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

[[1]](#footnote-1)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Document Name** | **Version** | **Status** | **Date** | **Viewable** | **Reviewer** | **Responsible** |
| **Documents** | | | | | | |
| **OEGP –Software Design Document\_v.0.1.docx** | **0.1**   * 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**   * 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**   * Update the figure number | Release | July 6, 2014 | NS, PW, CD | NS, PW | NS, PW |
| **OEGP-Software Design Document\_v1.1** | **0.1**   * 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 |

Contents

[Document History ii](#_Toc394518640)

[Chapter 1 | Introduction 1](#_Toc394518641)

[1.1. Identification 1](#_Toc394518642)

[1.2. Acronyms 2](#_Toc394518643)

[Chapter 2 | Sub-Feature and class diagram relationship 2](#_Toc394518644)

[Chapter 3 | Class Diagram 3](#_Toc394518645)

[3.1. Sub-Feature 5: Manage the drug substance property 3](#_Toc394518646)

[3.1.1- CD-01: Solubility Class diagram 4](#_Toc394518647)

[3.1.2- CD-02: Degradation mechanism Class Diagram 9](#_Toc394518648)

[3.1.3- CD-03: Kinetic Reaction Class Diagram 14](#_Toc394518649)

[3.1.4- CD-04: Pka Class diagram 19](#_Toc394518650)

[3.1.5- CD-05: PartitionCoefficient Class diagram 24](#_Toc394518651)

[3.1.6- CD-06: Solidstate Class Diagram 29](#_Toc394518652)

[3.1.7- CD-07: Hygroscopicity Class Diagram 34](#_Toc394518653)

[3.1.8- CD-08: ParticleSize Class diagram 39](#_Toc394518654)

[3.1.9- CD-09: Flowability Class diagram 44](#_Toc394518655)

[3.1.10- CD-10: Density Class diagram 49](#_Toc394518656)

[3.1.11- CD-11: CompoundFunction Class diagram 54](#_Toc394518657)

[3.2 Sub-Feature 6: Manage the drug substance 59](#_Toc394518658)

[3.2.1- CD-12: Substance Class diagram 59](#_Toc394518659)

[3.3 Sub-Feature 7: Manage the drug Excipient 65](#_Toc394518660)

[3.3.1- CD-13: Excipient Class diagram 66](#_Toc394518661)

[3.4 Sub-Feature 8: Manage the drug formulation 71](#_Toc394518662)

[3.4.1- CD-14: TabletFormulation Class diagram 71](#_Toc394518663)

[Chapter 4 | Sequence Diagram 76](#_Toc394518664)

[4.1- Sub-Feature 5: Manage the drug substance property 78](#_Toc394518665)

[4.1.1 URS-09: The user adds a new substance property into the system. 78](#_Toc394518666)

[4.1.2 URS-10: The user updates an existing substance property into the system. 89](#_Toc394518667)

[4.1.3 URS-11: The user deletes an existing substance property from the system. 100](#_Toc394518668)

[4.2- Sub-Feature 6: Manage the drug substance 111](#_Toc394518669)

[4.2.1 URS-12: The user adds a new substance to the system. 111](#_Toc394518670)

[4.2.2 URS-13: The user updates an existing substance in the system. 112](#_Toc394518671)

[4.2.3 URS-14: The user deletes an existing substance from the system. 113](#_Toc394518672)

[4.2.4 URS-15: The user views the substance in the system. 114](#_Toc394518673)

[4.3- Sub-Feature 7: Manage the drug substance 115](#_Toc394518674)

[4.3.1 URS-16: The user adds a new excipient to the system. 115](#_Toc394518675)

[4.3.2 URS-17: The user updates an existing excipient in the system. 116](#_Toc394518676)

[4.3.3 URS-18: The user deletes an existing excipient from the system. 117](#_Toc394518677)

[4.3.4 URS-19: The user views the excipient in the system. 118](#_Toc394518678)

[4.4- Sub-Feature 8: Manage the drug formulation 119](#_Toc394518679)

[4.4.1 URS-20: The user adds a new drug’s formulation to the system. 119](#_Toc394518680)

[4.4.2 URS-21: The user updates an existing drug’s formulation in the system. 120](#_Toc394518681)

[4.4.3 URS-22: The user deletes an existing drug’s formulation from the system. 121](#_Toc394518682)

[4.4.4 URS-23: The user views the drug’s formulation in the system. 122](#_Toc394518683)

[Chapter 5 | User Interface 123](#_Toc394518684)

[5.1- Sub-Feature 5: Manage the drug substance property 124](#_Toc394518685)

[5.1.1 URS-09: The user adds a new substance property into the system. 124](#_Toc394518686)

[5.1.2 URS-10: The user updates an existing substance property into the system. 127](#_Toc394518687)

[5.1.3 URS-11: The user deletes an existing substance property from the system. 128](#_Toc394518688)

[5.2- Sub-Feature 6: Manage the drug substance 129](#_Toc394518689)

[5.2.1 URS-12: The user adds a new substance to the system. 129](#_Toc394518690)

[5.2.2 URS-13: The user updates an existing substance in the system. 131](#_Toc394518691)

[5.2.3 URS-14: The user deletes an existing substance from the system. 132](#_Toc394518692)

[5.2.4 URS-15: The user views the substance in the system 132](#_Toc394518693)

[5.3- Sub-Feature 7: Manage the drug excipient 133](#_Toc394518694)

[5.3.1 URS-16: The user adds a new excipient to the system. 133](#_Toc394518695)

[5.3.2 URS-17: The user updates an existing excipient in the system. 135](#_Toc394518696)

[5.2.3 URS-18: The user deletes an existing excipient from the system. 135](#_Toc394518697)

[5.2.4 URS-19: The user views the excipient in the system 135](#_Toc394518698)

[5.4- Sub-Feature 8: Manage the drug formulation 136](#_Toc394518699)

[5.4.1 URS-20: The user adds a new formulation to the system. 136](#_Toc394518700)

[5.4.2 URS-21: The user updates a tablet formulation in the system. 138](#_Toc394518701)

[5.4.3 URS-22: The user deletes an existing tablet formulation from the system. 139](#_Toc394518702)

[5.4.4 URS-23: The user views the tablet formulation in the system 139](#_Toc394518703)

# Introduction

## 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 Figure 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 in the 1st progress

## 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 |

# Use Case and class diagram relationship

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

**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)

**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)

# Class Diagram

## 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.

### CD-SV-01: Substance Class diagram (Server Side)

##### Class diagram

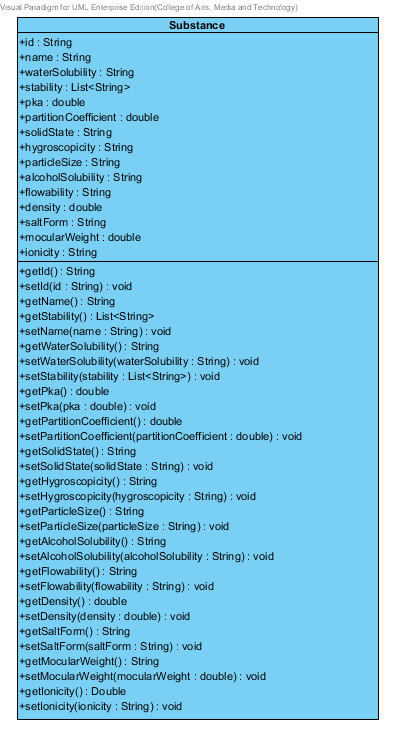


**Figure 55 - CD-12: Substance Class diagram**

##### Class description

From the figure 55, it can divide into 4 important classes. The detail of each class is described on the next paragraph.

###### Substance class



**Figure 56 – The substance class**

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.

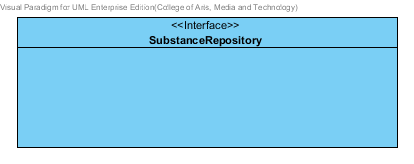
Attribute description

* + - **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.

Method description

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

###### SubstanceRepository Interface



**Figure 57 – The SubstanceRepository Interface**

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.

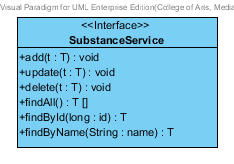
Attribute description

N/A

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.

###### SubstanceService



**Figure 58 – The SubstanceService Interface.**

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

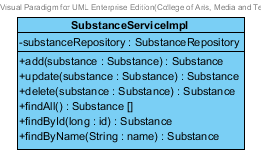
Attribute description

N/A

Method description

* + - **add (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.
    - **update (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.
    - **delete (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 object that input by the user is not contained in the database. This method will return a null value to the user.
    - **findAll() : Substance []** – The findAll 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.
    - **findById(id : long) : Substance** – The findById 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.
    - **findByName(name : String)** –The findByName method is used when the user wants to get 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 name that input by user is not contained in the database. This method will return null value to the user.

###### SubstanceServiceImpl



**Figure 59 – The SubstanceServiceImpl class**

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

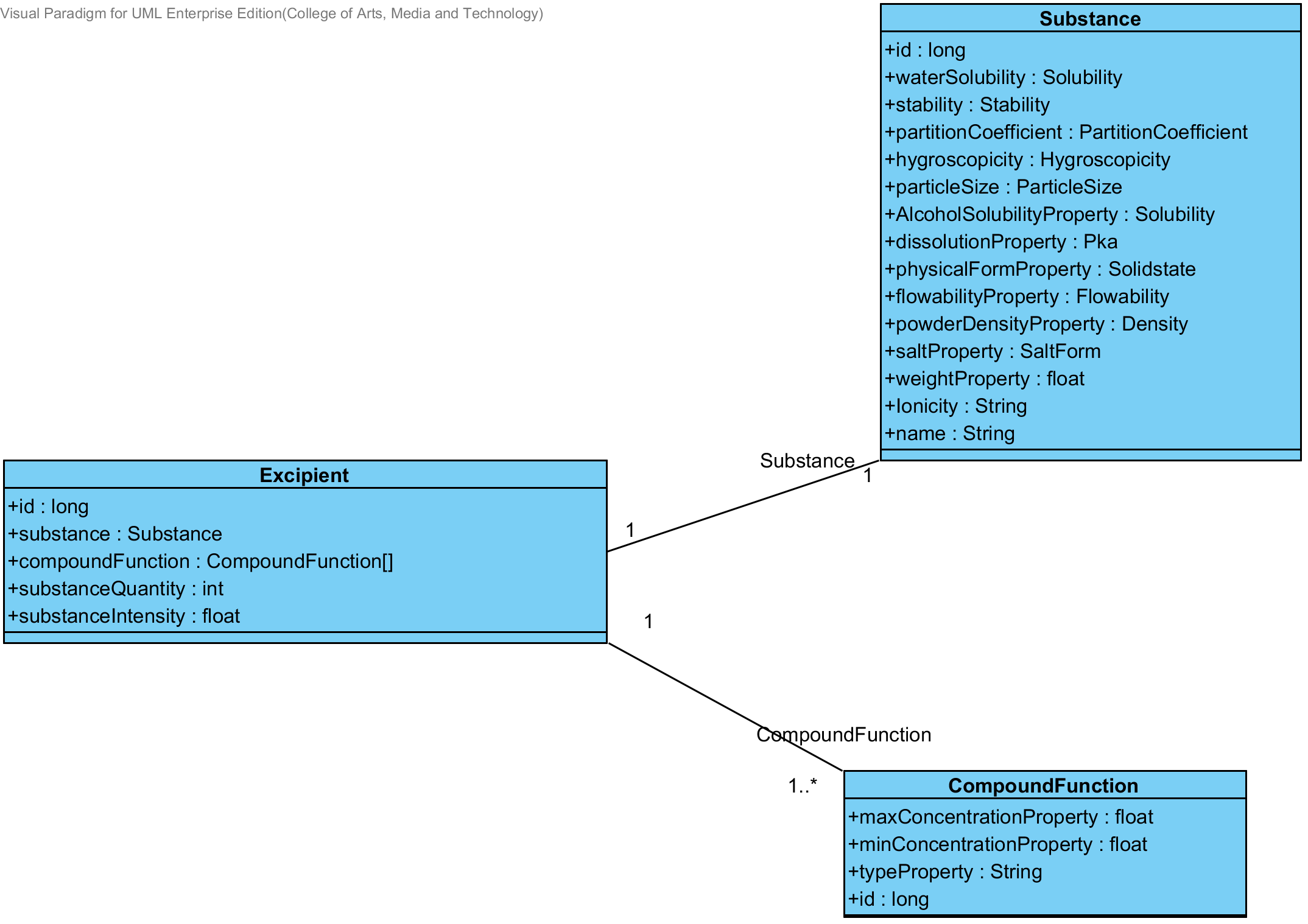
Attribute description

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

Method description

Same as SubstanceService

## 3.3 Sub-Feature 7: Manage the drug Excipient

 The Sub-Feature 7 is the excipient management. The user can add, update, delete and view the excipient. Each excipient is created from the substance and compoundfunction. The relationship between excipient entity, substance entity, and compoundfunction is illustrated in the Figure 60 below on this passage.

**Figure- 60: The entity relationship between an excipient, substance and compoundFunction.**

### 3.3.1- CD-13: Excipient Class diagram

##### 3.3.1.1: Class diagram

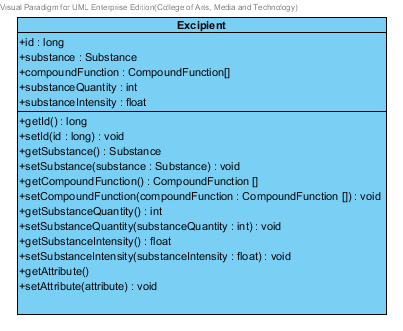


**Figure 61- CD-13: Excipient Class diagram**

##### 3.3.1.2: Class description

From the figure 61, it can divide into 4 important classes. The detail of each class is described on the next paragraph.

###### 3.3.1.2.1 Excipient class



**Figure 62 – The excipient class**

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.

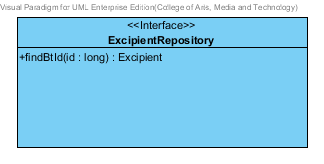
3.3.1.2.1.1: Attribute description

* + - **Id** – the identity of the excipient class. Id attribute is a long number.
    - **Substance** – the substance of the excipient class. The substance attribute is a substance object.
    - **CompoundFunction –**the compoundFunction of the excipient class .The user can set the compoundFunction in each substance.The compoundFunction is a compoundFunction object.
    - **substanceQuantity** – the substance quantity of the excipient. The substance quantity is integer number.
    - **substanceIntensity** –the substance intensity of the excipient. The substance intensity is the float value.

3.3.1.2.1.2: Method description

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

###### 3.3.1.2.2 ExcipientRepository Interface



**Figure 63 – The excipient repository interface**

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

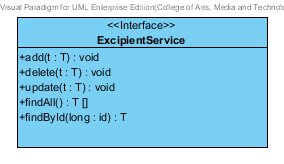
3.2.1.2.2.1: Attribute description

N/A

3.2.1.2.2.2: 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 a 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.
    - **findById (id: long): Excipient** – The findById 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.

###### 3.3.1.2.3 ExcipientService



**Figure 64 – The ExcipientService Interface.**

ExcipientService is business processing logic for excipient entity. ExcipientService manages the excipient data through the ExcipientRepository interface. ExcipientService consists of 6 methods follow the list below this passage.

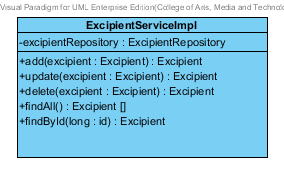
3.2.1.2.3.1: Attribute description

N/A

3.2.1.2.3.2: Method description

* + - **add (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.
    - **update (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.
    - **delete (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 the user is not contained in the database. This method will return a null value to the user.
    - **findAll() : Excipient []** – The findAll 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.
    - **findById(id : long) :** Excipient – The findById 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.

###### 3.3.1.2.4 ExcipientServiceImpl



**Figure 65 – The ExcipientServiceImpl class**

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

3.3.1.2.4.1: Attribute description

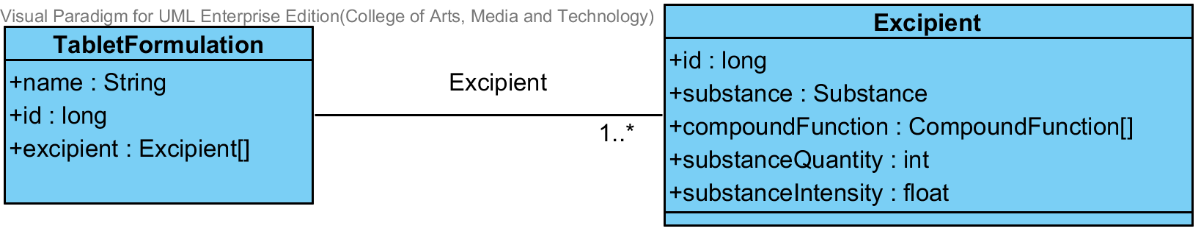
* + - **ExcipientRepository** – the repository of excipient. This attribute is used for excipient data management.

3.3.1.2.4.2: Method description

Same as ExcipientService

## 3.4 Sub-Feature 8: Manage the drug formulation

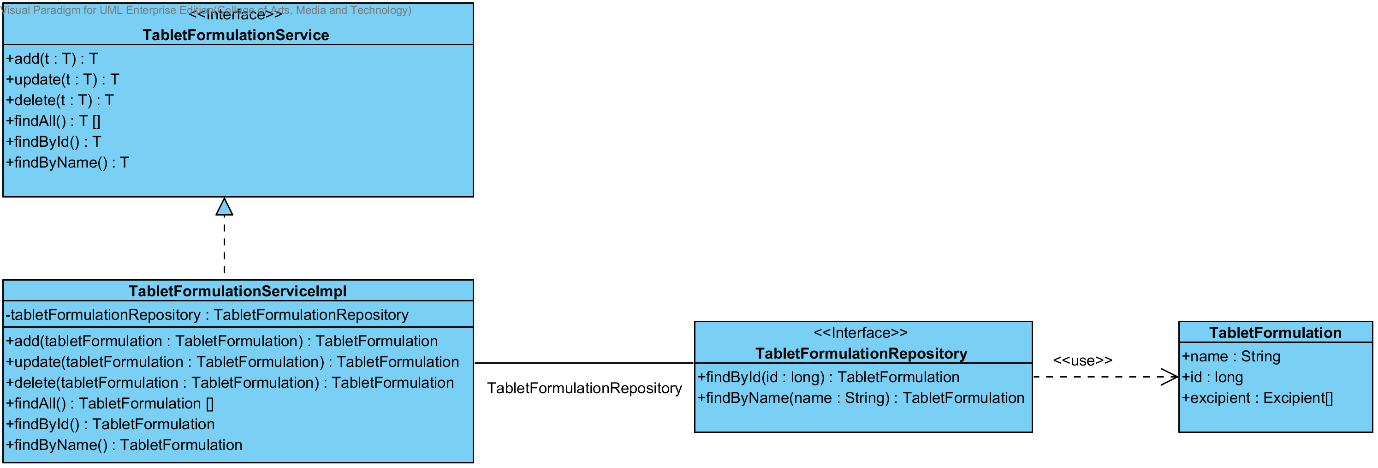
The Sub-Feature 8 is the drug’s formulation management. The users can add, update, delete and view the formulation. Each drug’s formulation is created from the excipient. The relationship between drug’s formulation and the excipient is illustrated in the Figure 66 below on this passage.



**Figure- 66: The entity relationship between a drug’s formulation and the excipient.**

### 3.4.1- CD-14: TabletFormulation Class diagram

##### 3.4.1.1: Class diagram

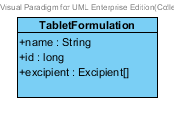


**Figure 67- CD-14: TabletFormulation Class diagram**

##### 3.4.1.2: Class description

From the figure 67, it can divide into 4 important classes. The detail of each class is described on the next paragraph.

###### 3.4.1.2.1 TabletFormulation class



**Figure 68 – The TabletFormulation class**

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

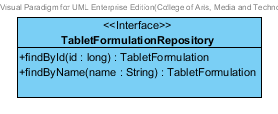
3.4.1.2.1.1: Attribute description

* + - **Id** – the identity of the TabletFormulation class. Id attribute is a long number.
    - **Name** – the name of the TabletFormulation class. The TabletFormulation attribute is a TabletFormulation object.
    - **Excipient –**the excipient of the TabletFormulation. The user can set the excipient more than 1 excipient in each tabletformulation.

3.4.1.2.1.2: Method description

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

###### 3.4.1.2.2 TabletFormulationRepository Interface



**Figure 69 – The TabletFormulation interface**

TabletFormulationRepository Interface is an interface that use for CRUD with entity classes in the system. The most of TabletFormulationRepository interface’s method is generated from Spring data MongoDB framework. TabletFormulationRepository interface consists of 5 methods is shown below this passage.

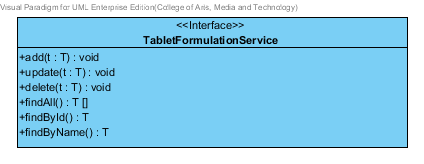
3.4.1.2.2.1: Attribute description

N/A

3.4.1.2.2.2: Method description

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

###### 3.4.1.2.3 TabletFormulationService



**Figure 70 – The TabletFormulationService Interface.**

TabletFormulationService is business processing logic for TabletFormulation entity. TabletFormulationService manages the TabletFormulation data through the TabletFormulationRepository interface. TabletFormulationService consists of 6 methods follow the list below this passage.

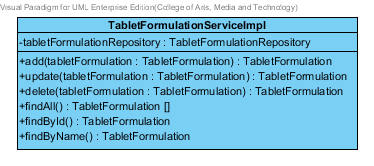
3.4.1.2.3.1: Attribute description

N/A

3.4.1.2.3.2: Method description

* + - **add (tabletFormulation: TabletFormulation)** – The adding tabletFormulation method is used, when the user wants to add a new tabletFormulation to the database. This method adds a new tabletFormulation by input variable of tabletFormulation object. If the tabletFormulation object that input by the user is not contained in the database, this method will add a new tabletFormulation to the database and return the tabletFormulation object from the database to the user after the adding tabletFormulation is successful. On the other hand, when the tabletFormulation object that input by the user is contained in the database. This method will return a null value to the user.
    - **update (tabletFormulation: TabletFormulation)** - The updating tabletFormulation method is used, when the user wants to update an existing tabletFormulation on in the database. This method update the existing tabletFormulation by input variable of tabletFormulation object. If the tabletFormulation object that input by the user is contained in the database, this method will update an existing tabletFormulation in the database and return the tabletFormulation object from the database to the user after the updating tabletFormulation is successful. On the other hand, when the tabletFormulation object that input by the user is not contained in the database. This method will return a null value to the user.
    - **delete (tabletFormulation: TabletFormulation)** The deleting tabletFormulation method is used when the user wants to deletes the existing tabletFormulation from the database. This method delete the tabletFormulation by input variable of tabletFormulation object. If the tabletFormulation object that input by the user is contained in the database, this method will delete an existing tabletFormulation from the database and return the tabletFormulation object to the user after the deleting tabletFormulation 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.
    - **findAll() : TabletFormulation []** – The findAll method is used, when the user wants to get all tabletFormulation data in the database. This method is return as a list of tabletFormulation object from the database.
    - **findById(long : id) : TabletFormulation**  – The findById method is used, when the user wants to get the tabletFormulation data in the system. This method gets tabletFormulation 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.
    - **findByName(name : String) : TabletFormulation** – The findById method is used, when the user wants to get the tabletFormulation data in the system. This method gets tabletFormulation object from the database by id that input by the user. On the other hand, if the name that input by user is not contained in the database. This method will return null value to the user.

###### 3.4.1.2.4 TabletFormulationServiceImpl



**Figure 71 – The TabletFormulationServiceImpl class**

TabletFormulationServiceImpl is the tabletformulation service class that implements the method from TabletFormulationService. So, the method of TabletFormulationServiceImpl is same as TabletFormulationService.

3.4.1.2.4.1: Attribute description

* + - **TabletFormulationRepository** – the repository of tabletFormulation. This attribute is used for tabletFormulation data management.

3.4.1.2.4.2: Method description

Same as TabletFormulationService

# Chapter 4 | Sequence Diagram

In the 1st progress the URS is related with the list of sequence diagram that shown below this passage.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Sub Sub-Feature Name** | **URS No.** | **URS Name** | **Sequence Diagram** |
| 5 | Manage the drug substance property | URS-09 | The user adds a new substance property into the system. | SQD-9A, SQD-9B, SQD-9C, SQD-9D, SQD-9E, SQD-9F, SQD-9G, SQD-9H, SQD-9I, SQD-9J, SQD-9K |
| URS-10 | The user updates an existing substance property into the system. | SQD-10A, SQD-10B, SQD-10C, SQD-10D, SQD-10E, SQD-10F, SQD-10G, SQD-10H, SQD-10I, SQD-10J, SQD-10K |
| URS-11 | The user deletes an existing substance property from the system. | SQD-11A, SQD-11B, SQD-11C, SQD-11D, SQD-11E, SQD-11F, SQD-11G, SQD-11H, SQD-11I, SQD-11J, SQD-11K |
| 6 | Manage the drug substance | URS-12 | The user adds a new substance into the system. | SQD-12 |
| URS-13 | The user updates an existing substance into the system. | SQD-13 |
| URS-14 | The user deletes an existing substance from the system. | SQD-14 |
| URS-15 | The user views the substance in the system. | SQD-15 |
| 7 | Manage the drug excipient | URS-16 | The user adds a new excipient to the system. | SQD-16 |
| URS-17 | The user updates an existing drug excipient in the system. | SQD-17 |
| URS-18 | The user delete an existing drug excipient in the system. | SQD-18 |
| URS-19 | The user views all the drug excipient in the system. | SQD-19 |
| 8 | Manage the drug formulation | URS-20 | The user adds a new drug formulation case into the system. | SQD-20 |
| URS-21 | The user updates an existing drug formulation case in the system. | SQD-21 |
| URS-22 | The user deletes an existing drug formulation case in the system. | SQD-22 |
| URS-23 | The user views all of the formulation in the system. | SQD-23 |

## 4.1- Sub-Feature 5: Manage the drug substance property

### 4.1.1 URS-09: The user adds a new substance property into the system.

In the sequence diagram, the user can add a new substance property to the system. Firstly, the user opens the substance property adding page, then the user inputs substance property value such as name, max value min value and substance value. The substance property controller gets the data from the user, after that the controller chooses appropriate the SubstancePropertyService for saving a new substance property to the system. Finally, the substance property controller shows a new substance property on the successful adding substance property page to the user.

