Requirements Specification for Book My Room

July 2015
Version 1.2

Adam Dubicki - V00799637
Anthony Kohan - V00740142
Chrisanda Ann - V00811197
Jose Gordillo - V00773366
Mike Weicker - V00255449
Tim Salomonsson - V00807959
Ushanth Loganathan - V00810681

Change History

Date	Version	Description	Updated By
May 19, 2015	0.1	Initial Draft	Team Book My Room
May 21, 2015	0.2	Major updates to all sections	Team Book My Room
May 23, 2015	0.3	Editing and proofreading for all sections	Team Book My Room
May 24, 2015	0.4	Document Formatting	Team Book My Room
June 4, 2015	0.5	Update Document after feedback	Team Book My Room
June 7, 2015	0.6	Minor update to edited sections after feedback	Team Book My Room
June 9, 2015	1.0	Final updates for second submission and formatting	Team Book My Room
July 10, 2015	1.1	Updated SRS to correspond with implementation	Team Book My Room
July 12, 2015	1.2	More implementation update edits	Team Book My Room

Document Approvals

Name	Role	Signature

Table of Contents

1. Introduction

1.1 Purpose

This document is the requirements specification for a McPherson library study room booking web application. It includes a system description describing the purpose and use of the application. Functional requirements and non-functional requirements describe how the application must operate, while the use case diagrams provide a reference for how the application is used. Architecture and component diagrams describe how the application will be built.

1.2 Document Conventions

In this document, Crow's notation is used for the Entity Relationship diagram. Arrows with the <<include>> tag indicate the prerequisites of a use case. This document refers to Book My Room as BMR.

1.3 Project Scope

The below sections define the problem the system solves and the need for a computerized solution.

1.3.1 Problem Definition

The library department at the University of Victoria (UVic) is unable to handle the high volume of calls that it receives to book study rooms in the library. Since the bookings are not automated, users are only able to book rooms by calling during library working hours. The library staff who answer these calls have to manually find available rooms, laptops and projectors. This is highly inefficient. Thus there is a need to develop an online automated system for booking and maintaining study rooms.

1.3.2 Need for Developing a Computerized Solution

As the process to book a study room is not computerized, users are only able to book rooms during the library’s working hours. Users have to call again if they want to edit a booking they’ve already made. Without computer assistance, monitoring these changes is difficult. When a user requests for the availability of rooms, laptops or projectors, the library staff have to go through raw data to gather the required information. Thus there is a need for a computerized solution to be implemented. Using an automated computerized system could record and keep track of all the changes made by the staff and the user, significantly reducing time spent doing manual management.

1.3.3 Application proposed

Book My Room (BMR) is an application that solves the above stated problem. It is a scheduler application for UVic's Staff, Faculty and Students to book library rooms with laptops and projectors in the McPherson Library. Users will be able to edit and cancel appointments as well as maintain a user a profile online. User accounts will be created from the Netlink ID's already in use at UVic.

1.3.4 Constraints

The BMR server must be implemented with MongoDB and Node.js. The web application must run on both android and iOS platform. The architecture must be implemented using model view controller architectural pattern.

1.4 References

1. SRS Template: http://www.jaysonjc.com/programming/how-to-write-a-software-requirements specification-srs-document.html

2. Diagrams built from Lucidchart: https://www.lucidchart.com/

2. System Description

The web application BMR will provide a library room scheduling service to students, faculty, and staff of UVic. Users will be able to sign into the BMR web application using their Netlink ID and schedule a library room with an optional laptop or projector.

2.1 Client Facility Resources

Facilities the system should work with are:

- 10 bookable study rooms
- 5 laptops
- 5 projectors

These resources are available to users on a first come first serve basis.

2.2 User Profile

The system should maintain profiles for users which contains the following relevant user data:

- First name
- Last name
- Email
- Status (student, faculty, staff, admin)
- Booked appointments
- Users phone number (optional)

Users should be able to edit their displayed email and profile.

2.3 Schedules and Booking

When a user accesses BMR, a schedule with time slots should be displayed which indicate available times for booking in a list format for the selected day.

The booking schedules are as follows:

- Weekdays: 8am-10pm
- Weekends: 11am - 6pm.

