****

**Design Documentation**

**Software Design and development project**

**Unit Coordinator:** Jamie Shield

**Tutor:** Dr.Zakiullah Khan

**Group Team member:**

Adarsha Mani Lamichhane, 12140071

Bandana Kapali, 12127043

Javagal Srinath Narayan, 12131231

Maruf Siddique, 12132167

Contents

[1. Project Brief 3](#_Toc79339578)

[2. Software Architecture 3](#_Toc79339579)

[3. Layer Modelling 4](#_Toc79339580)

[4. Database Design 6](#_Toc79339581)

[5. Process flow diagram (Lab Members End) 7](#_Toc79339582)

[6. Process flow at admin’s end 8](#_Toc79339583)

[7. UML Use case diagram 9](#_Toc79339584)

[8. Class Diagram of Lab equipment sourcing system. 10](#_Toc79339585)

[9. Sequential diagram 11](#_Toc79339586)

[10. User Interface Prototype 13](#_Toc79339587)

[Figure 1 : N -tier Architecture 4](#_Toc79339615)

[Figure 2 MVP pattern 5](#_Toc79339616)

[Figure 3 : Database Design ERD 6](file:///C:\Users\Bandana\Desktop\software%20engineer\design.docx#_Toc79339617)

[Figure 4 : Process flow (Member’s End) 7](#_Toc79339618)

[Figure 5 : Process flow (Admin's End) 8](#_Toc79339619)

[Figure 6 : Use Case Diagram 9](#_Toc79339620)

[Figure 7: Class Diagram 10](#_Toc79339621)

[Figure 8: Sequence diagram (admin) 11](#_Toc79339622)

[Figure 9: Sequence diagram (Lab members) 12](#_Toc79339623)

[Figure 10 Login screen 13](#_Toc79339624)

[Figure 11 : Student Dashboard 14](#_Toc79339625)

[Figure 12 : Admin Dashboard 15](#_Toc79339626)

[Figure 13 : Search window 16](#_Toc79339627)

[Figure 14: History Page 16](#_Toc79339628)

[Figure 15 : My settings 17](#_Toc79339629)

[Figure 16 Student dashboard 18](#_Toc79339630)

[Figure 17 : Student Dashboard 19](#_Toc79339631)

[Figure 18 : Admin Dashboard, Student Search 19](#_Toc79339632)

[Figure 19 Add or delete student page 20](#_Toc79339633)

[Figure 20 : Equipment details 20](#_Toc79339634)

[Figure 21 : Campus Details 21](#_Toc79339635)

# Project Brief

CQU lab management system is an application build for the staffs and student to quickly find lab equipment available in all the campuses. Lab management system offers a set of key features that will support a modern laboratory operation. The main aim of our application will be to eliminate the spreadsheet and paperwork and to organize and record all the lab equipment. CQU lab management system will set the key features that support a modern lab operation like data tracking support, flexible architecture and smart data exchange interfaces that will support its use in everyday regulated environment.

# Software Architecture

The CQU lab management system will be developed in the single tier architecture that means that all the 3 layers that is presentation, logical and data layers will be tightly connected, and all code will keep on single machine. This will also enable high security.

To understand in a very simple ways, single tier architecture has components as presentation logic, business logic and data access logic. Presentation logic where the tasks are translated to something that user can understand and logic tier coordinated the application, process commands and makes logical decision and performs calculation. Logic tier also moves and process the data between the surrounding layers. However, data access tier stores the data and information and retrieved its from the database and passed back to the logic tier so that logic tier can then process it.

N-tier is very easy to use because the application is divided into independent tiers, one can easily reuse each tier for other software projects (Alvatar,2017). N-tier is very simple and friendly for the development so that different team members can work in different tiers.

Diagram

Description automatically generated

Figure 1 : N -tier Architecture

# Layer Modelling

CQU lab Management system will follow Model View presenter pattern (MVP). The MVC pattern divides an application into three parts and they are:

* Model

The model is a set of classes that illustrates business logic and data information.

* View

View is the component which directly interact with user like Activity, Fragments.

* Presenter

Presenter receives the input form users and process the user’s data with the help of model and passes the data results back to the view. The presenter manipulates the model and also updates the view (Sinhal,A. 2017).

Diagram

Description automatically generated

Figure 2 MVP pattern

# Database Design

Database design starts with a conceptual data model and produces specification of a logical schema (Watt, 2014). The main objective of database design is to produce logical and physical design models of the database system. The database design will then decide how and what data must be stored in the system. Here is the design documentation of the CQU lab Equipment Sourcing system:

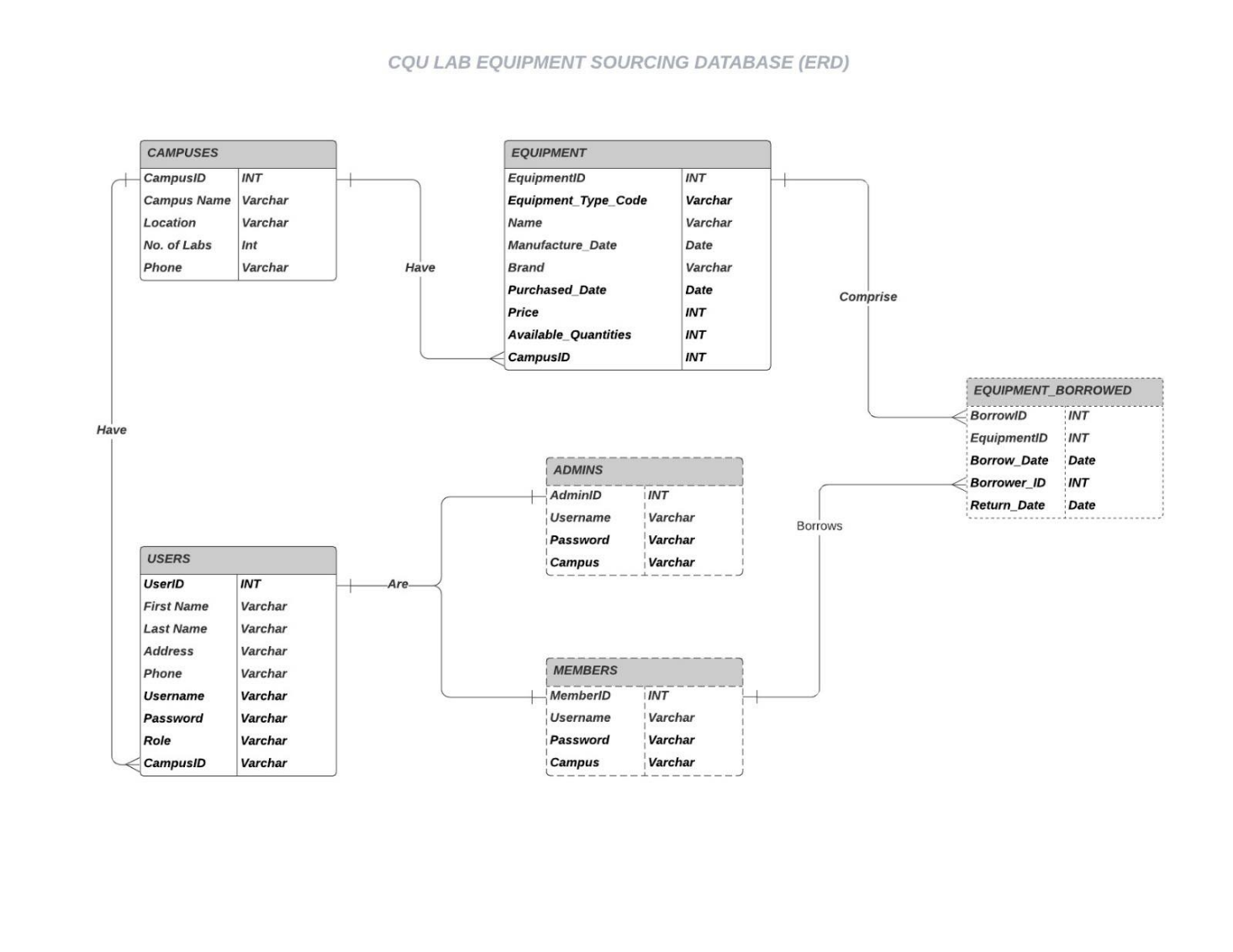


Figure 3 : Database Design ERD

# Process flow diagram (Lab Members End)

Project flow diagram is a flowchart used to illustrate high level processes. Below you can see the flow chart diagram of Lab Equipment Sourcing software through Members End:

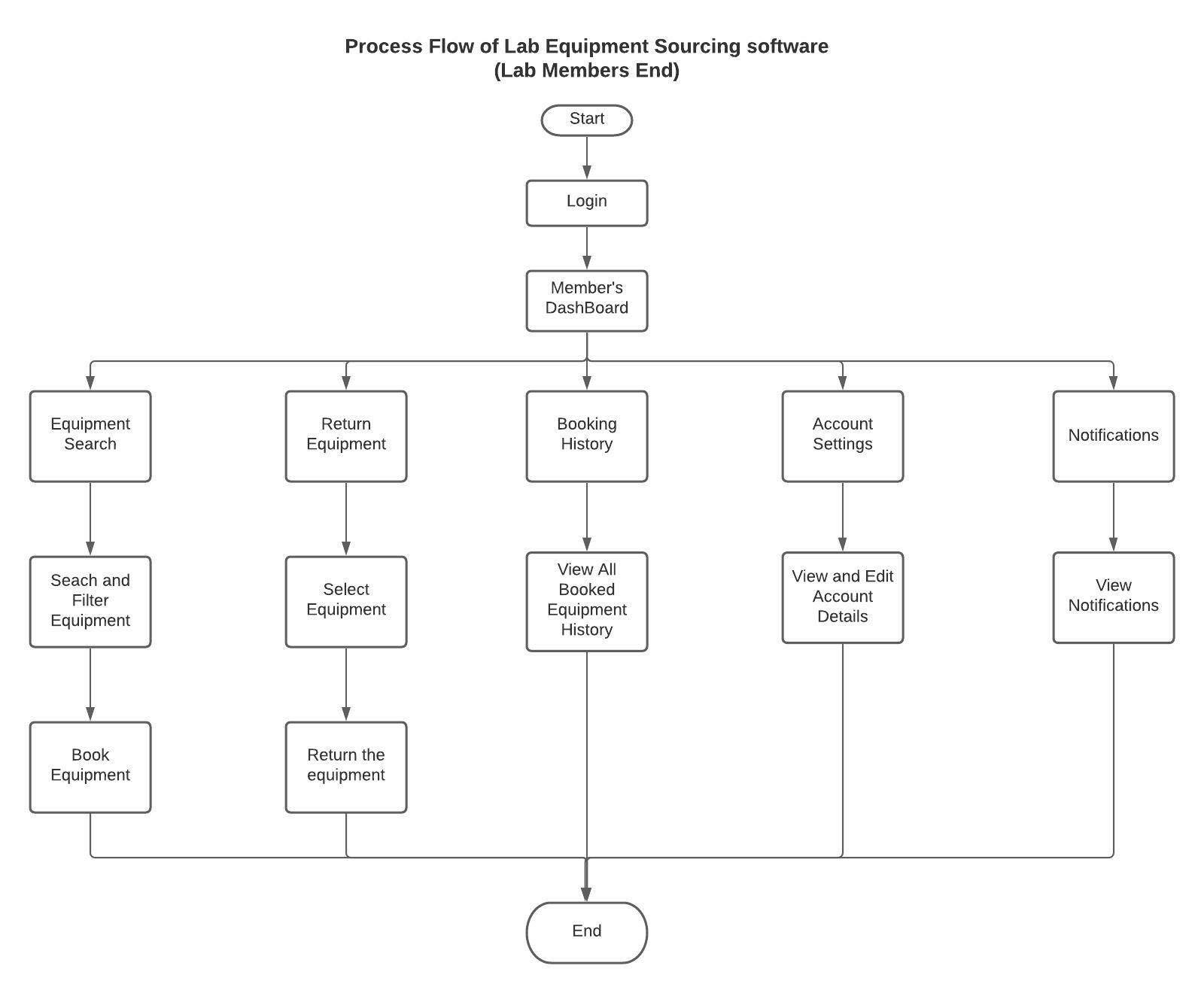


Figure 4 : Process flow (Member’s End)

# Process flow at admin’s end

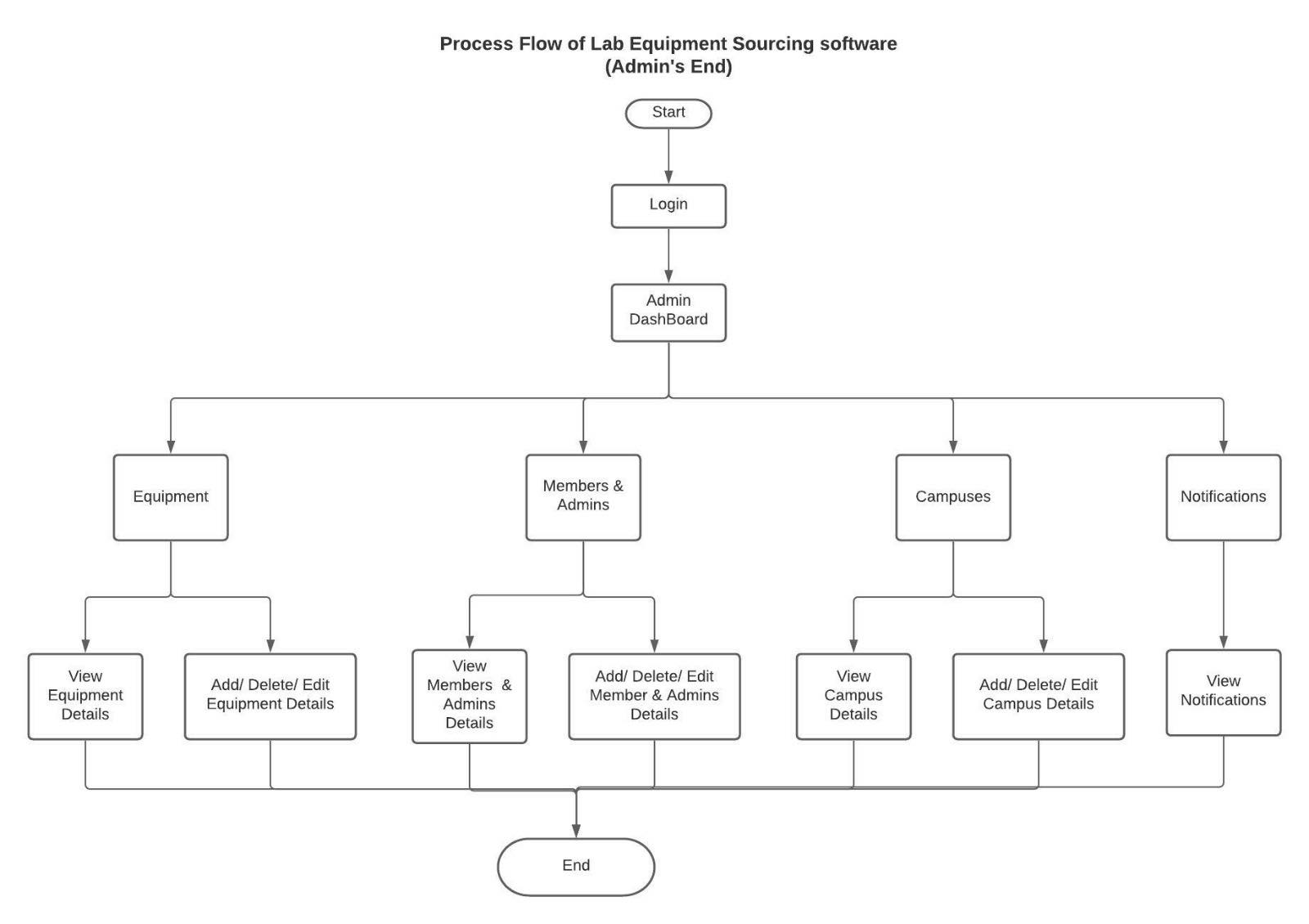


Figure 5 : Process flow (Admin's End)