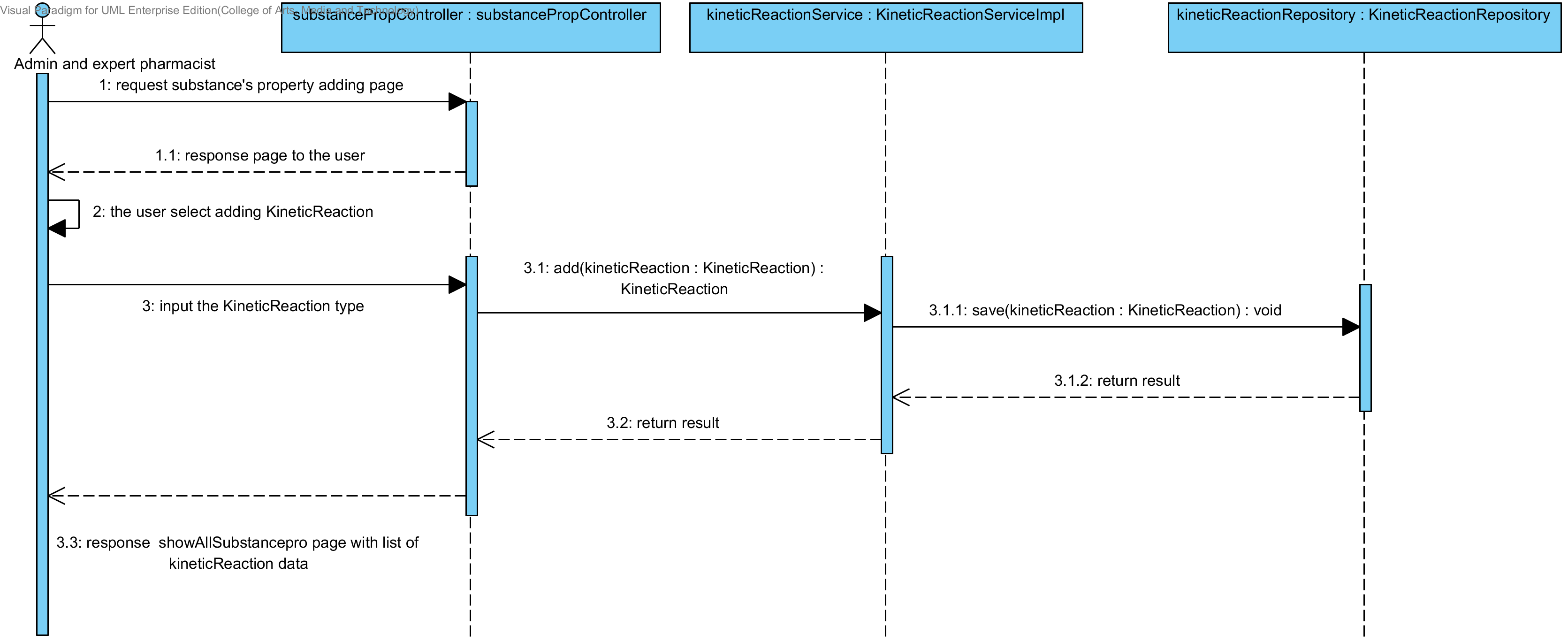
##### 4.1.1.1 SQD-09A: The user add a new drug substance property into the system (Solubility).



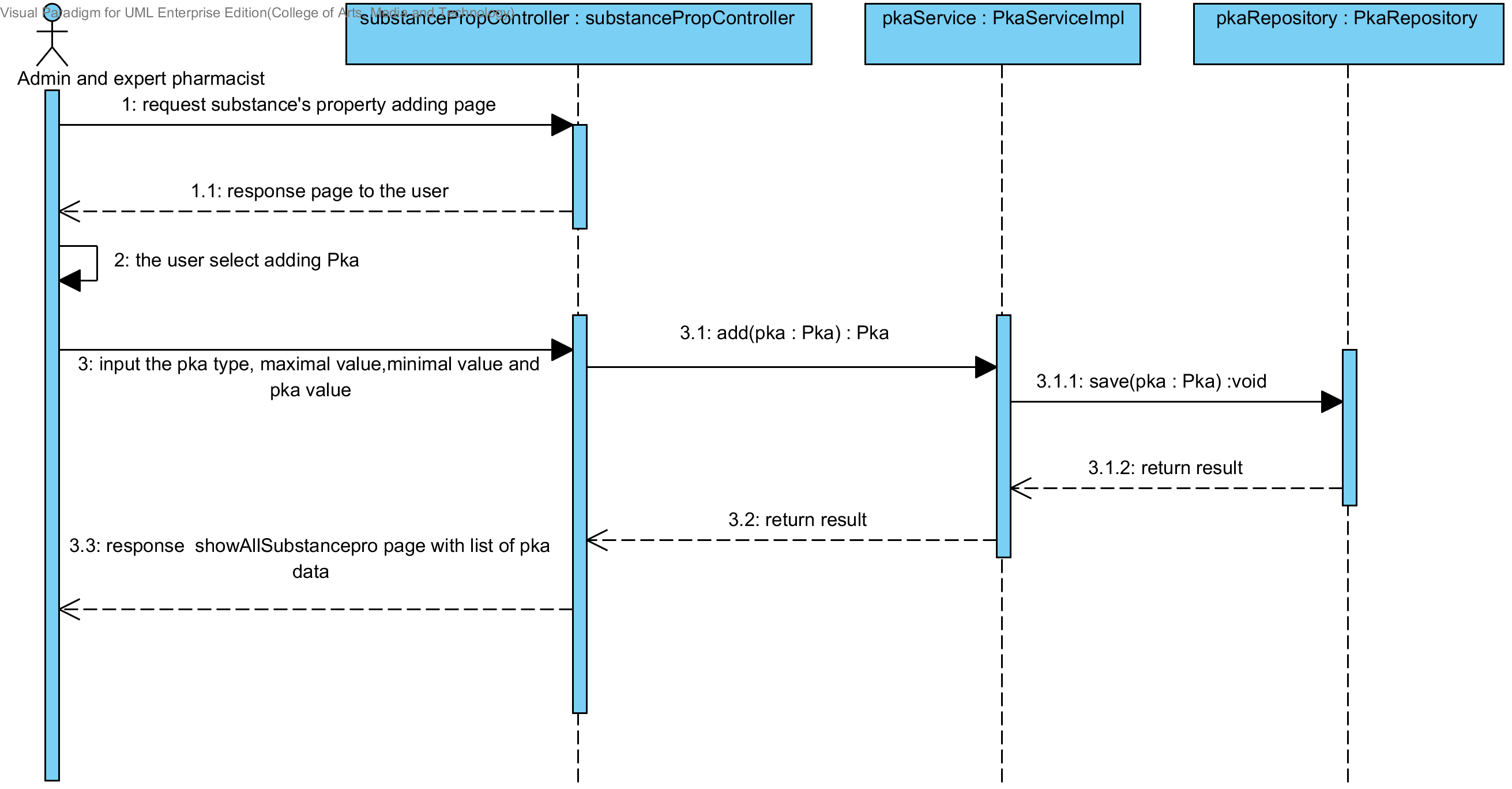
##### 4.1.1.2 SQD-09B: The user add a new drug substance property into the system (DegradationMechanism).



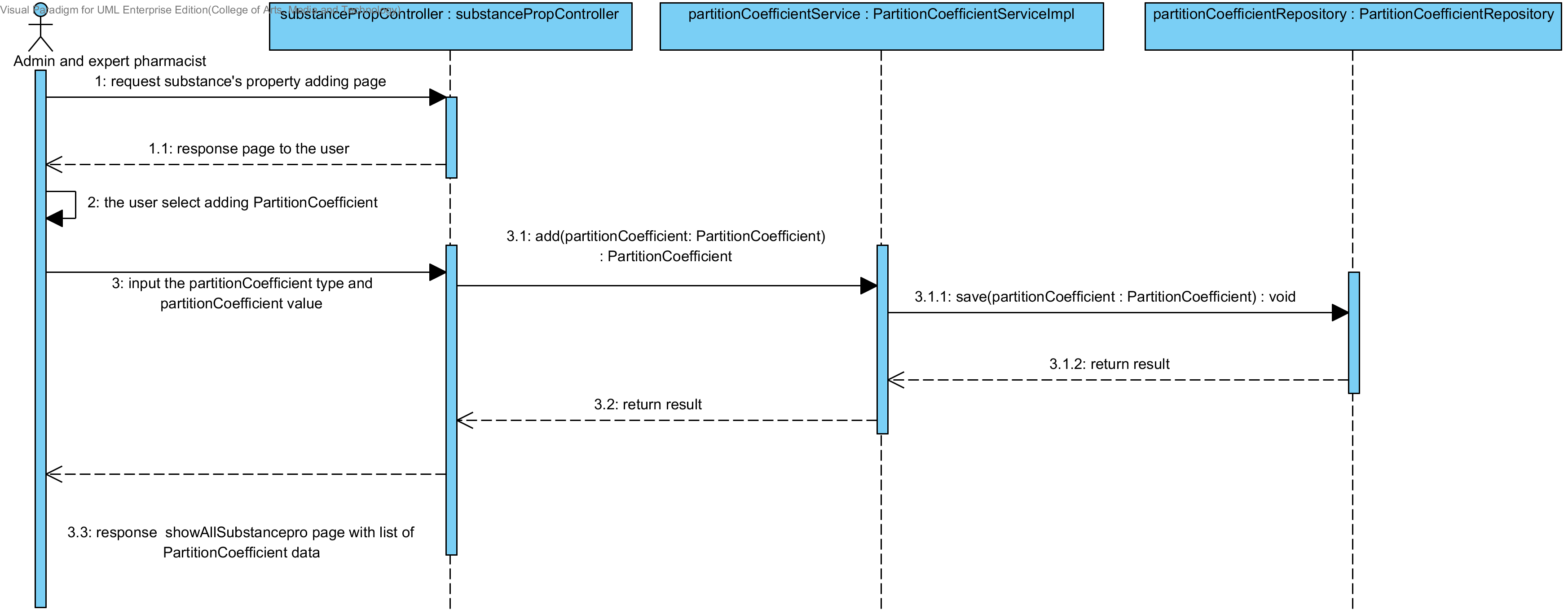
##### 4.1.1.3 SQD-09C: The user add a new drug substance property into the system (Kinetic Reaction).



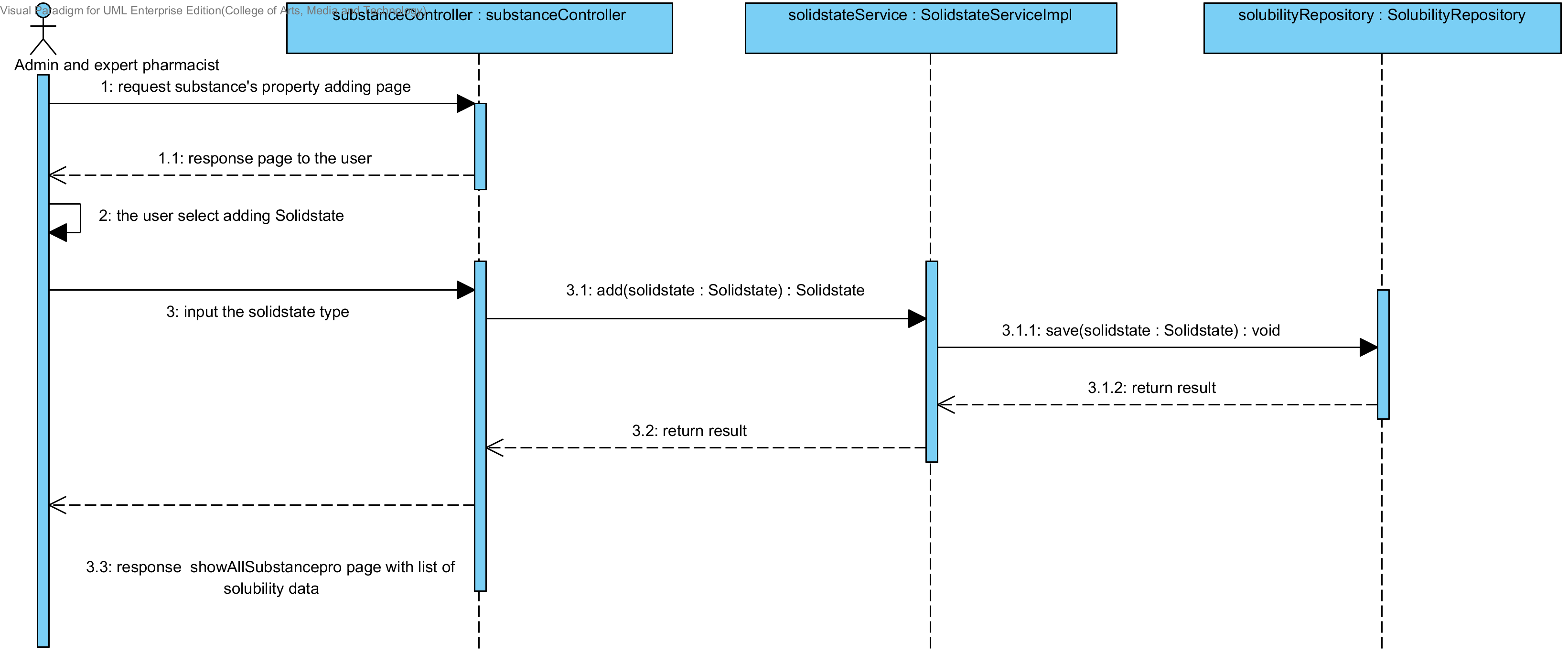
##### 4.1.1.4 SQD-09D: The user add a new drug substance property into the system (Pka).



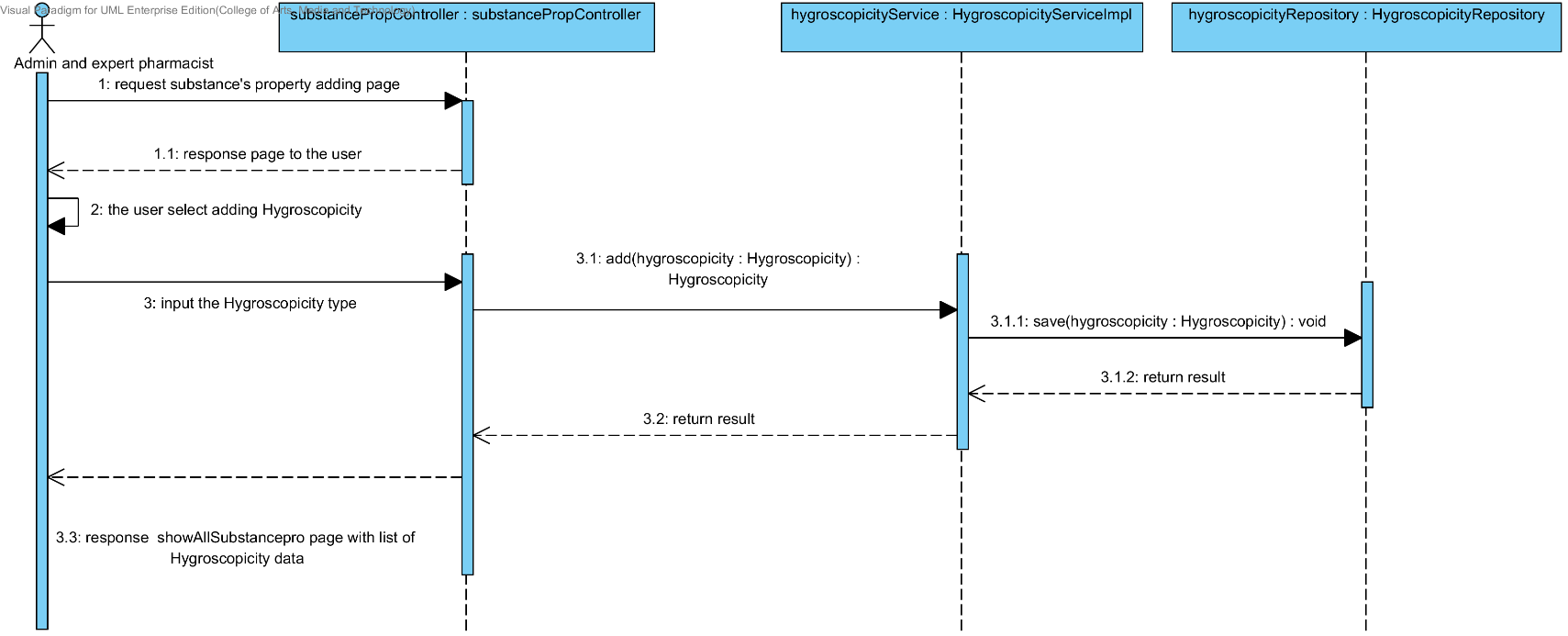
##### 4.1.1.5 SQD-09E: The user add a new drug substance property into the system (PartitionCoefficient).



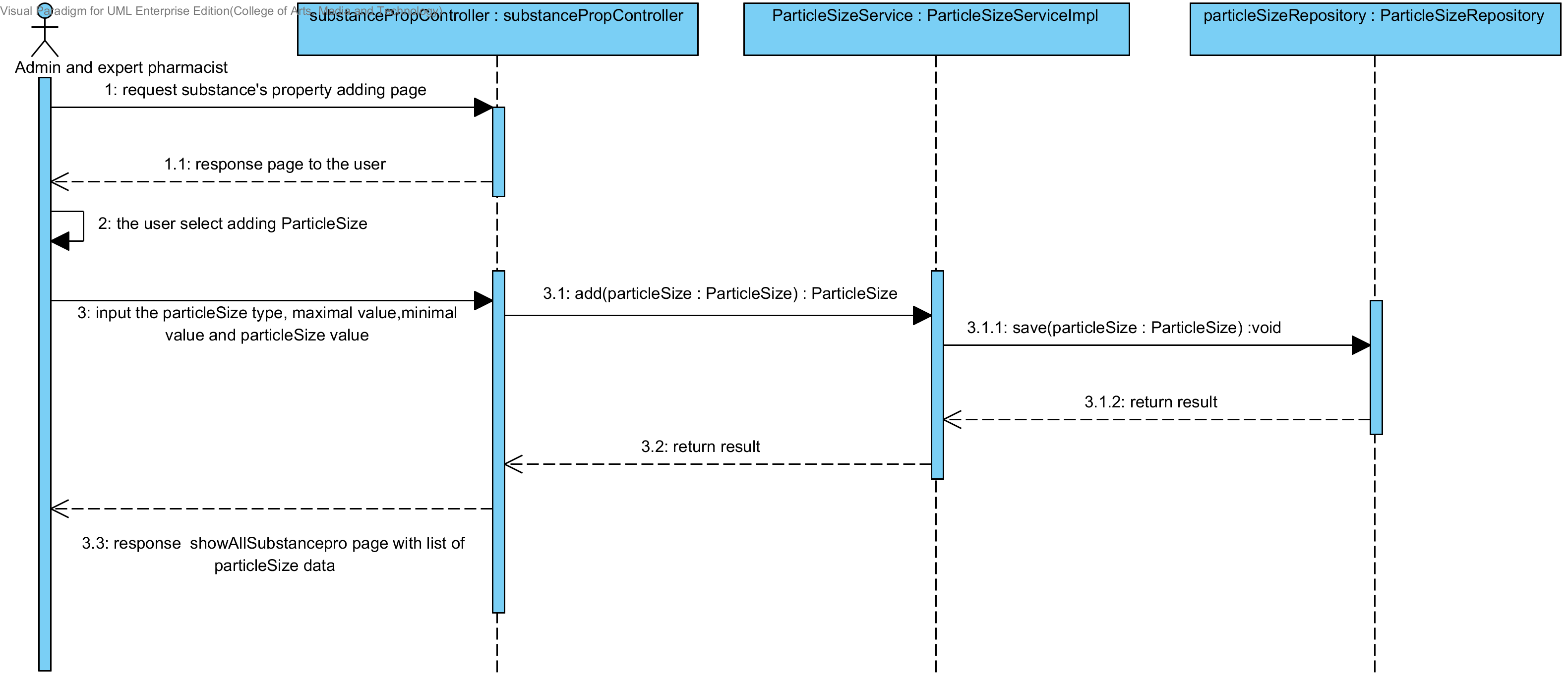
##### 4.1.1.6 SQD-09F: The user add a new drug substance property into the system (Solidstate).



##### 4.1.1.7 SQD-09G: The user add a new drug substance property into the system (Hygroscopicity).



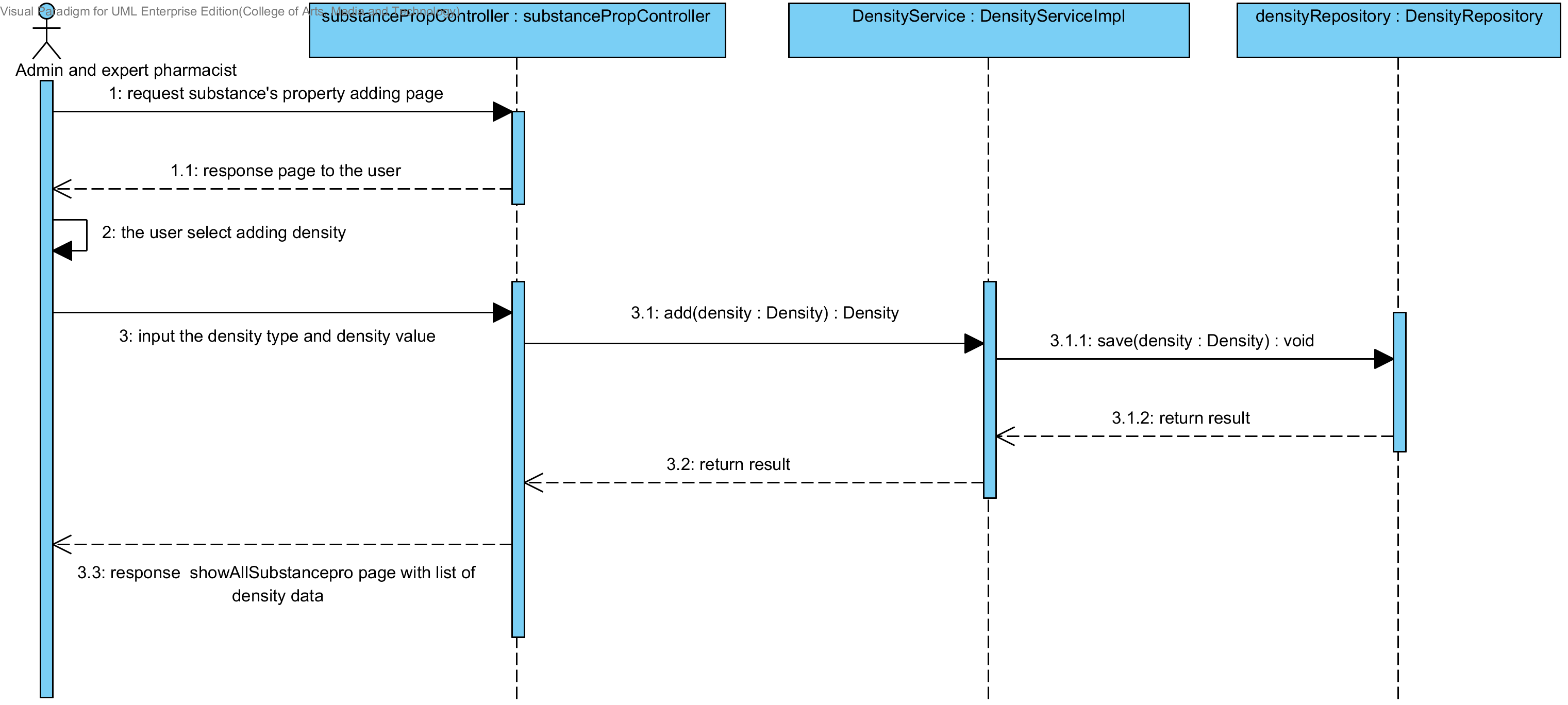
##### 4.1.1.8 SQD-09H: The user add a new drug substance property into the system (ParticleSize).



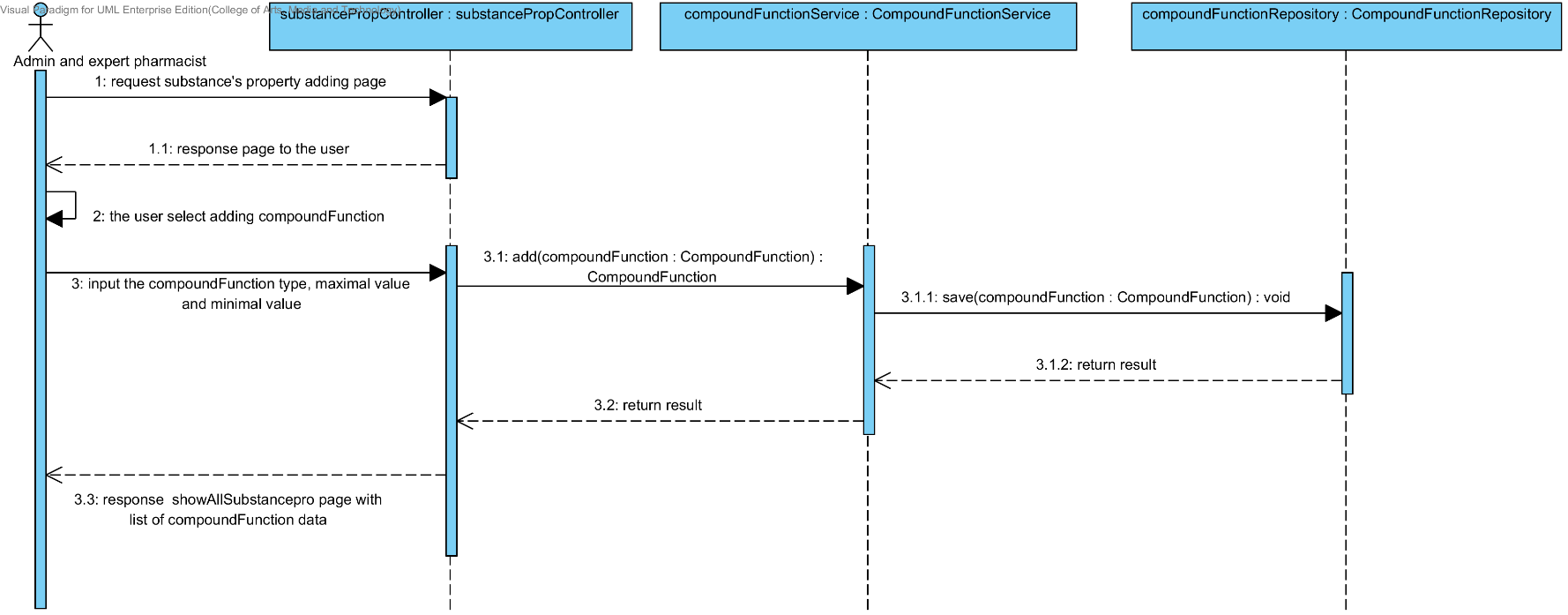
##### 4.1.1.9 SQD-09I: The user add a new drug substance property into the system (Flowability).



