



# Online Customer Care and Service Centre

Project Id : A073 <SC./D>

Release Id :

## Application Design



	Prepared By	Reviewed by	Approved By
Name	Linjo Rejoy		
Role	Programmer Analyst Trainee		
Signature	Linjo		
Date	18 Dec 2020		





# Contents

<b>1.0 Introduction .....</b>	<b>4</b>
<b>1.1 Purpose of the Document.....</b>	<b>4</b>
<b>1.2 Scope .....</b>	<b>4</b>
1.2.1 In Scope .....	4
1.2.2 Out Scope .....	4
<b>1.3 Intended Audience .....</b>	<b>4</b>
<b>2.0 Conventions and Standards .....</b>	<b>5</b>
<b>3.0 Assumptions, Dependencies and Risks .....</b>	<b>5</b>
<b>4.0 System Architecture.....</b>	<b>6</b>
<b>5.0 Use Case Realization .....</b>	<b>7</b>
<b>5.1 Flow of Events .....</b>	<b>7</b>
5.1.1 Login Sequence .....	7
5.1.2 Admin Sequence .....	7
5.1.3 Analyst Sequence .....	8
5.1.4 Email Analyst Sequence .....	9
5.1.5 Admin Controller Feedback.....	9
5.1.6 Admin Add Feedback .....	10
5.1.7 Admin Email to Analyst .....	10
<b>5.2 System Object Model .....</b>	<b>11</b>
5.2.1 Model Classes.....	11
5.2.2 Repositories .....	13
5.2.3 Validator Classes .....	13
5.2.4 Service Classes.....	14
5.2.5 Controller Classes.....	14
5.2.6 Security .....	15
5.2.7 Exporter Classes .....	15
5.2.8 Application Classes .....	15
<b>5.3 UI Controls .....</b>	<b>16</b>
5.3.1 User Registration.....	16
5.3.2 Analyst Registration .....	17
5.3.3 Login Admin .....	18
5.3.4 Login User .....	18
5.3.5 Login Analyst.....	18
5.3.6 Forgot User ID-User.....	19
5.3.7 Forgot Password-User .....	19
5.3.8 Reset Password-User .....	20
5.3.9 Forgot Analyst ID-Analyst .....	20
5.3.10 Forgot Password-Analyst .....	21
5.3.11 Reset Password-Analyst .....	21
5.3.12 Create Complaint-User .....	22
5.3.13 Complaint View-User .....	22
5.3.14 Complaint View-Analyst .....	23
5.3.15 Complaint View-Admin.....	24
5.3.16 Add Feedback-Admin.....	24





5.3.17	Email View-Analyst.....	25
5.3.18	Email-Admin .....	25
<b>5.4 UI Design .....</b>		<b>27</b>
5.4.1	Admin Login .....	27
5.4.2	Register User/Analyst .....	27
5.4.3	Register Analyst .....	28
5.4.4	Create Complaint User.....	28
5.4.5	View Complaint User.....	29
5.4.6	All Complaints .....	29
5.4.7	User to Analyst Mail .....	30
5.4.8	5.4.6 Analyst/Admin Home.....	30
5.4.9	Admin Home.....	31
5.4.10	View All Complaints Analyst/Admin .....	31
5.4.11	View Complaint Analyst/Admin .....	32
5.4.12	Create Feedback Admin .....	33
5.4.13	Analyst Emails.....	33
5.4.14	Analyst Email View.....	34
<b>5.5 Component Description.....</b>		<b>35</b>
5.5.1	Controller Classes .....	35
5.5.2	Service Classes.....	49
<b>6.0 Database Design.....</b>		<b>63</b>
<b>6.1 Table Structure .....</b>		<b>63</b>
6.1.1	User Table .....	63
<b>6.2 E- R Diagram (Relationship between Tables).....</b>		<b>69</b>
<b>6.3 Standards and Guidelines Followed .....</b>		<b>70</b>
6.3.1	JAVA .....	70
<b>7.0 Change Log.....</b>		<b>70</b>





# 1.0 Introduction

## 1.1 Purpose of the Document

The purpose of the detail design document is to define the detailed design for all the components of Online Customer Care and Service Centre that are specified in the Business Requirement document. All the low-level components, interfaces and integrations are detailed out in this document.

## 1.2 Scope

### 1.2.1 In Scope

- User Register/Login/Create Complaint/View Complaint
- Analyst Register/Login/View Complaint
- Admin Login/View Complaint/Download
- Create/View RL
- Create/View Status Report

### 1.2.2 Out Scope

- Migration of current employee list into the application

## 1.3 Intended Audience

The intended audience for this document is

- Developers, testers, other associates in the project
- Client
- Delivery Assurance Group, Process Engineering group





## 2.0 Conventions and Standards

*Online Customer Care and Services system will follow Oracle's java coding standard*

## 3.0 Assumptions, Dependencies and Risks

### Assumptions

- The Customer Care Application will employ 3 types of users. a normal User, an Analyst, and an Admin.
- Once the User after registration login to the webapp, He/she can submit a complaint.
- Once submitted the admin can forward the complaint and assign it to any of the existing Analysts'.
- The Analyst can view the issue and do the necessary action and mark it as resolved.
- If the issue requires more attention, the analyst can forward it to the next level analyst.

### Dependencies

- MySQL 8.0 Relational Database
- JDK 1.8 or above
- Spring Boot starter web
- Spring boot starter security
- Tomcat Server
- Bootstrap
- JQuery
- Junit and Mockito Testing
- Spring Core

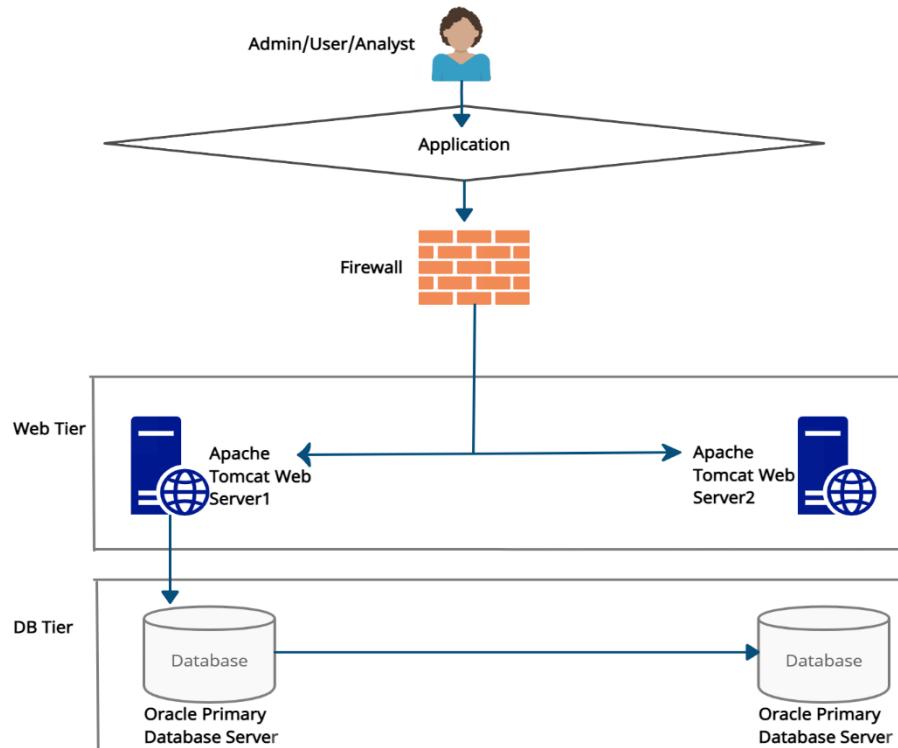
### Risks

- 1.0 The current system design application can perform optimally with less than 1000 employees.





## 4.0 System Architecture



We have used Spring security to perform authorization and authentication.

