# Module CO550WBL Web Applications

## Assignment CW1A – Team Project Design

Student names:

* Natalia Fernández Pintos (Student ID: 2191814)
* Wafa Zaidan (Student ID: 22000104)

1. The project idea

This project consists of a web application for the teachers and students of the Digital and Technology Solutions degree. The application provides a user interface in which the users can log in to visualise a calendar that displays a series of events related to the modules of the degree. It also allows users to add, edit and delete events, as well as notify other users of these changes. The application also features a profile page to access and edit user details, as well as an admin page to manage users and access a list of all notifications sent.

The functional requirements for the application were first gathered through brainstorming of ideas and creation of a list of possible functionalities. These then evolved into user stories which define the criteria for each feature of the application:

### User stories for user authentication and management

As a Student/Teacher

I want to be able to Sign Up

So that I can have an account for the Bucks Calendar application

As a Student/Teacher/Admin

I want to be able to Log In with my user details

So that I can get access to the Bucks Calendar application

As an Admin

I want to be able to view a list of existing users

So that I can keep track of who is registered in the application

As an Admin

I want to be able to edit the details of existing users

So that I can update their details if I need to

As an Admin

I want to be able to delete an existing user

So that I can remove their account from the application

### User stories for calendar management

As a Student/Teacher/Admin

I want to be able to create a new event in the calendar

So that I can track university events

As a Student/Teacher/Admin

I want to be able to view university events in the calendar

So that I can be up-to-date with the latest events

As a Student/Teacher

I want to be able to edit my existing university events

So that I can update them when needed

As a Student/Teacher

I want to be able to delete my existing university events

So that I can remove unwanted events

As an Admin

I want to be able to edit or delete events created by any user

So that I can manage the calendar having full access

### User stories for event notifications

As a Student/Teacher/Admin

I want to have the option to notify other users of my event by email or SMS

So that I can inform them of the event

As a Student/Teacher/Admin

I want other users to be subscribed to an event when I choose to notify them

So that I they can receive notifications when an event is created, updated or deleted

As an Admin

I want to be able to see a list of the notification logs

So that I can track sent notifications for events

In order to draft the designs for the application and its different pages, wireframes were created using Adobe XD to generate a prototype of the application. The login and sign up pages (Figures 1 and 2), feature user-friendly forms in a single column. The main page (Figure 3) displays a calendar with the current events and legends for each event type, as well as a navigation bar to access the profile page and sign out. Hovering over an event in the calendar displays a modal (Figure 4) with summarised information about the event. Additionally, there is a plus button which displays a modal that will allow a user to create a new event (Figure 5), as well as the option to click on an existing event to display another modal (Figure 6) which will allow the user to edit the event or delete it. There is also a profile page (Figure 7) with the details of the user, as well as a list of the events that the user has created. Clicking on the edit button in these events will display a modal (Figure 8) which will allow the user to edit or delete the events. Lastly, admin users can access the admin page (Figure 9), which provides a list of all registered users and a list of all notifications sent.

