Software Requirements Specifications
CARSA Registration System Enhancement

Group 5

Thursday October 29th, 2015

Table of Contents

Revision History

Name	Date	Reason for Changes	Version
Document Creation	2015-10-29		1.0

Section 1: Introduction

1.1 Purpose

This document describes the requirements that have been gathered for the CRSE project, and other related design artifacts. The objective of this project is to address the delays caused by the large influx of new members seeking to register at CARSA during the first two weeks of each semester.

1.2 Project Scope

This project is an enhancement to CARSA's current membership registration system. The primary objective is to improve the front desk's ability to meet the increased membership demand that occurs during the first two weeks of each academic term. This in turn will address the delays that occur as a result of overall demand exceeding the speed at which these demands can be met.

This project will produce a solution that will comply with the service requirements that exist within the currently operational system. The improvements will be focused on processes that physically occur in or around CARSA's front desk lobby area.

1.3 Glossary of Terms

CARSA: Center for Athletics, Recreation, and Special Abilities

CRSE: CARSA Registration System Enhancement

COTS: Commercial Off-the-Shelf

UVic Member: A CARSA member who also has a UVic account/OneCard.

Community Member: A CARSA member who has no UVic account/OneCard.

Membership: The information stored by the CARSA systems about a member. This information includes the member's name, address, current pass, and expiry information.

Basic Pass: A membership type that provides access to the sports facilities of the gym, but not the weight room or the climbing wall.

Fit Pass: A membership type that provides all the benefits of the Basic Pass, as well as access to the weight room.

All-In Pass: A membership type that provides all the benefits of the Basic and Fit Pass as well as access to the climbing wall.

Secondary Pass: Any short term (less than a month) pass that is not covered by the Basic, Fit, or All-In pass categories. This category includes day passes, month long passes, guest passes and McKinnon Gym passes.

1.4 References

[1] University of Victoria. *Accessibility*, [Online].
Available: http://www.uvic.ca/facilities/service/accessibility/index.php

[2] The Canadian Legal Information Institute. *Personal Information Protection and Electronic Documents Act, SC 2000, c 5,* [Online].
Available: http://canlii.ca/t/52hmg

[3] University of Victoria. *Privacy & access to information policies*, [Online].
Available:
https://www.uvic.ca/universitysecretary/privacy/policiesandguidelines/

1.5 Overview

This document includes five sections and three appendices. The sections that follow are separated into three general topics: description, features and requirements.

Section 2 describes the system's high level features, introduces the known stakeholders and their interactions with the system, and describes assumptions that were made regarding CARSA's existing system.

Section 3 describes the functional requirements that exist in the current system and that must be maintained in the proposed solution.

Section 4 details the features of the proposed solution and their associated non-functional requirements.

Section 5 outlines additional requirements, including various infrastructure, quality, privacy and security guidelines.

Appendices A and B contain models and use cases that represent the current operations within the CARSA registration system. They highlight both important relationships among different entities and user and stakeholder interaction with the system.

Appendix C outlines issues that have yet to be addressed or resolved. These issues include matters such as pending decisions, TBDs, and conflicts that have yet to be addressed.

Section 2: Overall Description

2.1 Product Perspective

The planned system would be a modification of CARSA's existing registration and membership process. This system will either extend or enhance the ability of the existing CARSA registration system to handle increased user traffic during peak intervals in the school year.

2.2 Product Features

Improved Customer Experience: The layout and design of the current registration process will be revamped to provide a more intuitive, fast and friendly customer experience.

Easy Implementation: The new solution will not require significant software changes or new hardware, it will reorganize existing methods in order to improve efficiency while limiting overhead. Existing employees should have no trouble learning the new system.

2.3 User Classes and Characteristics

The primary users of the system can be divided into two groups. CARSA staff are the primary operators and the intermediary between clients and the backend software system. CARSA customers primarily interact with the system through front desk staff in the building lobby.

2.3.1 CARSA Staff

In the context of the registration system, CARSA staff includes all the employees who work at the front desk. These staff are the first point of interaction for users looking to register at CARSA, and spend the most time out of any user group interacting with the backend system. All front desk staff are trained in operating the registration and membership system, and thus are presumed to have technical expertise with the system. CARSA staff are also responsible for providing membership information to customers.

2.3.2 CARSA Customers

Customers are divided into members and non-members. Members include UVic students, staff and faculty, as well as community members who have applied for a membership at CARSA. Non-members encompass all other users of the CARSA facilities. Even within the three sub-categories, customer usage of the system varies greatly from daily to intermittent, so it is not possible to describe a general usage pattern.

Following is a breakdown of the three main customer groups: UVic members, community members and non-member customers.

- **UVic members:** UVic students and staff who register with their ONECards and use CARSA facilities. All of these members have ONECards for identification, and many are automatically eligible for memberships through mandatory recreation fees.

- **Community members:** Customers from the community who are not part of the UVic systems. These clients are issued dedicated member cards upon registration.

