SMART PRESCRIPTION APPLICATION

Software Design

By

Mr. Natthakan	Kaeokanpai	552115020
Mr. Phithiwat	Sitthitun	552115051

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

Project Advisor

_____

Dr.Prompong Sugunnasil

Document History

Document Name	Version	Status	Date	Viewable	Editable	Responsible
Documents						
Smart prescription application _Software design	Software Design _V0.1	Reviewed	June 10th 2016	NAT,PHI, PROM	NAT,PHI	NAT,PHI
Smart prescription application _Software design	Software Design _V1.0	Released	July 10th 2016	NAT,PHI, PROM	NAT,PHI	NAT,PHI
Smart prescription application _Software design	Software Design _V.1.1	Released	July 20th 2016	NAT,PHI, PROM	NAT,PHI	NAT,PHI
Smart prescription application _Software design	Software Design _V.2.0	Released	July 30th 2016	NAT,PHI, PROM	NAT,PHI	NAT,PHI

Acronym

NAT = Natthakan Kaeokanpai

PHI = Phithiwat Sitthitun

PROM = Prompong Sugunnasil

Content

Chapter One | Introduction 5

1.1 Objective 5

1.2 Project Scope 5

1.3Acronyms and Definitions 5

1.3.1 Acronyms 5

1.3.2Definitions 6

Chapter Two | System Architecture 7

Chapter Three | Detailed Design 8

3.1 Class Diagram of smart prescription application 8

3.2 Class Description 10

(Mobile application) 27

Chapter Four | Database Design 45

4.1 ER Diagram 45

4.2 Table Description 46

Chapter Five | Sequence Diagram 49

SD-01: Doctors can view a doctor home page on the web application. 49

SD-02: Doctors can create a patient's profile on the web application. 50

SD-03: Doctors can update a patient's profile on the web application. 51

SD-04: Doctors can delete a patient's profile on the web application. 52

SD-05: Doctors can search a patient's profile on the web application. 53

SD-06: Doctors can view a patient's profile on the web application. 54

SD-07: Doctors can view a list of patient's profile on the web application. 55

SD-08: Administrators can view an admin home page on the web application. 56

SD-09: Administrators can create a user's profile on the web application. 57

SD-10: Administrators can update a user's profile on the web application. 58

SD-11: Administrators can delete a user's profile on the web application. 59

SD-12: Administrators can search a user's profile on the web application. 60

SD-13: Administrators can view a user's profile on the web application. 61
SD-14: Administrators can view a list of user's profile on the web application. 62
SD-15: Doctors, Administrators, and FDAs can login to the web application. 63
SD-16: Doctors, Administrators, and FDAs can logout from the web application. 64
SD-17: Patients can login to the mobile application. 65
SD-18: Patients can logout from the mobile application. 66
SD-19: Patients can view the QR code on the mobile application. 67
SD-20: Pharmacists can scan the QR code on the mobile application 68
SD-21: Pharmacists can add the time of dispensation to the patient's profile on the mobile application. 69
SD-22: Doctors can create an allergy report on the web application. 70
SD-23: Doctors can update an allergy report on the web application. 71
SD-24: Doctors can delete an allergy report on the web application. 72
SD-25: Doctors can search an allergy report on the web application. 73
SD-26: Doctors can view an allergy report on the web application. 74
SD-27: Doctors can view a list of allergy report on the web application. 75
SD-28: FDAs can view a FDA home page on the web application. 76
SD-29: FDAs can view a list of allergy report on the web application. 77
SD-30: FDAs can view a list of allergy report on the web application. 78
Chapter Six | User Interface Design 79

Chapter One | Introduction

1.1 Objective

The objective of the Software Design Document for "Smart prescription application" both Android mobile application and web application are to provide a description and explain of the design system including activity following requirements specification and before programming. This document is based on the project proposal, project plan and software requirement specification and help developers to understand how to work for this application.

1.2 Project Scope

Smart prescription application consists of a web application and a mobile application. Smart prescription application requires a web application for creating a patient's profile and allergy report by a doctor and the Smart prescription application require a mobile application to generate and scan the QR code on a mobile phone by a pharmacist.
The features of the web application consist of:
Feature#1: Prescription system
Feature#2: Account management system
Feature#3: Authentication system
Feature#5: Report allergy system
Feature#6: Allergy drug summary report system
The features of the mobile application consist of:
Feature#4: Prescription verification system

1.3Acronyms and Definitions

1.3.1 Acronyms

SDD	Software Design Document
CD	Class Diagram
URS	User Requirement Specification
UI	User Interface
SD	Sequence Diagram

1.3.2Definitions

Class diagram

A type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.[1]

Feature

Activity after input parameters and output the processing to user interface. It explain about functionality of an application used for requirements analysis, coding, design, and testing

Sequence diagram

A sequence diagram is an interaction diagram that shows how processes operate with one another and in what order. A sequence diagram shows object interactions arranged in time sequence.

UML

A standard notation for the modeling of real-world objects as a first step in developing an object-oriented design methodology [2]

Chapter Two | System Architecture

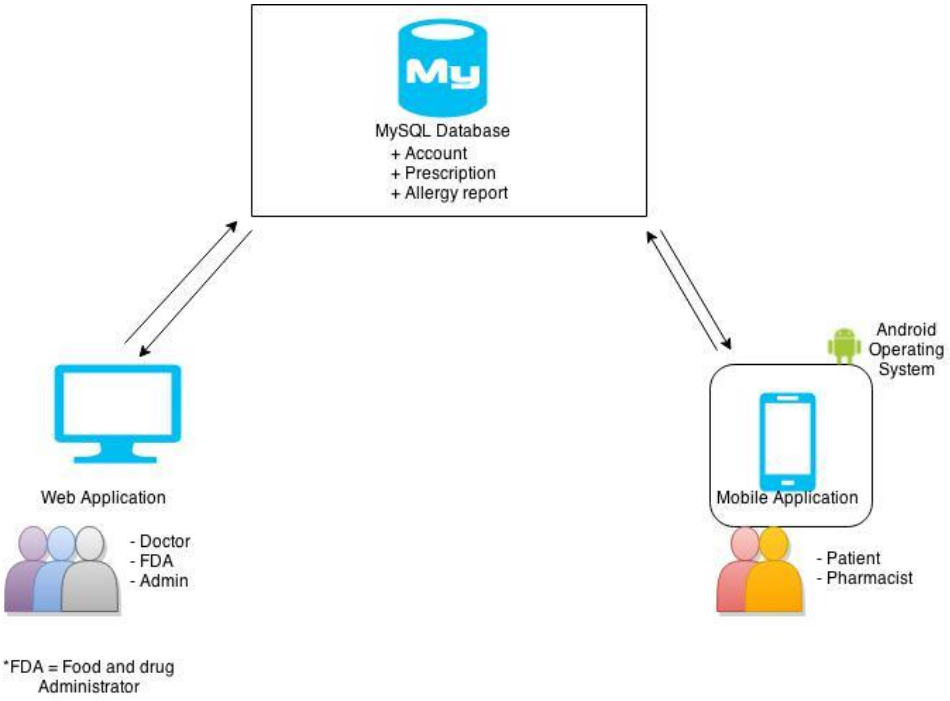


Figure 1: System Architecture (Overview)

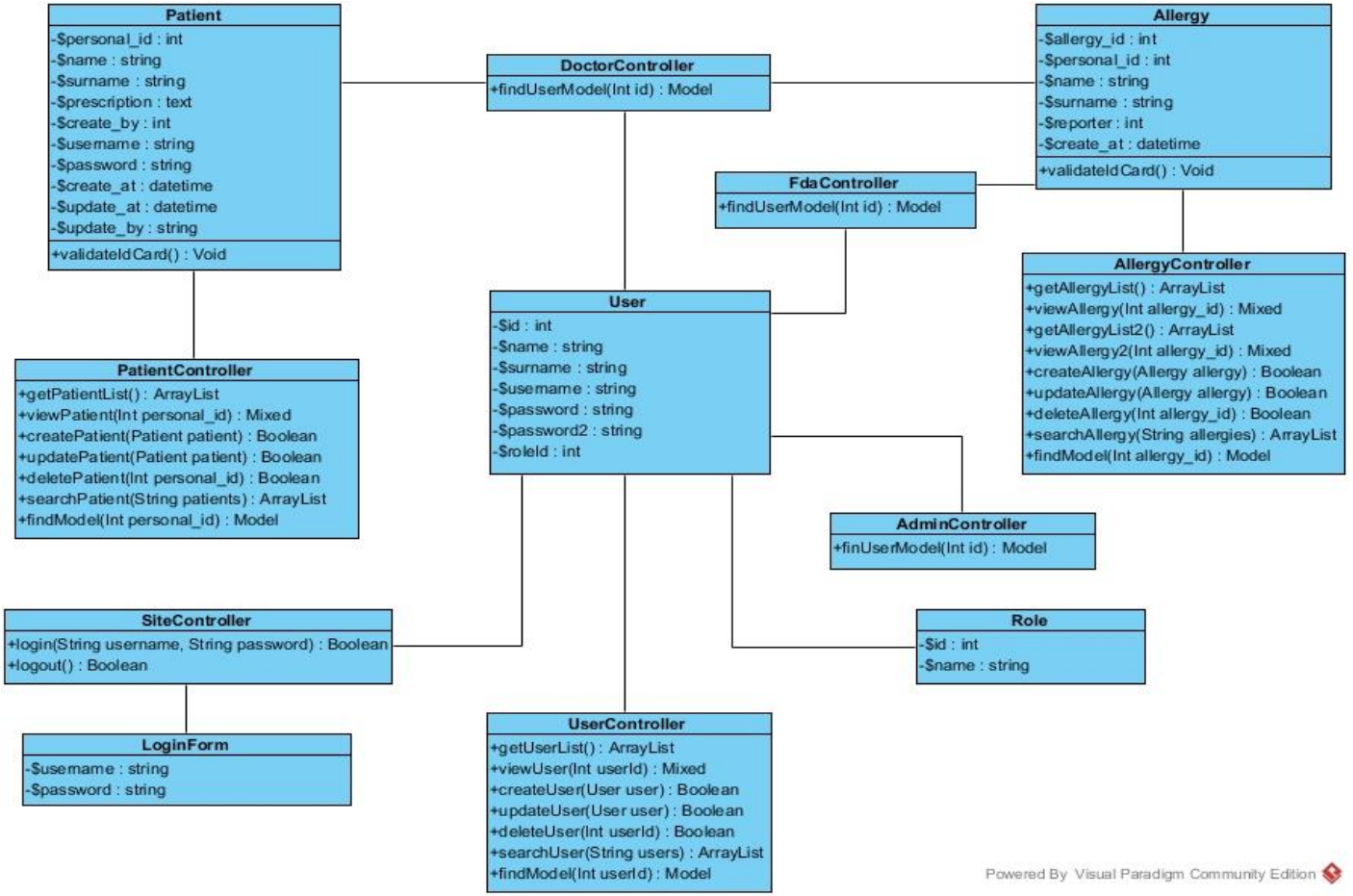
The architecture of "Smart Prescription Application", show in the figure 1, consists of two parts. The first part is the web application. In this system, there are three types of user doctor, FDA and admin. The users can use web application via a web browser by access to the web application by login. Doctors can create the patient's profiles and allergy report to MySQL database. The FDAs can view the allergy report on the web application. An admin can create and manage account of doctor and FDA to MySQL database.

The second part is the mobile application Smart Prescription Application stand for android operating system. There have two types of users; patients and pharmacists. Patients can view the QR code on the mobile application. Pharmacist can verify QR code and get the information of patient's profile by scan QR code function from database on the mobile application.

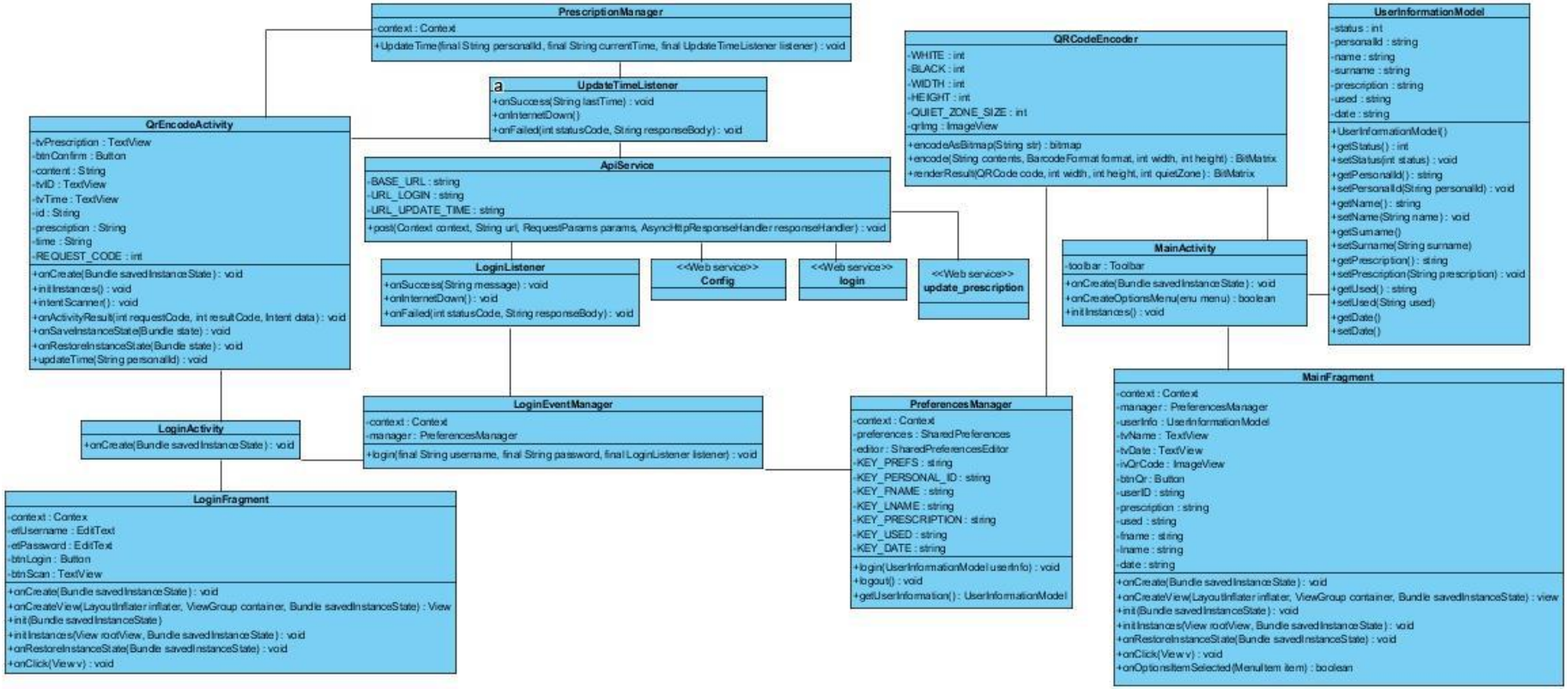
Chapter Three | Detailed Design

3.1 Class Diagram of smart prescription application

(Web application)



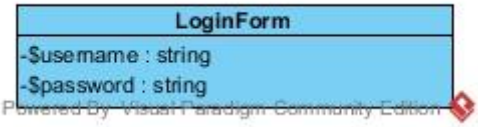
(Mobile application)



3.2 Class Description

(Web application)

3.2.1 Class-01: LoginForm



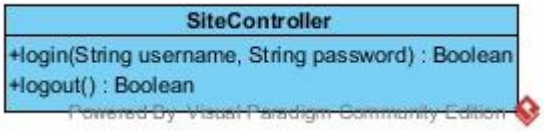
Description:

The LoginForm model class is using for store the username and password to login.

Attributes:

No.	Name	Description	Type
1	$username	This attribute is username of users	String
2	$password	This attribute is password of users	String

3.2.2 Class-02: SiteController



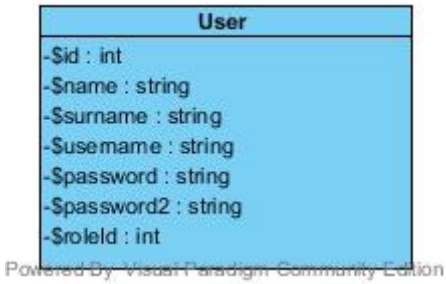
Description:

This class controls the login and logout function method.

Methods :

No.	Name	Description	Parameter Lists	Return Type
1	login()	- This method is logs in the user using the given username and password in the model. - This method returns true when users input username and password are correct. - This method returns false when users input username or password is incorrect.	String username String password	Boolean
3	logout()	- This is the action to logs out the current user and redirect to homepage.	-	Boolean

3.2.3 Class-03: User



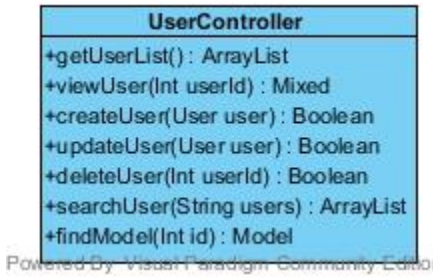
Description:

This is the User model class for table "tbl_user". It used to store the user's information. This model class manages and invokes the user's information.

Attributes:

No.	Name	Description	Type
1	$id	This attribute is id of user.	Int
2	$name	This attribute is name of user.	String
3	$surname	This attribute is surname of user.	String
5	$username	This attribute is username of user.	String
6	$password	This attribute is password of user.	String
7	$password2	This attribute is confirm password of user.	String
8	$roleId	This attribute is role of user, which include 1, 2, and 3.	Int

3.2.4 Class-04: UserController



Description:

This class is UserController using for controls and manages (CRUD) the user's profile and information. This class controls the user by admin.

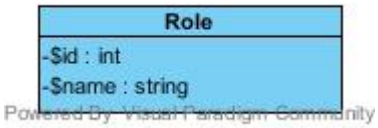
Methods :

No.	Name	Description	Parameter Lists	Return Type
1	getUserList()	- This method is used to providing the list all users. - This method is used to display over all list of user that exist on the database.	-	ArrayList<User>
2	viewUser(Int userId)	- userId is id of user from user table. - This method used for view the single user's information which includes id, position, name, surname, username, and password to display on the 'view' page by using id of user. - The mechanism of this method involves other method: findModel(int id). - This method returns user's information when user id that exist on the user table via user model. - This method returns false if the user id that does not exist on the user table.	Int userId	Mixed

3	createUser(User user)	- This method is used to creating the new data of user into database, the data that input to create will be validated by AJAX validation. - The mechanism of this method involves other method: validate() and save() from framework. - This method returns true if admin input overall information is correct. -This method returns false if admin does not input some information or inputting is incorrect.	User user	Boolean
4	updateUser(User user)	- This method is used to updating the data of user from the database by using user id, the data that input will be validated by AJAX validation. - The mechanism of this method involves other method: fineModel(int id), validate(), and save() from framework. - This method returns true if admin input the new overall information is correct. -This method returns false if admin does not input some information or inputting is incorrect.	User user	Boolean

5	deleteUser(Int userId)	- userId is id of user from user table. - This method is used to deleting the user from the database by using delete() method from framework via user id. - The mechanism of this method involves other method: findModel(int id) and delete() from framework. - This method returns true when delete user id that exist on the user table. - This method will return false when delete user id that does not exist on the user table.	Int userId	Mixed
6	searchUser(String users)	- This method is search query applied. - This method to get ActiveDataProvider instance which will filter models according to data in model fields. - This method returns data items currently available when input attributes that exist in the database.	String users	ArrayList<User>
7	findModel(Int id)	- id is user id from user table. - This method is used to finding the model based on its primary key value. - If the model is not found, a 404 HTTP exception will be thrown. - This method returns model object by id.	Int id	Model

3.2.5 Class-05: Role



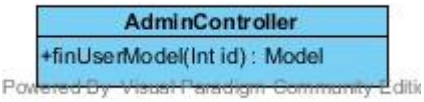
Description:

This is the Role model class for table "tbl_role". This model class used to store the role information.

Attributes:

No.	Name	Description	Type
1	$id	Id of role	Int
2	$name	The name of role	String

3.2.6 Class-06: AdminController



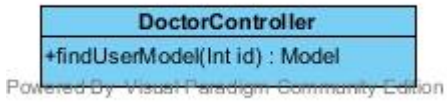
Description:

The AdminController class is the controller to separates the particular user on role type of admin and lead to the specific index page about the User model.

Methods :

No.	Name	Description	Parameter Lists	Return Type
1	findUserModel(Int id)	- id is user id from user table. - This method is used to finding the model based on its primary key value of User model. - If the model is not found, a 404 HTTP exception will be thrown. - This method returns model object by id.	Int id	Model

3.2.7 Class-07: DoctorController



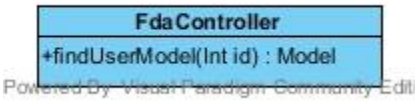
Description:

The DoctorController class is the controller to separates the particular user on role type of doctor and lead to the specific index page about the User model.

Methods :

No.	Name	Description	Parameter Lists	Return Type
1	findUserModel(Int id)	- id is user id from user table. - This method is used to finding the model based on its primary key value of User model. - If the model is not found, a 404 HTTP exception will be thrown. - This method returns model object by id.	Int id	Model

3.2.8 Class-08: FdaController



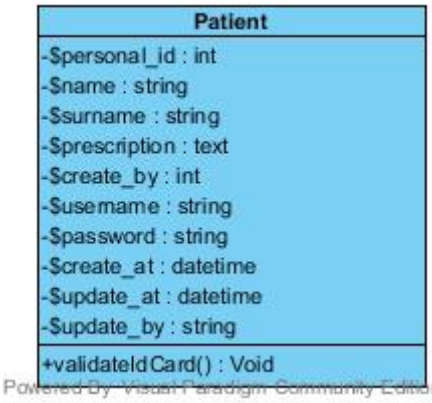
Description:

The FdaController class is the controller to separates the particular user on role type of FDA and lead to the specific index page about the User model.

Methods :

No.	Name	Description	Parameter Lists	Return Type
1	findUserModel(Int id)	- id is user id from user table. - This method is used to finding the model based on its primary key value of User model. - If the model is not found, a 404 HTTP exception will be thrown. - This method returns model object by id.	Int id	Model

3.2.9 Class-09: Patient



Description :

This is the Patient model class for table "tbl_patient". It used to store the patient's information. This model class manages and invokes the patient's information.

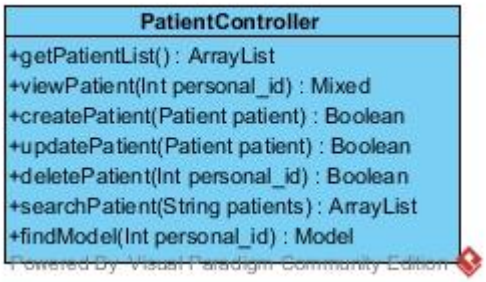
Attributes:

No.	Name	Description	Type
1	$personal_id	This attribute is personal ID of patient.	Int
2	$name	This attribute is name of patients.	String
3	$surname	This attribute is surname of patients.	String
4	$prescription	This attribute is prescription.	String
5	$create_by	This attribute is name of doctor who that create the prescription.	String
6	$username	This attribute is username of patient.	String
7	$password	This attribute is password of patient.	String
8	$create_at	This attribute is date time of prescription that created.	Date Time
9	$update_by	This attribute is name of doctor who that update the prescription.	String
10	$update_at	This attribute is date time of prescription that updated.	Date Time

Methods :

No.	Name	Description	Parameter Lists	Return Type
1	validateIdCard()	- This method is using for validate the personal id. It invoked in form validator via ajax validation.	-	Void

3.2.10 Class-10: PatientController



Description :

This class use for controls and manages (CRUD) the information of the patient. This class controls the patient's information by doctor.

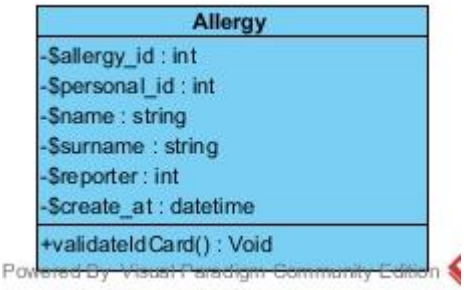
Methods :

No.	Name	Description	Parameter Lists	Return Type
1	getPatientList()	- This method is used to providing the list all patients. - This method is used to display over all list of patients that exist on the database.	-	ArrayList<Patient>
2	viewPatient(Int personal_id)	- personal_id is personal id of patient from patient table. - This method used for view the single user's information which includes personal id, name, surname, prescription, responsible, username, password and create date to display on the 'view' page by using personal id of patient.	Int personal_id	Mixed

		- The mechanism of this method involves other method: findModel(int personal_id). - This method returns patient's information when personal id that exist on the patient table via user model. - This method returns false if the personal id that does not exist on the patient table.		
3	createPatient(Patient patient)	- This method is used to creating the new data of patient into database, the data that input to create will be validated by AJAX validation. - The mechanism of this method involves other method: validate() and save() from framework. - This method returns true if doctor input overall information is correct. -This method returns false if doctor does not input some information or inputting is incorrect.	Patient patient	Boolean
4	updatePatient(Patient patient)	- This method is used to updating the data of patient from the database by using personal id, the data that input will be validated by AJAX validation. - The mechanism of this method involves other method: fineModel(int personal_id), validate(), and save() from framework. - This method returns true if doctor input the new overall information is correct. -This method returns false if doctor does not input some information or inputting is incorrect.	Patient patient	Boolean

5	deletePatient(Int personal_id)	- personal_id is personal id of patient from patient table. - This method is used to deleting the patient from the database by using delete() method from framework via user id. - The mechanism of this method involves other method: findModel(int personal_id) and delete() from framework. - This method returns true when delete personal id that exist on the user table. - This method will return false when delete personal id that does not exist on the user table.	Int personal_id	Boolean
6	searchPatient(String patients)	- This method is search query applied. - This method to get ActiveDataProvider instance which will filter models according to data in model fields. - This method returns data items currently available when input attributes that exist in the database.	String patients	ArrayList<Patient>
6	findModel(Int personal_id)	- personal_id is personal id of patient from patient table. - This method is used to finding the model based on its primary key value. - If the model is not found, a 404 HTTP exception will be thrown. - This method returns model object by personal id.	Int personal_id	Model

3.2.11 Class-11: Allergy



Description :

This is the Allergy model class for table "tbl_allergy". It is using to store the allergy report's information. This model class manages and invokes the allergy report.