##### 4.1.1.10 SQD-09J: The user add a new drug substance property into the system (Density).



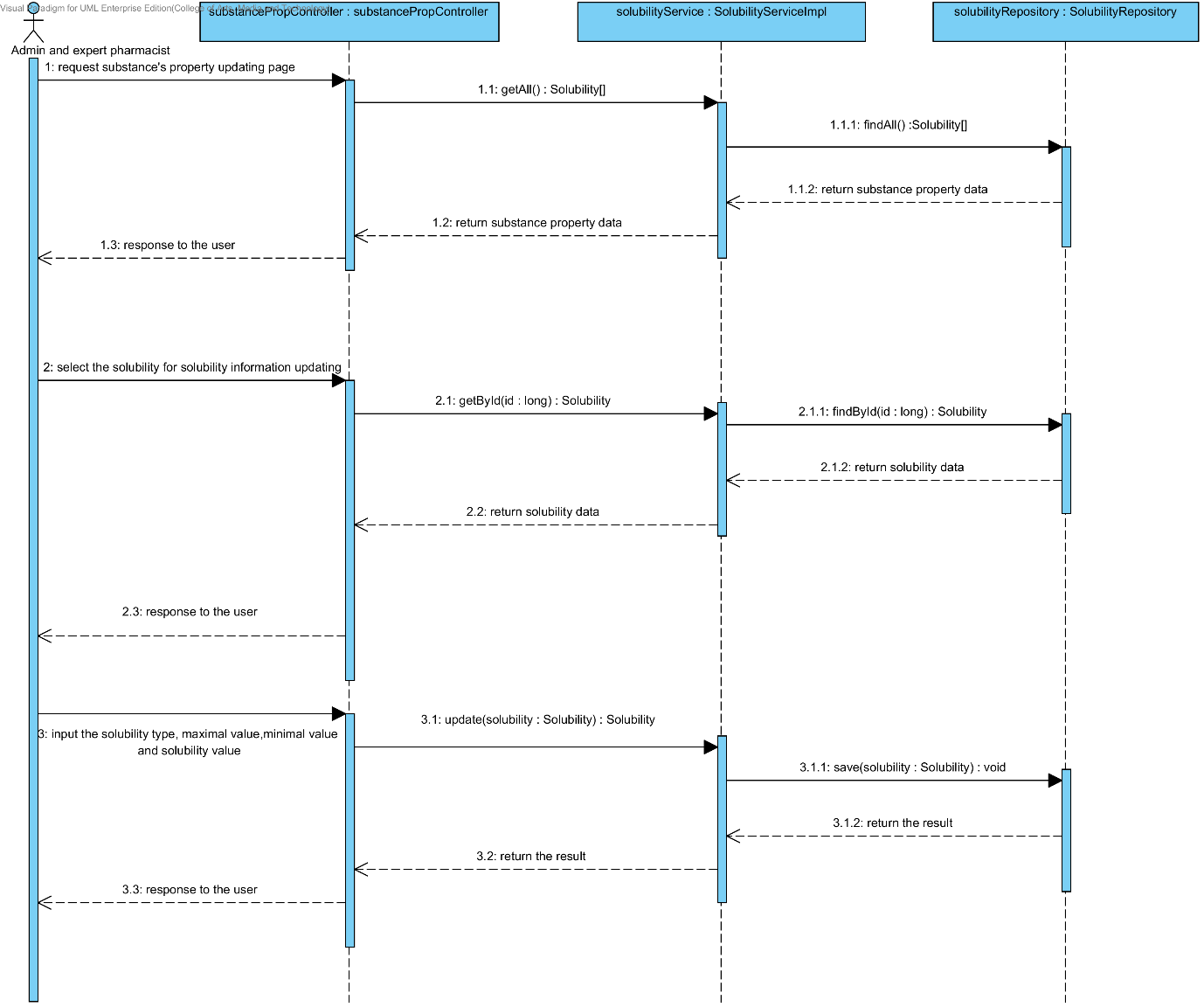
##### 4.1.1.11 SQD-09K: The user add a new drug substance property into the system (Compound function).



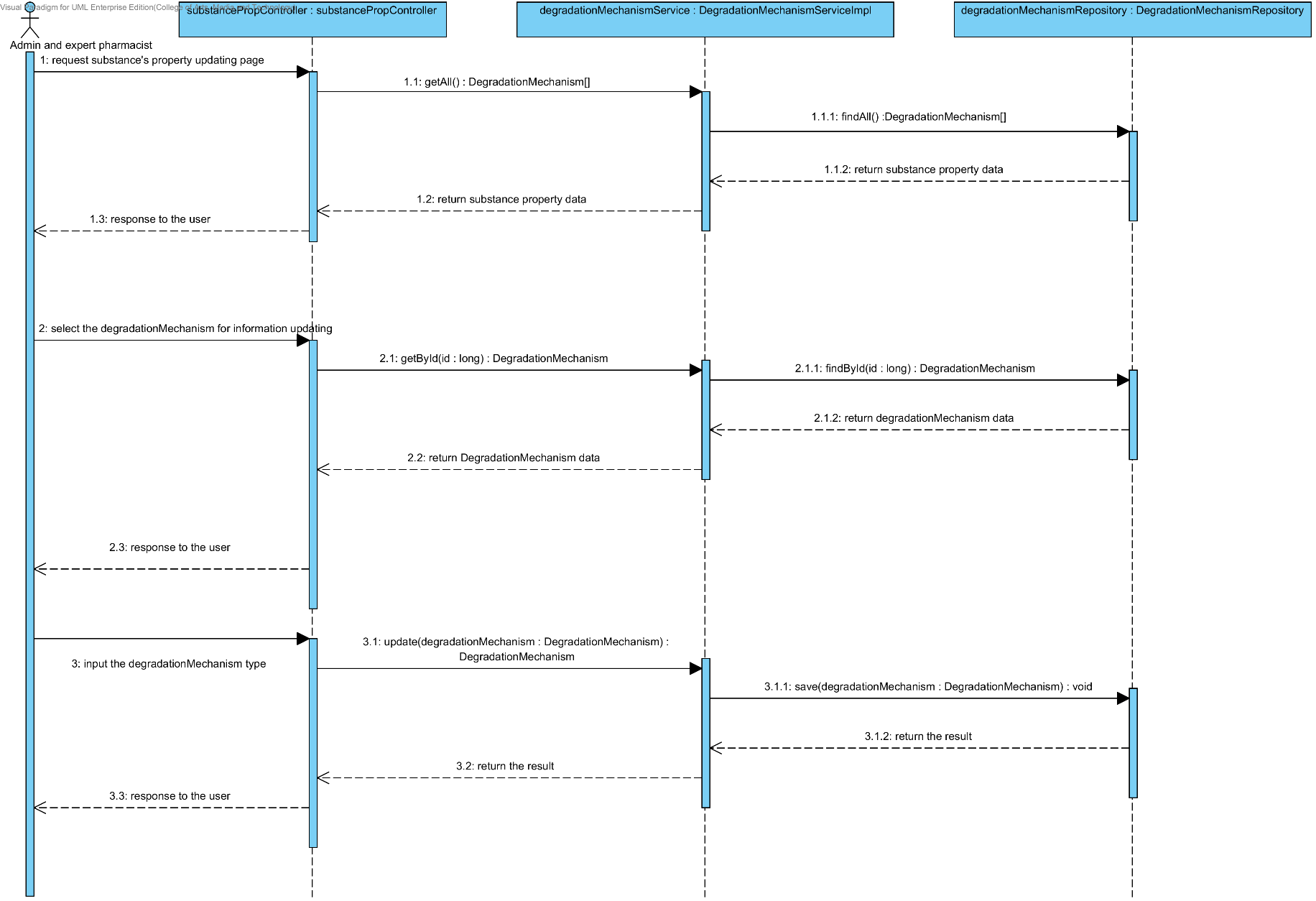
### 4.1.2 URS-10: The user updates an existing substance property into the system.

In the sequence diagram, the user can updates an existing substance property to the system. Firstly, the user opens the substance property updating page. The system shows all substance property data to the user, then the user selects the substance property for making substance property updating, after that, the user inputs substance property value such as name, max value min value and substance's property value. The substance property controller gets the data from the user, after that the controller chooses appropriate SubstanceProperty’s service for updating the existing substance property to the system. Finally, the substance property controller shows a substance property that already update on the successful adding substance property page to the user.

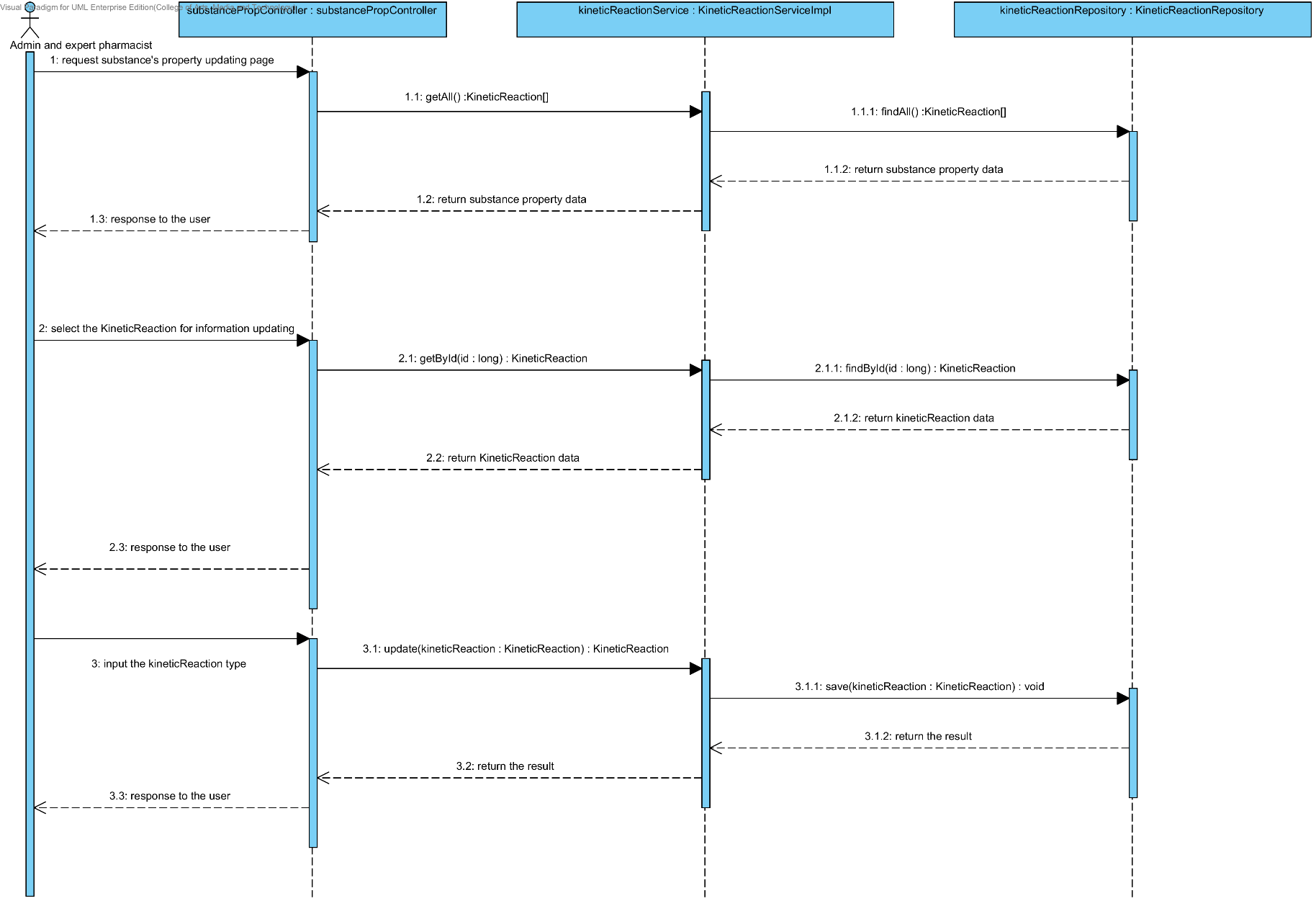
##### 4.1.2.1 SQD-10A: The user updates an existing substance’s property into the system (Solubility).



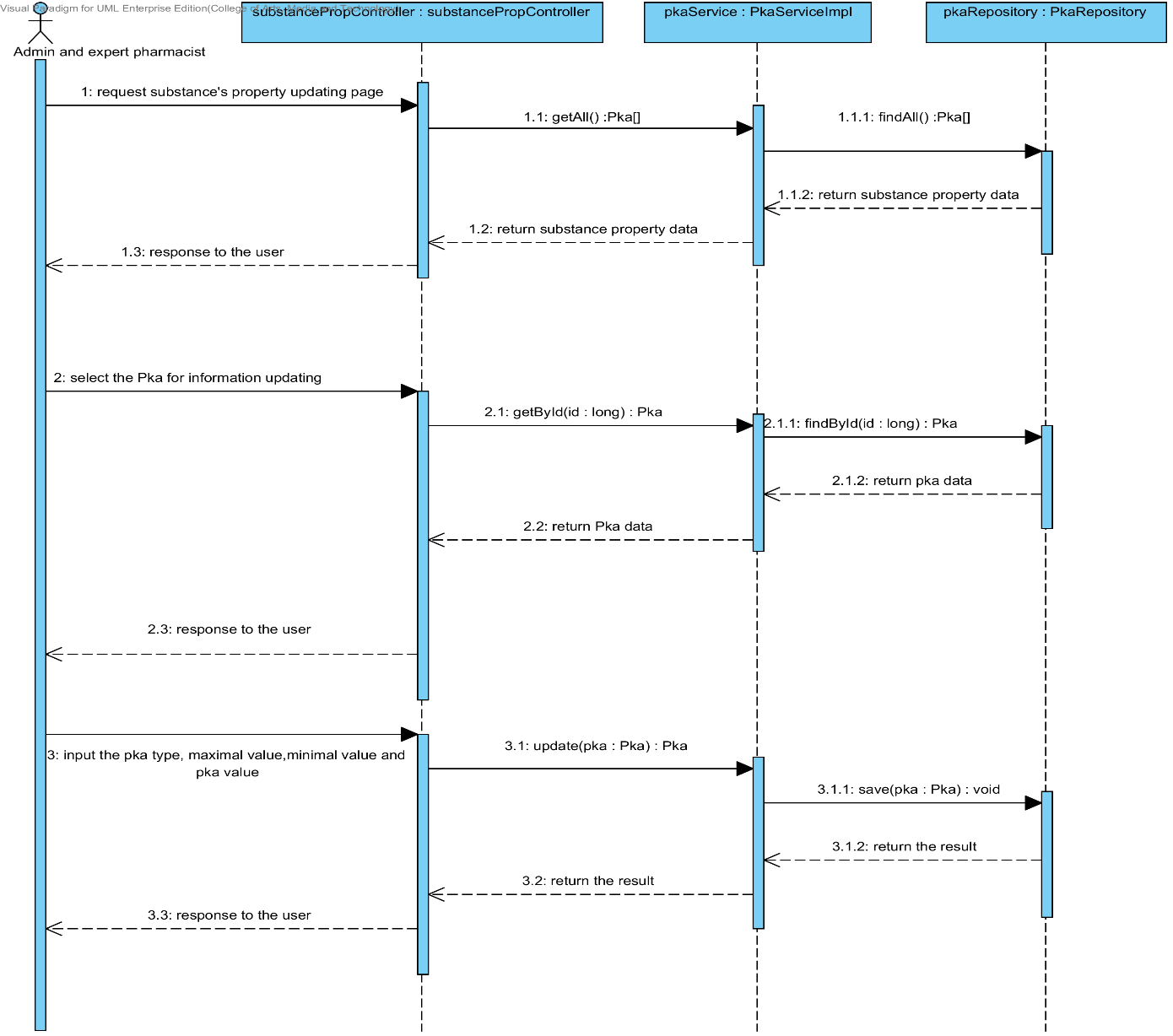
##### 4.1.2.2 SQD-10B: The user updates an existing substance property into the system (DegradationMechanism).



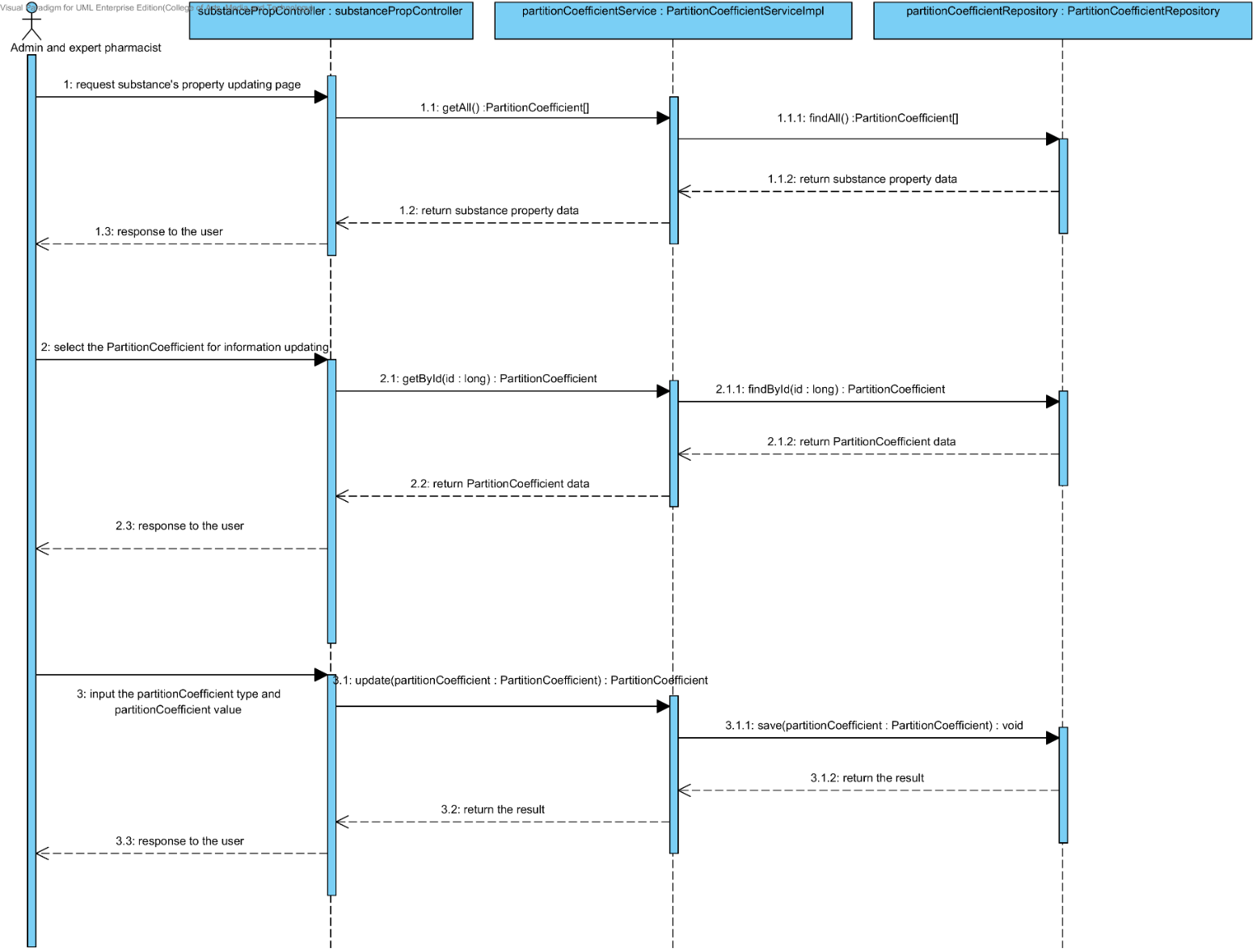
##### 4.1.2.3 SQD-10C: The user updates an existing substance property into the system (Kinetic Reaction).



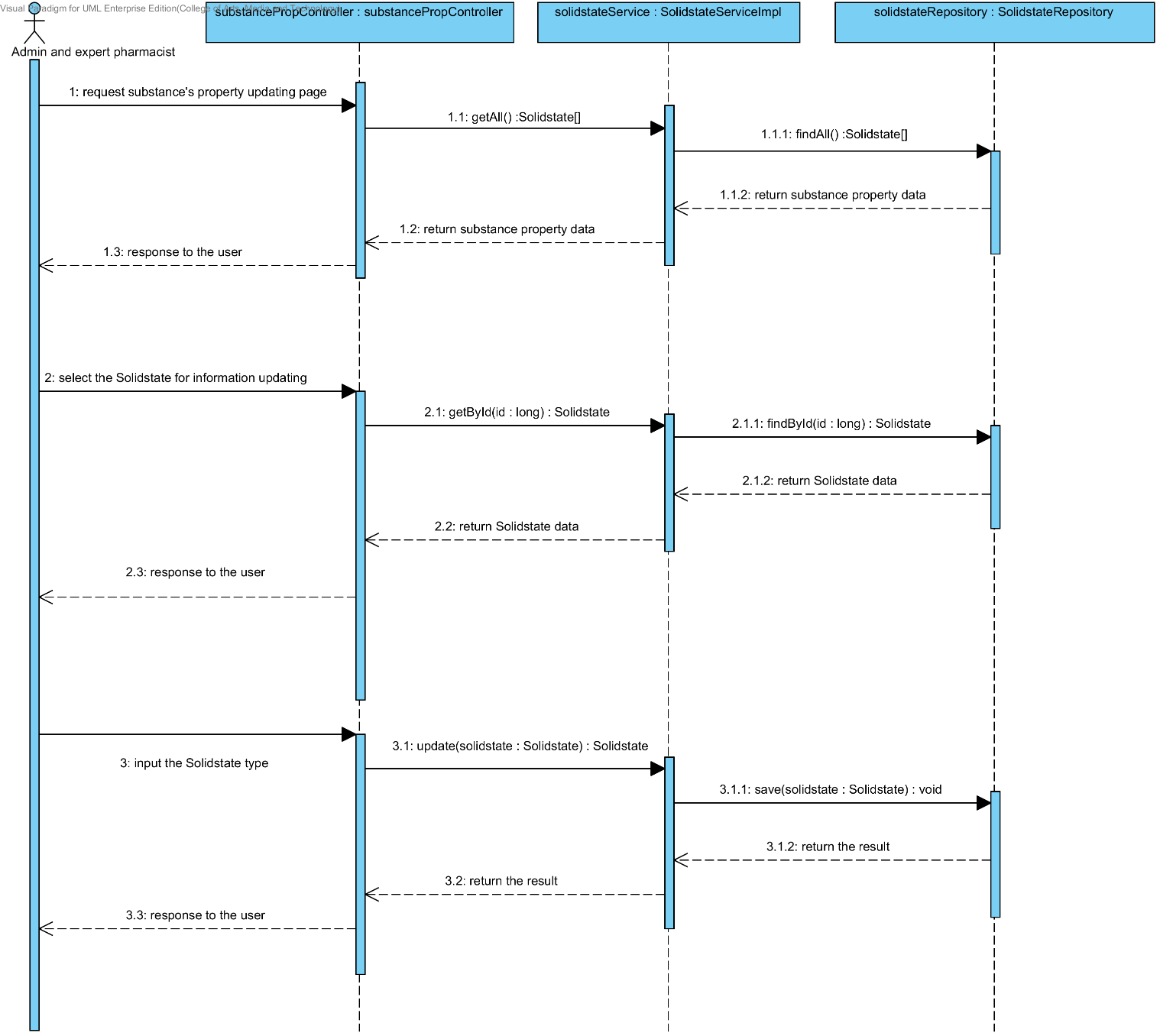
##### 4.1.2.4 SQD-10D: The user updates an existing substance property into the system (Pka).



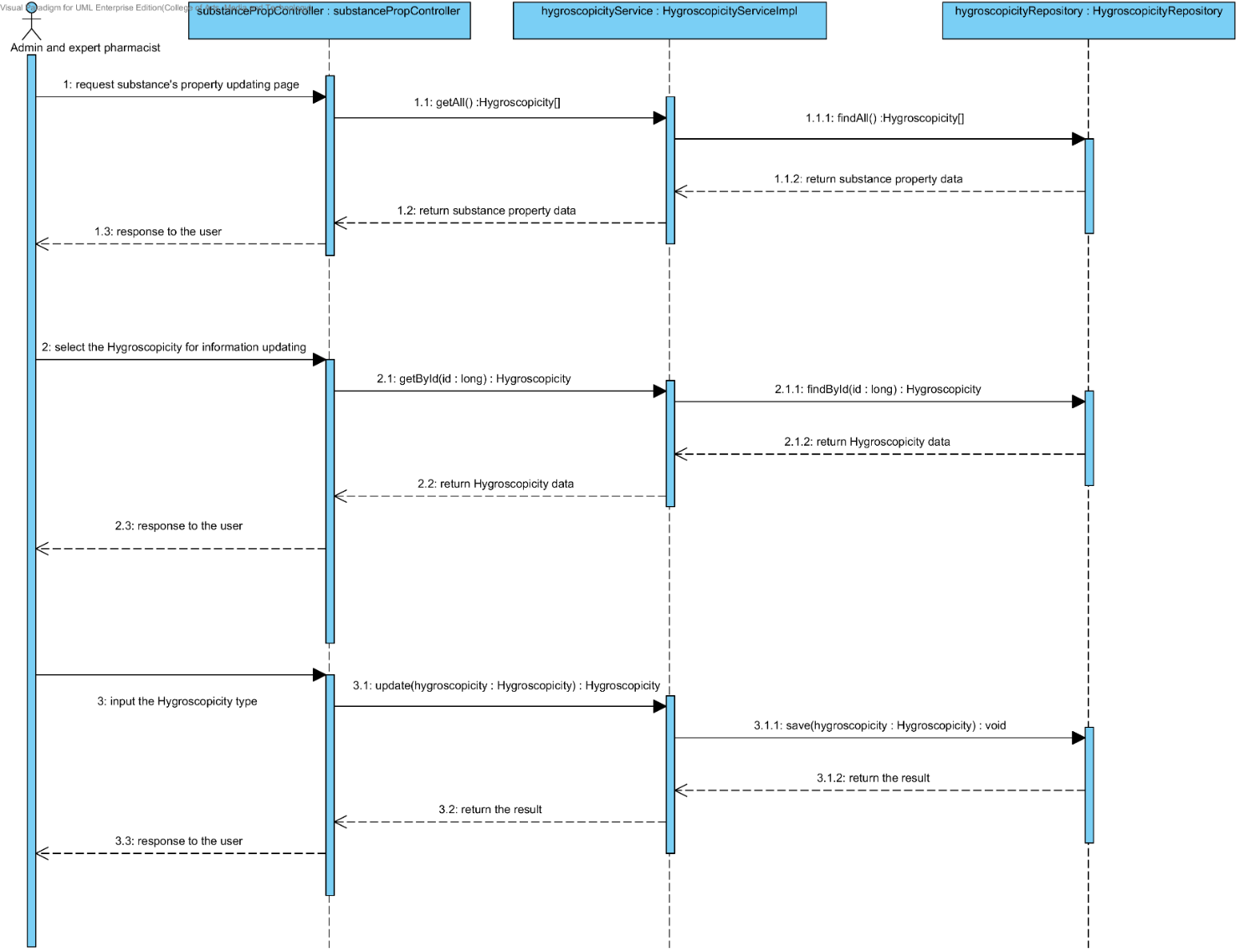
##### 4.1.2.5 SQD-10E: The user updates an existing substance property into the system (PartitionCoefficient).



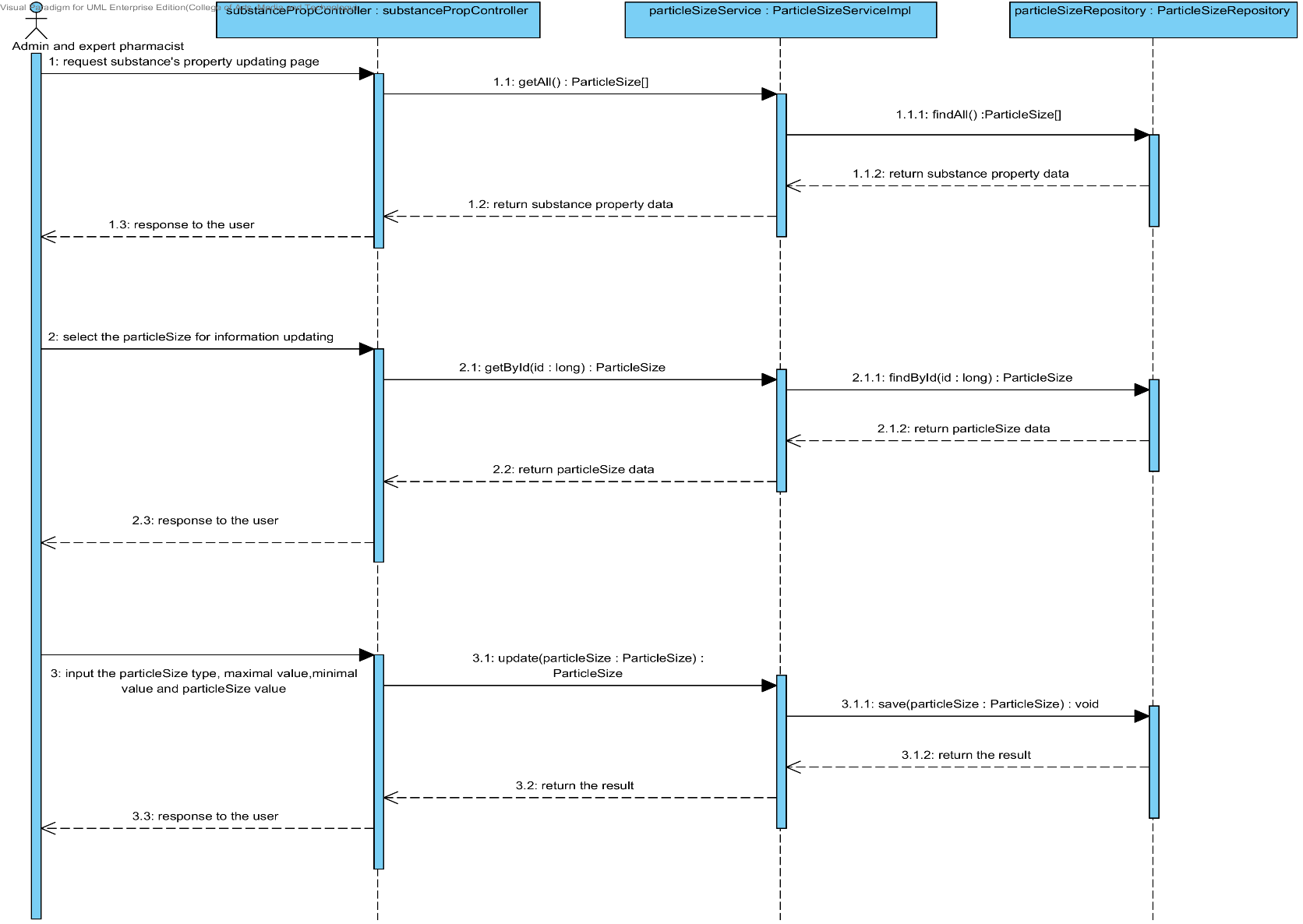
##### 4.1.2.6 SQD-10F: The user updates an existing substance property into the system (Solidstate).