- **Non-member customers:** Customers who have not applied for any form of membership fall into this category. To enter the facilities requires either a secondary pass or some otherwise specified reason such as supervising a minor. These customers generally have very limited interaction with the system.

2.4 Design and Implementation Constraints

2.4.1 Security

The new system must adhere to UVic's accessibility standards as detailed by [1]. Any modifications that interact with user information must abide by both the BC [2] and UVic [3] information privacy guidelines.

2.4.2 Modification of Existing System

The current CARSA registration system makes heavy use of a proprietary recreation management system called CLASS. Since CLASS is a COTS system, there will be no modification of existing software within the scope of this project.

2.5 Assumptions and dependencies

2.5.1 Assumptions

Our project assumes that UVic will not be replacing the underlying technical systems used by CARSA. Software changes would be likely to alter the workflow of front desk staff and may invalidate any proposed enhancements.

2.5.2 Dependencies

As this project builds off of the existing system, it is necessary to maintain the existing external dependencies:

- The UVIC account system
- The OneCard system
- CLASS and its associated functions
- Physical infrastructure provided for registration by CARSA (i.e. the front desk kiosks and terminals)

Section 3: Existing Functional Requirements

3.1 Description and Priority

These are functional requirements of the existing system that will be preserved in the new system's implementation. **Priority: High**

3.2 Requirements

REQ-01: Front desk kiosks must connect to UVic systems to retrieve UVic staff and student account information when creating their CARSA membership.

Backward Traceability: The existing kiosks have access to UVic systems for registration, making the registration process must faster for UVic members.

Forward Traceability: This requirement can be verified by checking whether a kiosk can retrieve account information from the UVic systems/ONECard when creating an account.

REQ-02: Front desk kiosks must not be able to modify/delete UVic accounts

Backward Traceability: The existing kiosks are limited to reading UVic account information to prevent interference in UVic's systems.

Forward Traceability: This requirement can be verified by attempting to delete or modify a UVic account from within the CARSA software system.

REQ-03: UVic members must have their UVic ID in their CARSA account, while community members have no UVic ID in their account.

Backward Traceability: The current system stores UVic member's UVic ID in their CARSA account, which allows customers identified as UVic members to qualify for lower membership fees than members of the general community.

Forward Traceability: This requirement can be verified by checking the account of a UVic member to verify that the UVic ID is present and by checking the account of a community member to verify that the UVic ID is absent.

REQ-04: The system must be able to register for a 4-month, 8-month or a full year pass at CARSA.

> **Backward Traceability:** The current system supports 4-month increments for the duration of purchased memberships, which allows members to make less of a commitment when purchasing their membership.
>
> **Forward Traceability:** This requirement can be verified by checking the possible membership options in the interface and through verification testing of the expiration date functionality.

REQ-05: The system must enable customers to register for any of the three membership tiers.

> **Backward Traceability:** The current system supports a three tier membership system that allows customers to opt out of paying for facilities they do not intend to use.
>
> **Forward Traceability:** This requirement can be verified by creating a new membership with each of the three membership tiers and testing the card on the card readers for each of the facilities.

REQ-06: The system must support the purchase of secondary passes.

> **Backward Traceability:** The current system allows customers to buy short term passes that do not require membership, which gives people a way to use gym facilities without committing to a longer membership.
>
> **Forward Traceability:** This requirement can be verified by checking the availability of non member passes at kiosks.

REQ-07: Front desk kiosks must be able to accept credit cards, debit cards, and cash payments.

> **Backward Traceability:** The current system allows customers to use cash, credit or debit. These methods of payment are all standard for most institutions.
>
> **Forward Traceability:** This requirement can be verified by ensuring that kiosks have a secure register for cash and a card reader for debit and credit cards.

REQ-10: Front desk kiosks need to be able to connect to the CLASS system in order to register new members and search for existing members.

> **Backward Traceability:** The current system uses the CLASS system to handle member accounts.
>
> **Forward Traceability:** This requirement can be verified by ensuring that kiosks have an active internet connection.

REQ-11: Users must sign a waiver against personal injury while using the facilities as part of the registration process.

> **Backward Traceability:** The waiver protects CARSA from liability if a member is injured while using the facilities.
>
> **Forward Traceability:** This requirement can be verified by ensuring that the account is not activated until the waiver has been signed.

REQ-12: Users must log on to gain access to CARSA systems.

> **Backward Traceability:** The current system has a login screen and requires an employee username and password. This prevents malicious use of CARSA systems and protects member privacy.
>
> **Forward Traceability:** This requirement can be verified by ensuring that there are no exploits by which a user could gain access to CARSA systems without valid login credentials.

REQ-13: The system must be able to search for members using any of the following parameters: first name, last name, student number or address.

> **Backward Traceability:** The current CARSA system includes support for searching with several different kinds of customer information. This is essential for retrieving membership information with limited data.
>
> **Forward Traceability:** This requirement can be verified by ensuring that the search field in the CLASS application accepts a first name, last name, address or UVic ID as search parameters.

