Business Modeling

* 1. Vision
* Project Proposition: Project is proposed by Dr. Kewal Dhariwal, Executive Director of ICCP to integrate two systems that are Back Office Management System and online Exam Management system and to build Online Exam Presentation System.
* Problems: Currently ICCP has two isolated systems that are Back Office Management System which is using C# Web Forms as frontend and Exam Management System which is using C # Angular MVC Core and Exams are currently presented offline.
* Proposed Solution: Proposed Solution would be to bring modifications in the existing ICCP website by integrating the Back-Office Management System and the Online Exam Management System. The complete system’s frontend will be coded in C # using Razor Pages and Angular MVC Core.
* Stakeholders:

|  |  |
| --- | --- |
| Stakeholder | Needs |
| * Dr. Kewal Dhariwal (Executive Director, ICCP) * Cindy Blaese (Director of Finance & Administration) * Kendra Gurke (Director Member Services) * ICCP Board of Directors * ICCP Certification Council * Student Team | * Integration of BOS and TMS. * Modifications in existing BOS. * Online Exam Presentation System. * Students can appear in online exams * Students can view their exam results online. |

* Feature List: The features that are requested by Client are as follows:
* Integration of BOS and TMS
* Exam Presentation System (Web Based)
* Modifications to BOS
* TMS Interface
* Web Service
* Password Reset
* Preferred Email field
* Actor Goal List

|  |  |
| --- | --- |
| Actor | Goals |
| Board Members, Executive committee | Maintain Inventory, Customers, Order, Credentials, Calendar, and Organization. |
| Potential Applicant | Ask information or apply for membership application |
| Candidate Member | Lookup and buy inventory materials |
| Certified Member | Qualify and receive credentials |
| Volunteers | Provide help |
| Trainers | Provide trainings online and onsite |
| Course Developer | Design syllabus for exam |
| Employees  (full time and contracted employees | Can handle issues related to website |
| Organization | Participating as partners or business association |
| Student | Can take online exam |

* 1. Business Case
     1. Business Bid
* Project Plan
* Phase Plan

Phase: Construction

|  |  |
| --- | --- |
| Major Milestone: | Initial Operational Capability |
| Start Date: | 03/25/2020 |
| End Date: | 04/10/2020 |
| Total Time(hours): | 150 |
| Total resources(roles): | 3 Use Case Specifiers,  3 System Analysts,  3 Use Case Engineers,  3 Component Engineers,  3 Test Engineers |

* Phase Plan

Phase: Transition

|  |  |
| --- | --- |
| Major Milestone: | Product Release |
| Start Date: | 04/10/2020 |
| End Date: | 04/22/2020 |
| Total Time(hours): | 108 |
| Total resources(roles): | 3 Use Case Specifiers,  3 System Analysts,  3 Use Case Engineers,  3 Component Engineers,  3 Test Engineers |

* Project Schedule
* Phase Plan

Phase: Construction

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Iteration | Minor Milestone | Start Date | End Date | Time(hours) | Resources (Roles) |
| 1 | Integration of BOS and TMS | 03/25/2020 | 04/10/2020 | 150 | 3 Use Case Specifiers,  3 System Analysts,  3 Use Case Engineers,  3 Component Engineers,  3 Test Engineers |
|  |  | Total |  | 150 | 3 Use Case Specifiers,  3 System Analysts,  3 Use Case Engineers,  3 Component Engineers,  3 Test Engineers |

* Phase Plan

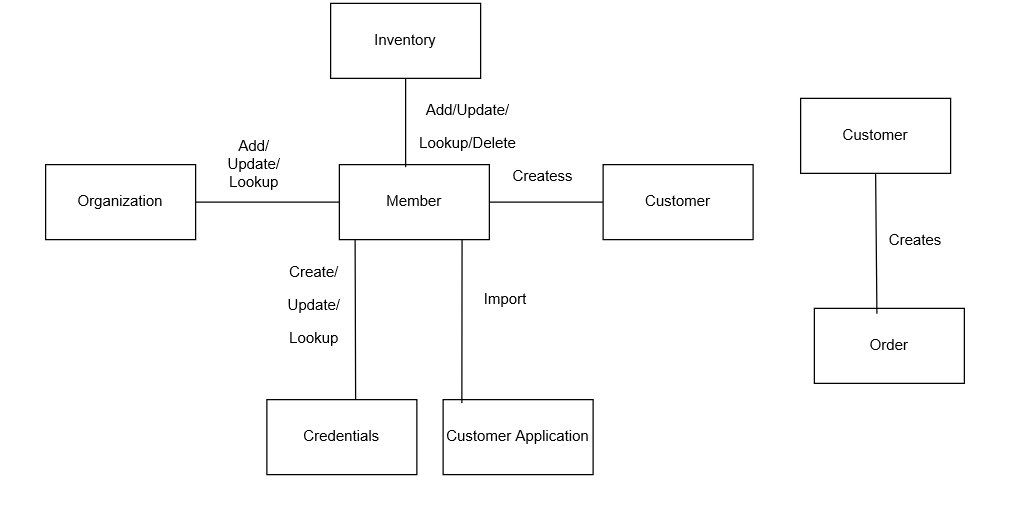
Phase: Transition

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Iteration | Minor Milestone | Start Date | End Date | Time(hours) | Resources (Roles) |
| 1 | Beta Release | 04/10/2020 | 04/16/2020 | 50 | 3 Use Case Specifiers,  3 System Analysts,  3 Use Case Engineers,  3 Component Engineers,  3 Test Engineers |
| 2 | Final Release | 04/16/2020 | 04/22/2020 | 58 | 3 Use Case Specifiers,  3 System Analysts,  3 Use Case Engineers,  3 Component Engineers,  3 Test Engineers |
|  |  | Total |  | 108 | 3 Roles:  3 Use Case Specifiers,  3 System Analysts,  3 Use Case Engineers,  3 Component Engineers,  3 Test Engineers |

* Economic Gains

|  |  |
| --- | --- |
| Stakeholders | Benefits |
| * Dr. Kewal Dhariwal (Executive Director, ICCP) * Cindy Blaese (Director of Finance & Administration) * Kendra Gurke (Director Member Services) * ICCP Board of Directors * ICCP Certification Council * Student Team | * The integration of BOS and TMS systems would provide ease of access to manage Back Office and Test Management data. * With this BOS system, it will be easy to use the interface in terms of menu selection. * Users can reset their passwords easily. * The business operations proving high business values such as online exams are automated. * Students can appear in exams anytime and can see their result reports. * Students can export their results. |

1.3 Domain Model



Analysis

2.1) Use Case: ImportCustomerApplication

* Brief Description: The purpose of this use case is to allow ICCP members to submit new application for Customer.
* Flow of Events:
* Basic Flow
* Use Case narrative:

Member wants to submit an application for new Customer.

For submitting Customer Application, Member may upload the application form and submits the Load Spreadsheet requests to the system. When completed, system displays the values in the form after reading the uploaded file. After reviewing the field of the form, Cindy submits the Submit Application request to the system. The System processes the request and add a new Customer. When completed, System returns the generated CustomerID for the new customer to Member.

For submitting new Customer Application, member provides the following details:

First Name:

Last Name:

Middle Name:

Organization Name:

Job Position:

Job Description:

Employment Date:

Education Degree:

Education Major:

Years Business Exp.:

Years IT Exp.:

Email Address:

Phone Number:

Street Address:

Unit Number:

Country:

Region/Province:

City:

Zip/Postal Code:

Use Home Email for ICCP Business:

Use Home Email for Prof. Communications:

For CBIP mailing use:

For TDWI mailing use:

For ICCP mailing use:

For related mailing:

Previously filed for CBIP Certification with TDWI?:

Previously filed for Certification with ICCP?:

Passed an ICCP exam and received a designation?:

Designation Received:

Disability Check:

TDWI Membership:

When completed, member submits the Submit Application request to the System. The System processes the request and add new Customer. When completed, System returns the generated CustomerID for the new customer to member.

* Primary scenario narrative:

Primary scenario: Cindy, ICCP member wants to submit a new customer application for Navjot Deol.

Cindy, ICCP member wants to submit an application for Navjot, new Customer.

For submitting Customer Application, Cindy may upload the application form and submits the Load Spreadsheet requests to the system. When completed, system displays the values in the form after reading the uploaded file. After reviewing the field of the form, Cindy submits the Submit Application request to the system. The System processes the request and add a new Customer. When completed, System returns the generated CustomerID for the new customer to Cindy.

For submitting new Customer Application, Cindy provides the following details:

First Name: Navjot

Last Name: Deol

Middle Name: Kaur

Organization Name: NAIT

Job Position: Analyst

Job Description: Technical Solutions

Employment Date: 2020-05-04

Education Degree: Some College (selected from drop down list)

Education Major: Business (selected from drop down list)

Years Business Exp.: 2-4 Years (selected from drop down list)

Years IT Exp.: 2-4 Years (selected from drop down list)

Email Address: [deolnavjot@yahoo.com](mailto:deolnavjot@yahoo.com)

Phone Number: 587-090-1234

Street Address: 001 01 Street

Unit Number: 12

Country: Canada (selected from drop down list)

Region/Province: Alberta (selected from drop down list)

City: Edmonton

Zip/Postal Code: T7G 1P6

Use Home Email for ICCP Business: Yes

Use Home Email for Prof. Communications: Yes

For CBIP mailing use: Home Address (selected from drop down list)

For TDWI mailing use: Home Address (selected from drop down list)

For ICCP mailing use: Home Address (selected from drop down list)

For related mailing: Home Address (selected from drop down list)

Previously filed for CBIP Certification with TDWI?: No

Previously filed for Certification with ICCP? Yes

Passed an ICCP exam and received a designation? Yes

Designation Received: 2020-03-02

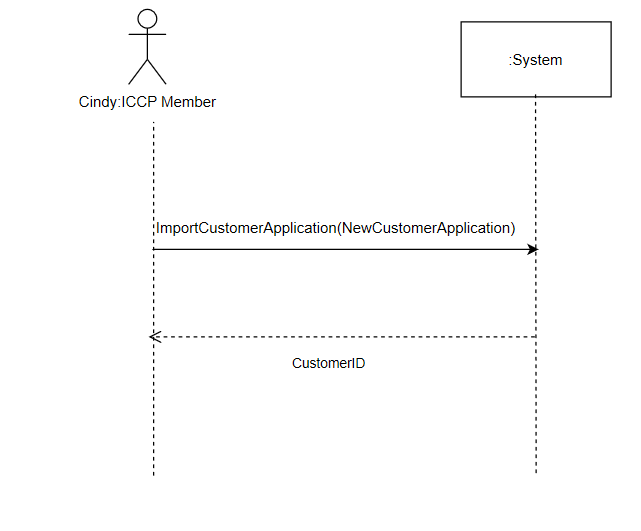
Disability Check: No

TDWI Membership: I would like to become a TDWI member (selected from drop down list)

When completed, Cindy submits the Submit Application request to the System. The System processes the request and add the user as a new Customer. When completed, System returns the generated CustomerID for the new customer to Cindy.

* Alternative Flows
* Secondary/ Alternative scenario narratives: N/A
* Special Requirements: N/A
* Preconditions: Completely filled Customer application form is required.
* Postconditions: Customer Application form is recorded into the ICCP database and new Customer is created.
* Other diagrams
* System Sequence Diagram (SSD)

Primary scenario: Cindy, ICCP member wants to submit a new customer application.



* User Interface Prototype

A screen shot of a computer

Description automatically generated

2.2) Use Case: CreatePerson

* Brief Description: The purpose of this use case is to allow ICCP members to add new Customer.
* Flow of Events:
* Basic Flow
* Use case narrative:

Member wants to create a new Customer.

For this, member provides the following details:

First Name:

Last Name:

Middle Name:

Membership Type:

Customer Roles:

Organization:

Email Address:

Phone Number:

Street Address:

Unit Number:

Country:

Region/Province:

City:

Zip/Postal Code:

Customer Notes:

When completed, member submits the AddPerson request to the System. The System processes the request and add a new Customer. When completed, System returns the Confirmation back to member.

* Primary scenario narrative:

Primary scenario: Cindy, ICCP member wants to create Navjot Deol as new Customer.

Cindy, ICCP member wants to create a Navjot Deol as a new Customer.

For this, Cindy provides the following details:

First Name: Navjot

Last Name: Deol

Middle Name: Kaur

Membership Type: Candidate (selected from drop down list)

Customer Roles: Employee (selected from drop down list)

Organization: NAIT (selected from drop down list)

Email Address: [deolnavjot@yahoo.com](mailto:deolnavjot@yahoo.com)

Phone Number: 587-090-1234

Street Address: 001 01 Street

Unit Number: 12

Country: Canada (selected from drop down list)

Region/Province: Alberta (selected from drop down list)

City: Edmonton

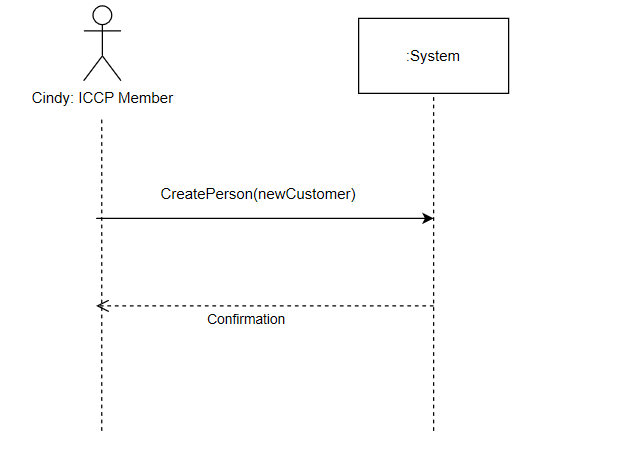
Zip/Postal Code: T7G 1P6

Customer Notes: I want to be added as new person

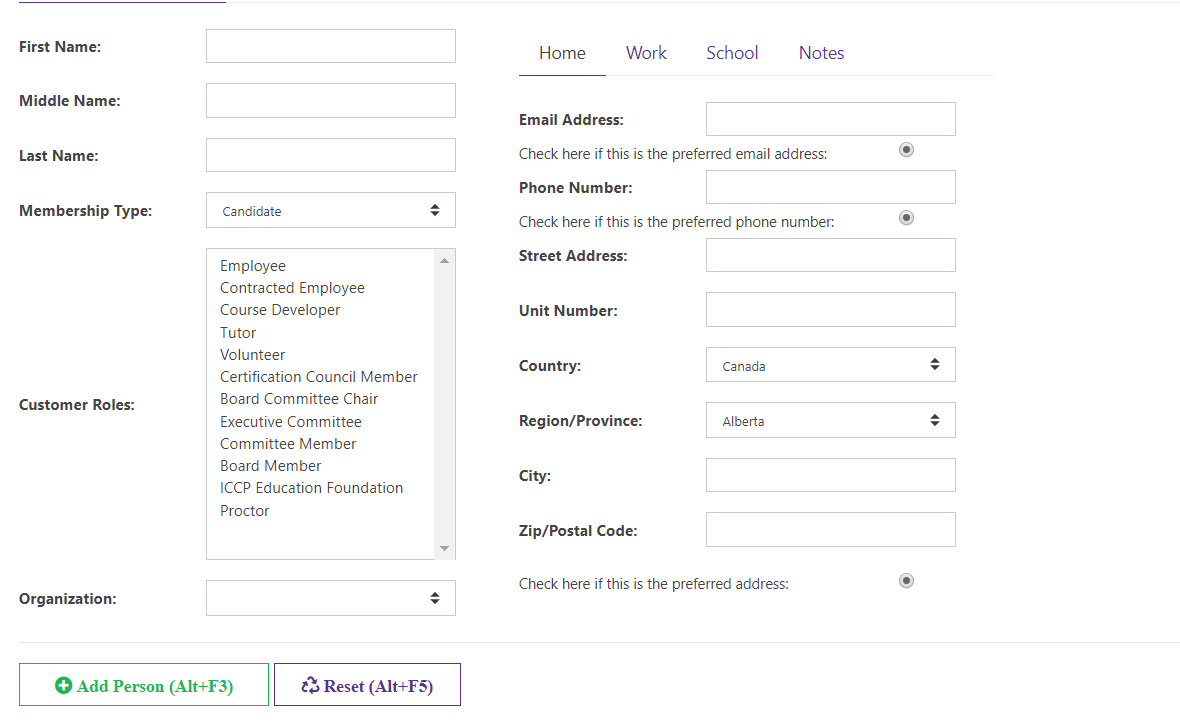
When completed, Cindy submits the AddPerson request to the System. The System processes the request and add Navjot as a new Customer. When completed, System returns the Confirmation back to Cindy.

* Alternative Flows
* Secondary/ Alternative scenario narratives: N/A
* Special Requirements: N/A
* Preconditions: N/A
* Postconditions: New Customer is created.
* Other diagrams
* System Sequence Diagram (SSD)

Primary scenario: Cindy, ICCP member wants to create Navjot Deol as new Customer.



* User Interface Prototype



2.3) Use Case: LookUpPerson

* Brief Description: The purpose of this use case is to allow ICCP members to view information of Persons added in the database.
* Flow of Events:
* Basic Flow
* Use Case narrative:

Member wants to view Person’s information.

