>(WEB APP) | https://univus.co<br><br>https://univus.co/lecture/5yxl7hvdyfa0ergt8yue | Copy link, go to site on website and view students | As expected | Success |  3:31<br>**Mobile App Dev 2**<br>28/01/2024 15:20 - 17:20<br>Room<br>LH401<br>Attendance Link (Click to Copy)<br>univus.co/lecture/5yxl7hvdyfa0ergt8yue |

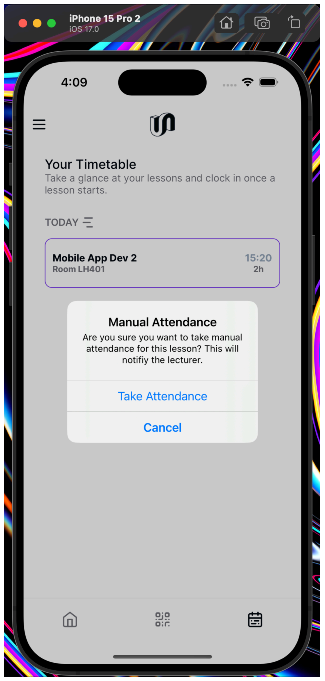
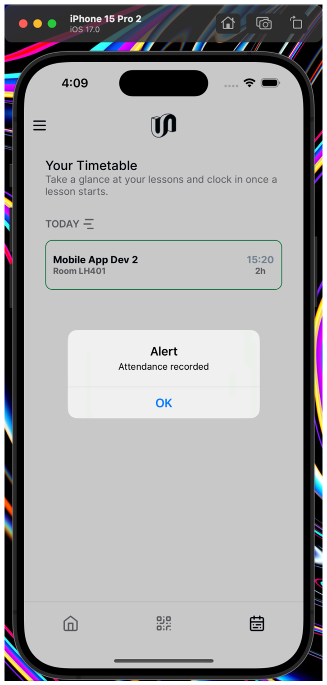
| | | | | | |
|---|---|---|---|---|---|
| | | | | |  Mobile App Dev 2<br>28/01/2024, 15:20 - 17:20<br>All Students<br>Brandon H<br>Computing (Web Technology and Security) ✓ |

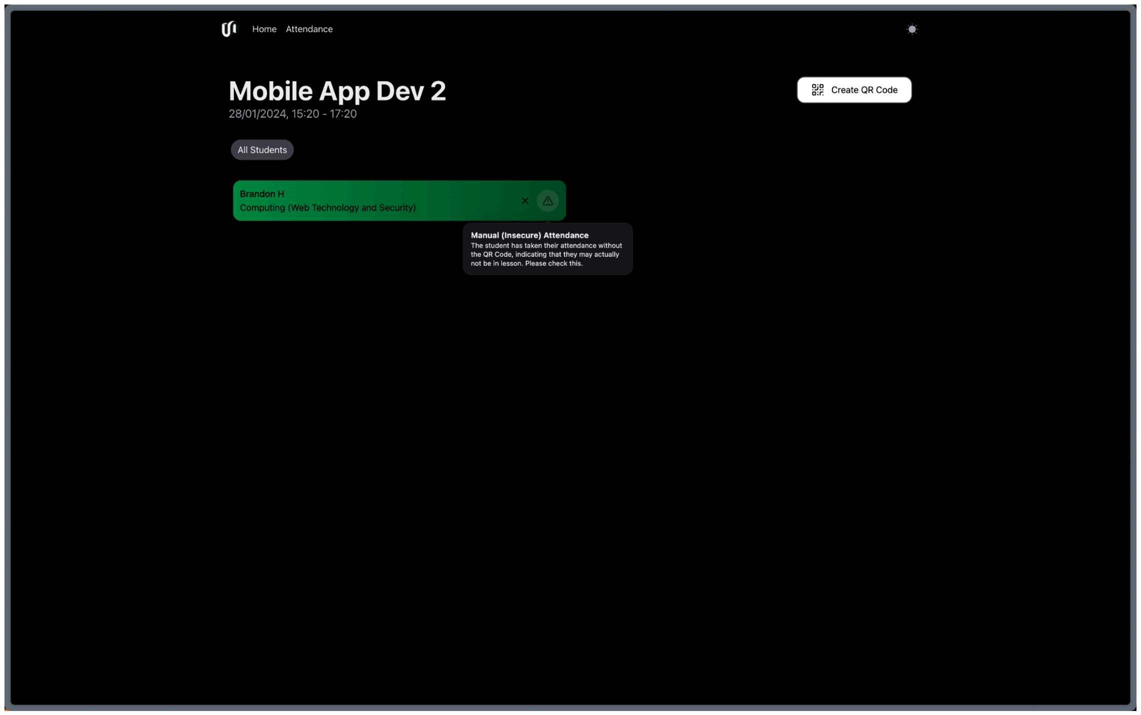
| | | | | | |
|---|---|---|---|---|---|
| Mark in and out students manually on attendance site | Mark in and out Brandon H | Mark attendance for brandon to green when marked in manually,<br><br>Mark it grey when marked out. | As expected | Success |  <br>Mobile App Dev 2<br>28/01/2024, 15:20 - 17:20<br>Brandon H<br>Computing (Web Technology and Security) ✓ |

