SOFTWARE REQUIREMENTS SPECIFICATION

for

The Force Awakens

PREPARED BY:

Name	Student ID	Email
Erin Benderoff	27768478	erin.benderoff@gmail.com
Syrine Krim	29773118	syrine_krim@hotmail.com
Ryan Lee	27752504	rswllee@hotmail.com
Jeremy Melnyk	26374603	jeremyjm91@gmail.com
Dimitri Topaloglou	29358269	dtopaloglou@gmail.com
Kevin Yasmine	27195346	kevinyasmine@gmail.com

Instructor: Dr. Constantinides

Course: SOEN 344

Date: April 4, 2017

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

Students in the Engineering and Computer Science (ENCS) faculty at Concordia are to complete several large academic projects, including an 8-month project known as Capstone that takes a lot of care and time. To work on these projects, students must have access to private rooms throughout the year.

The Force Awakens room reservation system was created by another group in SOEN 343 for this purpose. For SOEN 344, the system was refactored with an updated list of requirements. New requirements include reservation limits, such as a maximum of 3 repeat reservations, as well as a list of equipment that can be reserved along with a room in limited quantities. In order to incorporate all these changes and have the system function efficiently, the system was refactored appropriately.

1.1 PURPOSE AND SCOPE OF THE SRS

This document provides a detailed description of the functional and non-functional requirements of The Force Awakens system. The most significant functional requirements are captured as use cases that describe scenarios of user interaction with the system. From these use cases, system sequence diagrams are constructed that describe the messages sent between the user and the system viewed as a black box. Critical system operations are identified and described by operation contracts. Finally, a domain model visualizes the relationships between the conceptual classes of the problem domain. The SRS is used a starting point for architectural design and implementation.

Because the SRS contains no details on the actual architecture or implementation of the system, it can be read and understood by all stakeholders, even those with little to no expertise in software development, allowing them to provide important feedback.

1.2 DEFINITIONS, ACRONYMS AND ABBREVIATIONS

SRS: Software Requirements Specification

UC: Use Case

SSD: System Sequence Diagram

2. OVERALL DESCRIPTION

2.1 PRODUCT FUNCTIONS

This system is a room reservation system that allows ENCS students to reserve rooms at selected times throughout the week. Users can create reservations in 5 different rooms on a 24 hours basis. Users can also request equipment to rent during their reservation. At any point before the reservation start time, users can modify their reservation and/or equipment requests. If a room is already reserved at the user's desired time, or if the requested equipment is unavailable, the user is placed on a waitlist for that room. In the event of a cancellation, any Capstone students on the waitlist are given priority for the reservation, followed by non-Capstone students.

2.2 USER CHARACTERISTICS

This system is intended to be used by Concordia students in the ENCS faculty wishing to reserve a private room to work on their Capstone and/or other academic projects. Given that they are in university, the users will likely possess a medium to high level of technical expertise and should not have any trouble using the online system.

2.3 CONSTRAINTS

This system shall constrain its users such that they cannot create too many reservations at once, in order to make the system fair. To do this, users shall be limited to 3 hours of reservations per week, and cannot create a repeated reservation for more than 3 weeks. While a user is creating, modifying or deleting a reservation in a given room, all other users will denied write access to that room, providing mutual exclusion. Furthermore, users are not allowed to access the system without valid credentials, providing security for the system.

2.4 ASSUMPTIONS AND DEPENDENCIES

It is assumed that the user is using either a computer or mobile phone and has a working connection to the Internet. The user is assumed to have enough technical expertise to be able to navigate the system and use its functionalities. The users are assumed to know the constraints of the system.

3. SPECIFIC REQUIREMENTS

3.1 FUNCTIONAL REQUIREMENTS

3.1.1 LOGIN/LOGOUT

- System shall authenticate existing users with a username and password prior to giving them access to any read (view registry/reservations) or write (create/modify/cancel) function.
- System shall terminate the user's session upon logout request.

3.1.2 VIEW REGISTRY

- System shall display a complete room registry showing the reserved and available timeslots of each room.
- The room registry contains 5 rooms available on a 24-hour basis.
- Multiple users can view the registry at a time.

3.1.3 VIEW RESERVATIONS

- System shall allow users to view a list of their own reservations and/or any waitlisted reservations.

3.1.4 RESERVE ROOM

- System shall allow users to reserve a room at a given timeslot.
    - If the timeslot is already reserved by another user, the reservation is not created and the user is placed on a waitlist pending availability of the room.
- Users can reserve a maximum of 3 hours per week.
    - Once the user reaches the maximum, they are removed from any waitlists they were on and cannot place themselves on another waitlist.
- Users can create a repeated reservation (in the same room at same time) for up to 3 consecutive weeks.
- Priority for reservations is given to Capstone students
- Only one user can access a room for reservation creation at a time.
- During a room reservation, system shall allow users to request equipment (computers and projectors) for the room they wish to reserve.
    - If the requested equipment is unavailable, the reservation is not created and the user is placed on a waitlist pending availability of the equipment.

3.1.5 MODIFY RESERVATION

- System shall allow users to modify the time, room, title and/or equipment requests for their own existing reservation at any time up until the start time of the reservation.
- Upon modification, if a waitlist exists for the room and timeslot that was just freed, the first Capstone student on the waitlist is automatically given the reservation, or if there are no Capstone students on the waitlist, it is given to the first non-Capstone student.
- Only one user can access a room for reservation modification at a time.

3.1.6 CANCEL RESERVATION

- System shall allow users to cancel their own existing reservation at any time up until the start time of the reservation.
- System shall allow users to cancel a pending (waitlisted) reservation at any time up until the start time of the reservation, otherwise it will be automatically cancelled by the system.
- Upon cancellation, if a waitlist exists for the room and timeslot that was just freed, the first Capstone student on the waitlist is automatically given the reservation, or if there are no Capstone students on the waitlist, it is given to the first non-Capstone student.
- Only one user can access a room for reservation cancellation at a time.

