



Online Customer Care and Service Centre

Project Id : A073 <SC.ID>

Release Id :

Application Design



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





Contents

1.0 Introduction	4
1.1 Purpose of the Document.....	4
1.2 Scope	4
1.2.1 In Scope	4
1.2.2 Out Scope	4
1.3 Intended Audience	4
2.0 Conventions and Standards.....	5
3.0 Assumptions, Dependencies and Risks.....	5
4.0 System Architecture.....	6
5.0 Use Case Realization	7
5.1 Flow of Events	7
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
5.2 System Object Model	11
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
5.3 UI Controls	16
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
5.4	UI Design	27
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
5.5	Component Description.....	35
5.5.1	Controller Classes	35
5.5.2	Service Classes.....	49
6.0	Database Design.....	62
6.1	Table Structure	62
6.1.1	User Table	62
6.2	E- R Diagram (Relationship between Tables).....	68
6.3	Standards and Guidelines Followed	69
6.3.1	JAVA	69
7.0	Change Log.....	69





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

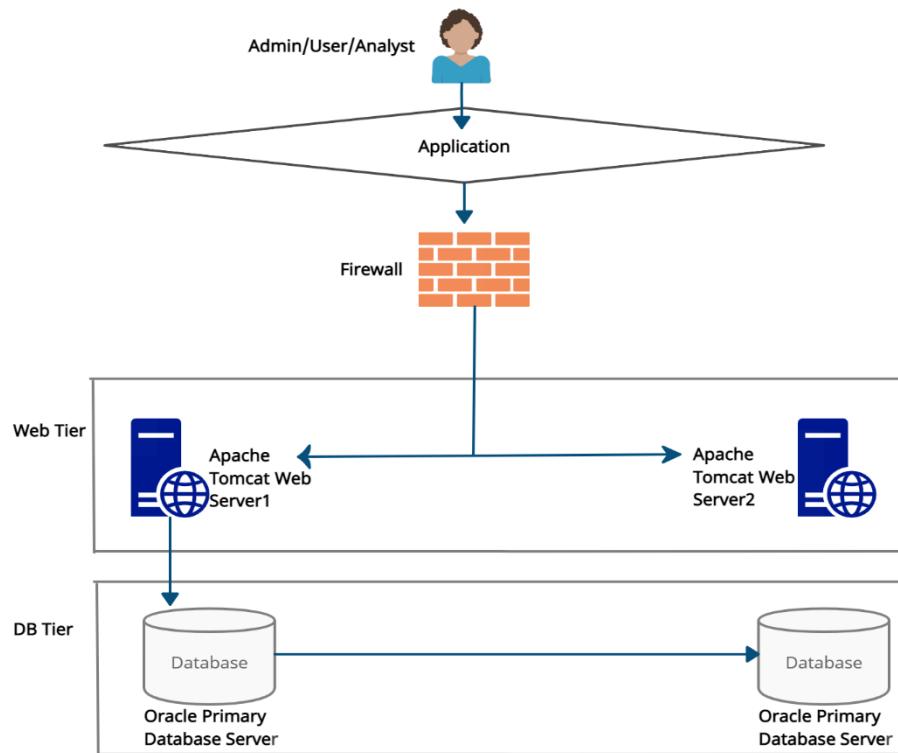
Risks

- 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.

