

# Software Requirements Specification for a TeamSpace Booking System

Authors<sup>1</sup>:

Willy Finish; Slim Chance; Percy Veer; Peter Out; Miles Behind

---

<sup>1</sup> This document is based on a student submission that, with permission, has been anonymised and edited to introduce many errors. Accordingly, we hereby re-attribute it to these (imaginary) authors.

## Document History

Revision	Date	Description
0.1	9/05/2010	Initial document created by a student group Headings defined by [student] and assigned to team members by [student]
0.2	15/05/2010	[student]: edit of functional requirements. [student]: initial creation of non-functional requirements [student]: creation of overview section [student]: creation of introduction and definitions
0.3	16/05/2010	Team reorganization of existing content Sections extended Addition of Use-Case tables
0.4	17/05/2010	[student]: edit of Functional Requirements [student]: Addition of Use cases [student]: Edit of Section1
0.5	18/05/2010	Addition of Availability, Maintainability and Design Constraints
1.0	19/05/2010	First draft printed and revised Content finalization
1.1	23/03/2011	Anonymised and edited by course staff, introducing many more errors for students to identify in review. Document reattributed to an imaginary group of engineers.

## TABLE OF CONTENTS

---

1	Introduction .....	1
1.1	Purpose .....	1
1.2	Scope .....	1
1.3	Definitions, Acronyms, Abbreviations .....	1
1.4	Overview .....	3
1.5	Product Perspective .....	5
1.5.1	System Interfaces .....	5
1.5.2	User interfaces .....	5
1.5.3	Hardware interfaces .....	6
1.5.4	Software interfaces .....	6
1.5.5	Communications interfaces .....	6
1.5.6	Memory .....	6
1.5.7	Operations .....	7
1.5.8	Site adaptation requirements .....	7
1.6	Product Functions .....	7
1.7	User Characteristics .....	8
1.8	Constraints .....	8
1.9	Assumptions and Dependencies .....	9
2	Specific Requirements .....	9
2.1	External Interfaces .....	9
2.1.1	ITS network .....	9
2.1.2	UQ Security Network .....	12
2.2	Hardware Interfaces .....	14
2.2.1	Dumb Terminal .....	14
2.2.2	RF-ID and Magnetic Swipe Card Readers .....	15
	Functional Requirements .....	15
2.2.3	Student Level Use Cases .....	18
2.2.4	Course Administrator Use Cases .....	24
2.2.5	room Administrator Use Cases .....	31
	Non-Functional Requirements .....	41
2.2.6	Software Operating Environment .....	41
2.2.7	Software Network Impact .....	41
2.2.8	Software Database Sizing .....	41
2.2.9	Hardware .....	41
2.2.10	Reliability/Dependability .....	42
2.2.11	Security .....	42
2.2.12	Availability .....	43

2.2.13 Maintainability .....43

## 1 INTRODUCTION

---

### 1.1 PURPOSE

---

This Software Requirements Specification provides a complete description of all the functions and constraints of a meeting room booking system for the proposed TeamSpace laboratory at University of Queensland, henceforth referred to by the acronym UQTSBS. The expected audience of this document includes EAIT faculty staff members, school of ITEE staff members, the software developer and persons desiring a model for software development documentation.

### 1.2 SCOPE

---

This document concerns the implementation details of the proposed UQTSBS, and makes some assumptions as to the physical design of the UQ TeamSpace laboratory. As far as possible, such assumptions are limited to where absolutely necessary, and are explicitly stated. It is not intended as a complete design plan for any physical aspect of the laboratory itself. Similarly, the exact design and implementation of the UQTSBS user interfaces and protocols is beyond the scope of this document. Such details will be settled during the implementation phase of the project. Furthermore, the security of physical assets, such as the computers, tables and chairs, is beyond the scope of this document.

### 1.3 DEFINITIONS, ACRONYMS, ABBREVIATIONS

---

Term	Description
UQTSBS	The University of Queensland TeamSpace Booking System
EAIT	Engineering, Architecture and Information Technology
ITEE	Information Technology and Electrical Engineering
SQL	Structured Query Language. A database computer language designed for retrieving and modifying data in relational database systems
Database	An organized body of related information
Terminal	A device which is capable of communicating over a data line to a remote server.
Non-interactive Terminal	A dumb terminal is a "computer unit" that has a monitor and a keyboard that must connect to another computer for external processing of the user commands. This device contains little actual 'intelligence' instead relying on a server for data and content
Swipe Card	A plastic card with a magnetic strip from which information may be obtained, such as a credit card or the like

## TeamSpace Booking System - Software Requirements Specification

Polarising OLED	An organic light emitting diode screen in which correctly applying a small voltage changes its permeability to light
RF-ID	Radio-frequency identification: Identification and tracking using radio waves
Self arresting, soft close hinge	A door hinge that offers resistance on closing to prevent the door slamming
Heat sink	A mass of metal attached to a chip carrier or socket for the purpose of dissipating heat usually into the local atmosphere
Tolerance	Allowance: a permissible difference; allowing some freedom to move within limits
Mono colour display	A display that uses only one color and tonal range to create images. For example a black and white screen
Room server	A computer that provides client stations with access to files and printers as shared resources to a computer network within the UQTSBS room
HTML	Hyper-Text Markup Language, a common internet format for rendering of text and media content
HTTP(S)	Hyper-Text Transfer Protocol (Secure), a common internet protocol used to send HTML to internet browsers
PHP	A scripting programming language widely used to write web applications
Apache	A HTTP server, which serves up web pages
mod_php	A module within Apache that embeds a PHP interpreter allowing server side conversion of PHP scripts to HTML
Windows	An operating system created by Microsoft
Linux	A freely available operating system
Macintosh or Mac	An operating system created by Apple
Oracle 10G	A relational database management system (RDBMS) produced and marketed by Oracle Corporation
XHTML	A computer language used to create world-wide-web pages which are read by browsers
Javascript	A scripting programming language most commonly used to add interactive features to web pages
API	Application Programming Interface, an interface that a software program implements in order to allow other software to interact with it
Booking	The act of reserving a resource for personal use

## TeamSpace Booking System - Software Requirements Specification

Allocating	The act of reserving a resource for booking by another entity.
HUPP	UQ's Handbook of University Policies and Procedures
Subnet	A logically visible, distinctly addressed part of a single Internet Protocol network
Virtual Machine	A software implementation of a machine (i.e. a computer) that executes programs like a physical machine
UPS	Uninterruptable Power Supply, An electrical apparatus that provides emergency power to a load when the input power source fails.

## 1.4 OVERVIEW

---

The proposed TeamSpace laboratory is a collaboration facility designed for the school of ITEE to support at most seven teams of up to six students each collaborating on software engineering and information technology projects. The team work spaces will have flexible screen configurations to support collaborative software development and will also support remote team participants using Internet-based videoconferencing

The new learning space will accommodate both scheduled (supervised) class sessions and sessions booked by private teams. The scheduled classes will be limited to no more than 50% of the working week (8 am to 5 pm) so that private teams have reasonable opportunities to book and use the facilities. In addition, 24-hour access will be provided via swipe cards for students enrolled in designated team-based courses (refer to Appendix ?).

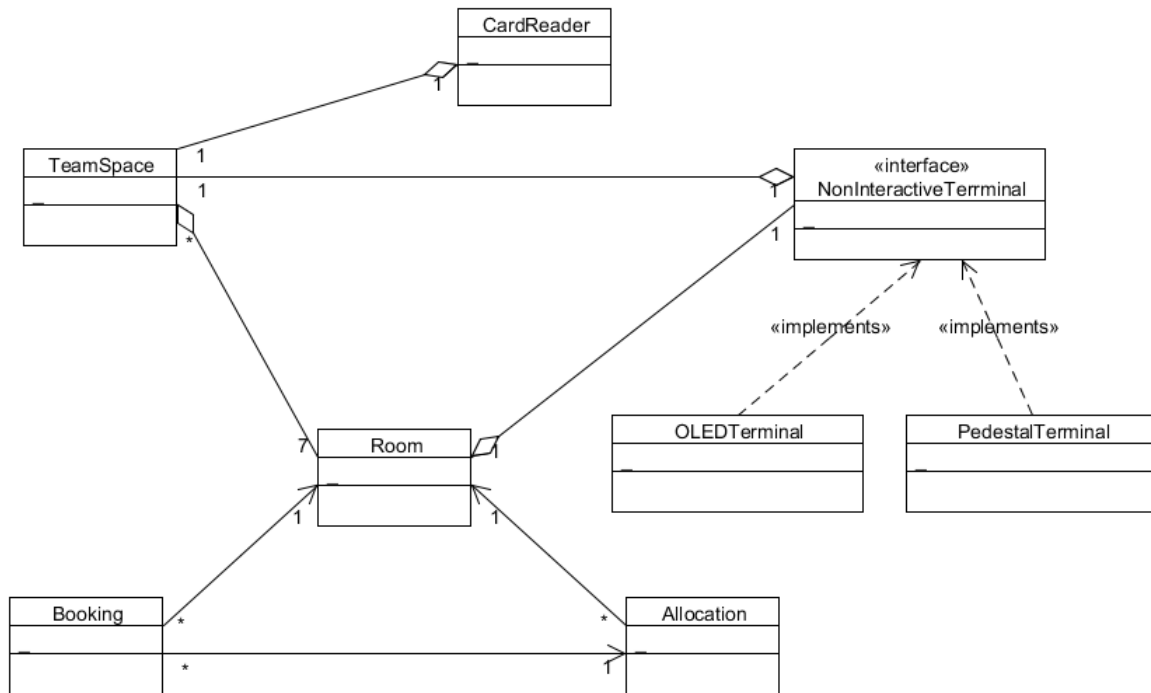
An application is required that allows ITEE teaching staff and ITEE students to book the TeamSpace workspaces. This web-based application is referred to as the UQTSBS. and will provide a facility for creating and monitoring bookings and usage of the lab. Admin staff will be able to manage booking schedules and lab facilities using the system.

Admin staff will also be able to book the entire TeamSpace (all workspaces) for regular timeslots for an entire semester – for example, for use within a regular course tutorial. This would automatically cancel any student bookings of workstations during those timeslots, and notify the affected students.

UQ currently supports authentication of users with established technology and protocols. The UQTSBS will be integrated into the existing UQ student and security networks.

TeamSpace Booking System - Software Requirements Specification

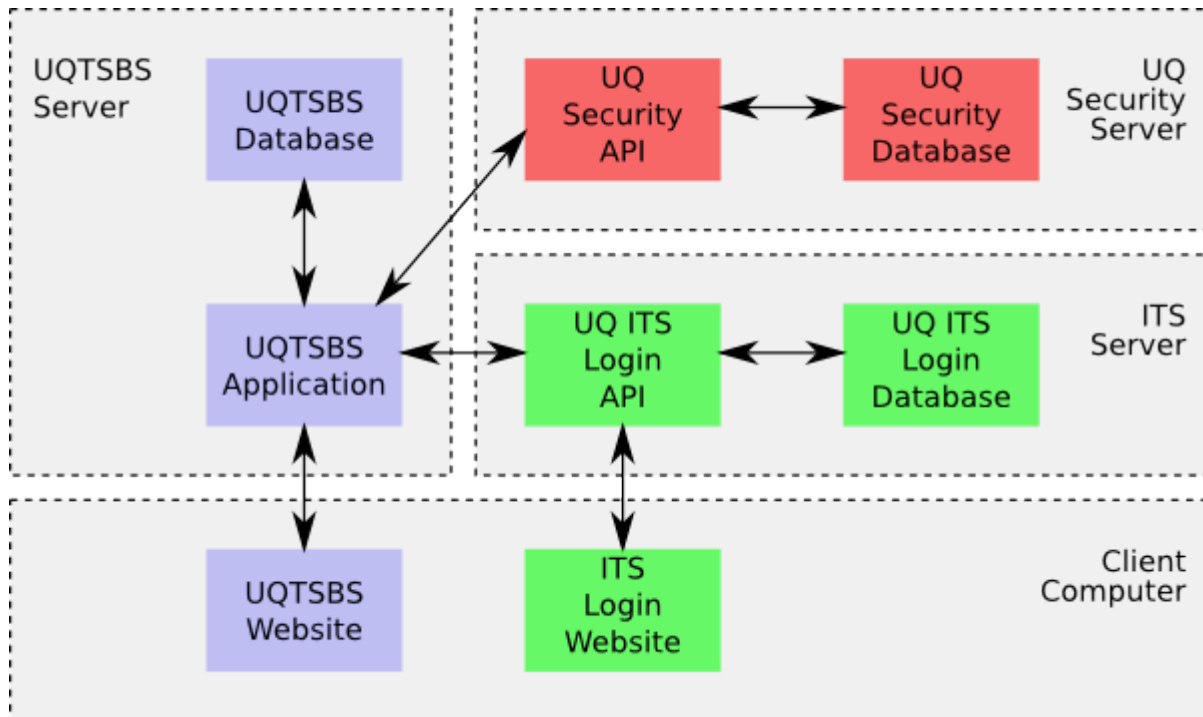
The class diagram for the system is shown below.



## Overall Description

## 1.5 PRODUCT PERSPECTIVE

The UQTSBS is integrated with both the ITS network and UQ's security network as shown below. It will be considered part of the UQ student services.



UQTSBS Product Perspective - arrows indicate TCP connections.

### 1.5.1 SYSTEM INTERFACES

The UQTSBS uses several external services including the ITS Login API, the UQ Security API and the ITEE course database. The UQTSBS application will communicate with the ITS Login API to find out if a client has the necessary privileges to use the UQTSBS service. The security service provides the UQTSBS protection against misuse and logs user activity and the ITEE course database provides the UQTSBS additional information about the authenticated user.

### 1.5.2 USER INTERFACES

The UQTSBS user interface is composed of two components, software and hardware. The software component of the user interface consists of the online web-based booking system. The hardware components of the user interface consist of a non-interactive terminal in the centre of the main room, non-interactive terminals on the doors of each group room and at least 1 standard ITS workstation in the main room to be used only for booking. Each terminal

## TeamSpace Booking System - Software Requirements Specification

on the room doors is required to display information about the current booking status of that room as a timeline. The centre terminal will display a timeline of all current bookings.