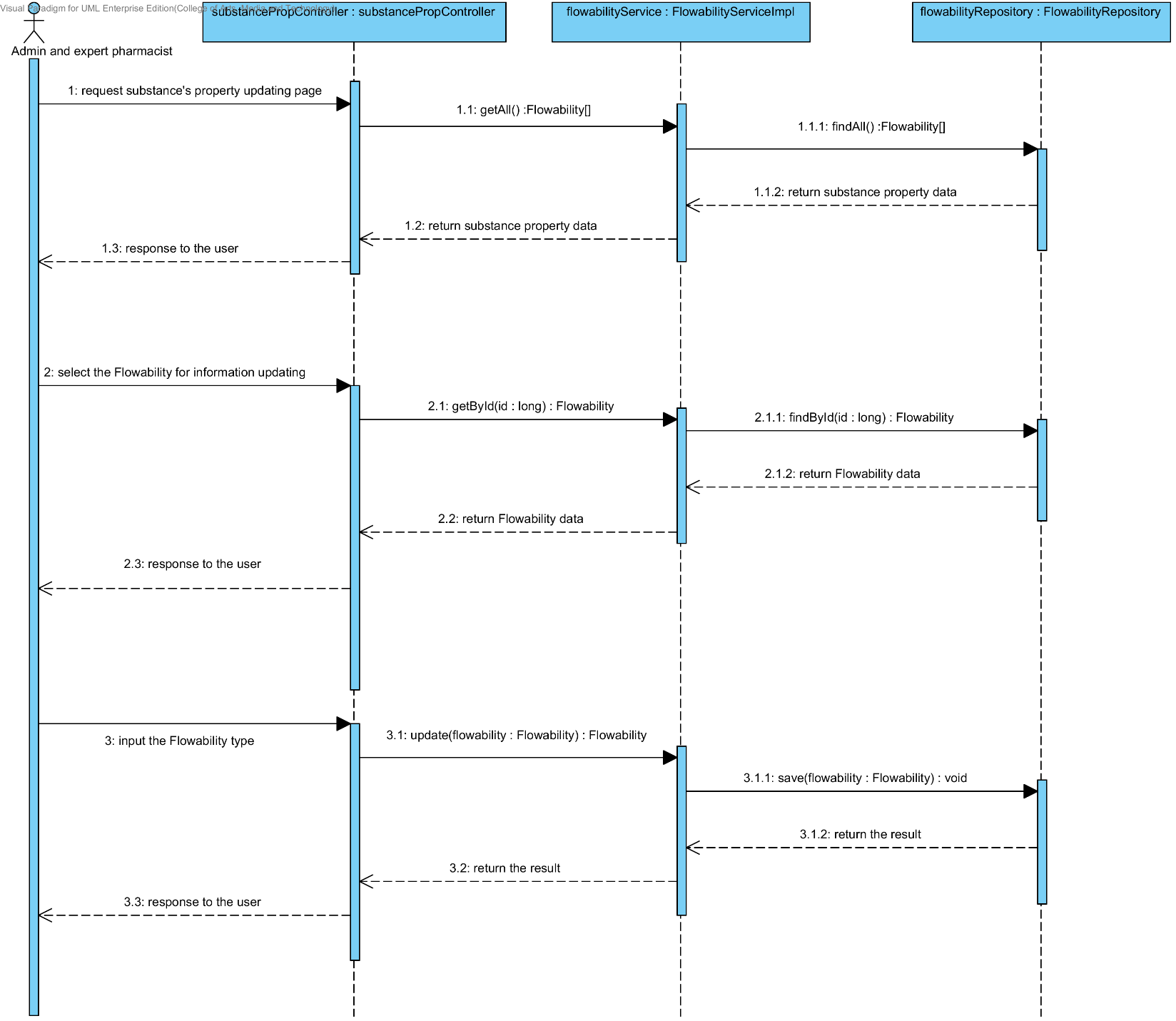
##### 4.1.2.7 SQD-10G: The user updates an existing substance property into the system (Hygroscopicity).



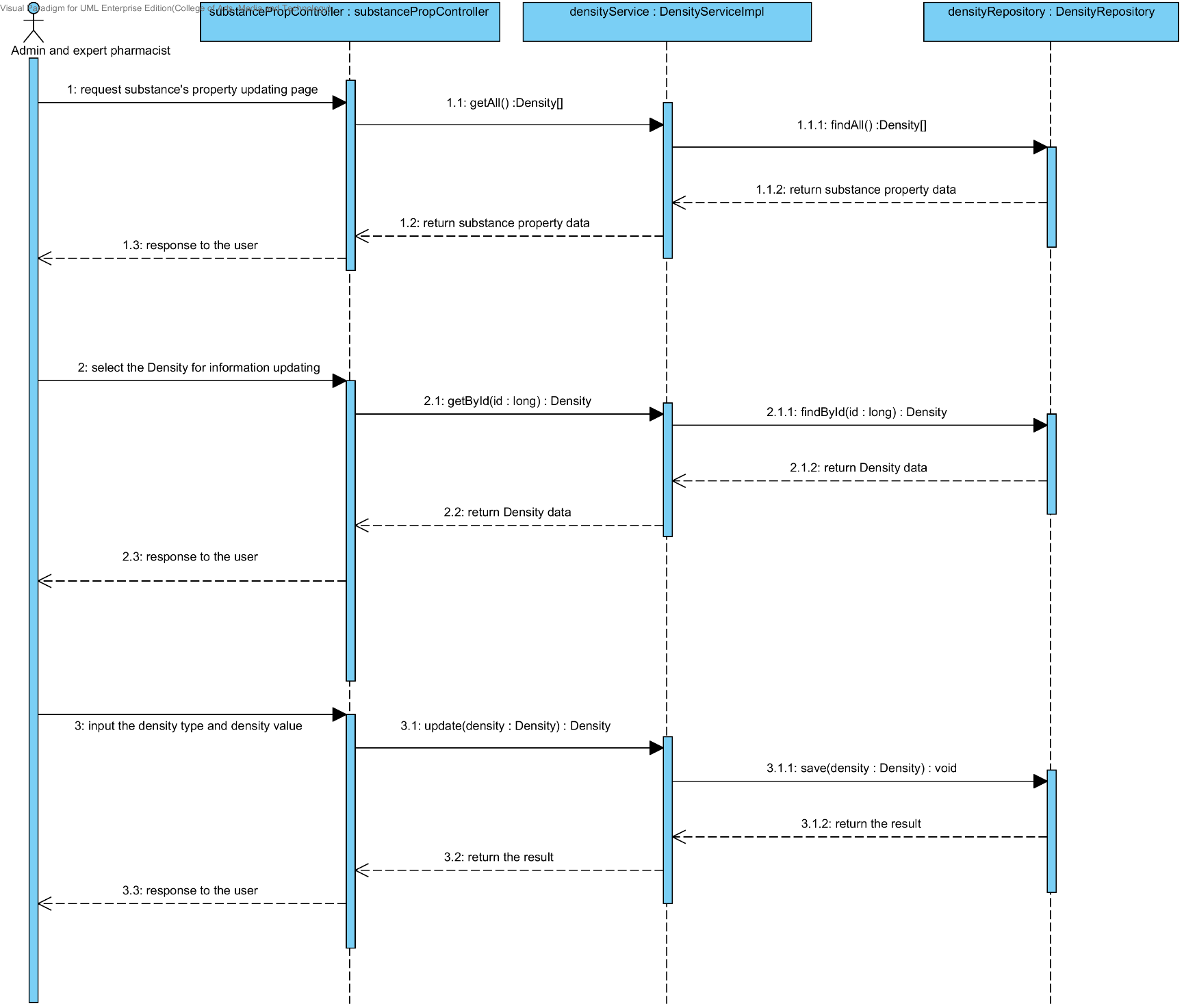
##### 4.1.2.8 SQD-10H: The user updates an existing substance property into the system (ParticleSize).



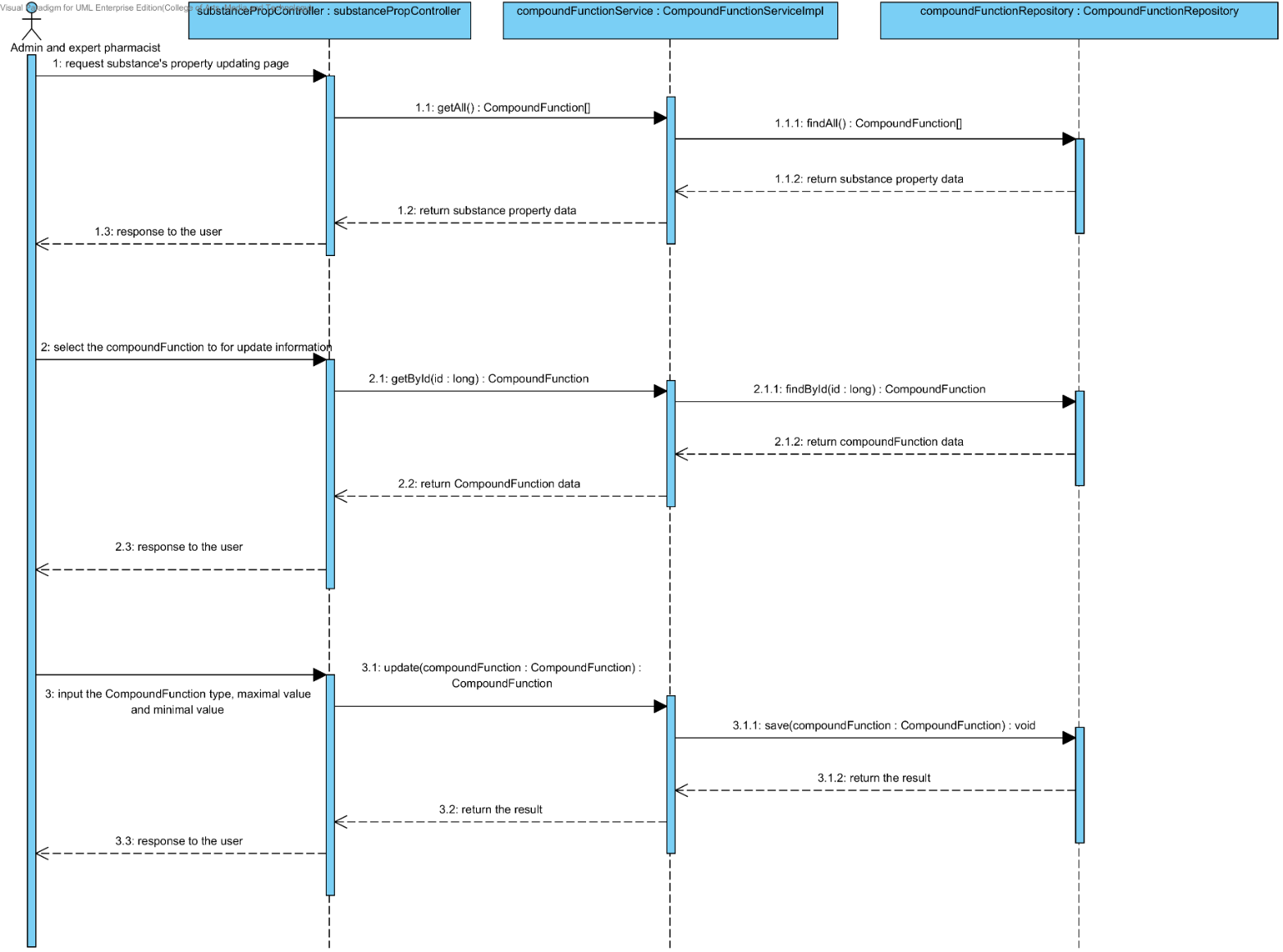
##### 4.1.2.9 SQD-10I: The user updates an existing substance property into the system (Flowability).



##### 4.1.2.10 SQD-10J: The user updates an existing substance property into the system (Density).



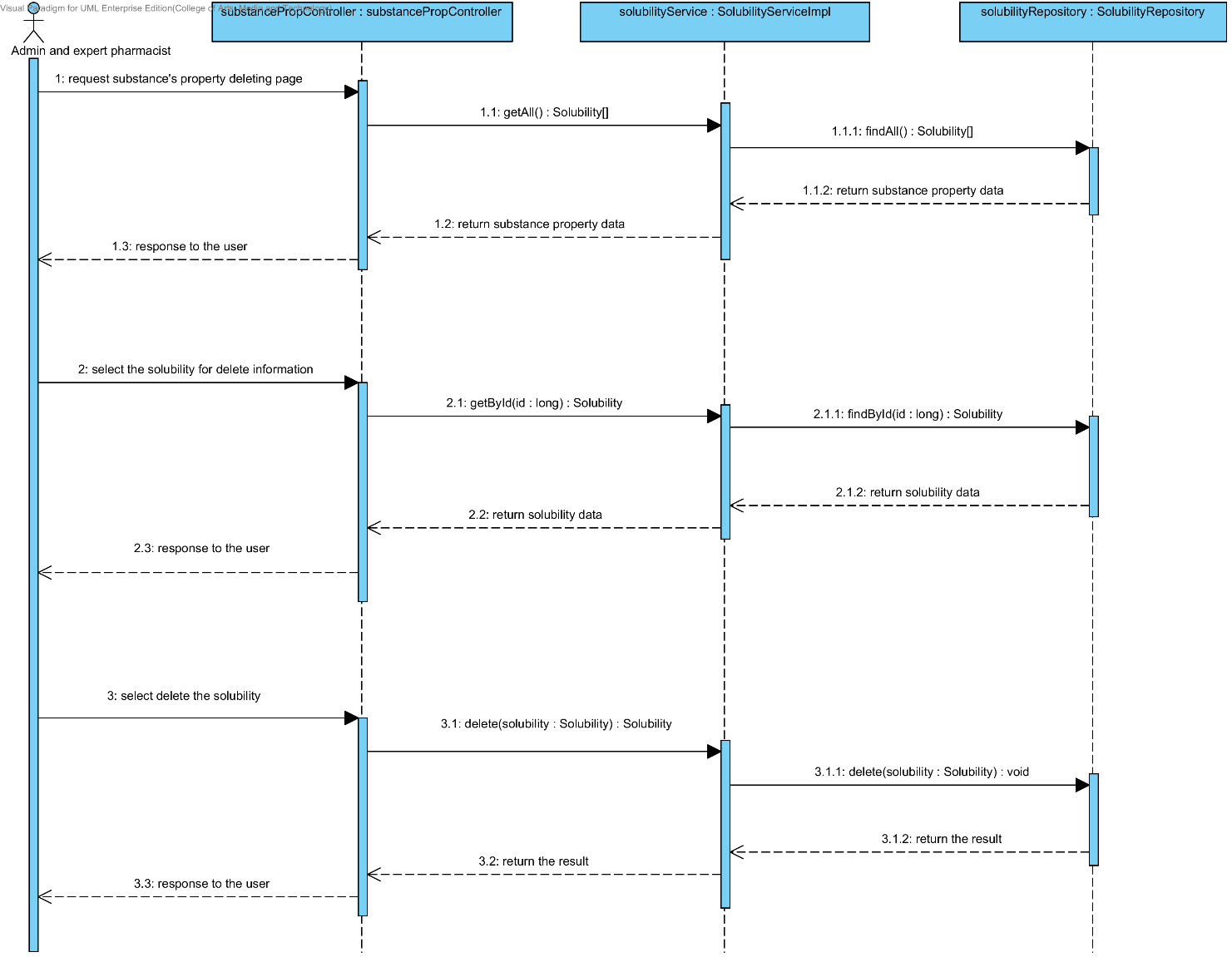
##### 4.1.2.11 SQD-10K: The user updates an existing substance property into the system (Compound function).



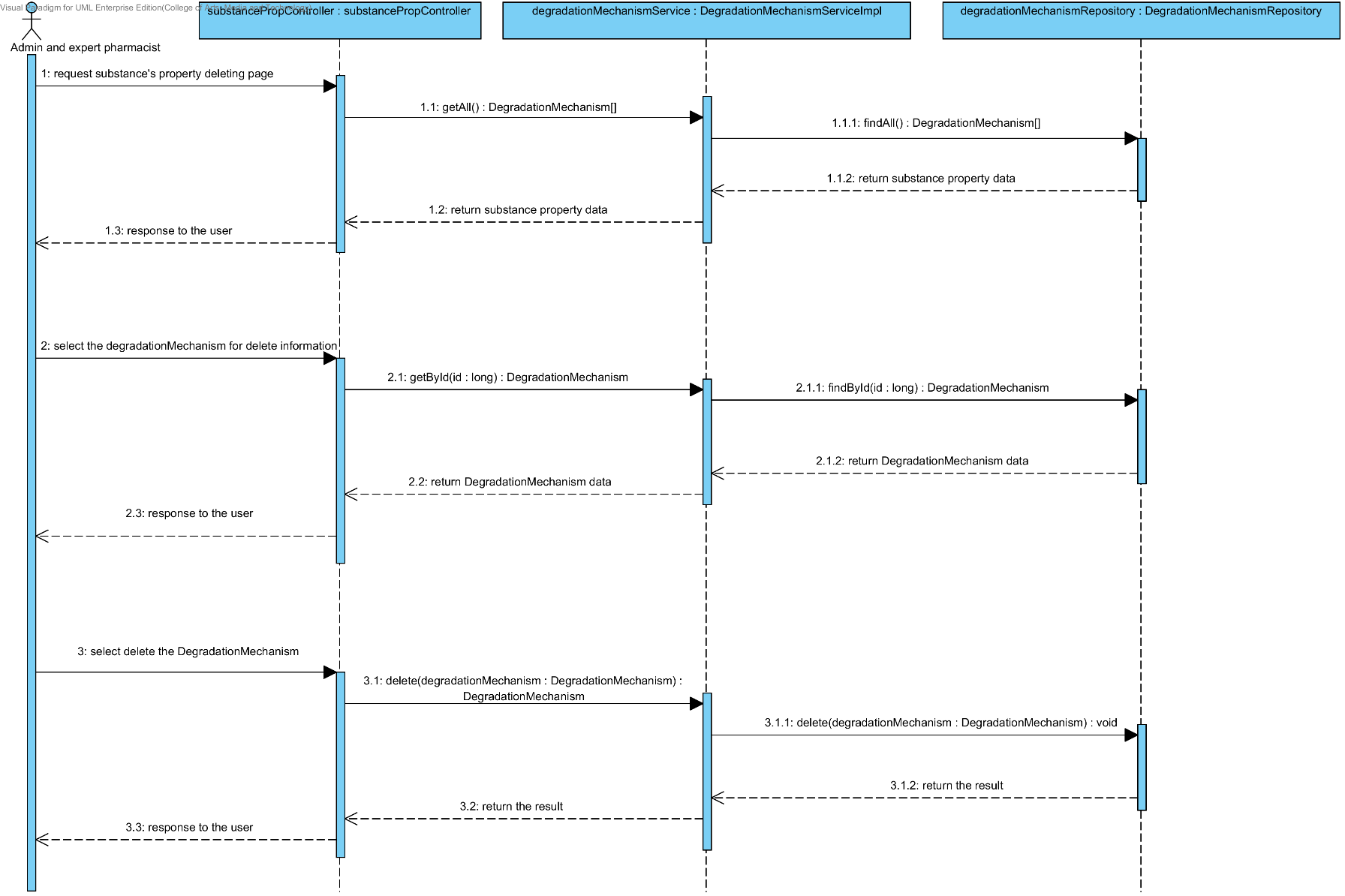
### 4.1.3 URS-11: The user deletes an existing substance property from the system.

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

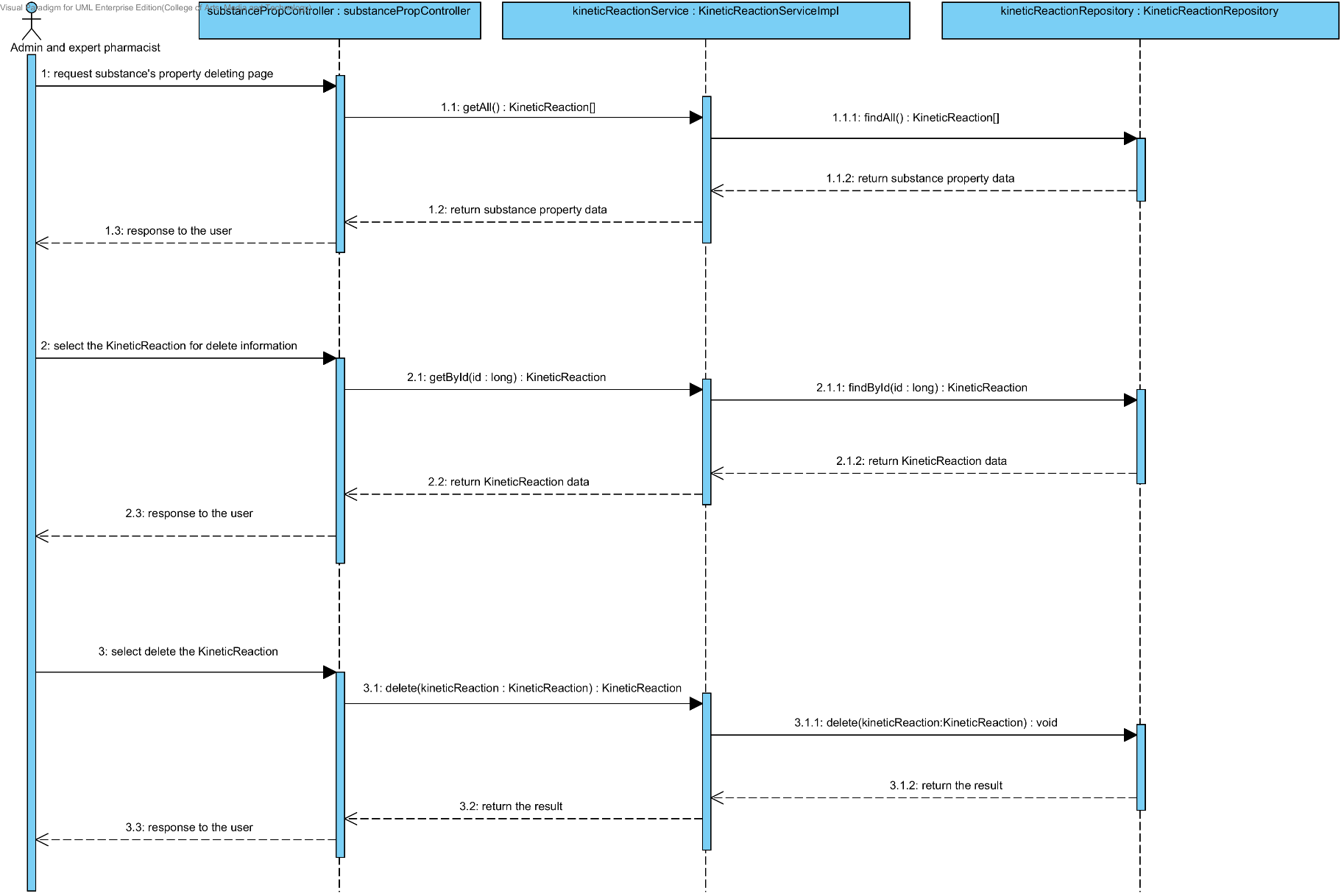
##### 4.1.3.1 SQD-11A: The user deletes an existing substance property from the system (Solubility).



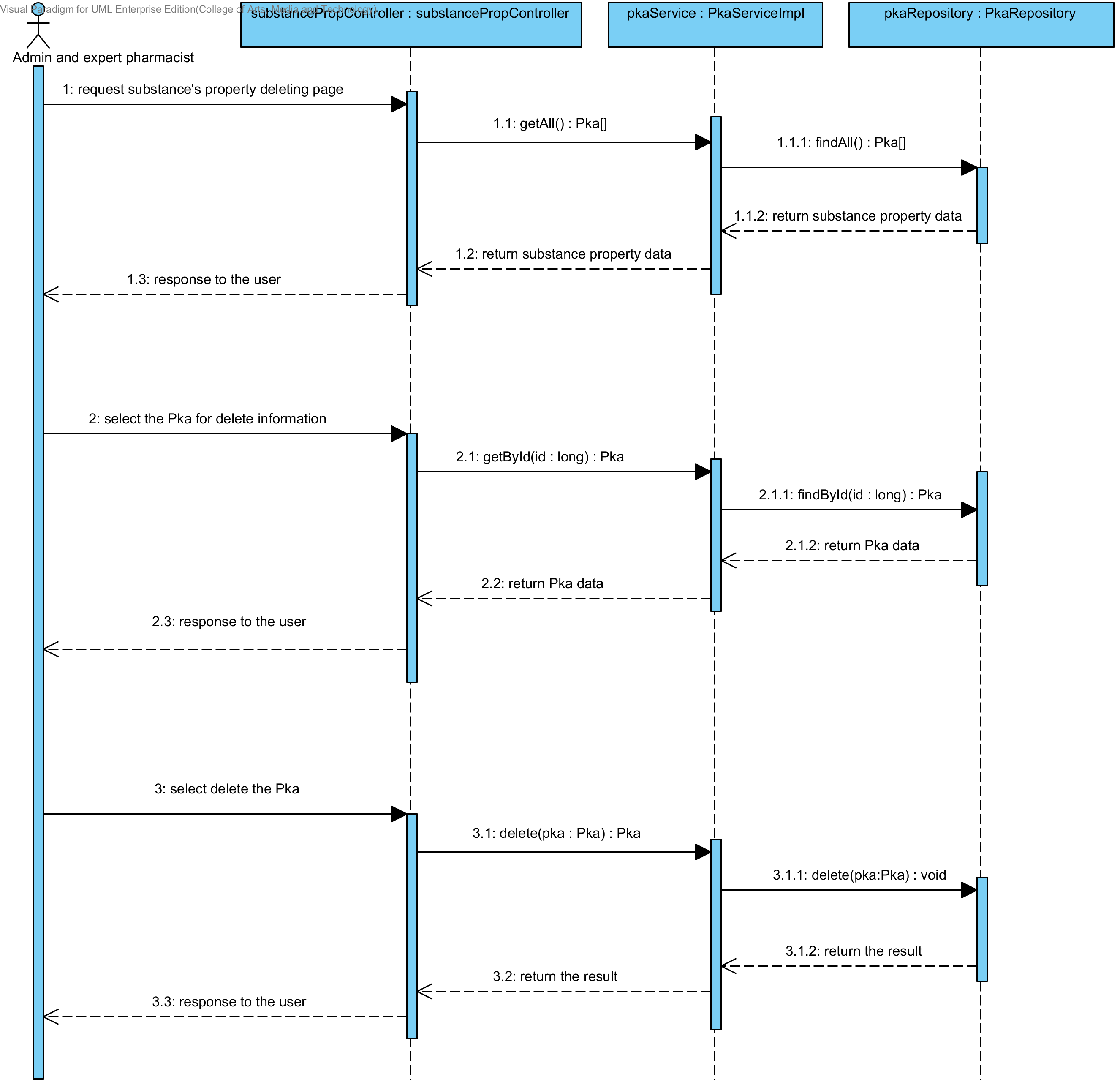
##### 4.1.3.2 SQD-11B: The user deletes an existing substance property from the system (DegradationMechanism).