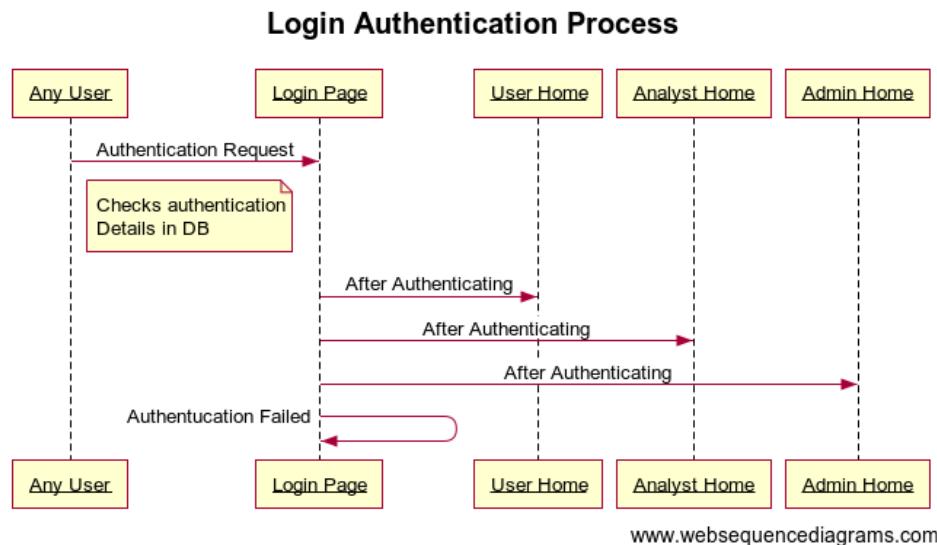




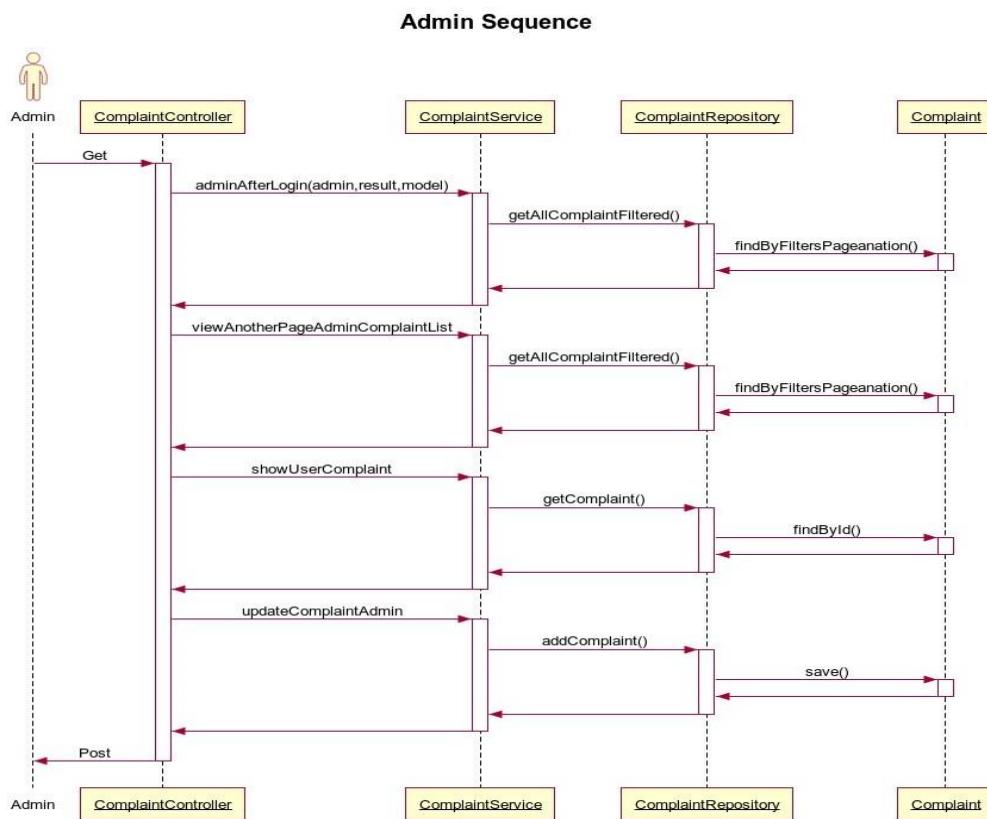
5.0 Use Case Realization

5.1 Flow of Events

5.1.1 Login Sequence

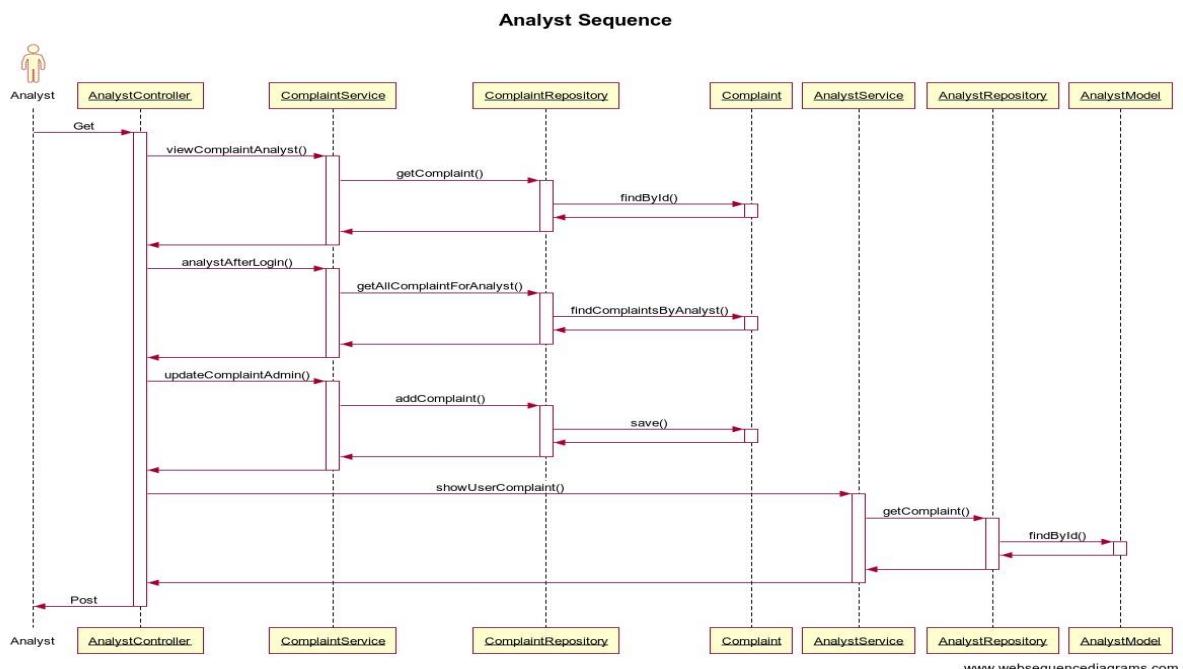


5.1.2 Admin Sequence