For this, member provides the following:

Customer ID:

After this, member submits the LookupPerson request to the system. The System processes the request and find the details of Person and returns the details back to member.

* Primary scenario narrative:

Primary scenario: Cindy, ICCP member wants to view Navjot’s information from the database.

Cindy, ICCP member wants to view Navjot’s information.

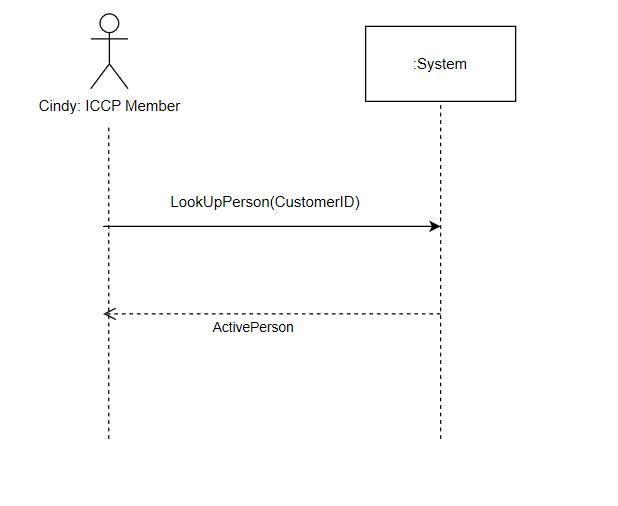
For this, Cindy provides the following:

Customer ID: 1

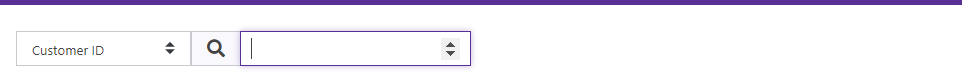
After this, Cindy submits the LookupPerson request to the system. The System processes the request and find the details of Navjot and returns the details back to Cindy.

* Alternative Flows
* Secondary/ Alternative scenario narratives: N/A
* Special Requirements: N/A
* Preconditions: Customer has to be added.
* Postconditions: N/A.
* Other diagrams
* System Sequence Diagram (SSD)

Primary scenario: Cindy, ICCP member wants to view Navjot’s information from the database.



* User Interface Prototype



2.4) Use Case: UpdatePerson

* Brief Description: The purpose of this use case is to allow ICCP members to update the person information already added in the database.
* Flow of Events:
* Basic Flow
* Use Case narrative:

Member wants to update Person’s information. For this, member must view the details of Person that are already added in the database.

For this, member provides the following:

Customer ID:

After this, member submits the LookupPerson request to the system. The System processes the request and find the details of Person and returns the details back to member. After viewing all the details, member enters the details that needs to be updated.

First Name:

Last Name:

Middle Name:

Membership Type:

Customer Roles:

Organization:

Email Address:

Phone Number:

Street Address:

Unit Number:

Country:

Region/Province:

City:

Zip/Postal Code:

When completed, member submits the SaveChanges request to the system. The system processes the request and update the Person’s information and returns Confirmation back to member.

* Primary scenario narrative:

Primary scenario: Cindy, ICCP member wants to update Navjot’s information in the database.

Cindy, ICCP member wants to update Navjot’s information. For this, Cindy must view the details of Navjot that are already added in the database.

For this, Cindy provides the following:

Customer ID: 1

After this, Cindy submits the LookupPerson request to the system. The System processes the request and find the details of Navjot and returns the details back to Cindy. After viewing all the details, Cindy enters the details that needs to be updated.

First Name: Nav

Last Name: Deol

Middle Name: Kaur

Membership Type: Candidate (selected from drop down list)

Customer Roles: Employee (selected from drop down list)

Organization: NAIT (selected from drop down list)

Email Address: [deolnavjot@yahoo.com](mailto:deolnavjot@yahoo.com)

Phone Number: 587-090-1234

Street Address: 001 01 Street

Unit Number: 1

Country: Canada (selected from drop down list)

Region/Province: Alberta (selected from drop down list)

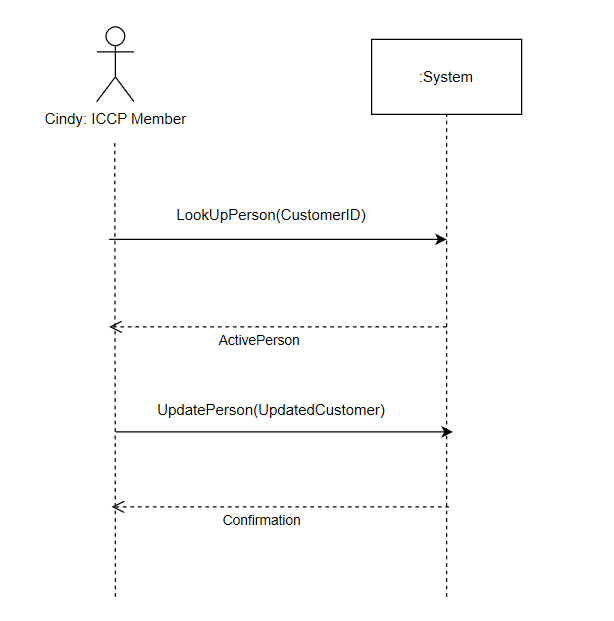
City: Edmonton

Zip/Postal Code: T6G 1P6

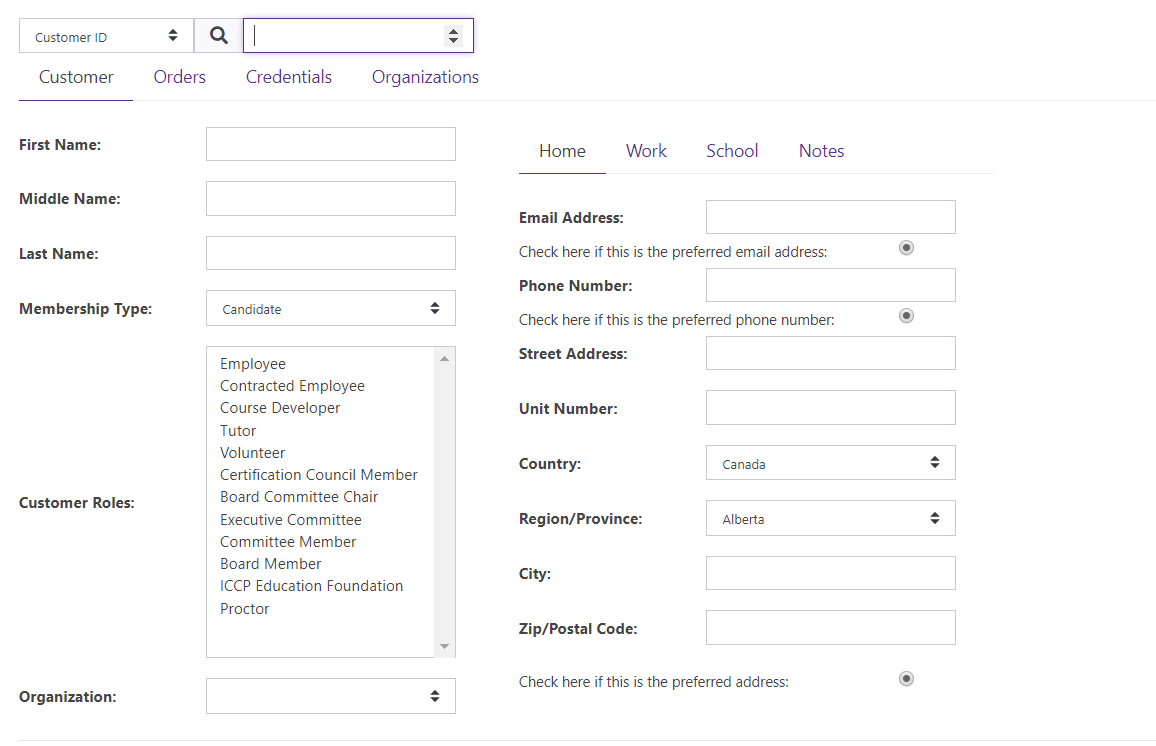
When completed, Cindy submits the SaveChanges request to the system. The system processes the request and update the Navjot’s information and returns Confirmation back to Cindy.

* Alternative Flows
* Secondary/ Alternative scenario narratives: N/A
* Special Requirements: N/A
* Preconditions: Person has to be added.
* Postconditions: Person information has been updated.
* Other diagrams
* System Sequence Diagram (SSD)

Primary scenario: Cindy, ICCP member wants to update Navjot’s information in the database.



* User Interface Prototype



2.5) Use Case: CreateOrder

* Brief Description: The purpose of this use case is to allow customers to place orders.
* Flow of Events:
* Basic Flow
* Use case narrative:

Customer of ICCP wants to place order for inventory items. For this, Customer must enter CustomerID and provides the following:

Inventory Item Name:

Quantity:

When completed, Customer submits the AddToPreview request to the system. The system processes the request and add the items ordered by Customer in the Preview and returns the list of items ordered along with OrderID to Customer. After reviewing the order, Customer submits the Place Order request to the system. The system processes the request and place the order and returns Confirmation back to Customer.

* Primary scenario narrative:

Primary scenario: Navjot, Customer of ICCP wants to place order for CAD and ABMP (inventory item).

Navjot, a customer of ICCP wants to place order for 2 CAD inventory items. For this, Navjot must enter CustomerID and provides the following:

Inventory Item Name: CAD (selected from the drop-down list)

Quantity: 2

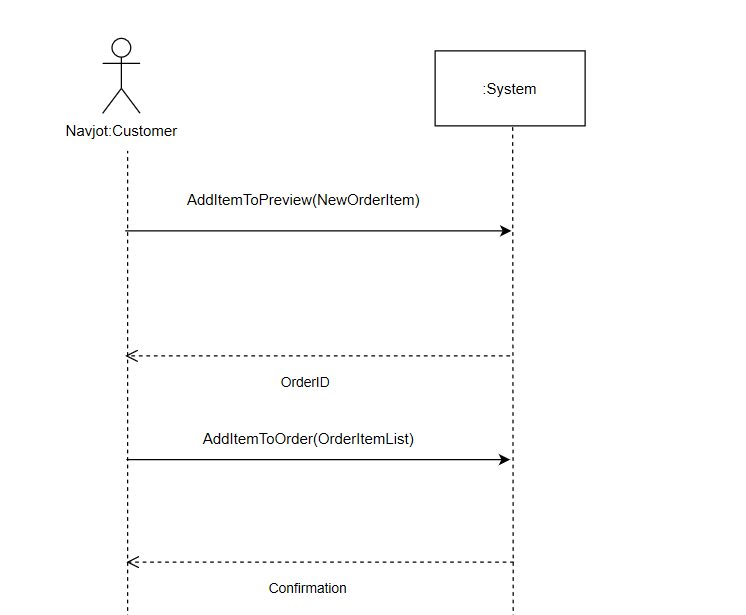
Inventory Item Name: ABMP

Quantity:3

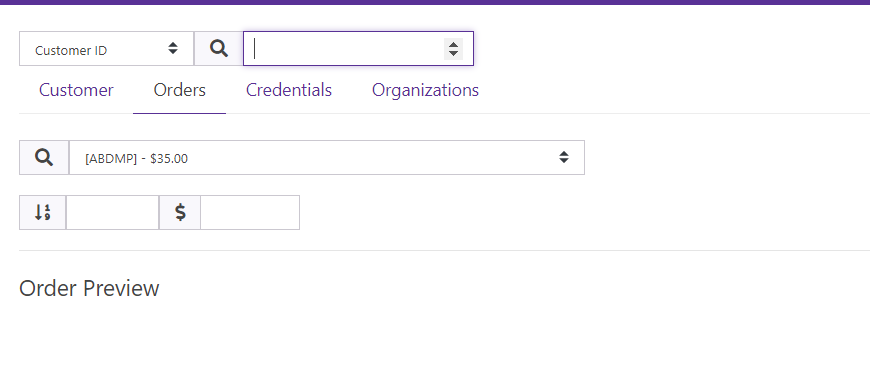
When completed, Navjot submits the AddToPreview request to the system. The system processes the request and add the items ordered by Navjot in the Preview and returns the list of items ordered along with OrderID to Navjot. After reviewing the order, Navjot submits the Place Order request to the system. The system processes the request and place the order for Navjot and returns Confirmation back to Navjot.

* Alternative Flows
* Secondary/ Alternative scenario narratives: N/A
* Special Requirements: N/A
* Preconditions: Inventory items has to be present in the database.
* Postconditions: Order has been placed.
* Other diagrams
* System Sequence Diagram (SSD)

Primary scenario: Navjot, Customer of ICCP wants to place order for CAD and ABMP (inventory item).



* User Interface Prototype



2.6) Use Case: AddInventory

* Brief Description: The purpose of this use case is to allow ICCP members to add new inventory items to the database.
* Flow of Events:
* Basic Flow
* Use case narrative:

Member of ICCP wants to add new Inventory item to the database. For this, member provides the following:

Name:

Description:

Unit Price:

Category:

When completed, member submits the CreateNewItem request to the system. The system processes the request and add the inventory item to the database and returns Confirmation back to member.

* Primary scenario narrative:

Primary scenario: Cindy, member of ICCP wants to add new Inventory Item Data Analytics under Self Study category.

Cindy, a member of ICCP wants to add new Inventory item to the database. For this, Cindy provides the following:

Name: Data Analytics

Description: Analysis of Data Sets

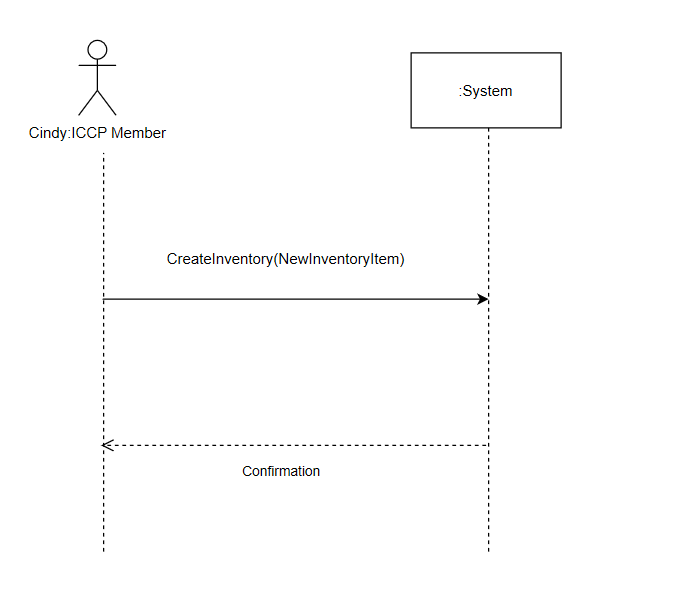
Unit Price: $125.00

Category: Self Study (selected from the drop-down list)

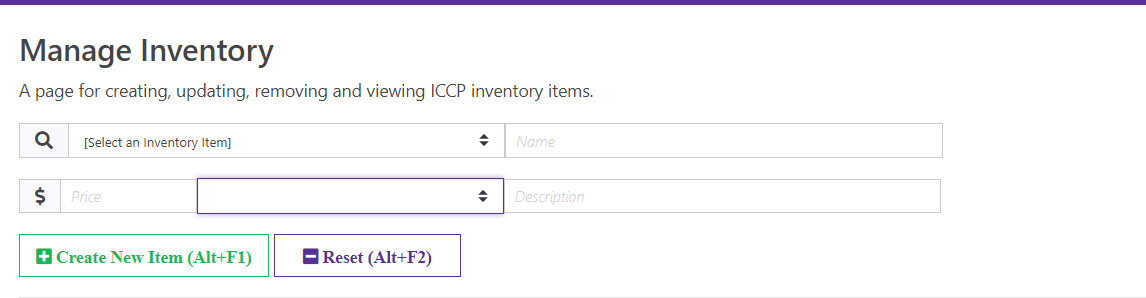
When completed, Cindy submits the CreateNewItem request to the system. The system processes the request and add the inventory item to the database and returns Confirmation back to Cindy.

* Alternative Flows
* Secondary/ Alternative scenario narratives: N/A
* Special Requirements: N/A
* Preconditions: N/A
* Postconditions: Inventory Item has been added in the database.
* Other diagrams
* System Sequence Diagram (SSD)

Primary scenario: Cindy, member of ICCP wants to add new Inventory Item Data Analytics under Self Study category.



* User Interface Prototype



2.7) Use Case: UpdateInventory

* Brief Description: The purpose of this use case is to allow ICCP members to update the existing inventory items.
* Flow of Events:
* Basic Flow
* Use Case narrative:

Member of ICCP wants to update existing Inventory item. For this, member provides the following:

Inventory ID:

After reviewing the details of Inventory Item, member provides the details that needs to be updated as:

Name:

Description:

Unit Price:

Category:

When completed, member submits the Save Changes request to the system. The system processes the request and update the inventory items in the database and returns Confirmation back to member.

* Primary scenario narrative:

Primary scenario: Cindy, member of ICCP wants to update unit price and name of Inventory Item Data Analytics.