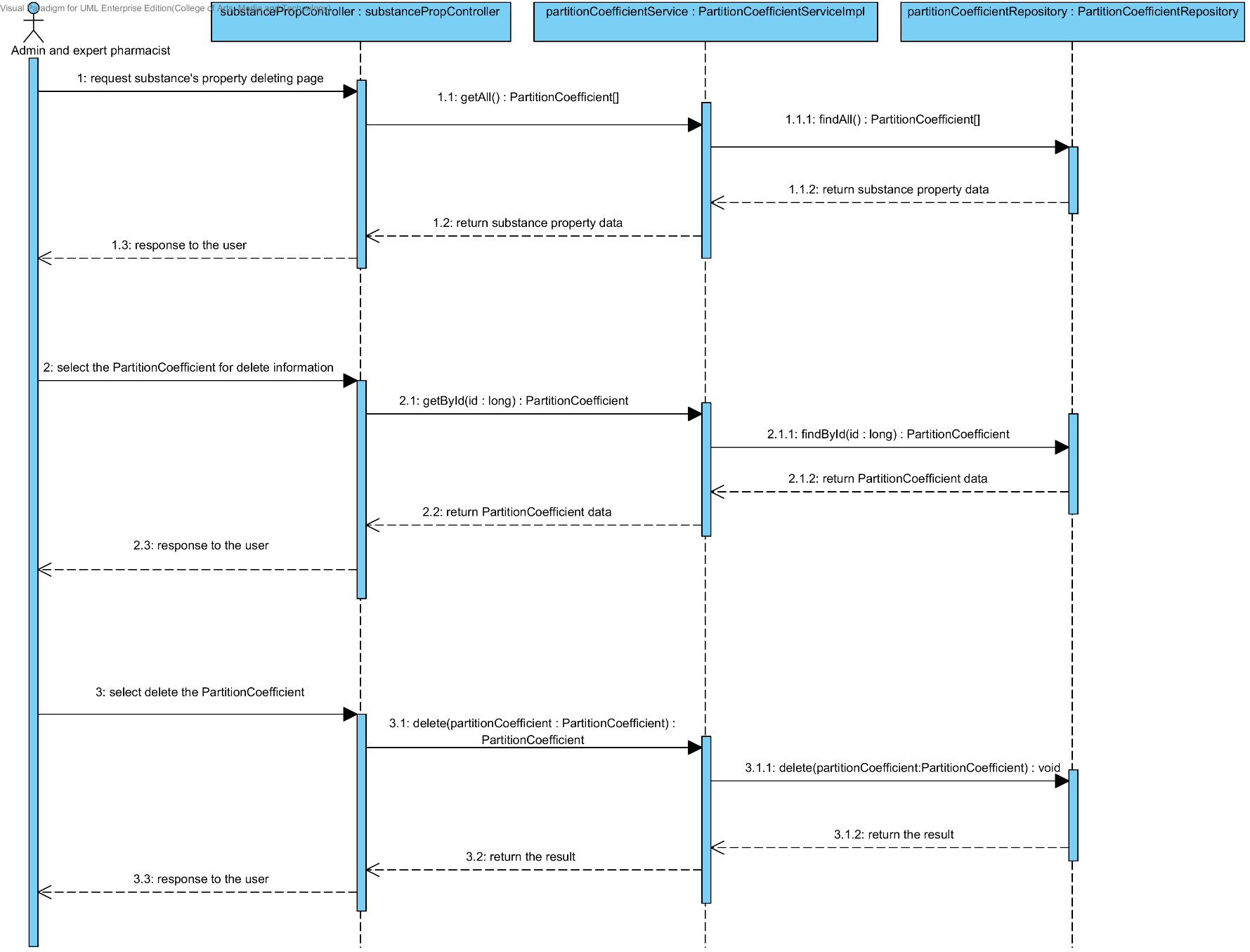
##### 4.1.3.3 SQD-11C: The user deletes an existing substance property into the system (Kinetic Reaction).



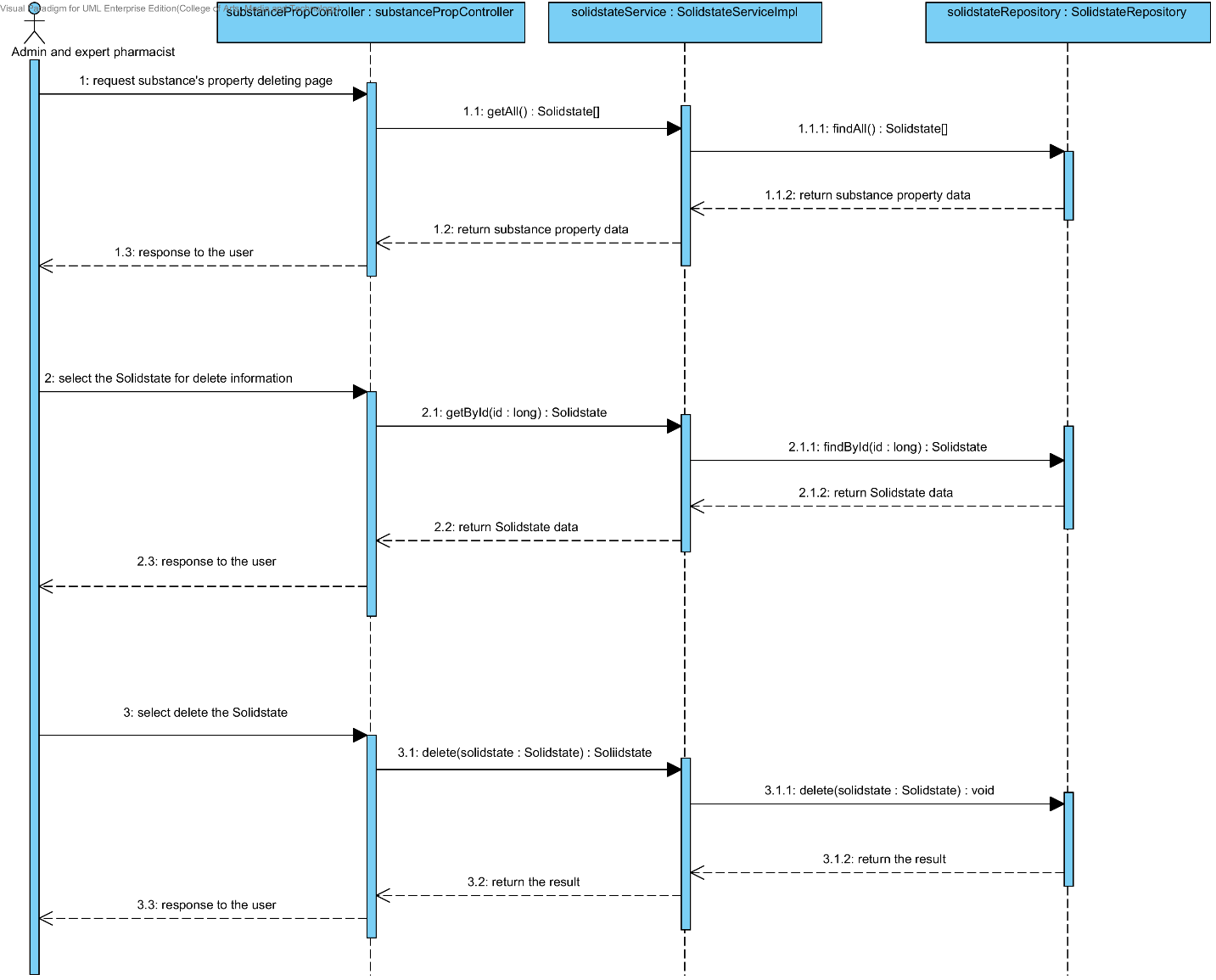
##### 4.1.3.4 SQD-11D: The user deletes an existing substance property from the system (Pka).



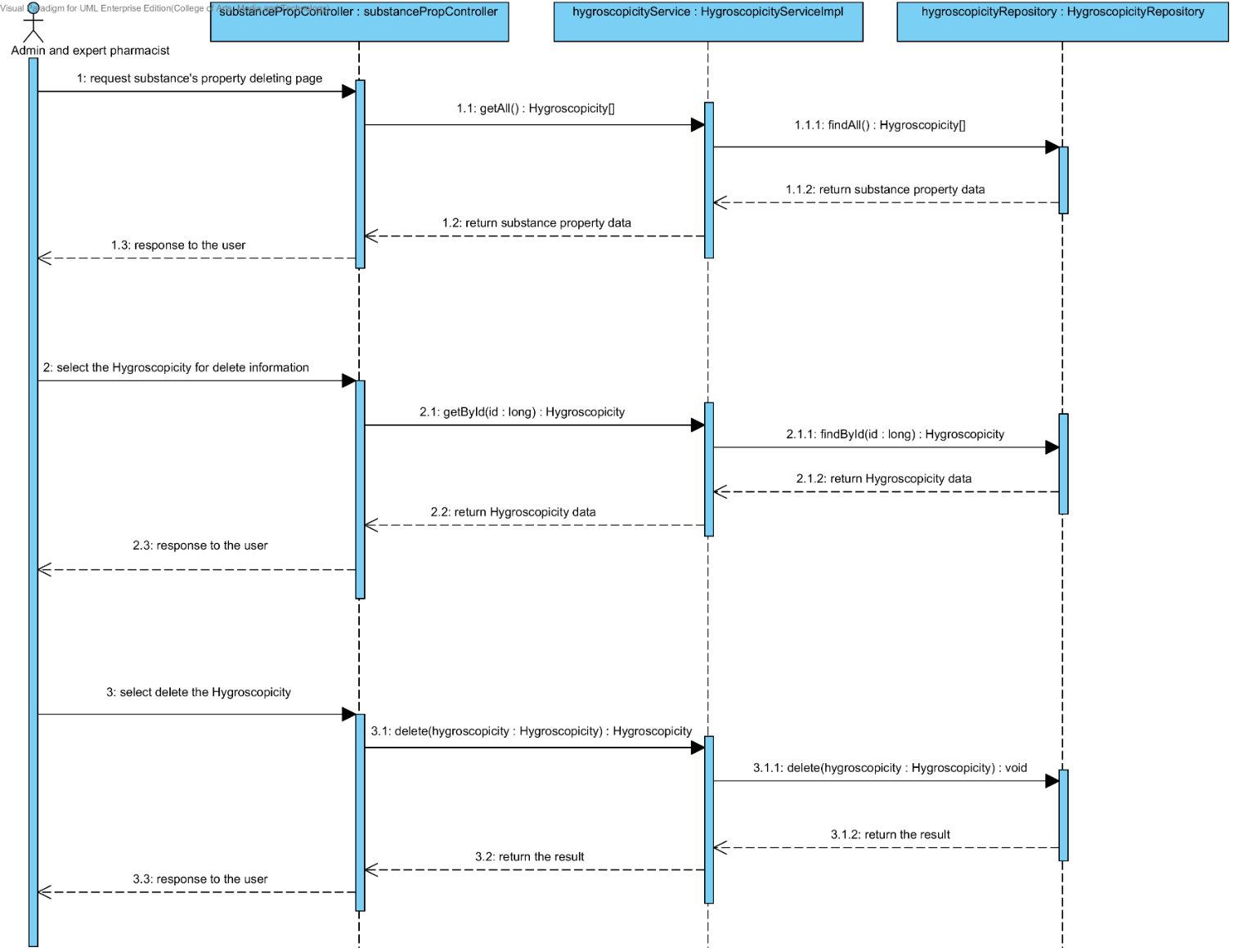
##### 4.1.3.5 SQD-11E: The user deletes an existing substance property from the system (PartitionCoefficient).



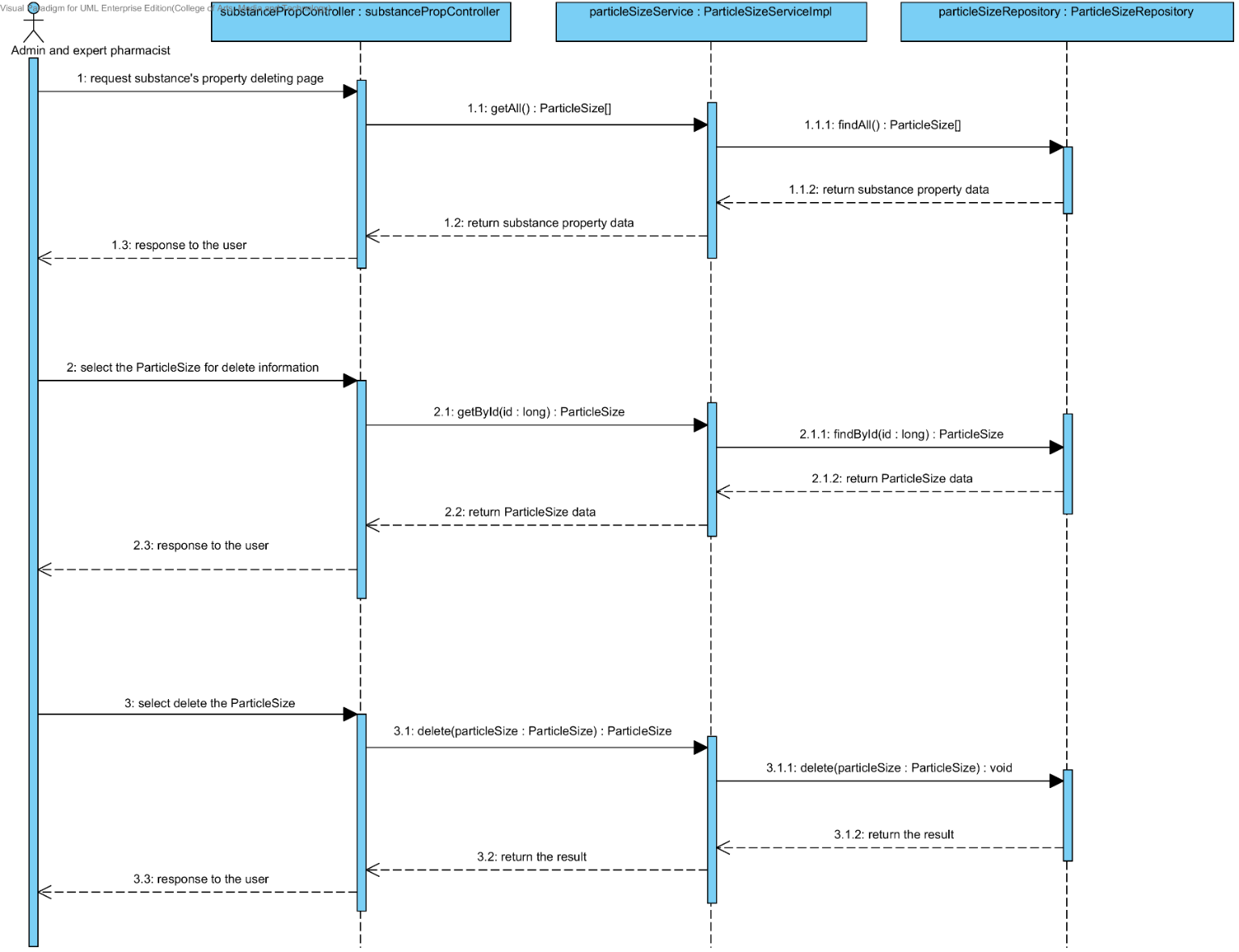
##### 4.1.3.6 SQD-11F: The user deletes an existing drug substance property from the system (Solidstate).



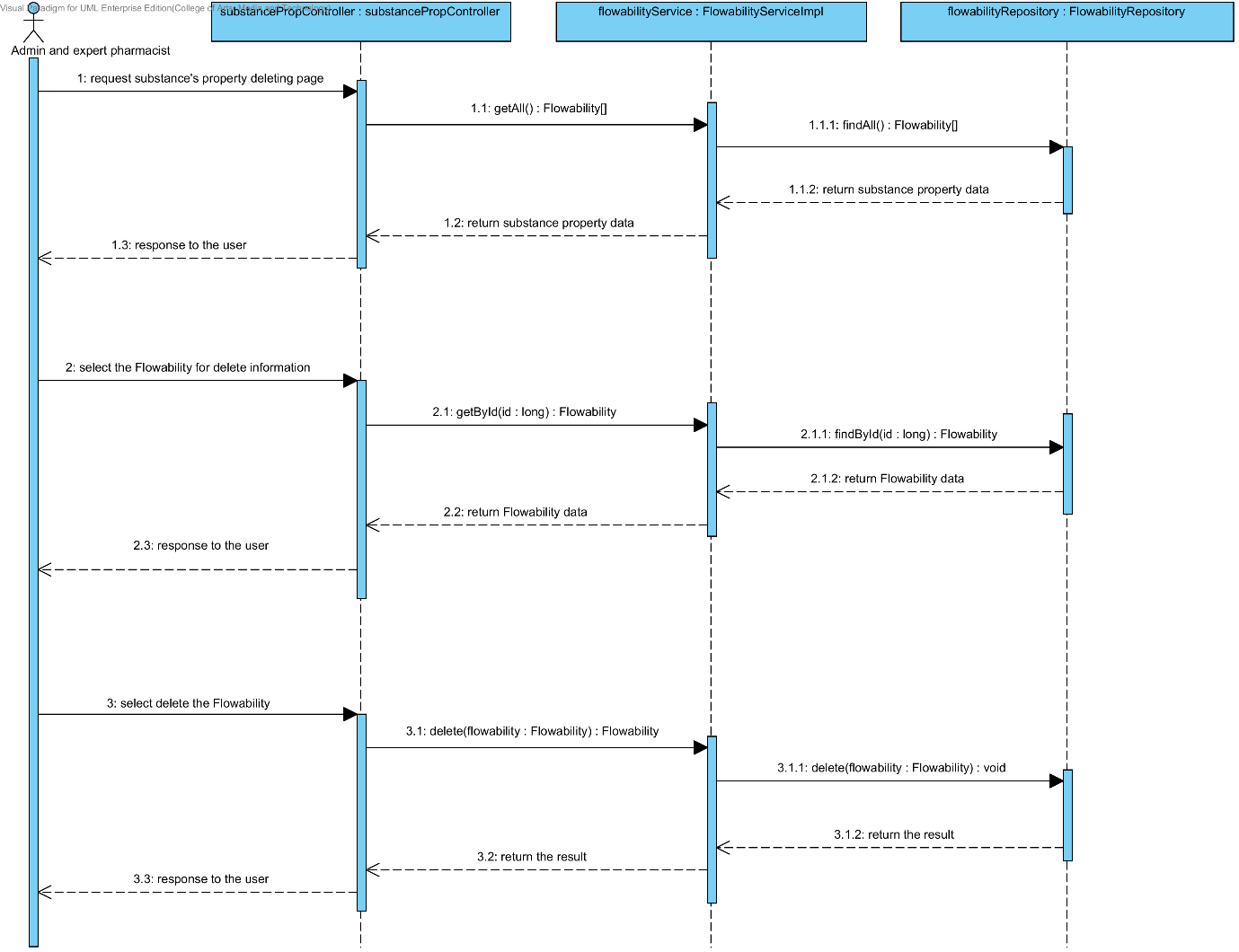
##### 4.1.3.7 SQD-11G: The user deletes an existing substance property from the system (Hygroscopicity).



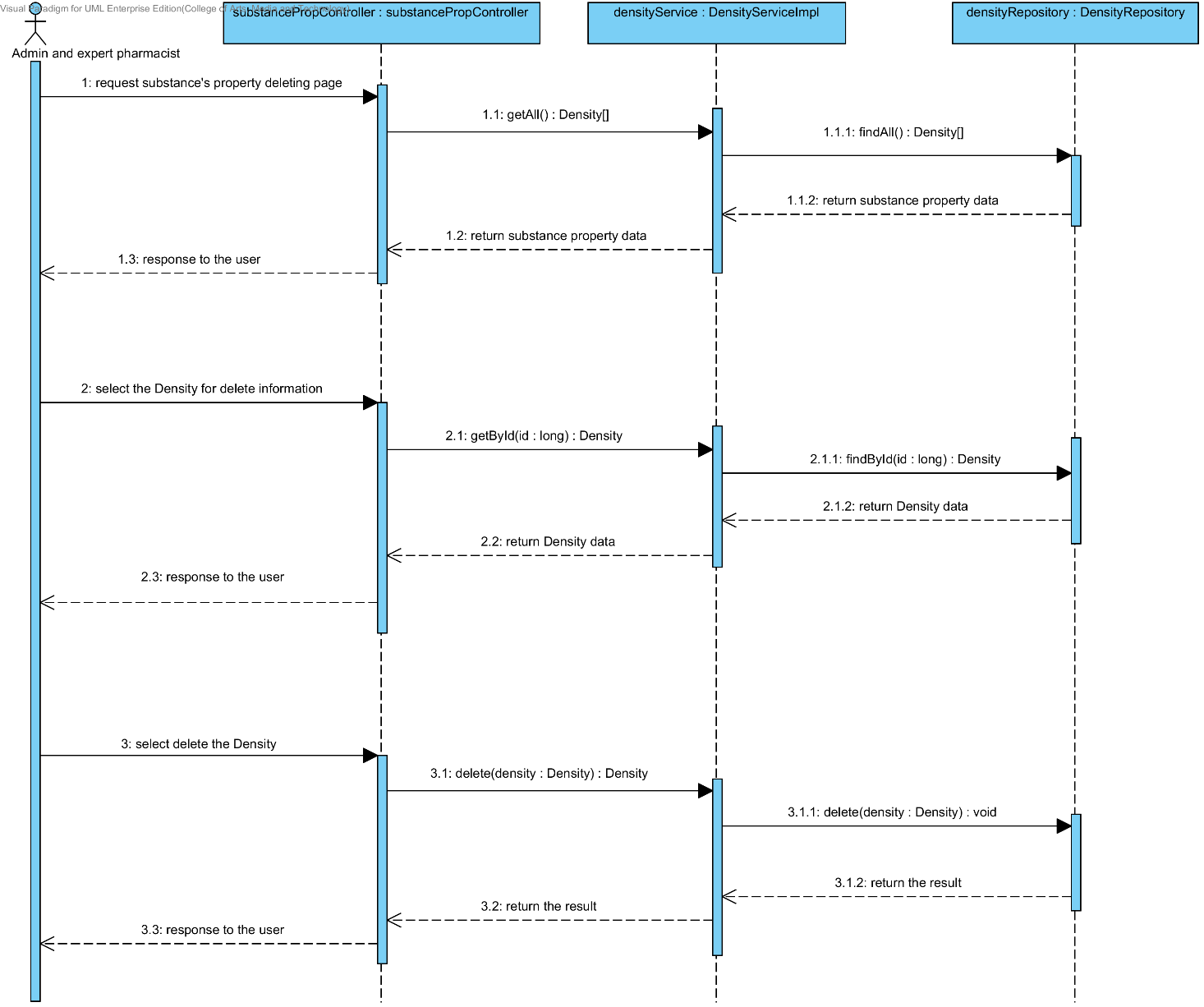
##### 4.1.3.8 SQD-11H: The user deletes an existing substance property from the system (ParticleSize).



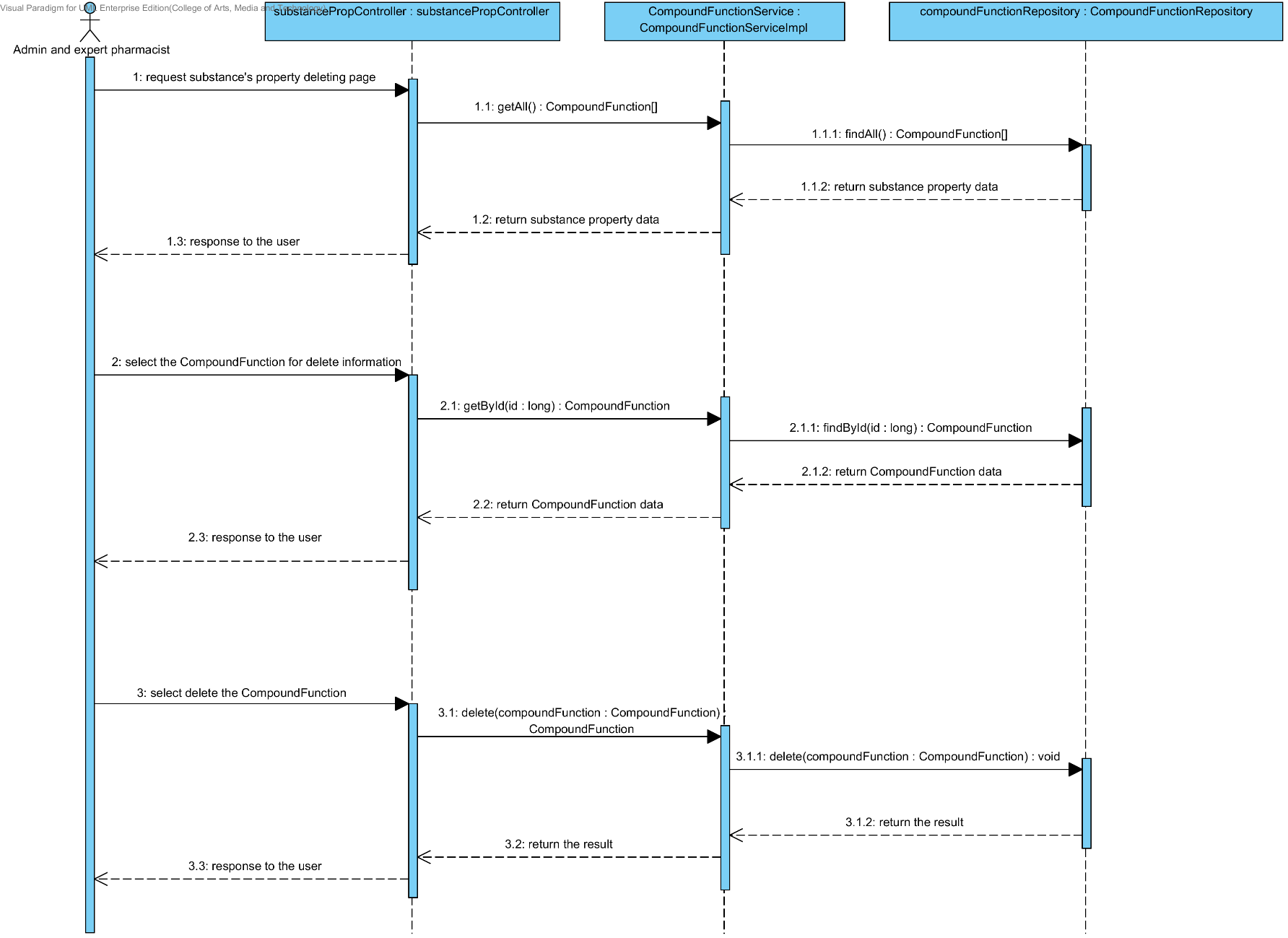
##### 4.1.3.9 SQD-11I: The user deletes an existing substance property from the system (Flowability).



