

# Software Design Document, Implementation and Testing

---

## Complaint Management System (CMS)

Prepared for: Public Works Department,  
Dhaka, Bangladesh.

By: Nextech Limited, Dhaka, Bangladesh.

# TABLE OF CONTENTS

- 1. Introduction ..... 1
- 2. Design considerations ..... 2
- 3. Architecture ..... 3
- 4. Data design .....12
- 5. Component design.....23
- 6. Software Interface design .....30
- 7. Class Diagram.....97
- 8. Implement and Testing .....99

# **1. Introduction**

The project is to create a unified Complaint Management System for all the departments of the Public Works Department. The system should be able to allocate Complaint Management to the authenticated users, keep track of changes, and be accessed only by the authorized personnel.

This design document presents the designs used or intended to be used in implementing the project. The designs described, follow the requirements specified in the Software Requirements Specifications document prepared for the project.

## **1.1. Purpose**

The purpose of this document is to present a detailed description of the designs of the Complaint Management System, created for the Public Works Department. Firstly, this document is intended for the programming group in Team 1, to use the designs as guidelines to implement the project. Equally, this document is also for the team's Leader, Mr.Mizanul Hoque Howlader, as it fulfils one of the requirements of the project. Lastly, this document could be used for designers who try to upgrade or modify the present design of the Complaint Management System.

## **1.2. Scope**

This document gives a detailed description of the software architecture of the Complaint Management system. It specifies the structure and design of some of the modules discussed in the SRS. It also displays some of the use cases that had transformed into sequential and activity diagrams. The class diagrams show how the programming team would implement the specific module.

### **1.3. Overview**

This document is written according to the standards for Software Design Documentation explained in “IEEE Recommended Practice for Software Design Documentation”.

Sections 3 – 5 contain discussions of the designs for the project with diagrams, section 6 shows samples of UI from the system, and section 7 contains the class diagrams. The appendices contain the setup and configuration needed for the CMS, a list of functions that are implemented in this version, and that are to be implemented in future version, and a list of tools and environment used in the entire project, along with the time contribution of team members. The appendices also include the test report and test cases.

## **2. Design considerations**

### **2.1. Assumptions**

The user of the Complaint Management system is aware of basic operations of a computer and web pages. The user also understands the standard terms used for operation.

### **2.2. Constraints**

The system is built accessible only through public works department’s website. The system is implemented using PHP (Laravel) and MySQL technologies.

### **2.3. System environment**

The web based unified Complaint Management system is designed to work on all operating systems. The system is accessible through any laptop and desktop, that is connected to the server. It is accessible at all times.

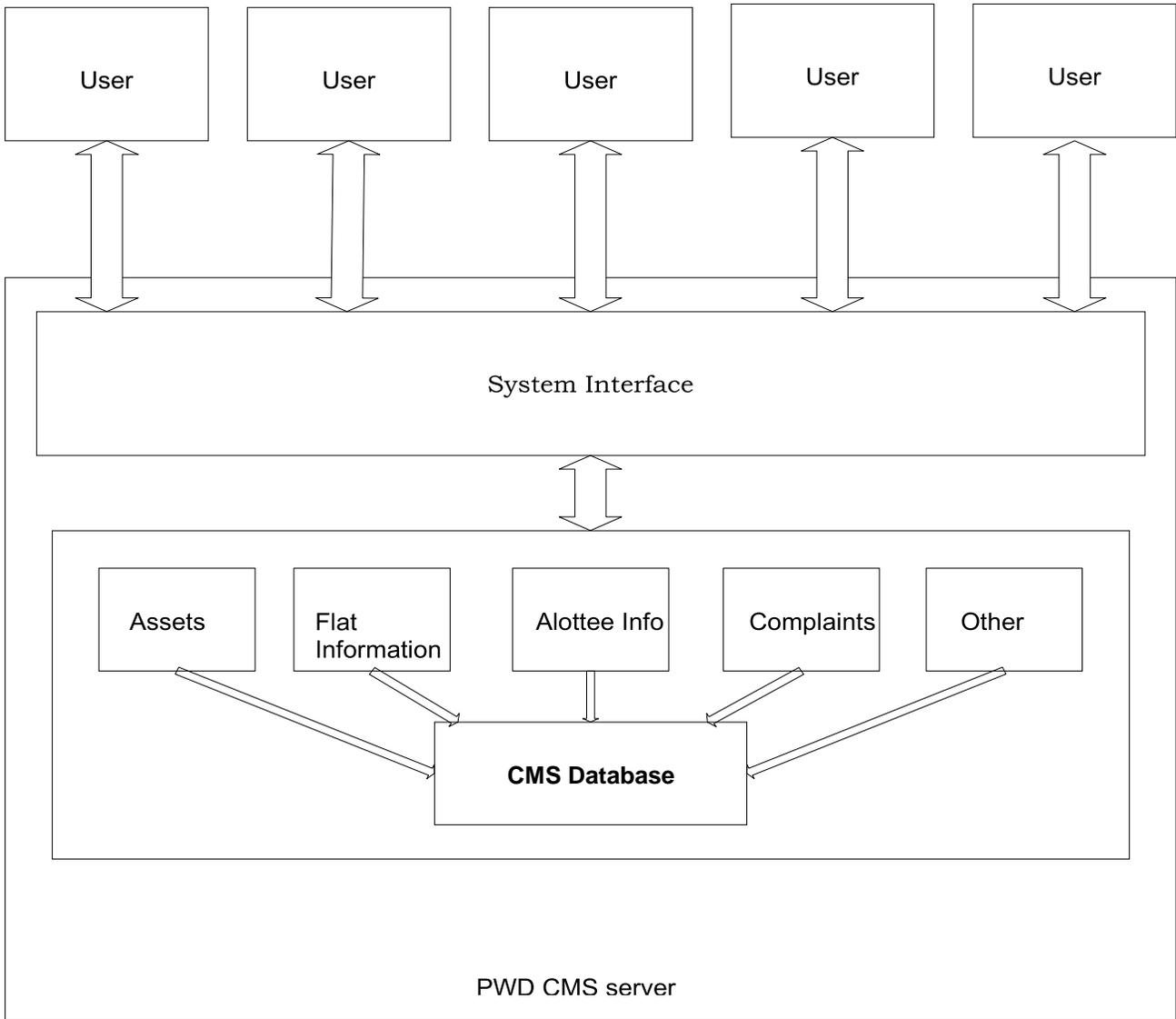
### **2.4. Design methodology**

The system is designed with flexibility for further development and/or modification. The system is divided into manageable processes that are grouped to sub-modules and modules that are built with abstraction.

## **3. Architecture**

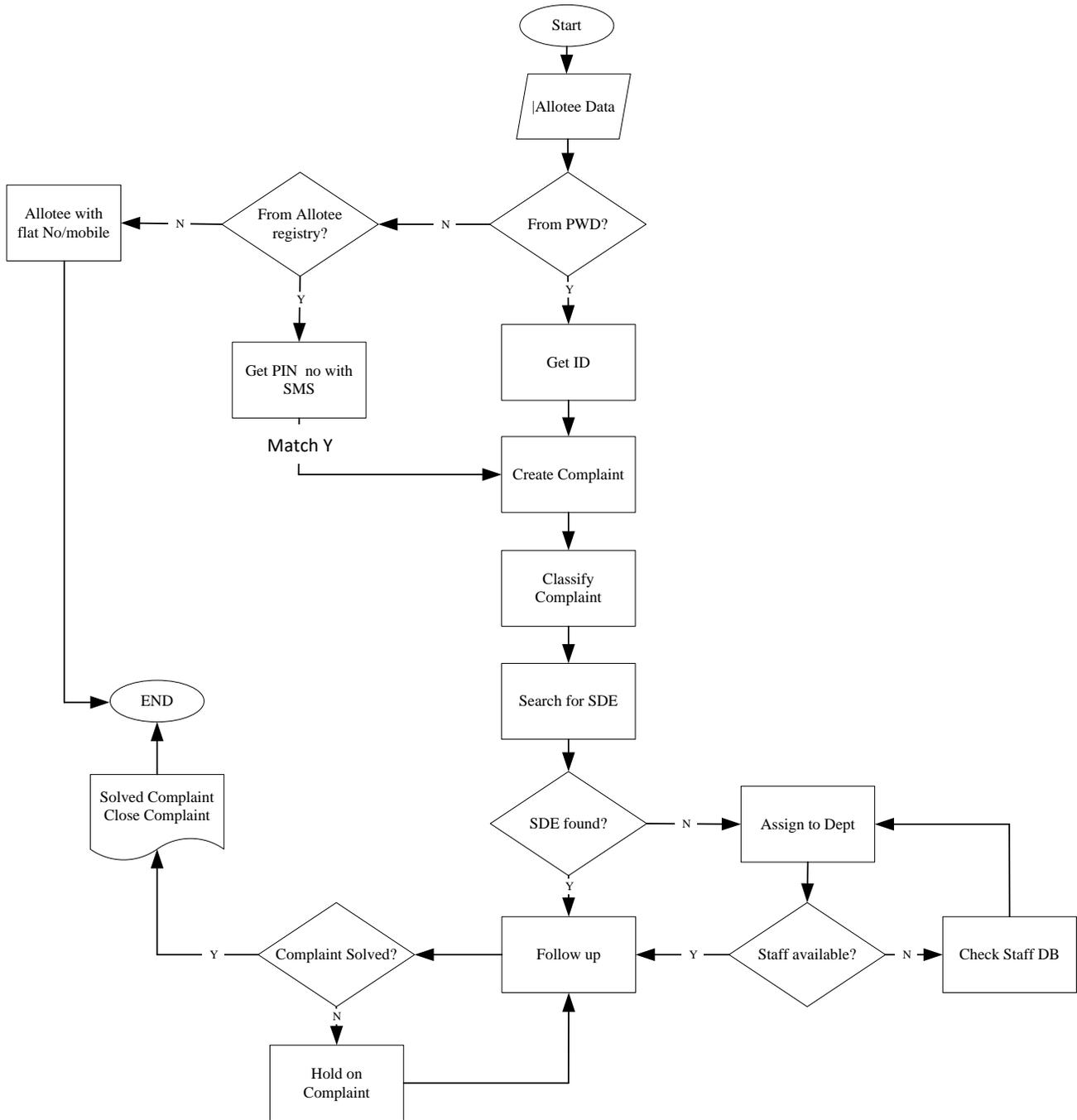
### **3.1. System design**

The block diagram below shows the principal parts of the system and their interactions.



**Figure 1.** Block diagram for CMS

The development of the proposed model is not only depending on how the system works. It also depends on the working flow process that being identified and need to be implemented and followed. The proposed complaint handling model is a method, platform or web-application to ensure that the complaint process is addressed and handled properly. The proposed Work Flow Process is shown in figure.



Proposed Working Flow Process

Figure shows the main processes in the working flow process of the proposed model. First, verifying the Allottee validity, then creating the Allottee complaint followed by a detailed processes to manage each complaint according it's classification priority and searching in the knowledgebase for related reference or similar case in order to find a quick handling procedure for finding a solution with the contribution of the responsible staff who will be dedicated to identify the rules on how to solve and get appropriate solution. In this paper the Nextech Ltd. tries to develop a model in terms of electronic complaints, which can support complaining from poor service quality and delivery. The advantage behind this model is the simple method for handling the Allottee's complaints about what dissatisfies them.

### 3.2 System Design

This model was designed to remove the main complaint handling department. In consequence, it made a small complaint handling section in each department. These small complaint handling sections provide complaint services for irritated Allottee's. Due to each organization has its own complaint handling procedure therefore each complaint section provides specific complaints handling method according to its department. [5]

The Nextech Ltd. disagrees with the previous model concerning the idea of removing the centralization of handling the complaints as this is against the idea behind e-CRM which is the coordination and common view of all the processes related to customer and proposed the below architecture for e- Complaint System based on SOA. The Nextech Ltd. enhanced the previous model by redesigning and centralizing the Complaint Management System, as shown in figure 4.

*The proposed model is divided into 3-tiers that consist of the following:*

- ***Database - tier***

It contains data about system users and their profiles, Allottee s information, available resources, and social association profiles.

- ***Business – tier***

It consists of the core of the system. i.e. complaint handling and feedback components.

- ***Presentation - tier***

It consists of web-based user interface.

Fig.4. Proposed Model for e-complaint handling system using SOA

The proposed system could be generally applicable for any web-based e-complaining system.

### 3.3 Implementation Design

After explaining the working flow process and the proposed model, the next step is the implementation of the

system as follows:

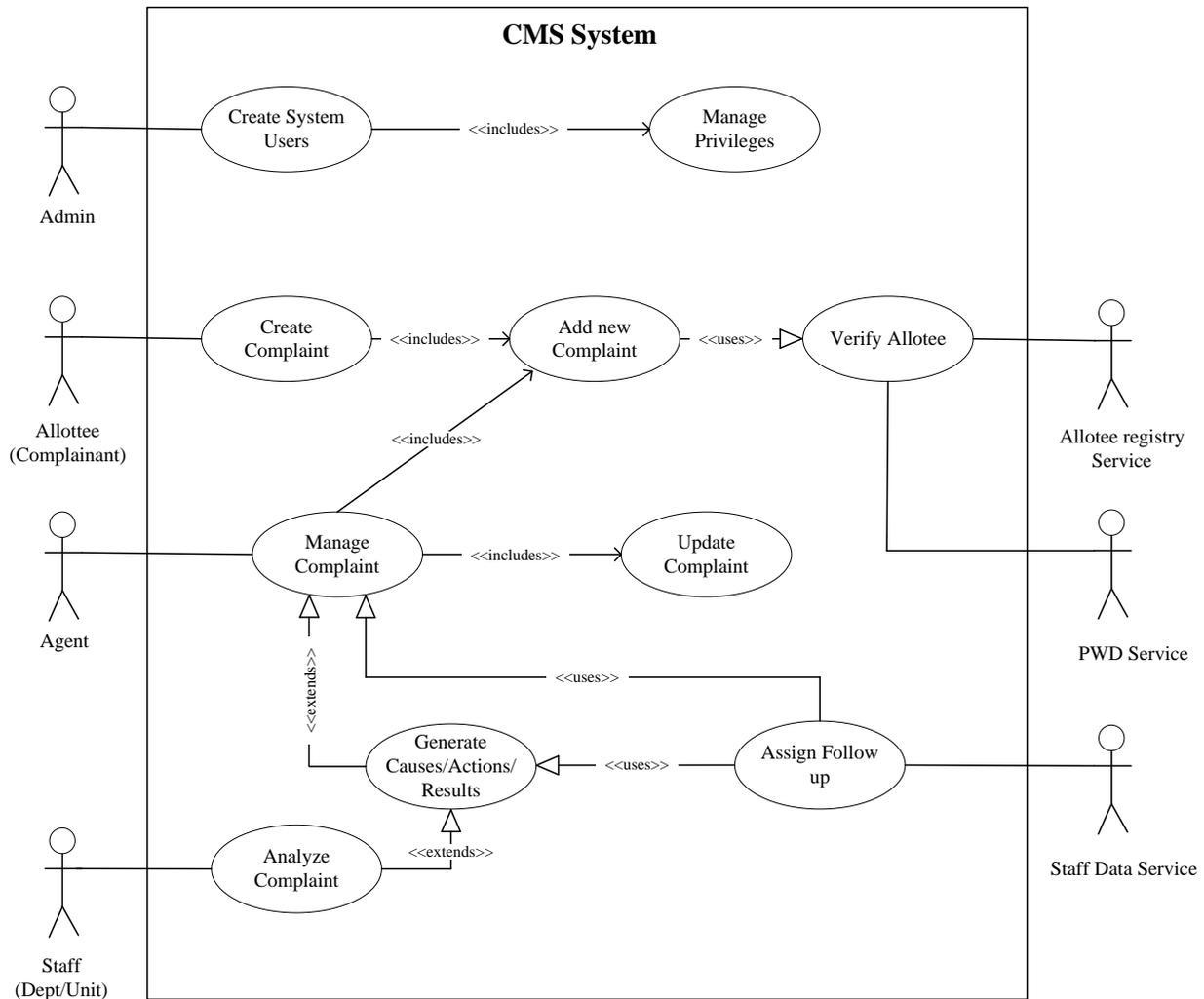
### **3.3.1 Use Case Diagram**

#### **3.3.1.1 ROLES**

- *Admin*- Create system users and manage their privileges,
- *Allottee* - Make Complaints against the provided services,
- *Agent*- Record the Complaint in the web-based then assigns follow up to available staff,
- *Staff*- Manage the Complaints causes and actions in the system then provide solutions,
- *Civil registry*- Has all Allottee s Name, Mobile, Flat... etc,
- *PWD*- Has Allottee s details that deserve services,
- *Staff Data*- Has all staff details who works on PWD and could be assigned to deal with Complaints.

#### **3.3.1.2 Scenario Overview**

An Allottee encounters a problem with the PWD; he enters the web and fills his new complaint in the complaint form, or he contacts the agent and explains his problem. Then the agent records the problem; opens the complaint system and creates the new complaint entry, also updates the complaints and enters information on each failure associated with the complaint. In a subsequent process the staff of the service department handles the complaints and delegates each complaint to the responsible one. Each department deal with the assigned complaints according to the management rules then analyzes actions and causes to get results. After that, propose a satisfying solution for the Allottee 's problem. If this solution is satisfactory for the Allottee then the statement of the complaint is closed. If not, the responsible department edits the original complaint and the process starts again. The new Complaint Management System is needed to improve these workflows and enable each department to contribute to the process of solving the Allottee complaints.



Proposed Use case Diagram

### 3.3.2 Sequence Diagram

For more understanding, the Nextech Ltd. summarizes the flow of the major functions of the system using the sequence diagram to show how processes operate one with another and in which order. The major functions of this system need to answer the following questions as: Which type of users deal with it? Who manage the system users and assign roles? Who make the complaint? Who deal with each complaint and according to which criteria? Who solve the complaint? Who follows up each complaint? Fig. shows the flow for the four main system users dealing with the system.