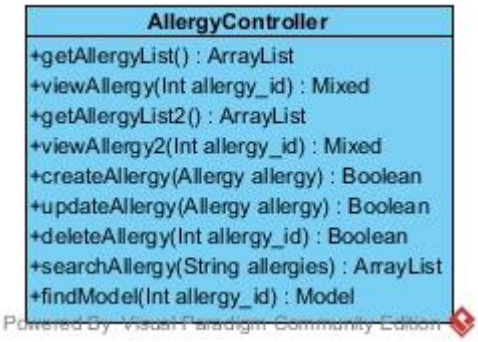
Attributes:

No.	Name	Description	Type
1	$allergy_id	This attribute is id of allergy report	Int
2	$personal_id	This attribute is personal id of the patient.	Int
3	$name	This attribute is name of the patient.	String
4	$surname	This attribute is surname of patients.	String
5	$allergy_drug	This attribute is information of allergy report	String
5	$reporter	This attribute is name of doctor who are reporter.	String
6	$create_at	This attribute is date time of report that doctor created.	Date Time

Methods :

No.	Name	Description	Parameter Lists	Return Type
1	validateIdCard()	- This method is using for validate the personal id. It invoked in form validator via Ajax validation.	-	Void

3.2.12 Class-12: AllergyController



Description :

This class use for controls and manages (CRUD) the information of the allergy and some patients. This class controls the allergy report's information by doctor.

Methods :

No.	Name	Description	Parameter Lists	Return Type
1	getAllergyList()	- This method is used to providing the list all allergy reports. - This method is used to display over all list of report that exist on the database. - This method using for role type doctor.	-	ArrayList<Allergy>
2	viewAllergy(Int allergy_id)	- allergy_id is allergy id of allergy report from allergy table. - This method used for view the single allergy report which includes allergy id, personal id, name, surname, allergy drug, reporter and create date to display on the 'view' page by using allergy id of report. - The mechanism of this method involves other method: findModel(int id). - This method returns allergy report when allergy id that exist on the allergy table via user model.	Int allergy_id	Mixed

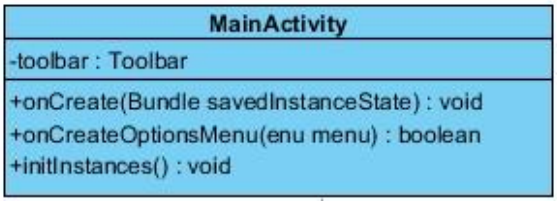
		- This method returns false if the allergy id that does not exist on the allergy table. - This method is using for role type doctor.		
3	getAllergyList2()	- This method is used to providing the list all allergy reports. - This method is used to display over all list of report that exist on the database. - This method using for role type FDA.	-	ArrayList<Allergy>
4	viewAllergy2(Int allergy_id)	- allergy_id is allergy id of allergy report from allergy table. - This method used for view the single allergy report which includes allergy id, personal id, name, surname, allergy drug, reporter and create date to display on the 'view' page by using allergy id of report. - The mechanism of this method involves other method: findModel(int id). - This method returns allergy report when allergy id that exist on the allergy table. - This method returns false if the allergy id that does not exist on the allergy table. - This method is using for role type FDA.	Int allergy_id	Mixed

5	createAllergy(Allergy allergy)	- This method is used to creating the new data of allergy report into database, the data that input to create will be validated by AJAX validation. - The mechanism of this method involves other method: validate() and save() from framework. - This method returns true if doctor input overall information is correct. -This method returns false if doctor does not input some information or inputting is incorrect.	Allergy allergy	Boolean
6	updateAllergy(Allergy allergy)	- This method is used to updating the data of allergy report from the database by using allergy id, the data that input will be validated by AJAX validation. - The mechanism of this method involves other method: fineModel(int allergy_id), validate(), and save() from framework. - This method returns true if doctor input the new overall information is correct. -This method returns false if doctor does not input some information or inputting is incorrect.	Allergy allergy	Boolean

7	deleteAllergy (Int allergy_id)	- allergy_id is allergy id of allergy report from allergy table. - This method is used to deleting the allergy report from the database by using delete() method from framework via user id. - The mechanism of this method involves other method: findModel(int allergy_id) and delete() from framework. - This method returns true when delete allergy id that exist on the allergy table. - This method will return false when delete allergy id that does not exist on the allergy table.	Int allergy_id	Boolean
8	searchAllergy(String allergies)	- This method is search query applied. - This method to get ActiveDataProvider instance which will filter models according to data in model fields. - This method returns data items currently available when input attributes that exist in the database.	String allergies	ArrayList
9	findModel(Int allergy_id)	- allergy_id is allergy id of allergy report from allergy table. - This method is used to finding the model based on its primary key value. - If the model is not found, a 404 HTTP exception will be thrown. - This method returns model object by allergy_id.	Int allergy_id	Model

(Mobile application)

3.2.13 Class-13: MainActivity



Description:

This class is container uses for control and manages the MainFragment class.

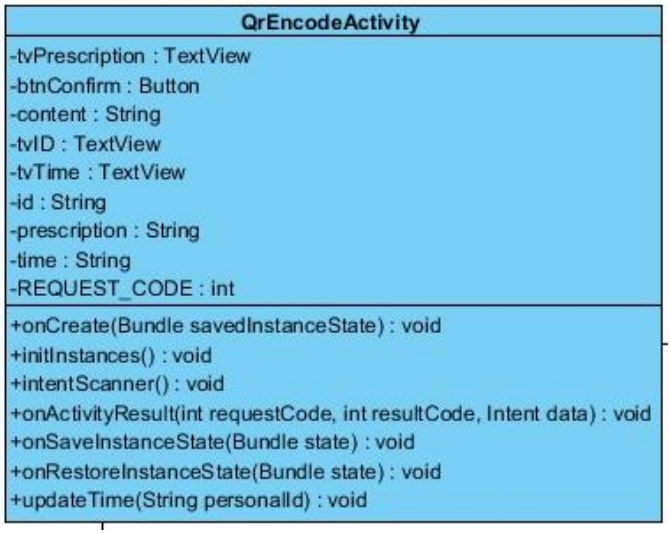
Attributes:

No.	Name	Description	Type
1	toolbar	Display a toolbar of the main page.	Toolbar

Methods:

No.	Name	Description	Parameter Lists	Return Type
1	onCreate(bundle savedInstanceState)	This method is use for re-creating activity on the MainFragment called when the activity is starting.	Bundle savedInstance State	void
2	onCreateOptionsMenu()	This method is use to callback method after user click menu on the MainFragment. - This method returns true when user select button on menu bar. - This method will return false when user do not select button on menu bar.	Menu menu	boolean
3	initInstances()	This method is use to handle for click the button on the MainFragment.	-	void

3.3.14 Class-14: QrEncodeActivity



Description:

This class is uses for control the scan QR code function and manage the update time function by pharmacist side.

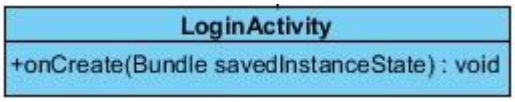
Attributes:

No.	Name	Description	Type
1	tvPrescriptionDetail	Display the text editor of the prescription detail.	TextView
2	tvId	Display the text editor of the ID	TextView
3	tvTime	Display the text editor of the time	TextView
4	btnConfirm	Display the button for confirms.	Button
5	content	Stored the value of content.	String
6	id	Stored the value of id.	String
7	prescription	Stored the value of prescription.	String
8	time	Stored the value of time.	String

Methods:

No.	Name	Description	Parameter Lists	Return Type
1	onCreate(Bundle savedInstanceState)	This method is use for re-creating activity called when the activity is starting.	Bundle savedInstanceState	void
2	initInstances()	This method is use to handle for click the button.	-	void
3	intentScanner()	This operation is use for call to use scanner activity.	-	void
4	onActivityResult()	This method is use for receive the requestCode call when activity launched exits.	Int requestCode Int resultCode Intent data	void
5	onSaveInstanceState()	This method is retrieve per-instance state.	Bundle state	void
6	onRestoreInstanceState()	This method is use for receive the SaveInstanceState	Bundle state	void
7	updateTime()	This method is updates the time value to the personal ID column with.	String personalId	void

3.3.15 Class-15: LoginActivity



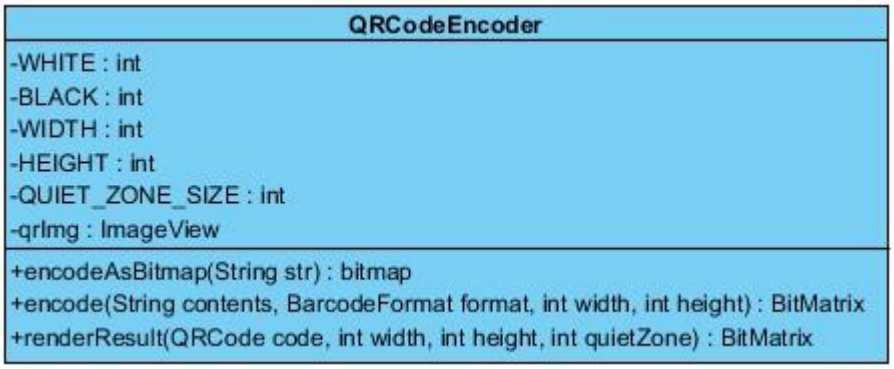
Description:

This class is container uses for control and manages LoginFragment class.

Methods:

No.	Name	Description	Parameter Lists	Return Type
1	onCreate(Bundle savedInstanceState)	This method is use for re-creating activity on the LoginFragment called when the activity is starting.	Bundle savedInstance State	void

3.3.16 Class-16: QRCodeEncoder



Description:

This class is model use for controls and manage the generate the QR code function of web service.

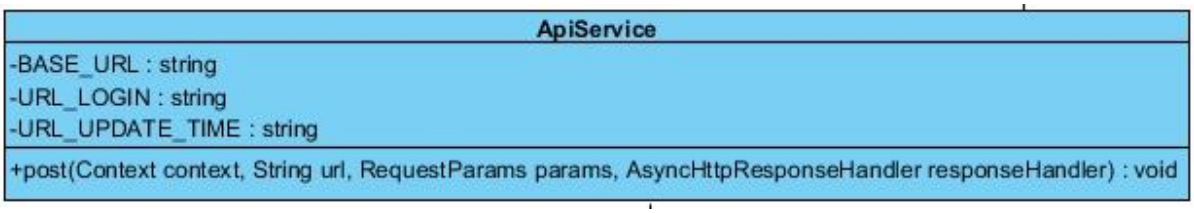
Attributes:

No.	Name	Description	Type
1	WHITE	This attribute use for stored the amount of white box on a QR code.	Int
2	BLACK	This attribute use for stored the amount of black box on a QR code.	Int
3	WIDTH	This attribute use for stored the amount of width on a QR code.	Int
4	HEIGHT	This attribute use for stored the amount of height on a QR code.	Int
5	QUIET_ZONE_SIZE	This attribute use for stored the amount of size on a QR code.	Int
6	qrImg	This attribute use for stored the image of QR code.	ImageView

Methods:

No.	Name	Description	Parameter Lists	Return Type
1	encodeAsBitmap()	This method use for create the bitmap.	String str	bitmap
2	encode()	This method use for encode a QR code.	String contents, BarcodeFormat format, Int width, Int height	BitMatrrix
3	renderResult()	This method is render the QR code.	QRCode code, Int width, Int height, Int quietZone	BitMatrrix

3.3.17 Class-17: ApiService



Description:

This class use for connect with the API service.

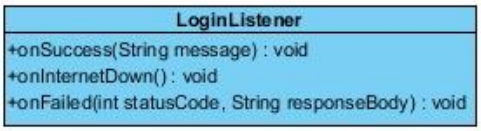
Attributes:

No.	Name	Description	Type
1	BASE_URL	This attribute use for connect the URL of IPv4 address .	String
2	URL_LOGIN	This attributes use for connect login.php of web service.	String
3	URL_UPDATE_TIME	This attributes use for connect update_prescription.php of web service.	String

Methods :

No.	Name	Description	Parameter Lists	Return Type
1	post()	This method use to request the server.	Context context, String url, RequestParams params, AsyncHttpResponseHandler responseHandler	void

3.3.18 Class-18: LoginListener



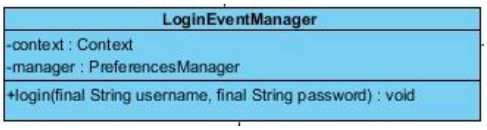
Description :

This class is pattern use when the application connects with the login API service.

Methods :

No.	Name	Description	Parameter Lists	Return Type
1	onSuccess()	This method is use for create a message call back to user after the activity successful.	String message	void
2	onInternetDown()	This method is call back to user after cannot the internet.	-	void
3	onFailed()	This method is use for create a message call back to user after the activity fail.	Int statusCode, String responseBody	void

3.3.19 Class-19 LoginEventManager



Description:

This class is model use for manage the login function connect with login API service.

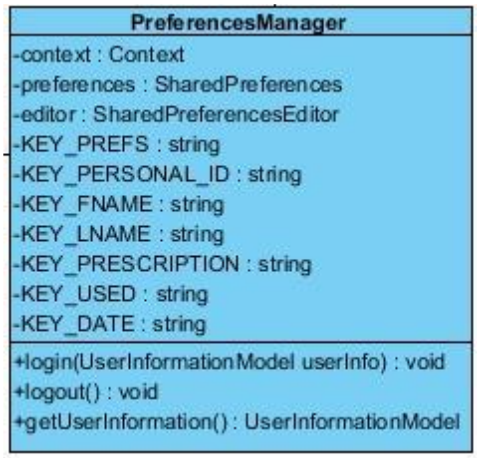
Attributes:

No.	Name	Description	Type
1	context	This attribute use for stored a data from API.	Context
2	manager	This attribute use for stored a PreferencesManager context.	PreferencesManager

Methods:

No.	Name	Description	Parameter Lists	Return Type
2	Login()	This method use for login to the system.	String username, String password,	void

3.3.20 Class-20 PreferencesManager



Description:

This class uses for stored the information of prescription.

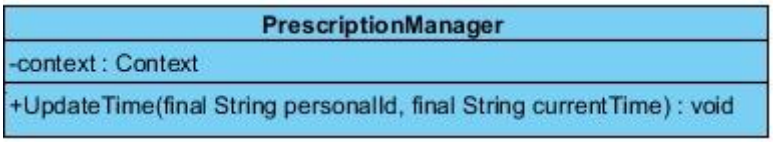
Attributes:

No.	Name	Description	Type
1	context	This attributes use for stored context.	Context
2	preference	This attributes use for stored preference.	SharedPreferences
3	editor	This attributes use for stored editor.	SharedPreferences.Editor
4	KEY_PREFS	This attributes use for stored mPrefs.	String
5	KEY_PERSONAL_ID	This attributes use for stored personal ID.	String
6	KEY_FNAME	This attributes use for stored name.	String
7	EY_LNAME	This attributes use for stored surname.	String
8	KEY_PRESCRIPTION	This attributes use for stored prescription.	String
9	KEY_USED	This attributes use for stored used	String
10	KEY_DATE	This attributes use for stored date.	String

Methods :

No.	Name	Description	Parameter Lists	Return Type
1	login(UserInformationModel userInfo)	This method uses for store the information of patient's profile after login successful.	UserInformationModel userInfo	void
2	logout()	This method uses for clear the value when user logout.	-	void
3	getUserInformation()	This method uses for get the value of UserInformationModel.	-	UserInformationModel

3.3.21 Class-21: PrescriptionManager



Description:

This class is model uses for connect with update_prescription API service.

Attributes:

No.	Name	Description	Type
1	context	This attributes use for stored context.	Context

Methods:

No.	Name	Description	Parameter Lists	Return Type
1	updateTime()	This method use for update current time to the prescription.	String personalId, String currenTtime	void

3.3.22 Class-22: UpdateTimeListener

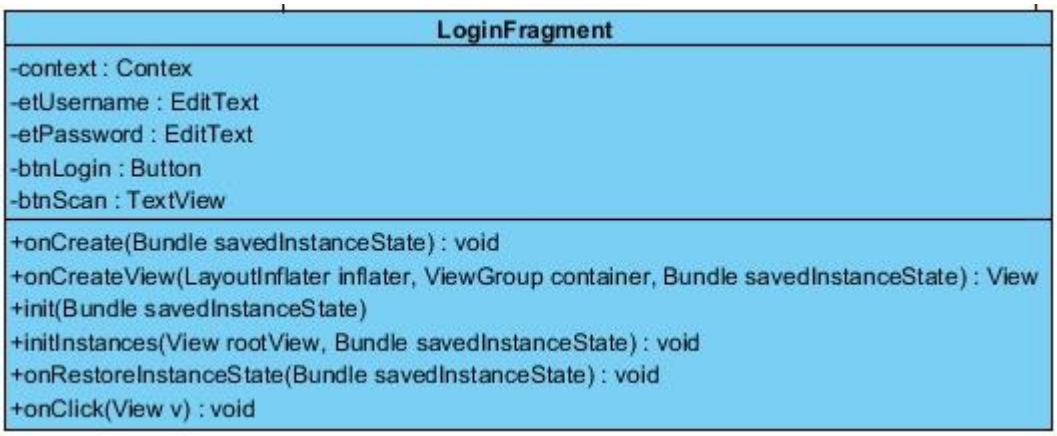
Description:

This class is pattern used when connect with the update_prescription API service.

Methods :

No.	Name	Description	Parameter Lists	Return Type
1	onInternetDown()	This method is interface for callback invocation on an application action when the internet failed.	-	void
2	onSuccess()	This method is interface for callback invocation on an application action when the operation failed.	String lastTime	void
3	onFailed()	This method is interface for callback invocation on an application action when the operation succeeded	Int statusCode, String responseBody	void

3.3.23 Class-23: LoginFragment



Description:

This class use for controls and manage interface of login page.

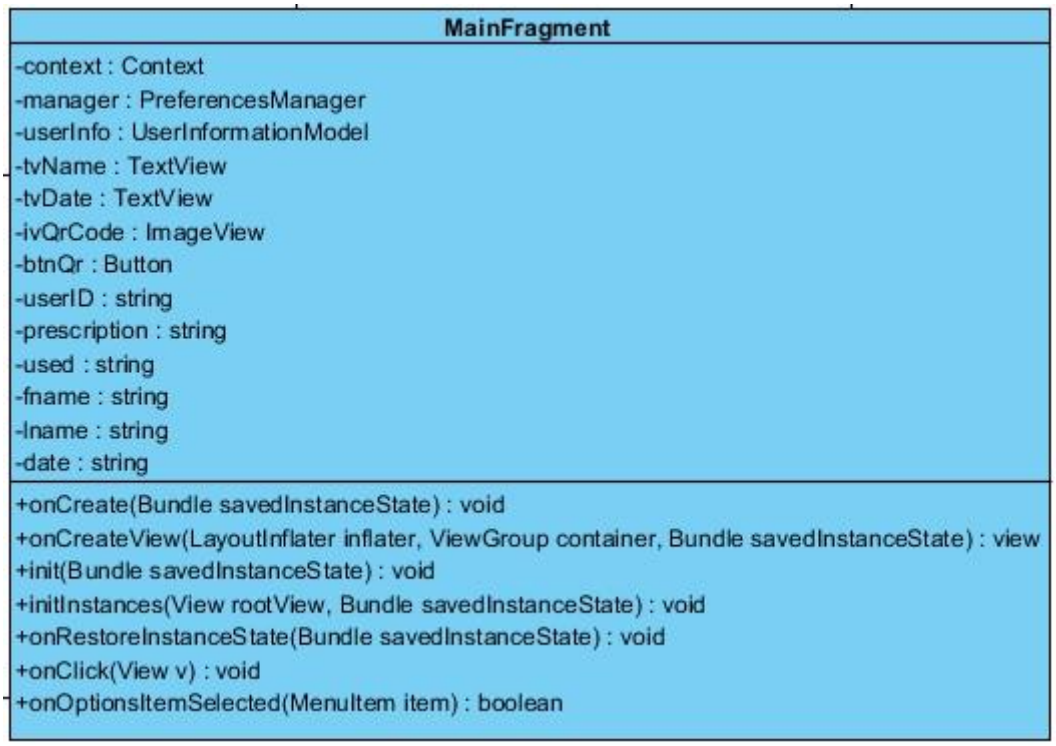
Attributes:

No.	Name	Description	Type
1	context	This attributes use for stored context.	Context
2	etUsername	This attributes use for stored username.	EditText
3	etPassword	This attributes use for stored password.	EditText
4	btnLogin	This attributes use for display login button.	Button
5	btnScan	This attributes use for display link scan code button.	TextView

Methods :

No.	Name	Description	Parameter Lists	Return Type
1	onCreate()	This method called to do initial creation of a LoginFragment.	Bundle savedInstanceState	void
2	onCreateView()	This method called to have the LoginFragment instantiate its user interface view.	LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState	view
3	init()	This method use for set menu item to show in action bar.	-	void
4	initInstances()	This method uses for set the component.	-	void
5	onRestoreInstanceState()	This method uses for restore the value from state.	-	void
6	onClick()	This method is call when a view have been clicked	View v	void

3.3.24 Class-24: MainFragment



Description :

This class use for controls and manage interface of main page.

Attributes:

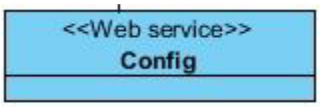
No.	Name	Description	Type
1	context	This attribute use for stored a context.	Context
2	manager	This attribute use for stored a manager.	PreferencesManager
3	btnQr	This attribute use for control a button	Button
4	TextView	This attribute use for stored a TextView.	UserInformationModel
5	tvName	This attribute use for display a tvName.	TextView
6	tvDate	This attribute use for display a tvDate.	TextView
7	ivQrCode	This attributes use for display a QrCode.	ImageView
8	userID	This attributes use for stored personal ID.	String
9	prescription	This attributes use for stored prescription.	String
10	used	This attribute use for stored used.	String

11	fname	This attributes use for stored name.	String
12	lname	This attribute use for stored surname.	String

Methods :

No.	Name	Description	Parameter Lists	Return Type
1	onCreate()	This method use for create a new activity when the application is starting.	Bundle savedInstanceState	void
2	onCreateView()	This method use for define the layout of MainFragment.	LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState	view
3	init()	This method use for set menu item to show in action bar.	-	void
4	initInstances()	This method uses for set the component.	-	void
5	onRestoreInstanceState()	This method uses for restore the value from state.	Bundle savedInstanceState	void
6	onClick()	This method is call when a view have been clicked	View v	void
7	onOptionsItemSelected()	This method uses for control logout system. - This method returns true when user select logout button. - This method will return false when user do not select logout button.	MenuItem item	boolean

3.3.25 Class-25: Config (web service)



Description :

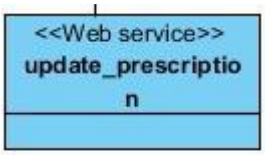
This class is web service use for connector between ApiService class and database.

3.3.26 Class-26: login (web service)

Description :

This class is web service use for connector between LoginEventManager class and database.

3.3.27 Class-27: update_prescription (web service)

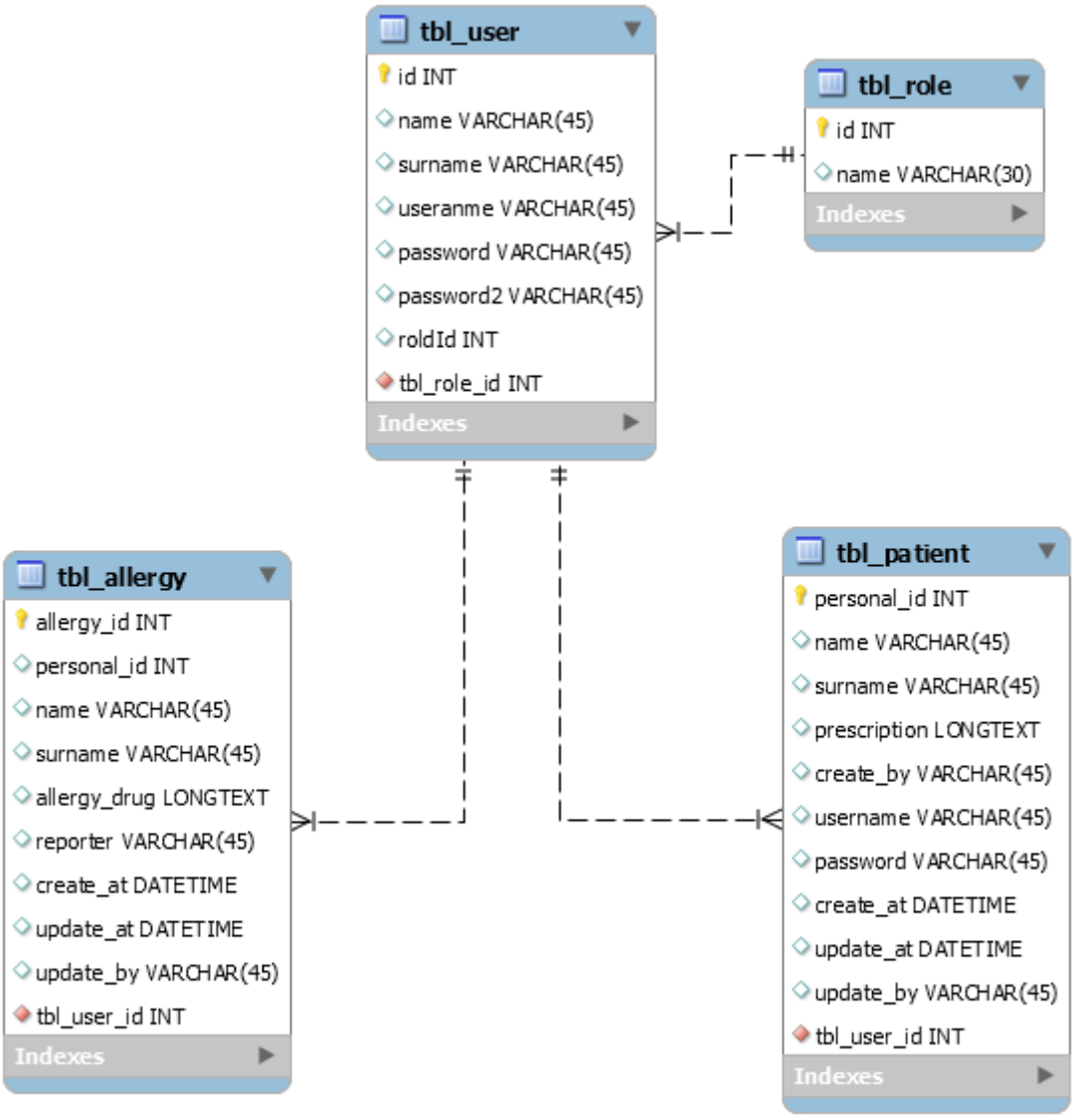


Description :

This class is web service use for connector between PrescriptionManager class and database.