Cindy, a member of ICCP wants to update existing Inventory item. For this, Cindy provides the following:

Inventory ID: Provided by selecting Inventory Item from the drop-down list

After reviewing the details of Inventory Item, Cindy provides the details that needs to be updated as:

Name: Data Analytics Processing

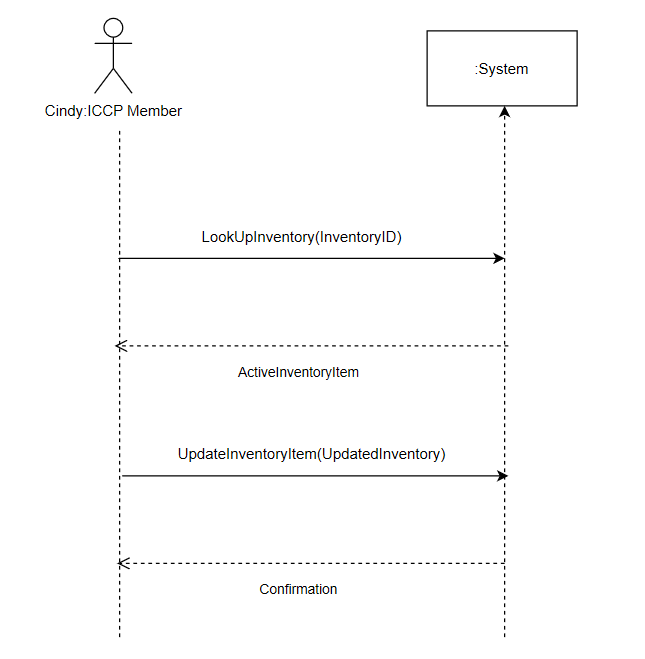
Description: Analysis of Data Sets Processing

Unit Price: $ 80.00

When completed, Cindy submits the Save Changes request to the system. The system processes the request and update the inventory items in the database and returns Confirmation back to Cindy.

* Alternative Flows
* Secondary/ Alternative scenario narratives: N/A
* Special Requirements: N/A
* Preconditions: Inventory Item has to be present in the database.
* Postconditions: Inventory Item has been updated in the database.
* Other diagrams
* System Sequence Diagram (SSD)

Primary scenario: Cindy, member of ICCP wants to update unit price and name of Inventory Item Data Analytics.



* User Interface Prototype



2.8) Use Case: RemoveInventory

* Brief Description: The purpose of this use case is to allow ICCP members to remove the existing inventory items.
* Flow of Events:
* Basic Flow
* Use Case narrative:

Member of ICCP wants to delete the existing Inventory item from the database. For this, member provides the following:

Inventory ID:

When completed, member submits the Remove Inventory request to the system. The system processes the request and delete the inventory items from the database and returns Confirmation back to member.

* Primary scenario narrative:

Primary scenario: Cindy, member of ICCP wants to delete Inventory Item Data Anaytics.

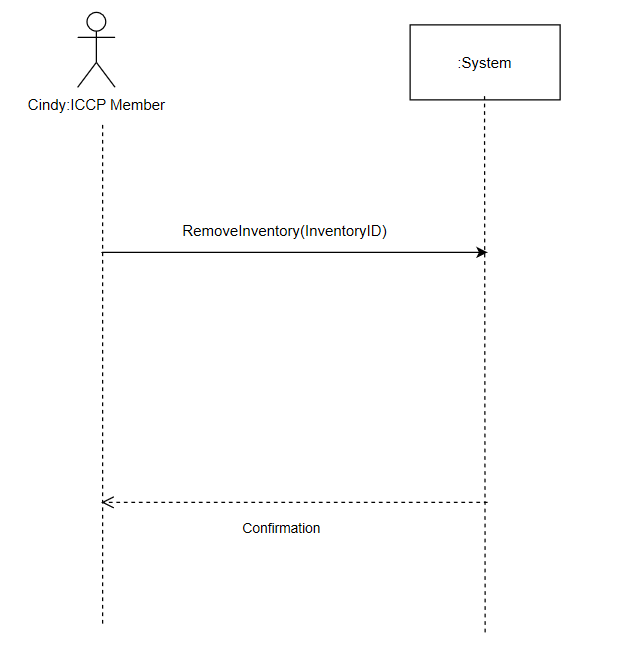
Cindy, Member of ICCP wants to delete the existing Inventory item from the database. For this, Cindy provides the following:

Inventory ID: Data Analytics

When completed, Cindy submits the Remove Inventory request to the system. The system processes the request and delete the inventory items from the database and returns Confirmation back to member.

* Alternative Flows
* Secondary/ Alternative scenario narratives: N/A
* Special Requirements: N/A
* Preconditions: Inventory Item has to be present in the database.
* Postconditions: Inventory Item has been deleted from the database.
* Other diagrams
* System Sequence Diagram (SSD)

Primary scenario: Cindy, member of ICCP wants to delete Inventory Item Data Anaytics.



2.9) Use Case: Add Organization

Brief Description: The purpose of this use case is to allow ICCP staff to add an organization.

* Use Case narrative:

ICCP staff wants to add Organization.

ICCP staff provides the following:

* Organization Name
* Organization Type
* Billing Street Address
* Billing Unit Number
* Billing Region/Province
* Billing City
* Billing Zip/Postal Code
* Billing Country
* Shipping Street Address
* Shipping Unit Number
* Shipping Region/Province
* Shipping City
* Shipping Zip/Postal Code
* Shipping Country

When completed, ICCP staff submits the request to the system to Add Organization. System processes the request, adds an organization into the database and returns confirmation back to ICCP Staff.

* Primary Scenario narrative:

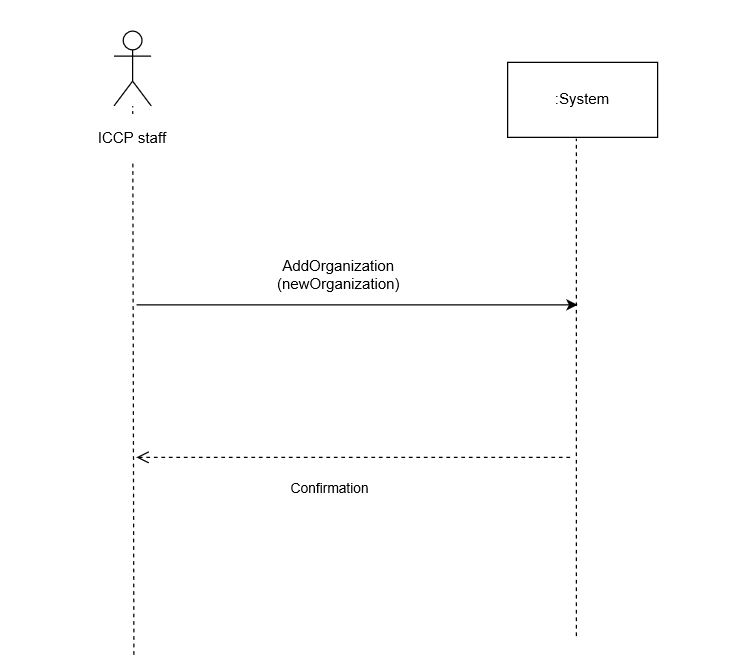
Primary Scenario: Kedre Gurre, Director of Member Services, ICCP wants to add an organization NAIT, selects an Organization Type as University, fills in all the contact and address details of the organization. He then submits the request Add Organization to the system. System then processes the request, adds NAIT as an organization in the database and returns confirmation back to Kedre Gurre.

So, Kedre Gurre provides the following:

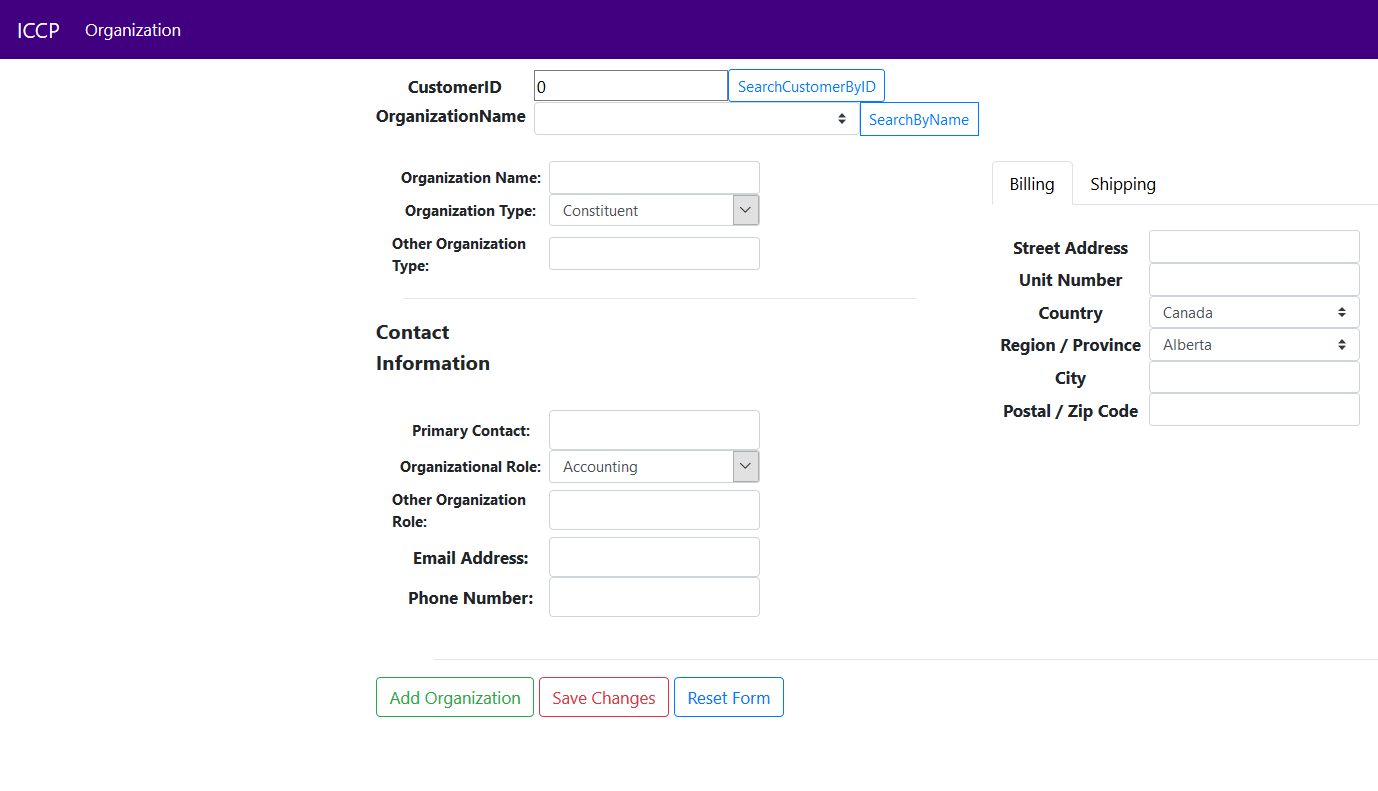
* Organization Name: NAIT
* Organization Type: University/College
* Other Organization Type:
* Primary Contact: Phone
* Organizational Role:
* Other Organizational Role:
* Email Address: AskNait@nait.ca
* Phone Number: 7907161616
* Billing Street Address:11762 106 Str
* Billing Unit Number:11
* Billing Region/Province: Alberta
* Billing City: Edmonton
* Billing Zip/Postal Code: T5G 241
* Billing Country: Canada
* Shipping Street Address:11762 106 Str
* Shipping Unit Number:11
* Shipping Region/Province: Alberta
* Shipping City: Edmonton
* Shipping Zip/Postal Code: T5G 241
* Shipping Country: Canada

When completed, Kedre Gurre submits the request to the system to Add Organization. System processes the request, adds an organization into the database and returns confirmation back to Kedre Gurre.

* System Sequence Diagram



* UI



2.10) Use Case: Search Organization

Brief Description: The purpose of this use case is to allow ICCP staff to search any Organization

* Use Case narrative:

ICCP staff wants to search an organization.

ICCP staff provides the following:

* CustomerID

When completed, ICCP staff sends request to the system to search organization. System processes the request and sends Organization details to ICCP staff.

* Primary Scenario narrative:

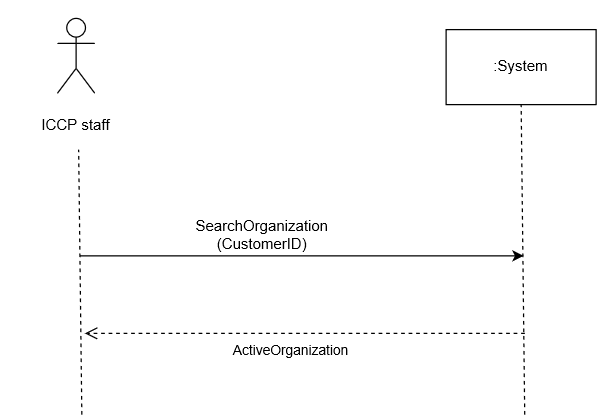
Primary Scenario: Kedre Gurre, Director of Member Services, ICCP wants to search an Organization NAIT.

ICCP provides the following:

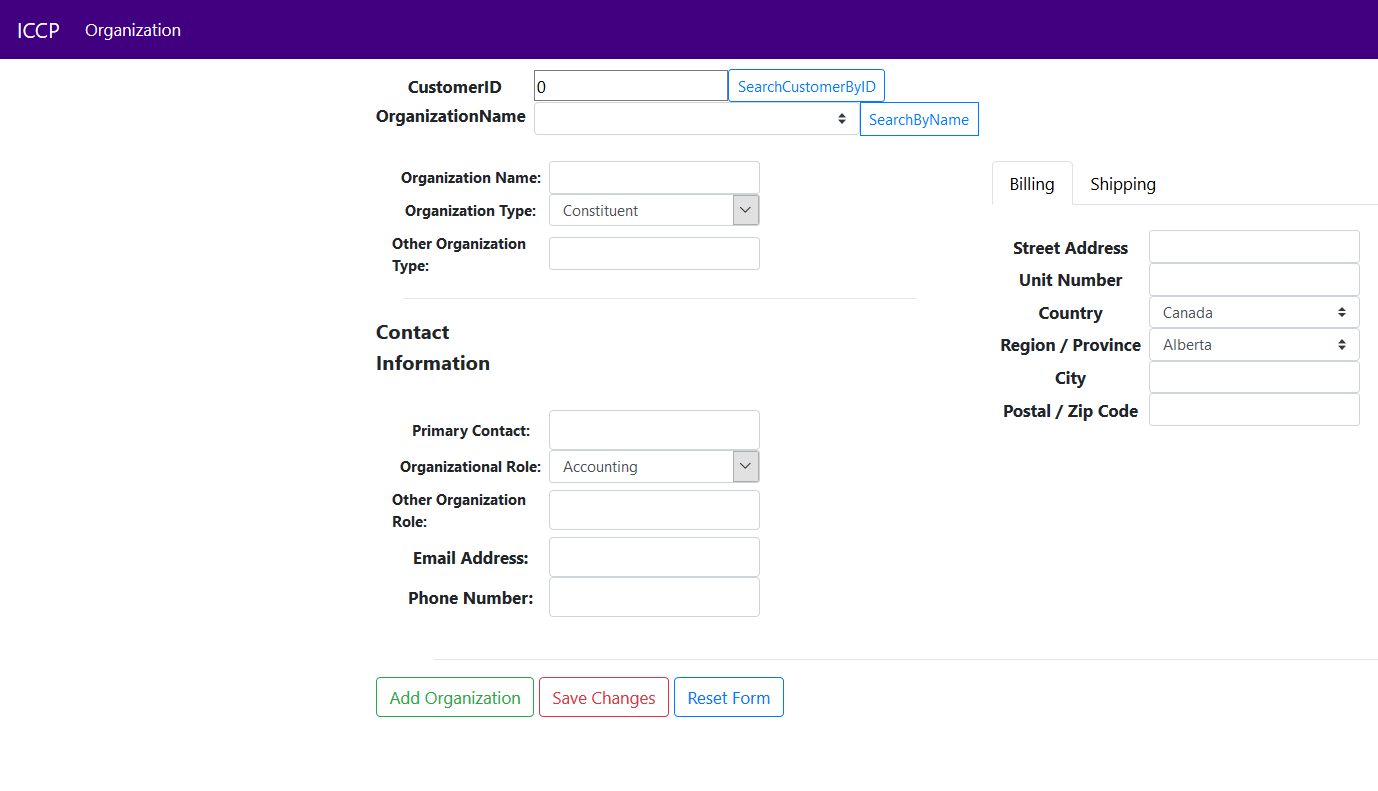
* CustomerID:1173

When completed, ICCP staff sends request to the system to search organization. System processes the request and sends back Organization details to ICCP staff.

* System Sequence Diagram



UI



2.11) Use Case: Modify Organization

Brief Description: The purpose of this use case is to allow ICCP staff to update an organization.

* Use Case narrative:

ICCP staff wants to update an Organization.

ICCP staff provides the following:

* Customer ID

When completed, ICCP staff sends request to the system to search organization. System processes the request and sends back Organization details to ICCP staff.