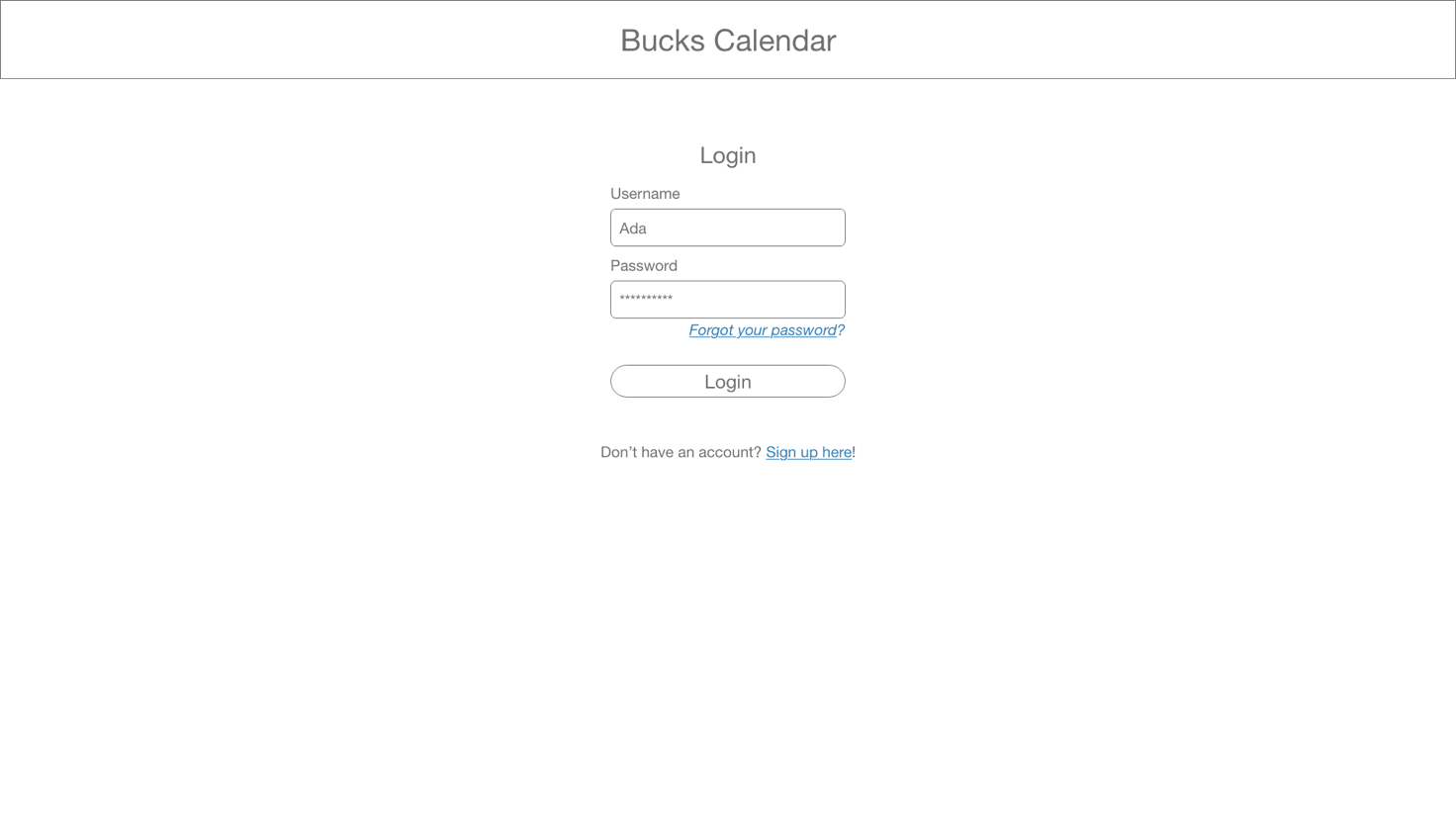
****

Figure 1 - Login page

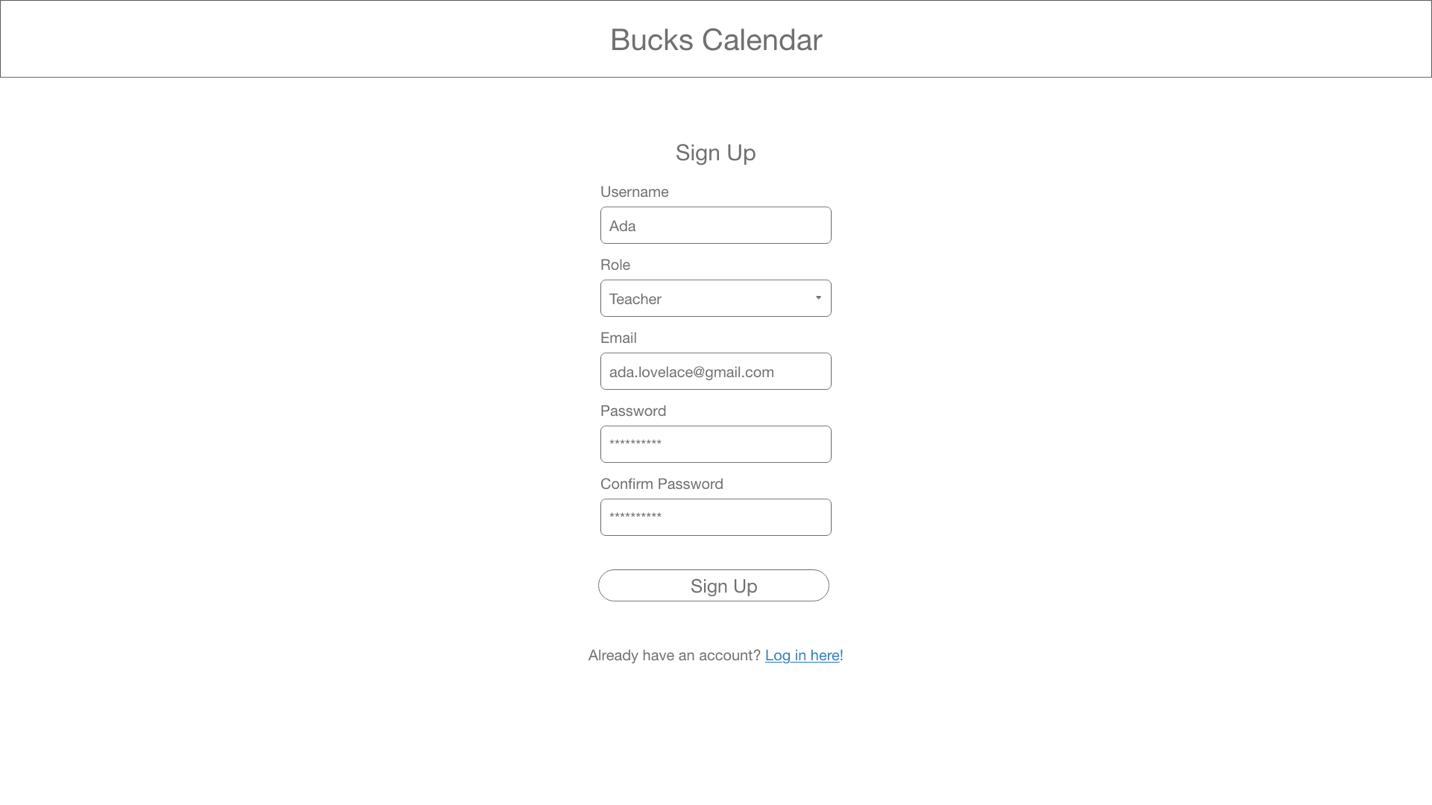


Figure 2 - Sign up page

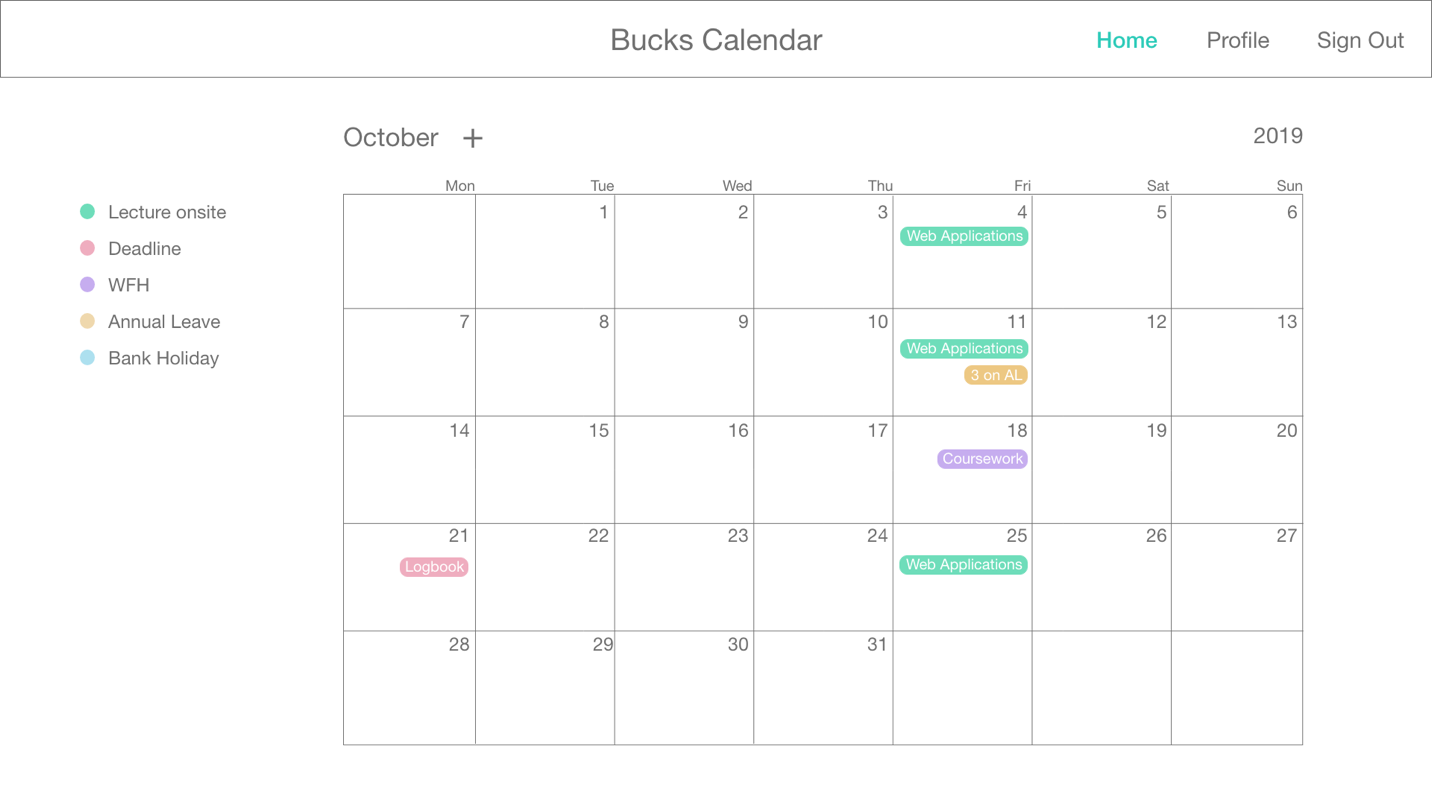


Figure 3 - Main page

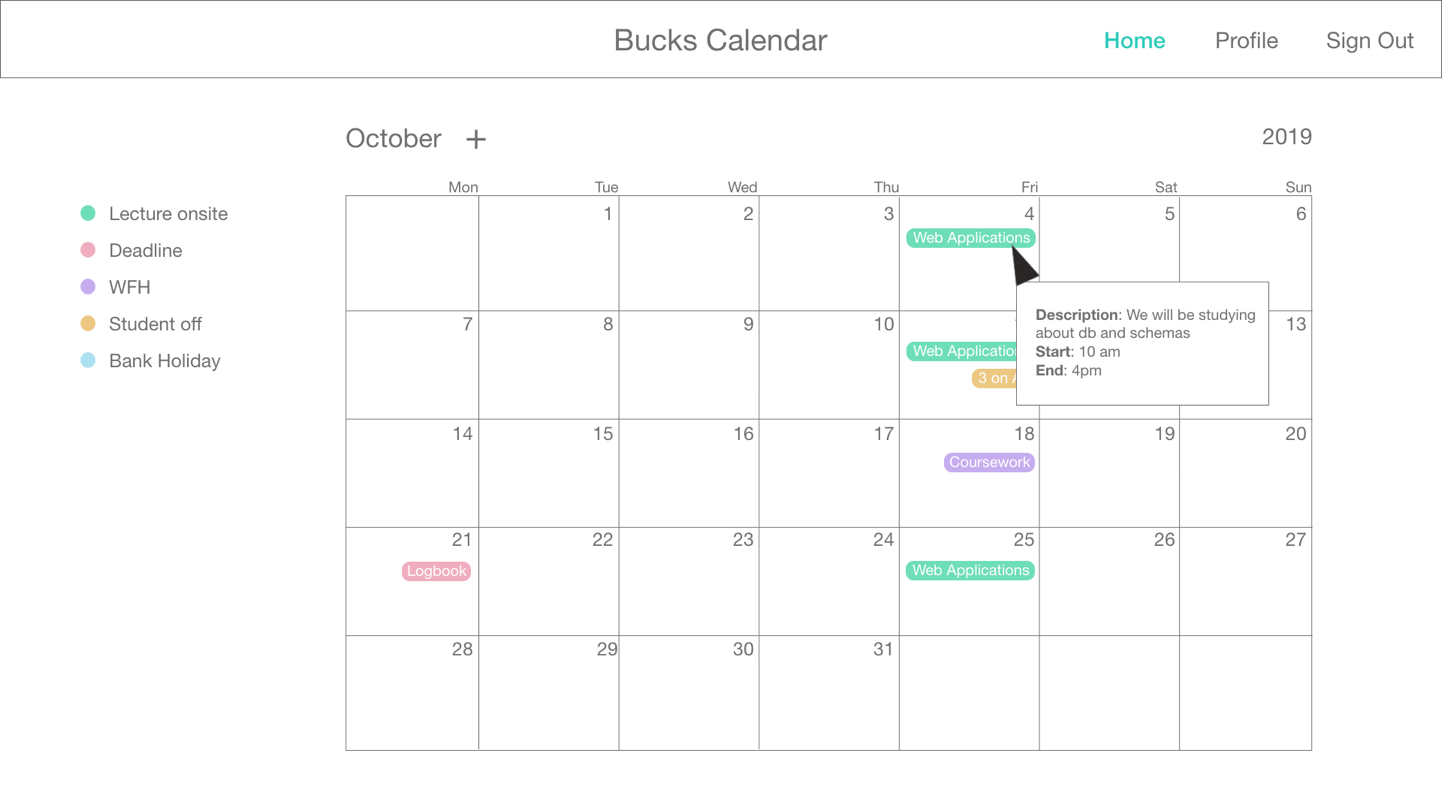
****

Figure 4 - Hover over event

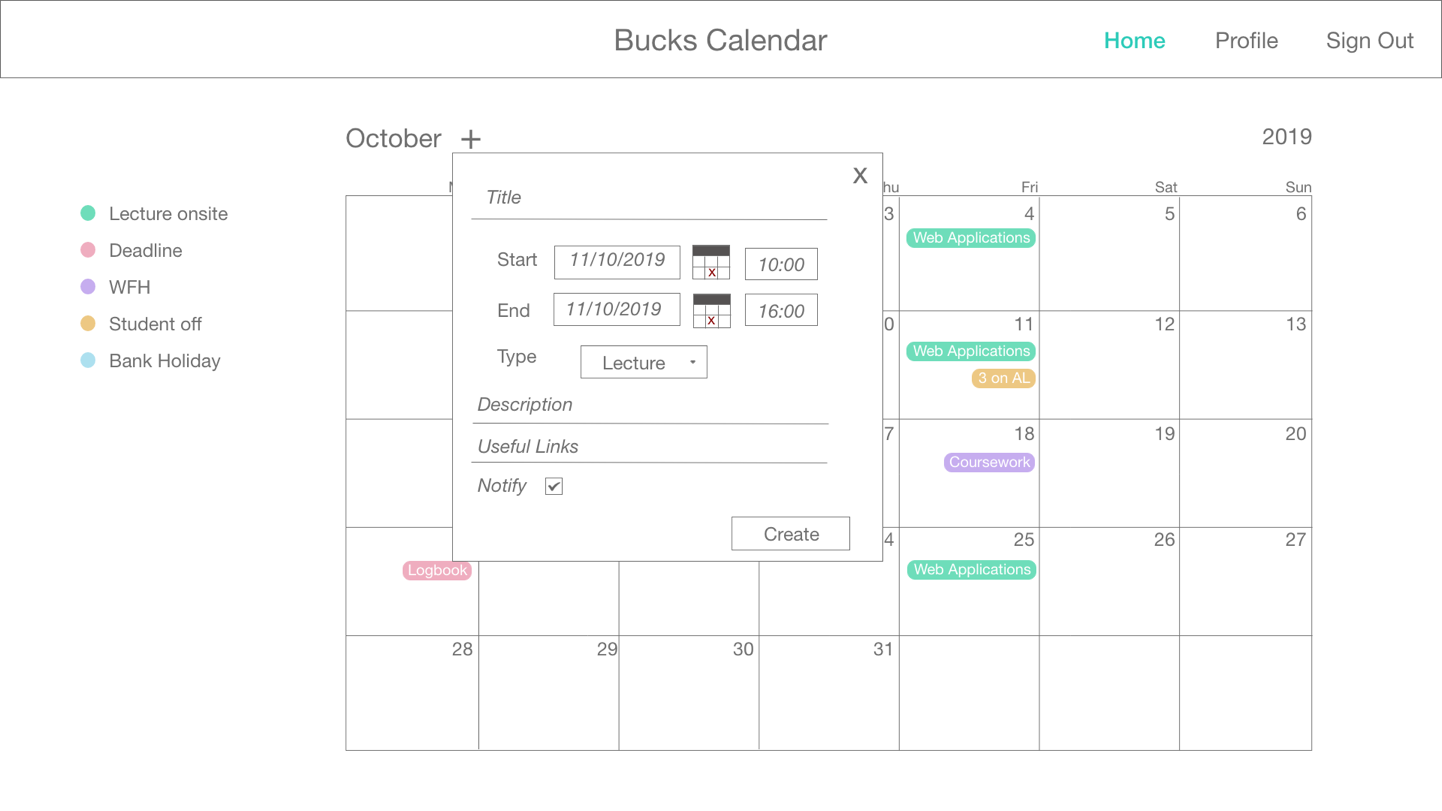