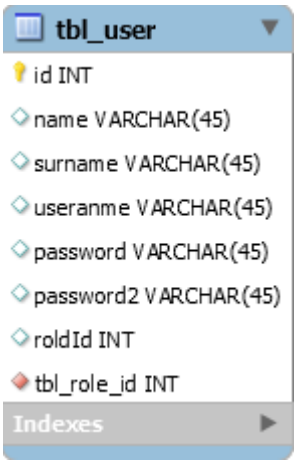
Chapter Four | Database Design

4.1 ER Diagram



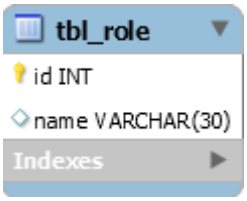
4.2 Table Description

4.2.1 User Table



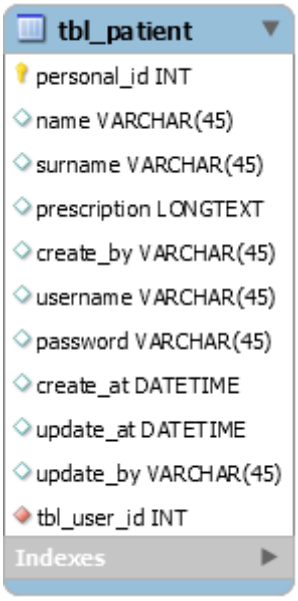
No.	Field Name	Type	Length	PK/FK
1	id	int	10	PK
2	name	varchar	45	-
3	surname	varchar	45	-
4	username	varchar	45	-
5	password	varchar	45	-
6	password2	varchar	45	-
7	roleId	int	2	FK

4.2.2 Role Table



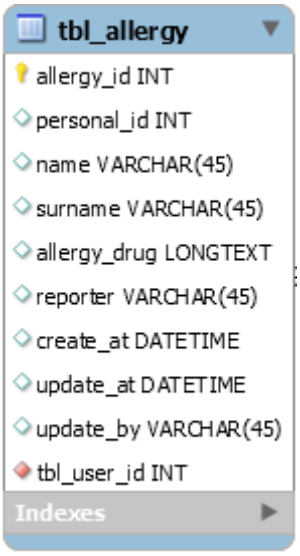
No.	Field Name	Type	Length	PK/FK
1	id	tinyint	1	PK
2	name	varchar	30	-

4.2.3 Patient Table



No.	Field Name	Type	Length	PK/FK
1	personal_id	int	17	PK
2	name	varchar	45	-
3	surname	varchar	45	-
4	prescription	longtext	-	-
5	create_by	varchar	45	-
4	username	varchar	45	-
6	password	varchar	45	-
7	create_at	datetime	-	-
8	update_at	datetime	-	-
9	update_by	varchar	45	-
10	userId	int	10	FK

4.2.4 Allergy Table



No.	Field Name	Type	Length	PK/FK
1	allergy_id	int	10	PK
2	personal_id	int	17	-
3	name	varchar	45	-
4	surname	varchar	45	-
5	username	varchar	45	-
6	allergy_drug	longtext	-	-
7	reporter	varchar	45	-
8	create_at	datetime	-	-
9	update_at	datetime	-	-
10	update_by	varchar	45	-
11	userId	int	10	FK

Chapter Five | Sequence Diagram

SD-01: Doctors can view a doctor home page on the web application.

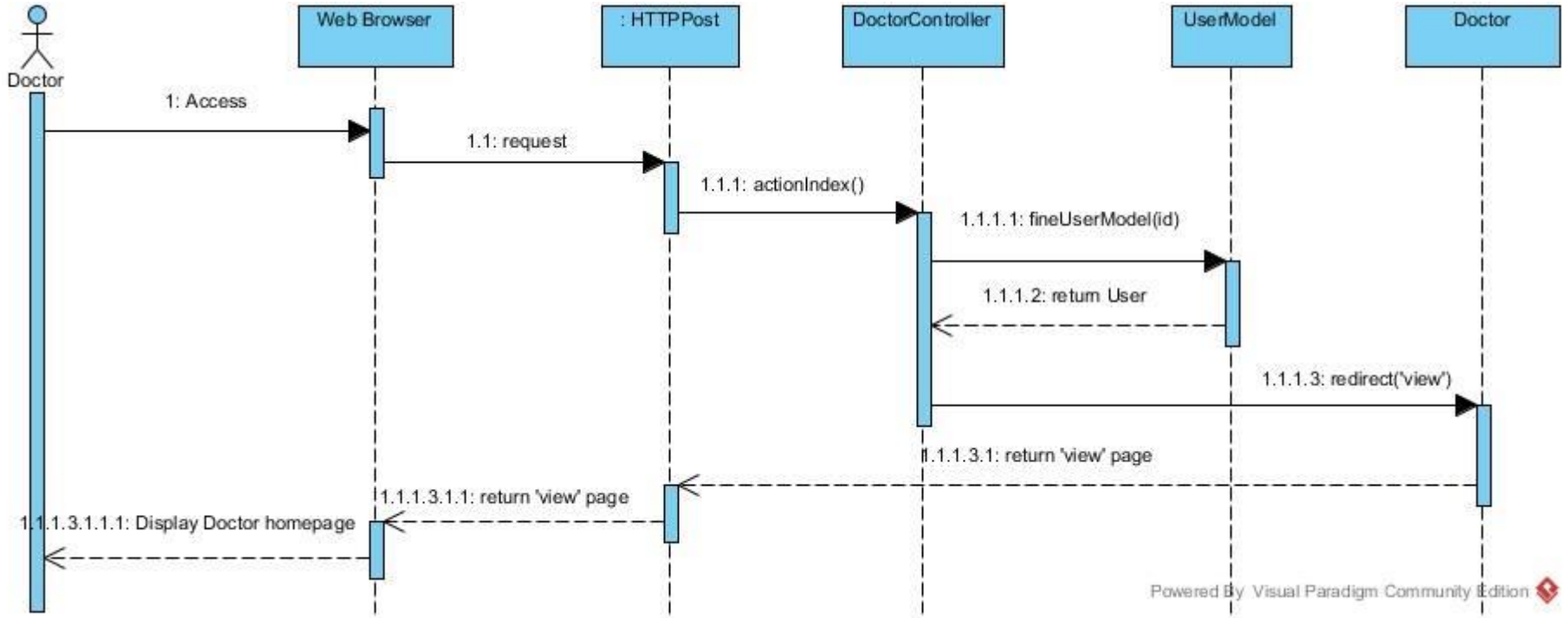


Figure 1: SD-01 Doctors can view a doctor home page on the web application.

SD-02: Doctors can create a patient's profile on the web application.

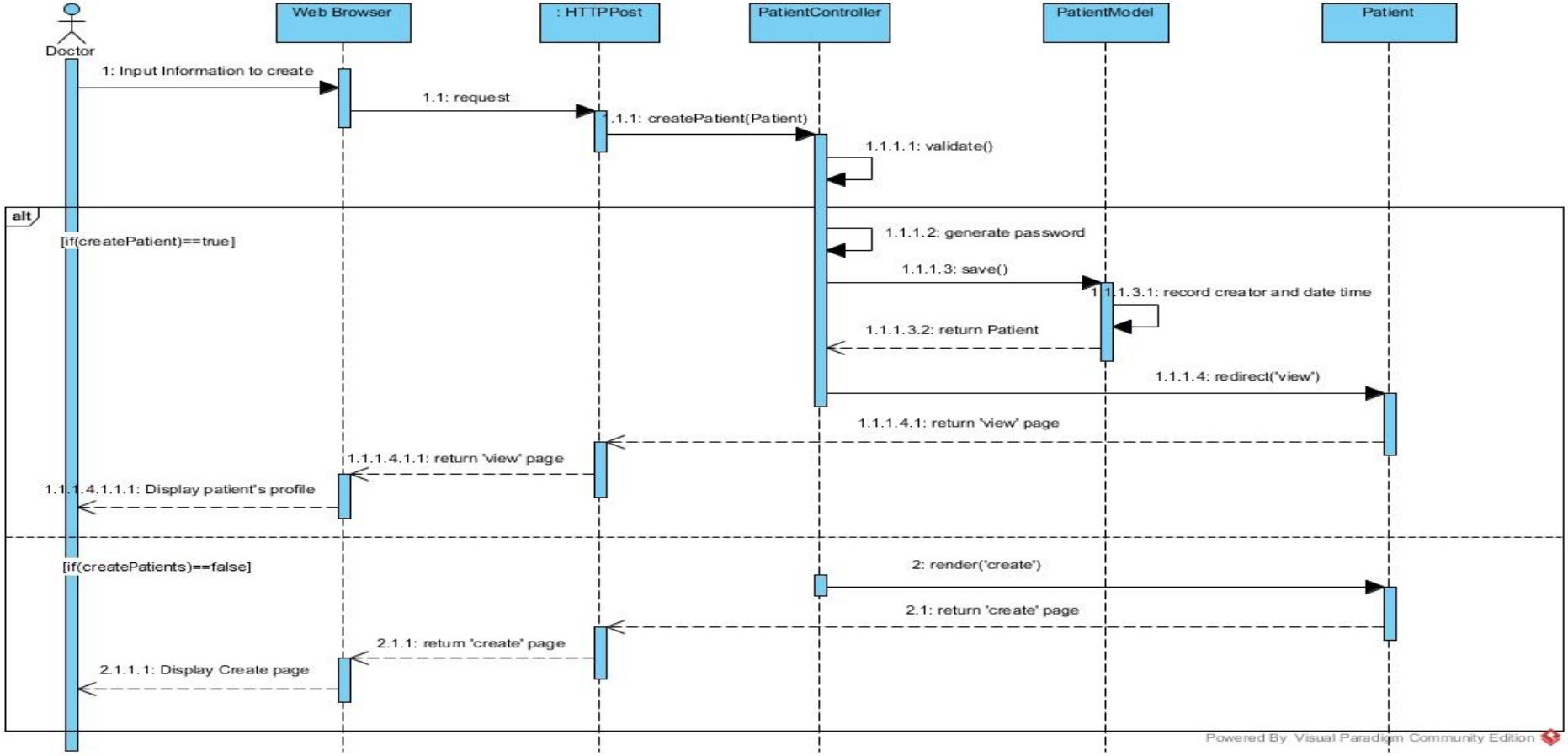


Figure 2: SD-02 Doctors can create a patient's profile on the web application.

SD-03: Doctors can update a patient's profile on the web application.

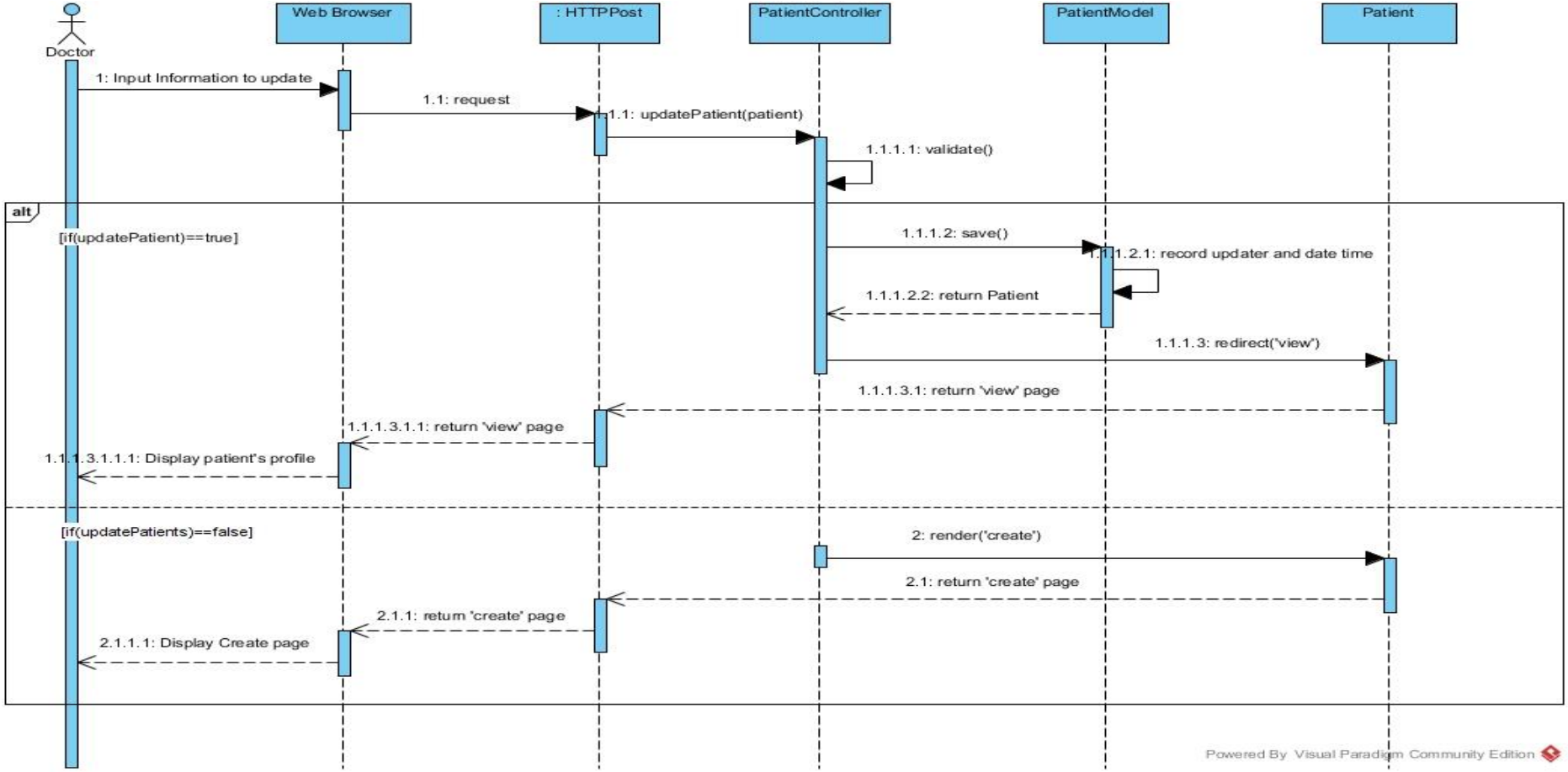


Figure 3: SD-03 Doctors can update a patient's profile on the web application.

SD-04: Doctors can delete a patient's profile on the web application.

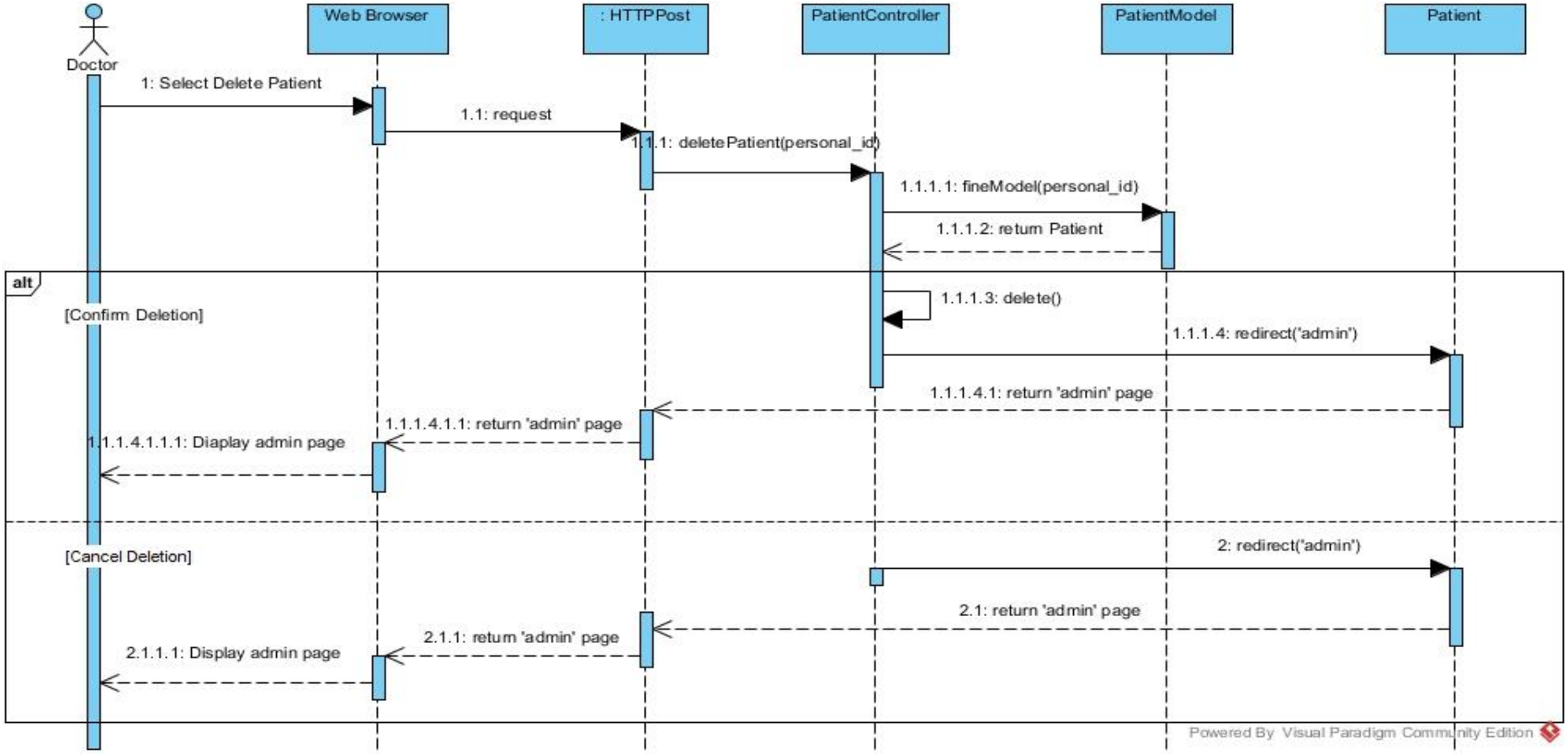


Figure 4: SD-04 Doctors can delete a patient's profile on the web application.

SD-05: Doctors can search a patient's profile on the web application.

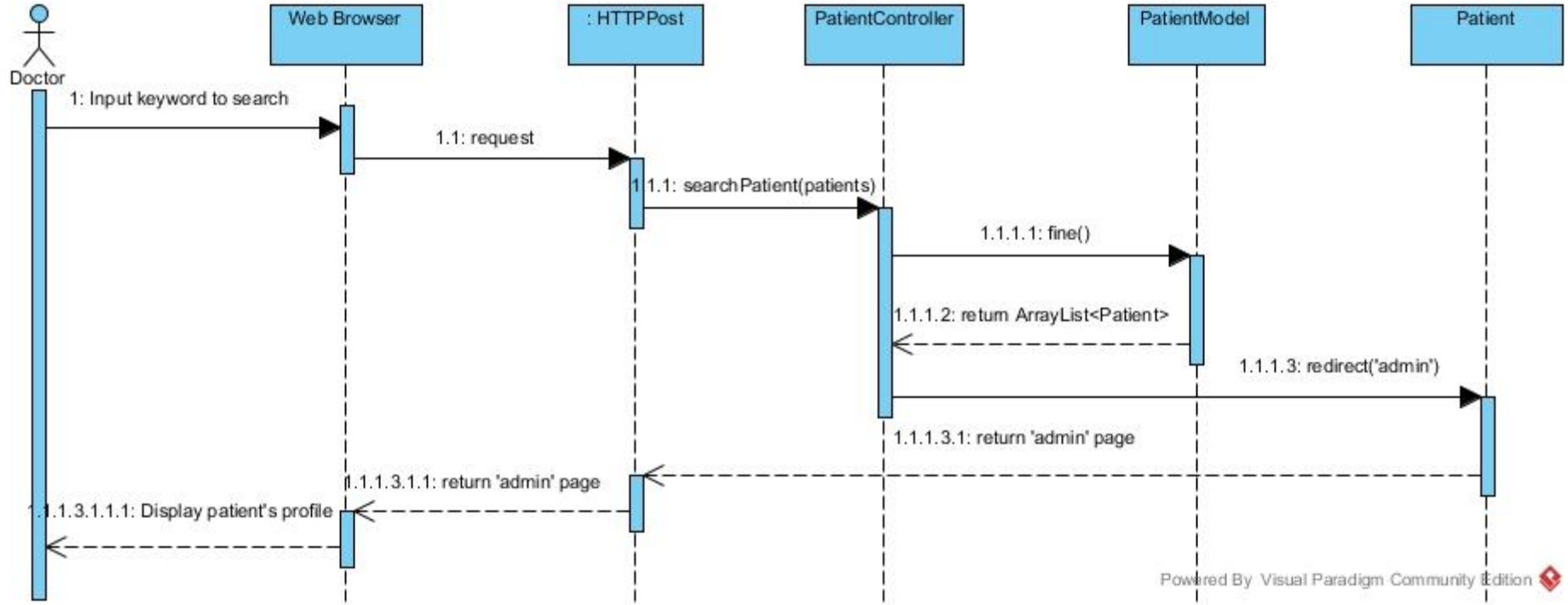


Figure 5: SD-05 Doctors can search a patient's profile on the web application.

SD-06: Doctors can view a patient's profile on the web application.

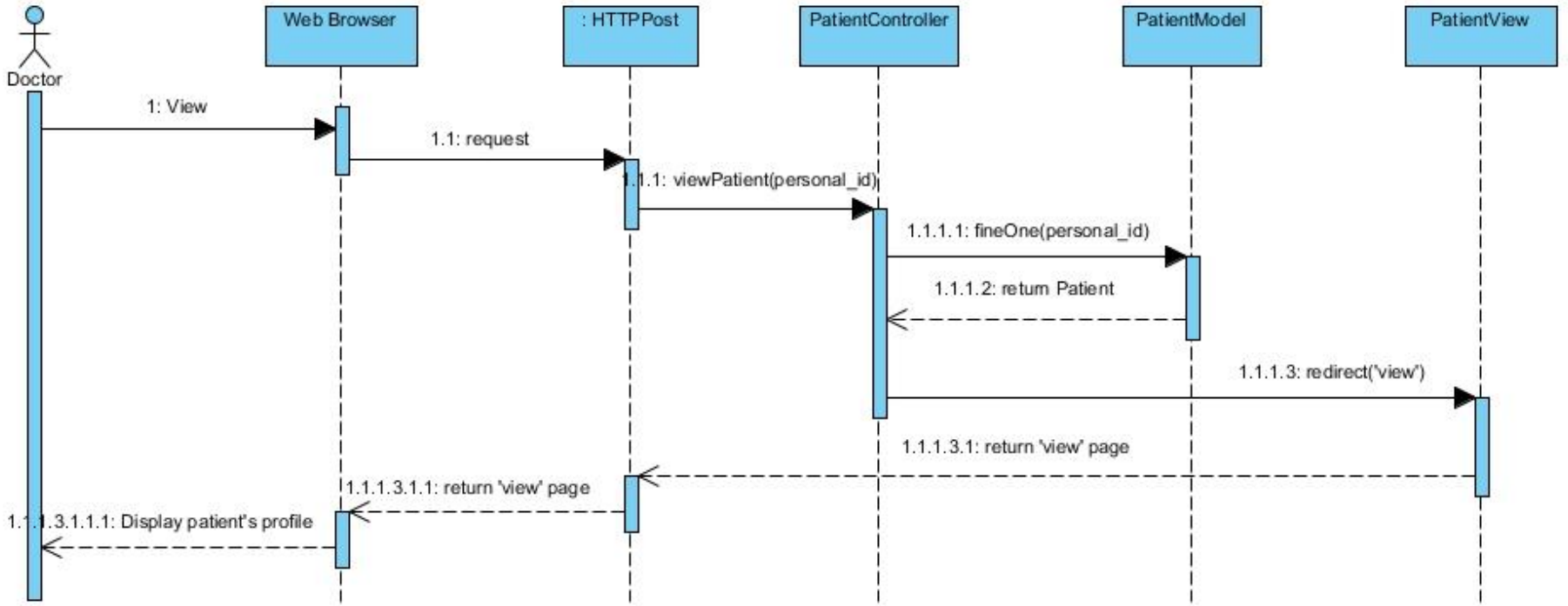


Figure 6: SD-06 Doctors can view a patient's profile on the web application.

SD-07: Doctors can view a list of patient's profile on the web application.

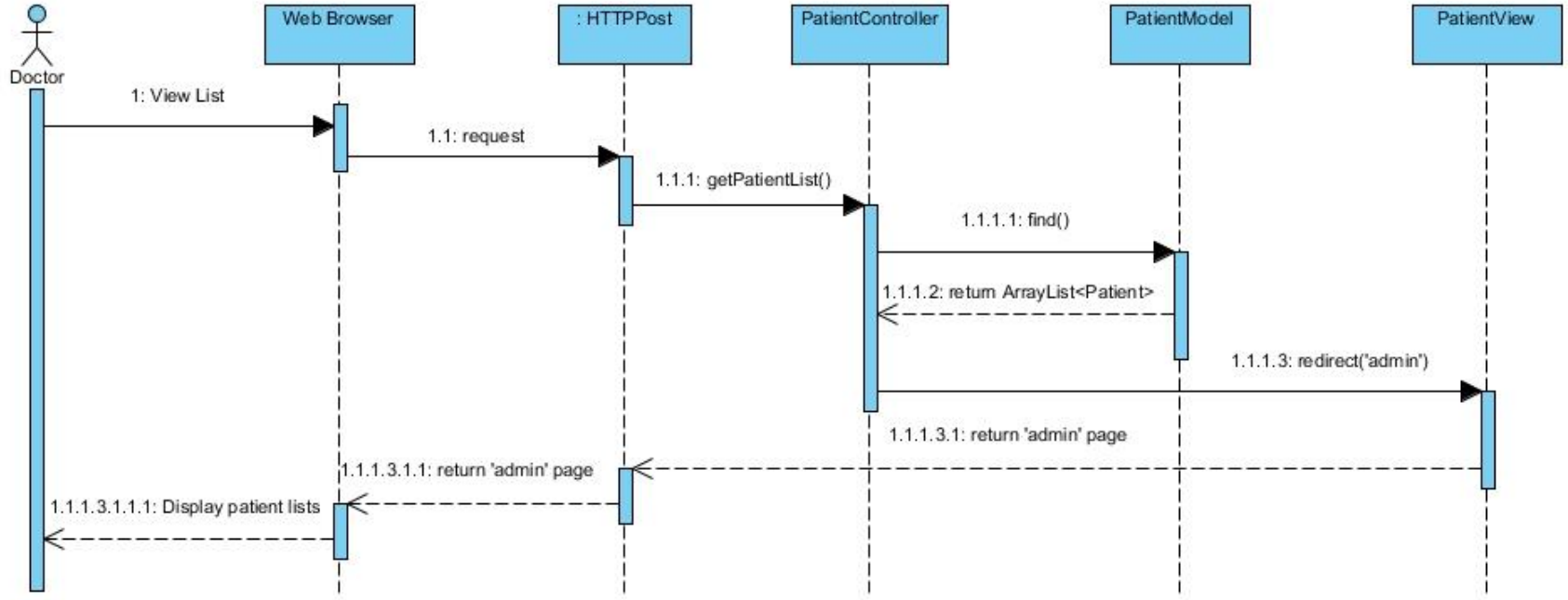


Figure 7: SD-07 Doctors can view a list of patient's profile on the web application.