Then, ICCP staff provides the following:

* Organization Name
* Organization Type
* Primary Contact
* Organization Role
* Email Address
* Phone Number
* Billing Street Address
* Billing Unit Number
* Billing Region/Province
* Billing City
* Billing Zip/Postal Code
* Billing Country
* Shipping Street Address
* Shipping Unit Number
* Shipping Region/Province
* Shipping City
* Shipping Zip/Postal Code
* Shipping Country

When completed, ICCP staff submits the request to the system to Update Organization. System processes the request, updates organization in the database and returns confirmation back to ICCP Staff.

* Primary Scenario narrative:

Primary Scenario: Kedre Gurre, Director of Member Services, ICCP wants to update an Organization to change Billing street address of NAIT to 11762 105 Str.

ICCP provides the following:

* Organization Name: NAIT

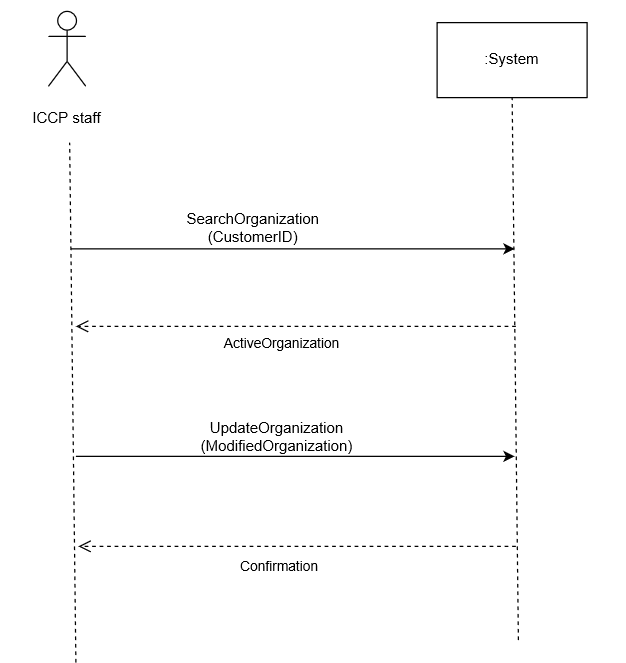
When completed, ICCP staff sends request to the system to search organization. System processes the request and sends back Organization details to ICCP staff.

Then, ICCP staff provides the following:

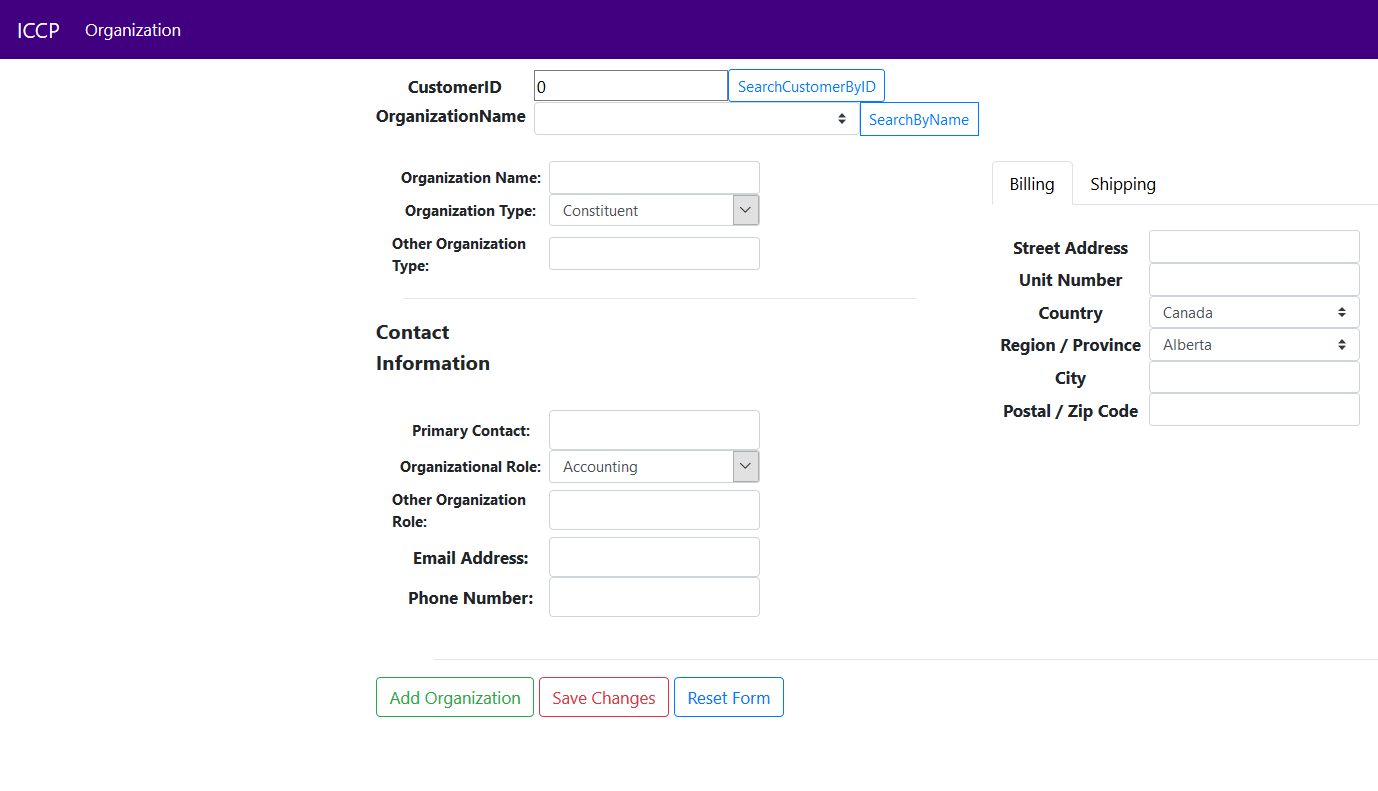
* Organization Name: NAIT
* Organization Type: University/College
* Primary Contact: 7804716248
* Organization Role: Education/ Teacher
* Email Address: AskNait@nait.ca
* Phone Number: 7804716248
* Billing Street Address: 11762 105 Str
* Billing Unit Number: 12345
* Billing City: Edmonton
* Billing Zip/Postal Code: T5G 241
* Billing Country: Canada
* Billing Region/Province: Alberta
* Shipping Street Address: 11762 105 Str
* Shipping Unit Number: 12345
* Shipping City: Edmonton
* Shipping Zip/Postal Code: T5G 241
* Shipping Country: Canada
* Shipping Region/Province: Alberta

When completed, ICCP staff submits the request to the system to Update Organization. System processes the request, updates organization in the database and returns confirmation back to ICCP Staff.

* System Sequence Diagram



UI



2.12) Use Case: Lookup Credentials

Use Case narrative:

ICCP staff wants to lookup candidate data.

ICCP staff provides the following:

* Customer Name or
* Customer ID or
* Order ID

When completed, ICCP staff submits the request to the system and system look up candidate record. System processes the request, After ICCP staff search candidate data into the database and returns confirmation.

* Primary Scenario narrative:

Primary Scenario: John Smith is Member of ICCP, he wants to search candidate data by providing customer name or customer id or order id . He then submits the request to lookup candidate data to the system. System then processes the request, find candidate information in the database and returns confirmation.

So, John Smith provides the following:

* Customer Name: Shweta Mehta or
* Customer Id: 25 or
* Order Id: 1002

When completed, John Smith submits the request to the system to search candidate information. System processes the request, search candidate data into the database and returns confirmation.

* System Sequence Diagram

Primary Scenario: John Smith is Member of ICCP, he wants to search candidate data.

: System

John Smith

: Iccp Staff

Search Candidate Data

(customer ID)

ActiveCustomer

2.13) Use Case: Update Credentials

Brief Description: The purpose of this use case is to allow ICCP staff to update candidate Data.

* Use Case narrative:

ICCP staff wants to update candidate data.

ICCP staff provides the following:

* Customer name or
* Customer ID or
* Order ID

When completed, ICCP staff sends request to the system to search candidate information. System processes the request and sends back candidate details to ICCP staff.

Then, ICCP staff provides the following:

* Certificate Name
* Birthdate
* Gender
* Career start date
* Organization type
* Job Position
* Education degree
* Education Major
* Years Business Exp:
* Years IT Exp :
* Retirement Status :

When completed, ICCP staff submits the request to the system to Update candidate information. System processes the request, updates candidate data in the database and returns confirmation.

* Primary Scenario narrative:

Primary Scenario: John Smith is Member of ICCP, he wants to update candidate data by providing customer name or customer id or order id .

ICCP provides the following:

* Customer Name: Jennie smith

When completed, ICCP staff sends request to the system to search candidate. System processes the request and sends back candidate details to ICCP staff.

Then, ICCP staff provides the following:

* Certificate Name: Jennie smith
* Birthdate: 10/11/1993
* Gender: Female
* Career start date: 01/14/2011
* Organization type: Corporate Contractor
* Job Position: Accounting
* Education degree: Bachelor’s degree
* Education Major: Computer Science
* Years Business Exp: 2-4 years
* Years IT Exp: 2-4 years
* Retirement Status: none

When completed, ICCP staff submits the request to the system to Update candidate data. System processes the request, updates candidate information in the database and returns confirmation.

* System Sequence Diagram

Primary Scenario: John Smith is Member of ICCP, wants to update candidate data.

: System

John Smith

: Iccp Staff

Search Candidate Data

(customer ID)

Confirmation

Update Candidate Data

(activecustomer)

Confirmation

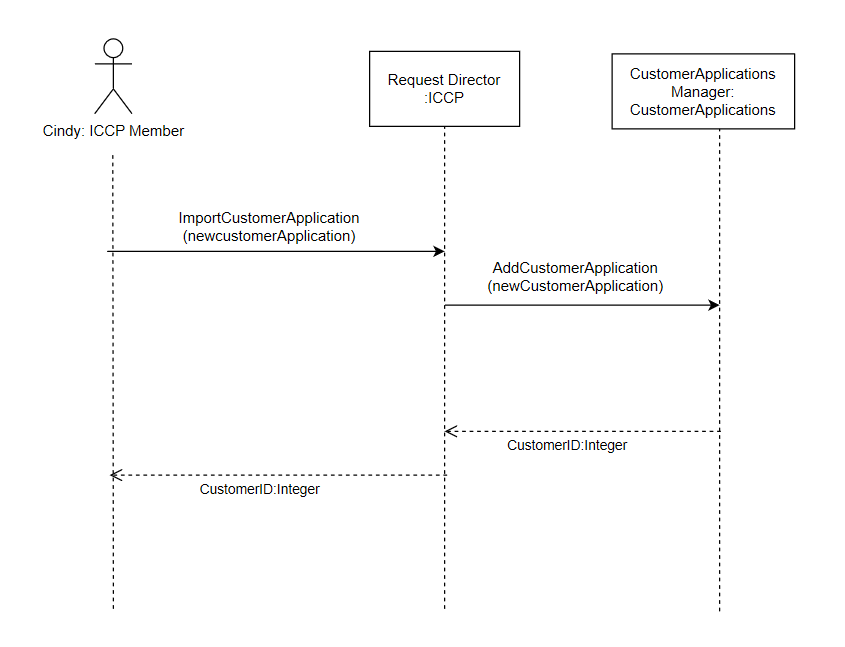
Design

3.1) Design model use case realizations

3.1.1 Use case: ImportCustomerApplication

Interaction Diagram:

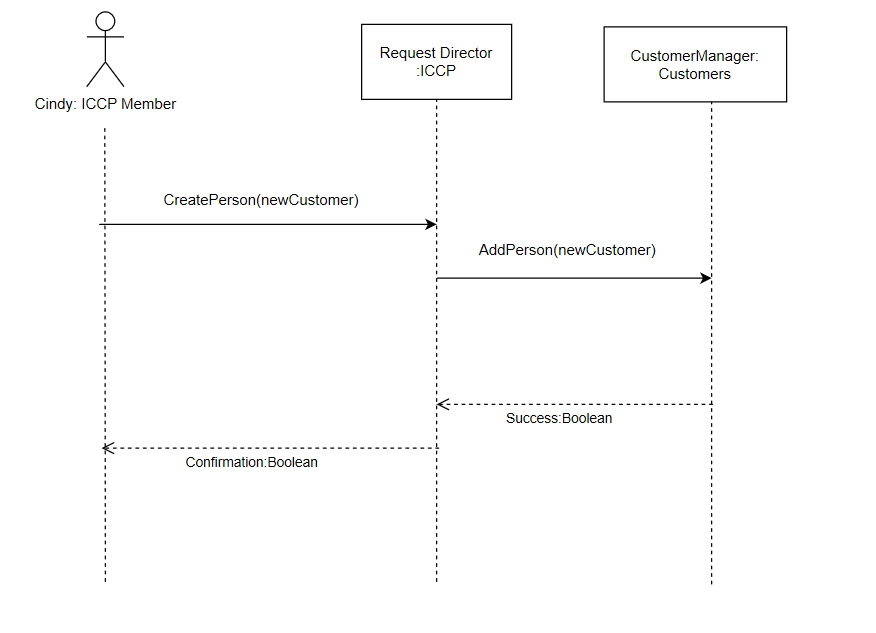
Primary scenario: Cindy, ICCP member wants to submit a new customer application.



3.1.2 Use case: CreatePerson

Interaction Diagram:

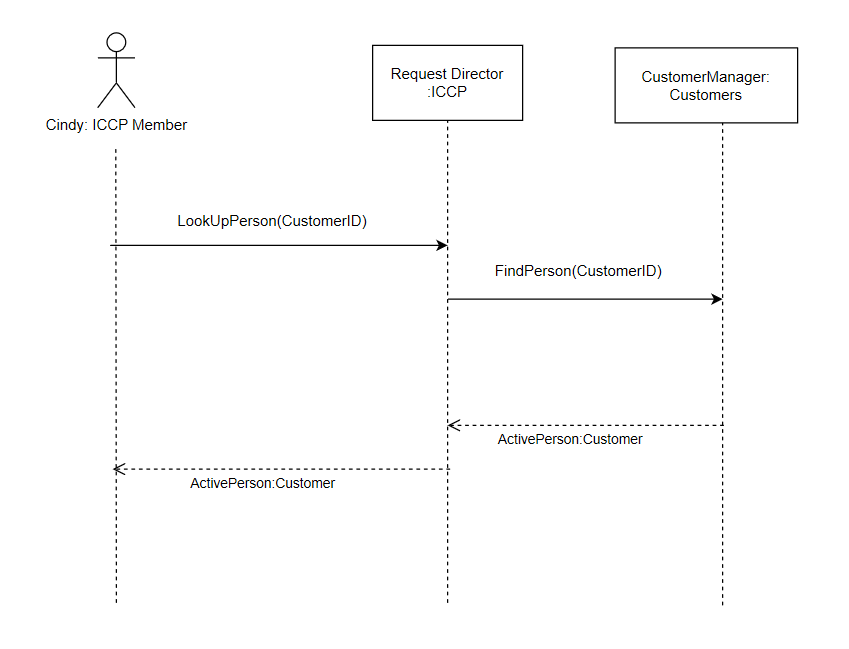
Primary scenario: Cindy, ICCP member wants to create Navjot Deol as new Person.



3.1.3 Use case: LookUpPerson

Interaction Diagram:

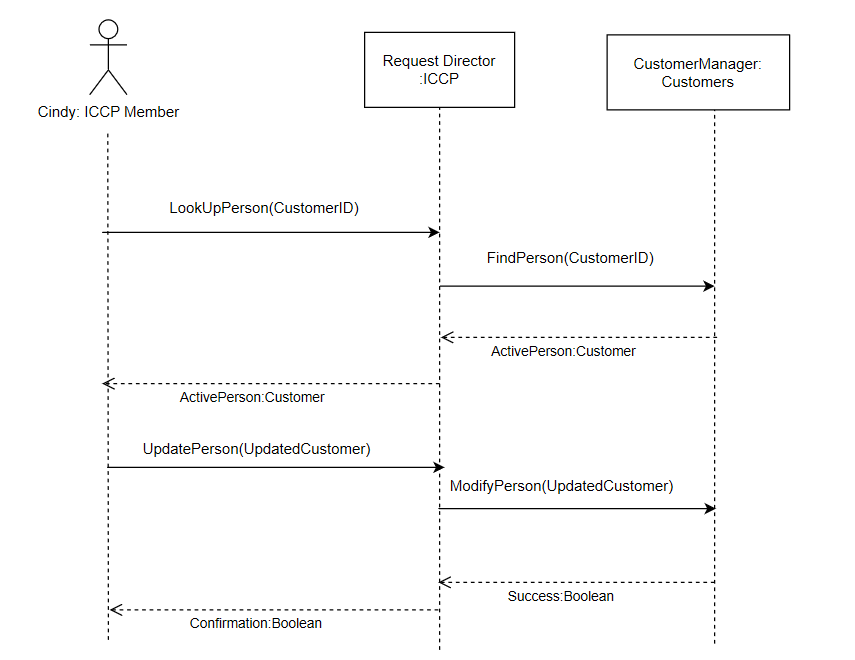
Primary scenario: Cindy, ICCP member wants to view Navjot’s information from the database.



