

# RUP Design Model Report

## for PRMS

Issue 2.1



Date:

13/7/2016

Filing ref:

PRC/PRMS/TS.6

CREATED WITH  **ENTERPRISE  
ARCHITECT**

# Revision History

<b>Date</b>	<b>Issue</b>	<b>Description</b>	<b>Author</b>
10 August 2012	1.0	Initial version	Heng Boon Kui, Swarna Ashok
6 August 2013	1.1	Revised for SE-OOAD 2013 run. Refined the interaction diagram.	Heng Boon Kui, Saisudha Rajagopal
19 August 2013	1.2	Revised for SE-OOAD 2013 run. Refined the interaction diagram.	Heng Boon Kui
6 August 2015	2.0	Ported to Enterprise Architect. Refined the class and interaction diagrams.	Heng Boon Kui
13 July 2016	2.1	Included the specification of interfaces missing from the previous report.	Heng Boon Kui

# Table of Contents

<b>1 Design Objects</b>	<b>7</b>
1.1 Class Diagram for Entity Classes (Users)	7
1.2 Class Diagram of Entity Classes (Radio Programs)	8
1.3 phoenix	8
1.3.1 core	9
1.3.1.1 controller	9
1.3.1.1.1 Class: Public PhoenixFrontController	9
1.3.1.1.1.1 Operation: Public void doGet()	9
1.3.1.1.1.2 Operation: Public void doPost()	9
1.3.1.1.1.3 Operation: Public void processRequest()	9
1.3.1.2 dao	10
1.3.1.2.1 Class: Public DAOFactory	10
1.3.1.2.1.1 Operation: Public DAOFactory()	10
1.3.1.2.1.2 Operation: Public ProgramDAO getProgramDAO()	10
1.3.1.2.1.3 Operation: Public RoleDAO getRoleDAO()	10
1.3.1.2.1.4 Operation: Public ScheduleDAO getScheduleDAO()	10
1.3.1.2.1.5 Operation: Public UserDAO getUserDAO()	10
1.3.2 authenticate	11
1.3.2.1 controller	11
1.3.2.1.1 Class: Public LoginCmd	11
1.3.2.1.1.1 Operation: Public String perform()	11
1.3.2.1.2 Class: Public LogoutCmd	11
1.3.2.1.2.1 Operation: Public String perform()	11
1.3.2.3 dao	12
1.3.2.3.1 impl	12
1.3.2.3.1.1 Class: Public RoleDAOImpl	12
1.3.2.3.1.1.1 Operation: Public List<Role> loadAll()	12
1.3.2.3.1.1.2 Operation: Public RoleDAOImpl()	12
1.3.2.3.1.1.3 Operation: Public List<Role> searchMatching()	12
1.3.2.3.1.2 Class: Public UserDAOImpl	12
1.3.2.3.1.2.1.1 Operation: Public List<User> loadAll()	12
1.3.2.3.1.2.1.2 Operation: Public List<User> searchMatching()	12
1.3.2.3.1.2.1.3 Operation: Public UserDAOImpl()	12
1.3.2.3.2 Interface: Public RoleDAO	12
1.3.2.3.2.1.1 Operation: Public List<Role> loadAll()	12
1.3.2.3.2.1.2 Operation: Public List<Role> searchMatching()	12
1.3.2.3.3 Interface: Public UserDAO	12
1.3.2.3.3.1.1 Operation: Public List<User> loadAll()	12
1.3.2.3.3.1.2 Operation: Public List<User> searchMatching()	12
1.3.2.5 delegate	13
1.3.2.5.1 Class: Public AuthenticateDelegate	13
1.3.2.5.1.1.1 Operation: Public AuthenticateDelegate()	13
1.3.2.5.1.1.2 Operation: Public User validateUserIdPassword()	13
1.3.2.7 entity	14
1.3.2.7.1 Class: Public Role	14
1.3.2.7.1.1.1 Attributes	14

1.3.2.7.1.1.2	Operation: Public getter()	14
1.3.2.7.1.1.3	Operation: Public Role()	14
1.3.2.7.1.1.4	Operation: Public setter()	14
1.3.2.7.2	Class: Public User	14
1.3.2.7.2.1.1	Attributes	14
1.3.2.7.2.1.2	Operation: Public getter()	14
1.3.2.7.2.1.3	Operation: Public setter()	14
1.3.2.7.2.1.4	Operation: Public User()	14
1.3.2.9	service	15
1.3.2.9.1	Class: Public AuthenticateService	15
1.3.2.9.1.1.1	Operation: Public AuthenticateService()	15
1.3.2.9.1.1.2	Operation: Public User validateUserIdPassword()	15
1.3.2.10	Class: Public homePage	15
1.3.2.10.1.1	Operation: Public init()	15
1.3.2.10.1.2	Operation: Public manageRadioProgram()	15
1.3.2.10.1.3	Operation: Public selectLogout()	15
1.3.2.11	Class: Public loginPage	15
1.3.2.11.1.1	Operation: Public enterUserIdAndPassword()	15
1.3.2.11.1.2	Operation: Public init()	15
1.3.2.12	Class: Public logoutPage	15
1.3.2.12.1.1	Operation: Public init()	15
1.3.3	radioprogram	16
1.3.3.1	controller	16
1.3.3.1.1	Class: Public AddEditRadioProgramCmd	16
1.3.3.1.1.1.1	Operation: Public String perform()	16
1.3.3.1.2	Class: Public DeleteRadioProgramCmd	16
1.3.3.1.2.1.1	Operation: Public String perform()	16
1.3.3.1.3	Class: Public EnterProgramDetailsCmd	16
1.3.3.1.3.1.1	Operation: Public String perform()	16
1.3.3.1.3.1.2	Operation: Private void validateFormat()	16
1.3.3.1.4	Class: Public ManageRadioProgramCmd	16
1.3.3.1.4.1.1	Operation: Public String perform()	16
1.3.3.3	dao	17
1.3.3.3.1	impl	17
1.3.3.3.1.1	Class: Public ProgramDAOImpl	17
1.3.3.3.1.1.1.1	Operation: Public bool checkProgramExists()	17
1.3.3.3.1.1.1.2	Operation: Public void createProgram()	17
1.3.3.3.1.1.1.3	Operation: Public void deleteProgram()	17
1.3.3.3.1.1.1.4	Operation: Public List<RadioProgram> findProgram()	17
1.3.3.3.1.1.1.5	Operation: Public void modifyProgram()	17
1.3.3.3.1.1.1.6	Operation: Public ProgramDAOImpl()	17
1.3.3.3.2	Interface: Public ProgramDAO	17
1.3.3.3.2.1.1	Operation: Public bool checkProgramExists()	17
1.3.3.3.2.1.2	Operation: Public void createProgram()	17
1.3.3.3.2.1.3	Operation: Public void deleteProgram()	17
1.3.3.3.2.1.4	Operation: Public List<RadioProgram> findProgram()	17
1.3.3.3.2.1.5	Operation: Public void modifyProgram()	17
1.3.3.5	delegate	18
1.3.3.5.1	Class: Public ProgramDelegate	18
1.3.3.5.1.1.1	Operation: Public void processCreate()	18

