

Ontology-based Expert System for a Generic Drug Production of Pharmaceutical Dosage Forms

Software Design Document

By

Mr. Narongrit Saisuwan 542115017

Mr. Panupak Wichaidit 542115047

Department of Software Engineering
College of Arts, Media and Technology
Chiang Mai University

Project Advisor

.....
Dr. Chartchai Doungsa-ard

Document History

Document Name	Version	Status	Date	Viewable	Reviewer	Responsible
Documents						
OEGP –Software Design Document _v.0.1.docx	0.1 <ul style="list-style-type: none"> • Introduction • Class Diagram • Sequence Diagram • UI Design 	Draft	Jun. 4, 14	NS, PW, CD	NS, PW	NS, PW
OEGP –Software Design Document _v.0.2.docx	0.2 <ul style="list-style-type: none"> • Acronyms • Class Diagram • Method description 	Draft	July 6, 2014	NS, PW, CD	NS, PW	NS, PW
OEGP –Software Design Document _v.1.0.docx	1.0 <ul style="list-style-type: none"> • Update the figure number 	Release	July 6, 2014	NS, PW, CD	NS, PW	NS, PW
OEGP-Software Design Document _v1.1	1.1 <ul style="list-style-type: none"> • Update substance, excipient and formulation class diagram and sequence diagram • Remove substance Properties class diagram and sequence diagram • Add class diagram and sequence at client part • Redesign UI 	Draft	October 16, 2014	NS, PW, CD	NS, PW	NS, PW
OEGP-Software Design Document _v2.0	2.0 <ul style="list-style-type: none"> • Update the class diagram • Update method description • Update figure numbering 	Release	October 20, 2014	NS, PW, CD	NS, PW	NS, PW

Contents

Document History	ii
Chapter 1 Introduction	1
1.1. Identification	1
1.2. Acronyms	2
Chapter 2 Use Case and class diagram relationship	3
Chapter 3 Class Diagram	4
3.1. Sub-Feature 5: Manage the drug substance	4
3.1.1. CD-CI- 01: Substance Class Diagram (Client Side)	4
3.1.2. CD-SV-01: Substance Class Diagram (Server Side)	11
3.2. Sub-Feature 6: Manage the drug Excipient	17
3.2.1. CD-CI- 02: Excipient Class Diagram (Client Side)	18
3.2.2. CD-SV-02: Excipient Class Diagram (Server Side)	24
3.3. Sub-Feature 7: Manage the drug formulation	28
3.3.1. CD-CI-03: Formulation Class diagram (Client Side)	28
3.3.2. CD-SV-03: Formulation Class Diagram (Server Side)	33
3.4. Sub-Feature 2: Calculate the drug reformulation by using the inference engine	37
3.4.1. CD-CI- 04: Reformulation Class Diagram (Client Side)	37
3.4.2. CD-SV-04: Reformulation Class Diagram (Server Side)	43
3.5. Sub-Feature 3: View the drug reformulation history	49
3.5.1. CD-CI- 05: Reformulation History Class Diagram (Client Side)	50
3.5.2. CD-SV-05: Reformulation History Class Diagram (Server Side)	54
Chapter 4 Sequence Diagram	59
4.1. Sub-Feature 6: Manage the drug substance	60
4.1.1. URS-09: The user adds a new substance to the system.	60
4.1.2. URS-10: The user updates an existing substance in the system.	62
4.1.3. URS-11: The user deletes an existing substance from the system.	64
4.1.4. URS-12: The user views the substance in the system.	66
4.2. Sub-Feature 7: Manage the drug excipient	68
4.2.1. URS-13: The user adds a new excipient to the system.	68
4.2.2. URS-14: The user updates an existing excipient in the system.	70
4.2.3. URS-15: The user deletes an existing excipient from the system.	72
4.2.4. URS-16: The user views the excipient in the system.	74
4.3. Sub-Feature 8: Manage the drug formulation	76
4.3.1. URS-17: The user adds a new drug's formulation to the system....	76

4.3.2.	URS-18: The user updates an existing drug's formulation in the system.....	78
4.3.3.	URS-19: The user deletes an existing drug's formulation from the system.	80
4.3.4.	URS-20: The user views the drug's formulation in the system.....	82
4.4.	Sub-Feature 2: Calculate the drug reformulation by using the inference engine.	84
4.4.1.	URS-06: The user calculates a drug reformulation by using an inference engine.....	84
4.5.	Sub-Feature 3: View the drug reformulation history.....	86
4.5.1.	URS-07: The user views their drug reformulation history.	86
Chapter 5	User Interface.....	88
5.1.	Sub-Feature 6: Manage the drug substance.....	89
5.1.1.	URS-09: The user adds a new substance into the system.	89
5.1.2.	URS-10: The user updates an existing substance into the system.	91
5.1.3.	URS-11: The user deletes an existing substance from the system.	93
5.1.4.	URS-12: The user views the substance in the system.....	93
5.2.	Sub-Feature 7: Manage the drug excipient	94
5.2.1.	URS-13: The user adds a new excipient to the system.....	94
5.2.2.	URS-14: The user updates an existing excipient in the system.	96
5.2.3.	URS-15: The user deletes an existing excipient from the system.....	98
5.2.4.	URS-16: The user views the excipient in the system	98
5.3.	Sub-Feature 8: Manage the drug formulation.....	99
5.3.1.	URS-17: The user adds a new formulation to the system.	99
5.3.2.	URS-18: The user updates a formulation in the system.	101
5.3.3.	URS-19: The user deletes an existing formulation from the system.	103
5.3.4.	URS-20: The user views the formulation in the system.....	103
5.4.	Sub-Feature 2: Calculate the drug reformulation by using the inference engine.	104
5.4.1.	URS 06: The user calculates a drug reformulation by using an inference engine.	104
5.5.	Sub-Feature 3: View the drug reformulation history.....	107
5.5.1.	URS-07: The user views their drug reformulation history.	107

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	4 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

Chapter 1 | Introduction

1.1. Identification

Software Design Document (SDD) is translated the requirement into the detail design. The SDD also explains the system architecture in detail. The purposes of the description are making a same understanding about the system. The software design consists of the list of Sub-Feature in a software requirement specification that is illustrated in Table 1.

No	Sub-Feature	URS No.	URS Name	Actor
2	Calculate the drug reformulation by using the inference engine.	URS-06	The user calculates a drug reformulation by using an inference engine.	General Pharmacists, Expert Pharmacists, Administrator
3	View the drug reformulation history	URS-07	The user views their drug reformulation history.	General Pharmacists, Expert Pharmacists, Administrator
5	Manage the drug substance	URS-09	The user adds a new substance into the system.	Expert Pharmacists, Administrator
		URS-10	The user updates an existing substance into the system.	
		URS-11	The user deletes an existing substance from the system.	
		URS-12	The user views the substance in the system.	
6	Manage the drug excipient	URS-13	The user adds a new excipient to the system.	Expert Pharmacists, Administrator
		URS-14	The user updates an existing drug excipient in the system.	
		URS-15	The user delete an existing drug excipient in the system.	
		URS-16	The user views all the drug excipient in the system.	
7	Manage the drug formulation	URS-17	The user adds a new drug formulation case into the system.	Expert Pharmacists, Administrator
		URS-18	The user updates an existing drug formulation case in the system.	
		URS-19	The user deletes an existing drug formulation case in the system.	
		URS-20	The user views all of the formulation in the system.	

Table 1: The list of Sub-Feature in software requirement specification

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	1 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

1.2. Acronyms

OEGP	Ontology-based Expert System for a Generic Drug Production of Pharmaceutical Dosage Forms
PDPO	Pharmaceutical Dosage Form Production Ontology
URS	User Requirement Specification
SRS	System Requirement Specification
CD	Class diagram
CD-CI	Class diagram For Client Side
CD-SV	Class diagram For Server Side.
SQD	Sequence Diagram
SQD-CI	Sequence Diagram For Client Side
SQD-SV	Sequence Diagram For Server Side
UI	User interface
SDD	Software Design Document

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	2 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

Chapter 2 | Use Case and class diagram relationship

The Sub-Feature in the table 1 is related with the class diagram that show on the list below this passage.

Feature 5: PDPO System

Sub-Feature 6: Manage the drug substance

- Client Side
 - CD-CI-01 : Substance Class Diagram (Client Side)
- Server Side
 - CD-SV-01 : Substance Class Diagram (Server Side)

Sub-Feature 7: Manage the drug excipient

- Client Side
 - CD-CI-02 : Excipient Class Diagram (Client Side)
- Server Side
 - CD-SV-02 : Excipient Class Diagram (Server Side)

Sub-Feature 8: Manage the drug formulation

- Client Side
 - CD-CI-03 : Formulation Class Diagram (Client Side)
- Server Side
 - CD-SV-03 : Formulation Class Diagram (Server Side)

Feature 2: Rule Base System and Feature 4: Case Base Reasoning System

Sub-Feature 2: Calculate the drug reformulation by using the inference engine.

- Client Side
 - CD-CI-04 : Reformulation Class Diagram (Client Side)
- Server Side
 - CD-SV-04 : Reformulation Class Diagram (Server Side)

Sub-Feature 3: View the drug reformulation history.

- Client Side
 - CD-CI-05 : Reformulation History Class Diagram (Client Side)
- Server Side
 - CD-SV-05 : Reformulation History Class Diagram (Server Side)

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	3 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

Chapter 3 | Class Diagram

3.1. Sub-Feature 5: Manage the drug substance

The Sub-Feature 5 is the substance management. The user can add, update, delete and view the substance. The substance is a part of excipient and formulation.

3.1.1. CD-CI- 01: Substance Class Diagram (Client Side)

3.1.1.1. Class diagram

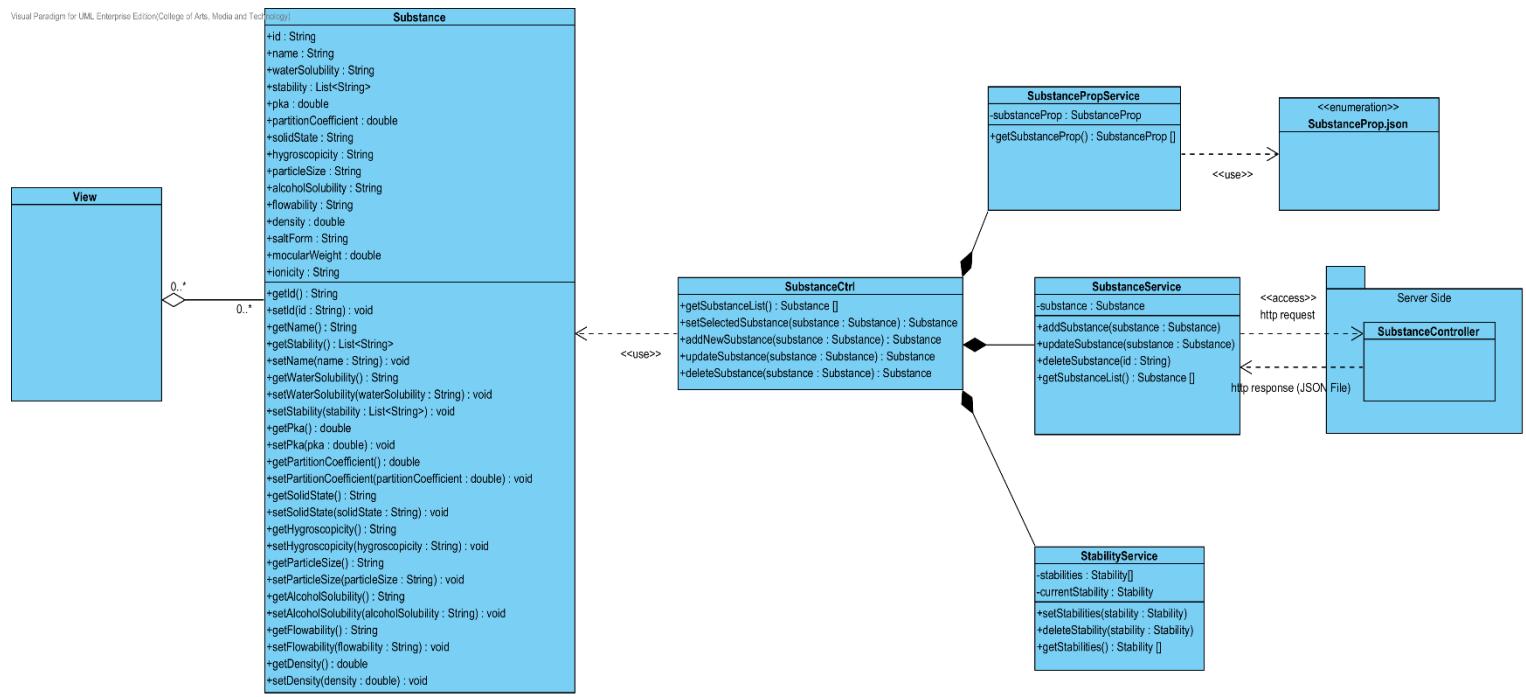


Figure 1: CD-CI-01 – Substance Class Diagram (Client Side)

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	4 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

3.1.1.2. Class description

From the Figure 1. It can divide into 5 Classes .The detail of each class is described on the next paragraph.

3.1.1.2.1. Substance Class

Visual Paradigm for UML Enterprise Edition(College of Arts, Media and Technology)

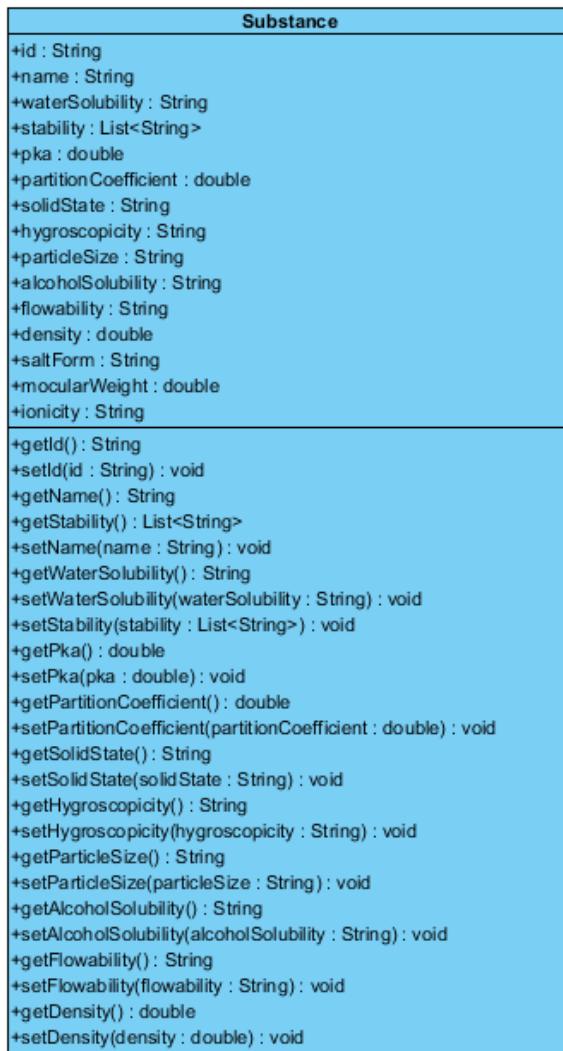


Figure 2: Substance Class (Client Side)

The Substance Class in Client Side is the model class that used for receiving substance data from the user and showing a substance data to the user. This model is controlled by SubstanceCtrl. The substance model consists of 15 attribute.

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	5 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

3.1.1.2.1.1. Attribute Description

- **Id** – the identity of the substance class. Id attribute is a String.
- **Name** – the name of the substance class. The substance name is String.
- **waterSolubility** – the soluble of water in each substance. The water solubility is String.
- **Stability** – the stability of substance. The stability attribute is the list of String.
- **pKa** – the dissociation constant of the drug. It used for estimating the result of drug. The pKa is a double number.
- **PartitionCoefficient** – the partition coefficient is the distribution between substances. The partition coefficient is double number.
- **Solid State** – the statement that change to solid. The solid state is String.
- **Hygroscopicity** – the ability to absorb the moisture from the air and water. The hygroscopicity is the String
- **ParticleSize** – the size of particle in each substance. The particle size is String.
- **AlcoholSolubility** – the soluble of alcohol in each substance. The alcohol solubility is String.
- **FlowAbility** – the format of flowing that estimate from the medium. The flow ability is the String.
- **Density** – the ration between mass and volume. The density is the double number.
- **SaltForm** – the form of salt that used in each substance. The salt form is String.
- **MolecularWeight** – the weight of each molecular in each substance. The molecular weight is double number.
- **Ionicity** – the ionicity of substance. The ionic is String.

3.1.1.2.1.2. Method Description

- **Getter and Setter method** – It used when the system set value and get value.

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	6 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

3.1.1.2.2. SubstanceCtrl

Visual Paradigm for UML Enterprise Edition(College of Arts, Media and Technology)

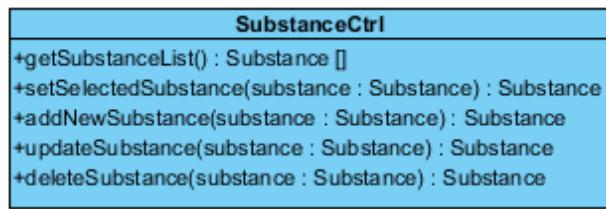


Figure 3 : SubstanceCtrl (Client Side)

The SubstanceCtrl is the substance controller that used for controlling the substance data that receive from the user or server. The SubstanceCtrl consists of 5 method. The detail of each method are shown on next paragraph.

3.1.1.2.2.1. Attribute Description

N/A

3.1.1.2.2.2. Method Description

- **getSubstanceList() : Substance[]** - This method is used for getting the list of substance from the service. The method will return the list of substance object. On the other hand, if there are not any substance object in the database. The SubstanceCtrl will return null.
- **setSelectedSubstance(substance : Substance) : Substance** – This method is used for setting the substance data that selected by the user. The selected substance can use for updating or deleting substance object. The input variable is the substance object that user selected.
- **addNewSubstance (substance : Substance) : Substance** – This method is used for adding a new substance data to the database. The method will receive data from the user. The input variable is the new substance that create by user.
- **UpdateSubstance(substance : Substance) : Substance** – This method is used for updating the existing substance that user selected. The input variable is the substance that user selected.
- **DeleteSubstance(substance : Substance) : Substance** – This method is used for deleting the existing substance from the database. The input variable is the substance that user selected.

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	7 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

3.1.1.2.3. SubstancePropService

Visual Paradigm for UML Enterprise Edition|College of Arts, Media :

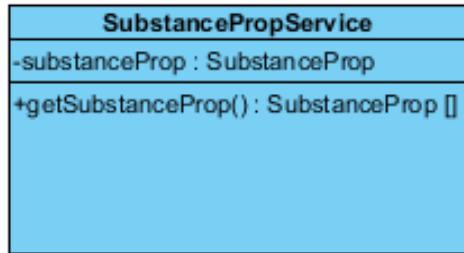


Figure 4 : SubstancePropService (Client Side)

The SubstancePropService used for getting the substance properties data from the JSON file. The SubstanceCtrl use this class for setting the properties of substance object.

3.1.1.2.3.1. Attribute Description

N/A

3.1.1.2.3.2. Method Description

- **getSubstanceProp() : SubstanceProp[]** : The getSubstanceProp method is used for getting substance data from the JSON file.

3.1.1.2.4. SubstanceService

Visual Paradigm for UML Enterprise Edition|College of Arts, Media and

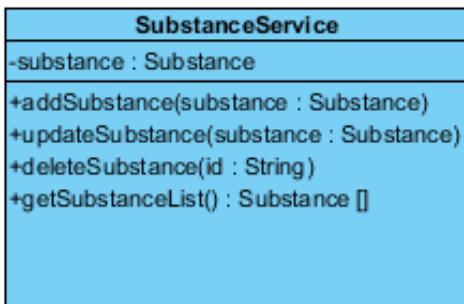


Figure 5 : Substance Service (Client Side)

The Substance Service is the service that use for adding, updating, deleting and getting data from the server. The SubstanceService can do its business by using http request such as POST, PUT, GET and DELETE.

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	8 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

3.1.1.2.4.1. Attribute Description

- **Substance** – The Substance object that receive from SubstanceCtrl and get from the server.

3.1.1.2.4.2. Method Description

- **addSubstance (substance: Substance)** – This method is used for adding a new substance to database. The addSubstance method send the data that user input to the server by POST request. The input variable of this method is substance object that receive from the SubstanceCtrl.
- **updateSubstance (substance : Substance)** – This method is used for updating the existing substance object. The updateSubstance method send the edited data that user input to the server by PUT request. The input variable of this method is substance object that receive from the Substance Ctrl.
- **deleteSubstance (id: String)** – This method is used for deleting the existing substance object. The deleteSubstance method send the id of the substance object to the server by DELETE request. The input variable of this method is id of substance that receive from SubstanceCtrl.
- **getSubstanceList() : Substance []** – This method is used for getting the substance object from the database. This method send the GET request to server for getting the data.

3.1.1.2.5. StabilityService

Visual Paradigm for UML Enterprise Edition(College of Arts,

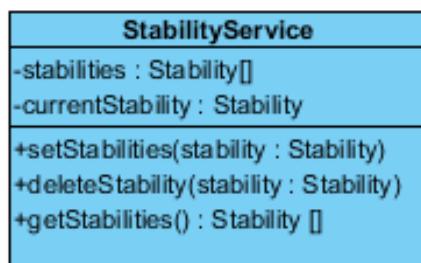


Figure 6 : StabilityService (Client Side)

The StabilityService is the service that use for setting, deleting and getting the stability object to the list. The stability object is one of substance properties.

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	9 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

3.1.1.2.5.1. Attribute Description

- **Stabilities** – The list of stability object that receive from the server.
- **currentStability** – The stability object that user want to add to the substance object.

3.1.1.2.5.2. Method Description

- **setStabilities(stability: Stability)** - This method use for setting the stability that user input. The input variable is the stability object that create by the user.
- **deleteStability(stability : Stability)** - This method user for deleting the stability that user input. The input variable is the stability object that user selected for deleting.
- **getStabilities () : Stabilities []** – This method is used for getting the stability object list for adding to the substance.

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	10 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

3.1.2. CD-SV-01: Substance Class Diagram (Server Side)

3.1.2.1. Class diagram

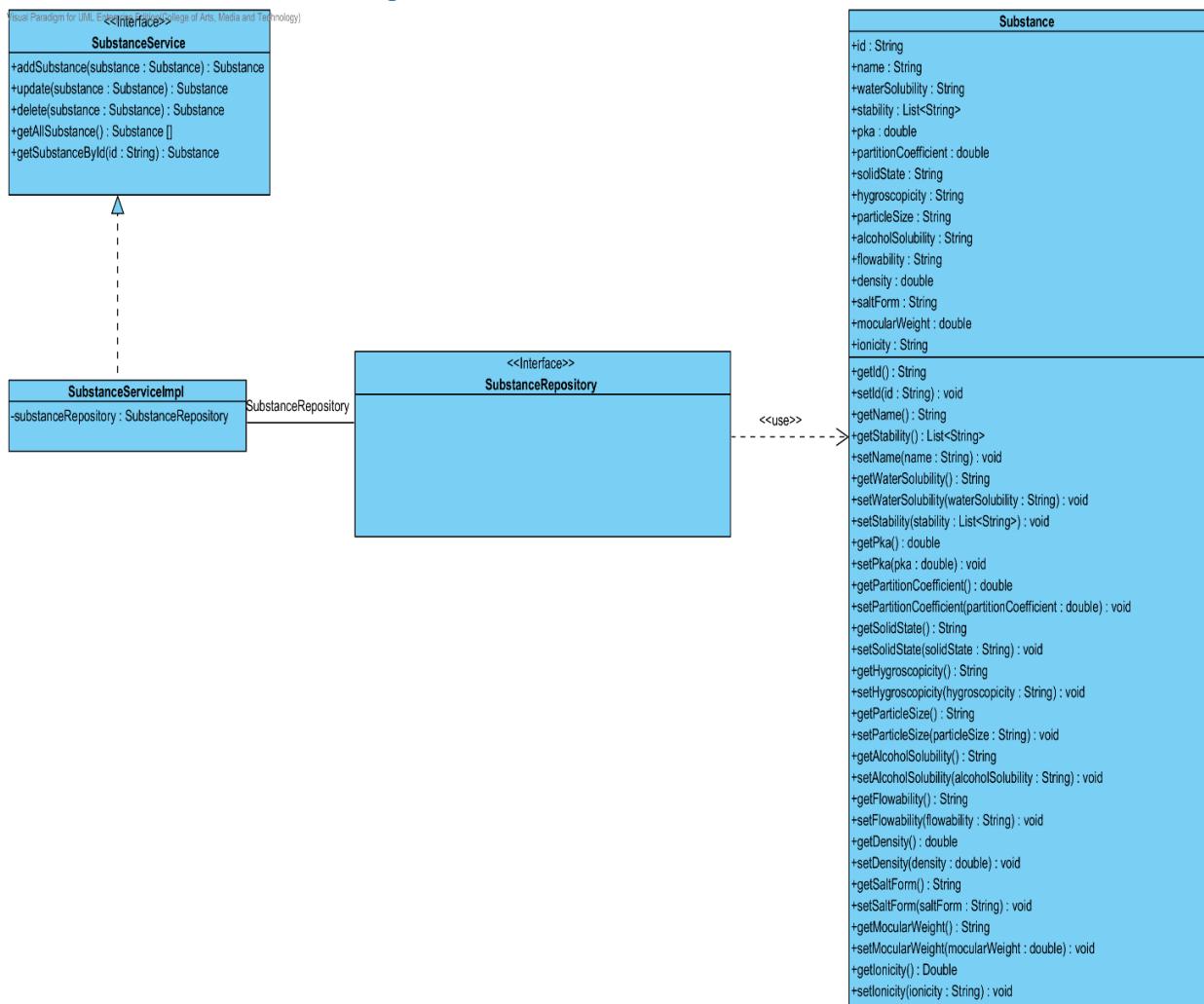


Figure 7 : CD-SV-01 – Substance Class Diagram (Server Side).

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	11 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

3.1.2.2. Class description

From the figure 7, it can divide into 4 important classes. The detail of each class is described on the next paragraph.

3.1.2.2.1. Substance class

Visual Paradigm for UML Enterprise Edition(College of Arts, Media and Technology)

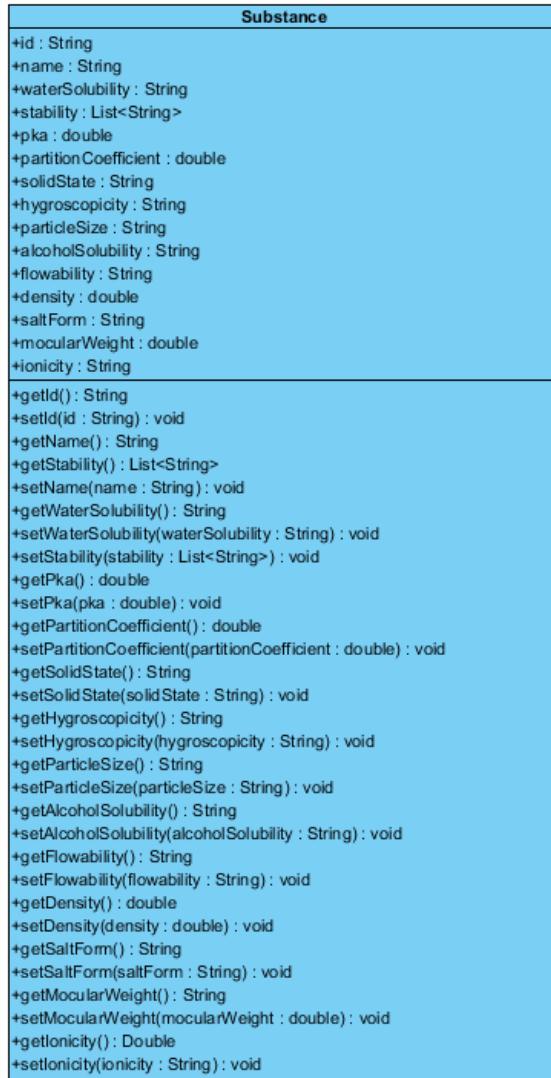


Figure 8 : Substance Class (Server Side)

Substance is a part of drug's excipient. Substance class is an entity class that will be saved to the system. The substance class consists of 15 attributes follow the list below this passage.

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	12 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

3.1.2.2.1.1. Attribute description

- **Id** – the identity of the substance class. Id attribute is a String.
- **Name** – the name of the substance class. The substance name is String.
- **waterSolubility** – the soluble of water in each substance. The water solubility is String.
- **Stability** – the stability of substance. The stability attribute is the list of String.
- **pKa** – the dissociation constant of the drug. It used for estimating the result of drug. The pKa is a double number.
- **PartitionCoefficient** – the partition coefficient is the distribution between substances. The partition coefficient is double number.
- **Solid State**- the statement that change to solid. The solid state is String.
- **Hygroscopicity** – the ability to absorb the moisture from the air and water. The hygroscopicity is the String
- **ParticleSize** – the size of particle in each substance. The particle size is String.
- **AlcoholSolubility**- the soluble of alcohol in each substance. The alcohol solubility is String.
- **FlowAbility** – the format of flowing that estimate from the medium. The flow ability is the String.
- **Density** – the ration between mass and volume. The density is the double number.
- **SaltForm** – the form of salt that used in each substance. The salt form is String.
- **MolecularWeight** – the weight of each molecular in each substance. The molecular weight is double number.
- **Ionicity** – the ionicity of substance. The ionic is String.

3.1.2.2.1.2. Method description

- **Getter and Setter method** – It used when the system set value and get value.

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	13 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

3.1.2.2.2. SubstanceRepository Interface

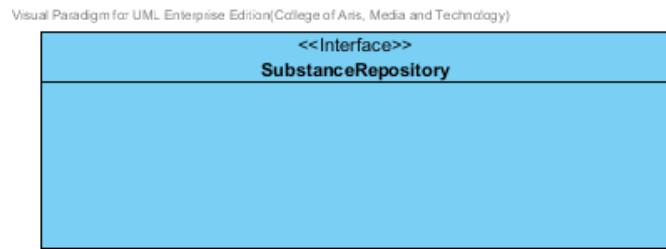


Figure 9 : SubstanceRepository (Server Side)

SubstanceRepository Interface is an interface that use for CRUD with entity classes in the system. All of SubstanceRepository interface's method is generated from Spring Data MongoDB framework. SubstanceRepository interface consists of 4 methods is shown below this passage.