---

### 1.5.3 HARDWARE INTERFACES

---

The UQTSBS application and database will be designed to run on a centralised networked server. This server provides command processing and interaction for the other hardware interfaces within the room.

There will be a OLED, non-interactive terminal in the middle of the TeamSpace room, displaying all existing bookings for the day in real time. In addition, every room will feature an OLED interactive terminal displaying the bookings for that specific room for the day in real-time.

Each entry to the TeamSpace room features a card reader which is capable of reading RF-ID cards and magnetic swipe cards. This allows the UQ student cards to be used with the RF-ID component of the card reader, and the ITEE swipe cards to be used with the magnetic component.

The other hardware interface consists of the various workstations throughout the TeamSpace room for use of the UQ students. This document does not specify the requirements and specifications for these workstations, or the centralised server.

---

### 1.5.4 SOFTWARE INTERFACES

---

The online component consists of an Internet based calendar, where interaction occurs through a web browser. The UQTSBS will be written in PHP and will require Apache 2.2 with mod\_php installed on Windows, Linux or Macintosh. Oracle 10G will be used as a database server to hold the booking details for UQTSBS. The UQTSBS will render compliant XHTML 1.1 and Javascript 1.8.1. The web interface should support at least Mozilla Firefox 2.0 and greater, Internet Explorer 6.0 and greater, Google Chrome 4.0 and greater and Opera 9.0 and greater.

---

### 1.5.5 COMMUNICATIONS INTERFACES

---

HTTP/HTTPS will be used to communicate between the UQTSBS website and the user's web browser. The interface used to communicate between the backend UQTSBS application and database will be ANSI compliant SQL. Custom TCP/TLS based protocols will be used to communicate between UQTSBS and either ITS or the UQ Security Network.

---

### 1.5.6 MEMORY

---

The UQTSBS application is designed to run using under 1GB of RAM in normal use, however, this value scales linearly with the number of concurrent requests. The Oracle 10G database requires hard disk space proportional to the number of bookings stored in the system. It is expected to be below 5GB in normal use. Logging of user activity is expected to consume no more than  $5N^2 - 5N + 50$  megabytes.

The UQTSBS System will support both archiving of data and warehousing of data from the main database. Archiving of data may be performed by the room administrator in order to remove some of the data from the main database to free up space. Archiving consists of taking the data from the main database and exporting into an XML format that can be easily added back into the database later by the room administrator. Warehousing consists of

storing the data in such a way that it is more efficient to search through it along a specified axis.

---

### 1.5.7 OPERATIONS

---

There are two types of operation supported by UQTSBS, user-initiated operations and system-initiated operations. User-initiated operations will consist of all the use-cases (see Product Functions 3.1). System-initiated operations will be time based and will consist only of removing redundant or stale records and a periodic backup operation.

---

### 1.5.8 SITE ADAPTATION REQUIREMENTS

---

The software should be easily installable on a new site running Windows, Linux or Macintosh as an operating system and with Oracle 10G installed. Previous data may be moved by copying a single backup file to the same location on the new site and restoring the backed up database.

---

## 1.6 PRODUCT FUNCTIONS

---

The Booking system for the TeamSpace Laboratory is structured around three distinct user groups each representing an escalating level of control of the TeamSpace booking system. These levels of access are, listed in order of privilege: Room Administrator, Course Administrator and Student. The base privilege level for any user is that of 'Student' the highest is that of Room Administrator. The remainder of this section describes how each user category interacts with the system, their privileges, restrictions and their relationship to one another.

Common to all the user levels is the ability to make private group bookings in unallocated time slots. These bookings are subject to the 50% supervised time mandate and to user limits set by the room administrator. The user does not need to define with whom they will be using the room however, regardless of user access level. The system logs all user interactions with the system in order to ensure transparency and accountability.

As previously stated, the Room Administrator has all privileges typically associated with the root user of a system, specifically the user may add, edit or delete information and reservations as necessary. The Room Administrator is also responsible for managing restrictions to the other user groups and allocating room time to course and student users. In the case of student misuse of the rooms, the room administrator may lock access for the offending students for a set duration or permanently. It is also the responsibility of the Room Administrator to allocate the total amount of time to each course. This amount can vary from course to course depending on the subjects' requirements subject to the mandated 50% scheduled time limit. The room administrator is able to promote any user to Course Administrator or above and confer any combination of user privileges.

The next user level, that of Course Administrator, may allocate periods of time, up to their set maximum, during which use of rooms is reserved for students participating in his/her course. In some respects the Course Administrator functions as a restricted Room Administrator, they have many of the same privileges, able to modify and delete other user's bookings and set time restrictions upon subordinate users, however these powers only extend to the time periods allocated for their course. Course Administrators are also

## TeamSpace Booking System - Software Requirements Specification

responsible for the management of group allocations for the students in their course with up to a maximum of six students in each group. Like the Room Administrator, the Course Administrator can confer their system privileges to student users in order for more than one course administrator to be assigned to a single course. This concept aims to allow course lecturers to make full use of their tutors.

A user with Student level privileges may reserve a room online or in person at the TeamSpace laboratory.

All records of student bookings and course allocations maintained by the system will be stored on a university server with database interactions provided by a SQL interface. Security precautions must be in place to prevent unauthorised access to the data; only the room administrator has unrestricted access and may modify records.

For private bookings, the reservation process is straightforward: the student gives his or her student number and, assuming the student is authorised, the interface provides an interactive calendar through which the student may reserve a time period. The allocation of a room itself is transparent to the user as they do not need to know which rooms are available; only if a time period is available. The amount of time for which each student may reserve a room is limited in order to ensure fair opportunity for all who wish to use the rooms. Rooms may be reserved only a limited time in advanced with this period specified by the Room or Course Administrator.

If a user of any privilege class later wishes to remove or edit a booking, he/she need only select the option from the menu of the user interface. They will then be presented with details of the current and future bookings and a sub-interface with which to make changes.

## 1.7 USER CHARACTERISTICS

---

There are three user classifications Room Administrator, Course Administrator and Student for the UQTSBS user base.

Students' education levels can be assumed to be at least secondary school completion. Course Administrators can be assumed to have completed tertiary education, while Room Administrators may be considered highly proficient with computers, relational database management systems, web pages and the Internet.

## 1.8 CONSTRAINTS

---

Prospective developers of the UQTSBS must be aware of the following constraints to any proposed implementation of the system. These include policies set out by the University and ITEE facility, hardware limitations provided by the ITS service, restrictions to the external database interfaces, audit and accountability mechanisms and security of people and assets.

The developer must be aware of and conform to the University's code of conduct and standards of service as defined in the Handbook of University Policy and Procedures (HUPP). Specifically sections: 1.60 Administrative Accountability, section 3.40.4 Private access to Computers by Students and section 6.10 Information Technology.

## TeamSpace Booking System - Software Requirements Specification

In addition to these policies, the developer must be aware of the ITEE Acceptable Use Policy, User Privacy Policy, Swipe-cards and Building Keys Policy (where applicable) and Student Misconduct Policy. Where conflicts occur, the ITEE policy takes precedence.

The developer has access to the server and desktop hardware provided by the ITS services. The developer may petition for new hardware if required, but approval will not necessarily be granted.

The external interfaces the system must interact with, the ITEE student login interface, the ITS security interface, and the ITEE course database, all expose a custom PHP API to external web applications. These interfaces cannot be modified and it is preferred that existing services provided by these systems are not duplicated.

Course administrators must be authenticated against the UQ course co-ordinator database, which is a service provided by mySINet.

The system must have the ability to be remotely audited with the options of creating hard-copy 'paper trail' of user activity and usage metrics. The usage metrics must be able to indicate historical performance data for a minimum of one year prior to the current date. This system must also provide data to ensure that users can be held accountable for damages to room assets or other people.

## 1.9 ASSUMPTIONS AND DEPENDENCIES

---

As already expressed, the successful implementation of this system relies heavily on existing UQ subsystems.

## 2 SPECIFIC REQUIREMENTS

---

This section discusses the parameters and acceptable tolerances required in the final system.

### 2.1 EXTERNAL INTERFACES

---

This section defines the technical existing external interfaces that the UQTSBS will interact with. These systems include such software interfaces as the ITS student login system and the UQ security Network; and hardware interfaces such as the non-interactive terminals and RF-ID readers.

#### 2.1.1 ITS NETWORK

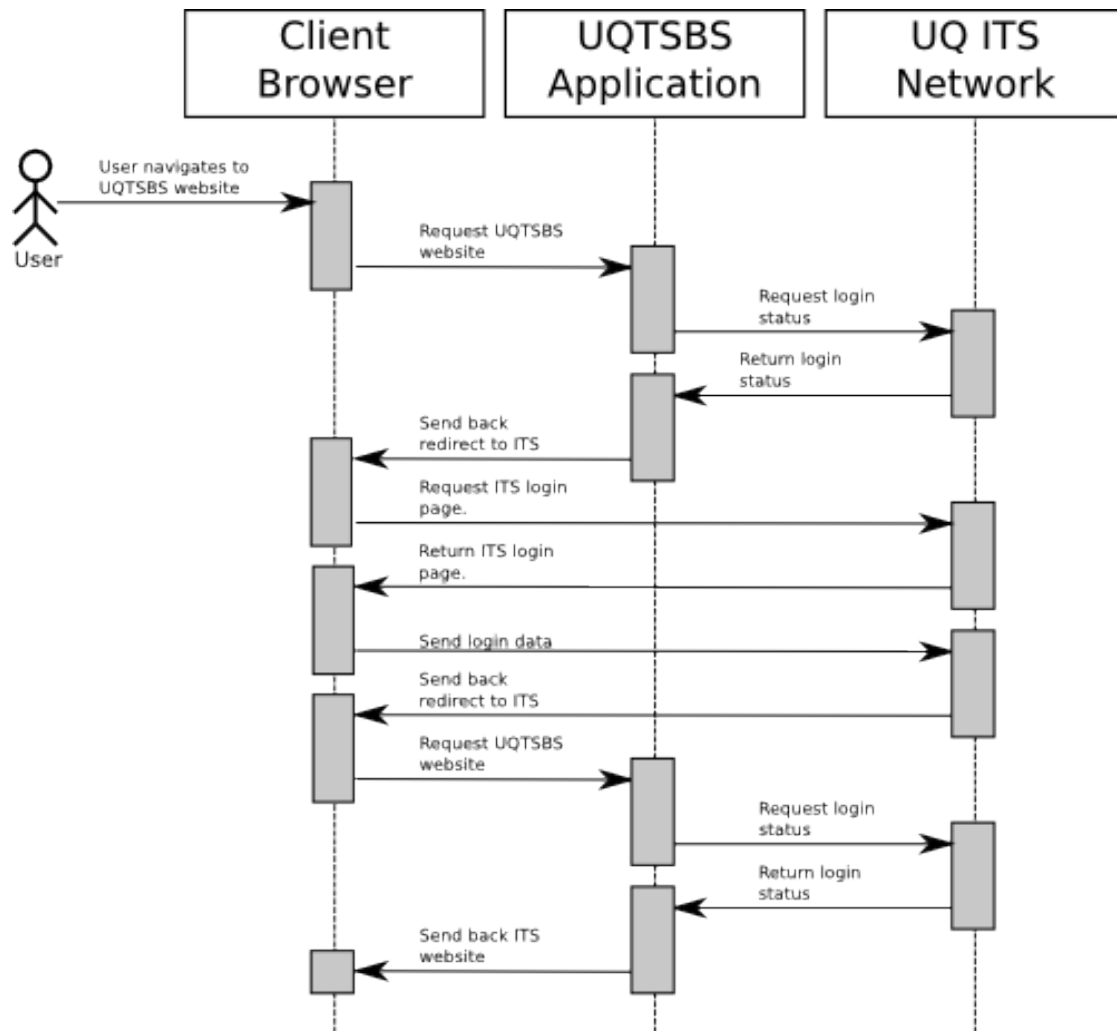
---

Name	ITS Network
Description	The ITS network controls all student logins and internet access at UQ using internet proxies.
Inputs / Outputs	The ITS network will receive queries about users status and will return whether the user is authenticated. The ITS webpage will be used for users logging

## TeamSpace Booking System - Software Requirements Specification

	into the website.
Valid Range, Accuracy or Tolerance	N/A
Units of Measure	N/A
Timing	See below.
Relationships to other inputs / outputs	The ITS network is only used by UQTSBS for authentication, so no other outputs are considered.
Screen Formats/Organisation	The ITS webpage will be in HTML.
Window Formats/Organisation	N/A
Data Formats	The ITS system will communicate with UQTSBS using HTTPS.
Command Formats	Commands / queries will be implemented in HTTPS
End Messages	There is no requirement for an end message as logins within the ITS network are persistent for an unknown amount of time and UQTSBS will constantly poll the ITS network whenever users attempt to use the main page.

Timing diagram for UQ ITS network as used in Login Use Case (see Login).



---

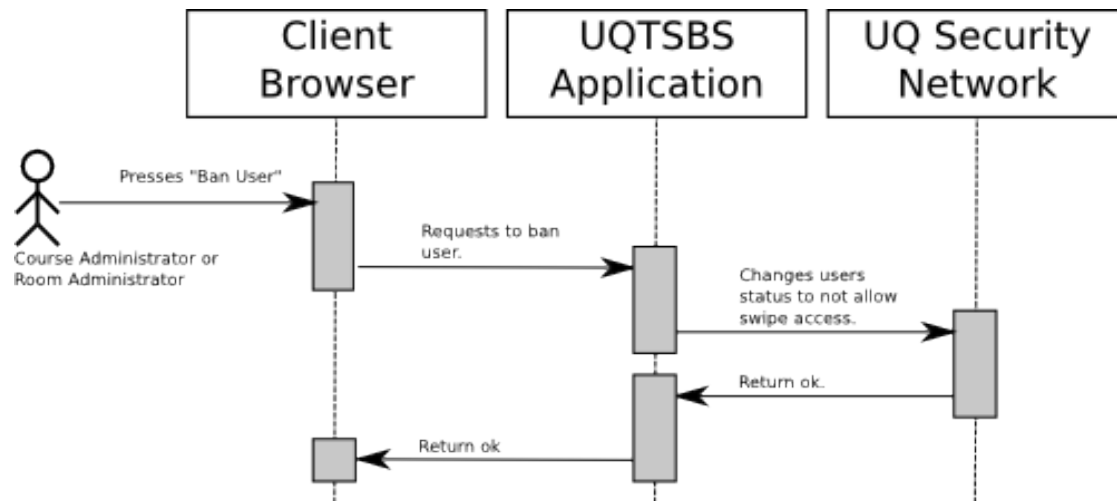
### 2.1.2 UQ SECURITY NETWORK

---

Name	UQ Security Network
Description	The UQ Security Network authenticates all UQ student and staff logins
Inputs / Outputs	The UQ security network will receive queries about user's id number and password.
Valid Range, Accuracy or Tolerance	N/A
Units of Measure	N/A
Timing	See Below
Relationships to other inputs / outputs	The UQ Security Network is only used by UQTSBS for authentication, so no other outputs are considered.
Screen Formats/Organisation	The UQ security network is transparent to the user, and is not displayed publicly.
Window Formats/Organisation	N/A
Data Formats	The UQ security system will communicate with UQTSBS using HTTP over the TCP protocol.
Command Formats	Commands will be sent through POST data using HTTP.
End Messages	This will be determined based on the UQ security system's response.