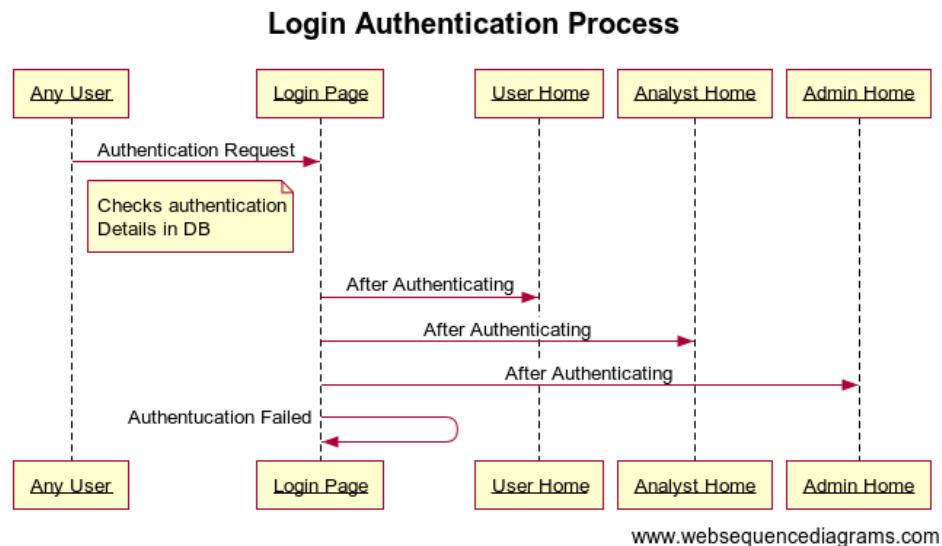




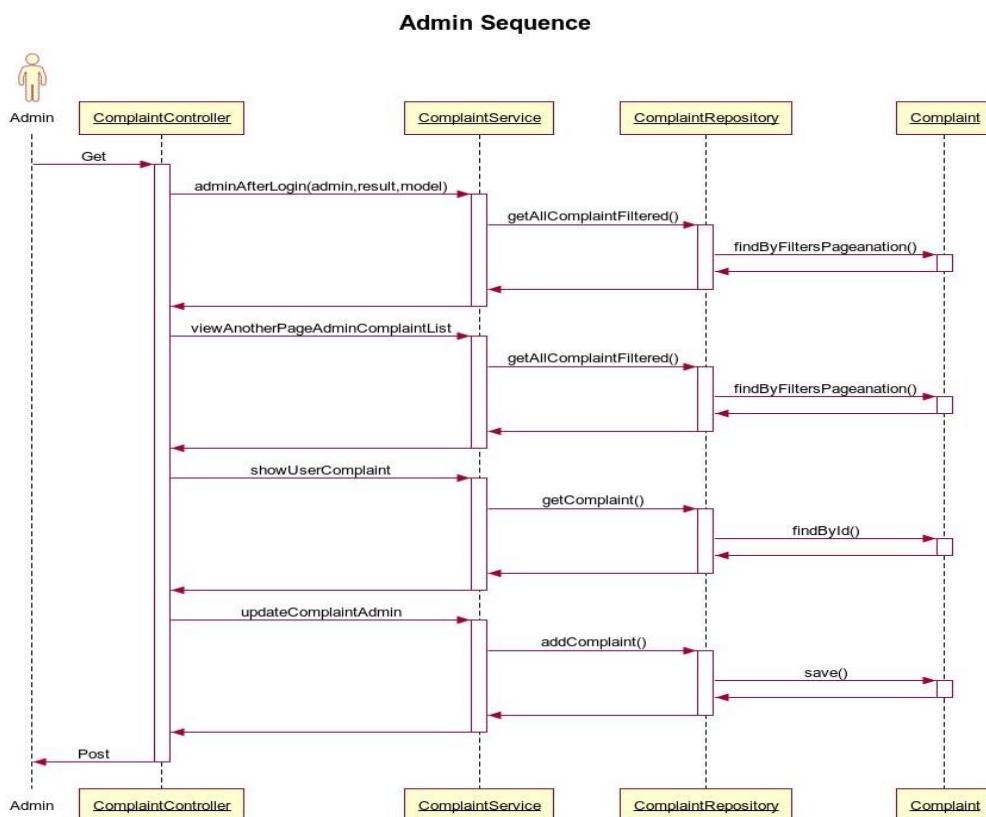
## 5.0 Use Case Realization

### 5.1 Flow of Events

#### 5.1.1 Login Sequence

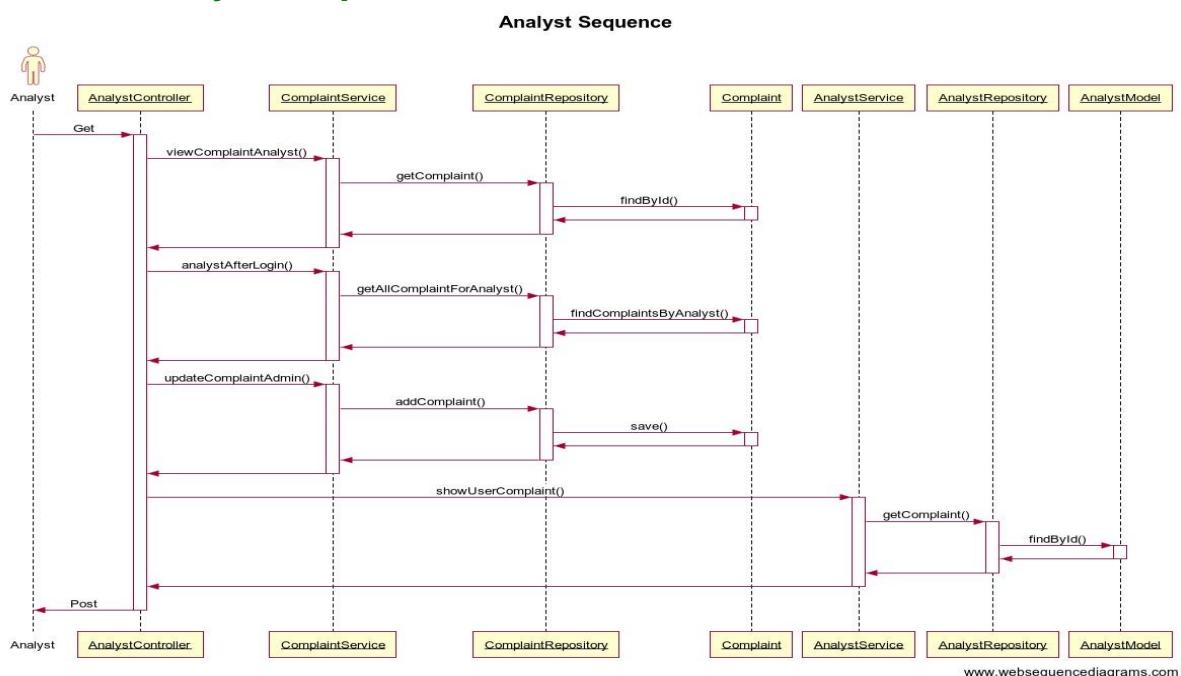


#### 5.1.2 Admin Sequence

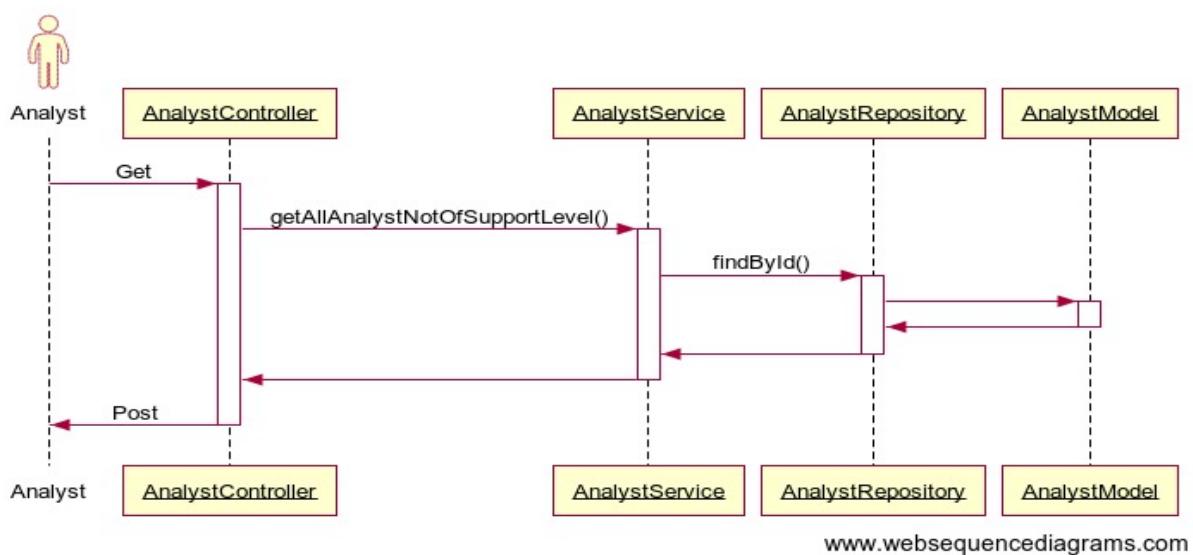




### 5.1.3 Analyst Sequence



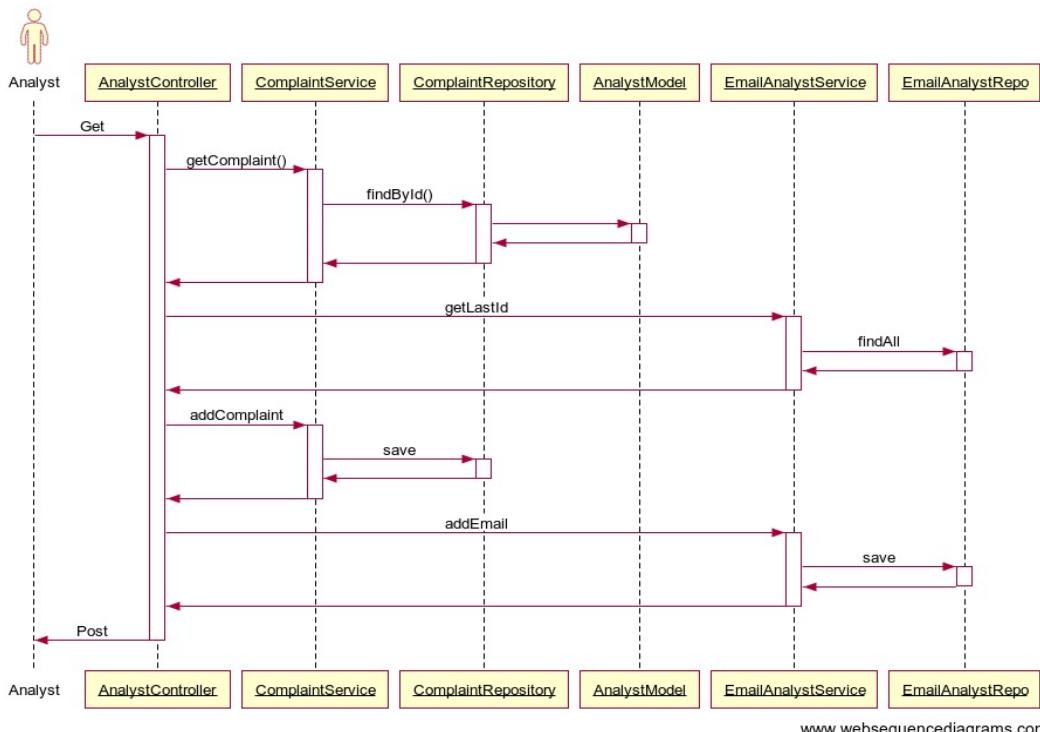
### Analyst Sequence





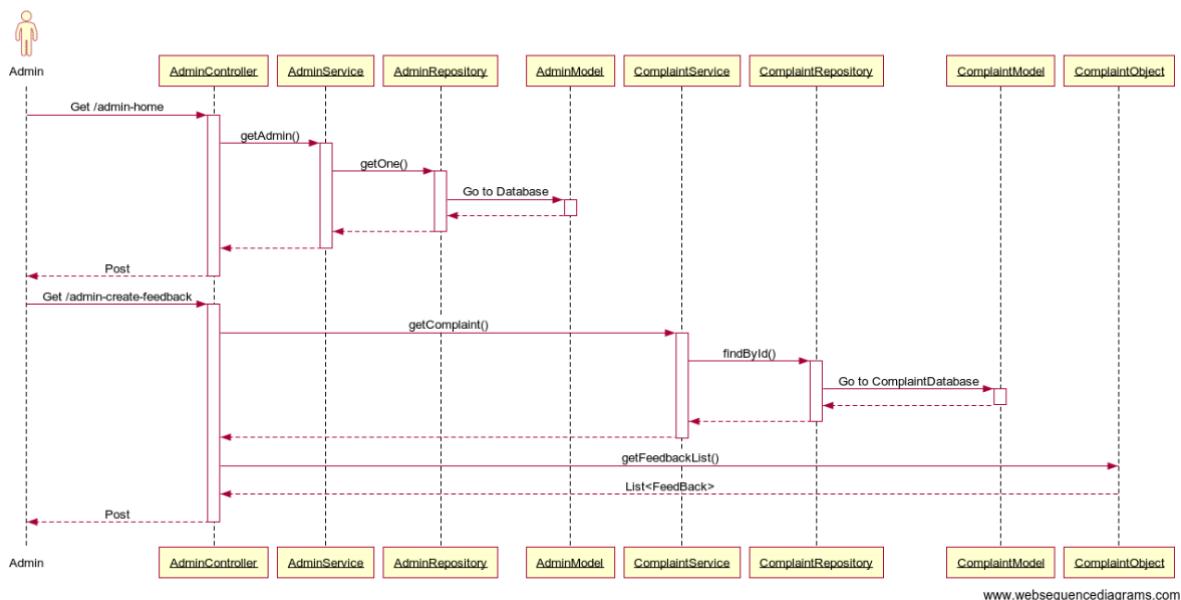
## 5.1.4 Email Analyst Sequence

Email Analyst Sequence - Update Complaint



## 5.1.5 Admin Controller Feedback

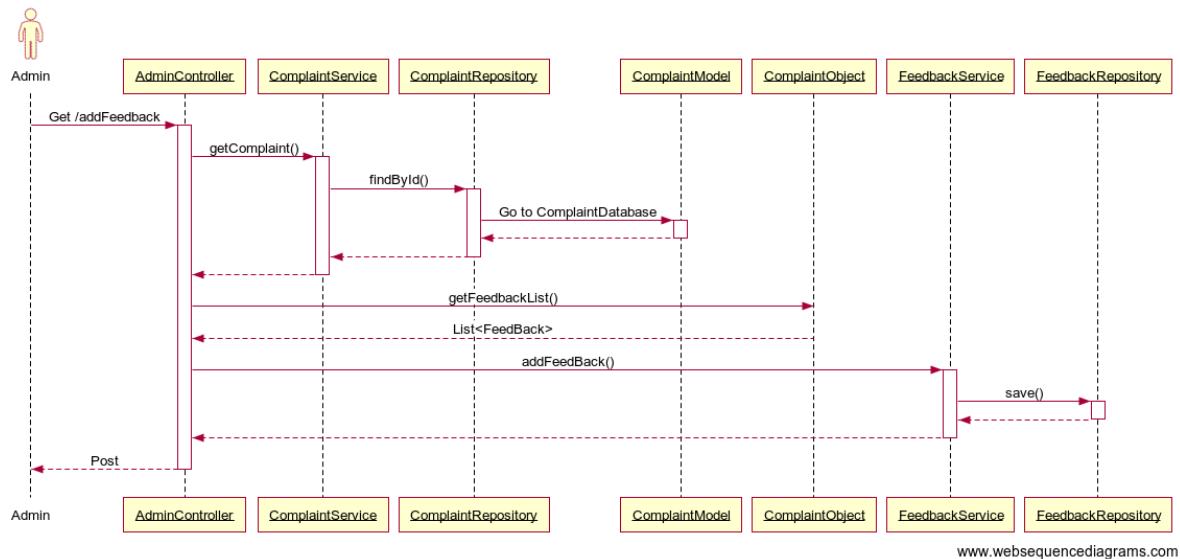
Admin Sequence





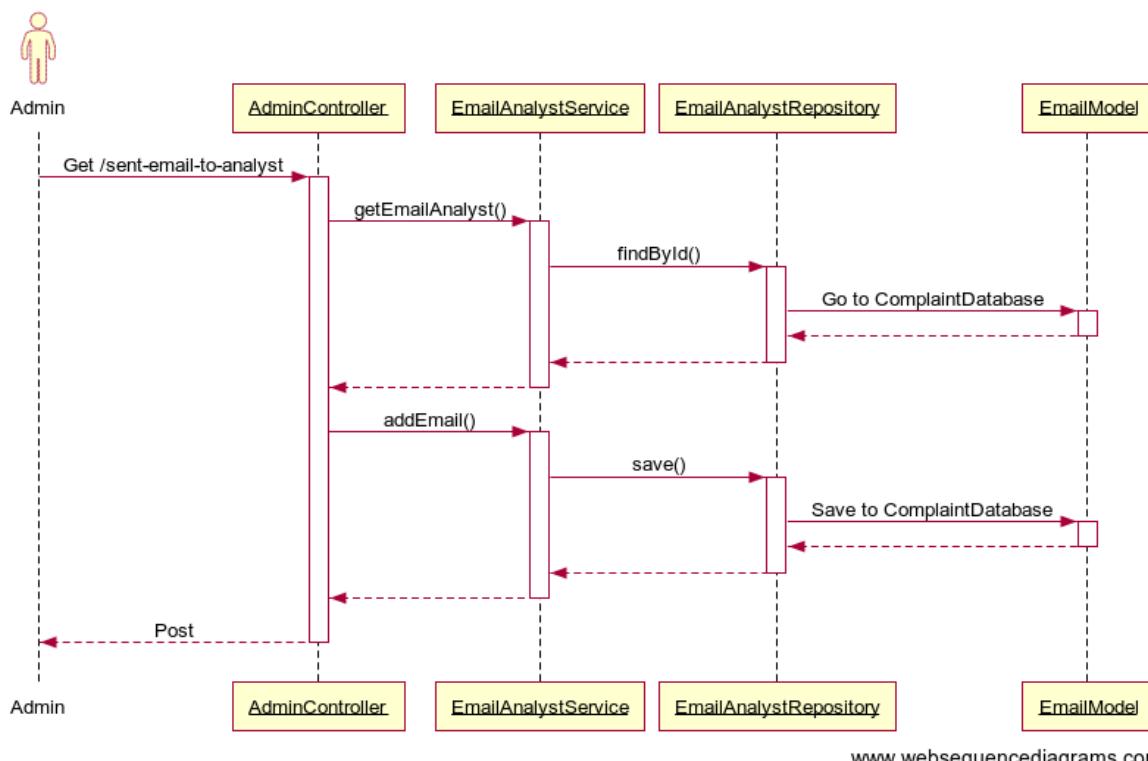
## 5.1.6 Admin Add Feedback

Admin Add Feedback



## 5.1.7 Admin Email to Analyst

Admin Sent Email to Analyst





## 5.2 System Object Model

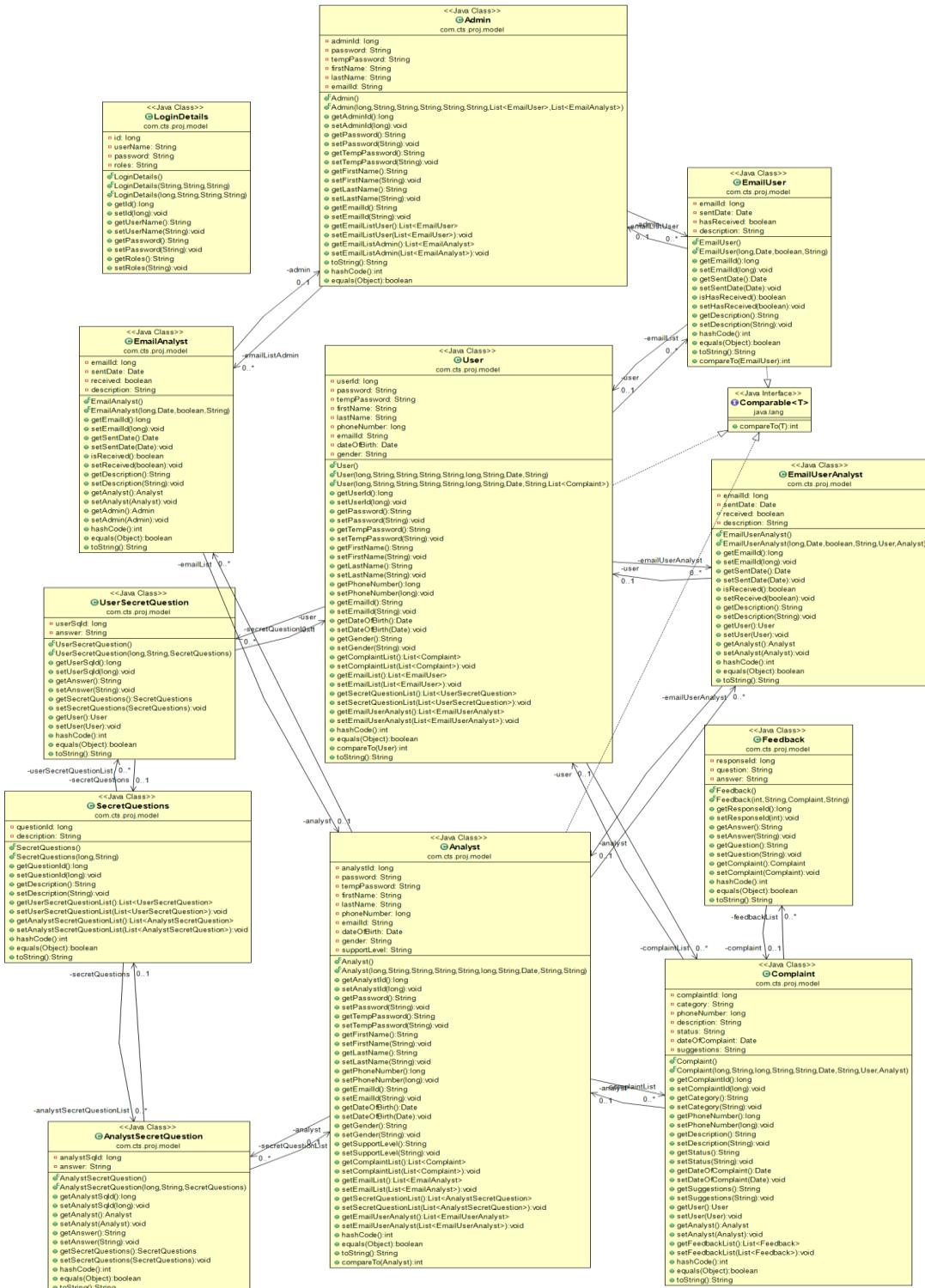
### 5.2.1 Model Classes

Model Classes required for storing the information as objects for easy writing and reading.