3.1.4 Use case: UpdatePerson

Interaction Diagram:

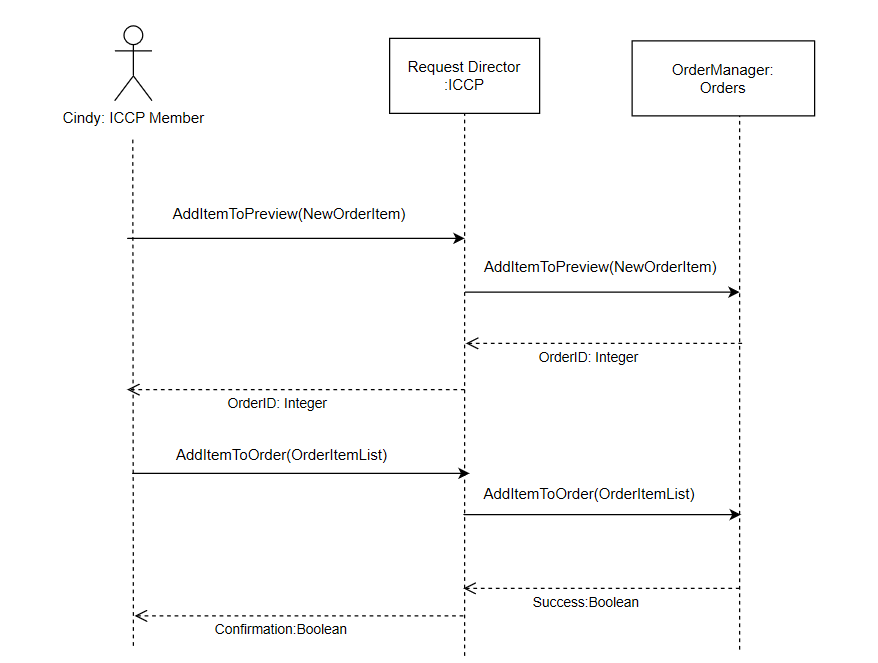
Primary scenario: Cindy, ICCP member wants to update Navjot’s information in the database.



3.1.5 Use case: CreateOrder

Interaction Diagram:

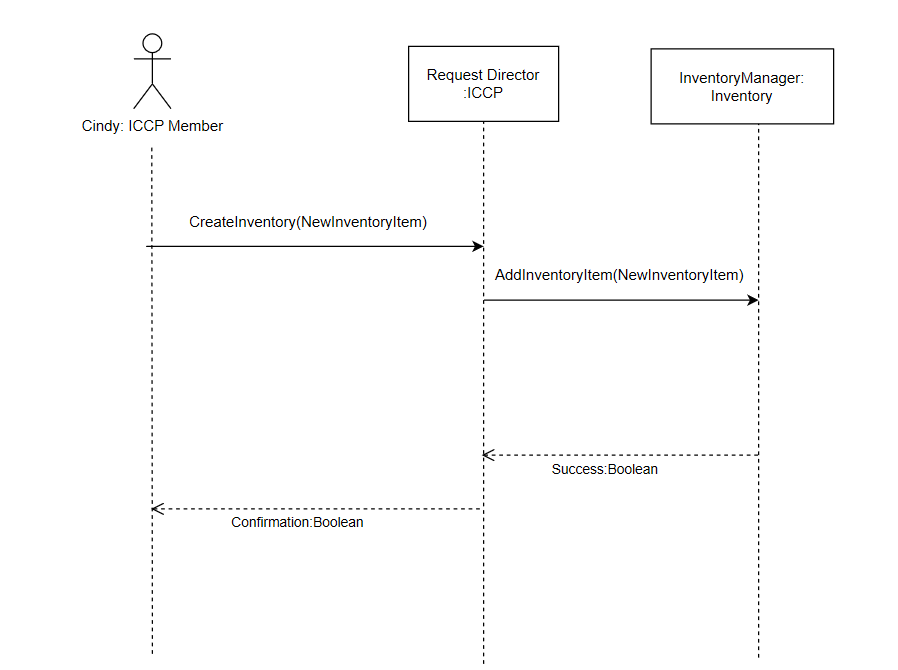
Primary scenario: Navjot, Customer of ICCP wants to place order for CAD and ABMP (inventory item).



3.1.6 Use case: AddInventory

Interaction Diagram:

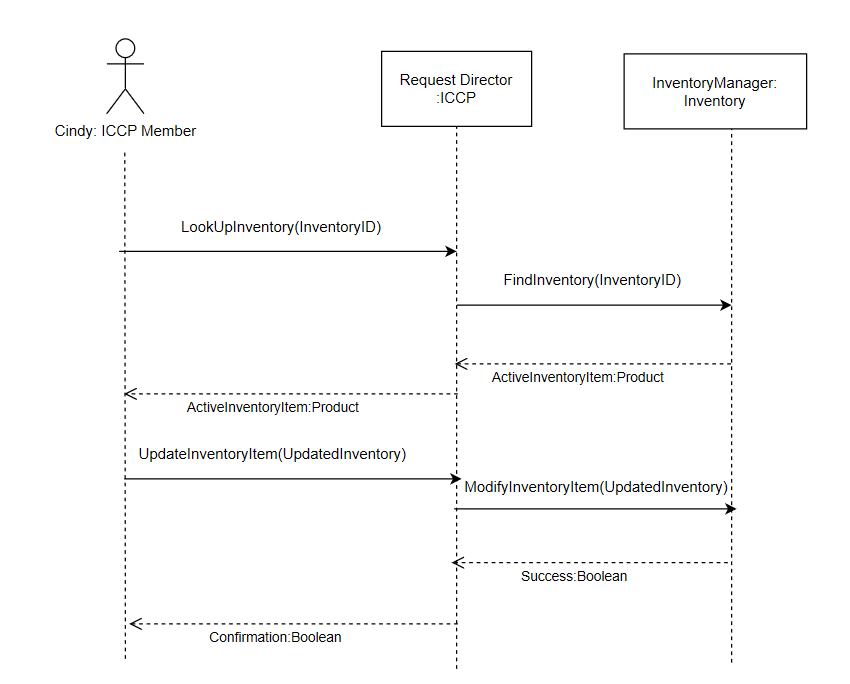
Primary scenario: Cindy, member of ICCP wants to add new Inventory Item Data Analytics under Self Study category.



3.1.7 Use case: UpdateInventory

Interaction Diagram:

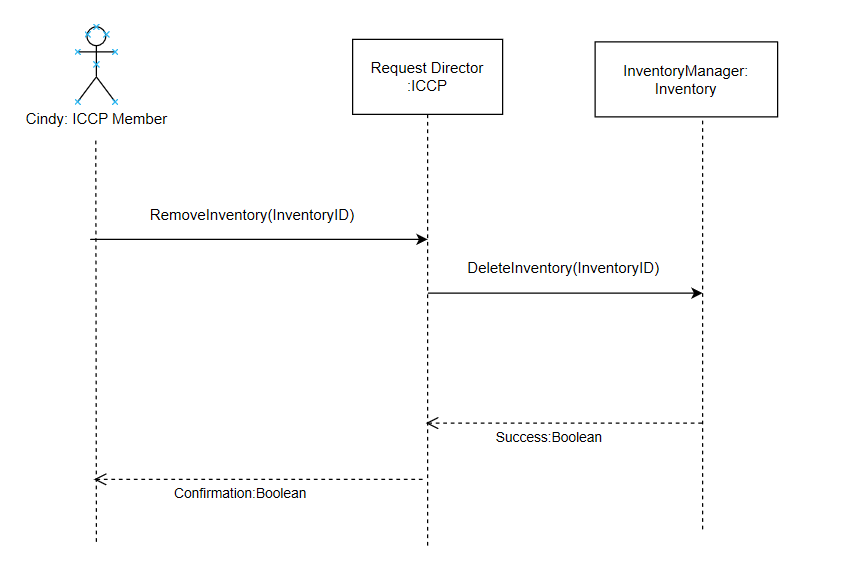
Primary scenario: Cindy, member of ICCP wants to update unit price and name of Inventory Item Data Analytics.



3.1.8 Use case: RemoveInventory

Interaction Diagram:

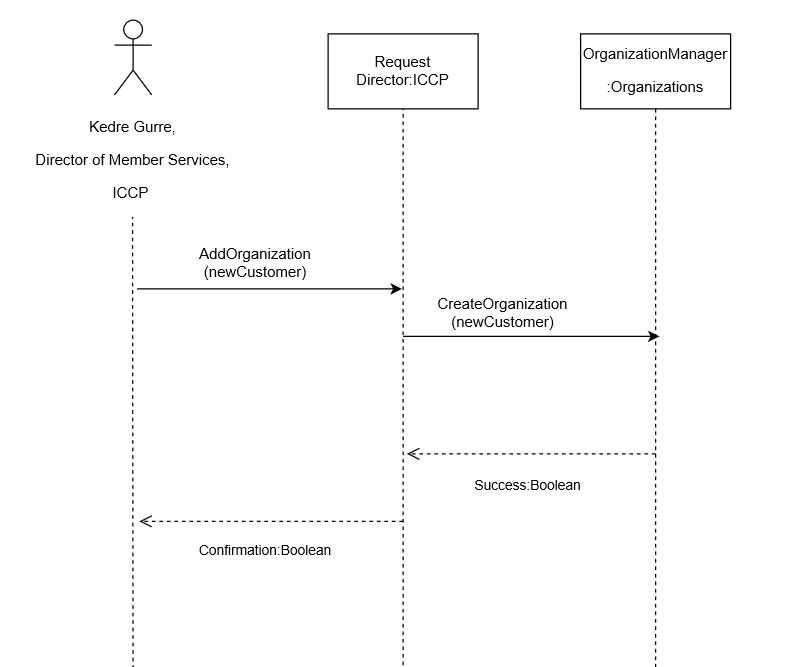
Primary scenario: Cindy, member of ICCP wants to remove Inventory Item Data Analytics



3.1.9 Use case: AddOrganization

Interaction Diagram

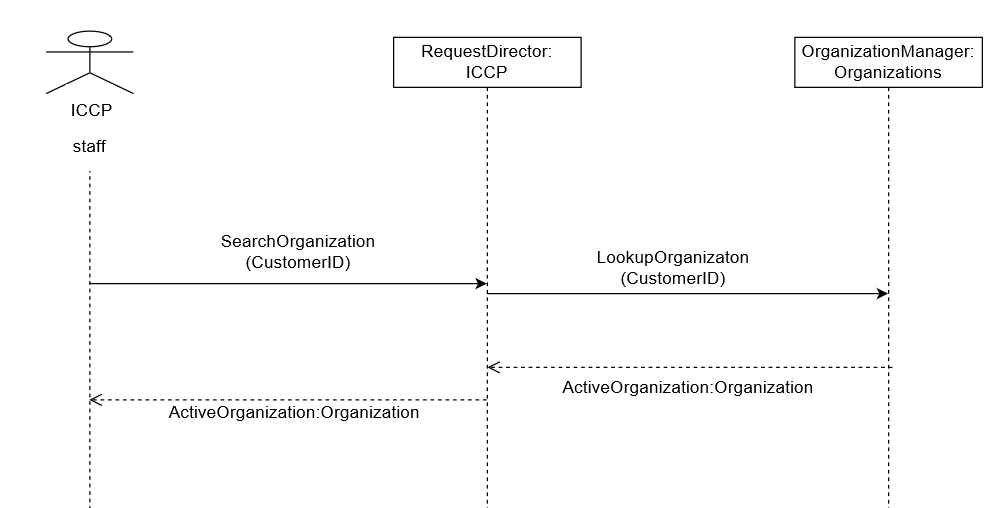
Primary Scenario: Kedre Gurre, Director of Member Services, ICCP wants to add an organization NAIT



3.1.10 Use case: SearchOrganization

Interaction Diagram

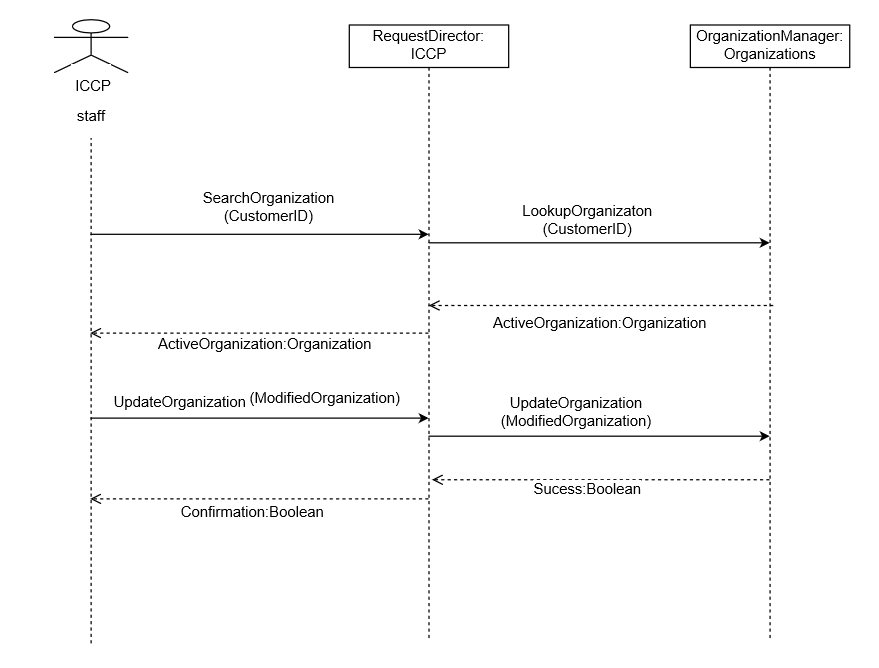
Primary Scenario: Kedre Gurre, Director of Member Services, ICCP wants to search an Organization NAIT .



3.1.11 Use case: ModifyOrganization

Interaction Diagram

Primary Scenario: Kedre Gurre, Director of Member Services, ICCP wants to update an Organization to change Billing street address of NAIT to 11762 105 Str



3.1.12 Use case: LookUpCrredentials

Interaction Diagram

Primary Scenario: John Smith is Member of ICCP, he wants to search candidate data.

System Manager: Customers

: TeeTimes

Request Director

: ICCP

John Smith

Iccp Staff

Search Candidate Data

Lookup Candidate Data

(Customer ID)

(Customer ID)

)

ActiveCustomer: Customer

ActiveCustomer: Customer

3.1.13 Use case: UpdateCredentials

Interaction Diagram

Primary Scenario: John Smith is Member of ICCP, wants to update candidate data.

System Manager: Customers

: TeeTimes

Request Director

: ICCP

John Smith

Iccp Staff

Search Candidate Data

(Customer ID)

Lookup Candidate Data

(Customer ID)

)

ActiveCustomer: Customer

ActiveCustomer: Customer

Update Candidate Data

activecustomer

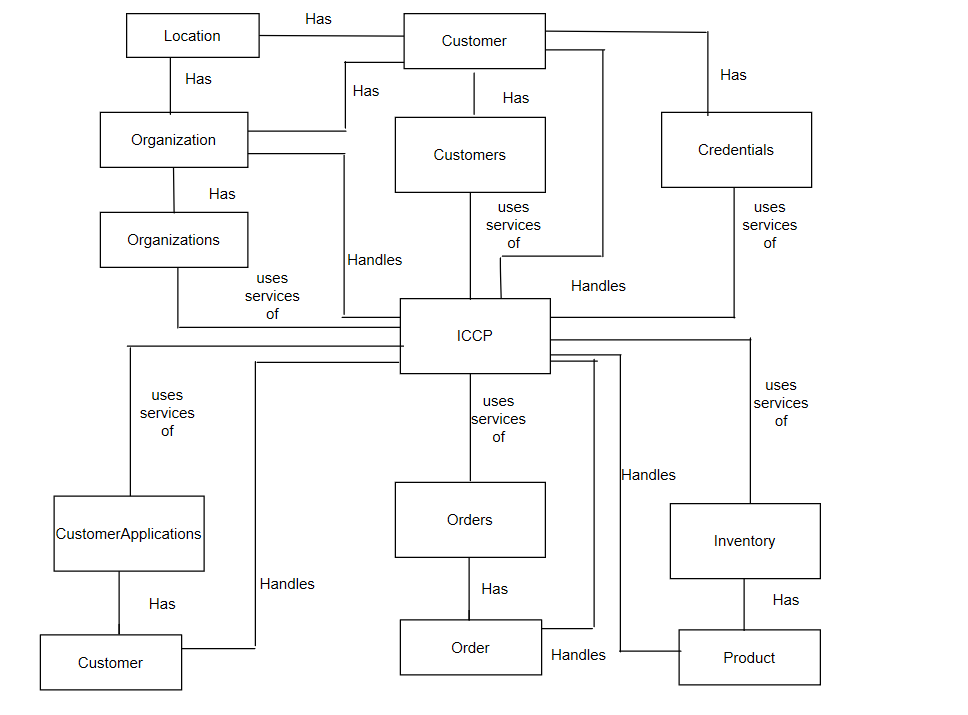
Update Candidate Data

activecustomer

Success : Boolean

Confirmation: Boolean

3.2) Design class diagram



3.3) Data Dictionary

Data Entities:

|  |
| --- |
| Address Type |
| <<PK>> AddressTypeID:Number {seed=1, increment=1}  AddressType:Unicode String{ size=10} |