TeamSpace Booking System - Software Requirements Specification

Timing diagram for UQ security network that would be used in the Modify User Constraints use case (see Modify User Constraints).



## 2.2 HARDWARE INTERFACES

### 2.2.1 DUMB TERMINAL

Name	Dumb Terminal
Description	The dumb terminal in the TeamSpace room will display the bookings for all rooms. The dumb terminal outside each booking room will display the bookings for that specific room.
Inputs / Outputs	The terminal will receive its input from the centralised server and its output will be displayed on the OLED screen.
Valid Range, Accuracy or Tolerance	N/A
Units of Measure	N/A
Timing	Not available.
Relationships to other inputs / outputs	There are no interactions with any other systems, therefore there are no other inputs or outputs.
Screen Formats/Organisation	The screen in the TeamSpace is a 24" LCD. The screens outside each booking rooms are 17" OLEDs.
Window Formats/Organisation	N/A
Data Formats	The dumb terminals will be able to receive data via VGA or HDMI.
Command Formats	There are no commands specified.
End Messages	N/A

---

## 2.2.2 RF-ID AND MAGNETIC SWIPE CARD READERS

---

Name	RF-ID and Magnetic Swipe Card Readers
Description	The card readers authenticate the entry of UQ staff and students into the TeamSpace room.
Inputs / Outputs	The card reader receives input via RF-ID or a magnetic field and outputs the card values to the UQ authentication system.
Valid Range, Accuracy or Tolerance	The accuracy of RF-ID is 10 CM. The magnetic region of a card must touch the card reader.
Units of Measure	Centimetres
Timing	Not available.
Relationships to other inputs / outputs	There are no other inputs/outputs.
Screen Formats/Organisation	N/A
Window Formats/Organisation	N/A
Data Formats	The data is transmitted over a custom protocol.
Command Formats	Unknown/Proprietary
End Messages	N/A

---

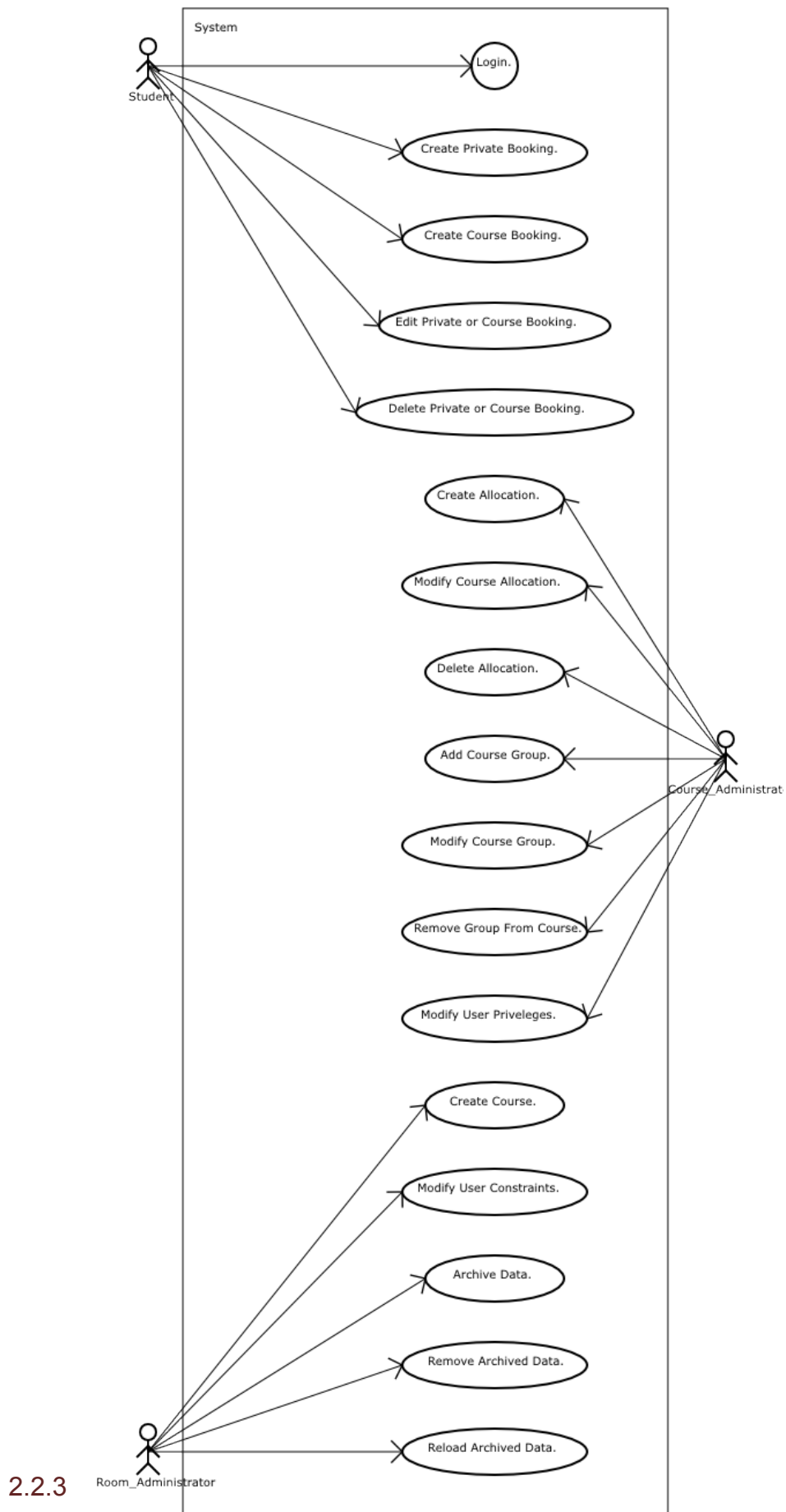
## FUNCTIONAL REQUIREMENTS

---

This section defines example top level use cases for each of the three user groups in the UQTSBS system. The first subsection defines the student use cases, the second the Course Administrators and finally the last the Room Administrators. Each of the processes illustrated in this section are cumulative; the Course Administrator is capable of performing both Course Administrator level processes and Student level processes and the Room Administrator is capable of performing all the tasks explained in this section.

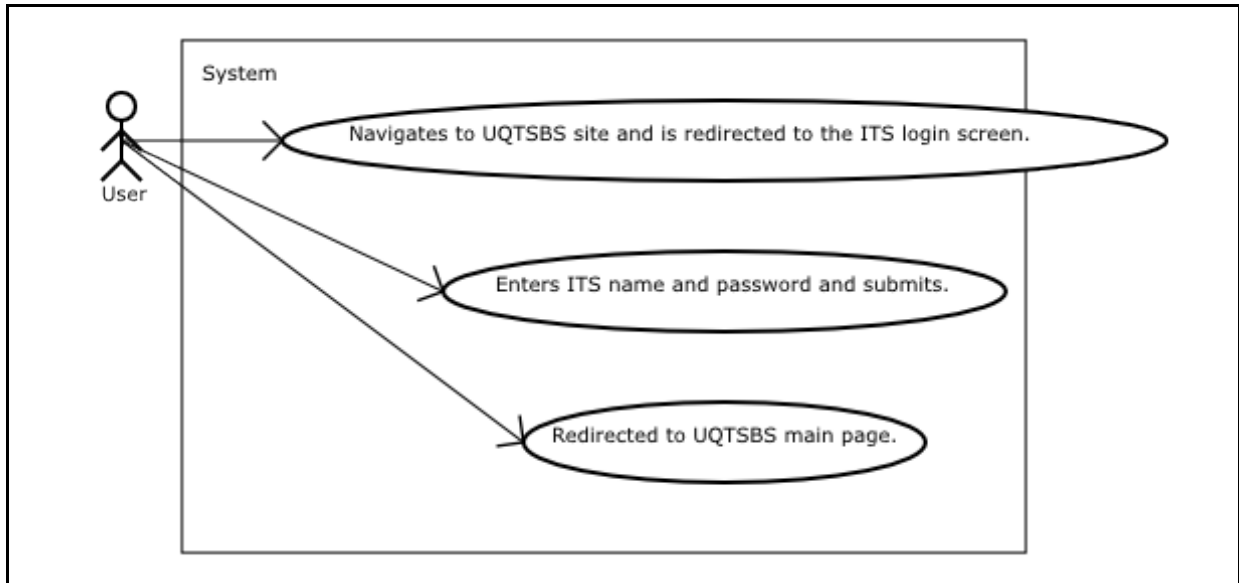


TeamSpace Booking System - Software Requirements Specification



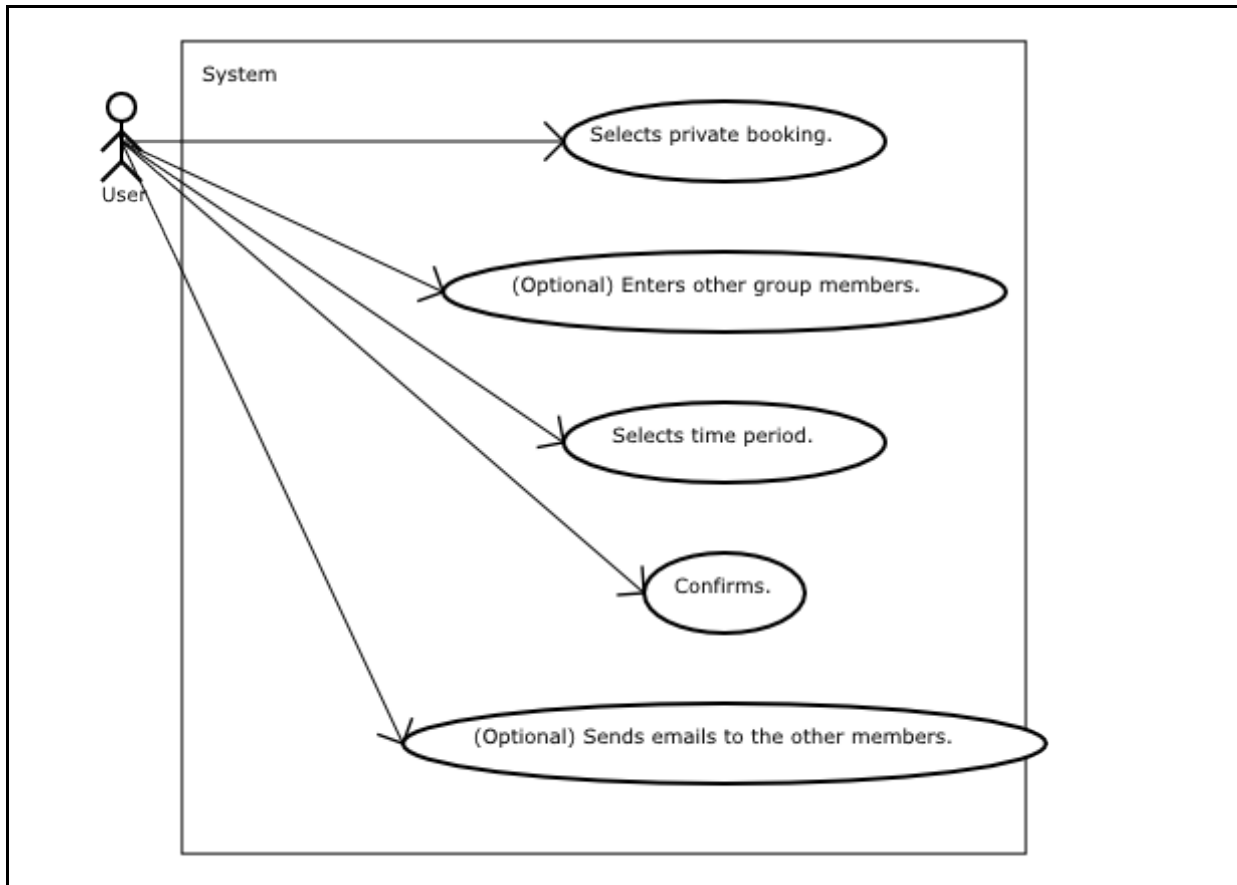
STUDENT LEVEL USE CASES

2.2.3.1 LOGIN TO THE UQTSBS



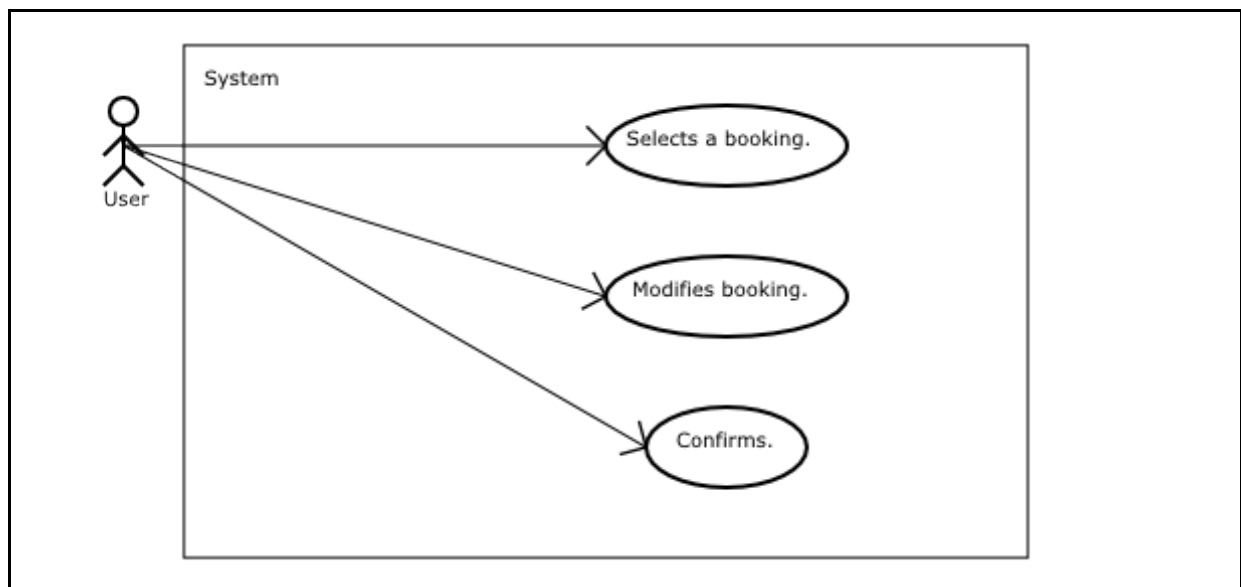
Brief Description	User logs into UQTSBS to allow managing bookings.
Actors	All users.
Preconditions	User is not logged in. User has a browser open.
Main flow of events	<ol style="list-style-type: none"> <li>1. User navigates to UQTSBS site and is redirected to the ITS login screen.</li> <li>2. User enters ITS name and password and submits.</li> <li>3. User is redirected to UQTSBS main page.</li> </ol>
Alternate flow of events	<ol style="list-style-type: none"> <li>2a. User enters incorrect name or password and submits.</li> <li>3a. User is redirected to an error page.</li> </ol>
Post-conditions	User receives privileges.