Section 4: Features

4.1 Improved Customer Experience

4.1.1 Description and Priority

The new system will be easy for customers to use and understand, but also considerably faster than the existing solution, resulting in shorter lineups.
Priority: High

4.1.2 Requirements

REQ-CX-01: Information about CARSA memberships and passes should be available online.

Backward Traceability: Providing information about the available passes and memberships online helps customers decide what they will purchase before they come to CARSA, leading to a faster registration process.

Forward Traceability: This requirement can be verified by finding the information on the CARSA webpage.

REQ-CX-02: The system must meet UVic's accessibility standards[1].

Backward Traceability: As a UVic facility, CARSA must adhere to the university's accessibility guidelines.

Forward Traceability: This requirement can be verified by reviewing UVic's accessibility standards.

REQ-CX-03: The system must be active during CARSA's daily operational hours: 6:30AM - 11PM every day.

Backward Traceability: Any new system must not interfere with the regularly scheduled hours for CARSA

Forward Traceability: This requirement can be verified by checking staff numbers and availability against the work schedule.

REQ-CX-04: The system must be capable of handling customer inquiries at the front desk.

> **Backward Traceability:** The front desk is the only avenue for customers to get help from a human about CARSA facilities and services.
>
> **Forward Traceability:** This requirement can be verified by ensuring that there is a protocol in place for handling customer inquiries.

REQ-CX-05: The system must allow users to immediately (in less than one second) identify open kiosks.

> **Backward Traceability:** Making open kiosks obvious to customers reduces the time spent in line.
>
> **Forward Traceability:** This requirement can be verified by observing the visibility of the kiosks and clerks from the head of the line.

REQ-CX-06: The system needs to provide users with enough resources to assist them in choosing memberships. Resources are provided before and after the users reach the front desk.

> **Backward Traceability:** The current system provides users with some information of CARSA registration options. However, the users are not finding these information to be sufficient.
>
> **Forward Traceability:** This requirement can be verified by providing the users with registration information while they are at the front desk, and also allow the users to receive information before they reach the front desk.

REQ-CX-07: No customer should have to wait in line for more than 15 minutes.

> **Backward Traceability:** Long waits in line lead to a negative customer experience.
>
> **Forward Traceability:** This requirement can be verified by timing the progress of customers through the line.

REQ-CX-08: The system must be able to register a new customer within five minutes of their arrival at the front desk

> **Backward Traceability:** Quick service is essential to customer satisfaction.
>
> **Forward Traceability:** This requirement can be verified by recording registration times.

REQ-CX-09: The line for registration should never exceed 20 people

Backward Traceability: Large numbers of customers in line can have a deterring effect on new customers trying to sign up for membership.

Forward Traceability: This requirement can be verified by observing and recording line sizes

REQ-CX-10: There must be some form of line management.

Backward Traceability: Currently during rush times there are crowd control stanchions that are placed leading up to the front desk. These are necessary to manage the large influx of customers.

Forward Traceability: This requirement can be verified by checking if stanchions are placed in the lobby.

4.2 Easy Implementation

4.2.1 Description and Priority

Training existing staff to use the new system will be quick and any increases in operational cost will be small and largely limited to the first two weeks of semester. **Priority: Medium**

4.1.2 Requirements

REQ-IMP-01: The system needs to be implemented within the confines of the CARSA lobby.

Backward Traceability: There is no other feasible space in which to conduct registration.

Forward Traceability: This requirement can be verified by checking the CARSA lobby.

REQ-IMP-02: It must be possible to train existing employees to use the new system within 10 minutes.

Backward Traceability: The system must not too complicated to easily explain to CARSA staff. Ideally a staff member could learn the new arrangement during their shift.

Forward Traceability: This requirement can be verified when training CARSA staff in the new system.

REQ-IMP-03: Front desk staff must have an immediately available means of contacting support in the event of technical problems.

Backward Traceability: The front desk is currently equipped with a phone that can be used to call UVic technical support services. This is essential for handling unexpected technical problems.

Forward Traceability: This requirement can be verified by ensuring that there is a phone at the front desk and that front desk staff know the appropriate support number and procedures.

Section 5: Other Requirements

REQ-MISC-01: The system must adhere to the BC Provincial Government's privacy laws[2]

Backward Traceability: In order to avoid legal and ethical risks, CARSA must strictly adhere to the BC Government's privacy laws.

Forward Traceability: This requirement must be verified by a BC provincial law expert.

REQ-MISC-02: The system must adhere to UVic's privacy guidelines[3]

Backward Traceability: As a UVic service, CARSA must adhere to UVic policy in matters of information privacy.

Forward Traceability: This requirement can be verified by referencing the UVic privacy guidelines.

REQ-MISC-03: Card payment options must be secured.

Backward Traceability: Secure card payments are essential to prevent any instances of identity theft or wire fraud from occurring at CARSA facilities.