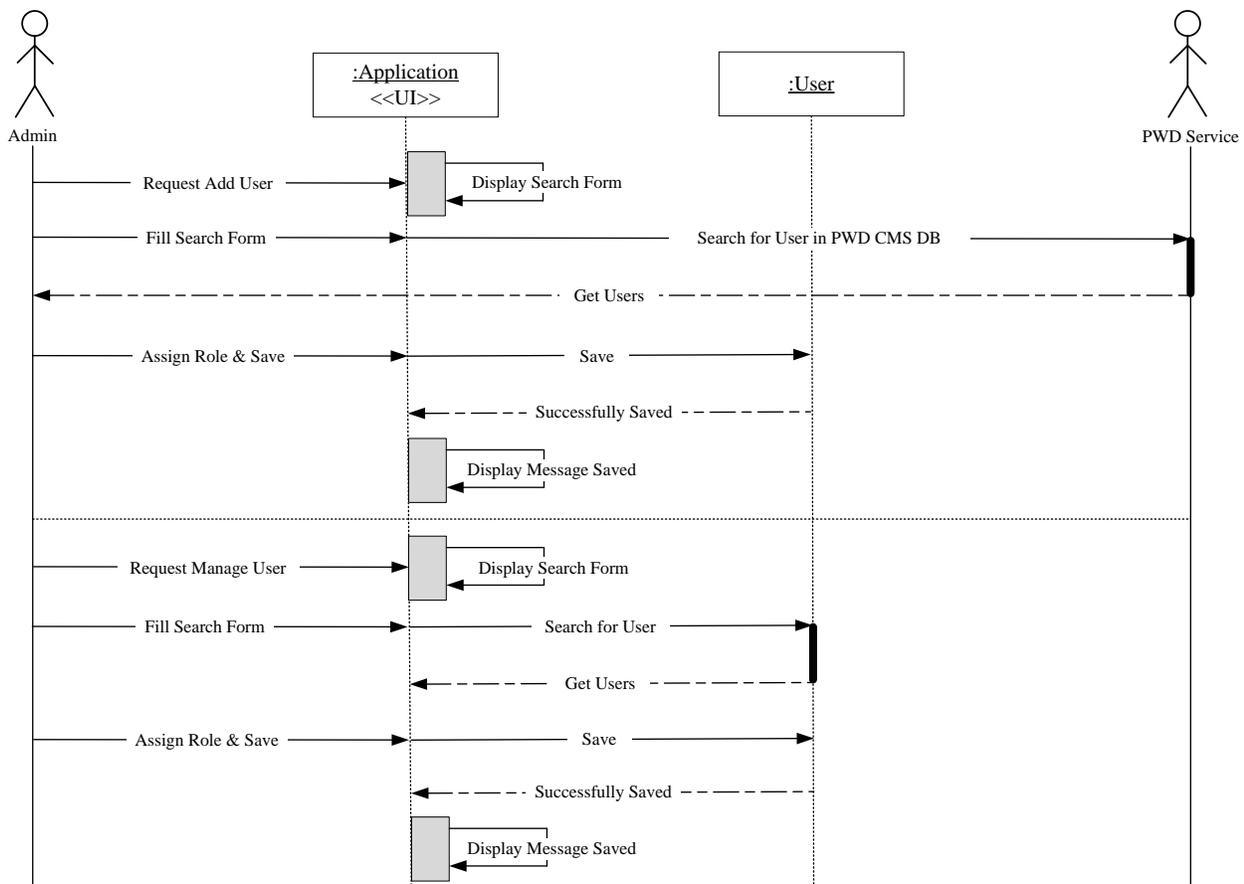
- *First user:* the administrator who adds new users or manages system privilege for whom exist on the system by assigning their roles.
- *Second user:* the Allottee (complainant) who makes complaint and check for existing complaint status.
- *Third user:* the agent who registers or manages Allottee Complaint each according to its classification and follows up through knowledge base in case of solution not found then forward the complaint case to the

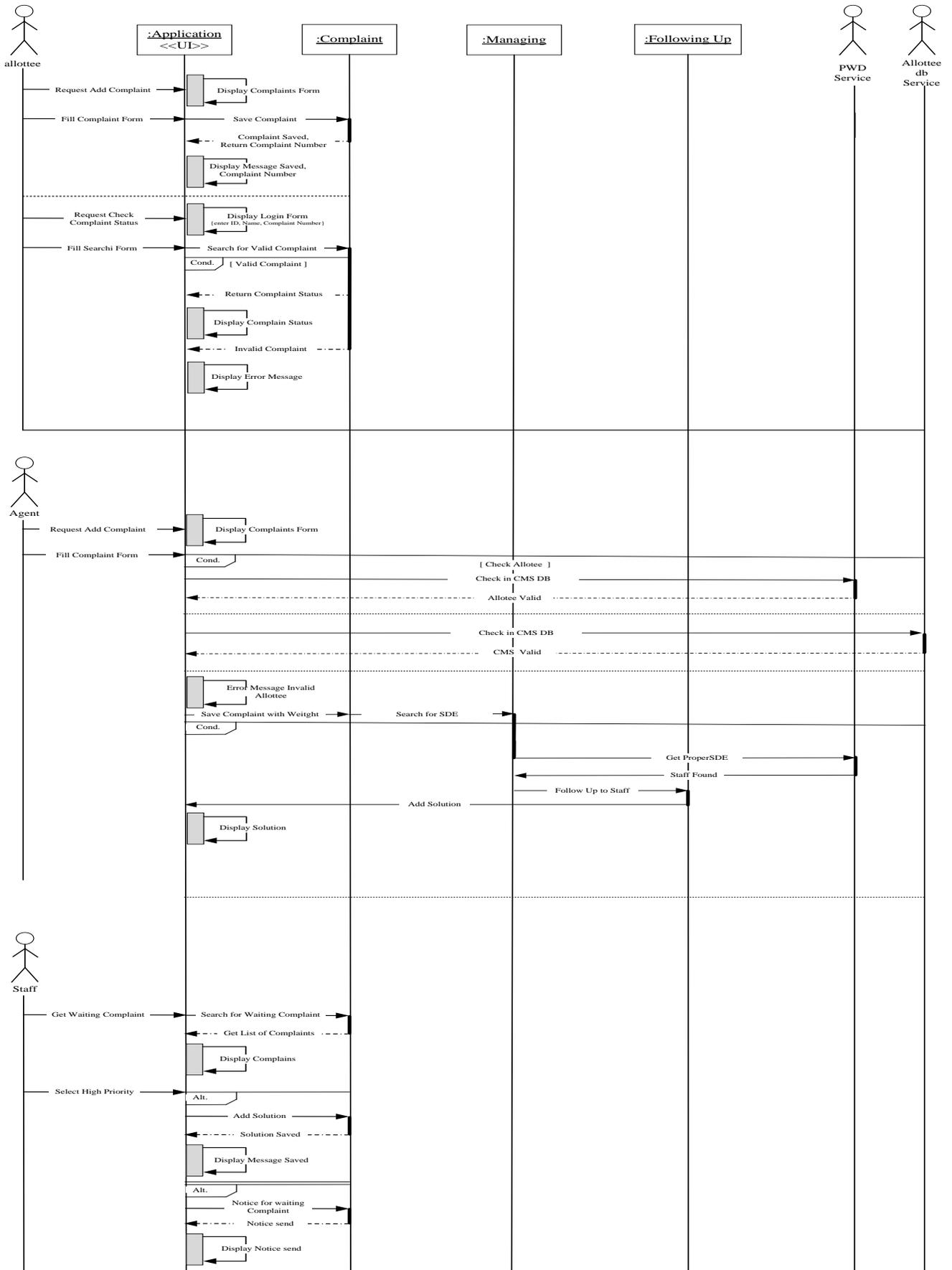
responsible staff of the service department to handle it.

- *Fourth user:* the responsible staff of the service department who deals with the waiting complaints by identifying and analyzing the causes and actions for each complaint then resolving it according to the management rules and propose a preventive action in order to ensure that this complaint will not be raised again. Then finally, back the complaint case to the agent in order to follow it up. After that, the agent updates the complaint status or closes the complaint case if it resolved.

Also there are two main services that integrate with the proposed system, as showed in fig.

- *First service:* the PWD service which contains the entire Allottee s' information that deserves the services each according to his region and Governorate, also it contains the entire staffs' information, each according to his department and role to deal with cases on the system.
- *Second service:* the Allottee db service which contains the entire Allottee's identity information, if they deserve services or not, also to validate the identity of the Allottee who make complaint in order to improve the provided service as general.

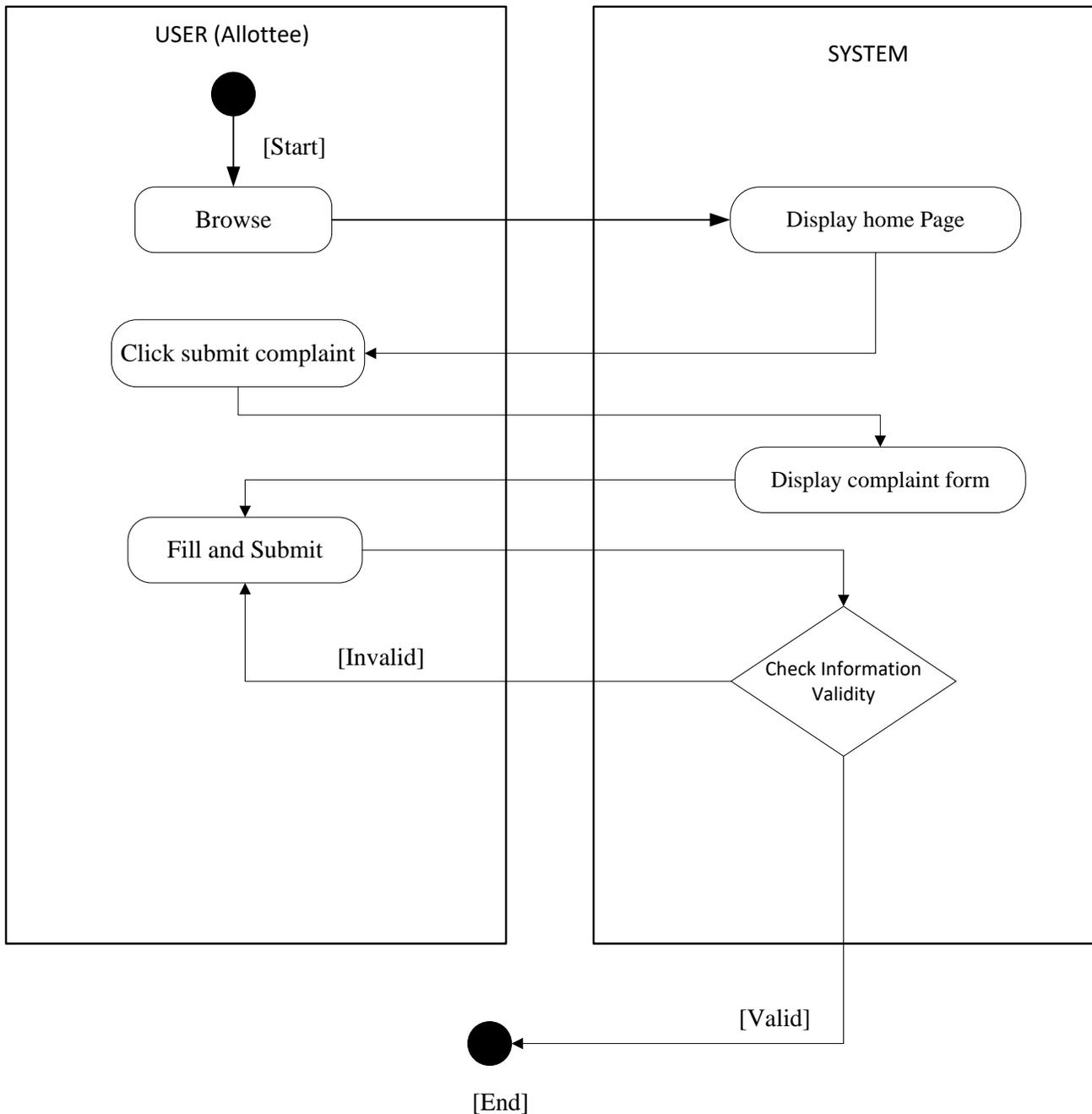


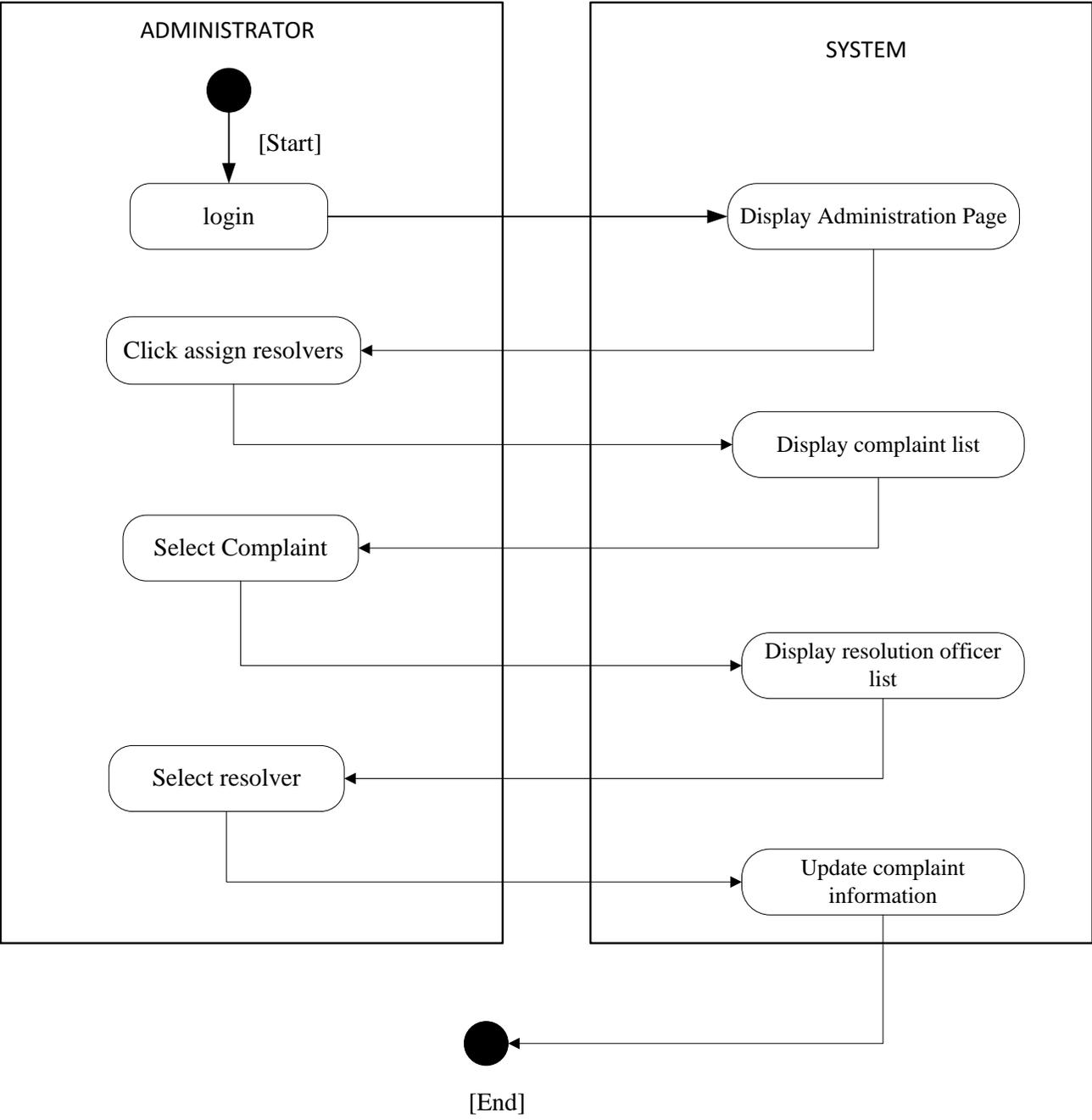


## Sequence Diagram for the proposed CMS

### 3.3.3 Activity Diagrams

Activity diagram is another important diagram in UML used to describe dynamic aspects of the system. An activity diagram visually presents a series of actions or flow of control in a system similar to a flowchart or a data flow diagram.





## 4. Data design

### 4.1. Data description

MySQL database and PHP (Laravel) to communicate with the database that is installed locally on the server.