3.2 USE CASE VIEW

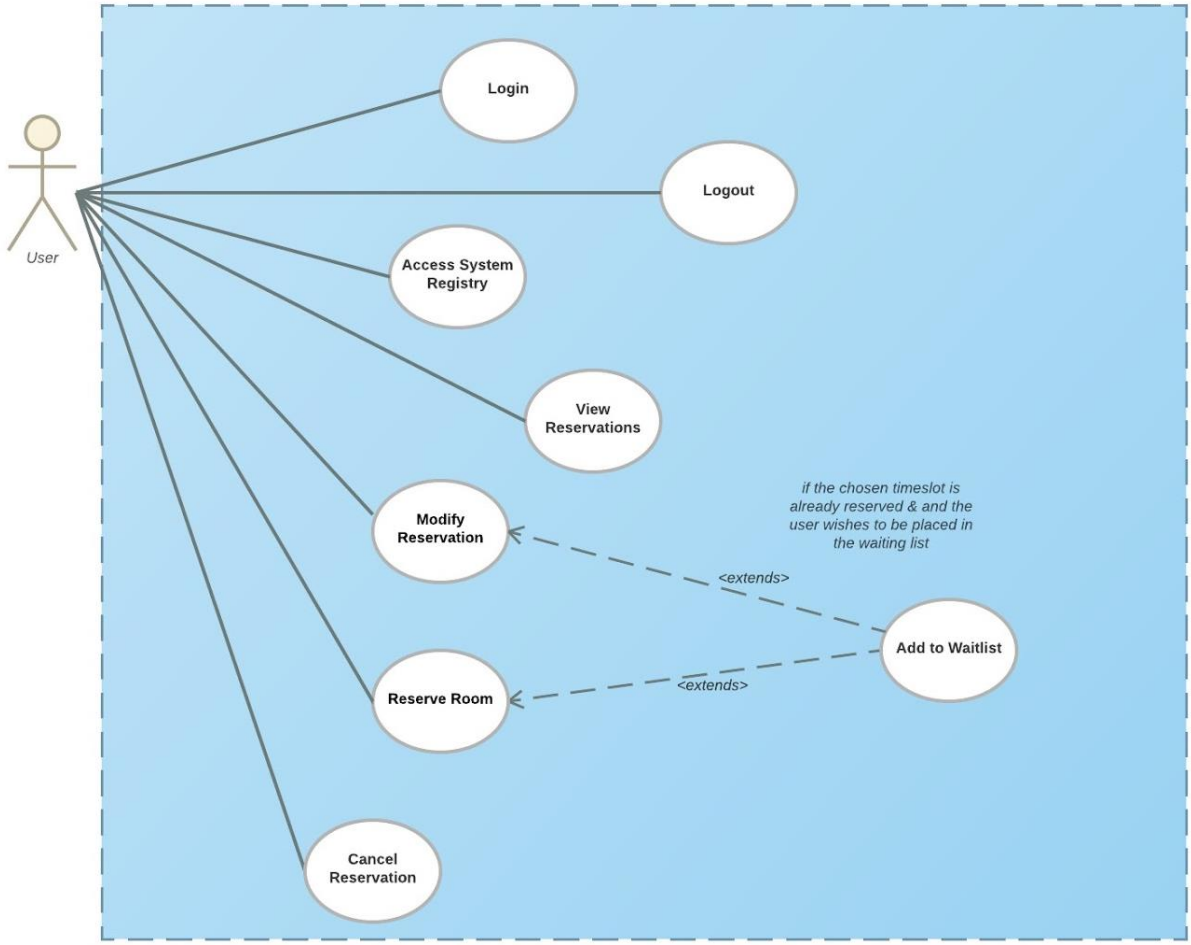

Figure 1: Use Case View

3.3 USE CASES

3.3.1 UC 1 – LOGIN

Use Case ID:	**UC1**
Use Case Name:	Login
Created By:	Ryan Lee
Actor(s):	User
Goal/Actor Goals:	The user gains access to the system.
Description/Summary:	In order to access the system, the user enters his username and password, which are verified. Once they are successfully authenticated the user gains access.
Preconditions:	• The user is not yet logged into the system.
Post-conditions:	• The user wishes to log in to the system.
Minimum Guarantee:	Unregistered users are denied access to the system.
Basic Flow:	1. The user provides a valid username and password. 2. The system indicates that the user has successfully logged in.
Alternative flows	• The inputted username is invalid. System displays an error message and prompts the user to try again. • The inputted password is invalid. System displays an error message and prompts the user to try again.

3.3.2 UC 2 – LOGOUT

Use Case ID:	UC2
Use Case Name:	Logout
Created By:	Ryan Lee
Actor(s):	User
Goal/Actor Goals:	The user wishes to log out from the system.
Description/Summary:	The user asks the system to log out, which consequently terminates his/her session.
Preconditions:	• The user is logged into the system.
Post-conditions:	• The user's access to the system is terminated.
Minimum Guarantee:	The user's session is preserved.
Basic Flow:	1. The user requests to log out from the system. 2. The system indicates that the user has successfully logged out.