Forward Traceability: This requirement can be verified by using only secured card readers and payment services.

REQ-MISC-04: There must be a contingency plan for technical failures in the computer system.

Backward Traceability: In the event of technical failure of computers at the CARSA facility, there should be a plan in place to handle registration requests and other inquiries.

Forward Traceability: This requirement can be verified by reviewing procedure documentation.

REQ-MISC-05: Front desk kiosks must be equipped with a UVic OneCard reader.

Backward Traceability: The existing kiosks are all equipped with OneCard readers to accelerate registration for UVic members.

Forward Traceability: This requirement can be verified by checking whether kiosks are equipped with OneCard readers.

REQ-MISC-06: Front desk kiosks must be able to print membership cards with photos for community members.

> **Backward Traceability:** Currently the front desk is equipped with a card printer and photo equipment for producing membership cards. This is essential for members who do not have a OneCard.
>
> **Forward Traceability:** This requirement can be verified by ensuring that the appropriate printing and photographic equipment is available at the front desk.

REQ-MISC-07: There must be a way for new members to sign the waiver on site.

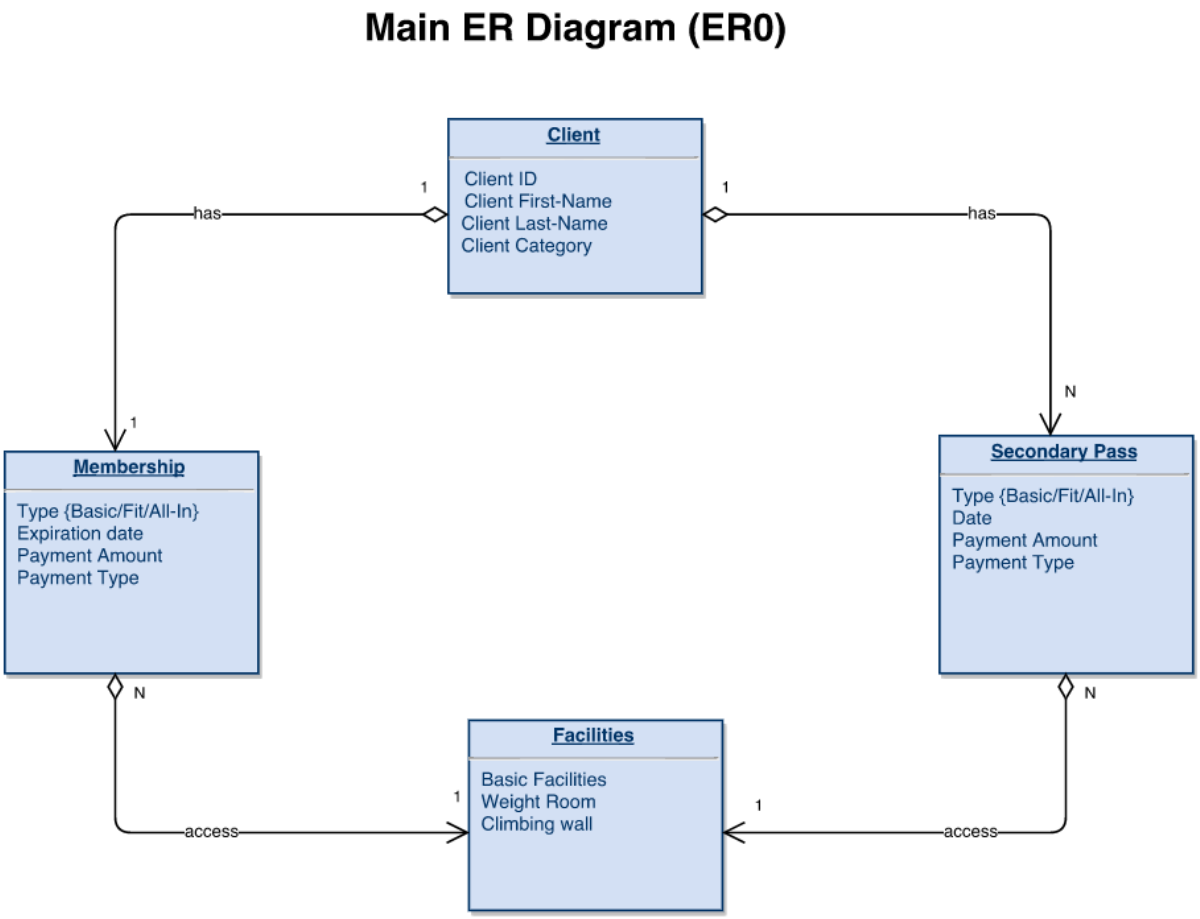
> **Backward Traceability:** There is currently a special self service kiosk for UVic members at the front desk that allows them to sign the digital waiver. Physical waivers for community members are provided on site.
>
> **Forward Traceability:** This requirement can be verified by ensuring that there is a way for both UVic and community members to sign the waiver available in the lobby.