# UML Use case diagram

Use case diagram describes the high-level functions and the scope system. Use case diagram model describes the behaviour of the system and helps to capture all the requirements of the application. The figure below shows the Use case diagram of the Lab Equipment System:

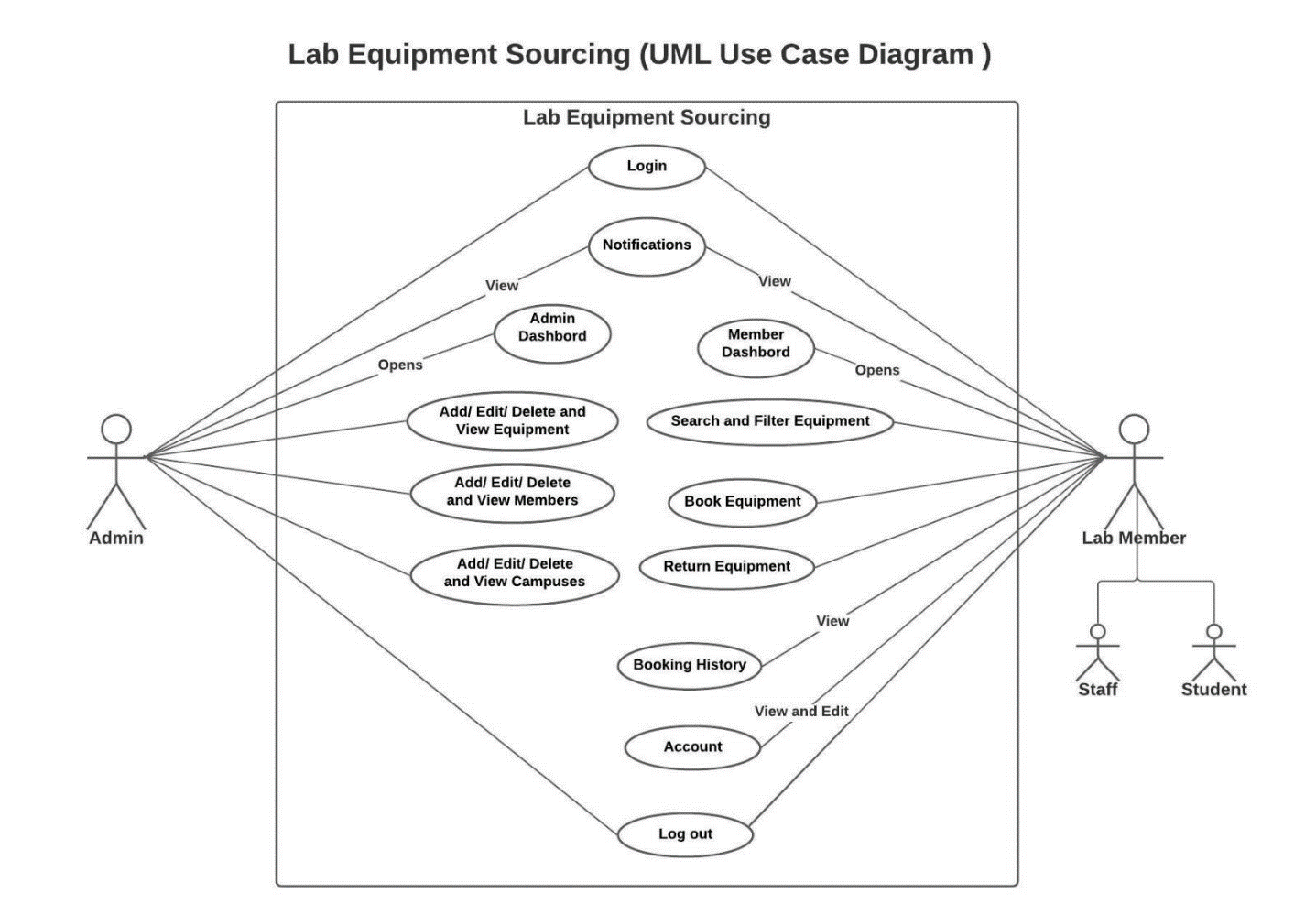


Figure 6 : Use Case Diagram

# Class Diagram of Lab equipment sourcing system.

Class diagrams are the building blocks in the object-oriented language. Class diagrams are used to show the different entities and attributes of the system.

Classes in the class diagram are represented in the box, the top part contains the name of the class whereas the middle part contains attributes, and the last bottom part represents the operations which are associated with the class.

The following figures is a class diagram of lab equipment sourcing system:

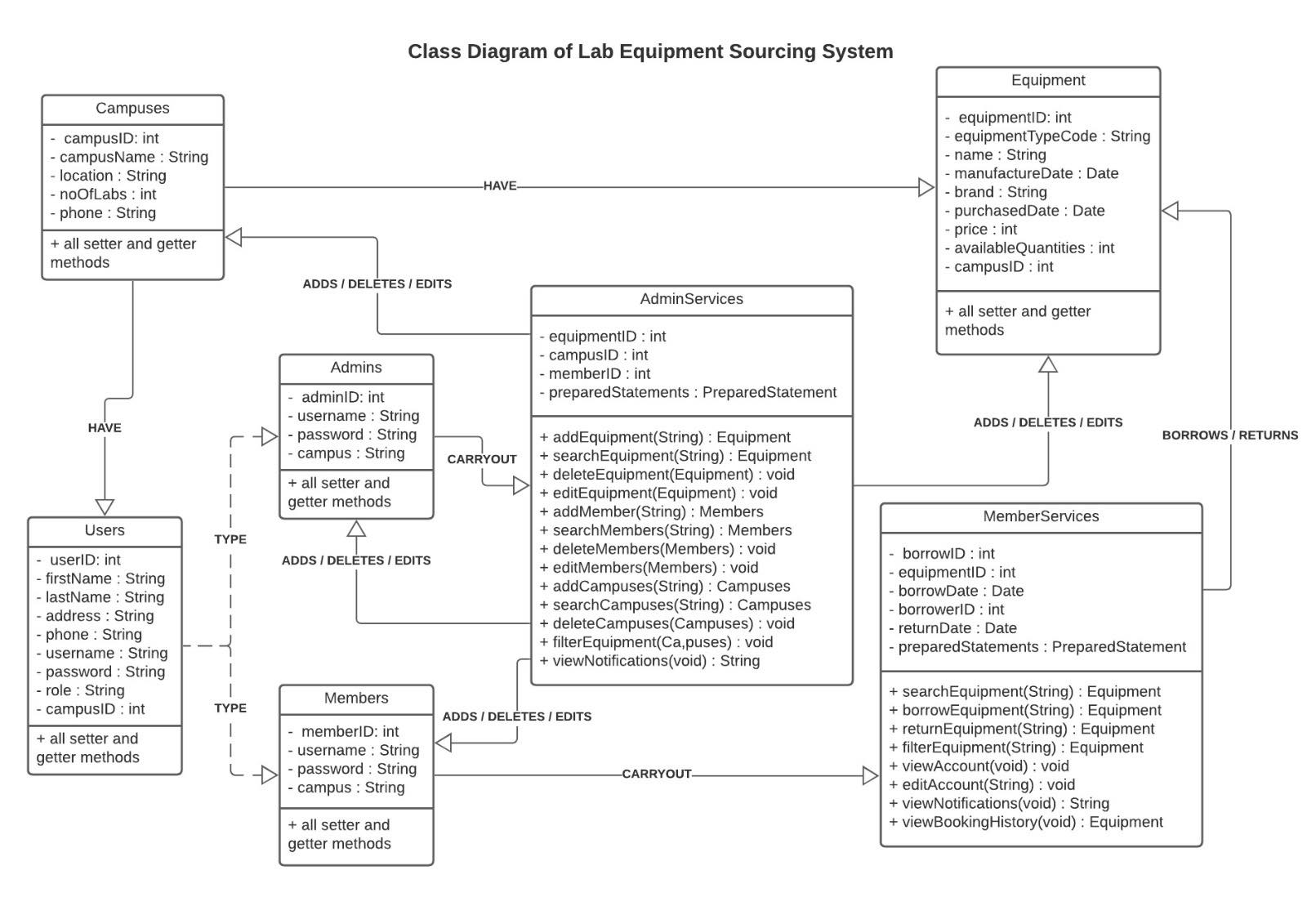


Figure : Class Diagram

# Sequential diagram

Sequence diagram is defined by the UML manual as “a diagram that shows object interactions arranged in time sequence” (Baqais and Alshayeb, 2018). Sequence diagram is a UML diagram which illustrates the sequence of the message between the objects. Sequence diagram helps to understand the relation between the objects. Figure below shows the interaction of admin(actor) and the layer entities:

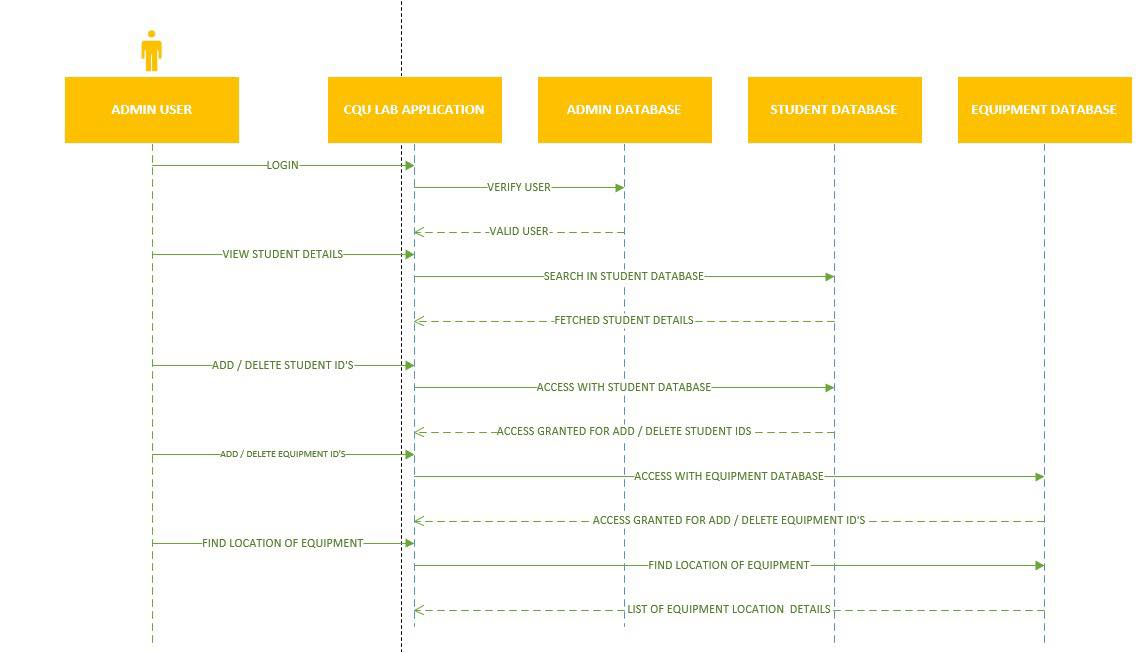


Figure : Sequence diagram (admin)

Above diagram shows the interaction between admin and the layer entities. The admin logins to the CQU application. In the homepage we can see admin logins with their credentials authentication and only admin can view staff and student details and can add, delete staffs and students.

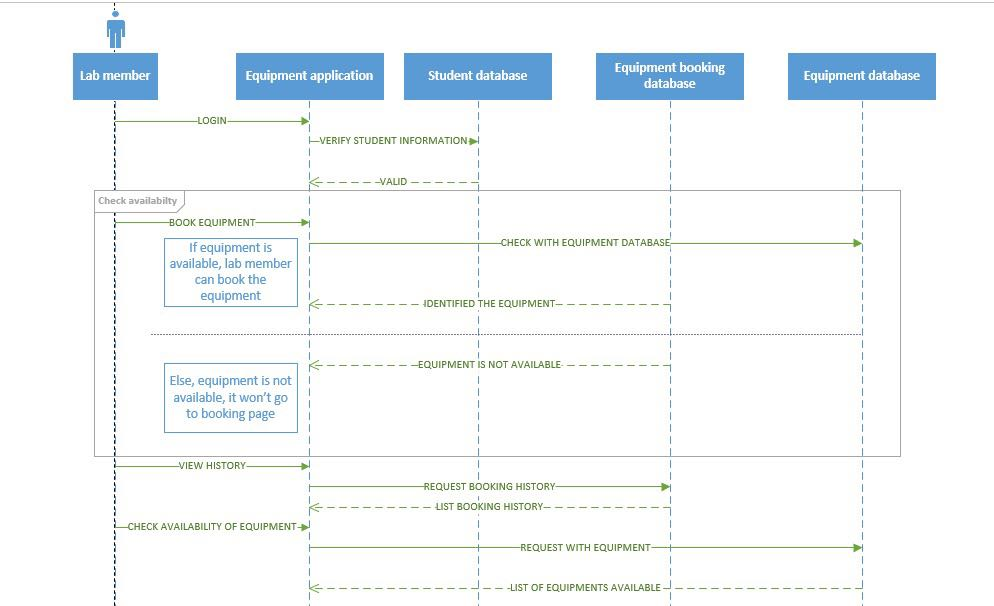


Figure : Sequence diagram (Lab members)

Here in the above diagram, we can see the interaction between the lab members and the layers. In the homepage we can see that the lab members logins with there credentials and if that is valid then the lab members can use rest of the function like booking equipment and checking the availability of the equipment.