2.4 System Functionality and Flow

The user should be able to select a day, duration and equipment and be able to view the times available. Then the user will then be prompted to login with their Netlink ID, if they are not currently authenticated. All users have a limit of 1 projector and/or laptop per booking. Students can book a room for 1 hour while staff, faculty and administrators can book for up to 3 hours. The Booking is then automatically saved to their schedule. The scheduled appointment will now appear on a table in the user's profile page. ~~A user can only have 1 booking per day.~~

From their profile, users can edit their booking time which allows them to change to another available time slot. Users will have to re-request projectors and laptops due to possible scheduling conflicts. A user can also cancel their booking which if done within 5 hours of the scheduled booking will cause the user to be unable to book a room for the next 24 hours. Users can extend their booking time in the room if it is available after original scheduling. The extension must have 2 hours of prior notification. Students have an extension length of 1 hour while staff, faculty and administrators have an extension length of 3 hours.

2.5 Development Environment

The system will be developed using the MEAN stack. We will implement the database through MongoDB for agile development and scalability. The development will be created in javascript using node for implementation in HTML so that it may be viewed on any browser. The system will be built on UVic lab computers and on Team Book My Room's personal computers.

2.6 System Environment

The system will be hosted on a server with minimal downtime as to be accessible as often as possible. The system will invoke a regular backup system with redundancy for minimal data loss potential. The system will interface with any major web browser. The system will be tested on Chrome and Firefox.. Other web browser interfaces may be functional but are unsupported. The system is primarily designed to be viewed on mobile devices so it will therefore be tested and designed primarily for small screen mobile devices. The system will be primarily developed for android based devices . The system will be tested on Android, iOS and Windows Phone environments.

3. Functional Requirements

The application's functional requirements are all web application requirements.

- The system must provide a medium for the vendors to convey the schedule to the users and edit appointments made by the users.
- The system must allow users to edit their own profile.
- The system must allow admins to edit room schedules, user appointments, and profiles.
- The system must display daily availabilities.

3.1 System Features

3.1.1 Mobile Library Study Room Booking Application

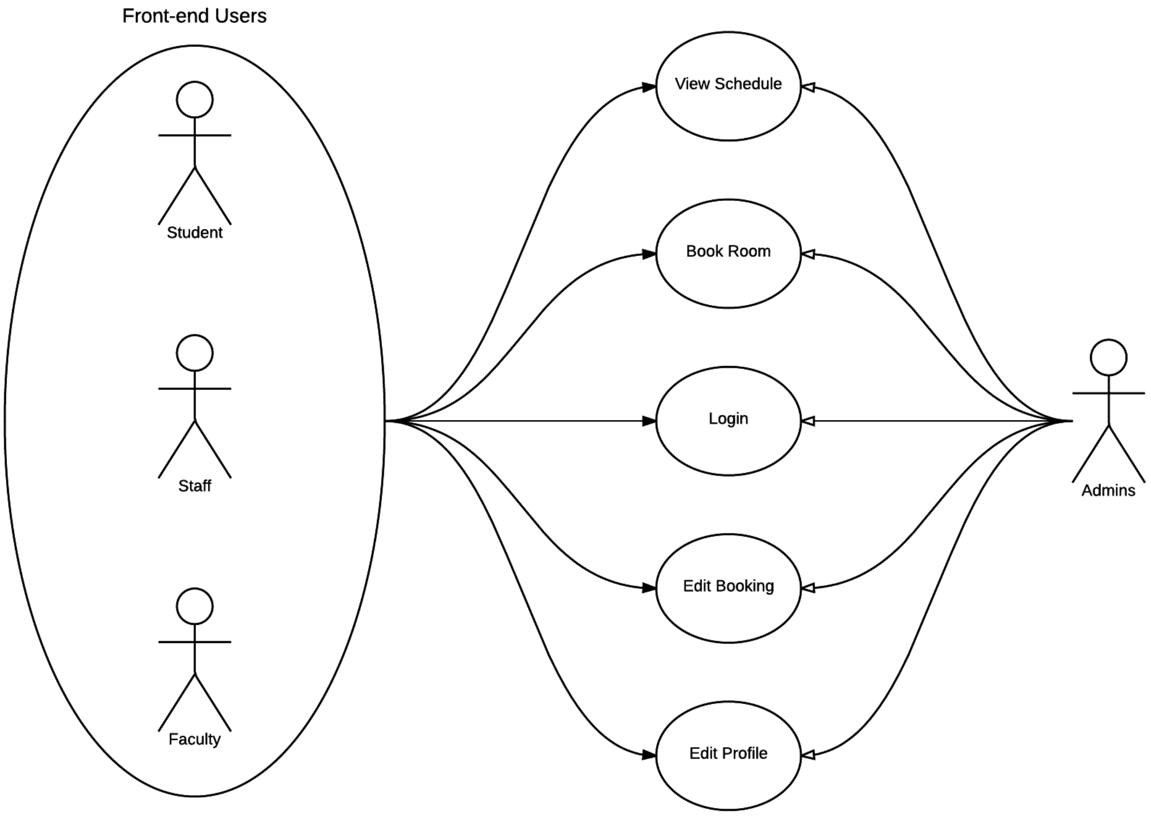
A web application targeting mobile audience which allows users to book, edit, and cancel study rooms bookings in the University's library. Users can also book laptops and/or projectors if they have existing room bookings.