2.2.3.2 CREATE PRIVATE BOOKING



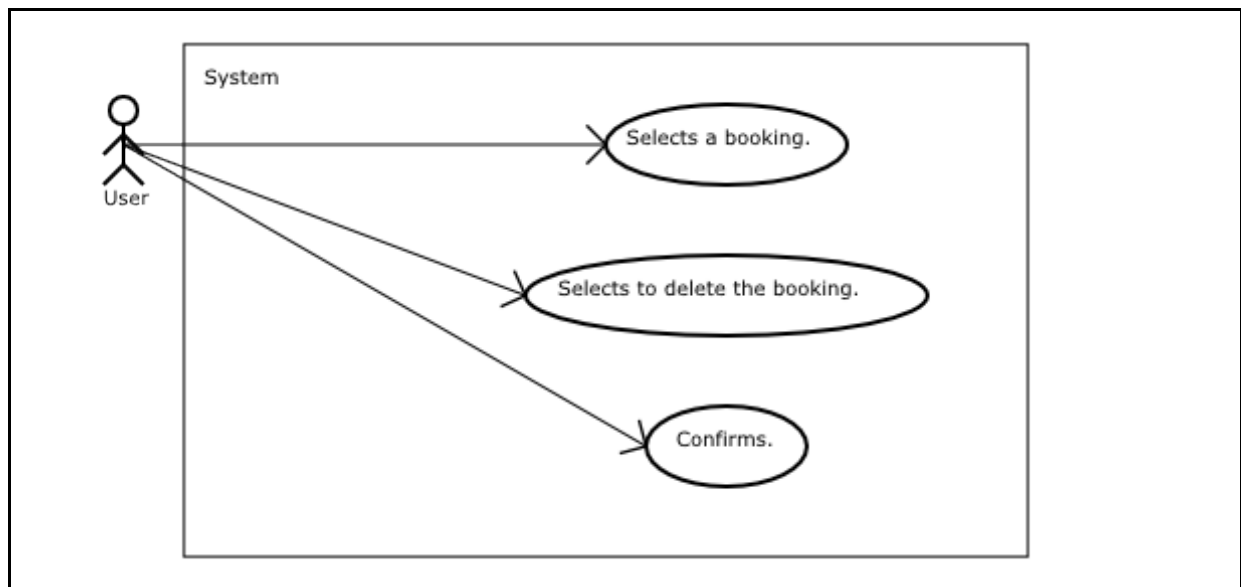
Brief Description	User creates a private booking.
Actors	All users.
Preconditions	User is logged in.
Main flow of events	<ol style="list-style-type: none"> <li>1. User selects the private booking option (default upon successful login).</li> <li>2. (Optional) User enters other group members.</li> <li>3. User selects a time period that has not been allocated.</li> <li>4. User confirms their booking for the selected time period.</li> <li>5. (Optional) The system sends emails to the user and the user's group members.</li> </ol>
Alternate flow of events	<p>4a. User selects a time period however the time period is not available.                      5a. User is shown an error message. System reverts to prior state.</p> <p>4b. User selects a time period that exceeds their allowed reserved time or the global private booking maximum.                      5b. User is shown an error message.</p>
Post-conditions	User is allocated the time period in the booking system.

## 2.2.3.3 EDIT PRIVATE OR COURSE BOOKING



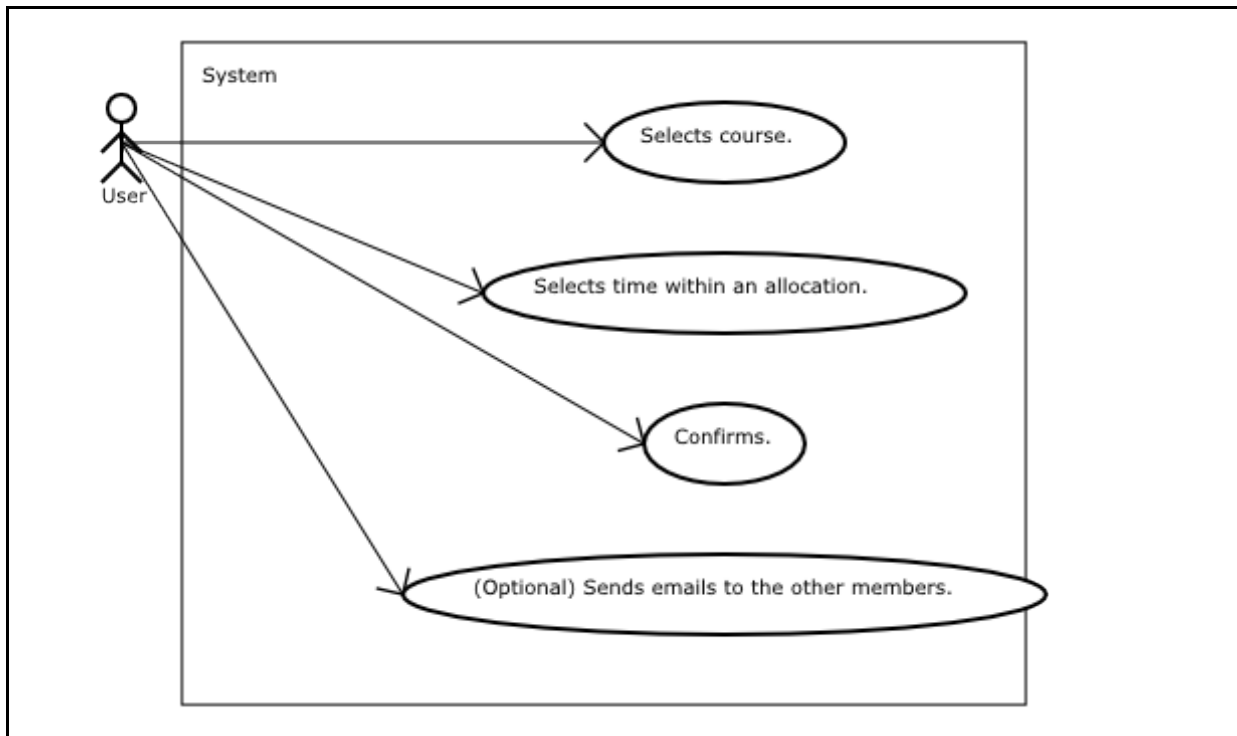
Brief Description	User edits a booking.
Actors	All Users.
Preconditions	User is logged in. User has previously made a booking.
Main flow of events	<ol style="list-style-type: none"> <li>1. User selects a booking from either a user booking history or from a TeamSpace booking timeline.</li> <li>2. User can modify:</li> <li>3. The time of the booking to another free time of the same or smaller duration</li> <li>4. The length of the booking</li> <li>5. The group members in a private booking</li> <li>6. Convert a private booking to a course booking</li> <li>7. User confirms changes.</li> </ol>
Alternate flow of events	3a. User receives an invalid change error message. System reverts to prior state.
Post-conditions	Booking properties are modified.

## 2.2.3.4 DELETE PRIVATE OR COURSE BOOKING



Brief Description	User deletes a booking.
Actors	All users.
Preconditions	User is logged in.
Main flow of events	<ol style="list-style-type: none"> <li>1. User selects a booking from either a user booking history or from a TeamSpace booking timeline.</li> <li>2. User selects to delete the booking.</li> <li>3. User confirms.</li> </ol>
Alternate flow of events	None.
Post-conditions	Booking is removed from the system.

2.2.3.5 CREATE COURSE BOOKING IN PRE-ALLOCATED TIME



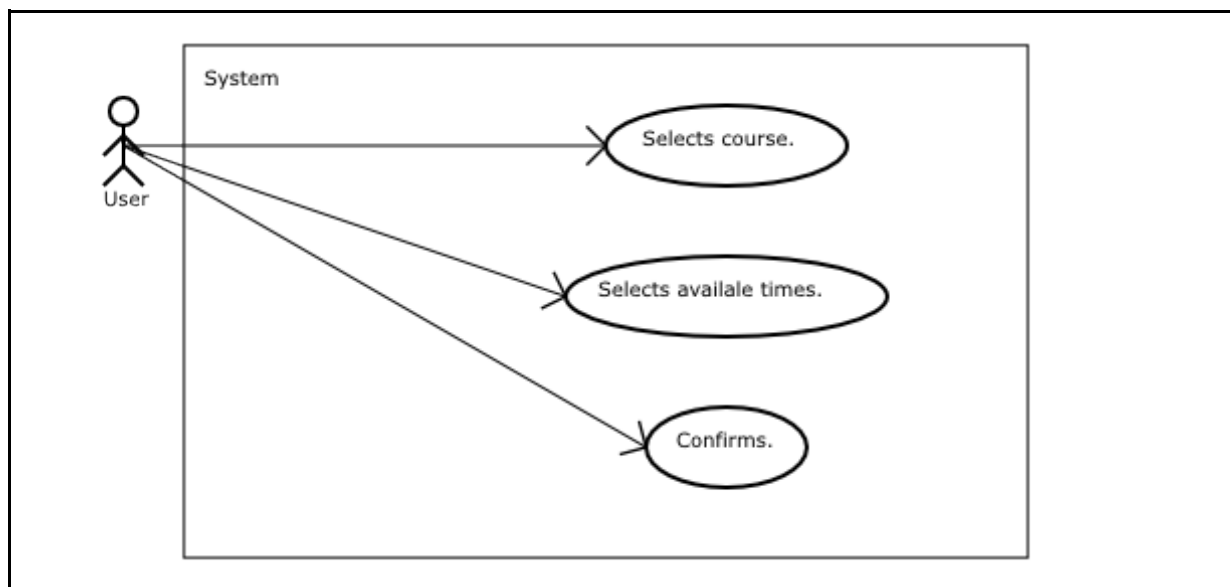
Brief Description	User creates a course booking within a time that has been pre-allocated to the course.
Actors	All users but designed for students.
Preconditions	Course coordinator has previously booked time for the course. Student is logged in.
Main flow of events	<ol style="list-style-type: none"> <li>1. User selects the course he/she is doing and is presented with possible times from allocations.</li> <li>2. User selects time within an allocation.</li> <li>3. User confirms their booking for the selected period.</li> <li>4. (Optional) If selected the system sends emails to the user and the user's group members.</li> </ol>
Alternate flow of events	<p>2a. User selects a time period however the time period is not available.</p> <p>3a. User is shown an error message. System reverts to prior state.</p> <p>2b. User selects a time period that exceeds their permitted reservation duration or the amount of time allocated for the course.</p> <p>3b. User is shown an error message. System</p>

## TeamSpace Booking System - Software Requirements Specification

	reverts to prior state.
Post-conditions	User's group is allocated the time period.