REQ-MISC-08: Front desk kiosks must have an internet connection.

> **Backward Traceability:** The kiosks rely on an internet connection to function, as it allows the use of card readers and is necessary for use of the CLASS system.
>
> **Forward Traceability:** This requirement can be verified by ensuring that kiosks have internet access.

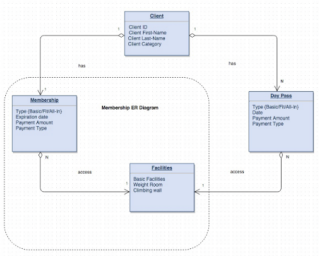
Appendix A: Analysis Models

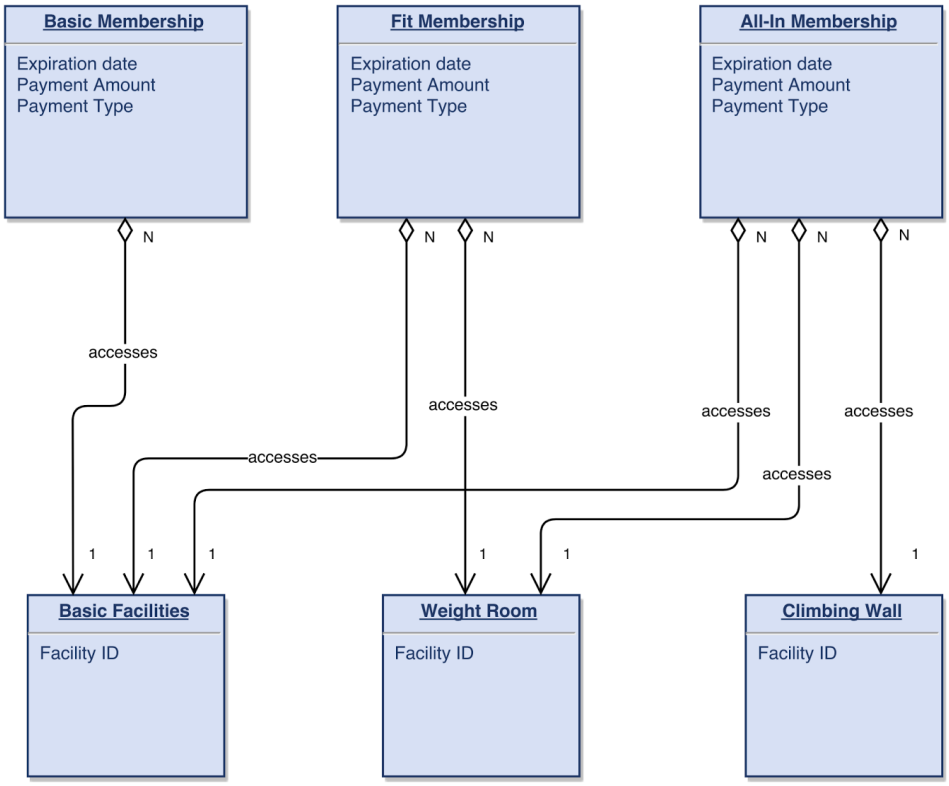
Section A1: ER Diagrams



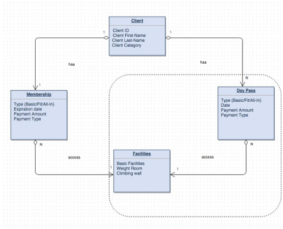
Memberships (ER1)

ER1 focuses on the relationships between the Membership and Facilities entities from ER0.