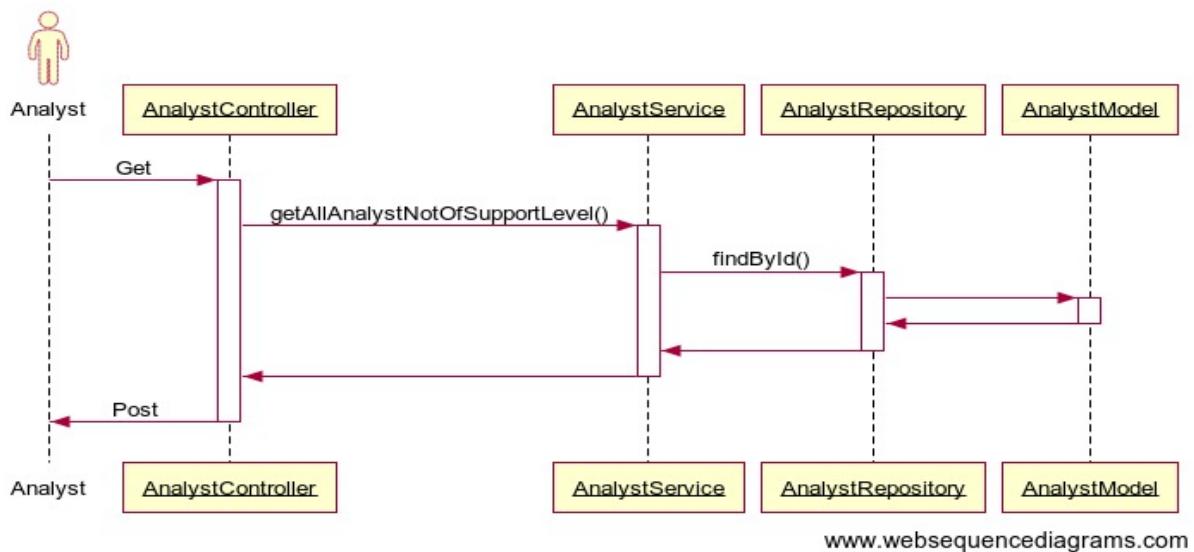




5.1.3 Analyst Sequence



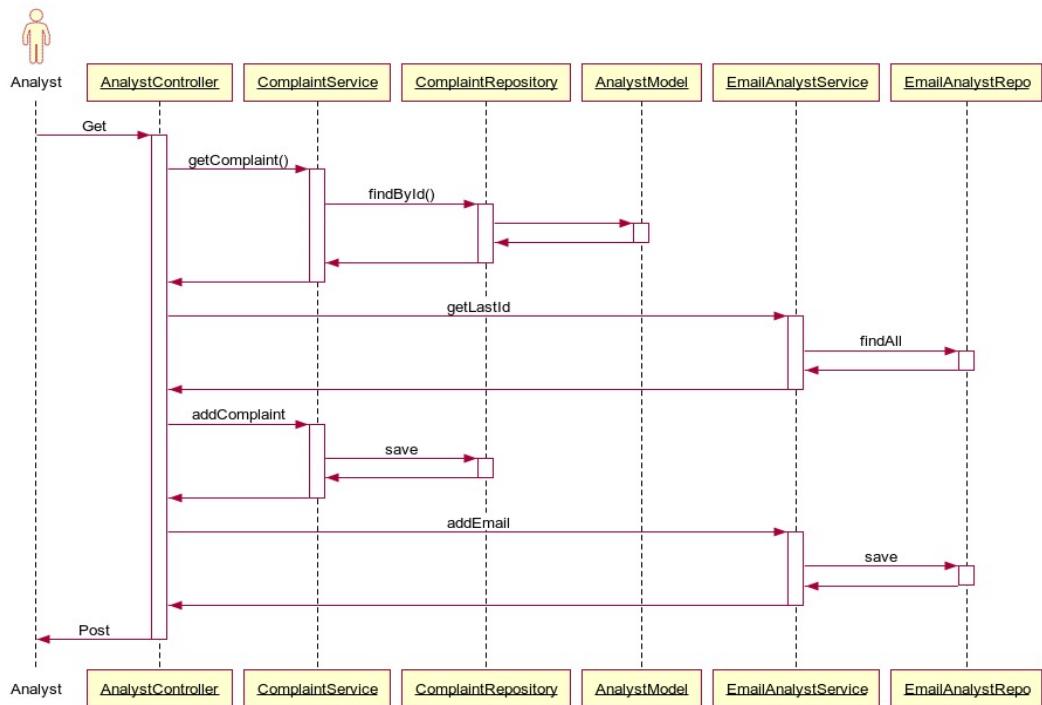
Analyst Sequence





5.1.4 Email Analyst Sequence

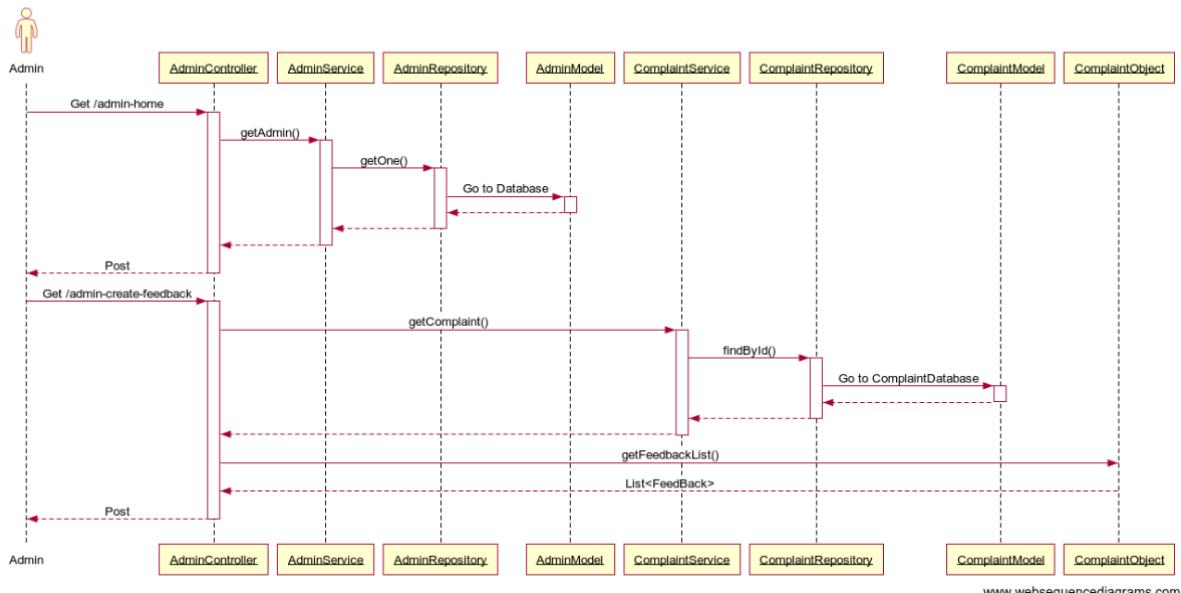
Email Analyst Sequence - Update Complaint