# 10. User Interface Prototype

**1. LOGIN SCREEN**



Figure 10 Login screen

**2. STUDENT DASHBOARD**

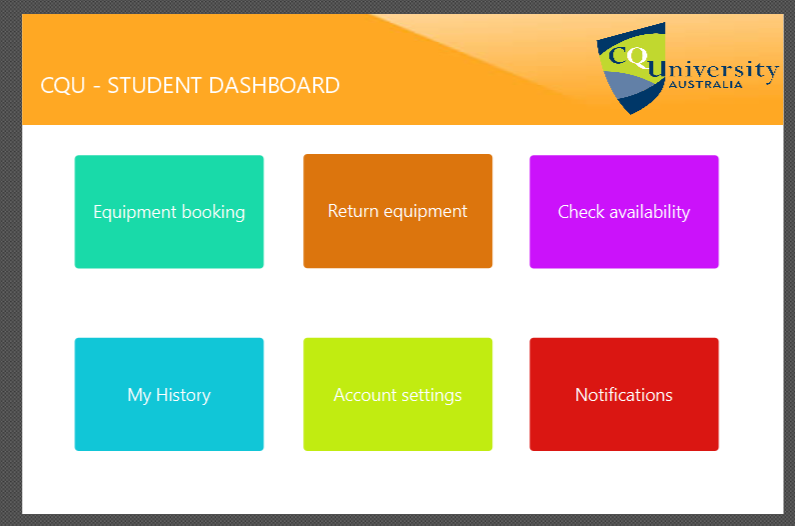


Figure 11 : Student Dashboard

**3. ADMIN DASHBOARD**



Figure 12 : Admin Dashboard

**4. EQUIPMENT SEARCH WINDOW – PROTOTYPE DESIGN**

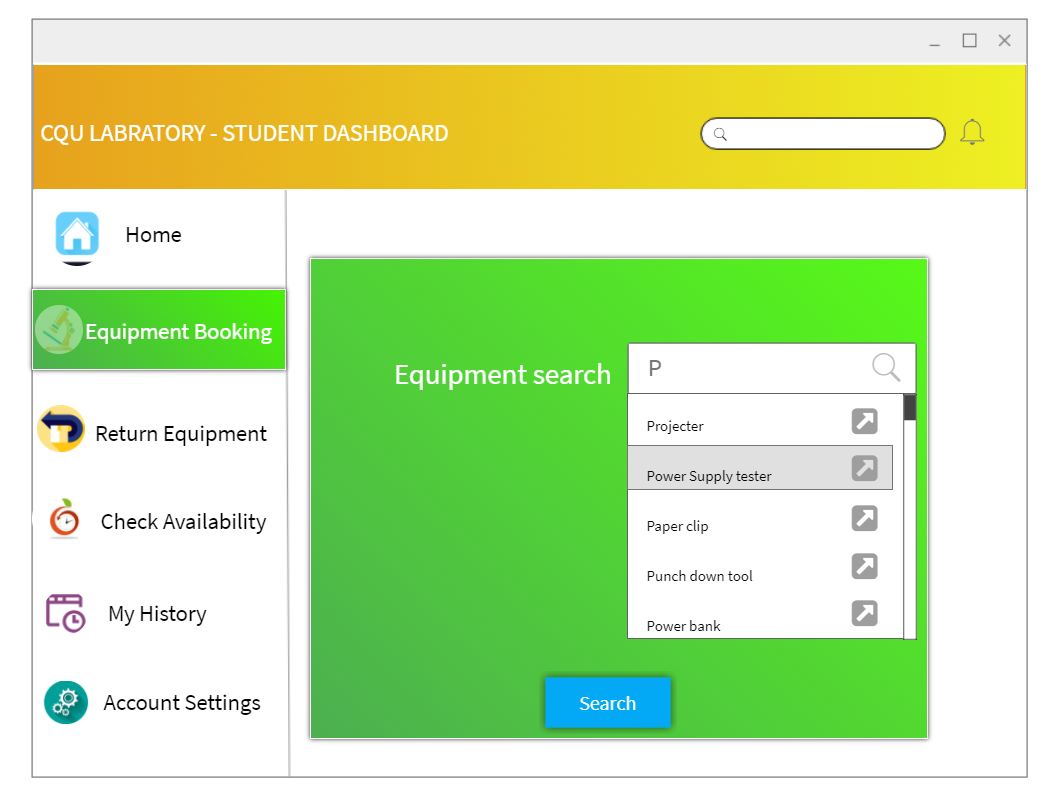


Figure 13 : Search window

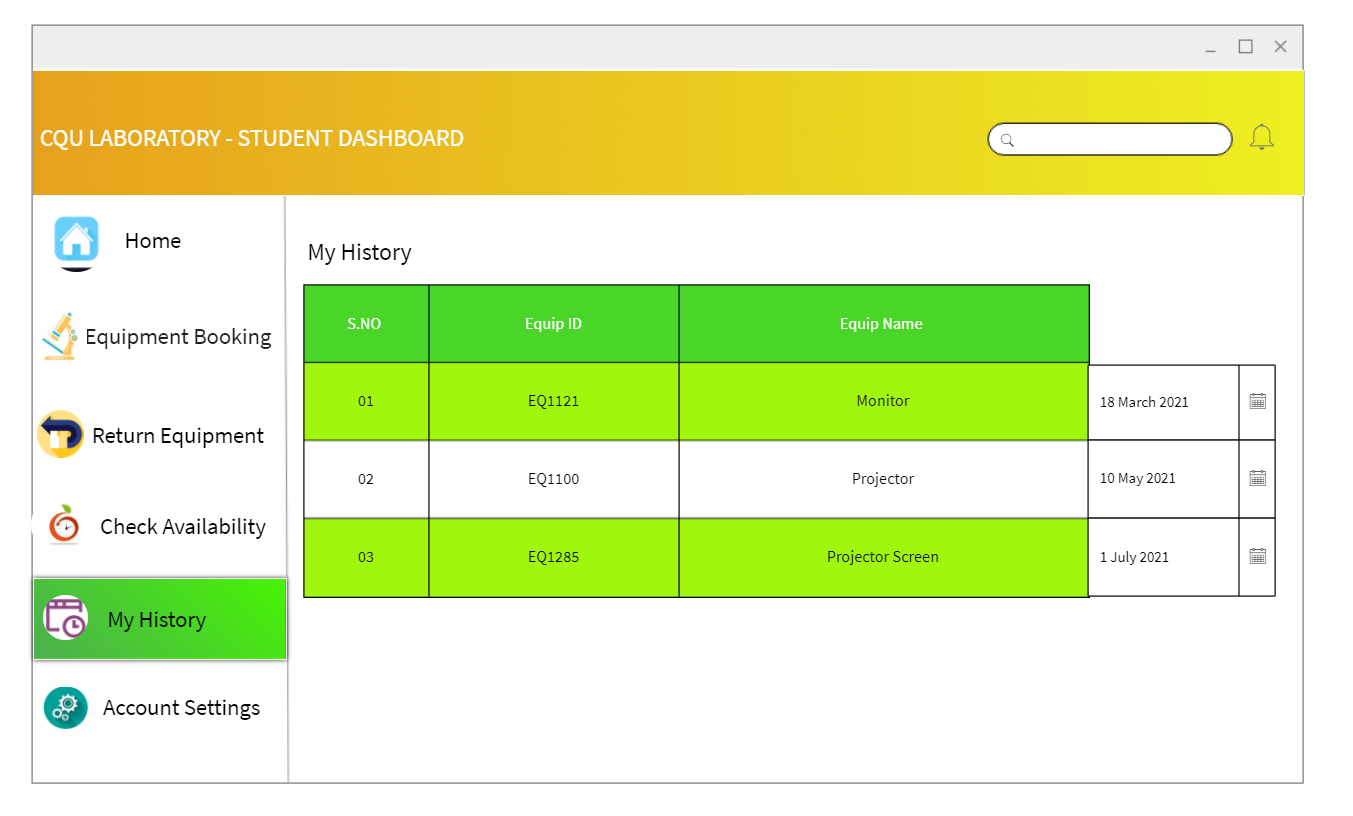
**5. MY HISTORY**

Figure 14: History Page

**6. MY SETTINGS**

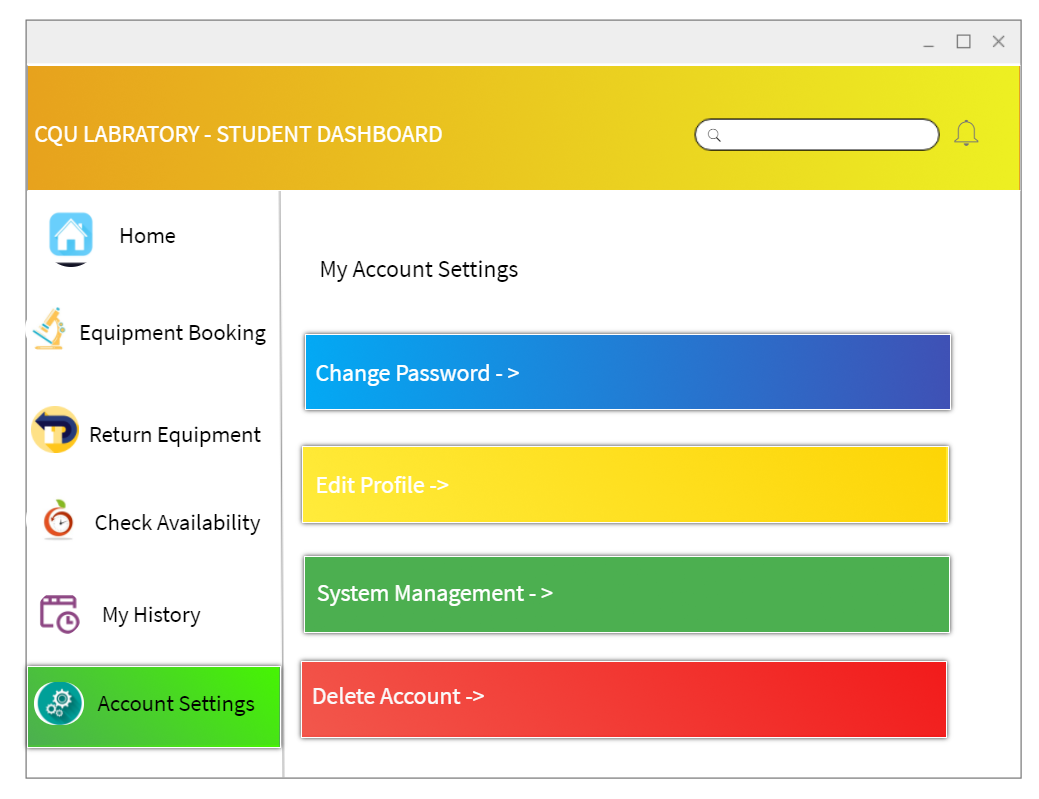


Figure 15 : My settings

**7. CHECK AVAILABILITY**

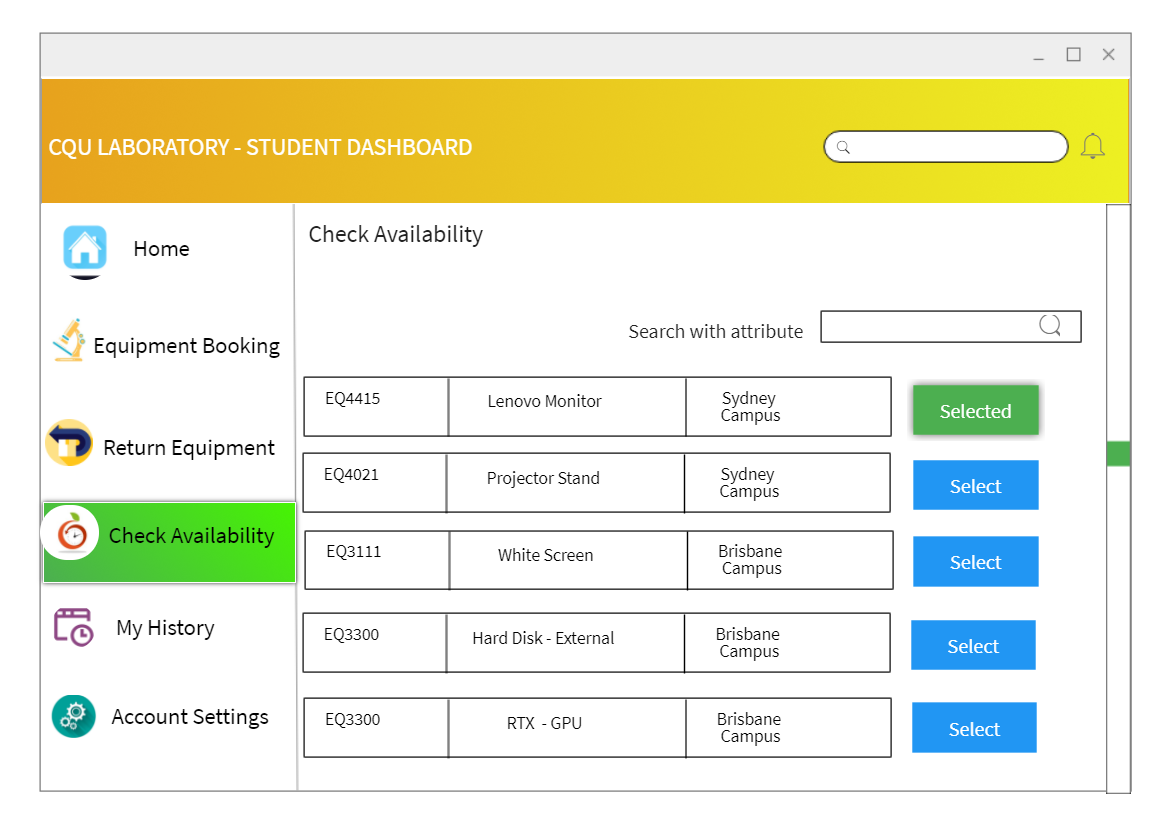