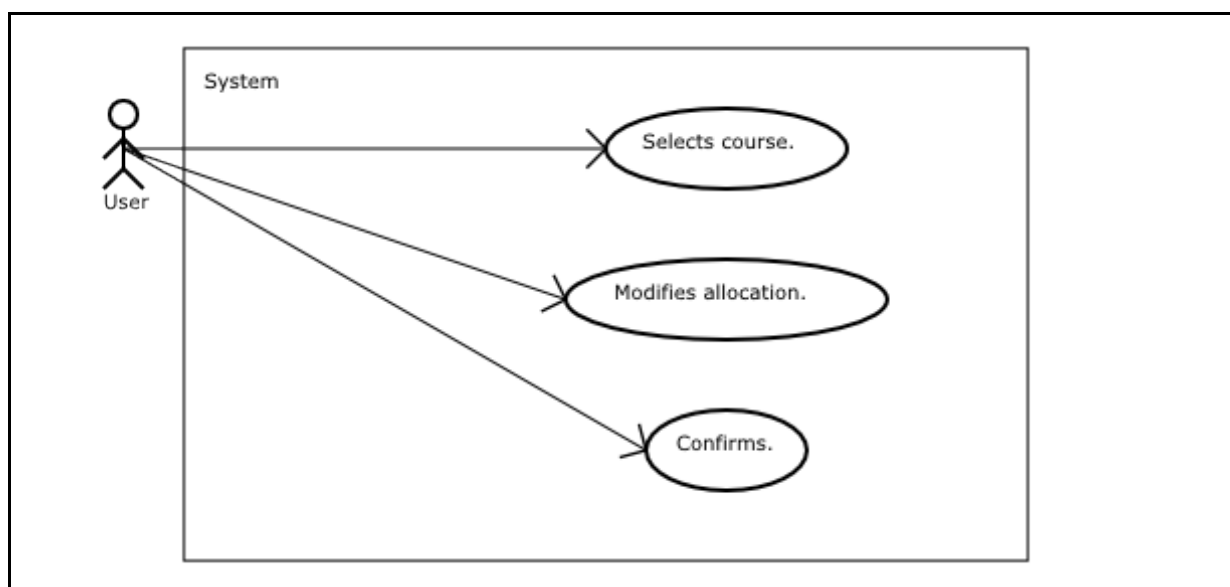
## 2.2.4 COURSE ADMINISTRATOR USE CASES

### 2.2.4.1 CREATE ALLOCATION



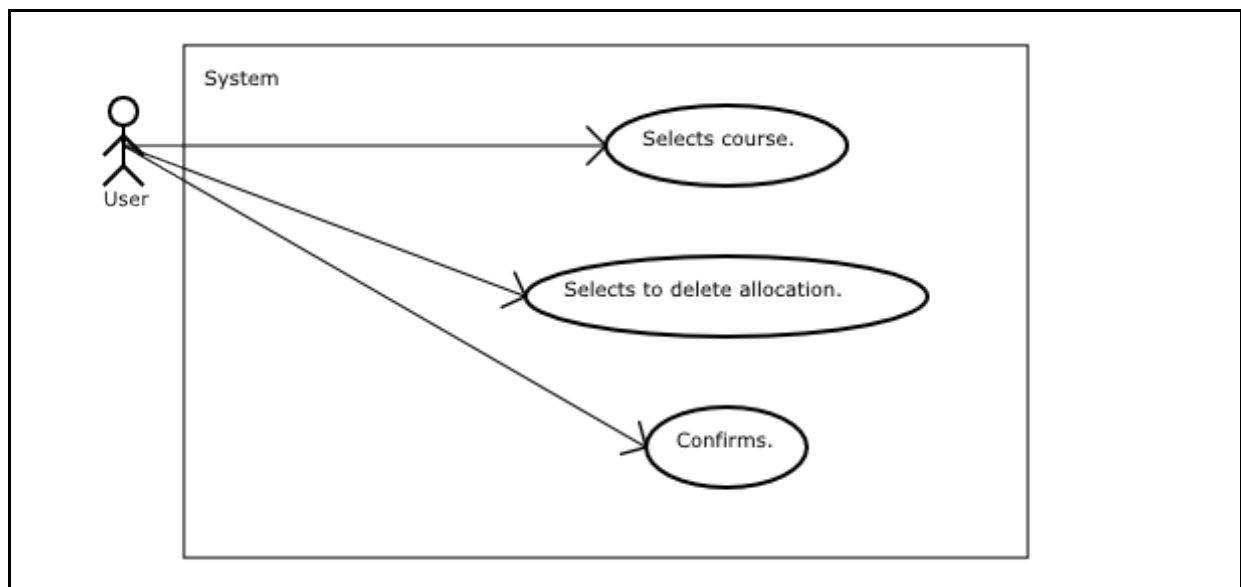
Brief Description	Course Administrator creates allocation.
Actors	Course Administrator and Room Administrator but designed for Course Administrator.
Preconditions	User is logged in.
Main flow of events	<ol style="list-style-type: none"> <li>1. User selects course that they wish to manage and is presented with available times.</li> <li>2. User selects available times for allocation.</li> <li>3. User confirms allocation.</li> </ol>
Alternate flow of events	<ol style="list-style-type: none"> <li>2a. User selects times that are not available.</li> <li>3a. User receives error message. System reverts to prior state.</li> <li>2b. User selects times that violate their constraints.</li> <li>3b. User receives error message.</li> </ol>
Post-conditions	Allocation is stored in the system.

## 2.2.4.2 MODIFY COURSE ALLOCATION



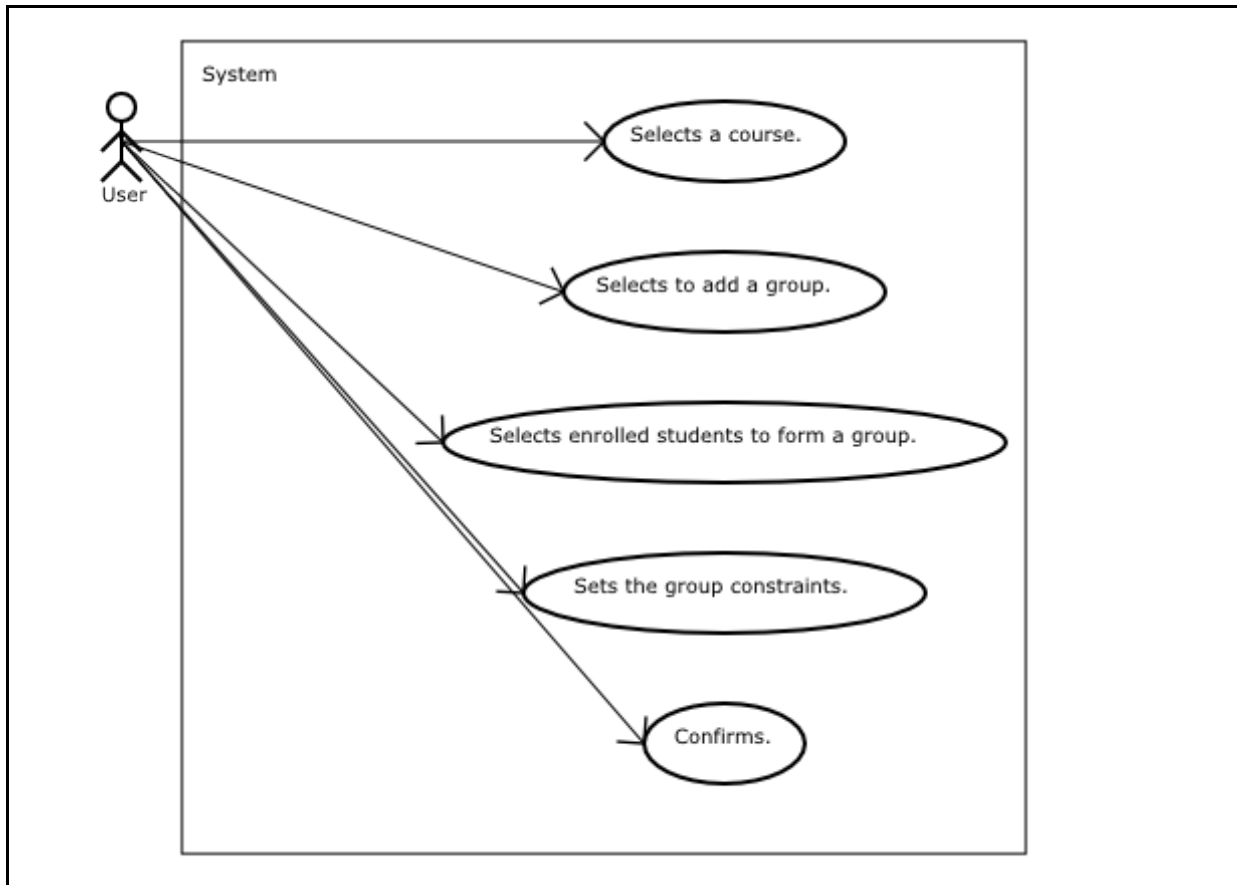
Brief Description	Course Administrator modifies an allocation in part or whole.
Actors	Course Administrator and Room Administrator but designed for Course Administrator.
Preconditions	User is logged in. User has previously created an allocation.
Main flow of events	<ol style="list-style-type: none"> <li>1. User selects the course they wish to manage and is presented with available allocations.</li> <li>2. User can modify:</li> <li>3. The time of the booking to another free time of the same or smaller duration.</li> <li>4. The length of the booking.</li> <li>5. The number of rooms required for each time period.</li> <li>6. User confirms changes.</li> </ol>
Alternate flow of events	<ol style="list-style-type: none"> <li>2a. User selects times that are not available.</li> <li>3a. User receives error message. System reverts to prior state.</li> <li>2b. User selects times that violate their constraints.</li> <li>3b. User receives error message.</li> </ol>
Post-conditions	Modifications are saved.

## 2.2.4.3 DELETE ALLOCATION



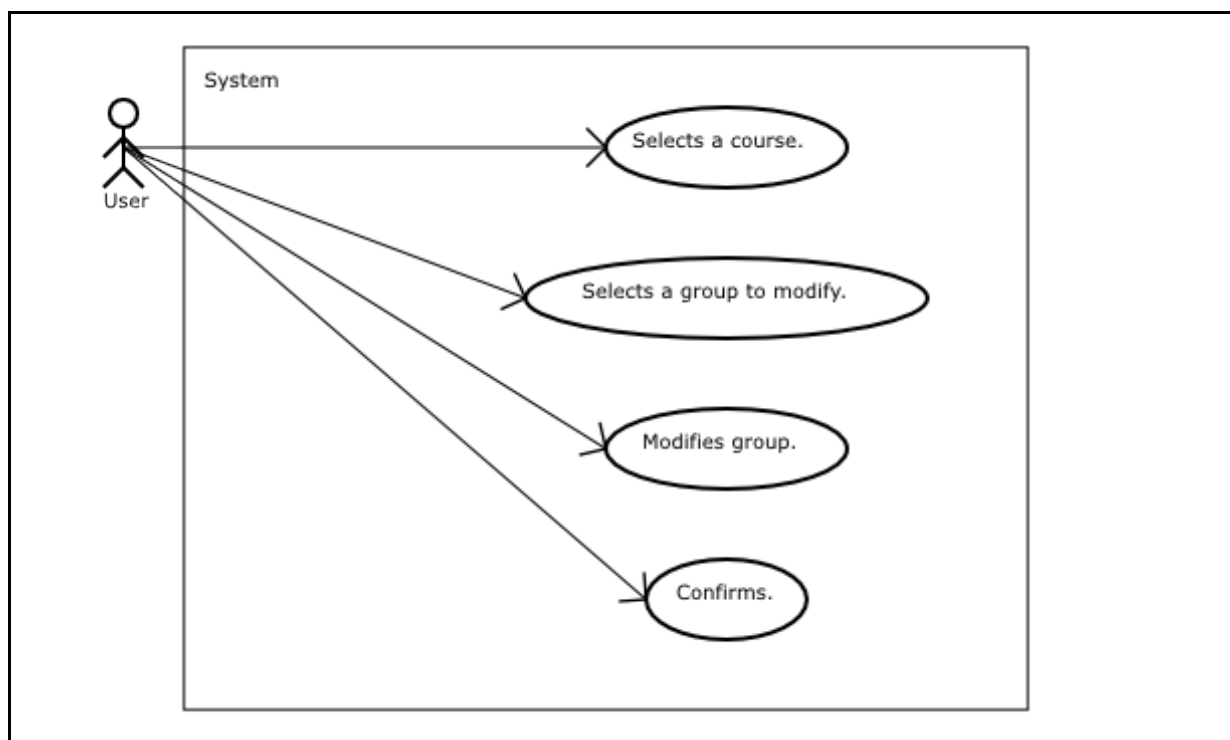
Brief Description	Course Administrator removes an allocation in part or whole.
Actors	Course Administrator and Room Administrator but designed for Course Administrator.
Preconditions	User is logged in. User has previously created an allocation.
Main flow of events	<ol style="list-style-type: none"> <li>1. User selects the course they wish to manage and is presented with available allocations.</li> <li>2. User can modify:</li> <li>3. The time of the booking to another free time of the same or smaller duration.</li> <li>4. The length of the booking.</li> <li>5. The number of rooms required for each time period.</li> <li>6. User confirms changes.</li> </ol>
Alternate flow of events	None.
Post-conditions	Modifications are saved.

2.2.4.4 ADD COURSE GROUP



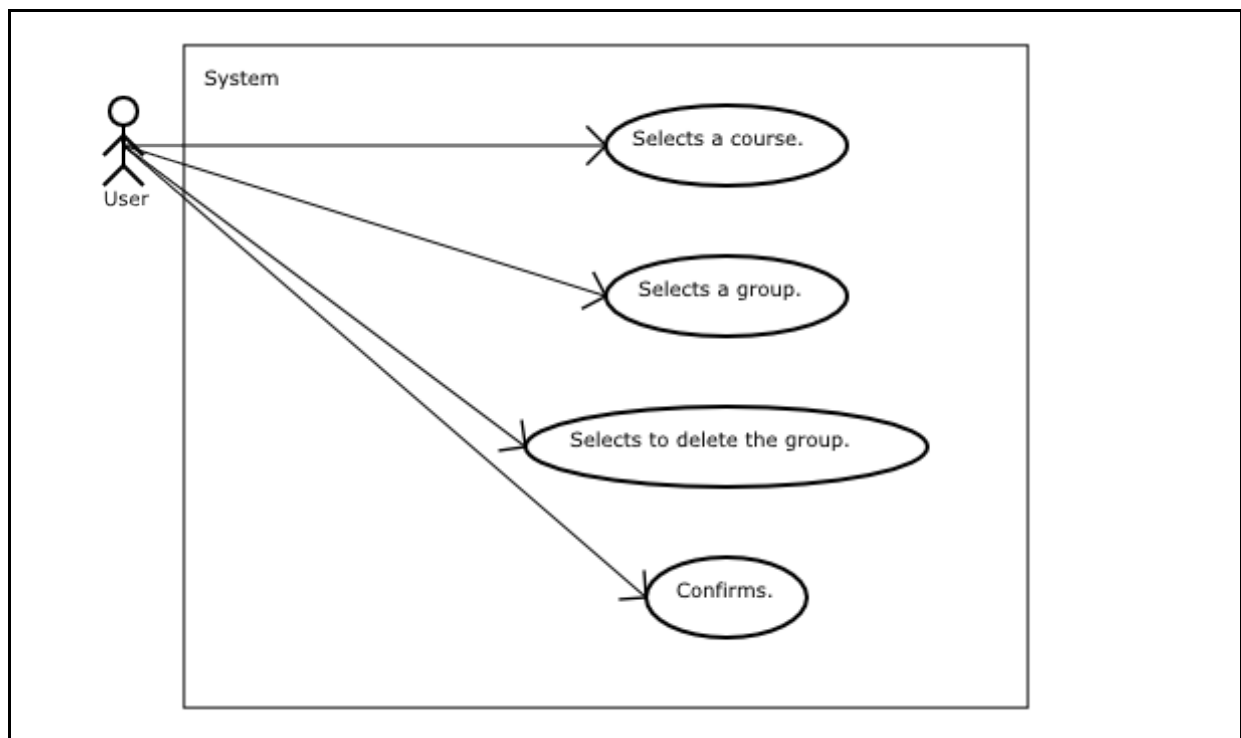
Brief Description	Course Administrator adds a group to their course.
Actors	Course Administrator and Room Administrator but designed for Course Administrator.
Preconditions	User is logged in Course has previously been created.
Main flow of events	<ol style="list-style-type: none"> <li>1. The user selects a course.</li> <li>2. The user selects to add group.</li> <li>3. The user selects students' details from the list of enrolled students retrieved from the ITEE database.</li> <li>4. Set group constraints</li> <li>5. The user confirms.</li> </ol>
Alternate flow of events	<ol style="list-style-type: none"> <li>4a. the selected constraints violates the course constraints.</li> <li>5a error message is reported.</li> </ol>
Post-conditions	The specified group is added to the system under that course code.