SD-08: Administrators can view an admin home page on the web application.

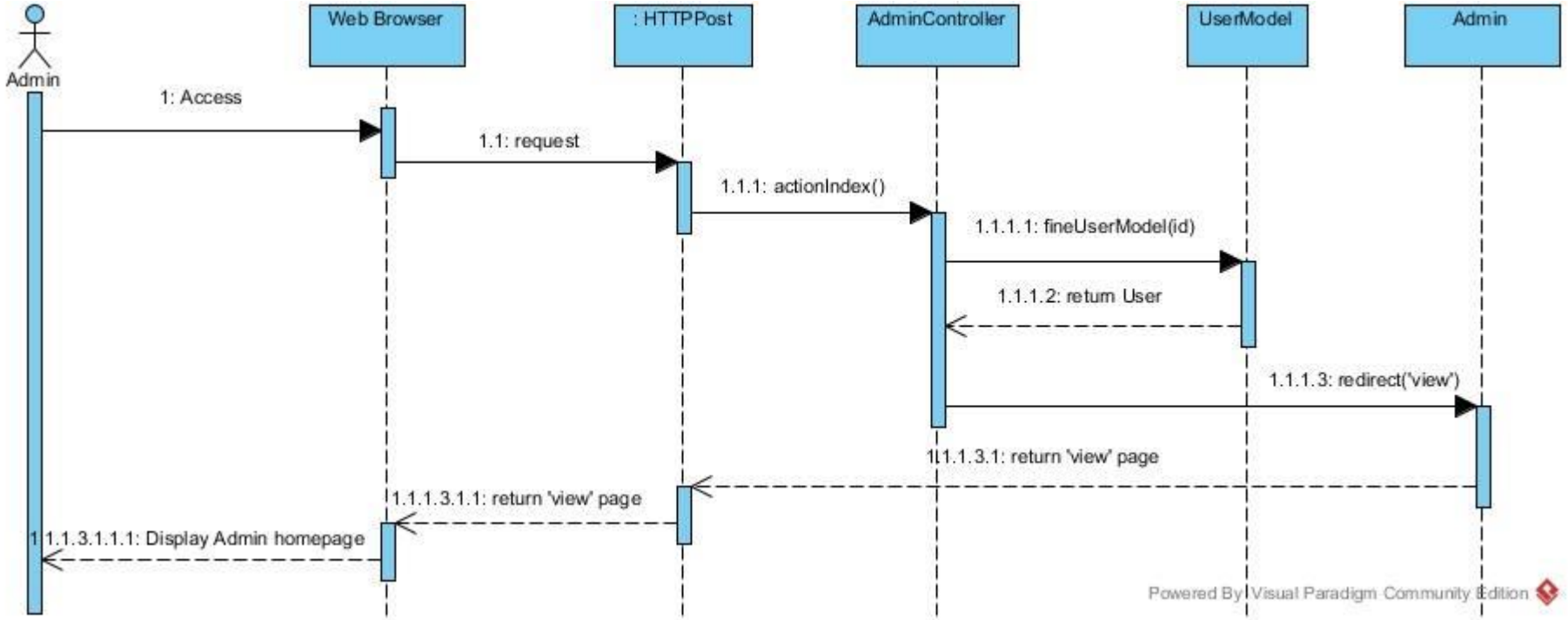


Figure 8: SD-08 Administrators can view an admin home page on the web application.

SD-09: Administrators can create a user's profile on the web application.

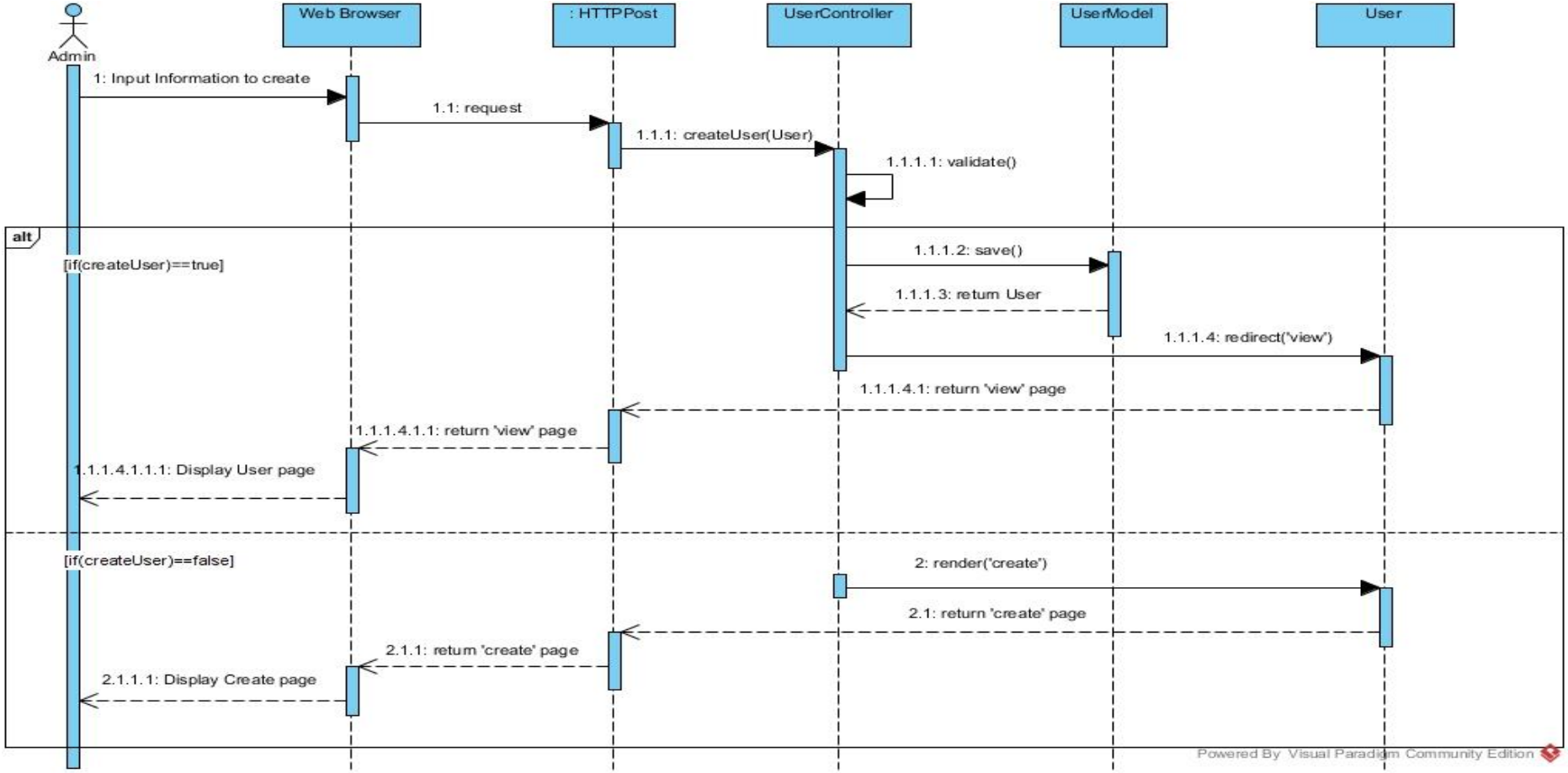


Figure 9: SD-09 Administrators can create a user's profile on the web application.

SD-10: Administrators can update a user's profile on the web application.

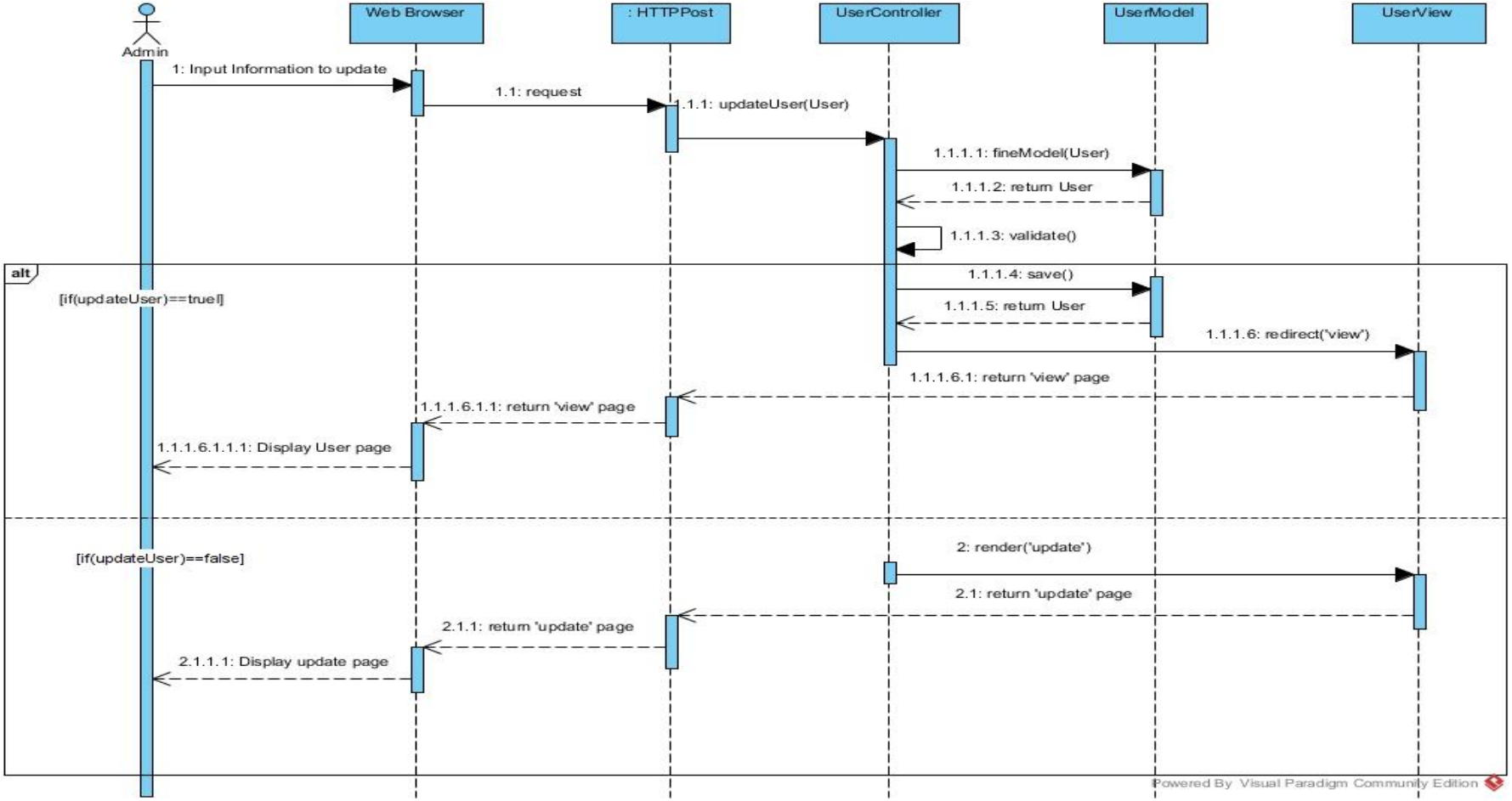


Figure 10: SD-10 Administrators can update a user's profile on the web application.

SD-11: Administrators can delete a user's profile on the web application.

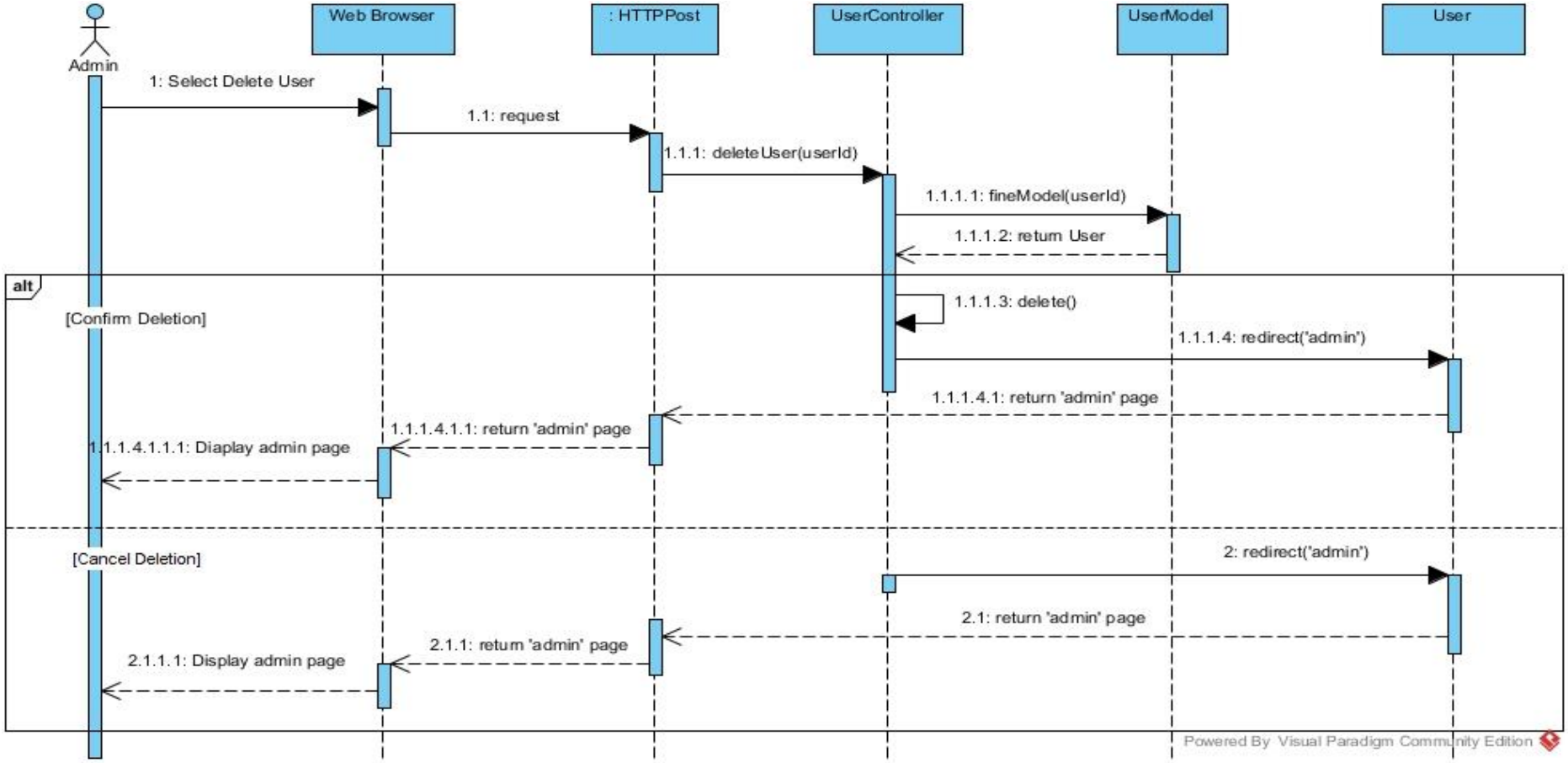


Figure 11: SD-11 Administrators can delete a user's profile on the web application.

SD-12: Administrators can search a user's profile on the web application.

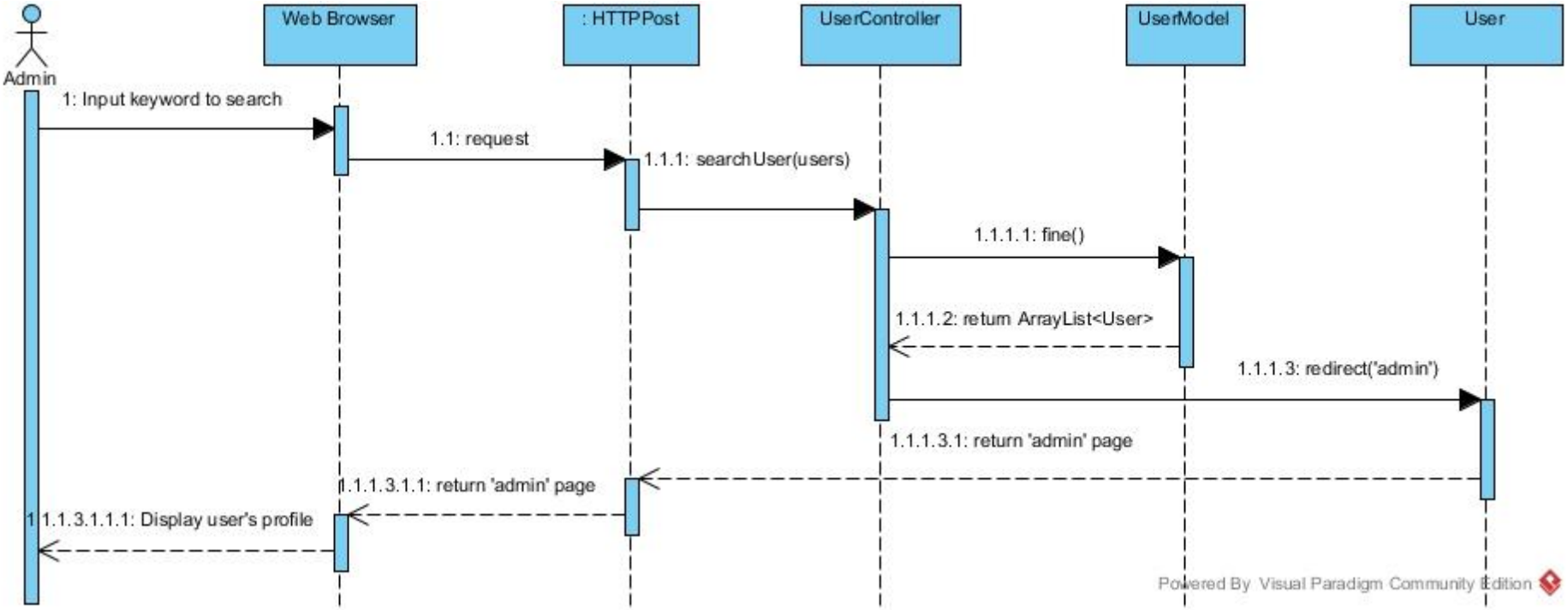


Figure 12: SD-12 Administrators can search a user's profile on the web application.

SD-13: Administrators can view a user's profile on the web application.

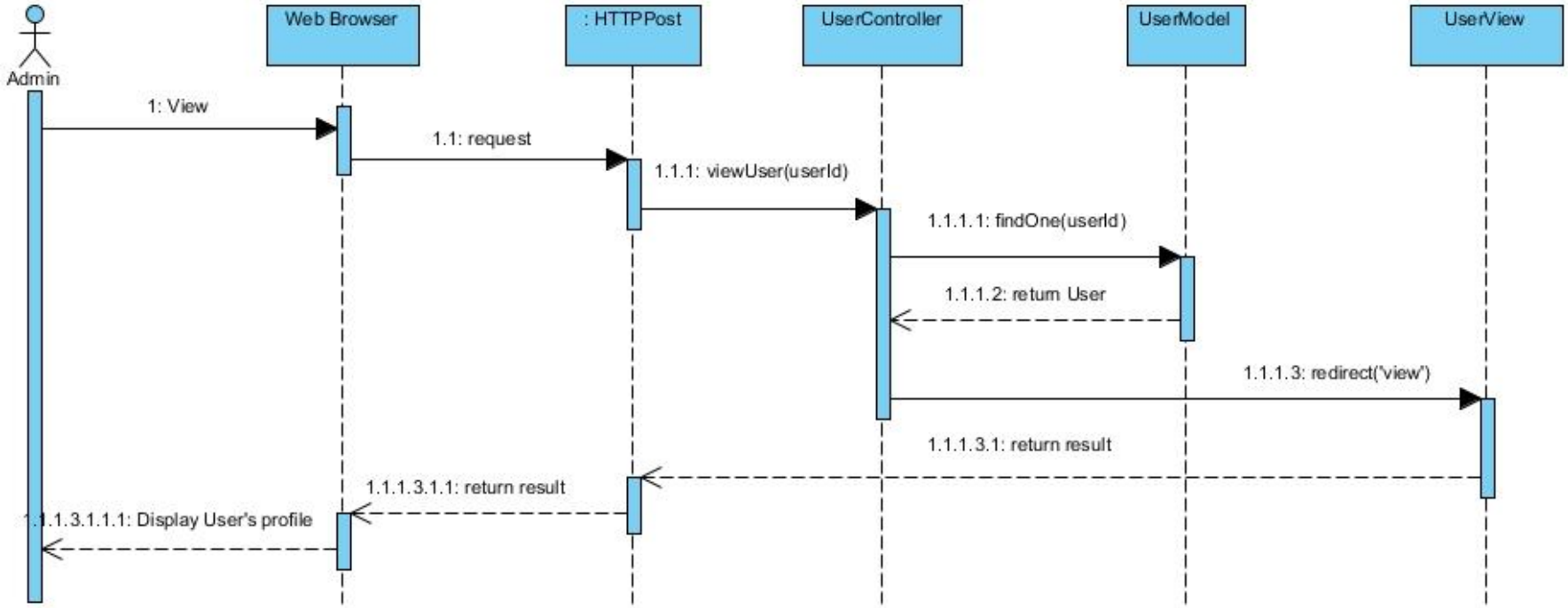


Figure 13: SD-13 Administrators can view a user's profile on the web application.

SD-14: Administrators can view a list of user's profile on the web application.

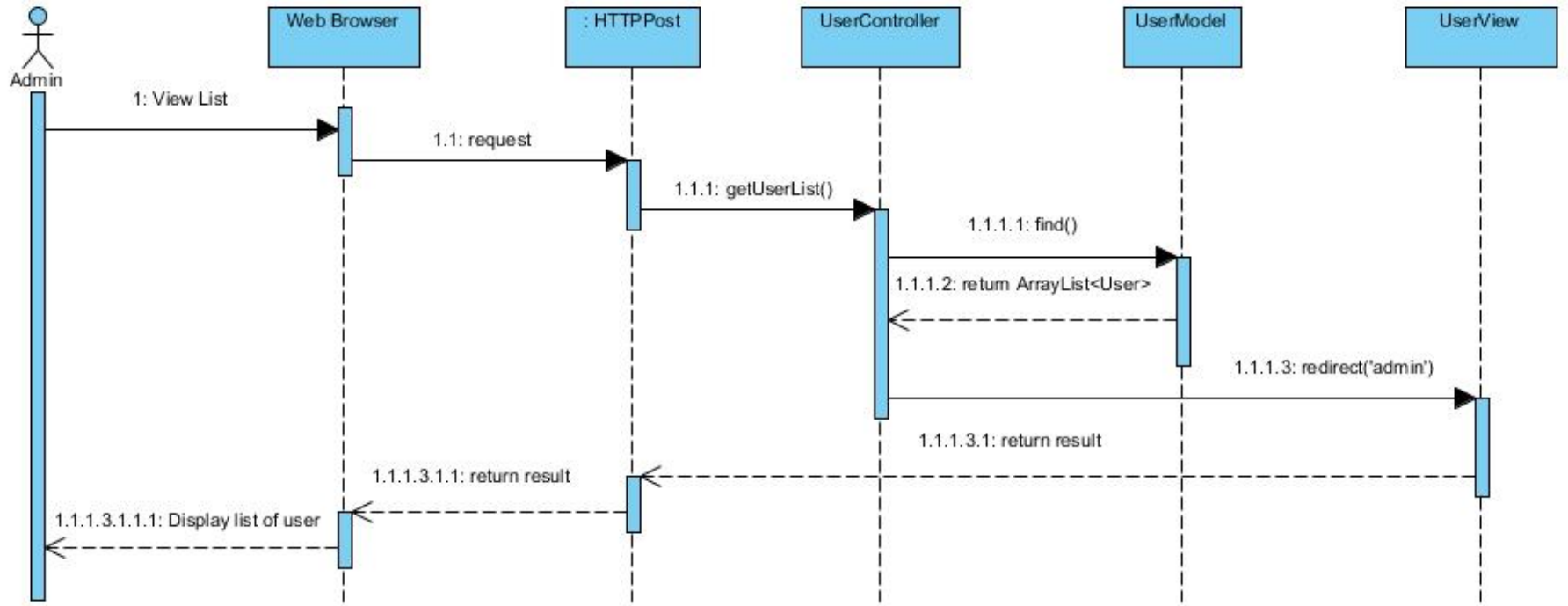


Figure 14: SD-14 Administrators can view a list of user's profile on the web application.

SD-15: Doctors, Administrators, and FDAs can login to the web application.