##### 4.1.3.10 SQD-11J: The user deletes an existing substance property from the system (Density).



##### 4.1.3.11 SQD-11K: The user deletes an existing substance property from the system (Compound function).

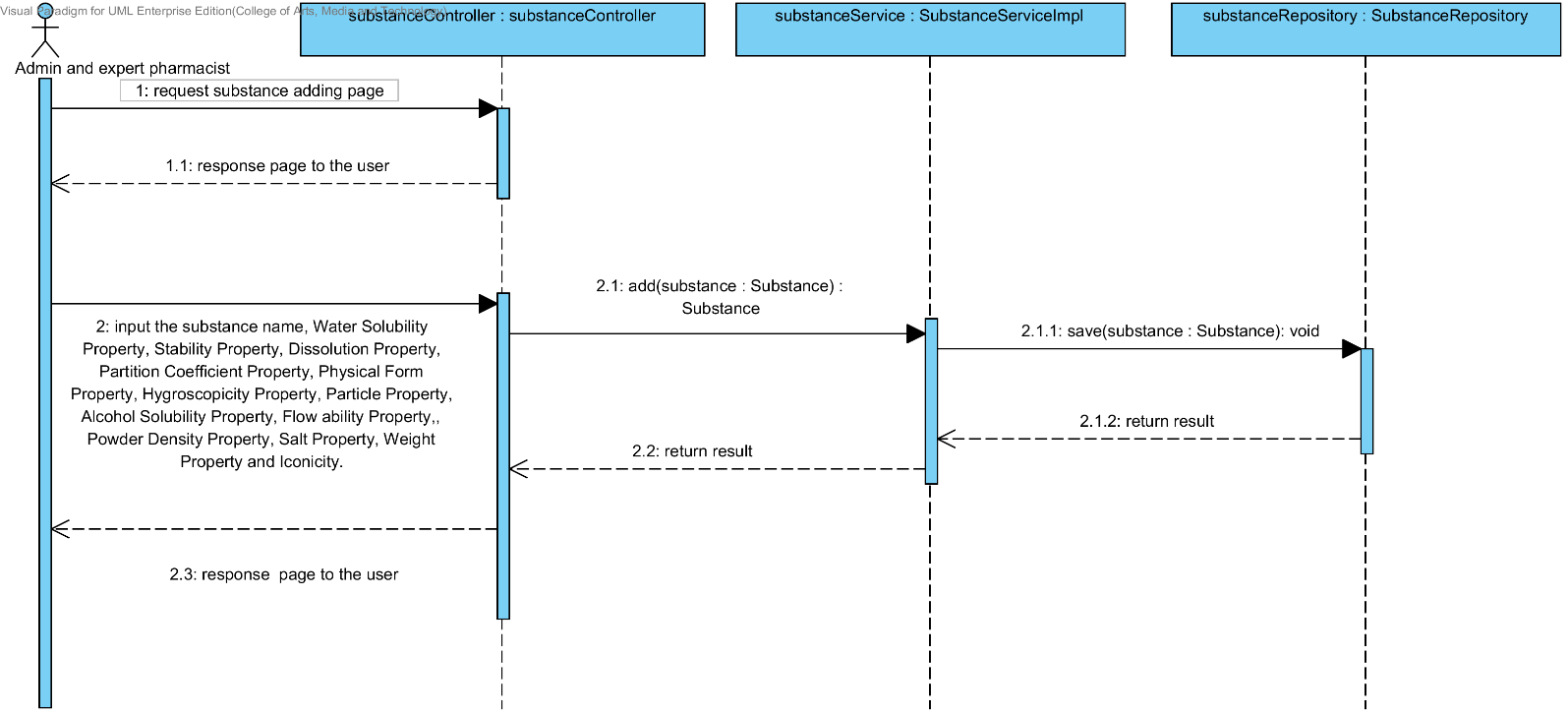


## 4.2- Sub-Feature 6: Manage the drug substance

### 4.2.1 URS-12: 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, substance property, weight property and lonicity. 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.

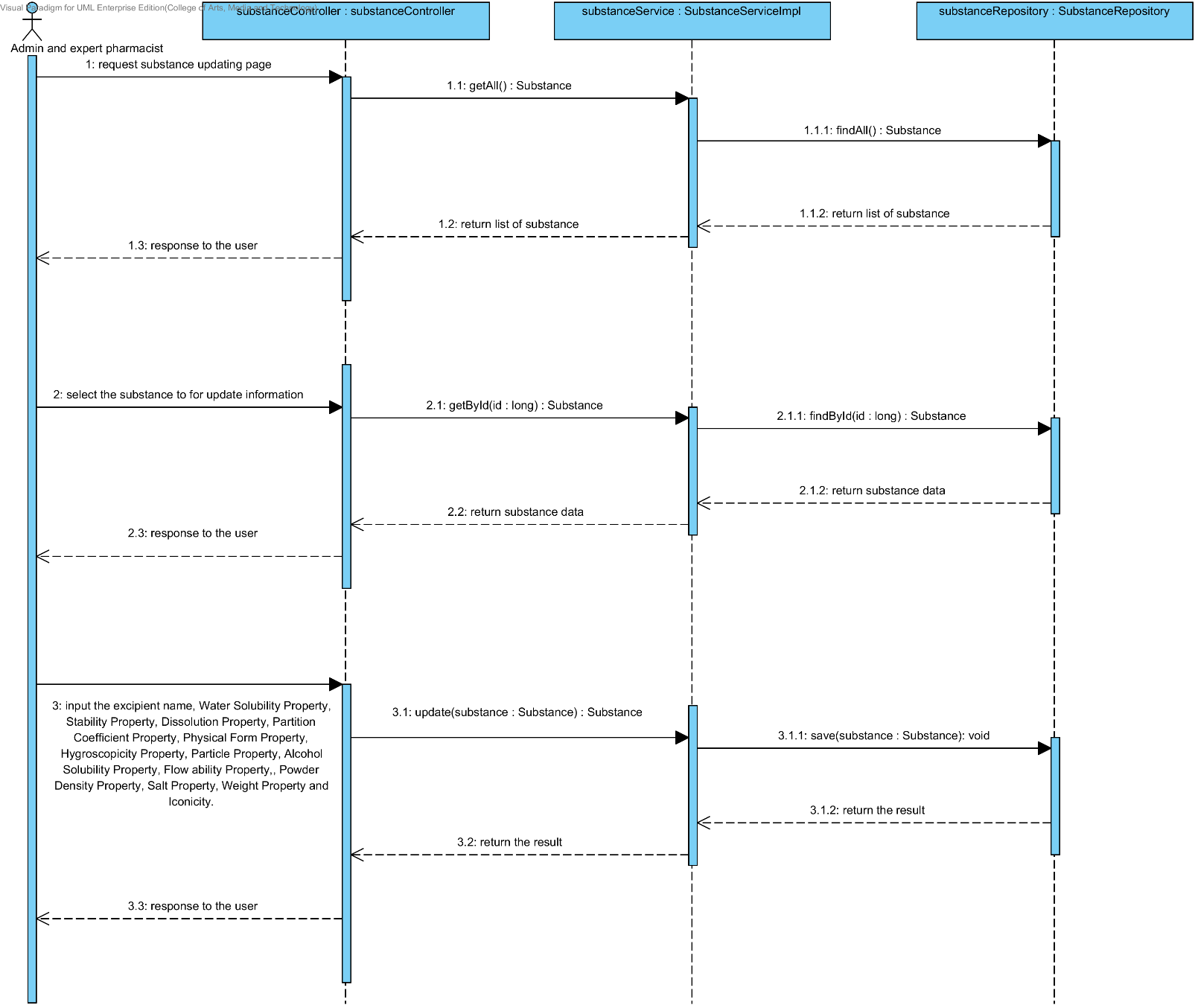
##### 4.2.1.1 SQD-12: The user adds a new substance to the system.



### 4.2.2 URS-13: 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 substance property, weight property and lonicity. 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 adding successful page.

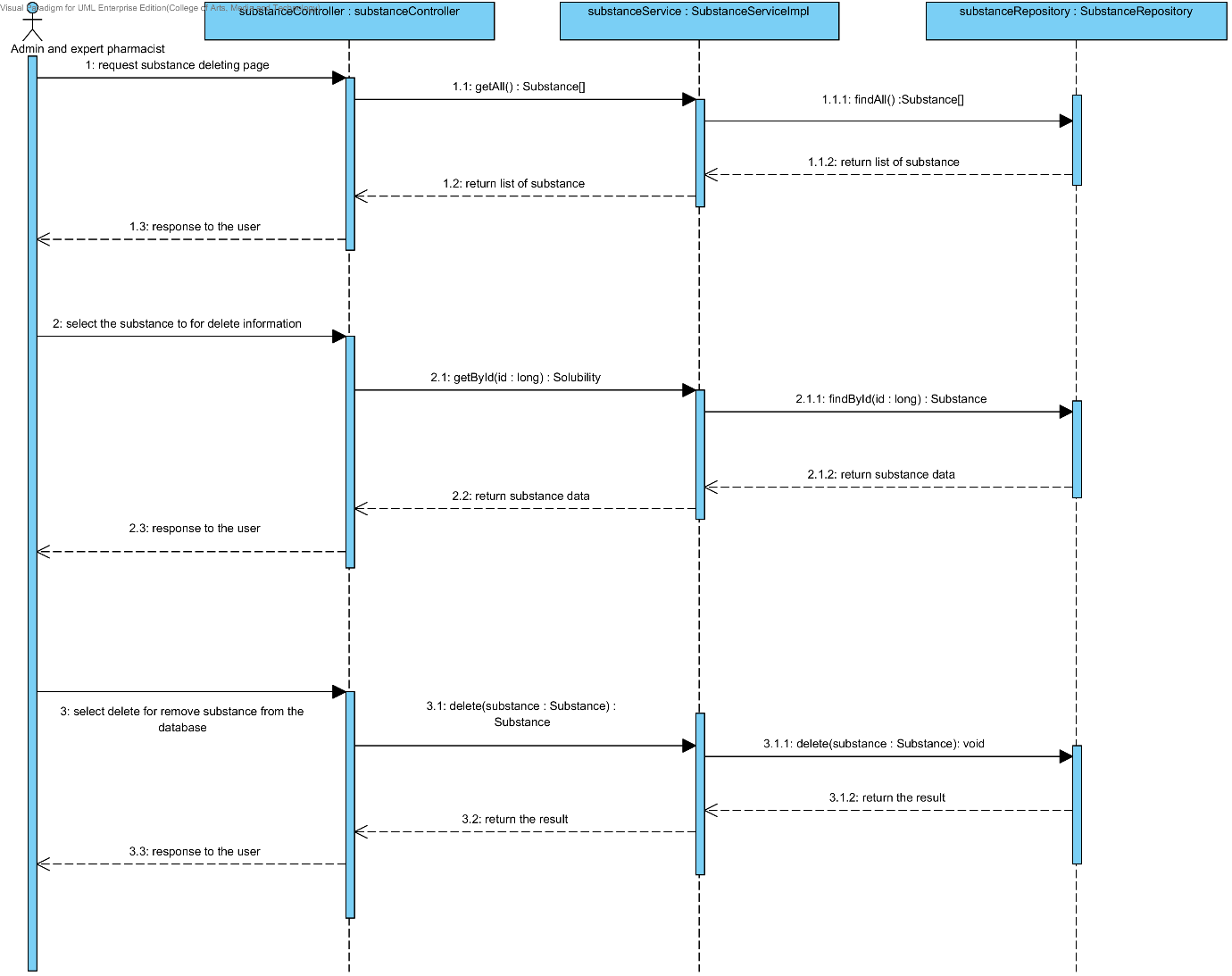
##### 4.2.2.1 SQD-13: The user updates an existing substance in the system.



### 4.2.3 URS-14: 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.

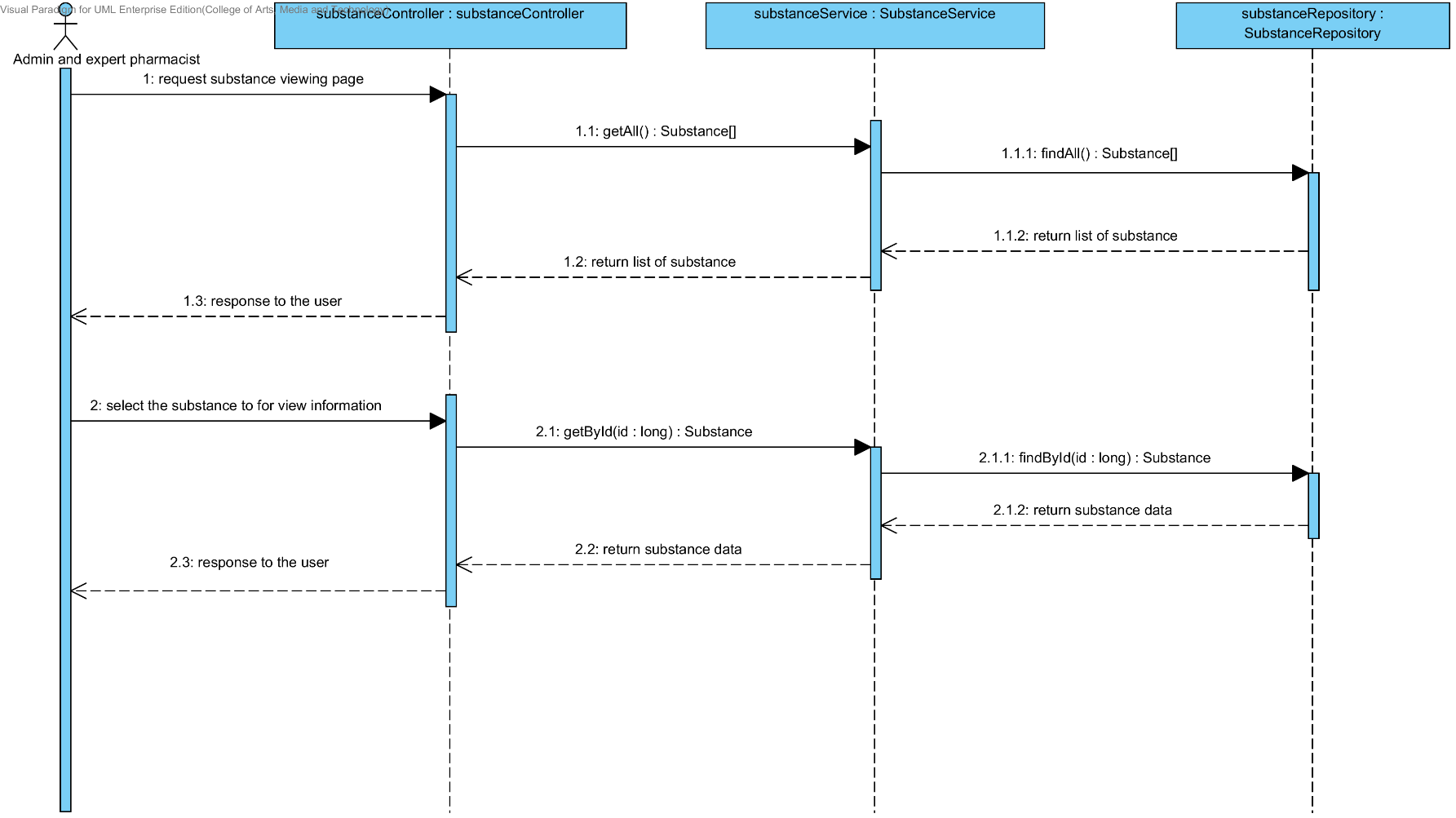
##### 4.2.3.1 SQD-14: The user deletes an existing substance from the system.



### 4.2.4 URS-15: 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.

##### 4.2.4.1 SQD-15: The user views the substance in the system.

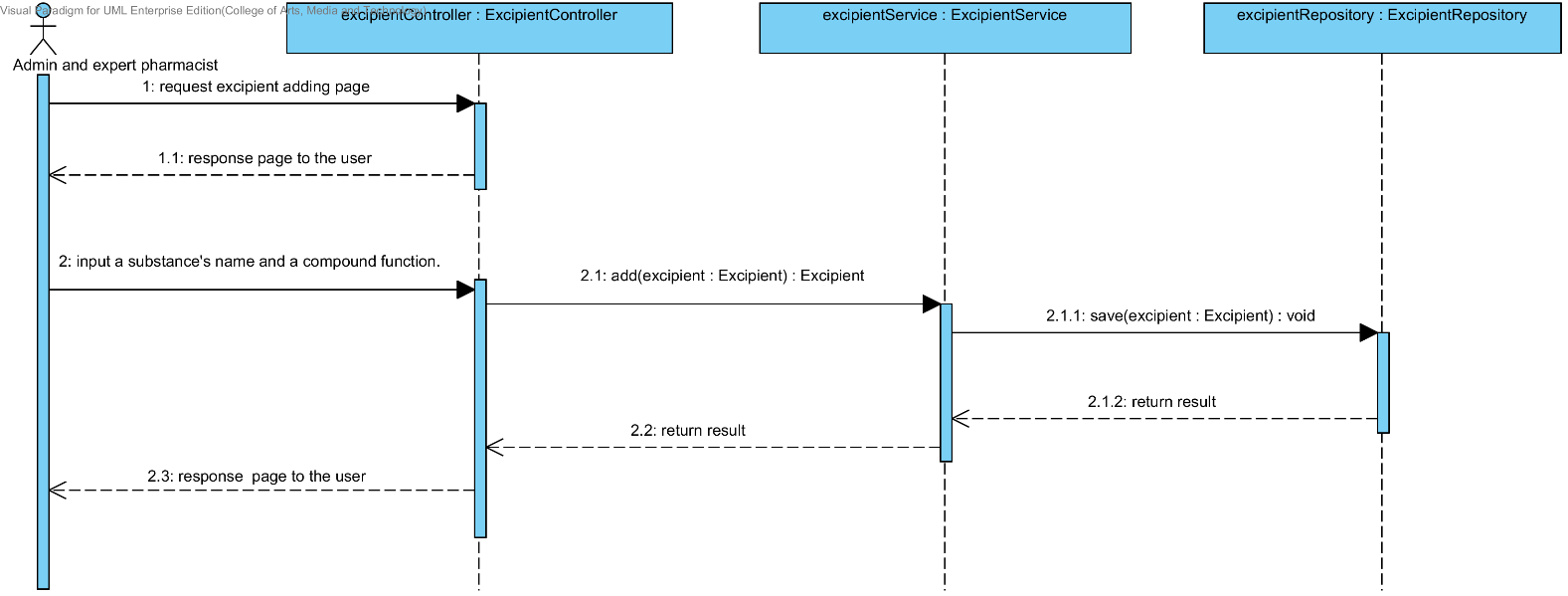


## 4.3- Sub-Feature 7: Manage the drug substance

### 4.3.1 URS-16: 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 property and compound function. 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.

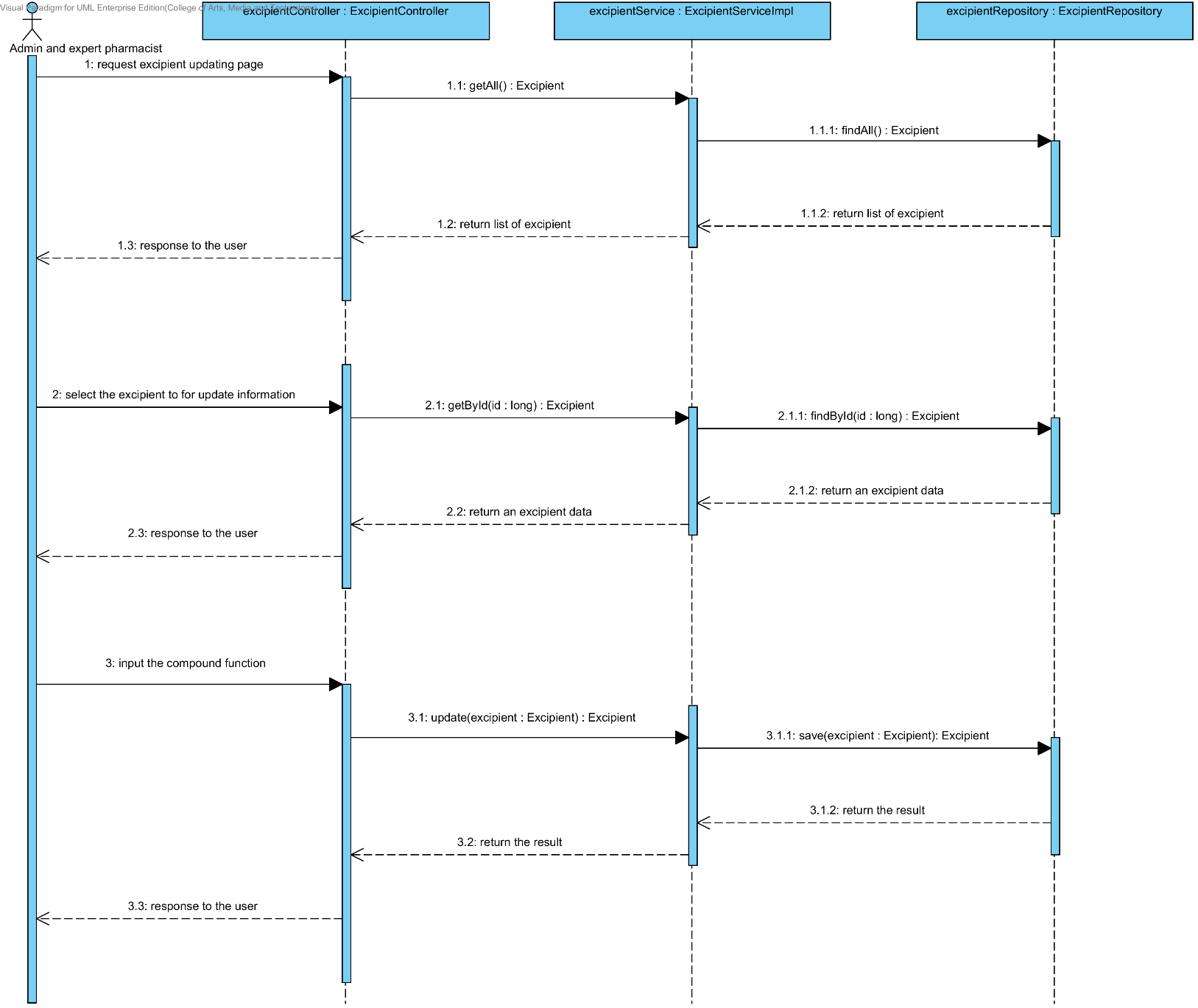
##### 4.3.1.1 SQD-16: The user adds a new excipient to the system.



### 4.3.2 URS-17: 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 property and compound 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.

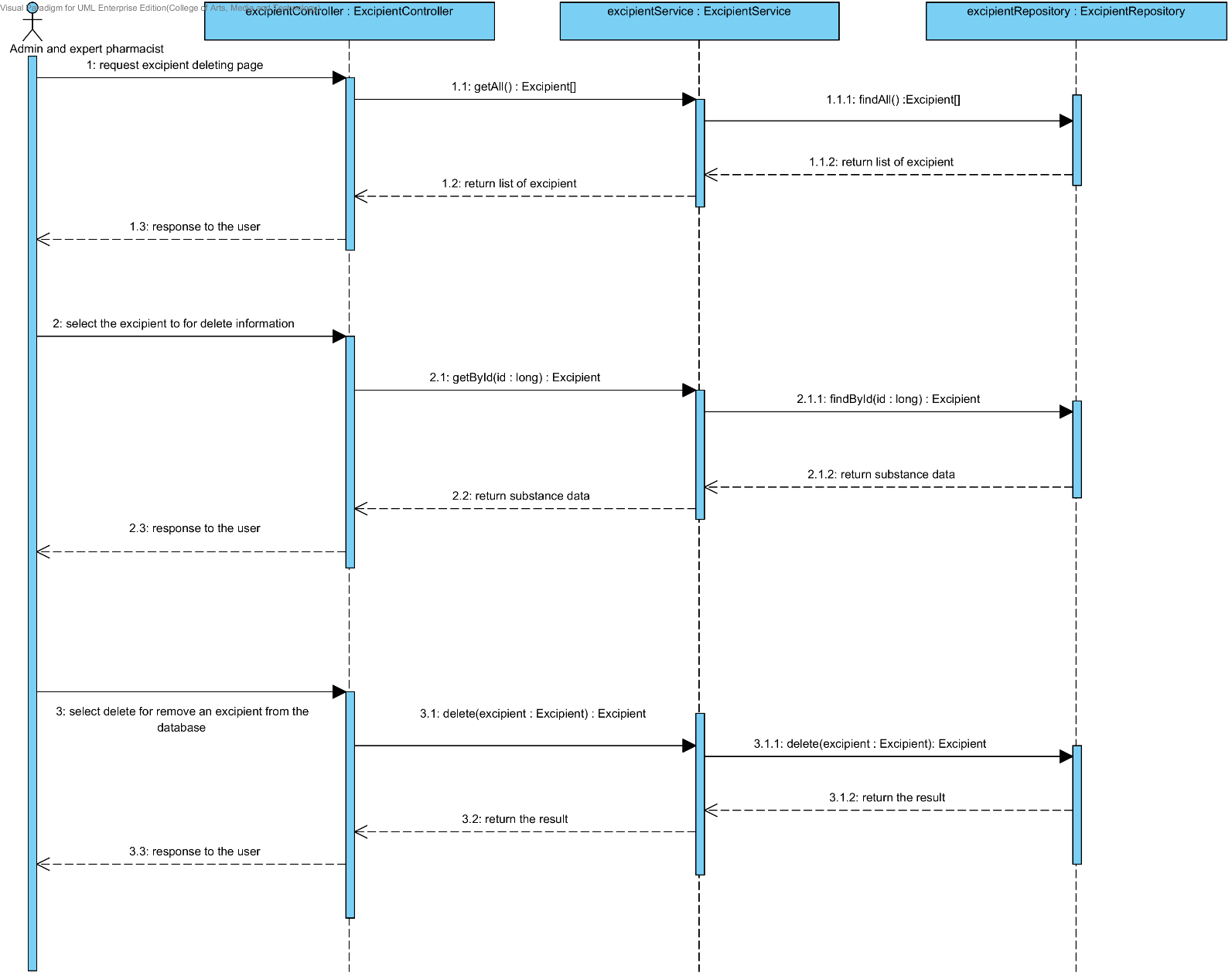
##### 4.3.2.1 SQD-17: The user updates an existing excipient in the system



### 4.3.3 URS-18: 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.

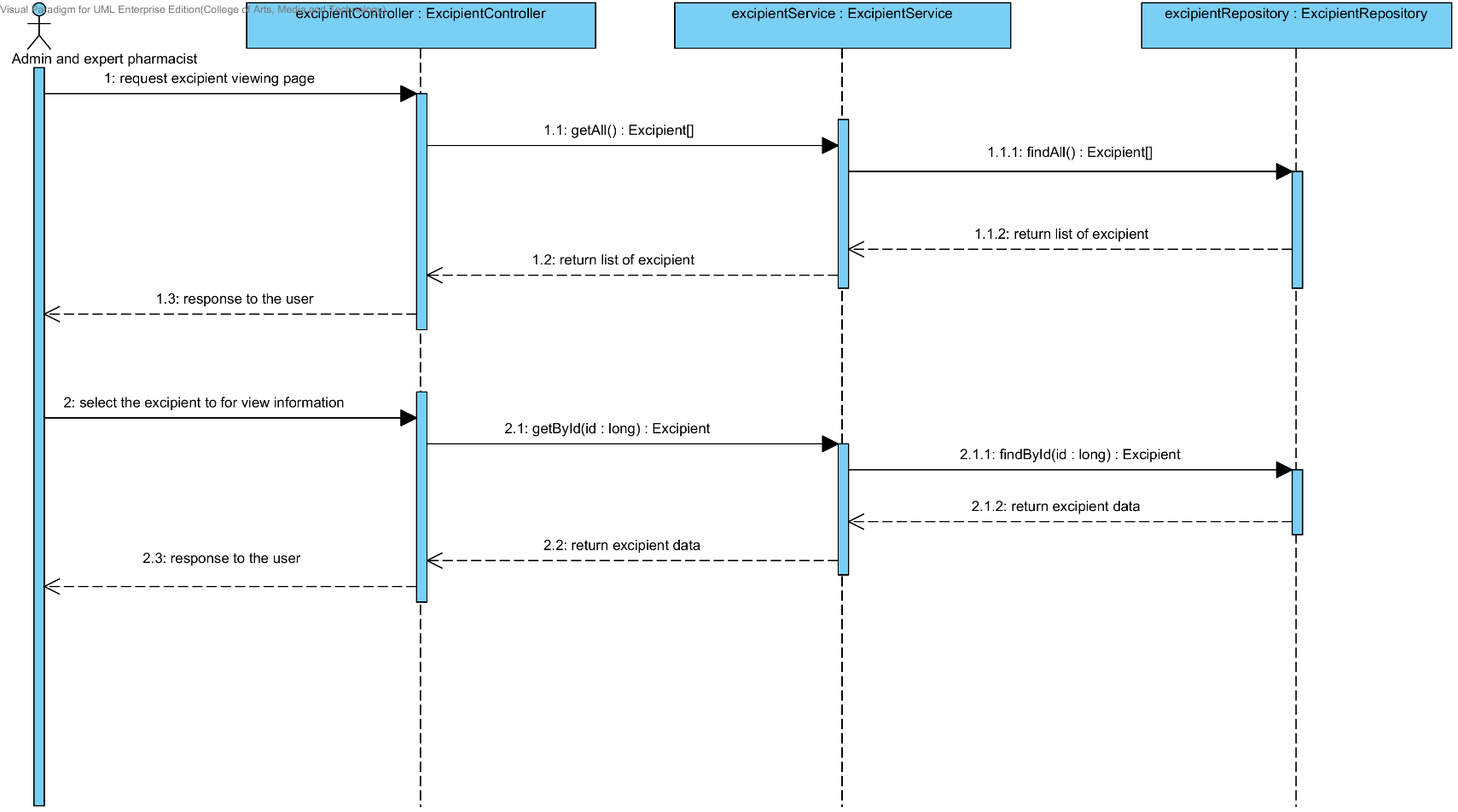
##### 4.3.3.1 SQD-18: The user deletes an existing excipient from the system.