Overall 12 model classes are present.

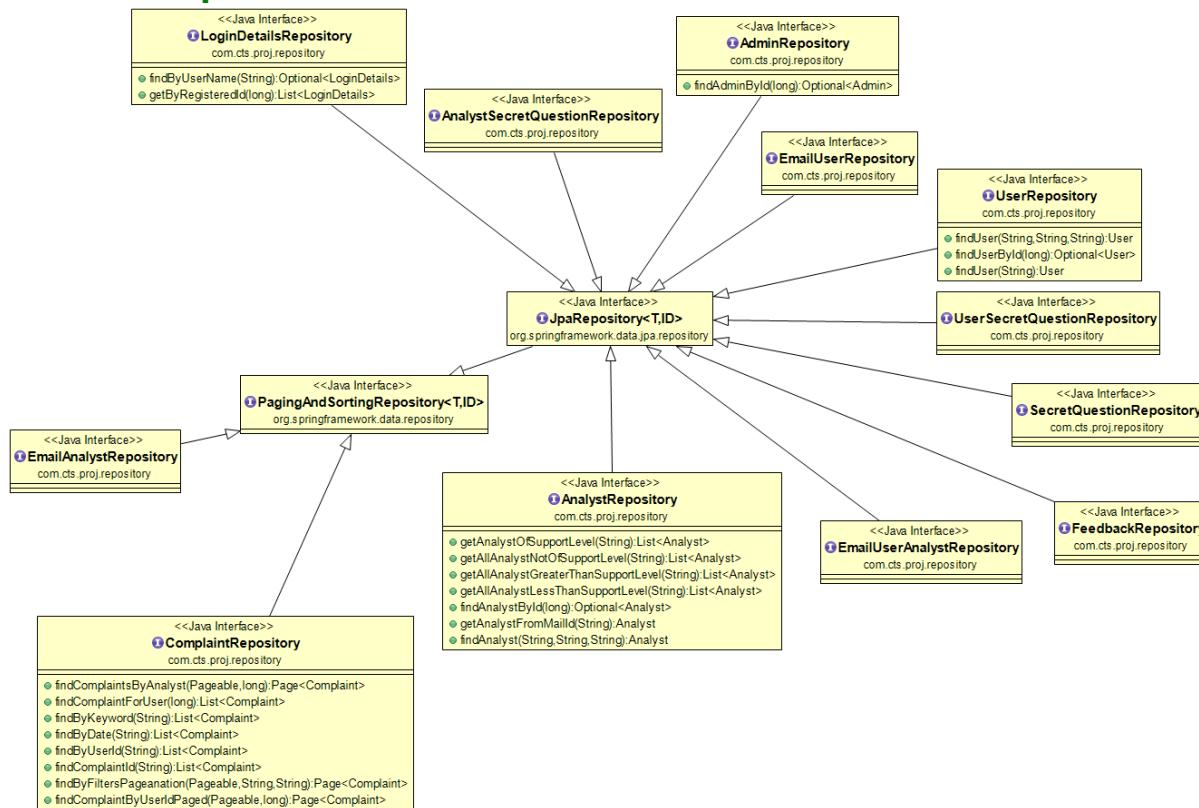
1. **User:** to store the information about the user
2. **Analyst:** to store info about the analyst
3. **Admin:** to store the information about the admin
4. **Complaint:** to store the info about the Complaint
5. **Feedback:** to store the feedback provided by user
6. **EmailAnalyst:** To store email object from 1 analyst or admin to another analyst
7. **EmailUser:** To store email object from admin to user
8. **EmailUserAnalyst:** To store email object from user to analyst
9. **UserSecretQuestions:** To Store secret question submitted by user
10. **AnalystSecretQuestion:** To Store secret question submitted by Analyst
11. **SecretQuestions:** To Store secret questions description
12. **LoginDetails:** This database holds all the login details info which is to be extracted by Spring Security for verification.