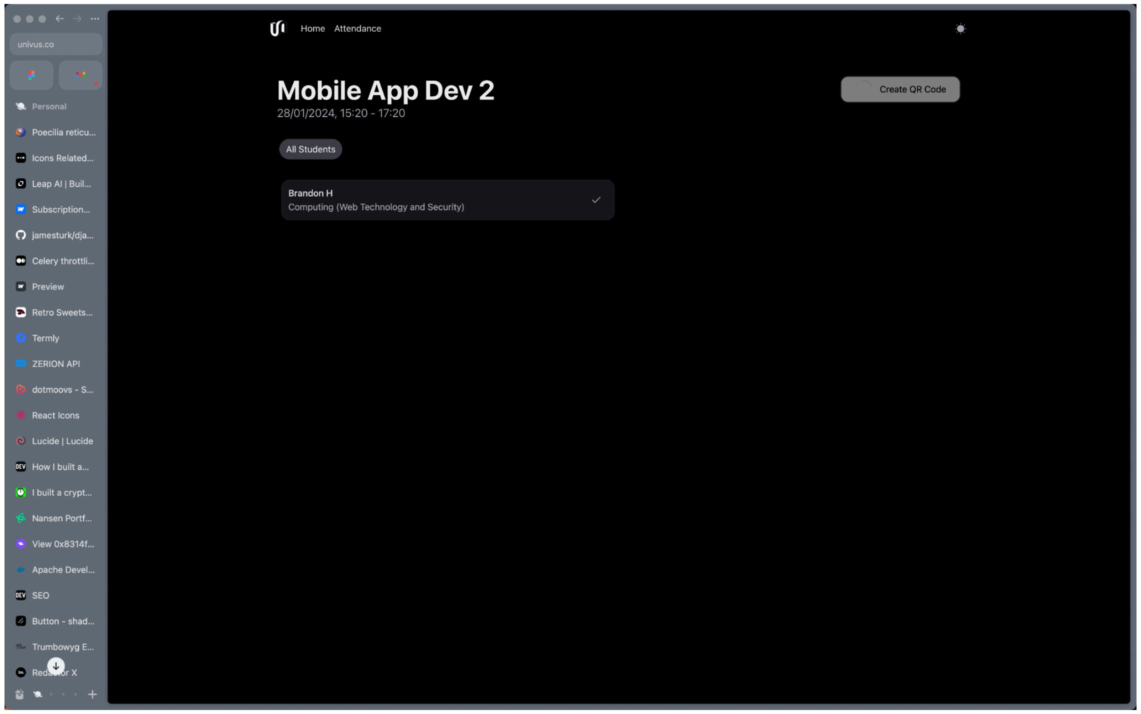
| | | | | | |
|---|---|---|---|---|---|
| | | | | |  Home Attendance<br>Mobile App Dev 2<br>28/01/2024, 15:20 - 17:20<br>Create QR Code<br>All Students<br>Brandon H<br>Computing (Web Technology and Security) |