### 4.3.4 URS-19: 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.

##### 4.3.4.1 SQD-19: The user views the excipient in the system.



## 4.4- Sub-Feature 8: Manage the drug formulation

### 4.4.1 URS-20: 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 data such as name, excipient, substance quantity and substance intensity. 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.

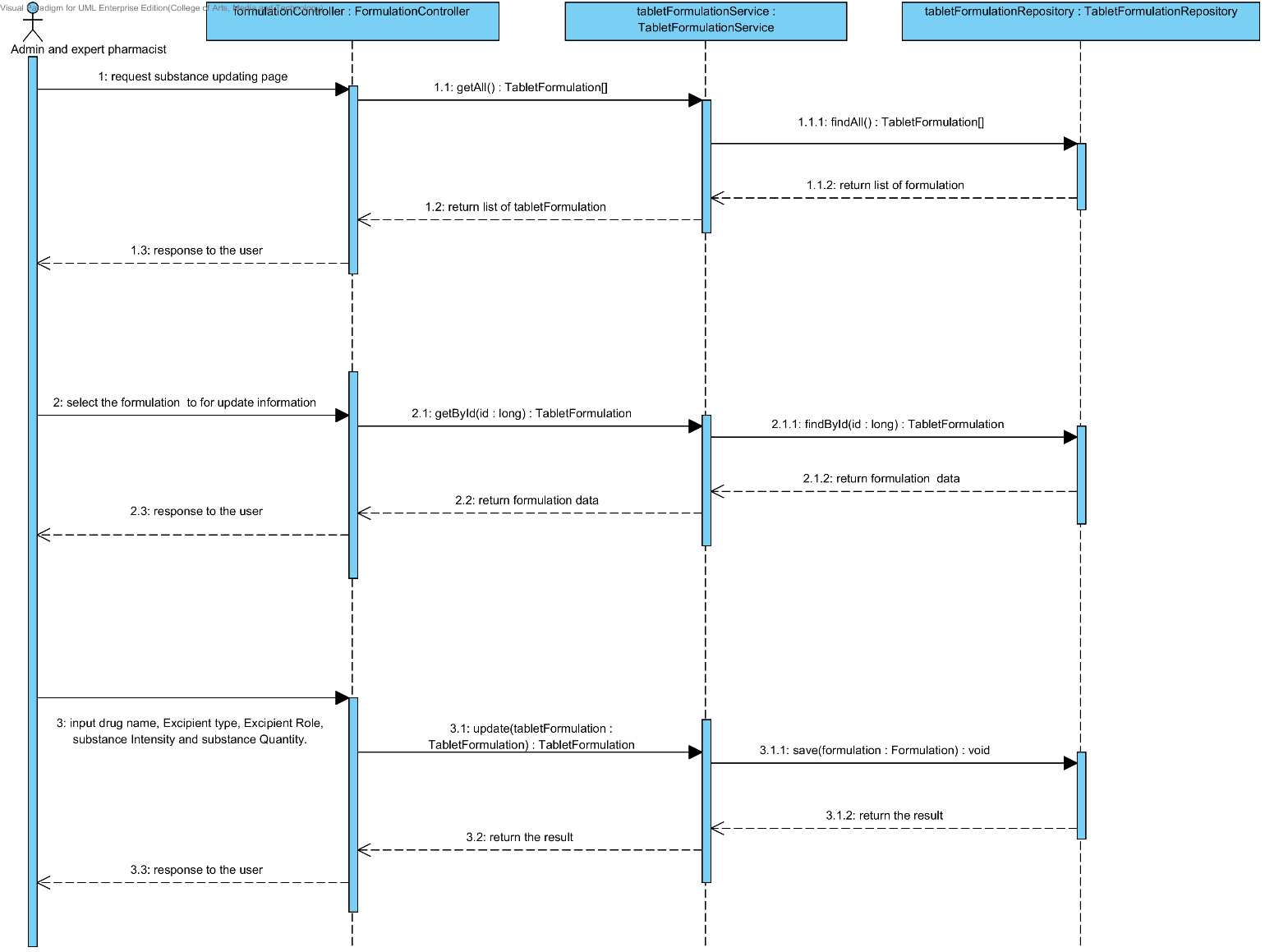
##### 4.4.1.1 SQD-20: The user adds a new drug’s formulation to the system.



### 4.4.2 URS-21: 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 name, excipient, substance quantity and substance intensity. 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.

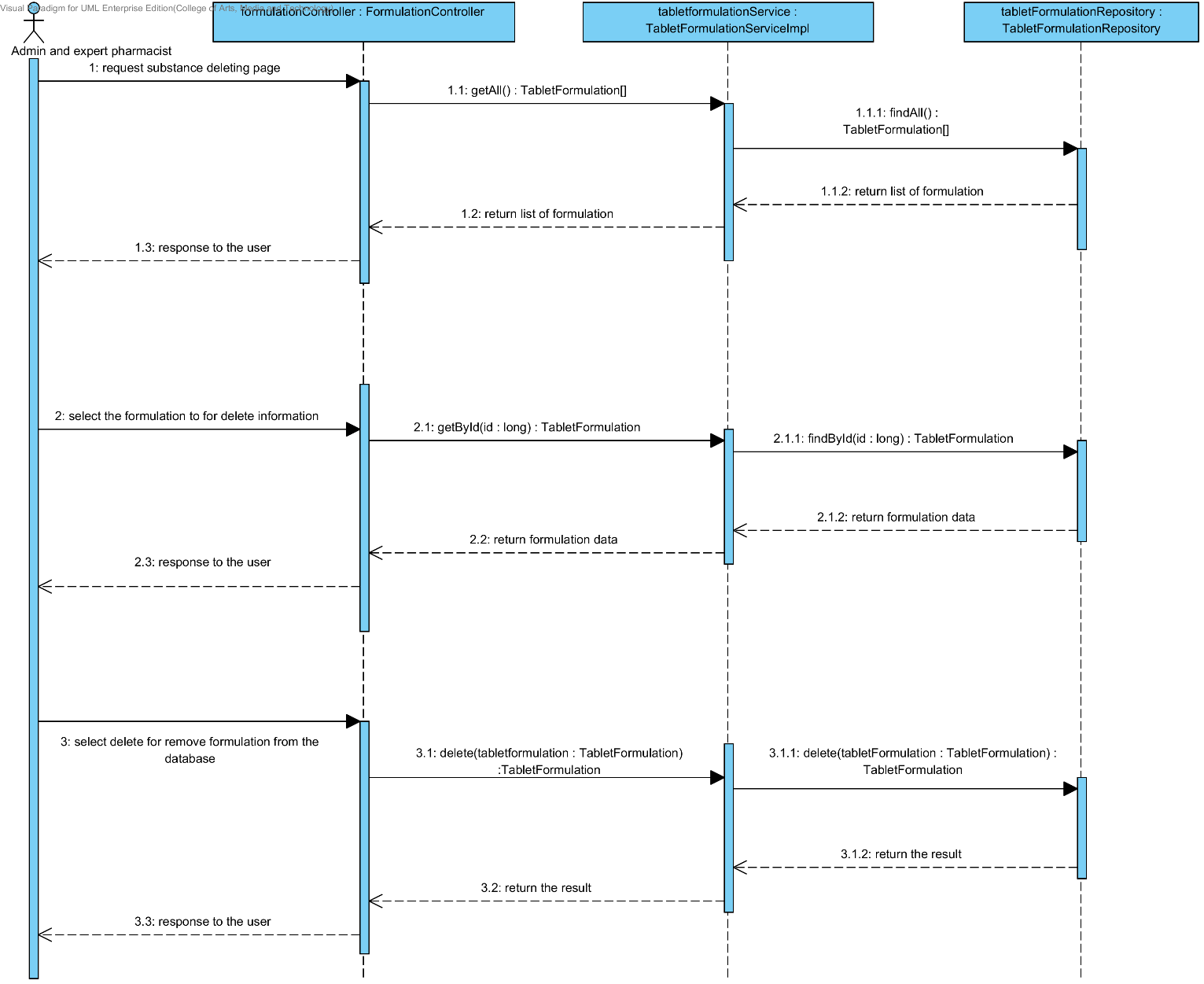
##### 4.4.2.1 SQD-21: The user updates an existing drug’s formulation in the system.



### 4.4.3 URS-22: 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.

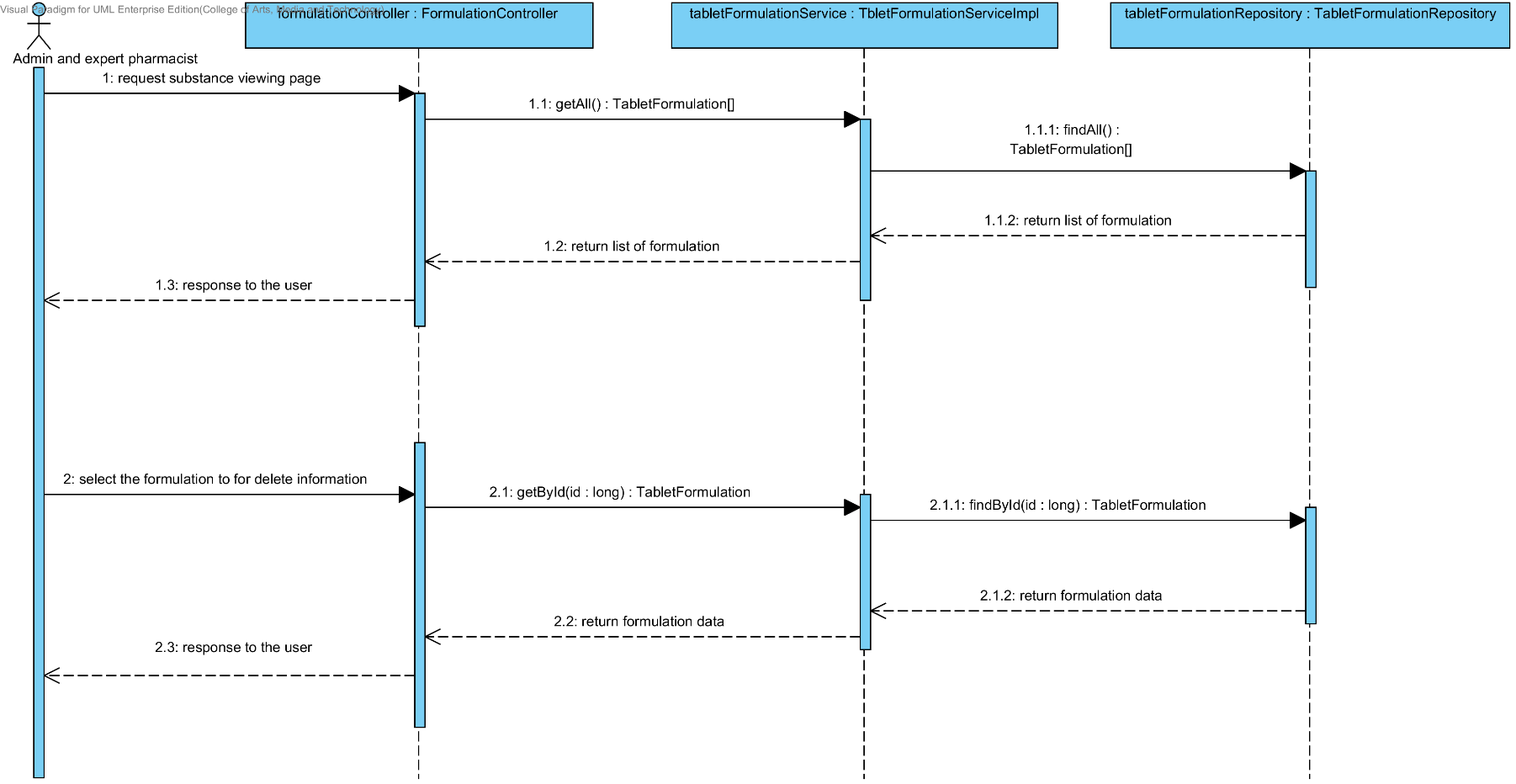
##### 4.4.3.1 SQD-22: The user deletes an existing drug’s formulation from the system.



### 4.4.4 URS-23: 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.

##### 4.4.4.1 SQD-23: The user views the drug’s formulation in the system.



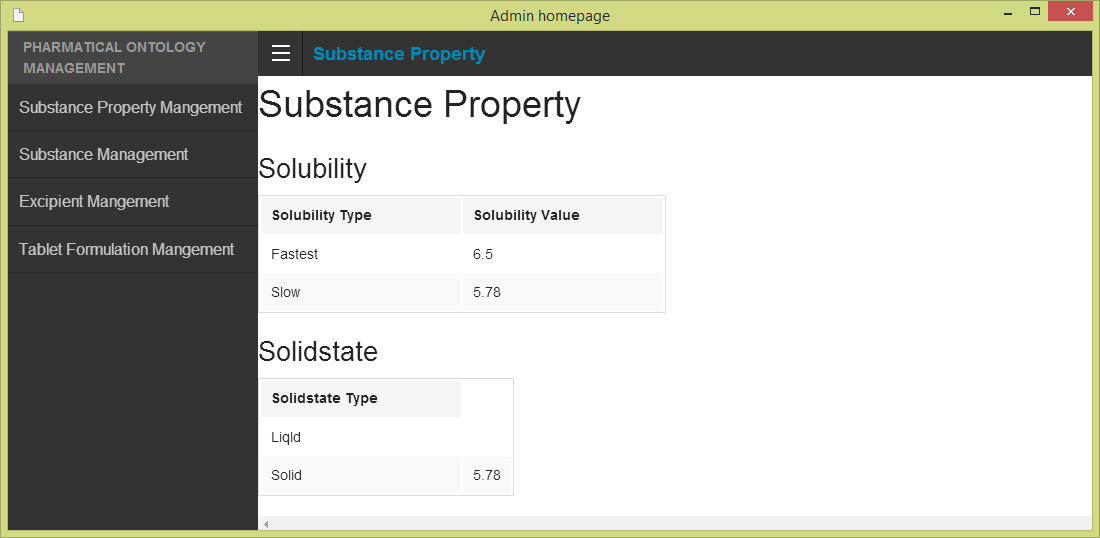
# Chapter 5 | User Interface

In the 1st progress the URS is related with the list of user interface that shown below this passage.

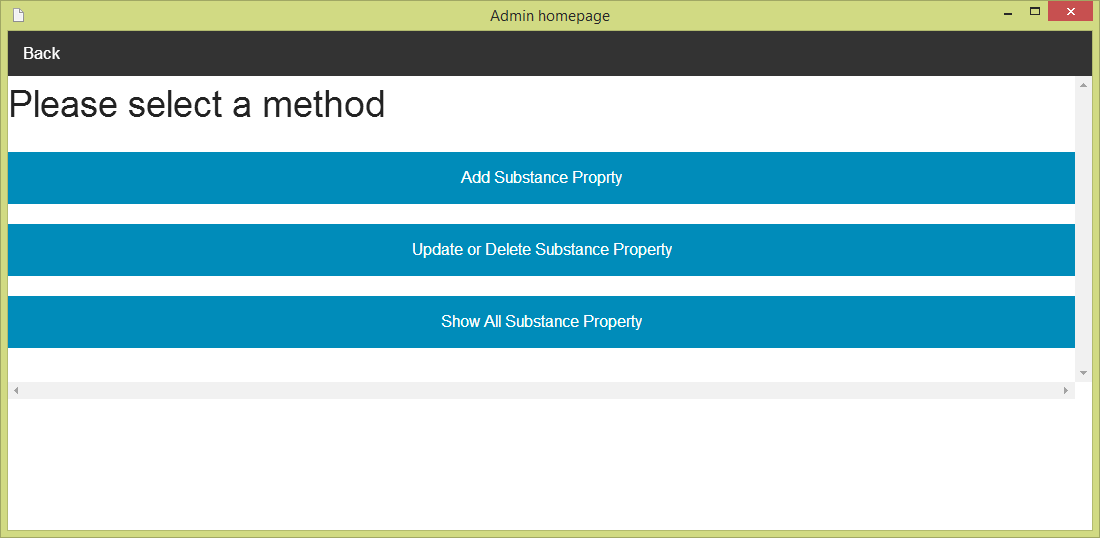
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Sub-Feature Name** | **URS No.** | **URS Name** | **Sequence Diagram** |
| 5 | Manage the drug substance property | URS-09 | The user adds a new substance property into the system. | UI-01,UI-02,UI-03,UI-04 |
| URS-10 | The user updates an existing substance property into the system. | UI-01,UI-02,UI-03,UI-05,UI-06 |
| URS-11 | The user deletes an existing substance property from the system. | UI-01,UI-02,UI-03,UI-05,UI-06 |
| 6 | Manage the drug substance | URS-12 | The user adds a new substance into the system. | UI-01,UI-07,UI-08 |
| URS-13 | The user updates an existing substance into the system. | UI-01,UI-07,UI-09,UI-10 |
| URS-14 | The user deletes an existing substance from the system. | UI-01,UI-07,UI-09,UI-10 |
| URS-15 | The user views the substance in the system. | UI-01,UI-07,UI-09,UI-10 |
| 7 | Manage the drug excipient | URS-16 | The user adds a new excipient to the system. | UI-01,UI-11,UI-12 |
| URS-17 | The user updates an existing drug excipient in the system. | UI-01,UI-09,UI-11,UI-13 |
| URS-18 | The user delete an existing drug excipient in the system. | UI-01,UI-09,UI-11,UI-13 |
| URS-19 | The user views all the drug excipient in the system. | UI-01,UI-09,UI-11,UI-13 |
| 8 | Manage the drug formulation | URS-20 | The user adds a new drug formulation case into the system. | UI-01,UI-14,UI-15 |
| URS-21 | The user updates an existing drug formulation case in the system. | UI-01,UI-14,UI-16,UI-17 |
| URS-22 | The user deletes an existing drug formulation case in the system. | UI-01,UI-14,UI-16,UI-17 |
| URS-23 | The user views all of the formulation in the system. | UI-01,UI-14,UI-16,UI-17 |

## 5.1- Sub-Feature 5: Manage the drug substance property

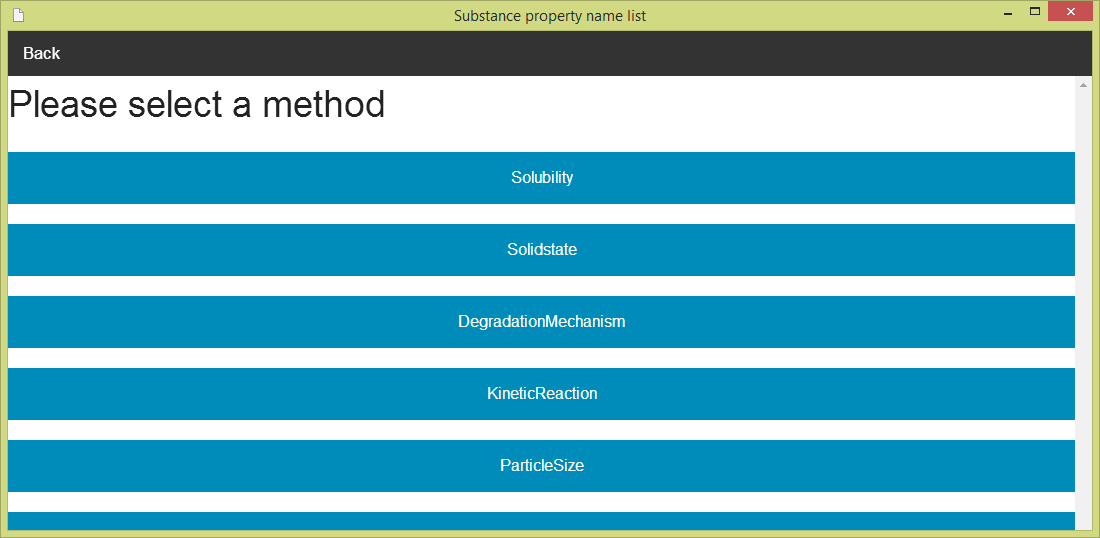
### 5.1.1 URS-09: The user adds a new substance property into the system.

 In the user interface design, the user can add a new substance property by opening the main page of a program, then the user select “Substance Property Management”. The main menu shows on the figure 68.

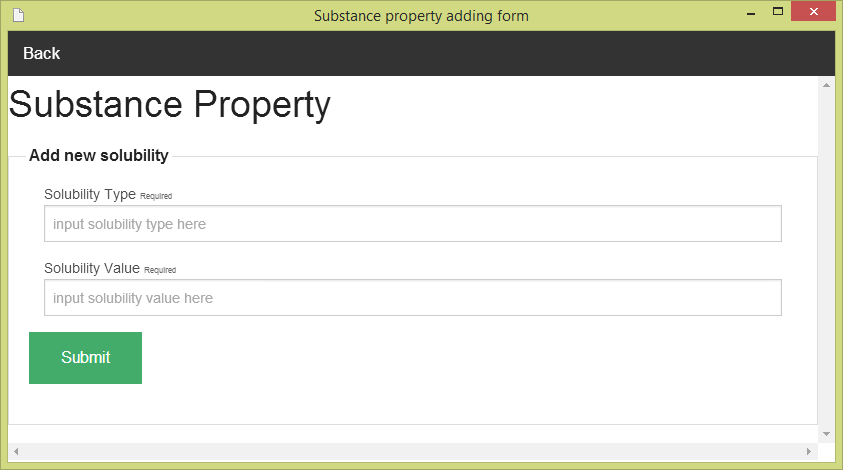
**Figure 68- UI-01: Admin main menu**

After the user selects “Substance Property Management”. The Substance management menu will show on the screen. The user must selects “Adding substance property” for making substance property adding. The substance management menu is shown on the figure 69.

**Figure 69- UI-02: Substance property management menu**

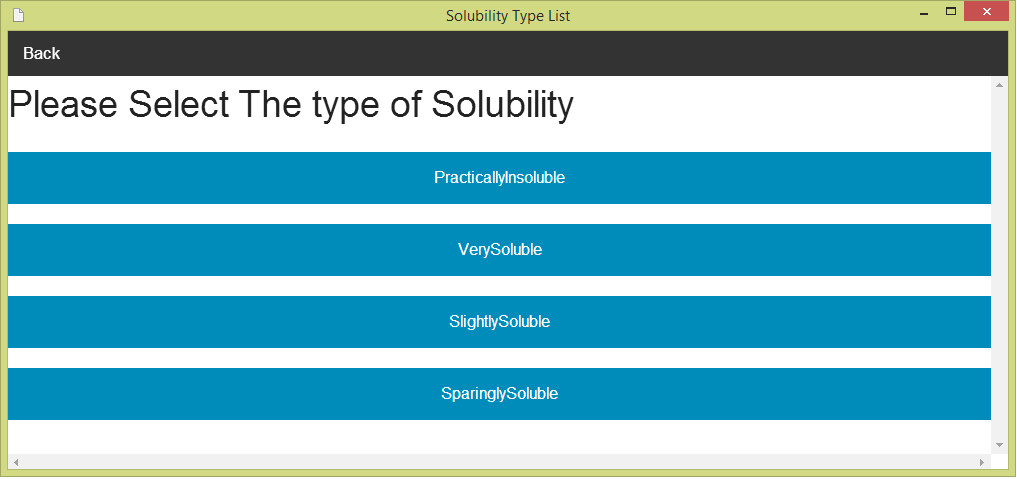
After the user selects “Add Substance Property”. The Substance property name list will show on the screen. The user must can select one substance property (e.g. Solubility) for making substance property adding. The substance property name list is shown on the figure 70.

**Figure 70 - UI-03: Substance property management menu**

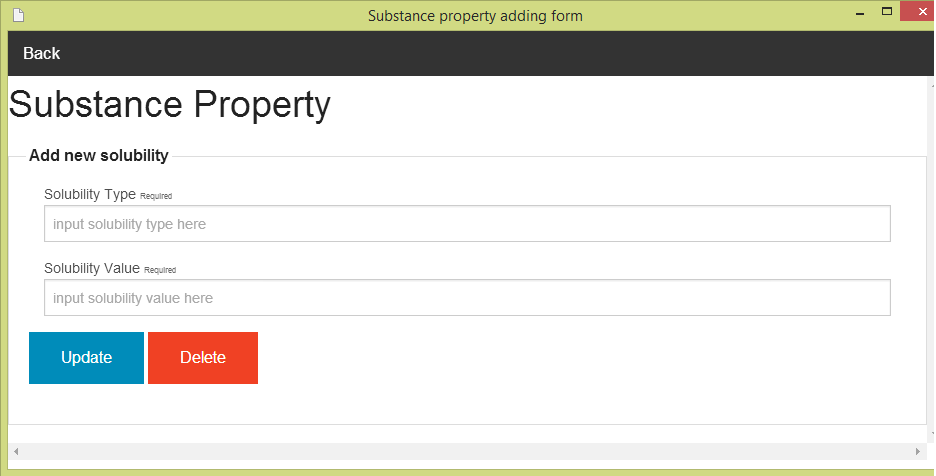
****After the user selects one of substance property in the list (e.g. solubility). The Substance property adding form will show on the screen. The user must input data of substance property to the form. The Substance adding form can check the input format. The substance property adding form is show on the figure 71

**Figure 71 - UI-04: Substance property adding form**

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

 In the user interface design, the user can update an existing substance property by opening the main page of the program (figure 68).Then, the user select “Substance Property Management." After that, the system will show substance management menu on the screen (figure 69). The user can select “update and delete substance property” for making update and delete substance property. Next, the system will show the substance property name list on the screen (figure 70).The user must select one of substance property (e.g. solubility). Then, the system will show a list of substance property type (e.g. if user selects solubility, the list of a type will be solubility type).The user must select one type from the list of substance property type. The list of substance property type is shown on the figure 72.