### 4.2. Data dictionary

Data Dictionary

#### tbl\_allottee

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	<b>allottee_id</b>	int(11)			No	<i>None</i>		AUTO_INCREMENT
2	<b>allottee_name</b>	varchar(255)	latin1_swedish_ci		No	<i>None</i>		
3	<b>allottee_email</b>	varchar(100)	latin1_swedish_ci		No	<i>None</i>		
4	<b>allottee_mobile</b>	varchar(15)	latin1_swedish_ci		No	<i>None</i>		
5	<b>nid</b>	varchar(30)	latin1_swedish_ci		No	<i>None</i>		
6	<b>govt_department_id</b>	int(11)			No	<i>None</i>		
7	<b>designation</b>	varchar(30)	latin1_swedish_ci		No	<i>None</i>		
8	<b>permanent_address</b>	text	latin1_swedish_ci		No	<i>None</i>		
9	<b>division_id</b>	int(11)			No	<i>None</i>		
10	<b>district_id</b>	int(11)			No	<i>None</i>		
11	<b>thana_id</b>	int(11)			No	<i>None</i>		
12	<b>picture</b>	varchar(50)	latin1_swedish_ci		No	<i>None</i>		
13	<b>entry_by</b>	int(11)			Yes	<i>NUL L</i>		
14	<b>entry_date</b>	date			Yes	<i>NUL L</i>		
15	<b>update_by</b>	int(11)			Yes	<i>NUL L</i>		
16	<b>update_date</b>	date			Yes	<i>NUL L</i>		

#### *tbl\_apartment*

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	apartment_id	int(11)			No	None		AUTO_INCREMENT
2	apartment_no	varchar(200)	latin1_swedish_ci		No	None		
3	apartment_description	text	latin1_swedish_ci		No	None		
4	apartment_building_id	int(11)			No	None		
5	apartment_quarter_id	int(11)			No	None		
6	allottee_name	varchar(200)	latin1_swedish_ci		No	None		
7	allottee_designation	varchar(100)	latin1_swedish_ci		No	None		
8	allottee_office_loc	varchar(200)	latin1_swedish_ci		No	None		
9	allottee_mobile	varchar(100)	latin1_swedish_ci		No	None		
10	email	varchar(100)	latin1_swedish_ci		No	None		
11	entry_by	int(11)			No	None		
12	entry_date	datetime			No	None		
13	update_by	int(11)			No	None		
14	update_date	datetime			No	None		

[tbl\\_building](#)

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	building_id	int(11)			No	None		AUTO_INCREMENT
2	building_no	varchar(200)	latin1_swedish_ci		No	None		
3	building_name	varchar(100)	latin1_swedish_ci		No	None		
4	building_description	text	latin1_swedish_ci		No	None		
5	building_quarters_id	int(11)			No	None		
6	building_type	int(11)			No	None		
7	entry_by	int(11)			Yes	NULL		
8	entry_date	datetime			Yes	NULL		
9	update_by	int(11)			Yes	NULL		
10	update_date	datetime			Yes	NULL		

**tbl\_buildingtype**

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	buildingtype_id	int(11)			No	None		AUTO_INCREMENT
2	rank_id	int(11)			Yes	NULL		
3	buildingtype_name	varchar(50)	latin1_swedish_ci		No	None		
4	entry_by	int(11)			Yes	NULL		
5	entry_date	date			Yes	NULL		
6	update_by	int(11)			Yes	NULL		
7	update_date	date			Yes	NULL		

**tbl\_circle**

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	circle_id	int(11)			No	None		AUTO_INCREMENT
2	circle_name	varchar(100)	latin1_swedish_ci		No	None		
3	entry_by	int(11)			Yes	NULL		
4	entry_date	date			Yes	NULL		
5	update_by	int(11)			Yes	NULL		
6	update_date	date			Yes	NULL		

**tbl\_department**

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	department_id	int(10)		UNSIGNED	No	None		AUTO_INCREMENT
2	department_name	varchar(50)	utf8mb4_unicode_ci		No	None		
3	entry_by	int(10)		UNSIGNED	Yes	NULL		
4	entry_date	datetime			No	None		
5	update_by	int(10)		UNSIGNED	Yes	NULL		
6	update_date	datetime			No	None		
7	created_at	timestamp			Yes	NULL		
8	updated_at	timestamp			Yes	NULL		

*tbl\_designation*

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	<b>designation_id</b>	int(10)		UNSIGNED	No	<i>None</i>		AUTO_INCREMENT
2	<b>designation_name</b>	varchar(50)	utf8mb4_unicode_ci		No	<i>None</i>		
3	<b>entry_by</b>	int(10)		UNSIGNED	Yes	<i>NULL</i>		
4	<b>entry_date</b>	datetime			Yes	<i>NULL</i>		
5	<b>update_by</b>	int(10)		UNSIGNED	Yes	<i>NULL</i>		
6	<b>update_date</b>	datetime			Yes	<i>NULL</i>		
7	<b>created_at</b>	timestamp			Yes	<i>NULL</i>		
8	<b>updated_at</b>	timestamp			Yes	<i>NULL</i>		

*tbl\_district*

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	<b>district_id</b>	int(11)			No	<i>None</i>		AUTO_INCREMENT
2	<b>district_name</b>	varchar(150)	latin1_swedish_ci		No	<i>None</i>		
3	<b>district_geo</b>	varchar(11)	latin1_swedish_ci		No	<i>None</i>		
4	<b>division_geo_code</b>	int(11)			No	<i>None</i>		
5	<b>entry_by</b>	varchar(150)	latin1_swedish_ci		Yes	<i>NULL</i>		
6	<b>entry_date</b>	date			Yes	<i>NULL</i>		
7	<b>update_by</b>	varchar(150)	latin1_swedish_ci		Yes	<i>NULL</i>		
8	<b>update_date</b>	date			Yes	<i>NULL</i>		

*tbl\_division*

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	division_id	int(11)			No	None		
2	division_name	varchar(150)	latin1_swedish_ci		No	None		
3	division_circle_id	int(11)			No	None		
4	entry_by	int(11)			Yes	NULL		
5	entry_date	date			Yes	NULL		
6	update_by	int(11)			Yes	NULL		
7	update_date	date			Yes	NULL		

*tbl\_division\_geo*

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	division_id	int(11)			No	None		AUTO_INCREMENT
2	division_name	varchar(150)	latin1_swedish_ci		No	None		
3	division_geo_code	varchar(5)	latin1_swedish_ci		No	None		
4	entry_by	int(11)			Yes	NULL		
5	entry_date	date			Yes	NULL		
6	update_by	int(11)			Yes	NULL		
7	update_date	date			Yes	NULL		

*tbl\_govt\_department*

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	govt_department_id	int(11)			No	None		AUTO_INCREMENT
2	govt_department_name	varchar(150)	latin1_swedish_ci		No	None		
3	entry_by	int(11)			Yes	NULL		
4	entry_date	date			Yes	NULL		
5	update_by	int(11)			Yes	NULL		
6	update_date	date			Yes	NULL		

*tbl\_problemtypes*

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	prob_type_id	int(10)		UNSIGNED	No	None		AUTO_INCREMENT

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
2	prob_type_desc	varchar(50)	utf8mb4_unicode_ci		No	None		
3	pro_type_maxday	int(10)		UNSIGNED	Yes	NULL		
4	prob_type_department	int(10)		UNSIGNED	Yes	NULL		
5	entry_by	int(10)		UNSIGNED	Yes	NULL		
6	entry_date	datetime			Yes	NULL		
7	update_by	int(10)		UNSIGNED	Yes	NULL		
8	update_date	datetime			Yes	NULL		
9	created_at	timestamp			Yes	NULL		
10	updated_at	timestamp			Yes	NULL		

### *tbl\_quarters*

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	quarters_id	int(11)			No	None		AUTO_INCREMENT
2	quarters_name	varchar(250)	latin1_swedish_ci		No	None		
3	quarters_sup_cv	int(11)			No	None		
4	quarters_sup_cv_mo	varchar(50)	latin1_swedish_ci		No	None		
5	quarters_sup_em	int(11)			No	None		
6	quarters_sup_em_mo	varchar(50)	latin1_swedish_ci		No	None		
7	quarters_ee	int(11)			No	None		
8	quarters_ee_mob	varchar(50)	latin1_swedish_ci		No	None		
9	quarters_ee_em	int(11)			No	None		
10	quarters_ee_em_mob	varchar(50)	latin1_swedish_ci		No	None		
11	quarters_se	int(11)			No	None		
12	quarters_se_mob	varchar(50)	latin1_swedish		N	None		

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
		0)	h_ci		o			
13	quarters_se_em	int(11)			No	None		
14	quarters_se_em_mob	varchar(50)	latin1_swedish_ci		No	None		
15	circle_id	int(11)			No	None		
16	division_id	int(11)			No	None		
17	subdiv_id	int(11)			No	None		
18	quarters_district_id	int(11)			No	None		
19	entry_by	int(11)			Yes	NULL		
20	entry_date	date			Yes	NULL		
21	update_by	int(11)			Yes	NULL		
22	update_date	date			Yes	NULL		

*tbl\_quarter\_supervisor*

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	supervisor_id	int(11)			No	None		AUTO_INCREMENT
2	quarter_id	int(11)			No	None		
3	department_id	int(11)			No	None		
4	designation_id	int(11)			No	None		
5	user_id	int(11)			No	None		
6	entry_by	int(11)			Yes	NULL		
7	entry_date	date			Yes	NULL		
8	update_by	int(11)			Yes	NULL		
9	update_date	date			Yes	NULL		

*tbl\_role*

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	role_id	int(10)		UNSIGNED	No	None		
2	role_name	varchar(50)	utf8mb4_unicode_ci		No	None		

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
3	entry_by	int(10)		UNSIGNED	Yes	NULL		
4	entry_date	datetime			No	None		
5	update_by	int(10)		UNSIGNED	Yes	NULL		
6	update_date	datetime			No	None		

### *tbl\_sms\_send*

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id	int(11)			No	None		AUTO_INCREMENT
2	number	varchar(150)	latin1_swedish_ci		No	None		
3	description	text	latin1_swedish_ci		No	None		
4	entry_date	date			No	None		
5	enrty_datetime	datetime			No	None		
6	type	int(11)			No	None		
7	code	varchar(100)	latin1_swedish_ci		No	None		
8	status	int(11)			No	None		

### *tbl\_status*

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	status_id	int(10)		UNSIGNED	No	None		AUTO_INCREMENT
2	status_name	varchar(50)	utf8mb4_unicode_ci		No	None		
3	entry_by	int(10)		UNSIGNED	Yes	NULL		
4	entry_date	datetime			Yes	NULL		
5	update_by	int(10)		UNSIGNED	Yes	NULL		
6	update_date	datetime			Yes	NULL		

### *tbl\_subdiv*

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	subdiv_id	int(11)			No	None		AUTO_INCREMENT
2	subdiv_name	text	latin1_swedish_ci		No	None		

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
			ci					
3	subdiv_division_id	int(11)			No	None		
4	subdiv_circle_id	int(11)			No	None		
5	enrty_by	int(11)			Yes	NULL		
6	entry_date	datetime			Yes	NULL		
7	update_by	int(11)			Yes	NULL		
8	update_date	date			Yes	NULL		

*tbl\_survey\_user*

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	survey_user_id	int(11)			No	None		AUTO_INCREMENT
2	survey_user_name	varchar(50)	latin1_swedish_ci		No	None		
3	survey_pass	varchar(50)	latin1_swedish_ci		No	None		

*tbl\_task*

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	task_id	int(11)			No	None		AUTO_INCREMENT
2	circle_id	int(11)			No	None		
3	task_no	varchar(150)	latin1_swedish_ci		No	None		
4	division_id	int(11)			No	None		
5	subdiv_id	int(11)			No	None		
6	task_district	int(11)			No	None		
7	task_quaters	int(11)			No	None		
8	task_building	int(11)			No	None		
9	task_apartment	int(11)			No	None		
10	task_buildingtype	int(11)			No	None		
11	task_user_comments	text	utf8_general_ci		No	None		
12	task_name	varchar(150)	latin1_swedish_ci		No	None		
13	task_mobile	varchar(100)	latin1_swedish_ci		No	None		
14	task_status	int(11)			No	None		
15	task_admin_commen	text	utf8_general		No	None		

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
	<b>ts</b>		_ci					
16	<b>task_status_em</b>	int(11)			No	None		
17	<b>task_admin_comments_em</b>	text	utf8_general_ci		No	None		
18	<b>dpt_civil</b>	int(11)			No	None		
19	<b>dpt_em</b>	int(11)			No	None		
20	<b>task_update_date</b>	date			No	None		
21	<b>task_date</b>	date			No	None		
22	<b>task_update_by</b>	int(11)			No	None		

### *tbl\_task\_problem\_type*

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	<b>task_problem_type_id</b>	int(11)			No	None		AUTO_INCREMENT
2	<b>task_type_id</b>	int(11)			No	None		
3	<b>task_desc</b>	varchar(200)	latin1_swedish_ci		No	None		
4	<b>task_id</b>	int(11)			No	None		
5	<b>department_id</b>	int(11)			No	None		
6	<b>status_id</b>	int(11)			No	None		
7	<b>task_comments</b>	varchar(300)	latin1_swedish_ci		No	None		
8	<b>allocation_for</b>	int(11)			No	None		
9	<b>entry_date</b>	datetime			Yes	NULL		
10	<b>update_by</b>	int(11)			Yes	NULL		
11	<b>update_date</b>	datetime			Yes	NULL		

### *tbl\_users\_history*

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	<b>users_his_id</b>	int(11)			No	None		AUTO_INCREMENT
2	<b>first_name</b>	varchar(50)	latin1_swedish_ci		No	None		
3	<b>user_last_name</b>	varchar(50)	latin1_swedish_ci		No	None		
4	<b>email</b>	varchar(50)	latin1_swedish_ci		No	None		
5	<b>password</b>	varchar(20)	latin1_swedish_ci		No	None		

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
		)	ci					
6	<b>user_phnoe</b>	varchar(15)	latin1_swedish_ci		No	None		
7	<b>entry_by</b>	int(11)			No	None		
8	<b>entry_date</b>	date			No	None		

### *tbl\_ustatus*

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	<b>ustatus_id</b>	int(11)			No	None		
2	<b>ustatus_name</b>	varchar(50)	latin1_swedish_ci		No	None		

### *tree\_entries*

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	<b>id</b>	int(10)		UNSIGNED	No	None		
2	<b>pid</b>	int(10)		UNSIGNED	Yes	NULL		
3	<b>nodeName</b>	varchar(50)	utf8mb4_unicode_ci		No	None		
4	<b>url</b>	varchar(50)	utf8mb4_unicode_ci		No	None		
5	<b>icon</b>	varchar(50)	utf8mb4_unicode_ci		No	None		
6	<b>view_status</b>	varchar(5)	utf8mb4_unicode_ci		No	None		
7	<b>created_at</b>	timestamp			Yes	NULL		
8	<b>updated_at</b>	timestamp			Yes	NULL		

### *Users*

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	<b>id</b>	int(11)		UNSIGNED	No	None		AUTO_INCREMENT
2	<b>name</b>	varchar(150)	latin1_swedish_ci		No	None		
3	<b>user_last_name</b>	varchar(150)	latin1_swedish_ci		No	None		
4	<b>user_phone</b>	varchar(100)	latin1_swedish_ci		No	None		
5	<b>email</b>	varchar(150)	latin1_swedish_ci		No	None		
6	<b>user_name</b>	varchar(100)	latin1_swedish_ci		No	None		
7	<b>password</b>	varchar(200)	latin1_swedish_ci		No	None		
8	<b>user_role</b>	int(50)			No	None		
9	<b>user_status</b>	varchar(50)	latin1_swedish_ci		No	None		

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
10	<b>user_zone</b>	int(11)			No	<i>None</i>		
11	<b>user_circle_id</b>	int(11)			No	<i>None</i>		
12	<b>user_division_id</b>	int(11)			No	<i>None</i>		
13	<b>user_subdivision_id</b>	int(11)			No	<i>None</i>		
14	<b>user_department</b>	int(11)			No	<i>None</i>		
15	<b>user_designation</b>	int(11)			No	<i>None</i>		
16	<b>remember_token</b>	varchar(100)	latin1_swedish_ci		Yes	<i>NULL</i>		
17	<b>picture</b>	varchar(7)	latin1_swedish_ci		No	<i>None</i>		
18	<b>user_entryby</b>	varchar(100)	latin1_swedish_ci		Yes	<i>NULL</i>		
19	<b>user_entrydate</b>	date			Yes	<i>NULL</i>		
20	<b>user_updateby</b>	varchar(100)	latin1_swedish_ci		Yes	<i>NULL</i>		
21	<b>user_updatedate</b>	date			Yes	<i>NULL</i>		

### *user\_permission*

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	<b>perm_id</b>	int(10)		UNSIGNED	No	<i>None</i>		
2	<b>user_id</b>	int(10)		UNSIGNED	Yes	<i>NULL</i>		
3	<b>id</b>	int(10)		UNSIGNED	Yes	<i>NULL</i>		
4	<b>pid</b>	int(10)		UNSIGNED	Yes	<i>NULL</i>		
5	<b>created_at</b>	timestamp			Yes	<i>NULL</i>		
6	<b>updated_at</b>	timestamp			Yes	<i>NULL</i>		

### *\_nisl\_tree\_entries*

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	<b>id</b>	int(11)			No	0		
2	<b>pid</b>	int(11)			No	0		
3	<b>nodeName</b>	varchar(50)	utf8_general_ci		No			
4	<b>url</b>	varchar(200)	utf8_general_ci		No			
5	<b>icon</b>	varchar(100)	utf8_general_ci		No	<i>None</i>		
6	<b>view_status</b>	varchar(5)	utf8_general_ci		No			
7	<b>serials</b>	int(11)			No	<i>None</i>		

### *\_nisl\_user\_permission*

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
---	------	------	-----------	------------	------	---------	----------	-------

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	<b>perm_id</b>	int(11)			No	<i>None</i>		AUTO_INCREMENT
2	<b>user_role</b>	int(11)			No	<i>None</i>		
3	<b>id</b>	int(11)			No	<i>None</i>		
4	<b>pid</b>	int(11)			No	0		

## 5. Component design

“Users with appropriate permission” in the diagram refers to the users who are given exemptions or/and users of a particular level. Precise permissions are listed in Software requirements specification (SRS).

For diagrams with multiple functions, the design is the same for those functions, except for parameters/methods/classes.