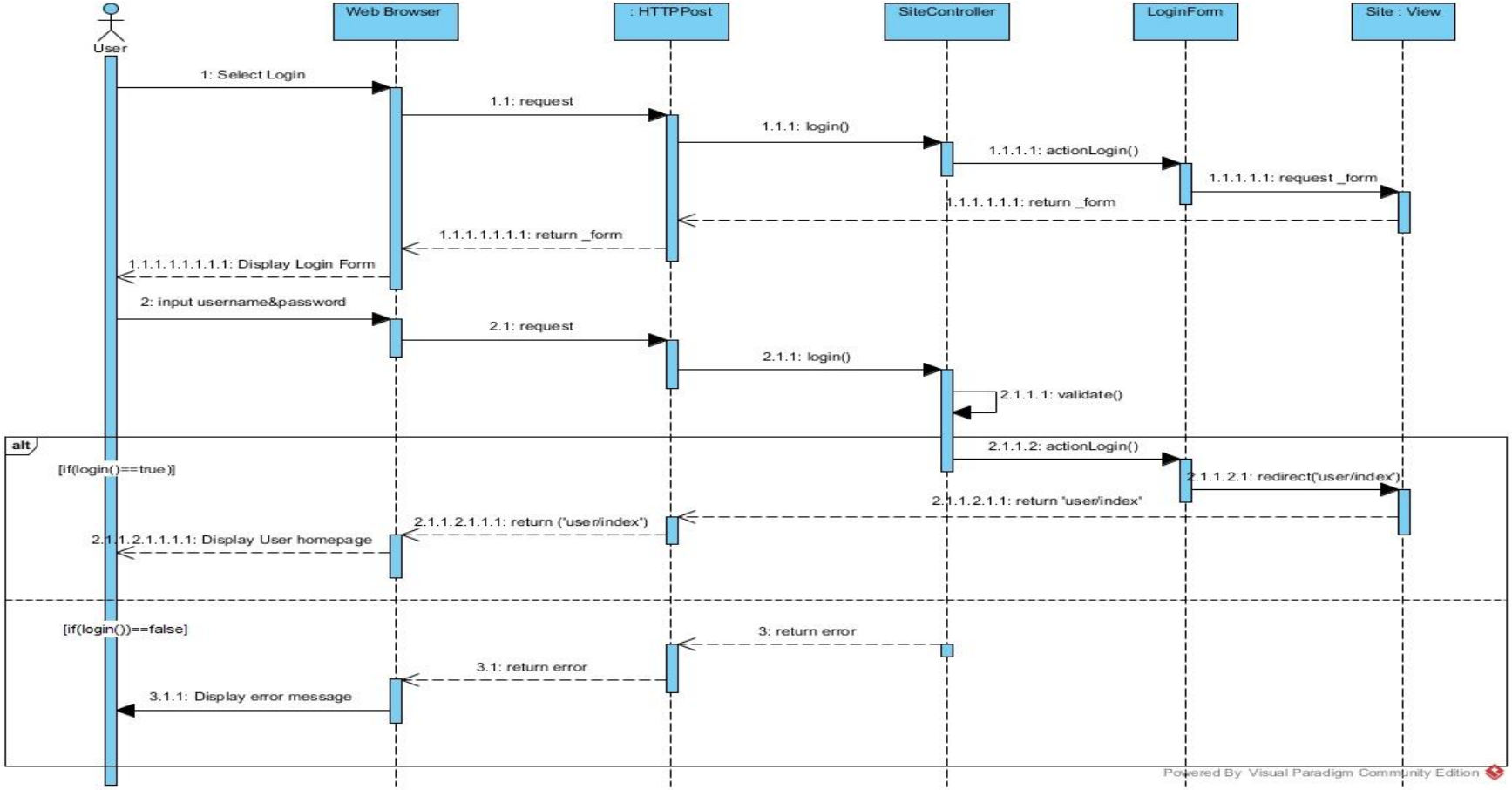


Figure 15: SD-15 Doctors, Administrators, and FDAs can login to the web application.

SD-16: Doctors, Administrators, and FDAs can logout from the web application.

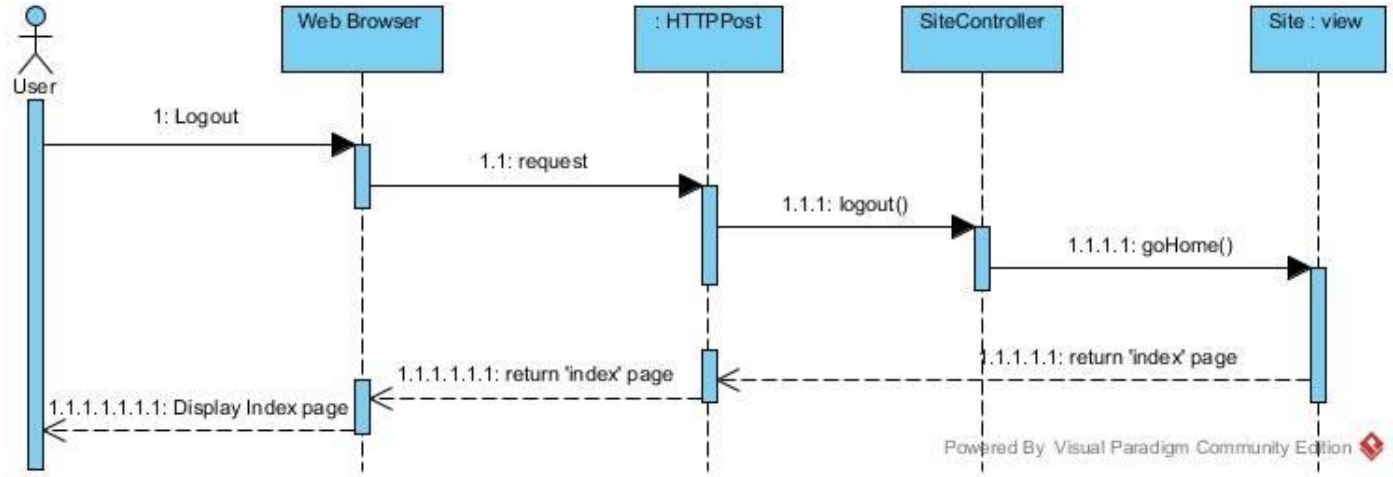


Figure 16: SD-16 Doctors, Administrators, and FDAs can logout from the web application.

SD-17: Patients can login to the mobile application.

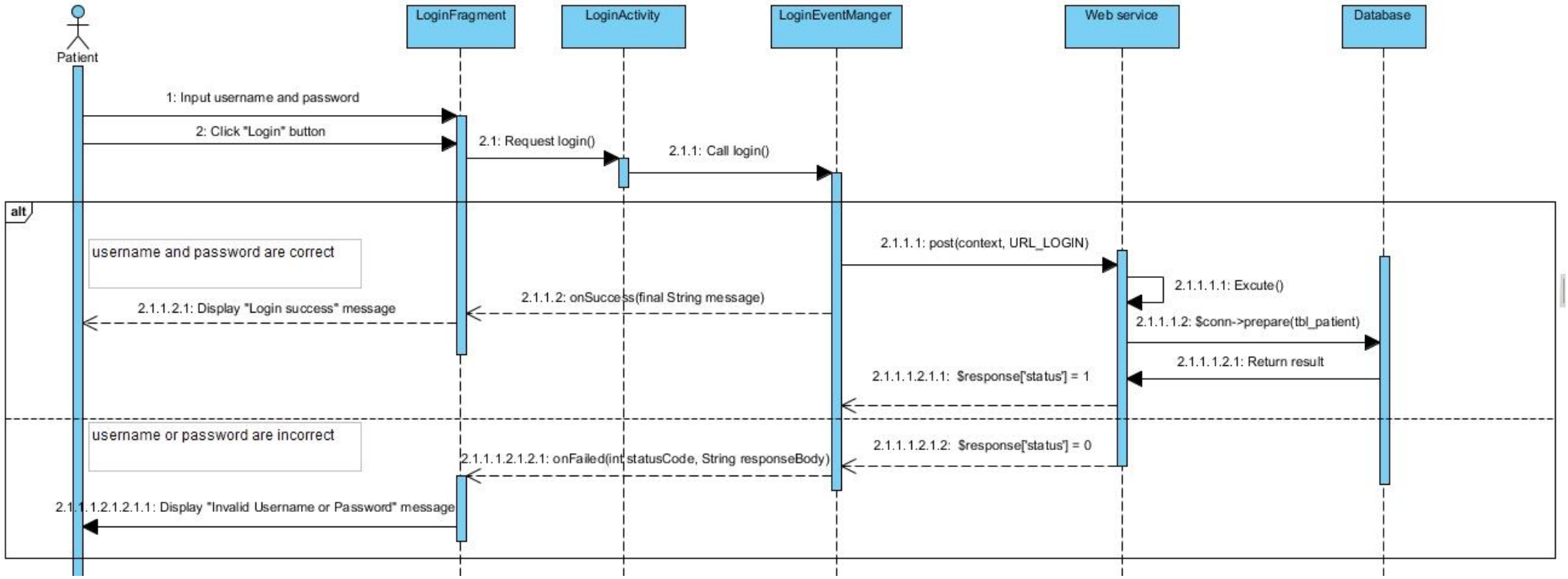


Figure 17: SD-17 Patients can login to the mobile application.

SD-18: Patients can logout from the mobile application.

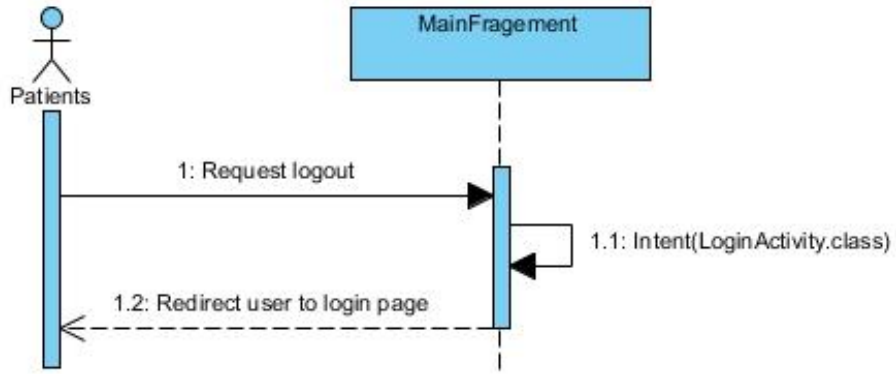


Figure 18: Patients can logout from the mobile application.

SD-19: Patients can view the QR code on the mobile application.

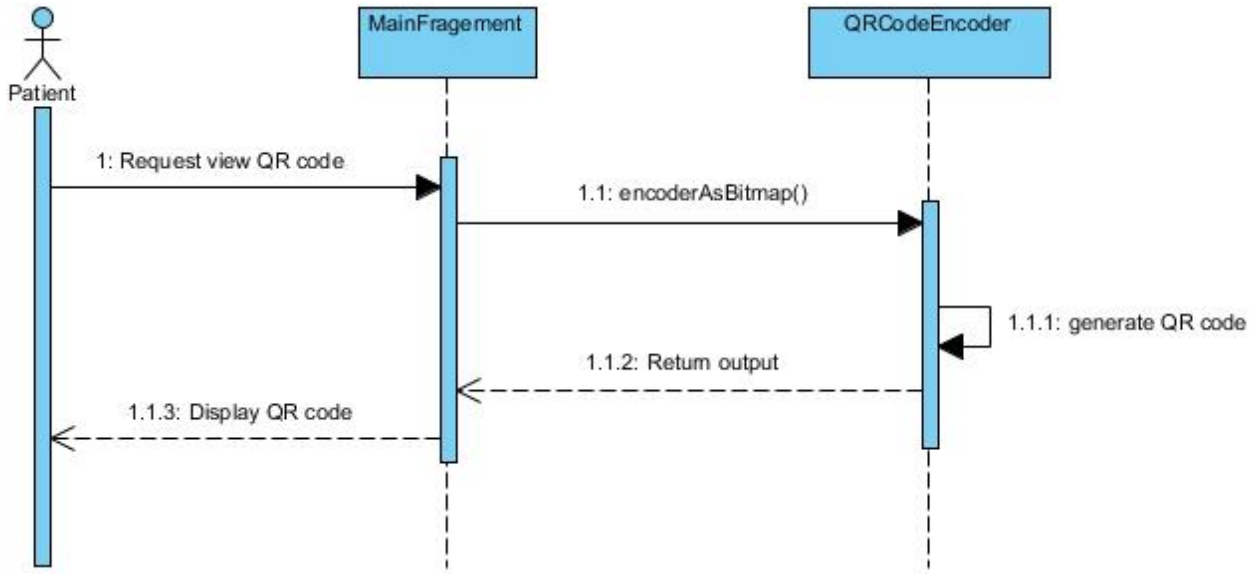


Figure 19: SD-19 Patients can view the QR code on the mobile application

SD-20: Pharmacists can scan the QR code on the mobile application.

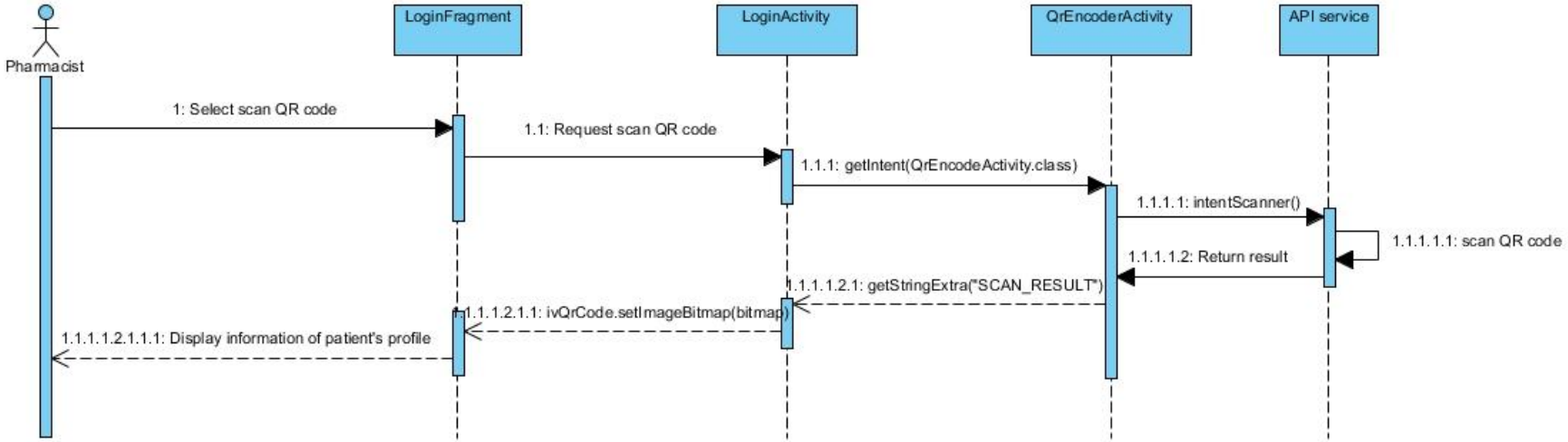


Figure 20: SD-20 Pharmacists can scan the QR code on the mobile application.

SD-21: Pharmacists can add the time of dispensation to the patient's profile on the mobile application.

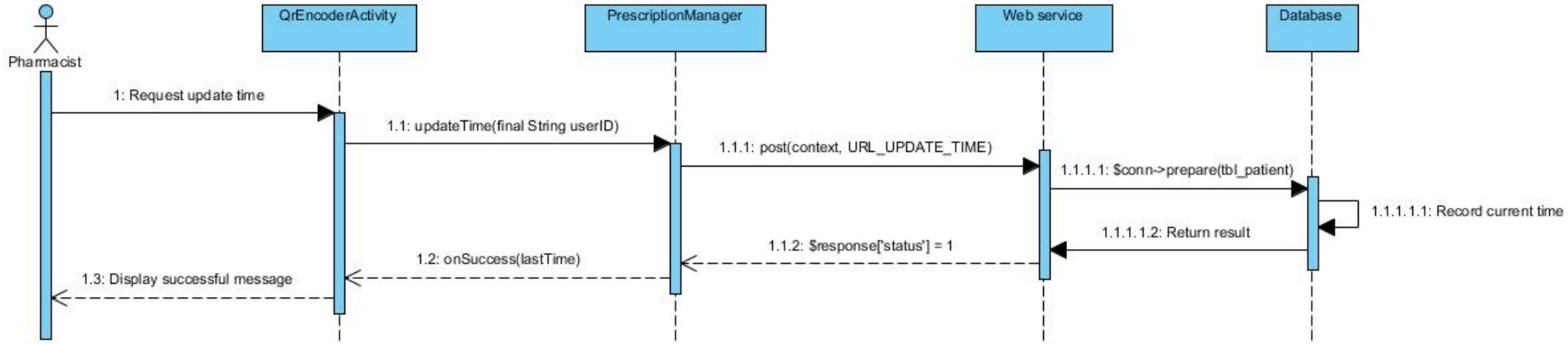


Figure 21: SD-21 Pharmacists can add the time of dispensation to the patient's profile on the mobile application.

SD-22: Doctors can create an allergy report on the web application.

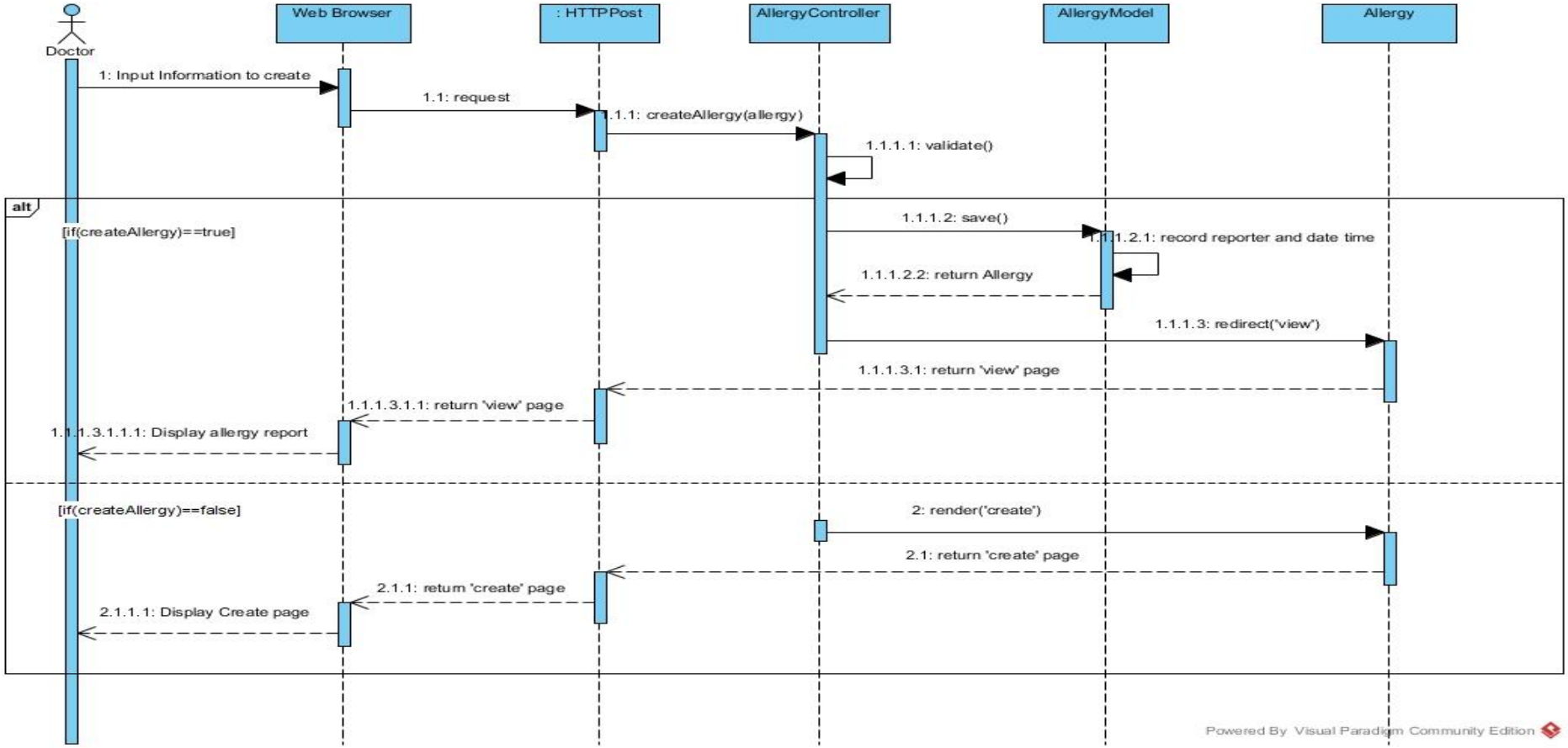


Figure 22: SD-22 Doctors can create an allergy report on the web application.

SD-23: Doctors can update an allergy report on the web application.

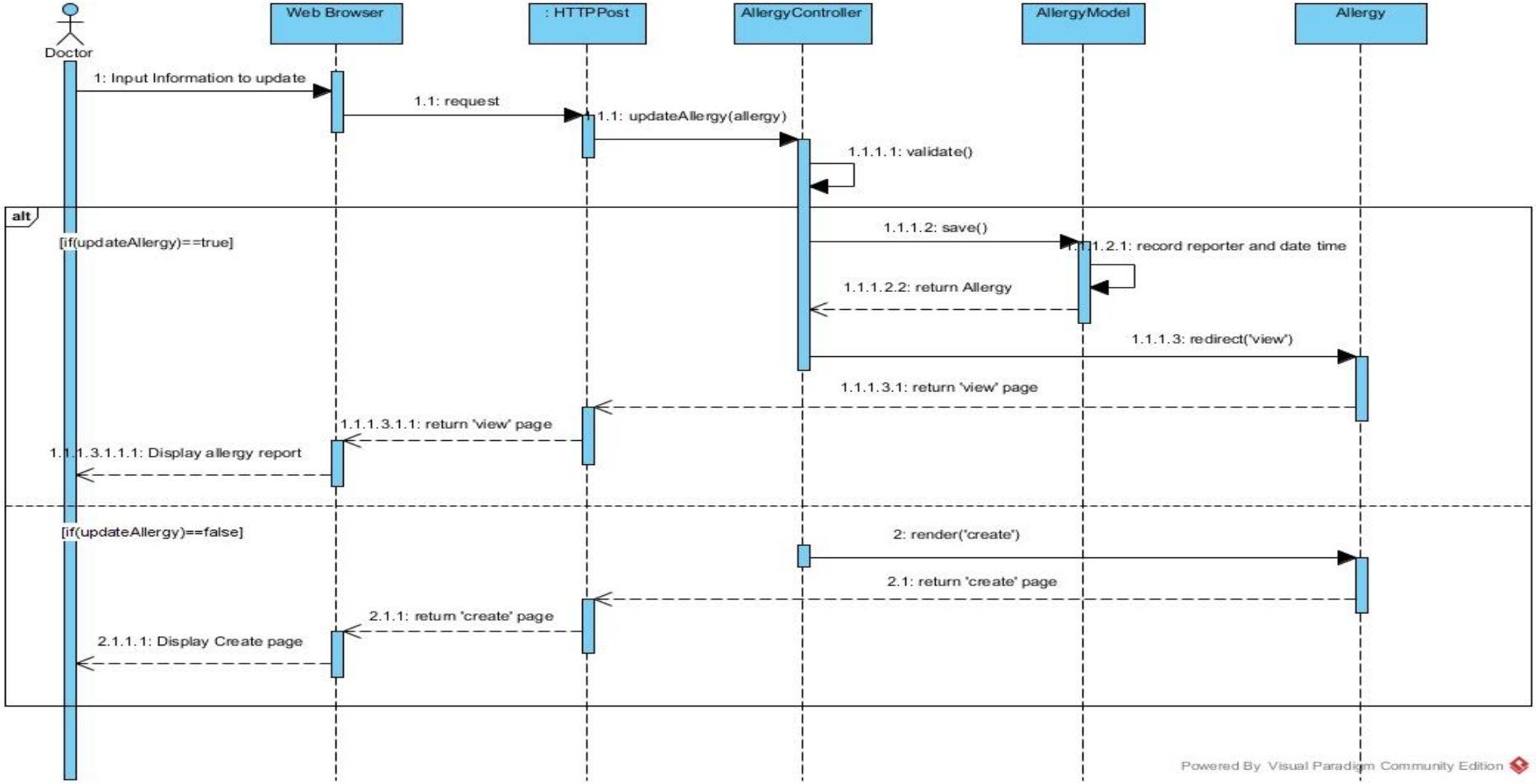


Figure 23: SD-23 Doctors can update an allergy report on the web application.

SD-24: Doctors can delete an allergy report on the web application.

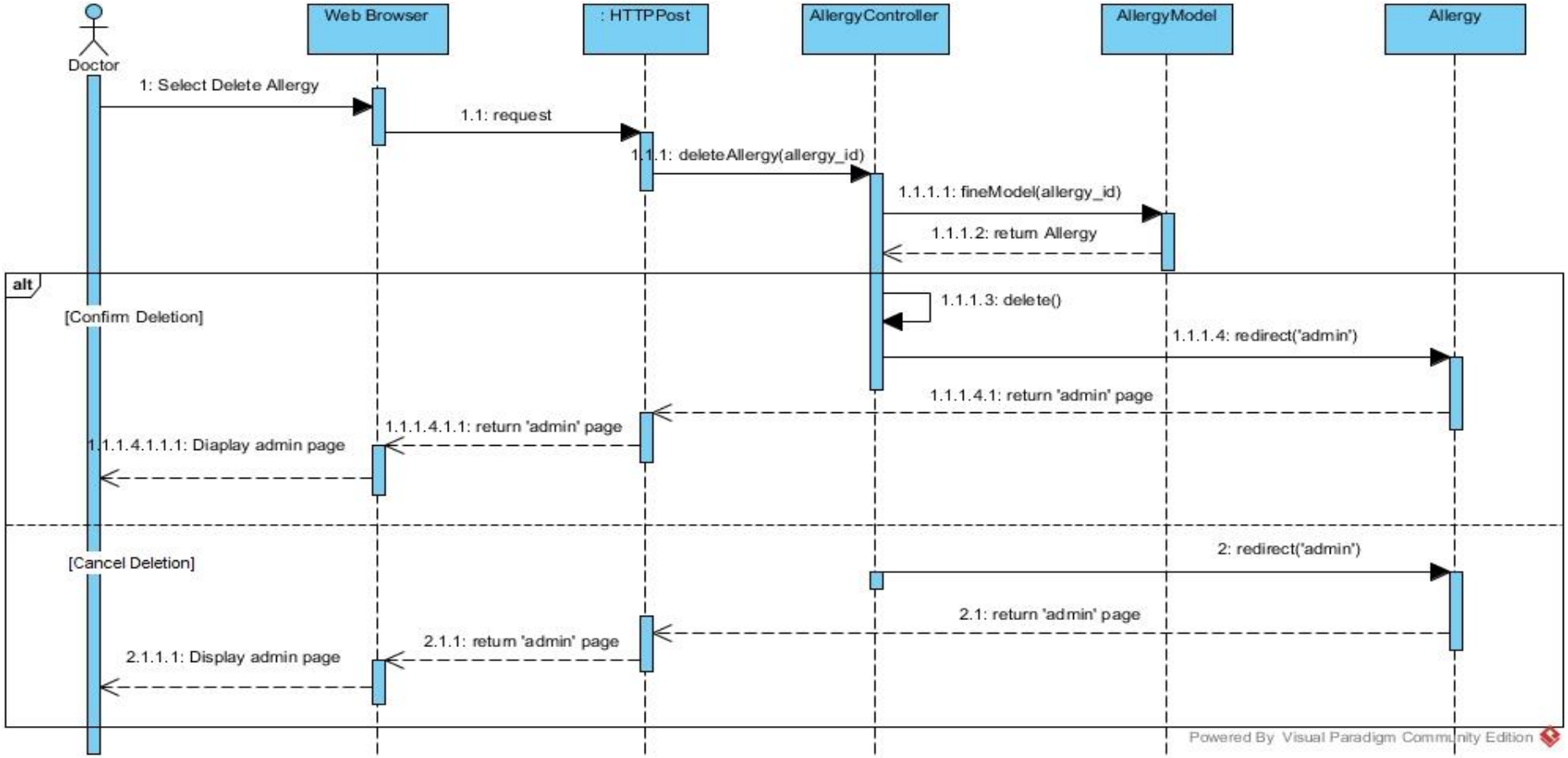


Figure 24: SD-24 Doctors can delete an allergy report on the web application.