3.3.3 UC 3 - ACCESS REGISTRY

Use Case ID:	**UC3**
Use Case Name:	Access System Registry
Created By:	Ryan Lee
Actor(s):	User
Goal/Actor Goals:	The user wishes to view all current availabilities and reservations in the system.
Description/Summary:	The user would like to know which timeslots are currently available for each room in the registry, and which ones are reserved. The system retrieves this information and displays it to the user.
Preconditions:	• The user is logged into the system.
Post-conditions:	• A list of available and reserved timeslots for each room is displayed
Minimum Guarantee:	The system fails to fulfill the request and the registry is not displayed
Basic Flow:	1. User requests access to system registry 2. System displays a list to the user currently available and reserved timeslots for each room from the reservation registry

3.3.4 UC 4 - VIEW RESERVATIONS

Use Case ID:	**UC4**
Use Case Name:	View Reservations
Created By:	Ryan Lee
Actor(s):	User
Goal/Actor Goals:	Allow users to view their own reservations
Description/Summary:	This allows the user to view a list of their existing reservations as well as a list of any waitlists they placed themselves on.
Preconditions:	• The user is already logged in
Post-conditions:	• A list of the user's reservations and waitlists is displayed
Minimum Guarantee:	The system fails to fulfill the request and does not display the user's reservations
Basic Flow:	1. User requests to view their own reservations 2. System displays a list of all reservations created by the user as well as any waitlists the user is on

3.3.5 UC 5 - RESERVE ROOM

Use Case ID:	**UC5**
Use Case Name:	Reserve Room
Created By:	Ryan Lee
Actor(s):	User
Goal/Actor Goals:	The user wishes to book a timeslot for a given room with equipment
Description/Summary:	The user selects an available time period and equipment they wish to book in a given room. The system creates a reservation for the user during this time period.
Preconditions:	• The user is logged into the system • The user was able to view the list of availabilities and reservations (UC3) • The room is not locked • The user has not exceeded their maximum of 3 reservations per week • The user has not exceeded their maximum of 3 hours reserved per week • Any equipment that the user has selected is available
Post-conditions:	• The timeslot the user requested to book is reserved in the user's name for the given room • The user is removed from any waitlists they were on at the same timeslot in other rooms • Any equipment that was selected is marked as in use at that time
Minimum Guarantee:	The reservation is not created and the state of the reservation registry is unchanged
Basic Flow:	1. From the reservation dashboard, the user selects an available time period for their reservation, between 30 minutes and 3 hours, and equipment for a given room 2. System asks the user to confirm the reservation for this period 3. User confirms the reservation 4. System generates a reservation in the user's name at the indicated timeslot(s) in the given room 5. System adds the reservation to the registry 6. If applicable, system removes the user from any waitlists they were on at the same timeslot in other rooms. 7. System displays a "success" message to the user
Alternative Flows	• Step 1: user selects a time period that exceeds the maximum bookable time period of 3 hours. System displays an error message. • Step 1: the desired time period includes one or more unavailable time slots. In this case, the flow changes to that of "Add to Waitlist" (UC6) • Step 1: the desired time period doesn't have the desired equipment available. In this case, the flow changes to that of "Add to Waitlist" (UC6) • Step 3: the user does not confirm the reservation and instead chooses to cancel their request. System brings user back to the dashboard.

3.3.6 UC 6 - ADD TO WAITLIST

Use Case ID:	**UC6**
Use Case Name:	Add to Waitlist
Created By:	Ryan Lee
Actor(s):	User
Goal/Actor Goals:	The user wishes to be placed on a waitlist of a room at a specific timeslot
Description/Summary:	This use case allows the user to place themselves onto a waitlist for a timeslot of a given room that has already been reserved by someone else or has unavailable equipment at the specified time.
Preconditions:	• The user is already logged in • The requested room is reserved at that specific time slot or the equipment is unavailable at that specific time slot
Post-conditions:	• The user is placed in a waitlist of a room at a specific timeslot
Minimum Guarantee:	The user is not added to any waitlists and the state of the internal system remains unchanged
Basic Flow:	1. From the reservation dashboard, the user selects an unavailable timeslot for a given room or selects a timeslot while equipment is unavailable 2. The system asks the user if they wish to be waitlisted for this timeslot 3. User confirms they would like to be placed on waitlist 4. The system adds the user to the waitlist of that room at the specified timeslot. 5. The system displays “success” message indicating the user has been placed on the waitlist.
Alternative Flows	• Step 3: the user does not confirm they wish to be placed on the waitlist and instead cancels their request. System returns the user to the reservation dashboard.

3.3.7 UC 7 - MODIFY RESERVATION

Use Case ID:	UC7
Use Case Name:	Modify Reservation
Created By:	Ryan Lee
Actor(s):	User
Goal/Actor Goals:	The user wishes to change the timeslot, room or equipment of their reservation
Description/Summary:	The user chooses to change their reservation's time, room and/or equipment. The system verifies that the new reservation or equipment is available. The system then modifies the user's reservation to the new time and room and updates the registry
Preconditions:	• The user is logged into the system • The user was able to view their own reservations (UC4) • The user has at least one reservation • The room is not locked
Post-conditions:	• The user's reservation is changed is updated to the appropriate room, timeslot and/or equipment.
Minimum Guarantee:	The user's reservation remains unchanged
Basic Flow:	1. On the list of the user's reservations, user selects the "modify" option on one of them 2. System prompts the user to select a new room, start time, end time and equipment change 3. User selects a new room, start time and end time 4. User confirms their selection 5. System modifies the time of the user's reservation to match the new room, start, end time and equipment change 6. System displays a "success" message to the user
Alternative Flows	• Step 4: the user does not confirm their request to modify their reservation. System returns user to their list of reservations, which remains unchanged. • Step 3: user's new reservation exceeds the maximum of 3 hours. System displays an error and prompts the user to choose a new time period. • Step 3: user's new reservation includes one or more timeslots that are already reserved. In this case, the flow switches to that of "add to waitlist" (UC6). • Step 3: User's new reservation includes equipment that is already being used at the time. In this case, the flow switches to that of "add to waitlist" (UC6).