www.websequencediagrams.com

5.1.5 Admin Controller Feedback

Admin Sequence

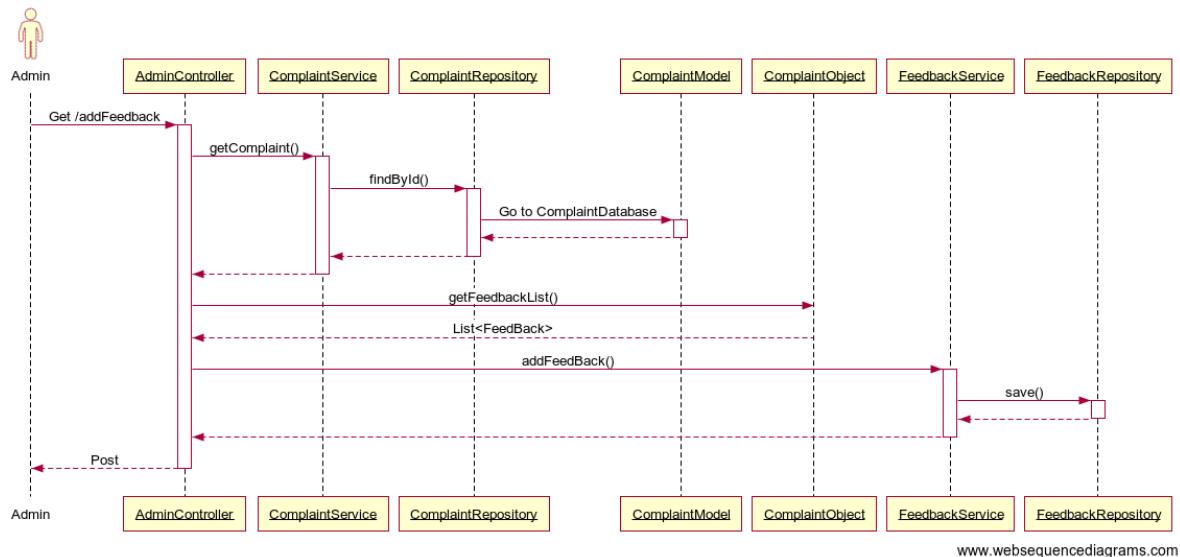


www.websequencediagrams.com

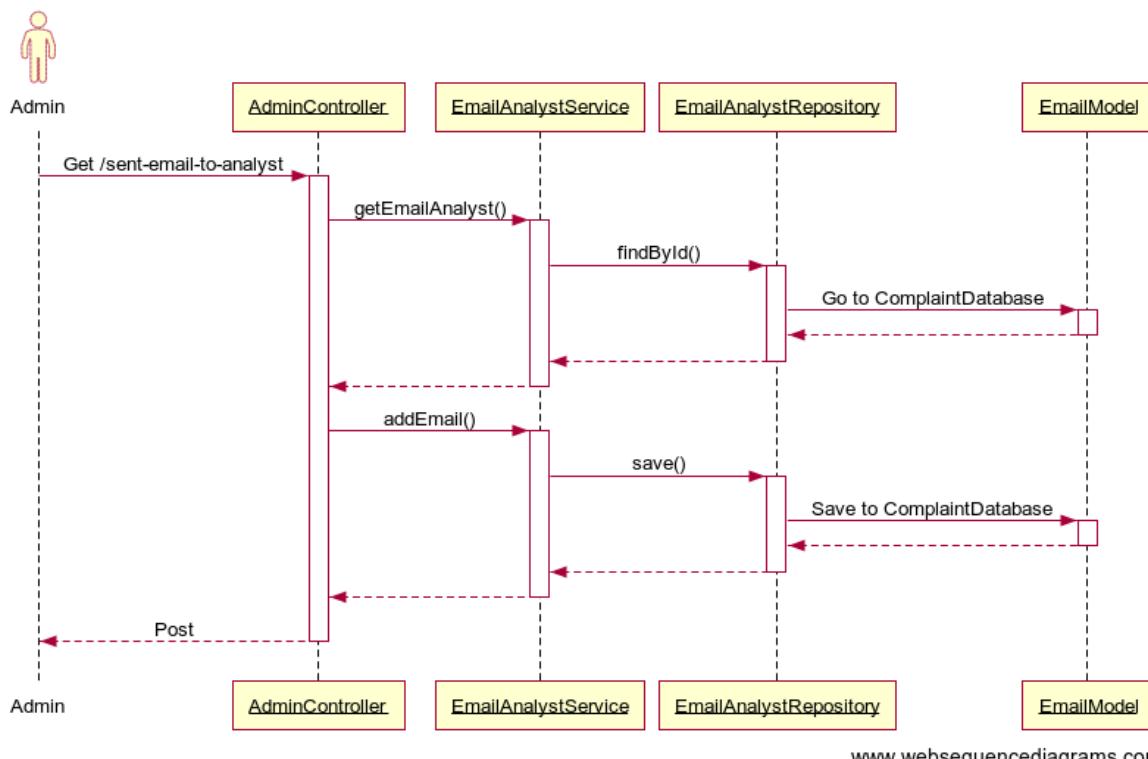




5.1.6 Admin Add Feedback

Admin Add Feedback

5.1.7 Admin Email to Analyst

Admin Sent Email to Analyst



5.2 System Object Model

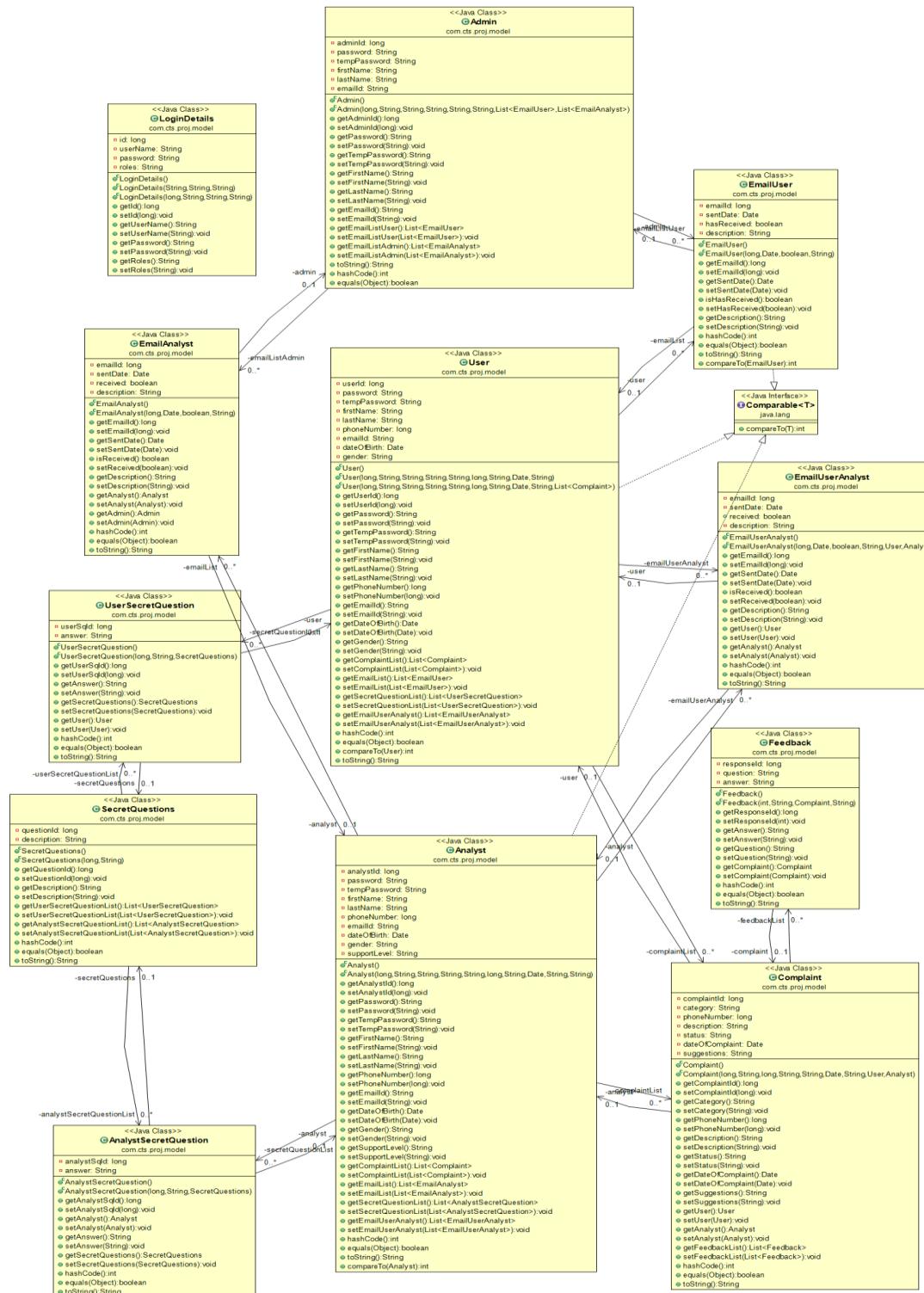
5.2.1 Model Classes

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

Overall 12 model classes are present.

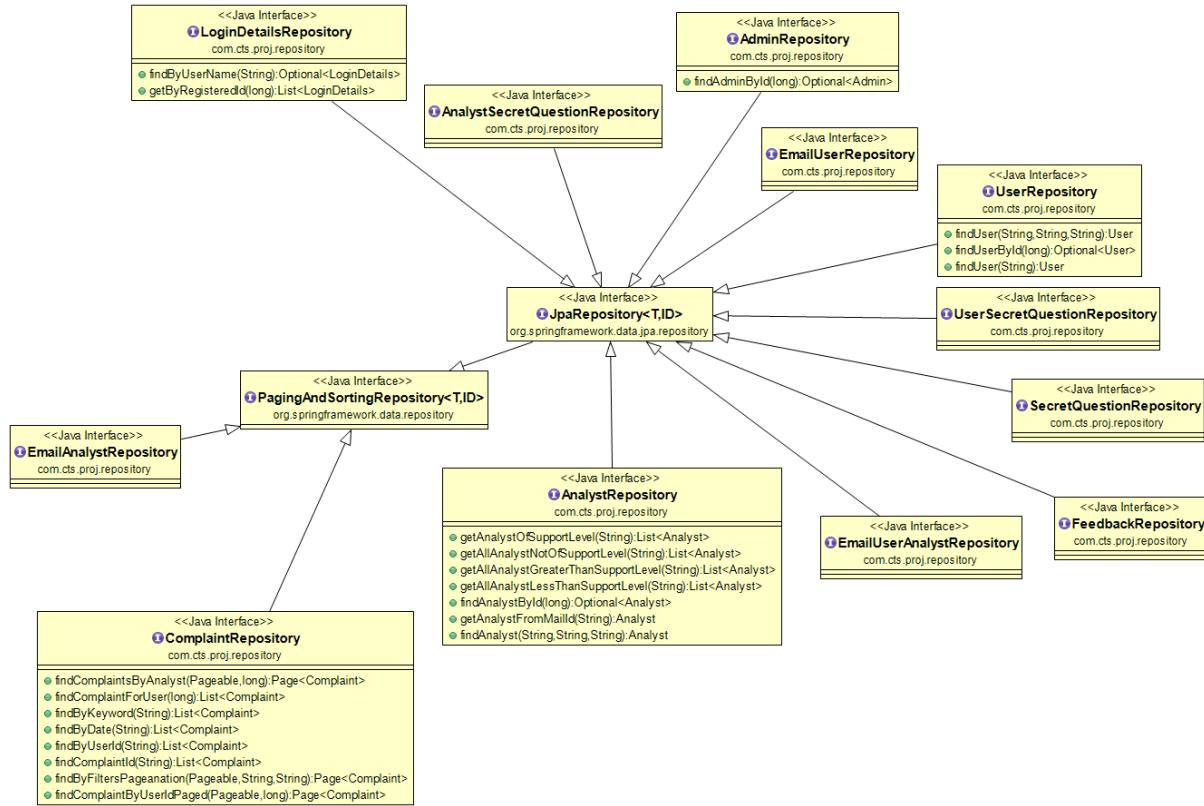
1. **User**: to store the information about the user
2. **Analyst**: to store info about the analyst
3. **Admin**: to store the information about the admin
4. **Complaint**: to store the info about the Complaint
5. **Feedback**: to store the feedback provided by user
6. **EmailAnalyst**: To store email 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.