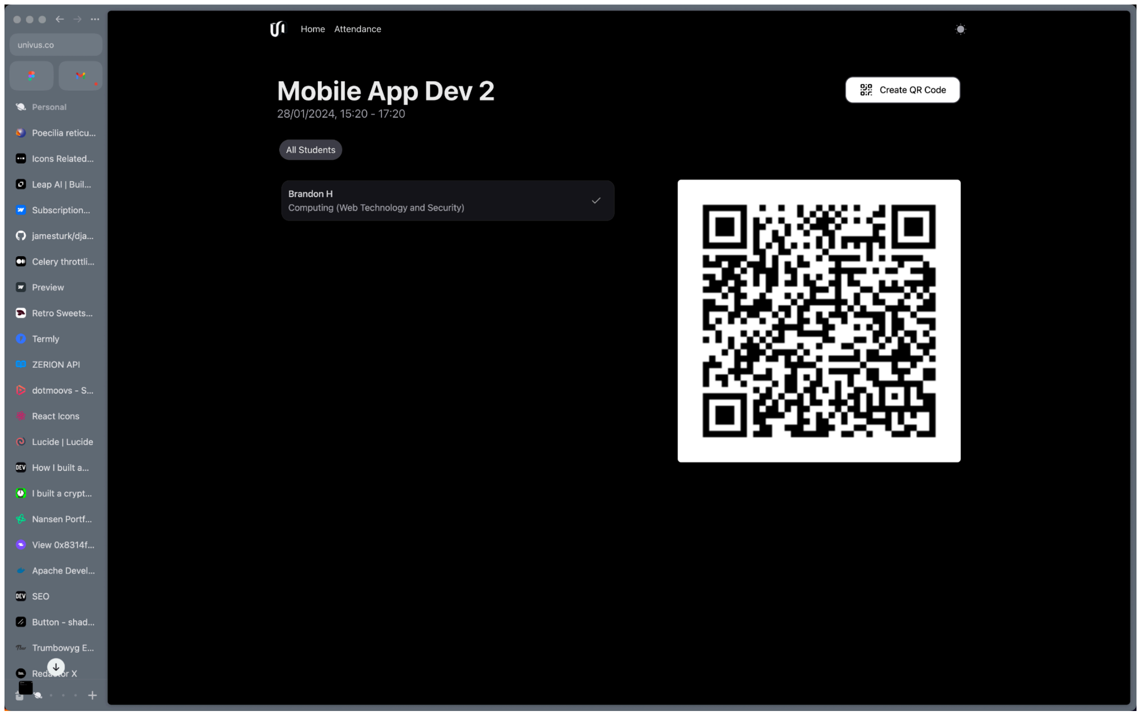
| | | | | | |
|---|---|---|---|---|---|
| | | | | |  Home Attendance Mobile App Dev 2 28/01/2024, 15:20 - 17:20 Create QR Code All Students Brandon H Computing (Web Technology and Security) |

