

# Reporting Tool System

Rajashree Majumder, Jingya Wang, Yixiao Gao

30th September 2020

## 1 Project Vision

The vision of the project is to create a system that can make the life of architects, civil engineers, and material engineers better by allowing them to purchase licence easily to gain information related to their field and also by allowing them to store the information for their convenience.

## 2 Project Requirements

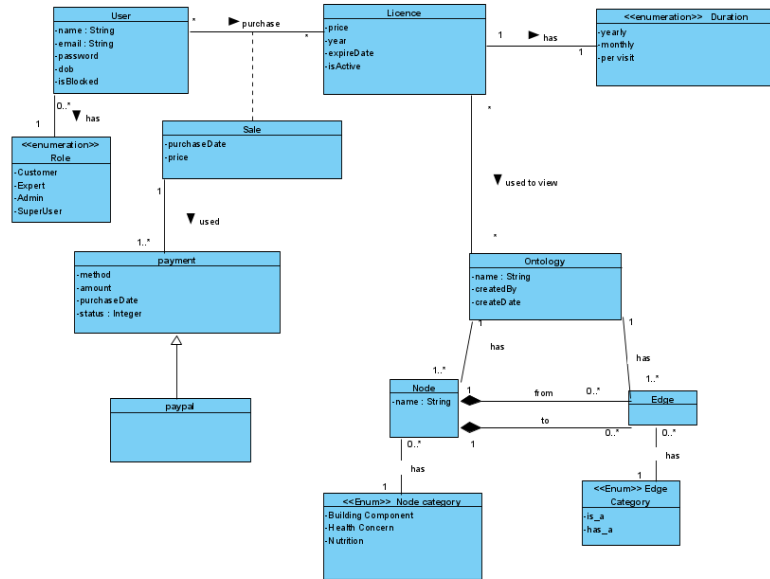
### 2.1 Functional Requirements

- Customer can register to the system.
- Customer can log in to the system.
- Customer will have three type of licenses: Monthly, yearly and per visit for choosing a license plan.
- Customer can purchase licenses.
- Customer will get email notification upon license approval.
- Customers can manage their own account.
- Customers can manage their licences.
- Customers can download the ontology's in an excel spreadsheet and content of the GD as PDF or excel sheet.

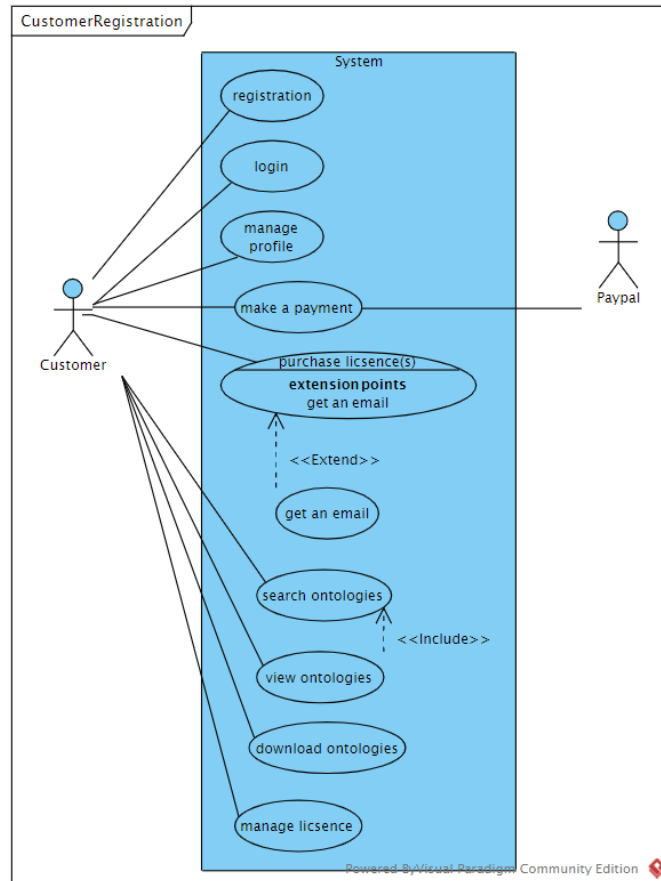
### 2.2 Non Functional Requirements

- System Admin must approve licences requested by customers so that only authorized customer's have access to the ontology's.
- Customer must have license to view the ontology's.

### 3 Domain Model



## 4 Use Case Model



## 5 Use Cases

### 5.1 Registration

#### 5.1.1 Scope

Reporting Tool

#### 5.1.2 Level

User Goal

### **5.1.3 Primary Actor**

User

### **5.1.4 Stakeholders and Interests**

User: Wants to register a new account

### **5.1.5 Preconditions**

User is new to this system.

### **5.1.6 Post conditions**

The user has an account of this system.

### **5.1.7 Main Success Scenario**

1. User opens the system.
2. User input his/her username and password.
3. The system records the username and password, then redirects the user to log in page

### **5.1.8 Extensions**

- 1-3 User is not connected to the internet.
1. System informs the user to connect to the internet.
  - 2a. The username exists.
    1. System tells the user to change another username.
  - 2b. The password is too simple.
    1. The system informs the user to set a more complex password.

## **5.2 Log In**

### **5.2.1 Scope**

Reporting Tool

### **5.2.2 Level**

User Goal

### **5.2.3 Primary Actor**

User

### **5.2.4 Stakeholders and Interests**

User: Wants to log in the system in a reliable way.

#### **5.2.5 Preconditions**

User has an account of this system.

#### **5.2.6 Post conditions**

The user successfully log into this system.