## 2.2.4.5 MODIFY COURSE GROUP



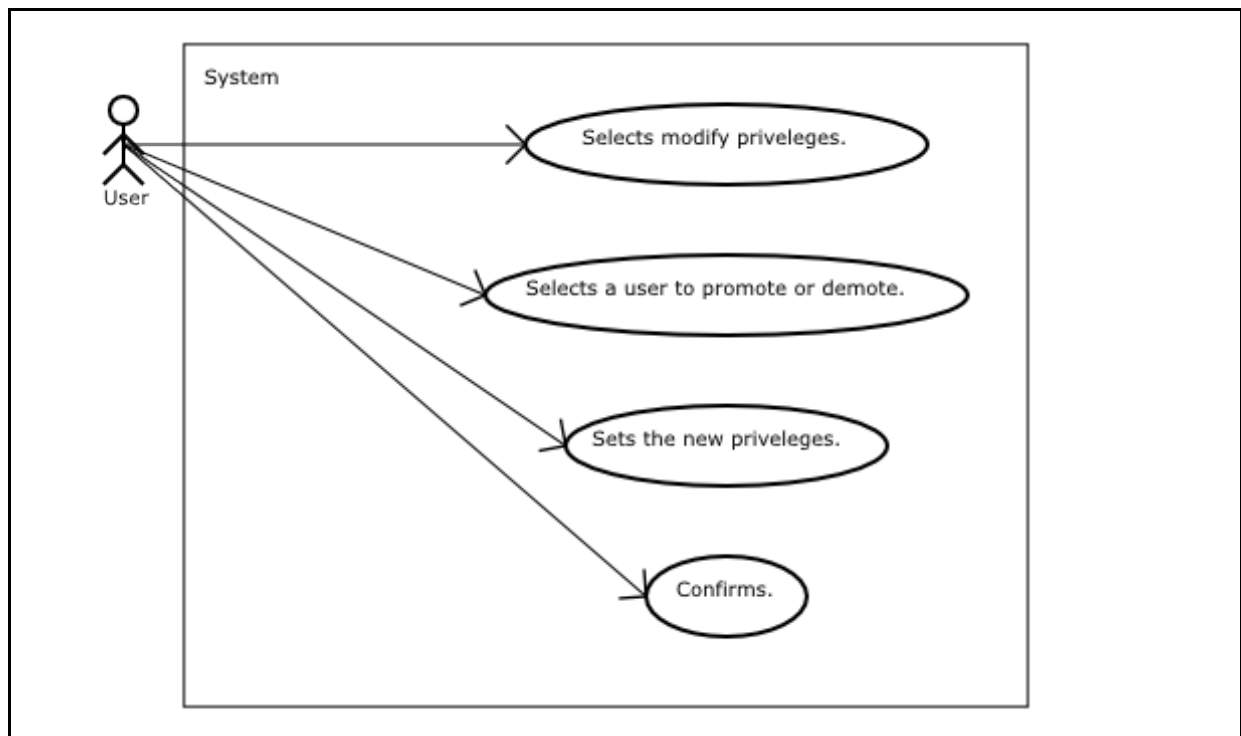
Brief Description	Course Administrator adds a modification to an existing group.
Actors	Course Administrator and Room Administrator but designed for Course Administrator.
Preconditions	User is logged in. Group has previously been created.
Main flow of events	<ol style="list-style-type: none"> <li>1. The user selects a course.</li> <li>2. The user selects a group to modify.</li> <li>3. Can modify:</li> <li>4. Students assigned to group.</li> <li>5. Group constraints.</li> <li>6. The user confirms.</li> </ol>
Alternate flow of events	<p>4a. the selected constraints violates the course constraints.</p> <p>5a error message is reported.</p>
Post-conditions	The specified group is modified.

## 2.2.4.6 REMOVE GROUP FROM COURSE



Brief Description	Course Administrator removes a group.
Actors	Course Administrator and Room Administrator but designed for Course Administrator.
Preconditions	User is logged in Group has previously been created.
Main flow of events	<ol style="list-style-type: none"> <li>1. The user selects a course.</li> <li>2. The user selects a group.</li> <li>3. The user nominates course deletion.</li> <li>4. The user confirms.</li> </ol>
Alternate flow of events	
Post-conditions	The specified group is removed from the course.

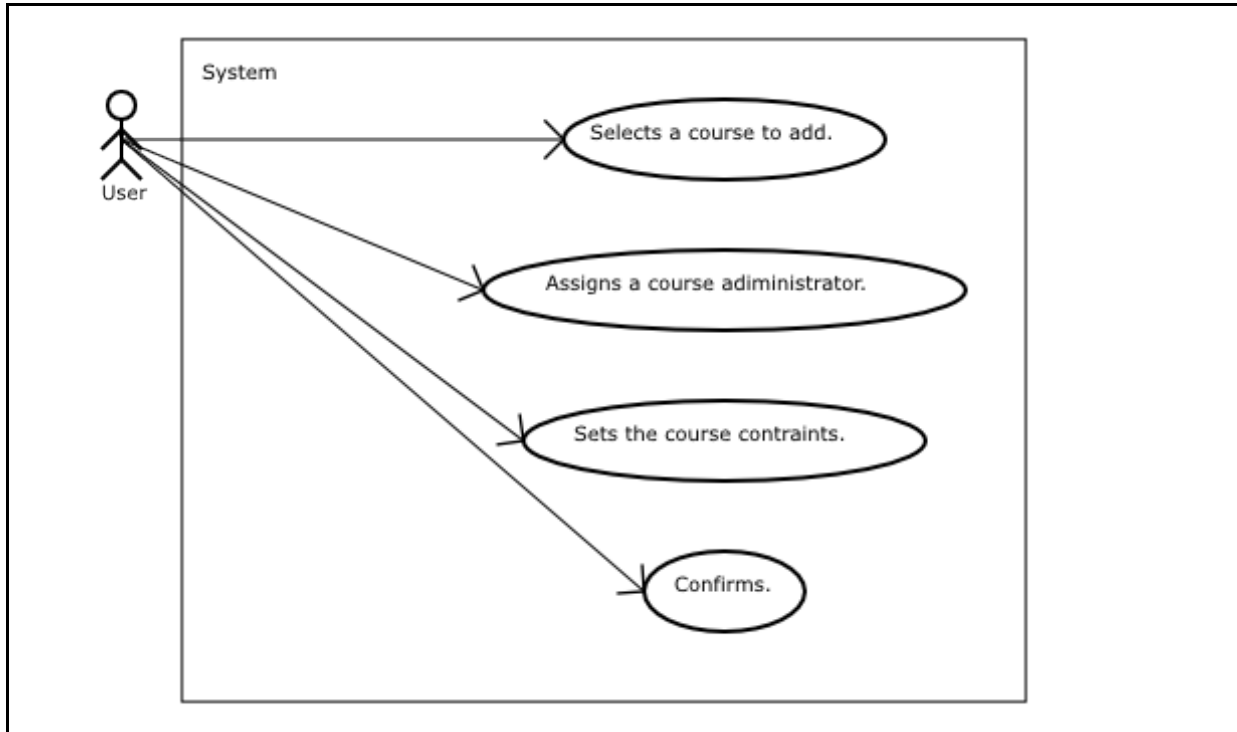
## 2.2.4.7 MODIFY USER PRIVILEGES



Brief Description	Grants the user additional privileges of selected by the Course Administrator
Actors	Room Administrator or other Course Administrator
Preconditions	User is logged in. User has rights to the required course
Main flow of events	<ol style="list-style-type: none"> <li>1. User selects promote user.</li> <li>2. User selects the recipient from the list of ITEE users.</li> <li>3. User sets the privileges of the recipient</li> <li>4. User confirms changes.</li> </ol>
Alternate flow of events	None.
Post-conditions	Recipient receives or loses privileges.

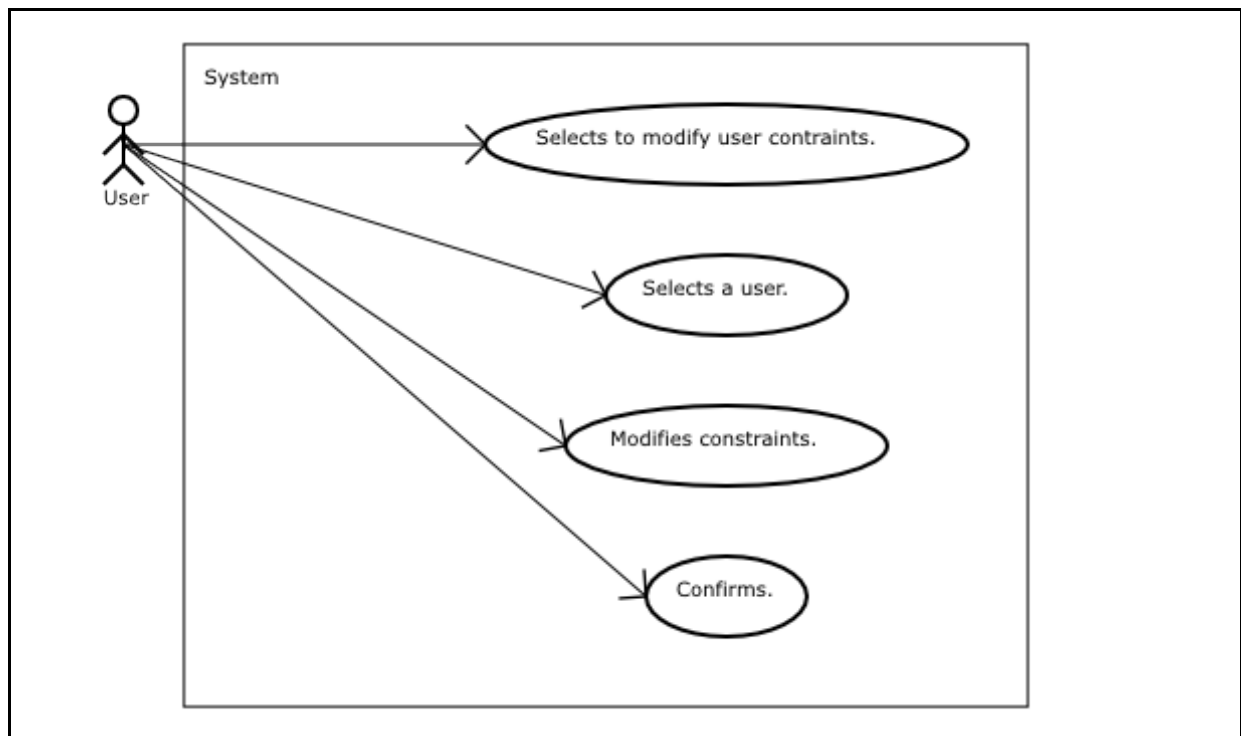
2.2.5 ROOM ADMINISTRATOR USE CASES

2.2.5.1 CREATE COURSE



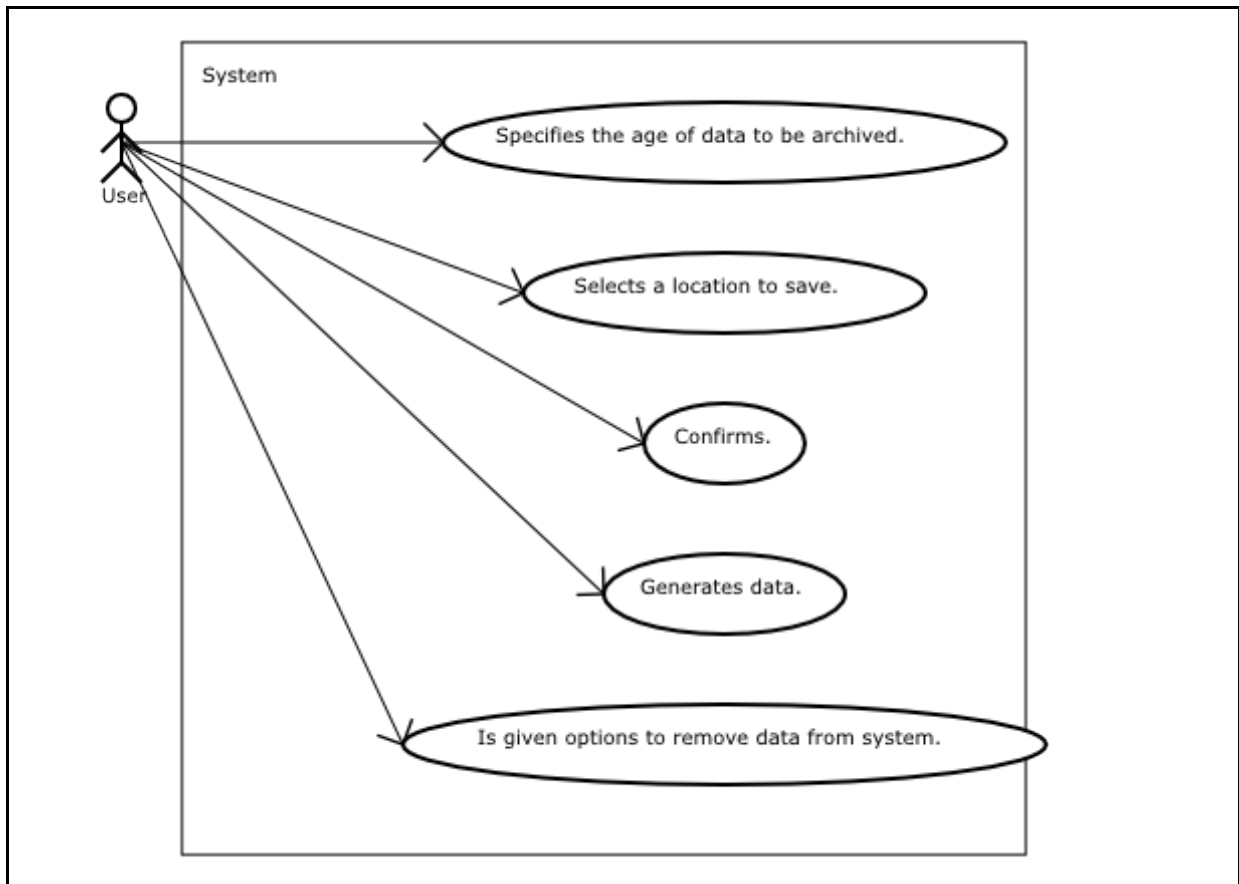
Brief Description	Administrator allows course to be allocated time in the TeamSpace Lab.
Actors	Room Administrator.
Preconditions	User is logged in.
Main flow of events	<ol style="list-style-type: none"> <li>1. User selects to add a course from the list of ITEE courses.</li> <li>2. user assigns an existing course Administrator or promotes one to administer the course.</li> <li>3. User sets the course constraints</li> <li>4. User confirms changes.</li> </ol>
Alternate flow of events	2a User aborts the process. System returns unchanged.
Post-conditions	Course is added to the system