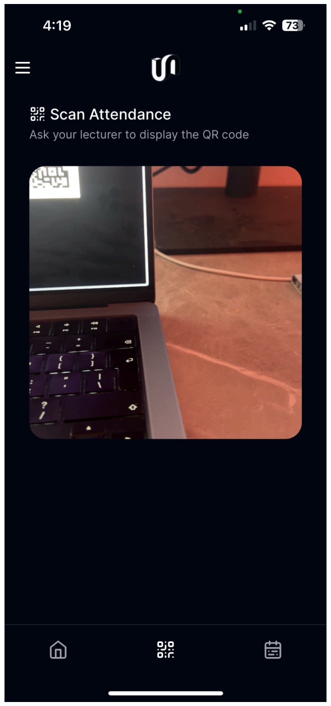
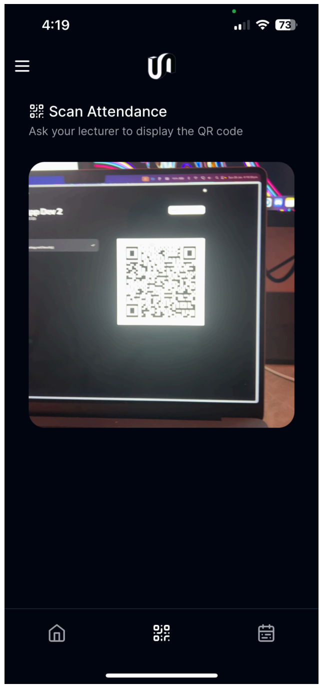
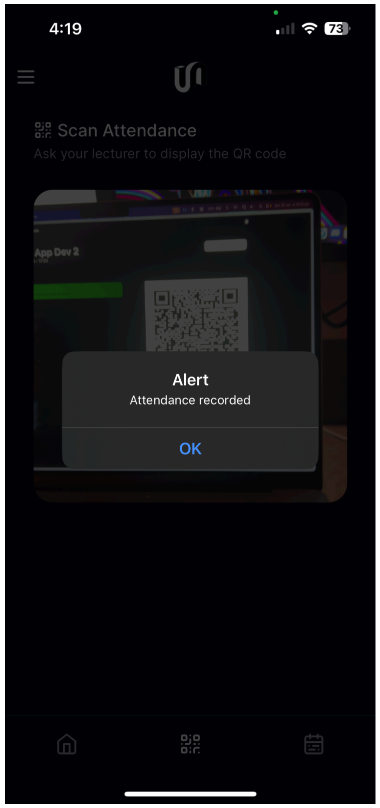
|  |  |  |  |  |  |
|---|---|---|---|---|---|
|  |  |  |  |  |  <br>Mobile App Dev 2<br>28/01/2024, 15:20 - 17:20<br>Brandon H<br>Computing (Web Technology and Security) ✓ |

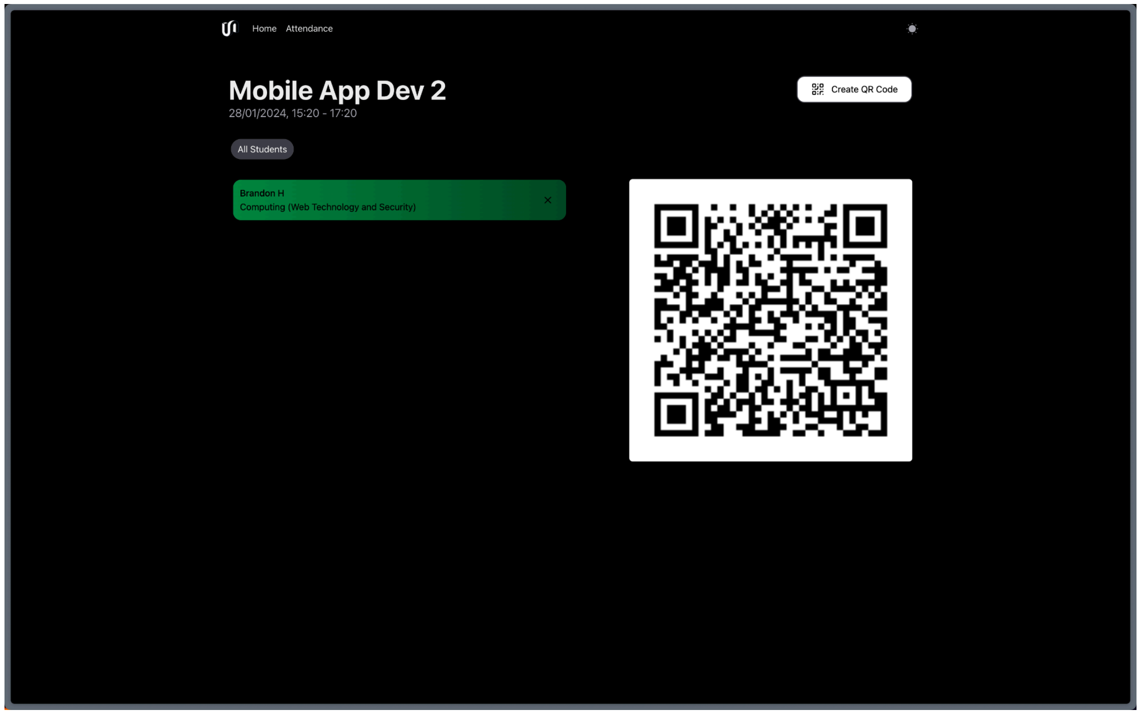
| | | | | | |
|---|---|---|---|---|---|
| Student takes attendance without lecturer (manual) | Brandon H, Mobile app dev 2, Manual | It should show an Alert to say it has been recorded on the student phone, and show it's been recorded on the lecturer website attendance screen with an indication it was insecure because it was manually recorded by the student, instead of QR Code scan. | As expected | Success |  iPhone 15 Pro 2 iOS 17.0 4:09 Your Timetable Take a glance at your lessons and clock in once a lesson starts. TODAY Mobile App Dev 2 Room LH401 15:20 2h **Manual Attendance** Are you sure you want to take manual attendance for this lesson? This will notifiy the lecturer. Take Attendance Cancel  iPhone 15 Pro 2 iOS 17.0 4:09 Your Timetable Take a glance at your lessons and clock in once a lesson starts. TODAY Mobile App Dev 2 Room LH401 15:20 2h **Alert** Attendance recorded OK |