3.1.2.2.1. Attribute description

N/A

3.1.2.2.2. Method description

- **Save (substance: Substance)** - The save method is generated from Spring Data MongoDB framework. This method is used when the user wants to add a new substance or update existing substance to the system. The input variable is a substance object.
- **Delete (substance: Substance)** – The delete method is generated from Spring Data MongoDB framework. This method is used when the user wants to delete the substance from the system. The input variable is substance object.
- **findAll (): Substance []** – The findAll method is generated from Spring Data MongoDB framework. This method is used when the user wants to retrieve all of substance data from the system. The result of this method is a list of substance object.
- **findOne (id: String): Substance** – The findOne method is used when the user wants to retrieve the substance data from the system. The system gets a substance object by the id of substance.

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	14 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

3.1.2.2.3. SubstanceService

Visual Paradigm for UML Enterprise Edition(College of Arts, Media and Technology)

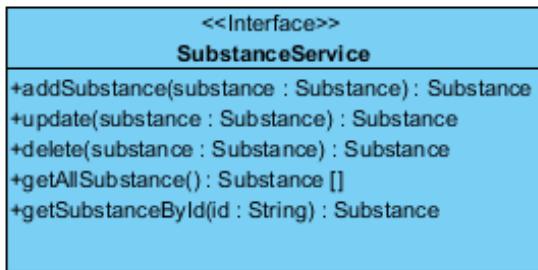


Figure 10 : SubstanceService (Server Side)

SubstanceService is business processing logic for substance entity. SubstanceService manages the substance data through the SubstanceRepository interface. SubstanceService consists of 6 methods follow the list below this passage.

3.1.2.2.3.1. Attribute description

N/A

3.1.2.2.3.2. Method description

- **addSubstance (substance: Substance) : Substance** – The adding substance method is used, when the user wants to add a new substance to the database. This method adds a new substance by input variable of substance object. If the substance object that input by the user is not contained in the database, this method will add a new substance to the database and return the substance object from the database to the user after the adding substance is successful. On the other hand, when the substance object that input by the user is contained in the database. This method will return a null value to the user.
- **updateSubstance (substance: Substance): Substance** - The updating substance method is used, when the user wants to update an existing substance on in the database. This method update the existing substance by input variable of substance object. If the substance object that input by the user is contained in the database, this method will update an existing substance in the database and return the substance object from the database to the user after the updating substance is successful. On the other hand, when the substance object that input by the user is not contained in the database. This method will return a null value to the user.
- **deleteSubstance (substance: Substance) : Substance** – The deleting substance method is used when the user wants to deletes the existing substance from the database. This method delete the substance by input variable of substance object. If the substance object that input by the user is contained in the database, this method will delete an existing substance from the database and return the substance object to the user after the deleting substance is successful. On the other hand, when the substance

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	15 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

object that input by the user is not contained in the database. This method will return a null value to the user.

- **getAllSubstance() : Substance []** – The getAllSubstance method is used, when the user wants to get all substance data in the database. This method is return as a list of substance object database.
- **getSubstanceById(id String) : Substance** – The getSubstanceById method is used, when the user wants to get the substance data in the system. This method gets substance object from the database by id that input by the user. On the other hand, if the id that input by user is not contained in the database. This method will return null value to the user.

3.1.2.2.4. SubstanceServiceImpl

Visual Paradigm for UML Enterprise Edition|College of Arts, Media and Techn

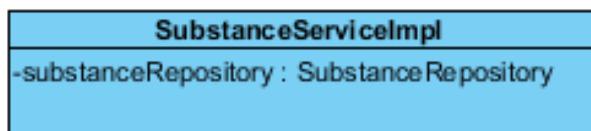


Figure 11 : SubstanceServiceImpl (Server Side)

SubstanceServiceImpl is the substance service class that implements the method from SubstanceService. So, the method of SubstanceServiceImpl is same as SubstanceService.

3.1.2.2.4.1. Attribute description

- **SubstanceRepository** – the repository of substance. This attribute is used for substance data management.

3.1.2.2.4.2. Method description

Same as SubstanceService

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	16 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

3.2. Sub-Feature 6: Manage the drug Excipient

The Sub-Feature 6 is the excipient management. The user can add, update, delete and view the excipient. Each excipient is created from the substance and substanceFunction. The relationship between excipient entity, substance entity, and substanceFunction is illustrated in the Figure 12 below on this passage.

Visual Paradigm for UML Enterprise Edition(College of Arts, Media and Technology)

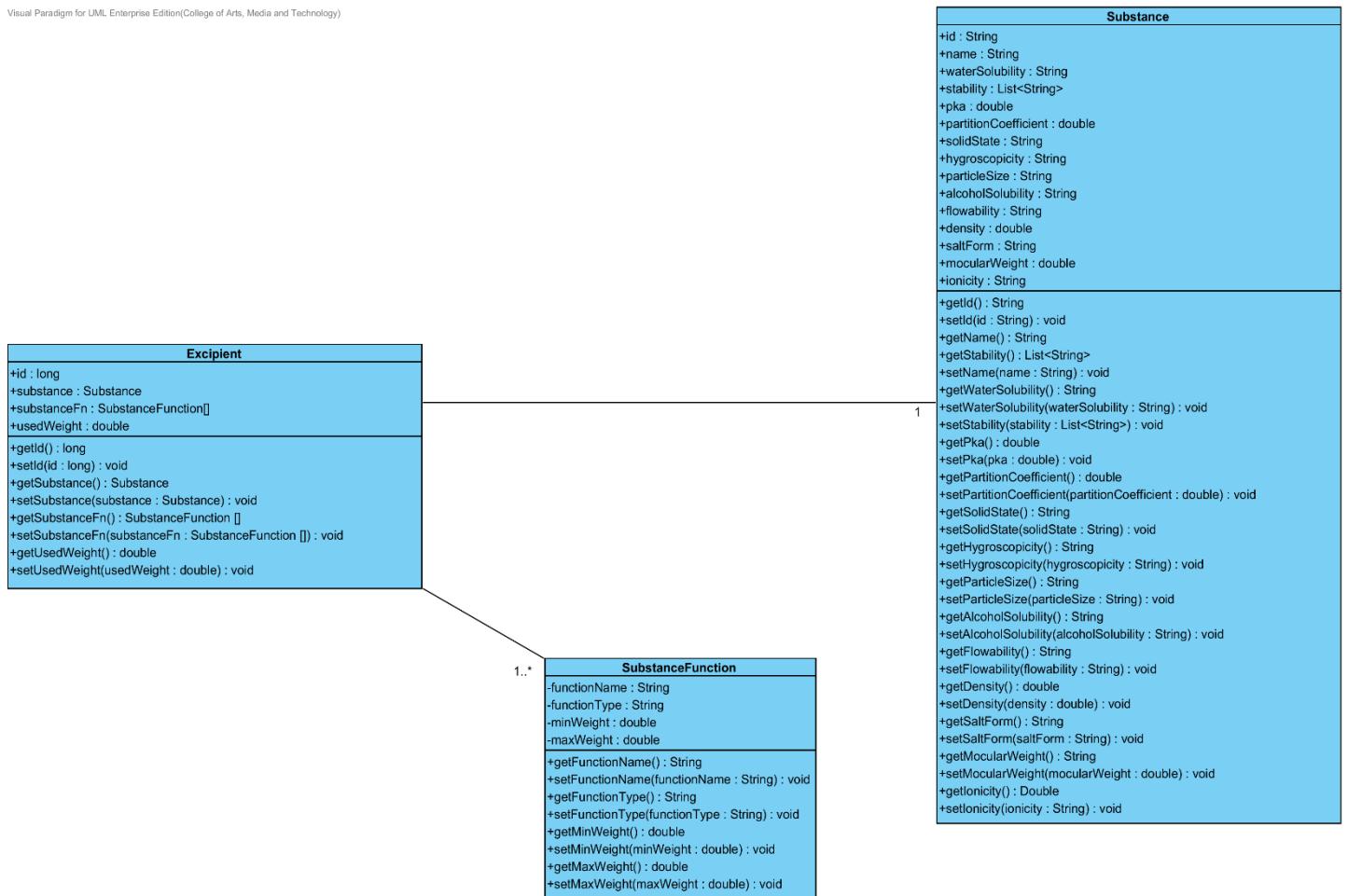


Figure 12: The Entity Relationship between Excipient, Substance Function and Substance

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	17 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

3.2.1. CD-Cl- 02: Excipient Class Diagram (Client Side)

3.2.1.1. Class diagram

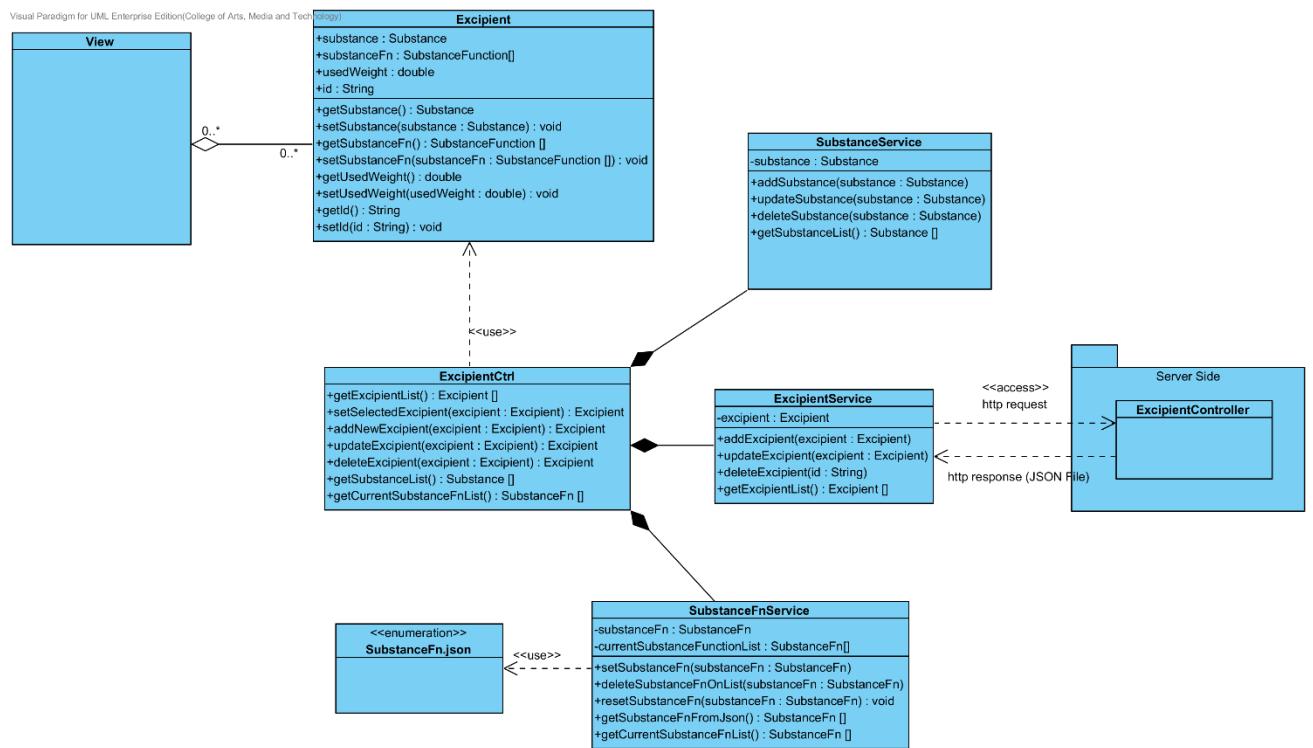


Figure 13: CD-CI-02 – Excipient Class Diagram (Client Side).

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	18 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

3.2.1.2. Class description

From the Figure number 13 .It can divide into 5 Classes .The detail of each class is described on the next paragraph.

3.2.1.2.1. Excipient Class

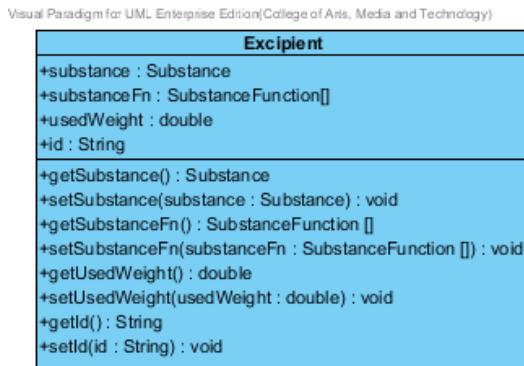


Figure 14: Excipient Class (Client Side)

The Excipient Class in Client Side is the model class that used for receiving excipient data from the user and showing an excipient data to the user. This model is controlled by ExcipientCtrl.

3.2.1.2.1.1. Attribute Description

- **Id** – the identity of the excipient class. Id attribute is a String number.
- **Substance** – the substance of the excipient class. The substance attribute is a substance object.
- **SubstanceFn**–the substance Function of the excipient class .The user can set the substance Function in each excipient .The substance Function is a substance function object.
- **usedWeight** – the weight of substance function that will be use in each excipient.

3.2.1.2.1.2. Method Description

- **Getter and Setter method** – It used when the system set value and get value.

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	19 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

3.2.1.2.2. ExcipientCtrl

Visual Paradigm for UML Enterprise Edition(College of Arts, Media and Technology)

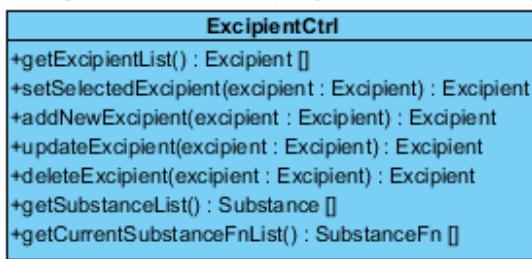


Figure 15 : ExcipientCtrl (Client Side)

The ExcipientCtrl is the excipient controller that used for controlling the excipient data that receive from the user or server. The ExcipientCtrl consists of 5 method. The detail of each method are shown on next paragraph.

3.2.1.2.2.1. Attribute Description

N/A

3.2.1.2.2.2. Method Description

- **getExcipientList() : Excipient[]** - This method is used for getting the list of excipient from the server. The method will return the list of excipient object. On the other hand, if there are not any excipient object in the database. The ExcipientCtrl will return null.
- **setSelectedExcipient(excipient : Excipient) : Excipient** - This method is used for setting the excipient data that selected by the user. The selected excipient can use for updating or deleting excipient object. The input variable of this method is the excipient object that selected by the user.
- **addNewExcipient (excipient : Excipient) : Excipient** – This method is used for adding a new excipient data to the database. The input variable of this method is the excipient object that create by the user.
- **UpdateExcipient(excipient : Excipient) : Excipient** – This method is used for updating the existing excipient in the database that user selected. The input variable of this method is the excipient object that selected by the user.
- **DeleteExcipient(excipient : Excipient) : Excipient** – This method is used for deleting the existing excipient from the database. The input variable of this method is the excipient object that selected by the user.
- **getExcipientList() : Excipient[]** - This method is used for getting the list of excipient from the service. The method will return the list of excipient object. On the other hand, if there are not any excipient object in the database. The getExcipientList will return null.
- **getCurrentSubstanceFnList() : SubstanceFn []** - This method is used for getting substance function object from the substanceFnService. This method will return the substance Function list for setting a new excipient.

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	20 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

3.2.1.2.3. SubstanceService

Visual Paradigm for UML Enterprise Edition|College of Arts, Media and

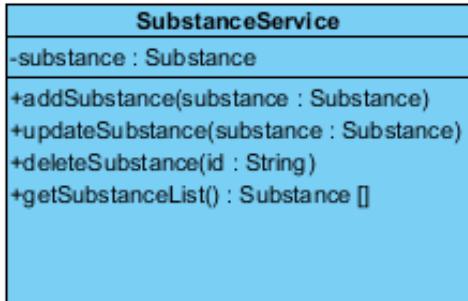


Figure 16 : Substance Service (Client Side)

The Substance Service is used for adding, updating, deleting and getting data from the server. The ExcipientCtrl will use this service for creating excipient object.

3.2.1.2.3.1. Attribute Description

- **Substance** – The Substance object that receive from SubstanceCtrl and get from the server.

3.2.1.2.3.2. Method Description

- **addSubstance (substance: Substance)** – This method is used for adding a new substance to database. The addSubstance method send the data that user input to the server by POST request. The input variable is the substance object that created by the user.
- **updateSubstance (substance : Substance)** – This method is used for updating the existing substance object. The updateSubstance method send the edited data that user input to the server by PUT request. The input variable is the substance object that selected by the user.
- **deleteSubstance (id: String)** – This method is used for deleting the existing substance object. The deleteSubstance method send the id of the substance object to the server by DELETE request. The input variable is the id of the substance that user wants to delete.
- **getSubstanceList() : Substance []** – This method is used for getting the substance object from the database. This method send the GET request to server for getting the data.

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	21 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

3.2.1.2.4. ExcipientService

Visual Paradigm for UML Enterprise Edition(College of Arts, Medi

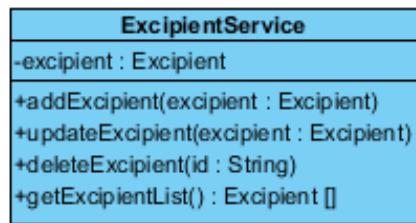


Figure 17 : Excipient Service (Client Side)

The Excipient Service is the service that use for adding, updating, deleting and getting data from the server. The ExcipientService can do its business by using http request such as POST, PUT, GET and DELETE.

3.2.1.2.4.1. Attribute Description

- **Excipient** – The Excipient object that receive from ExcipientCtrl and get from the user.

3.2.1.2.4.2. Method Description

- **addExcipient (excipient: Excipient)** – This method is used for adding a new excipient to database. The addExcipient method send the data that user input to the server by POST request. The input variable of this method is the excipient that created by the user.
- **updateExcipient (excipient : Excipient)** – This method is used for updating the existing excipient object. The updateExcipient method send the edited data that user input to the server by PUT request. The input variable of this method is the excipient that selected by the user.
- **deleteExcipient (id: String)** – This method is used for deleting the existing excipient object. The deleteExcipient method send the id of the excipient object to the server by DELETE request. The input variable of this method is the id of excipient that user wants to delete.
- **getExcipientList() : Excipient []** – This method is used for getting the excipient object from the database. This method send the GET request to server for getting the data.

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	22 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

3.2.1.2.5. SubstanceFnService

Visual Paradigm for UML Enterprise Edition(College of Arts, Media and Technology)

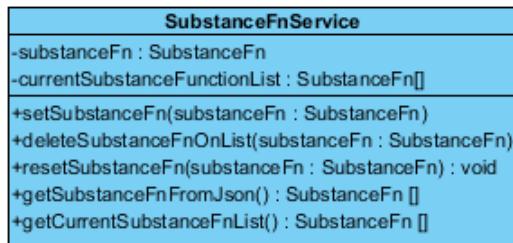


Figure 18 : SubstanceFnService (Client Side)

The SubstanceFnService used for getting the substance function data from the JSON file. The ExcipientCtrl use this class for setting the substance function of excipient object.

3.2.1.2.5.1. Attribute Description

- **SubstanceFn** – Substance Function data that get from the JSON file.
- **CurrentSubstanceFunctionList** – Substance Function that user selected for using in excipient object.

3.2.1.2.5.2. Method Description

- **setSubstanceFn (substanceFn : SubstanceFn)** – This method is used for setting the substance function object that user selected. The input variable of this method is the substance function that selected by the user.
- **deleteSubstanceFnOnList (substanceFn : SubstanceFn)** – This method is used for deleting the substance function object from the current substance function list. The input variable of this method is the substance function that selected by the user.
- **resetSubstanceFn(substanceFn : SubstanceFn)** – This method is used for reset the substance function in the list. Which mean the substance function list will be a null value. The input variable of this method is the substance function that selected by the user.
- **getSubstanceFnFromJson() : SubstanceFn[]** – This method is used for getting the substance function from the JSON file.

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	23 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

3.2.2. CD-SV-02: Excipient Class Diagram (Server Side)

3.2.2.1. Class diagram

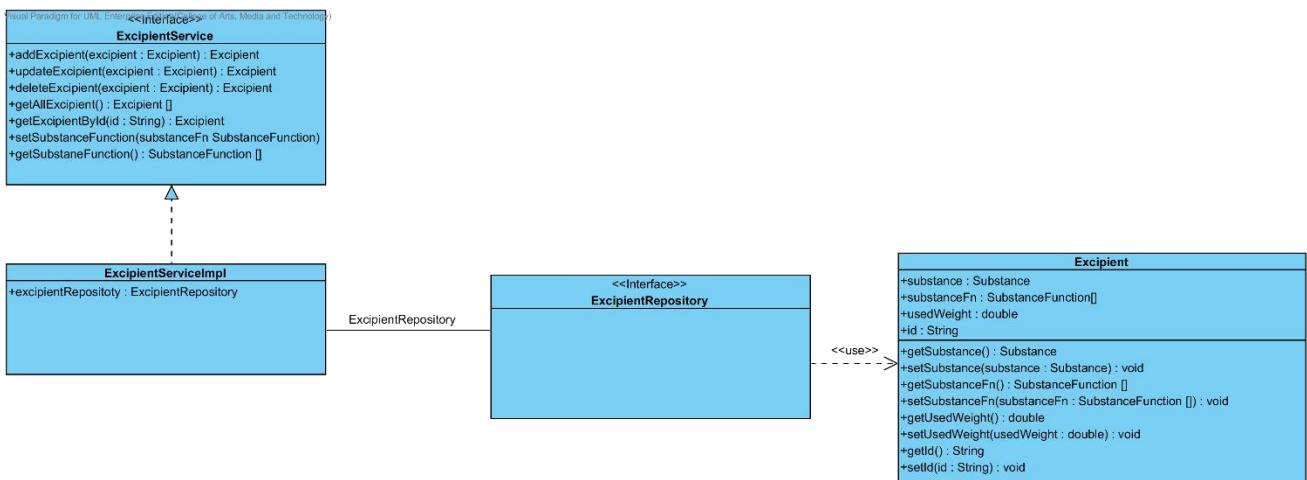


Figure 19 : CD-SV-02 – Excipient Class Diagram (Server Side).

3.2.2.2. Class description

From the figure 19, it can divide into 4 important classes. The detail of each class is described on the next paragraph.

3.2.2.2.1. Excipient class

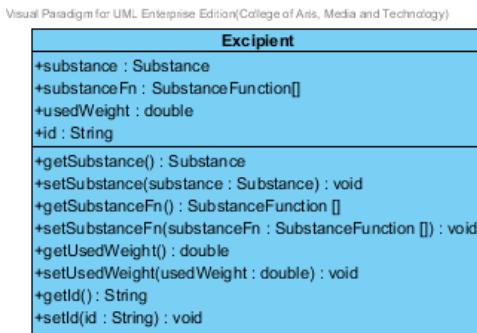


Figure 20 : Excipient Class (Server Side)

Excipient is a part of drug's formulation. Excipient class is an entity class that will be saved to the system. The excipient class consists of 5 attributes follow the list below this passage.

3.2.2.2.1.1. Attribute description

- **Id** – the identity of the excipient class. Id attribute is a String number.
- **Substance** – the substance of the excipient class. The substance attribute is a substance object.

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	24 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

- **SubstanceFn**—the substance Function of the excipient class .The user can set the substance Function in each excipient .The substance Function is a substance function object.
- **usedWeight** – the weight of substance function that will be used in each excipient.

3.2.2.2.1.2. Method description

- **Getter and Setter method** – It is used when the system sets value and gets value.

3.2.2.2.2. ExcipientRepository Interface

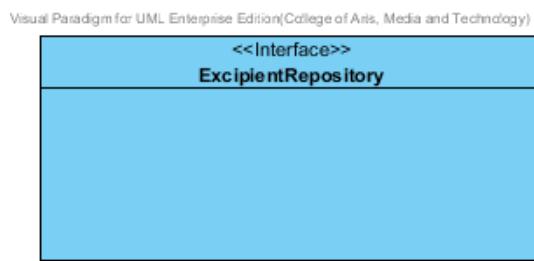


Figure 21: Excipient Repository (Server Side)

ExcipientRepository Interface is an interface that is used for CRUD with entity classes in the system. The most of ExcipientRepository interface's methods are generated from Spring data MongoDB framework. ExcipientRepository interface consists of 4 methods as shown below this passage.

3.2.2.2.2.1. Attribute description

N/A

3.2.2.2.2.2. Method description

- **Save (excipient: Excipient)** - The save method is generated from Spring data MongoDB framework. This method is used when the user wants to add a new excipient or update existing excipient to the system. The input variable is an excipient object.
- **Delete (excipient: Excipient)** – The delete method is generated from Spring data MongoDB framework. This method is used when the user wants to delete the excipient from the system. The input variable is excipient object.
- **findAll ():Excipient []** – The findAll method is generated from Spring data MongoDB framework. This method is used when the user wants to retrieve all of excipient data from the system. The result of this method is a list of excipient object.
- **findOne (id: String): Excipient** – The findOne method is used when the user wants to retrieve the excipient data from the system. The system gets an excipient object by the id of excipient. The input variable of this method is the id of the excipient that the user wants to find.

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	25 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

3.2.2.2.3. ExcipientService

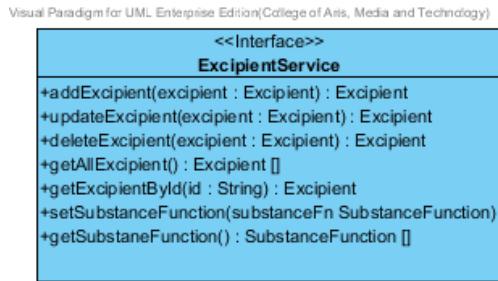


Figure 22: Excipient Service (Server Side)

ExcipientService is business processing logic for excipient entity. ExcipientService manages the excipient data through the ExcipientRepository interface. ExcipientService consists of 7 methods follow the list below this passage.

3.2.1.2.3.1: Attribute description

N/A

3.2.1.2.3.2: Method description

- **addExcipient (excipient: Excipient)** – The adding excipient method is used, when the user wants to add a new excipient to the database. This method adds a new excipient by input variable of excipient object. If the excipient object that input by the user is not contained in the database, this method will add a new excipient to the database and return the excipient object from the database to the user after the adding excipient is successful. On the other hand, when the excipient object that input by the user is contained in the database. This method will return a null value to the user.
- **updateExcipient (excipient: Excipient)** - The updating excipient method is used, when the user wants to update an existing excipient on in the database. This method update the existing excipient by input variable of excipient object. If the excipient object that input by the user is contained in the database, this method will update an existing excipient in the database and return the excipient object from the database to the user after the updating excipient is successful. On the other hand, when the excipient object that input by the user is not contained in the database. This method will return a null value to the user.
- **deleteExcipient (excipient: Excipient)** – The deleting excipient method is used when the user wants to deletes the existing excipient from the database. This method delete the excipient by input variable of excipient object. If the excipient object that input by the user is contained in the database, this method will delete an existing excipient from the database and return the excipient object to the user after the deleting excipient is successful. On the other hand, when the excipient object that input by

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	26 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

the user is not contained in the database. This method will return a null value to the user.

- **getAllExcipient() : Excipient []** – The getAllExcipient method is used, when the user wants to get all excipient data in the database. This method is return as a list of excipient object from the database.
- **getExcipientById(id : String) : Excipient** – Excipient – The getExcipientById method is used, when the user wants to get the excipient data in the system. This method gets excipient object from the database by id that input by the user. On the other hand, if the id that input by user is not contained in the database. This method will return null value to the user. The input variable of this method is the id of excipient that user wants to delete.
- **setSubstanceFunction(substanceFn : SubstanceFn)** – This method is used for setting the substance function to the substance object. The input variable of this method is the substance function that created by the user.
- **getSubstanceFunction() : SubstanceFunction[]** – This method is used for getting the substance function that set at setSubstanceFunction method.

3.2.2.2.4. ExcipientServiceImpl

Visual Paradigm for UML Enterprise Edition(College of Arts, Media and Technology)

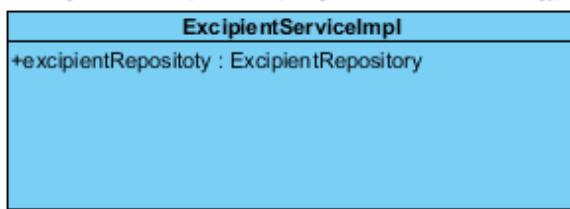


Figure 23: Excipient Service Implementation (Server Side)

ExcipientServiceImpl is the excipient service class that implements the method from ExcipientService. So, the method of ExcipientServiceImpl is same as ExcipientService.

3.2.2.2.4.1. Attribute description

- **ExcipientRepository** – the repository of excipient. This attribute is used for excipient data management.

3.2.2.2.4.2. Method description

Same as ExcipientService

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	27 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

3.3. Sub-Feature 7: Manage the drug formulation

The Sub-Feature 7 is the drug's formulation management. The users can add, update, delete and view the formulation. Each drug's formulation is created from the excipient. The relationship between drug's formulation and the excipient is illustrated in the Figure 24 below on this passage.