1.3.3.5.1.1.2	Operation: Public void processDelete()	18
1.3.3.5.1.1.3	Operation: Public void processModify()	18
1.3.3.5.1.1.4	Operation: Public ProgramDelegate()	18
1.3.3.5.2	Class: Public ReviewSelectRadioProgramDelegate	18
1.3.3.5.2.1.1	Operation: Public void reviewSelectRadioProgram()	18
1.3.3.5.2.1.2	Operation: Public ReviewSelectRadioProgramDelegate()	18
1.3.3.7	entity	19
1.3.3.7.1	Class: Public RadioProgram	19
1.3.3.7.1.1.1	Attributes	19
1.3.3.7.1.1.2	Operation: Public getAttribute()	19
1.3.3.7.1.1.3	Operation: Public RadioProgram()	19
1.3.3.7.1.1.4	Operation: Public setAttribute()	19
1.3.3.9	service	20
1.3.3.9.1	Class: Public ProgramService	20
1.3.3.9.1.1.1	Operation: Public void processCreate()	20
1.3.3.9.1.1.2	Operation: Public void processDelete()	20
1.3.3.9.1.1.3	Operation: Public void processModify()	20
1.3.3.9.1.1.4	Operation: Public ProgramService()	20
1.3.3.9.2	Class: Public ReviewSelectRadioProgramService	20
1.3.3.9.2.1.1	Operation: Public void reviewSelectRadioProgram()	20
1.3.3.9.2.1.2	Operation: Public ReviewSelectRadioProgramService()	20
1.3.3.10	Class: Public programListPage	20
1.3.3.10.1.1	Operation: Public void init()	20
1.3.3.10.1.2	Operation: Public selectAdd()	20
1.3.3.10.1.3	Operation: Public selectDelete()	20
1.3.3.10.1.4	Operation: Public selectEdit()	20
1.3.3.11	Class: Public setupProgramPage	20
1.3.3.11.1.1	Operation: Public enterProgramDetails()	20
1.3.3.11.1.2	Operation: Public init()	20
<b>2</b>	<b>Use-Case Realizations</b>	<b>21</b>
2.1	Design of Authenticate Users	21
2.1.1	Class Diagram for Authenticate Users	21
2.1.2	Sequence Diagram for Log-in	22
2.1.3	Sequence Diagram for Log-out	23
2.2	Design of Maintain Radio Programs	24
2.2.1	Class Diagram for Maintain Radio Programs	24
2.2.2	Sequence Diagram for Create/Modify Program	25
2.2.3	Sequence Diagram for Delete Program	26
2.3	Design of Review/Select Radio Program	27
2.3.1	Class Diagram for Review/Select Program	27
2.3.2	Sequence Diagram for Review/Select Program	28



# 1 Design Objects

## 1.1 Class Diagram for Entity Classes (Users)

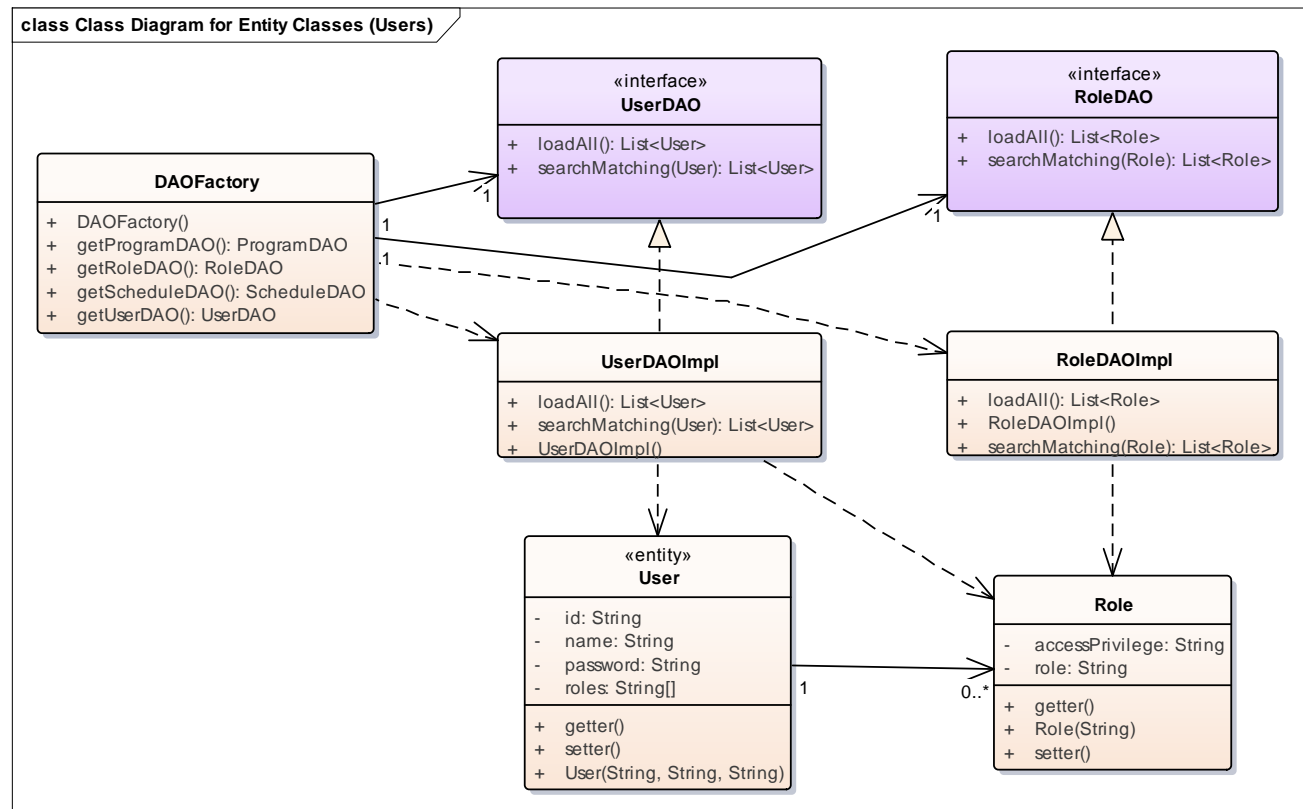


Figure 1: Class Diagram for Entity Classes (Users)

## 1.2 Class Diagram of Entity Classes (Radio Programs)

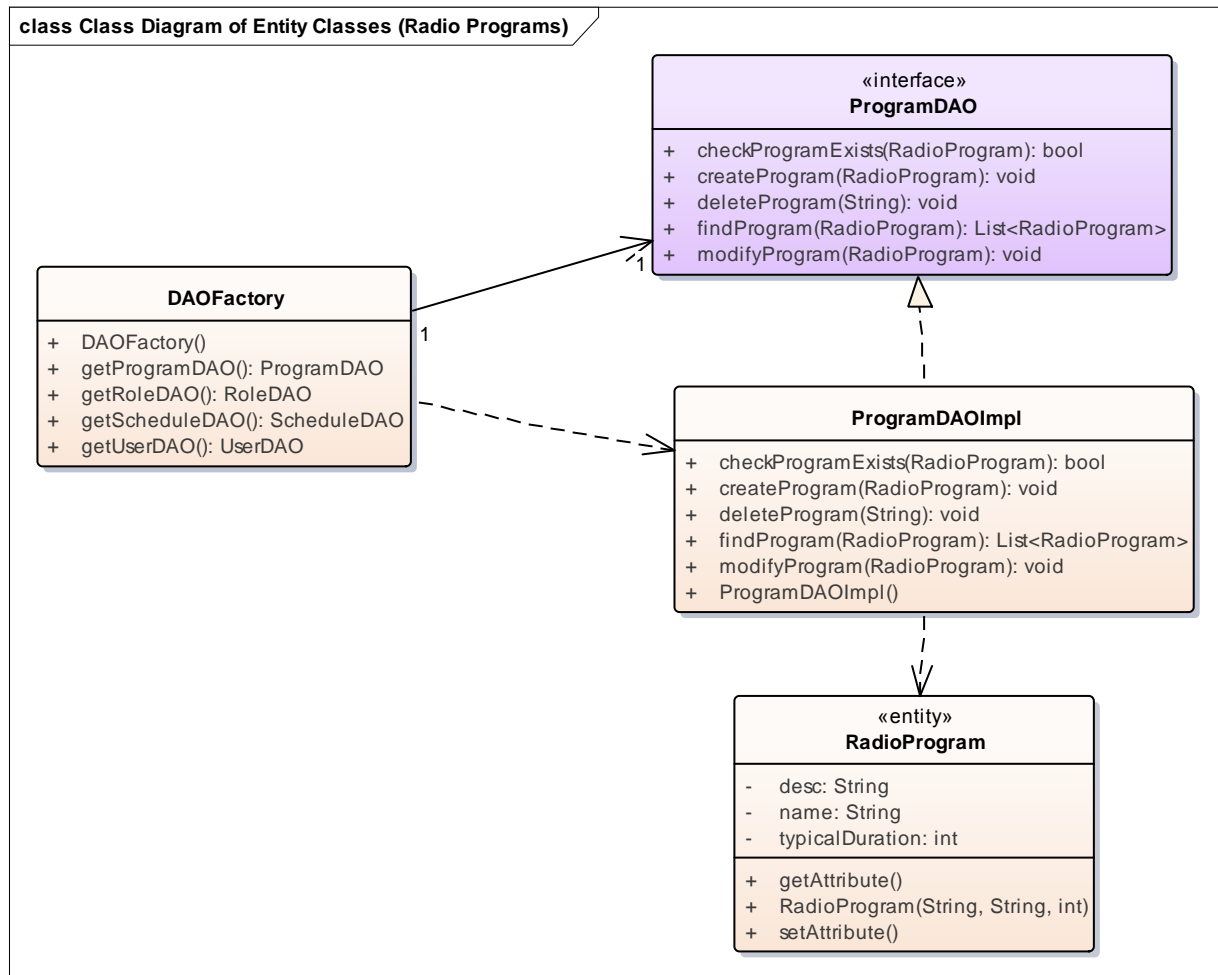


Figure 2: Class Diagram of Entity Classes (Radio Programs)

## 1.3 phoenix



## 1.3.1 core

### 1.3.1.1 controller

#### 1.3.1.1.1 Class: Public PhoenixFrontController

The Nocturne Application Controller that is the entry point of all the HTTP requests.