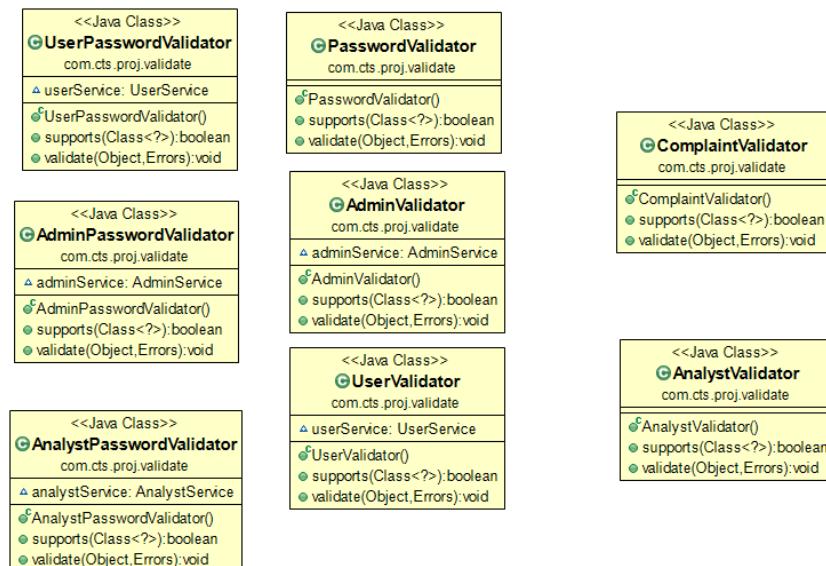




## 5.2.2 Repositories

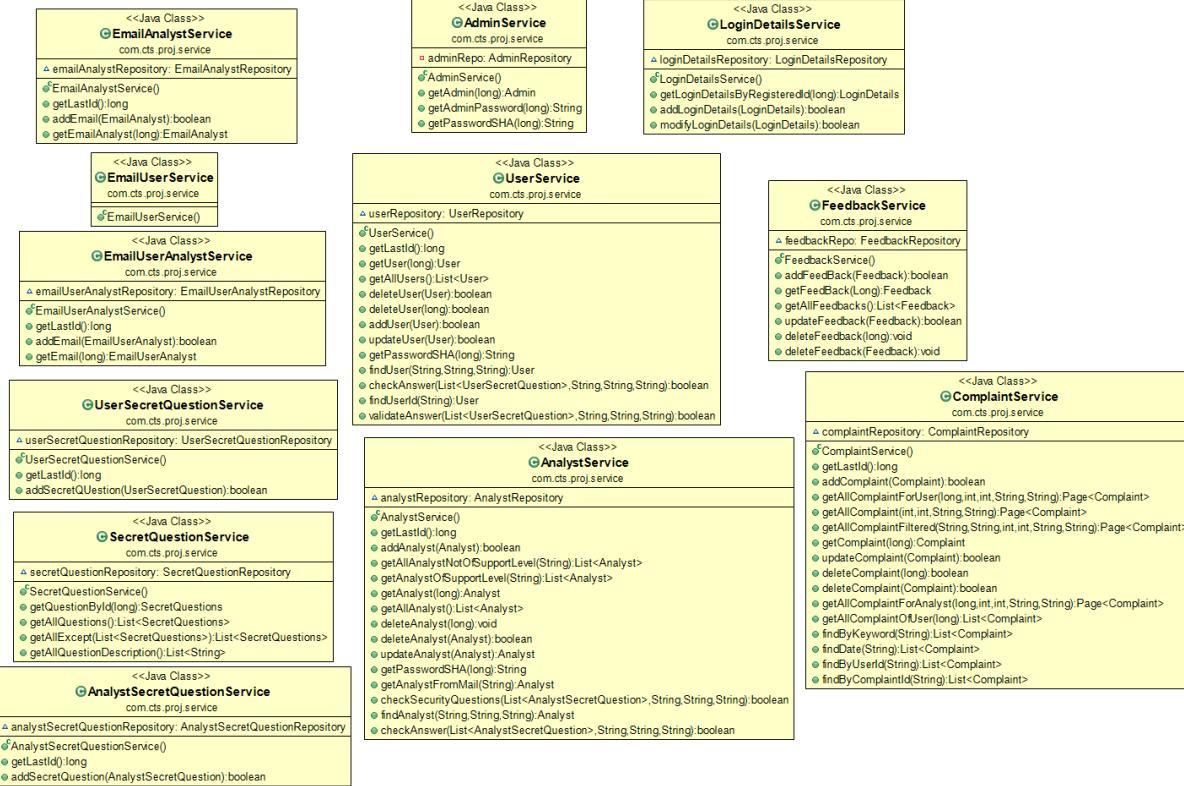


### 5.2.3 Validator Classes

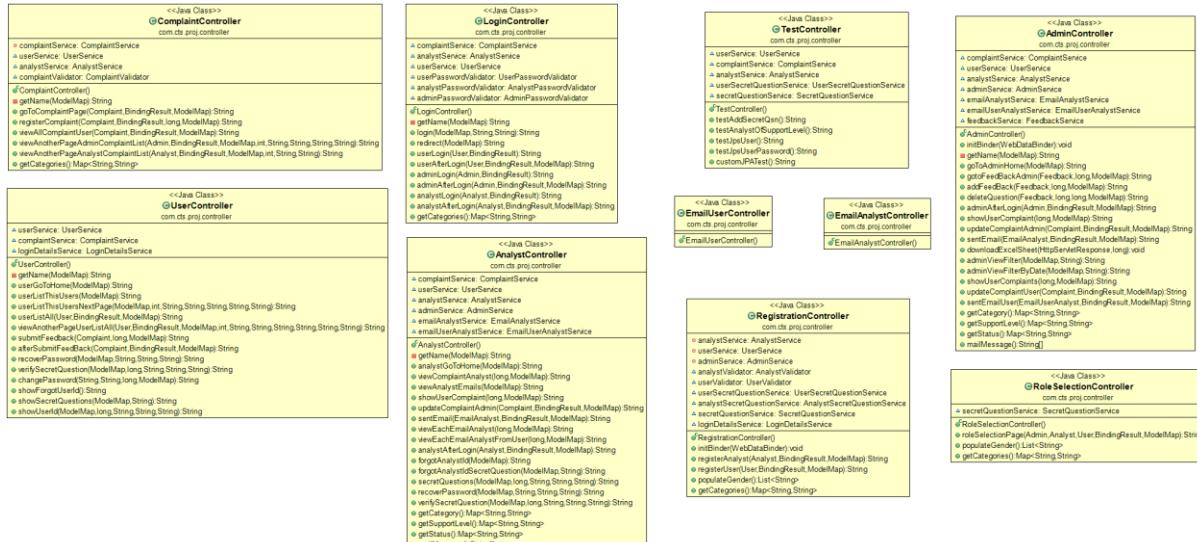




## 5.2.4 Service Classes

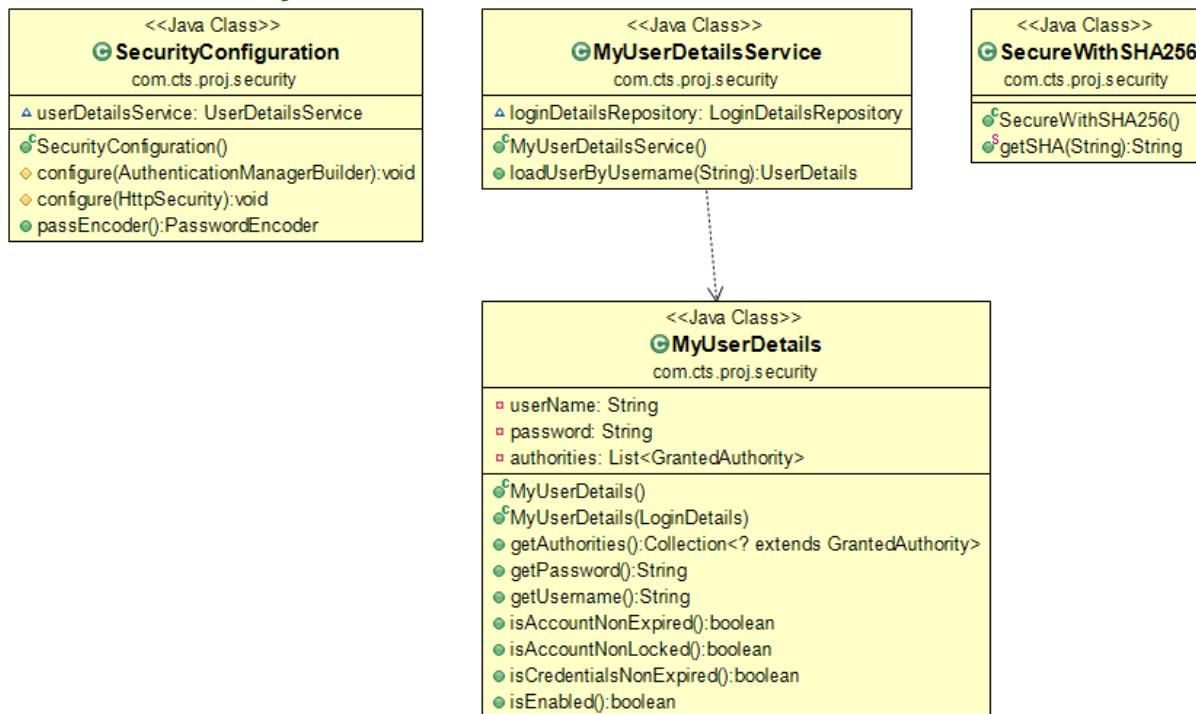


## 5.2.5 Controller Classes





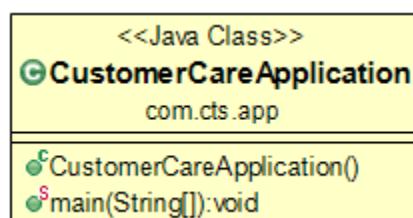
## 5.2.6 Security



## 5.2.7 Exporter Classes



## 5.2.8 Application Classes





## 5.3 UI Controls

### 5.3.1 User Registration

Data Element	Control type	Default Values	Editable Field	Restrictions	Mandatory/ Non Mandatory	Data Type	Data Size
First Name	Textbox	No	Yes	Only Alphabets	Mandatory	Varchar	45
Last Name	Textbox	No	Yes	Only Alphabets	Mandatory	Varchar	45
Phone Number	Textbox	No	Yes	Should be 10 digits long	Mandatory	Varchar	15
Email	Textbox	No	Yes	Should be a Proper email ID Format	Mandatory	Varchar	45
DOB	Textbox	No	Yes	Valid Date	Mandatory	Date	-
Gender	Drop Down		Yes	-	Mandatory	Varchar	10
Password	Textbox	No	Yes		Mandatory	Varchar	256
Confirm Password	Textbox	No			Mandatory	Varchar	256
Security Question 1	Textbox	No	Yes	-	Mandatory	Varchar	45
Security Question 2	Textbox	No	Yes	-	Mandatory	Varchar	45
Security Question 3	Textbox	No	Yes	-	Mandatory	Varchar	45





### 5.3.2 Analyst Registration

Data Element	Control type	Default Values	Editable Field	Restrictions	Mandatory/ Non Mandatory	Data Type	Data Size
First Name	Textbox	No	Yes	Only Alphabets	Mandatory	Varchar	45
Last Name	Textbox	No	Yes	Only Alphabets	Mandatory	Varchar	45
Phone Number	Textbox	No		Should be 10 digits long	Mandatory	Varchar	15
Email	Textbox	No		Should be a Proper email ID Format	Mandatory	Varchar	45
DOB	Textbox	No	Yes	Valid Date	Mandatory	Date	-
Gender	Drop Down		Yes	-	Mandatory	Varchar	10
Support Level	Drop Down	No	Yes	-	Mandatory	Varchar	2
Password	Textbox	No	Yes		Mandatory	Varchar	256
Confirm Password	Textbox	No	Yes		Mandatory	Varchar	256
Security Question 1	Textbox	No	Yes	-	Mandatory	Varchar	45
Security Question 2	Textbox	No	Yes	-	Mandatory	Varchar	45
Security Question 3	Textbox	No	Yes	-	Mandatory	Varchar	45





--	--	--	--	--	--	--	--

### 5.3.3 Login Admin

Data Element	Control type	Default Values	Editable Field	Restrictions	Mandatory/ Non Mandatory	Data Type	Data Size
Admin ID	Textbox	No	Yes		Mandatory	INT	
Password	Textbox	No	Yes		Mandatory	Varchar	256

### 5.3.4 Login User

Data Element	Control type	Default Values	Editable Field	Restrictions	Mandatory/ Non Mandatory	Data Type	Data Size
User ID	Textbox	No	Yes		Mandatory	INT	
Password	Textbox	No	Yes		Mandatory	Varchar	256

### 5.3.5 Login Analyst

Data Element	Control type	Default Values	Editable Field	Restrictions	Mandatory/ Non Mandatory	Data Type	Data Size
Analyst ID	Textbox	No	Yes		Mandatory	INT	
Password	Textbox	No	Yes		Mandatory	Varchar	256





### 5.3.6 Forgot User ID-User

Data Element	Control type	Default Values	Editable Field	Restrictions	Mandatory/ Non Mandatory	Data Type	Data Size
Email	Textbox	No	Yes	Should be a Proper email ID Format	Mandatory	Varchar	45
Security Question 1	Textbox	No	Yes	-	Mandatory	Varchar	45
Security Question 2	Textbox	No	Yes	-	Mandatory	Varchar	45
Security Question 3	Textbox	No	Yes	-	Mandatory	Varchar	45

### 5.3.7 Forgot Password-User

Data Element	Control type	Default Values	Editable Field	Restrictions	Mandatory/ Non Mandatory	Data Type	Data Size
User ID	Textbox	No	Yes		Mandatory	INT	
Phone Number	Textbox	No	Yes	Should be 10 digits long	Mandatory	Varchar	15
Email	Textbox	No	Yes	Should be a Proper email ID Format	Mandatory	Varchar	45
Security Question 1	Textbox	No	Yes	-	Mandatory	Varchar	45





Security Question 2	Textbox	No	Yes	-	Mandatory	Varchar	45
Security Question 3	Textbox	No	Yes	-	Mandatory	Varchar	45

### 5.3.8 Reset Password-User

Data Element	Control type	Default Values	Editable Field	Restrictions	Mandatory/ Non Mandatory	Data Type	Data Size
New Password	Textbox	No	Yes		Mandatory	Varchar	256
Confirm Password	Textbox	No	Yes		Mandatory	Varchar	256

### 5.3.9 Forgot Analyst ID-Analyst

Data Element	Control type	Default Values	Editable Field	Restrictions	Mandatory/ Non Mandatory	Data Type	Data Size
Email	Textbox	No	Yes	Should be a Proper email ID Format	Mandatory	Varchar	45
Security Question 1	Textbox	No	Yes	-	Mandatory	Varchar	45
Security Question 2	Textbox	No	Yes	-	Mandatory	Varchar	45
Security Question 3	Textbox	No	Yes	-	Mandatory	Varchar	45





### 5.3.10 Forgot Password-Analyst

Data Element	Control type	Default Values	Editable Field	Restrictions	Mandatory/ Non Mandatory	Data Type	Data Size
Analyst ID	Textbox	No	Yes		Mandatory	INT	
Phone Number	Textbox	No	Yes	Should be 10 digits long	Mandatory	Varchar	15
Email	Textbox	No	Yes	Should be a Proper email ID Format	Mandatory	Varchar	45
Security Question 1	Textbox	No	Yes	-	Mandatory	Varchar	45
Security Question 2	Textbox	No	Yes	-	Mandatory	Varchar	45
Security Question 3	Textbox	No	Yes	-	Mandatory	Varchar	45

### 5.3.11 Reset Password-Analyst

Data Element	Control type	Default Values	Editable Field	Restrictions	Mandatory/ Non Mandatory	Data Type	Data Size
New Password	Textbox	No	Yes		Mandatory	Varchar	256
Confirm Password	Textbox	No	Yes		Mandatory	Varchar	256





### 5.3.12 Create Complaint-User

Data Element	Control type	Default Values	Editable Field	Restrictions	Mandatory/ Non Mandatory	Data Type	Data Size
Complaint ID	Textbox	Yes	No	Only Digits	Mandatory	INT	
User ID	Textbox	Yes	No	Only Digits	Mandatory	INT	
Phone Number	Textbox	Yes	Yes	Should be 10 digits long	Mandatory	Varchar	15
Category	Drop Down	No	Yes	-	Mandatory	Varchar	45
Complaint	Textbox	No	Yes	Minimum 10 words	Mandatory	Varchar	1024

### 5.3.13 Complaint View-User

Data Element	Control type	Default Values	Editable Field	Restrictions	Mandatory/ Non Mandatory	Data Type	Data Size
Complaint ID	Textbox	Yes	No	Only Digits	Mandatory	INT	
User ID	Textbox	Yes	No	Only Digits	Mandatory	INT	
Phone Number	Textbox	Yes	No	Should be 10 digits long	Mandatory	Varchar	15
Category	Drop Down	Yes	No	-	Mandatory	Varchar	45





Complaint	Textbox	Yes	No	Minimum 10 words	Mandatory	Varchar	1024
-----------	---------	-----	----	------------------	-----------	---------	------

### 5.3.14 Complaint View-Analyst

Data Element	Control type	Default Values	Editable Field	Restrictions	Mandatory/ Non Mandatory	Data Type	Data Size
Complaint ID	Textbox	Yes	No	Only Digits	Mandatory	INT	
User ID	Textbox	Yes	No	Only Digits	Mandatory	INT	
Phone Number	Textbox	Yes	No	Should be 10 digits long	Mandatory	Varchar	15
Email	Textbox	No	Yes	Should be a Proper email ID Format	Mandatory	Varchar	45
Category	Drop Down	Yes	No	-	Mandatory	Varchar	45
Status	Drop Down	Yes	Yes	-	Mandatory	Varchar	45
Assigned Analyst	Drop Down	Yes	Yes	-	Mandatory	Varchar	45
Complaint	Text Area	Yes	No	Minimum 10 words	Mandatory	Varchar	1024





### 5.3.15 Complaint View-Admin

Data Element	Control type	Default Values	Editable Field	Restrictions	Mandatory/ Non Mandatory	Data Type	Data Size
Complaint ID	Textbox	Yes	No	Only Digits	Mandatory	INT	
User ID	Textbox	Yes	No	Only Digits	Mandatory	INT	
Phone Number	Textbox	Yes	No	Should be 10 digits long	Mandatory	Varchar	15
Email	Textbox	No	Yes	Should be a Proper email ID Format	Mandatory	Varchar	45
Category	Drop Down	Yes	No	-	Mandatory	Varchar	45
Status	Drop Down	Yes	Yes	-	Mandatory	Varchar	45
Assigned Analyst	Drop Down	Yes	Yes	-	Mandatory	Varchar	45
Complaint	Textbox	Yes	No	Minimum 10 words	Mandatory	Varchar	1024

### 5.3.16 Add Feedback-Admin

Data Element	Control type	Default Values	Editable Field	Restrictions	Mandatory/ Non Mandatory	Data Type	Data Size
Question Description	Textbox	No	Yes	-	Mandatory	Varchar	1024





### 5.3.17 Email View-Analyst

Data Element	Control type	Default Values	Editable Field	Restrictions	Mandatory/ Non Mandatory	Data Type	Data Size
Sent By	Textbox	Yes	No	-	Mandatory	Varchar	45
Recipient	Textbox	Yes	No	-	Mandatory	Varchar	45
Date	Textbox	Yes	No	-	Mandatory	Date	-
Description	Text Area	Yes	No	-	Mandatory	Varchar	1024

### 5.3.18 Email-Admin

Data Element	Control type	Default Values	Editable Field	Restrictions	Mandatory/ Non Mandatory	Data Type	Data Size
Email ID	Textbox	Yes	No	-	Mandatory	INT	
Email To	Textbox	Yes	Yes	-	Mandatory	Varchar	45
Email To ID	Textbox	Yes	Yes	-	Mandatory	INT	
Date	Textbox	Yes	Yes	-	Mandatory	Date	
Complaint	Textbox	Yes	Yes	-	Mandatory	Varchar	1024
Mail Description	Text Area	No	Yes	-	Mandatory	Varchar	1024







## 5.4 UI Design

### 5.4.1 Admin Login

Customer Care

User Analyst Admin

Login

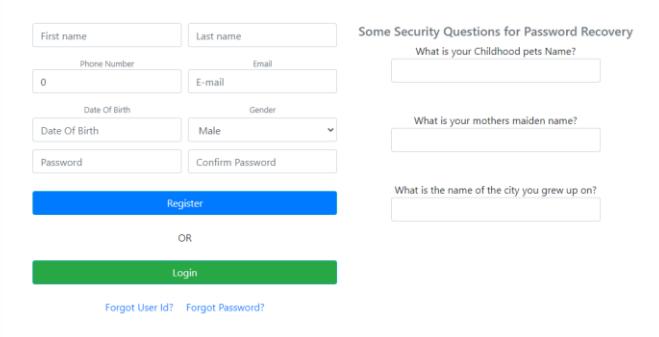


© 2020 Copyright:

### 5.4.2 Register User/Analyst

Customer Care

User Analyst Admin



First name  Last name   
Phone Number  E-mail   
Date Of Birth  Gender   
Date Of Birth  Male   
Password  Confirm Password   
**Register** **OR** **Login**

Some Security Questions for Password Recovery  
What is your Childhood pets Name?   
What is your mothers maiden name?   
What is the name of the city you grew up on?

[Forgot User Id?](#) [Forgot Password?](#)

© 2020 Copyright:





### 5.4.3 Register Analyst

Customer Care

User **Analyst** Admin

First name  Last name   
Phone Number  Email   
0  E-mail   
Date Of Birth  Gender   
Date Of Birth  Male   
Support Level   
Level 1   
Password  Confirm Password   
**Register**  
OR  
**Login**  
Forgot User Id? | Forgot Password?