|  |
| --- |
| Country Group |
| <<PK>> CountryCode:Unicode Character {size=5}  CountryName: Unicode Character{size=40}  CountryGroupID: Number |

|  |
| --- |
| Region Group |
| RegionCode: Unicode Character {size=5}  RegionName:Unicode Character {size=40}  <<FK>> CountryCode: Unicode Character {size=5, key=FK, reference= CountryGroup} |

|  |
| --- |
| Status |
| <<PK>> StatusID:Number {seed=1, increment=1}  Status: Unicode Character {size=30} |

|  |
| --- |
| Membership Roles |
| <<PK>> CustomerRoleID:Number {seed=1, increment=1}  CustomerRole: Unicode Character {size=30} |

|  |
| --- |
| Customer Type |
| <<PK>> CustomerTypeID:Number {seed=1, increment=1}  CustomerType: Unicode Character {size=20} |

|  |
| --- |
| Membership Type |
| <<PK>> MembershipTypeID:Number {seed=1, increment=1}  MembershipType: Unicode Character {size=40} |

|  |
| --- |
| Education Level |
| <<PK>> EducationLevelID:Number {seed=1, increment=1}  EducationLevel: Unicode Character {size=25} |

|  |
| --- |
| Education Major |
| <<PK>> EducationMajorID:Number {seed=1, increment=1}  EducationMajor: Unicode Character {size=20} |

|  |
| --- |
| Years Experience |
| <<PK>> YearsExperienceID:Number {seed=1, increment=1}  YearsExperience: Unicode Character {size=30} |

|  |
| --- |
| TDWI Membership Pref |
| <<PK>> TDWIMembershipPrefID:Number {seed=1, increment=1}  TDWIMembershipPref:Unicode Character {size=50} |

|  |
| --- |
| Retirement Status |
| <<PK>> RetirementStatusID:Number {seed=1, increment=1}  RetirementStatus:Unicode Character {size=50} |

|  |
| --- |
| Shipment Type |
| <<PK>> ShipmentTypeID:Number {seed=1, increment=1}  ShipmentType:Unicode Character {size=30} |

|  |
| --- |
| Payment Type |
| <<PK>> PaymentTypeID:Number {seed=1, increment=1}  PaymentType:Unicode Character {size=30} |

|  |
| --- |
| Payment Apportion Type |
| <<PK>> ApportionmentTypeID:Number {seed=1, increment=1}  ApportionmentType:Unicode Character {size=30} |

|  |
| --- |
| Inventory Category |
| <<PK>> CategoryID:Number {seed=1, increment=1}  Category:Unicode Character {size=35} |

|  |
| --- |
| Inventory |
| <<PK>> InventoryID:Number {seed=1, increment=1}  ItemName: Unicode String {size=90}  ItemDescription: Unicode String {size=90}  ItemUnitPrice: Currency  <<FK>> CategoryID: Number {key=FK, reference=Inventory Category}  <<FK>> StatusID: Number {key=FK, reference=Status}  CreateDate: DateTime  UpdateDate:DateTime  AccessKey: Unicode String {size=10} |

|  |
| --- |
| Customer |
| <<PK>> CustomerID:Number {seed=1, increment=1}  <<FK>> CustomerTypeID: Number {key=FK, reference=Customer Type}  CustomerName: Unicode String {size=90}  <<FK>> StatusID: Number {key=FK, reference=Status}  CreateDate: DateTime  UpdateDate:DateTime |

|  |
| --- |
| Customer Details |
| <<FK>> CustomerID:Number {key=FK, reference=Customer }  FirstName: Unicode String {size=50}  MiddleName: Unicode String {size=30}  LastName: Unicode String {size=50}  BirthDate: Unicode String {size=20}  Gender: Unicode Character  <<FK>> MembershipTypeID: Number {key=FK, reference=Membership Type}  MembershipNumber: Unicode Character {size=10}  OrganizationID: Number  <<FK>> EducationLevelID: Number {key=FK, reference= Education Level}  <<FK>> EducationMajorID: Number {key=FK, reference= Education Major}  <<FK>> RetirementStatusID: Number {key=FK, reference=Retirement Status}  CareerStartDate: Unicode String {size=20}  <<FK>> YearsBusExp: Number {key=FK, reference= Years Experience}  <<FK>> YearsITExp:Number {key=FK, reference= Years Experience}  CertificateName: Unicode String {size=90}  UseEmailforICCPBus: Bit  UseEmailforProfComms: Bit  <<FK>> CBIPMailingAddress: Number {key=FK, reference=Address Type}  <<FK>> TDWIMailingAddress: Number {key=FK, reference=Address Type}  <<FK>> ICCPMailingAddress: Number {key=FK, reference=Address Type}  <<FK>> RelatedMailingAddress: Number {key=FK, reference=Address Type}  AppliedforCBIPCert: Bit  AppliedforICCPCert:Bit  PassedICCPExam: Bit  DesignationDate: Unicode String {size=20}  DisabilityCheck: Bit  <<FK>> TDWIMembershipPrefID: Number {key=FK, reference=TDWI Membership Pref}  OrganizationName: Unicode String {size=20}  JobPosition: Unicode String {size=50}  JobPositionDesc: Unicode String {size=50}  EmploymentDate: Unicode String {size=20} |

|  |
| --- |
| Customer Address |
| <<PK>> CustomerAddressID:Number {seed=1, increment=1}  <<FK>> CustomerID: Number {key=FK, reference=Customer}  OrgLocationID: Number  <<FK>> AddressTypeID: Number {key=FK, reference= Address Type}  Address: Unicode String {size=50}  Address2: Unicode String {size=50}  Country: Unicode Character {size=5}  Region: Unicode Character {size=5}  City: Unicode String {size=40}  PostalCode: Unicode String {size=10}  CompanyName: Unicode String {size=50}  CompanyEmail: Unicode String {size=70}  CompanyPhone: Unicode String {size=30}  CreateDate: DateTime  UpdateDate: DateTime  PreferredAddress: Bit |

|  |
| --- |
| Customer Notes |
| <<PK>> CustomerNoteID:Number {seed=1, increment=1}  <<FK>> CustomerID: Number {key=FK, reference=Customer}  NoteText: Unicode String {size=1100}  DateAdded: DateTime |

|  |
| --- |
| Customer Email |
| <<FK>> CustomerID: Number {key=FK, reference=Customer}  <<FK>> AddressTypeID: Number {key=FK, reference= Address Type}  EmailAddress: Unicode String {size=70}  PreferredEmail: Bit |

|  |
| --- |
| Customer Phone |
| <<FK>> CustomerID: Number {key=FK, reference=Customer}  <<FK>> AddressTypeID: Number {key=FK, reference= Address Type}  PhoneNumber: Unicode String {size=30}  PreferredPhone: Bit |

|  |
| --- |
| Customer Role |
| <<FK>> CustomerID: Number {key=FK, reference=Customer}  <<FK>> PersonRoleID: Number {key=FK, reference= Membership Roles} |

|  |
| --- |
| Order |
| <<PK>> OrderID: Number {seed=1, increment=1}  <<FK>> CustomerID:Number {key=FK, reference=Customer}  <<FK>> AddressID: Number {key=FK, reference=Customer Address}  <<FK>> AddressTypeID: Number {key=FK, reference=Address Type}  OrderTotal: Currency  ShipTotal: Currency  <<FK>> StatusID: Number {key=FK, reference=Status}  CreateDate: DateTime  UpdateDate:DateTime |

|  |
| --- |
| Order Notes |
| <<FK>> OrderID: Number {key=FK, reference=Order}  CreateDate: DateTime  NoteText: Unicode String {size=90} |

|  |
| --- |
| Order Preview |
| <<FK>> CustomerID: Number {key=FK, reference=Customer}  <<FK>> InventoryID: Number {key=FK, reference=Inventory}  ItemUnitPrice: Currency  ItemQTYOrdered: Number  OrderPrice: Currency  OrderNotes: Unicode String {size=50}  CreateDate: DateTime |

|  |
| --- |
| Order Details |
| <<PK>> OrderDetailID: Number {seed=1, increment=1}  <<FK>> OrderID: Number {key=FK, reference=Order}  <<FK>> InventoryID: Number {key=FK, reference=Inventory}  ItemUnitPrice: Currency  ItemQTYOrdered: Number  OrderPrice: Currency  OrderDate: DateTime  ItemQTYShipped: Number  ShipPrice: Currency  ShipDate: DateTime  <<FK>> ShipmentTypeID: Number {key=FK, reference= Shipment Type}  <<FK>> StatusID: Number {key=FK, reference= Status}  PurchaseOrderID: Unicode Character {size=10}  ExpirationDate: Unicode Character {size=10} |

|  |
| --- |
| Payment |
| <<PK>> PaymentID: Number {seed=1, increment=1 }  <<FK>> OrderID : Number {key=FK, reference=Order}  PaymentAmt: Currency  PaymentCurrency: Unicode Character {size=5}  PaymentDateRcvd: DateTime  <<FK>> PaymentTypeID: Number {key=FK, reference= Payment Type} |

|  |
| --- |
| Payment Apportionment |
| <<PK>> PaymentApportionID: Number {seed=1, increment=1 }  <<FK>> ApportionmentTypeID : Number {key=FK, reference=Payment Apportionment Type }  AmountApportioned: Currency  DateApplied: DateTime  <<FK>> PaymentID: Number {key=FK, reference= Payment } |

|  |
| --- |
| Organization |
| <<PK>><<FK>>OrganizationID: Number {reference=Customer}  <<FK>>Organization Type: Number {reference= [Organization Type]  Organization Name: string {size=50} |

|  |
| --- |
| Organization Location |
| <<PK>>OrgLocationID: Number  <<FK>>OrganizationID: Number {reference=Organization}  OrgLocationName: String{size=50}  OrgContactID: Number {reference= [Organization Contact]}  NumOfEmployees: Number {Default=’0’}  PreferredLocation: Number {Default=’0’} |

|  |
| --- |
| Organization Contact |
| <<PK>>OrgContactID: Number  <<FK>>OrganizationID: Number {reference=Organization}  <<FK>>OrgLocationID: Number {reference=Organization Location}  <<FK>>OrgCustomerID: Number {reference= Customer}  <<FK>>OrgPersonRole: Number {reference= [Organization Role]}  PhoneNumber: string {size=30}  EmailAddress: string {size=70} |

|  |
| --- |
| Organization Role |
| <<PK>>OrganizationRoleID: Number  OrganizationRole: string {size=70} |
| Organization Type |
| <<PK>>OrganizationTypeID: Number  OrganizationRole: string {size=70} |

Design Classes:

|  |
| --- |
| ICCP |
| +AddOrganization (NewCustomer: Organization): Boolean  +ViewOrganizations (): Organization [\*]  +ModifyOrganization (ActiveCustomer: Organization): Boolean  +ViewCountries (): Location [\*]  +SearchOrganization (CustomerID: int): Organization  +LookupOrganizationByLocation (CustomerID: int, OrgLocationID: int): Location [\*]  +ImportCustomerApplication(NewCustomerApplication: CustomerApplication):Integer  +CreateCustomer(NewCustomer:Customer):Boolean  +CreateInventory (NewInventoryItem: Product):Boolean  +RemoveInventory(InventoryID:Integer):Boolean  +LookUpInventory(InventoryID:Integer): Product  +UpdateInventoryItem(UpdatedInventory:Product):Boolean  +AddItemToPreview(NewOrderItem:Order):Integer  +AddItemToOrder(OrderItemList:Order):Boolean  +LookUpPerson(CustomerID:Integer):Customer  +UpdatePerson(UpdatedCustomer: Customer): Boolean  + LookupCandidatedata(): Customer [\*]  +UpdateCandidate( ActiveCustomer:Customer): Boolean |

|  |
| --- |
| CustomerApplications |
| +AddCustomerApplication(NewCustomerApplication: CustomerApplication):Integer |

|  |
| --- |
| Inventory |
| +AddInventory (NewInventoryItem: Product):Boolean  +DeleteInventory(InventoryID:Integer):Boolean  +FindInventory(InventoryID:Integer): Product  +ModifyInventoryItem(UpdatedInventory:Product):Boolean |

|  |
| --- |
| Orders |
| +AddItemToPreview(NewOrderItem:Order):Integer  +AddItemToOrder(OrderItemList:Order):Boolean |

|  |
| --- |
| Customers |
| +AddCustomer(NewCustomer:Customer):Boolean  +FindPerson(CustomerID:Integer):Customer  +ModifyPerson(UpdatedCustomer: Customer):Boolean  +LookupCandidate(CustomerID: Integer): Customer  +UpdateCandidatedata(ActiveCustomer: Customer): Boolean |

|  |
| --- |
| Customer |
| +CustomerID:Integer  +FirstName:String  +MiddleName:String  +LastName:String  +BirthDate: String  +Gender:String  +MembershipTypeID: Integer  +MembershipType: String  +PersonRoleID: String  +PersonRole: String  +StatusID: Integer  +AddressTypeID: Integer  +Addresses: Location [\*]  +CertificateName: String  +CareerStartDate: String  +EducationLevelID: Integer  +EducationMajorID: Integer  +EducationMajor:String  +EducationLevel:String  +YearsBusExpID: Integer  +YearsITExp: Integer  +RetirementStatusID: Integer  +EmailAddress: String  +EmailICCP:Byte  +EmailComms:Byte  +MailingCBIP: Integer  +MailingTDWI: Integer  +MailingICCP: Integer  +MailingRelated: Integer  +OrganizationName: String  +JobPosition: String  +JobPositionDesc: String  +EmploymentDate: String  +AppliedforCBIPCert: Byte  +AppliedforICCPCert: Byte  +PassedICCPExam: Byte  +DesignationDate: String  +DisabilityCheck: Byte  +TDWIMembershipPref: String  +TDWIMembershipPrefID: Integer  +CustomerNote: String  +PrefEmailAddress: String |

|  |
| --- |
| Order |
| +CustomerID: Integer  +CustomerName: String  +AddressTypeID: Integer  +AddressID: Integer  +InventoryID: Integer  +ItemQTYOrdered: Integer  +ItemUnitPrice: Decimal  +OrderPrice: Decimal  +OrderNote: String  +OrderTotal: Decimal  +OrderDate: String  +OrderDetailID: Integer  +Status: String  +StatusID: Integer |

|  |
| --- |
| Product |
| +InventoryID: Integer  +ItemName: String  +Description: String  +UnitPrice: Double  +CategoryID: Integer  +Category: String  +Status: String  +StatusID: Integer |

|  |
| --- |
| Location |
| +AddressTypeID: Number  +CompanyName: string  +Address: string  +Address2: string  +CountryCode: string  +CountryName: string  +RegionCode: string  +RegionName: string  +City: string  +PostalCode: string  +EmailAddress: string  +PhoneNumber: string  +PreferredAddress: bool  +OrgLocationID: Number  +OrgCustomerID: Number  +OrgLocationName: string  +OrgContactID: Number  +NumberOfEmployees: Number  +PreferredLocation: bool |

|  |
| --- |
| Organization |
| +OrganizationID: Number  +CustomerID: Number  +OrgName: string  +OrgTypeID: Number  +OrgType: string  +StatusID: Number  +OrgContactID: Number  +OrganizationRoleID: Number  +OrganizationRole: string  +EmailAddress: string  +PrimaryContact: string  +PhoneNumber: string |

|  |
| --- |
| Organizations |
| +CreateOrganization (newCustomer: Organization): Boolean  +UpdateOrganization (activeCustomer: Organization): Boolean  +LookupOrganization (customerID: number): Organization  +GetOrganizations (): Organization [\*]  +GetOrganizationID (OrgName: string): number  +GetOrganizationLocationID (OrganizationID: number): Int  +GetCountries: Location [\*]  +GetRegions: Location [\*]  +GetOrganizationType: Organization [\*]  +GetOrganizationRole: Organization [\*] |

Implementation

4.1) Software Architecture – architecture representation

* + 1. Architecture Representation

Architecturally Significant Use cases for Back office management system is maintain customer, maintain order, maintain inventory, maintain credentials, maintain calendar and maintain organization. The entire system relies on the functionality and the relationship between Customer, Admin, inventory, order, credentials, calendar. We have grouped all the use cases under these five ones. All system maintain customer, maintain order, maintain inventory, maintain credentials, maintain calendar and maintain organization use cases entail creating, modifying, viewing details and viewing statistics. Presently ICCP facing two major isolated system that are Backoffice management system and Time management system. For the solution of ICCP system that is Integration of BOS and TMS.