### 5.1. Login

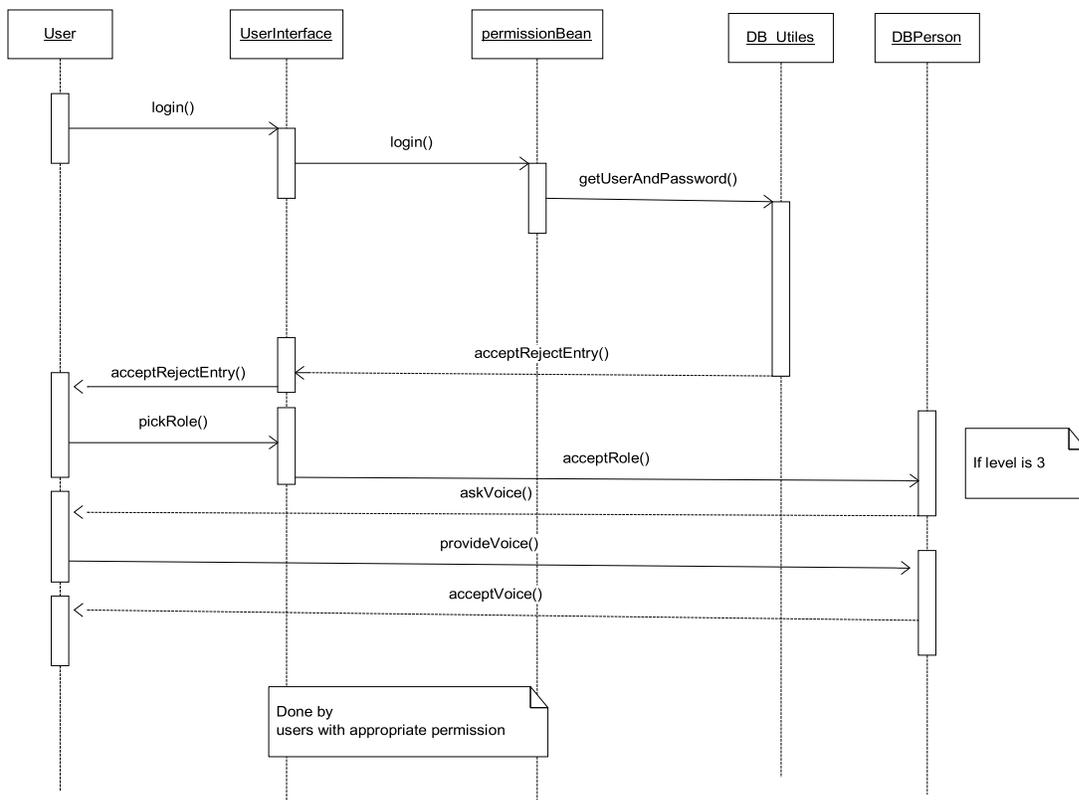


Figure 4. Sequence diagram for Login

5.2. Select language

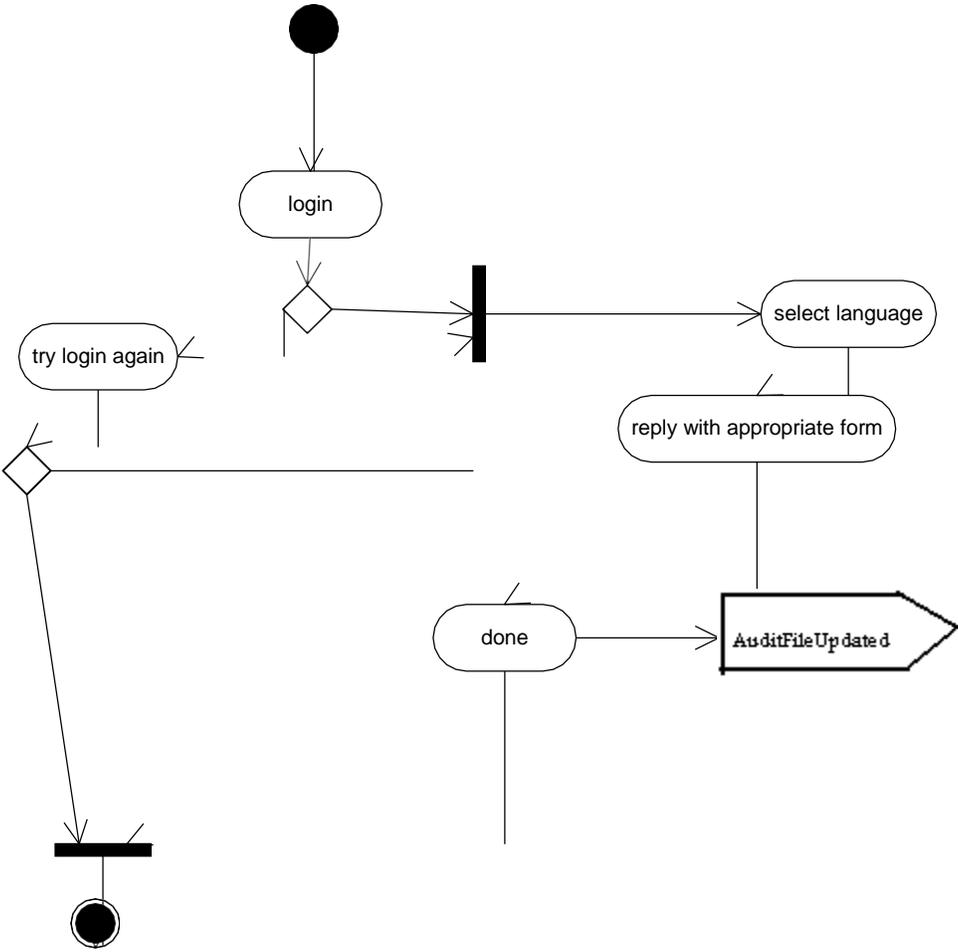
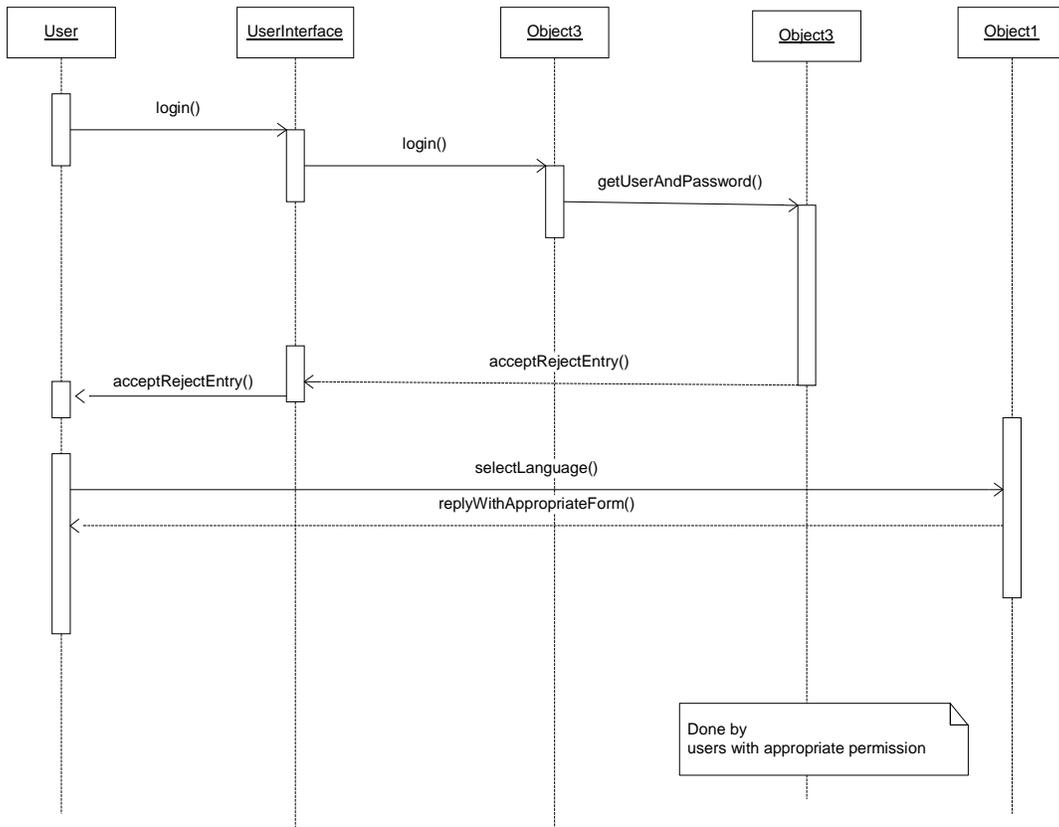
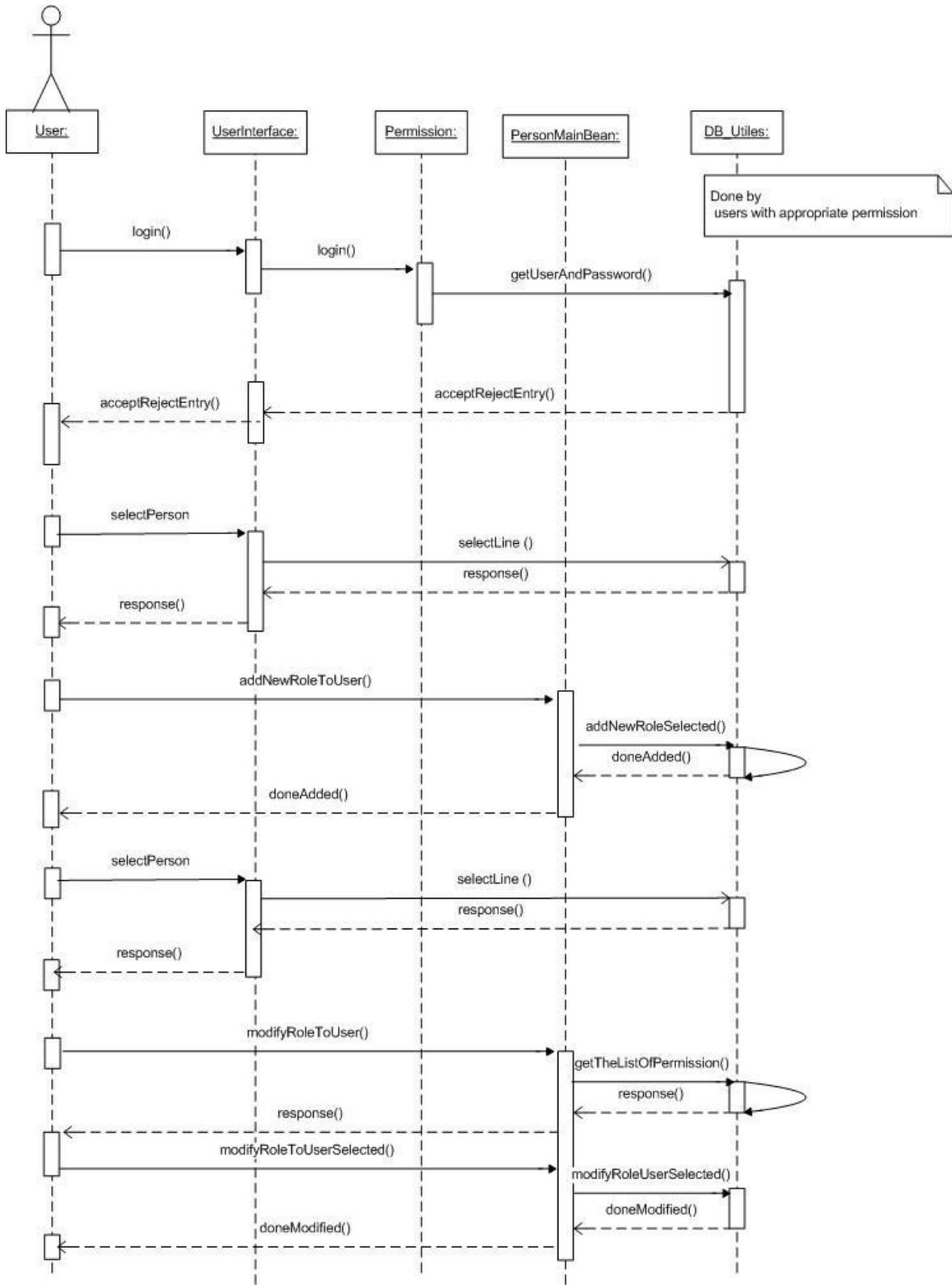


Figure 7. Activity diagram for Select language



**Figure 8.** Sequence diagram for Select language





**Figure 15.** Sequence diagram for Assign / Modify - role

5.4. Report

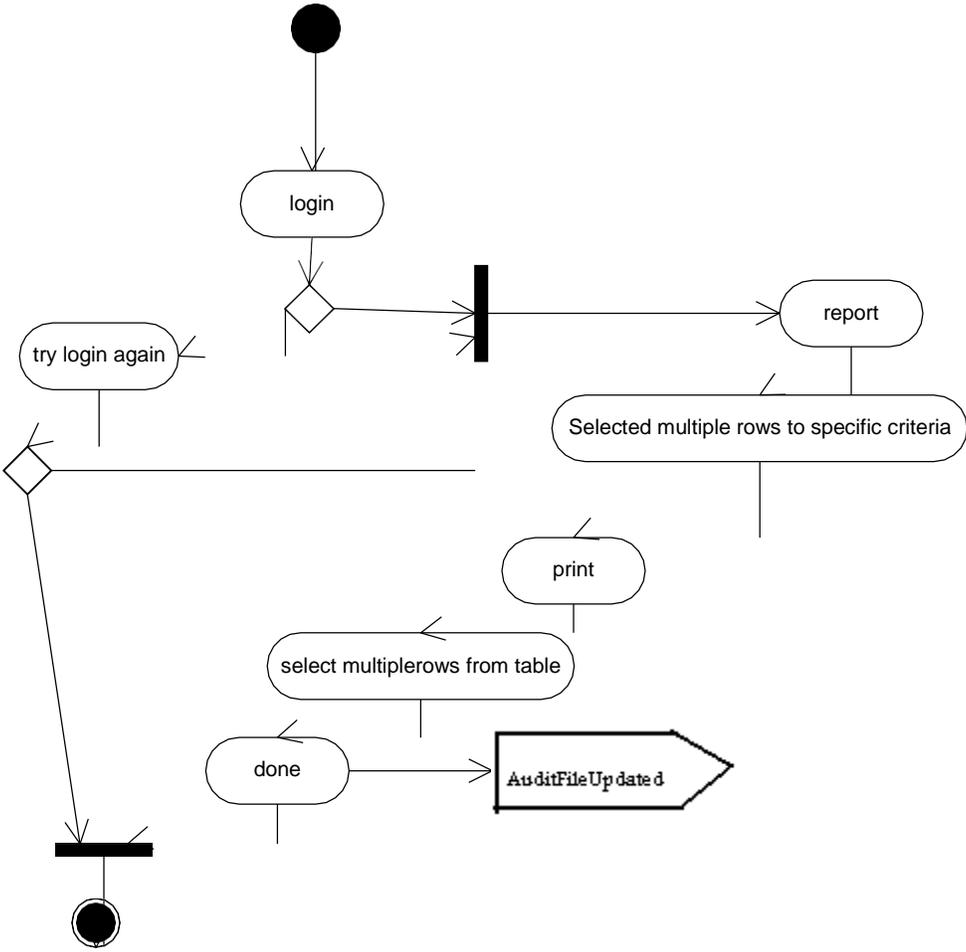
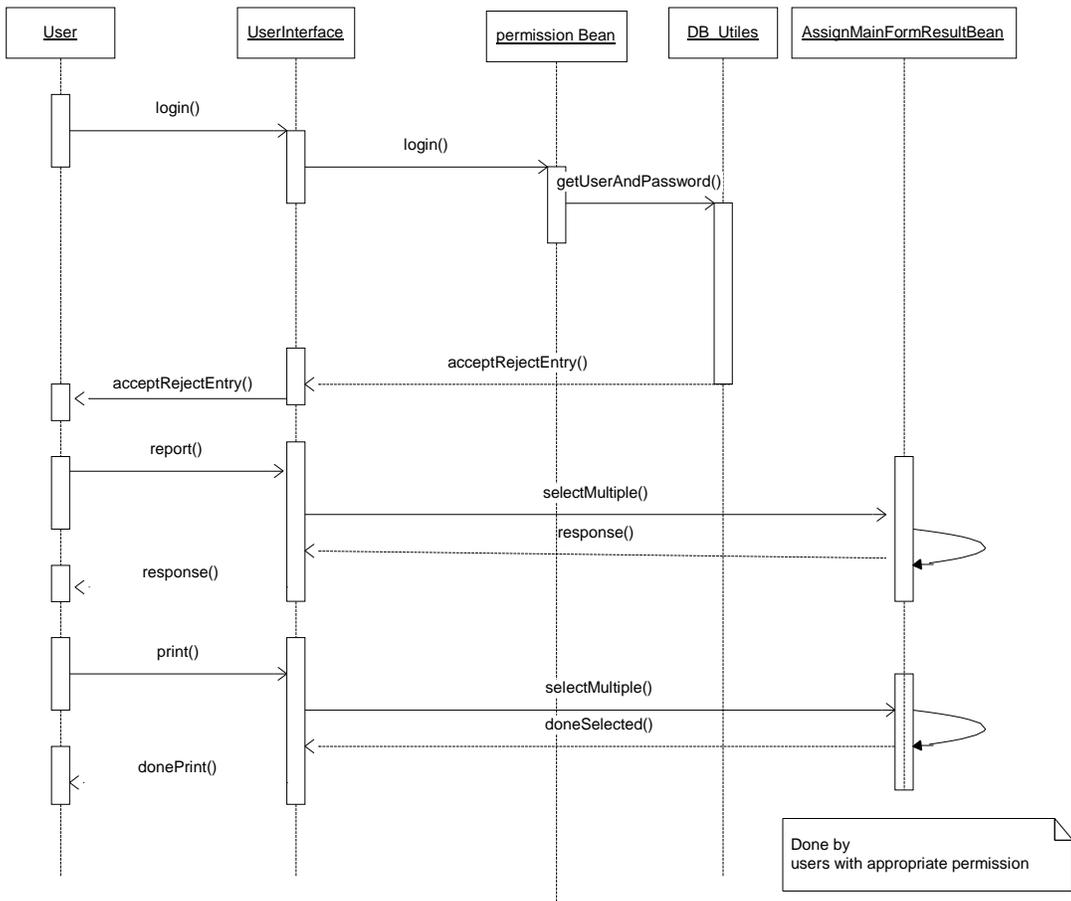


Figure 27. Activity diagram for Report



**Figure 28.** Sequence diagram for Report

## 6. Software interface design

### 6.1. User interface design

UI is designed according to UI design principles.

**The structure principle:** UI is organized in such a way that related things are combined together and unrelated things are separated.

**The simplicity principle:** It is easy to follow the provided interface. In the case of mistake, system displays error message.

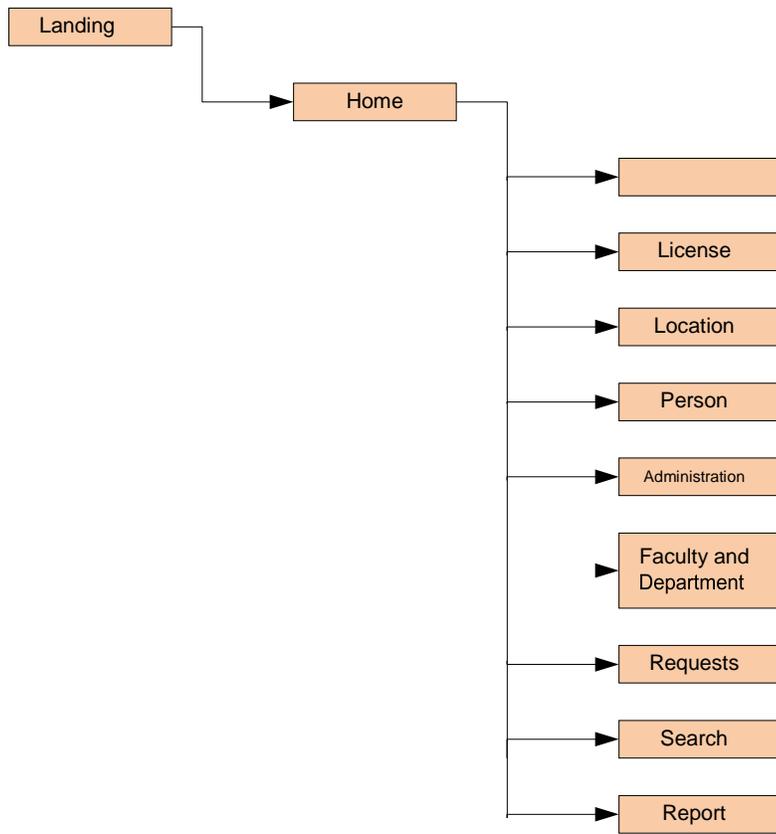
**The visibility principle:** All system's functions are available through UI. It does not overwhelm users with too many alternatives.

**The feedback principle:** Through the system of messages, the design keeps users informed of actions, errors, or exceptions.

**The reuse principle:** In design, same names were used to perform the same operations with different objects in order to reduce ambiguity.

