Contents:

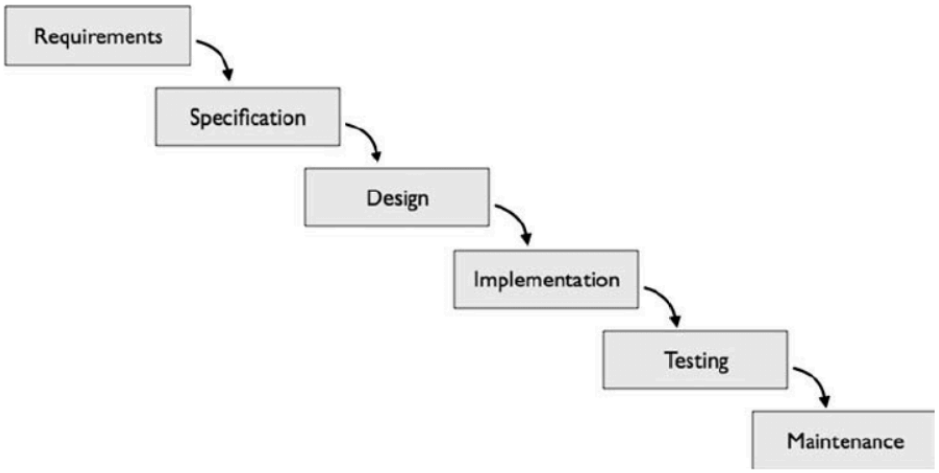
Student information Kiosk

Introduction:

For this assignment, we are assigned a project where we will be designing a Kiosk system as a group that allows students to access an electronic one stop service shop. The purpose of the student information kiosk is to provide up to date information to the students. The Information that will be provided will include events, student discounts, course details that are taking place on the university. It is essential that the university's system focuses on enabling the students from both overseas and home to gain access to the application and have capability of its using the it's functionalities.

Software Development Lifecycle

Waterfall Diagram:



In order to create a well performing student information system, we followed the steps in the following waterfall software development life cycle. This will allow us to effectively plan stages and move towards a common goal, it will force structure and will be simple to follow and arrange designated tasks.

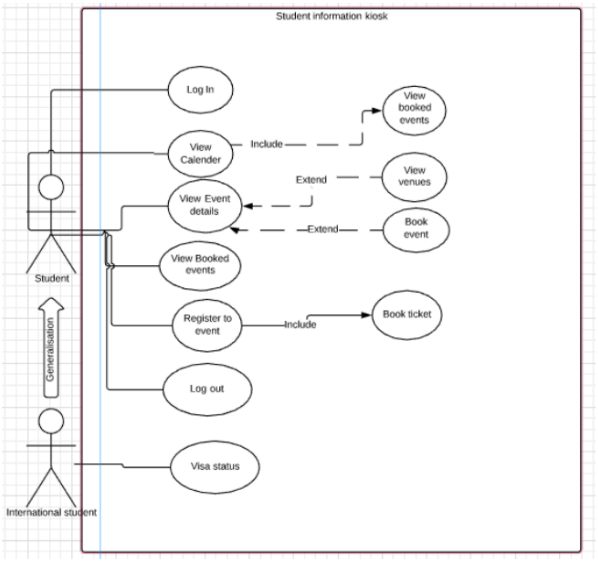
List of requirements (Non functional & Functional):

Requirements summary			
Non-Functional-Requirements:			
ID:	**Name:**	**P:**	**Owner:**
#NF201	Access permissions for the particular system information can only be changed by system admin.		**A**
#NF202	Application should be accessible on 3g/4g mobile connections.		**A/S**
#NF203	The response time should be no longer than 5 seconds on the time it takes to a view a page.		**A/S**
#NF204	Modified data within the database should be updated for all of the users accessing it within 5 seconds.		**A/S**
#NF205	Ability of the system to be able to operate correctly/accordingly, if large number of users login.		**A/S**
#NF206	System returns error when user has entered wrong password/username		**A/S**
#NF207	System returns error when user has entered wrong data type within the field.		**A/S**
#NF208	System requires validation of fields (input) by student user		**S**
#NF209	System requires validation of fields (input) by admin user		**A**
#NF210	Users easily able to navigate by a navigation menu & buttons provided within the application		**A/S**
#NF211	Users able to access a service page, displaying various services within the University and outside.		**A/S**
#NF212	Layout and colour scheme must be user friendly and fully visible for the user in various lighting conditions		**A/S**

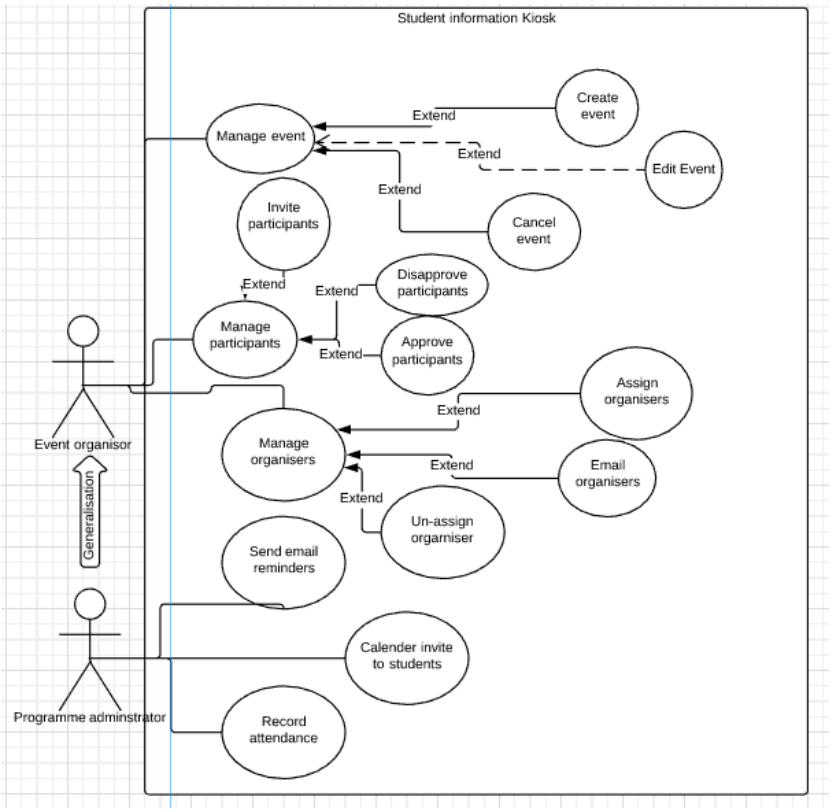
	Functional Requirements:		
#F301	Registered users must be able to log in to the application via the login page and be redirected.		**A/S**
#F302	A student user must be able book an available event		**S**
#F303	A student user must be able to see their own event booking		**S**
#F304	Users must be able to view the event venues		**A/S**
#F305	Student users should be able to delete their own event bookings		**S**
#F306	A student must be able to view their calendar with their booked events displayed		**S**
#F307	Users must be able log out of kiosk & be redirected back to login page.		**A/S**
#F308	Admin users must be able to delete zero to many users event bookings		**A**
#F309	Admin users must be able to have the option of functionality to modify all users event bookings		**A**
#F310	Admin users must be able to add an event/venue		**A**
#F311	Admin users must be able to create user accounts & modify		**A**

UML Diagram

In the UML diagram i have shown below, I have presented a plan on the functions a student may have on student information kiosk. This includes the student being to look at their calendar for upcoming events. The student being able to view event details, register to event and check their booking status. International students will also be able to get forwarded information about their visa

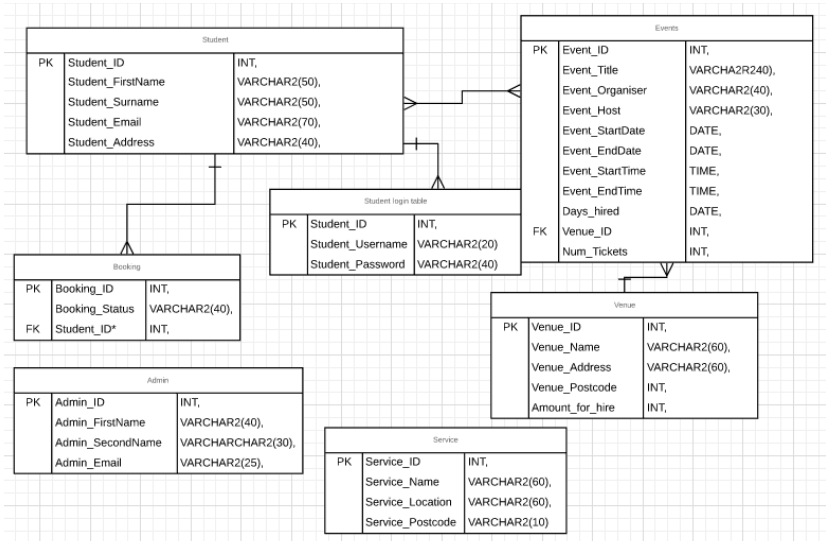


In the UML diagram i have shown below. I have presented how a plan on how it would look like management an event as an event organiser. In the use case diagram it shows that the event organiser is able to manage events and manage participants. A generalisation was also done to present the programme administrator who sends out emails to students and records attendance.