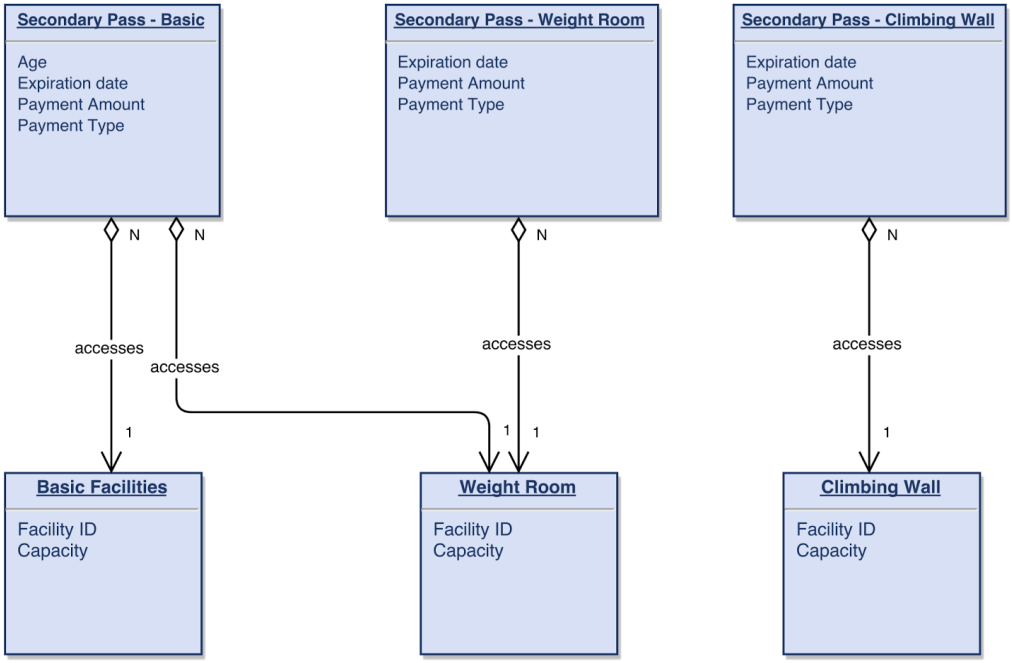




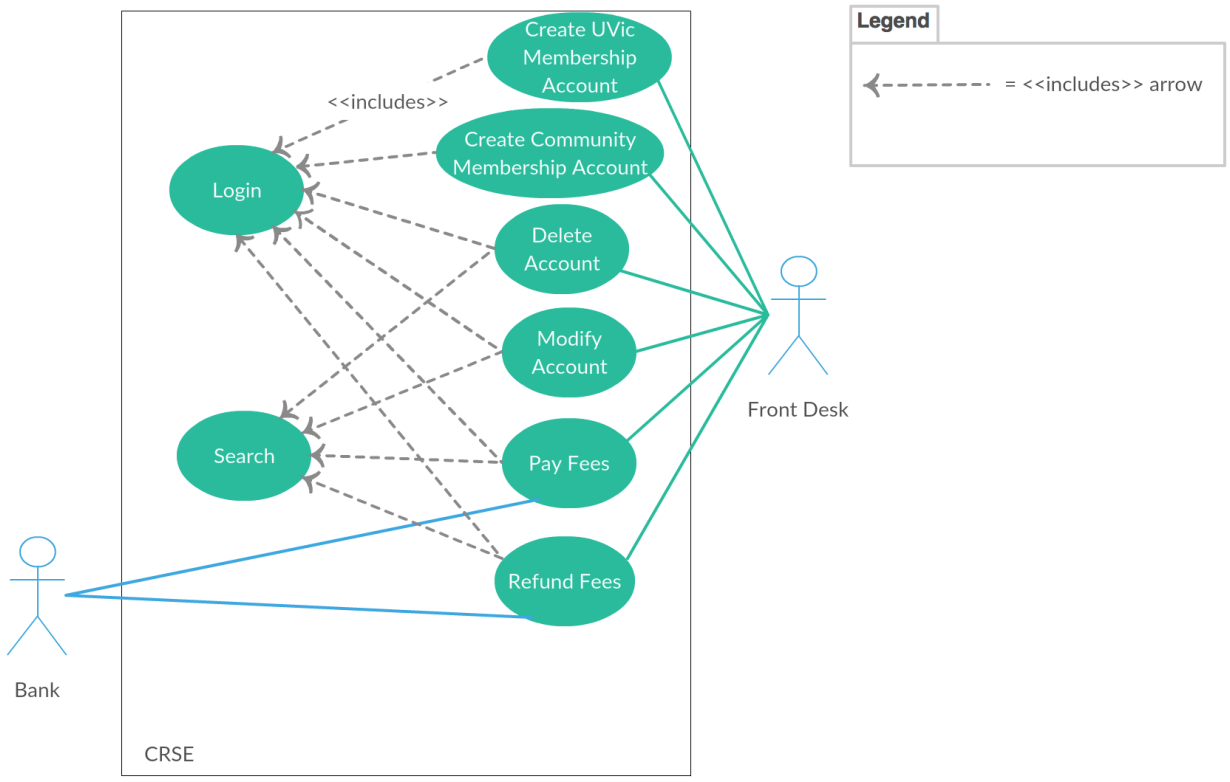
Secondary Passes (ER2)



ER2 focuses on the relationships between the Secondary Pass and Facilities entities from ER0.



Section A2: Use Case Diagram



Appendix B: Use Cases

Use Case 1: Login

Description

This use case describes how a Front Desk actor logs in to the system.

Actors

Front Desk

Pre-Conditions

The Front Desk must have an active account within the system including a username, a password, and access to the front desk computer terminals.

Main Flow

1. The use case begins when the Front Desk turns the front desk computer terminal on.
2. The system prompts for a user-name and password.
3. **<Enter Credentials>** The Front Desk enters their user-name and password.
4. **<Authentication>** The system authenticates the actor.
5. The system displays the administrative menu.
6. The use case ends.

Post-Conditions

The Front Desk has access to the system's main view.

Alternative Flows

A. At **<Enter Credentials>**, if any invalid parameters are entered in the input fields, then
   1. The system colours the background of the input sections red.
   2. The system displays "Inputs are invalid, please try again."

Return to **<Enter Credentials>**.