3.3.8 UC 8 - CANCEL RESERVATION

Use Case ID:	UC8
Use Case Name:	Cancel Reservation
Created By:	Ryan Lee
Actor(s):	User
Goal/Actor Goals:	The user wishes to cancel one of their reservations.
Description/Summary:	This use case allows the user to have full control over the reservations that they have created. It gives them the right to cancel their existing reservation.
Preconditions:	• The user is already logged in • The user was able to view their own reservations (UC4) • The user has at least one reservation • The cancellation request is made before the start time of the reserved time slot • There is no other user performing modification/cancellation/creation action on the same room at the same moment.
Post-conditions:	• The user has one less reservation after each successful cancellation request • If there are not capstone students, the user who is on the top of the waiting list of that room at that time slot reserves the room automatically, otherwise the capstone student on top of the waiting list gets the room • In case of empty waiting list, the system indicates to viewing users that the room is now available.
Minimum Guarantee:	The reservation is not cancelled and the system maintains its previous internal state
Basic Flow:	1. The user selects to view their reservations from the reservation dashboard 2. The system displays a list of all the user's reservations 3. The system indicates to the user what actions each reservation has 4. The user requests to cancel one of their reservations 5. The system asks the user to confirm the cancellation request 6. The user confirms the request 7. The system displays an updated version of the user's reservation list 8. The system displays a "success" message
Alternative Flows	• Step 6: the user does not confirm the cancellation and instead, cancels the request. The system returns the user to the list of their reservations.

3.4 NON-FUNCTIONAL REQUIREMENTS

- Only a single write operation (create, modify, cancel) can occur at any given time.
- Multiple view operations can occur simultaneously.
- System must display safety, liveness and fairness.
    - Only one user may access a room at any given time in order to create, modify or cancel a reservation
- System must be updated at real time.
- System must have high maintainability (easily identifiable errors and inefficient areas).
- System must be reliable (no crashes).
- System must be reusable.
- System must have quick response times.
    - Must create proper database indexing (need good ER model)
- System must run efficiently (as low load times as possible, given above circumstances).
- System must be portable between operating systems (Windows, Mac, Linux) and to mobile devices (iOS, Android).

3.4.1 RELIABILITY

- Unless the web application is closed for maintenance, users should be able to access the system at any given time.

3.4.2 USABILITY

- Reservations in schedule will be displayed in color.
- The user will be able to select reservations and be waitlisted through pop out menus.
- The system must centralize the functionalities

3.4.3 SECURITY

- The user cannot perform any functions if they are not logged in.
- Only a single user can make modifications to a given schedule slot at once. This means that a student cannot try altering their schedule while an administrator is making changes to it or while another logged in user is using this time slot.
- The system should be immune to security attacks such as SQL injections

3.4.4 DESIGN CONSTRAINTS

- Object Oriented Languages must be used to implement the system
- Relational Databases must be used to persist the data
- System components must be reusable in different systems.

4. ANALYSIS MODELS

4.1 SYSTEM SEQUENCE DIAGRAMS

SSDs describe the interactions between the user and the system for different scenarios of use. They define the behaviour of the system when viewed as a black box.