Student information Kiosk
Create event
Extend
Manage event
Extend
Edit Event
Extend
Invite participants
Cancel event
Extend
Extend
Disapprove participants
Manage participants
Extend
Approve participants
Assign organisers
Extend
Event organisor
Manage organisers
Extend
Email organisers
Extend
Generalisation
Un-assign orgarniser
Send email reminders
Calender invite to students
Programme adminstrator
Record attendance

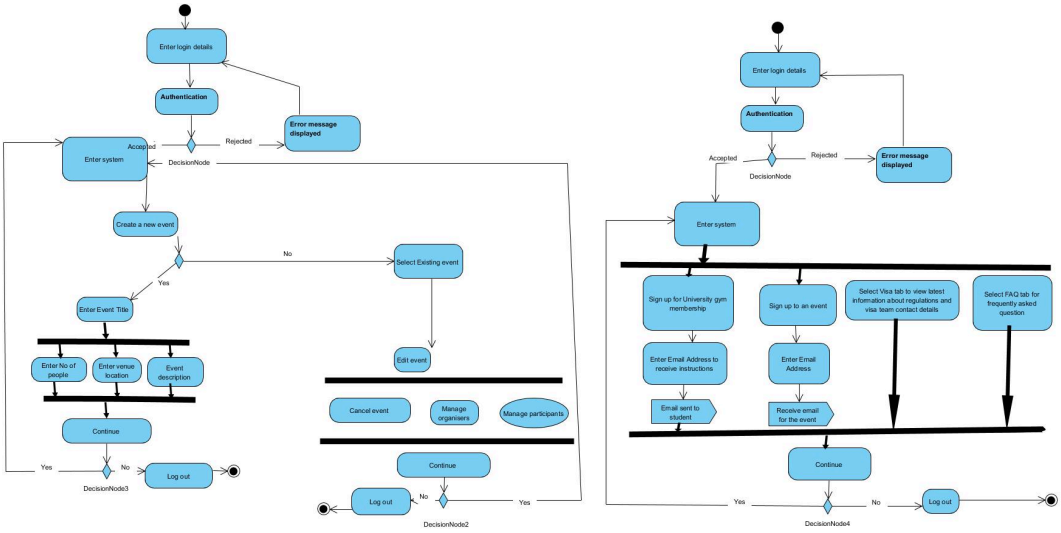
ERD

In the entity relationship diagram I have presented all the relationships in our kiosk database system. Student can join many events and vice versa. Data types have been chosen for the relevant columns and keys have also been implemented.



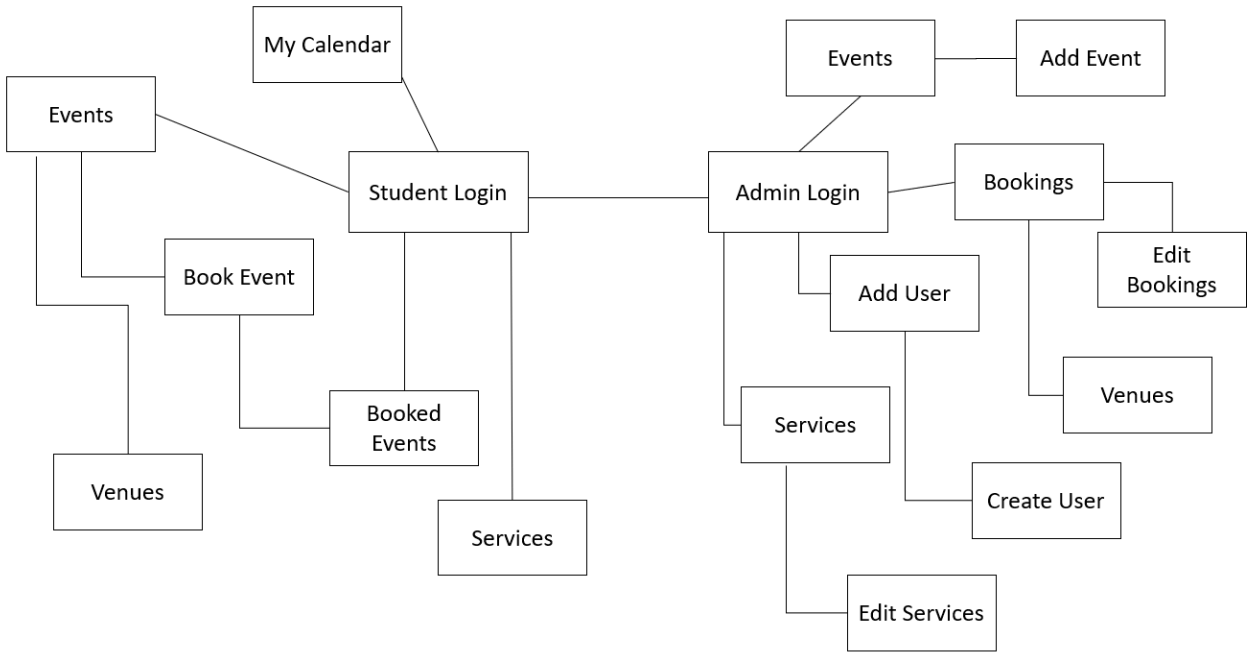
Activity diagram

This activity diagram below provides further details following on from our use case diagram, on how we want the student information kiosk to work.



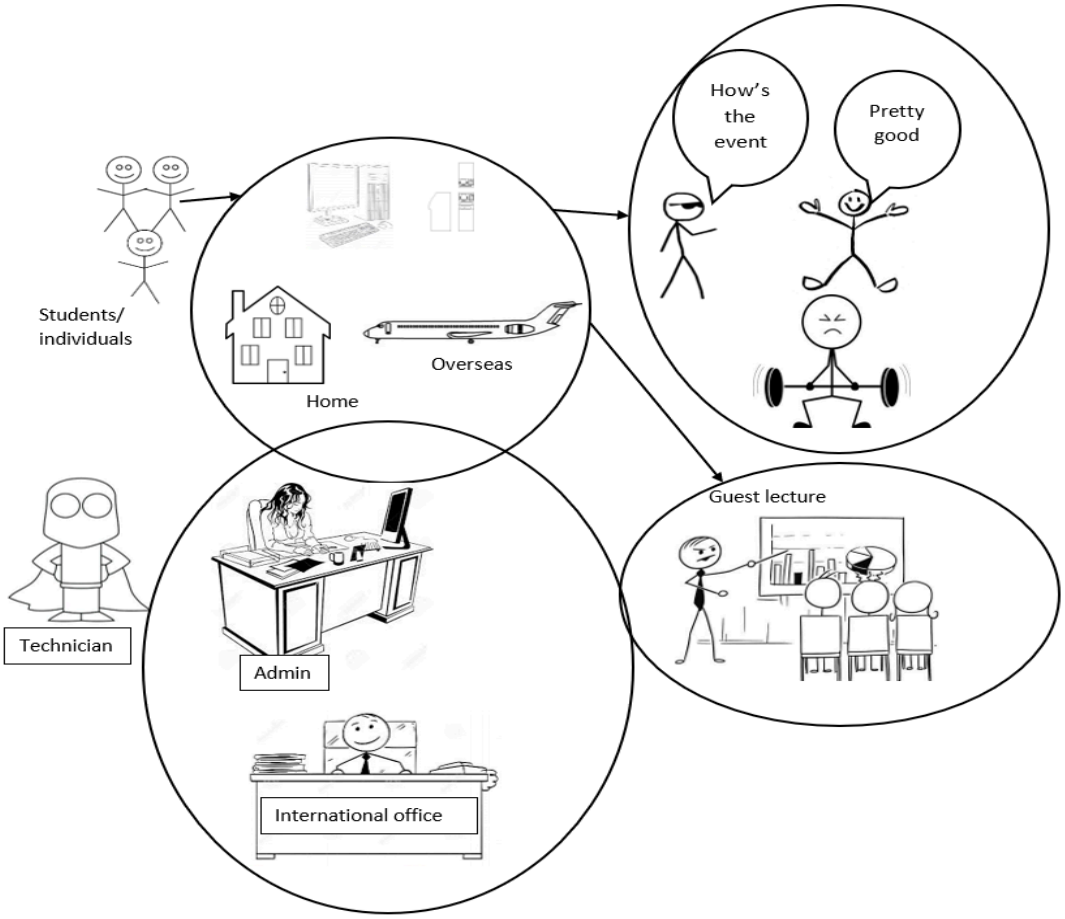
Navigation mapping:

This is the navigation map for our Student Information System. It shows the pages that can be accessed by both the user and the admin This navigation map also shows the prevellages the students have compared to the admin



Rich picture:

We created this rich picture as a way to acknowledge, explore and define different situations a define them through a diagram. This rich picture shows that students/ individuals are able to access the kiosk. This includes students both from home and overseas.The student kiosk will have events such as sports etc. The rich picture states that the admin is able to access the kiosk, but he will have more privileges than the students. The guests are also able to use the kiosk but they can only interact with it, they are unable to book. The technician is there incase of any technical failure about the Student Information System.



Root definition:

A operated university-based system, to allow students to book & receive up to date information about all events on campus & to provide sources of information in regards to Luton & it's opportunities, by the means of current students having the capability to have access to a one stop service shop. The purpose of this is to increase the awareness of the events/services, whilst allowing students to have easy access to them

Catwoe:

C Customers: Students
A Actors: University staff
T Transformations: Students -> awareness of events & services in regards to the relevancy of the campus and Luton as a whole.
W Worldview:Belief that providing up to date information of all events & services on campus & in the Luton community is a good way to get students to be aware of them, involved & to take part.

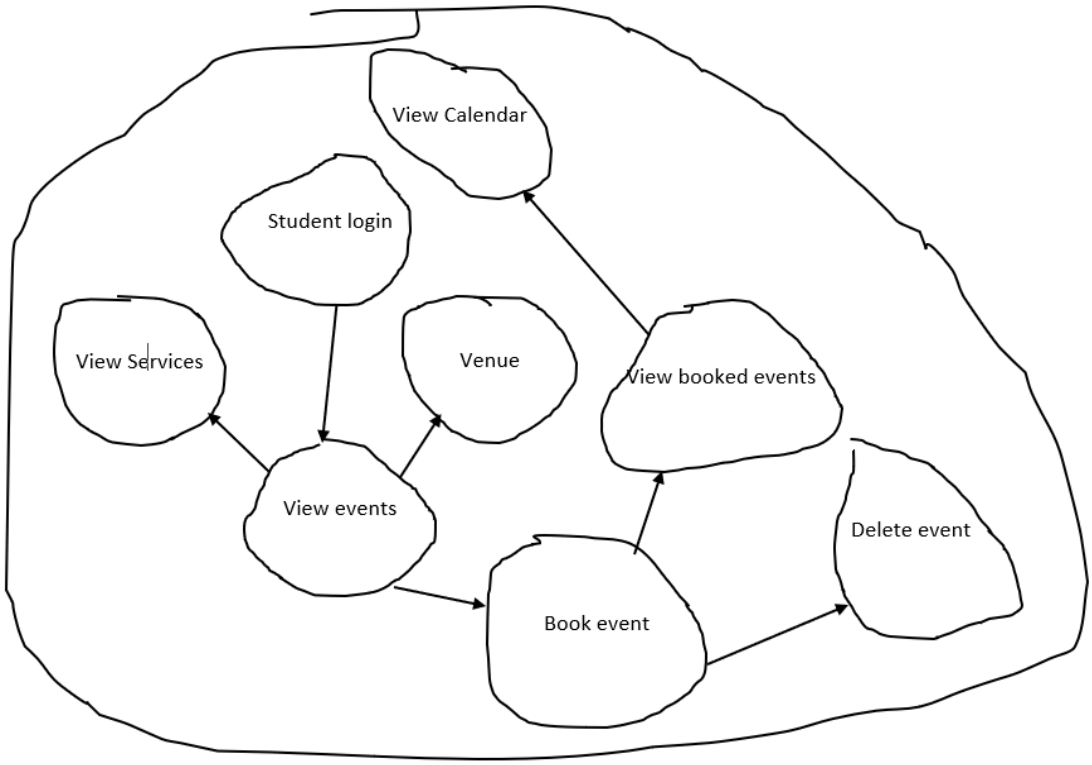
O Owners: The University
E Environmental Constraints: University (term periods)

Performance Measurements:

E1 – Efficacy
The student information system performs very well and it meets the students' goals to update the them about every new information that takes place at the university like events.
E2 - Efficiency .
students at the campus are satisfied with the services provided as it helps improve their university life and helped let them network with others
E3 - Effectiveness
The students at the campus find the student information system very useful. This is because the system helps to make harder activities easier for the students.