##### 1.3.1.1.1.1 Operation: Public void doGet()

Parameter Name	Type	Documentation
request	HttpServletRequest	Refer to Java API specification.
response	HttpServletResponse	Refer to Java API specification.

##### 1.3.1.1.1.2 Operation: Public void doPost()

Parameter Name	Type	Documentation
request	HttpServletRequest	Refer to Java API specification.
response	HttpServletResponse	Refer to Java API specification.

##### 1.3.1.1.1.3 Operation: Public void processRequest()

Processes a HTTP request and forwards to another page. The processing involves mapping the request path, executing the corresponding command object and obtaining the response path.

Parameter Name	Type	Documentation
request	HttpServletRequest	Refer to Java API specification.
response	HttpServletResponse	Refer to Java API specification.

## 1.3.1.2 dao

### 1.3.1.2.1 Class: Public DAOFactory

The factory that instantiates all the DAO objects.

#### 1.3.1.2.1.1.1 Operation: Public DAOFactory()

The constructor of the factory.

#### 1.3.1.2.1.1.2 Operation: Public ProgramDAO getProgramDAO()

Returns the DAO for radio programs.

#### 1.3.1.2.1.1.3 Operation: Public RoleDAO getRoleDAO()

Returns the DAO for user roles.

#### 1.3.1.2.1.1.4 Operation: Public ScheduleDAO getScheduleDAO()

Returns the DAO for schedule-related objects.

#### 1.3.1.2.1.1.5 Operation: Public UserDAO getUserDAO()

Returns the DAO for users.

## 1.3.2 authenticate

### 1.3.2.1 controller

#### 1.3.2.1.1 Class: Public LoginCmd

This command captures the Login request.

##### 1.3.2.1.1.1 Operation: Public String perform()

Performs the login request by passing the input user id and password to the Business Layer, setting the authenticated User object in HTTP session and forwarding to the home page.

Parameter Name	Type	Documentation
path	String	Refer to Nocturne Application Controller framework.
req	HttpServletRequest	Refer to Java API specification.
resp	HttpServletResponse	Refer to Java API specification.

#### 1.3.2.1.2 Class: Public LogoutCmd

This command captures the Logout request.

##### 1.3.2.1.2.1 Operation: Public String perform()

Performs the login request by invalidating the HTTP session and forwarding to the logout page.

Parameter Name	Type	Documentation
path	String	Refer to Nocturne Application Controller framework.
req	HttpServletRequest	Refer to java API specification.
resp	HttpServletResponse	Refer to java API specification.

### 1.3.2.3 dao

#### 1.3.2.3.1 impl

##### 1.3.2.3.1.1 Class: Public RoleDAOImpl

The implementation of the DAO for user roles. Refer to the documentation of the interface for the details of the class.

##### 1.3.2.3.1.1.1 Operation: Public List<Role> loadAll()

Refer to the documentation of interface RoleDAO.

##### 1.3.2.3.1.1.1.2 Operation: Public RoleDAOImpl()

The constructor of the DAO implementation.

##### 1.3.2.3.1.1.1.3 Operation: Public List<Role> searchMatching()

Refer to the documentation of interface RoleDAO.

Parameter Name	Type	Documentation
input	Role	

##### 1.3.2.3.1.2 Class: Public UserDAOImpl

The implementation of the DAO for users. Refer to the documentation of the interface for the details of the class.

##### 1.3.2.3.1.2.1.1 Operation: Public List<User> loadAll()

Refer to the documentation of interface UserDAO.

##### 1.3.2.3.1.2.1.2 Operation: Public List<User> searchMatching()

Refer to the documentation of interface UserDAO.

Parameter Name	Type	Documentation
input	User	

##### 1.3.2.3.1.2.1.3 Operation: Public UserDAOImpl()

The constructor of the DAO implementation.

#### 1.3.2.3.2 Interface: Public RoleDAO

The interface of the DAO for user roles.

##### 1.3.2.3.2.1.1 Operation: Public List<Role> loadAll()

Reads all contents from database table and build a List of user objects. Note that this operation will consume huge amount of resources if the table has a lot of rows. It should only be used when the target table has only small amount of data.

##### 1.3.2.3.2.1.2 Operation: Public List<Role> searchMatching()

Searches the database for value objects that match the criteria specified by the attributes of input value object. Attributes with null values are excluded as criteria. The result is a list of value objects.

Parameter Name	Type	Documentation
input	Role	The input user role object.

#### 1.3.2.3.3 Interface: Public UserDAO

The interface of the DAO for users.

##### 1.3.2.3.3.1.1 Operation: Public List<User> loadAll()

Reads all contents from database table and build a List of user objects. Note that this operation will consume huge amount of resources if the table has a lot of rows. It should only be used when the target table has only small amount of data.

##### 1.3.2.3.3.1.2 Operation: Public List<User> searchMatching()

Parameter Name	Type	Documentation
input	User	The input user object.

## 1.3.2.5 delegate

### 1.3.2.5.1 Class: Public AuthenticateDelegate

The delegate of the Authenticate Users use case.

#### 1.3.2.5.1.1.1 Operation: Public AuthenticateDelegate()

The constructor of the delegate.

#### 1.3.2.5.1.1.2 Operation: Public User validateUserIdPassword()

Validates the user id and password of the user.

Parameter Name	Type	Documentation
input	User	The input user object specifying the user id and password.

## 1.3.2.7 entity

### 1.3.2.7.1 Class: Public Role

This object represents the access privileges of a role that is assigned to users.

#### 1.3.2.7.1.1.1 Attributes

Scope	Name	Type	Documentation
private	accessPrivilege	String	The access privileges for the user role.
private	role	String	The name of the user role.

#### 1.3.2.7.1.1.2 Operation: Public getter()

The generic representation of getter operations.

#### 1.3.2.7.1.1.3 Operation: Public Role()

The constructor of the user role.

Parameter Name	Type	Documentation
role	String	The name of the user role.

#### 1.3.2.7.1.1.4 Operation: Public setter()

The generic representation of setter operations.

### 1.3.2.7.2 Class: Public User

This entity object represents the common user characteristics.

#### 1.3.2.7.2.1.1 Attributes

Scope	Name	Type	Documentation
private	id	String	The user id of the user.
private	name	String	The name of the user.
private	password	String	The password of the user.
private	roles	String[]	The names of all roles assigned to the user.

#### 1.3.2.7.2.1.2 Operation: Public getter()

The generic representation of getter operations.

#### 1.3.2.7.2.1.3 Operation: Public setter()

The generic representation of setter operations.

#### 1.3.2.7.2.1.4 Operation: Public User()

The constructor of the user.

Parameter Name	Type	Documentation
id	String	The user id of the user.
name	String	The name of the user.
password	String	The password of the user.

## 1.3.2.9 service

### 1.3.2.9.1 Class: Public AuthenticateService

The service of the Authenticate Users use case. Refer to the documentation of the delegate for the details of the class.

#### 1.3.2.9.1.1.1 Operation: Public AuthenticateService()

The constructor of the service.