Figure 16 Student dashboard

**8. RETURN EQUIPMENT**

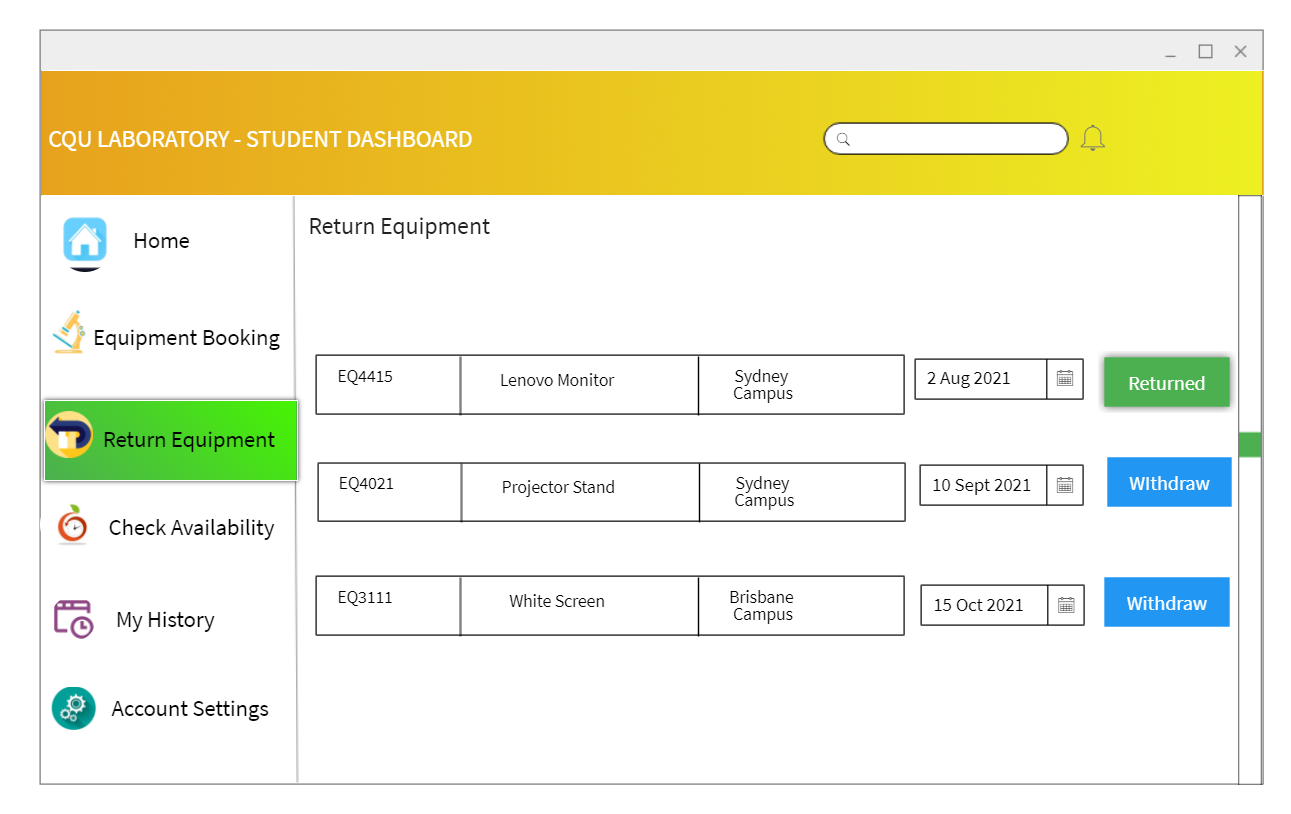


Figure 17 : Student Dashboard

**ADMIN DASHBOARDS**

**1. STUDENT SEARCH**

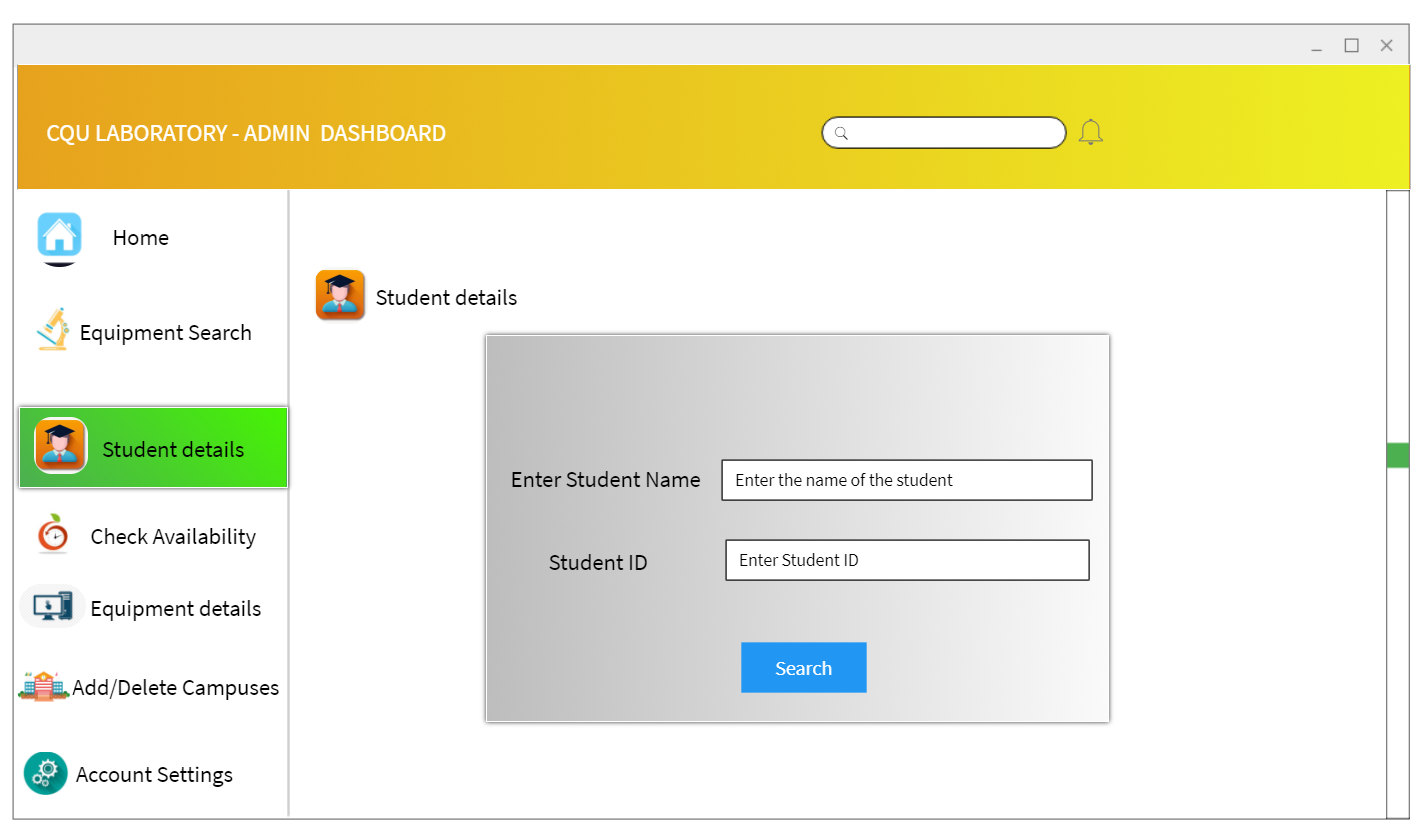


Figure 18 : Admin Dashboard, Student Search

**2. STUDENT DETAILS – ADD / DELETE**

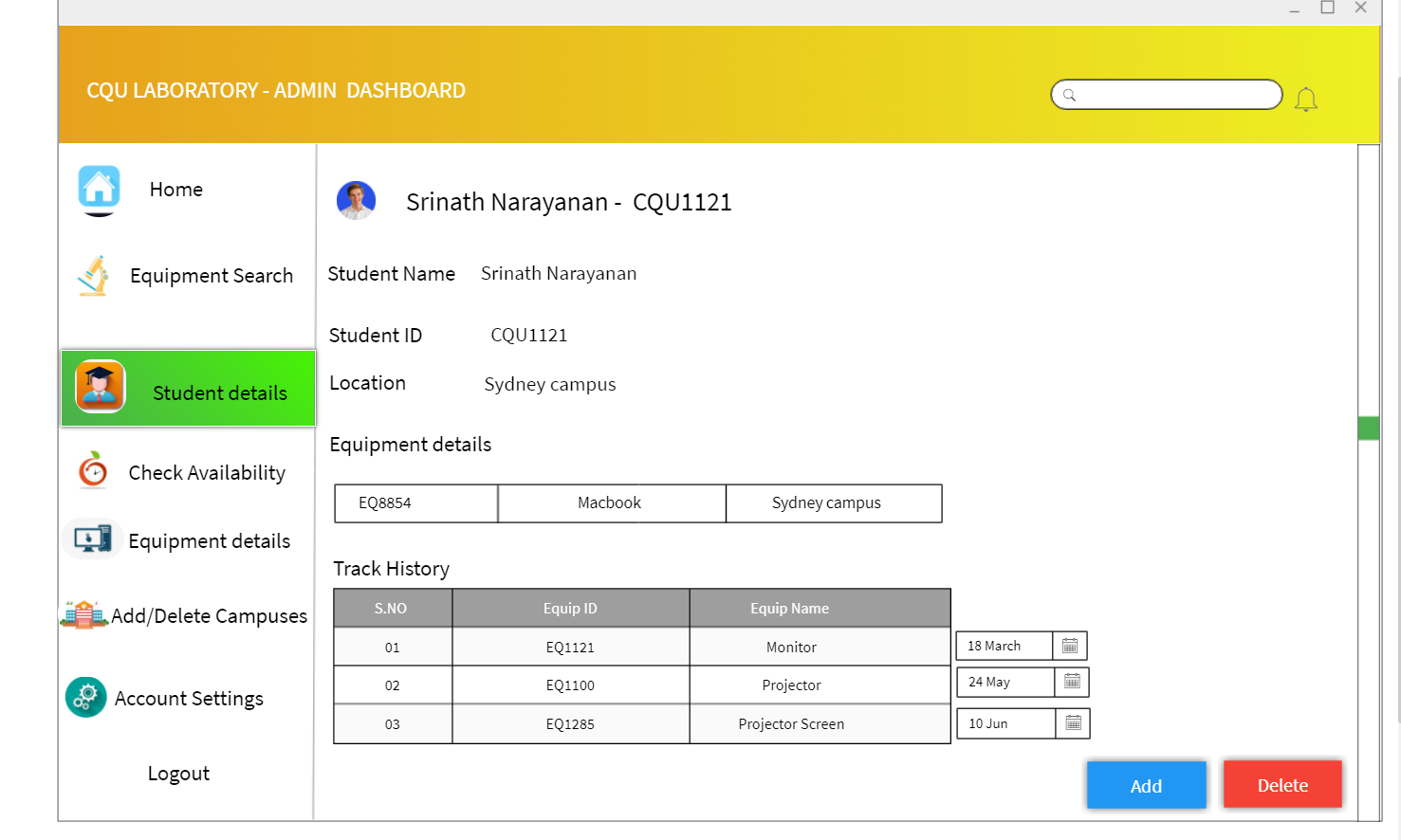


Figure 19 Add or delete student page

**3. EQUIPMENT DETAILS - ADD / DELETE**

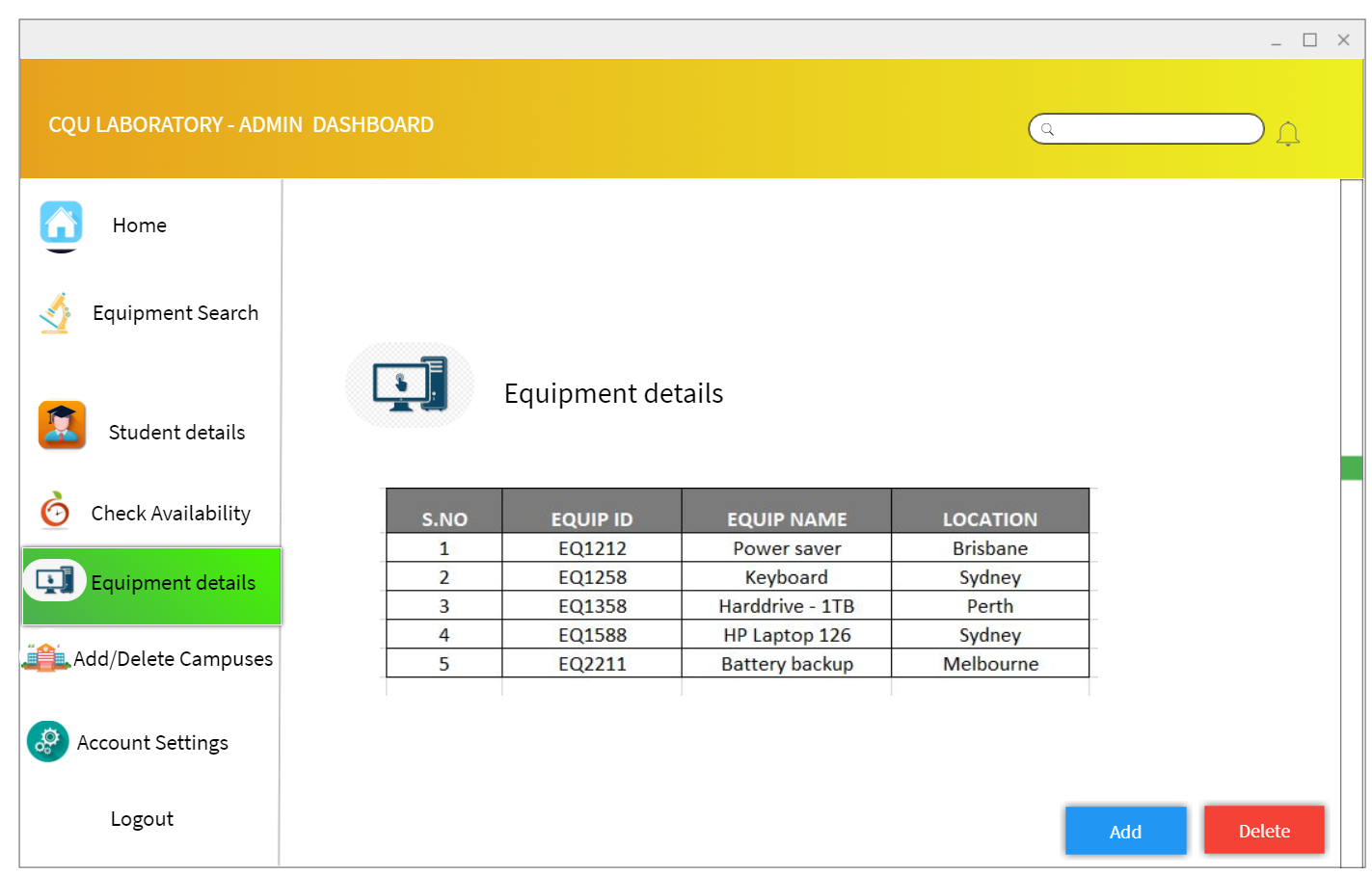


Figure 20 : Equipment details