Monitor and control is vital, In which case we should establish performance measures, we should consider to use the three E'S, (efficacy, efficiency, effectiveness.) The purpose of this to understand our system and be able to evaluate the performance and to also regulate itself if performance is not met.

Conceptual Model:



Communication

Every Time before we met we had each other's contacts so we used social media as our means of communication. This helped us to make plans for our group meetings.and whenever a member of our group came up with an idea they could message it to the rest of the group members if they were not in a group meeting.

Week by week steps - Meeting log:

Week 1
In the first week, we started by distributing different activities within the group. Each student was allocated a task to work on.
We then started working on the prototype for our student information system by planning our navigation and pages. We did this by using a powerpoint presentation and also creating a storyboard designs for our kiosk. One of the key tasks we did is identifying functional and non-functional requirements.

Week 2
In week 2 with started to focus on the planning of our application. We did this by using UML, the first model we worked on was a use case diagram we modelled this according to have we wanted on the application by identifying actors and functions. We identified entities and attributes which we then normalised to create an ERM which identifies the relationships of the kiosk system.

Week 3:
In week 3 we created an activity diagram presenting the flow of the kiosk system. We also created the rich picture to illustrate the structure, process, soft facts and the environment.
We began started working on the report. Altogether we used a rich picture to express the situation. We also did triple E and Catwoe.

Week 4:
We started to create Oracle Apex accounts and started to follow the practicals in orders to gain a better understanding & knowledge of the web development tool. We then started to learn how to navigate through oracle apex and how the creation process of the database operates & connects with application side of Apex.

Week 5:
Since we completed our entity relationship diagrams and normalisation we prepared to make our tables through sql using sql commands & the object browser within oracle. The powerful web application development tool did the necessary jobs by having the capability to be able to select identifier keys and the relationships for us.

Week 6:
These are some of the ideas that we had for the student kiosk. Most of them went through and some did not. It is also where we did our brainstorming for the kiosk and the kiosk prototype.

Week 7:

We started implementing some of the ideas that we had during our brainstorm group sessions/meetings.

Week 8:

We then finally started to create our prototype application, we came to the conclusion to create two applications; one for the student user & for the admin user. We then back to our user requirements and identified which particular tables, functionalities and options, (permissions) are required in the student app and which ones are needed in the admin app. We first started to display the relevant information we wanted onto modal & non-modal; forms,interactive forms, interactive grids. We started by adding an event page by using sql query (SELECT STATEMENT) on the page designer to select our columns that we wanted to display on to the grid we, used a inner join to join two columns from two different tables together. We added a button region to our page allowing the student user to book an available event from a select list with a list of values saved from our shared components. The select list consists of the following sql query.

```
'select NAME as display,
        EVENT_ID as return
 from event
 order by 1'
```

This query generates data from the event table into the select list, allowing the user to select an event. We made other buttons in which redirect to another page we also made a submit button in which uses an insert statement in order to insert the data into the table.

Week 8.1:
After completing the booking page, we again used our functional requirements in order for us to identify what we need to create next. We started to create a 'booked events' page, we did this by displaying tables together into a interactive report. We used an inner join and made sure that the student that is logged in is only able to see his/her's events by using 'app_user'. Below is the sql query that is used for the source.

```
select b.B_ID, b.EVENT_ID, e.NAME
FROM BOOKING b
inner join EVENT e
on b.EVENT_ID = e.EVENT_ID
where b.USERNAME = :APP_USER
```

We created a button on the side of the row to be accessible for student users to be able to delete there booking we did this by using a region button directly the page to a form where they can delete the select booking, this works by a delete statement.

Week 8.2:
Now we decided to create a calendar for student users to be able to view their booked events on the displayed calendar.we did this by creating a calendar form and then we linked the page to a booking table in our database. We then created the calendar in a way that when a user logs in they can only see their bookings that they made on the calendar. We did this by using the :APP_USER query.

The last page we created for the student application side was a service page to allow students to be able to have information on relevant services. This was simply created using a report and to source it to display the data in our database tables.

Week 8.3:
We began making the admin application side. Then admin event page pretty much generalises off student event page however instead of being able to book an event they only having the two options of being able to delete any users booked event. We created the venue page by populating data from tables by the use of a sql select statement & a join. The venue ID will be used as the primary key however will be used as a foreign key in the events table. Admin can create a new venue ID so that they are able to create a new event.

Created a form page to allow Admins to create Student accounts, allowing the functionality for them to be able to add users and insert the data into our tables.

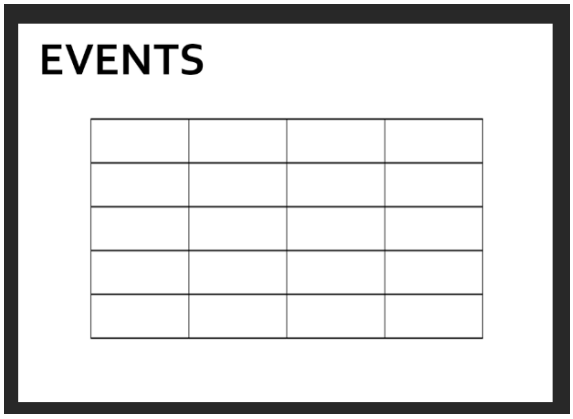
Added a service page giving the admin an selection to add a service.

Week 8.4:
Ensuring Validation is checked on all relevant form fields, we did this by going in page designer and changing the setting.

User Interface Design Storyboards

This is a prototype model that we created to get an idea of how our student information system will look like. We made some major changes along the way and our system turned out looking different.

We used this page to hold the information for the students who make any bookings.



This calendar is used by student users. When they log in, they able to check upcoming events that are coming and see which ones they are booked too on which day. It is like a timetable.

Requirement catalogue

Requirement catalogue	
Requirement Id:	#NF201
Requirement name	Access permissions
Description	Access permissions for the particular system information can only be changed by A system Admin, this should include various parts of tables, and the admin having the capability to add/modify events, user accounts and venues.
Owner	A
Comments/suggested solutions	Create two separate applications one for the student side & one for the Admin allowing particular functionalities.
Benefits	Will allow to add/edit via the front end instead of making changes via the back end, adding an Admin account with a particular granted set of permissions is vital and can be very beneficial to the application.
Related requirements	

Requirement catalogue	
Requirement Id:	#NF202

Requirement name	Application accessibility
Description	Application should be accessible on 3g/4g mobile connections.
Priority	D
Owner	A/S
Comments/suggested solutions	Oracle apex allows mobile UI and 3g/4g connectivity.
Benefits	Wider levels of access for users to be able to interactive with the application.
Related requirements	

Requirement catalogue

Requirement Id:	#NF203
Requirement name	Response time (Page)
Description	The response time should be no longer than five seconds on the time it takes to a view a page.
Priority	M
Owner	A/S
Comments/suggested solutions	
Benefits	Too allow a user to be able to view a page without having to wait a long time for the page to load.
Related requirements	#NF203.1 #NF203.2

Requirement catalogue

Requirement Id:	#NF203.1
Requirement name	Response time (Data)
Description	Response time should take no longer than five seconds when displaying any particular data from a database onto a page
Priority	M
Owner	A/S
Comments/suggested solutions	To use a powerful web development tool such as oracle that handle fast response time.
Benefits	Too allow the user to retrieve relevant information and data quickly
Related requirements	#NF203 #NF203.2

Requirement catalogue	
Requirement Id:	#NF203.2
Requirement name	Update Time (Data)
Description	Modified data within the database should be updated for all of the users accessing it within five seconds.
Priority	M
Owner	A/S
Comments/suggested solutions	To use a powerful web development tool such as oracle that handle fast update time.
Benefits	Users able to immediately be able to see updated data at least 5 seconds after it has been updated. This is beneficial as it is important users can see new data and not old data.
Related requirements	

Requirement catalogue	
Requirement Id:	#NF205
Requirement name	Ensuring user maintainability
Description	Ability of the system to be able to operate correctly/accordingly, if large number of users login or are trying to login.
Priority	D
Owner	A/S
Comments/suggested solutions	
Benefits	Too allow many users to correctly be able to operate without any issues.
Related requirements	

Requirement catalogue	
Requirement Id:	#NF206
Requirement name	Wrong password/username

Description	System returns error when user has entered wrong password/username
Priority	
Owner	A/S
Comments/suggested solutions	SQL commands shall be used to in order to check users login details. The authentication scheme will be changed and i function could be used.
Benefits	User is able to identify their mistake, from the response of the return error.
Related requirements	

Requirement catalogue	
Requirement Id:	#NF207
Requirement name	Wrong data type

Description	System returns error when user has entered wrong data type in a field.
Priority	M
Owner	A/S
Comments/suggested solutions	Set column data types to particular ones, ensuring a user has to enter that particular one to proceed.
Benefits	Too inform user that they have entered a wrong data type value so that they can easily understand their mistake.
Related Requirements	