Some Security Questions for Password Recovery

What is your Childhood pets Name?   
What is your mothers maiden name?   
What is the name of the city you grew up on?

© 2020 Copyright:  
localhost:9066/roleSelection?tag#pills-analyst

### 5.4.4 Create Complaint User

Customer Care

Home  
Logout

**Complaint Submission**

Complaint ID:  User Id:  Phone Number:  Category:   
4007  3001  9876543210  Software

Write Your Complaint

Write your Complaint here...

**Submit**

© 2020 Copyright:





### 5.4.5 View Complaint User

The screenshot shows a web application interface titled "Complaint Notifications". At the top, there are navigation links for "Home" and "Menu". Below the title, there are search filters: "Sort by" (set to "Complaint Id"), "Date", and "Category". It displays "Total Complaints : 3" and "Page 1 of 1". There are four search buttons: "Search by Category", "YYYY-MM-DD" (set to "3001"), "Enter complaintId", and three other "Search" buttons. Two complaints are listed:

- Complaint Id : 4001**  
Description: Put description Here  
View button
- Complaint Id : 4002**  
Description: Put description Here

### 5.4.6 All Complaints

The screenshot shows a web application interface titled "Complaint Notifications". At the top, there is a "Menu" link. Below the title, there are search filters: "Sort by" (set to "Complaint Id"), "Date", and "Category". It displays "Total Complaints : 3" and "Page 1 of 1". There are four search buttons: "Search by Category", "YYYY-MM-DD" (set to "3001"), "Search by User Id", and "Enter complaintId". Two complaints are listed:

- Complaint Id : 4001**  
Description: Put description Here  
View button
- Complaint Id : 4002**  
Description: Please Fill The Description

© 2020 Copyright:





## 5.4.7 User to Analyst Mail

Customer Care

Email Number : 12007

Email To :

Email To Id :

Date :

Complaint

Put description Here

Dear, Unasigned

Please do the needfull to rectify the complaint the user Linjo is facing  
Hope this issue is rectified soon,  
Regards,  
Linjo

Description

Sent Email

© 2020 Copyright:

## 5.4.8 5.4.6 Analyst/Admin Home

Customer Care

Welcome

View All Complaints

Any Complaints which you may have previously submitted can be viewed here.

View All Complaints

© 2020 Copyright:





## 5.4.9 Admin Home

Welcome

Create FeedBack Survey  
Create A new Feedback Survey for a User

Create FeedBack

© 2020 Copyright:

## 5.4.10 View All Complaints Analyst/Admin

Complaint Notifications

Sort by : Complaint Id Analyst Id Date Category

Total Complaints : 7 Page 1 of 2

1 2

Complaint Id	Description	Action
4001	Put description Here	<a href="#">View</a>
4002	Please Fill The Description	<a href="#">View</a>
4003	DSAV f ceger vgew v sg c gd dsfc w fcvs eg ve gv w vs e svh sbv b sh caedatc e gt srt cv w srvv e th sr dhv wr	<a href="#">View</a>

© 2020 Copyright:





## 5.4.11 View Complaint Analyst/Admin

Customer Care Menu ▾

### Complaint View

Complaint Id :	4001
User Id :	3001
Phone Number :	1212121212
Email Id :	linjo@cts.com
Category :	Software
Status :	reopen
Assigned Analyst	L1 : 2006

Description  
Put description Here

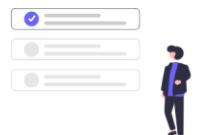
Assign Download Excel File

© 2020 Copyright:

Customer Care Home Menu ▾

### Complaint Notifications

Sort by :	Complaint Id	Date	Category
Total Complaints : 7			
1 2			
Search by Category YYYY-MM-DD			
<span style="border: 1px solid #ccc; padding: 2px 5px;">Search</span> <span style="border: 1px solid #ccc; padding: 2px 5px;">Search</span>			
<p><b>Complaint Id : 4001</b></p> <p>Description      Complaint Created By : 3001 Assigned Analyst Id 2001 <span style="border: 1px solid #ccc; padding: 2px 5px;">reopen</span></p> <p>Put description Here</p> <p style="text-align: center;"><span style="border: 1px solid #ccc; padding: 2px 5px;">View</span> <span style="border: 1px solid #ccc; padding: 2px 5px;">Create Feedback</span></p>			
<p><b>Complaint Id : 4002</b></p> <p>Description      Complaint Created By : 3001 Assigned Analyst Id 2001 <span style="border: 1px solid #ccc; padding: 2px 5px;">Active</span></p> <p>Please Fill The Description</p> <p style="text-align: center;"><span style="border: 1px solid #ccc; padding: 2px 5px;">View</span> <span style="border: 1px solid #ccc; padding: 2px 5px;">Create Feedback</span></p>			
<p><b>Complaint Id : 4003</b></p> <p>Description      Complaint Created By : 3001 Assigned Analyst Id 2001 <span style="border: 1px solid #ccc; padding: 2px 5px;">reopen</span></p> <p>DSAV f ceger vgew v sg c gd dsfc w fcvs eg ve gv w vs e svh sbv b sh caedatc e gt srt cv w srvv e th sr dhv wr</p>			


© 2020 Copyright:



### 5.4.12 Create Feedback Admin

Customer Care

Menu ▾

#### Complaint Feedback Questionnaire

[Finish Survey](#)

#	Question Description	Remove
1	How would you rate the CustomerCare	<a href="#">Delete</a>
2	Would You suggest this to your friends	<a href="#">Delete</a>
3	Will you come to us in the future	<a href="#">Delete</a>

Question Description

Write your Question here...

Add New

### 5.4.13 Analyst Emails

Customer Care

Menu ▾

Email Number	Date	Description	
10001	2020-12-16 21:52:34.0	Dear, Unsigned Please do the needfull to rectify the complaint the user Linj.....	<a href="#">View</a>
10002	2020-12-16 22:11:17.0	Dear, Unsigned Please do the needfull to rectify the complaint the user LINJ.....	<a href="#">View</a>
12005	2020-12-16 12:17:53.0	Dear, Unsigned Please do the needfull to rectify the complaint the user Linjo .....	<a href="#">View</a>
12006	2020-12-16 12:19:51.0	Dear, Unsigned Please do the needfull to rectify the complaint the user Linj.....	<a href="#">View</a>
12007	2020-12-16 17:54:41.0	Dear, Unsigned Please do the needfull to rectify the complaint the user Linj.....	<a href="#">View</a>
12008	2020-12-17 08:58:57.0	Dear, Unsigned Please do the needfull to rectify the complaint the user Linj.....	<a href="#">View</a>

© 2020 Copyright:



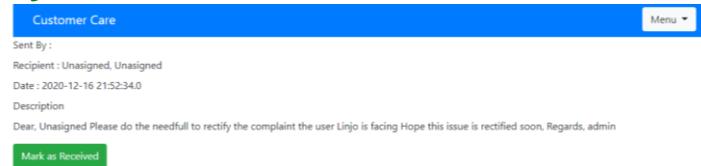
### 5.4.14 Analyst Email View

Customer Care

Menu ▾

Sent By :  
Recipient : Unsigned, Unsigned  
Date : 2020-12-16 21:52:34.0  
Description  
Dear, Unsigned Please do the needfull to rectify the complaint the user Linjo is facing Hope this issue is rectified soon, Regards, admin

[Mark as Received](#)





## 5.5 Component Description

### 5.5.1 Controller Classes

Class	Method	Program Logic
AdminController	<b>initBinder</b>  <b>Parameters</b> WebDataBinder binder  <b>Return Type</b> void	<ul style="list-style-type: none"><li>Convert date into “ yyyy-MM-dd” format.</li></ul>
	<b>getName</b>  <b>Parameters</b> ModelMap model  <b>Return Type</b> String	<ul style="list-style-type: none"><li>Return the name of user logged in.</li></ul>
	<b>goToAdminHome</b>  <b>Parameters</b> ModelMap model  <b>Return Type</b> String	<ul style="list-style-type: none"><li>Returns the admin-home.jsp page.</li></ul>
	<b>gotoFeedBackAdmin</b>  <b>Parameters</b> @ModelAttribute Feedback feedback, @RequestParam long complaintId, ModelMap model  <b>Return Type</b> String	<ul style="list-style-type: none"><li>Returns the feedback-creation-admin.jsp page.</li></ul>





	<p><b>addFeedBack</b></p> <p><b>Parameters</b></p> <pre>@ModelAttribute Feedback feedback, @RequestParam long complaintId, ModelMap model</pre> <p><b>Return Type</b></p> <p>String</p>	<ul style="list-style-type: none"><li>Accept Feedback object as parameter and add to database. Then redirect to feedback-creation-admin.jsp page.</li></ul>
	<p><b>deleteQuestion</b></p> <p><b>Parameters</b></p> <pre>@ModelAttribute Feedback feedback, @RequestParam long responseId, @RequestParam long complaintId, ModelMap model</pre> <p><b>Return Type</b></p> <p>String</p>	<ul style="list-style-type: none"><li>Accept response Id as parameter and delete the corresponding feedback. Then redirect to feedback-creation-admin.jsp page.</li></ul>
	<p><b>adminAfterLogin</b></p> <p><b>Parameters</b></p> <pre>@Validated @ModelAttribute Admin admin, BindingResult result, ModelMap model.</pre> <p><b>Return Type</b></p> <p>String</p>	<ul style="list-style-type: none"><li>Return complaint-notification-admin.jsp page</li></ul>
	<p><b>showUserComplaint</b></p> <p><b>Parameters</b></p> <pre>@RequestParam long complaintId, ModelMap model</pre> <p><b>Return Type</b></p> <p>String</p>	<ul style="list-style-type: none"><li>Return complaint-admin-view.jsp page.</li></ul>
	<p><b>updateComplaintAdmin</b></p> <p><b>Parameters</b></p> <pre>@Validated @ModelAttribute Complaint complaint, BindingResult result, ModelMap model)</pre>	<ul style="list-style-type: none"><li>Accept Complaint object as parameter and assign support analyst.</li><li>Then add mail content and return admin-to-analyst-mail.jsp page.</li></ul>





	<p><b>Return Type</b> String</p> <p><b>sentEmail</b></p> <p><b>Parameters</b></p> <pre>@ModelAttribute EmailAnalyst emailAnalyst, BindingResult results, ModelMap model</pre> <p><b>Return Type</b> String</p>	<ul style="list-style-type: none"><li>• Sent mail to analyst and return admin-home.jsp page.</li></ul>
	<p><b>downloadExcelSheet</b></p> <p><b>Parameters</b></p> <pre>HttpServletResponse response, @RequestParam long complaintId</pre> <p><b>Return Type</b> void</p>	<ul style="list-style-type: none"><li>• Accept complaint Id and provide option to download excel file.</li></ul>
	<p><b>adminViewFilter</b></p> <p><b>Parameters</b></p> <pre>ModelMap model, String keyword</pre> <p><b>Return Type</b> String</p>	<ul style="list-style-type: none"><li>• Filter based on keyword and return complaint-notification-admin.jsp page.</li></ul>
	<p><b>adminViewFilterByDate</b></p> <p><b>Parameters</b></p> <pre>ModelMap model, String date</pre> <p><b>Return Type</b> String</p>	<ul style="list-style-type: none"><li>• Filter based on date and return complaint-notification-admin.jsp page.</li></ul>
	<p><b>showUserComplaints</b></p> <p><b>Parameters</b></p> <pre>@RequestParam long complaintId, ModelMap model</pre>	<ul style="list-style-type: none"><li>• Return complaint-view-user.jsp page.</li></ul>





	<p><b>Return Type</b> String</p> <p><b>updateComplaintUser</b></p> <p><b>Parameters</b></p> <pre>@Validated @ModelAttribute Complaint complaint, BindingResult result, ModelMap model</pre> <p><b>Return Type</b> String</p> <p><b>sentEmailUser</b></p> <p><b>Parameters</b></p> <pre>@ModelAttribute EmailUserAnalyst email, BindingResult results, ModelMap model</pre> <p><b>Return Type</b> String</p> <p><b>getCategory()</b></p> <p><b>Parameters</b></p> <p>None</p> <p><b>Return Type</b> Map&lt;String, String&gt;</p> <p><b>getSupportLevel</b></p> <p><b>Parameters</b></p> <p>None</p> <p><b>Return Type</b> Map&lt;String, String&gt;</p> <p><b>getStatus()</b></p> <p><b>Parameters</b></p> <p><b>Return Type</b> Map&lt;String, String&gt;</p>	<ul style="list-style-type: none"><li>Set mail content and return user-to-analyst-mail.jsp page.</li></ul> <ul style="list-style-type: none"><li>Sent mail to analyst and return user-home.jsp page.</li></ul> <ul style="list-style-type: none"><li>Return category list.</li></ul> <ul style="list-style-type: none"><li>Return supportLevel list.</li></ul> <ul style="list-style-type: none"><li>Return sstatus list.</li></ul>
--	--	---





	<b>mailMessage()</b>  <b>Parameters</b> <b>None</b>  <b>Return Type</b> String[]	<ul style="list-style-type: none"><li>• Return mail body,</li></ul>
AnalystController	<b>getName</b>  <b>Parameters</b> ModelMap model  <b>Return Type</b> String	<ul style="list-style-type: none"><li>• Return the name of user logged in.</li></ul>
	<b>analystGoToHome</b>  <b>Parameters</b> ModelMap model  <b>Return Type</b> String	<ul style="list-style-type: none"><li>• Return analyst-home.jsp page.</li></ul>
	<b>viewComplaintAnalyst</b>  <b>Parameters</b> @RequestParam long complaintId, ModelMap model  <b>Return Type</b> String	<ul style="list-style-type: none"><li>• Return complaint-analyst-view.jsp page.</li></ul>
	<b>viewAnalystEmails</b>  <b>Parameters</b> ModelMap model  <b>Return Type</b> String	<ul style="list-style-type: none"><li>• Return emails-analyst.jsp page.</li></ul>