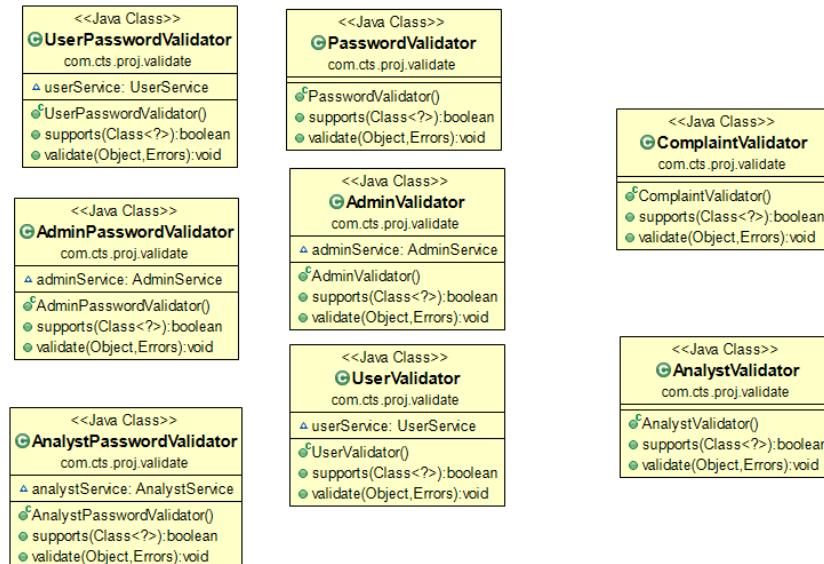




5.2.2 Repositories

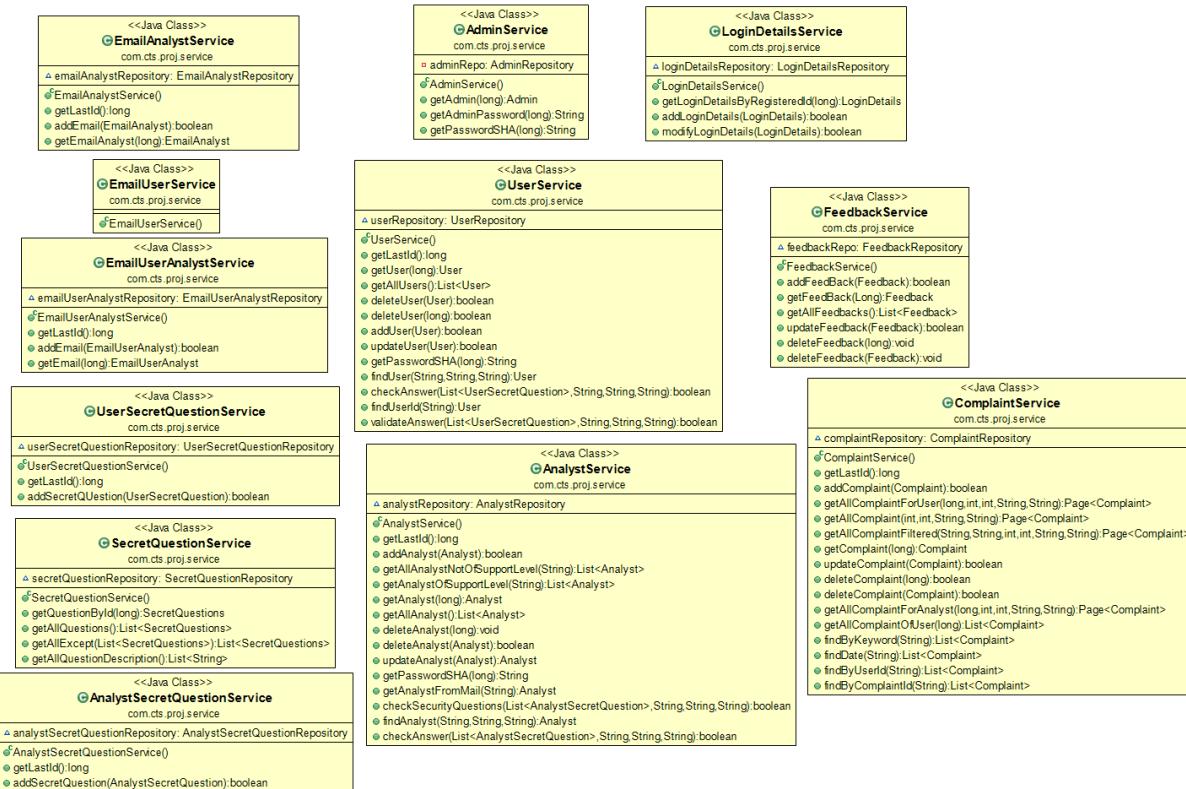


5.2.3 Validator Classes

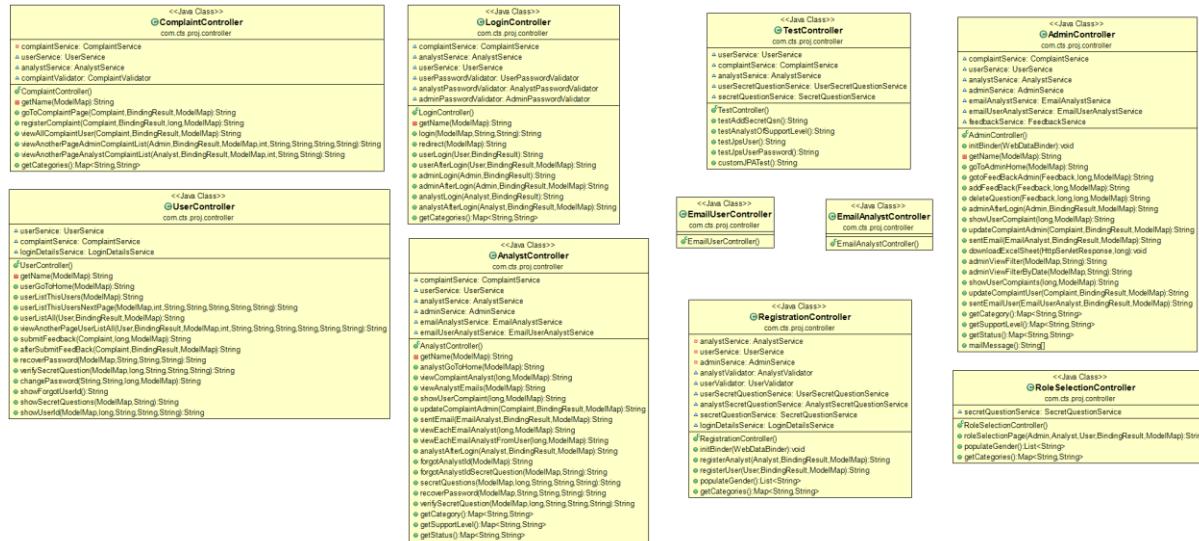




5.2.4 Service Classes

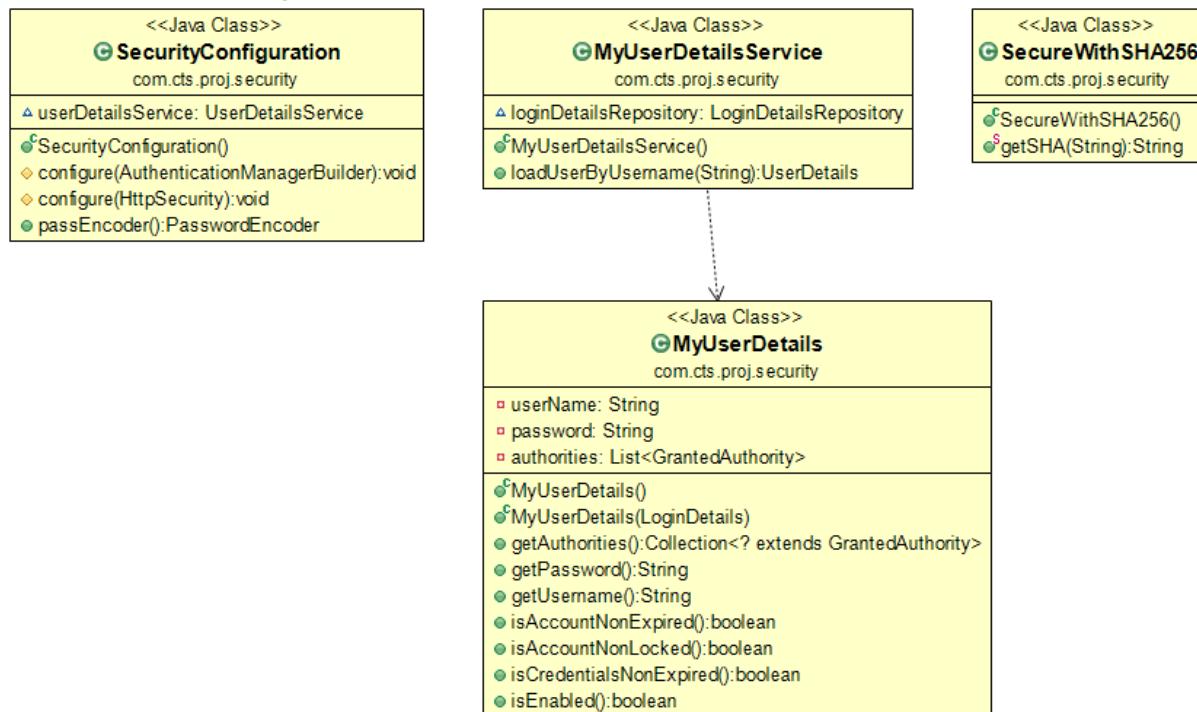


5.2.5 Controller Classes





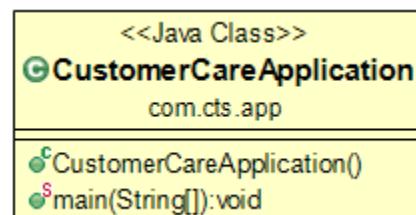
5.2.6 Security



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	Email
Date Of Birth	Gender
Date Of Birth	Male
Password	Confirm 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?