Requirement catalogue	
Requirement Id:	#NF208/9

Requirement name	Require validation
Description	System requires validation of fields (input) by student user/admin user.
Owner	A/S
Comments/suggested solutions	Use page designer to ensure validation is checked on relevant columns
Benefits	Before submitting page, the system will check for any field that require validation, the benefit of this that a user will not be able to pursue to the next page.
Related requirements	

Requirement catalogue	
Requirement Id:	#NF210

Requirement name	An easy accessibility friendly navigation
Description	Users easily able to navigate by a navigation menu & buttons provided within the application
Priority	M
Owner	A/S
Comments/suggested solutions	Use the shared components option and access ‘user attributes’ & the ‘navigation menu’ to enhance the navigation. We should also adjust the sequence to order the pages appropriately.
Benefits	Too allow users to easily & effectively navigate through the application, without having issues finding pages.
Related requirements	

Requirement catalogue	
Requirement Id:	#NF211

Requirement name	Include service page
Description	Users able to access a service page, displaying various services within the University and outside.
Priority	M
Owner	A/S
Comments/suggested solutions	Include various sources of services within Luton that are relevant to students. Display information into a interactive grid/form.
Benefits	To give users up to date, relevant information about services within the University and outside.
Related requirements	

Requirement catalogue	
Requirement Id:	#NF212

Requirement name	Layout and colour scheme
Description	Layout and colour scheme must be user friendly and fully visible for the user in various lighting conditions.
Priority	M
Owner	A/S
Comments/suggested solutions	Use clear colours on the design on the pages including the colour of the font & the size.
Benefits	Too allow users to easily see the display without mis-reading and being able to clearly visualise everything.
Related requirements	

Requirement catalogue

Requirement Id:	#F301
Requirement name	Logging into the application
Description	Registered users must be able to log in to the application via the login page and be redirected.
Priority	M
Owner	A/S
Comments/suggested solutions	A forgot password function is provided if the student is unable to log in
Benefits	Users can log in and be redirected to the main side of the application if which they can have particular access to.
Related requirements	

Requirement catalogue	
Requirement Id:	#F302
Requirement name	Booking event
Description	A student user must be able book an available event
Priority	D
Owner	S
Comments/suggested solutions	Create tables in-regards to event information and replicate it on the application itself.
Benefits	Students can book themselves into available events.
Related requirements	

Requirement catalogue	
Requirement Id:	#F303

Requirement name	Viewing Booking
Description	A student user must be able to see their own event booking
Priority	M
Owner	S
Comments/suggested solutions	SQL query should be used to select only ‘app users’ id from login and show only their bookings.
Benefits	Students are able to see there booked events for reference.
Related requirements	

Requirement Id:	#F304
Requirement name	Viewing Venues
Description	Users must be able to view the event venues
Priority	M
Owner	S
Comments/suggested solutions	Each event shall have an event ID and Venue ID attached. Use a form to display venue the table from parts of the database.
Related requirements	

Requirement Id:	#F305
Requirement name	Deleting own bookings
Description	Student users should be able to delete their own event bookings
Priority	M
Owner	S
Comments/suggested solutions	Use a region button to direct the report to a form in which Students users can choose to delete their booked event. An SQL delete command will be used.
Benefits	If student user decides he wants to cancel they can have the functionality to delete their booked event.

Requirement catalogue	
Requirement Id:	#F306
Requirement name	View calendar
Description	A student must be able to view their calendar with their own booked events displayed
Priority	D
Owner	S
Comments/suggested solutions	SQL code shall be used to display only app users, so that only that particular user signed in is able to see only their booked events on their calendar.
Benefits	They are able to see upcoming events they are booked too in a calendar format.
Related requirements	

Requirement catalogue	
Requirement Id:	#F307
Requirement name	Log in & Log out
Description	Users must be able log out of kiosk & be redirected back to login page.
Priority	M
Owner	A/S
Comments/suggested solutions	
Benefits	Keeps the kiosk looking consistent and relevant to user needs. For example if sessions times out, you will be directed to home page
Related requirements	

Requirement catalogue	
Requirement Id:	#F308
Requirement name	Deleting user event bookings
Description	Admin users must be able to delete zero to many users event bookings
Owner	A
Comments/suggested solutions	Unlike student users, admins shall have a report with a form on their side application to allow them to modify the report and data inside it.
Benefits	To remove redundant user accounts for example past students who have graduated
Related requirements	

Requirement catalogue	
Requirement Id:	#F309
Requirement name	Modify all users event bookings
Description	Admin users must be able to have the option of functionality to modify all users event bookings
Priority	M
Owner	A
Comments/suggested solutions	
Benefits	Give up to date information for student to participate and be satisfied with the ease of use on the service
Related requirements	

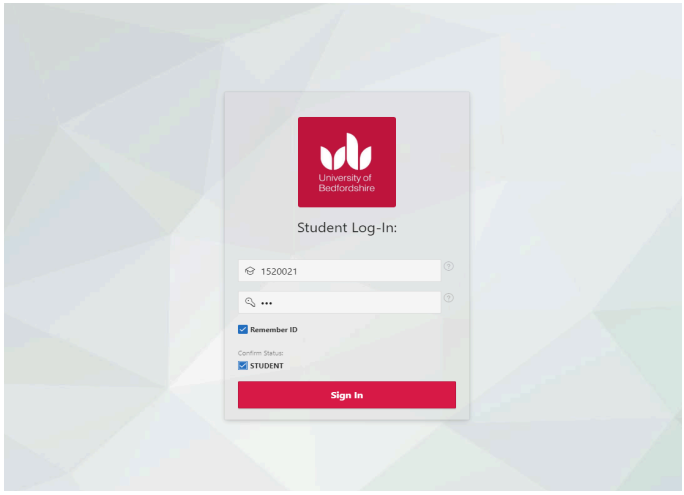
Requirement catalogue	
Requirement Id:	#F310
Requirement name	Add an event/venue
Description	Admin users must be able to add an event/venue.
Priority	M
Owner	A
Comments/suggested solutions	A button can be used to redirect the Admin to form page in order to add an Event. The Admin should fill in the event details and click the submit button in order for a 'SQL insert command' to proceed, allow the data to be inserted into the relevant tables.
Benefits	Give up to date information so students are able to participate in events.
Related requirements	

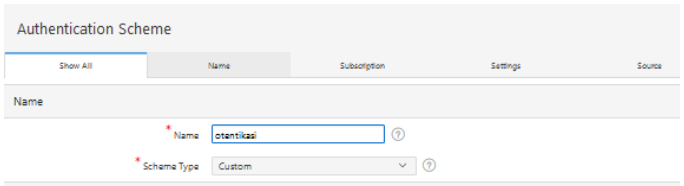
Requirement catalogue	
Requirement Id:	#F311
Requirement name	Create user accounts & modify
Description	Admin users must be able to create user accounts & modify.
Priority	M
Owner	A
Comments/suggested solutions	Admins shall have access to an additional page & table which should include users details.
Benefits	To help manage student accounts for example if students forget their password or if any faults or human errors occur due to incorrect information.
Related requirements	

Student Information System test

Student login:

The student login is can only be used by the students to access the student information system. The information that they get to see or modify is set appropriately. The student details are stored in a table in which therefore once the user click signs in the system will run a particular function to check details entered.



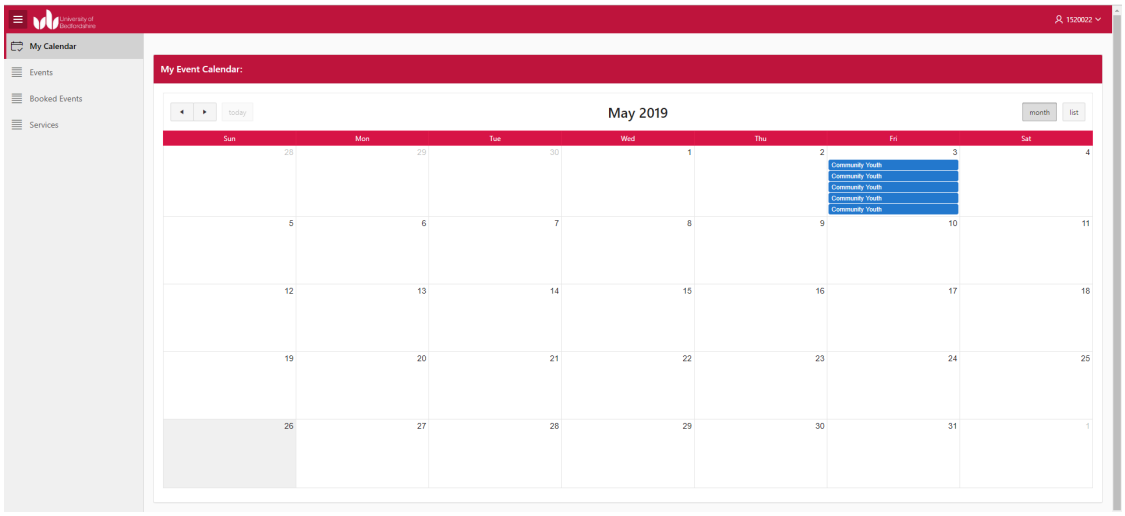



```
FUNCTION cek_otentikasi (p_username IN VARCHAR2, p_password IN VARCHAR2)
RETURN BOOLEAN
AS
hasil NUMBER := 0;
BEGIN
SELECT 1 INTO hasil FROM TLOGIN
WHERE UPPER(USERNAME) = UPPER(p_username)
AND PASSWORD = p_password
AND STATUS= :P101_STATUS
AND KET= 'AKTIF';
RETURN TRUE;
EXCEPTION
WHEN NO_DATA_FOUND THEN
RETURN FALSE;
END cek_otentikasi;

```

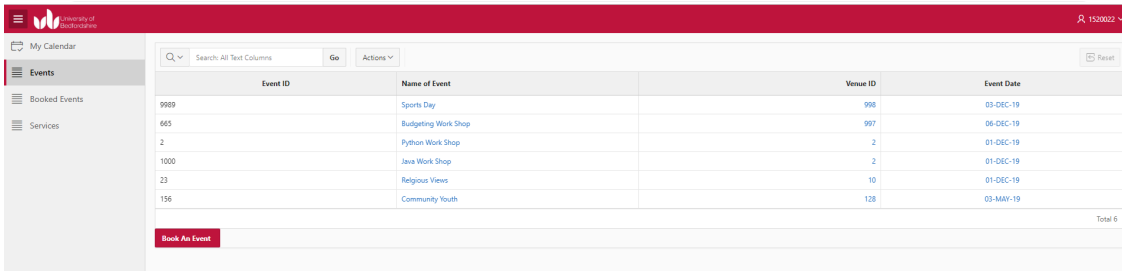
My calendar:

This is the calendar. The calendar is used to show which bookings have been made by the students, it also shows the date on which the booking has been made.