#### 1.3.2.9.1.1.2 Operation: Public User validateUserIdPassword()

Refer to the documentation of the delegate.

Parameter Name	Type	Documentation
input	User	

### 1.3.2.10 Class: Public homePage

This boundary object represents the main screen of the PRMS application.

#### 1.3.2.10.1.1 Operation: Public init()

Initializes the JSP page.

#### 1.3.2.10.1.2 Operation: Public manageRadioProgram()

User request to manage radio programs.

#### 1.3.2.10.1.3 Operation: Public selectLogout()

User request to logout from the system.

### 1.3.2.11 Class: Public loginPage

This boundary object represents the screen used by users to log-in to the system.

#### 1.3.2.11.1.1 Operation: Public enterUserIdAndPassword()

User request to login to the system.

#### 1.3.2.11.1.2 Operation: Public init()

Initializes the JSP page.

### 1.3.2.12 Class: Public logoutPage

This boundary object represents a confirmation dialog to prompt if the user wants to logout.

#### 1.3.2.12.1.1 Operation: Public init()

Initializes the JSP page.

## 1.3.3 radioprogram

### 1.3.3.1 controller

#### 1.3.3.1.1 Class: Public AddEditRadioProgramCmd

This command captures the request to add a new radio program or edit an existing radio program.

##### 1.3.3.1.1.1 Operation: Public String perform()

Performs the request forwarding to the setup program page for creation or editing of radio program.

Parameter Name	Type	Documentation
path	String	Refer to Nocturne Application Controller framework.
req	HttpServletRequest	Refer to Java API specification.
resp	HttpServletResponse	Refer to Java API specification.

#### 1.3.3.1.2 Class: Public DeleteRadioProgramCmd

This command captures the request to delete an existing radio program.

##### 1.3.3.1.2.1 Operation: Public String perform()

Performs the request by passing the name of the radio program to delete to the Business Layer, invoking the Review/Select Radio Program use case to display the resultant list of radio programs.

Parameter Name	Type	Documentation
path	String	Refer to Nocturne Application Controller framework.
req	HttpServletRequest	Refer to Java API specification.
resp	HttpServletResponse	Refer to Java API specification.

#### 1.3.3.1.3 Class: Public EnterProgramDetailsCmd

This command captures the request to submit the details of a radio program entered by the actor.

##### 1.3.3.1.3.1 Operation: Public String perform()

Performs the request by passing the input details of the radio program to be created/modified to the Business Layer, invoking the Review/Select Radio Program use case to display the resultant list of radio programs.

Parameter Name	Type	Documentation
path	String	Refer to Nocturne Application Controller framework.
req	HttpServletRequest	Refer to Java API specification.
resp	HttpServletResponse	Refer to Java API specification.

##### 1.3.3.1.3.1.2 Operation: Private void validateFormat()

Validates the format of details entered by the actor.

#### 1.3.3.1.4 Class: Public ManageRadioProgramCmd

This command captures the request to manage radio programs.

##### 1.3.3.1.4.1 Operation: Public String perform()

Performs the request by invoking the Review/Select Radio Program use case to display the current list of radio programs.

Parameter Name	Type	Documentation
path	String	Refer to Nocturne Application Controller framework.
req	HttpServletRequest	Refer to Java API specification.
resp	HttpServletResponse	Refer to Java API specification.



### 1.3.3.3 dao

#### 1.3.3.3.1 impl

##### 1.3.3.3.1.1 Class: Public ProgramDAOImpl

The implementation of the DAO for radio programs. Refer to the documentation of the interface for the details of the class.

##### 1.3.3.3.1.1.1.1 Operation: Public bool checkProgramExists()

Refer to the documentation of interface ProgramDAO.

Parameter Name	Type	Documentation
input	RadioProgram	

##### 1.3.3.3.1.1.1.2 Operation: Public void createProgram()

Refer to the documentation of interface ProgramDAO.

Parameter Name	Type	Documentation
input	RadioProgram	

##### 1.3.3.3.1.1.1.3 Operation: Public void deleteProgram()

Refer to the documentation of interface ProgramDAO.

Parameter Name	Type	Documentation
name	String	

##### 1.3.3.3.1.1.1.4 Operation: Public List<RadioProgram> findProgram()

Refer to the documentation of interface ProgramDAO.

Parameter Name	Type	Documentation
input	RadioProgram	

##### 1.3.3.3.1.1.1.5 Operation: Public void modifyProgram()

Refer to the documentation of interface ProgramDAO.

Parameter Name	Type	Documentation
input	RadioProgram	

##### 1.3.3.3.1.1.1.6 Operation: Public ProgramDAOImpl()

The constructor of the DAO implementation.

#### 1.3.3.3.2 Interface: Public ProgramDAO

The interface of the DAO for radio programs.

##### 1.3.3.3.2.1.1 Operation: Public bool checkProgramExists()

Checks if a radio program is existing in the database by matching the name of the input radio program. The result is a boolean value.

Parameter Name	Type	Documentation
input	RadioProgram	The input radio program.

##### 1.3.3.3.2.1.2 Operation: Public void createProgram()

Create a new database record for the input radio program.

Parameter Name	Type	Documentation
input	RadioProgram	The input radio program.

##### 1.3.3.3.2.1.3 Operation: Public void deleteProgram()

Delete an existing database record matching the input name of nradio program.

Parameter Name	Type	Documentation
name	String	The name of the radio program.

##### 1.3.3.3.2.1.4 Operation: Public List<RadioProgram> findProgram()

Searches for database records matching the attribute values of the input radio program. A list of radio programs is returned.

Parameter Name	Type	Documentation
input	RadioProgram	The input radio program.

##### 1.3.3.3.2.1.5 Operation: Public void modifyProgram()

Update an existing database record with the attribute values of the input radio program.

Parameter Name	Type	Documentation
input	RadioProgram	The input radio program.

### 1.3.3.4 delegate

#### 1.3.3.4.1 Class: Public ProgramDelegate

The delegate of the Maintain Radio Programs use case.

##### 1.3.3.4.1.1.1 Operation: Public void processCreate()

Creates a new radio program based on the input radio program.

Parameter Name	Type	Documentation
input	RadioProgram	The input radio program.

##### 1.3.3.4.1.1.2 Operation: Public void processDelete()

Deletes the radio program with the input name.

Parameter Name	Type	Documentation
name	String	The name of the radio program.

##### 1.3.3.4.1.1.3 Operation: Public void processModify()

Modifies an existing radio program based on the input radio program.

Parameter Name	Type	Documentation
input	RadioProgram	The input radio program.

##### 1.3.3.4.1.1.4 Operation: Public ProgramDelegate()

The constructor of the delegate.

#### 1.3.3.4.2 Class: Public ReviewSelectRadioProgramDelegate

The delegate of the Review/Select Radio Program use case.

##### 1.3.3.4.2.1.1 Operation: Public void reviewSelectRadioProgram()

Starts the selection of a radio program from the list of existing radio programs.

##### 1.3.3.4.2.1.2 Operation: Public ReviewSelectRadioProgramDelegate()

The constructor of the delegate.

### 1.3.3.6 entity

#### 1.3.3.6.1 Class: Public RadioProgram

The entity object represents a radio program as defined by the radio station manager.

##### 1.3.3.6.1.1.1 Attributes

Scope	Name	Type	Documentation
private	desc	String	The description of a radio program.
private	name	String	The name of a radio program.
private	typicalDuration	int	The typical duration of a radio program.

##### 1.3.3.6.1.1.2 Operation: Public getAttribute()

The generic representation of getter operations.

##### 1.3.3.6.1.1.3 Operation: Public RadioProgram()

The constructor of the class.

Parameter Name	Type	Documentation
desc	String	The description of the radio program.
name	String	The name of the radio program.
typicalDuration	int	The typical duration of the radio program.

##### 1.3.3.6.1.1.4 Operation: Public setAttribute()

The generic representation of setter operations.

### 1.3.3.8 service

#### 1.3.3.8.1 Class: Public ProgramService

The service of the Maintain Radio Programs use case.

##### 1.3.3.8.1.1.1 Operation: Public void processCreate()

Refer to the documentation of the delegate.