4.1.1 SSD - LOGIN

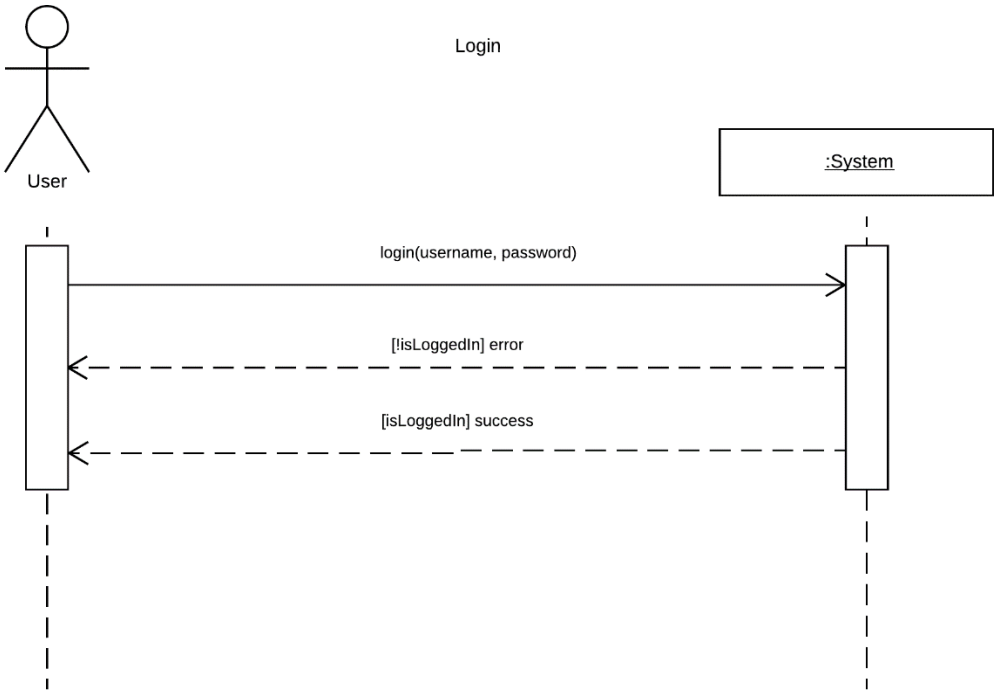

Figure 2: SSD for Login

4.1.2 SSD – LOGOUT

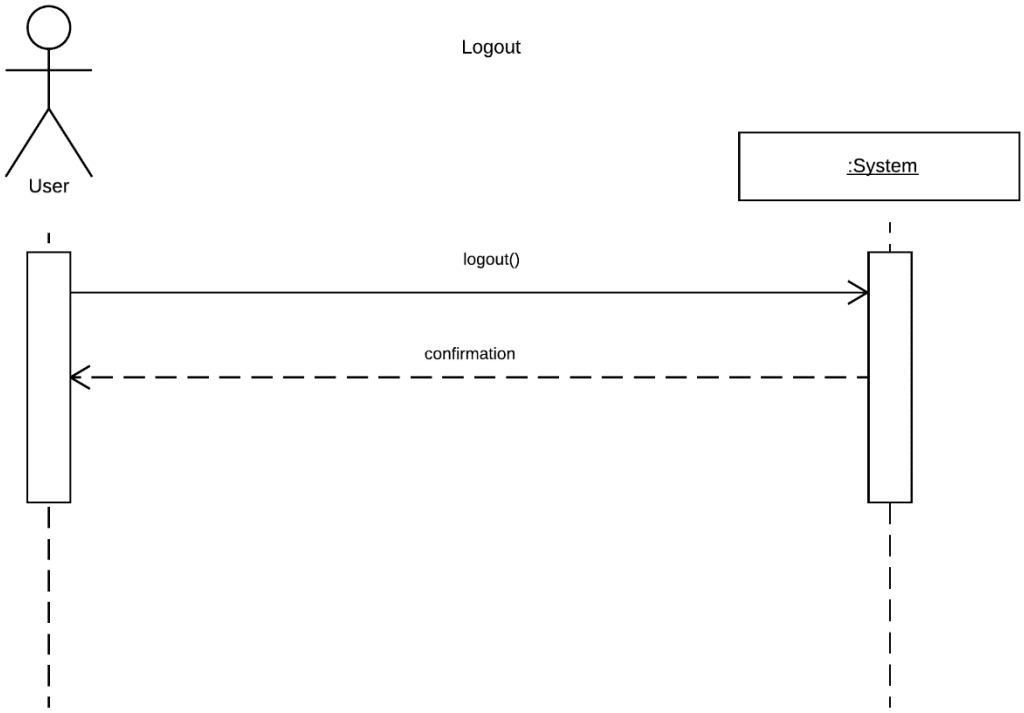

Figure 3: SSD for Logout

4.1.3 SSD - ACCESS REGISTRY

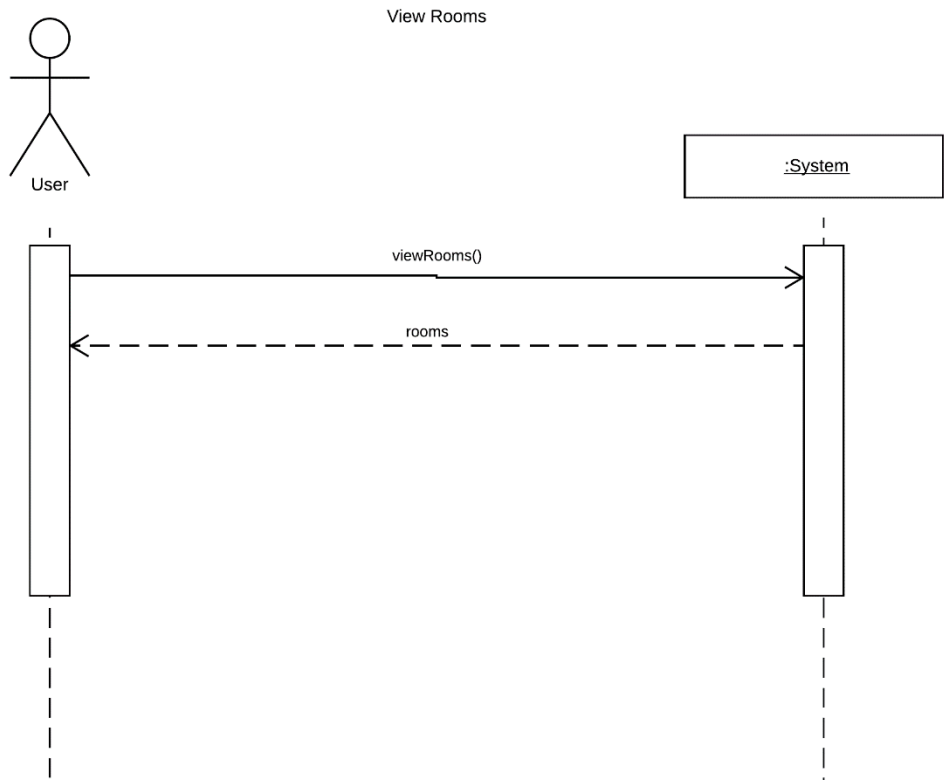

Figure 4: SSD for Access Registry

4.1.4 SSD - VIEW RESERVATIONS

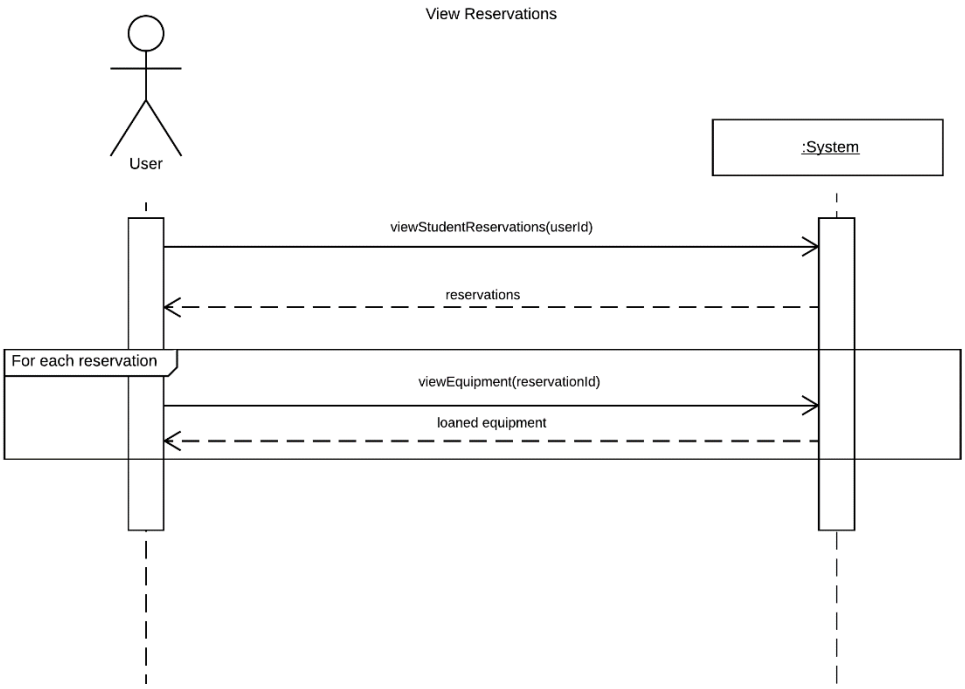

Figure 5: SSD for View Reservations

4.1.5 SSD - RESERVE ROOM

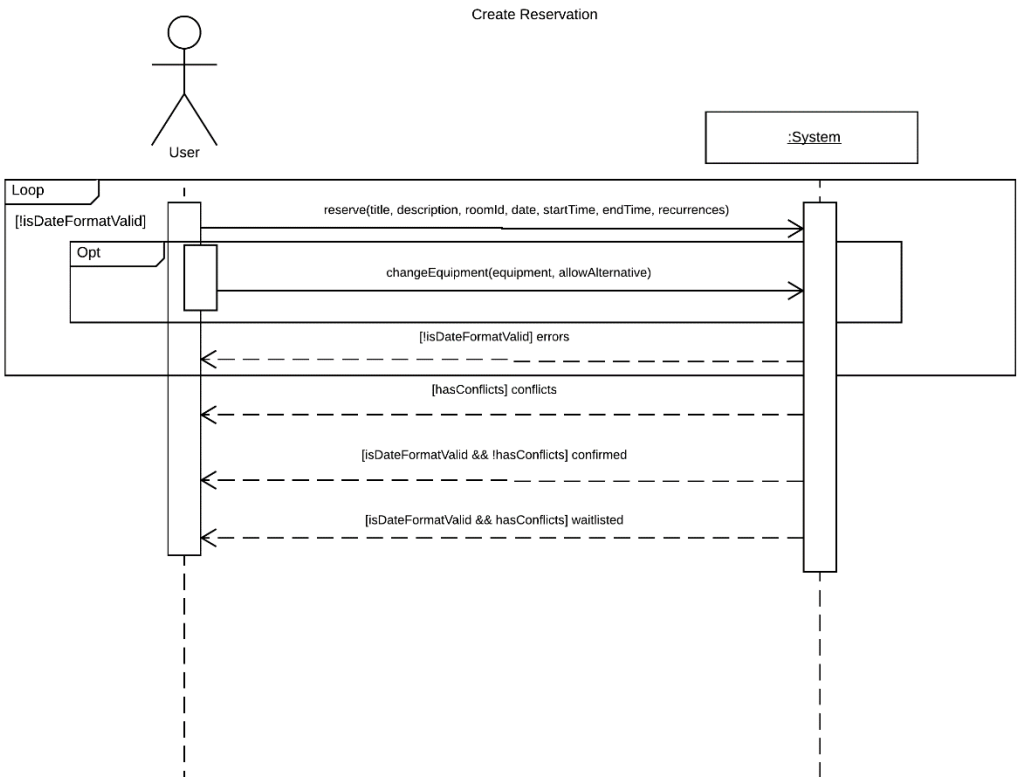

Figure 6: SSD for Reserve Room

4.1.6 SSD - MODIFY RESERVATION

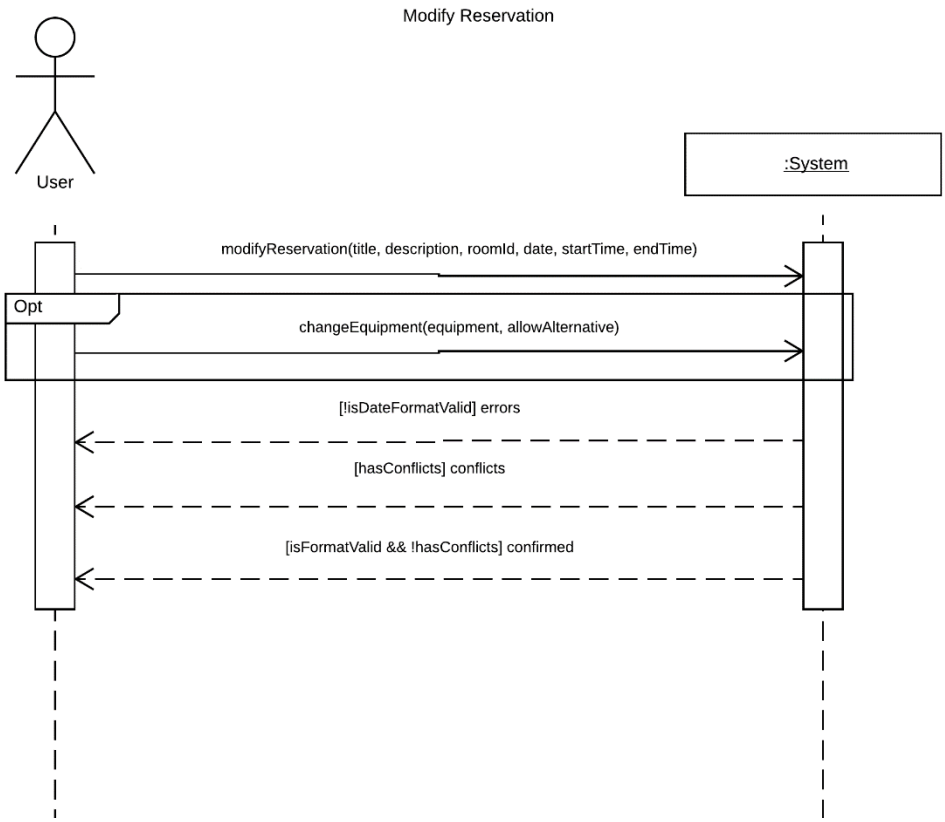

Figure 7: SSD for Modify Reservation

4.1.7 SSD - CANCEL RESERVATION

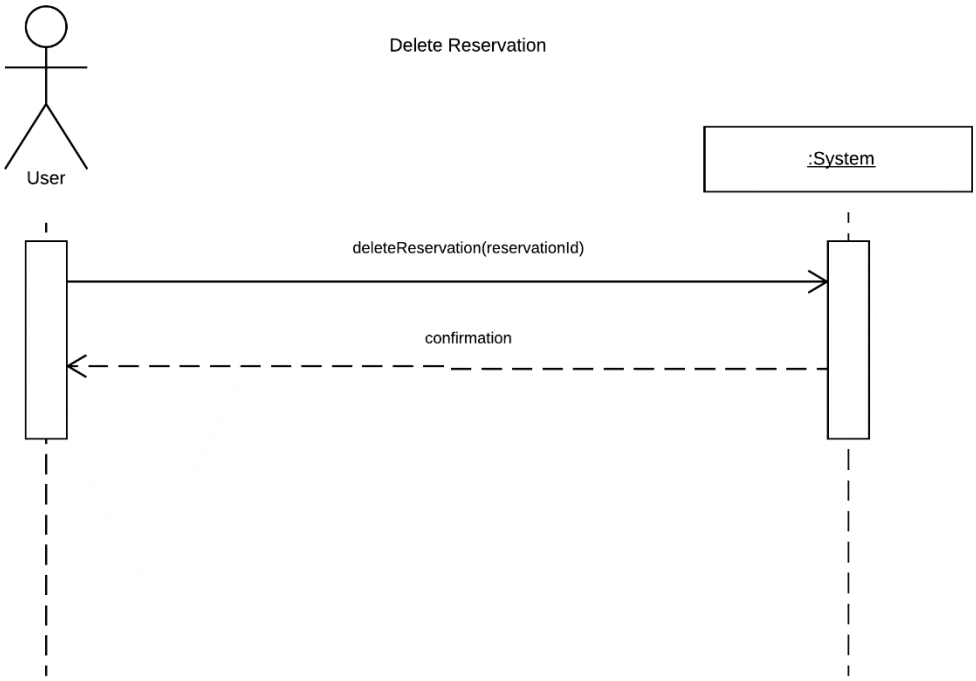

Figure 8: SSD for Cancel Reservation

4.2 LIST OF SYSTEM OPERATIONS

The list below provides a list of operations offered by the public system interface. It represents all the methods that are accessible by the client in order to receive a set of services meeting the system specifications.

System
login(username, password)
logout()
viewRooms()
viewStudentReservations(userId)
viewEquipment(reservationId)
reserve(title, description, roomId, date, startTime, endTime, recurrences)
modifyReservation(title, description, roomId, date, startTime, endTime)
deleteReservation(reservationId)

4.3 OPERATION CONTRACTS

Contracts for each system operation are provided below, with the exception of login and logout.

4.3.1 VIEWROOMS

Contract 1	