#### **5.2.7 Main Success Scenario**

1. User opens this system.
  2. User inputs his/her user name, password and verification code
  3. The system verifies the username, password and verification code, and then redirect the user into the home page of the system.

#### **5.2.8 Extensions**

- 1-3 User is not connected to the internet.
  1. System informs the user to connect to the internet.
- 2a. The username doesn't exist.
  1. System tells user to register an account.
- 2b. the password is incorrect.
  1. System tells user to try another password.
- 2c. The user resets the password.
  1. System asks the user to input the registration email.
  2. User inputs the new password.
- 2d. The verification code is incorrect.
  1. The system tells the user to input a new verification code.

### **5.3 Choosing a license Plan**

#### **5.3.1 Scope:**

Reporting Tool

#### **5.3.2 Level:**

User goal

#### **5.3.3 Primary Actor:**

User

#### **5.3.4 Stakeholders & Interests:**

- User: Wants to be able to get multiple options for purchasing a license. Wants to be able to select a particular license option as per his/her interest.

### **5.3.5 Precondition**

User must be registered to the system.

### **5.3.6 Success Guarantee**

User is successfully redirected to the payment Page.

### **5.3.7 Main Success Scenario**

1. User requests to view the available licences.
2. System redirect user to the license details page.
3. User selects a particular licence plan.
4. System redirect user to payment page.
5. User makes payment
6. System updates user's licence status.

### **5.3.8 Extensions**

\*3. User cancels the license purchase plan.

1. System redirects user back to the home page.

\*a. User wants to change the license type from yearly to monthly or per visit after payment.

1. User contacts system admin.

User gets refunded.

\*6. User license information is not updated in the system.

1. User contacts system admin.

User's licences are updated.

## **5.4 Make a Payment**

### **5.4.1 Scope**

Reporting Tool

### **5.4.2 Level**

User Goal

### **5.4.3 Primary Actor**

User

#### **5.4.4 Stakeholders and Interests**

User: Wants an accurate, reliable payment process.

#### **5.4.5 Preconditions**

User logs into the system.

#### **5.4.6 Post conditions**

The user successfully pay for the licences.

#### **5.4.7 Main Success Scenario**

1. User opens the system.
2. User selects a type of licences.
3. User selects the paypal as the payment method.
4. User inputs his/her paypal username and password to login the paypal sys-tem.
5. User make a payment.
6. The system redirects the user to the purchasing page.

#### **5.4.8 Extensions**

- 1-3 User is not connected to the internet.
  1. System informs the user to connect to the internet.
- 1-3 User cancels the payment.
  1. System redirects the user to the home page.
- 3a The user doesn't have an paypal account.
  1. The system tells the user to register a paypal account.
  2. The system tells the user to log in the paypal system.
- 3b. The payment process fails.
  1. The system informs the user the payment fails.
  2. The system redirects the user to the purchasing page.

### **5.5 Managing User Account**

#### **5.5.1 Scope:**

Reporting Tool

#### **5.5.2 Level:**

User goal

#### **5.5.3 Primary Actor:**

User

#### **5.5.4 Stakeholders & Interests:**

- User: Wants to be able to change his/her account information at any time and with less effort.

#### **5.5.5 Precondition**

User must be registered to the system.

#### **5.5.6 Success Guarantee**

User account information's were modified successfully.

#### **5.5.7 Main Success Scenario**

1. User requests for current account modification.
2. System returns user a form to modify the current account information.
3. Customer fills the form and submit it.
4. System updates the customer account information.

#### **5.5.8 Extensions**

\*3. Customer enters invalid inputs

1. System aborts the save operation and notifies customer about the invalid fields.

#### **5.5.9 Technology and Data Variations List**

\*7 Customer enters invalid inputs

1. Customer information is invalid like email address has wrong format and date of birth has invalid format.

### **5.6 Managing licenses**

#### **5.6.1 Scope:**

Reporting Tool

#### **5.6.2 Level:**

User goal

#### **5.6.3 Primary Actor:**

User



#### **5.6.4 Stakeholders & Interests:**

- User: Wants to be able to change his/her license preferences at any time and with less effort.

#### **5.6.5 Precondition**

User must be registered to the system.

#### **5.6.6 Success Guarantee**

User license is modified and user is redirected to the payment system.

#### **5.6.7 Main Success Scenario**

1. User requests to view his/her current license plan.
2. System returns user the current license type and it's expiration date.
3. User wants to renew current license.
4. System redirect user to payment page.
5. User makes payment.
6. System updates user license expiration date.

#### **5.6.8 Extensions**

\*a User changes license plan from monthly to per visit or yearly to monthly or per visit.

1. User contacts the admin.

- User is refunded

\*3 User wants to change license plan type.

1. User requests to view the available licences.
2. System redirect user to available license plans.
3. User a chooses new type of license.
4. System redirect user to the payment page.
5. User makes payment.
6. System updates user licence type in the system.

## **5.7 Search Ontologies**

### **5.7.1 Scope**

Reporting Tool System

### **5.7.2 Level**

User goal

### **5.7.3 Primary Actor**

Customer

### **5.7.4 Stakeholder and interests**

- Customer: person would like to search ontologies.
- Reporting Tool System: system would like to properly respond to customer.

### **5.7.5 Precondition**

Customer is logged in the system. Customer is not blocked.

### **5.7.6 Postcondition**

Return the search results(the list of ontologies) to the user.

### **5.7.7 Main success scenario**

1. Customer starts to search ontologies.
2. Reporting Tool System confirms customer's license is valid.
3. Customer submits the parameters for search.
4. Reporting Tool System return the list of ontologies to User.