Parameter Name	Type	Documentation
input	RadioProgram	

##### 1.3.3.8.1.1.2 Operation: Public void processDelete()

Refer to the documentation of the delegate.

Parameter Name	Type	Documentation
name	String	

##### 1.3.3.8.1.1.3 Operation: Public void processModify()

Refer to the documentation of the delegate.

Parameter Name	Type	Documentation
input	RadioProgram	

##### 1.3.3.8.1.1.4 Operation: Public ProgramService()

The constructor of the service.

#### 1.3.3.8.2 Class: Public ReviewSelectRadioProgramService

This service of the Review / Select Radio Program use case.

##### 1.3.3.8.2.1.1 Operation: Public void reviewSelectRadioProgram()

Refer to the documentation of the delegate.

##### 1.3.3.8.2.1.2 Operation: Public ReviewSelectRadioProgramService()

The constructor of the service.

### 1.3.3.9 Class: Public programListPage

This boundary object represents the screen used to list radio programs.

#### 1.3.3.9.1.1 Operation: Public void init()

Initializes the JSP page.

#### 1.3.3.9.1.2 Operation: Public selectAdd()

User request to create a new radio program.

#### 1.3.3.9.1.3 Operation: Public selectDelete()

User request to delete an existing radio program.

#### 1.3.3.9.1.4 Operation: Public selectEdit()

User request to modify an existing radio program.

### 1.3.3.10 Class: Public setupProgramPage

This boundary object represents the screen used to maintain radio programs.

#### 1.3.3.10.1.1 Operation: Public enterProgramDetails()

User request to submit the details of a new or modified radio program.

#### 1.3.3.10.1.2 Operation: Public init()

Initializes the JSP page.