Created By:	Kevin Yasmine
Operation:	viewRooms()
Cross-reference:	UC3 - Access Registry
Pre-conditions	• User is authenticated • There is an Instance of RoomDirectory roomDir
Post-conditions	• A collection of Room instances is associated with roomDir

4.3.2 VIEWSTUDENTRESERVATIONS

Contract 2	
Created By:	Kevin Yasmine
Operation:	viewStudentReservations(userId)
Cross-reference:	UC4 - View Reservations
Pre-conditions	• User is authenticated • There is an instance of ReservationRegistry resReg • A User instance with id equal to userId exists
Post-conditions	• A collection of Reservation instances belonging to User with id userid is associated with resReg

4.3.3 VIEWEQUIPMENT

Contract 3	
Created By:	Kevin Yasmine
Operation:	viewEquipment(reservationId)
Cross-reference:	UC4 - View Reservations
Pre-conditions	• User is authenticated • There is an instance of Reservation res with id equal to reservationId
Post-conditions	• A collection of LoanedEquipment instances is associated with res

4.3.4 RESERVE

Contract 4	
Created By:	Kevin Yasmine
Operation:	reserve(title, description, roomId, date, startTime, endTime, recurrences)
Cross-reference:	UC5 - Reserve Room
Pre-conditions	• User is authenticated • There is a Room instance room with id equal to roomId • There is an instance of CreateReservationSession underway