****

Figure 5 - Add event

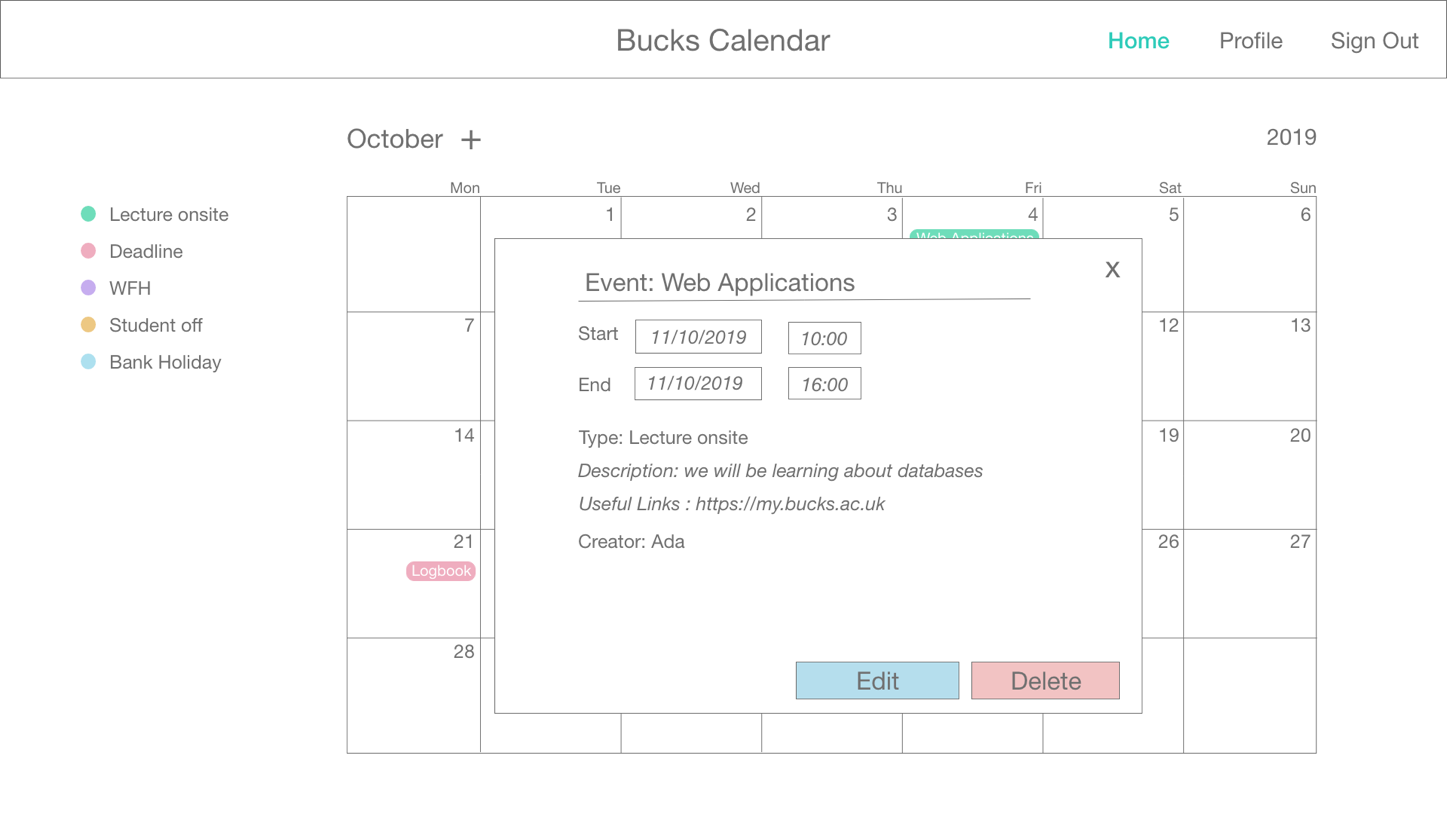
****

Figure 6 - Edit or delete event

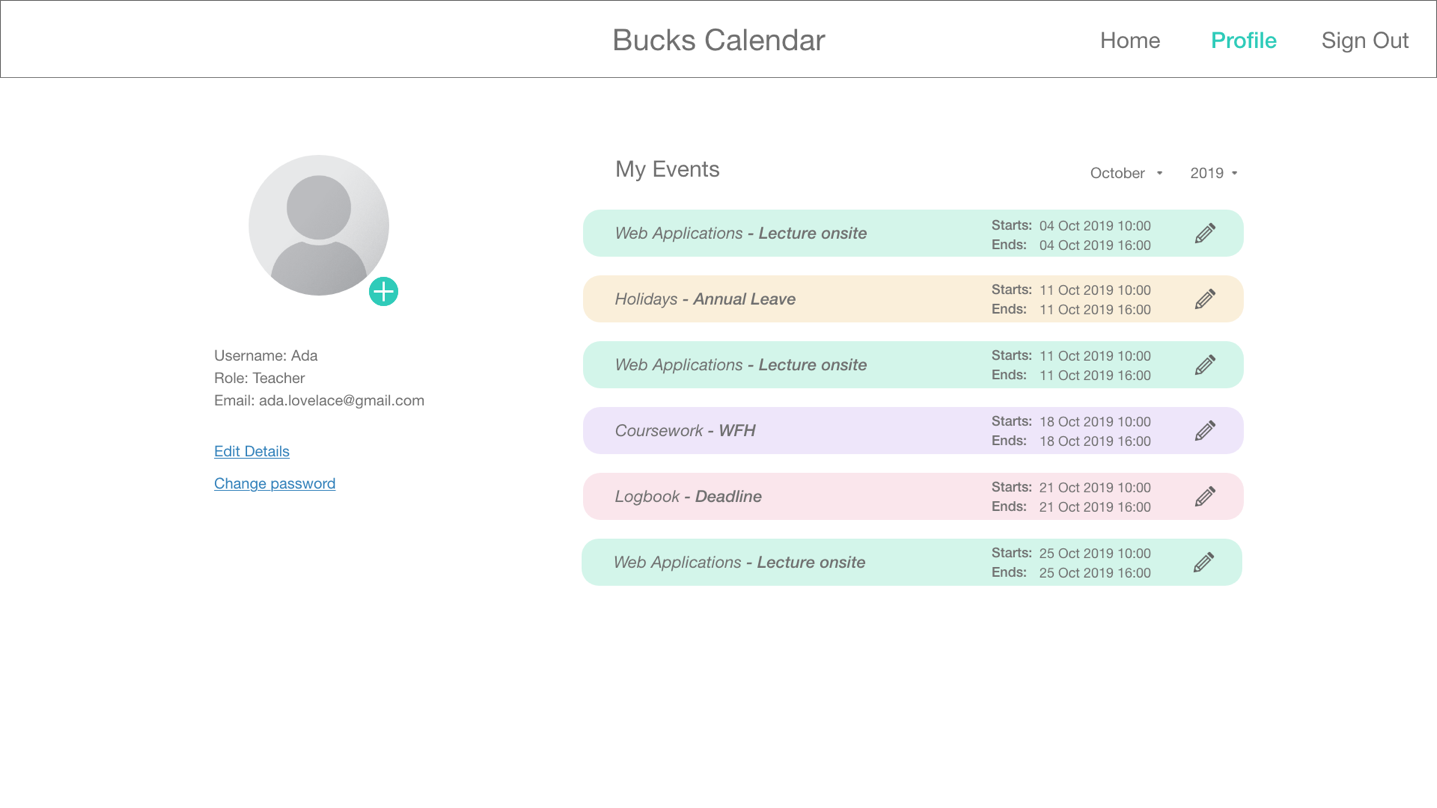
****

Figure 7 - Profile page

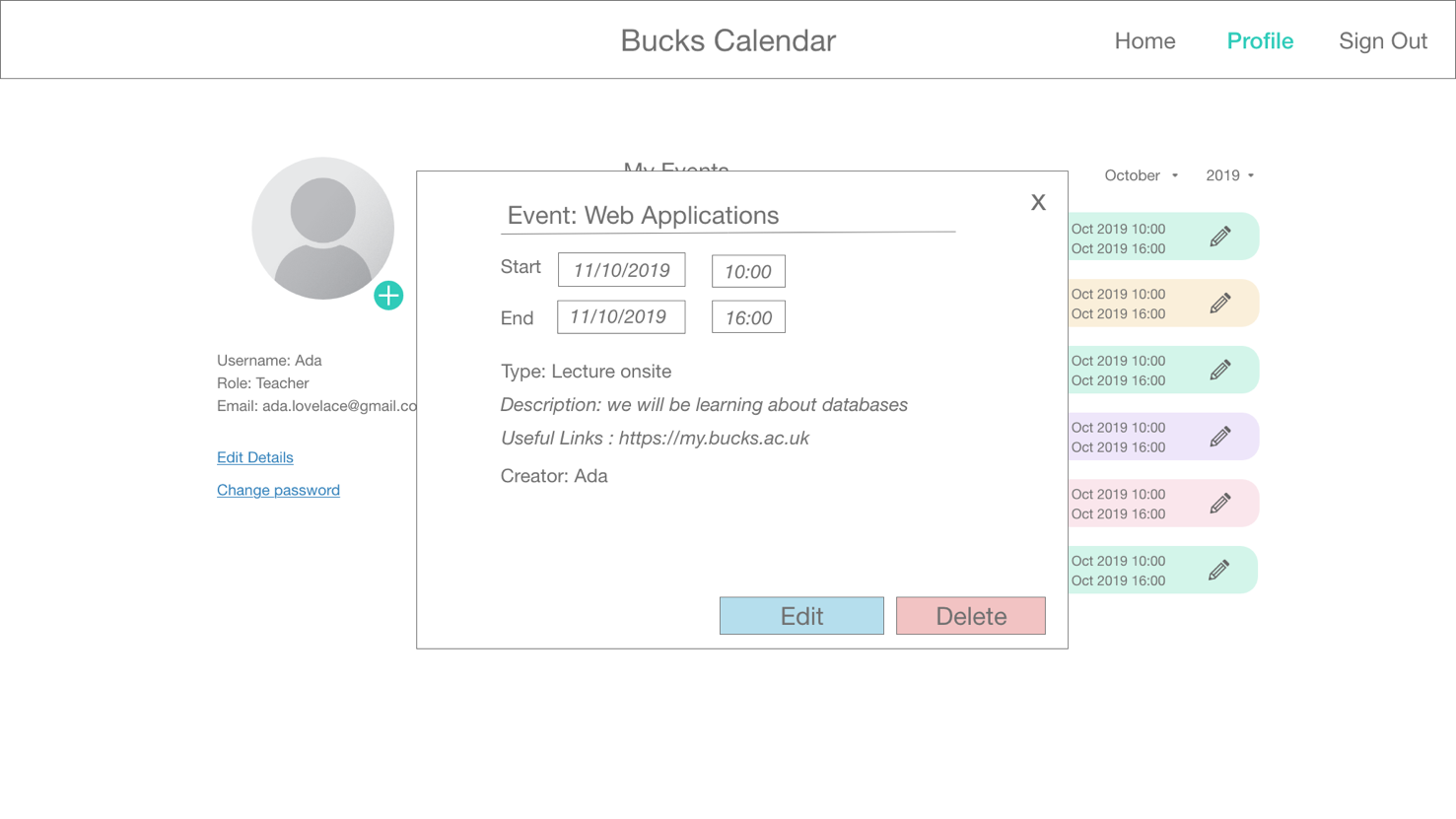
****

Figure 8 - Edit or delete event in profile

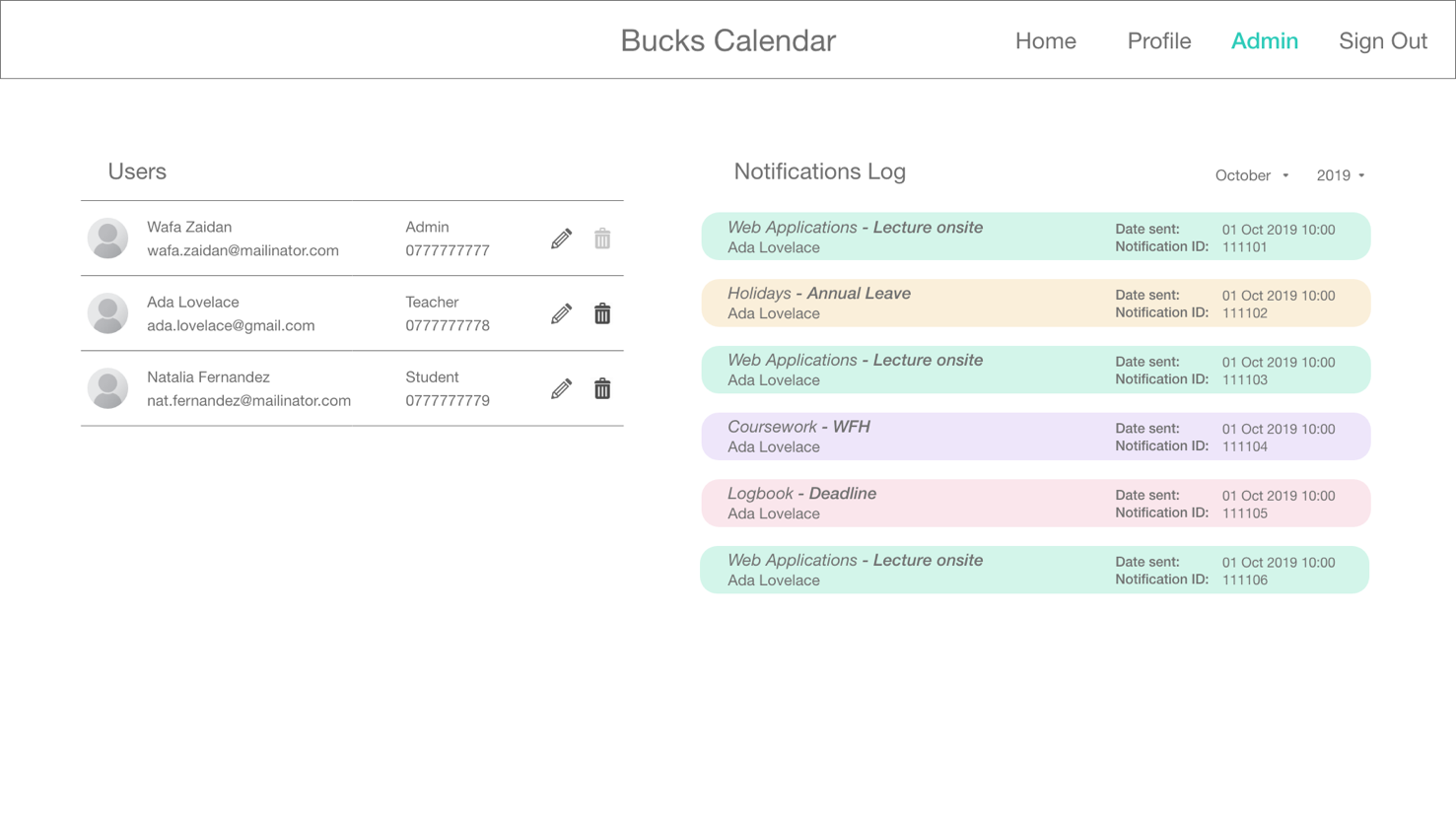
****

Figure 9 - Admin page

1. Background

The idea for this project originates from the need of a unified location in which to easily display information about the timetables and deadlines for the Digital and Technology Solutions degree, as well as details for the attendance to the onsite lectures. Currently, the information about the timetables and deadlines for the modules of the degree is available in a module guide file, and attendance is communicated via different messaging channels (social media or email). The disadvantages of the current methods arise from the difficulty of updating the timetables document, as it is recorded in a static file and would require the teacher to create and upload a new one each time there is the requirement to do any changes to it, and additionally, it would need every student to download a new copy of the document. Furthermore, keeping record of each time a student or teacher is unable to attend a lesson can also be quite inconvenient.