#### 6.1.1. Web pages in a tree

The system's web pages are presented in a tree in **Error! Reference source not found.**<sup>31</sup>. From "Landing" page user can reach "Main/Admin" page. From "Main" page user can reach following pages through credentials: "Login", "Dashboard", "User Control (Users, User Permission)", "Setup(Designation, Status, ProblemType, Govt. Department, Division GEO, District GEO, Building Class, Building Type)", "Entry Form(Circle, Division, Subdivision, Quarter, Building, Allottee, Apartment, Complaint)", All Reports(Complaint Report, User Report, Quarter Report, Building Report, Allottee Report, Summary Report, Task Report) , Profile. All these pages cover necessary functionality of system. It is easy to navigate between these pages. User constantly has access to it through the menu on the left side of page. Note: user has access to welcome page only after login, he can't come back to it, because this page has descriptive characters and doesn't have influence on functionalities of the system.



**Figure 32.** A tree of web pages

### 6.1.1.1. Description

“**Landing**” page has descriptive characters; it contains a list of main system’s functionality and new complaint, admin, survey, language selector information.

“**New Complaint**” page has descriptive characters; it contains a list of main system’s functionality for entry of new complaint for the registered allottee.

**Login** page is constantly present on the top side of the screen for admin and survey menu, which covers main functionalities of the system. After login, authenticate user can access the dashboard.

**Dashboard** page open default manually after successful login and get the function of showing the all complaint (new, InProgress, completed and canceled ) of current day and current month showing total quarter, building, apartment, and allottee. Also showing the report several with the bar diagram.

#### User Control:

1. **User:** User has its own menu on the left of the page, which contains all required operations that could be performed with user.
2. **User Permission:** User Permission has its own menu on the left of page which contains all required operations that could be performed with user permission.

#### Setup:

1. **Designation:** User has its own menu on the left of the page, which contains all required operations that could be performed with designation.
2. **Status:** User Permission has its own menu on the left of page which contains all required operations that could be performed with status.
3. **Problem type:** Problem type has its own menu on the left of page which contains all required operations that could be performed with problem type.
4. **Govt. Department:** Govt. Department has its own menu on the left of page which contains all required operations that could be performed with govt. department.
5. **Division GEO:** Division GEO has its own menu on the left of the page, which contains all required operations that could be performed with division geo.
6. **District GEO:** User Permission has its own menu on the left of page which contains all required operations that could be performed with district geo.
7. **Building Class:** Problem type has its own menu on the left of page which contains all required operations that could be performed with building class.
8. **Building type:** Govt. Department has its own menu on the left of page which contains all required operations that could be performed with building type.

#### Entry form:

1. **Circle:** Circle has its own menu on the left of the page, which contains all

required operations that could be performed with circle.

2. **Division:** Division has its own menu on the left of page which contains all required operations that could be performed with division.
3. **Subdivision:** Subdivision has its own menu on the left of page which contains all required operations that could be performed with subdivision.
4. **Quarter:** Quarter has its own menu on the left of page which contains all required operations that could be performed with quarter.
5. **Building :** Building has its own menu on the left of the page, which contains all required operations that could be performed with building.
6. **Allottee:** Allottee has its own menu on the left of page which contains all required operations that could be performed with allottee.
7. **Apartment:** Apartment has its own menu on the left of page which contains all required operations that could be performed with apartment.
8. **Complaint:** Complaint has its own menu on the left of page which contains all required operations that could be performed with complaint.

#### **All Reports:**

1. **Complaint Report:** Complaint Report has its own menu on the left of the page under all reports section, which contains all required operations that could be performed with complaint report.
2. **Report of User:** Report of user has its own menu on the left of page all reports section which contains all required operations that could be performed with report of user.
3. **Report of user:** Report of user has its own menu on the left of page under all reports which contains all required operations that could be performed with report of user.
4. **Report of Quarter:** Report of Quarter has its own menu on the left of page under all reports which contains all required operations that could be performed with report of quarter.
5. **Report of Building :** Report of Building has its own menu on the left of the page, under all reports which contains all required operations that could be performed with report of building.
6. **Report of Allottee:** Report of Allottee has its own menu on the left of page under all reports which contains all required operations that could be performed with report of allottee.
7. **Summary Report:** Summary report has its own menu on the left of page under all reports which contains all required operations that could be performed with summary report.
8. **Task Report:** Task report has its own menu on the left of page under all reports which contains all required operations that could be performed with task report.

**Profile:** Profile has its own menu on the left of page which contains all required operations that could be performed with task report.

### *Access to Information Idea(A2I) [ Webpage ]*

**Dashboard** page open default manually after successful login and get the function of showing the all complaint (new, InProgress, completed and canceled ) showing the report several with the bar diagram.

**Total Complaint:** Total Complaint has its own menu on the left of page which contains all required operations that could be performed with total complaint

**Total Complaint of this month:** Total Complaint of this month has its own menu on the left of page which contains all required operations that could be performed with total complaint of this month.

**New Complaint:** New Complaint has its own menu on the left of page which contains all required operations that could be performed with new complaint.

**New Complaint of this month:** New Complaint of this month has its own menu on the left of page which contains all required operations that could be performed with new complaint of this month.

**In progress Complaints:** In Progress Complaints has its own menu on the left of page which contains all required operations that could be performed with in progress complaints.

**Completed Complaint:** Complete Complaint has its own menu on the left of page which contains all required operations that could be performed with complete complaint.

**Canceled Complaint:** Canceled Complaint has its own menu on the left of page which contains all required operations that could be performed with canceled complaint.

**Change Password:** Change Password has its own menu on the left of page which contains all required operations that could be performed with change password.

### 1.1.1.2. Objects and Actions

In the “**Landing**” page there is a button named admin ,after clicking the admin login area, user provides login and password in appropriate text boxes and confirms this operation, clicking on button “Sign in”. Also user has opportunity to select language from menu.

After logging “**Dashboard**” page is available to user. It contains following menu: ““User Control (Users, User Permission)”, “Setup(Designation, Status, ProblemType, Govt. Department, Division GEO, District GEO, Building Class, Building Type)”, “Entry Form(Circle, Division, Subdivision, Quarter, Building, Allottee, Apartment, Complaint)”, All Reports(Complaint Report, User Report, Quarter Report, Building Report, Allottee Report, Summary Report, Task Report) , Profile, “Logout”. User clicks on necessary item in the menu in order to transfer in the next page.

Page “Users Control” has menu with submenus on the left of the page:

1. **Users:**
  - Add new
  - View
  - Search with 10 fields.
  - Edit
2. **User Permission:**
  - Permission
3. **Setup**
  - **Designation:**
    - Add new
    - View
    - Edit
  - **Status:**
    - Add new
    - View
    - Edit
  - **Problem Type:**
    - Add new
    - View
    - Edit
  - **Govt. Department:**
    - Add new
    - View
    - Edit
  - **Division GEO:**
    - Add new
    - View
    - Edit

➤ **District GEO:**

- Add new
- View
- Edit

➤ **Building Type :**

- Add new
- View
- Edit

#### 4. Entry Form

➤ **Circle:**

- Add new
- View
- Edit

➤ **Division:**

- Add new
- View
- Edit
- Search with two fields

➤ **Subdivision:**

- Add new
- View
- Edit
- Search with three fields

➤ **Quarter:**

- Add new
- View
- Edit
- Search with Five Fields

➤ **Building:**

- Add new
- View
- Edit
- Search with Five Fields

➤ **Allottee:**

- Add new
- View
- Edit
- Search with Five Fields

➤ **Apartment :**

- Add new
- View
- Edit

- Search with Five Fields
- **Complaint :**
  - View
  - Update only for SDE
  - Search with Eight Fields

## 5. All Reports

- **Complaint Report:**
  - Search with Eight Fields
  - View
  - Print with PDF
  - Download or Export with Excel
- **Report of User :**
  - Search with Ten Fields
  - View
  - Print with PDF
  - Download or Export with Excel
- **Report of Quarter:**
  - Search with Five Fields
  - View
  - Print with PDF
  - Download or Export with Excel
- **Report of Building:**
  - Search with Five Fields
  - View
  - Print with PDF
  - Download or Export with Excel
- **Report of Building:**
  - Search with Four Fields
  - View
  - Print with PDF
  - Download or Export with Excel
- **Report of Allottee:**
  - Search with Four Fields
  - View
  - Print with PDF
  - Download or Export with Excel
- **Profile:**
  - Change Name, Email, password and Picture

User clicks on necessary item in the menu in order to perform operation. Description of operation can be found in the Use Cases.

In addition user can perform next actions:

- see all available fields and records in the table, using scroll bar;
- 
-

set a filter selecting type of asset from drop down menu in the group “Filter” and then clicking button “Search”;

- hide/show columns in the table by clicking link “Hide/Show” next to the name of field;
- perform basic search for current page putting string in appropriate text box and clicking button “Search”;
- edit record in the table by clicking link “Edit” next to the appropriate record;

User clicks on necessary item in the menu in order to perform operation. Description of operation can be found in the Use Cases.

Page “**Search**” allows to user perform Basic search and advanced search. Page contains text box for searched string, button “Search”, table with names of databases and fields where on each page.

### 1.1.2. User interface

Used GUI components are menus, submenus, buttons, text boxes, check boxes, down drop lists, links, and tables. The only means of access to the entire database, by all users, is through this UI.

#### 1.1.2.1. Screen image

Some examples of UI are presented below:

##### **Landing page:**

This is the landing or welcome page of the Complaint Management System(CMS). Allottee, User, Survey user can access to their specific menu.

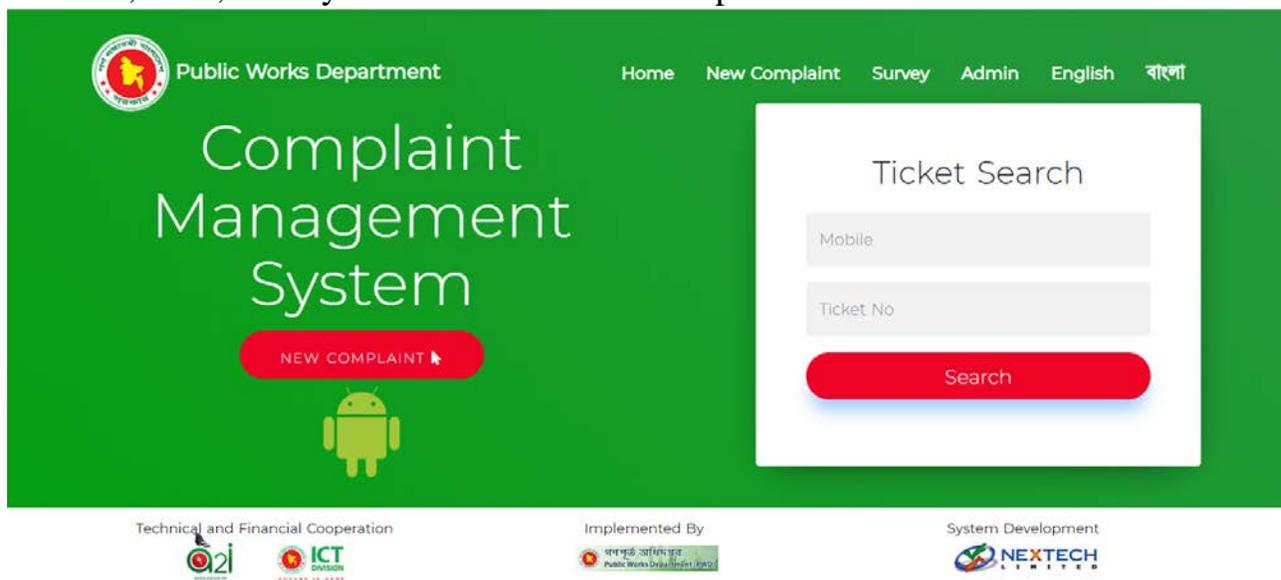


Figure: Landing main page of CMS.

**New Complaint:** This menu is used for allottee to complain to SDE (Sub divisional Engineer).

**Complaint Form for Allottee :** After clicking new complaint this section is appeared on web page and registered can access this form by selecting every credential information as district, quarter, building, flat, building type, name, mobile email(if any ), problem type, problem description.



**Public Works Department**      **Complaint Management System**

**Complaint Form** Home

**District \*** Dhaka      **Quarter / Area / Zone Name \*** Dhaka PWD Railway Diversion...      **Building Name / No \*** Belly (Khilgaon)      **Flat No \*** Belly-101      **Building Type** Dormitory Building

**Name \*** Md Dudu Mia      **Mobile \*** 01854313531      **Email** Allottee Email

**Problem Type\***

Door / Window Problem     Fan repair or replacement     Electrical Switch and circuit breaker     Roof water drainage and water tank repair

Sanitary and Plumbing

**Problem Description \***

Door and Electrical switch problem

**Continue**

Technical and Financial Cooperation:  **ICT DIVISION** FUTURE IS HERE

Implemented By:  গণস্বর্ত্ত অধিদপ্তর Public Works Department (PWD)

System Development:  **NEXTECH LIMITED**

**Review Form for Allottee :** After clicking new complaint successfully this is the review page and get a new code for validation the real allottee. Then putting the real code and sending the final submission.



Public Works Department

## Complaint Management System

**Review**

District	Dhaka
Quarter Name	Motijheel colony (Ideal Zone )
Building No	B-1
Flat no and Type	E-10
Name	Md. A. Al Faisal Mia
Mobile	01714522292
Email	af_apon@yahoo.com
Problem Type	Door / Window Problem, Electrical Switch and circuit breaker
Problem Description	Door and Electrical Switch Problem

Please check SMS and fill verification code

PIN NUMBER NOT MATCHED

**Login:** This Login page is for PWD Login, To reach the panel dashboard this page is used for user authentication.

**PWD CMS**

Sign in

Username

Password

[Home](#) [I forgot my password](#)

## Dashboard

### Today's Complaints

**New Complain:** It shows the no. of today's complaint.

**In Progress Complaint:** It shows the no. of today's in progress complaint.

**Completed Complaint:** It shows the no. of complaint which have been completed today.

**Canceled Complaint:** It shows the no. of complaint which have been canceled today.

**This Month's Complaints**

**New Complain:** It shows the no. of complaint for running month.

**In Progress Complaint:** It shows the no. of in progress complaint for running month.

**Completed Complaint:** It shows the no. of complaint which have been completed during running month.

**Canceled Complaint:** It shows the no. of complaint which have been canceled during running month.

**Total Quarters, Buildings, Apartments and Allottees**

**Total Quarters:** It shows the total number of quarters.

**Total Building:** It shows the total number of buildings.

**Total Apartments:** It shows the total number of apartments.

**Total Allottees:** It shows the total number of allottees.

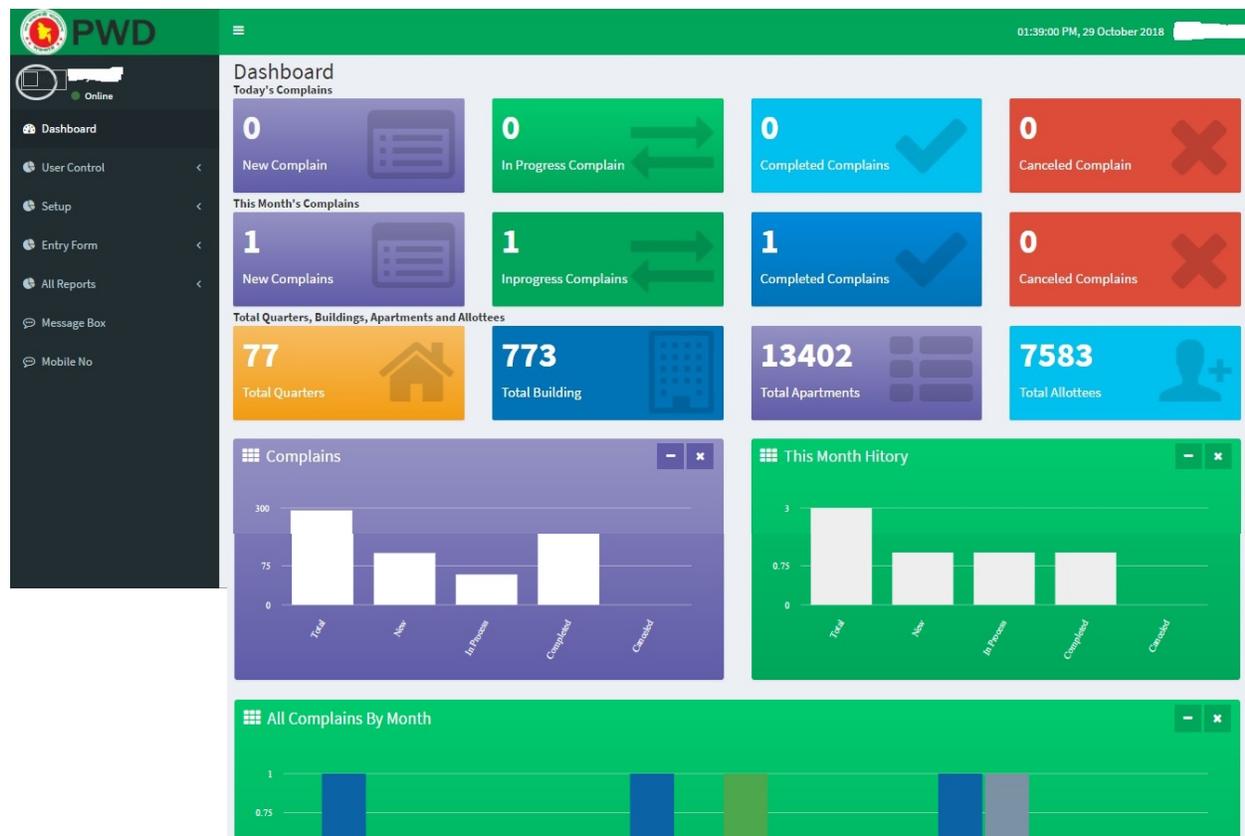
**Complains:** This graph shows the overall view of all complaints.

**This Month History:** This graph shows the running month complaints history.

**All Complaints By Month:** This graph shows all complaints monthly.

**All Complaints By Year:** This graph shows all complaints yearly.

**All Building By Type:** This graph shows all buildings chart as their type





# User Control

## 3.1 User :

From this section, users can be added or managed.  
Click +Add new User icon to add a new user.

The screenshot shows the 'User Information' form. At the top left, there is a blue button labeled '+ Add new User' which is circled. At the top right, there is a blue 'Search' button. The form is divided into several sections:
 

- Name**: A text input field.
- Mobile**: A text input field.
- User Name**: A text input field.
- User Role**: A dropdown menu with 'Choose Role' selected.
- User Status**: A dropdown menu with 'Choose Status' selected.
- Department**: A dropdown menu with 'Choose Dept.' selected.
- Designation**: A dropdown menu with 'Choose Designation' selected.
- Circle**: A dropdown menu with 'Choose Circle' selected.
- Division**: A dropdown menu with 'Choose Circle First' selected.
- Sub Division**: A dropdown menu with 'Choose Division First' selected.