Architecturally Significant nodes and physical configuration between nodes

Internet

Data server

(SQL server 2017)

web server

Internet

User

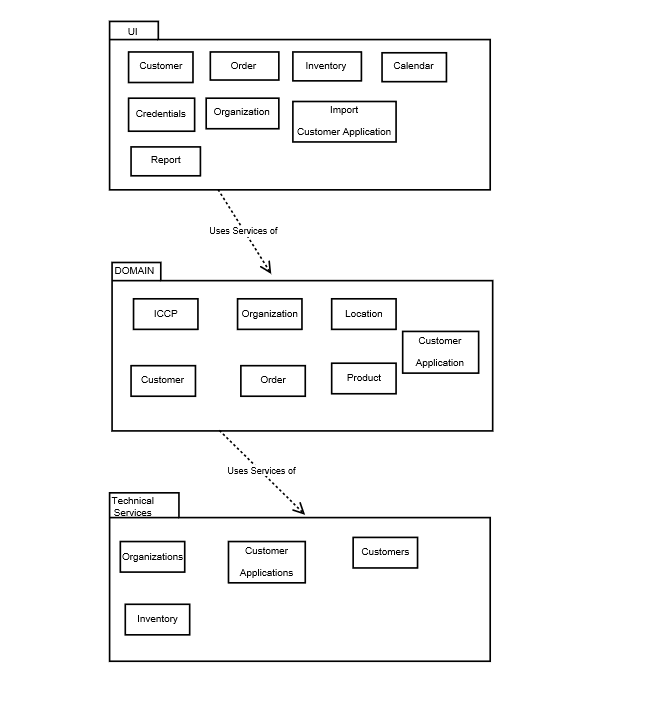
Discussions & Motivations:

The database will be run at a data server that ICCP will be providing that includes BOS, TMS, Online exam presentation and will be implemented in SQL Server 2017. The web server will hold the application code for the Back-office Management system and Test Management system. The application will be written using ASP.NET Core 3.1 with razor pages and database SQL Server. The web server will also be provided by ICCP.

4.1.2) Use case view

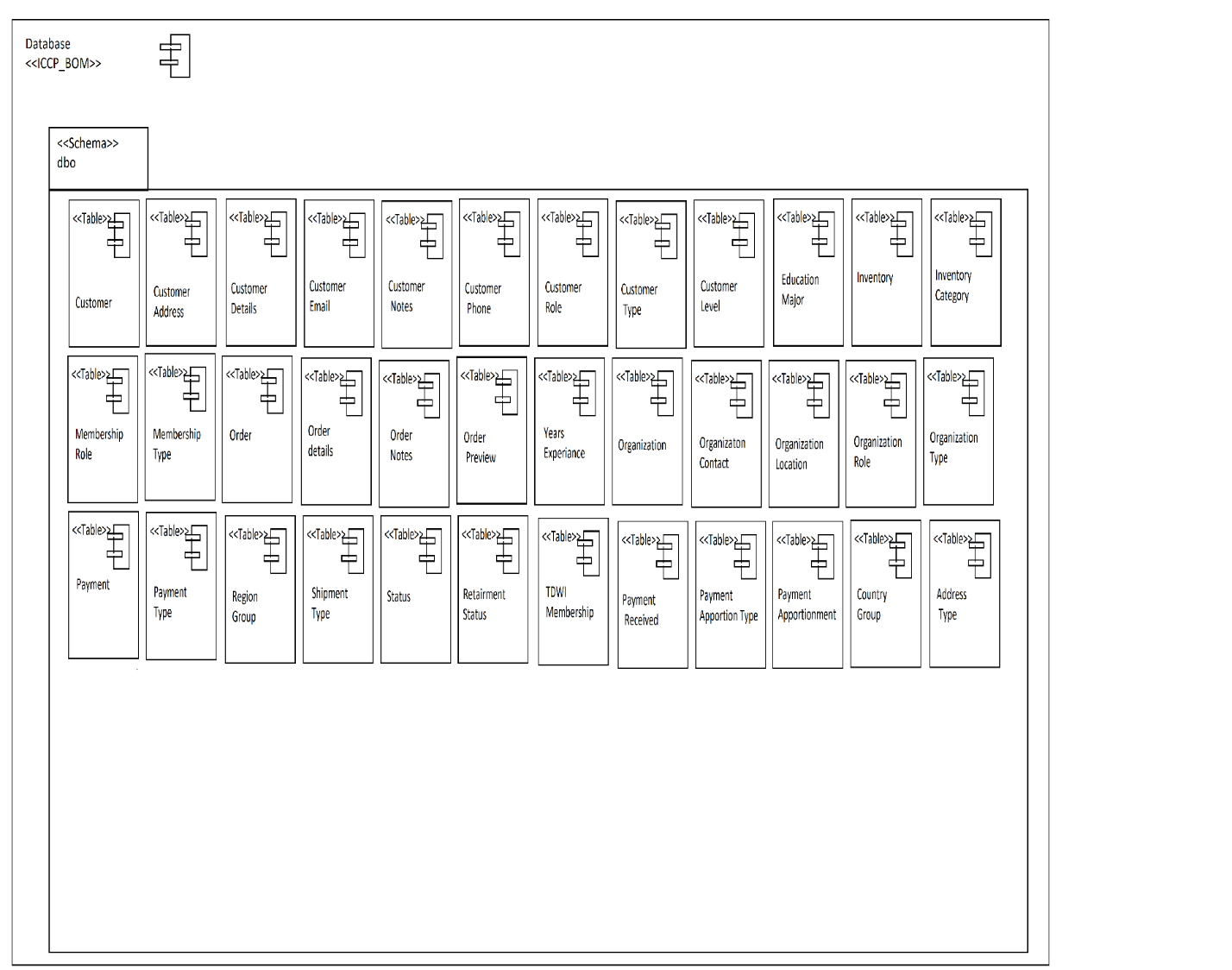


4.1.3) Logical view



4.2) Implementation Model

4.2.1) Component Diagram

****

|  |
| --- |
| <<Stored Procedure>> dbo.storedprocedure |
| GetEducationDegree()  GetEducationMajor()  GetYearsExperience()  GetTDWIMembershipPref()  GetCountriesList()  GetRegionsList()  spImportApplication()  spAddPersonLocation()  GetMembershipType()  GetMembershipRoles()  GetOrganization()  spCreatePerson()  spAddPersonLocation()  spAddCustomerNotes()  spAddPreferredEmail()  spLookupcandidatedata()  spUpdateCandidateData()  GetInventoryList()  GetCategoryList()  spCreateInventoryItem()  GetAllInventory()  spUpdateInventoryItem()  RemoveInventoryItem()  GetInventoryDetails()  spCreateOrganization()  spUpdateOrganization()  spAddOrganizationLocation()  spUpdateOrganizationLocation()  spUpdateOrganizationContact()  GetRegions()  GetCountries()  GetOrganizationRole()  GetOrganizationType()  spLookupOrganizationByValue()  GetOrganizationID()  spCreateOrder()  spAddItemToPreview()  spAddItemToOrder()  spAddPaymentToOrder()  spAddOrderNotes() |



Test

5.1) Test Model

5.1.1) Test Case: ImportCustomerApplication – Cindy submits customer application form of Navjot Deol.

Use Case: ImportCustomerApplication

**Input**

Test Procedure: ImportCustomerApplication – Cindy submits customer application form of Navjot Deol.

Enter URL in Internet Browser: <http://localhost/BOM/ImportCustomerApplication>

Enter the following values:

First Name: Navjot

Last Name: Deol

Middle Name: Kaur

Organization Name: NAIT

Job Position: Analyst

Job Description: Technical Solutions

Employment Date: 2020-05-04

Education Degree: Some College (selected from drop down list)

Education Major: Business (selected from drop down list)

Years Business Exp.: 2-4 Years (selected from drop down list)

Years IT Exp.: 2-4 Years (selected from drop down list)

Email Address: [deol@yahoo.com](mailto:deol@yahoo.com)

Phone Number: 587-090-1234

Street Address: 707

Unit Number: 40

Country: Canada (selected from drop down list)

Region/Province: Alberta (selected from drop down list)

City: Edmonton

Zip/Postal Code: T6G 1P8

Email Address: [grewal@yahoo.com](mailto:grewal@yahoo.com)

Phone Number: 587-090-1234

Street Address: 104

Unit Number: 67

Country: Canada (selected from drop down list)

Region/Province: Alberta (selected from drop down list)

City: Edmonton

Zip/Postal Code: T6G 1P8

Use Home Email for ICCP Business: Yes

Use Home Email for Prof. Communications: Yes

For CBIP mailing use: Home Address (selected from drop down list)

For TDWI mailing use: Home Address (selected from drop down list)

For ICCP mailing use: Home Address (selected from drop down list)

For related mailing: Home Address (selected from drop down list)

Previously filed for CBIP Certification with TDWI?: Yes

Previously filed for Certification with ICCP? Yes

Passed an ICCP exam and received a designation? Yes

Designation Received: 2020-03-02

Disability Check: No

TDWI Membership: I would like to become a TDWI member (selected from drop down list)

Click Submit Application Button

Message “Customer Application is Submitted successfully” displayed

Verify Results

**Result**

Database: ICCP\_BOM Table: Customer

|  |  |  |
| --- | --- | --- |
| CustomerID | CustomerTypeID | CustomerName |
| 1 | 1 | Navjot Deol |

-New rows created

Database: ICCP\_BOM Table: Customer Details

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CustomerID | FirstName | LastName | EducationLevelID | EducationMajorID |
| 1 | Navjot | Deol | 1 | 1 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| YearsBusExp | YearsITExp | CertificateName | UseEmailforICCP | UseEmailforProfComms |
| 2 | 2 | Navjot Deol | 1 | 1 |

|  |  |  |  |
| --- | --- | --- | --- |
| CBIPMailingAddress | TDWIMailingAddress | ICCPMailingAddress | RelatedMailingAddress |
| 1 | 1 | 1 | 1 |

|  |  |  |  |
| --- | --- | --- | --- |
| AppliedForCBIPCert | AppliedForICCPCert | PassedICCPExam | DisabilityCheck |
| 1 | 1 | 1 | 0 |

|  |  |  |  |
| --- | --- | --- | --- |
| TDWIMembershipPref | OrganizationName | Job Position | JobPositionDesc |
| 2 | NAIT | Analyst | Technical Solutions |

|  |  |
| --- | --- |
| EmploymentDate | DesignationDate |
| 2020-05-04 | 2020-03-02 |

-New rows created

Database: ICCP\_BOM Table: Customer Address

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CustomerAddressID | CustomerID | AddressTypeID | Address | Address2 | Country |
| 1 | 1 | 1 | 707 | 40 | CA |
| 2 | 1 | 2 | 104 | 67 | CA |

|  |  |  |  |
| --- | --- | --- | --- |
| City | Region | CompanyName | PostalCode |
| Edmonton | Alberta |  | T6G 1P8 |
| Edmonton | Alberta | IT Solutions | T6G 1P8 |

-New rows created

Database: ICCP\_BOM Table: Customer Email

|  |  |  |
| --- | --- | --- |
| CustomerID | AddressTypeID | EmailAddress |
| 1 | 1 | deol@yahoo.com |
| 1 | 2 | grewal@yahoo.com |

-New rows created

Database: ICCP\_BOM Table: Customer Phone

|  |  |  |
| --- | --- | --- |
| CustomerID | AddressTypeID | PhoneNumber |
| 1 | 1 | 587-090-1234 |
| 1 | 2 | 587-090-1234 |

-New rows created

**Conditions**

Workstation WB206-12

* Test Procedures

5.1.2) Test Case: CreatePerson – Cindy, ICCP member wants to create Navjot Deol as new Customer.

UseCase: CreatePerson

**Input**

Customer.cshtml control values:

First Name: Navjot

Last Name: Deol

Middle Name: Kaur

Membership Type: Candidate (selected from drop down list)

Customer Roles: Employee (selected from drop down list)

Organization: NAIT (selected from drop down list)

Email Address: [deolnavjot@yahoo.com](mailto:deolnavjot@yahoo.com)

Phone Number: 587-090-1234

Street Address: 001 01 Street

Unit Number: 12

Country: Canada (selected from drop down list)

Region/Province: Alberta (selected from drop down list)

City: Edmonton

Zip/Postal Code: T7G 1P6

Customer Notes: I want to be added as new person

**Result**

**Conditions**

Workstation WB206-12

When completed, Cindy submits the AddPerson request to the System

Navjot, a customer of ICCP wants to place order for 2 CAD inventory items. For this, Navjot must enter CustomerID and provides the following:

Inventory Item Name: CAD (selected from the drop-down list)

Quantity: 2

Inventory Item Name: ABMP

Quantity:3

When completed, Navjot submits the AddToPreview request to the system.

Use Case: Create Inventory

Primary scenario: Cindy, member of ICCP wants to add new Inventory Item Data Analytics under Self Study category.

Cindy, a member of ICCP wants to add new Inventory item to the database. For this, Cindy provides the following:

Name: Data Analytics

Description: Analysis of Data Sets

Unit Price: $125.00

Category: Self Study (selected from the drop-down list)

When completed, Cindy submits the CreateNewItem

Test Case: AddOrganization – Kedre Gurre submits Organization form of NAIT.

**Input**

Organization.cshtml control values:

* Organization Name: NAIT
* Organization Type: University/College
* Primary Contact: 7804716248
* Organization Role: Education/ Teacher
* OtherOrgRole: N/A
* PrimaryContact: Phone
* Email Address: Nait@nait.ca
* Phone Number: 7804716248
* Billing Street Address: 11762 106 Str
* Billing Unit Number: 12345
* Billing City: Edmonton
* Billing Zip/Postal Code: T5G 241
* Billing Country: Canada
* Billing Region/Province: Alberta
* Shipping Street Address: 11762 106 Str
* Shipping Unit Number: 12345
* Shipping City: Edmonton
* Shipping Zip/Postal Code: T5G 241
* Shipping Country: Canada
* Shipping Region/Province: Alberta

**Result**

Database: ICCP\_BOM                                                    Table: Organization

|  |  |  |
| --- | --- | --- |
| OrganizationID | OrganizationTypeID | OrganizationName |
| 1 | 6 | NAIT |

-New rows created

Database: ICCP\_BOM                                                    Table: [Organization Location]

|  |  |  |
| --- | --- | --- |
| OrganizationLocationID | OrganizationID | OrgLocationName |
| 1 | 1 | 12345(Edmonton, AB, Canada) |

-New rows created

Database: ICCP\_BOM                                                    Table: Customer

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CustomerID | CustomerTypeID | CustomerName | CreateDate | UpdateDate | StatusID |
| 1 | 1 | 12345(Edmonton, AB, Canada) | 2020-03-12 13:00:09.400 | 2020-03-12 13:00:09.417 | 1 |

|  |  |
| --- | --- |
| CreateDate | UpdateDate |
| 2020-03-12 13:00:09.417 | 2020-03-12 13:00:09.417 |
| 2020-03-12 13:00:09.417 | 2020-03-12 13:00:09.417 |

-New rows created

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Org  ContactID | OrganizationID | OrgLocationID | OrgCustomerID | OrgPersonRoleID |
| 1 | 1 | 1 | 1 | 8 |

|  |  |  |  |
| --- | --- | --- | --- |
| OtherOrgRole | PrimaryContact | Phone | Email |
| N/A | Phone | 7804716248 | Nait@nait.ca |

Database: ICCP\_BOM                                                    Table: [Customer Address]

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CustomerAddressID | CustomerID | OrgLocationID | AddressTypeID | Address1 | Address2 |
| 1 | 1 | 1 | 4 | 12345 | 11762 106 Str |
| 2 | 1 | 1 | 5 | 12345 | 11762 106 Str |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Country | Region | City | PostalCode | CreateDate | UpdateDate |
| CA | AB | Edmonton | T5G 241 | 2020-03-12 13:00:09.417 | 2020-03-12 13:00:09.417 |
| CA | AB | Edmonton | T5G 241 | 2020-03-12 13:00:09.417 | 2020-03-12 13:00:09.417 |

**Conditions**

Workstation WB206-12

* Test Procedures

Test Procedure: AddOrganization – Kedre Gurre submits request to add Organization NAIT

Test Case: AddOrganization – Kedre submits Organization form of NAIT.

Enter URL in Internet Browser: http://localhost/BOM/Organization

Enter the following values:

* Organization Name: NAIT
* Organization Type: University/College
* Primary Contact: 7804716248
* Organization Role: Education/ Teacher
* OtherOrgRole: N/A
* PrimaryContact: Phone
* Email Address: Nait@nait.ca
* Phone Number: 7804716248
* Billing Street Address: 11762 106 Str
* Billing Unit Number: 12345
* Billing City: Edmonton
* Billing Zip/Postal Code: T5G 241
* Billing Country: Canada
* Billing Region/Province: Alberta
* Shipping Street Address: 11762 106 Str
* Shipping Unit Number: 12345
* Shipping City: Edmonton
* Shipping Zip/Postal Code: T5G 241
* Shipping Country: Canada
* Shipping Region/Province: Alberta

           Click Add Organization Button

           Message “Organization added” is displayed

           Verify Results

Use Case: AddOrganization

Test Case: AddOrganization – Kedre Gurre submits Organization form of NAIT.