3.1.2 User Profile

The application will maintain a profile for all the users with their biographic information and a log of upcoming bookings. Users can edit and update their profiles.

3.2 Use Cases

3.2.1 Use Case Diagram



3.2.2 Use Case 1 - View Schedule

ID	001
Description	An actor desires to check the available time slots for a given day. They then select a day and view the schedule.
Actors	Staff, Students, Faculty and Administrator
Preconditions	The user must be logged in.
Basic Steps	User opens the web page. The user selects the desired day to view. The schedule is displayed.
Alternate Steps	
Exceptions	If the schedule is altered after the user submits a view request then the user will need to resubmit a view request to get the latest version of the schedule.
Business validations/Rules	
Postconditions	None, viewing the schedule does not change the schedule.

3.2.3 Use Case 2 - Log in

ID	002
Description	A user must log in to the system to affect it. This use case describes the process of logging in to the system.
Actors	Staff, Students, Faculty and Administrator
Preconditions	None, users may initiate a login from any stage.
Basic Steps	The user is prompted to login. The user enters their login credentials. The credentials are checked in the system. The user is logged in.
Alternate Steps	
Exceptions	
Business validations/Rules	Users may only log in to their own accounts.
Postconditions	The user now has access to the system as specified by the type of account they have just logged in to.

3.2.4 Use Case 3 - Book Room

ID	003
Description	Selecting specifics and booking a room.
Actors	Staff, Students, Faculty and Administrator
Preconditions	User must be logged in to book a room.
Basic Steps	An actor decides that they want to book a specific day. The user is displayed the default booking details. The default settings are a 1/2 hour time slot with no extra equipment. The user is given the option to change these defaults. A user may change the duration, add a projector or laptop. The user is then prompted to confirm the booking and then the room is booked.
Alternate Steps	If the user is not already logged in the system prompts the user to log in before selecting the specific room details.
Exceptions	If a user attempts to book a room which causes a scheduling conflict, then they will be given a scheduling conflict error.
Business validations/Rules	Students and staff will be able to book maximum of 1 or 3 hour time slots respectively and will not be able to book back to back rooms until 2 hours before the start of the subsequent hour.
Postconditions	The user has one additional booking.

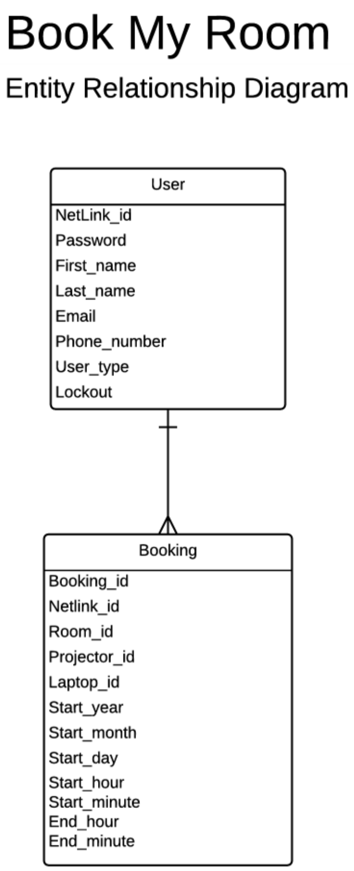
3.2.5 Use Case 4 - Edit Room Booking

ID	004
Description	A user may edit an already existing room booking.
Actors	Staff, Students, Faculty and Administrator
Preconditions	A room has been booked by the user. The User must be logged in.
Basic Steps	The user views their current bookings. The user makes changes to their booking. These changes may include extending the booking or canceling the booking all together. The user then confirms the changes to their booking.
Alternate Steps	A user could change their mind about editing their booking and just return to their booking list without making any changes..
Exceptions	A user will be unable to select an extension if they are not within 2 hours of the start of the booking. If the user attempts to cancel a booking within 5 hours of the start time they will receive a warning of being locked out of the system for 24 hours.
Business validations/Rules	Extensions cannot be created more than two hours before the extension time block. Users will be issued a 24 hour lockout penalty if a room is canceled less than 5 hours before the start of the booking.
Postconditions	The room booking is edited.