 At the bottom right of the form, there is a green 'Search' button.

After clicking +Add new user icon, a new box will be showed.

The screenshot shows the 'Add User Information' modal form. It contains the following fields:
 

- First Name \***: Text input field.
- Last Name**: Text input field.
- Mobile \***: Text input field.
- Email**: Text input field.
- User Name \***: Text input field.
- Password \***: Text input field.
- Re Password**: Text input field.
- Designation**: Dropdown menu with 'Select a designation' selected.
- Department**: Dropdown menu with 'Select a department' selected.
- User Type**: Dropdown menu with 'Select a User Type' selected.
- Circle**: Dropdown menu with 'Select Circle' selected.
- Division**: Dropdown menu with 'Select Circle First' selected.
- Subdivision**: Dropdown menu with 'Select Division First' selected.
- Status**: Dropdown menu with 'Select a Status' selected.

 At the bottom right, there are two buttons: a green 'Submit' button and a red 'Back' button.

In this box, input new user's all information and submit it.  
If you want to see all users, click Search option.

The screenshot shows the 'User Information' form. At the top left, there is a blue '+ Add new User' button. At the top right, there is a blue 'Search' button which is circled. The form contains the same fields as in the previous screenshot: Name, Mobile, User Name, User Role, User Status, Department, Designation, Circle, Division, and Sub Division. At the bottom right, there is a green 'Search' button.

If you want to see a specific user, then input his name and click this Search option.

User Information

+ Add new User
Search

Name	Mobile	User Name	User Role Choose Role
User Status Choose Status	Department Choose Dept.	Designation Choose Designation	Circle Choose Circle
Division Choose Circle First	Sub Division Choose Division First		

Search

If you want to edit a specific user, input his name and click this Search option. Then click edit option and edit as you want.

User Information

+ Add new User
Search

Name	Mobile	User Name	User Role Choose Role
User Status Choose Status	Department Choose Dept.	Designation Choose Designation	Circle Choose Circle
Division Choose Circle First	Sub Division Choose Division First		

Search

Sl	Name	Phone	Email	U.Name	Role	Status	Circle	Division	Subdiv	Dept.	Degsignatin	Edit
1	██████████	01-██████████	██████████.com ██████████.bd	██████████	Administrator	Active				Admin		<span>Edit</span>

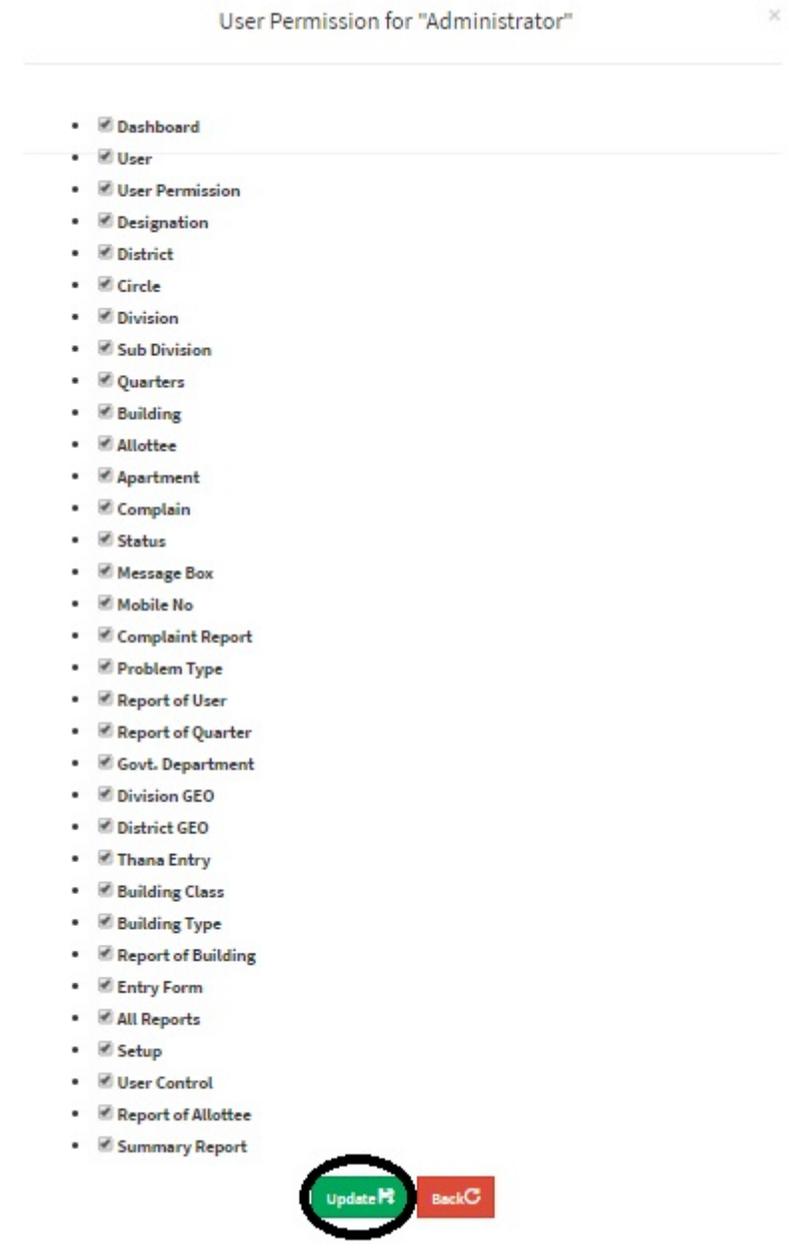
### 3.2 User Permission:

From this section, you can give the access permission of anything to any user as their category.

User Permission

#SL	Name	Role	Status	Permission
1	Administrator	Divisional Admin	Given	<span>Permission</span>
2	General	Divisional Admin	Given	<span>Permission</span>
3	Divisional Admin	Divisional Admin	Not Given	<span>Permission</span>
4	SE	Divisional Admin	Not Given	<span>Permission</span>

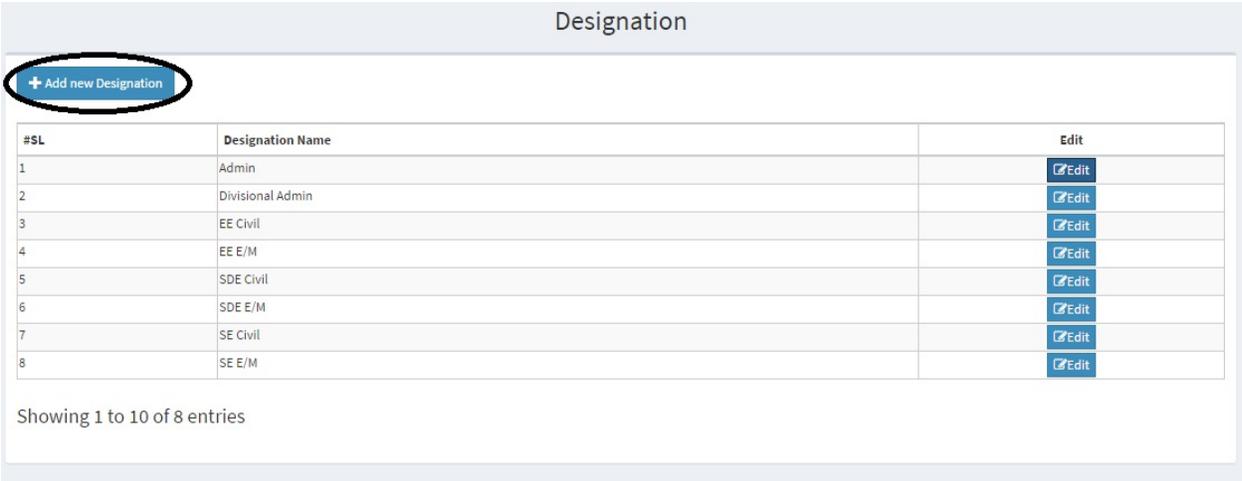
After clicking permission option, a new box will show. From this box, give permission by clicking these options (if you want) and update it.



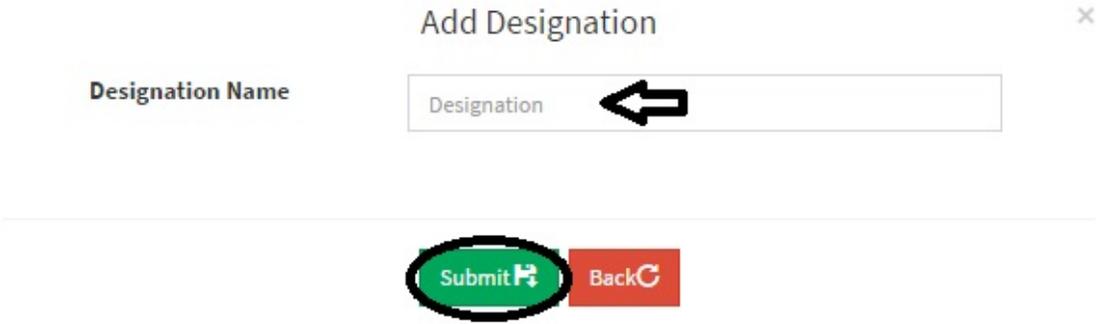
# Setup

## 4.1 Designation

From this section, you can add or edit designation for the users. Click + Add new Designation option to add new.



A new box will be come. Then input here the new designation and submit it.



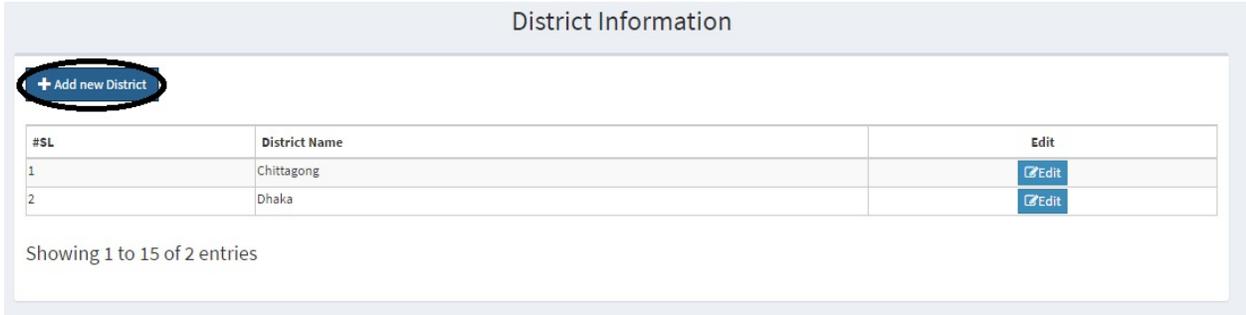
If you want to edit any designation, then click the edit option, edit it and update it.



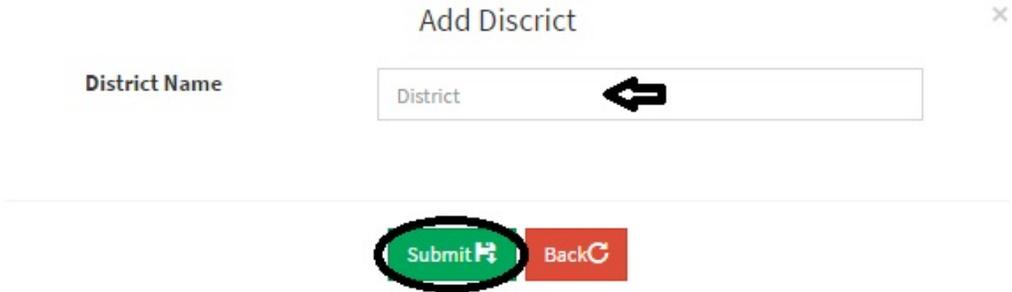
Showing 1 to 10 of 8 entries

## 4.2 District

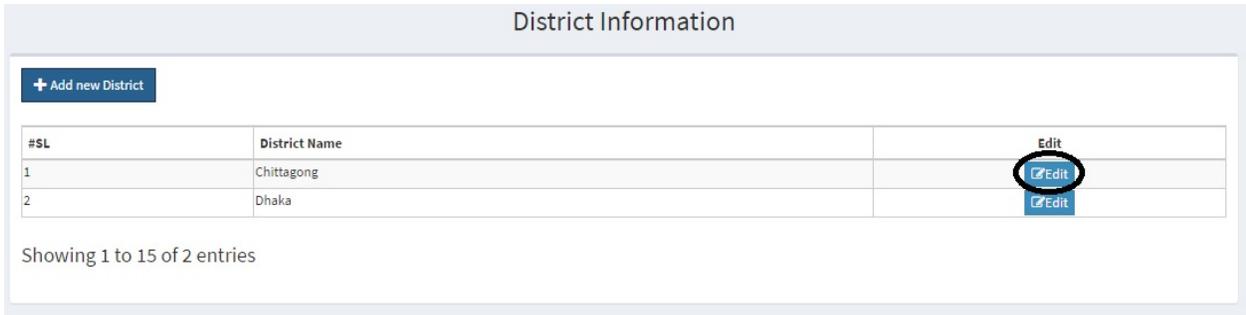
From this section, you can add or edit district name for the users. Click + Add new District option to add new.



A new box will be come. Then input here the new district and submit it.

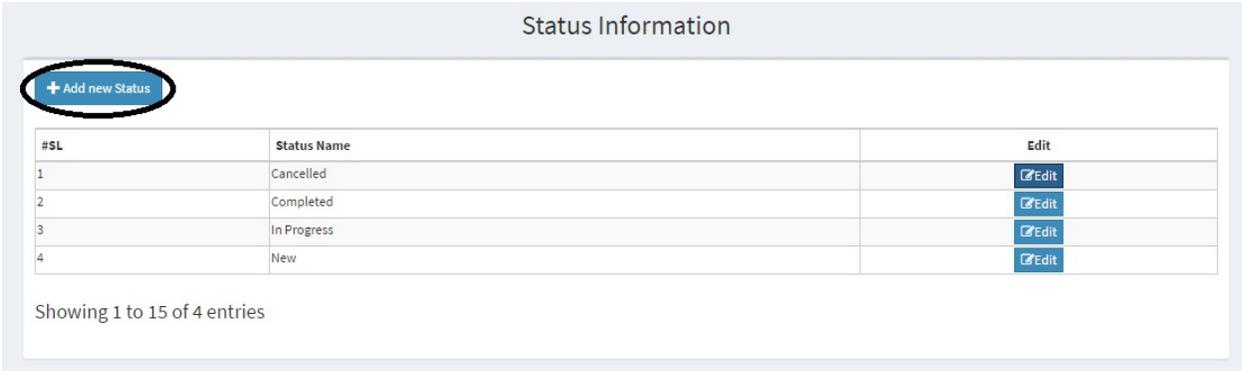


If you want to edit any district name, then click the edit option, edit it and update it.

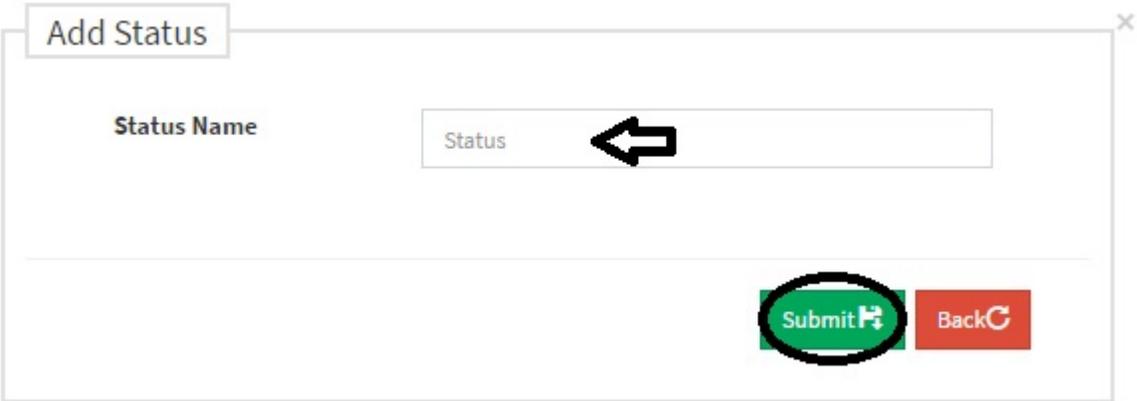


### 4.3 Status

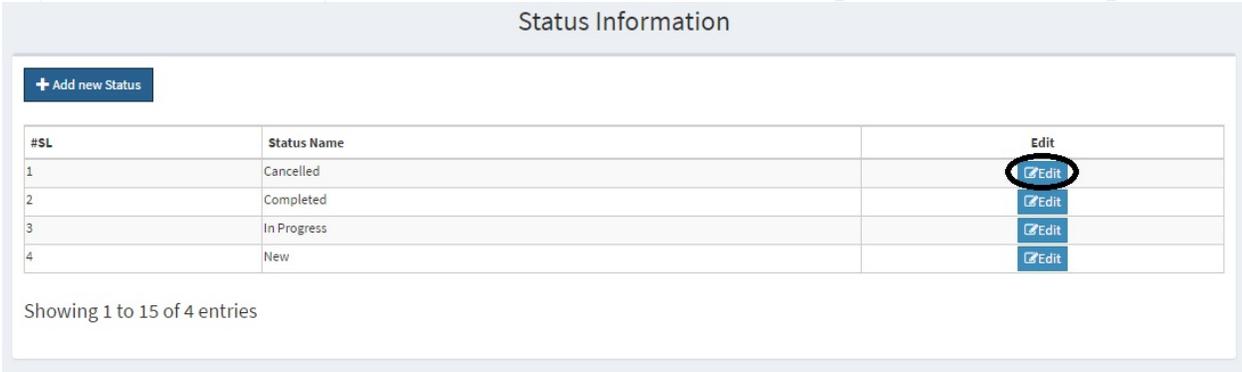
From this section, you can add or edit status name for the users.  
Click + Add new Status option to add new.



A new box will be come. Then input here the new status and submit it.



If you want to edit any status name, then click the edit option, edit it and update it.



### 4.4 Problem Type

From this section, you can add or edit problem type for the users. Click + Add new Problem Type option to add new.