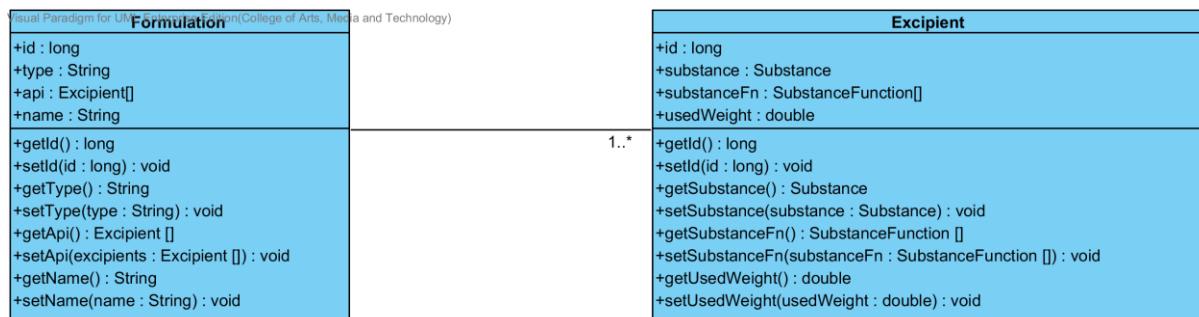


Figure 24: Entity Relationship between Formulation and Excipient

3.3.1. CD-Cl-03: Formulation Class diagram (Client Side)

3.3.1.1. Class diagram

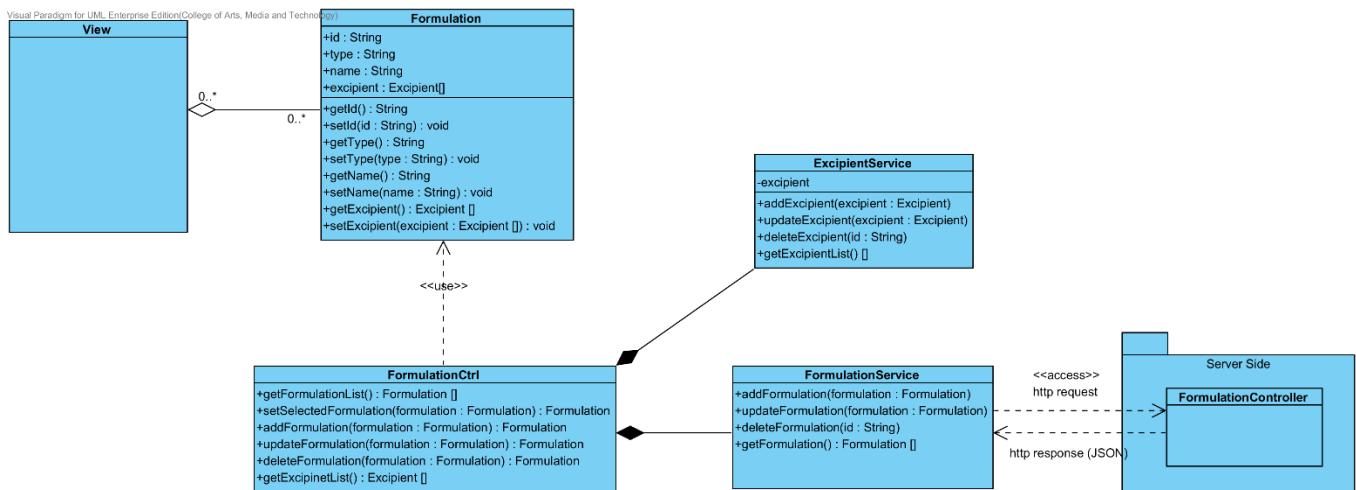


Figure 25: CD-Cl-03 – Formulation Class Diagram (Client Side).

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	28 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

3.3.1.2. Class description

From the Figure number 25 .It can divide into 4 Classes .The detail of each class is described on the next paragraph

3.3.1.2.1. Formulation Class

Visual Paradigm for UML Enterprise Edition|College of Arts, Media and Te

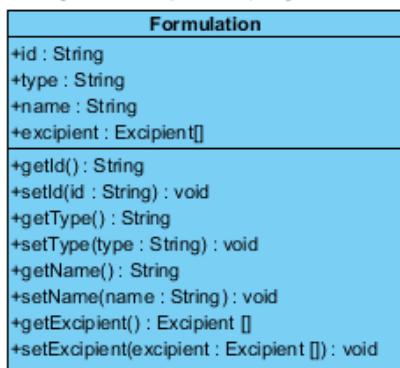


Figure 26: Formulation Class (Client Side)

The Formulation Class in Client Side is the model class that used for receiving the formulation data from the user and showing the formulation data to the user. This model is controlled by FormulationCtrl.

3.3.1.2.1.1. Attribute Description.

- **Id** – the identity of the Formulation class. Id attribute is a String number.
- **Name** – the name of the Formulation class. The Formulation attribute is a Formulation object.
- **Excipient** –the excipient of the Formulation. The user can set the excipient more than 1 excipient in each formulation.
- **Type** – the type of each formulation. There will be “Solution” or “Tablet”.

3.3.1.2.1.2. Method Description

- **Getter and Setter method** – It used when the system set value and get value.

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	29 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

3.3.1.2.2. FormulationCtrl

Visual Paradigm for UML Enterprise Edition(College of Arts, Media and Technology)

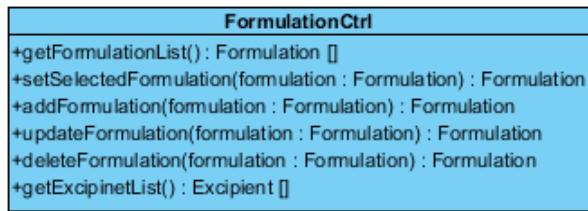


Figure 27 : FormulationCtrl (Client Side)

The FormulationCtrl is the formulation controller that used for controlling the formulation data that receive from the user or server. The FormulationCtrl consists of 5 method. The detail of each method are shown on next paragraph.

3.3.1.2.2.1. Attribute Description

N/A

3.3.1.2.2.2. Method Description

- **getFormulationList () : Formulation[]** - This method is used for getting the list of formulation from the server. The method will return the list of formulation object. On the other hand, if there are not any formulation object in the database. The getFormulationList method will return null.
- **setSelectedFormulation (formulation : Formulation) : Formulation** – This method is used for setting the formulation data that selected by the user. The selected formulation can use for updating or deleting formulation object. The input variable of this method is the formulation object that selected by the user.
- **addFormulation (formulation : Formulation) : Formulation** – This method is used for adding a new formulation data to the database. The method will receive data from the user. The input variable of this method is the formulation object the created by the user.
- **updateFormulation (formulation : Formulation) : Formulation** – This method is used for updating the existing formulation that user selected. The input variable of this method is the formulation object that user wants to update.
- **deleteFormulation (formulation : Formulation) : Formulation** – This method is used for deleting the existing formulation from the database. The input variable of this method is the formulation object that user wants to delete.
- **getExcipientList() : Excipient []** - This method is used for getting the list of excipient from the service. The method will return the list of excipient object. On the other hand, if there are not any excipient object in the database. The getExcipientList will return null.

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	30 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

3.3.1.2.3. ExcipientService

Visual Paradigm for UML Enterprise Edition(College of Arts, Medi

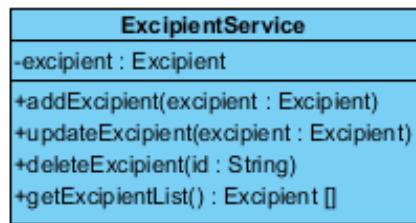


Figure 28 : Excipient Service (Client Side)

The Excipient Service is the service that use for adding, updating, deleting and getting data from the server. The ExcipientService can do its business by using http request such as POST, PUT, GET and DELETE.

3.3.1.2.3.1. Attribute Description

- **Excipient** – The Excipient object that receive from ExcipientCtrl and get from the user.

3.3.1.2.3.2. Method Description

- **addExcipient (excipient: Excipient)** – This method is used for adding a new excipient to database. The addExcipient method send the data that user input to the server by POST request. The input variable of this method is the excipient object that user created.
- **updateExcipient (excipient : Excipient)** – This method is used for updating the existing excipient object. The updateExcipient method send the edited data that user input to the server by PUT request. The input variable of this method is the excipient object that user wants to update.
- **deleteExcipient (id: String)** – This method is used for deleting the existing excipient object. The deleteExcipient method send the id of the excipient object to the server by DELETE request. The input variable of this method is the excipient object that user wants to delete.
- **getExcipientList() : Excipient []** – This method is used for getting the excipient object from the database. This method send the GET request to server for getting the data.

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	31 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

3.3.1.2.4. FormulationService

Visual Paradigm for UML Enterprise Edition(College of Arts, Media and Tech)

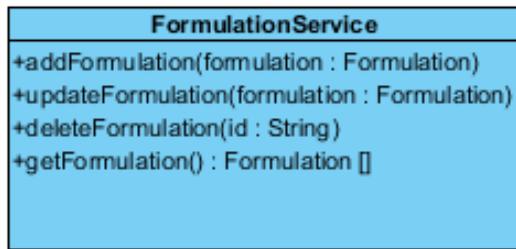


Figure 29 : Formulation Service (Client Side)

The Formulation Service is the service that use for adding, updating, deleting and getting data from the server. The FormulationService can do its business by using http request such as POST, PUT, GET and DELETE.

3.3.1.2.4.1. Attribute Description

- N/A

3.3.1.2.4.2. Method Description

- **addFormulation (formulation : Formulation)** – This method is used for adding a new formulation to database. The addFormulation method send the data that user input to the server by POST request. The input variable of this method is the formulation object that created by the user.
- **updateFormulation (formulation : Formulation)** – This method is used for updating the existing formulation object. The update Formulation method send the edited data that user input to the server by PUT request. The input variable of this method is formulation object that user wants update.
- **deleteFormulation(formulation: Formulation)** – This method is used for deleting the existing formulation object. The delete Formulation method send the id of the excipient object to the server by DELETE request. The input variable of this method is formulation object that user wants delete.
- **getFormulation() : Formulation[]** – This method is used for getting the formulation object from the server by GET request.

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	32 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

3.3.2. CD-SV-03: Formulation Class Diagram (Server Side)

3.3.2.1. Class diagram

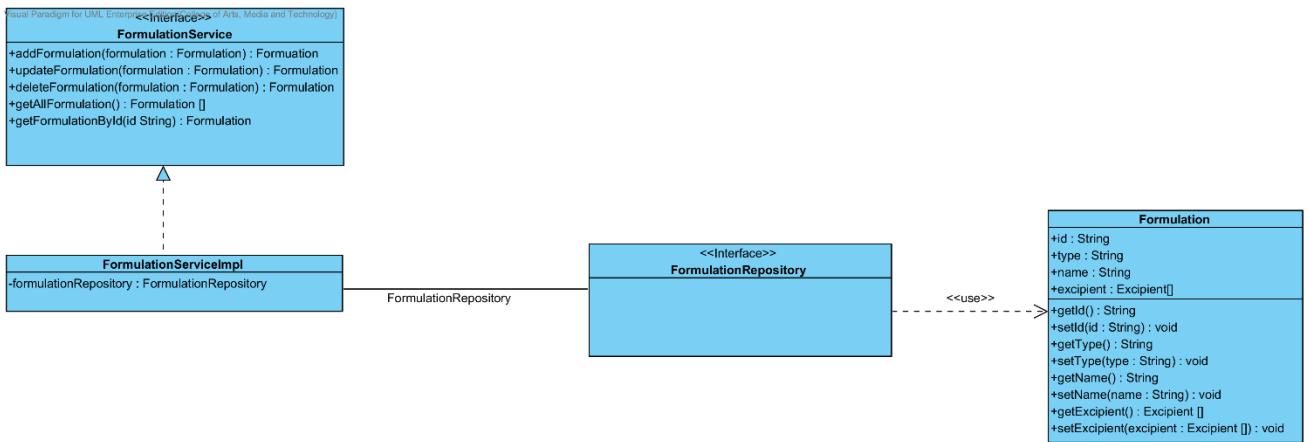


Figure 30 : CD-SV-03 – Formulation Class Diagram (Server Side).

3.3.2.2. Class description

From the figure 30, it can divide into 4 important classes. The detail of each class is described on the next paragraph.

3.3.2.2.1. Formulation class

Visual Paradigm for UML Enterprise Edition(College of Arts, Media and Technology)

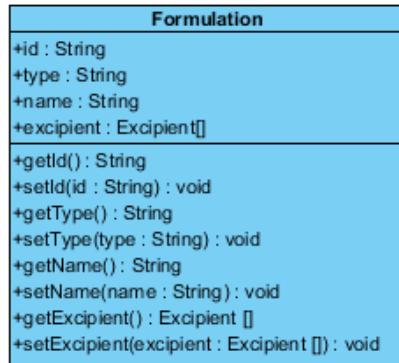


Figure 31: Formulation Class (Server Side)

Formulation class is an entity class that will be saved to the system. The Formulation class consists of 3 attributes follow the list below this passage.

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	33 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

3.3.2.2.1.1. Attribute description

- **Id** – the identity of the Formulation class. Id attribute is a String number.
- **Name** – the name of the Formulation class. The Formulation attribute is a Formulation object.
- **Excipient** –the excipient of the Formulation. The user can set the excipient more than 1 excipient in each formulation.
- **Type** – the type of each formulation. There will be “Solution” or “Tablet”.

3.3.2.2.1.2. Method description

- **Getter and Setter method** – It used when the system set value and get value.

3.3.2.2.2. FormulationRepository Interface

Visual Paradigm for UML Enterprise Edition(College of Arts, Media and Technology)

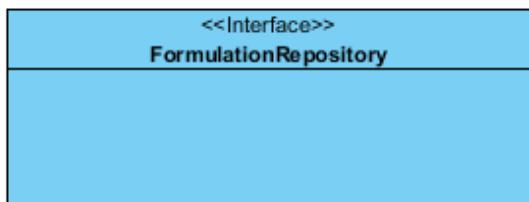


Figure 32 : Formulation Repository (Server Side)

FormulationRepository Interface is an interface that use for CRUD with entity classes in the system. The most of FormulationRepository interface's method is generated from Spring data MongoDB framework. FormulationRepository interface consists of 4 methods is shown below this passage.

3.3.2.2.2.1. Attribute description

N/A

3.3.2.2.2.2. Method description

- **Save (formulation: Formulation)** - The save method is generated from Spring data MongoDB framework. This method is used when the user wants to add a new Formulation or update existing Formulation to the system. The input variable is the Formulation object.
- **Delete (formulation: Formulation)** – The delete method is generated from Spring data MongoDB framework. This method is used when the user wants to delete the Formulation from the system. The input variable is Formulation object.
- **findAll ():Formulation []** – The findAll method is generated from Spring data MongoDB framework. This method is used when the user wants to retrieve all of Formulation data from the system. The result of this method is a list of Formulation object.

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	34 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

- **findOne (id: String): Formulation** – The findOne method is used when the user wants to retrieve the Formulation data from the system. The system gets a Formulation object by the id of Formulation.

3.3.2.2.3. FormulationService

Visual Paradigm for UML Enterprise Edition(College of Arts, Media and Technology)

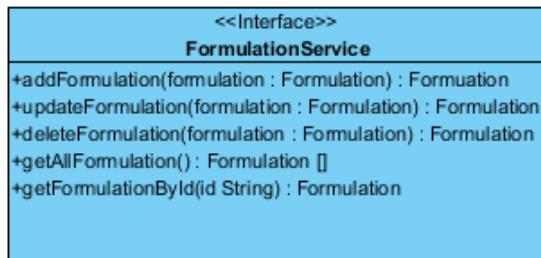


Figure 33: Formulation Service (Server Side)

FormulationService is business processing logic for Formulation entity. FormulationService manages the Formulation data through the FormulationRepository interface. FormulationService consists of 6 methods follow the list below this passage.

3.4.1.2.3.1: Attribute description

N/A

3.4.1.2.3.2: Method description

- **addFormulation (formulation: Formulation)** – The adding Formulation method is used, when the user wants to add a new Formulation to the database. This method adds a new Formulation by input variable of Formulation object. If the Formulation object that input by the user is not contained in the database, this method will add a new Formulation to the database and return the Formulation object from the database to the user after the adding Formulation is successful. On the other hand, when the Formulation object that input by the user is contained in the database. This method will return a null value to the user.
- **updateFormulation (formulation: Formulation)** - The updating Formulation method is used, when the user wants to update an existing Formulation on in the database. This method update the existing Formulation by input variable of Formulation object. If the Formulation object that input by the user is contained in the database, this method will update an existing Formulation in the database and return the Formulation object from the database to the user after the updating Formulation is successful. On the other hand, when the Formulation object that input by the user is not contained in the database. This method will return a null value to the user.

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	35 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

- **deleteFormulation (formulation:Formulation)** - The deleting Formulation method is used when the user wants to delete the existing Formulation from the database. This method delete the Formulation by input variable of Formulation object. If the Formulation object that input by the user is contained in the database, this method will delete an existing Formulation from the database and return the Formulation object to the user after the deleting Formulation is successful. On the other hand, when the formulation object that input by the user is not contained in the database. This method will return a null value to the user.
- **getAllFormulation() : Formulation []** – The getAllFormulation method is used, when the user wants to get all Formulation data in the database. This method is return as a list of Formulation object from the database.
- **getFormulationById(id : String) : Formulation** – The getFormulationById method is used, when the user wants to get the Formulation data in the system. This method gets Formulation object from the database by id that input by the user. On the other hand, if the id that input by user is not contained in the database. This method will return null value to the user.

3.3.2.2.4. FormulationServiceImpl

Visual Paradigm for UML Enterprise Edition(College of Arts, Media and Technology)

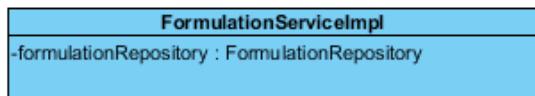


Figure 34 : Formulation Service Implementation (Server Side)

FormulationServiceImpl is the formulation service class that implements the method from FormulationService. So, the method of FormulationServiceImpl is same as FormulationService.

3.3.2.2.4.1. Attribute description

- **FormulationRepository** – the repository of Formulation. This attribute is used for Formulation data management.

3.3.2.2.4.2. Method description

Same as FormulationService

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	36 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

3.4. Sub-Feature 2: Calculate the drug reformulation by using the inference engine.

The Sub-Feature 2 is the making reformulation. The user can input the formulation and value for making reformulation by use the Rule base, Case Base Reasoning and Hybrid Reasoning. The user can save their reformulation to the history for viewing it after.

3.4.1. CD-CI- 04: Reformulation Class Diagram (Client Side)

3.4.1.1. Class diagram

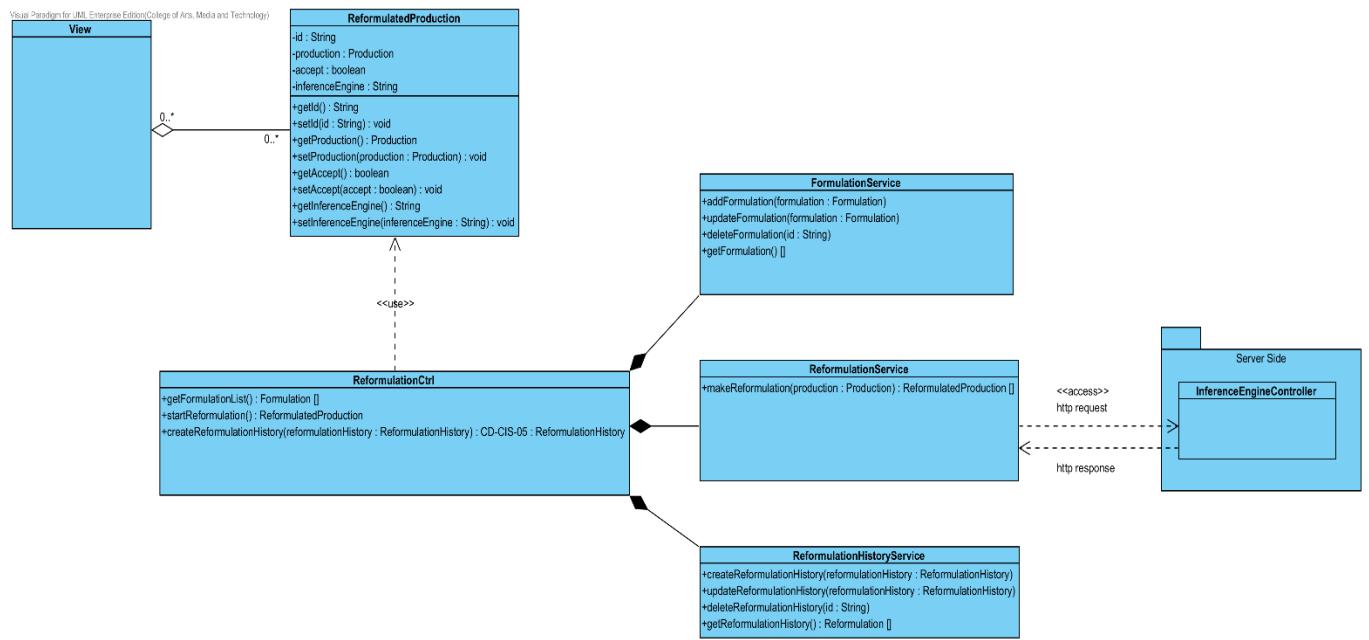


Figure 35: CD-CI-04 – Reformulation Class Diagram (Client Side).

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	37 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

3.4.1.2. Class description

From the Figure number 35 .It can divide into 5 Classes .The detail of each class is described on the next paragraph

3.4.1.2.1. ReformulatedProduction Class

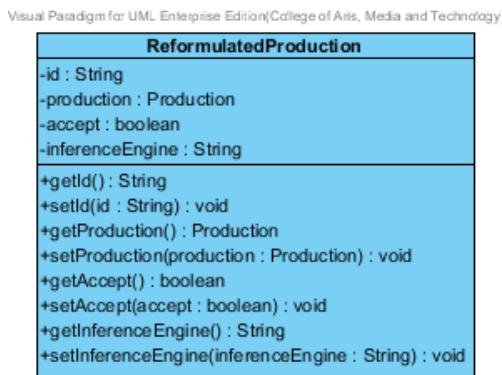


Figure 36: Reformulated Production Class (Client Side)

The ReformulatedProduction Class in Client Side is the model class that used for receiving reformulated production data from the user and showing the reformulated production data to the user. This model is controlled by ReformulationCtrl.

3.4.1.2.1.1. Attribute description

- **Id** – The identification of reformulated production
- **Production** – the production has been reformulated by inference engine.
- **Accept** – the default is “false”, if the accept equal “true”. Which mean this product is good for base data.
- **Inference Engine** – the inference engine of this reformulated Production.

3.4.1.2.1.2. Attribute description

- Getter and Setter Method.

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	38 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

3.4.1.2.2. ReformulationCtrl

Visual Paradigm for UML Enterprise Edition(College of Arts, Media and Technology)

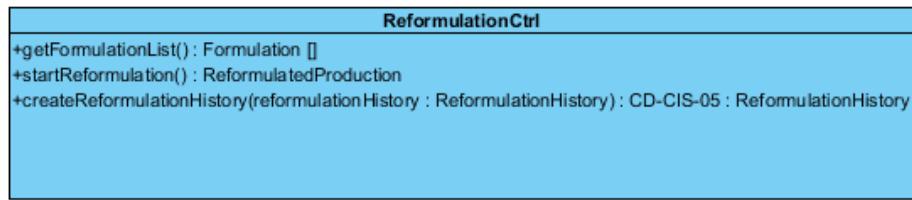


Figure 37 : ReformulationCtrl (Client Side)

The Reformulation Ctrl is the controller that used for managing the production data when the user making reformulation. The ReformulationCtrl also get the production after reformulate process for showing result to the user. The ReformulationCtrl consists of 3 method. The detail of each method are shown on next paragraph.

3.4.1.2.2.1. Attribute Description

N/A

3.4.1.2.2.2. Method Description

- **getFormulationList() : Formulation[]** – This method is used for getting the formulation list for making reformulation. On the other hand if there are not any formulation object it will return null.
- **startReformulation() : ReformulatedProduction** – This method is used for reformulating production. This method will get the reformulated production after reformulation. The input variable is the production object that the user wants to make reformulation.
- **createReformulationHistory(reformulatedProduction : ReformulatedProduction)** – This method is used for creating reformulation history after the user get reformulation result. The input variable is the reformulatedProduction that user wants to save to the history.

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	39 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

3.4.1.2.3. FormulationService

Visual Paradigm for UML Enterprise Edition(College of Arts, Media and Technology)

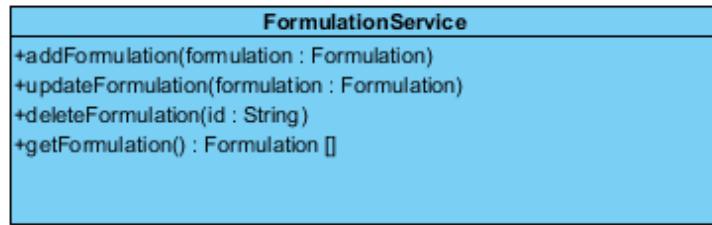


Figure 38 : Formulation Service (Client Side)

The Formulation Service is the service that use for adding, updating, deleting and getting data from the server. The FormulationService can do its business by using http request such as POST, PUT, GET and DELETE.

3.4.1.2.3.1. Attribute Description

- N/A

3.4.1.2.3.2. Method Description

- **addFormulation (formulation : Formulation)** – This method is used for adding a new formulation to database. The addFormulation method send the data that user input to the server by POST request. The input variable of this method is the formulation that created by the user.
- **updateFormulation (formulation : Formulation)** – This method is used for updating the existing formulation object. The updateFormulation method send the edited data that user input to the server by PUT request. The input variable of this method is the formulation object that user wants to update.
- **deleteFormulation(id : String)** – This method is used for deleting the existing formulation object. The deleteFormulation method send the id of the excipient object to the server by DELETE request. The input variable of this method is the id of formulation that user wants to delete.
- **getFormulation() : Formulation[]** – This method is used for getting the formulation object from the server by GET request.

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	40 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

3.4.1.2.4. ReformulationService

Visual Paradigm for UML Enterprise Edition(College of Arts, Media and Technology)

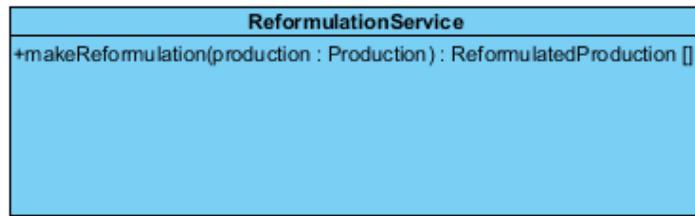


Figure 39 : Reformulation Service (Client Side)

The Reformulation Service is the service that use for making drug reformulation with the server.

3.4.1.2.4.1. Attribute Description

- N/A

3.4.1.2.4.2. Method Description

- **makeReformulation (production : Production)** – This method is used for making drug reformulation. The method will send the production object to server by POST request. After that, the method will receive the reformulated production list from the server.

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	41 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

3.4.1.2.5. ReformulationHistoryService

Visual Paradigm for UML Enterprise Edition(College of Arts, Media and Technology)

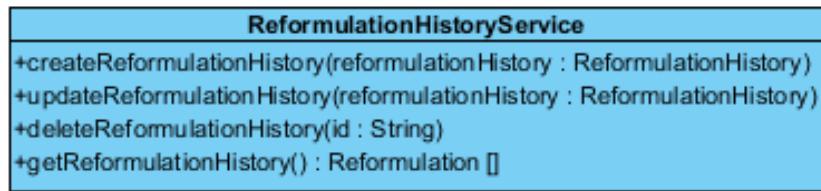


Figure 40 : ReformulationHistoryService (Client Side)

The Reformulation History Service is the service that use for adding, updating, deleting and getting data from the server. The Reformulation History can do its business by using http request such as POST, PUT, GET and DELETE.

3.4.1.2.5.1. Attribute Description

- N/A

3.4.1.2.5.2. Method Description

- **createReformulationHistory (reformulationHistory : ReformulationHistory)** – This method is used for adding a new reformulation history to database. The createReformulationHistory method sends the data that user input to the server by POST request. The input variable of this method is the reformulationHistory object.
- **updateReformulationHistory (reformulationHistory : ReformulationHistory)** – This method is used for updating the existing reformulation history. The updateReformulationHistory method sends the edited data that user input to the server by PUT request. The input variable of this method is the reformulationHistory object.
- **deleteReformulationHistory (reformulationHistory : Reformulation)** This method is used for deleting the existing reformulation history. The deleteReformulationHistory method sends the id of the reformulation history id to the server by DELETE request. The input variable of this method is the reformulation history object.
- **getReformulationHistory : ReformulationHistory[]** – This method is used for getting the reformulation history from the server by GET request.

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	42 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

3.4.2. CD-SV-04: Reformulation Class Diagram (Server Side)

3.4.2.1. Class diagram

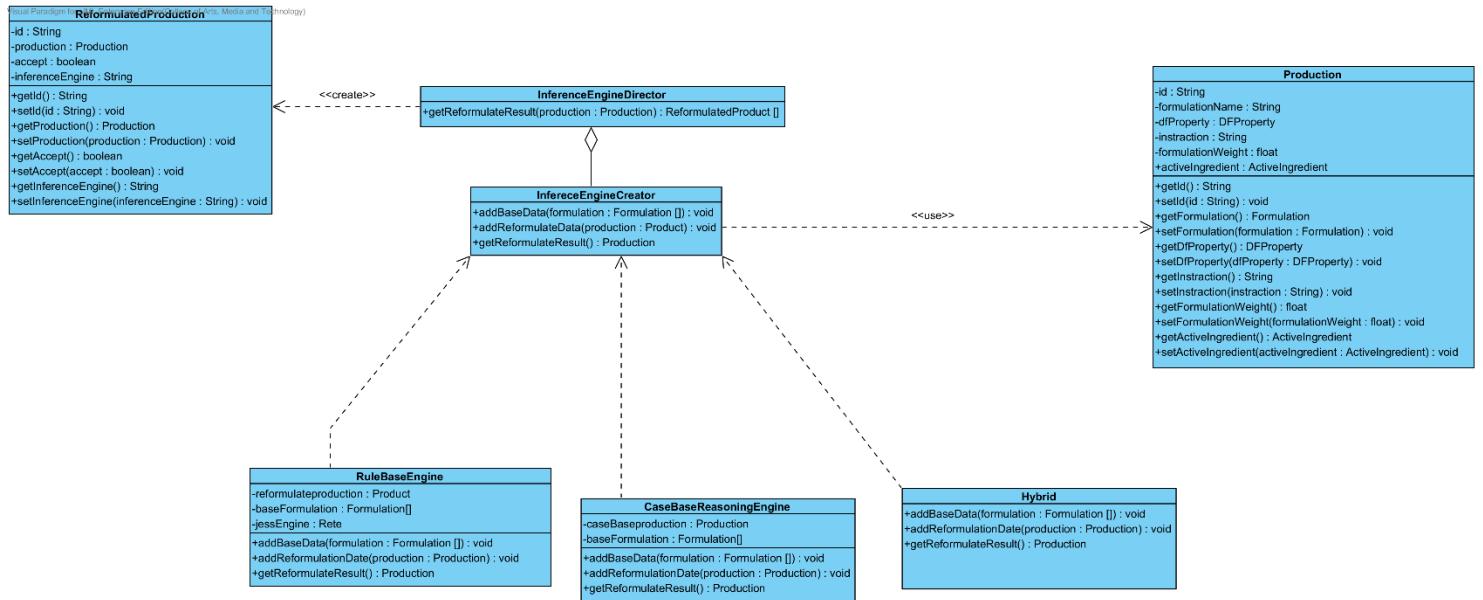


Figure 41 : CD-SV-04 – Reformulation Class Diagram (Server Side).

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	43 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

3.4.2.2. Class description

From the figure 41, it can divide into 7 important classes. The detail of each class is described on the next paragraph.