### **5.7.8 Extensions**

- a\*. Anytime Reporting Tool System doesn't respond
  1. User will restart the system
- 2.a If customer doesn't have valid license,
  1. Reporting Tool System will send customer a message:"Sorry, you don't have valid license to search ontologies. Would you like to purchase one?"
  2. Reporting Tool System directs customer to the purchasing a licence page.
- 4.a Customer wants to redo a search.
  1. Reporting Tool System directs customer to the submitting parameter page.

## **5.8 View ontologies**

### **5.8.1 Scope**

Reporting Tool System

### **5.8.2 Level**

User Goal

### **5.8.3 Primary Actor**

Customer

### **5.8.4 Stakeholder and interests**

- Customer: person would like to view a specific ontology.
- Reporting Tool System: system would like to properly respond to customer.

### **5.8.5 Precondition**

Customer searched the ontologies.

### **5.8.6 Postcondition**

Return the ontology graph to the user.

### **5.8.7 Main success scenario**

1. Customer starts to view ontologies.
2. Customer selects an ontology to view.
3. Reporting Tool System returns the ontology details(graph) to customer.

### **5.8.8 Extensions**

- a\*. Anytime, User requests a cancel operation,
  1. Reporting Tool System discard the search and return to the search results page.
- 3.a User wants to view another ontology.
  1. Reporting Tool System returns to the search results page.

## **5.9 Download ontologies**

### **5.9.1 Scope**

Reporting Tool System

### **5.9.2 Level**

User goal

### **5.9.3 Primary Actor**

Customer

### **5.9.4 Stakeholder and interests**

- Customer: person would like to download ontologies.
- Reporting Tool System: system would like to properly respond to customer.

### **5.9.5 Precondition**

Customer searched ontologies.

### **5.9.6 Postcondition**

An excel spreadsheet contains ontologies is ready for downloading.

### **5.9.7 Main success scenario**

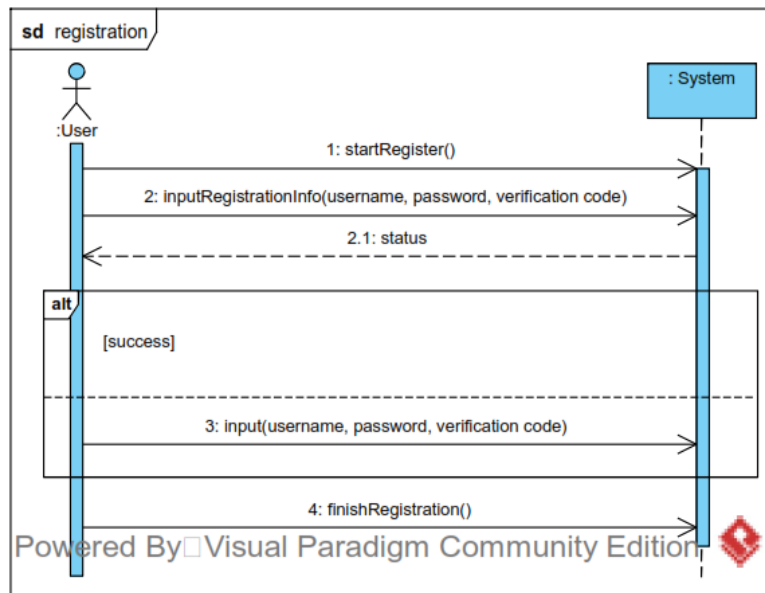
1. Customer starts to download ontologies.
2. Customer selects ontologies to download.
3. Reporting Tool System returns an excel spreadsheet that contains selected ontologies to customer.

### **5.9.8 Extensions**

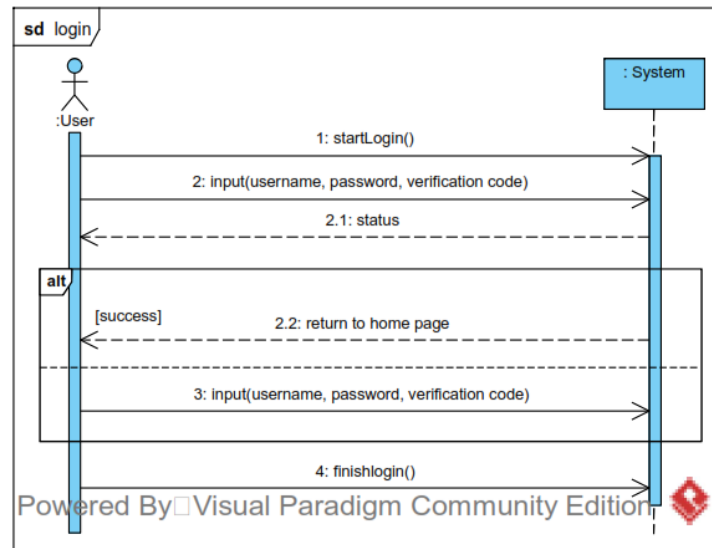
- a\*. Anytime, customer wants to cancel downloading.
  1. Reporting Tool System discards the excel file.
  2. Reporting Tool System discards returns to the search results page.
- \*.a Anytime, Reporting Tool System has no response.
  1. Customer restarts the system.