With the web application proposed in this project, the aim is to solve these problems by developing a data-driven solution in which users can keep all this information in a single place. In the first place, the web application would enable its users to keep track of all the information related to the different modules of the degree in a simple way. Secondly, any changes to the existing schedule would be effortless and instantaneous, improving the current need to work with static files. Lastly, the application would enhance the communication between teachers and students, providing the option for users to notify others of any changes in the schedule.

1. Using web applications

The reasons why a web application is suitable for this project are many. In the first place, creating a web application to give access to the schedule information provides availability of this data in a single place with only a couple of clicks, and across a variety of devices. It also has the added benefit of allowing the users to interact with the information in the website in an easy way, which is something that is expected in websites nowadays, as explained by Offutt (2002) we now refer to the visitors of a website as *users*, implying interaction. Moreover, a web application is extensible, and therefore allows the solution to grow and meet new requirements that can arise in the future. Additionally, using a web application that is data-driven has the advantage that we only need to install a web browser to access all the information and services that it provides (Richardson and Ruby, 2007, p. 2), therefore, removing the necessity for each person to install their own calendar applications, add and maintain updated the data across different users manually.

1. Data management:

As the primary objective of our web application is to facilitate timetable management between students and teachers at Bucks university, we recognize that we will be dealing with large amounts of data that need to be well managed and stored, easily accessible and highly maintainable.

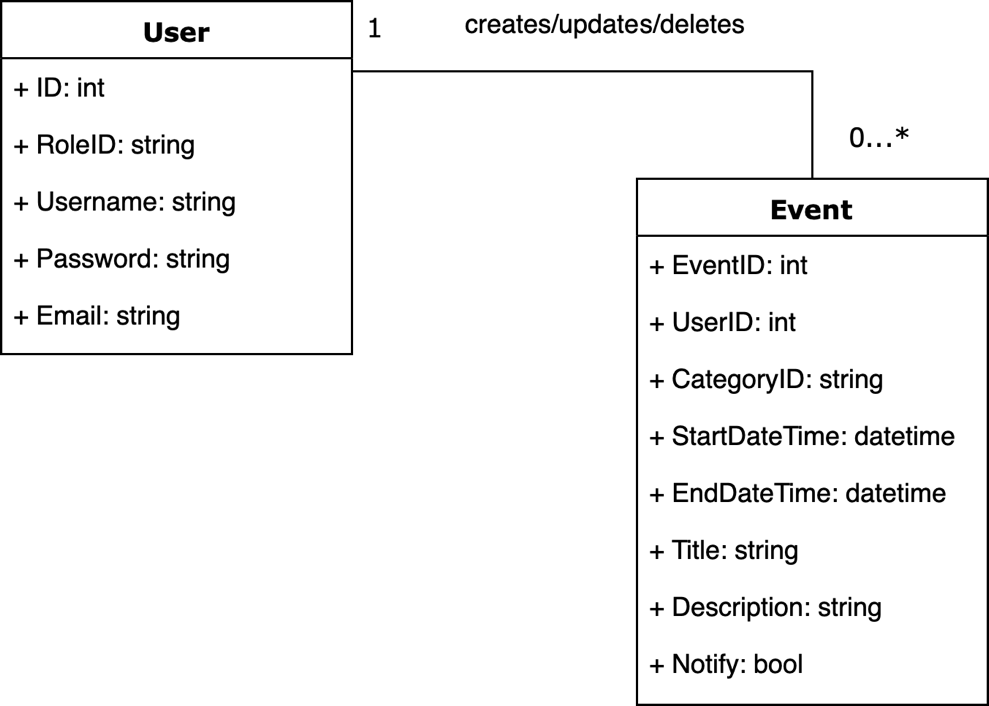
For that reason, we made sure to follow an iterative process of data modelling where we identify and determine real-life information and put them coherently into a well-formed data structure. This process is essential to determine and analyze the kind of information we need in our application which means we can get a clearer understanding of the exact data that we need early-on (Domanski and Irvine, 2000, p. 28).

Initially, we started off by build an Entity Relationship Diagram which is an abstract representation of the data that will be used in our program. The ERD format has three main elements; entities, relationships and attributes (Liu, 2011, p. 276).