Register

OR

Login

[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**

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:
localhost:9000/roleSelectionPage#role=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 "Customer Care", "Home", and "Menu". Below the title, it displays "Total Complaints : 3" and "Page 1 of 1". There are search filters for "Search by Category", "YYYY-MM-DD", "3001", and "Enter complaintId", each with a "Search" button. A "Sort by" dropdown menu is set to "Complaint Id". The main content area lists two complaints:

- 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 "Customer Care" link and a "Menu" dropdown. Below the title, it displays "Total Complaints : 3" and "Page 1 of 1". There are search filters for "Search by Category", "YYYY-MM-DD", "Search by User Id", and "Enter complaintId", each with a "Search" button. A "Sort by" dropdown menu is set to "Complaint Id". The main content area lists two complaints:

- 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 Menu ▾

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

© 2020 Copyright:

5.4.8 5.4.6 Analyst/Admin Home

Customer Care Menu ▾

Welcome

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

© 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 : 4001
Description
Put description Here [View](#)

Complaint Id : 4002
Description
Please Fill The Description [View](#)

Complaint Id : 4003
Description
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 srnv e th sr dhv wr [View](#)

© 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 Page 1 of 2

1 2

Search by Category YYYY-MM-DD

Search Search

Complaint Id : 4001
Description Complaint Created By : 3001 Assigned Analyst Id 2001 reopen
Put description Here
View Create Feedback
Complaint Id : 4002
Description Complaint Created By : 3001 Assigned Analyst Id 2001 Active
Please Fill The Description
View Create Feedback
Complaint Id : 4003
Description Complaint Created By : 3001 Assigned Analyst Id 2001 reopen
DSAV f ceger vgew v sg c gd dsfc w fcvs eg ve gv vs e svh sbv b sh caedact e gt srt cv w svvv e th sr dhv wr

© 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	Delete
2	Would You suggest this to your friends	Delete
3	Will you come to us in the future	Delete

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, Unassigned Please do the needfull to rectify the complaint the user Linj.....	View
10002	2020-12-16 22:11:17.0	Dear, Unassigned Please do the needfull to rectify the complaint the user LINJ.....	View
12005	2020-12-16 12:17:53.0	Dear, Unassigned Please do the needfull to rectify the complaint the user Linjo	View
12006	2020-12-16 12:19:51.0	Dear, Unassigned Please do the needfull to rectify the complaint the user Linj.....	View
12007	2020-12-16 17:54:41.0	Dear, Unassigned Please do the needfull to rectify the complaint the user Linj.....	View
12008	2020-12-17 08:58:57.0	Dear, Unassigned Please do the needfull to rectify the complaint the user Linj.....	View

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





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





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





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





	Parameters None Return Type String[]	
AnalystController	getName Parameters ModelMap model Return Type String	<ul style="list-style-type: none">• Return the name of user logged in.
	analystGoToHome Parameters ModelMap model Return Type String	<ul style="list-style-type: none">• Return analyst-home jsp page.
	viewComplaintAnalyst Parameters @RequestParam long complaintId, ModelMap model Return Type String	<ul style="list-style-type: none">• Return complaint-analyst-view jsp page.
	viewAnalystEmails Parameters ModelMap model Return Type String	<ul style="list-style-type: none">• Return emails-analyst jsp page.
	showUserComplaint Parameters	<ul style="list-style-type: none">• Accept complaint Id as parameter and add complaint details and return complaint-analyst-view jsp page.





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





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





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





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





	viewAnotherPageUserListAll Parameters <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> Return Type String	<ul style="list-style-type: none"> Filter the complaints based on category, date, user Id and complaint Id and return complaint-notification-user jsp page.
	submitFeedback Parameters <pre>@ModelAttribute Complaint complaint, @RequestParam long complaintId, ModelMap model</pre> Return Type String	<ul style="list-style-type: none"> Accept complaint Id as parameter and get complaint and feedback details and return user-feedback-submission jsp page.
	afterSubmitFeedBack Parameters <pre>@ModelAttribute Complaint complaint, BindingResult result, ModelMap model</pre> Return Type String	<ul style="list-style-type: none"> Set answer for feedback questions and add it to the complaint. Then return to user-home jsp page.
	recoverPassword Parameters <pre>ModelMap model, String userId, String mob, String email</pre> Return Type String	<ul style="list-style-type: none"> Accept user Id, email Id and mobile number as parameter and find corresponding user. If a valid user is obtained, then password-recovery-user jsp page else return forgot-password-user jsp page.
	verifySecretQuestion Parameters <pre>ModelMap model, @PathVariable long userId, String ans1, String ans2, String ans3</pre>	<ul style="list-style-type: none"> Accept answers of secret questions as parameter and verify user. If a valid user is obtained, then return set-new-pwd-user jsp page else return





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





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





	<p>@Validated @ModelAttribute Analyst analyst, BindingResult result, ModelMap model, @PathVariable int pageNumber, @Param String sortBy, @Param String sortDir</p> <p>Return Type</p> <p>String</p>	
	<p>getCategories()</p> <p>Parameters</p> <p>None</p> <p>Return Type</p> <p>Map<String, String></p>	<ul style="list-style-type: none">• Return complaint category list.
LoginController	<p>getName</p> <p>Parameters</p> <p>ModelMap model</p> <p>Return Type</p> <p>String</p>	<ul style="list-style-type: none">• Return the name of user logged in.
	<p>userLogin</p> <p>Parameters</p> <p>ModelMap model, String error, String logout</p> <p>Return Type</p> <p>String</p>	<ul style="list-style-type: none">• Return login jsp page.
	<p>Redirect</p> <p>Parameters</p> <p>ModelMap model</p> <p>Return Type</p> <p>String</p>	<ul style="list-style-type: none">• If id between 1000 and 2000 is received then redirect to admin-home.jsp page.• If id between 2000 and 3000 is received then redirect to analyst-home.jsp page.• If id between 3000 and 4000 is received then redirect to user-home.jsp page.• Else return admin-home.jsp page.
	<p>getCategory</p> <p>Parameters</p>	<ul style="list-style-type: none">• Return category list.





	<p>None</p> <p>Return Type</p> <p>Map<String, String></p>	
RegistrationController	<p>initBinder</p> <p>Parameters</p> <p>WebDataBinder binder</p> <p>Return Type</p> <p>void</p>	<ul style="list-style-type: none">Convert date into “ yyyy-MM-dd” format.
	<p>registerAnalyst</p> <p>Parameters</p> <p>@Validated @ModelAttribute Analyst analyst, BindingResult result, ModelMap model</p> <p>Return Type</p> <p>String</p>	<ul style="list-style-type: none">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.Else return role-selection.jsp page.
	<p>registerUser</p> <p>Parameters</p> <p>@Validated @ModelAttribute User user, BindingResult result, ModelMap model</p> <p>Return Type</p> <p>String</p>	<ul style="list-style-type: none">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.Else return role-selection.jsp page.
	<p>populateGender</p> <p>Parameters</p> <p>Return Type</p> <p>List<String></p>	<ul style="list-style-type: none">Return list.
	<p>getCategories</p> <p>Parameters</p> <p>Return Type</p> <p>Map<String, String></p>	<ul style="list-style-type: none">Return category list.