3.4.2.2.1. Production class

Visual Paradigm for UML Enterprise Edition(College of Arts, Media and Technology)

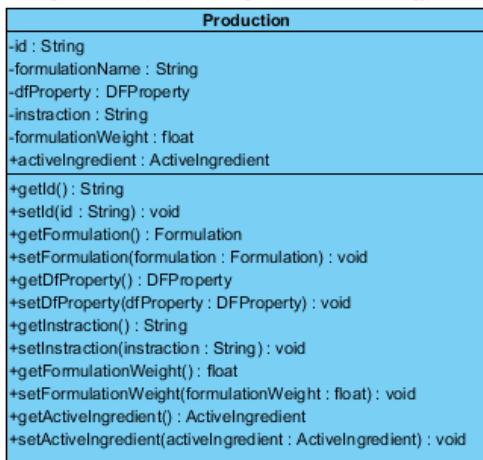


Figure 42 : Production Class (Server Side)

Production is a part of drug reformulation. Production object is created after reformulation successful. The production class consists of 6 attributes follow the list below this passage.

3.4.2.2.1.1. Attribute description

- **Id** – the identity of the production class. Id attribute is a long number.
- **Formulation Name** – the name of formulation that use for creating a new production.
- **DF Property** - The pharmaceutical value that come from the lab or a real drug.
- **Instruction** – The instruction about how to create this production. The instruction is created after reformulation.
- **FormulationWeight** – The weight of formulation in each production.
- **Active Ingredient** – The ingredient properties that user want to make reformulation.

3.4.2.2.1.2. Method description

- **Getter and Setter method** – It used when the system set value and get value.

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	44 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

3.4.2.2.2. InferenceEngineDirector

Visual Paradigm for UML Enterprise Edition(College of Arts, Media and Technology)

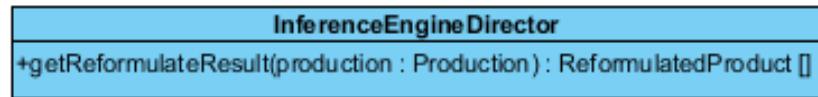


Figure 43: Inference Engine Director (Server Side)

The InferenceEngineDirector is director for managing the inference engine such as Rule base, Case base reasoning and hybrid reasoning. This class will help the inference engine working correctly.

3.4.2.2.2.1. Attribute description

N/A

3.4.2.2.2.2. Method description

- **getReformulateResult(production: Production): ReformulatedProduct[]**
 - This method will receive the production object from client side. After that, it will use the inference engine for reformulating the production. Finally, this method will send the reformulation result that call reformulatedProduct to the user. The input variable of this method is the production that user want to reformulate.

3.4.2.2.3. InferenceEngineCreator

Visual Paradigm for UML Enterprise Edition(College of Arts, Media and Technology)

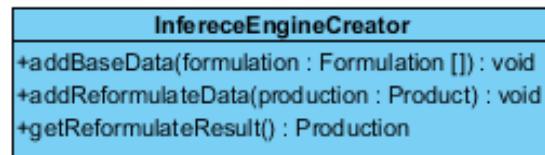


Figure 44: Inference Engine Creator (Server Side)

The inference engine creator will create the inference engine follow the InferenceEngineDirector command. This interface consists of 3 method that will be show on below this paragraph.

3.2.1.2.3.1: Attribute description

N/A

3.2.1.2.3.2: Method description

- **addBaseData(formulation : Formulation[]) : void** – This method is used for adding the base data from the database. The base data is used by the inference engine for calculating reformulation. The input variable of this method is the formulation object that retrieve from the database.

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	45 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

- **addReformulateData(production : Production) : void** – This method is used for adding the production data that user want to reformulation. The production will calculate with the base data. The input variable is the production that user wants to make reformulation.
- **getReformulateResult() : Production** – After reformulation, this method will return the reformulation result to the user.

3.4.2.2.4. RuleBaseEngine

Visual Paradigm for UML Enterprise Edition(College of Arts, Media and Technology)

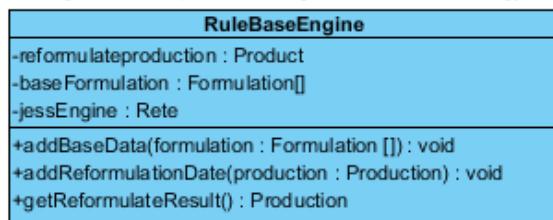


Figure 45: Rule Base Engine (Server Side)

Rule Base Engine, the rule base engine is the one of inference engine that use for reformulating the production. The Rule Base engine implement the method from inference engine creator.

3.4.2.2.4.1. Attribute description

- **Reformulateproduction** – the production that use for reformulation.
- **BaseFormulation** –the base formulation is get from the database.
- **jessEngine** – the rule base engine is used for making reformulation.

3.4.2.2.4.2. Method description

Same as InferenceEnigneCreator

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	46 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

3.4.2.2.5. CaseBaseReasoningEngine

Visual Paradigm for UML Enterprise Edition(College of Arts, Media and Technology)

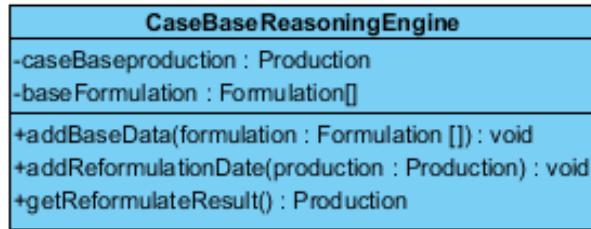


Figure 46: Rule Base Engine (Server Side)

Case Base Reasoning Engine, the Case Base Reasoning engine is the one of inference engine that use for reformulating the production. The Case Base Reasoning engine implement the method from inference engine creator.

3.4.2.2.5.1. Attribute description

- **Reformulateproduction** – the production that use for reformulation.
- **BaseFormulation** –the base formulation is get from the database.

3.4.2.2.5.2. Method description

Same as InferenceEnigneCreator

3.4.2.2.6. HybridReasoningEnigne

Visual Paradigm for UML Enterprise Edition(College of Arts, Media and Technology)

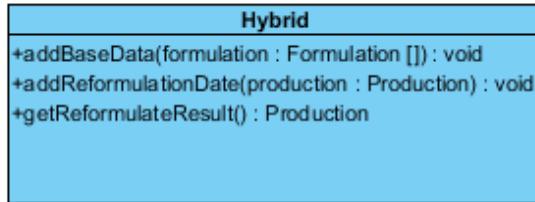


Figure 47: Rule Base Engine (Server Side)

Hybrid Engine, the hybrid engine is the one of inference engine that use for reformulating the production. The Hybrid engine implement the method from inference engine creator.

3.4.2.2.6.1. Attribute description

- **N/A**

3.4.2.2.6.2. Method description

Same as InferenceEnigneCreator

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	47 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

3.4.2.2.7. ReformulatedProduction

Visual Paradigm for UML Enterprise Edition|College of Arts, Media and Technology

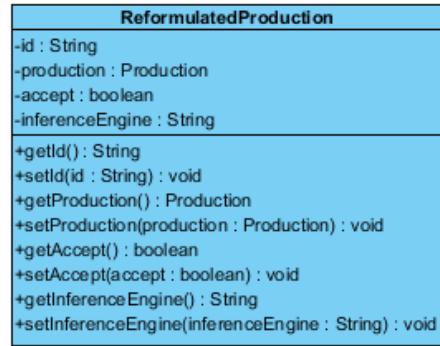


Figure 48: Reformulated Production (Server Side)

The Reformulated production is created after reformulation. The attribute of this class will show below this paragraph.

3.4.2.2.7.1. Attribute description

- **Id** – The identification of reformulated production
- **Production** – the production has been reformulated by inference engine.
- **Accept** – the default is “false”, if the accept equal “true”. Which means this product is good for base data.
- **Inference Engine** – the inference engine of this reformulated Production.

3.4.2.2.7.2. Method description

Getter and Setter Method.

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	48 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

3.5. Sub-Feature 3: View the drug reformulation history.

The Sub-Feature 3 is view the drug reformulation history. The user can view the reformulation history that user save. The relationship between reformulation history and reformulated production is illustrated in the Figure 49 below on this passage.

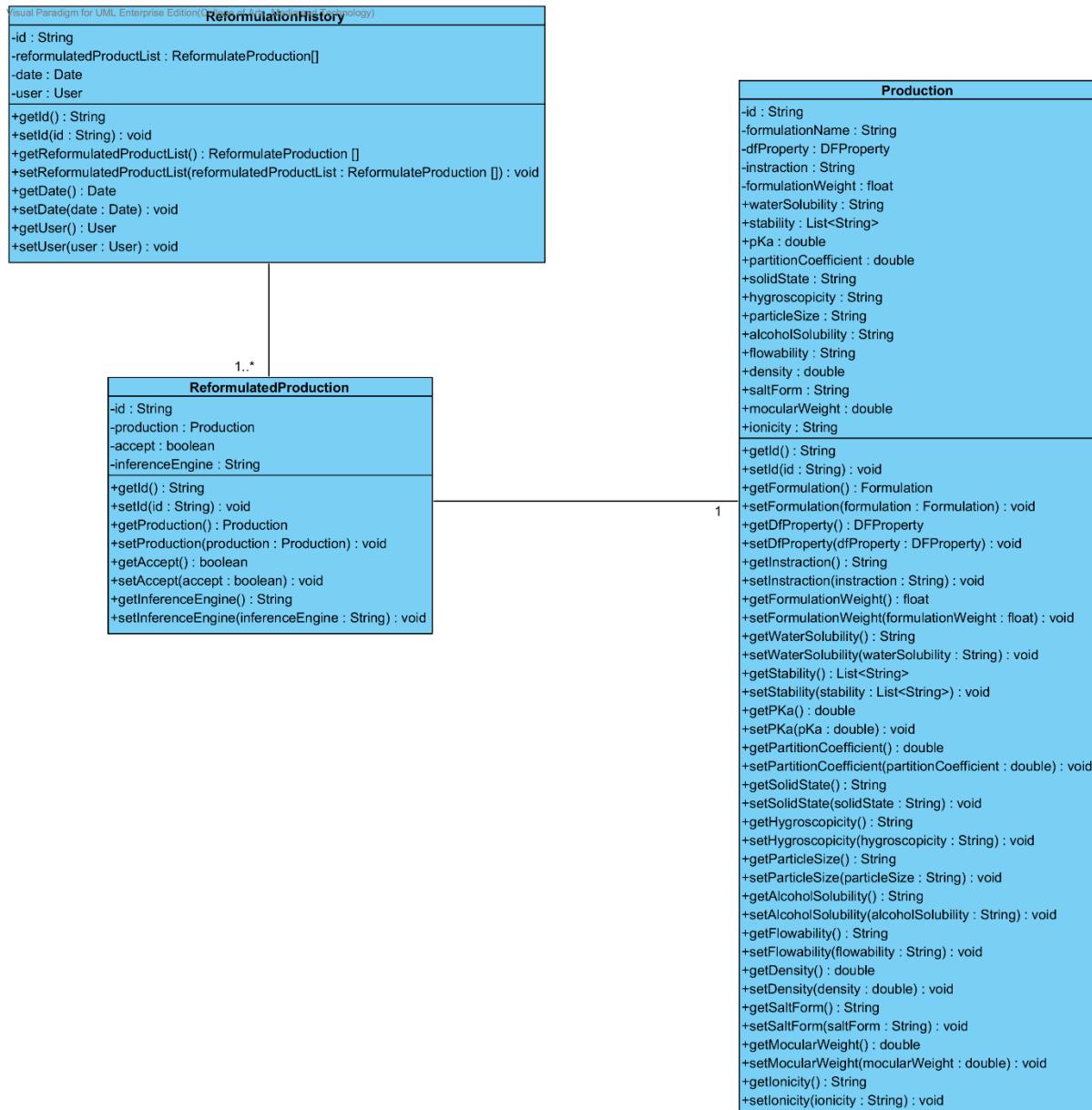


Figure 49: The Relationship between Reformulation History and Reformulated Production

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	49 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

3.5.1. CD-CI- 05: Reformulation History Class Diagram (Client Side)

3.5.1.1. Class diagram

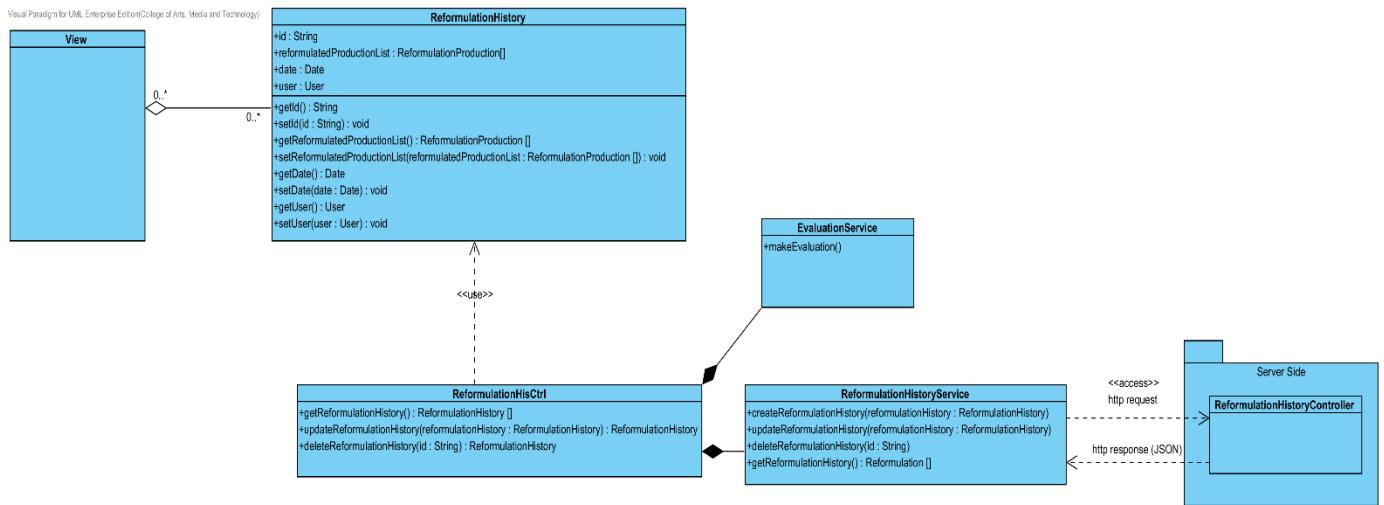


Figure 50: CD-CI-05 – Reformulation History Class Diagram (Client Side)

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	50 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

3.5.1.2. Class description

From the Figure number 51 .It can divide into 3 Classes .The detail of each class is described on the next paragraph

3.5.1.2.1. ReformulationHistory Class

Visual Paradigm for UML Enterprise Edition(College of Arts, Media and Technology)

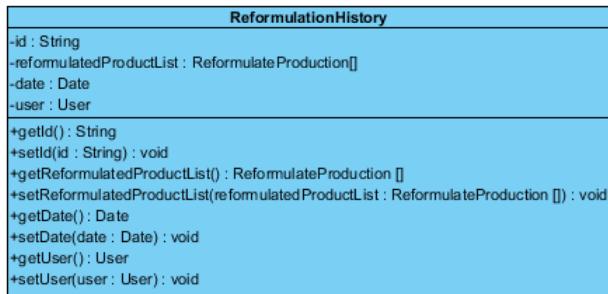


Figure 51: Reformulation History Class (Client Side)

The Reformulation history Class in Client Side is the model class that used for receiving excipient data from the user and showing an excipient data to the user. This model is controlled by ReformulationHistoryCtrl.

3.5.1.2.1.1. Attribute description

- **Id** – the identity of the reformulation history class. Id attribute is a String.
- **ReformulationProductionlist** – the list of reformulated production.
- **Date** –the created date.
- **User** – the user object who create the reformulation history.

3.5.1.2.1.2. Method description

- **Getter and Setter method** – It used when the system set value and get value.

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	51 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

3.5.1.2.2. ReformulationHistoryCtrl

Visual Paradigm for UML Enterprise Edition(College of Arts, Media and Technology)

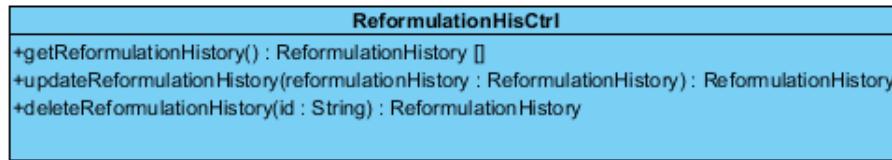


Figure 52 : Reformulation History Ctrl (Client Side)

The ReformulationHistoryCtrl is the reformulation history controller that used for managing the excipient data that receive from the user or server. The ReformulationHistoryCtrl consists of 3 method. The detail of each method are shown on next paragraph.

3.5.1.2.2.1. Attribute Description

N/A

3.5.1.2.2.2. Method Description

- **getReformulationHistory() : ReformulationHistory[]** - This method is used for getting the list of reformulation history from the server. The method will return the list of reformulation history object. On the other hand, if there are not any reformulation history object in the database. This method will return null.
- **UpdateReformulationHistory(reformulationHistory : ReformulationHsitory) : ReformulationHistory** – This method is used for updating the existing reformulation history that user selected. This method will use the service for saving an edited reformulation history. The input variable of this method is the reformulationHistory object that user wants to update
- **DeleteReformulationHistory(id : String) : ReformulationHistory** – This method is used for deleting the existing excipient from the database. The input variable of this method is the id of reformulationHistory object that user want to delete.

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	52 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

3.5.1.2.3. EvaluationService

Visual Paradigm for UML Enterprise Edition(College of Arts,

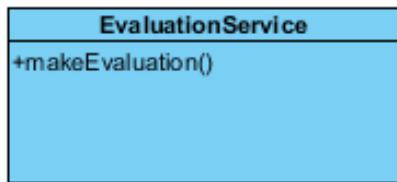


Figure 53 : Substance Service (Client Side)

The Evaluation Service is used for evaluate the reformulation that user making.

3.5.1.2.3.1. Attribute Description

- N/A

3.5.1.2.3.2. Method Description

- **makeEvaluation ()** – This method is used for evaluating the reformulation that user making.

3.5.1.2.4. ReformulationHistoryService

Visual Paradigm for UML Enterprise Edition(College of Arts, Media and Technology)

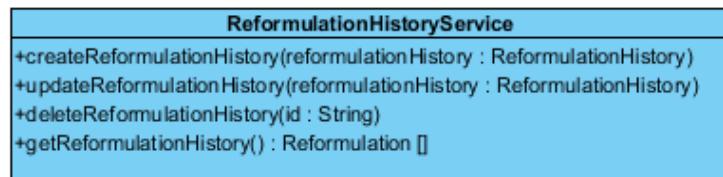


Figure 54 : Reformulation History Service (Client Side)

The Reformulation History Service is the service that use for adding, updating, deleting and getting data from the server. The Reformulation history Service can do its business by using http request such as POST, PUT, GET and DELELTE.

3.5.1.2.4.1. Method Description

- **createReformulationHistory (reformulationHistory: ReformulationHistory)** – This method is used for adding a new reformulation history to database. The createReformulationHistory method send the data that user input to the server by POST request. The input variable of this method is the reformulationHistory that user wants to create.

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	53 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

- **updateReformulationHistory (reformulationHistory : ReformulationHistory)** – This method is used for updating the existing reformulation history object. The update Reformulation History method send the edited data that user input to the server by PUT request. The input of this method is the reformulation history that user wants to update.
- **deleteReformulationHistory (id: String)** – This method is used for deleting the existing reformulation history object. The delete Reformulation History method send the id of the reformulation history object to the server by DELETE request. The input of this method is the reformulation history that user wants to delete.
- **getReformulationHistory() : ReformulationHistory []** – This method is used for getting the reformulation history object from the database. This method send the GET request to server for getting the data.

3.5.2. CD-SV-05: Reformulation History Class Diagram (Server Side)

3.5.2.1. Class diagram

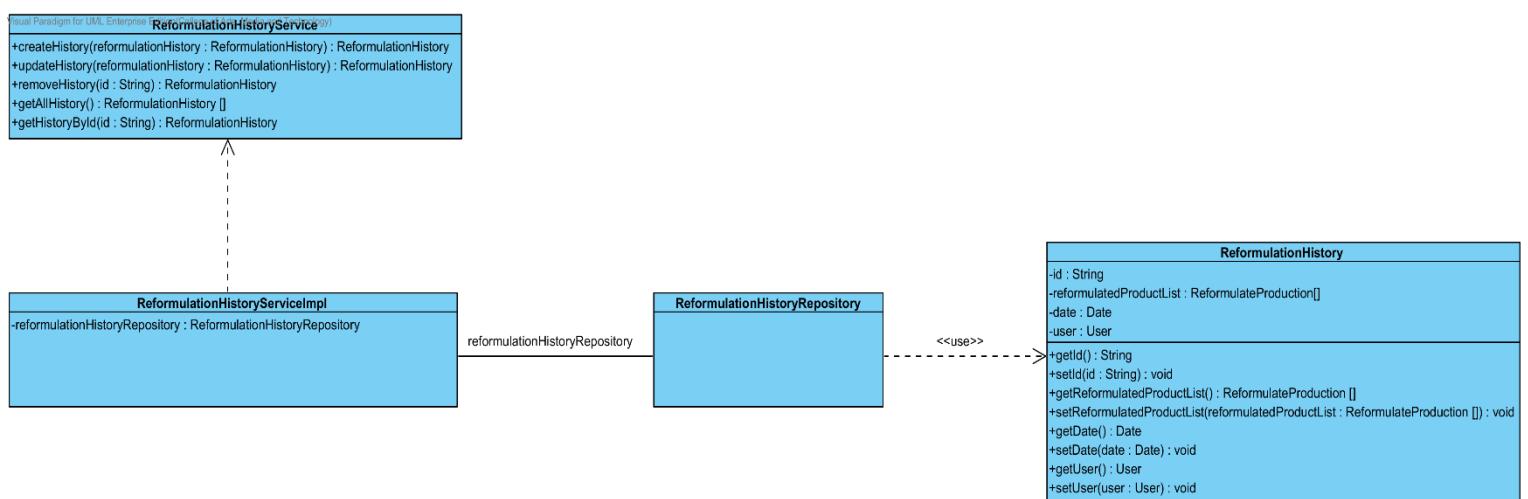


Figure 55 : CD-SV-05: Reformulation History Class Diagram (Server Side)

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	54 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

3.5.2.2. Class description

From the figure 57, it can divide into 4 important classes. The detail of each class is described on the next paragraph.

3.5.2.2.1. ReformulationHistory class

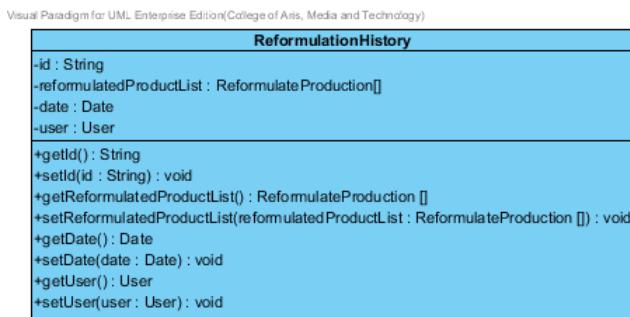


Figure 56 : Excipient Class (Server Side)

Reformulation History is the history of the reformulated production. Reformulation History class is an entity class that will be saved to the system. The excipient class consists of 4 attributes follow the list below this passage.

3.5.2.2.1.1. Attribute description

- **Id** – the identity of the reformulation history class. Id attribute is a String.
- **ReformulationProductionlist** – the list of reformulated production.
- **Date** –the created date.
- **User** – the user object who create the reformulation history.

3.5.2.2.1.2. Method description

- **Getter and Setter method** – It used when the system set value and get value.

3.5.2.2.2. ReformulationRepository Interface

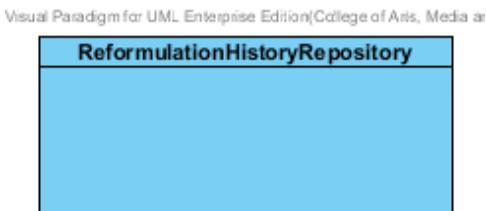


Figure 57: Reformulation History Repository (Server Side)

ReformulationHistoryRepository Interface is an interface that use for CRUD with entity classes in the system. All of ReformulationHistoryRepository interface's method is generated from Spring data MongoDB framework. ReformulationHistoryRepository interface consists of 4 methods is shown below this passage.

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	55 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

3.5.2.2.2.1. Attribute description

N/A

3.5.2.2.2.2. Method description

- **Save (reformulationHistory: ReformulationHistory)** - The save method is generated from Spring data MongoDB framework. This method is used when the user wants to add a new reformulation history or update existing reformulation history to the system. The input variable is the reformulation history object.
- **Delete (reformulationHistory: ReformulationHistory)** – The delete method is generated from Spring data MongoDB framework. This method is used when the user wants to delete the reformulation history from the system. The input variable is reformulation history object.
- **findAll ():ReformulationHistory []** – The findAll method is generated from Spring data MongoDB framework. This method is used when the user wants to retrieve all of reformulation history data from the system. The result of this method is a list of reformulation history object.
- **findOne (id: String): ReformulationHistory** – The findOne method is used when the user wants to retrieve the reformulation history data from the system. The system gets the reformulation history object by the id of reformulation history.

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	56 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

3.5.2.2.3. ReformulationHistoryService

Visual Paradigm for UML Enterprise Edition(College of Arts, Media and Technology)



Figure 58: Reformulation History Service (Server Side)

ReformulationHistoryService is business processing logic for reformulation history entity. ReformulationHistoryService manages the reformulation history data through the ReformulationHistoryRepository interface. ReformulationHistoryService consists of 5 methods follow the list below this passage.

3.2.1.2.3.1: Attribute description

N/A

3.2.1.2.3.2: Method description

- **createHistory (reformulationHistory: ReformulationHistory)** – The create history method is used, when the user wants to add a new history to the database. This method adds a new history by input variable of history object. If the history object that input by the user is not contained in the database, this method will add a new reformulation history to the database and return the reformulation history object from the database to the user after the adding reformulation history is successful. On the other hand, when the reformulation history object that input by the user is contained in the database. This method will return a null value to the user.
- **updateHistory (reformulationHistory: ReformulationHistory)** - The updating reformation history method is used, when the user wants to update an existing reformulation history on in the database. This method update the existing reformulation history by input variable of reformulation history object. If the reformulation history object that input by the user is contained in the database, this method will update an existing reformulation history in the database and return the reformulation history object from the database to the user after the updating reformulation history is successful. On the other hand, when the reformulation history object that input by the user is not contained in the database. This method will return a null value to the user.
- **Removehistory (id : String)** – The deleting formulation history method is used when the user wants to deletes the existing reformulation history from the database. This method delete the reformulation history by input variable of formulation history object. If the reformulation history object that input by the user is contained in the database, this method will delete an existing reformulation history from the database and return

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	57 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

the reformulation history object to the user after the deleting reformulation history is successful. On the other hand, when the reformulation history object that input by the user is not contained in the database. This method will return a null value to the user.

- **getAllHistory() : ReformulationHistory []** – The getAllHistory method is used, when the user wants to get all reformulation history data in the database. This method is return as a list of reformulation history object from the database.
- **getHistoryById(id : String) : Excipient** – The getHistoryById method is used, when the user wants to get the history data in the system. This method gets reformulation history object from the database by id that input by the user. On the other hand, if the id that input by user is not contained in the database. This method will return null value to the user.

3.5.2.2.4. ReformulationHistoryServiceImpl



Figure 59: Reformulation History Service Implementation (Server Side)

ReformulationHistoryServiceImpl is the reformulation history service class that implements the method from ReformulationHistoryService. So, the method of ReformulationHistoryServiceImpl is same as ReformulationHistoryService.

3.5.2.2.4.1. Attribute description

- **ReformulationHistoryRepository** – the repository of reformulation history. This attribute is used for reformulation history data management.

3.5.2.2.4.2. Method description

Same as ReformulationHistoryService

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	58 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

Chapter 4 | Sequence Diagram

In the 2nd progress the URS is related with the list of sequence diagram that shown below this passage.

No	Sub-Feature Name	URS No.	URS Name	Sequence Diagram
6	Manage the drug substance	URS-09	The user adds a new substance into the system.	SQD-CI-01, SQD-SV-01
		URS-10	The user updates an existing substance into the system.	SQD-CI-02, SQD-SV-02
		URS-11	The user deletes an existing substance from the system.	SQD-CI-03, SQD-SV-03
		URS-12	The user views the substance in the system.	SQD-CI-04, SQD-SV-04
7	Manage the drug excipient	URS-13	The user adds a new excipient to the system.	SQD-CI-05, SQD-SV-05
		URS-14	The user updates an existing drug excipient in the system.	SQD-CI-06, SQD-SV-06
		URS-15	The user delete an existing drug excipient in the system.	SQD-CI-07, SQD-SV-07
		URS-16	The user views all the drug excipient in the system.	SQD-CI-08, SQD-SV-08
8	Manage the drug formulation	URS-17	The user adds a new drug formulation case into the system.	SQD-CI-09, SQD-SV-09
		URS-18	The user updates an existing drug formulation case in the system.	SQD-CI-10, SQD-SV-10
		URS-19	The user deletes an existing drug formulation case in the system.	SQD-CI-11, SQD-SV-11
		URS-20	The user views all of the formulation in the system.	SQD-CI-12, SQD-SV-12
2	Calculate the drug reformulation by using the inference engine.	URS-06	The user calculates a drug reformulation by using an inference engine.	SQD-CI-13 , SQD-SV-13
3	View the drug reformulation history	URS-07	The user views their drug reformulation history.	SQD-CI-14, SQD-SV-14

Table 2: The Relationship between the Use Case and Sequence Diagram

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	59 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

4.1. Sub-Feature 6: Manage the drug substance

4.1.1. URS-09: The user adds a new substance to the system.

In a sequence diagram, the user can add a new substance to the system. Firstly, the user opens the substance adding page, then the user input substance data such as name, water solubility, stability and etc. The substance controller gets an input data from the user. After that, the substance controller send a new substance data to appropriate service for adding a new substance. Finally, the system show a new substance with the adding substance successful page.