B. At **<Authentication>**, if the Front Desk entered an incorrect username or password, then
   1. The system displays a message saying "User-name and password do not match, please try again."

Return to **<Enter Credentials>**.

Use Case 2: Search

Description

This use case describes how a Front Desk actor searches for accounts in the system.

Actors

Front Desk

Pre-Conditions

The actor must logged in to the administrative menu of the system.

Main Flow

1. The use case begins when the Front Desk chooses the search menu option.
2. The system displays the search view.
3. **<Enter Query>** The Front Desk enters any of the following information if it is available: first name, last name, UVic ID number, address.
4. **<Submit Query>** The Front Desk submits the query to the search interface provided.
5. The interface returns the search results.
6. The system displays the search results view.
7. The Front Desk selects an account from those displayed.
8. The system displays the account details view of the selected account.
9. The use case ends.

Post-Conditions

The actor has access to the account details view of the account matching the search query.

Alternative Flows

A. At **<Enter Query>**, if any invalid parameters are entered in the input fields, then
   1. The system colours the background of the input sections red.
   2. The system displays "Inputs are invalid, please try again."

Return to **<Enter Query>**.

B. At **<Submit Query>**, if the Front Desk entered parameters which do not match any account, then
   1. The system displays a message saying "No accounts were found with <parameter name> = <input parameter>."

Return to **<Submit Query>**.

Use Case 3: Create Community Membership Account

Description

This use case describes how a Front Desk actor creates a new Community account in the system.

Actors

Front Desk

Pre-Conditions

The actor must logged in to the administrative view of the system; all input parameters to an account must be available, including: first and last name of the future account holder, a selected available account type, and a selected available membership level; there must be no accounts associated with the exact parameters list being used; the actor must have seen valid identification confirming both the full name and the selected account type to be created as true.

Main Flow

1. The use case begins when the Front Desk chooses the create account menu option.
2. The system displays the create account screen.
3. **<Enter Data>** The Front Desk enters the information for the account being created, including the new member's full name and address
4. **<Submit Data>** The Front Desk submits the data to the system.
5. **<Generate Account>** The system generates a new account with the corresponding data inserted, along with the current date listed as the creation date.
6. The system displays the account details screen for the new account.
7. A picture is taken of the new community member and a card is printed out for their account.
8. The use case ends.

Post-Conditions

The system has a new inactive account corresponding to the submitted data.

Alternative Flows

A. At **<Enter Data>**, if any invalid parameters are entered in the input fields, then
   1. The system colours the background of the input sections red.
   2. The system displays "Inputs are invalid, please try again."

Return to **<Enter Data>**.

B. At **<Generate Account>**, if the entered parameters do not result in conflicting types, or are themselves invalid, then
   1. The system displays a message saying "Account not created: non-matching or invalid inputs submitted. Please try again."

Return to **<Submit Data>**.

Use Case 4: Create UVic Membership Account

Description

This use case describes how a Front Desk actor creates a new UVic account in the system.

Actors

Front Desk

Pre-Conditions

The actor must logged in to the administrative view of the system; all input parameters to an account must be available, including: first and last name of the future account holder, a selected available account type, a selected available membership level, and a UVic ID number; there must be no accounts associated with the exact parameters list being used; the actor must have seen valid identification confirming both the full name and the selected account type to be created as true.

Main Flow

1. The use case begins when the Front Desk chooses the create account menu option.
2. The system displays the create account screen.
3. **<Enter Data>** The Front Desk enters all parameters for the account being created either manually, or automatically by swiping the OneCard.
4. **<Submit Data>** The Front Desk submits the data to the system.
5. **<Generate Account>** The system generates a new account with the corresponding data inserted, along with the current date listed as the creation date.
6. The system displays the account details screen for the new account.
7. The use case ends.

Post-Conditions

The system has a new inactive account corresponding to the submitted data.

Alternative Flows

A. At **<Enter Data>**, if any invalid parameters are entered in the input fields, then
   1. The system colours the background of the input sections red.
   2. The system displays "Inputs are invalid, please try again."

Return to **<Enter Data>**.

B. At **<Generate Account>**, if the entered parameters do not result in conflicting types, or are themselves invalid, then
   1. The system displays a message saying "Account not created: non-matching or invalid inputs submitted. Please try again."

Return to **<Submit Data>**.

Use Case 5: Delete Account

Description

This use case describes how a Front Desk actor deletes an account from the system.

Actors

Front Desk

Pre-Conditions

The actor must logged in to the administrative view of the system.

Main Flow

1. The use case begins when the Front Desk chooses the delete account menu option.
2. The system displays the delete account screen.
3. **<Search>** The Front Desk searches for the account to be deleted.
4. **<Delete>** The actor chooses the delete-account option.
5. **<Confirm>** The system displays a message saying "Delete this account?" with cancel and confirm options.
6. The Front Desk chooses the confirm option.
7. The system deletes the account from the database.
8. **<Completion>** The system confirms the account has been removed from the database.
9. The system displays a message saying "Account successfully deleted."
10. The use case ends.