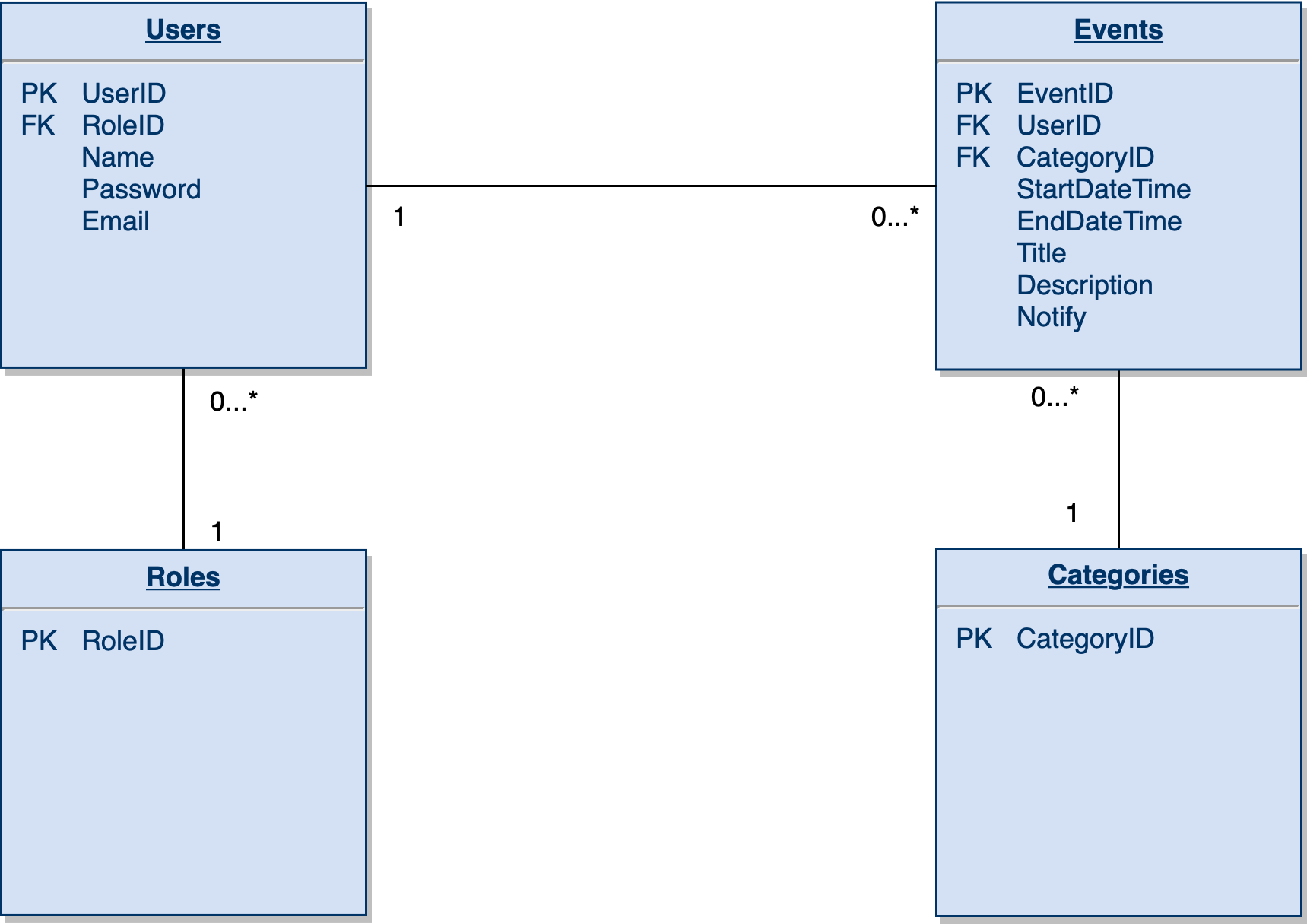
In our program, we defined two main entities which are the ‘User’ and the ‘Event’, as the application mainly revolves around users creating events in our calendar, and to share and notify other users with key events.

Then we identified the attributes that each entity and their data type. For example, for the User we included an Id and a role Id of type integer, and a username, password and email of type string. Whereas for the Event we identified an event Id, user Id, category Id of type integer, start-date and time and end-date and time of type datetime, a title, description of type string and finally a notify field of type Boolean. We also defined the entities’ relationship with each other and their multiplicities. For example, a user can create, update or delete zero to many numbers of events and an event can be managed by only one user.

After creating the ERD, we moved on to building our relational-database schema where we specify the description of the structure and behavior of our database. (Balsters, 2000, p. 1). This is where we visualize the entities that will exist in the relational-database with their tables, fields and kid of relationship whether they have a One-To-One, One-To-Many or Many-To-Many relationship (Gupta, 2007, p.15).

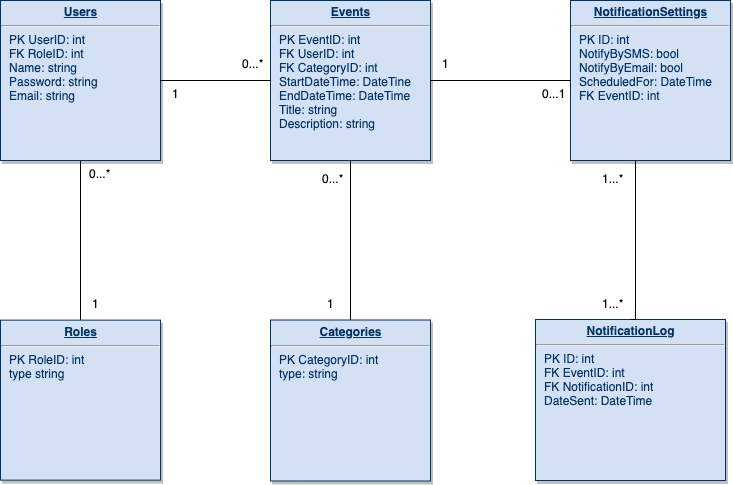


Building upon the ERD, we identified four tables which are users, events, categories and roles and depicted the relationship that they will have for each other. We also listed the fields that each table will have and marked those that will constitute the primary or the foreign key.



However, after going through our diagrams we spotted some major similarities between the ER diagram and our database schema so we decided to merge the two into one ER diagram to reduce the amount of duplicate data and to give a better view of the outcome of our data modelling process.

In the process of refactoring the diagrams, we came up with more interesting features that we could add to our programme to make it more efficient and useful, like the mechanism of notifying the users of events and the means by which they could do so. Based on that, we introduced two new tables, the first is to record data about notification settings that the owner of the event specified and the second to keep logs of notifications that have been sent.



This process of iterative thinking and constant refactoring to manage the data that will be used in our application has helped us understand what data we need to build the programme, and how it will be structured and stored.

Finally, we believe that this iterative process has provided us with solid foundation to build upon in the next stage where we start coding to build the Bucks calendar application.

1. References

Balsters, H., Brock, B., and Conrad, S. (2000) *Database Schema Evolution and Meta-Modeling*. Dagstuhl Castle: Springer.

Domanski, P. and Irvine, P. (2000) *A Practical Guide to Relational Database Design*. Herefordshire: Diaxon Ltd.

Gupta, K. (2007) *Taxonomy of Database Management System*. New Delhi: Firewall Media.

Liu, H. (2011) *Oracle Database Performance and Scalability: A Quantitative Approach*. Hoboken: Wiley.

Offutt, J. (2002) ‘Quality attributes of Web software applications’, *IEEE Software*, 19(2), pp. 25-32. doi: 10.1109/52.991329.

Richardson, L. and Ruby, S. (2007) *RESTful Web Services.* Sebastopol: O’Reilly.