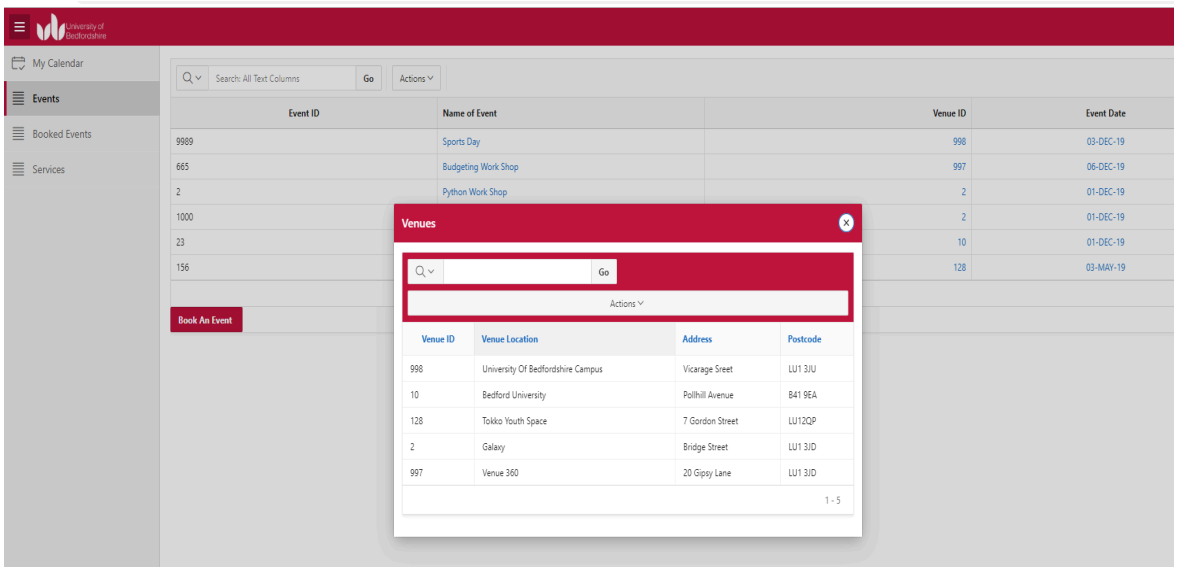
Events:

The events page shows the events that are available for the student user.



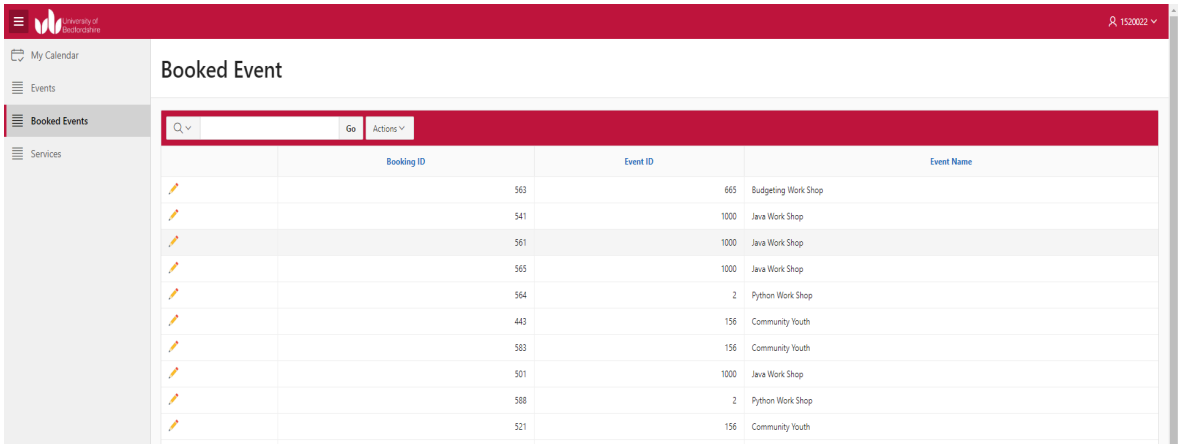
Venues

The venus page is opened by the student user when they click on any event on the events page. This works by the use of the interactive grid having the capability to be able to source a particular column to another page, such as this modal dialog, when it is opened, the user can see the location of the event he booked, the address and the postcode this is populated from the tables data in our DB and presented into a report type.



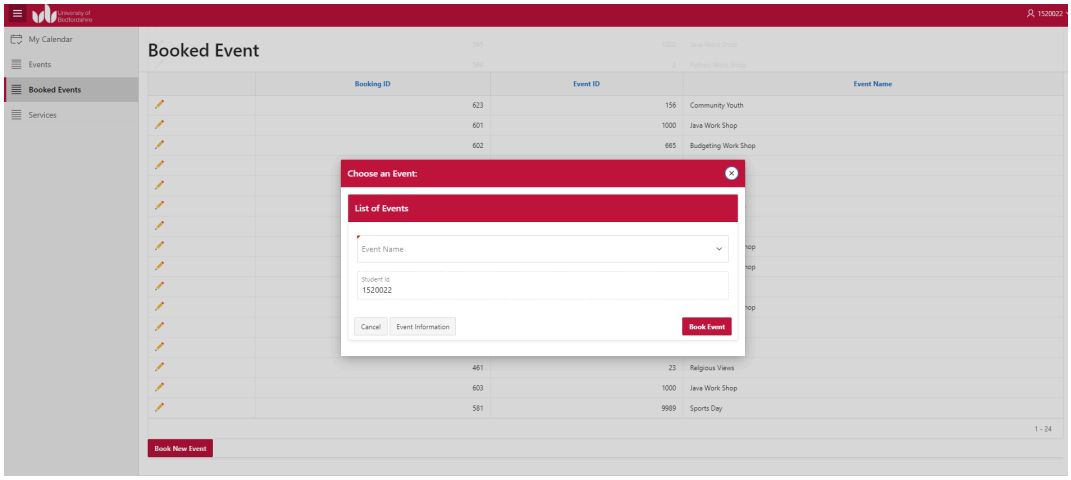
Booked Events

This page shows the events that the student user has booked, this done based on the student user that has logged in. So that means if the user that has logged in has no bookings, there will be no bookings displayed on this page



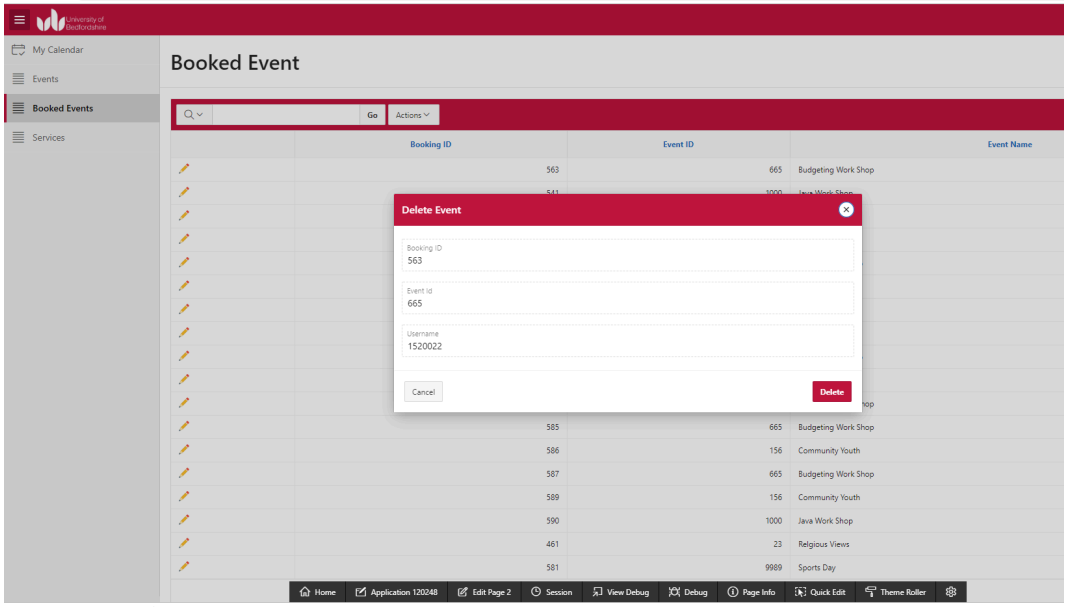
Book Event:

The book event page is opened by the student users when they click the key about ‘Book New Event’ to book an activity which is appear the list of events. Then a window will be come because it is about they can click the ‘event name’ to picking which event do they want to book and make sure they are student number. If they want to book the event which they choosing, they can click the red key that is ‘book event’. If they no sure, they can click ‘cancel’ or ‘event information’ to look over events again.