## 6 System Sequence Diagram's

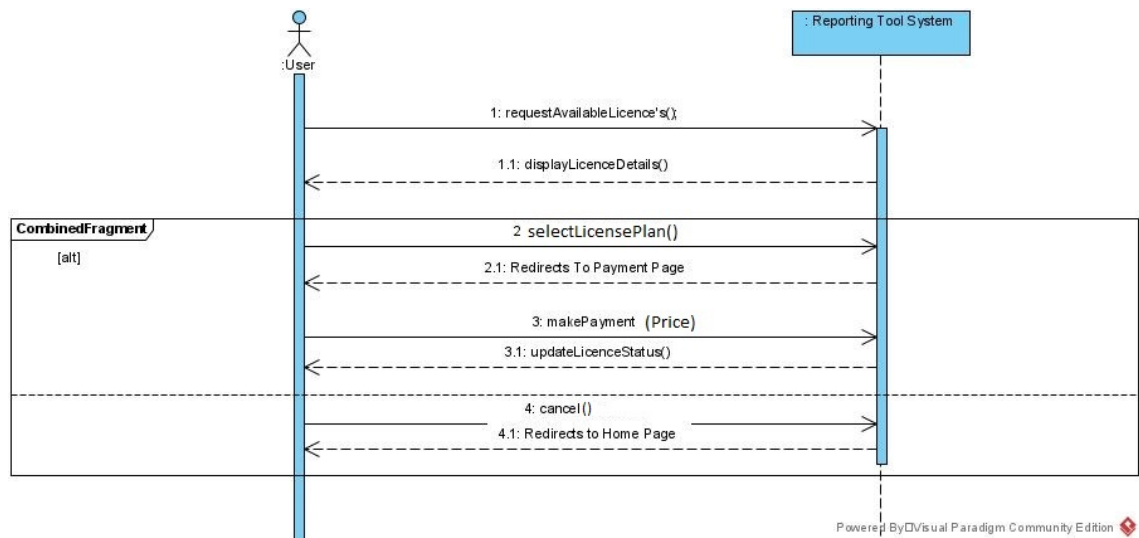
### 6.1 Registration



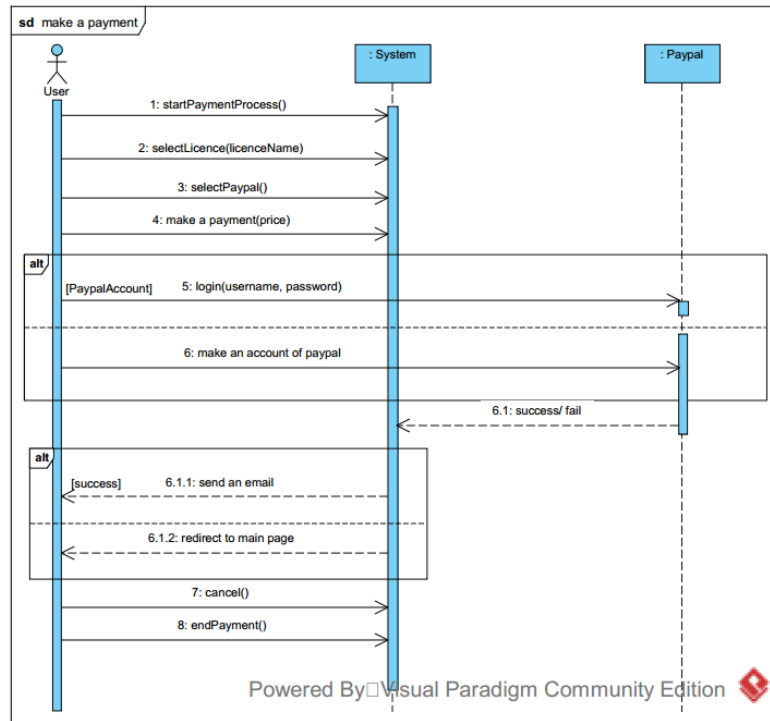
## 6.2 Login



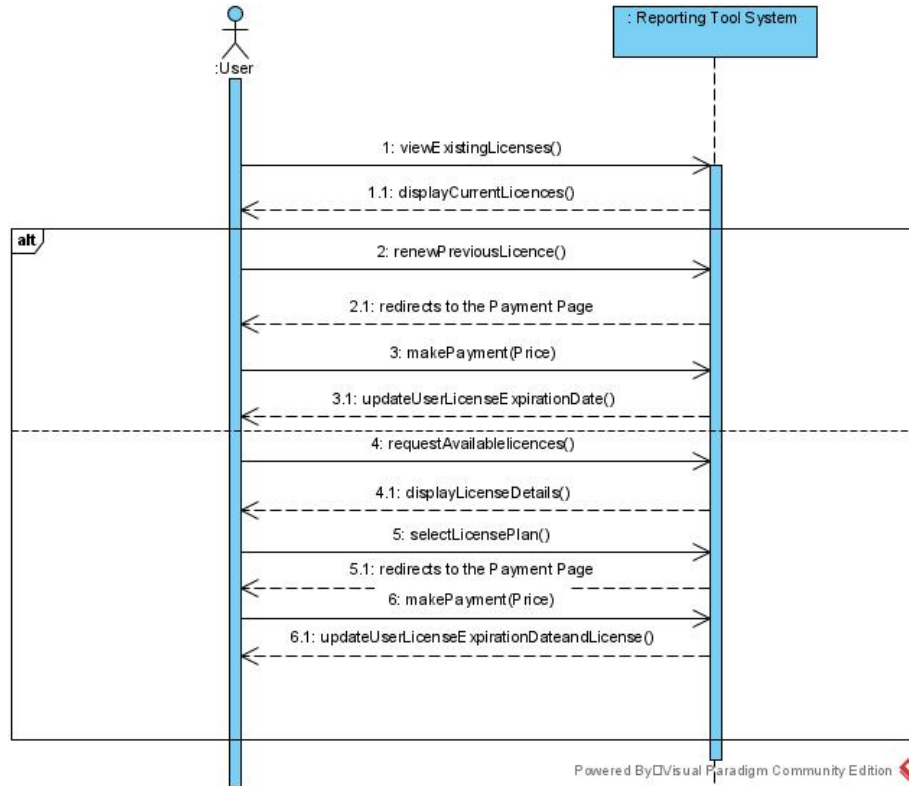
## 6.3 Choosing a License Plan



## 6.4 Make Payment

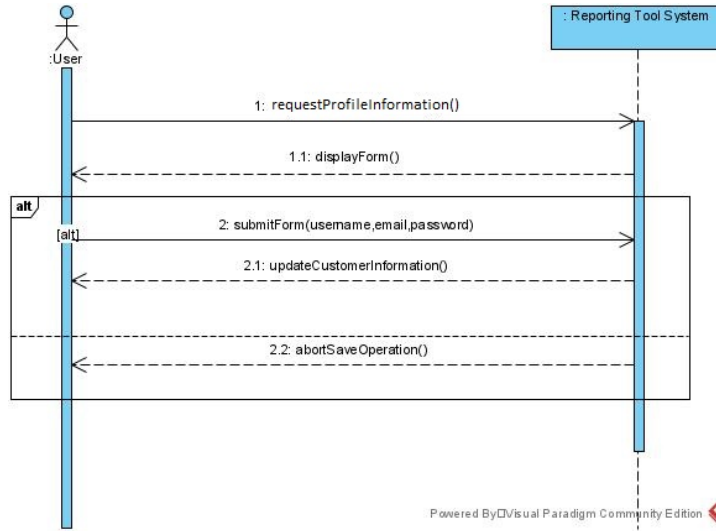


## 6.5 Manage license

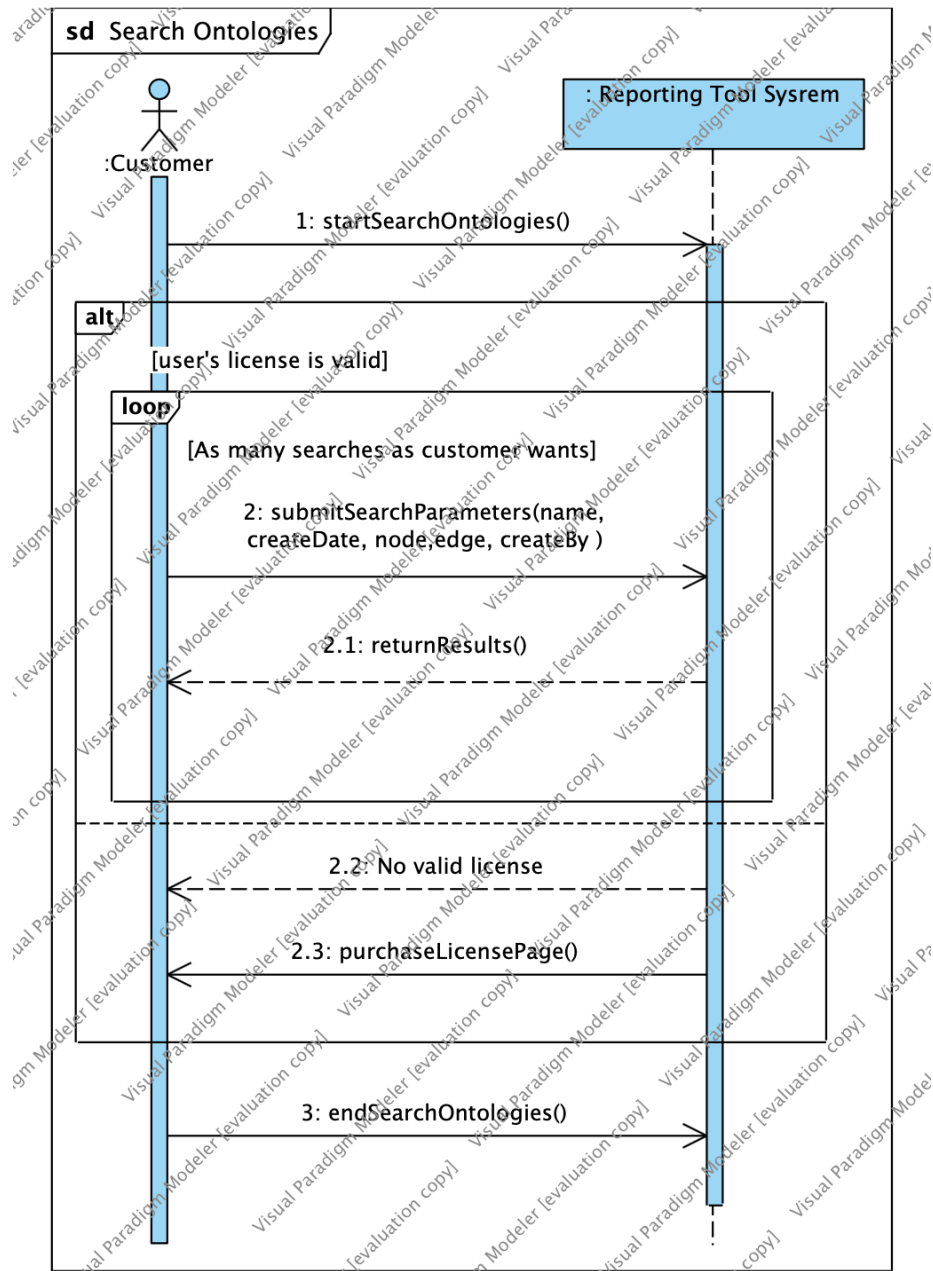




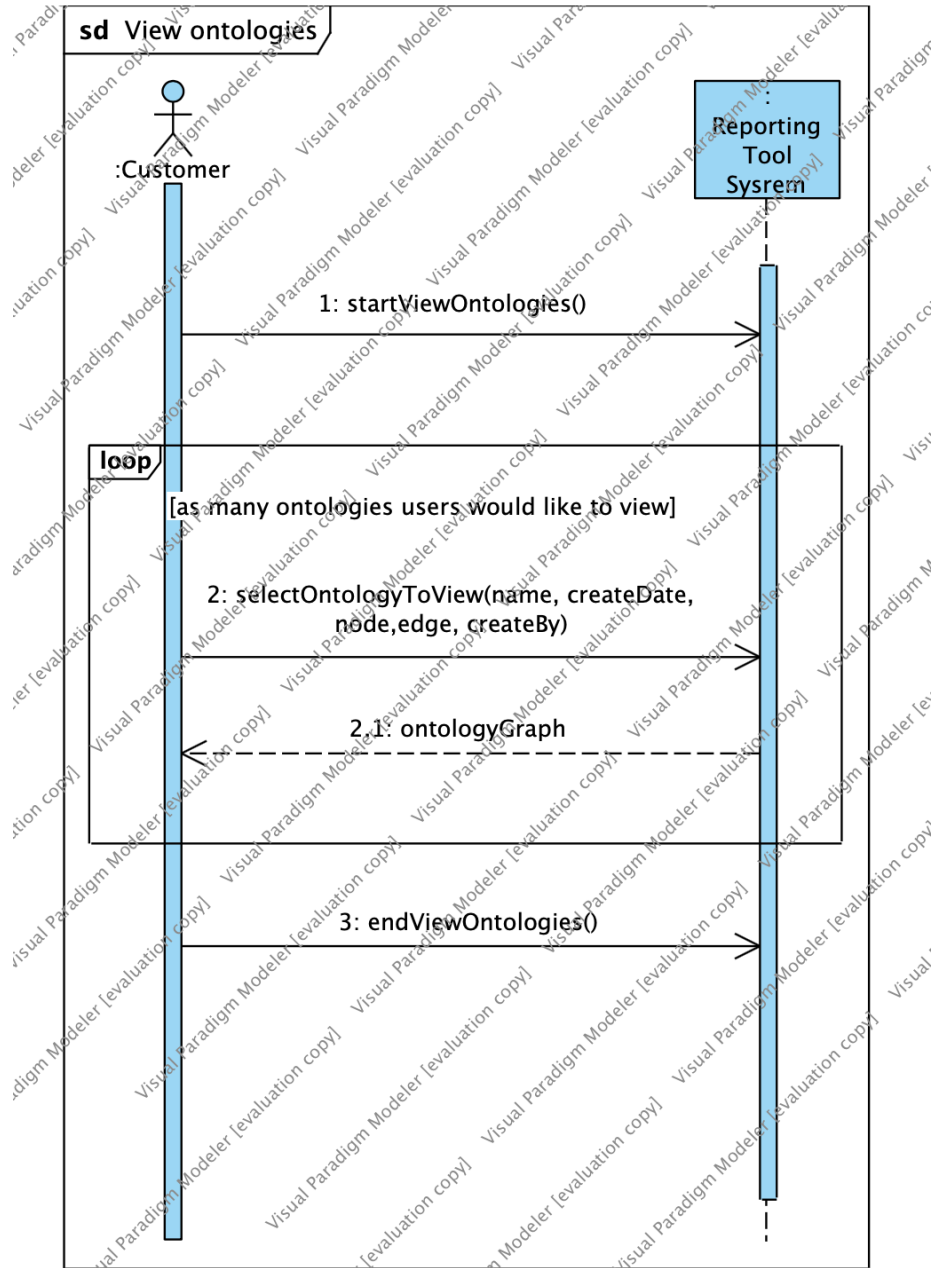
## 6.6 Manage Profile



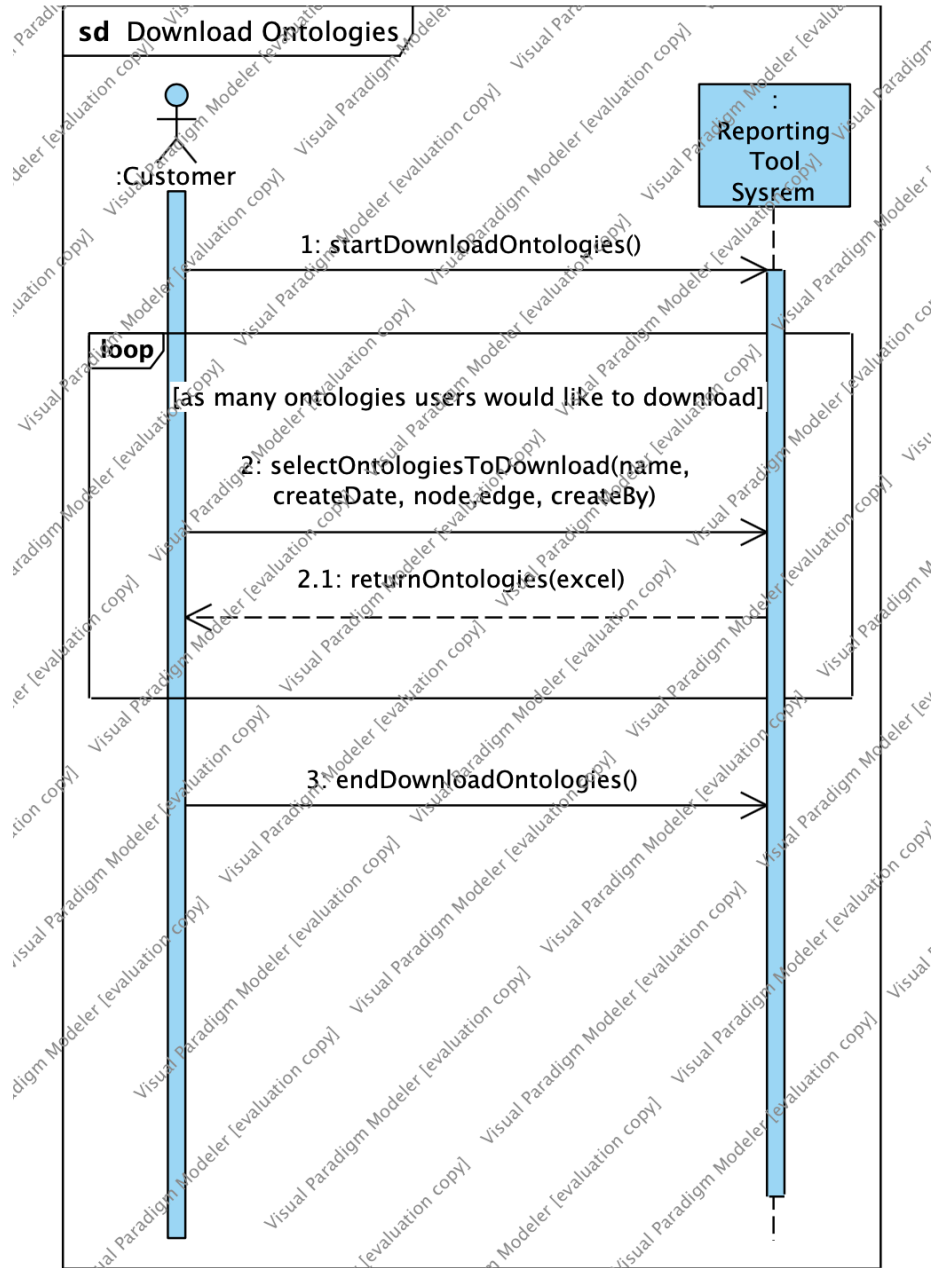
## 6.7 Search Ontologies



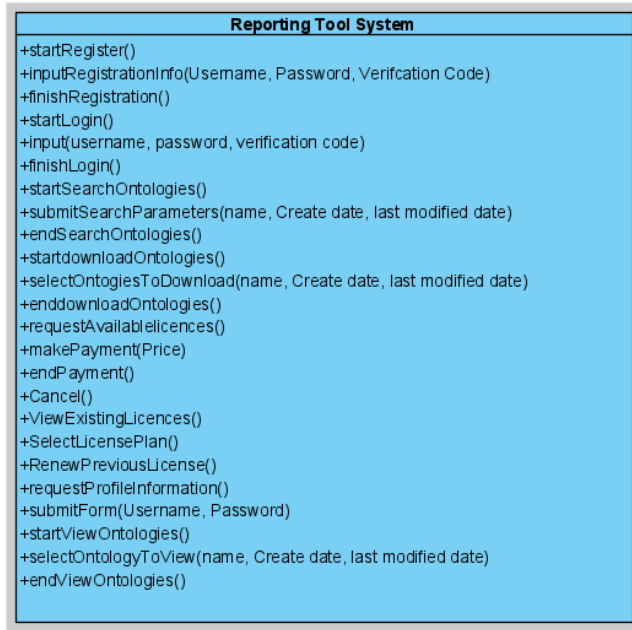
## 6.8 View Ontologies



## 6.9 Download Ontologies



## 7 System Operations



## 8 Operations Contracts

### 8.1 Contract 1: submitSearchParameters

Operation: submitSearchParameters(name, createDate, node, edge, createBy)