|  |  |  |  |  |  |
|---|---|---|---|---|---|
|  |  |  |  |  |  Home Attendance Mobile App Dev 2 28/01/2024, 15:20 - 17:20 Create QR Code All Students Brandon H Computing (Web Technology and Security) **Manual (Insecure) Attendance** The student has taken their attendance without the QR Code, indicating that they may actually not be in lesson. Please check this. |

| | | | | | |
|---|---|---|---|---|---|
| Lecturer can Create Attendance QR Code that Students can Scan to sign them selves in to lesson. | Lecturer should generate QR Code<br><br>Student Scans it on app. | Should mark student in without lecturer warning indication (all secure) and show alert on student device | As expected | Success | <br>Mobile App Dev 2<br>28/01/2024, 15:20 - 17:20<br>All Students<br>Brandon H ✓<br>Computing (Web Technology and Security) |

| | | | | | |
|---|---|---|---|---|---|
| | | | | |  univus.co<br>Personal<br>Poecilia reticu...<br>Icons Related...<br>Leap AI \| Buil...<br>Subscription...<br>jamesturk/dja...<br>Celery throttli...<br>Preview<br>Retro Sweets...<br>Termly<br>ZERION API<br>dotmoovs - S...<br>React Icons<br>Lucide \| Lucide<br>How I built a...<br>I built a crypt...<br>Nansen Portf...<br>View 0x8314f...<br>Apache Devel...<br>SEO<br>Button - shad...<br>Trumbowyg E...<br>Redactor X<br>Home<br>Attendance<br>Create QR Code<br>All Students<br>Mobile App Dev 2<br>28/01/2024, 15:20 - 17:20<br>Brandon H<br>Computing (Web Technology and Security) ✓ |