Post-Conditions

The system has removed all data corresponding to the account in question.

Alternative Flows

A. At **<Confirm>**, if the actor chooses the cancel option, then
   1. The system displays "Operation canceled. Account not deleted."

Return to **<Delete>**.

B. At **<Completion>**, if the account was not removed from the database, then
   1. The system displays "Error: account was not deleted."

Return to **<Delete>**.

C. At <Search>, if no account is found, then end the use case.

Use Case 6: Modify Account

Description

This use case describes how a Front Desk actor modifies an account in the system.

Actors

Front Desk

Pre-Conditions

The actor must logged in to the administrative view of the system.

Main Flow

1. **<Search>** The use case begins when the Front Desk searches for the account to be deleted.
2. The Front Desk selects the account from the search results.
3. The system displays the account details screen.
4. The actor chooses the modify account option.
5. The system displays the modify account screen.
6. **<Modifying>** The actor modifies any fields in the account, replacing them with different valid inputs.
7. The Front Desk chooses the finish option.
8. The system saves the new account information into the database.
9. **<Success>** The system confirms the new information has been successfully saved.
10. The system displays a message saying "Account successfully updated."
11. The system returns to the account details screen.
12. The use case ends.

Post-Conditions

The system has updated all data corresponding to the account in question.

Alternative Flows

A. At **<Modifying>**, if the actor inputs invalid parameters, then
   1. The system colours the background of the input section red.
   2. The system displays "Invalid input, please correct."

Return to **<Modifying>**.

B. At **<Success>**, if the new account information was not saved in the database, then
   1. The system displays "Error: account update not saved."

Return to **<Modifying>**.

C. At **<Search>**, if no account is returned, end the use case.

Use Case 7: Pay Fees

Description

This use case describes how a Front Desk actor takes payments for a membership.

Actors

Front Desk, Bank

Pre-Conditions

The Front Desk must logged in to the administrative view of the system, and must be in the account details view of the account to be paid.

Main Flow

1. The use case begins when the Front Desk chooses the pay option from the account details screen.
2. **<Select>** The Front Desk selects either the debit, credit, or cash payment option.
3. The system calculates the corresponding fee for the account.
4. The system charges the fee to the payment hardware.
5. **<Charge>** The payment hardware submits data to the Bank.
6. **<Confirm>** The Bank confirms payment received.
7. The system marks fees as paid.
8. The system updates the account status to activated.
9. The system updates the account deactivation date to match the membership type.
10. The system displays a message “Print receipt?”
11. **<Receipt>** The actor selects yes.
12. The system prints a receipt.
13. The new member signs the waiver in order to gain access to the gym facilities.
14. The use case ends.

Post-Conditions

The Bank has received payment for the membership fee, the account status has been updated to active, and the account deactivation date has been updated to match the membership type.

Alternative Flows

A. At **<Confirm>**, if the Bank does not confirm, then
    1. The system displays "Payment failed: <Bank error>."

Return to **<Charge>**.

B. At **<Receipt>**, if the Front Desk selects no, then
    1. Skip to step 13.

C. At **<Select>**, if cash is selected, then
    1. The system displays the cash-in-cash-out view.
    2. The Front Desk submits the total cash provided.
    3. The system displays the change to be provided.
    4. The Front Desk inserts the received cash into the till and retrieves the displayed change amount.
    5. The Front Desk returns the change.
    6. The system displays the print receipt view.

Return to **<Receipt>**.

7.

Use Case 8: Refund Fees

Description

This use case describes how a Front Desk actor refunds payments for a membership.

Actors

Front Desk, Bank

Pre-Conditions

The Front Desk actor must logged in to the administrative menu of the CLASS system, and must be in the account details view of the account to be paid. If the card reader is used, it must have a connection to the internet in order to function.

Main Flow

1. The use case begins when the Front Desk chooses the refund option from the account details screen.
2. The Front Desk selects either the debit or credit refund option.
3. The system selects the value of the fee already paid.
4. The system sends the fee and refund code to the payment hardware.
5. **<Refund>** The payment hardware submits data to the Bank.
6. **<Confirm>** The Bank confirms refund received.
7. The system marks fees as refunded.
8. The system updates the account status to deactivated.
9. The system updates the account deactivation date to undetermined.
10. The system displays a message “Print receipt?”
11. **<Receipt>** The user selects yes.
12. The system prints a receipt.
13. The use case ends.

Post-Conditions

The Bank has received refunded the previously received fee for membership, the account status has been updated to deactivated, and the account deactivation date has been updated to undetermined.

Alternative Flows

A. At **<Confirm>**, if the Bank does not confirm, then
   1. The system displays “Payment failed: <Bank error>.”

Return to **<Refund>**.

B. At **<Receipt>**, if the Front Desk selects no, then
   1. End the use case.

Appendix C: Issues List

There are currently no issues that have yet to be addressed.