## 2 Use-Case Realizations

### 2.1 Design of Authenticate Users

#### 2.1.1 Class Diagram for Authenticate Users

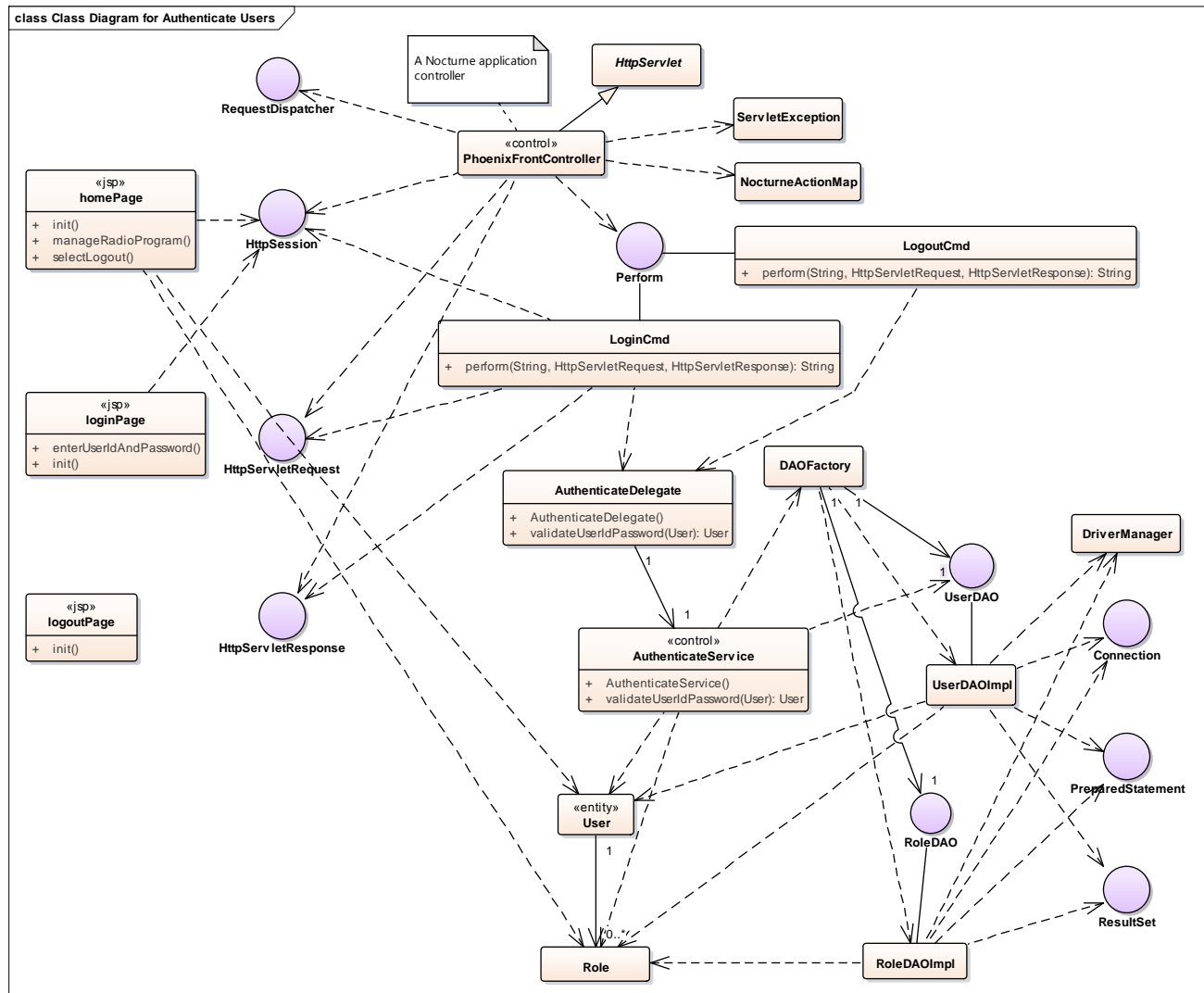


Figure 3: Class Diagram for Authenticate Users

2.1.2 Sequence Diagram for Log-in

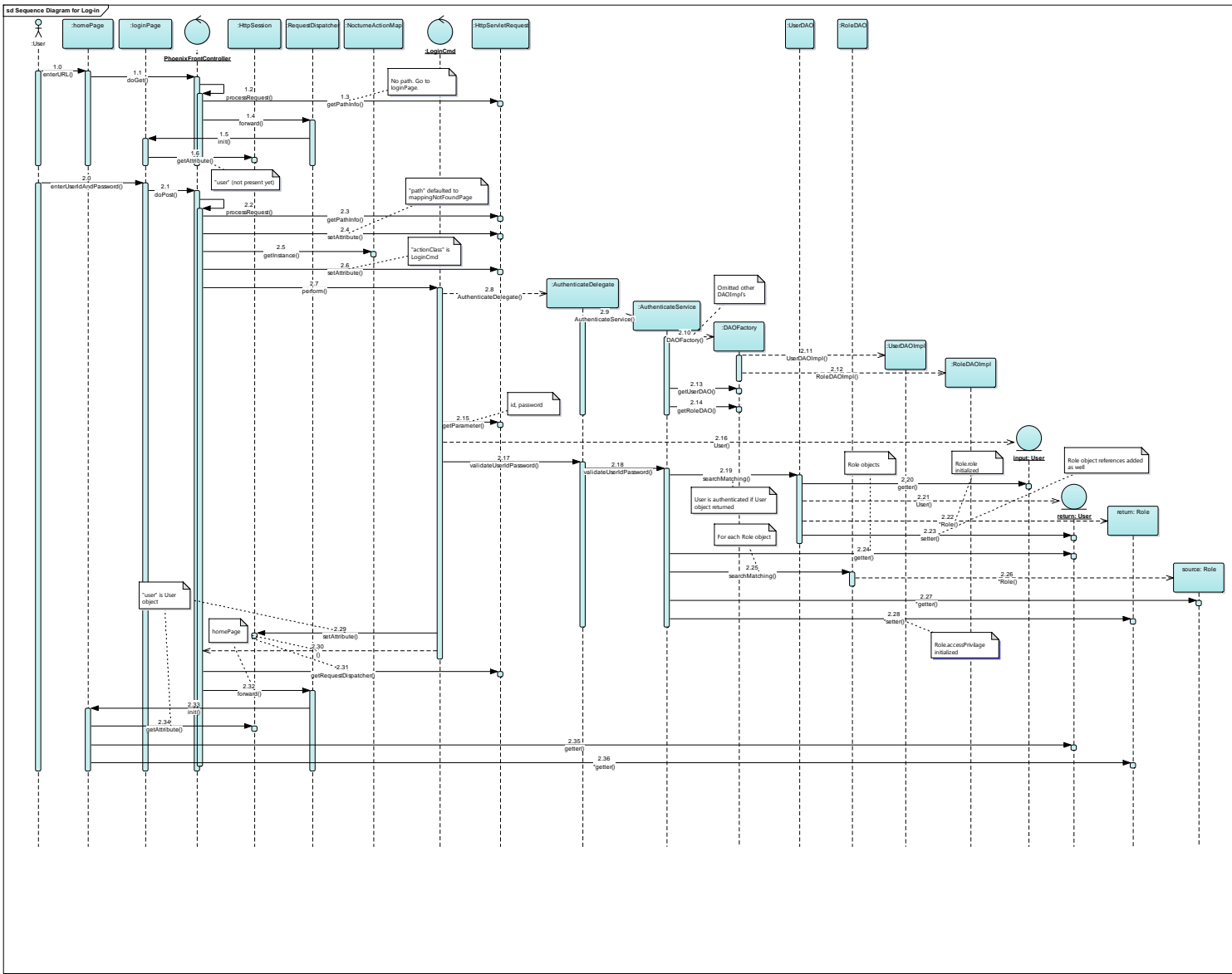


Figure 4: Sequence Diagram for Log-in

## 2.1.3 Sequence Diagram for Log-out

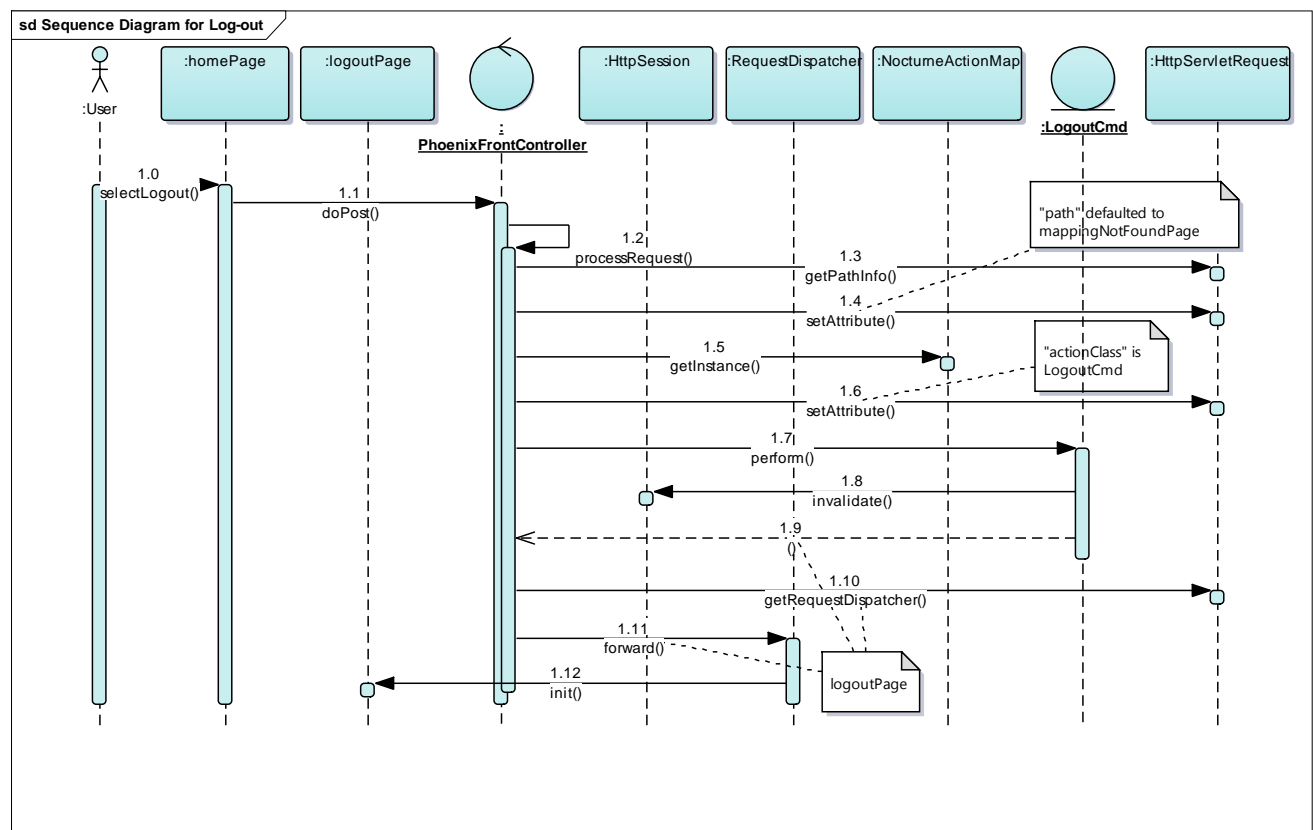