Problem Information

[+ Add new Problem type](#)

#SL	Problem Description	Department	Maximum day to solve	Edit
1	Door / Window Problem	Civil	1	<a href="#">Edit</a>
2	Fan repair or replacement	Electrical	1	<a href="#">Edit</a>
3	Electrical Switch and circuit breaker	Electrical	1	<a href="#">Edit</a>
4	Roof water drainage and water tank repair	Civil	1	<a href="#">Edit</a>
5	Sanitary and Plumbing	Civil	1	<a href="#">Edit</a>

Showing 1 to 15 of 5 entries

A new box will be come. Then input here the new problem name, department, and maximum day to solve this problem and submit it.

Add Problem

↓

**Problem Name\***

**Department\***

**Maximum day to solve problem\***

[Submit](#) [Back](#)

If you want to edit any problem type name, then click the edit option, edit it and update it.

Problem Information

[+ Add new Problem type](#)

#SL	Problem Description	Department	Maximum day to solve	Edit
1	Door / Window Problem	Civil	1	<a href="#">Edit</a>
2	Fan repair or replacement	Electrical	1	<a href="#">Edit</a>
3	Electrical Switch and circuit breaker	Electrical	1	<a href="#">Edit</a>
4	Roof water drainage and water tank repair	Civil	1	<a href="#">Edit</a>
5	Sanitary and Plumbing	Civil	1	<a href="#">Edit</a>

Showing 1 to 15 of 5 entries

## 4.5 Govt. Department

From this section, you can add or edit govt. department for the users. Click + Add new govt. department option to add new.

Government Department Information

[+ Add new govt. department](#)

#SL	Govt. Department Name	Edit
1	Planning Commission	<a href="#">Edit</a>
2	Bangladesh secretariat	<a href="#">Edit</a>

Showing 1 to 20 of 2 entries

A new box will be come. Then input here the new govt. department name and submit it.

Add govt. department Name

**Govt Department Name\***

Govt Department Name

[Submit](#) [Back](#)

If you want to edit any govt. department's name, then click the edit option, edit it and update it.

Government Department Information

[+ Add new govt. department](#)

#SL	Govt. Department Name	Edit
1	Planning Commission	<a href="#">Edit</a>
2	Bangladesh secretariat	<a href="#">Edit</a>

Showing 1 to 20 of 2 entries

### 4.6 Division GEO

From this section, you can add or edit division geo for the users. Click + Add new division geo option to add new.



A new box will be come. Then input here the new division geo name and submit it.



If you want to edit any division geo's name, then click the edit option, edit it and update it.



## 4.7 District GEO

From this section, you can add or edit district geo for the users.  
Click + Add new district geo option to add new.

District GEO Information

[+ Add new district geo](#)

#SL	Division GEO Name	District GEO Name	District Website	Edit
1	Barishal	Barguna	www.barguna.gov.bd	<a href="#">Edit</a>
2	Barishal	Barishal	www.barisal.gov.bd	<a href="#">Edit</a>
3	Barishal	Bhola	www.bhola.gov.bd	<a href="#">Edit</a>
4	Barishal	Jhalokati	www.jhalakathi.gov.bd	<a href="#">Edit</a>
5	Barishal	Patuakhali	www.patuakhali.gov.bd	<a href="#">Edit</a>
6	Barishal	Pirojpur	www.pirojpur.gov.bd	<a href="#">Edit</a>
7	Chattogram	Bandarban	www.bandarban.gov.bd	<a href="#">Edit</a>
8	Chattogram	Brahmanbaria	www.brahmanbaria.gov.bd	<a href="#">Edit</a>

A new box will be come. Then input here the new district geo name, division name and district website and then submit it.

Add District GEO ×

**District GEO**

**Division**

**District Website**

If you want to edit any district geo's name, then click the edit option, edit it and update it.

District GEO Information

[+ Add new district geo](#)

#SL	Division GEO Name	District GEO Name	District Website	Edit
1	Barishal	Barguna	www.barguna.gov.bd	<a href="#">Edit</a>
2	Barishal	Barishal	www.barisal.gov.bd	<a href="#">Edit</a>
3	Barishal	Bhola	www.bhola.gov.bd	<a href="#">Edit</a>
4	Barishal	Jhalokati	www.jhalakathi.gov.bd	<a href="#">Edit</a>
5	Barishal	Patuakhali	www.patuakhali.gov.bd	<a href="#">Edit</a>
6	Barishal	Pirojpur	www.pirojpur.gov.bd	<a href="#">Edit</a>
7	Chattogram	Bandarban	www.bandarban.gov.bd	<a href="#">Edit</a>
8	Chattogram	Brahmanbaria	www.brahmanbaria.gov.bd	<a href="#">Edit</a>

## 4.8 Thana Entry

From this section, you can add or edit thana name for the users.  
Click + Add new thana option to add new.

Thana Information				
#SL	Division Name	District Name	Thana Name	Edit
1	Barishal	Barguna	Amtali Upazila	<a href="#">Edit</a>
2	Barishal	Barguna	Bamna Upazila	<a href="#">Edit</a>
3	Barishal	Barguna	Barguna Sadar Upazila	<a href="#">Edit</a>
4	Barishal	Barguna	Betagi Upazila	<a href="#">Edit</a>
5	Barishal	Barguna	Patharghata Upazila	<a href="#">Edit</a>
6	Barishal	Barguna	Taltali Upazila	<a href="#">Edit</a>
7	Barishal	Barishal	Agailjhara Upazila	<a href="#">Edit</a>
8	Barishal	Barishal	Babuganj Upazila	<a href="#">Edit</a>

A new box will be come. Then input here the new thana name, division name and district name and then submit it.

Add Thana
✕



**Thana Name**

**Division Name**

**District Name**

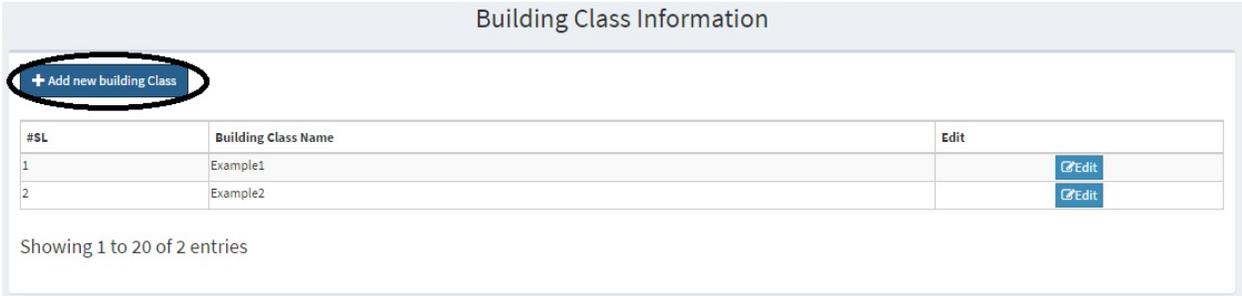
Submit
Back

If you want to edit any thana's name, then click the edit option, edit it and update it.

Thana Information				
#SL	Division Name	District Name	Thana Name	Edit
1	Barishal	Barguna	Amtali Upazila	<a href="#">Edit</a>
2	Barishal	Barguna	Bamna Upazila	<a href="#">Edit</a>
3	Barishal	Barguna	Barguna Sadar Upazila	<a href="#">Edit</a>
4	Barishal	Barguna	Betagi Upazila	<a href="#">Edit</a>
5	Barishal	Barguna	Patharghata Upazila	<a href="#">Edit</a>
6	Barishal	Barguna	Taltali Upazila	<a href="#">Edit</a>
7	Barishal	Barishal	Agailjhara Upazila	<a href="#">Edit</a>
8	Barishal	Barishal	Babuganj Upazila	<a href="#">Edit</a>

### 4.9 Building Class

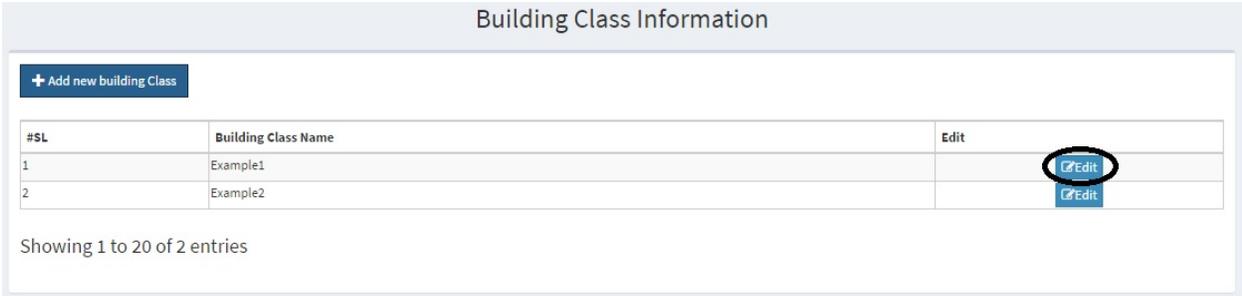
From this section, you can add or edit building class name for the users. Click + Add new building class option to add new.



A new box will be come. Then input here the new building class name and then submit it.



If you want to edit any building class's name, then click the edit option, edit it and update it.



### 4.10 Building Type

From this section, you can add or edit building type for the users. Click + Add new building type option to add new.

Building Type Information

[+ Add new buildingtype](#)

#SL	Building Class Name	Building Type Name	Edit
1		2-storied Building	<a href="#">Edit</a>
2		Banglow	<a href="#">Edit</a>
3		Barrack Building.(Res )	<a href="#">Edit</a>
4		D-Type ( 1000 Sft )	<a href="#">Edit</a>
5		Office Building.	<a href="#">Edit</a>
6		Post Office Building.	<a href="#">Edit</a>
7		1-storied Office Building	<a href="#">Edit</a>
8		12 Unite 140 sft	<a href="#">Edit</a>

A new box will be come. Then input here the new building type and then submit it.

**Add Buildingtype** ✕

**Buildingtype Name \***

**Building Class Name \***

If you want to edit any building type's, then click the edit option, edit it and update it.

Building Type Information

[+ Add new buildingtype](#)

#SL	Building Class Name	Building Type Name	Edit
1		2-storied Building	<a href="#">Edit</a>
2		Banglow	<a href="#">Edit</a>
3		Barrack Building.(Res )	<a href="#">Edit</a>
4		D-Type ( 1000 Sft )	<a href="#">Edit</a>
5		Office Building.	<a href="#">Edit</a>
6		Post Office Building.	<a href="#">Edit</a>
7		1-storied Office Building	<a href="#">Edit</a>
8		12 Unite 140 sft	<a href="#">Edit</a>

# Entry Form

## 5.1 Circle

From this section, you can add or edit circle for the users.

Click + Add new Circle option to add new.

Circle Information		
<a href="#">+ Add new Circle</a>		
#SL	Circle Name	Edit
1	Barishal	<a href="#">Edit</a>
2	Bogra Circle	<a href="#">Edit</a>
3	Chtg Circle-1	<a href="#">Edit</a>
4	Chtg Circle-2	<a href="#">Edit</a>
5	Comilla Circle	<a href="#">Edit</a>
6	Dhaka Circle-1	<a href="#">Edit</a>
7	Dhaka Circle-2	<a href="#">Edit</a>
8	Dhaka Circle-3	<a href="#">Edit</a>

A new box will be come. Then input here the new circle name and submit it.

Add Circle Information

**Circle Name**

Submit

Back

If you want to edit any circle's name, then click the edit option, edit it and update it.

Circle Information		
<a href="#">+ Add new Circle</a>		
#SL	Circle Name	Edit
1	Barishal	<a href="#">Edit</a>
2	Bogra Circle	<a href="#">Edit</a>
3	Chtg Circle-1	<a href="#">Edit</a>
4	Chtg Circle-2	<a href="#">Edit</a>
5	Comilla Circle	<a href="#">Edit</a>
6	Dhaka Circle-1	<a href="#">Edit</a>
7	Dhaka Circle-2	<a href="#">Edit</a>
8	Dhaka Circle-3	<a href="#">Edit</a>

## Division

From this section, you can add or edit division for the users.  
Click + Add new Division option to add new.

**Division Information**

+ Add new Division

**Circle**

**Division**

**Search**

#SL	Circle Name	Division Name	Edit
1	Khulna Circle	Bagerhat Division	<a href="#">Edit</a>
2	Chtg Circle-2	Bandarban Division	<a href="#">Edit</a>
3	Barishal	Barguna Division	<a href="#">Edit</a>
4	Barishal	Barishal Division	<a href="#">Edit</a>
5	Barishal	Bhola Division	<a href="#">Edit</a>
6	Bogra Circle	Bogra Division	<a href="#">Edit</a>
7	Comilla Circle	Brammanbaria Division	<a href="#">Edit</a>
8	Comilla Circle	Chandpur Division	<a href="#">Edit</a>

A new box will be come. Then input here the new division name and circle name and then submit it.

**Add Division** ×

**Division Name**

**Circle Name**

Submit

Back

You can search a specific circle or division to check or edit. To do this, write the circle name in the circle box, then division name in the division box and then click Search option.

**Division Information**

+ Add new Division

**Circle**

**Division**

**Search**

#SL	Circle Name	Division Name	Edit
1	Khulna Circle	Bagerhat Division	<a href="#">Edit</a>
2	Chtg Circle-2	Bandarban Division	<a href="#">Edit</a>
3	Barishal	Barguna Division	<a href="#">Edit</a>
4	Barishal	Barishal Division	<a href="#">Edit</a>
5	Barishal	Bhola Division	<a href="#">Edit</a>
6	Bogra Circle	Bogra Division	<a href="#">Edit</a>
7	Comilla Circle	Brammanbaria Division	<a href="#">Edit</a>
8	Comilla Circle	Chandpur Division	<a href="#">Edit</a>

If you want to edit any division's name, then click the edit option, edit it and update it.

**Division Information**

#SL	Circle Name	Division Name	Edit
1	Khulna Circle	Bagerhat Division	<a href="#">Edit</a>
2	Chtg Circle-2	Bandarban Division	<a href="#">Edit</a>
3	Barishal	Barguna Division	<a href="#">Edit</a>
4	Barishal	Barishal Division	<a href="#">Edit</a>
5	Barishal	Bhola Division	<a href="#">Edit</a>
6	Bogra Circle	Bogra Division	<a href="#">Edit</a>
7	Comilla Circle	Brammanbaria Division	<a href="#">Edit</a>
8	Comilla Circle	Chandpur Division	<a href="#">Edit</a>

### Sub Division

From this section, you can add or edit sub-division for the users. Click + Add new Subdivision option to add new

**Sub Division Information**

#SL	Circle Name	Division Name	Subdivision	Edit
1	Chtg Circle-2	Chittagong Division-4	Arboriculture PWD Subdivision ,Chittagong	<a href="#">Edit</a>
2	Barishal	Gopalganj Division	Arboriculture PWD Subdivision ,Gopalganj	<a href="#">Edit</a>
3	Dhaka Circle-4	Dhaka Azimpur PWD Division	Azimpur Sub-Division-1, Dhaka	<a href="#">Edit</a>
4	Dhaka Circle-4	Dhaka Azimpur PWD Division	Azimpur Sub-Division-2, Dhaka	<a href="#">Edit</a>
5	Dhaka Circle-1	Dhaka City PWD Division	Bangabhaban PWD Sub-Division	<a href="#">Edit</a>
6	Barishal	Bhola Division	Bhola PWD Subdivision , Bhola	<a href="#">Edit</a>
7	Dhaka Circle-2	Dhaka PWD Division-4	Dhaka PWD Sub-Division-5, Dhaka	<a href="#">Edit</a>
8	Dhaka Circle-2	Dhaka PWD Division-4	Dhaka PWD Sub-Division-6, Dhaka	<a href="#">Edit</a>

A new box will be come. Then input here the new subdivision name and circle name and then submit it.

**Add Subdivision** ×

**Subdivision Name**

**Circle Name**

**Division Name**

You can search a specific circle, division or subdivision to check or edit. To do this, write the circle name in the circle box, division name in the division box and subdivision name in the subdivision box and then click Search option.

**Sub Division Information**

[+ Add new Subdivision](#)

**Circle**

**Division**

**Subdivision**

**Search**

#SL	Circle Name	Division Name	Subdivision	Edit
1	Chtg Circle-2	Chittagong Division-4	Arboriculture PWD Subdivision ,Chittagong	<a href="#">Edit</a>
2	Barishal	Gopalganj Division	Arboriculture PWD Subdivision ,Gopalganj	<a href="#">Edit</a>
3	Dhaka Circle-4	Dhaka Azimpur PWD Division	Azimpur Sub-Division-1, Dhaka	<a href="#">Edit</a>
4	Dhaka Circle-4	Dhaka Azimpur PWD Division	Azimpur Sub-Division-2, Dhaka	<a href="#">Edit</a>
5	Dhaka Circle-1	Dhaka City PWD Division	Bangabhaban PWD Sub-Division	<a href="#">Edit</a>
6	Barishal	Bhola Division	Bhola PWD Subdivision , Bhola	<a href="#">Edit</a>
7	Dhaka Circle-2	Dhaka PWD Division-4	Dhaka PWD Sub-Division-5, Dhaka	<a href="#">Edit</a>
8	Dhaka Circle-2	Dhaka PWD Division-4	Dhaka PWD Sub-Division-6, Dhaka	<a href="#">Edit</a>

If you want to edit any subdivision’s name, then click the edit option, edit it and update it.

**Sub Division Information**

[+ Add new Subdivision](#)

**Circle**

**Division**

**Subdivision**

**Search**

#SL	Circle Name	Division Name	Subdivision	Edit
1	Chtg Circle-2	Chittagong Division-4	Arboriculture PWD Subdivision ,Chittagong	<a href="#">Edit</a>
2	Barishal	Gopalganj Division	Arboriculture PWD Subdivision ,Gopalganj	<a href="#">Edit</a>
3	Dhaka Circle-4	Dhaka Azimpur PWD Division	Azimpur Sub-Division-1, Dhaka	<a href="#">Edit</a>
4	Dhaka Circle-4	Dhaka Azimpur PWD Division	Azimpur Sub-Division-2, Dhaka	<a href="#">Edit</a>
5	Dhaka Circle-1	Dhaka City PWD Division	Bangabhaban PWD Sub-Division	<a href="#">Edit</a>
6	Barishal	Bhola Division	Bhola PWD Subdivision , Bhola	<a href="#">Edit</a>
7	Dhaka Circle-2	Dhaka PWD Division-4	Dhaka PWD Sub-Division-5, Dhaka	<a href="#">Edit</a>
8	Dhaka Circle-2	Dhaka PWD Division-4	Dhaka PWD Sub-Division-6, Dhaka	<a href="#">Edit</a>

## Quarters

From this section, you can add or edit quarters name for the users. Click + Add new Quarter option to add new.