3.2.6 Use Case 5 - Edit Profile

ID	005
Description	Users can edit their profile information, such as their preferred email address for notifications.
Actors	Staff, Students, Faculty and Administrator
Preconditions	User must be logged in.
Basic Steps	User selects their profile. User edits one or more available field. The user confirms their update and the profile is updated.
Alternate Steps	
Exceptions	
Business validations/Rules	No SQL Injection Vulnerabilities
Postconditions	The user profile has new values.

3.3 Entity Relationship Diagram



Book My Room

Entity Relationship Diagram

3.4 Data Dictionary

3.4.1 User

The user is the person using the program to book a room, laptop and/or projector. Users will obtain their profile by logging in with their already existing NetLink ID.

Attribute	**Type**	**Optional ?**	**Notes**
Netlink_id	String	No	Username used by the UVic students. This username must exist in the UVic servers.
Password	String	No	Password used by the UVic students. This password must exist in the UVic servers. Encrypted.
Name	String	No	First name of the user. This information is taken from the UVic Servers.
Email	String	No	Preferred email address. The default value is the UVic email address.
Phone_number	String	Yes	User’s phone number used as contact information.
User_type	Integer	No	Identifies the user as student, staff, or faculty.
Lockout	String	Yes	Shows if user is locked out of system.

3.4.2 Booking appointments

The Room Appointments keep track of when a specific room is booked.

Attribute	**Type**	**Optional?**	**Notes**
booking_id	String	No	A unique ID for each room appointment. The Appointment_id includes a timestamp and a correlative number.
room_id	Integer	No	A unique ID for each room.
laptop_id	Integer	Yes	Indicates which laptop was booked along with the corresponding room. It is possible for the user to book a room without booking a laptop.
projector_id	Integer	Yes	Indicates which projector was booked along with the corresponding room. It is possible for the user to book a room without booking a projector.
netlink_id	String	No	Indicates which user booked the room
start_day	Integer	No	Gives an integer for the day of the booking
start_hour	Integer	No	Gives an integer for the start hour of the booking
start_minute	Integer	No	Gives an integer for the start minute of the booking
start_month	Integer	No	Gives an integer for the start month of the booking
start_year	Integer	No	Gives an integer for the start year of the booking
end_minute	Integer	No	Gives an integer for the end year of the booking
end_hour	Integer	No	Gives an integer for the end hour of the booking

4. External Interface Requirements

External interfaces will include a web based input and output only. Users and administrators will be able to access the system through any standard web browser.

5. Technical Requirements (Non functional)

Each of the following requirements are non-functional requirements specified by both the client and the development team.

5.1 Performance

Given a high quality network connection our system should respond to simple task requests such as view schedule, edit profile, create booking, and edit booking within 5 seconds.

5.2 Scalability

BMR will scale up if more resources (rooms, laptops, projectors) are required by the system. This will also scale down if any one resource component becomes temporarily unavailable.

5.3 Security

BMR will have sensitive user data and will require encryption to be implemented on all sensitive data. BMR will encrypt users emails address, passwords and phone numbers.

5.4 Flexibility

BMR will be functional on most major web browsers. This includes, Chrome, Firefox, Internet Explorer, Safari and Opera.

5.5 Maintainability

The database must be easy to update in real time. Administrators can update room and equipment availability.

5.6 Usability

BMR will have an intuitive design and simple user interface which will make it easy to schedule a room. It will have the same resolution on different mobile platforms.

5.7 Availability

BMR may be unavailable from 2:00am to 2:30am (GMT-8) each day for updating and maintenance. Minor downtime during service hours is acceptable during initial beta testing.