Figure 5: Sequence Diagram for Log-out

## 2.2 Design of Maintain Radio Programs

### 2.2.1 Class Diagram for Maintain Radio Programs

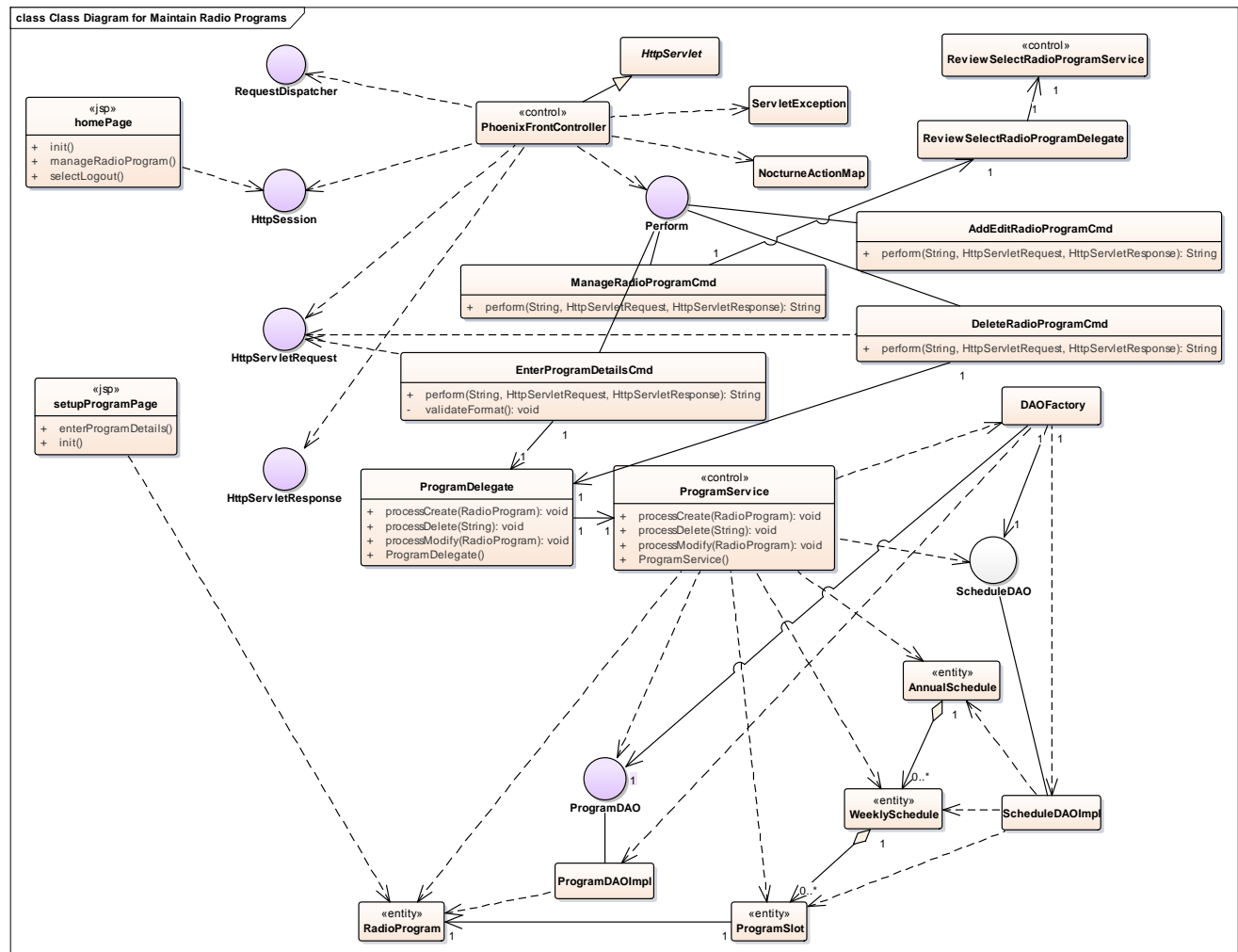


Figure 6: Class Diagram for Maintain Radio Programs



### 2.2.2 Sequence Diagram for Create/Modify Program

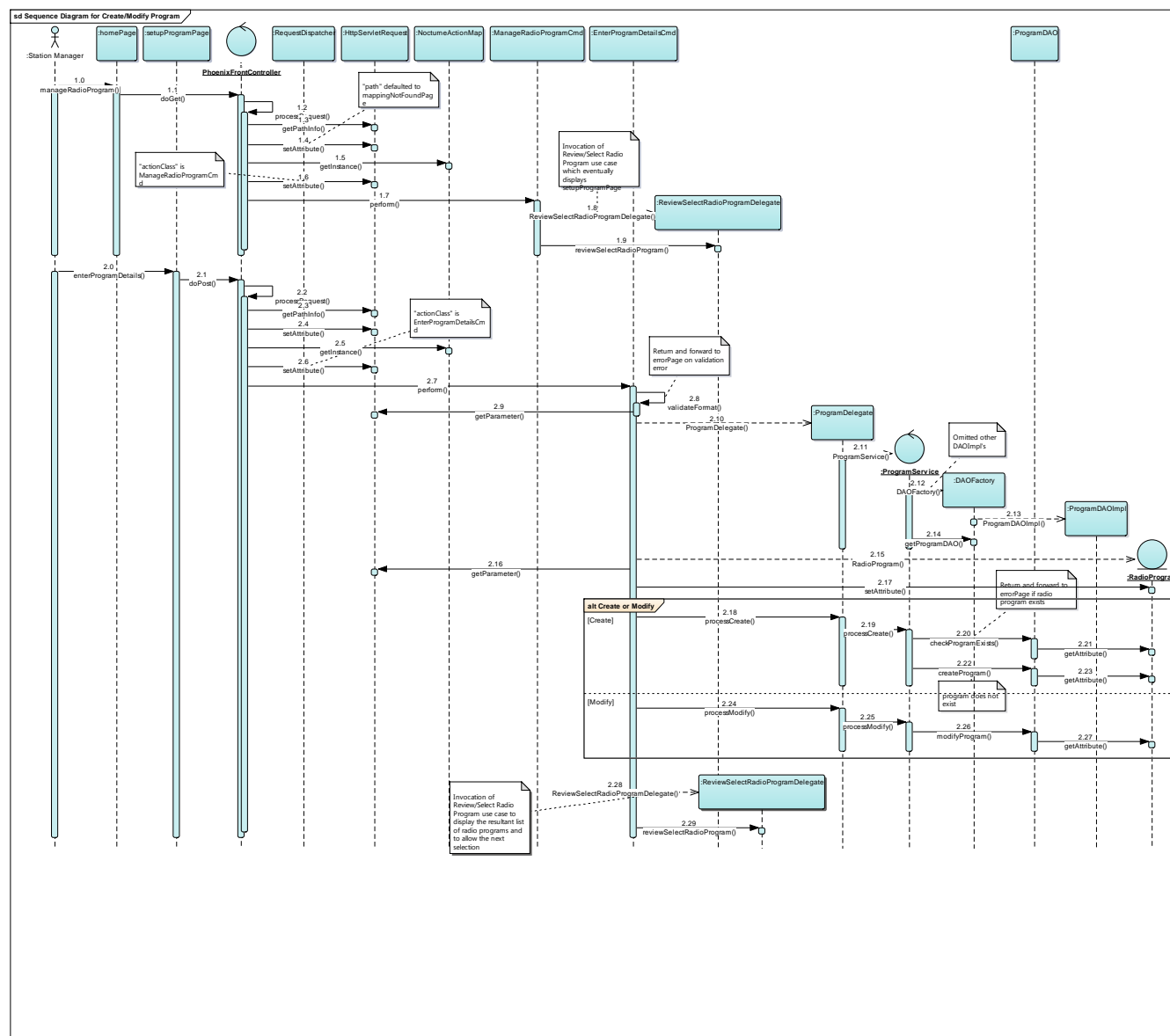


Figure 7: Sequence Diagram for Create/Modify Program

2.2.3 Sequence Diagram for Delete Program