Cross References: UC Search Ontologies

Preconditions:

- Ontology instance is created before.
- Ontology instance matches to the parameters.

Postconditions:

- ontology is associated with the previous Ontology.
- ontology.name became name.
- ontology.createDate became createDate.
- ontology.lastModifiedDate became lastModifiedDate.

### 8.2 Contract 2: selectOntologyToView

Operation: selectOntologyToView(name, createDate, node, edge, createBy)

Cross References: UC View Ontologies

Preconditions:

- Ontology instance is created before.

- Ontology instance equal to selected ontology.

Postconditions:

- ontology is associated with the previous Ontology.
- ontology.name became name.
- ontology.createDate became createDate.
- ontology.lastModifiedDate became lastModifiedDate.

### 8.3 Contract 3: selectOntologyToDownload

Operation: selectOntologyToDownload(name, createDate, node, edge, createBy)

Cross References: UC Download Ontologies

Preconditions:

- Ontology instance is created before.
- Ontology instance equal to selected ontology.

Postconditions:

- ontology is associated with the previous Ontology.
- ontology.name became name.
- ontology.createDate became createDate.
- ontology.lastModifiedDate became lastModifiedDate.

### 8.4 Contract 4: makePayment

Operation: makePayment(price)

Cross References: Use case UC4: Make a payment

Preconditions: A license instance is created before.

Postconditions:

- user is associated with license by sale.
- payment.price becomes price(attribution modification)
- payment.purchaseDate becomes the current date.
- licence.expireDate becomes the current date plus the duration of the license.
- payment.transactionStatus becomes 1 that stands for success.
- Licence.flag becomes 1 that stands of active license.

### 8.5 Contract 5: selectLicensePlan

Operation: SelectLicensePlan()

Cross References: UC Choosing a License Plan

Preconditions:

- User instance is created before.

Postconditions:

- A license instance will be created.
- license.duration becomes the selected license duration.
- User is redirected to the payment page

## 8.6 Contract 6: submitForm

Operation: submitForm(Username,Password)

Cross References: UC Manage Profile

Preconditions:

- User instance is created before.