	<p><b>showUserComplaint</b></p> <p><b>Parameters</b></p> <pre>@RequestParam long complaintId, ModelMap model</pre> <p><b>Return Type</b></p> <p>String</p>	<ul style="list-style-type: none"><li>Accept complaint Id as parameter and add complaint details and return complaint-analyst-view jsp page.</li></ul>
	<p><b>updateComplaintAdmin</b></p> <p><b>Parameters</b></p> <pre>@Validated @ModelAttribute Complaint complaint, BindingResult result, ModelMap model</pre> <p><b>Return Type</b></p> <p>String</p>	<ul style="list-style-type: none"><li>Set mail content and return analyst -to-analyst-mail jsp page.</li></ul>
	<p><b>sentEmail</b></p> <p><b>Parameters</b></p> <pre>@ModelAttribute EmailAnalyst emailAnalyst, BindingResult results, ModelMap model</pre> <p><b>Return Type</b></p> <p>String</p>	<ul style="list-style-type: none"><li>Sent mail to analyst and return analyst -home jsp page.</li></ul>
	<p><b>viewEachEmailAnalyst</b></p> <p><b>Parameters</b></p> <pre>@RequestParam long emailId, ModelMap model</pre> <p><b>Return Type</b></p> <p>String</p>	<ul style="list-style-type: none"><li>Return each-email-analyst jsp page.</li></ul>
	<p><b>viewEachEmailAnalystFromUser</b></p> <p><b>Parameters</b></p> <pre>@RequestParam long emailId, ModelMap model</pre> <p><b>Return Type</b></p>	<ul style="list-style-type: none"><li>Return each-email-analyst-from-user jsp page.</li></ul>





	<p>String</p> <p><b>analystAfterLogin</b></p> <p><b>Parameters</b></p> <pre>@Validated @ModelAttribute Analyst analyst, BindingResult result, ModelMap model</pre> <p><b>Return Type</b></p> <p>String</p>	<ul style="list-style-type: none"><li>• Return complaint-notification-analyst jsp page.</li></ul>
	<p><b>forgotAnalystId</b></p> <p><b>Parameters</b></p> <pre>ModelMap model</pre> <p><b>Return Type</b></p> <p>String</p>	<ul style="list-style-type: none"><li>• Return forgot-analyst-id jsp page.</li></ul>
	<p><b>forgotAnalystIdSecretQuestion</b></p> <p><b>Parameters</b></p> <pre>ModelMap model, String mail</pre> <p><b>Return Type</b></p> <p>String</p>	<ul style="list-style-type: none"><li>• Accept Email Id as parameter and find corresponding analyst.</li><li>• If a valid analyst is obtained, then return forgot-analyst-id-sq-question jsp page else return forgot-analyst-id jsp page.</li></ul>
	<p><b>secretQuestions</b></p> <p><b>Parameters</b></p> <pre>ModelMap model, @PathVariable long analystId, String ans1, String ans2, String ans3</pre> <p><b>Return Type</b></p> <p>String</p>	<ul style="list-style-type: none"><li>• Accept answers of security questions as parameter and find corresponding analyst.</li><li>• If a valid analyst is obtained, then return display-analyst-id jsp page else return forgot-analyst-id jsp page.</li></ul>
	<p><b>recoverPassword</b></p> <p><b>Parameters</b></p> <pre>ModelMap model, String analystId, String mob, String email</pre> <p><b>Return Type</b></p>	<ul style="list-style-type: none"><li>• Accept user Id, email Id and mobile number as parameter and find corresponding analyst.</li><li>• If a valid analyst is obtained, then password-recovery-analyst jsp page else return forgot-password-analyst jsp page.</li></ul>





	String	
	<b>verifySecretQuestion</b>  <b>Parameters</b>  ModelMap model, @PathVariable long analystId, String ans1, String ans2, String ans3  <b>Return Type</b>  String	<ul style="list-style-type: none"><li>• Accept answers of security questions as parameter and verify analyst.</li><li>• If a valid analyst is obtained, then return set-new-pwd-analyst.jsp page else return password-recovery-analyst.jsp page.</li></ul>
	<b>changePassword</b>  <b>Parameters</b>  String newPwd, String confirmPwd, long analystId, ModelMap model  <b>Return Type</b>  String	<ul style="list-style-type: none"><li>• Accept password as parameter.</li><li>• If password is valid, then set as new password and return analyst-home.jsp page. Else return set-new-pwd-analyst.jsp page.</li></ul>
	<b>getCategory</b>  <b>Parameters</b>  None  <b>Return Type</b>  Map<String, String>	<ul style="list-style-type: none"><li>• Return category list.</li></ul>
	<b>getSupportLevel</b>  <b>Parameters</b>  None  <b>Return Type</b>  Map<String, String>	<ul style="list-style-type: none"><li>• Return support level list.</li></ul>
	<b>getStatus</b>  <b>Parameters</b>  None  <b>Return Type</b>  Map<String, String>	<ul style="list-style-type: none"><li>• Return sstatus list.</li></ul>
	<b>mailMessage</b>  <b>Parameters</b>	<ul style="list-style-type: none"><li>• Return mail body.</li></ul>





	<p>None</p> <p><b>Return Type</b></p> <p>String[]</p>	
UserController	<p><b>getName</b></p> <p><b>Parameters</b></p> <p>ModelMap model</p> <p><b>Return Type</b></p> <p>String</p>	<ul style="list-style-type: none"><li>Returns the name of user logged in.</li></ul>
	<p><b>userGoToHome</b></p> <p><b>Parameters</b></p> <p>ModelMap model</p> <p><b>Return Type</b></p> <p>String</p>	<ul style="list-style-type: none"><li>Return user-home.jsp page.</li></ul>
	<p><b>userListThisUsers</b></p> <p><b>Parameters</b></p> <p>ModelMap model</p> <p><b>Return Type</b></p> <p>String</p>	<ul style="list-style-type: none"><li>Accept the parameters and add complaint details of user to page and return complaint-notification-user-personnel.jsp page.</li></ul>
	<p><b>userListThisUsersNextPage</b></p> <p><b>Parameters</b></p> <p>ModelMap model, @PathVariable int pageNumber, @Param String sortBy, @Param String sortDir, String keyword, String date, String complaintId</p> <p><b>Return Type</b></p> <p>String</p>	<ul style="list-style-type: none"><li>Accept the parameters and add complaint details of user to page and return complaint-notification-user-personnel.jsp page.</li></ul>
	<p><b>userListAll</b></p> <p><b>Parameters</b></p>	<ul style="list-style-type: none"><li>Add complaint details of all users to page and return complaint-notification-user.jsp page.</li></ul>





	<pre>@Validated @ModelAttribute User user, BindingResult result, ModelMap model</pre> <p><b>Return Type</b> String</p>	
	<pre>viewAnotherPageUserListAll</pre> <p><b>Parameters</b> <pre>@Validated @ModelAttribute User user, BindingResult result, ModelMap model, @PathVariable int pageNumber, @Param String sortBy, @Param String sortDir, String keyword, String date, String userId, String complaintId</pre><p><b>Return Type</b> String</p></p>	<ul style="list-style-type: none"><li>Filter the complaints based on category, date, user Id and complaint Id and return complaint-notification-user jsp page.</li></ul>
	<pre>submitFeedback</pre> <p><b>Parameters</b> <pre>@ModelAttribute Complaint complaint, @RequestParam long complaintId, ModelMap model</pre><p><b>Return Type</b> String</p></p>	<ul style="list-style-type: none"><li>Accept complaint Id as parameter and get complaint and feedback details and return user-feedback-submission jsp page.</li></ul>
	<pre>afterSubmitFeedBack</pre> <p><b>Parameters</b> <pre>@ModelAttribute Complaint complaint, BindingResult result, ModelMap model</pre><p><b>Return Type</b> String</p></p>	<ul style="list-style-type: none"><li>Set answer for feedback questions and add it to the complaint.</li><li>Then return to user-home jsp page.</li></ul>
	<pre>recoverPassword</pre> <p><b>Parameters</b> <pre>ModelMap model, String userId, String mob, String email</pre><p><b>Return Type</b></p></p>	<ul style="list-style-type: none"><li>Accept user Id, email Id and mobile number as parameter and find corresponding user.</li><li>If a valid user is obtained, then password-recovery-user jsp page else return forgot-password-user jsp page.</li></ul>





	String	
	<b>verifySecretQuestion</b>  <b>Parameters</b>  ModelMap model,@PathVariable long userId,String ans1,String ans2,String ans3  <b>Return Type</b>  String	<ul style="list-style-type: none"><li>• Accept answers of secret questions as parameter and verify user.</li><li>• If a valid user is obtained, then return set-new-pwd-user.jsp page else return password-recovery-user.jsp page.</li></ul>
	<b>changePassword</b>  <b>Parameters</b>  String newPassword, String confirmPassword, long userId, ModelMap model  <b>Return Type</b>  String	<ul style="list-style-type: none"><li>• Accept password as parameter.</li><li>• If password is valid, then set as new password and return user-home.jsp page. Else return set-new-pwd-user.jsp page.</li></ul>
	<b>showForgotUserId</b>  <b>Parameters</b>  <b>Return Type</b>  String	<ul style="list-style-type: none"><li>• Return forgot-user-id.jsp page.</li></ul>
	<b>showSecretQuestions</b>  <b>Parameters</b>  ModelMap model, String email  <b>Return Type</b>  String	<ul style="list-style-type: none"><li>• Accept email Id as parameter and find corresponding user.</li><li>• If a valid user is obtained, then forgot-user-id-sq-question.jsp page else return forgot-user-id.jsp page.</li></ul>
	<b>showUserId</b>  <b>Parameters</b>  ModelMap model,@PathVariable long userId, String ans1, String ans2, String ans3  <b>Return Type</b>  String	<ul style="list-style-type: none"><li>• Accept answers of secret questions as parameter and verify user.</li><li>• If a valid user is obtained, then return show-user-id.jsp page else return forgot-user-id-sq-question.jsp page.</li></ul>





ComplaintController	<p><b>getName</b></p> <p><b>Parameters</b></p> <p>ModelMap model</p> <p><b>Return Type</b></p> <p>String</p> <p><b>goToComplaintPage</b></p> <p><b>Parameters</b></p> <p>@ModelAttribute Complaint complaint, BindingResult result, ModelMap model</p> <p><b>Return Type</b></p> <p>String</p> <p><b>registerComplaint</b></p> <p><b>Parameters</b></p> <p>@Validated @ModelAttribute Complaint complaint, BindingResult result, @RequestParam long complaintId, ModelMap model</p> <p><b>Return Type</b></p> <p>String</p> <p><b>viewAllComplaintUser</b></p> <p><b>Parameters</b></p> <p>@ModelAttribute Complaint complaint, BindingResult result, ModelMap model</p> <p><b>Return Type</b></p> <p>String</p> <p><b>viewAnotherPageAdminComplaintList</b></p> <p><b>Parameters</b></p> <p>@Validated @ModelAttribute Admin admin, BindingResult result, ModelMap model, @PathVariable int pageNumber, @Param</p>	<ul style="list-style-type: none"><li>Return the name of user logged in.</li></ul> <ul style="list-style-type: none"><li>Add the complaint template and return complaint-creation.jsp page.</li></ul> <ul style="list-style-type: none"><li>Accept complaint as parameter. If a valid is received then set the complaint details and return complaint-submission-user.jsp page.</li><li>Else return complaint-creation.jsp page.</li></ul> <ul style="list-style-type: none"><li>Add all complaints and return complaint-notification-user.jsp page.</li></ul> <ul style="list-style-type: none"><li>Sort complaints based on id, date, category and return complaint-notification-admin.jsp page.</li></ul>
---------------------	---	--





	<p>String sortBy, @Param String sortDir, String keyword, String date</p> <p><b>Return Type</b></p> <p>String</p>	
	<p><b>viewAnotherPageAnalystComplaintList</b></p> <p><b>Parameters</b></p> <p>@Validated @ModelAttribute Analyst analyst, BindingResult result, ModelMap model, @PathVariable int pageNumber, @Param String sortBy, @Param String sortDir</p> <p><b>Return Type</b></p> <p>String</p>	<ul style="list-style-type: none"><li>Sort complaints based on id, date, category and return complaint-notification-analyst.jsp page.</li></ul>
	<p><b>getCategories()</b></p> <p><b>Parameters</b></p> <p>None</p> <p><b>Return Type</b></p> <p>Map&lt;String, String&gt;</p>	<ul style="list-style-type: none"><li>Return complaint category list.</li></ul>
LoginController	<p><b>getName</b></p> <p><b>Parameters</b></p> <p>ModelMap model</p> <p><b>Return Type</b></p> <p>String</p>	<ul style="list-style-type: none"><li>Return the name of user logged in.</li></ul>
	<p><b>userLogin</b></p> <p><b>Parameters</b></p> <p>ModelMap model, String error, String logout</p> <p><b>Return Type</b></p> <p>String</p>	<ul style="list-style-type: none"><li>Return login.jsp page.</li></ul>
	<p><b>Redirect</b></p> <p><b>Parameters</b></p>	<ul style="list-style-type: none"><li>If id between 1000 and 2000 is received then redirect to admin-home.jsp page.</li></ul>





	<p>ModelMap model</p> <p><b>Return Type</b></p> <p>String</p>	<ul style="list-style-type: none"><li>• If id between 2000 and 3000 is received then redirect to analyst-home.jsp page.</li><li>• If id between 3000 and 4000 is received then redirect to user-home.jsp page.</li><li>• Else return admin-home.jsp page.</li></ul>
	<p><b>getCategory</b></p> <p><b>Parameters</b></p> <p>None</p> <p><b>Return Type</b></p> <p>Map&lt;String, String&gt;</p>	<ul style="list-style-type: none"><li>• Return category list.</li></ul>
RegistrationController	<p><b>initBinder</b></p> <p><b>Parameters</b></p> <p>WebDataBinder binder</p> <p><b>Return Type</b></p> <p>void</p>	<ul style="list-style-type: none"><li>• Convert date into " yyyy-MM-dd" format.</li></ul>
	<p><b>registerAnalyst</b></p> <p><b>Parameters</b></p> <p>@Validated @ModelAttribute Analyst analyst, BindingResult result, ModelMap model</p> <p><b>Return Type</b></p> <p>String</p>	<ul style="list-style-type: none"><li>• If a valid analyst is received as parameter, then set the password and secret questions and save analyst to database and return analyst-reg-status.jsp page.</li><li>• Else return role-selection.jsp page.</li></ul>
	<p><b>registerUser</b></p> <p><b>Parameters</b></p> <p>@Validated @ModelAttribute User user, BindingResult result, ModelMap model</p> <p><b>Return Type</b></p> <p>String</p>	<ul style="list-style-type: none"><li>• If a valid user is received as parameter, then set the password and secret questions and save user to database and return user-reg-status.jsp page.</li><li>• Else return role-selection.jsp page.</li></ul>
	<p><b>populateGender</b></p> <p><b>Parameters</b></p>	<ul style="list-style-type: none"><li>• Return list.</li></ul>