## 2.2.5.2 MODIFY USER CONSTRAINTS



Brief Description	Changes the constraints applied to a user.
Actors	Room Administrator and Course Administrator
Preconditions	User is logged in User is responsible for recipient
Main flow of events	<ol style="list-style-type: none"> <li>1. User selects modify user constraints.</li> <li>2. User selects a User to modify from a list of subordinate users.</li> <li>3. User updates user constraints.</li> <li>4. User Confirms.</li> </ol>
Alternate flow of events	3a User specifies constraints that are outside their ability to confer
Post-conditions	Subordinate user is updated.

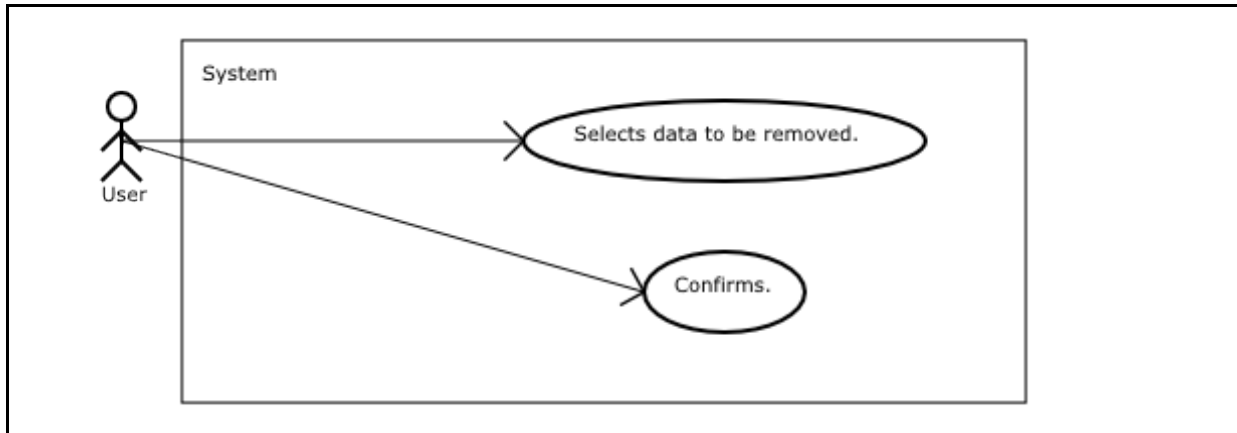
2.2.5.3 ARCHIVE DATA



## TeamSpace Booking System - Software Requirements Specification

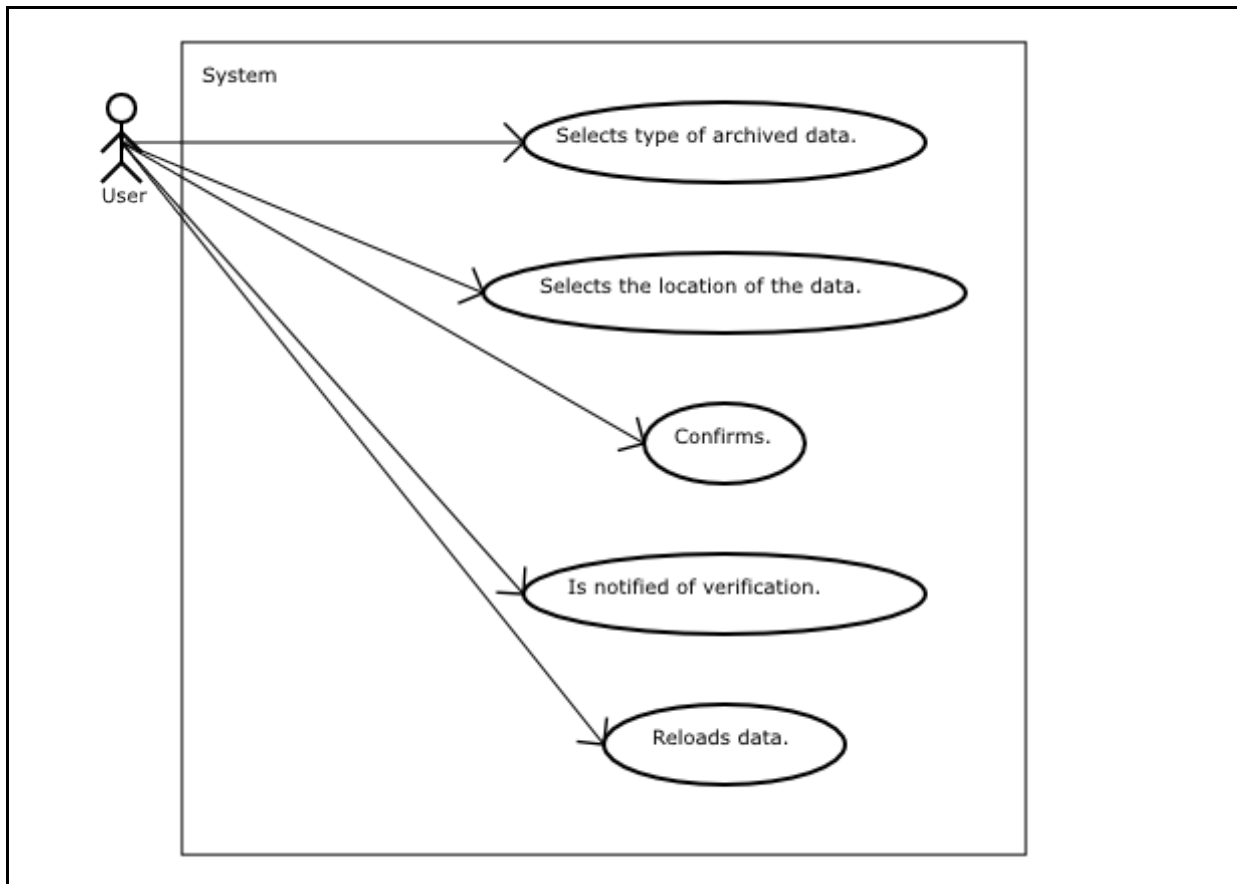
Brief Description	Room Administrator archives any data associated with the system.
Actors	Room Administrator
Preconditions	Room Administrator is logged into the UQTSBS system.
Main Flow of Events	<ol style="list-style-type: none"> <li>1. Room Administrator specifies the age of data to be archived.</li> <li>2. Room Administrator selects the location to save the archived data.</li> <li>3. Room Administrator confirms the archival.</li> <li>4. Room Administrator chooses whether or not to remove archived data from the system.</li> </ol>
Alternative Flow of Events	<p>3a System Administrator cancels archival process before confirmation; data is not archived.</p> <p>4b Archival process failed; system displays the appropriate error message and details of the failure.</p>
Post-conditions	Data is archived by the System. Archived data may be removed from the System.

2.2.5.4 REMOVE ARCHIVED DATA FROM SYSTEM



<p>Brief Description</p>	<p>The Room Administrator has an option to remove archived data from the system once an archival process has been completed successfully. This is to free up system resources by removing outdated (and archived) data.</p>
<p>Actors</p>	<p>System Administrator</p>
<p>Preconditions</p>	<p>The Room Administrator is logged into the UQTSBS System. The Room Administrator has completed the Archive Data process successfully and opted to remove the archived data from the system.</p>
<p>Main Flow of Events</p>	<ol style="list-style-type: none"> <li>1. System displays details of data to be removed and prompts user to confirm the removal.</li> <li>2. Room Administrator confirms the removal process.</li> <li>3. Archived data is removed from the system.</li> </ol>
<p>Alternative Flow of Events</p>	<p>2a Room Administrator cancels the remove archived data process before confirmation. No data will be removed from the system. 3b Removal process not completed due to some failure; appropriate error message and details of failure is displayed. No data will be removed from the system.</p>
<p>Post-conditions</p>	<p>Data recently archived is removed from the system and put into backup store.</p>

2.2.5.5 RELOAD ARCHIVED DATA INTO SYSTEM

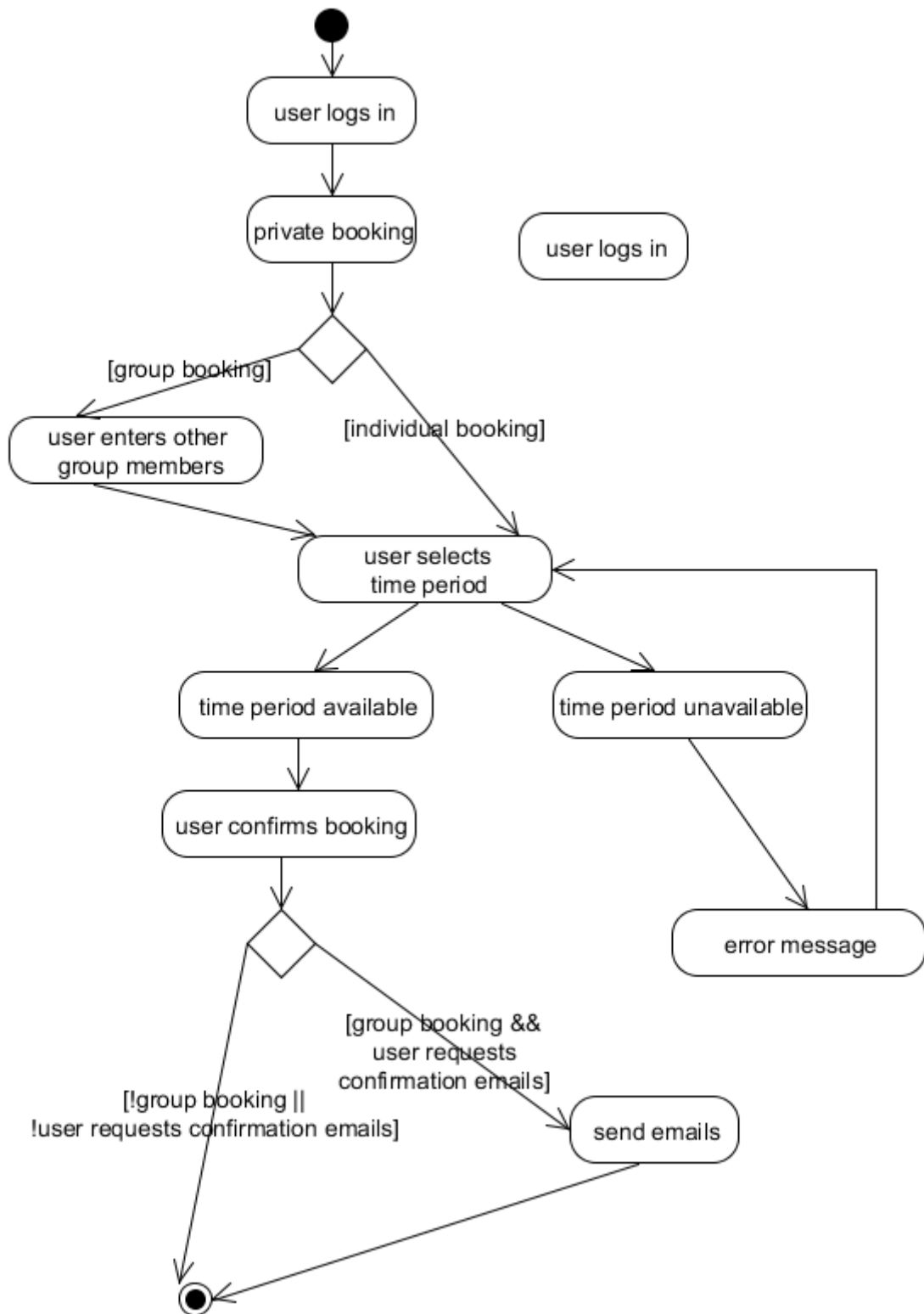


Brief Description	Room Administrator reloads archived data into the system.
Actors	Room Administrator
Preconditions	Room Administrator is logged into the UQTSBS System.
Main Flow of Events	<ol style="list-style-type: none"> <li>1. Room Administrator selects the type of archived data to be reloaded into the system.</li> <li>2. Room Administrator selects the location of the archived data file in XML format to be used.</li> <li>3. Room Administrator confirms the reload process.</li> <li>4. System verifies the contents of the archived data file.</li> <li>5. System reloads the archived data into the database.</li> </ol>
Alternative Flow of Events	2a Room Administrator cancels the reload process before confirmation; data is not