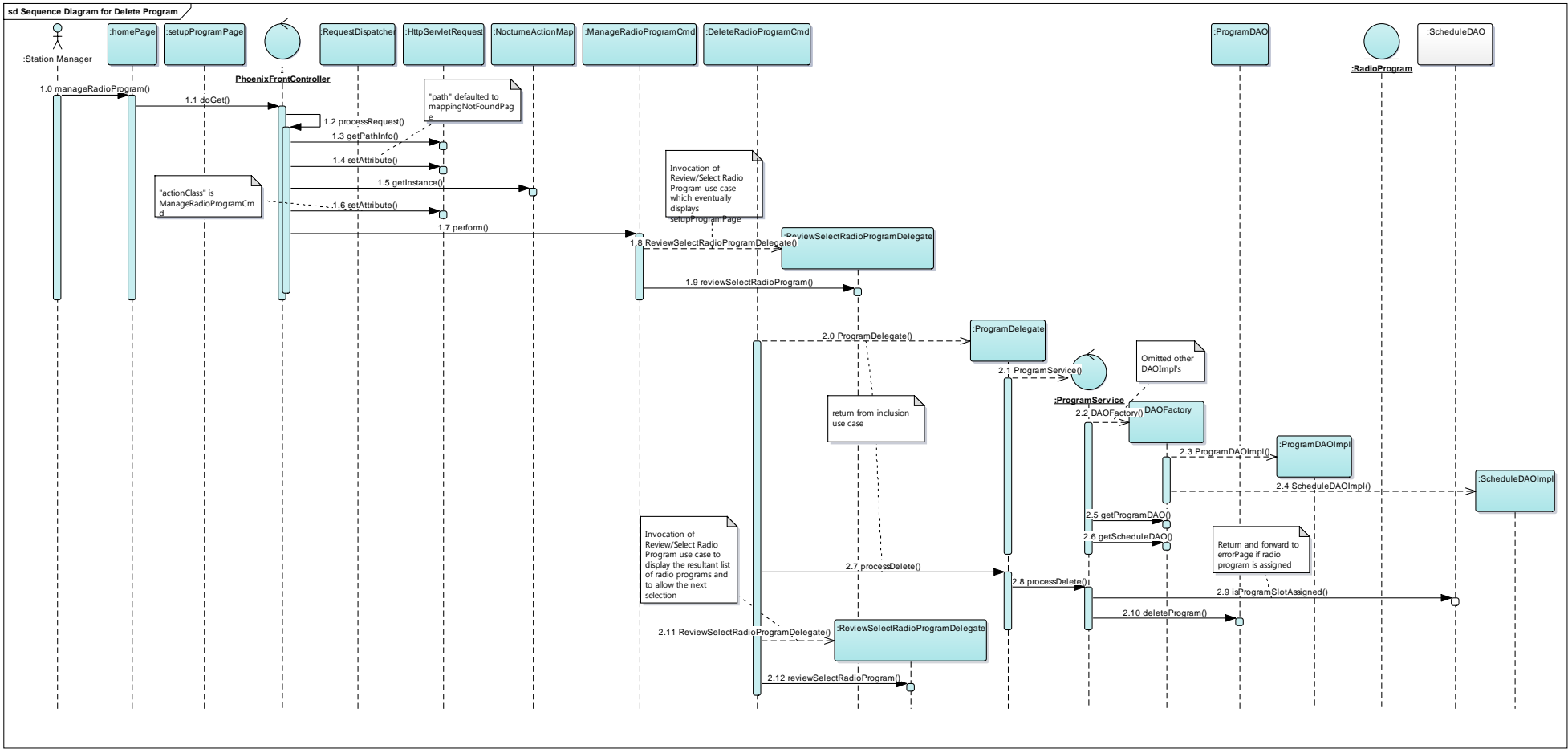


Figure 8: Sequence Diagram for Delete Program

## 2.3 Design of Review/Select Radio Program

### 2.3.1 Class Diagram for Review/Select Program

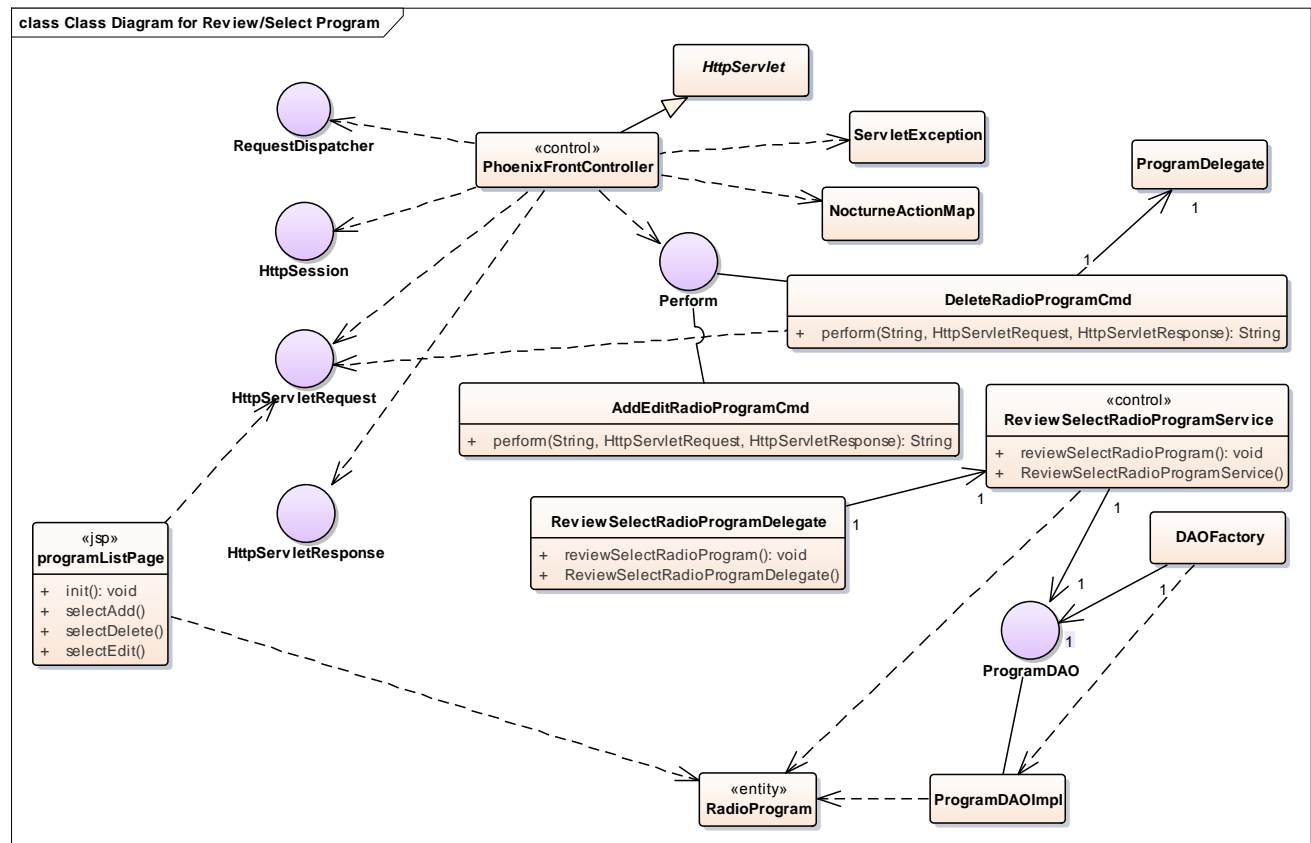


Figure 9: Class Diagram for Review/Select Program

2.3.2 Sequence Diagram for Review/Select Program

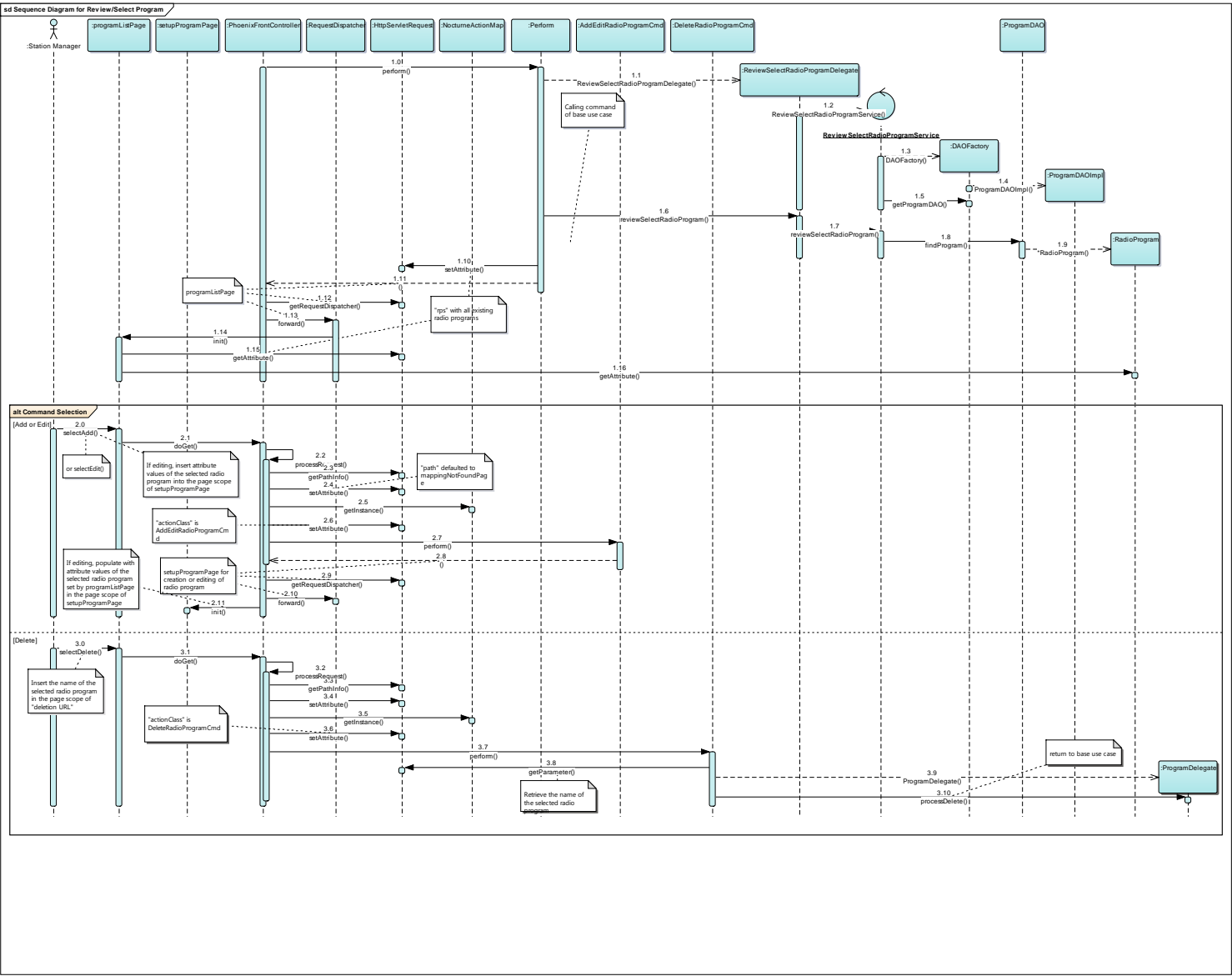


Figure 10: Sequence Diagram for Review/Select Program