	<b>Return Type</b> List<String> <b>getCategories</b> <b>Parameters</b>	
	<b>Return Type</b> Map<String, String>	<ul style="list-style-type: none"><li>• Return category list.</li></ul>
RoleSelectionController	<b>roleSelectionPage</b> <b>Parameters</b>  @ModelAttribute Admin admin, @ModelAttribute Analyst analyst, @ModelAttribute User user, BindingResult result, ModelMap model  <b>Return Type</b> String	<ul style="list-style-type: none"><li>• Return role-selection jsp for Admin, Analyst and user.</li></ul>
	<b>populateGender</b> <b>Parameters</b>	<ul style="list-style-type: none"><li>• Return list.</li></ul>
	<b>Return Type</b> List<String> <b>getCategories</b> <b>Parameters</b>	<ul style="list-style-type: none"><li>• Return category list.</li></ul>
	<b>Return Type</b> Map<String, String>	

### 5.5.2 Service Classes

Class	Method	Parameter Logic





AnalystService	<b>getLastId()</b>	<ul style="list-style-type: none"><li>• Call the getLastId() of AnalystService</li><li>• Check whether analystId greater than 2000 if it is greater then return analystId</li></ul>
	<b>Parameters</b> None	
	<b>Return Type</b> long lstId	
	<b>getAnalyst()</b> <b>Parameters</b> long analystId <b>Return Type</b> User	<ul style="list-style-type: none"><li>• Call the getAnalyst() of AnalystService by passing analystId</li><li>• it will return corresponding user</li></ul>
	<b>addAnalyst()</b> <b>Parameters</b> Analyst analyst <b>Return Type</b> boolean	<ul style="list-style-type: none"><li>• Call the addAnalyst() of AnalystService by passing analyst</li><li>• It will return true if analyst is added</li></ul>
	<b>getAllAnalystNotOfSupportLevel()</b> <b>Parameters</b> String supportLevel <b>Return Type</b> List<Analyst>	<ul style="list-style-type: none"><li>• Call the getAllAnalystNotOfSupportLevel() of AnalystService by passing analyst by passing supportLevel</li><li>• It will return list of analyst.</li></ul>





	<b>getAnalyst()</b>  <b>Parameters</b> None  <b>Return Type</b> List<Analyst>	<ul style="list-style-type: none"><li>Call the getAllAnalyst() of AnalystService. It will return the list of Analyst.</li></ul>
	<b>deleteAnalyst()</b>  <b>Parameters</b> long AnalystId  <b>Return Type</b> void	<ul style="list-style-type: none"><li>Call the deleteAnalyst() of AnalystService by passing analystId parameter to delete the corresponding analyst.</li></ul>
	<b>deleteAnalyst()</b>  <b>Parameters</b> Analyst  <b>Return Type</b> void	<ul style="list-style-type: none"><li>Call the deleteAnalyst() of AnalystService by passing analyst parameter to delete the corresponding analyst.</li></ul>
	<b>updateAnalyst()</b>  <b>Parameters</b> Analyst  <b>Return Type</b> Analyst	<ul style="list-style-type: none"><li>Call the updateAnalyst() of AnalystService by passing Analyst to update the Analyst.</li></ul>
	<b>getPasswordSHA()</b>  <b>Parameters</b> long analystId  <b>Return Type</b> String	<ul style="list-style-type: none"><li>Call the getPasswordSHA() of AnalystService by passing AnalystId to get the encrypted password.</li><li></li></ul>





	<b>checkAnswer()</b>  <b>Parameters</b> List<analystSecretQuestion> String ans1 String ans2 String ans3  <b>Return Type</b> Boolean	<ul style="list-style-type: none"><li>Call the CheckAnswer() of AnalystService by passing list of secret questions and answers to check whether answers typed by the analyst is correct or not to change password.</li></ul>
	<b>findAnalyst()</b>  <b>Parameters</b> long analystId String mob String email  <b>Return Type</b> Analyst	<ul style="list-style-type: none"><li>Call the findAnalyst() of AnalystService by passing AnalystId,mob,email to get the corresponding analyst</li></ul>
	<b>getLastId()</b>  <b>Parameters</b> None  <b>Return Type</b> long lstId	<ul style="list-style-type: none"><li>Call the getLastId() of AnalystSecretQuestionService</li><li>Check whether analystSecretQuestion id greater than 8000 if it is greater then return Id</li></ul>
EmailAnalystService	<b>getLastId()</b>  <b>Parameters</b> None  <b>Return Type</b> long	<ul style="list-style-type: none"><li>Call the getLastId () of AnalystSecretQuestionService</li></ul>
	<b>addMail()</b>  <b>Parameters</b>	<ul style="list-style-type: none"><li>Call the addEmail() of EmailAnalystService by passing email</li><li>It will return true if mail is added.</li></ul>





	<p>EmailAnalyst email</p> <p><b>Return Type</b></p> <p>boolean</p>	
	<p>getEmailAnalyst()</p> <p><b>Parameters</b></p> <p>long emailld</p> <p><b>Return Type</b></p> <p>EmailAnalyst</p>	<ul style="list-style-type: none"><li>Call the getEmailAnalyst() of EmailAnalystService by passing emailld</li><li>It will return EmailAnalyst</li></ul>
EmailUserAnalystService	<p>getLastId()</p> <p><b>Parameters</b></p> <p>None</p> <p><b>Return Type</b></p> <p>long lstld</p>	<ul style="list-style-type: none"><li>Call the getLastId() of EmailUserAnalystService</li><li>Check whether email is not empty and email id greater than 12000 if it is greater then return Id</li><li></li></ul>
	<p>addEMail()</p> <p><b>Parameters</b></p> <p>EmailUserAnalyst email</p> <p><b>Return Type</b></p> <p>boolean</p>	<ul style="list-style-type: none"><li>Call the addEmail() of EmailUserAnalystService by passing email</li><li>It will return true if mail is added.</li></ul>
	<p>geEtMail()</p> <p><b>Parameters</b></p> <p>long emailld</p> <p><b>Return Type</b></p> <p>EmailUserAnalyst</p>	<ul style="list-style-type: none"><li>Call the getEmail() of EmailUserAnalystService by passing emailld</li><li>It will return EmailUserAnalyst</li></ul>
FeedbackService	<p>addFeedback()</p> <p><b>Parameters</b></p>	<ul style="list-style-type: none"><li>Call the addFeedback() of FeedbackService by passing feedback</li><li>It will return true if feedback is added.</li></ul>





	<p>Feedback feedback</p> <p><b>Return Type</b></p> <p>boolean</p>	
	<p>getFeedback()</p> <p><b>Parameters</b></p> <p>long responseld</p> <p><b>Return Type</b></p> <p>Feedback</p>	<ul style="list-style-type: none"><li>Call the getFeedback() of FeedbackService by passing response id</li><li>It will return feedback object</li></ul>
	<p>getAllFeedback()</p> <p><b>Parameters</b></p> <p>None</p> <p><b>Return Type</b></p> <p>List&lt;Feedback&gt;</p>	<ul style="list-style-type: none"><li>Call the getAllFeedback() of FeedbackService .It will return list of feedback objects.</li></ul>
	<p>updateFeedback()</p> <p><b>Parameters</b></p> <p>Feedback feedback</p> <p><b>Return Type</b></p> <p>Boolean</p>	<ul style="list-style-type: none"><li>Call the updateFeedback() of FeedbackService by passing feedback</li><li>It will return true if it is updated</li></ul>
	<p>deleteFeedback()</p> <p><b>Parameters</b></p> <p>long responseld</p> <p><b>Return Type</b></p> <p>Void</p>	<ul style="list-style-type: none"><li>Call the deleteFeedback() of FeedbackService by passing response id</li></ul>
	<p>deleteFeedback()</p> <p><b>Parameters</b></p>	<ul style="list-style-type: none"><li>Call the deleteFeedback() of FeedbackService by passing feedback .</li></ul>





	<p>Feedback feedback</p> <p><b>Return Type</b></p> <p>Void</p>	
LoginDetailsService	<p>getLoginDetailsByRegisteredId()</p> <p><b>Parameters</b></p> <p>long registeredId</p> <p><b>Return Type</b></p> <p>LoginDetails</p>	<ul style="list-style-type: none"><li>Call the getLoginDetailsByRegisteredId() of LoginDetailsService by passing registeredId.</li><li>It will return loginDetails</li></ul>
	<p>addLoginDetails()</p> <p><b>Parameters</b></p> <p>LoginDetails loginDetails</p> <p><b>Return Type</b></p> <p>boolean</p>	<ul style="list-style-type: none"><li>Call the addLoginDetails() of LoginDetailsService by passing loginDetails.</li><li>It will return true if it is added.</li></ul>
	<p>modifyLoginDetails()</p> <p><b>Parameters</b></p> <p>LoginDetails loginDetails</p> <p><b>Return Type</b></p> <p>boolean</p>	<ul style="list-style-type: none"><li>Call the modifyLoginDetails() of LoginDetailsService by passing loginDetails.</li><li>It will return true if it is updated.</li></ul>
	<p>getQuestionById()</p> <p>Parameters</p> <p>long questionId</p> <p>Return Type</p> <p>SecretQuestions</p>	<ul style="list-style-type: none"><li>Call the getQuestionById() of SecretQuestionService by passing questionId.</li><li>It will return SecretQuestions object.</li></ul>
	<p>getAllQuestions()</p> <p>Parameters</p>	<ul style="list-style-type: none"><li>Call the getQuestion() of SecretQuestionService it will return list of secret questions.</li></ul>





	None  Return Type  List<SecretQuestion>	
	getAllExcept()  Parameters  List<SecretQuestion> alreadyAdded  Return Type  List<SecretQuestion>	<ul style="list-style-type: none"><li>• Call the getAllExcept() of SecretQuestionService by using list of secret questions it will return list of secret questions.</li></ul>
	getAllQuestionDescription()  Parameters  None  Return Type  List<String>	<ul style="list-style-type: none"><li>• Call the getAllQuestionDescription() of SecretQuestionService it will return list.</li></ul>
	getLastId()  Parameters  None  Return Type  long lstId	<ul style="list-style-type: none"><li>• Call the getLastId() of UserSecretQuestionService</li><li>• Check whether user secret question id is greater than 7000 if it is greater then return question id.</li></ul>
	addSecretQuestion()  Parameters  UserSecretQuestion question  Return Type  boolean	<ul style="list-style-type: none"><li>• Call the addSecretQuestion() of UserSecretQuestionService by passing question.it will return true if it is added.</li></ul>





	<b>getLastId()</b> Parameters None Return Type long	<ul style="list-style-type: none"><li>Call the getLastId() of ComplaintService .it will return id.</li></ul>
	<b>addComplaint()</b> Parameters Complaint complaint Return Type Boolean	<ul style="list-style-type: none"><li>Call the addComplaint() of ComplaintService by passing complaint.it will return true if complaint is added.</li></ul>
	<b>getComplaint()</b> Parameters long complaintId Return Type Complaint	<ul style="list-style-type: none"><li>Call the getComplaint() of ComplaintService by passing complaintId.it will return Complaint object.</li></ul>
	<b>updateComplaint()</b> Parameters Complaint complaint Return Type boolean	<ul style="list-style-type: none"><li>Call the updateComplaint() of ComplaintService by passing complaint.it will return true if it is updated.</li></ul>
	<b>deleteComplaint()</b> Parameters long complaintId Return Type Boolean	<ul style="list-style-type: none"><li>Call the deleteComplaint() of ComplaintService by passing complaintId .it will return true if deleted.</li></ul>





	<p>getAllComplaintOfUser()</p> <p>Parameters</p> <p>long userId</p> <p>Return Type</p> <p>List&lt;Complaint&gt;</p>	<ul style="list-style-type: none"><li>Call the getAllComplaintOfUser() of ComplaintService by passing userId .it will return list of complaint.</li></ul>
	<p>findByKeyword()</p> <p>Parameters</p> <p>String keyword</p> <p>Return Type</p> <p>List&lt;Complaint&gt;</p>	<ul style="list-style-type: none"><li>Call the findByKeyword() of ComplaintService by passing keyword .it will return list of complaint.</li></ul>
	<p>getAllComplaintForUser()</p> <p>Parameters</p> <p>long userId</p> <p>int pageNumber</p> <p>int count</p> <p>String sortBy</p> <p>String sortDir</p> <p>Return Type</p> <p>Page&lt;complaint&gt;</p>	<ul style="list-style-type: none"><li>Call the getAllComplaintUser() of ComplaintUser by passing userId,pageNumber,count,sortBy,sortDir.</li></ul>
	<p>getAllComplaint()</p> <p>Parameters</p> <p>int pageNumber</p> <p>int count</p> <p>String sortBy</p> <p>String sortDir</p> <p>Return Type</p> <p>Page&lt;complaint&gt;</p>	<ul style="list-style-type: none"><li>Call the getAllComplaint() of ComplaintUser by passing count,pageNumber,sortBy,sortDir. It will return page.</li></ul>





	<p>getAllComplaintFiltered()</p> <p>Parameters</p> <p>String keyword</p> <p>String date</p> <p>int pageNumber</p> <p>int count</p> <p>String sortBy</p> <p>String sortDir</p> <p>Return Type</p> <p>Page&lt;complaint&gt;</p>	<ul style="list-style-type: none"><li>Call the getAllComplaintFiltered() of ComplaintUser by passing keyword,date,count,pageNumber,sortBy,sortDir. It will return page</li></ul>
	<p>getAllComplaintForAnalyst()</p> <p>Parameters</p> <p>long analystId</p> <p>int pageNumber</p> <p>int count</p> <p>String sortBy</p> <p>String sortDir</p> <p>Return Type</p> <p>Page&lt;complaint&gt;</p>	<ul style="list-style-type: none"><li>Call the getAllComplaintForAnalyst() of ComplaintUser by passing analystId,count,pageNumber,sortBy,sortDir. It will return page.</li></ul>
	<p>findDate()</p> <p>Parameters</p> <p>String date</p> <p>Return Type</p> <p>List&lt;Complaint&gt;</p>	<ul style="list-style-type: none"><li>Call the findDate() of ComplaintUser by passing date. It will return list of complaint in the date.</li></ul>
	<p>findById()</p> <p>Parameters</p>	<ul style="list-style-type: none"><li>Call the findById() of ComplaintUser by passing userId. It will return list of complaint of userId.</li></ul>