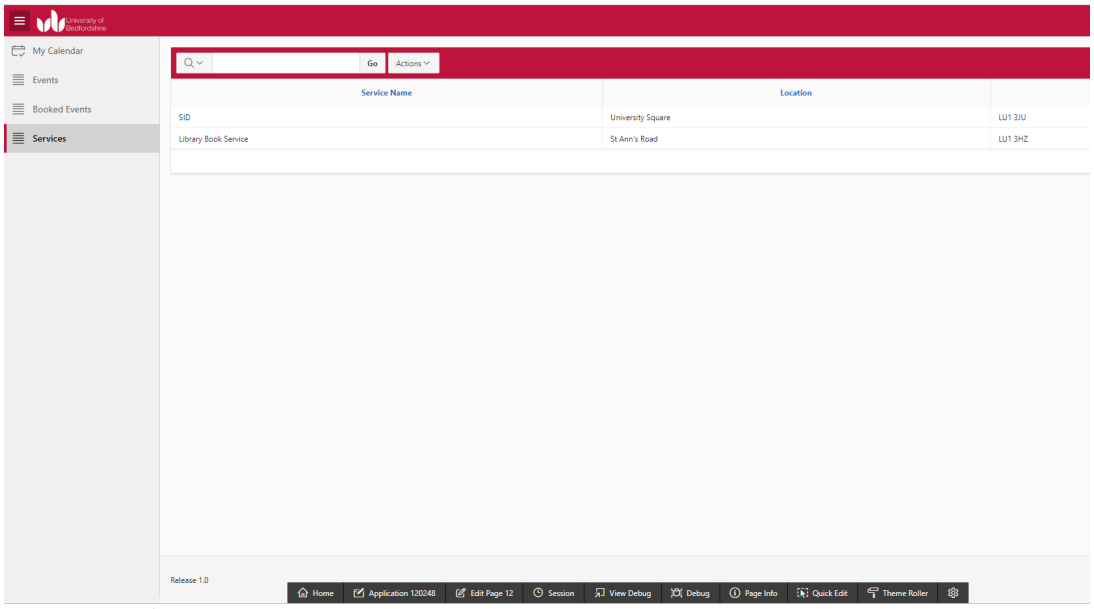
Delete Event:

When the student user logs in, he is given the option to delete any events that he has created. The user can do this by pressing the yellow pencil on the booked events page.



Services

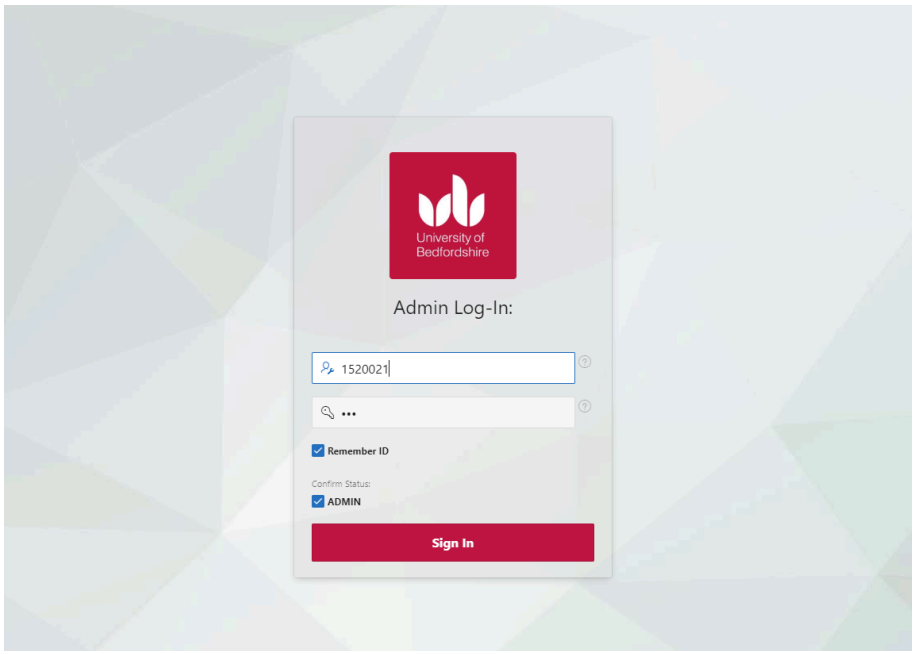
The page can be used to show the student user any service that the university can provide the students but the students are not given the permission to interact with the services.



Admin Test:

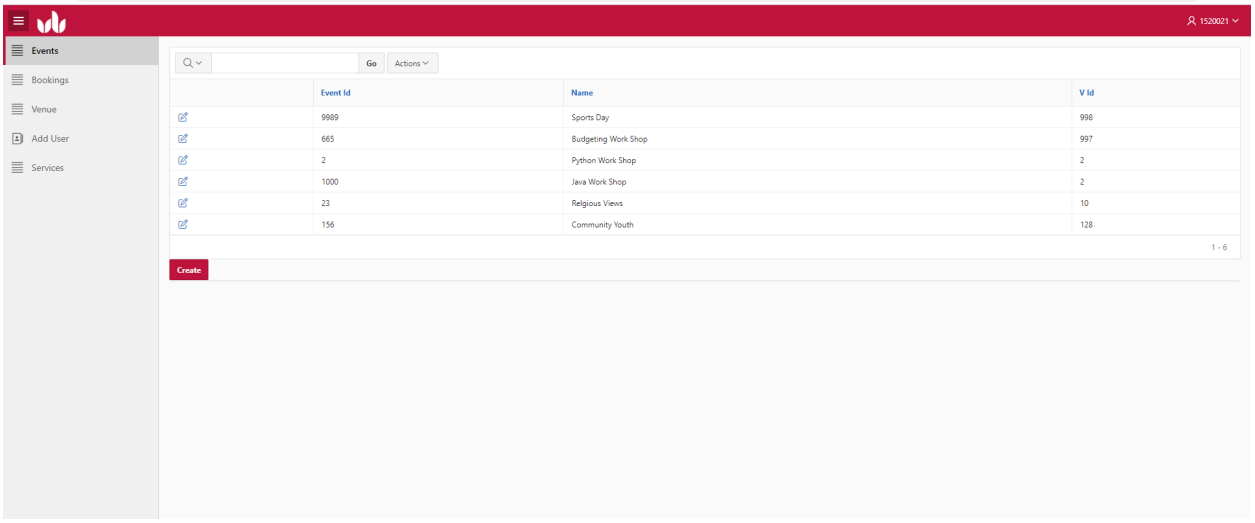
Admin Login

Just like the student login, the admin login works as the same but when the administrator logs in he is given more permission over the student information system when compared to the student.



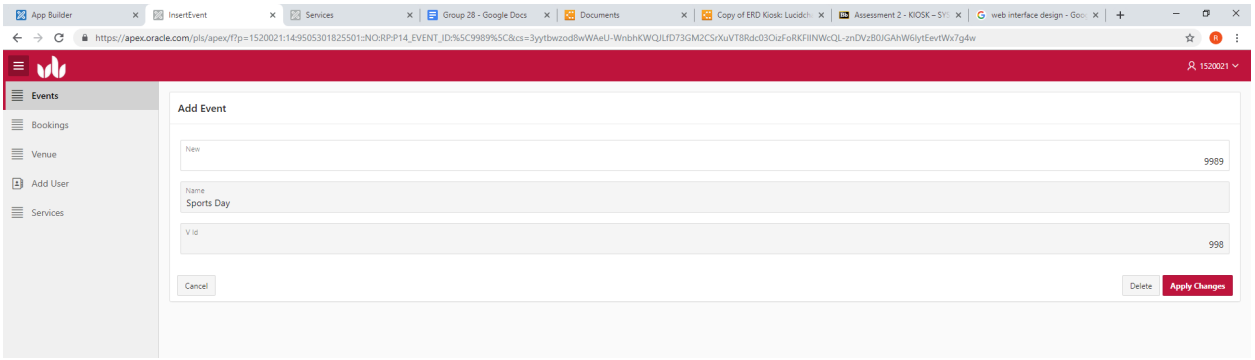
Events

The administrator can use this page to view any events that have been created either by him or any other administrator.



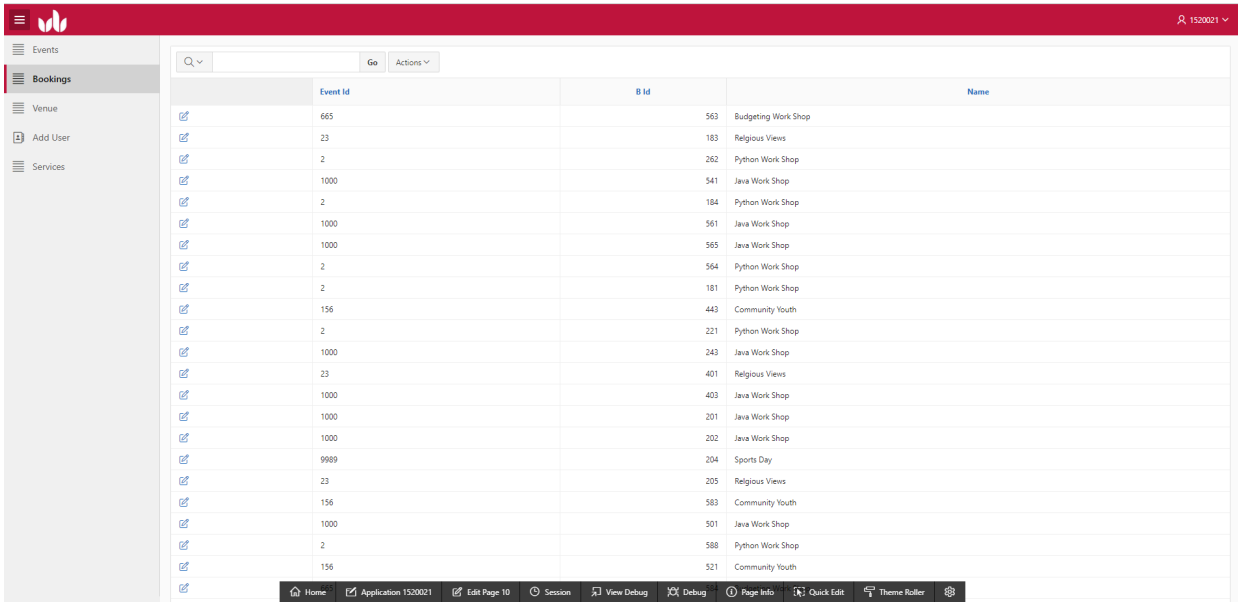
Add Event

Since the admin has more permission over the student information system than the student uer, he can create an event. He can do this by clicking on the create button and the add event page will come up. The admin can add an event by creating a new event ID, adding an event name and the venue ID.after an event is added, it is viewed by both the user and the admin.