**Input**

Organization.cshtml control values:

* Organization Name: NAIT
* Organization Type: University/College
* Primary Contact: 7804716248
* Organization Role: Education/ Teacher
* OtherOrgRole: N/A
* PrimaryContact: Phone
* Email Address: Nait@nait.ca
* Phone Number: 7804716248
* Billing Street Address: 11762 106 Str
* Billing Unit Number: 12345
* Billing City: Edmonton
* Billing Zip/Postal Code: T5G 241
* Billing Country: Canada
* Billing Region/Province: Alberta
* Shipping Street Address: 11762 106 Str
* Shipping Unit Number: 12345
* Shipping City: Edmonton
* Shipping Zip/Postal Code: T5G 241
* Shipping Country: Canada
* Shipping Region/Province: Alberta

**Result**

Database: ICCP\_BOM                                                    Table: Organization

|  |  |  |
| --- | --- | --- |
| OrganizationID | OrganizationTypeID | OrganizationName |
| 1 | 6 | NAIT |

-New rows created

Database: ICCP\_BOM                                                    Table: [Organization Location]

|  |  |  |
| --- | --- | --- |
| OrganizationLocationID | OrganizationID | OrgLocationName |
| 1 | Navjot | 12345(Edmonton, AB, Canada) |

-New rows created

Database: ICCP\_BOM                                                    Table: Customer

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CustomerID | CustomerTypeID | CustomerName | CreateDate | UpdateDate | StatusID |
| 1 | Navjot | 12345(Edmonton, AB, Canada) | 2020-03-12 13:00:09.400 | 2020-03-12 13:00:09.417 | 1 |

|  |  |
| --- | --- |
| CreateDate | UpdateDate |
| 2020-03-12 13:00:09.417 | 2020-03-12 13:00:09.417 |
| 2020-03-12 13:00:09.417 | 2020-03-12 13:00:09.417 |

-New rows created

Database: ICCP\_BOM                                                    Table: [Customer Address]

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CustomerAddressID | CustomerID | OrgLocationID | AddressTypeID | Address1 | Address2 |
| 1 | 1 | 1 | 4 | 12345 | 11762 106 Str |
| 2 | 1 | 1 | 5 | 12345 | 11762 106 Str |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Country | Region | City | PostalCode | CreateDate | UpdateDate |
| CA | AB | Edmonton | T5G 241 | 2020-03-12 13:00:09.417 | 2020-03-12 13:00:09.417 |
| CA | AB | Edmonton | T5G 241 | 2020-03-12 13:00:09.417 | 2020-03-12 13:00:09.417 |

**Conditions**

Workstation WB206-12

* Test Procedures

Test Procedure: AddOrganization – Kedre Gurre submits request to add Organization NAIT

Test Case: AddOrganization – Kedre submits Organization form of NAIT.

Enter URL in Internet Browser: http://localhost/BOM/Organization

Enter the following values:

* Organization Name: NAIT
* Organization Type: University/College
* Primary Contact: 7804716248
* Organization Role: Education/ Teacher
* Email Address: Nait@nait.ca
* Phone Number: 7804716248
* Billing Street Address: 11762 106 Str
* Billing Unit Number: 12345
* Billing City: Edmonton
* Billing Zip/Postal Code: T5G 241
* Billing Country: Canada
* Billing Region/Province: Alberta
* Shipping Street Address: 11762 106 Str
* Shipping Unit Number: 12345
* Shipping City: Edmonton
* Shipping Zip/Postal Code: T5G 241
* Shipping Country: Canada
* Shipping Region/Province: Alberta

           Click Add Organization Button

           Message “Organization added” is displayed

           Verify Results

Test Cases

Use Case: ModifyOrganization

Test Case: ModifyOrganization – Kedre Gurre, Director of Member Services, ICCP wants to update an Organization to change BillingStreetAddress of NAIT to 11762 105 Str

**Input**

Organization.cshtml control values:

* Organization Name: NAIT
* Organization Type: University/College
* Primary Contact: 7804716248
* Organization Role: Education/ Teacher
* OtherOrgRole: N/A
* PrimaryContact: Phone
* Email Address: AskNait@nait.ca
* Phone Number: 7804716248
* Billing Street Address: 11762 106 Str
* Billing Unit Number: 12345
* Billing City: Edmonton
* Billing Zip/Postal Code: T5G 241
* Billing Country: Canada
* Billing Region/Province: Alberta
* Shipping Street Address: 11762 105 Str
* Shipping Unit Number: 12345
* Shipping City: Edmonton
* Shipping Zip/Postal Code: T5G 241
* Shipping Country: Canada
* Shipping Region/Province: Alberta

**Result**

Database: ICCP\_BOM                                                    Table: Organization

|  |  |  |
| --- | --- | --- |
| OrganizationID | OrganizationTypeID | OrganizationName |
| 1 | 6 | NAIT |

-New rows created

Database: ICCP\_BOM                                                    Table: [Organization Location]

|  |  |  |
| --- | --- | --- |
| OrganizationLocationID | OrganizationID | OrgLocationName |
| 1 | 1 | 12345(Edmonton, AB, Canada) |

-New rows created

Database: ICCP\_BOM                                                    Table: Customer

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CustomerID | CustomerTypeID | CustomerName | CreateDate | UpdateDate | StatusID |
| 1 | 1 | 12345(Edmonton, AB, Canada) | 2020-03-12 13:00:09.400 | 2020-03-12 13:00:09.417 | 1 |

|  |  |
| --- | --- |
| CreateDate | UpdateDate |
| 2020-03-12 13:00:09.417 | 2020-03-12 13:00:09.417 |
| 2020-03-12 13:00:09.417 | 2020-03-12 13:00:09.417 |

-New rows created

Database: ICCP\_BOM                                                    Table: [Customer Address]

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CustomerAddressID | CustomerID | OrgLocationID | AddressTypeID | Address1 | Address2 |
| 1 | 1 | 1 | 4 | 12345 | 11762 105 Str |
| 2 | 1 | 1 | 5 | 12345 | 11762 106 Str |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Country | Region | City | PostalCode | CreateDate | UpdateDate |
| CA | AB | Edmonton | T5G 241 | 2020-03-12 13:00:09.417 | 2020-03-12 14:00:09.417 |
| CA | AB | Edmonton | T5G 241 | 2020-03-12 13:00:09.417 | 2020-03-12 14:00:09.417 |

-New rows created

**Conditions**

Workstation WB206-12

* Test Procedures

Test Procedure: ModifyOrganization – Kedre Gurre submits request to Modify Organization NAIT

Test Case: ModifyOrganization – Kedre submits modified Organization form of NAIT.

Enter URL in Internet Browser: http://localhost/BOM/Organization

Enter the following values:

* Organization Name: NAIT
* Organization Type: University/College
* Primary Contact: 7804716248
* Organization Role: Education/ Teacher
* OtherOrgRole: N/A
* PrimaryContact: Phone
* Email Address: AskNait@nait.ca
* Phone Number: 7804716248
* Billing Street Address: 11762 105 Str
* Billing Unit Number: 12345
* Billing City: Edmonton
* Billing Zip/Postal Code: T5G 241
* Billing Country: Canada
* Billing Region/Province: Alberta
* Shipping Street Address: 11762 106 Str
* Shipping Unit Number: 12345
* Shipping City: Edmonton
* Shipping Zip/Postal Code: T5G 241
* Shipping Country: Canada
* Shipping Region/Province: Alberta

           Click Modify Organization Button

           Message “Organization Modified” is displayed

           Verify Results

Use Case: LookupCandidate Data

Test Case: Lookup Candidate Data – John search candidate data by providing customerID.

**Input**

CustomerID:

**Conditions**

Workstation WB206-12

* Test Procedures

Test Procedure: Lookup Candidate Data – john search candidate data by providing customerID.

Enter the following values:

CustomerID: 149

Click Submit Application Button

Message “Candidate data found successfully” displayed

Verify Results

Use Case: UpdateCandidate Data

Test Case: Update Candidate Data – John update candidate data .

**Input**

Certificate Name: Navjot

Organization Name: NAIT

Birthdate:1993-5-1

Gender: Female

Carrer start Date: 2020-05-04

Job Position: Analyst

Education Degree: Some College

Education Major: Business

Years Business Exp.: 2-4 Years

Years IT Exp.: 3-4 Years

**Conditions**

Workstation WB206-12

* Test Procedures

Test Case: Update Candidate Data – John update candidate data.

Enter the following values:

Certificate Name: Navjot

Organization Name: NAIT

Birthdate:1993-5-1

Gender: Female

Carrer start Date: 2020-05-04

Job Position: Analyst

Education Degree: Some College (selected from drop down list)

Education Major: Business (selected from drop down list)

Years Business Exp.: 2-4 Years (selected from drop down list)

Years IT Exp.: 2-4 Years (selected from drop down list)

Click Save Application Button

Message “Candidate Data Update is Submitted successfully” displayed.

Support – Project Management

6.1) Software Development Plan

6.1.1) Risk List

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Description | Priority | Impact | Mitigation Strategy/ Contingency Plan |
| Integrating BOS and TMS  (Architectural risk) | Architectural factors for Performance/ accuracy | Critical | On stakeholders  System will remain unstable | Started doing in the 1st iteration of the elaboration phase and is continued in the 1st iteration of construction |
| Exam Presentation System | Architectural factors for Performance/ accuracy | Critical | On stakeholders, Customers | Started doing in the 2nd iteration of the elaboration phase and is continued in the 1st iteration of construction |
| Conversion of Web forms to Razor Pages  (Technology Risk | Performance/  Maintainability | Critical | On stakeholders, Customers | Started conversion to razor pages. |

* + 1. Requirement Ranking List

|  |  |  |
| --- | --- | --- |
| Rank | Use Case/ Requirement | Comment |
| High | Create Customers | Risk: Integrating BOS and TMS  Coverage: Core functionality  Criticality: High business value |
| High | Update Customers | Risk: Integrating BOS and TMS  Coverage: Core functionality  Criticality: High business value |
| High | Import Customer Application | Risk: Integrating BOS and TMS  Coverage: Core functionality  Criticality: High business value |
| High | Create Inventory | Risk: Integrating BOS and TMS  Coverage: Core Functionality  Criticality: High business value |
| High | Update Inventory | Risk: Integrating BOS and TMS  Coverage: Core Functionality  Criticality: High business value |
| High | Create Orders | Risk: Integrating BOS and TMS  Coverage: Core Functionality  Criticality: High business value |
| High | Manage Orders | Risk: Integrating BOS and TMS  Coverage: Core Functionality  Criticality: High business value |
| High | Create Credentials | Risk: Integrating BOS and TMS  Coverage: Core Functionality  Criticality: High business value |
| High | Update Credentials | Risk: Integrating BOS and TMS  Coverage: Core Functionality  Criticality: High business value |
| High | Maintain Calendar | Risk: Integrating BOS and TMS  Coverage: Core Functionality  Criticality: High business value |
| High | Add Organization | Risk: Integrating BOS and TMS  Coverage: Core Functionality  Criticality: High business value |
| High | Modify Organization | Risk: Integrating BOS and TMS  Coverage: Core Functionality  Criticality: High business value |
| High | Take Exam | Risk: Exam Presentation System  Coverage: Core Functionality  Criticality: High business value |
| High | Conversion from web forms to razor pages | Risk: Conversion to Razor Pages  Coverage: Core Functionality  Criticality: High business value |

6.1.3) Iteration Plan

Iteration 1 (Actual)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Core Workflows/ Disciplines** | **Use Case/ Requirements** | **Activity/ Task** | **Time**  **(hours)** | **Actual Time (hours)** | **Resources(role)** |
| Requirement | ImportCustomerApplication  CreatePerson  AddOrganization  CreateInventory  UpdateCandidateData  ModifyCredentials | ImportCustomerApplication  Write Scenario  CreatePerson  Write Scenario  AddOrganization  Write Scenario  CreateInventory  Write Scenario  UpdateCandidateData  Write Scenario  ModifyCredentials  Write Scenario | 2  2  2  2  2  2 | 3  3  3  3  3  3 | 3 Use case Specifier |
| Analysis | ImportCustomerApplication  CreatePerson  AddOrganization  CreateInventory  UpdateCandidateData  ModifyCredentials | ImportCustomerApplication  Domain Model  CreatePerson  Domain Model  AddOrganization  Domain Model  CreateInventory  Domain Model  UpdateCandidateData  Domain Model  ModifyCredentials  Domain Model | 2  2  1  2  3  3 | 2  1  2  1  2  2 | 3 System analyst |
| Design | ImportCustomerApplication  CreatePerson  AddOrganization  CreateInventory  UpdateCandidateData  ModifyCredentials | ImportCustomerApplication  Interaction Diagram, Design Class diagram  CreatePerson  Interaction Diagram, Design Class diagram  AddOrganization  Interaction Diagram, Design Class diagram  CreateInventory  Interaction Diagram, Design Class diagram  UpdateCandidateData  Interaction Diagram, Design Class diagram  ModifyCredentials  Interaction Diagram, Design Class diagram | 1  1  3  1  3  2 | 2  2  3  2  3  3 | 3 Use case Engineer |
| Implement | ImportCustomerApplication  CreatePerson  AddOrganization  CreateInventory  UpdateCandidateData  ModifyCredentials | ImportCustomerApplication  Implementing Object Oriented Design  CreatePerson  Implementing Object Oriented Design  AddOrganization  Implementing Object Oriented Design  -CreateInventory  Implementing Object Oriented Design  UpdateCandidateData  Implementing Object Oriented Design  ModifyCredentials  Implementing Object Oriented Design | 6  10  8  3  4  6 | 15  10  15  5  3  12 | 3 Component Engineer |
| Test | ImportCustomerApplication  CreatePerson  AddOrganization  CreateInventory  UpdateCandidateData  ModifyCredentials | ImportCustomerApplication   * Testcase * Test Procedure * Execution of test   CreatePerson   * Testcase * Test Procedure * Execution of test   AddOrganization   * Testcase * Test Procedure * Execution of test   CreateInventory   * Testcase * Test Procedure * Execution of test   UpdateCandidateData   * Testcase * Test Procedure * Execution of test   ModifyCredentials   * Testcase * Test Procedure * Execution of test | 3  2  2  2  3  2 | 4  4  4  2  2  4 | 3 Test Engineer |
|  |  | **TOTAL:-** | 93 | 150 | 3 Use Case Specifiers  3 System Analysts  3 Use Case Engineers  3 Component Engineers  3 Test Engineers |

Phase: Elaboration Iteration:2 Start Date: Mar 16th , 2020 End Date: Mar 30th , 2020

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Core Workflows/ Disciplines** | **Use Case/ Requirements** | **Activity/ Task** | **Time**  **(hours)** | **Resources(role)** |
| Requirement | CreateOrder  ManageOrder  UpdateInventory  LookupInventory  AddCredentials  LookupOrder  LookupOrganization  ModifyOrganization | CreateOrder  Write Scenario  ManageOrder  Write Scenario  UpdateInventory  Write Scenario  LookupInventory  Write Scenario  AddCredentials  Write Scenario  LookupOrder  Write Scenario  LookupOrganization  Write Scenario  ModifyOrganization  Write Scenario | 2  2  2  2  2  2  2  2 | 3 Use case Specifier |
| Analysis | CreateOrder  ManageOrder  UpdateInventory  LookupInventory  AddCredentials  LookupOrder  LookupOrganization  ModifyOrganization | CreateOrder  Domain Model  ManageOrder  Domain Model  UpdateInventory  Domain Model  LookupInventory  Domain Model  AddCredentials  Domain Model  LookupOrder  Domain Model  LookupOrganization  Domain Model  ModifyOrganization  Domain Model | 2  4  3  2  3  3  3  3 | 3 System analyst |
| Design | CreateOrder  ManageOrder  UpdateInventory  LookupInventory  AddCredentials  LookupOrder  LookupOrganization  ModifyOrganization | CreateOrder  Interaction Diagram, Design Class diagram  ManageOrder  Interaction Diagram, Design Class diagram  UpdateInventory  Interaction Diagram, Design Class diagram  LookupInventory  Interaction Diagram, Design Class diagram  AddCredentials  Interaction Diagram, Design Class diagram  LookupOrder  Interaction Diagram, Design Class diagram  LookupOrganization  Interaction Diagram, Design Class diagram  ModifyOrganization  Interaction Diagram, Design Class diagram | 4  3  3  3  3  2  4  4 | 3 Use case Engineer |
| Implement | CreateOrder  ManageOrder  UpdateInventory  LookupInventory  AddCredentials  LookupOrder  LookupOrganization  ModifyOrganization | CreateOrder  Implementing Object Oriented Design  ManageOrder  Implementing Object Oriented Design  UpdateInventory  Implementing Object Oriented Design  LookupInventory  Implementing Object Oriented Design  AddCredentials  Implementing Object Oriented Design  LookupOrder  Implementing Object Oriented Design  LookupOrganization  Implementing Object Oriented Design  ModifyOrganization  Implementing Object Oriented Design | 8  8  5  4  8  6  8  8 | 3 Component Engineer |
| Test | CreateOrder  ManageOrder  UpdateInventory  LookupInventory  AddCredentials  LookupOrder  LookupOrganization  ModifyOrganization | CreateOrder   * Testcase * Test Procedure * Execution of test   ManageOrder   * Testcase * Test Procedure * Execution of test   UpdateInventory   * Testcase * Test Procedure * Execution of test   LookupInventory   * Testcase * Test Procedure * Execution of test   AddCredentials   * Testcase * Test Procedure * Execution of test   LookupOrder   * Testcase * Test Procedure * Execution of test   LookupOrganization   * Testcase * Test Procedure * Execution of test   ModifyOrganization   * Testcase * Test Procedure * Execution of test | 3  4  4  4  2  3  4  4 | 3 Test Engineer |
|  |  | **TOTAL:-** | 164 | 3 Use Case Specifiers  3 System Analysts  3 Use Case Engineers  3 Component Engineers  3 Test Engineers |