4.1.1.1. SQD-CI-01: The user adds a new substance to the system (Client Side).

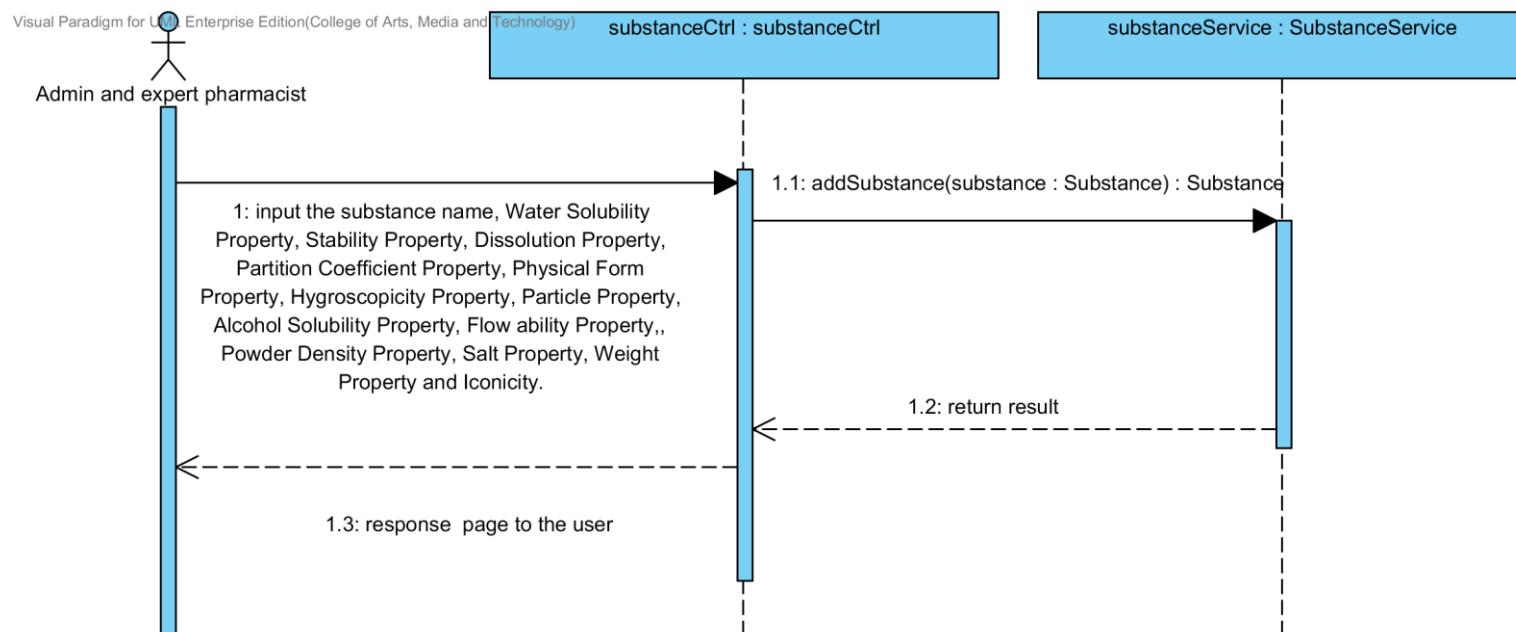


Figure 60 : SQD-CI-01- The user adds a new substance to the system (Client Side).

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	60 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

4.1.1.2. SQD-SV-01: The user adds a new substance to the system (Server Side)

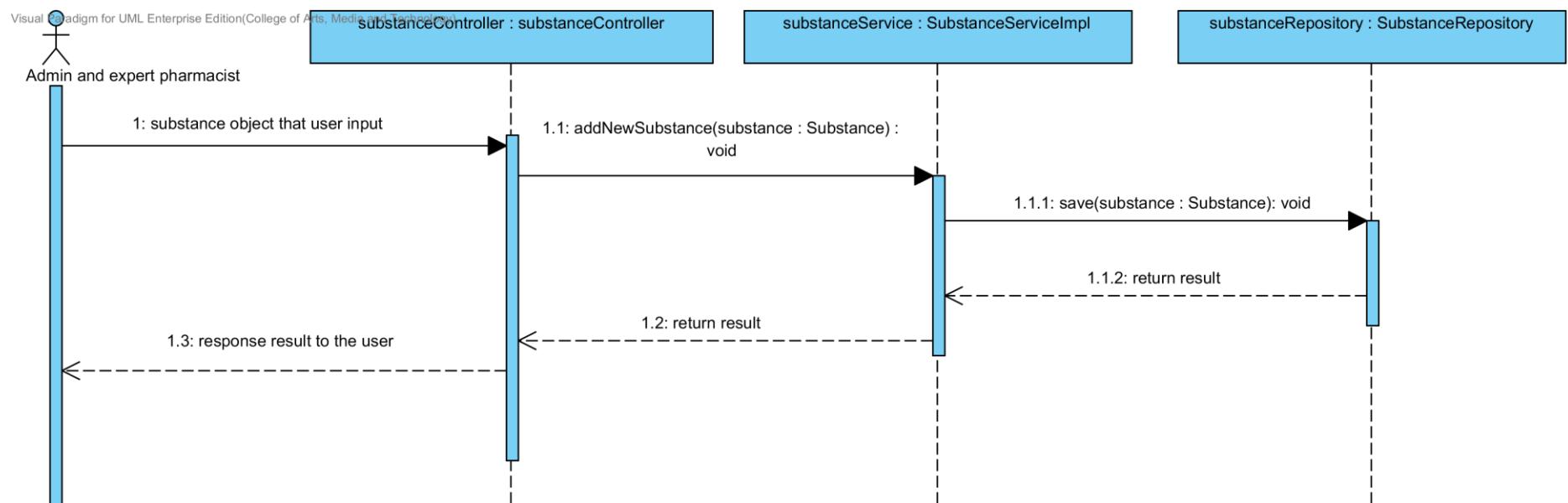


Figure 61: SQD-SV-01: The user adds a new substance to the system (Server Side).

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	61 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

4.1.2. URS-10: The user updates an existing substance in the system.

In a sequence diagram, the user can update an existing substance in the system. Firstly, the user opens the substance updating page, then the user input substance data such as water solubility, stability and etc. The substance controller gets a substance data from the user. After that, the substance controller send a new substance data to appropriate service for updating an existing substance in the system. Finally, the system show substance that already update with the substance updating successful page.

4.1.2.1. SQD-CI-02: The user updates an existing substance in the system (Client Side).

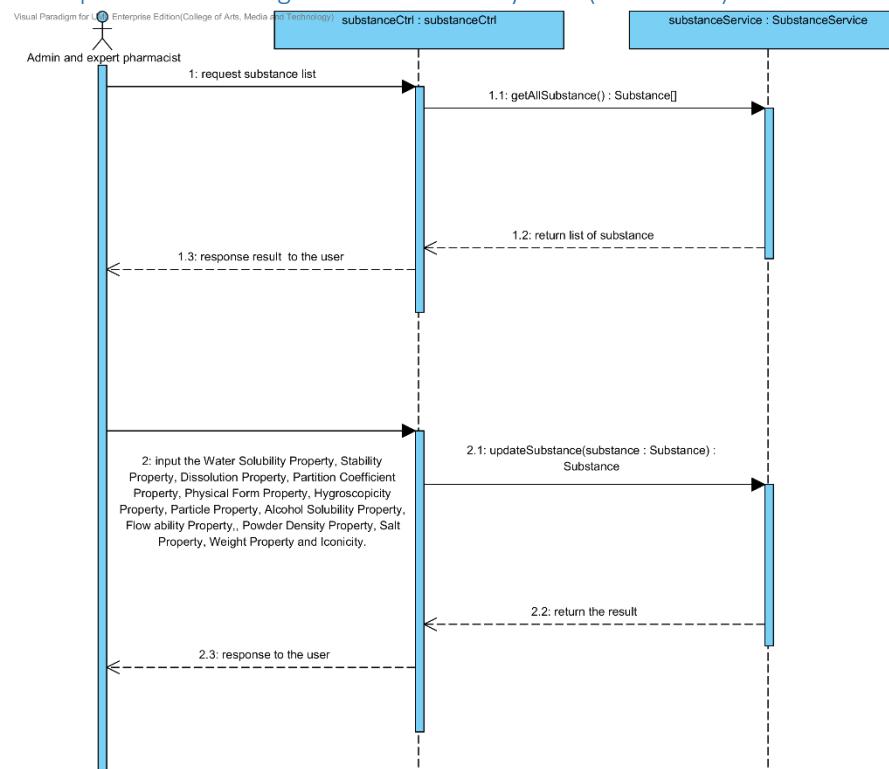


Figure 62: SQD-CI-02 - The user updates an existing substance in the system (Client Side)

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	62 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

4.1.2.2. SQD-SV-02: The user updates an existing substance in the system (Server Side).

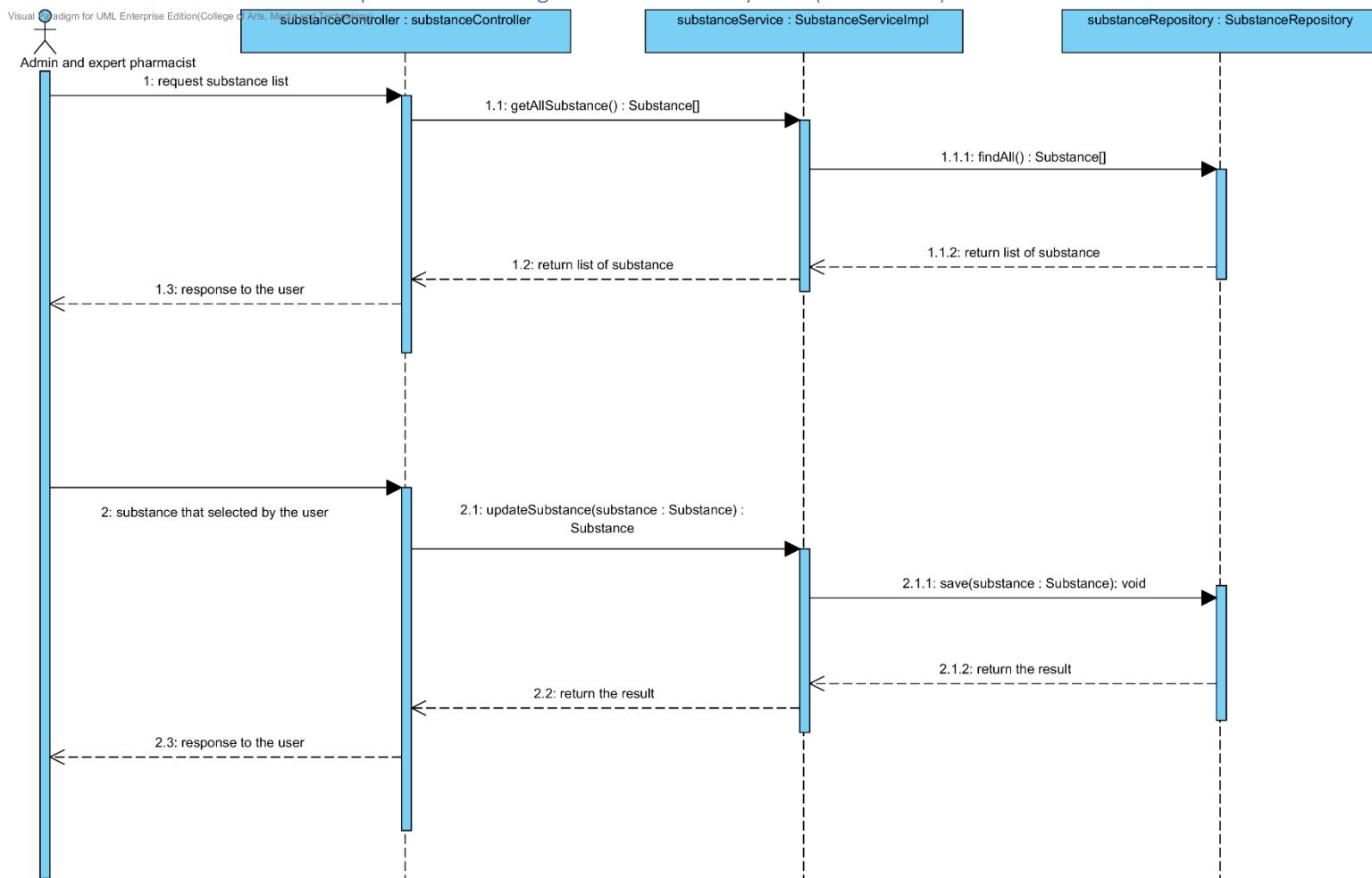


Figure 63: SQD-SV-02- The user updates an existing substance in the system (Server Side)

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	63 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

4.1.3. URS-11: The user deletes an existing substance from the system.

In the sequence diagram, the user can delete an existing substance from the system. Firstly, the user opens the substance deleting page. The system shows all substance data on the screen, then the user selects substance for deleting. After that, the substance controller finds an appropriate service for substance property deleting. Finally, the substance controller shows a substance that already deleted on the deleting substance successful page to the user.

4.1.3.1. SQD-CI-03: The user deletes an existing substance from the system (Client Side).

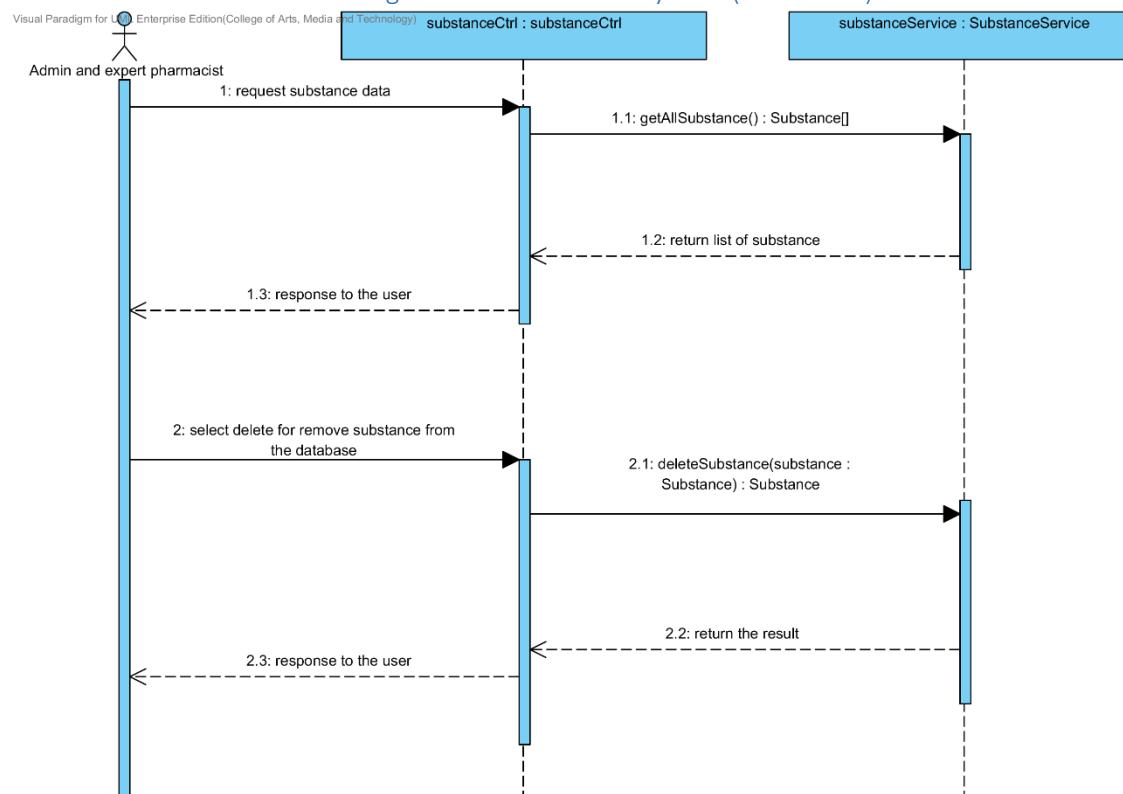


Figure 64: SQD-CI-03- The user deletes an existing substance from the system (Client Side)

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	64 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

4.1.3.2. SQD-SV-03: The user deletes an existing substance from the system (Server Side).

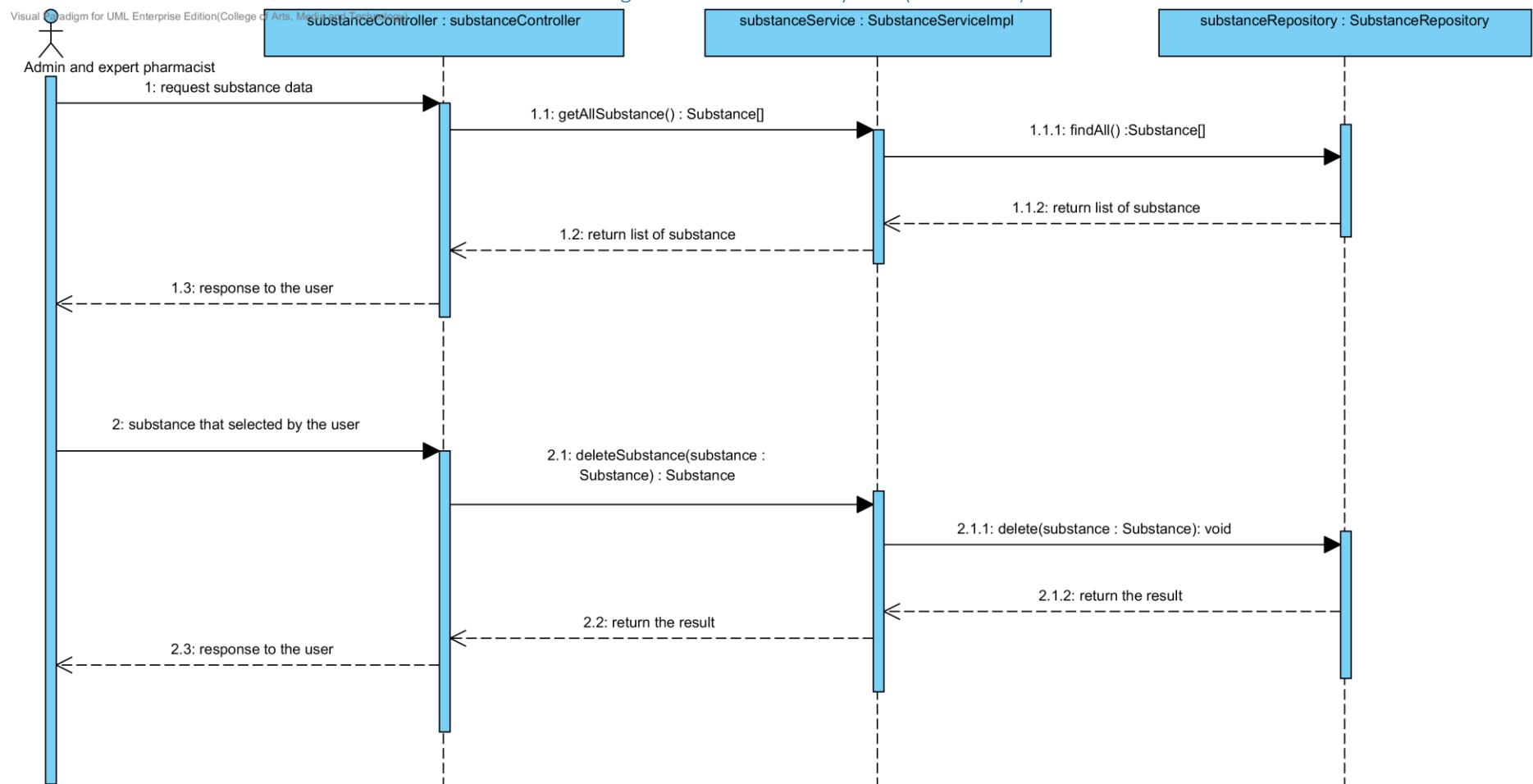


Figure 65: SQD-SV-03 – The user deletes an existing substance from the system (Server Side)

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	65 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

4.1.4. URS-12: The user views the substance in the system.

In the sequence diagram, the user can delete an existing substance from the system. Firstly, the user opens the substance deleting page, then the system shows all substance data on the screen.

4.1.4.1. SQD-CI-04: The user views the substance in the system (Client Side).

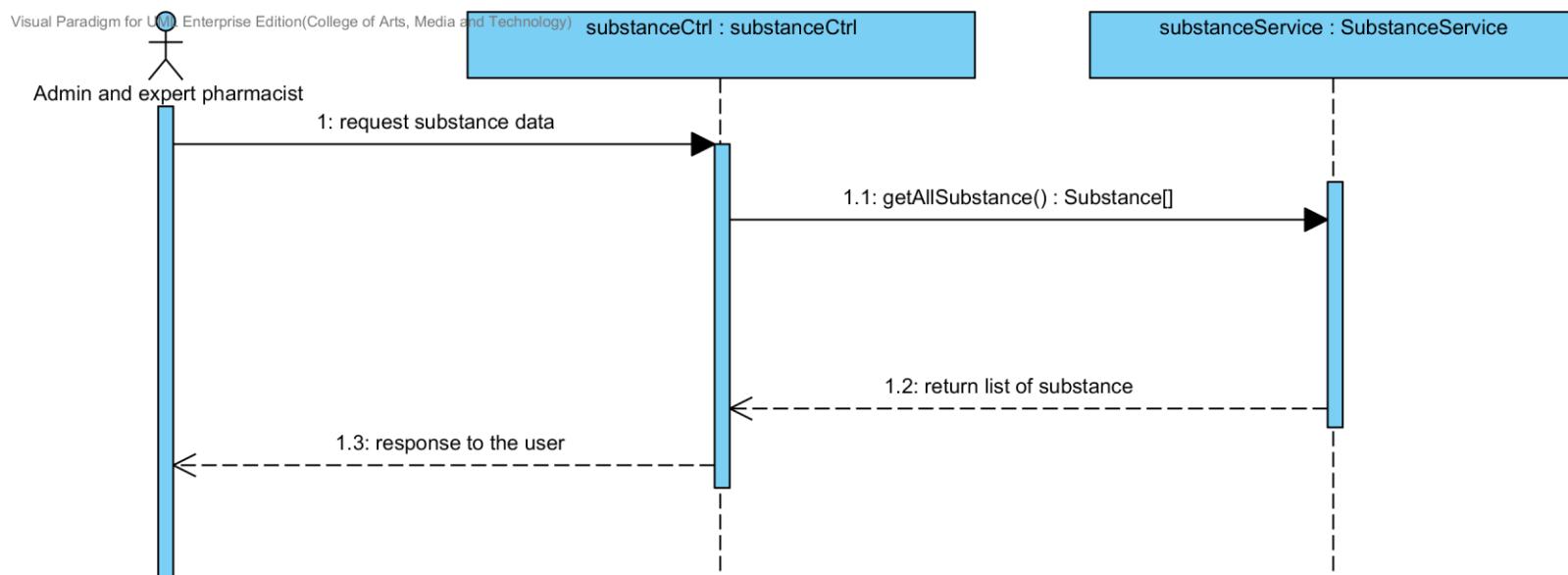


Figure 66 : SQD-CI-04 – The user views the substance in the system (Client Side)

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	66 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

4.1.4.2. SQD-SV-04: The user views the substance in the system (Server Side).

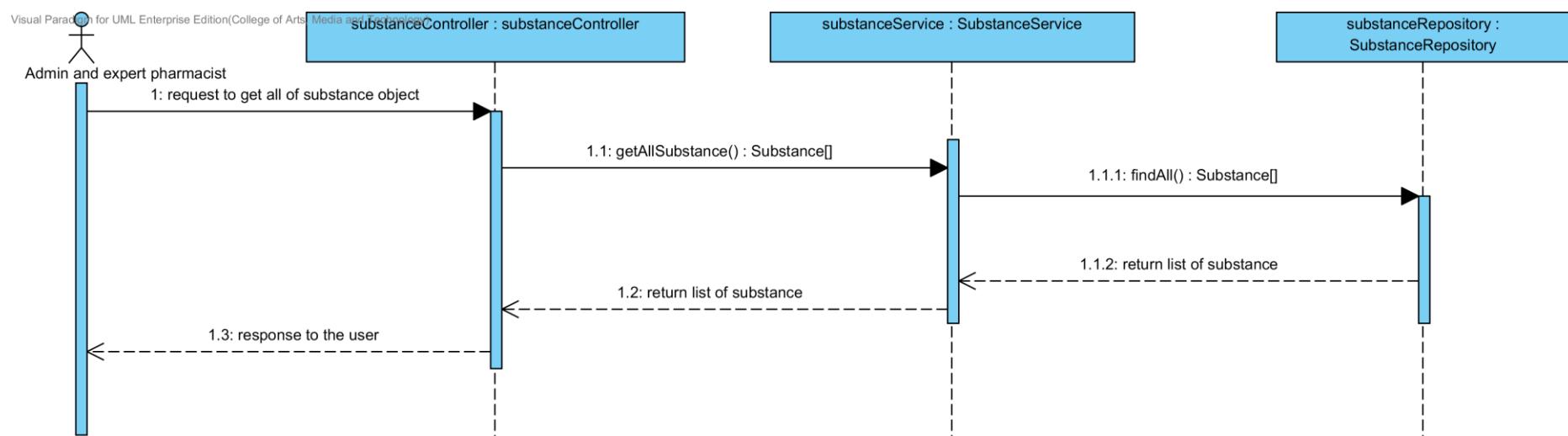


Figure 67: SQD-SV-04 – The user views the substance in the system (Server Side)

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	67 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

4.2. Sub-Feature 7: Manage the drug excipient

4.2.1. URS-13: The user adds a new excipient to the system.

In a sequence diagram, the user can add a new excipient to the system. Firstly, the user opens the excipient adding page, then the user input excipient data such as substance object, substance function, min weight and max weight . The excipient controller gets an input data from the user. After that, the excipient controller send a new excipient data to appropriate service for adding a new excipient. Finally, the system show a new excipient with the adding excipient successful page.

4.2.1.1. SQD-CI-05: The user adds a new excipient to the system (Client Side).

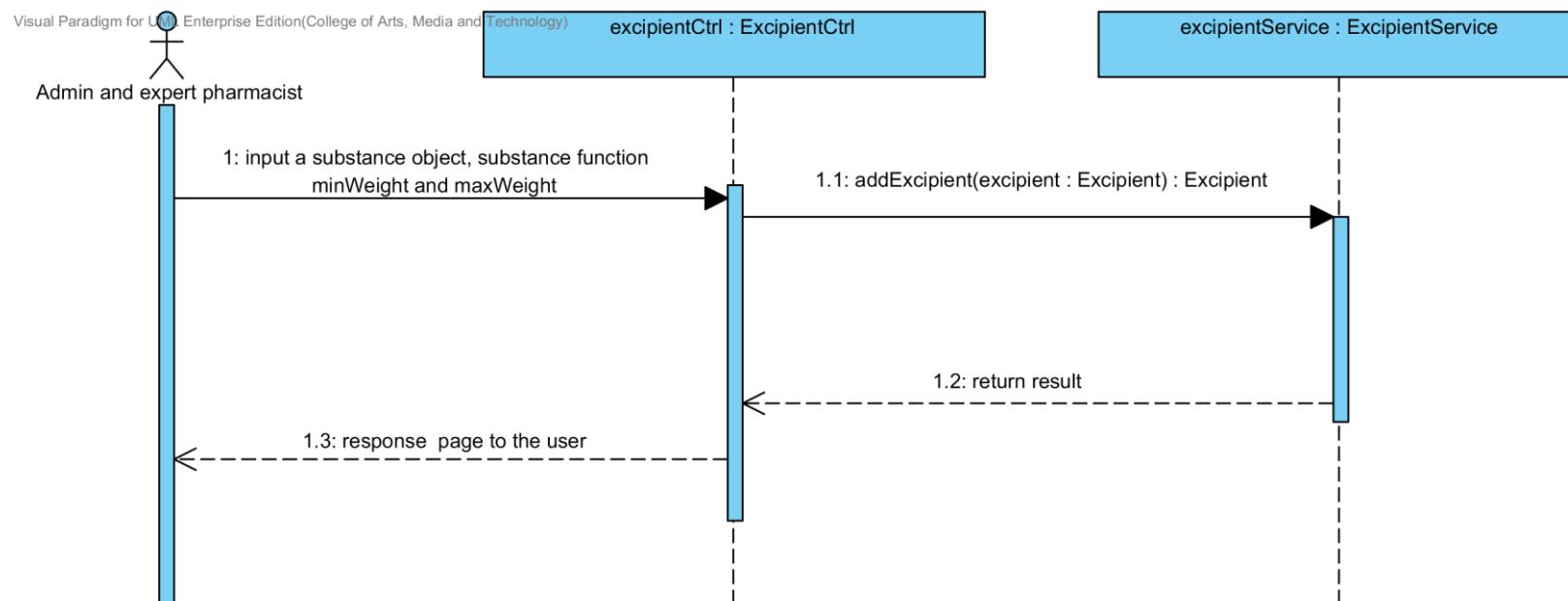


Figure 68: SQD-CI-05 – The user adds a new excipient to the system (Client Side)

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	68 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

4.2.1.2. SQD-SV-05: The user adds a new excipient to the system (Server Side).

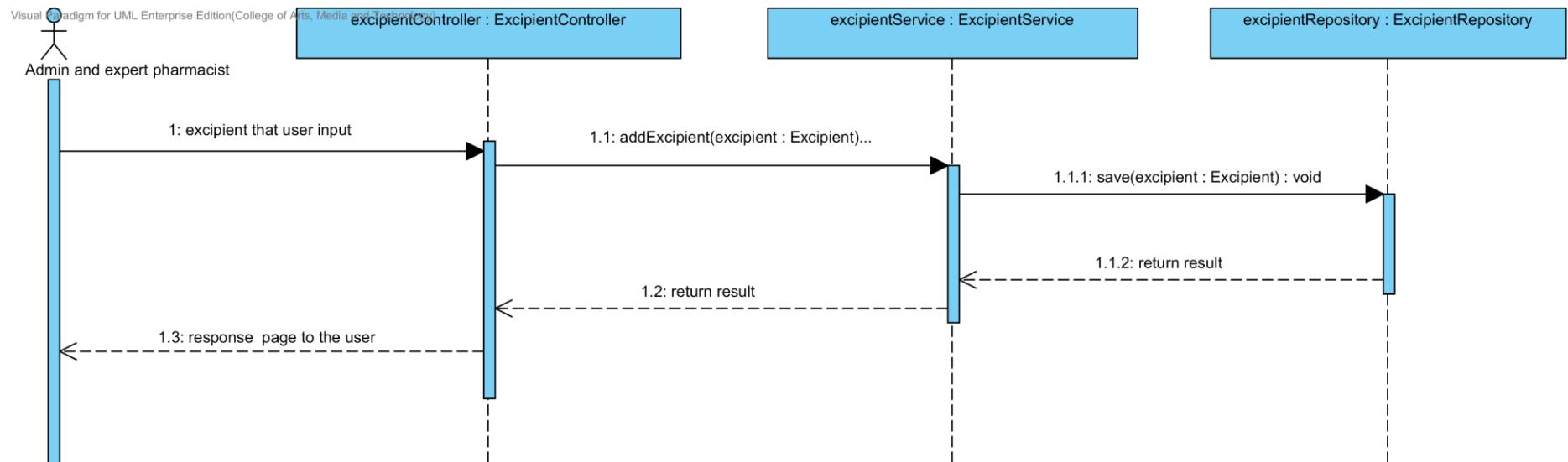


Figure 69: SQD-SV-05 – The user adds a new excipient to the system (Server Side).