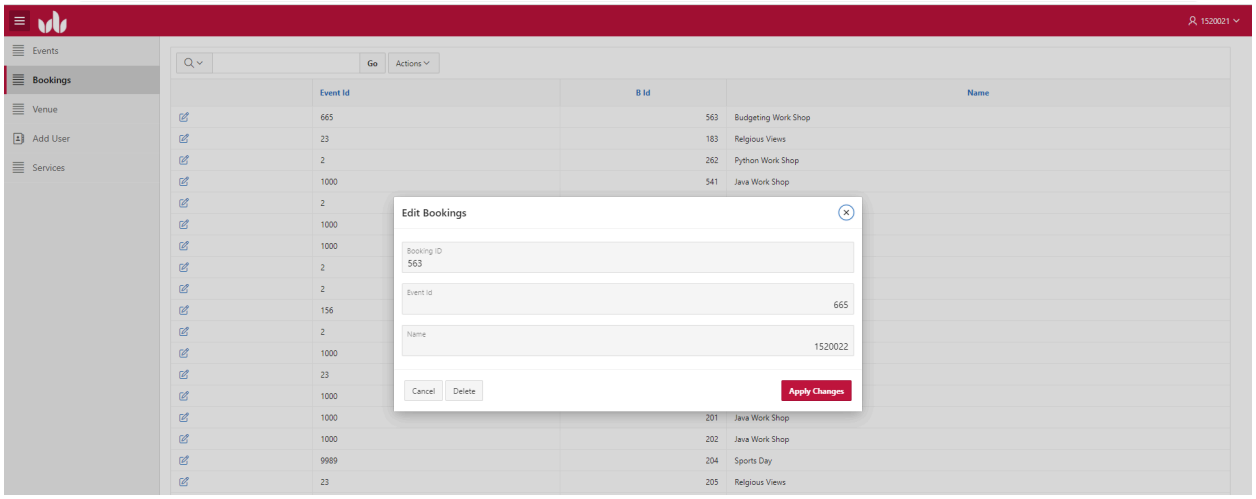
Bookings

The page which is booking is opened by the administrator. It is showed information about event id, booking id and event name. And the total number of booking is appeared on bottom right corner.



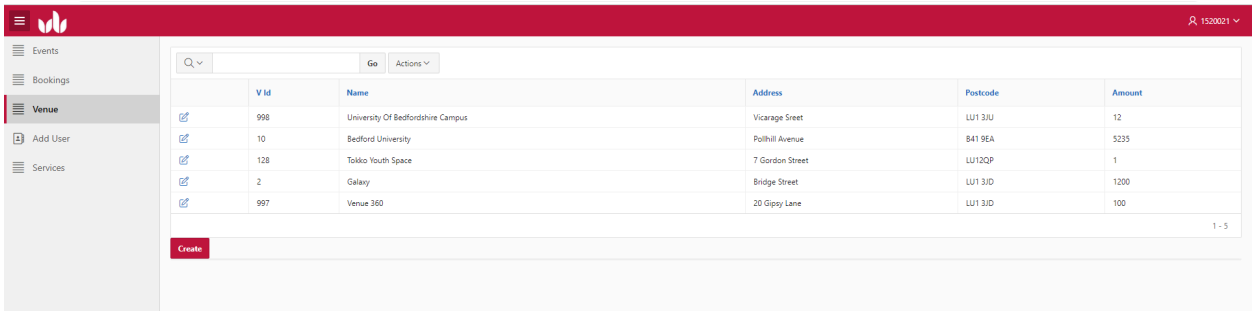
Edit Bookings

The edit booking page is opened by the administrator when they click the blue icon in booking page. There are a lot of details in here, which are about the student users booking such as booking id, event id and username. And the administrator has the right to change the all details, if any student users want to be fixed. And the administrator can delete the booking too.



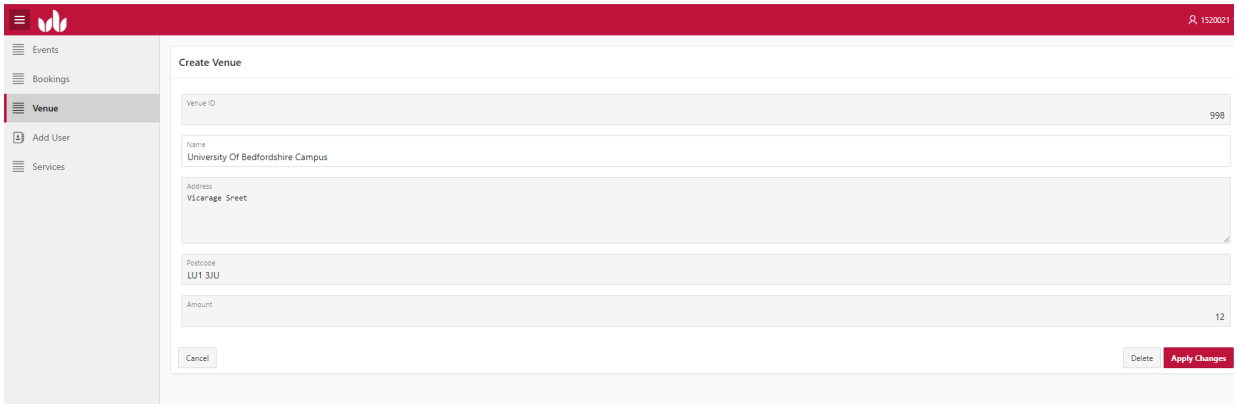
Venue

The page of venue is showed the information of location. For example, there are venue id, name, address postcode and amount in this page. The administrator can use the search bar to search keywords for the accurate information and make sure it right. If no, they can click the blue button to fix it.



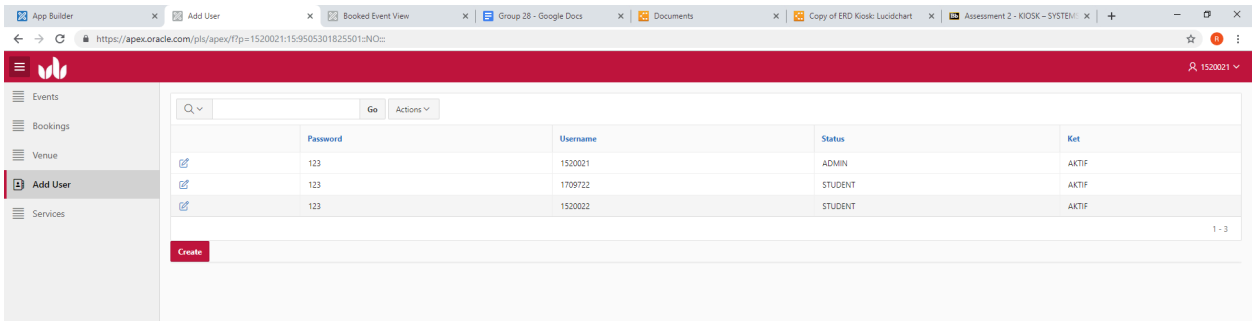
Create Venue

In order for the admin to create a venue, they have to add a venue ID, the name of the venue, the address and the postcode. This information will then be displayed on the venues page for both the user and the admin.



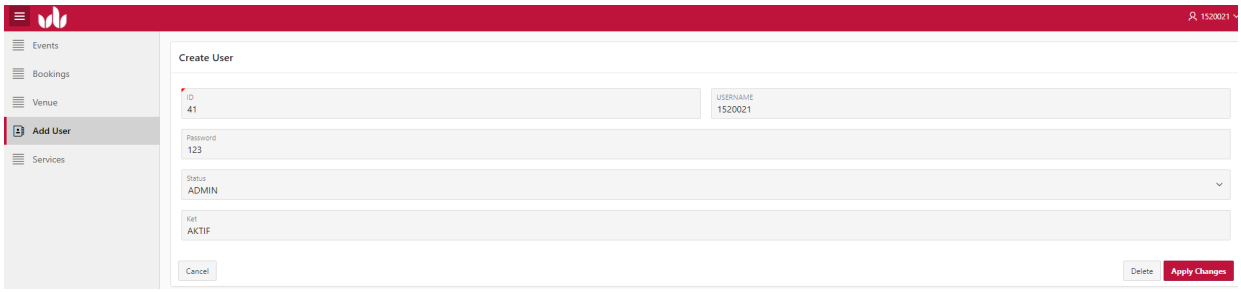
Add User

The admin is also given the permission to see the users accounts as well as the information that they have such as their passwords, usernames and their status.



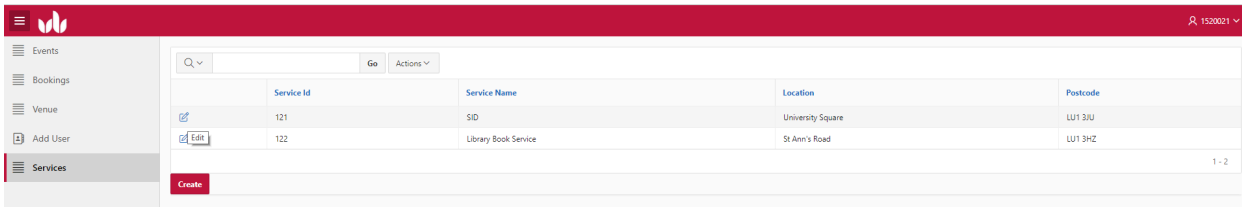
Create User

The administrator is also given permission to create new accounts both student user and admin.he can also deactivate and activate the same accounts and activate them too.