Post-conditions	• One or more instances of Reservation were created with the title, description, date, start time and end time equal to the supplied parameters • The number of Reservation instances created is equal to recurrences • Each Reservation instance is associated with room

4.3.5 MODIFYRESERVATION

Contract 5	
Created By:	Kevin Yasmine
Operation:	modifyReservation(title, description, roomId, date, startTime, endTime)
Cross-reference:	UC7 - Modify Reservation
Pre-conditions	• User is authenticated • There exists an instance of Reservation res • There is a Room instance room with id equal to roomId • There is an instance of ModifyReservationSession underway
Post-conditions	• res.title was set to title • res.description was set to description • res.date was set to date • res.startTime was set to startTime • res.endTime was set to endTime • res is associated with room

4.3.6 DELETERESERVATION

Contract 6	
Created By:	Kevin Yasmine
Operation:	deleteReservation(reservationId)
Cross-reference:	UC8 - Cancel Reservation
Pre-conditions	• User is authenticated • There is an instance of Reservation res with id equal to reservationId • There is an instance of DeleteReservationSession underway
Post-conditions	• The Reservation instance res was deleted

4.4 DOMAIN MODEL

The domain model is a diagram that gives a visual representation of the core concepts and an overview of them in the system in question. The main purpose of this diagram is to list objects of interest and their associations with one another.