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

	populateGender	<ul style="list-style-type: none"> Return list.
	Parameters List<String>	
	Return Type Map<String, String>	<ul style="list-style-type: none"> Return category list.

5.5.2 Service Classes

Class	Method	Parameter Logic
AnalystService	getLastId() Parameters None Return Type long lstId	<ul style="list-style-type: none"> Call the getLastId() of AnalystService Check whether analystId greater than 2000 if it is greater then return analystId
	getAnalyst() Parameters long analystId Return Type User	<ul style="list-style-type: none"> Call the getAnalyst() of AnalystService by passing analystId it will return corresponding user





	<p>addAnalyst()</p> <p>Parameters</p> <p>Analyst analyst</p> <p>Return Type</p> <p>boolean</p>	<ul style="list-style-type: none">• Call the addAnalyst() of AnalystService by passing analyst• It will return true if analyst is added
	<p>getAllAnalystNotOfSupportLevel()</p> <p>Parameters</p> <p>String supportLevel</p> <p>Return Type</p> <p>List<Analyst></p>	<ul style="list-style-type: none">• Call the getAllAnalystNotOfSupportLevel() of AnalystService by passing analyst by passing supportLevel• It will return list of analyst.
	<p>getAllAnalyst()</p> <p>Parameters</p> <p>None</p> <p>Return Type</p> <p>List<Analyst></p>	<ul style="list-style-type: none">• Call the getAllAnalyst() of AnalystService. It will return the list of Analyst
	<p>deleteAnalyst()</p> <p>Parameters</p> <p>long AnalystId</p> <p>Return Type</p> <p>void</p>	<ul style="list-style-type: none">• Call the deleteAnalyst() of AnalystService by passing analystId parameter to delete the corresponding analyst





	deleteAnalyst() Parameters Analyst Return Type void	<ul style="list-style-type: none">Call the deleteAnalyst() of AnalystService by passing analyst parameter to delete the corresponding analyst
	updateAnalyst() Parameters Analyst Return Type Analyst	<ul style="list-style-type: none">Call the updateAnalyst() of AnalystService by passing Analyst to update the Analyst.
	getPasswordSHA() Parameters long analystId Return Type String	<ul style="list-style-type: none">Call the getPasswordSHA() of AnalystService by passing AnalystId to get the encrypted password.
	checkAnswer() Parameters List<analystSecretQuestion> String ans1 String ans2 String ans3 Return Type Boolean	<ul style="list-style-type: none">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.





	findAnalyst() Parameters long analystId String mob String email Return Type Analyst	<ul style="list-style-type: none">Call the findAnalyst() of AnalystService by passing AnalystId,mob,email to get the corresponding analyst
	getLastId() Parameters None Return Type long lstId	<ul style="list-style-type: none">Call the getLastId() of AnalystSecretQuestionServiceCheck whether analystSecretQuestion id greater than 8000 if it is greater then return Id
EmailAnalystService	getLastId() Parameters None Return Type long	<ul style="list-style-type: none">Call the getLastId () of AnalystSecretQuestionService
	addMail() Parameters EmailAnalyst email Return Type boolean	<ul style="list-style-type: none">Call the addEmail() of EmailAnalystService by passing emailIt will return true if mail is added.
	getEmailAnalyst() Parameters long emailld Return Type EmailAnalyst	<ul style="list-style-type: none">Call the getEmailAnalyst() of EmailAnalystService by passing emailldIt will return EmailAnalyst





EmailUserAnalystService	getLastId() Parameters None Return Type long lstId	<ul style="list-style-type: none">• Call the getLastId() of EmailUserAnalystService• Check whether email is not empty and email id greater than 12000 if it is greater then return Id•
	addEmail() Parameters EmailUserAnalyst email Return Type boolean	<ul style="list-style-type: none">• Call the addEmail() of EmailUserAnalystService by passing email• It will return true if mail is added.
	getMail() Parameters long emaiId Return Type EmailUserAnalyst	<ul style="list-style-type: none">• Call the getEmail() of EmailUserAnalystService by passing emaiId• It will return EmailUserAnalyst
FeedbackService	addFeedback() Parameters Feedback feedback Return Type boolean	<ul style="list-style-type: none">• Call the addFeedback() of FeedbackService by passing feedback• It will return true if feedback is added.
	getFeedback() Parameters long responsId Return Type	<ul style="list-style-type: none">• Call the getFeedback() of FeedbackService by passing response id• It will return feedback object





	Feedback	
	getAllFeedback() Parameters None Return Type List<Feedback>	<ul style="list-style-type: none">Call the getAllFeedback() of FeedbackService .It will return list of feedback objects.
	updateFeedback() Parameters Feedback feedback Return Type Boolean	<ul style="list-style-type: none">Call the updateFeedback() of FeedbackService by passing feedbackIt will return true if it is updated
	deleteFeedback() Parameters long responseld Return Type Void	<ul style="list-style-type: none">Call the deleteFeedback() of FeedbackService by passing response id
	deleteFeedback() Parameters Feedback feedback Return Type Void	<ul style="list-style-type: none">Call the deleteFeedback() of FeedbackService by passing feedback .
LoginDetailsService	getLoginDetailsByRegiste redId() Parameters long registeredId Return Type	<ul style="list-style-type: none">Call the getLoginDetailsByRegisteredId() of LoginDetailsService by passing registeredId.It will return loginDetails





	LoginDetails	
	addLoginDetails() Parameters LoginDetails loginDetails Return Type boolean	<ul style="list-style-type: none">Call the addLoginDetails() of LoginDetailsService by passing loginDetails.It will return true if it is added.
	modifyLoginDetails() Parameters LoginDetails loginDetails Return Type boolean	<ul style="list-style-type: none">Call the modifyLoginDetails() of LoginDetailsService by passing loginDetails.It will return true if it is updated.
	getQuestionById() Parameters long questionId Return Type SecretQuestions	<ul style="list-style-type: none">Call the getQuestionById() of SecretQuestionService by passing questionId.It will return SecretQuestions object.
	getAllQuestions() Parameters None Return Type List<SecretQuestion>	<ul style="list-style-type: none">Call the getQuestion() of SecretQuestionService it will return list of secret questions.
	getAllExcept() Parameters List<SecretQuestion> alreadyAdded Return Type	<ul style="list-style-type: none">Call the getAllExcept() of SecretQuestionService by using list of secret questions it will return list of secret questions.





	List<SecretQuestion>	
	getAllQuestionDescription() Parameters None Return Type List<String>	<ul style="list-style-type: none">Call the getAllQuestionDescription() of SecretQuestionService it will return list.
	getLastId() Parameters None Return Type long lstId	<ul style="list-style-type: none">Call the getLastId() of UserSecretQuestionServiceCheck whether user secret question id is greater than 7000 if it is greater then return question id.
	addSecretQuestion() Parameters UserSecretQuestion question Return Type boolean	<ul style="list-style-type: none">Call the addSecretQuestion() of UserSecretQuestionService by passing question.it will return true if it is added.
	getLastId() Parameters None Return Type long	<ul style="list-style-type: none">Call the getLastId() of ComplaintService .it will return id.
	addComplaint() Parameters Complaint complaint Return Type	<ul style="list-style-type: none">Call the addComplaint() of ComplaintService by passing complaint.it will return true if complaint is added.