**Figure 72 - UI-05: Substance property adding form**

After the user selects one of substance property type in the list (e.g. Very soluble). The Substance property updating and deleing form will show on the screen. The user must input data of substance property to the form. The substance property updating can check the input format. The substance property updating and deleting form is show on the figure 73

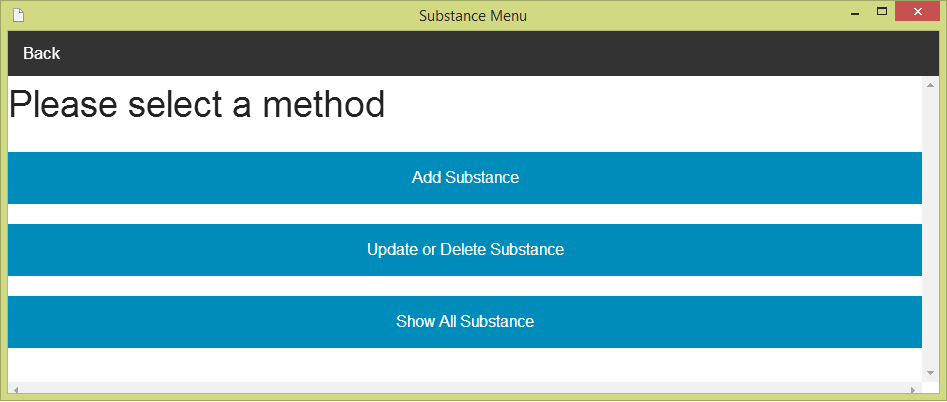
**Figure 73 - UI-06: Substance property updating and deleting form**

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

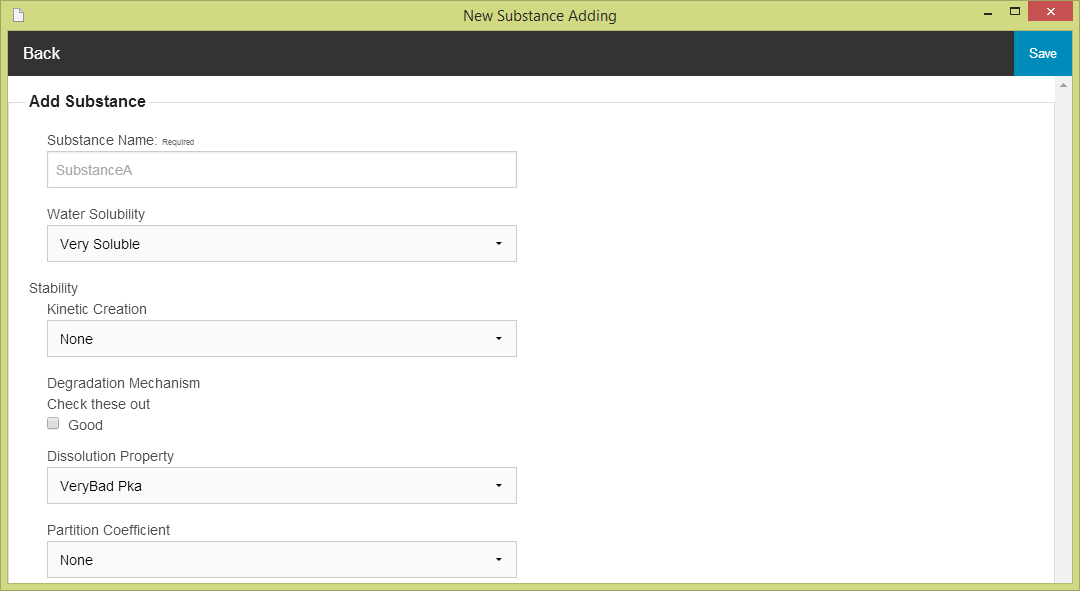
The user interface of this URS is same as URS-10.

## 5.2- Sub-Feature 6: Manage the drug substance

### 5.2.1 URS-12: The user adds a new substance to the system.

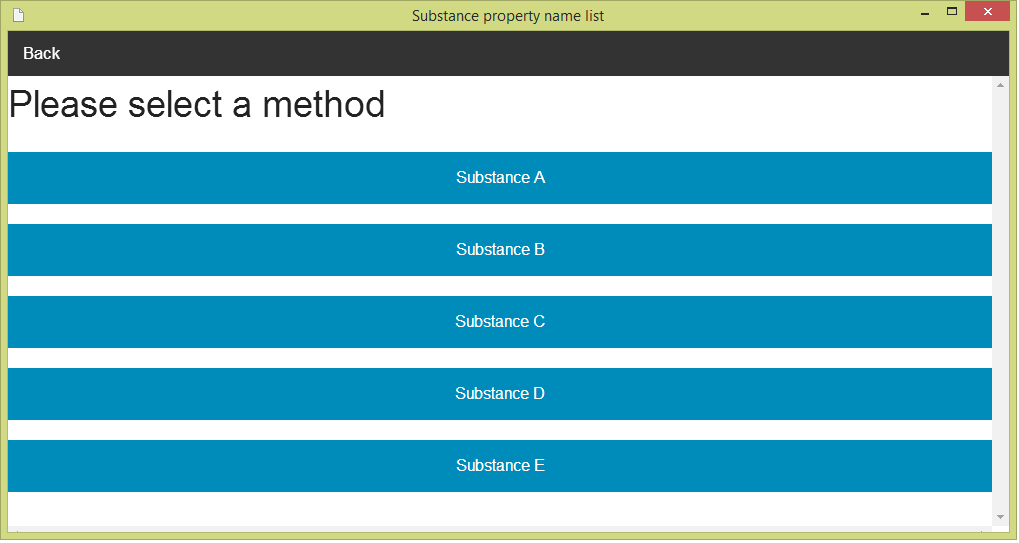
 In the user interface design, the user can add a new substance by opening the main page of a program (figure 68).Then the user select “Substance Management”. After that, the system will show substance management menu on the screen. The user can select “Add a new substance” for making substance adding. The substance management menu is show on the figure 74.

**Figure 74 - UI-07: Substance management menu**

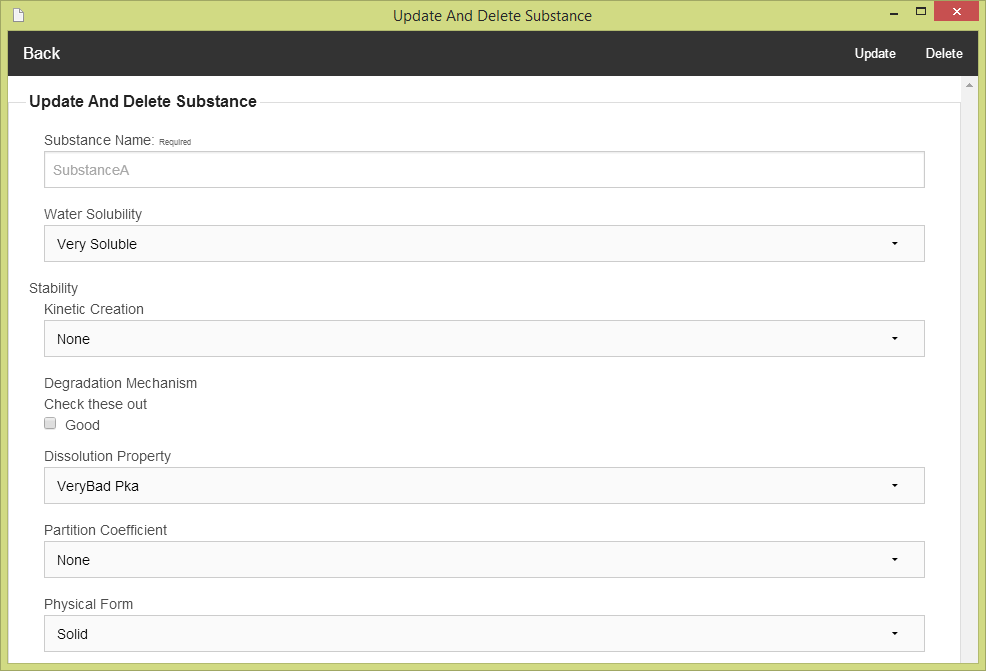
After the user selects “Adding substance”. The Substance adding form will show on the screen. The user must input data of substance to the form. The substance adding form can check the input format for the user. The substance adding form is show on the figure 75

**Figure 75 - UI-08: Substance adding form**

### 5.2.2 URS-13: The user updates an existing substance in the system.

 In the user interface design, the user can update an existing substance by opening the main page of the program (figure 68).Then, the user select “Substance Management." After that, the system will show substance management menu on the screen (figure 74). The user can select “update and delete substance”. Next, the system will show the substance name list on the screen .The user must select one of substance property (e.g. Substance A) for making update and delete the substance. The substance name list is show the figure 76.

**Figure 76 - UI-09: Substance name list**

****After the user selects one of substance property (e.g. Substance A). The Substance updating and deleting form will show on the screen. The user must input data of substance to the form. The substance updating and deleting form can check the input format from the user. The substance updating and deleting form is show on the figure 77.

**Figure 77 - UI-10: Substance updating and deleting form**

### 5.2.3 URS-14: The user deletes an existing substance from the system.

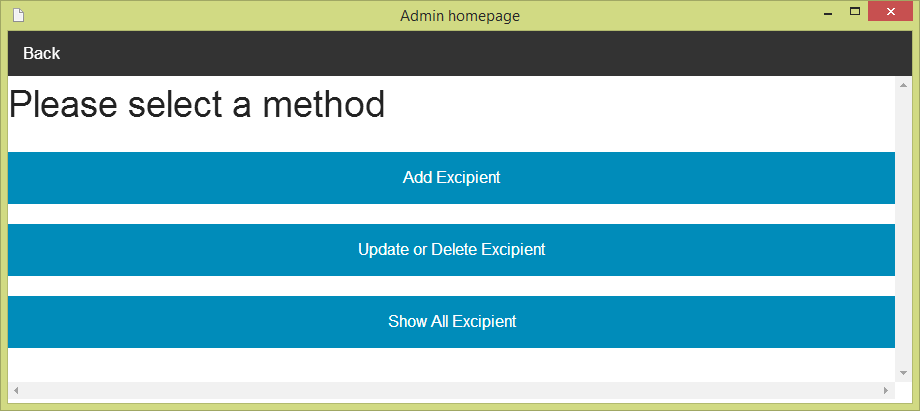
The user interface of this URS is same as URS-13.

### 5.2.4 URS-15: The user views the substance in the system

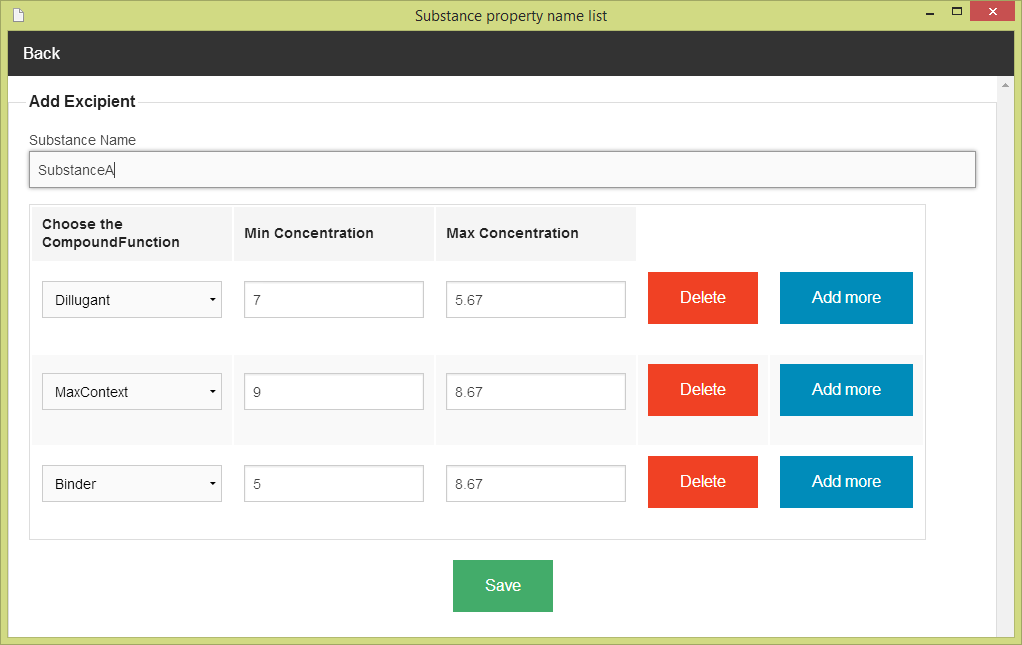
The user interface of this URS is same as URS-13.

## 5.3- Sub-Feature 7: Manage the drug excipient

### 5.3.1 URS-16: The user adds a new excipient to the system.

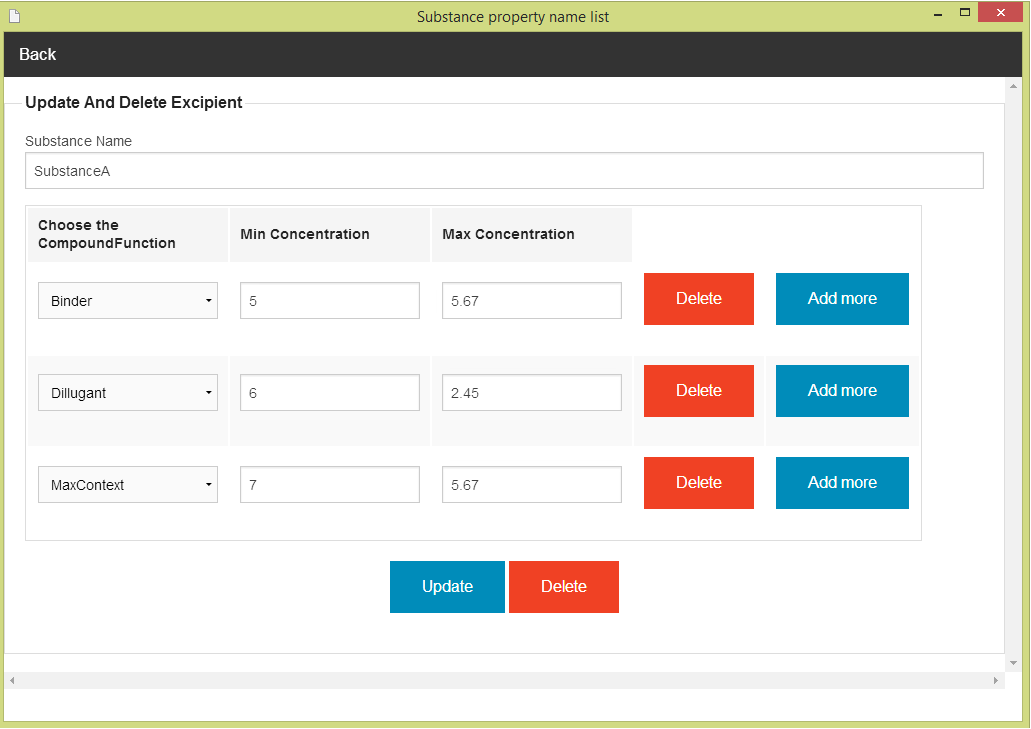
**** In the user interface design, the user can add a new excipient by opening the main page of a program (figure 68).Then the user select “Excipient Management”. After that, the system will show Excipient management menu on the screen. The user can select “Add Excipient” for making excipient adding. The excipient management menu is show on the figure 78.

**Figure 78 - UI-11: Excipient management menu**

****After the user selects “Adding Excipient”. The system show the list of substance name on the screen (Figure 76). The user must select one of substance from the substance name list. After that, the system will show excipient adding form. The user must input excipient data to the form. The excipient adding form can check the input format from the user. The excipient adding form is show on the figure 79.

**Figure 79 - UI-12: Excipient adding form**

### 5.3.2 URS-17: The user updates an existing excipient in the system.

**** In the user interface design, the user can update an existing excipient by opening the main page of the program (figure 68).Then, the user select “Excipient Management." After that, the system will show excipient management menu on the screen (figure 78). The user can select “update and delete excipient”. Next, the system will show the substance name list on the screen (Figure 76) .The user must select one of substance property (e.g. Substance A) for making update and delete the substance. After that, the system will show the excipient updating and deleting form. The user must input data of excipient to the form. The excipient adding form can check the input format from the user. The excipient updating and deleting form is show on the figure 80

**Figure 80 - UI-13: Excipient updating and deleting form**

### 5.2.3 URS-18: The user deletes an existing excipient from the system.

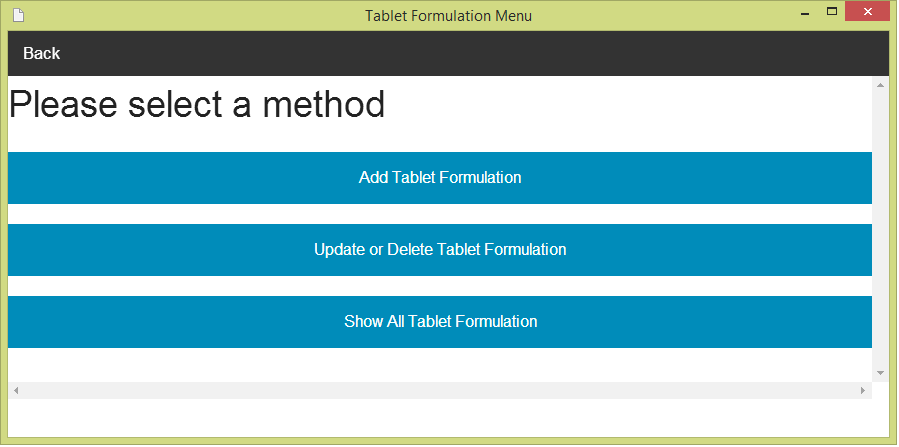
The user interface of this URS is same as URS-17.

### 5.2.4 URS-19: The user views the excipient in the system

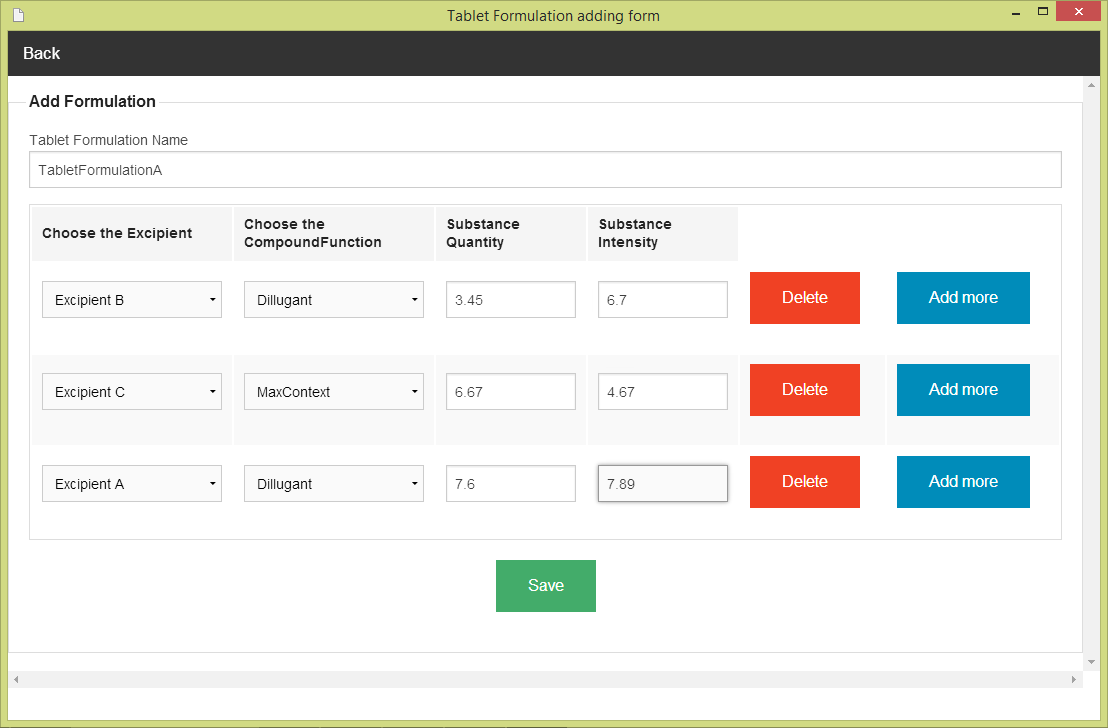
The user interface of this URS is same as URS-17.

## 5.4- Sub-Feature 8: Manage the drug formulation

### 5.4.1 URS-20: The user adds a new formulation to the system.

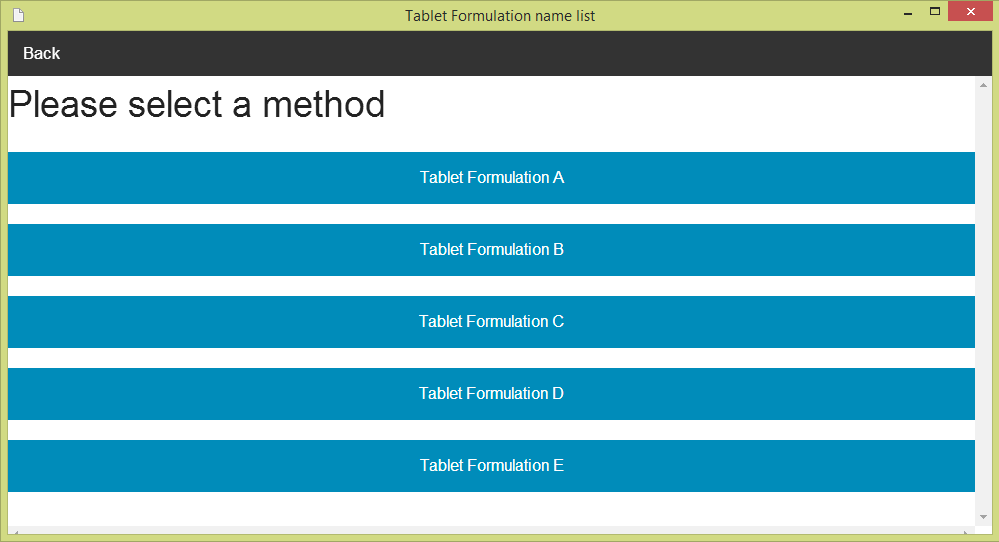
 In the user interface design, the user can add a new formulation by opening the main page of a program (figure 68).Then the user select “Tablet formulation Management”. After that, the system will show tablet formulation management menu on the screen. The user can select “Add a new formulation” for making tablet formulation adding. The tablet formulation management menu is show on the figure 81.

**Figure 81 - UI-14: Tablet Formulation management menu**

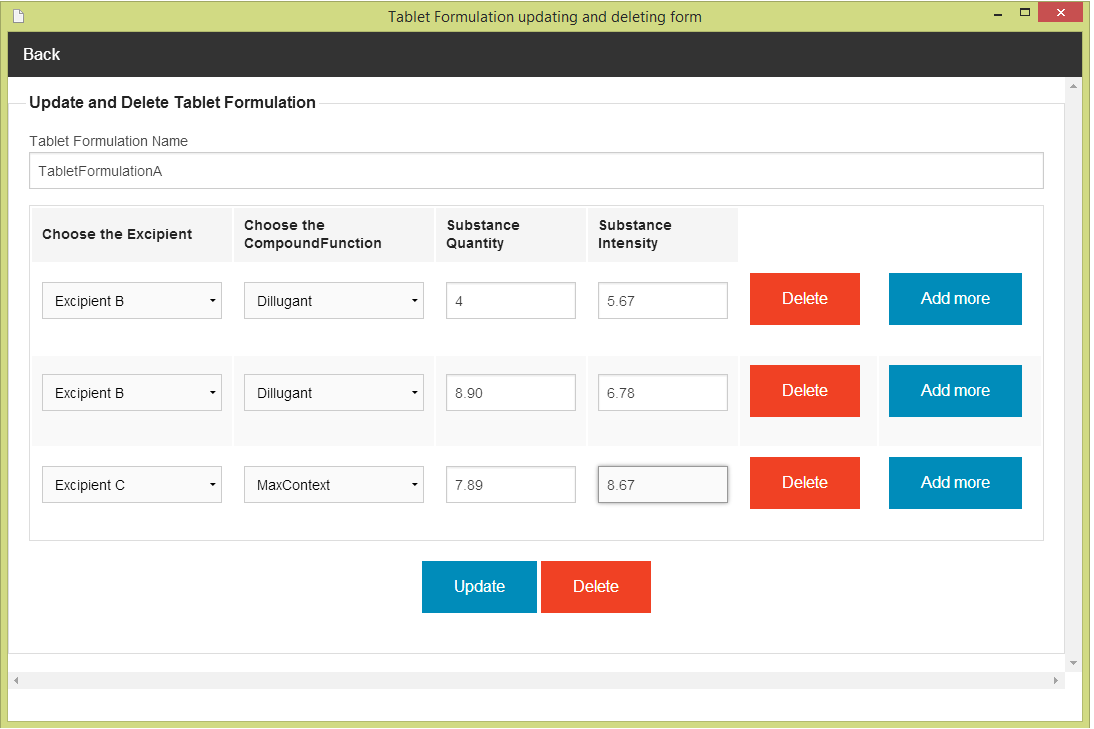
****After the user selects “Adding Tablet Formulation”. The Tablet Formulation adding form will show on the screen. The user must input the tablet formulation data to the form. The tablet formulation adding form can check the input format from the user. The tablet formulation adding form is show on the figure 82.

**Figure 82 - UI-15: Tablet Formulation adding form**

### 5.4.2 URS-21: The user updates a tablet formulation in the system.

 In the user interface design, the user can update an existing tablet formulation by opening the main page of the program (figure 68).Then, the user select “Tablet Formulation Management." After that, the system will show tablet formulation management menu on the screen (figure 81). The user can select “update and delete TabletFormulation”. Next, the system will show the list of tablet formulation name on the screen .The user must select one of tablet formulation (e.g. TabletFormulation A) for making update and delete the TabletFormulation. The list of tablet formulation is show the figure 83.

**Figure 83 - UI-16: The list of tablet formulation name**

After the user selects one of tablet formulation (e.g. TabletFormulation A). The TabletFormulation updating and deleting form will show on the screen. The user must input tablet formulation data to the form. The tabletformulation updating and deleting form can check the input format from the user. The tabletformulation updating and deleting form is show on the figure 84

**Figure 84 - UI-17: TabletFormulation updating and deleting form**

### 5.4.3 URS-22: The user deletes an existing tablet formulation from the system.

The user interface of this URS is same as URS-21.

### 5.4.4 URS-23: The user views the tablet formulation in the system

The user interface of this URS is same as URS-21.

1. NS – Narongrit Saisuwan , PW – Panupak Wichaidit , CD - Chartchai Doungsa-ard [↑](#footnote-ref-1)