SD-25: Doctors can search an allergy report on the web application.

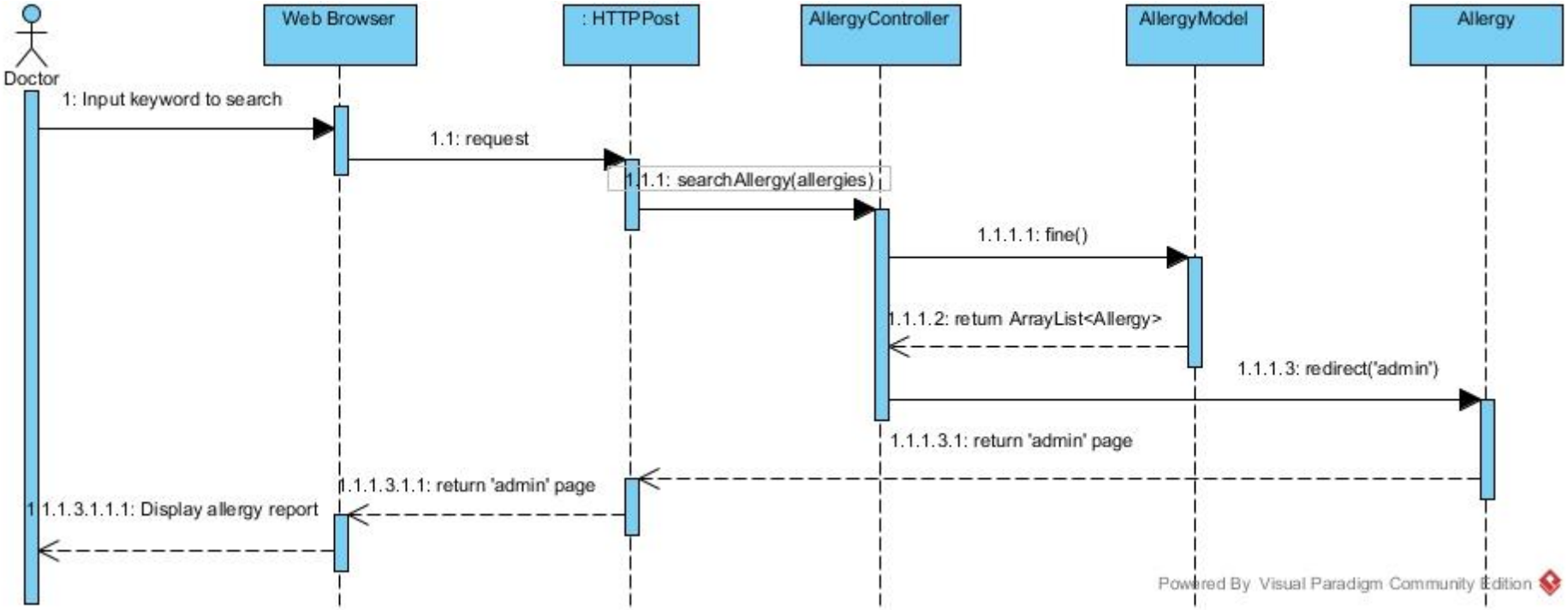


Figure 25: of SD-25 Doctors can search an allergy report on the web application.

SD-26: Doctors can view an allergy report on the web application.

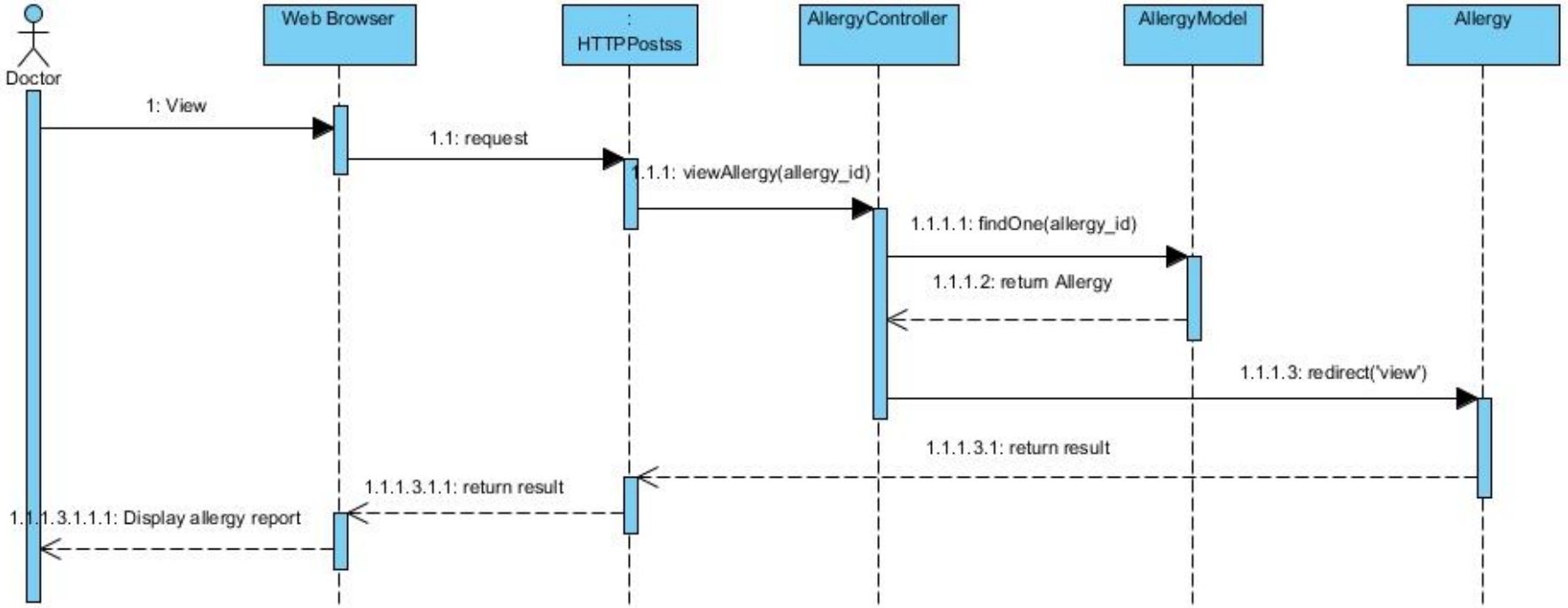


Figure 26: SD-26 Doctors can view an allergy report on the web application.

SD-27: Doctors can view a list of allergy report on the web application.

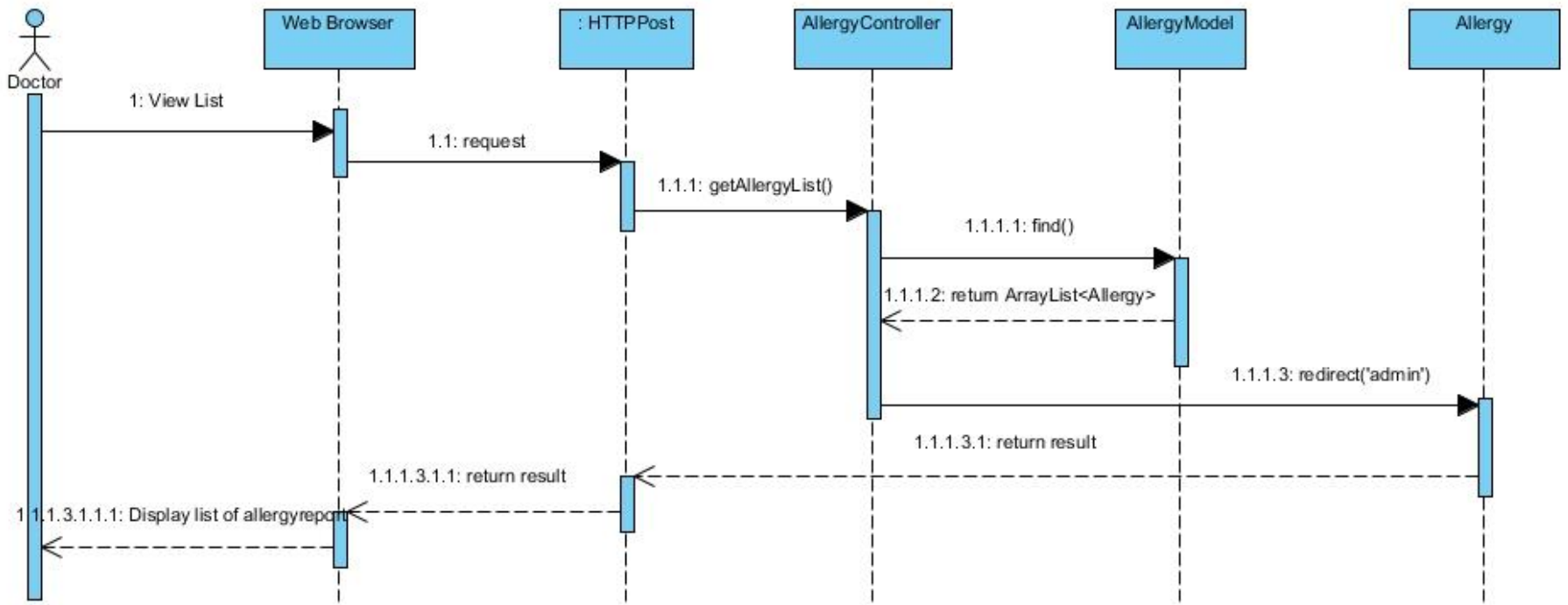


Figure 27: SD-27 Doctors can view a list of allergy report on the web application.

SD-28: FDAs can view a FDA home page on the web application.

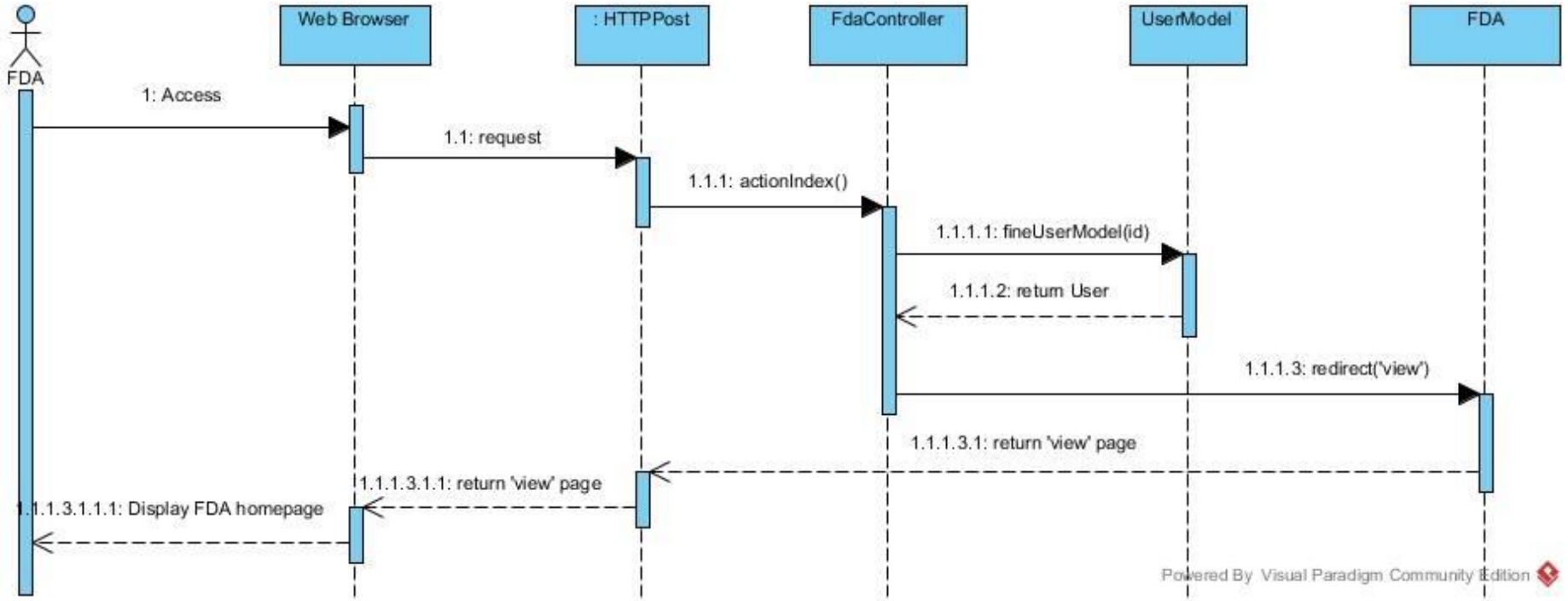


Figure 28: SD-28 FDAs can view a FDA home page on the web application.

SD-29: FDAs can view a list of allergy report on the web application.

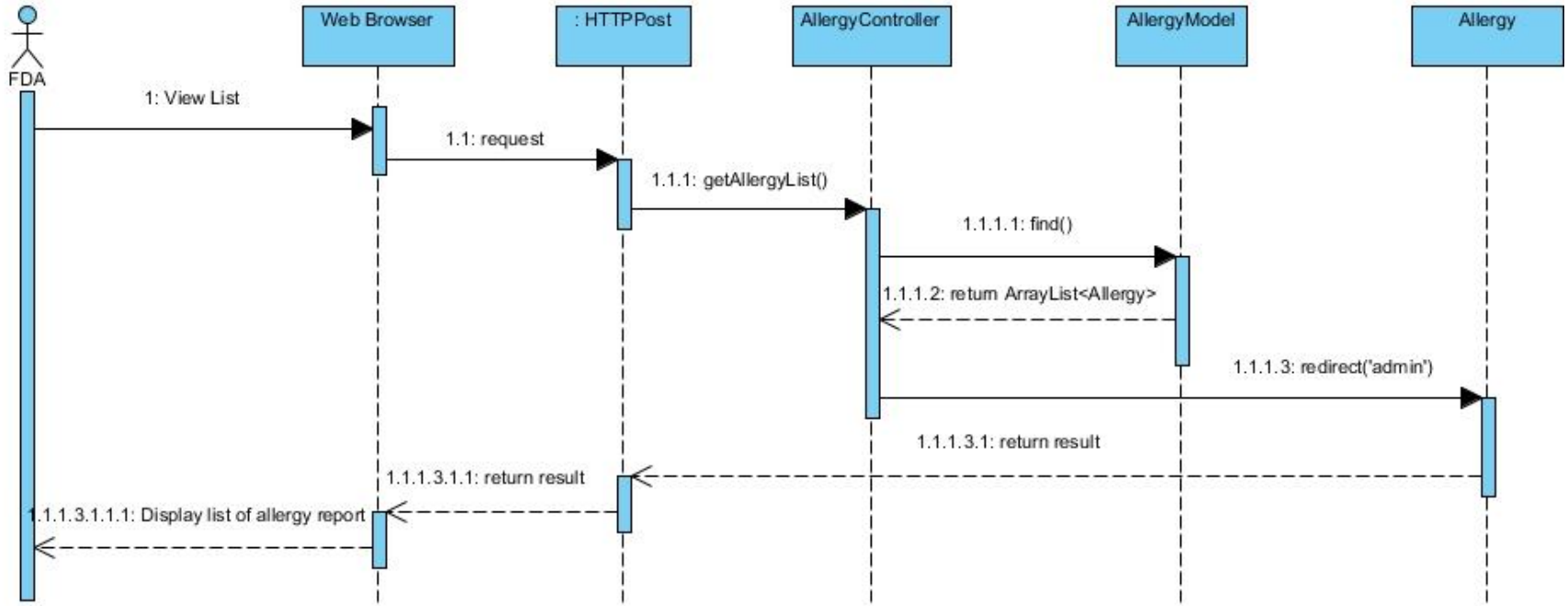


Figure 29: SD-29 FDAs can view a list of allergy report on the web application.

SD-30: FDAs can view a list of allergy report on the web application.

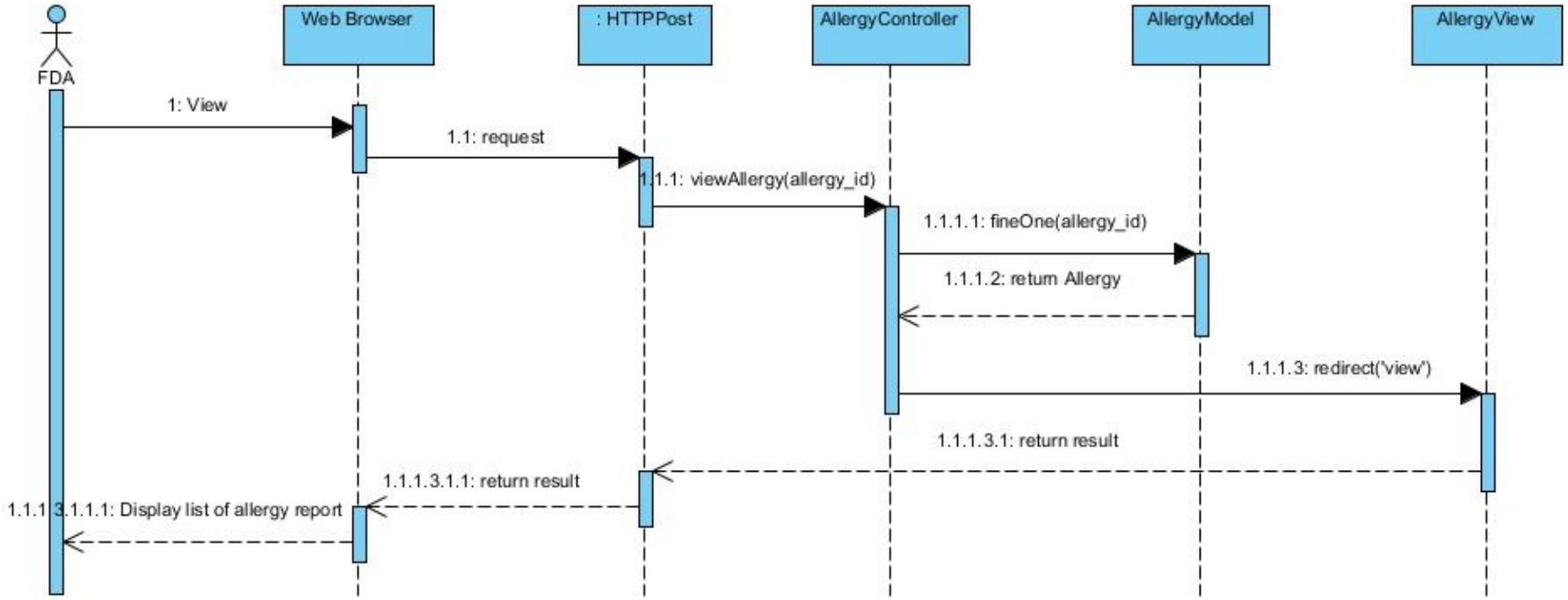
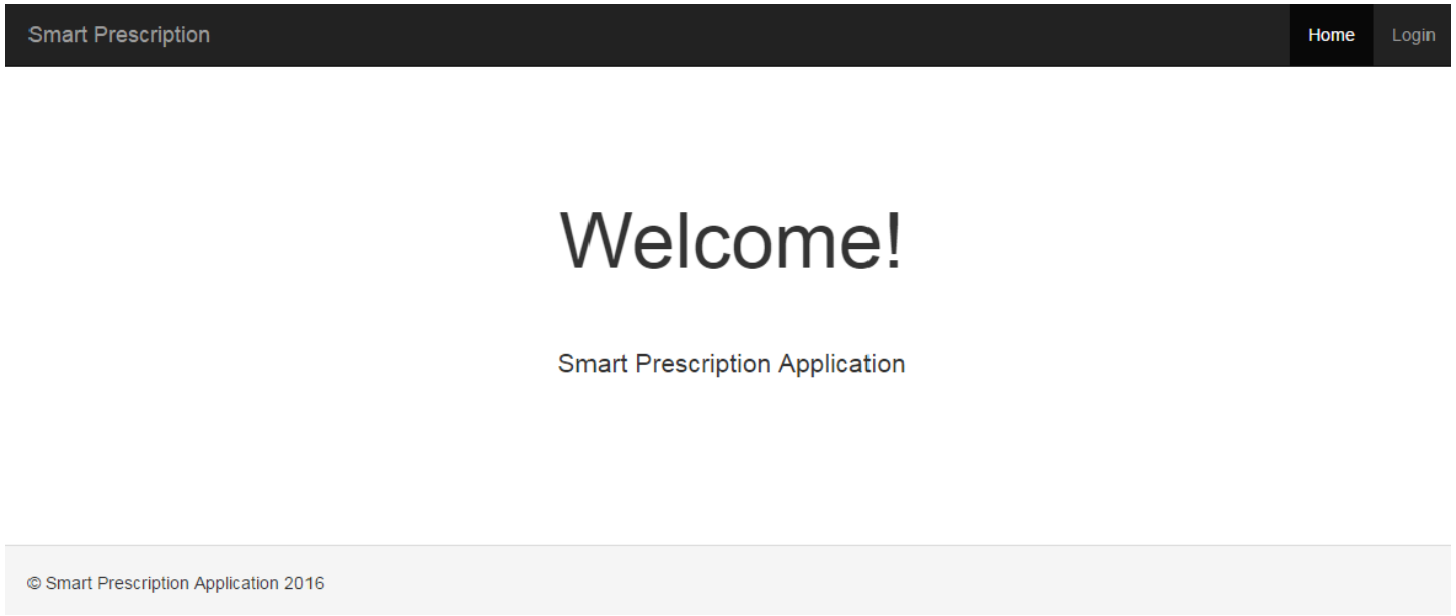


Figure 30: SD-30 FDAs can view a list of allergy report on the web application.

Chapter Six | User Interface Design

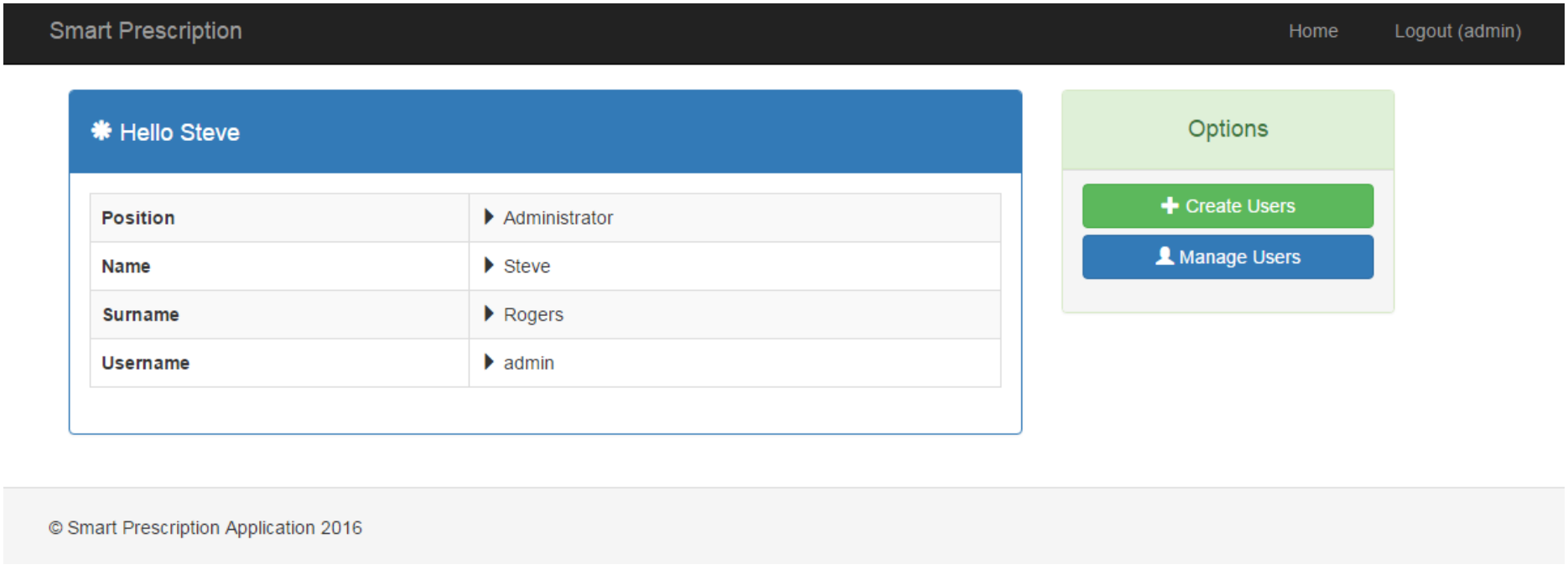
UI-01: Main page



This page is main page use for administration, doctor, and FDA that consist of

(1) "Login" button use for login to the web application.
(2) "Home" button use for moving to the Home page.