4.4.1 CORE PACKAGE

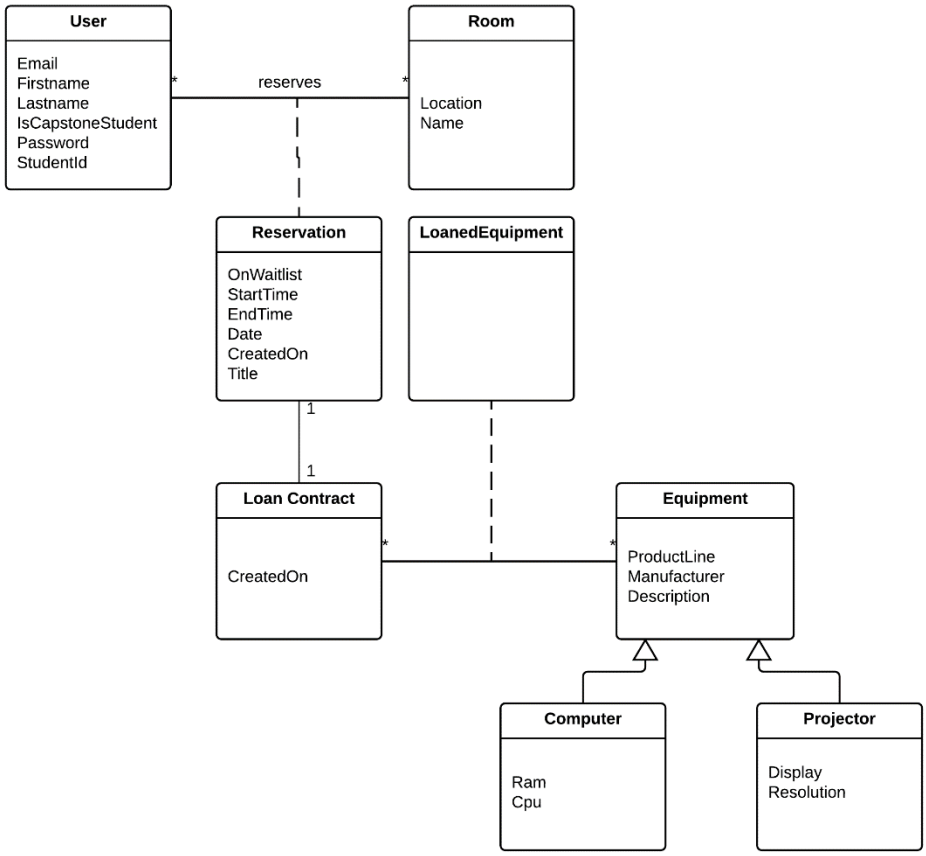

Figure 9: Domain Model Core Package

4.4.1.1 USER

The user is an ENCS student who accesses the system in order to view, create, modify or delete reservations. Each user has an email (username) and password which they can use to access the system. The user may be a Capstone student, which gives them priority over non-Capstone students on the waiting list. Each user can reserve a maximum of 3 hours per week and cannot create a repeated reservation for more than 3 weeks.

4.4.1.2 RESERVATION

A reservation is an association between a user, a room, date, start time, and end time. It indicates that the user has made a request to claim this room on a specific date and time slot. If the reservation conflicts with another one at the same time, the reservation is placed on a waitlist. If there is no conflict, it is made active and will appear on the public calendar.

4.4.1.3 ROOM

There are 5 rooms in this system that are each available to reserve on a 24-hour basis.

4.4.1.4 EQUIPMENT

There is a limited list of equipment available for students to request for the room they wish to reserve. If the requested equipment is not available, the reservation is waitlisted regardless of whether or not the room is available at the requested time slot.

4.4.1.5 COMPUTER

Laptop computers are a type of equipment available for students to request.

4.4.1.6 PROJECTOR

Projectors are a type of equipment available for students to request.

4.4.1.7 LOAN CONTRACT

A loan contract is associated with a reservation. The presence of one indicates that one or more equipment has been requested for the associated reservation and the user who made it. This was done to decouple reservations from needing to know about loan contracts.

4.4.1.8 LOANED EQUIPMENT

Loaned equipment is an association between a loan contract and equipment, representing the individual piece of rented equipment for a given loan contract. Multiple equipment can be associated with a single loan contract (and thus a single reservation).

4.4.2 RESERVATION SESSIONS PACKAGE

Figure 10: Domain Model Reservation Sessions Package

4.4.2.1 CREATE RESERVATION SESSION

This is a user session in which the user wishes to reserve a room at a given time. It checks the reservation registry to see if the time slot the user is trying to book is currently available. It also checks the equipment catalog to verify that the requested equipment, if any, is available to rent. Upon successful reservation creation, a new reservation is added to the reservation registry.

4.4.2.2 MODIFY RESERVATION SESSION

This is a user session in which the user wishes to modify one of their existing reservations. They are given the ability to change the date, start time, end time, title, description, room, and requested equipment. After a successful modification, the given reservation is updated in the reservation registry. If new equipment was requested, or loaned equipment removed, the equipment catalog is updated and loaned equipment records are created/removed accordingly.

4.4.2.3 DELETE RESERVATION SESSION

This is a user session in which the user may choose to cancel one or more of their existing reservations. The reservation is removed from the reservation registry, and any wait-listed reservations that can be made active as a result of the delete are automatically added to the schedule (based on wait list priority).