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	69 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

4.2.2. URS-14: The user updates an existing excipient in the system.

In a sequence diagram, the user can update an existing excipient in the system. Firstly, the user opens the excipient updating page, then the user input excipient data such as substance function. The excipient controller gets an excipient data from the user. After that, the excipient controller send a new excipient data to appropriate service for updating an existing excipient in the system. Finally, the system show excipient that already update with the excipient adding successful page.

4.2.2.1. SQD-CI-06: The user updates an existing excipient in the system (Client Side)

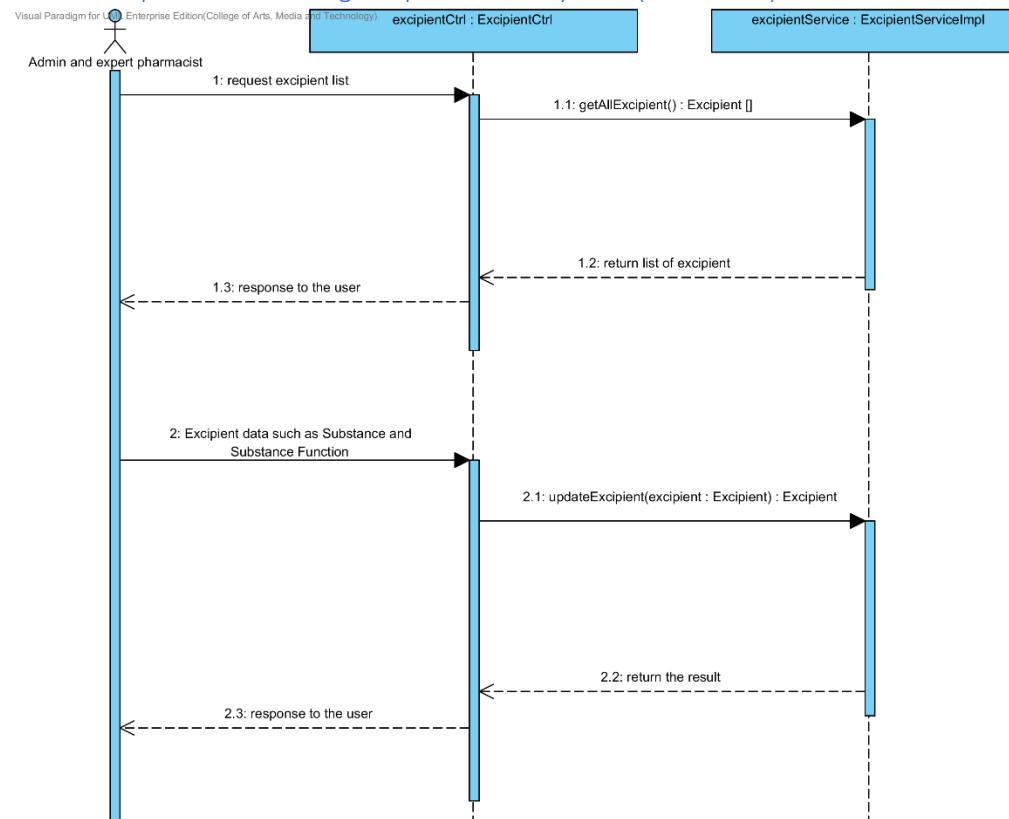


Figure 70 : SQD-CI-06 – The user updates an existing excipient in the system (Client Side).

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	70 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

4.2.2.2. SQD-SV-06: The user updates an existing excipient in the system (Server Side)

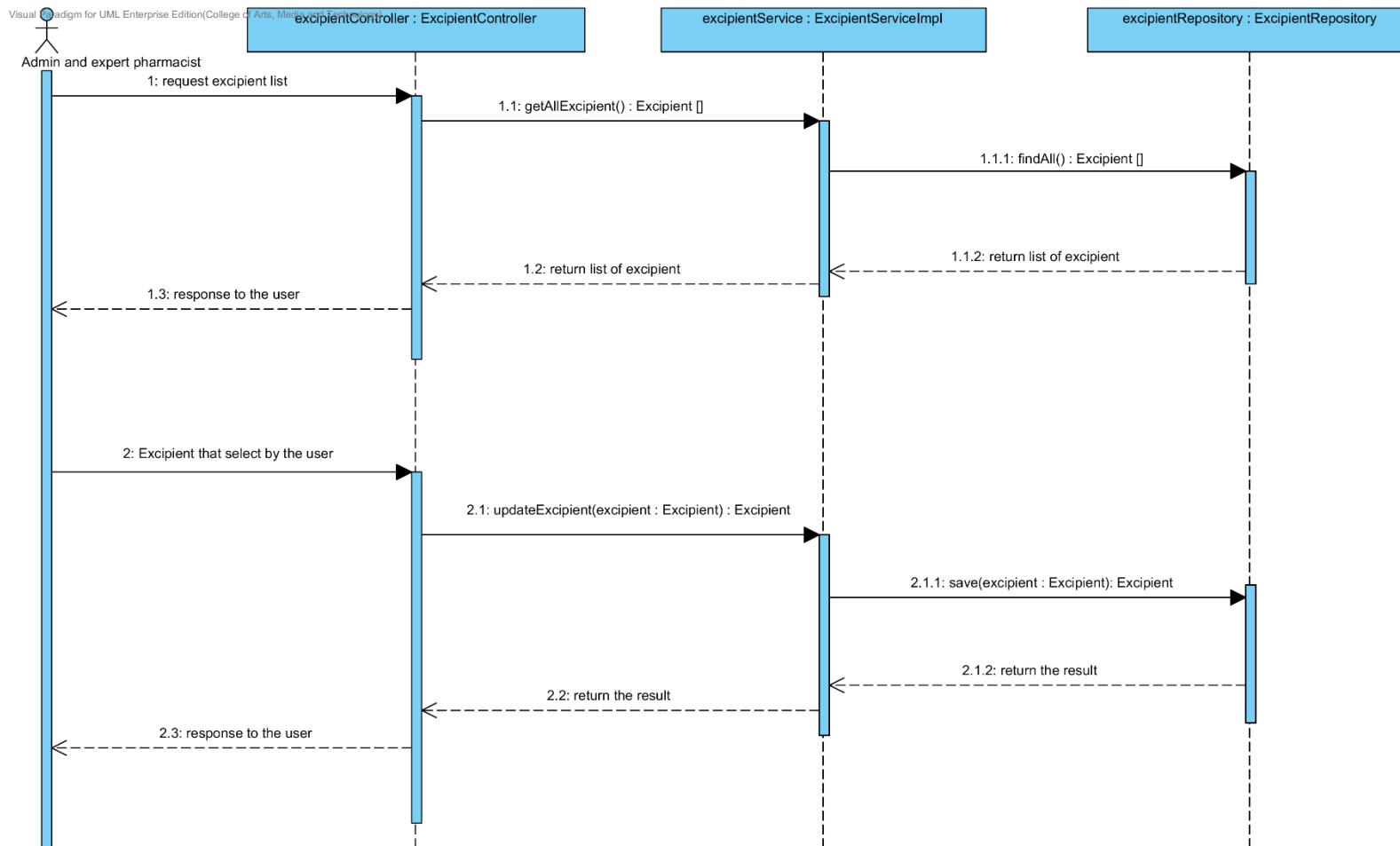


Figure 71: SQD-SV-06 – The user updates an existing excipient in the system (Server Side)

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	71 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

4.2.3. URS-15: The user deletes an existing excipient from the system.

In the sequence diagram, the user can delete an existing excipient from the system. Firstly, the user opens the excipient deleting page. The system shows all excipient data on the screen, then the user selects excipient for deleting. After that, the excipient controller finds an appropriate service for excipient deleting. Finally, the excipient controller shows the excipient that already deleted on the deleting excipient successful page.

4.2.3.1. SQD-SV-07: The user deletes an existing excipient from the system (Client Side).

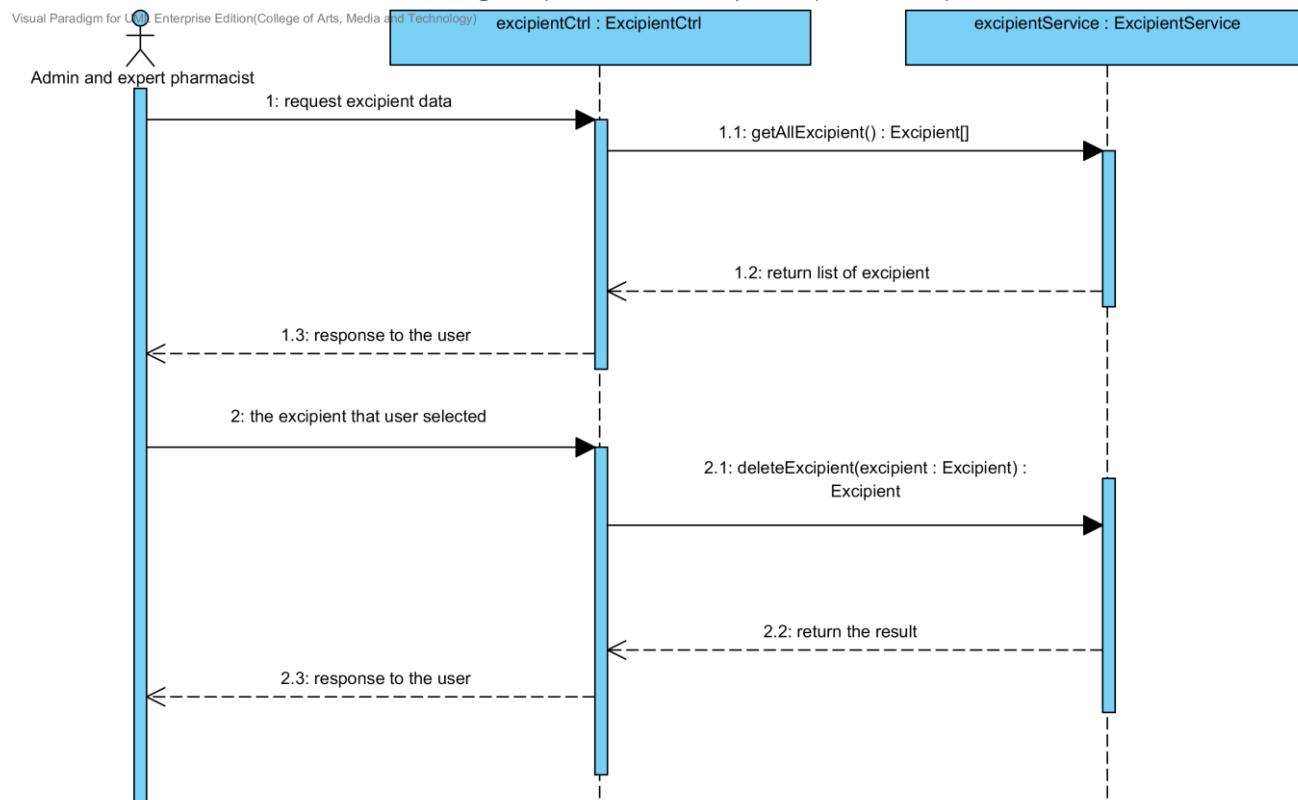


Figure 72 : SQD-SV-07 – The user deletes an existing excipient from the system (Client Side)

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	72 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

4.2.3.2. SQD-SV-07: The user deletes an existing excipient from the system (Server Side).

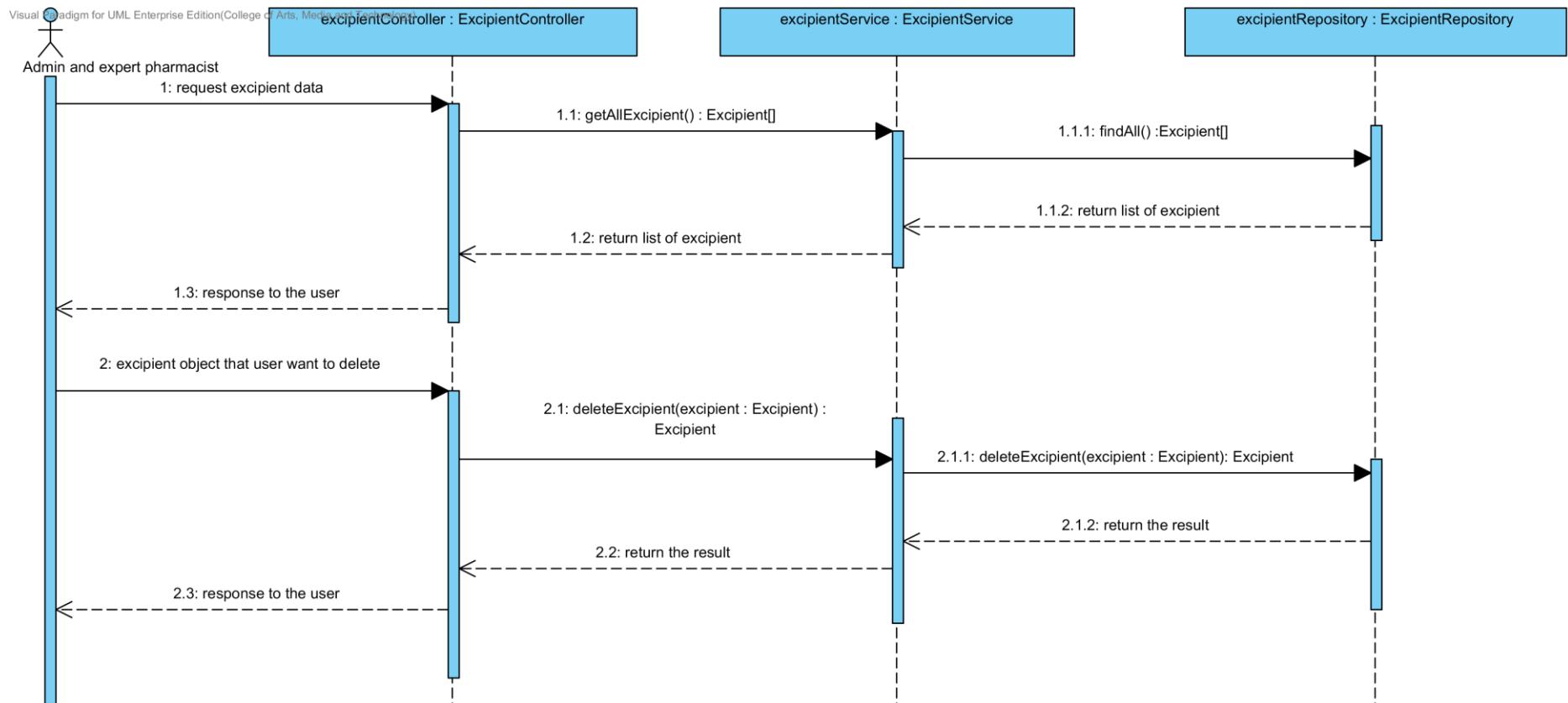


Figure 73 : SQD-SV-07 – The user deletes an existing excipient from the system (Server Side)

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	73 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

4.2.4. URS-16: The user views the excipient in the system.

In the sequence diagram, the user can delete an existing excipient from the system. Firstly, the user opens the excipient deleting page, then the system shows all excipient data on the screen.

4.2.4.1. SQD-CI-08: The user views the excipient in the system (Client Side).

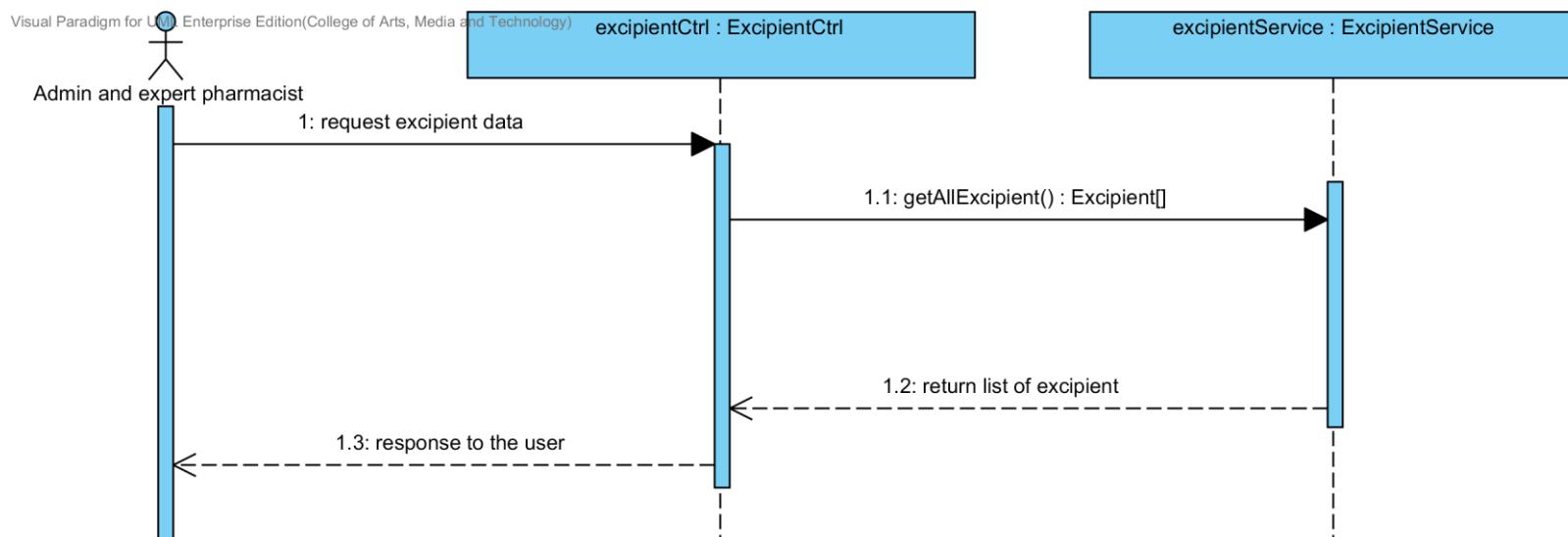


Figure 74 : SQD-CI-08 – The user views the excipient in the system (Client Side)

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	74 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

4.2.4.2. SQD-SV-08: The user views the excipient in the system (Server Side).

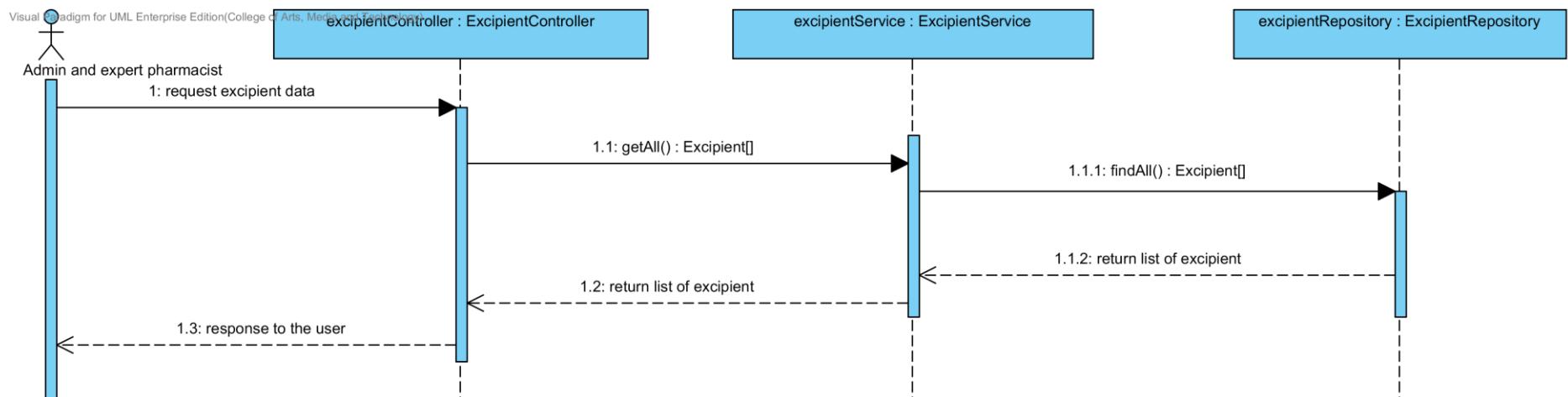


Figure 75 : SQD-SV-08 – The user views the excipient in the system (Server Side)

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	75 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

4.3. Sub-Feature 8: Manage the drug formulation

4.3.1. URS-17: The user adds a new drug's formulation to the system.

In a sequence diagram, the user can add a new drug's formulation to the system. Firstly, the user opens the drug's formulation adding page, then the user input drug's formulation data such as name, excipient and used weight. The drug's formulation controller gets an input data from the user. After that, the drug's formulation controller send a new drug's formulation data to appropriate service for adding a new drug's formulation. Finally, the system show a new drug's formulation with the adding drug's formulation successful page.

4.3.1.1. SQD-CI-09: The user adds a new drug's formulation to the system (Client Side).

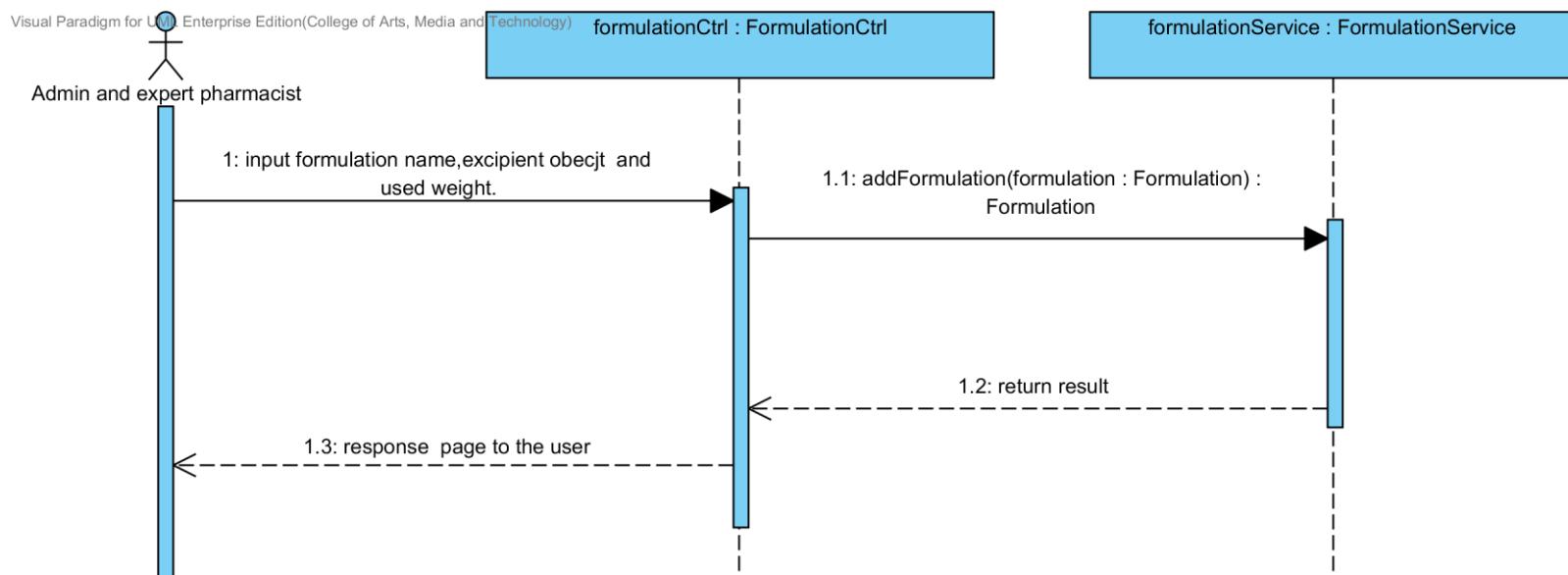


Figure 76: SQD-CI-09: The user adds a new drug's formulation to the system (Client Side).

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	76 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

4.3.1.2. SQD-SV-09: The user adds a new drug's formulation to the system (Server Side).

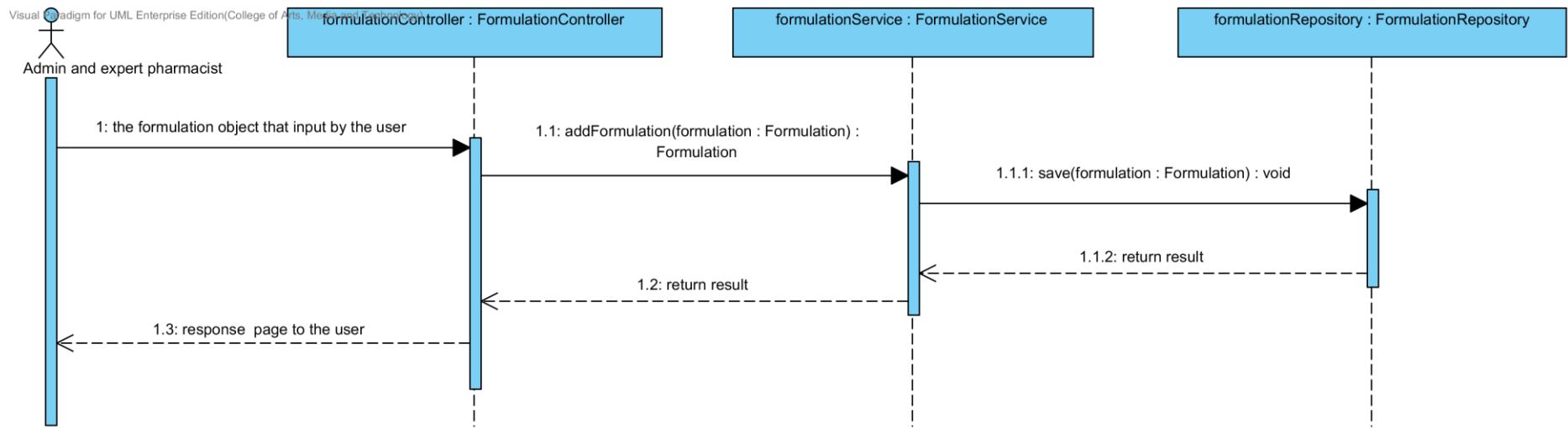


Figure 77: SQD-SV-09 – The user adds a new drug formulation to the system (Server Side).

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	77 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

4.3.2. URS-18: The user updates an existing drug's formulation in the system.

In a sequence diagram, the user can update an existing drug's formulation in the system. Firstly, the user opens the drug's formulation updating page, then the user input drug's formulation data such as excipient and used weight. The drug's formulation controller gets a drug's formulation data from the user. After that, the drug's formulation controller send a new drug's formulation data to appropriate service for updating an existing substance in the system. Finally, the system show drug's formulation that already update with the drug's formulation updating successful page.

4.3.2.1. SQD-CI-10: The user updates an existing drug's formulation in the system (Client Side).

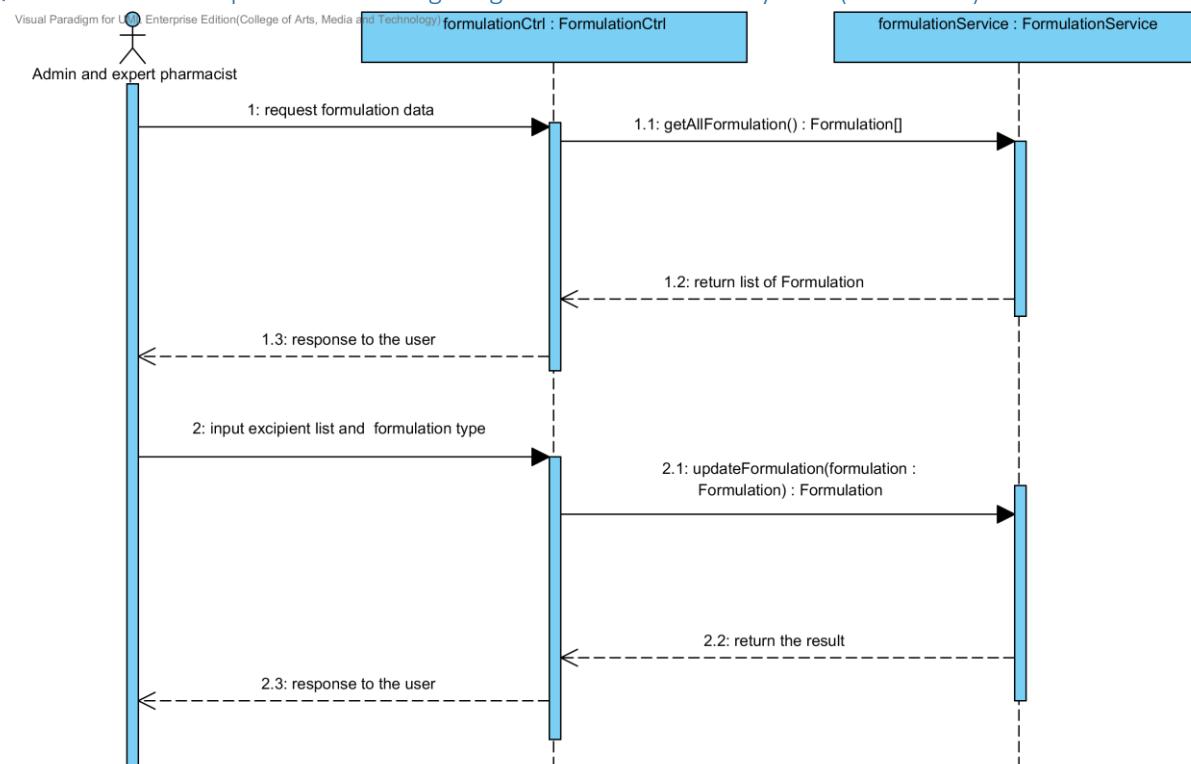


Figure 78: SQD-CI-10 – The user updates an existing drug's formulation in the system (Client Side)

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	78 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

4.3.2.2. SQD-SV-10: The user updates an existing drug's formulation in the system (Server Side).

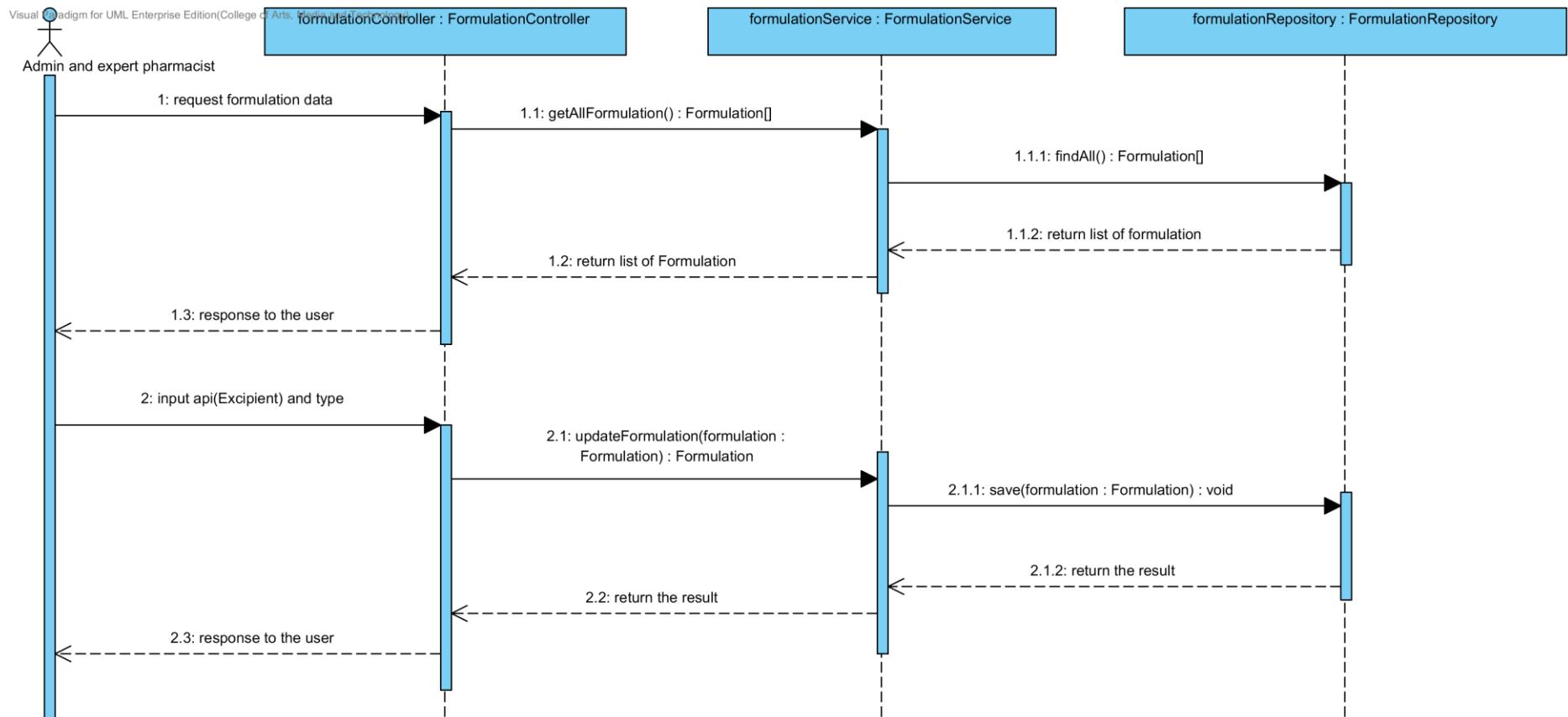


Figure 79: SQD-SV-10: The user updates an existing drug's formulation in the system (Server Side).