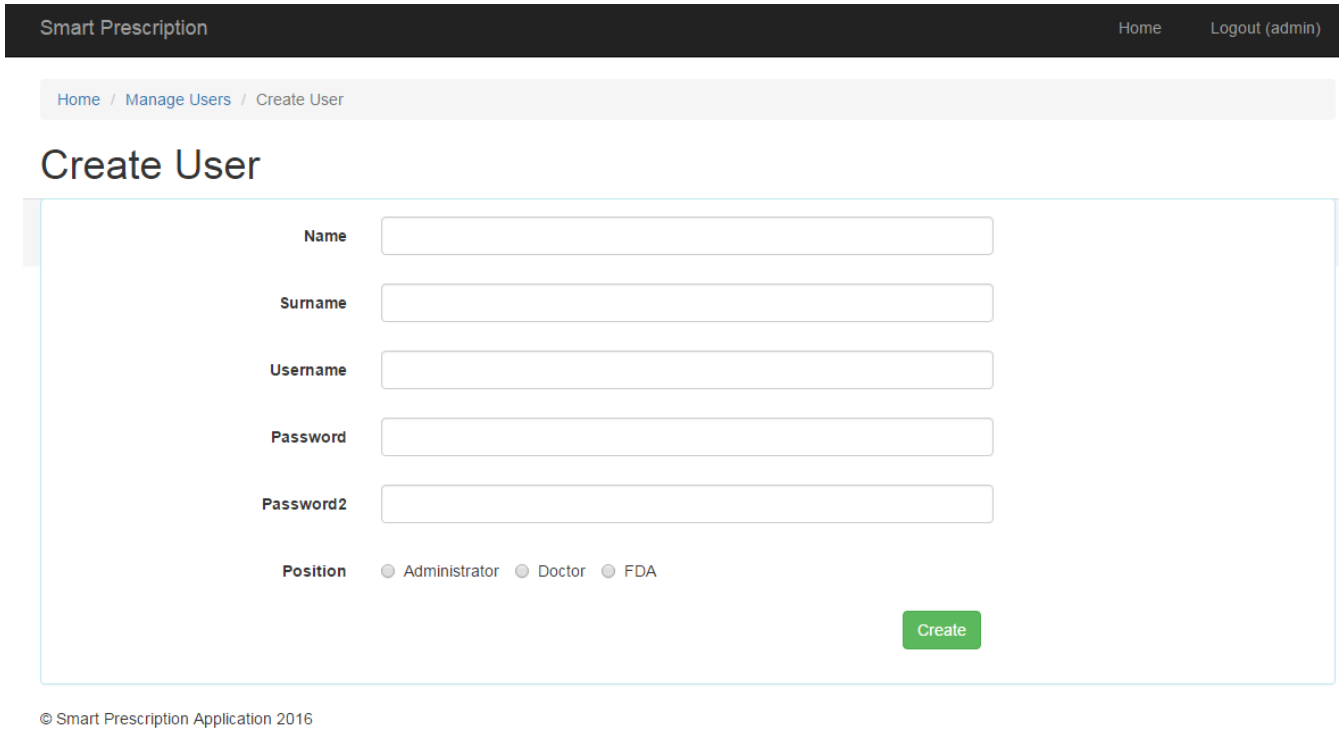
UI-02: Administration home page



This page is administration home page for administration that consist of

(1) "Logout" button use for logout from the web application.
(2) "Home" button use for moving to the Home page.
(3) Information of user that working on the web application.
(4) Option menu for create users and manage users.

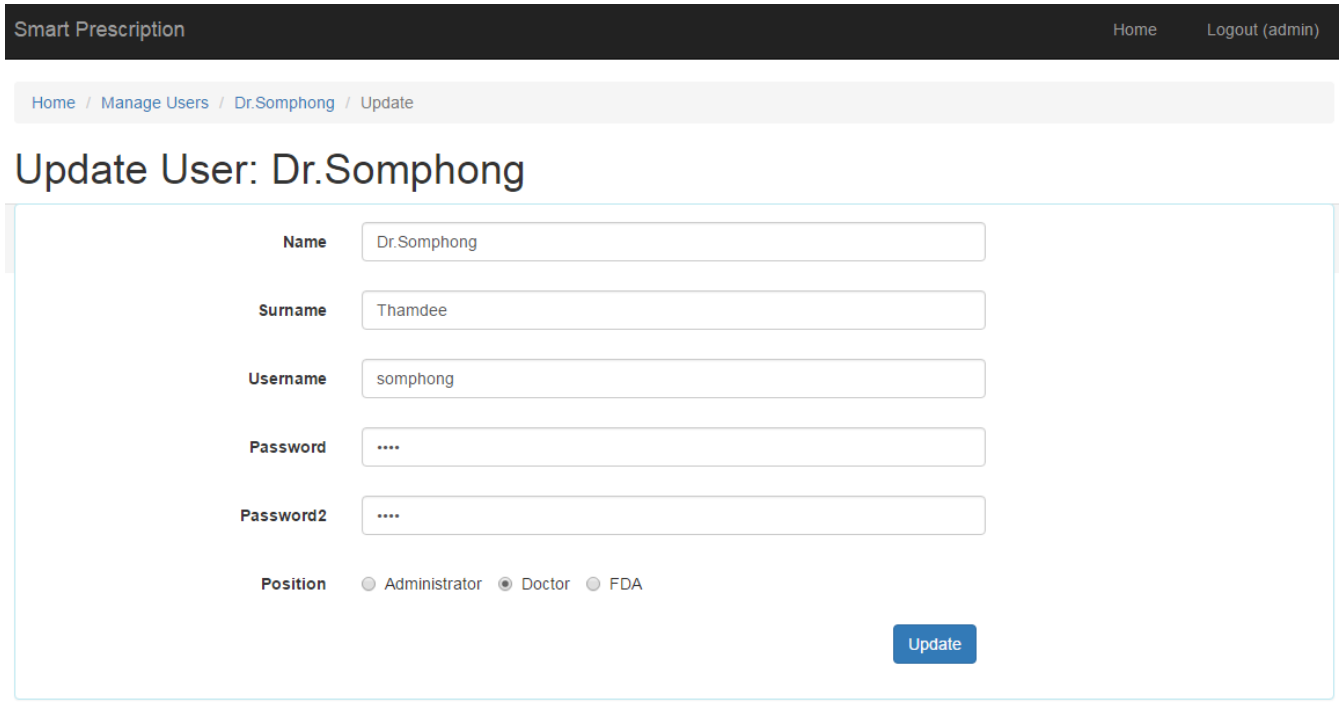
UI-03: Create user page



This page is create user page for administration that consist of

(1) "Logout" button use for logout from the web application.
(2) "Home" button use for moving to the Home page.
(3) Breadcrumb link use for select to the previous page.
(4) Users create form for input the information of user's profile.
(5) "Create" button for select to create user's profile.

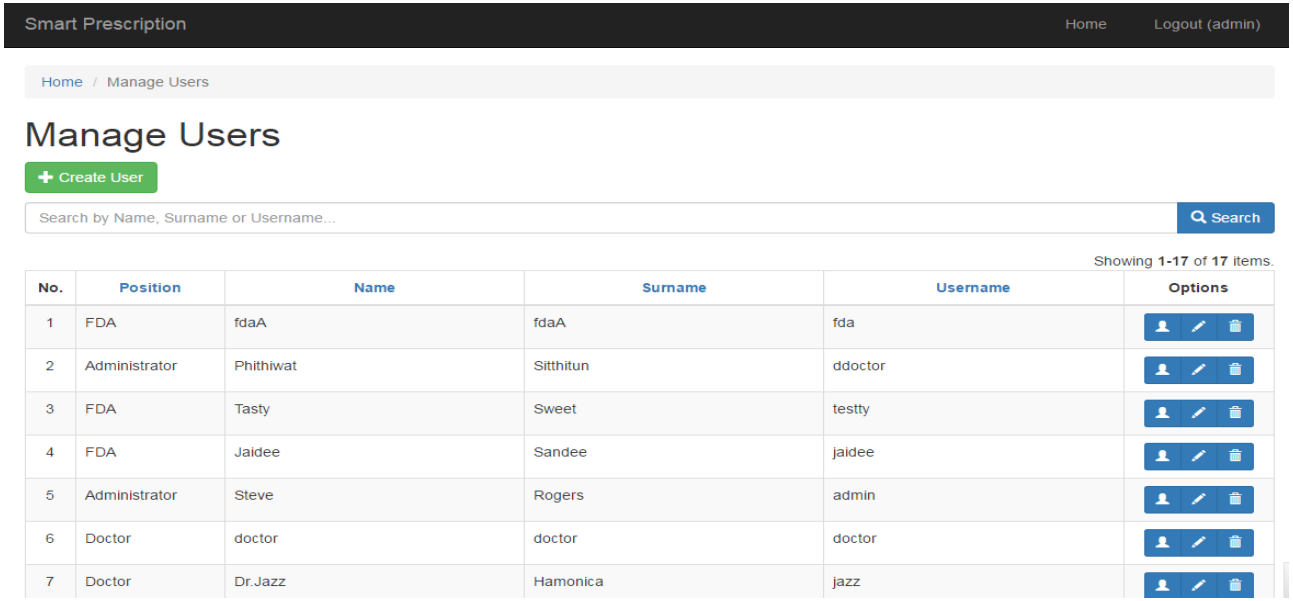
UI-04: Update user page



This page is update user page for administration that consist of

(1) "Logout" button use for logout from the web application.
(2) "Home" button use for moving to the Home page.
(3) Breadcrumb link use for select to the previous page.
(4) Users update form for input the information of user's profile.
(5) "Update" button for select to update user's profile.

UI-05: Manage users page

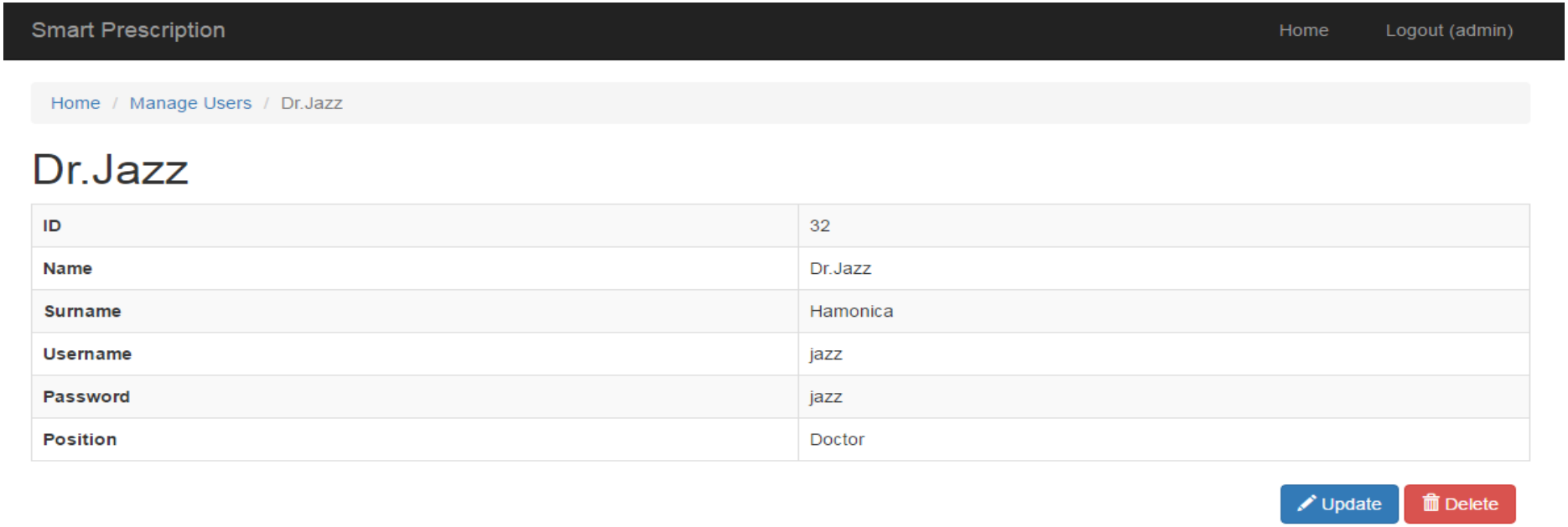


This page is manage user page for administration that consist of

(1) "Logout" button use for logout from the web application.
(2) "Home" button use for moving to the Home page.
(3) Breadcrumb link use for select to the previous page.
(4) The number and total amount of user's profiles.

(5) List of user's profile include six columns (no., position, name, surname, username, and options.).
(6) "Create User" button for select to create user's profile.
(7) "Options column of table" uses for view, edit, and delete the user's profile.
(8) Search text field use for input a name, surname or username of user's profile.
(9) "Search" button for select to search user's profiles.

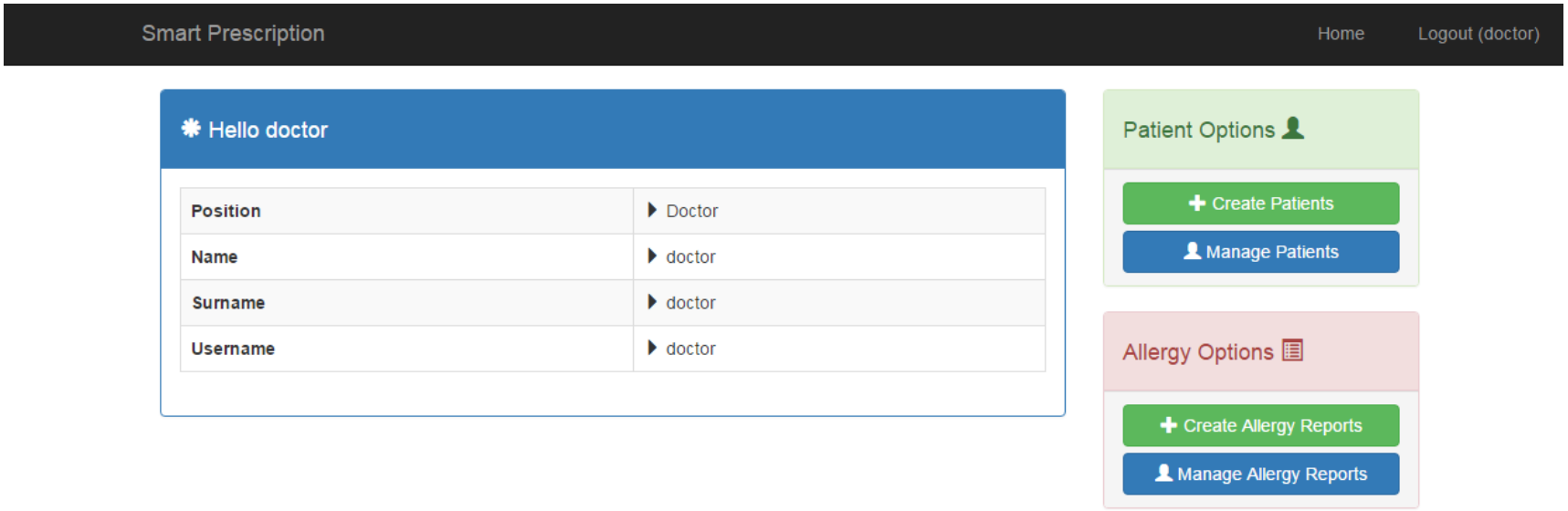
UI-06: User's profile page



This page is user's profile page for administration that consist of

(1) "Logout" button use for logout from the web application.
(2) "Home" button use for moving to the Home page.
(3) Breadcrumb link use for select to the previous page.
(4) Information of user's profile.
(5) "Update" button use for select to update the user's profile.
(6) "Delete" button use for select to delete the user's profile.

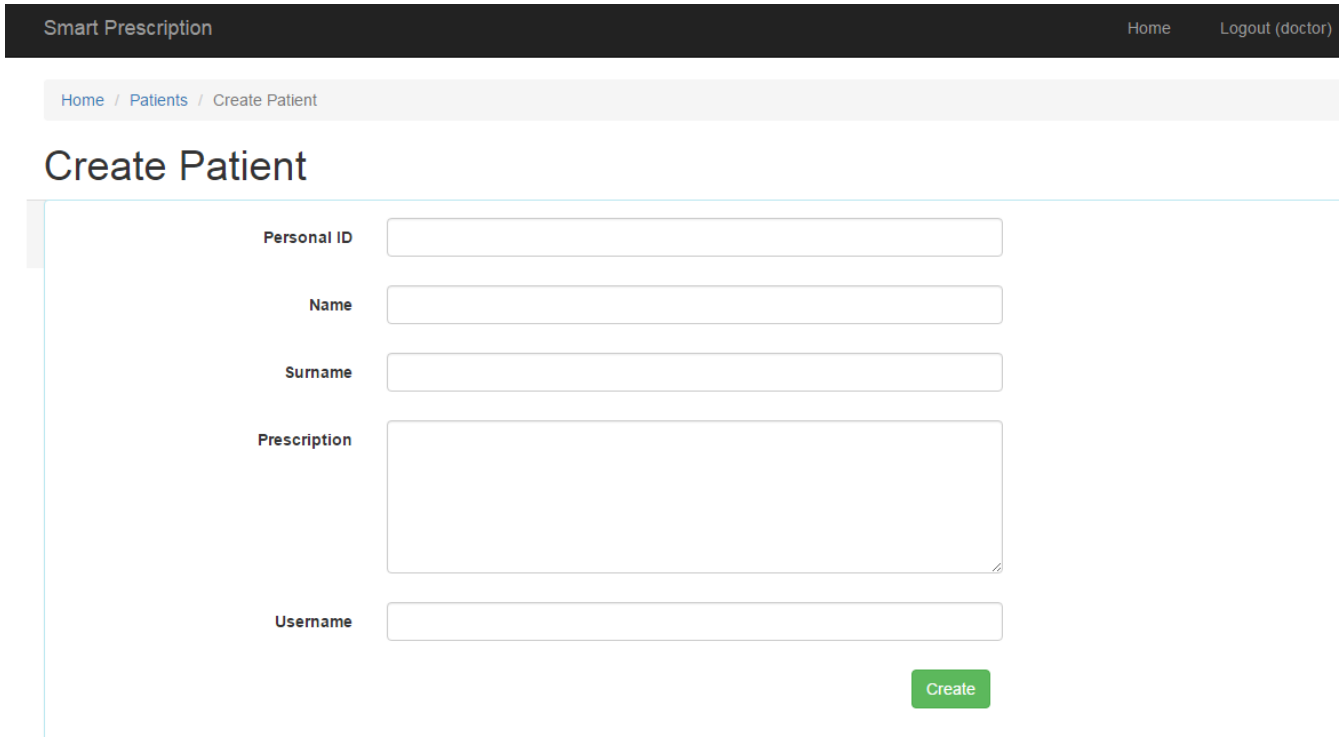
UI-07: Doctor home page



This page is doctor home page for doctor that consist of

(1) "Logout" button use for logout from the web application.
(2) "Home" button use for moving to the Home page.
(3) Information of doctor that working on the web application.
(4) Option menu for create patient and manage patient.

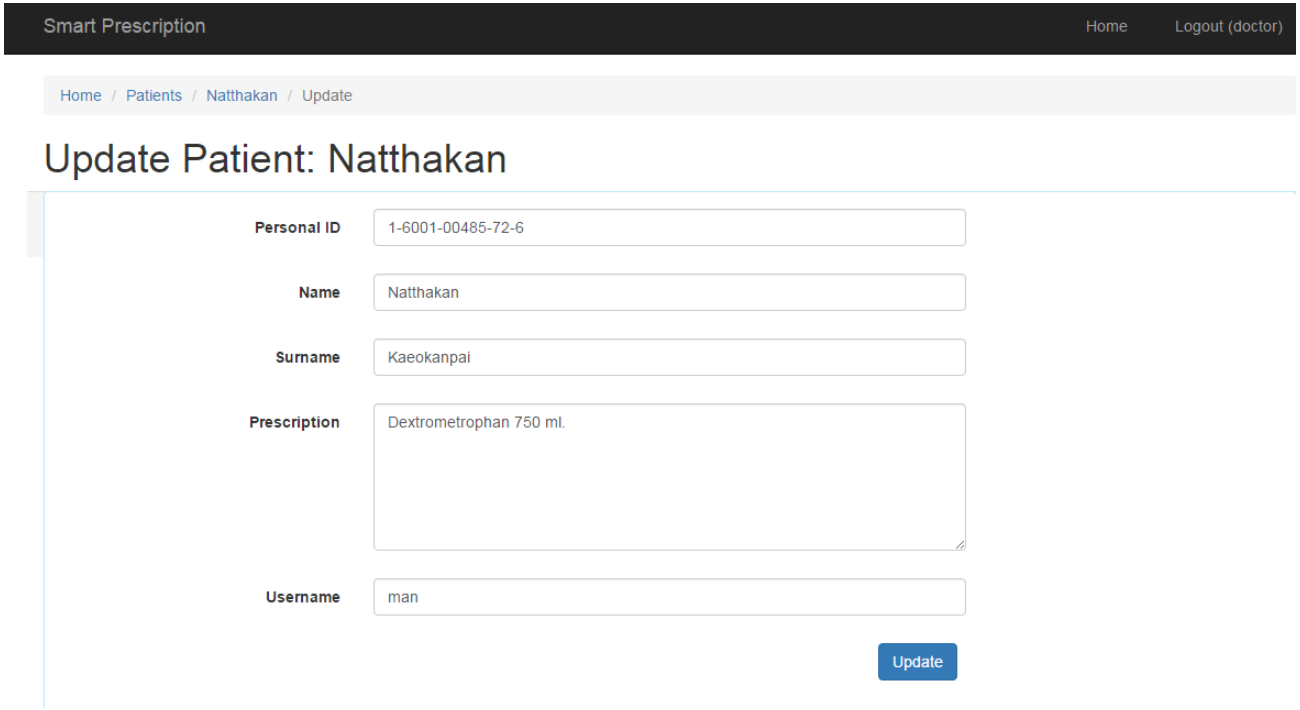
UI-08: Create patient page



This page is create patient page for doctor that consist of

(1) "Logout" button use for logout from the web application.
(2) "Home" button use for moving to the Home page.
(3) Breadcrumb link use for select to the previous page.
(4) Patients create form for input the information of patient's profile.
(5) "Create" button for select to create a patient's profile.

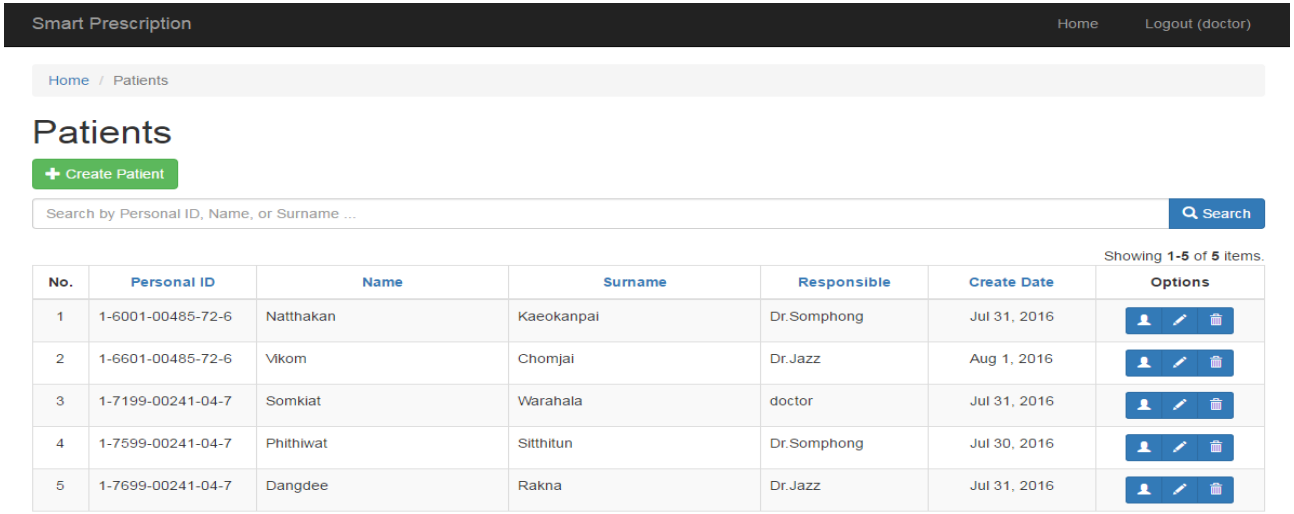
UI-09: Update patient page



This page is update patient page for doctor that consist of

(1) "Logout" button use for logout from the web application.
(2) "Home" button use for moving to the Home page.
(3) Breadcrumb link use for select to the previous page.
(4) Patient update form for input the information of patient's profile.
(5) "Update" button for select to update patient's profile.

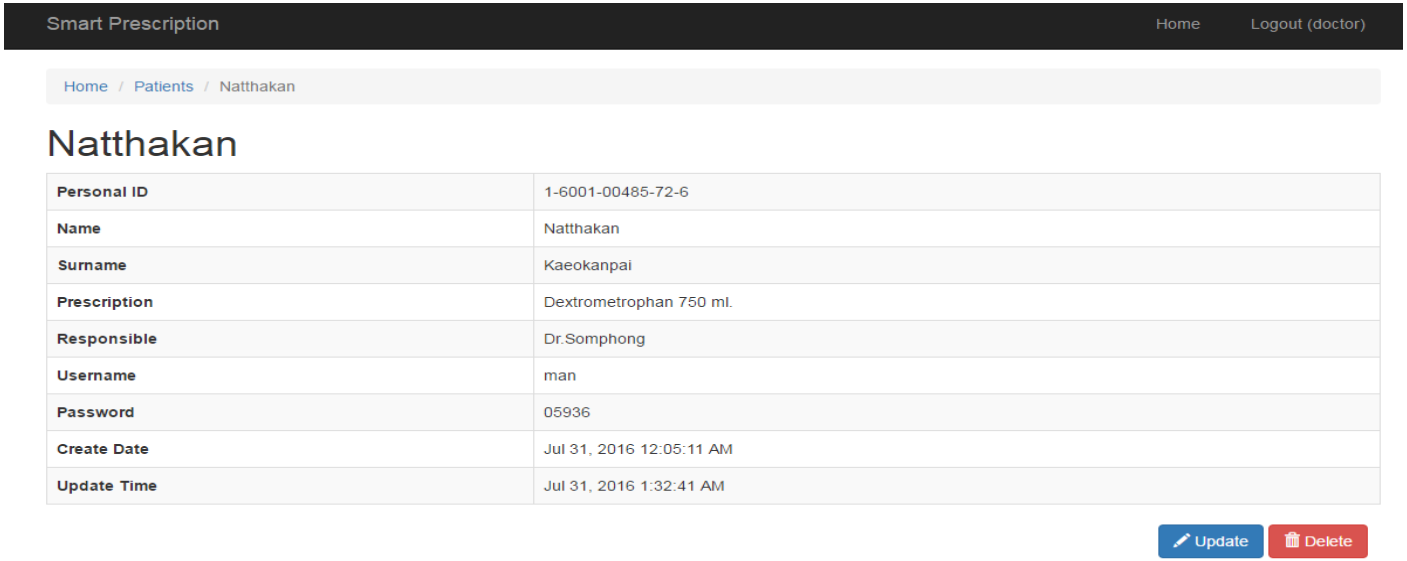
UI-10: Manage patient page



This page is manage patient page for doctor that consist of

(1) “Logout” button use for logout from the web application.
(2) “Home” button use for moving to the Home page.
(3) Breadcrumb link use for select to the previous page.
(4) The number and total amount of patient’s profiles.
(5) List of patient’s profile include seven columns (no., personal ID, name, surname, responsible, create date, and options.).
(6) “Create patient” button for select to create patient’s profile.
(7) “Options column of table” uses for view, edit, and delete the patient’s profile.
(8) Search text field use for input a name, surname or personal ID of patient’s profile.
(9) “Search” button for select to search patient’s profiles.