5.8 Compatibility

The system will be compatible with web browsers such as Internet Explorer, Google Chrome, Safari and Mozilla Firefox. The system must be compatible with mobile platforms such as iOS, Android and Windows Phone.

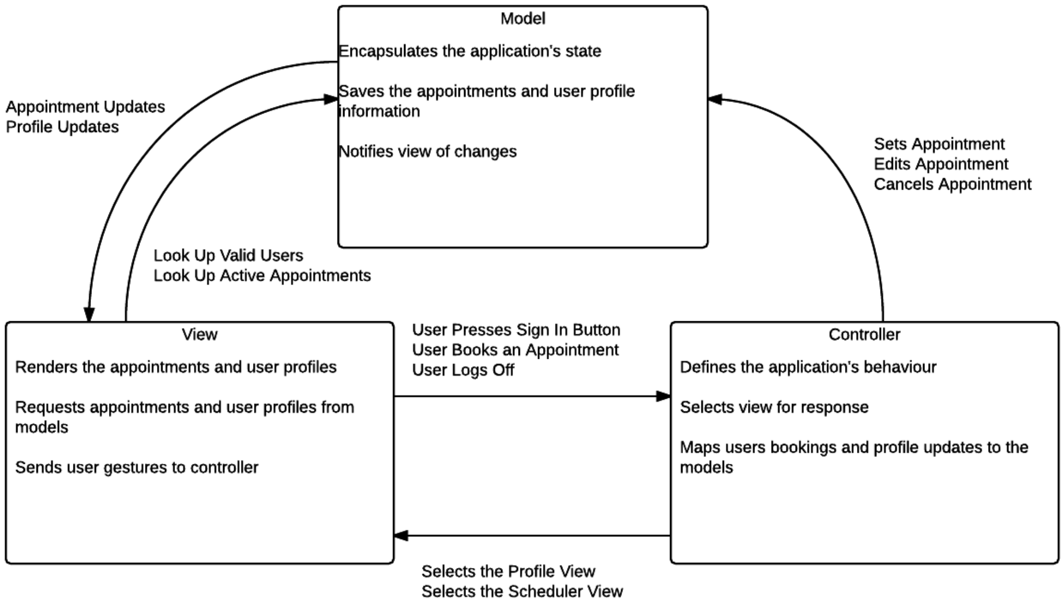
6. Open Issues

Future updates to this system could include features such as card readers at the entrance of each study room to check if users actually showed up for their scheduled appointments. This could be used to enforce more strict consequences for misuse of the system. This would also allow the university to hold specific users accountable for the state of the rooms. The university library has several different types of rooms, for example, some have outdoor facing windows and some do not. In future implementations we could have preferences for specific room types. In future versions the system, user biographic information could allow you to create groups which would automatically notify other group members of a booking for ongoing group projects. This system could integrate with common calendar applications such as google calendar, outlook or iCal.