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	79 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

4.3.3. URS-19: The user deletes an existing drug's formulation from the system.

In the sequence diagram, the user can delete an existing drug's formulation from the system. Firstly, the user opens the drug's formulation deleting page. The system shows all drug's formulation data on the screen, then the user selects drug's formulation for deleting. After that, the drug's formulation controller finds an appropriate service for drug's formulation deleting. Finally, the drug formulation controller shows a drug's formulation that already deleted on the deleting drug's formulation successful page.

4.3.3.1. SQD-CI-11: The user deletes an existing drug's formulation from the system (Client Side)

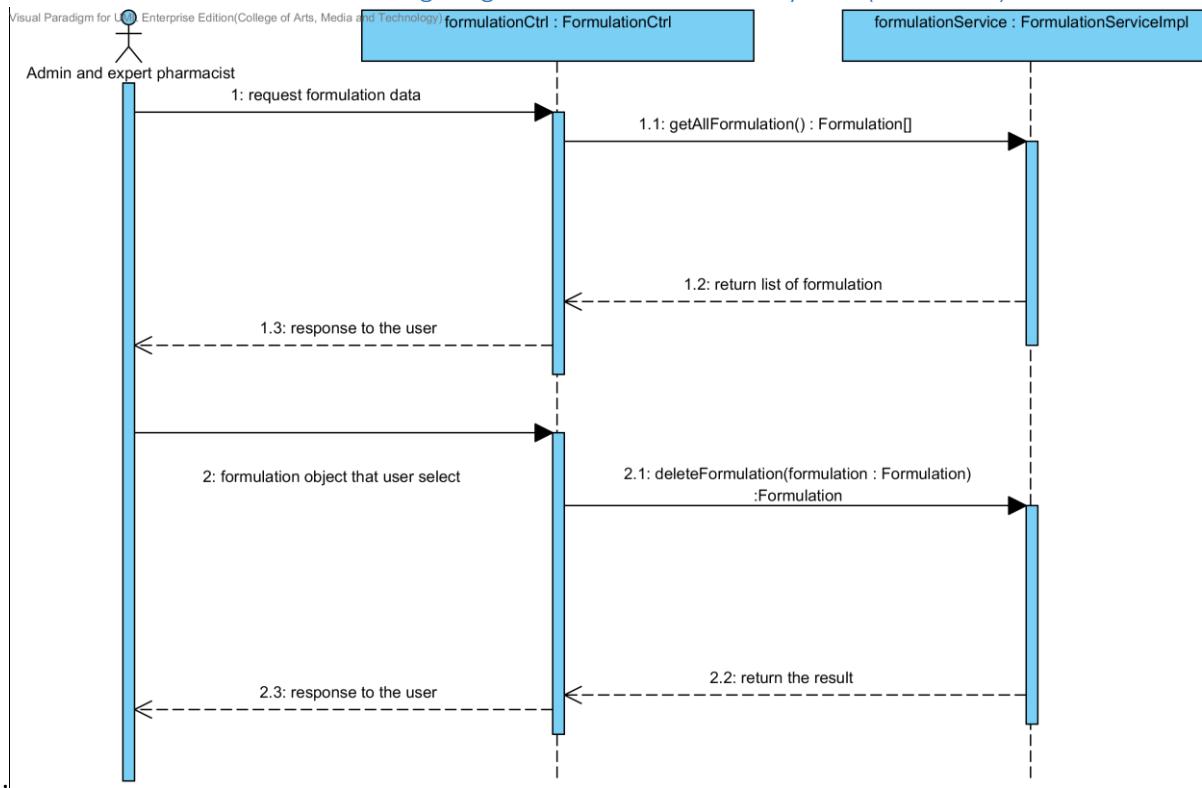


Figure 80: SQD-CI-11: The user deletes an existing drug formulation from the system (Client Side)

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	80 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

4.3.3.2. SQD-SV-11: The user deletes an existing drug's formulation from the system (Server Side).

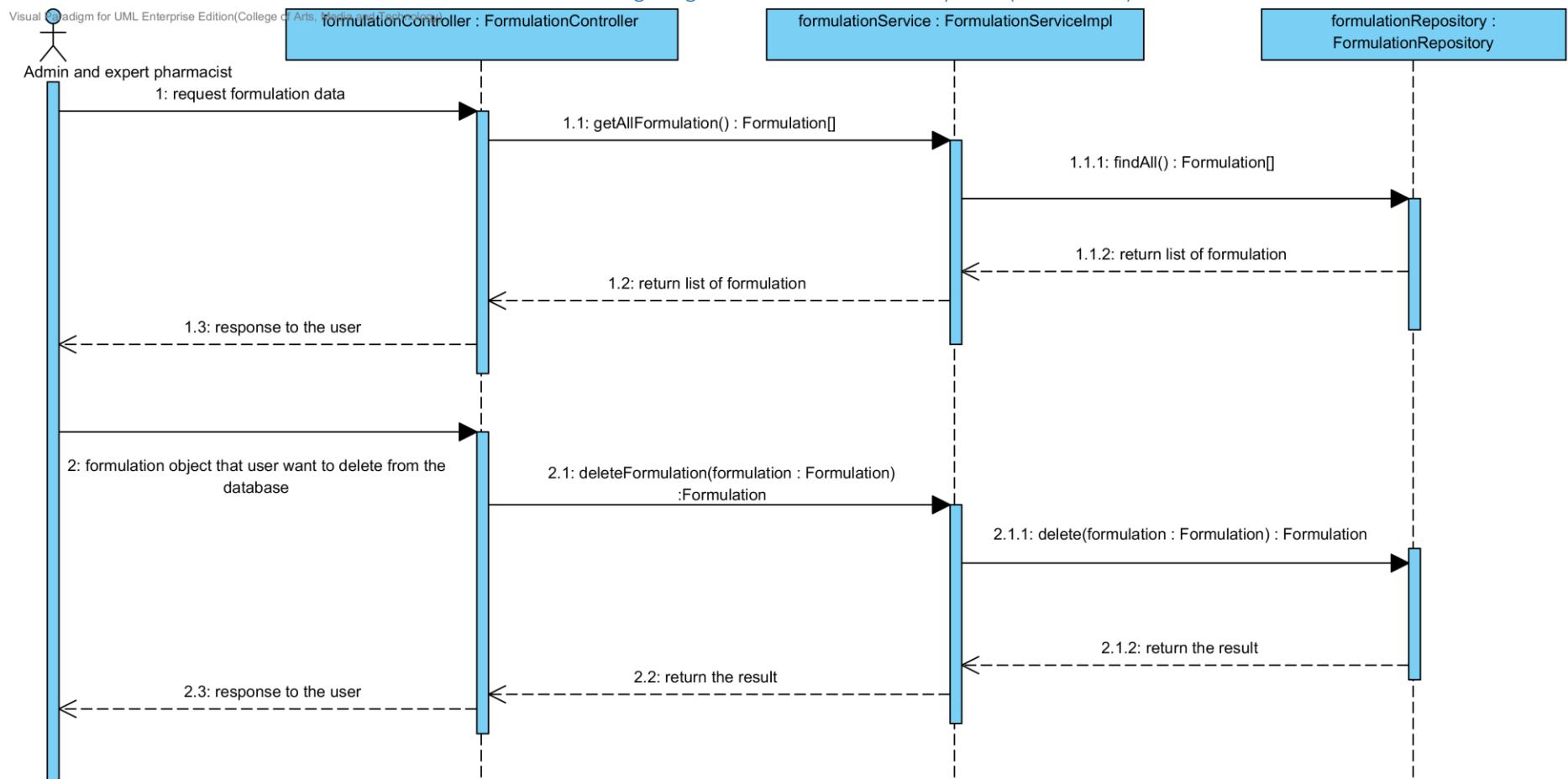


Figure 81: SQD-SV-11: The user deletes an existing drug formulation from the system (Server Side)

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	81 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

4.3.4. URS-20: The user views the drug's formulation in the system.

In the sequence diagram, the user can delete an existing drug's formulation from the system. Firstly, the user opens the drug's formulation deleting page, then the system shows all drug's formulation data on the screen.

4.3.4.1. SQD-CI-12: The user views the drug's formulation in the system (Client Side).

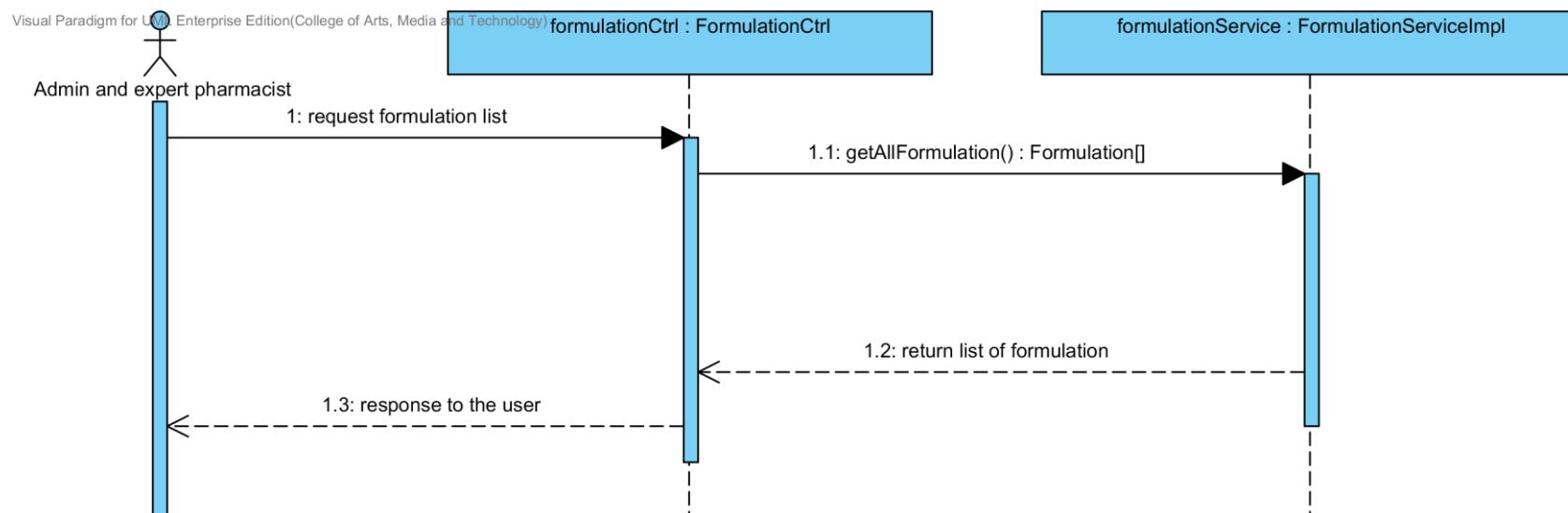


Figure 82: SQD-CI-12: The user views the drug's formulation in the system (Client Side).

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	82 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

4.3.4.2. SQD-SV-12: The user views the drug's formulation in the system (Server Side).

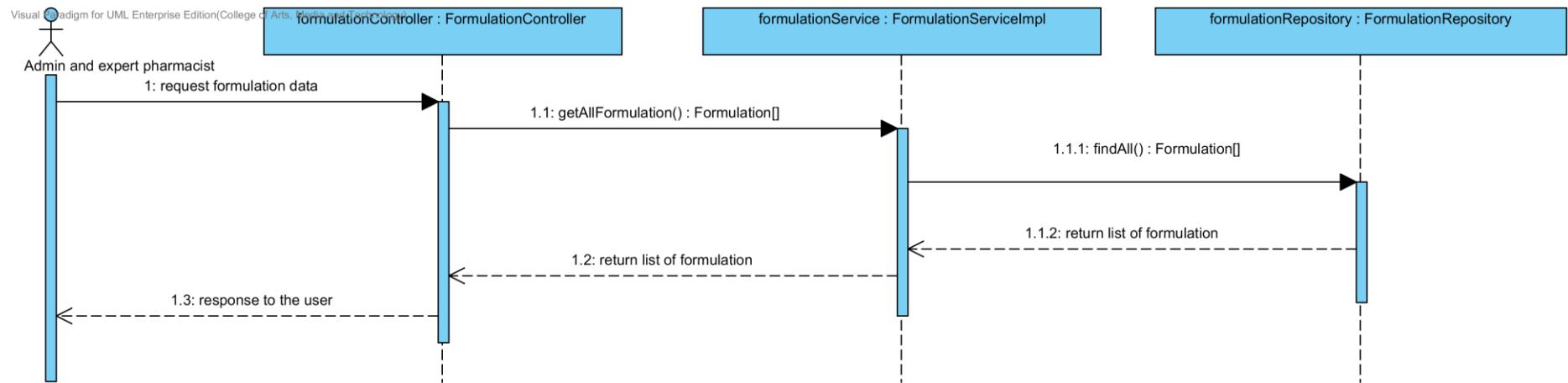


Figure 83: SQD-SV-12 – The user views the drug's formulation in the system (Server Side).

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	83 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

4.4. Sub-Feature 2: Calculate the drug reformulation by using the inference engine.

4.4.1. URS-06: The user calculates a drug reformulation by using an inference engine.

4.4.1.1. SQD-CI-13: The user calculates a drug reformulation by using an inference engine (Client Side).

In a sequence diagram, the user can make drug reformulation by using the inference engine such as Rule Base System, Case Base Reasoning and Hybrid Reasoning. Firstly, the user input the production object that they want to reformulation. Then, the formulationCtrl will get the production data and call the formulation service for sending the production data to server. After that, the server make the reformulation by using the production object. Finally, the server will sends the reformulation result to client side for showing to the user.

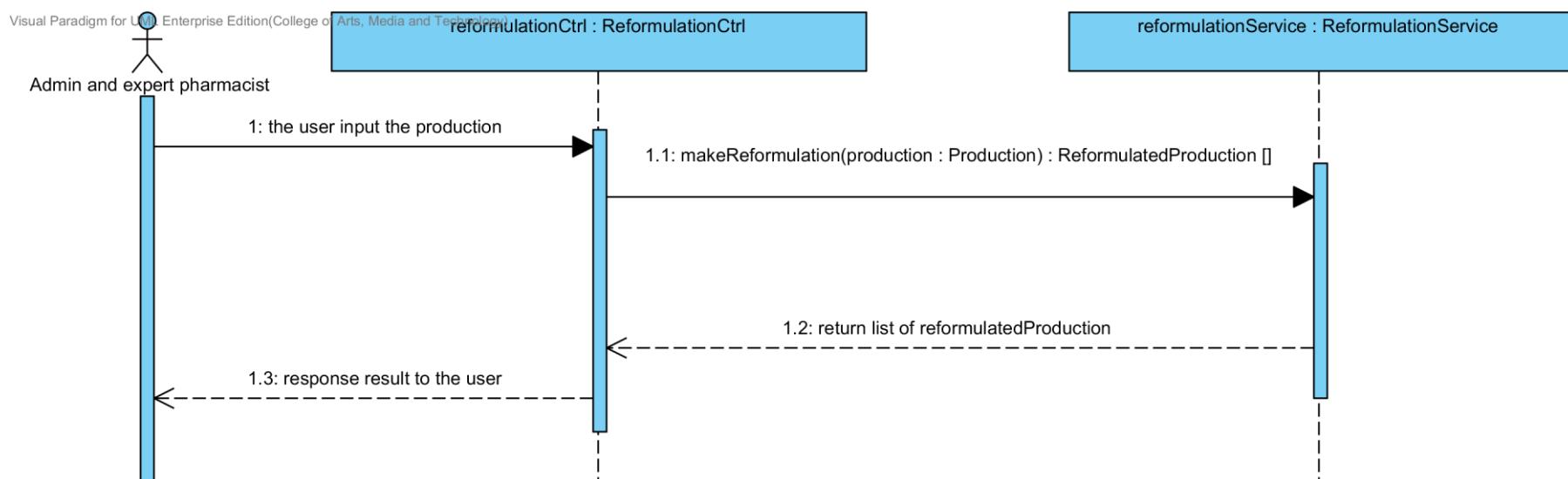


Figure 84: SQD-CI-13 – The user calculates a drug reformulation by using an inference engine (Client Side).

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	84 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

4.4.1.2. SQD-SV-13: The user calculates a drug reformulation by using an inference engine (Server Side).

In a sequence diagram, the server can make drug reformulation by using the inference engine such as Rule Base System, Case Base Reasoning and Hybrid Reasoning. Firstly, the server receives production object from client Side by InferenceEngineController. Then, the InferenceEngineDirector will get the production data from the controller and call the inferenceEngineCreator for making drug reformulation. After that, the server will send the reformulation result to the client for showing to the user.

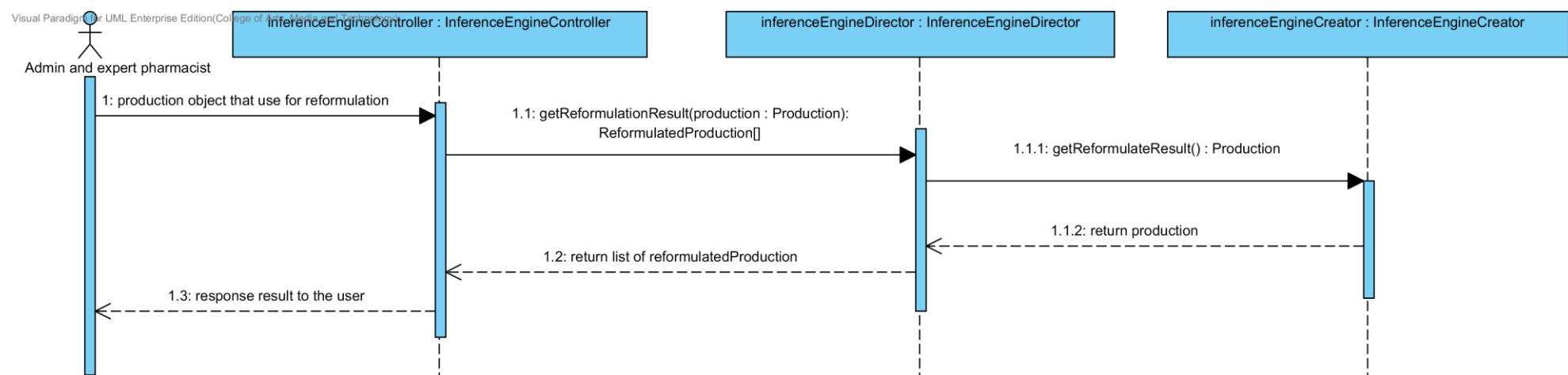


Figure 85: SQD-SV-13 – The user calculates a drug reformulation by using an inference engine (Server Side).

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	85 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

4.5. Sub-Feature 3: View the drug reformulation history

4.5.1. URS-07: The user views their drug reformulation history.

In a sequence diagram, the user can view the reformulation history. Firstly, the user opens the reformulation history page. Then the system will show the reformulation history list to the user. The reformulation history gets from the database.

4.5.1.1. SQD-CI-14: The user views the drug reformulation history (Client Side).

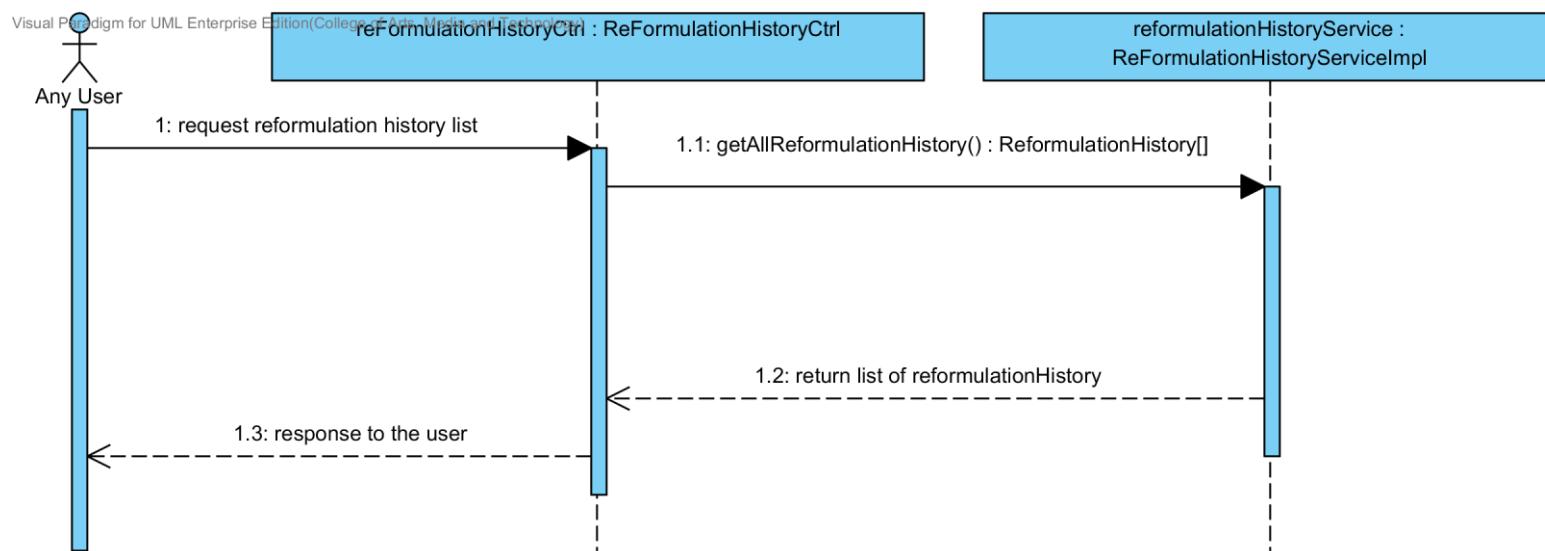


Figure 86: SQD-CI-14 – The user adds a new drug formulation to the system (Client Side).

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	86 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

4.5.1.2. SQD-SV-14: The user views the drug reformulation history (Server Side).

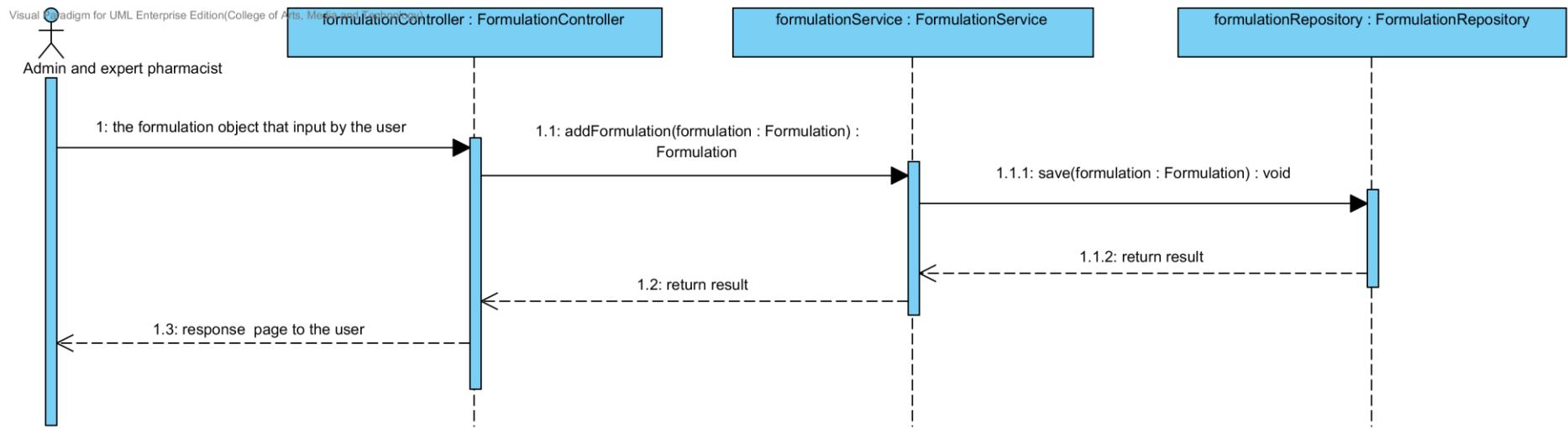


Figure 87: SQD-SV-14 – The user adds a new drug formulation to the system (server side).

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	87 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

Chapter 5 | User Interface

The URS is related with the list of User Interface (UI) that shown on the table 3 below this passage.

No	Sub-Feature Name	URS No.	URS Name	User Interface
6	Manage the drug substance	URS-09	The user adds a new substance into the system.	UI-01,UI-02
		URS-10	The user updates an existing substance into the system.	UI-01,UI-03,UI-04
		URS-11	The user deletes an existing substance from the system.	UI-01,UI-03,UI-05
		URS-12	The user views the substance in the system.	UI-01
7	Manage the drug excipient	URS-13	The user adds a new excipient to the system.	UI-06,UI-07
		URS-14	The user updates an existing drug excipient in the system.	UI-06,UI-08,UI-09
		URS-15	The user delete an existing drug excipient in the system.	UI-06,UI-08,UI-10
		URS-16	The user views all the drug excipient in the system.	UI-06
8	Manage the drug formulation	URS-17	The user adds a new drug formulation case into the system.	UI-11,UI-12
		URS-18	The user updates an existing drug formulation case in the system.	UI-11,UI-13,UI-14
		URS-19	The user deletes an existing drug formulation case in the system.	UI-11,UI-13,UI-15
		URS-20	The user views all of the formulation in the system.	UI-11
2	Calculate the drug reformulation by using the inference engine.	URS-06	The user calculates a drug reformulation by using an inference engine.	UI-16,UI-17,UI-18
3	View the drug reformulation history	URS-07	The user views their drug reformulation history.	UI-19,UI-20

Table 3 : The Relationship of URS and UI

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	88 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

5.1. Sub-Feature 6: Manage the drug substance

5.1.1. URS-09: The user adds a new substance into the system.

In the user interface design, the user can add a new substance by opening the main page of a program, then the user select “Substance” at the main menu on their left hand side. The main content of substance will show the substance data list that contained in the database like the figure 88.

The screenshot shows the OEGP software interface. On the left, there is a vertical sidebar titled 'OEGP Menu' containing the following sections and items:

- Manage Pharmaceutical Value**
 - Substance
 - Excipient
 - Formulation
- Drug Reformulation Analysis**
 - Drug Reformulation
 - Reformulation History

To the right of the sidebar, the main content area is titled 'All Substance'. It features a search bar with a magnifying glass icon and a clear button. Below the search bar, there is a table-like structure displaying substance data:

Substance	Water Solubility	pKa Value
Substance A	Sparingly Soluble	2.43
Substance B	Very Soluble	2.43

A blue '+' button is located in the top right corner of the main content area.

Figure 88 : UI-01 Substance Content

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	89 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

After the main content showed. The user can select “+” symbol on right-top corner for adding a new substance. Then the screen will show substance input form like the figure 89 below this passage.

Excipient	Alcohol Solubility :	
Formulation	Salt Form :	
Drug Reformulation Analysis		
Drug Reformulation	pKa Value :	Required!
	2.34	
Reformulation History	Partition Coefficient Value :	Required!
	6.00	
	Density Value :	Required!
	5.67	
	Molecular Weight Value :	Required!
	8.34	
	Ionicity Type :	Required!
	Very Ionicity	
	Select the Stability >	
Create Substance		

Figure 89: UI-02 Substance Adding Form

Finally, the user inputs the substance follow the substance adding form. If the user agree with the substance data that they input. They can select “Create Substance” for adding a new substance to the database. Then the system will get the substance object that user input and save it to the data base.