	<p>String userId Return Type List&lt;Complaint&gt;</p>	
	<p>findByComplaintId() Parameters String complaintId Return Type List&lt;Complaint&gt;</p>	<ul style="list-style-type: none"><li>Call the findByComplaintId() of ComplaintUser by passing complaintId. It will return list of complaint .</li></ul>
	<p>getAdmin() Paramenters long adminId Return Type Admin</p>	<ul style="list-style-type: none"><li>Call the getAdmin() of AdminService by passing adminId</li><li>To get the Admin having corresponding adminId</li></ul>
	<p>getAdminPassword() Parameters long adminId Return Type String</p>	<ul style="list-style-type: none"><li>Call the getAdminPassword() of AdminService by passing adminId.</li></ul>
	<p>getAdminPassword() Parameters long adminId Return Type String</p>	<ul style="list-style-type: none"><li>Call the getPasswordSHA() of AdminService by passing analystId.</li></ul>
	<p>getLastId() Parameters None</p>	<ul style="list-style-type: none"><li>Call the getLastId() of UserService</li><li>Check whether userId greater than 3000 if it is greater then return userId</li></ul>





	Return Type long lstId	
	getUser()  Parameters long userId  Return Type User	<ul style="list-style-type: none"><li>Call the getUser() of UserService by passing userId it will return corresponding user</li></ul>
	getAllUsers()  Parameters None  Return Type List<user>	<ul style="list-style-type: none"><li>Call the getAllUsers() of UserService</li><li>It will return the list of users</li></ul>
	deleteUser()  Parameters User user  Return Type boolean	<ul style="list-style-type: none"><li>Call the deleteUser() of UserService by passing user parameter to delete the corresponding user</li></ul>
	getPasswordSHA()  Parameters long userId  Return Type String	<ul style="list-style-type: none"><li>Call the getPasswordSHA() of UserService by passing userId to get the encrypted password.</li></ul>
	findUser()  Parameters long userId String mob String email	<ul style="list-style-type: none"><li>Call the findUser() of UserService by passing userId,mob,email to get the corresponding user</li></ul>





	Return Type User	
	checkAnswer()  Parameters List<userSecretQuestion>  String ans1  String ans2  String ans3  Return Type  Boolean	<ul style="list-style-type: none"><li>Call the CheckAnswer() of UserService by passing list of secret questions and answers to check whether answers typed by the user is correct or not to change password.</li></ul>
	findUserId()  Parameters String email  Return Type  User	<ul style="list-style-type: none"><li>Call the findUserId() by passing email to get corresponding user.</li></ul>
	validateAnswer()  Parameters List<userSecretQuestion>  String ans1  String ans2  String ans3  Return Type  boolean	<ul style="list-style-type: none"><li>Call the validAnswer() of UserService by passing list of secret questions and answers to check whether answers typed by the user correct or not to get user id.</li></ul>
	updateUser()  Parameters User user  Return Type	<ul style="list-style-type: none"><li>Call the updateUser() of UserService by passing user to update the user.</li><li></li></ul>





	boolean	
--	---------	--

## 6.0 Database Design

### 6.1 Table Structure



db\_schema\_v5.pdf

#### 6.1.1 User Table

Table Name	Column Name	Column Description	Data Type	Data Size	Primary Key	Other Constraint
USER	USER_ID	Column is used for storing User Id	INT		YES	NOT NULL AUTO_INCREMENT
	FIRST_NAME	Column for storing First Name	VARCHAR	45		NOT NULL
	LAST_NAME	Column for storing Last Name of User	VARCHAR	45		NOT NULL
	PHONE_NUMBER	Column for storing Phone Number of User	VARCHAR	15		
	EMAIL_ID	Column for storing Email ID of User	VARCHAR	45		NOT NULL
	DATE_OF_BIRTH	Column for storing Date of birth of User	DATETIME			NOT NULL
	GENDER	Column for storing gender of User	VARCHAR	10		NOT NULL
	PASSWORD	Column for storing password of User	VARCHAR	256		NOT NULL
	TEMP_PASSWORD	Column for storing temporary password	VARCHAR	256		NOT NULL





ADMIN	ADMIN_ID	Column is used for storing Admin Id	INT		YES	NOT NULL AUTO_INCREMENT
	FIRST_NAME	Column for storing first name of admin	VARCHAR	45		NOT NULL
	LAST_NAME	Column for storing last name of admin	VARCHAR	45		NOT NULL
	EMAIL_ID	Column for storing Email ID of Admin	VARCHAR	45		NOT NULL
	PASSWORD	Column for storing password of Admin	VARCHAR	256		NOT NULL
	TEMP_PASSWORD	Column for storing temporary password	VARCHAR	256		NOT NULL
ANALYST	ANALYST_ID	Column is used for storing Analyst id.	INT		YES	NOT NULL AUTO_INCREMENT
	FIRST_NAME	Column for storing the first name of Analyst.	VARCHAR	45		NOT NULL
	LAST_NAME	Column is used for storing last name of analyst	VARCHAR	45		NOT NULL
	PHONE_NUMBER	Column is used for Phone Number of Analyst	VARCHAR	15		NOT NULL
	EMAIL	Column is used for storing Email ID	VARCHAR	45		NOT NULL
	DATE_OF_BIRTH	Column is used for storing date of birth of Analyst	DATETIME			NOT NULL
	GENDER	Column is used for storing gender of Analyst	VARCHAR	45		NOT NULL
	SUPPORT_LEVEL	Column is used for storing support level of Analyst	VARCHAR	45		NOT NULL
	PASSWORD	Column is used for storing password of Analyst	VARCHAR	256		NOT NULL
	TEMP_PASSWORD	Column is used for storing temporary password of Analyst	VARCHAR	256		NOT NULL





COMPLAINT	COMPLAINT_ID	Column is used for storing Complaint id.	INT		YES	NOT NULL AUTO_INCREMENT
	USER_ID	Column is used for storing User id	INT			FOREIGN KEY (USER.USER_ID), NOT NULL
	ASSIGNED_ANALYST_ID	Column is used for storing Analyst id	INT			FOREIGN KEY (ANALYST.ANALYST_ID), NOT NULL
CATEGORY		Column is used for storing Category of Complaint	VARCHAR	45		NOT NULL
PHONE_NUMBER		Column is used for storing Phone Number	VARCHAR	15		NOT NULL
STATUS		Column is used for storing Status of Complaint	VARCHAR	45		NOT NULL
DATE_OF_COMPLAINT		Column is used for storing date of Complaint	DATETIME			NOT NULL
DESCRIPTION		Column is used for storing description of Complaint	VARCHAR	1024		NOT NULL
SUGGESTIONS		Column is used for storing Suggestions	VARCHAR	1024		
SECRET_QUESTIONS	QUESTION_ID	Column is used for storing Question Id	INT		YES	AUTO_INCREMENT NOT NULL
	QUESTION_DESCRIPTION	Column is used for storing Question description	VARCHAR	45		NOT NULL
U_SQ_QUESTIONS	U_SQ_ID	Column is used for storing User secret Question id	INT		YES	NOT NULL
	USER_ID	Column is used for storing User id	INT			NOT NULL FOREIGN KEY (USER.USER_ID)
	QUESTION_ID	Column is used for storing Question id	INT			NOT NULL FOREIGN KEY (SECRET_QUESTION.QUESTION_ID)





	ANSWER	Column is used for storing Answer	VARCHAR	1024		
EMAIL_ANALYST	EMAIL_ID	Column is used for storing Email id	INT		YES	NOT NULL AUTO_INCREMENT
	ADMIN_ID	Column is used for storing Admin id	INT			NOT NULL FOREIGN KEY (ADMIN.ADMIN_ID)
	ANALYST_ID	Column is used for storing Analyst id	INT			NOT NULL FOREIGN KEY (ANALYST.ANALYST_ID)
	SENT_DATE	Column is used for storing Sent date	DATETIME			NOT NULL
	RECIEVED	Column is used for storing Received	TINYINT			NOT NULL
	DESCRIPTION	Column is used for storing Description of complaint	VARCHAR	1024		NOT NULL
EMAIL_USER	EMAIL_ID	Column is used for storing Email id	INT		YES	NOT NULL AUTO_INCREMENT
	ADMIN_ID	Column is used for storing Admin id	INT			NOT NULL FOREIGN KEY (ADMIN.ADMIN_ID)
	USER_ID	Column is used for storing User id	INT			NOT NULL FOREIGN KEY (USER.USER_ID)
	SENT_DATE	Column is used for storing Sent date	DATETIME			NOT NULL
	RECIEVED	Column is used for storing Received	TINYINT			NOT NULL
	DESCRIPTION	Column is used for storing Description of complaint	VARCHAR	1024		NOT NULL
FEEDBACK_RESPONSE	FEEDBACK_RESPONSE_ID	Column is used for storing Feedback response id	INT		YES	NOT NULL AUTO_INCREMENT
	COMPLAINT_COMPLAINT_ID	Column is used for storing Complaint id	INT			NOT NULL FOREIGN KEY (COMPLAINT.COMPLAINT_COMPLAINT_ID)





	QUESTION	Column is used for storing Questions	VARCHAR	1024		
	ANSWER	Column is used for storing Answers	VARCHAR	1024		
LOGIN_DETAIL_S	LOGIN_DETAILS_ID	Column is used for storing Login details id	INT		YES	NOT NULL AUTO_INCREMENT
	USER_NAME	Column is used for storing User name	VARCHAR	45		NOT NULL
	PASSWORD	Column is used for storing Password	VARCHAR	45		NOT NULL
	ROLES	Column is used for storing Roles	VARCHAR	45		NOT NULL
A_SQ_QUESTI ONS	A_SQ_ID	Column is used for storing Analyst secret question id	INT		YES	NOT NULL AUTO_INCREMENT
	ANALYST_ID	Column is used for storing Analyst id	INT			NOT NULL FOREIGN KEY (ANALYST.ANALYST_ID)
	QUESTION_ID	Column is used for storing Question id	INT			NOT NULL FOREIGN KEY (SECRET_QUESTI ON.QUESTION_ID)
	ANSWER	Column is used for storing Analyst secret question Answer	VARCHAR	1024		
EMAIL_USER_ ANALYST	EMAIL_ID	Column is used for storing email id.	INT		YES	NOT NULL AUTO_INCREMENT
	USER_ID	Column is used for storing user id.	INT			NOT NULL FOREIGN KEY (USER.USER_ID)
	ANALYST_ID	Column is used for storing analyst id.	INT			NOT NULL FOREIGN KEY (ANALYST.ANALYST_ID)
	SENT_DATE	Column is used for storing sent date.	DATETIME			
	RECIEVED	Column is used for storing received.	TINYINT			NOT NULL



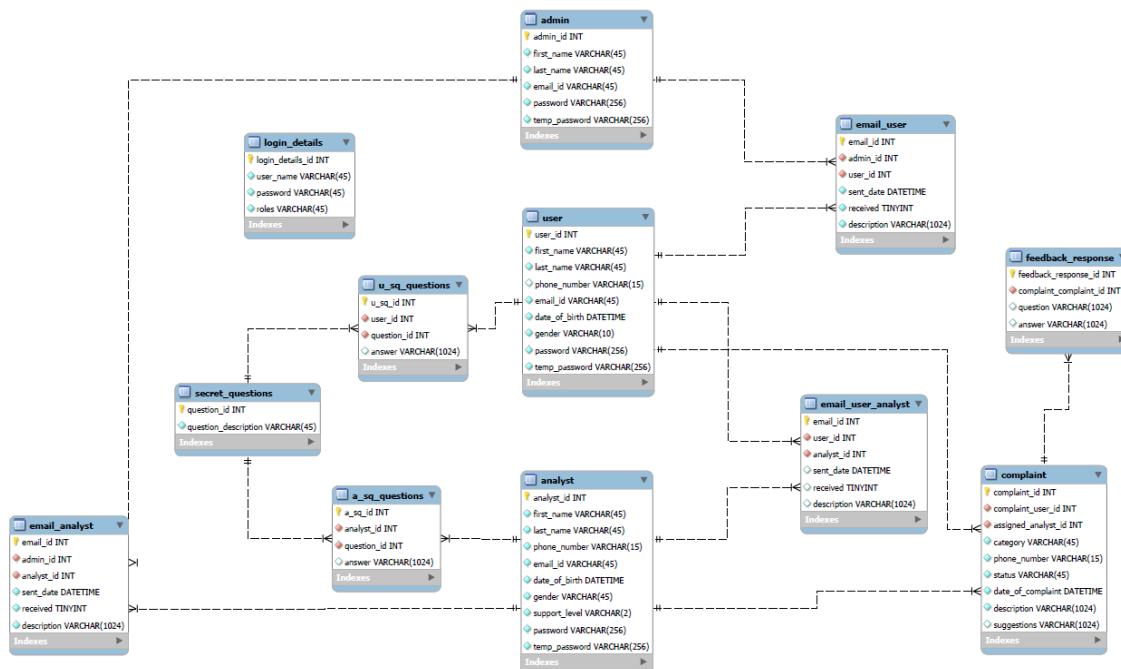


	DESCRIPTION	Column is used for storing description.	VARCHAR	1024			
--	-------------	---	---------	------	--	--	--





## 6.2 E-R Diagram (Relationship between Tables)





## 6.3 Standards and Guidelines Followed

### 6.3.1 JAVA

- 1) Ensure that the class names, method names and variable names are followed exactly as specified in the class diagram
- 2) Naming standards to be followed:
  - a) Variable
    - i) loweCamelCase
    - ii) Short variable
  - b) Class
    - i) Is a noun
  - c) Methods
    - i) Is a Verb

## 7.0 Change Log

*Please note that this table needs to be maintained even if a Configuration Management tool is used.*

Version Number	Changes made
V1.0	<First version>