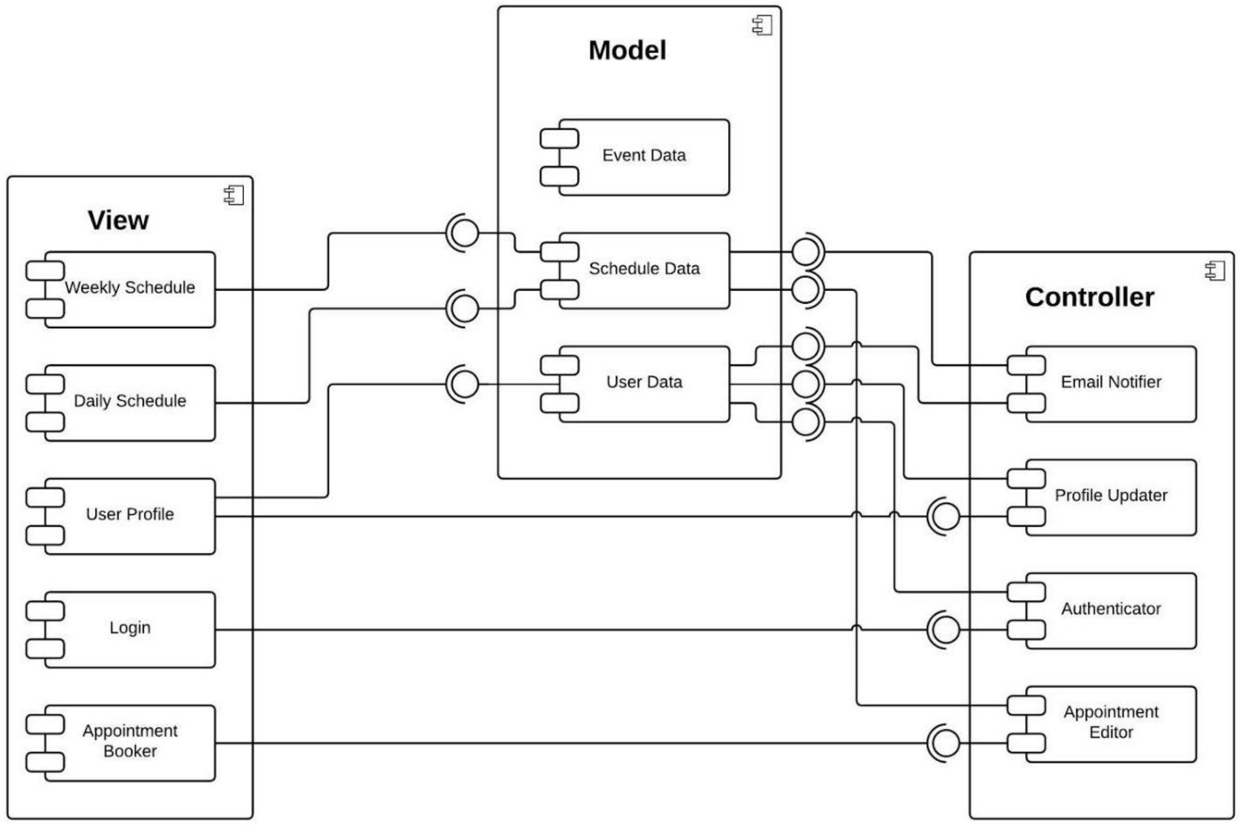
Future native applications for mobile devices could be implemented along side of this web application to be customized for each major type of mobile device. These apps could integrate directly with other native system devices and push notifications to the users. Users which would want to create a recurring event could set up temporary recurring bookings. This would require user confirmation for each subsequent booking.

With the implementation of ID card active logging upon arrival and possibly when users leave the rooms we could implement a live view of occupancy information for administrators.

7. Appendix A: Model-View-Controller Diagram


Model
Encapsulates the application's state
Saves the appointments and user profile information
Notifies view of changes
Appointment Updates
Profile Updates
Look Up Valid Users
Look Up Active Appointments
Sets Appointment
Edits Appointment
Cancels Appointment
View
Renders the appointments and user profiles
Requests appointments and user profiles from models
Sends user gestures to controller
User Presses Sign In Button
User Books an Appointment
User Logs Off
Controller
Defines the application's behaviour
Selects view for response
Maps users bookings and profile updates to the models
Selects the Profile View
Selects the Scheduler View

8. Appendix B: Component Diagram



Component Description

View

Daily Schedule: These views simply pull from “Schedule Data”, a collection of booking objects, to show what appointments have been made for the day ~~or week~~. If the user is logged in and is an admin, each appointment will be editable.

User Profile: When using this view, the user’s profile information is pulled from “User Data”. If the user alters the information, the changes will be sent to the “Profile Updater”.

Login: When the user inputs their information, the data will be sent to “Authenticator” for verification.

Appointment Booker: Where the user books a room, laptop, or projector. The page will be different depending on whether the logged in user is a student or a staff member.

Model

Event Data: Stores all system events. For example, a user booking or editing an appointment or an admin deleting a user would be events. This component is connected to practically every other component in the system so those connections are not shown in the diagram.

Schedule Data: Provides access to the room, laptop, and projector schedule information stored in the database. All queries to the schedules are done through this component.

Users Data: Provides access to the user information stored in the database. All queries to the user information is done through this component.

Controller

Email Notifier: Reads the booking times from “Schedule Data” and email addresses from “User Data” and sends a notification email to a user before their booking.

Profile Updater: Takes information from the “User Profile” view when a user edits their profile, encrypts it, and sends it to “User Data” to be stored in the database.

Authenticator: When a user submits their login information, the authenticator encrypts the information and compares it against the data stored in “User Data”. If it is valid, the authenticator determines the route the user will use depending on whether he or she is a student or a staff member.

Appointment Editor: When a user books or edits an appointment, this component takes the information, packages it, and sends it to be stored in “Schedule Data”.