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	90 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

5.1.2. URS-10: The user updates an existing substance into the system.

In the user interface design (UI), the user select the “substance” on the main menu at their left hand side. The system will show the main content with the substance list (look on the figure 88). If the user wants to edit the substance. The user should select one of substance that appear on the substance list. After that, the system will shows the detail of each substance like the figure 90

OEGP Menu		All Substance		Substance Detail		...
Manage Pharmaceutical Value						
Substance		Substance Name :		Substance A		
Excipient		Water Solubility Type :		SparinglySoluble		
Formulation		Stability :		["FFFF"]		
Drug Reformulation Analysis		Pka Value :		2.43		
Drug Reformulation		Partition Coefficient Value :		5.43		
Reformulation History		Solid State :		AmorphousMaterials		
		Hygroscopicity :		Slightly hygroscopic		

Figure 90 : UI-03 Substance Detail

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	91 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

After the substance detail shows on the screen. The user should selects “...” symbol for updating or deleting a substance. The system will show the substance option. The option consist of 2 choice the first choice is “Edit Substance” and second is “delete substance”. Next, the user select “Edit Substance” for updating a substance. Then, the system will show the substance editing form like figure XX below this passage.

Excipient	AICONOL Solubility :
	Soluble
Formulation	Salt Form :
	saltFormC
Drug Reformulation Analysis	pKa Value :
	2.43
Reformulation History	Partition Coefficient Value :
	5.43
	Density Value :
	5.43
	Molecular Weight Value :
	5.5
	ionicity Type :
	DDD
	Select the Stability >
Update Substance	

Figure 91: UI-04 - The substance editing form

After the substance editing form showed on the screen. The can edit substance data such as the water solubility, stability, pKa value and etc. If the user agree with their new substance data. The must select the “Update Substance” for updating the substance data. On the on other hand, they can select “Cancel” on top-left corner for backing to the substance detail page.

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	92 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

5.1.3. URS-11: The user deletes an existing substance from the system.

In the user interface design (UI), the user select the “substance” on the main menu at their left hand side. The system will show the main content with the substance list (look on the figure 88). If the user wants to delete the substance. The user should select one of substance that appear on the substance list. After that, the system will shows the detail of each substance (look on the figure 90). After the substance detail shows on the screen. The user should selects “...” symbol for updating or deleting a substance. The system will show the substance option. The option consist of 2 choice, the first choice is “Edit Substance” and second is “delete substance”. Next, the user select “Delete Substance” for deleting a substance. Then, the system will show the YES/No dialog for confirming substance deleting. The screen will show the YES/NO like figure 92 below this passage.

The screenshot shows a software interface for managing pharmaceutical values. On the left, there's a sidebar with navigation options: 'Manage Pharmaceutical Value' (selected), 'Substance' (selected), 'Excipient', 'Formulation', 'Drug Reformulation Analysis' (selected), 'Drug Reformulation', and 'Reformulation History'. The main content area displays substance details for 'Substance A': Water Solubility Type (SparinglySoluble), Stability (["FFFF"]), Pka Value (2.43), Partition Coe (5.43), Solid State (AmorphousMaterials), Hygroscopicity (Slightly hygroscopic), Particle Size (NanosizeParticle), and Alcohol Solubility. A modal dialog box is overlaid on the screen, asking 'Are you sure you want to delete this substance?' with 'Delete Substance' and 'Cancel' buttons.

Figure 92: UI-05 Substance deleting form

After that, the user can select “Ok” for deleting substance from the system. On the other hand, they can select “Cancel” for canceling the substance deleting.

5.1.4. URS-12: The user views the substance in the system

The user interface of the URS-12 is same as the figure 88 (UI-01: The Substance Main Content).

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	93 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

5.2. Sub-Feature 7: Manage the drug excipient

5.2.1. URS-13: The user adds a new excipient to the system.

In the user interface design, the user can add a new excipient by select “Excipient” at the menu on their left hand side. After that, the system will show main content of the excipient with the excipient list like figure 93 below this passage.

The screenshot shows the OEGP software interface. On the left, there is a vertical navigation menu with the following sections:

- OEGP Menu
- Manage Pharmaceutical Value
 - Substance
 - Excipient
- Drug Reformulation Analysis
 - Drug Reformulation
 - Reformulation History

The main content area is titled "All Excipient". It features a search bar with a magnifying glass icon and a clear button. Below the search bar, there is a table-like structure displaying three rows of excipient data:

Excipient Name :
Substance A
Substance B
Substance B

A blue circular button with a white plus sign is located in the top right corner of the main content area.

Figure 93: UI-06 Excipient Main Content

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	94 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

After the system shows the excipient main content with the excipient list. The user can select “+” at the top-right corner for going to select substance. This substance will be used for creating a new excipient. After the user select substance that they want. The system will show the excipient adding form like figure 94.

The screenshot shows a software interface titled "OEGP Menu". On the left, there is a vertical navigation menu with several options: "Manage Pharmaceutical Value" (selected), "Substance", "Excipient", "Formulation", "Drug Reformulation Analysis" (selected), "Drug Reformulation", and "Reformulation History". The main content area is titled "Substance Selection" and "Create a new Excipient". It contains two input fields: "Excipient Name : Substance A" and "Select Substance Function". At the bottom right of this section is a large blue button labeled "Create Excipient".

Figure 94: UI-07 Excipient adding form

From the Figure 94, the user can select “Substance Function” for adding a new substance function list to the excipient. If the user accept the substance data and substance function data. They can select “create Excipient” for creating a new excipient to the database. On the other hand, the user can select “cancel” for cancel excipient adding and back to previous page.

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	95 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

5.2.2. URS-14: The user updates an existing excipient in the system.

In the user interface design, the user can edited the existing excipient by select “Excipient” at the menu on their left hand side. After that, the system will show main content of the excipient with the excipient list (look on figure 93). If the user wants to update the excipient. They must select one of the excipient that show on the excipient list. The system will show the detail of excipient data follow item that user selected. The detail of excipient look like figure 95 below this passage.

The screenshot displays the OEGP software's user interface. On the left, a vertical sidebar titled "OEGP Menu" contains several options under "Manage Pharmaceutical Value": "Substance", "Excipient" (which is currently selected), and "Formulation". Below these are sections for "Drug Reformulation Analysis" with "Drug Reformulation" and "Reformulation History". The main content area is titled "All Excipients" and "Excipient Detail". It shows detailed information for an excipient named "Substance A", including its substance function: "Function Name : binder , Function Type : binderC , Min Weight : 2.43 , Max Weight : 4.32" and "Function Name : disintegrant , Function Type : disintegrantB , Min Weight : 2.43 , Max Weight : 4.43". There is also a three-dot menu icon in the top right corner of the main content area.

Figure 95: UI-08 Excipient Detail

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	96 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

After the excipient detail shows on the screen. The user should selects “...” symbol for updating or deleting an excipient. The system will show the excipient option. The option consist of 2 choice the first choice is “Edit Excipient” and second is “Delete Excipient”. Next, the user select “Edit Excipient” for updating an excipient. Then, the system will show the excipient editing form like the figure 96 on the screen.

The screenshot shows a software interface titled 'OEGP Menu'. On the left is a sidebar with several options under 'Manage Pharmaceutical Value': 'Substance' (selected), 'Excipient' (disabled), and 'Formulation'. Below this is a section for 'Drug Reformulation Analysis' with 'Drug Reformulation' (selected) and 'Reformulation History'. The main area is titled 'Update Excipient' and contains two input fields: 'Excipient Name : Substance A' and 'Select Substance Function'. At the bottom right of this area is a large red button labeled 'Update Excipient'.

Figure 96: UI-09 - The excipient updating form

After the excipient editing form showed on the screen. The can edit excipient data such as the substance function. If the user agree with their new excipient data. The must select the “Update Excipient” for updating the excipient data. On the on other hand, they can select “Cancel” on top-left corner for backing to the excipient detail page. (Look on figure 95).

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	97 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

5.2.3. URS-15: The user deletes an existing excipient from the system.

In the user interface design (UI), the user select the “excipient” on the main menu at their left hand side. The system will show the main content with the excipient list (look on the figure 93). If the user wants to delete the excipient. The user should select one of excipient that appear on the excipient list. After that, the system will shows the detail of each excipient (look on the figure 95). After the excipient detail shows on the screen. The user should selects “...” symbol for updating or deleting an excipient. The system will show the excipient option. The option consist of 2 choice, the first choice is “Edit Excipient” and second is “Delete Excipient”. Next, the user select “Delete Excipient” for deleting an excipient. Then, the system will show the YES/No dialog for confirming excipient deleting. The screen will show the YES/NO like figure 97 below this passage.

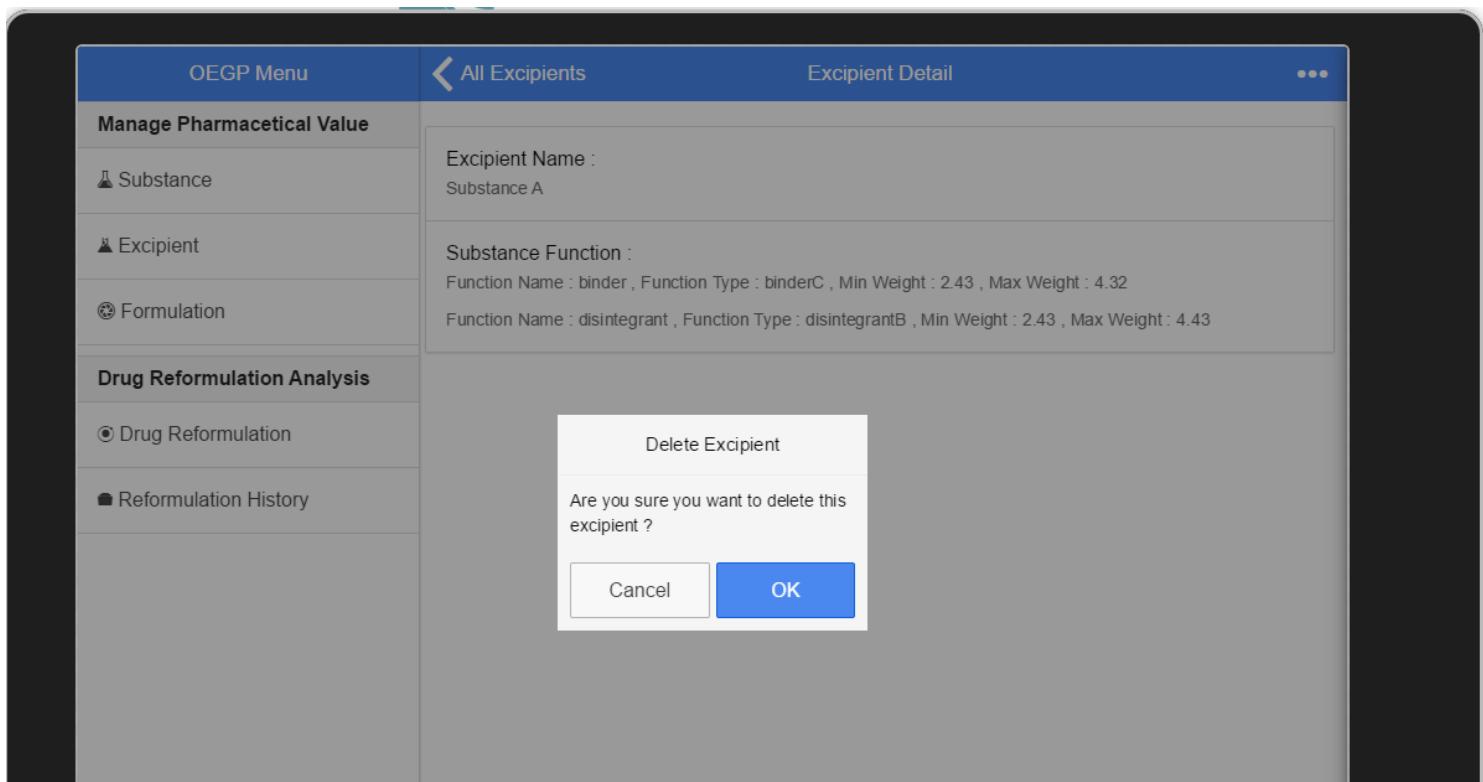


Figure 97: UI-10 The Excipient deleting form

After that, the user can select “Ok” for deleting excipient from the system. On the other hand, they can select “Cancel” for canceling the excipient deleting.

5.2.4. URS-16: The user views the excipient in the system

The user interface of the URS-16 is same as the figure 93 (UI-06: The Excipient Main Content).

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	98 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

5.3. Sub-Feature 8: Manage the drug formulation

5.3.1. URS-17: The user adds a new formulation to the system.

In the user interface design, the user can add a new formulation by select “Formulation” at the menu on their left hand side. After that, the system will show main content of the formulation with the formulation list like figure 98 below this passage.

The screenshot shows the OEGP software interface. On the left, there is a vertical sidebar titled "OEGP Menu" with the following options:

- Manage Pharmaceutical Value
 - Substance
 - Excipient
 - Formulation
- Drug Reformulation Analysis
 - Drug Reformulation
 - Reformulation History

The "Formulation" option under "Manage Pharmaceutical Value" is currently selected, indicated by a blue background. The main content area is titled "All Formulation". It contains a search bar with the placeholder "Search" and a clear button (X). Below the search bar, there is a list of formulations:

- Formulation Name : Formulation A
- Formulation Name : Formulation B
- Formulation Name : Atorvastatin Sodium

A blue "+" button is located in the top right corner of the main content area.

Figure 98: UI-11 - The Formulation Main Content

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	99 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

After the system shows the formulation main content with the formulation list. The user can select “+” at the top-right corner for going to formulation adding form. This formulation adding form consist of formulation name, formulation type and Excipient. The system will show the formulation adding form like figure 99.

Figure 99: UI-12 - Formulation adding form

From the Figure 99, the user can select “Excipient” for adding a new excipient list to the formulation. If the user accept the formulation name, formulation type and excipient. They can select “create Formulation” for creating a new formulation to the database. On the other hand, the user can select “cancel” for cancel formulation adding and back to previous page.

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	100 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

5.3.2. URS-18: The user updates a formulation in the system.

In the user interface design, the user can edit the existing formulation by select “Formulation” at the menu on their left hand side. After that, the system will show main content of the formulation with the formulation list (look on figure 98). If the user wants to update the formulation. They must select one of the formulation that show on the formulation list. The system will show the detail of formulation data follow item that user selected. The detail of formulation look like figure XX below this passage.

Figure 100: UI-13 the Formulation Detail

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	101 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

After the formulation detail shows on the screen. The user should selects “...” symbol for updating or deleting the formulation. The system will show the formulation option. The option consist of 2 choice, the first choice is “Edit Formulation” and second is “Delete Formulation”. Next, the user select “Edit Formulation” for updating the formulation. Then, the system will show the formulation editing form like the figure XX on the screen.

The screenshot shows a software interface for managing pharmaceutical values. On the left, there's a vertical menu with sections like 'Manage Pharmaceutical Value', 'Substance', 'Excipient', 'Formulation', 'Drug Reformulation Analysis', 'Drug Reformulation', and 'Reformulation History'. The 'Formulation' section is currently selected. To the right of the menu, there's a main content area titled 'Update Formulation'. It contains fields for 'Formulation Name' (set to 'Formulation A'), 'Select the formulation type' (set to 'Solution Formulation'), and 'Select the excipient'. At the bottom right of this area is a large blue button labeled 'Update Formulation'. In the top-left corner of the main content area, there's a smaller blue button labeled 'Cancel'.

Figure 101 : UI-14- Formulation editing form

After the formulation editing form showed on the screen. The can edit formulation data such as the formulation type and excipient. If the user agree with their new formulation data. They must select the “Update Formulation” for updating the formulation data. On the other hand, they can select “Cancel” on top-left corner for backing to the formulation detail page. (Look on figure 100).

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	102 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

5.3.3. URS-19: The user deletes an existing formulation from the system.

In the user interface design (UI), the user select the “formulation” on the main menu at their left hand side. The system will show the main content with the formulation list (look on the figure 98). If the user wants to delete the formulation. The user should select one of formulation that appear on the formulation list. After that, the system will shows the detail of each formulation (look on the figure 100). After the formulation detail shows on the screen. The user should selects “...” symbol for updating or deleting the formulation. The system will show the formulation option. The option consist of 2 choice, the first choice is “Edit Formulation” and second is “Delete Formulation”. Next, the user select “Delete Formulation” for deleting the formulation. Then, the system will show the YES/No dialog for confirming formulation deleting. The screen will show the YES/NO like figure XX below this passage.

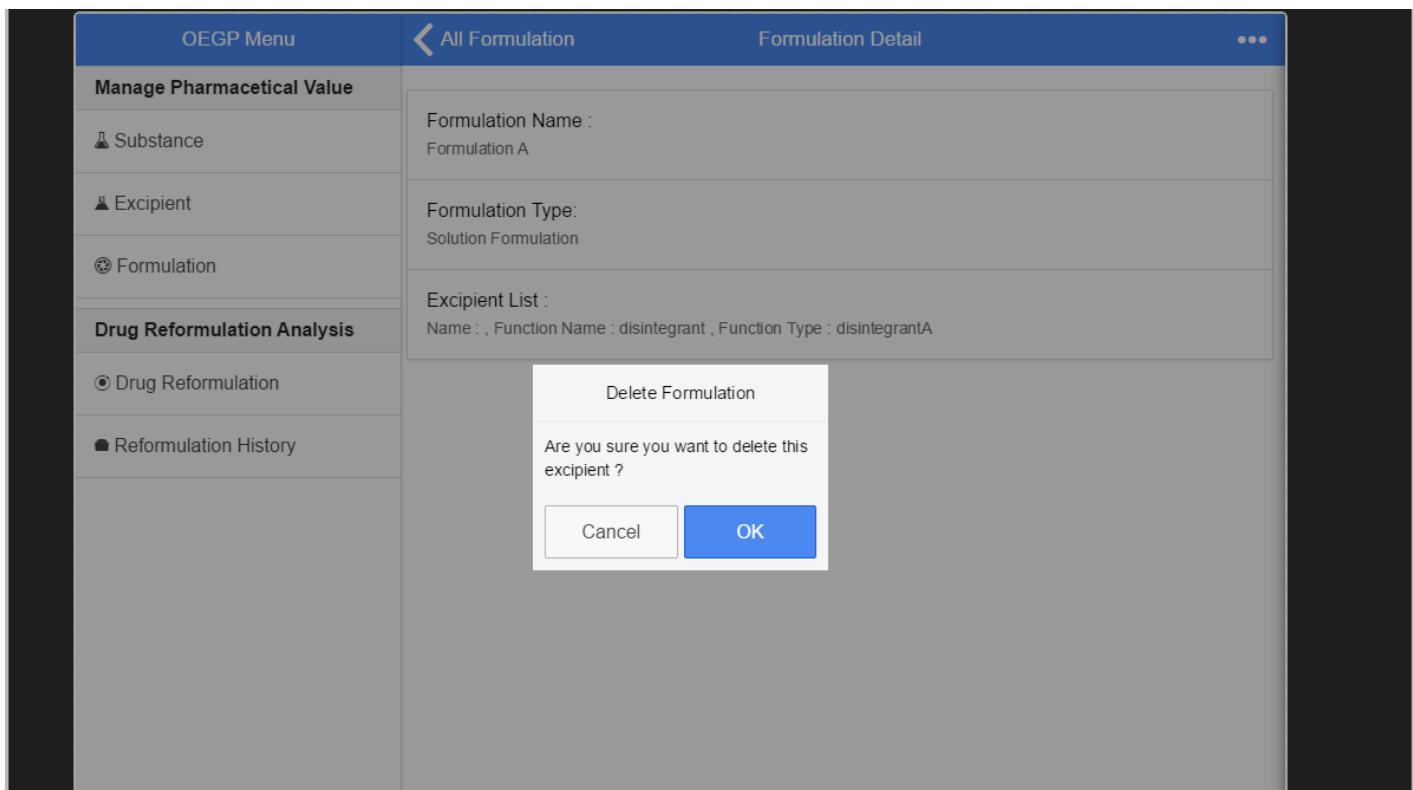


Figure 102: UI-15- The Formulation deleting form

After that, the user can select “Ok” for deleting formulation from the system. On the other hand, they can select “Cancel” for canceling the formulation deleting.

5.3.4. URS-20: The user views the formulation in the system.

The user interface of the URS-12 is same as the figure 98 (UI-11: The Formulation Main Content).

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	103 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

5.4. Sub-Feature 2: Calculate the drug reformulation by using the inference engine.

5.4.1. URS 06: The user calculates a drug reformulation by using an inference engine.

In the user interface, the user selects the “Drug Reformulation” at the menu on their left hand side. The system will show the drug reformulation main content with the list of formulation on the screen.

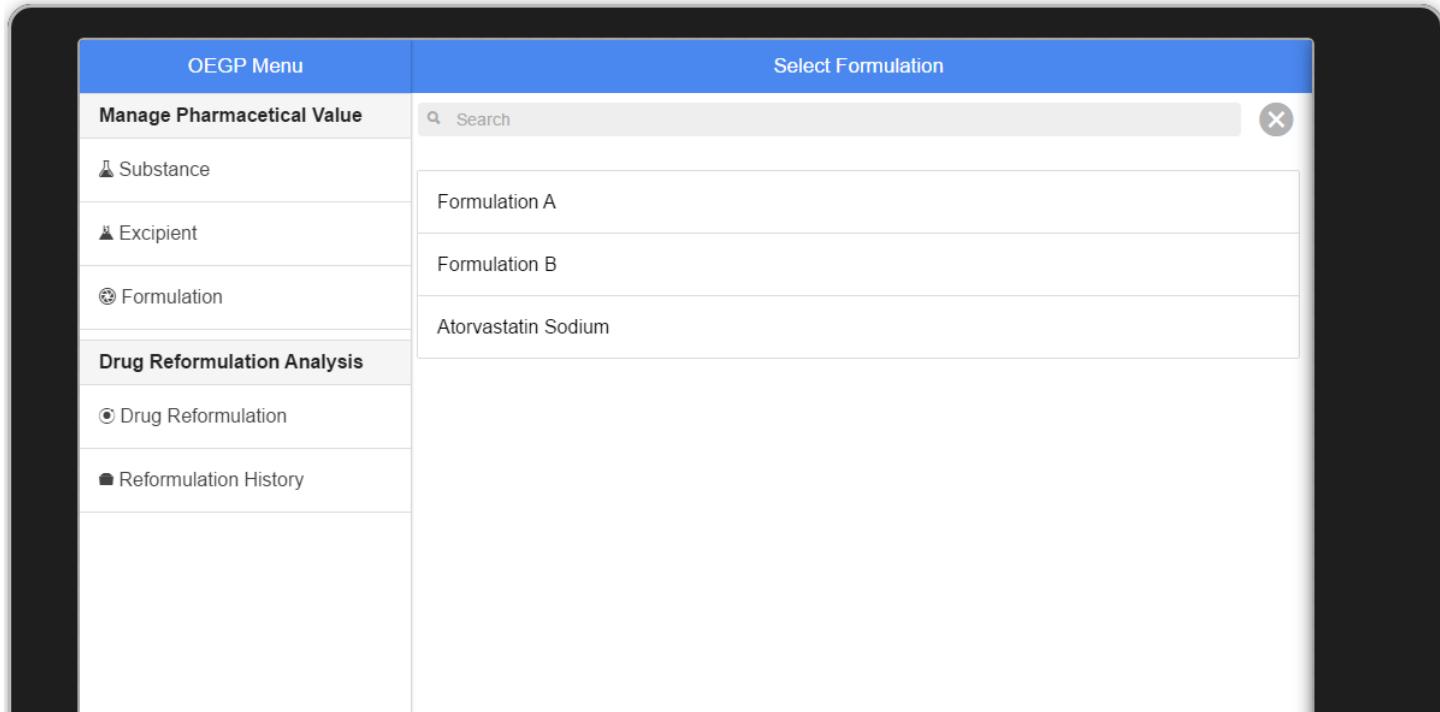


Figure 103 : UI-16- Drug Reformulation Main Content

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	104 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

After the drug reformulation main content shows on the screen, if the user wants to make drug reformulation. The user must select one of the formulation name in formulation name list. Next, the system will show the drug reformulation setting. The user must set value before make the reformulation. The reformulation setting form is showed on Figure XX.

The screenshot displays a software interface for 'Drug Reformulation Analysis'. On the left, there's a sidebar with icons for Substance, Excipient, Formulation, Drug Reformulation (which is selected), and Reformulation History. The main area contains several input fields:

- Substance:** Partition Coefficient Value : 6.00
- Excipient:** Density Value : 5.67
- Formulation:** Molecular Weight Value : 8.34
- Drug Reformulation Analysis:** Ionicity Type : Very Ionicity
- Reformulation History:** Select the Stability >

Below these, under 'Original Dosage Form Properties', are the following values:

- Total Weight (mg.) : 650.87
- Disintegration time (second) : 28
- Friability (%) : 16.87

An 'Add the Dissolution Profile' button is present. At the bottom is a large blue button labeled 'Start Reformulation'.

Figure 104: UI-17- The reformulation setting form

The user must input the setting value follow the UI-17. If they agree with their setting, they must select “Start Reformulation” for making reformulation. On the other hand, the user can select “cancel” for canceling the reformulation.

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	105 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

After the user selects the start reformulation, the system sends the production to server for making reformulation with the inference engine such as Rule Base, Case Base Reasoning and Hybrid Reasoning. After that, the server sends the reformulation result to the user. The reformulation result is shown on figure 105.

The screenshot shows a software interface titled 'OEGP Menu' on the left. The main area is titled 'Setting' and contains a 'Reformulation Result' section. Within this section, there is a tab bar with 'Rule Base' selected, followed by 'Case Base Reasoning' and 'Hybrid Reasoning'. Below the tabs, there are two sections: 'Formulation' and 'Instruction'. The 'Formulation' section displays 'Formulation Name : Atorvastatin Sodium' and 'Weight : 0'. The 'Instruction' section displays 'Instruction :'. The entire interface has a dark background with light-colored text and buttons.

Figure 105: UI -18- Drug Reformulation Result

From the Figure 105, the user can views the reformulation result. They can select the result section that they want to use such as rule base result. They also select “...” symbol and select “save reformulation result” for saving this reformulation to the history. On the other hand, if they are not agree with this reformulation result. They can select “setting” for resetting the input value and make reformulation again.

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	106 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

5.5. Sub-Feature 3: View the drug reformulation history.

5.5.1. URS-07: The user views their drug reformulation history.

In the user interface, the user selects the “Reformulation History” at the menu on their left hand side. The system will show the reformulation history main content with the list of reformulation history on the screen. The reformulation history main content is shown on figure XX.

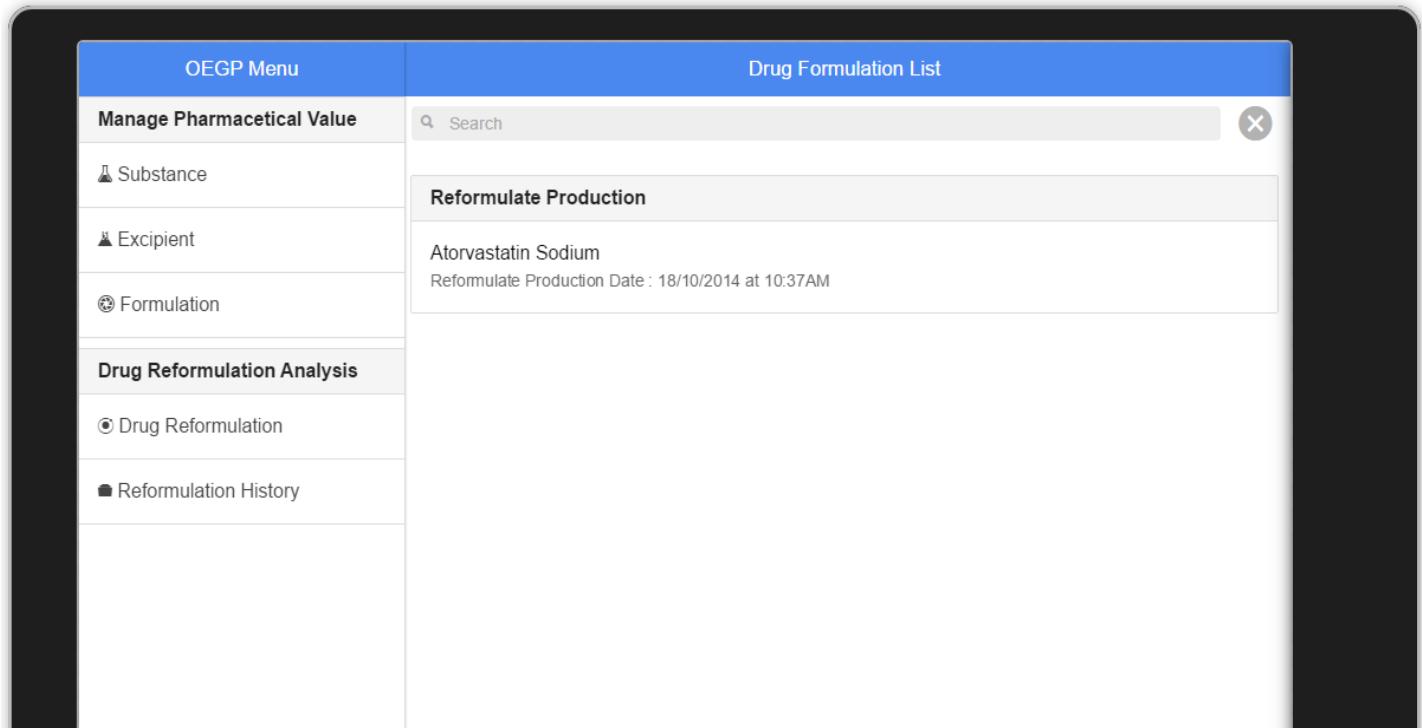


Figure 106 : UI-19 - Drug Reformulation History Main Content

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	107 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014

If the user wants to see the detail of reformulation history. They must select one of the formulation that show on the reformulation history list. The system will show the detail of reformulation history data follow item that user selected. The detail of reformulation look like figure XX below this passage.

The screenshot shows the OEGP Menu interface. On the left, there is a sidebar with the following navigation options:

- OEGP Menu**
- Manage Pharmaceutical Value**
 - Substance
 - Excipient
 - Formulation
- Drug Reformulation Analysis**
 - Drug Reformulation
 - Reformulation History

The main content area is titled "Reformulation History". It has three tabs at the top: "Rule Base" (which is selected), "Case Base Reasoning", and "Hybrid Reasoning". Below the tabs, there are several sections with data:

- Formulation**: Formulation Name : Atorvastatin Sodium
- DF Property**: Weight : 6.43
- Instarction**: Total Weight : 14, Disgradation Time : 50, hardness : 90
- Instruction**: Instruction :

Figure 107: UI-20 the reformulation History Detail

Document Name	OEGP-Software Design document_V.2.0.docx	Owner	NS,PW	Page	108 / 112
Document Type	Software Design Document	Release Date	October 22, 2014	Print Date	October 22, 2014