**4. CAMPUS DETAILS – ADD / DELETE**

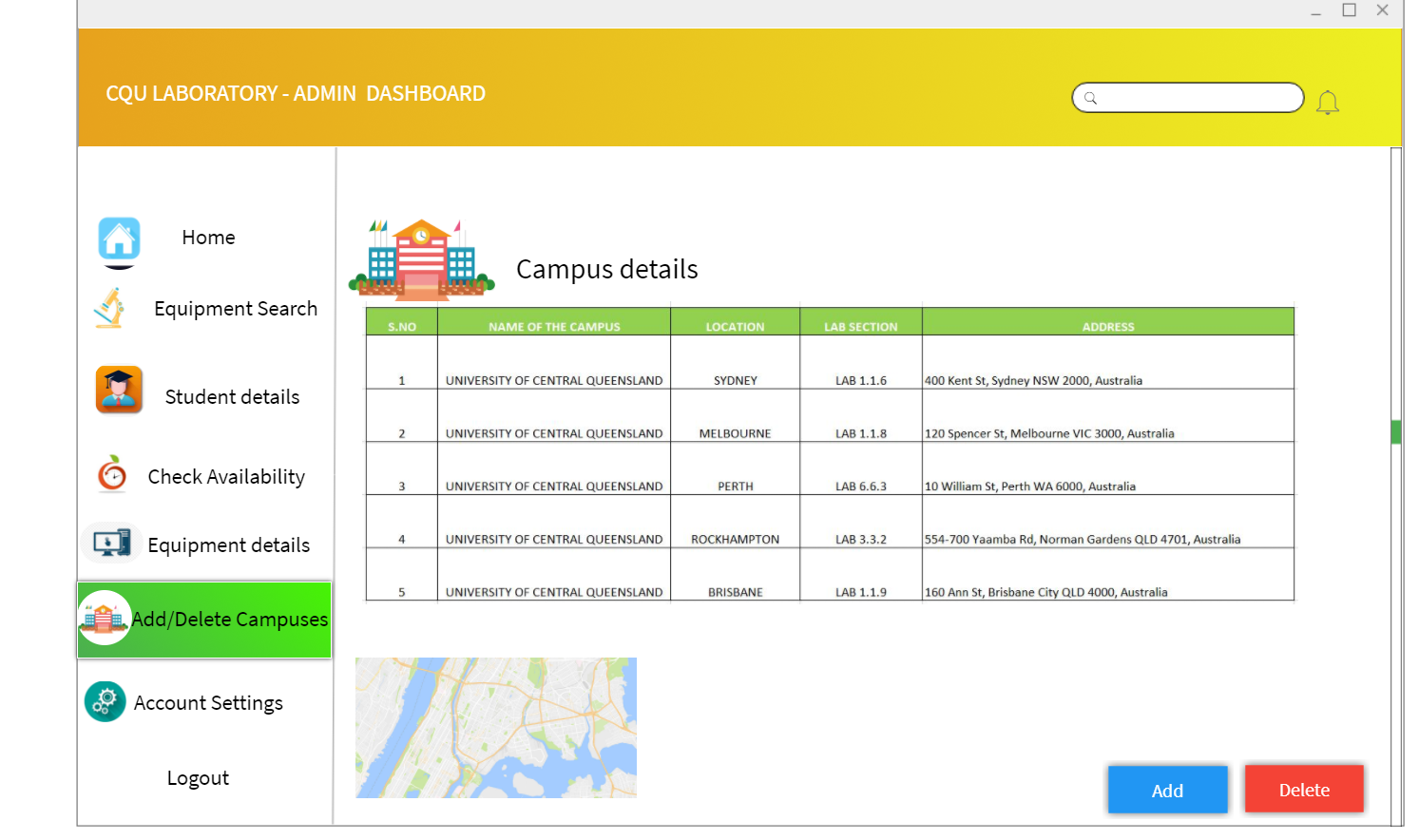


Figure 21 : Campus Details

Reference list

ALTVATER, A. (2017). *What is N-Tier Architecture? Examples, Tutorials & More*. [online] Stackify. Available at: https://stackify.com/n-tier-architecture/.

Baqais, A.A.B. and Alshayeb, M. (2018). Sequence diagram refactoring using single and hybridized algorithms. *PLOS ONE*, 13(8), p.e0202629.

Fowler, M. (2019). *Software Architecture Guide*. [online] https://martinfowler.com/architecture/. Available at: https://martinfowler.com/architecture/.

Sinhal, A. (2017). *MVC, MVP and MVVM Design Pattern*. [online] Medium. Available at: https://medium.com/@ankit.sinhal/mvc-mvp-and-mvvm-design-pattern-6e169567bbad.

slidetodoc.com. (n.d.). *Web Application Architecture multitier 2 tier 3 tier*. [online] Available at: https://slidetodoc.com/web-application-architecture-multitier-2-tier-3-tier/ [Accessed 7 Aug. 2021].

Watt, A. (2014). *Chapter 13 Database Development Process – Database Design – 2nd Edition*. [online] Opentextbc.ca. Available at: https://opentextbc.ca/dbdesign01/chapter/chapter-13-database-development-process/.