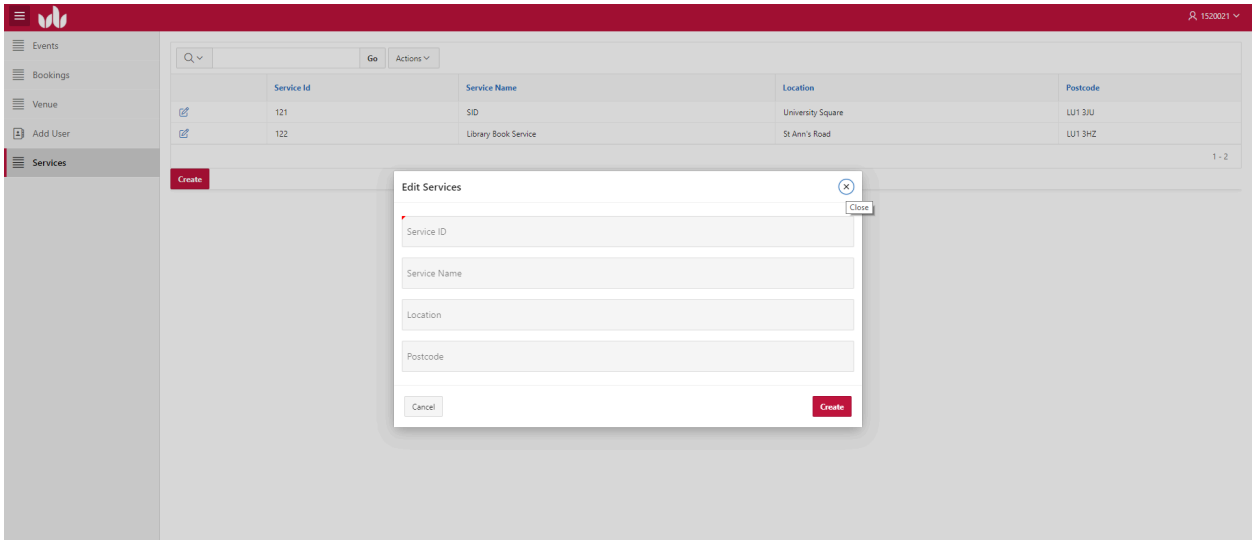
Services

The admin uses this page to see the services that he has added.



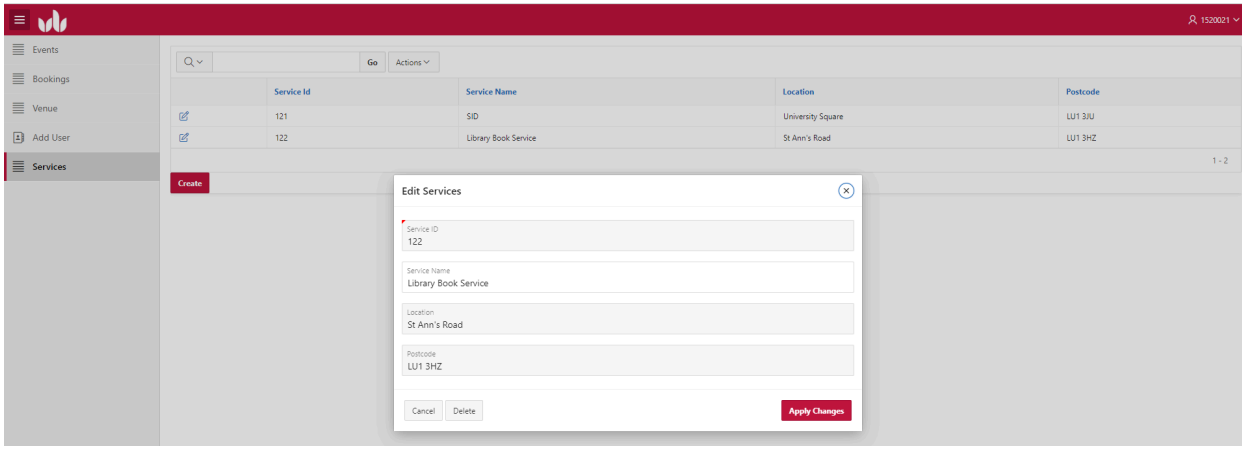
Add services

The admin is given the permission to add services, in order to do this the admin must enter the service ID, the service Name, location and the postcode.



Delete Services

when the administrator clicks on the delete services page he will be able to edit the events by deleting them this is done by clicking the blue edit button on the right. They can also update the services like name, location and postcode in this.



Customisation:

The student information system be laid out in a way that when there is a student user with an account, he will have to first login before he sees his information. This is because all the information that the user has will be based on his login details. The login page will have a logo, confirmation check box to make sure that it is the student logging in and a sign in button. This page will have a red colour scheme.

After logging in, the student user will be able to access the pages. The first page will be the calendar and it will display the user's bookings, the bookings displayed will be displayed according to the student user's calendar.
Another means of customization was the option of the student users to click on the event when they see it, everytime they see it they will see the venue of the event.
In the booked events page, we customize the pages by giving the student users an option to view and delete their events to add more to our customisation, we added a logo to both the student user and administrator's account.

Page by page information:

Page 1
Log in
The first page in our student information system is the login. The login has two types. These include student and admin. Admin has all the features a student has however they are able to manage user accounts. The admin is also able to make changes to events such as the venue of the event

Page 2
Calendar
Whenever a user logs in the student information system, this is the first page that they will see. This page displays the events that the student user has booked as well as the current date, time and year.

Page 3
Events
The events page is where the information about the the name of the event, the name of the venue, the address and postcode are displayed it is where the students get to find out more information about the event that they would like to book.

Page 4
Venue
This page shows the information from the venue like the the Venue ID, Venue Location, Address and the postcodet. As the administrator, you have the option you give all the venue details for your students. When the venue is created, the information entered will be displayed on the student information system.

Page 5
Booked Events
After the user books an event, their events booked will be displayed on this page. The page shows the booking ID , event ID and the name of the event that was booked. The student user also has the option to delete any events on this page if they want to, this is incase they book the wrong event.

Additional Administrator Pages

Page 6

Event information

As an administrator, you have the permission to add an event to the student information system. Adding an event to the student information system an only be done by the administrator, this means that the student will not be allowed to use or access this page.

Page 7

Add User

The add user page holds sensitive data about the students. It holds data such as their passwords, username etc. Because of this, it can only be accessed by the administrator. The administrator can also create/add a new user/student to the student information system through this page.

Meetings & feedback:

To enable us to identify & gather user requirements from a real potential client perspective and a student perspective by carrying out formal meetings

Meeting	Client/User	Feedback
1	Ashraf (CLIENT)	Requirement to allow students to view their bookings
2	Student (USER)	Requirement for a list of services to be added and a better layout/colour scheme.

Normalisation:

UNF	1NF	2NF	3NF
Student table Student_Name Student_Surname Student_Email Student_Address Student login Student_Username Student_Password Admin table Admin_Name Admin_email Event table Booking_status Event_Title Event_Orgasiner Event_StartDate Event_EndDate Event_StartTime Event_EndTime Num_Tickets Days_hired Venue table: Venue_Name Venue_Address Venue_Postcode Amount_for_hire Service table Service_Name Service_Location Service_Postcode	Student Table **Student_ID** Student_Name Student_Surname Student_Email Student_Address Student_Username Student_Password Admin table **Admin_ID** Admin_Name Admin_email Booking table **Booking_ID** Booking_status **Event_ID** Event_Title Event_Orgasiner Event_StartDate Event_EndDate Event_StartTime Event_EndTime Num_Tickets Days_hired **Venue_ID** Venue_Name Venue_Address Venue_Postcode Amount_for_hire Service table **Service_ID** Service_Name Service_Location Service_Postcode	Student table **Student_ID** Student_Name Student_Surnam e Student_Email Student_Address Admin table **Admin_ID** Admin_Name Admin_Email Booking table **Booking_ID** Booking_status Event table **Event_ID** Event_Title Event_Orgasiner Event_StartDate Event_EndDate Event_StartTime Event_EndTime Num_Tickets Days_hired Junction table **Event_ID** **Venue_ID** Venue table **Venue_ID** Venue_Name Venue_Address Venue_Postcode Amount_for_hire Service table **Service_ID**	Student Table **Student_ID** Student_FirstName Student_Surname Student_Email Student_Address Admin Table **Admin_ID** Admin_FirstName Admin_Surname Admin_email Booking Table **Booking_ID** Student_ID***** Booking_status Events Table **Event_ID** Event_Title Event_Orgasiner Event_StartDate Event_EndDate Event_StartTime Event_EndTime Num_Tickets Days_hired Venue_ID* Venue Table **Venue_ID** Venue_Name Venue_Address Venue_Postcode Amount_for_hire Service Table **Service_ID** Service_Name Service_Location

		Service_Name Service_Location Service_Postcode Student_login table **Student_ID** Student_Username Student_Password	Service_Postcode Student_login Table **Student_ID** Student_Username Student_Password

Future plans

During our planning a lot of changes were made upon our diagrams and many things we did include did not make it to the final project due to us running out of time. Whilst this system is put in place we would look to add the following into the kiosk system. A section for clubs which is similar to how to can create and join events. Include communities such as the student union Have an estate team who manages events. Different faculties. Bigger event format which can handle events such as careers or charity events which is where the estate team will come into play. With bigger events we may have provided information on catering companies.

Allocated tasks

	Nadim Uddin	Junjie Shao	Deon Shah	Ronald Bukenya
UML				
ERD				
CATWOE				
REQ CAT				
Rich picture				
Normalisation				
SQL				
App builder				
Page by page				

information				
Week by week steps				

Conclusion

To conclude we planned how we were going to look to create our student information kiosk. This was done by identifying what functional and non requirements requirements the system may have. Following up diagrams such as UML, ERD with normalisation and activity diagrams were made. Further research and designs were made through storyboards and SSM. Using all the data we gathered we were able to create a kiosk system which fits the students needs when it comes to be accomodating the students with all the university has to offer and give the students all the tools needed to achieve

References

Drob. 2006. Drupal groups. [Online]. [February 20th 2019]. Available at: **https://groups.drupal.org/node/374** [Accessed 11 May 2019].

Docs.oracle.com. (2016). *Creating Lists of Values*. [online] Available at: https://docs.oracle.com/cd/E11882_01/appdev.112/e11947/bldapp_lov.htm [Accessed 11 May 2019].

Docs.oracle.com. (2017). *Establishing User Identity Through Authentication*. [online] Available at: https://docs.oracle.com/database/121/HTMDB/sec_authentication.htm#HTMDB12003 [Accessed 10 May 2019].

Docs.oracle.com. (2017). *Joins*. [online] Available at: https://docs.oracle.com/cd/B19306_01/server.102/b14200/queries006.htm [Accessed 23 May 2019].

Anon, (2019). [online] Available at: https://docs.oracle.com/cd/E11882_01/appdev.112/e11947/bldapp_lov.htm auth scheme [Accessed 19 May 2017].

Docs.oracle.com. (2016). *Configuring Application Users and Application Roles*. [online] Available at: https://docs.oracle.com/database/121/DBFSG/users_roles.htm#DBFSG20021 [Accessed 21 May 2019].