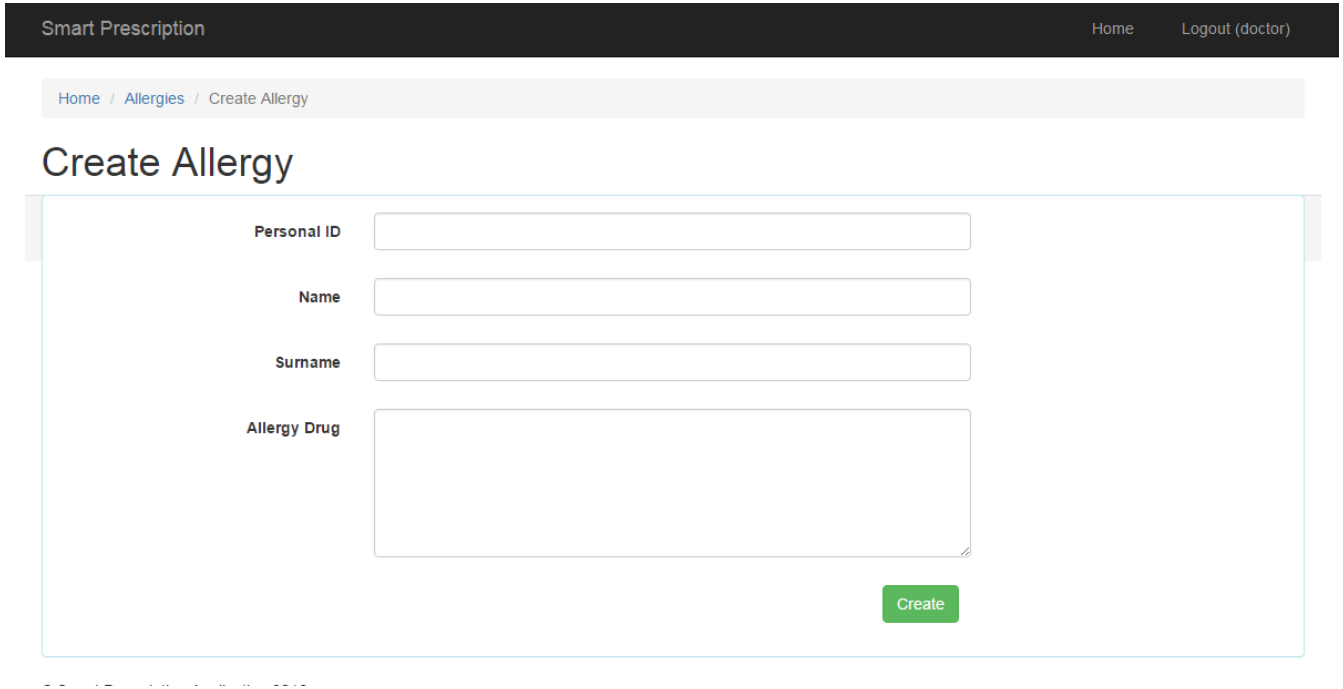
UI-11: Patient's profile page



This page is patient's profile page for doctor that consist of

(1) "Logout" button use for logout from the web application.
(2) "Home" button use for moving to the Home page.
(3) Breadcrumb link use for select to the previous page.
(4) Information of patient's profile.
(5) "Update" button use for select to update the patient's profile.
(6) "Delete" button use for select to delete the patient's profile.

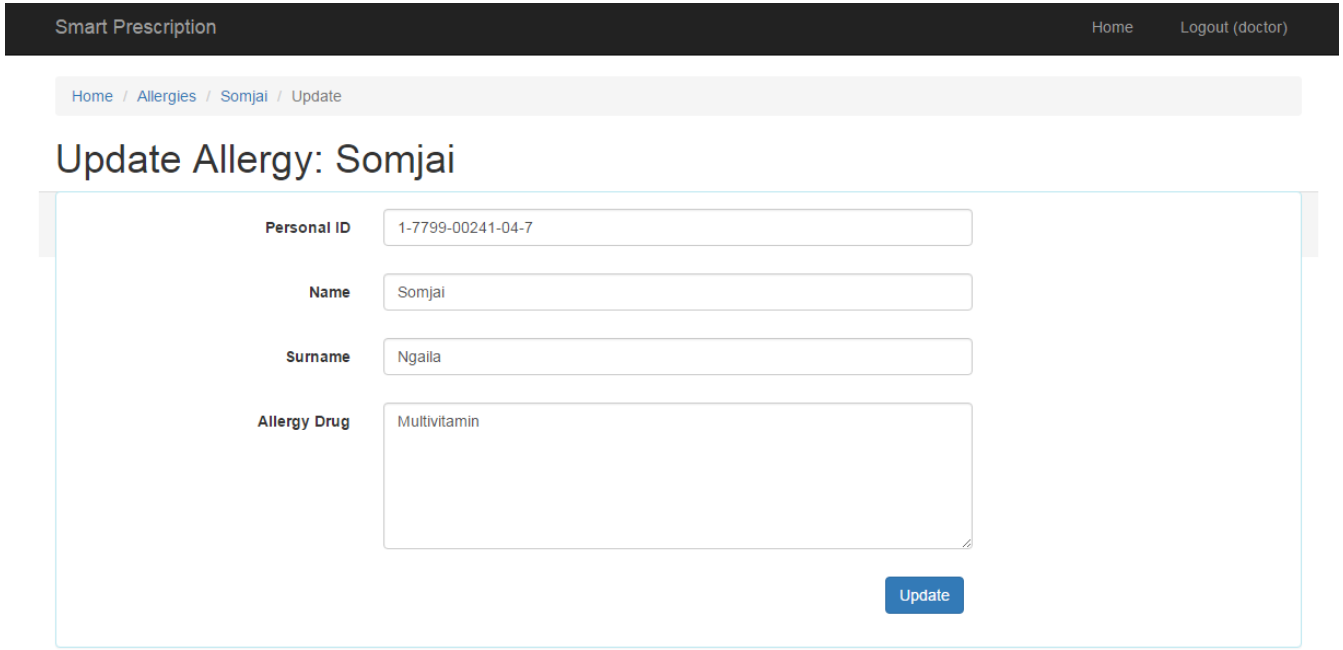
UI-12: Create allergy report page



This page is create allergy report page for doctor that consist of

(1) “Logout” button use for logout from the web application.
(2) “Home” button use for moving to the home page.
(3) Breadcrumb link use for select to the previous page.
(4) Allergy reports create form for input the information of allergy report.
(5) “Create” button for select to create allergy report.

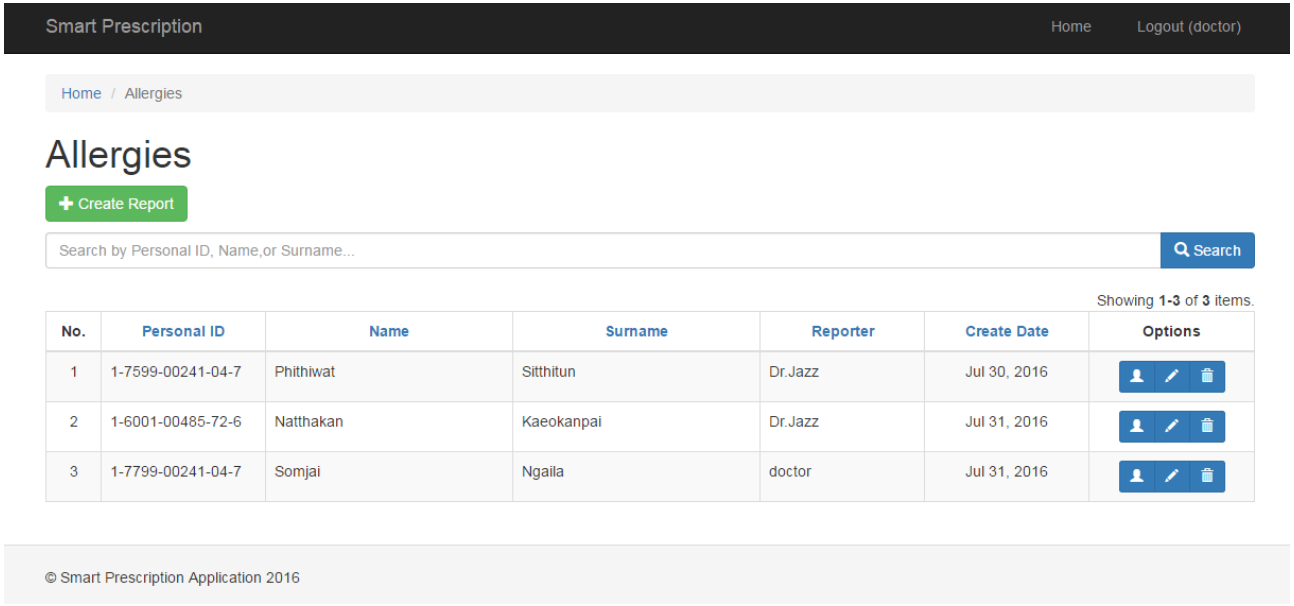
UI-13: Update allergy report page



This page is update allergy report page for doctor that consist of

(1) "Logout" button use for logout from the web application.
(2) "Home" button use for moving to the Home page.
(3) Breadcrumb link use for select to the previous page.
(4) Allergy report updates form for input the information of allergy report.
(5) "Update" button for select to update allergy report.

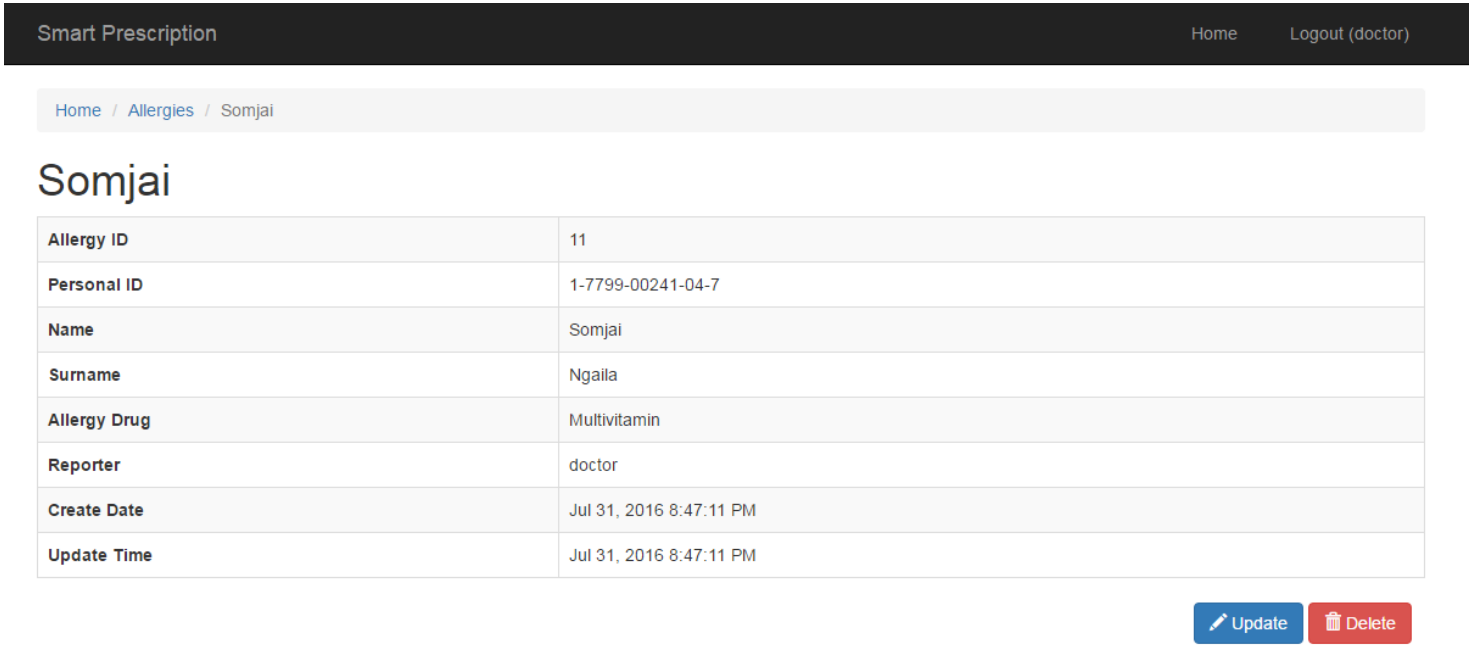
UI-14: Manage allergy reports page



This page is manage allergy report page for doctor that consist of

(1) "Logout" button use for logout from the web application.
(2) "Home" button use for moving to the home page.
(3) Breadcrumb link use for select to the previous page.
(4) The number and total amount of allergy report.
(5) List of allergy report include six columns (no., personal ID, name, surname, reporter, create date, and options.).
(6) "Create report" button for select to create allergy report.
(7) "Options column of table" uses for view, edit, and delete the allergy report.
(8) Search text field use for input a name, surname or personal ID of allergy report.
(9) "Search" button user for select to search allergy report.

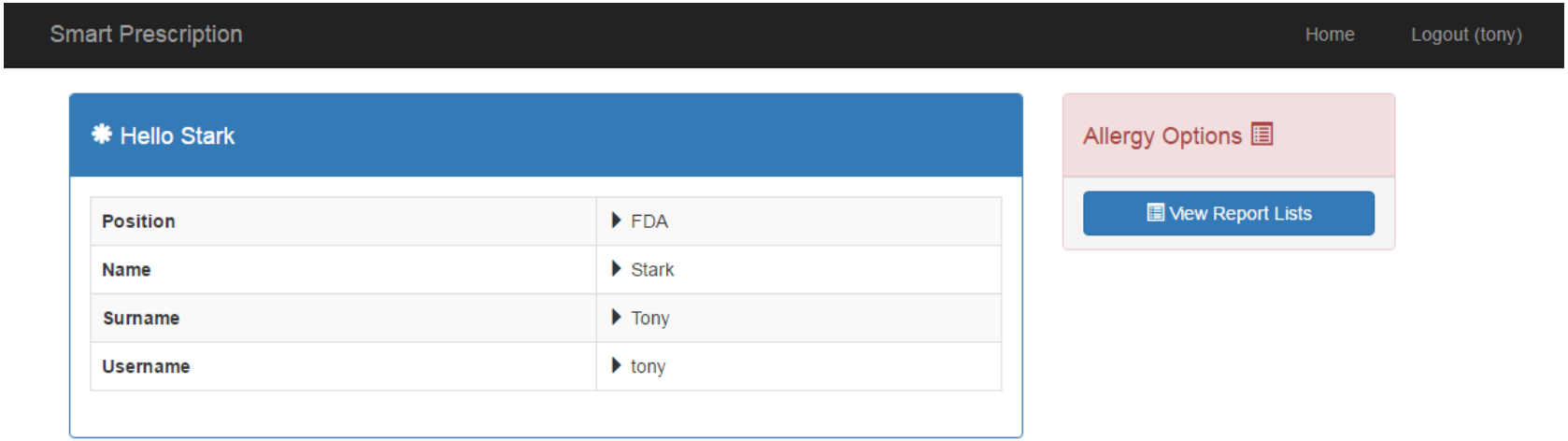
UI-15: Allergy report page



This page is allergy report page for doctor that consist of

(1) "Logout" button use for logout from the web application.
(2) "Home" button use for moving to the home page.
(3) Breadcrumb link use for select to the previous page.
(4) Information of allergy report.
(5) "Update" button use for select to update the allergy report.
(6) "Delete" button use for select to delete the allergy report.

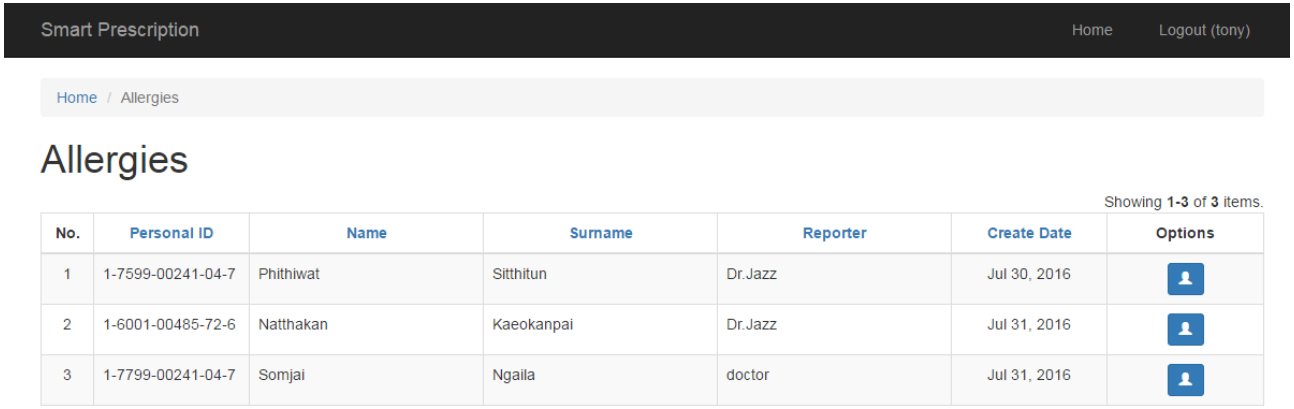
UI-16: FDA home page



This page is FDA home page for FDA that consist of

(1) "Logout" button use for logout from the web application.
(2) "Home" button use for moving to the Home page.
(3) Information of FDA that working on the web application.
(4) Option menu for view allergy report.

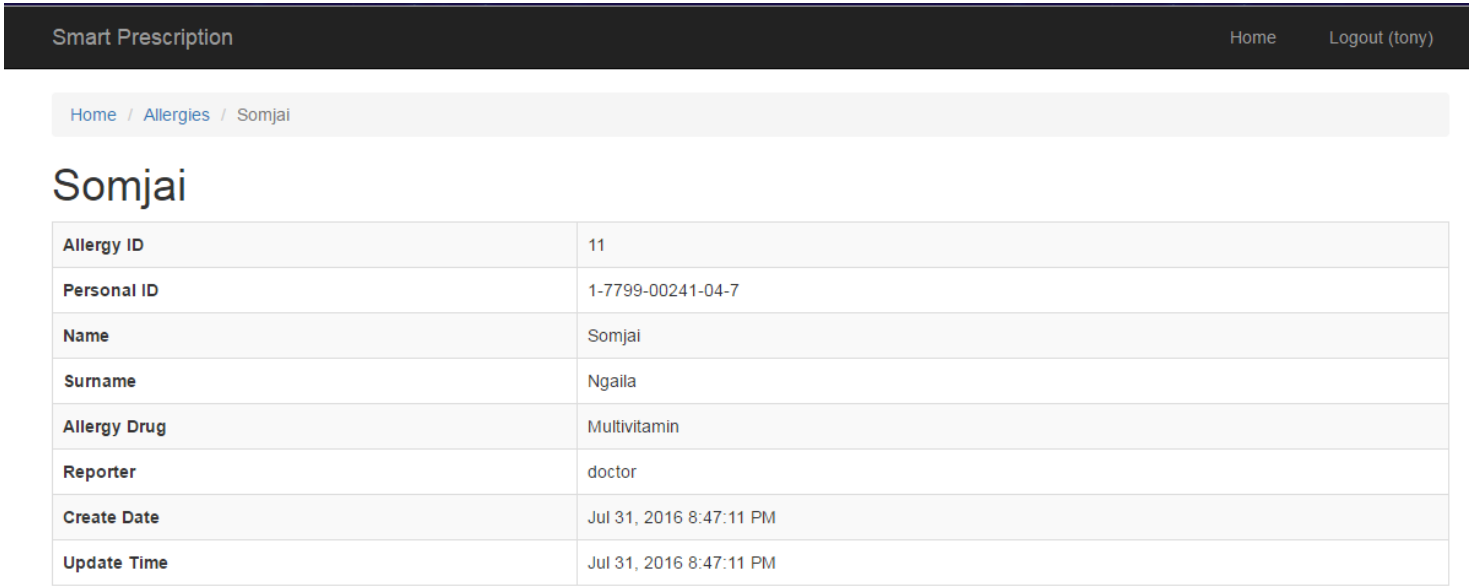
UI-17: List allergy report page



This page is list allergy report page for FDA that consist of

(1) “Logout” button use for logout from the web application.
(2) “Home” button use for moving to the home page.
(3) Breadcrumb link use for select to the previous page.
(4) The number and total amount of allergy report.
(5) List of allergy report include six columns (no., personal ID, name, surname, reporter, create date, and options.).
(6) “Options column of table” uses for view the allergy report.

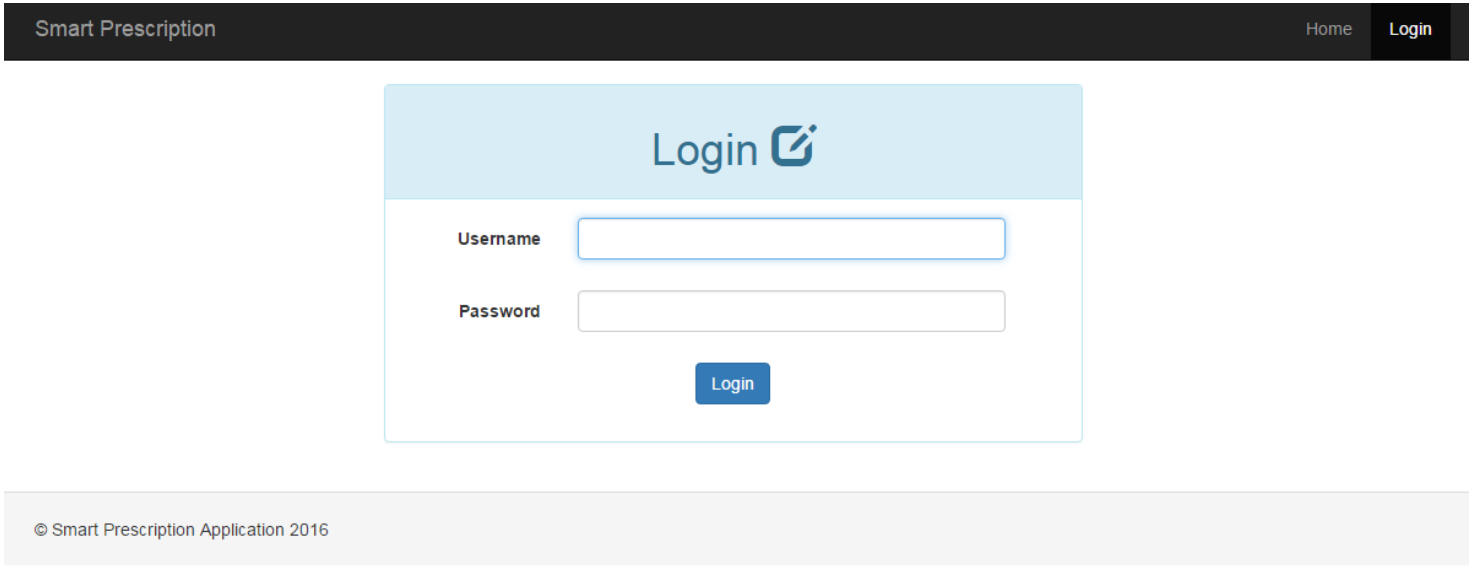
UI-18: FDA Allergy report page



This page is FDA allergy report page for FDA that consist of

(1) "Logout" button use for logout from the web application.
(2) "Home" button use for moving to the home page.
(3) Breadcrumb link use for select to the previous page.
(4) Information of allergy report.

UI-19: Login page

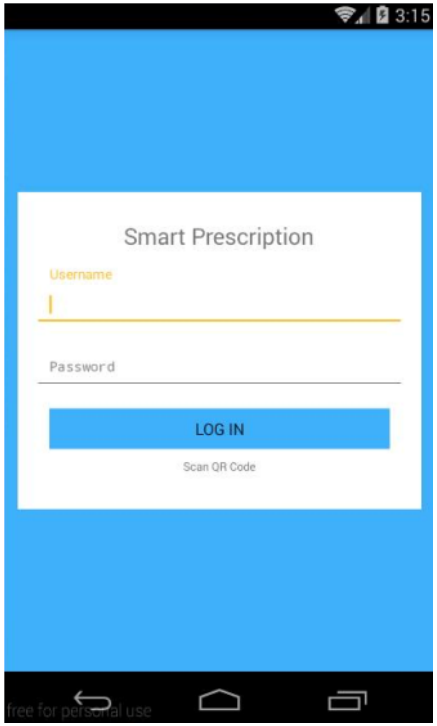


This page is Login page for doctor, administration, and FDA that consist of

(1) "Login" button use for moving to the login page.
(2) "Home" button use for moving to the home page.
(3) Login form for input a username and password.
(4) "Login" button for click to login.

(Mobile Application)

UI-20: Login page (mobile)



This page is Login page (mobile) for patient that consist of

(1) Login form for input a username and password.
(2) “Login” button for click to login.
(3) “Scan QR code” button for click to scan QR code.

UI-21: Main page (mobile)

This page is main page (mobile) for patient that consist of

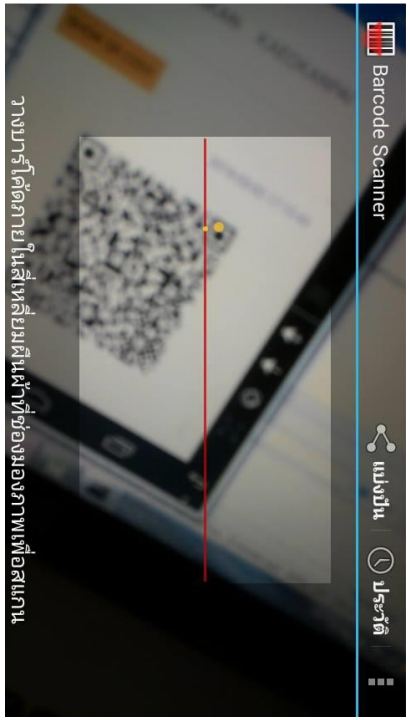
(1) "Logout" button for logout from the application.
(2) Name of patient's profile.
(3) "SHOW QR CODE" button for click to display QR code.

UI-22: QR code page (mobile)

This page is QR code page for patient that consist of

(1) "Logout" button for logout from the application.
(2) Name of patient's profile.
(3) "SHOW QR CODE" button for click to display QR code.
(4) QR code of the patient's profile.

UI-23: Scan QR code page (mobile)



This page is scan QR code page for pharmacist that consist of

(1) Scan QR code function.

UI-24: Pharmacist page (mobile)

This page is pharmacist page for pharmacist that consist of

(1) Information of patient's profile.
(2) "CONFIRM TO USE" button for click to record the current time.