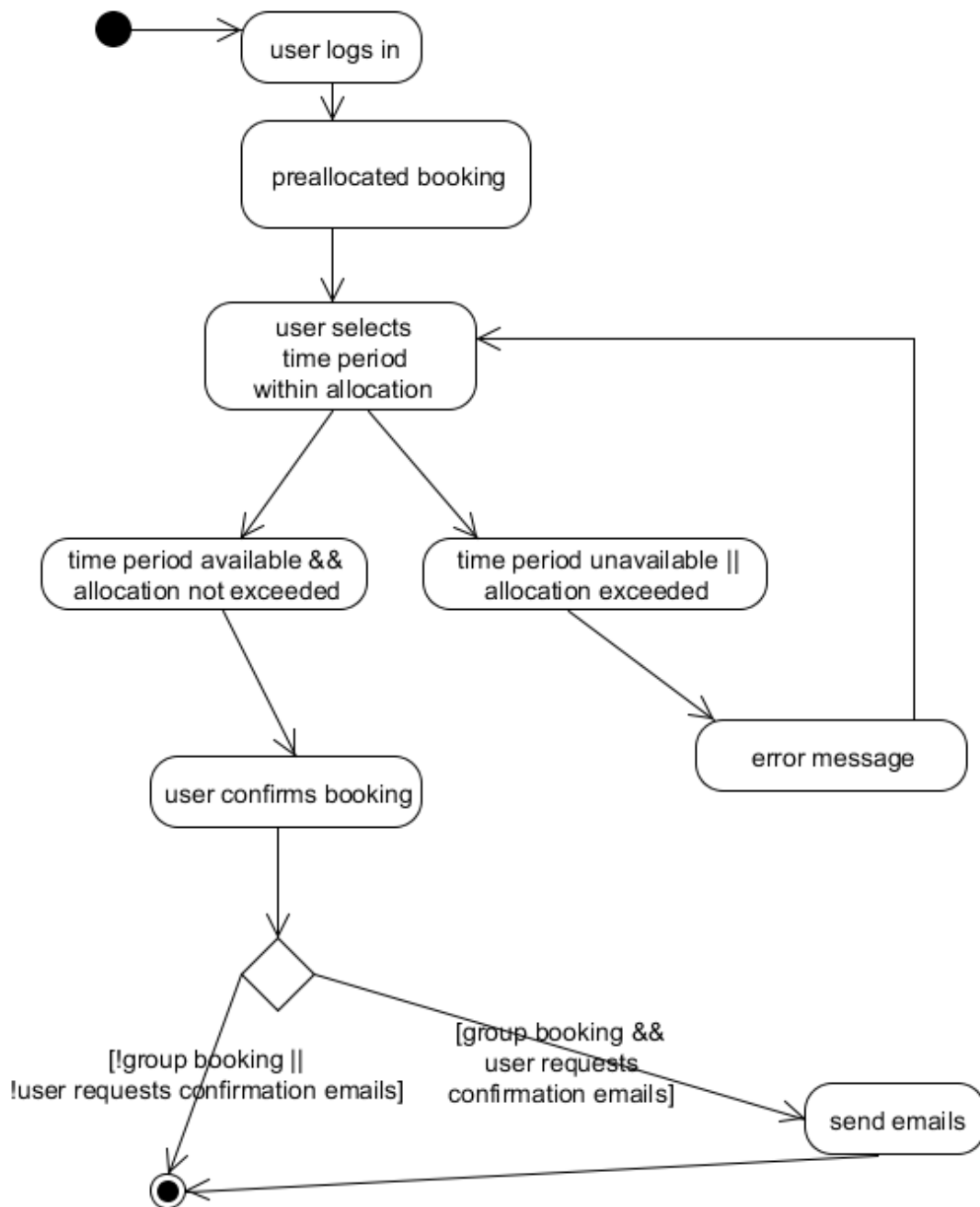
## TeamSpace Booking System - Software Requirements Specification

	reloaded. 4b System detects that the contents of the archived data file is invalid; System displays the appropriate error message. 5c Reload process failed; System displays the appropriate error message and details of the failure.
Post-conditions	Archived data is restored back into the System.

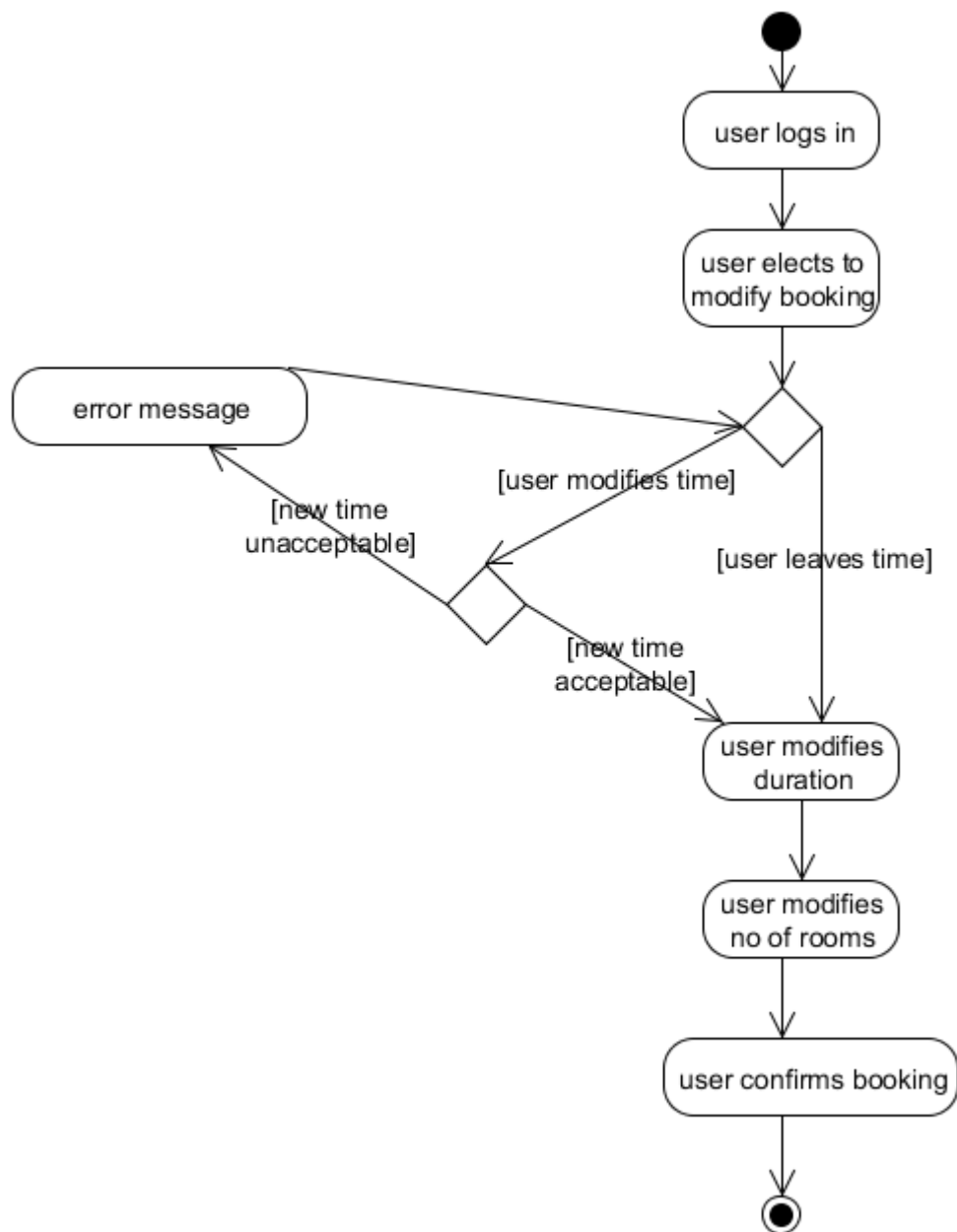
Activity diagram – student: make private booking.



Activity diagram – student: make booking in pre-allocated time.



Activity diagram – student: modify booking.



## NON-FUNCTIONAL REQUIREMENTS

---

This section is divided into three subsections addressing software and hardware respectfully. These sections outline requirements of the system that do relate directly to the software feature set.

---

### 2.2.6 SOFTWARE OPERATING ENVIRONMENT

---

The system must run on an Internet browser. The Internet browsers supported will be Internet Explorer (>6), Firefox (>2), Opera (>9) and Google Chrome (>2). The client operating systems supported will be Microsoft Windows (XP and later), Macintosh OS X (10.4 and later), and any Linux distribution with a Kernel later than 2.6.24. However, the system is expected to work on any other configuration which supports rendering w3c compliant XHTML 1.0 and Javascript 1.8.

---

### 2.2.7 SOFTWARE NETWORK IMPACT

---

The system must support up to 50 simultaneous requests and bookings over all three access levels.

---

### 2.2.8 SOFTWARE DATABASE SIZING

---

The system should be able to hold at least 1,500,000,000 records before archiving is undertaken. With archiving the system should be able to hold as many records as memory permits.

---

### 2.2.9 HARDWARE

---

The TeamSpace room will contain several physical interfaces that provide in-room access to the booking system and display the real-time usage of the TeamSpace lab resources. Some of the facilities present in the room can also be controlled in part by the booking system.

The primary user interface will be available both on the Internet and via stationary terminals in the common area of the Lab itself. A simplified interface will be available in the meeting rooms, informing users of current and future bookings.

The room has a common information terminal near the entrance that serves to show all the room allocations for the next four hours, the current availability of the provided desktop computers and the real-time lab usage. This terminal is mounted in a standalone enclosed structure that contains a screen with a minimum display area of 24". This terminal is an information terminal only and does not have any user accessible input devices.

The walls and doors of the meeting areas are made of glass that is tinted to provide privacy. People within the room can see out; however, people cannot see into the room easily. The doors to the meeting rooms will also feature a 45cm thick 'window stripe' at average head height made of a electrically polarising OLED tint that can be adjusted both manually (from inside the room), and by the booking system from clear to tinted. The OLED tinting can be made clear by the booking system if no booking exists for the meeting room. No restrictions exist for the unpowered state of the window stripe.

Fitted to the meeting room doors and located under the window stripe, is a 17" OLED display that shows the current room booking schedule. A user can make an impromptu booking of

## TeamSpace Booking System - Software Requirements Specification

free room time (for only the current day) by selecting the slot and using their student card in the door-frames RF-ID reader.

The OLED display is connected to a dumb terminal housed in the meeting room door's handle which is provided power and data cabling via a customised, self arresting, soft close hinge. The doors handle plate can function as a heat sink for the dumb terminal. The door OLEDs can only display the current room schedule, up to 24 hours from the current session.

The RF-ID reader must also allow the use of traditional magnetic swipe cards. The RF-ID component must have a minimum effective detection range of 10 CM.

---

## 2.2.10 RELIABILITY/DEPENDABILITY

---

### 2.2.10.1 SOFTWARE

---

The system should be reliable in performing all operations described in this document. Software failure are recognised to be unavoidable however they should occur infrequently and the system should be reliable as long as the numbers of requests are less than the supported number of requests and the database size is less than the supported size. 99.999% uptime is expected to be achieved.

The booking system must have 24-hour access from on-campus computers. This includes the use of personal computers or handheld devices using the wireless network.

In the case of client or server software failure, it is required that the system be returned to fully operational status in no less than 2 hours. Should hardware fail, the permitted period of downtime may be extended to no less than three days.

### 2.2.10.2 HARDWARE

---

The non-interactive interface provided at the entrance of the room must have a mean weekly operation time of 170 hours. This machine must be operational during work days from 8AM to 5PM. The electronics of the pedestal terminal must also be able to be rapidly replaced in the event of a hardware failure.

The polarising view window must be able to transition completely at all times even when subjected to localised damage. The window must have a mean time to failure in excess of 40,000 hours.

The OLED schedule screens must have an expected lifetime of 40,000 hours and an impact tolerance of 2.5N/mm<sup>2</sup>. To achieve life expectancy, mono colour displays are acceptable however full colour is preferable. The dumb terminal hardware embedded in the door handles must have a mean time to failure in excess of 150,000 hours.

---

## 2.2.11 SECURITY

---

Access to the room is available to all staff and students between the hours of 8 am - 5pm weekdays. After-hours access is restricted by swipe card to approved students and staff who take responsibility for the facilities upon entry.

The software must log all actions taken by users within the software interface including the creation, editing and deletion of bookings, for a period of at least four years. The system

## TeamSpace Booking System - Software Requirements Specification

must also record student swipe history from after hour access for the duration of their degree. These logs must be stored both locally in the room server and at a remote location.

---

## 2.2.12 AVAILABILITY

---

### 2.2.12.1 SOFTWARE

---

The software which interacts with the database must be sufficiently abstracted whereby the software is not dependent on the database existing locally, or even on the same sub-net. This allows for instances where the database terminates unexpectedly, or a hardware failure occurs. In such cases, a transition over to a second, redundant, database can take place. The web server and database should also be executed within a virtual machine and periodically backed up. This allows for a quick recovery of data upon failure and the ability to transition to alternative hardware relatively quickly.

### 2.2.12.2 HARDWARE

---

The hardware within the server should support a redundant RAID array to allow for disk failure. It is recommended to use ECC ram to enhance the stability of memory paging. In the event of a power failure at UQ, a UPS will be installed for the server to maintain up-time and availability for external users.

---

## 2.2.13 MAINTAINABILITY

---

### 2.2.13.1 SOFTWARE

---

The use of a virtual machine will enhance the ability to maintain the code base, where rollbacks can occur often and the state of the system can be saved and archived at any point.

The redundancy of the database as previously specified also allows for further maintainability of the software, where another server can be temporarily used to interact with the database in the event that the code-base on the existing server is updated.

### 2.2.13.2 HARDWARE

---

Given the nature of the pedestal computers, as they are fully redundant and non-interactive, maintainability of the pedestal computers is simple as they are accessible.

The card readers are readily available components, and given the nature of their design they can be swapped out easily. This allows for easy maintainability of the card readers.

Various attributes have been previously specified to enhance the availability of the centralised server. These attributes also enhance the maintainability of the system, given the redundancy.