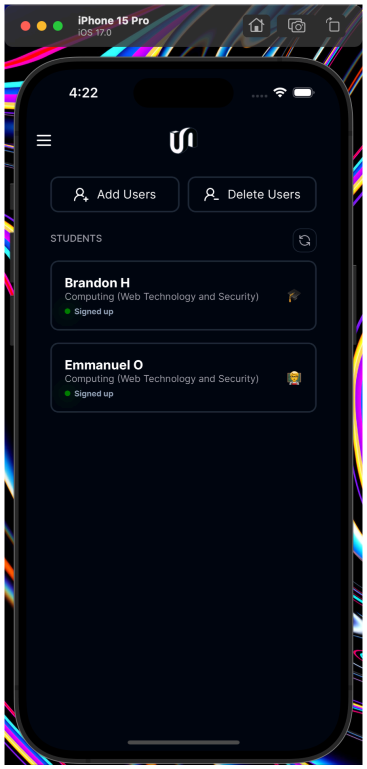
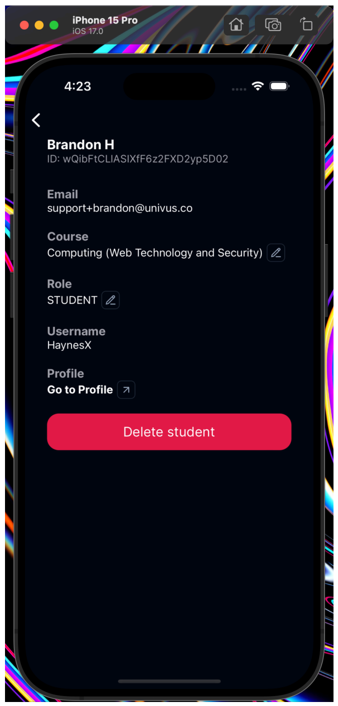
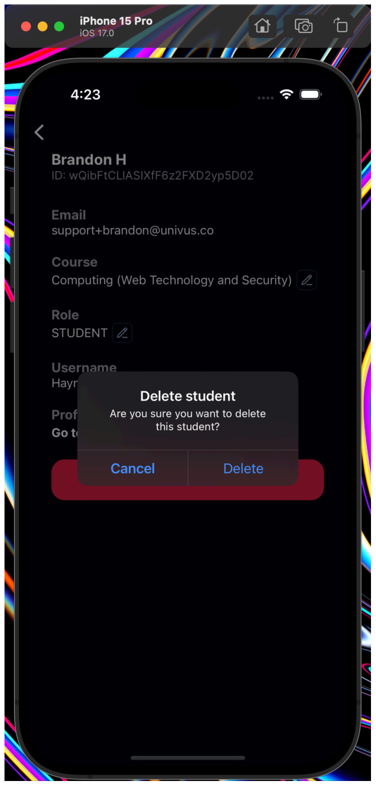
| | | | | | |
|---|---|---|---|---|---|
| | | | | |  4:19<br>Scan Attendance<br>Ask your lecturer to display the QR code  4:19<br>Scan Attendance<br>Ask your lecturer to display the QR code  4:19<br>Scan Attendance<br>Ask your lecturer to display the QR code<br>Alert<br>Attendance recorded<br>OK |

|  |  |  |  |  |  |
|---|---|---|---|---|---|
|  |  |  |  |  |  |

Mobile App Dev 2

28/01/2024, 15:20 - 17:20

Brandon H
Computing (Web Technology and Security)

| | | | | | |
|---|---|---|---|---|---|
| (MANDATORY)<br><br>**DELETE STUDENT** | Delete Brandon H Student | Deletes Student and removes from ManageStudents screen | As expected | Success |    |

iPhone 15 Pro
iOS 17.0

4:22

Add Users
Delete Users

STUDENTS

Brandon H
Computing (Web Technology and Security)
Signed up

Emmanuel O
Computing (Web Technology and Security)
Signed up

iPhone 15 Pro
iOS 17.0

4:23

Brandon H
ID: wQibFtCLIASIXfF6z2FXD2yp5D02

Email
support+brandon@univus.co

Course
Computing (Web Technology and Security)

Role
STUDENT

Username
HaynesX

Profile
Go to Profile

Delete student

iPhone 15 Pro
iOS 17.0

4:23

Brandon H
ID: wQibFtCLIASIXfF6z2FXD2yp5D02

Email
support+brandon@univus.co

Course
Computing (Web Technology and Security)