	Boolean	
	getComplaint() Parameters long complaintId Return Type Complaint	<ul style="list-style-type: none">Call the getComplaint() of ComplaintService by passing complaintId.it will return Complaint object.
	updateComplaint() Parameters Complaint complaint Return Type boolean	<ul style="list-style-type: none">Call the updateComplaint() of ComplaintService by passing complaint.it will return true if it is updated.
	deleteComplaint() Parameters long complaintId Return Type Boolean	<ul style="list-style-type: none">Call the deleteComplaint() of ComplaintService by passing complaintId .it will return true if deleted.
	getAllComplaintOfUser() Parameters long userId Return Type List<Complaint>	<ul style="list-style-type: none">Call the getAllComplaintOfUser() of ComplaintService by passing userId .it will return list of complaint.
	findByKeyword() Parameters String keyword Return Type List<Complaint>	<ul style="list-style-type: none">Call the findByKeyword() of ComplaintService by passing keyword .it will return list of complaint.





	<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<complaint></p>	<ul style="list-style-type: none">Call the getAllComplaintUser() of ComplaintUser by passing userId,pageNumber,count,sortBy,sortDir.
	<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<complaint></p>	<ul style="list-style-type: none">Call the getAllComplaint() of ComplaintUser by passing count,pageNumber,sortBy,sortDir. It will return page.
	<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<complaint></p>	<ul style="list-style-type: none">Call the getAllComplaintFiltered() of ComplaintUser by passing keyword,date,count,pageNumber,sortBy,sortDir. It will return page





	<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<complaint></p>	<ul style="list-style-type: none">Call the getAllComplaintForAnalyst() of ComplaintUser by passing analystId, count, pageNumber, sortBy, sort Dir. It will return page.
	<p>findDate()</p> <p>Parameters</p> <p>String date</p> <p>Return Type</p> <p>List<Complaint></p>	<ul style="list-style-type: none">Call the findDate() of ComplaintUser by passing date. It will return list of complaint in the date.
	<p>findByIdUserId()</p> <p>Parameters</p> <p>String userId</p> <p>Return Type</p> <p>List<Complaint></p>	<ul style="list-style-type: none">Call the findByIdUserId() of ComplaintUser by passing userId. It will return list of complaint of userId.
	<p>findByComplaintId()</p> <p>Parameters</p> <p>String complaintId</p> <p>Return Type</p> <p>List<Complaint></p>	<ul style="list-style-type: none">Call the findByComplaintId() of ComplaintUser by passing complaintId. It will return list of complaint .
	<p>getAdmin()</p> <p>Paramenters</p>	<ul style="list-style-type: none">Call the getAdmin() of AdminService by passing adminId





	<p>long adminId Return Type Admin</p> <p>getAdminPassword() Parameters long adminId Return Type String</p> <p>getAdminPassword() Parameters long adminId Return Type String</p> <p>getLastId() Parameters None Return Type long lstId</p> <p>getUser() Parameters long userId Return Type User</p> <p>getAllUsers() Parameters None Return Type List<user></p>	<ul style="list-style-type: none">To get the Admin having corresponding adminId <ul style="list-style-type: none">Call the getAdminPassword() of AdminService by passing adminId. <ul style="list-style-type: none">Call the getPasswordSHA() of AdminService by passing analystId. <ul style="list-style-type: none">Call the getLastId() of UserServiceCheck whether userId greater than 3000 if it is greater then return userId <ul style="list-style-type: none">Call the getUser() of UserService by passing userId it will return corresponding user <ul style="list-style-type: none">Call the getAllUsers() of UserServiceIt will return the list of users
--	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------





	<p>deleteUser()</p> <p>Parameters</p> <p>User user</p> <p>Return Type</p> <p>boolean</p>	<ul style="list-style-type: none">Call the deleteUser() of UserService by passing user parameter to delete the corresponding user
	<p>getPasswordSHA()</p> <p>Parameters</p> <p>long userId</p> <p>Return Type</p> <p>String</p>	<ul style="list-style-type: none">Call the getPasswordSHA() of UserService by passing userId to get the encrypted password.
	<p>findUser()</p> <p>Parameters</p> <p>long userId</p> <p>String mob</p> <p>String email</p> <p>Return Type</p> <p>User</p>	<ul style="list-style-type: none">Call the findUser() of UserService by passing userId,mob,email to get the corresponding user
	<p>checkAnswer()</p> <p>Parameters</p> <p>List<userSecretQuestion></p> <p>String ans1</p> <p>String ans2</p> <p>String ans3</p> <p>Return Type</p> <p>Boolean</p>	<ul style="list-style-type: none">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.
	<p>findUserId()</p> <p>Parameters</p>	<ul style="list-style-type: none">Call the findUserId() by passing email to get corresponding user.





	String email Return Type User	
	validateAnswer() Parameters List<userSecretQuestion> String ans1 String ans2 String ans3 Return Type boolean	<ul style="list-style-type: none">Call the validateAnswer() 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.
	updateUser() Parameters User user Return Type boolean	<ul style="list-style-type: none">Call the updateUser() of UserService by passing user to update the user.

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_DETAILS	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_QUESTIONS	A_SQ_ID	Column is used for storing Analyst secret question id	INT		YES	NOT NULL AUTO_INCREMENT



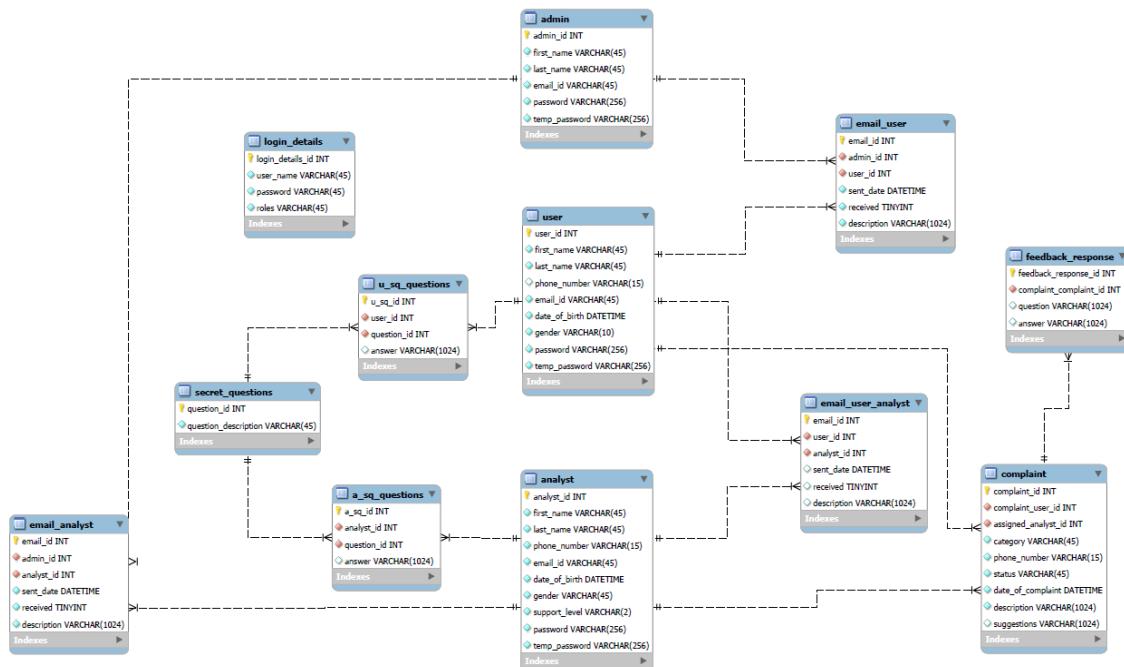


	ANALYST_ID	Column is used for storing Analyst id	INT			NOT NULL FOREIGN KEY (ANALYST.ANALYST_ID)
	QUESTION_ID	Column is used for storing Question id	INT			NOT NULL FOREIGN KEY (SECRET_QUESTION.QUESTION_ID)
	ANSWER	Column is used for storing Analyst secret question Answer	VARCHAR	1024		
EMAIL_USER_ANALYST	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	<i><First version></i>

