SyncUQ

Rationale

The project I am proposing is an app which will allow UQ students to see when their friends have breaks in between classes.

Students at UQ are part of a diverse and rich community, and many students have close friends from many different fields with very different timetables, and remembering when each friend is free, or messaging many friends hoping to get a response on a break is too troublesome. I am proposing an app which would take a user's uploaded timetable and compare it to his friends' timetables, and notify the user when a friend is out of class at the same time.

The key objectives of this app are to:

- Allow users to upload their timetables
- Allow users to add friends
- Notify users on breaks as to who else is on a break

All of these features are created with the purpose of making socialising and catching up with friends at university easier, with the aim of enriching the university experience and fostering a stronger social culture at UQ.

Theory

It is important to develop websites that comply with web standards in order to create a more stable web. In other words, to create a web that is equally accessible from any device or browser, and still maintain the same functionality. Adhering to standards also makes development for the web easier, as standards are developed in the interest of expanding the web.

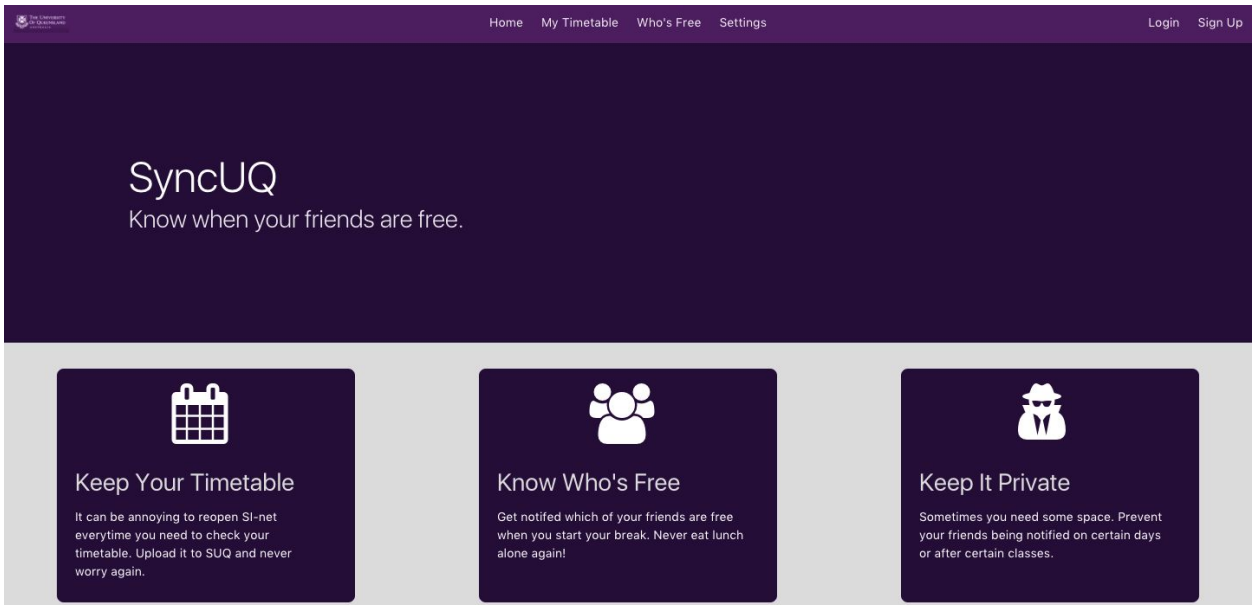
The development language I have chosen is Python, with the Flask framework. This will comprise the back end of the app, and will be used to query and update tables within the database, which we will be using PostgreSQL for. Python is a flexible and powerful language with many libraries available for use, which can make database management a lot simpler.

Design

The design will evoke the look and feel of the UQ design language (The official UQ website as well as UQ Timetable Planner) while also incorporating our own spin on the colour and layout. In order to do this, the main colour scheme will be comprised of different types of purple, with a light purple for the navigation bar and a dark purple for the title banner and other important

elements, while using blues, greens and reds for the timetable display and friends list. This design has been chosen in order to create a professional look using the purple colour scheme while also using other, more vibrant colours to differentiate the app from official UQ websites and create the impression that the app is made for social benefits.

The website will look different for logged in users and logged out users. Logged in users will see their homepage, with various tabs acting as the navigation bar, with only the personalised content appearing. Logged out users will see a sign in page, and a large banner urging visitors to sign up. Below is a basic mockup for the logged out screen.



Feature Coverage

Client Side Communication

Before the client can make any other interaction with the server, they must first create a user account. They will need to be able to fill out a form which creates a new entry in the user database. Clients will then need to be able to log in to the site. This will involve sending user details to the database, verifying that the username and password match, and sending a success or failure message back to the client.

When they are logged in, clients should be able to upload their timetable (in iCal format) to the database, view their timetable, make a friend request by username, accept friend requests, and view which of their friends are “on a break”. This will be achieved by using json files to communicate between the Purescript front end and the Flask back end.

Server side code

Clients will need to be able to create user accounts and add other users to their friend lists. The server side needs to be able to add new users to the database when necessary, as well as update friend lists when a user adds another user to their friends list.

The main unique feature of the project will be the ability to see which of a user's friends are free, or how long it will be until they are free. The back-end of the project will do this by querying the database and comparing the user's timetable with each of their friends' timetables, and determine shared breaks on the fly.

Advanced Javascript

The javascript framework I have chosen is Purescript, using the Pux framework. This will allow easy rendering of HTML, as well as a powerful type system. The two main advanced javascript features will be timetable rendering, and tabbed sections.

Timetable rendering will involve interpreting a json file sent by the server containing the iCal timetable, and rendering HTML elements to the DOM to create a visual representation of the timetable in one hour time periods. This will be generated on the fly when a user queries his or her timetable, as timetables will be stored only in ical format, they will not be saved to a table in an hour-by-hour format.

Tabbed sections will be implemented in order to maintain the same state across multiple "pages" with different functionality. Rather than clicking links to get to different sections of functionality, such as "Timetable View" or "Friends List", users will click tabs to swap between sections. This will involve re-rendering the DOM whenever a user switches tabs. The separate tabs will be:

- Timetable View
- Friends List
- Who's Free
- Settings

This will allow us to easily maintain a consistent state for a user when they are logged in, and also create an aesthetically pleasing user interface.