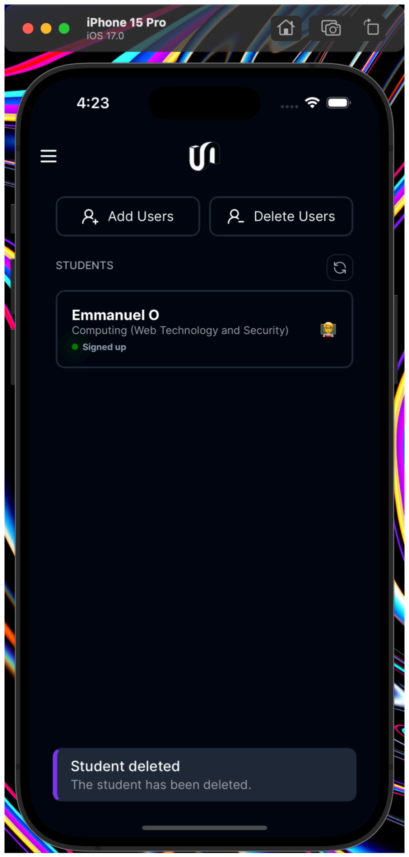
Role
STUDENT

Username
Hayr

Prof
Go t

Delete student
Are you sure you want to delete this student?

Cancel
Delete

| | | | | | |
|---|---|---|---|---|---|
| | | | | |  iPhone 15 Pro iOS 17.0 4:23 Add Users Delete Users STUDENTS **Emmanuel O** Computing (Web Technology and Security) ● Signed up **Student deleted** The student has been deleted. |

### Testing Conclusions

The testing phase for the app using React Native Expo, Firebase Emulators, and real builds was a key part of the project. Testing on both Android and iOS devices took a lot of time. Using Expo and an Apple development subscription helped make test builds for iOS. The main challenge was making sure the app worked well on both Android and iOS without any problems.

It took a lot of effort to figure out how to make everything work right on both types of devices. This process was about solving problems and learning a lot about app development. Testing wasn't just about finding and fixing issues. It was also about understanding how to make apps that work well on different devices.

Even though it was tough, testing the app was also fun and a great chance to learn. It was a big part of getting the app ready for people to use.

## Conclusion

In summary, making Univus for the University of GraceTech was a big project. This mobile app is made with React Native, Expo, and Firebase (and other big tech). It helps students mark attendance in classes and lets them chat and post on social media. A big thing about Univus is that it can work for lots of universities, not just one. This was done using Python for gathering data. But this also made it tricky to handle accounts from different universities.

The design of Univus was inspired by popular social media apps. The goal was to make it easy and intuitive to use. The technical side of the app uses Firebase for different tasks and Redux to keep track of what users are doing in the app. Building the app meant solving many problems, like making sure users could sign in properly and keeping the app running fast and smooth.

Firebase functions in the app are really important. They do things like checking data, managing user info, and other big tasks. These functions run on their own on Firebase, which helps the app respond quickly and work better.

In the end, "Univus" shows how well modern app-making tools work. It combines strong tech with a focus on making the app easy for users. The whole process was about coming up with new ideas, fixing problems, and paying attention to design. The result is an app that's not just right for the University of GraceTech but also flexible for other institutions. Personally (from author perspective) it’s an unfinished masterpiece - because there is so much missed due to time, but, given the time, it would be perfect.

## References Used

https://nextui.org/
https://twitter.com/
https://about.fb.com/news/2023/07/introducing-threads-new-app-text-sharing/
https://nile.northampton.ac.uk/?new_loc=%2Fultra%2Fcourse
https://expo.dev/
https://www.npmjs.com/package/@react-native-community/blur
https://docs.expo.dev/versions/latest/sdk/blur-view/
https://www.npmjs.com/package/@react-native-community/datetimepicker
https://www.npmjs.com/package/react-native-toast-message
https://firebase.google.com/docs/web/setup#available-libraries
https://firebase.google.com/docs/functions/config-env?gen=2nd
https://firebase.google.com/docs/database/admin/retrieve-data#node.js
https://stackoverflow.com/questions/46155/how-can-i-validate-an-email-address-in-javascri
pt
https://ui.shadcn.com/
https://nextjs.org/
https://vercel.com/
https://lucide.dev/icons/
https://firebase.google.com/docs/functions
https://stackoverflow.com/questions/76952406/firebase-blocking-function-beforeusercreate
d-to-save-user-record-into-firestor
https://stackoverflow.com/questions/76037951/google-cloud-platform-before-user-created-f
unction-not-showing-up-in-firebase-au
https://github.com/firebase/firebase-tools/issues/6235