Postconditions:

- User.username becomes username(attribute modification)
- User.password becomes password(attribute modification)

## 8.7 Contract 7: renewPreviousLicences

Operation: renewPreviousLicenses()

Cross References: UC Manage Licenses

Preconditions:

- User license instance has been created for the particular license duration.

Postconditions:

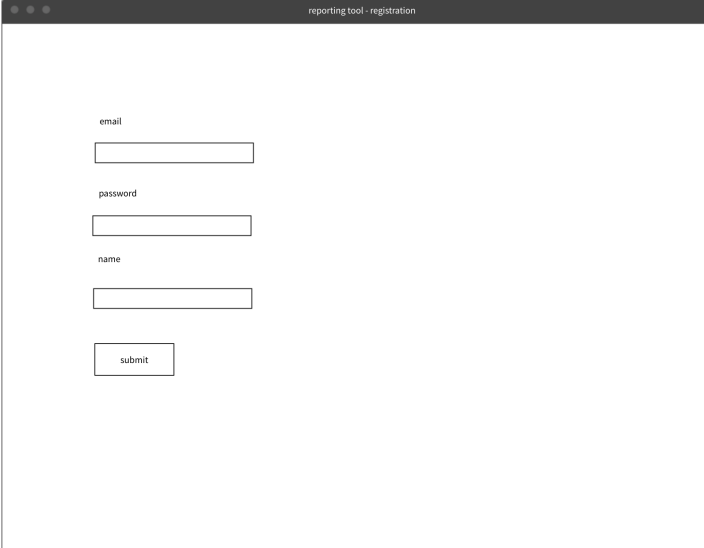
- User was redirected to payment page.
- After the payment operation license.expireDate was updated.

# 9 Wire Frames

## 9.1 Login

The wireframe shows a web browser window with the title 'Reporting Tool - login'. Inside the window, there is a central login form. The form has a title 'login' at the top. Below the title, there are two input fields: one labeled 'email' and another labeled 'password'. Below these fields, there are two buttons: 'signup' on the left and 'login' on the right.

## 9.2 Registration



reporting tool - registration

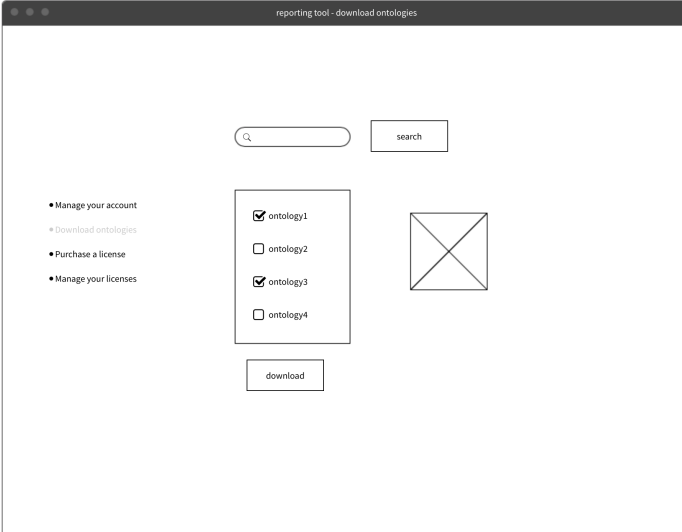
email

password

name

submit

## 9.3 Download Ontologies




reporting tool - download ontologies

Q  search

- Manage your account
- Download ontologies
- Purchase a license
- Manage your licenses

☒ ontology1  
☐ ontology2  
☒ ontology3  
☐ ontology4



download

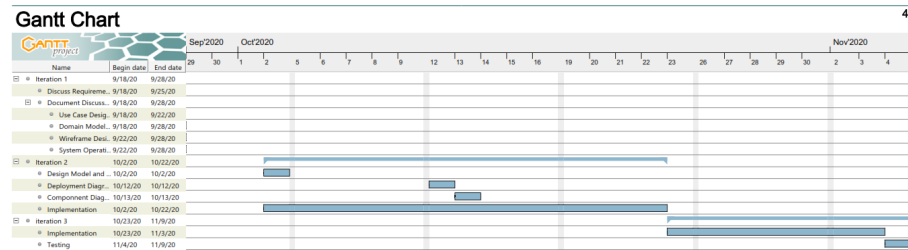


## 10 Tractability Matrix

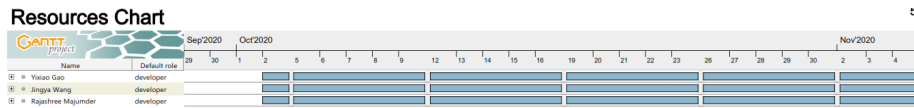
	REQ-1 Customer can register to the system	REQ-2 Customer logs in to the system.	REQ-3 Customer can purchase licenses using paypal.	REQ-4 Customer will get email notification upon license approval	REQ-5 The system display the ontology's.	REQ-6 Customers can manage their own account.	REQ-7 Customer's can manage their license.	REQ-8 Customers can download the ontology's in an excel spreadsheet and content of the GO as PDF or excel sheet.	REQ-9 customer can search ontology.	REQ-10 customer can choose the type of licences
UC-1 Login in		X								
UC-2 Registration	X									
UC-3 purchase a license	X	X	X	X						
UC-4 view ontology	X	X	X		X					
UC-5 search ontology	X	X	X		X					
UC-6 download ontology	X	X	X		X			X		
UC-7 manage profile	X	X				X			X	
UC-8 manage license	X	X	X				X			
UC-9 choosing a license plan	X	X								X

## 11 Gantt Chart and Resources Chart

### 11.1 Gantt Chart



### 11.2 Resources Chart



## 12 Issue Tracking

<https://gitlab.com/sbircrosswalk/reporting-tool/-/issues>