**Quarter Information**

[+ Add new Quarter](#)
[Search Show/Hide](#)

**Circle**

**Division**

**Sub Division**

**Quarters**

**District**

**Search**

#SL	District	Circle	Division	Subdivision	Quarters Name	Assigned Employee	Edit
1	Chittagong	Chtg Circle-2	Chittagong Division-4	PWD Resource Subdivision-Chittagong	PWD Resource Office Building	-	<a href="#">Edit</a>
2	Chittagong	Chtg Circle-2	Chittagong Division-4	PWD Resource Subdivision-Chittagong	Monsurabad Colony	-	<a href="#">Edit</a>
3	Chittagong	Chtg Circle-2	Chittagong Division-4	PWD Subdivision 8,Chittagong	Government CGS Colony	-	<a href="#">Edit</a>
4	Chittagong	Chtg Circle-2	Chittagong Division-4	PWD Subdivision 8,Chittagong	Government Bahutala Colony	-	<a href="#">Edit</a>
5	Chittagong	Chtg Circle-2	Chittagong Division-4	PWD Subdivision 8,Chittagong	Government C & B Colony	-	<a href="#">Edit</a>
6	Chittagong	Chtg Circle-2	Chittagong Division-4	PWD Subdivision 8,Chittagong	Government CGO Building No-1	-	<a href="#">Edit</a>
7	Chittagong	Chtg Circle-2	Chittagong Division-4	PWD Subdivision 9,Chittagong	Gazetted Hostel Colony	-	<a href="#">Edit</a>
8	Chittagong	Chtg Circle-2	Chittagong Division-4	PWD Subdivision 9,Chittagong	Government CGO Building No-2	-	<a href="#">Edit</a>

A new box will be come. Then input here the new quarter name and choose circle name, division name, district name and subdistrict name from the drop down menu and then submit it.

You can search a specific circle, division, subdivision, quarter or district to check or edit. To do this, select the circle name, division name, subdivision name, quarter name and district name from the drop down box and then click Search option.

**Quarter Information**

[+ Add new Quarter](#)
[Search Show/Hide](#)

**Circle** ↓

**Division** ↓

**Sub Division** ↓

**Quarters** ↓

**District** ↓

#SL	District	Circle	Division	Subdivision	Quarters Name	Assigned Employee	Edit
1	Chittagong	Chtg Circle-2	Chittagong Division-4	PWD Resource Subdivision-Chittagong	PWD Resource Office Building	-	<a href="#">Edit</a>
2	Chittagong	Chtg Circle-2	Chittagong Division-4	PWD Resource Subdivision-Chittagong	Monsurabad Colony	-	<a href="#">Edit</a>
3	Chittagong	Chtg Circle-2	Chittagong Division-4	PWD Subdivion 8,Chittagong	Government CGS Colony	-	<a href="#">Edit</a>
4	Chittagong	Chtg Circle-2	Chittagong Division-4	PWD Subdivion 8,Chittagong	Government Bahutala Colony	-	<a href="#">Edit</a>
5	Chittagong	Chtg Circle-2	Chittagong Division-4	PWD Subdivion 8,Chittagong	Government C & B Colony	-	<a href="#">Edit</a>
6	Chittagong	Chtg Circle-2	Chittagong Division-4	PWD Subdivion 8,Chittagong	Government CGO Building No-1	-	<a href="#">Edit</a>
7	Chittagong	Chtg Circle-2	Chittagong Division-4	PWD Subdivion 9,Chittagong	Gazetted Hostel Colony	-	<a href="#">Edit</a>
8	Chittagong	Chtg Circle-2	Chittagong Division-4	PWD Subdivion 9,Chittagong	Government CGO Building No-2	-	<a href="#">Edit</a>

If you want to edit any quarter's name, then click the edit option, edit it and update it.

**Quarter Information**

[+ Add new Quarter](#)
[Search Show/Hide](#)

**Circle**

**Division**

**Sub Division**

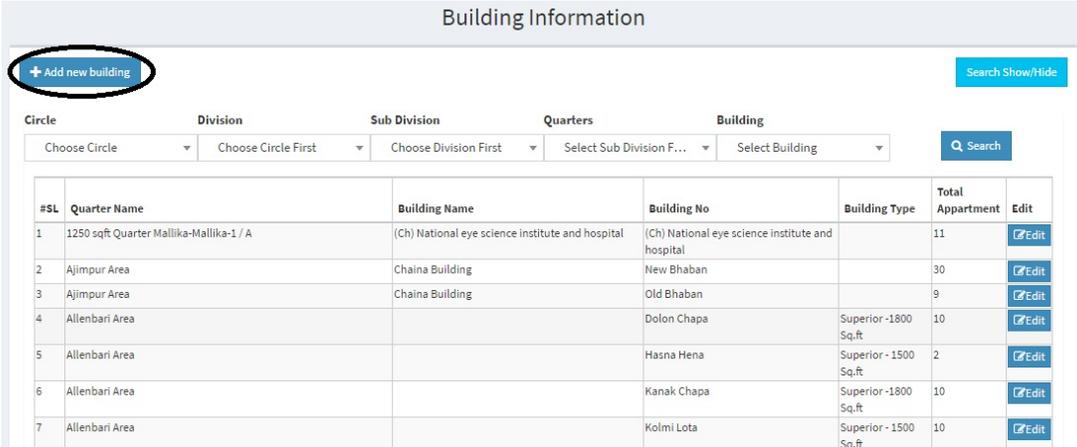
**Quarters**

**District**

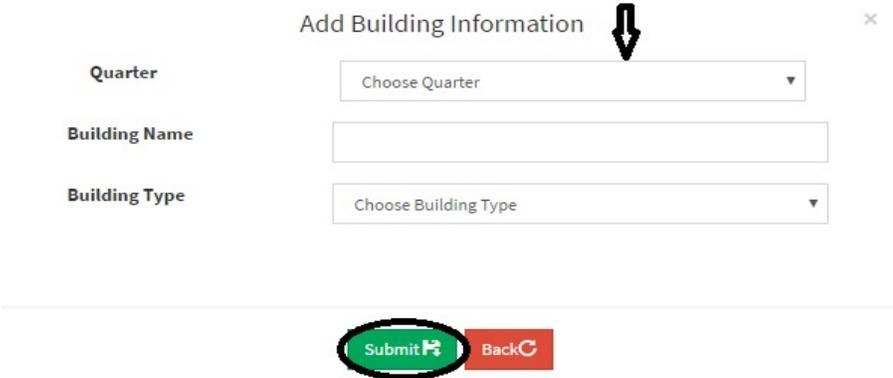
#SL	District	Circle	Division	Subdivision	Quarters Name	Assigned Employee	Edit
1	Chittagong	Chtg Circle-2	Chittagong Division-4	PWD Resource Subdivision-Chittagong	PWD Resource Office Building	-	<a href="#">Edit</a>
2	Chittagong	Chtg Circle-2	Chittagong Division-4	PWD Resource Subdivision-Chittagong	Monsurabad Colony	-	<a href="#">Edit</a>
3	Chittagong	Chtg Circle-2	Chittagong Division-4	PWD Subdivion 8,Chittagong	Government CGS Colony	-	<a href="#">Edit</a>
4	Chittagong	Chtg Circle-2	Chittagong Division-4	PWD Subdivion 8,Chittagong	Government Bahutala Colony	-	<a href="#">Edit</a>
5	Chittagong	Chtg Circle-2	Chittagong Division-4	PWD Subdivion 8,Chittagong	Government C & B Colony	-	<a href="#">Edit</a>
6	Chittagong	Chtg Circle-2	Chittagong Division-4	PWD Subdivion 8,Chittagong	Government CGO Building No-1	-	<a href="#">Edit</a>
7	Chittagong	Chtg Circle-2	Chittagong Division-4	PWD Subdivion 9,Chittagong	Gazetted Hostel Colony	-	<a href="#">Edit</a>
8	Chittagong	Chtg Circle-2	Chittagong Division-4	PWD Subdivion 9,Chittagong	Government CGO Building No-2	-	<a href="#">Edit</a>

## Building

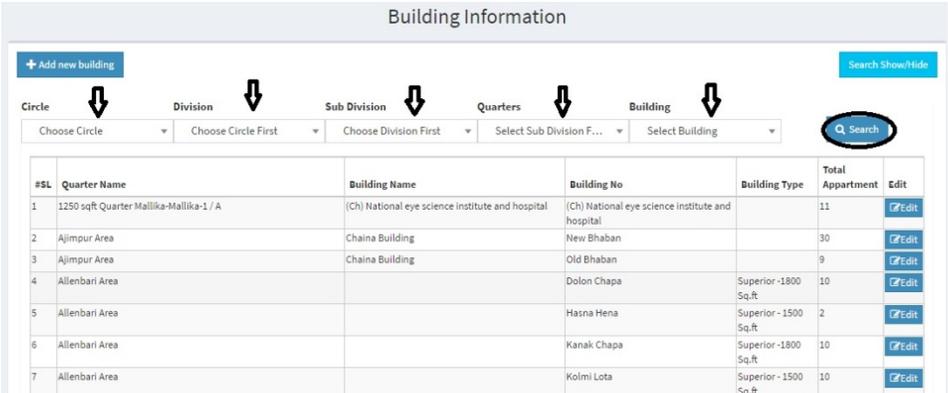
From this section, you can add or edit buildings name for the users. Click + Add new building option to add new.



A new box will be come. Then input here the new building name and choose quarter name and building type from the drop down box and then submit it.



You can search a specific circle, division, subdivision, quarter or building to check or edit. To do this, select the circle name, division name, subdivision name, quarter name and district name from the drop down box and then click Search option.



If you want to edit any building's name, then click the edit option, edit it and update it.

**Building Information**

[+ Add new building](#) [Search Show/Hide](#)

Circle: Choose Circle    Division: Choose Circle First    Sub Division: Choose Division First    Quarters: Select Sub Division F...    Building: Select Building    [Q Search](#)

#SL	Quarter Name	Building Name	Building No	Building Type	Total Apartment	Edit
1	1250 sqft Quarter Mallika-Mallika-1 / A	(Ch) National eye science institute and hospital	(Ch) National eye science institute and hospital		11	<a href="#">Edit</a>
2	Ajimpur Area	Chaina Building	New Bhaban		30	<a href="#">Edit</a>
3	Ajimpur Area	Chaina Building	Old Bhaban		9	<a href="#">Edit</a>
4	Allenbari Area		Dolon Chapa	Superior -1800 Sq.ft	10	<a href="#">Edit</a>
5	Allenbari Area		Hasna Hena	Superior - 1500 Sq.ft	2	<a href="#">Edit</a>
6	Allenbari Area		Kanak Chapa	Superior -1800 Sq.ft	10	<a href="#">Edit</a>
7	Allenbari Area		Kolmi Lota	Superior - 1500 Sq.ft	10	<a href="#">Edit</a>

## Allottee

From this section, you can add or edit allottees name for the users. Click + Add new allottee option to add new.

**Allottee Information**

[+ Add new allottee](#)

Circle: Choose Circle    Division: Choose Circle First    Sub Division: Choose Division First    Quarters: Select Sub Division First    Building: Select Quarter First    [Q Search](#)

#SL	Allottee Name	Email	Phone	Govt.Department	Designation	NID	Permanent Address	Division	District	Thana	Edit
1	Julhas Miah										<a href="#">Edit</a>
2	Abul Hasem										<a href="#">Edit</a>
3	Md. Ful Miah										<a href="#">Edit</a>
4	Monoara Begum										<a href="#">Edit</a>
5	Shekh Muzibur Rahman Montu										<a href="#">Edit</a>
6	Md. Mokhlesur Rahmahn										<a href="#">Edit</a>
7	Mostofa Kamal Mojumder										<a href="#">Edit</a>

A new box will be come. Then input here the new allottee's name and other information of the allottee and then submit it.

Add Allottee Name 

**Allottee Name**

**Email**

**Mobile**

**NID Number**

**Govt Department**

**Designation**

**Permanent Address**

**Division**

**District Name**

**Thana Name**

Submit Back

You can search a specific circle, division, subdivision, quarter or building to check or edit. To do this, select the circle name, division name, subdivision name, quarter name and district name from the drop down box and then click Search option.

**Allottee Information**

+ Add new allottee

**Circle**  **Division**  **Sub Division**  **Quarters**  **Building** 

Search

#SL	Allottee Name	Email	Phone	Govt.Department	Designation	NID	Permanent Address	Division	District	Thana	Edit
1	Julhas Miah										<span style="background-color: #2980b9; color: white; padding: 2px 5px; border-radius: 3px;">Edit</span>
2	Abul Hasem										<span style="background-color: #2980b9; color: white; padding: 2px 5px; border-radius: 3px;">Edit</span>
3	Md. Ful Miah										<span style="background-color: #2980b9; color: white; padding: 2px 5px; border-radius: 3px;">Edit</span>
4	Monoara Begum										<span style="background-color: #2980b9; color: white; padding: 2px 5px; border-radius: 3px;">Edit</span>
5	Shekh Muzibur Rahman Montu										<span style="background-color: #2980b9; color: white; padding: 2px 5px; border-radius: 3px;">Edit</span>
6	Md. Mokhlesur Rahmahn										<span style="background-color: #2980b9; color: white; padding: 2px 5px; border-radius: 3px;">Edit</span>
7	Mostofa Kamal Mojumder										<span style="background-color: #2980b9; color: white; padding: 2px 5px; border-radius: 3px;">Edit</span>

If you want to edit any allottee's name or other information, then click the edit option, edit it and update it.

**Allottee Information**

[+ Add new allottee](#)

Circle: Choose Circle    Division: Choose Circle First    Sub Division: Choose Division First    Quarters: Select Sub Division First    Building: Select Quarter First    [Search](#)

#SL	Allottee Name	Email	Phone	Govt.Department	Designation	NID	Permanent Address	Division	District	Thana	Edit
1	Julhas Miah										<a href="#">Edit</a>
2	Abul Hasem										<a href="#">Edit</a>
3	Md. Ful Miah										<a href="#">Edit</a>
4	Monoara Begum										<a href="#">Edit</a>
5	Shekh Muzibur Rahman Montu										<a href="#">Edit</a>
6	Md. Mokhlesur Rahmahn										<a href="#">Edit</a>
7	Mostofa Kamal Mojumder										<a href="#">Edit</a>

## Apartment

From this section, you can add or edit apartment's name for the users. Click + Add new apartment option to add new.

[+ Add new Apartment](#)

Circle: Choose Circle    Division: Select Circle First    Sub Division: Choose Division First    Quarters: Select Sub Division F...    Building: Select Quarter First    [Search](#)

#SL	Quarters Name	Building Name	Apartment Name	Assign Allottee	Edit
1	Elephant Road Medical College Staff Quarter,	Old Building No-01	01/A	Julhas Miah	<a href="#">Edit</a>
2	Elephant Road Medical College Staff Quarter,	Old Building No-01	01/B	Abul Hasem	<a href="#">Edit</a>
3	Elephant Road Medical College Staff Quarter,	Old Building No-01	01/C	Md. Ful Miah	<a href="#">Edit</a>
4	Elephant Road Medical College Staff Quarter,	Old Building No-01	01/D	Monoara Begum	<a href="#">Edit</a>
5	Elephant Road Medical College Staff Quarter,	Old Building No-01	01/E	Shekh Muzibur Rahman Montu	<a href="#">Edit</a>
6	Elephant Road Medical College Staff Quarter,	Old Building No-01	01/F	Md. Mokhlesur Rahmahn	<a href="#">Edit</a>
7	Elephant Road Medical College Staff Quarter,	Old Building No-01	01/G	Mostofa Kamal Mojumder	<a href="#">Edit</a>

A new box will be come. Then input here the new apartment's name and other information and then submit it

**Add Apartment**

Apartment Name:

Allottee Name:

Quarters Name:

Building:

[Submit](#) [Back](#)

You can search a specific circle, division, subdivision, quarter or building to check or edit. To do this, select the circle name, division name, subdivision name, quarter name and district name from the drop down box and then click Search option.

[+ Add new Apartment](#)

Circle Division Sub Division Quarters Building

Choose Circle  Select Circle First  Choose Division First  Select Sub Division F...  Select Quarter First  [Search](#)

#SL	Quarters Name	Building Name	Apartment Name	Assign Allottee	Edit
1	Elephant Road Medical College Staff Quarter,	Old Building No-01	01/A	Julhas Miah	<a href="#">Edit</a>
2	Elephant Road Medical College Staff Quarter,	Old Building No-01	01/B	Abul Hasem	<a href="#">Edit</a>
3	Elephant Road Medical College Staff Quarter,	Old Building No-01	01/C	Md. Ful Miah	<a href="#">Edit</a>
4	Elephant Road Medical College Staff Quarter,	Old Building No-01	01/D	Monoara Begum	<a href="#">Edit</a>
5	Elephant Road Medical College Staff Quarter,	Old Building No-01	01/E	Shekh Muzibur Rahman Montu	<a href="#">Edit</a>
6	Elephant Road Medical College Staff Quarter,	Old Building No-01	01/F	Md. Mokhlesur Rahmahn	<a href="#">Edit</a>
7	Elephant Road Medical College Staff Quarter,	Old Building No-01	01/G	Mostofa Kamal Mojumder	<a href="#">Edit</a>

If you want to edit any apartment's name or other information, then click the edit option, edit it and update it.

[+ Add new Apartment](#)

Circle  Division  Sub Division  Quarters  Building  [Search](#)

#SL	Quarters Name	Building Name	Apartment Name	Assign Allottee	Edit
1	Elephant Road Medical College Staff Quarter,	Old Building No-01	01/A	Julhas Miah	<a href="#">Edit</a>
2	Elephant Road Medical College Staff Quarter,	Old Building No-01	01/B	Abul Hasem	<a href="#">Edit</a>
3	Elephant Road Medical College Staff Quarter,	Old Building No-01	01/C	Md. Ful Miah	<a href="#">Edit</a>
4	Elephant Road Medical College Staff Quarter,	Old Building No-01	01/D	Monoara Begum	<a href="#">Edit</a>
5	Elephant Road Medical College Staff Quarter,	Old Building No-01	01/E	Shekh Muzibur Rahman Montu	<a href="#">Edit</a>
6	Elephant Road Medical College Staff Quarter,	Old Building No-01	01/F	Md. Mokhlesur Rahmahn	<a href="#">Edit</a>
7	Elephant Road Medical College Staff Quarter,	Old Building No-01	01/G	Mostofa Kamal Mojumder	<a href="#">Edit</a>

## Complain

From this section, you can check and update the status of complaints of the users. Fill up the boxes as you want to check complaints and click search option.

Complain Information

[Search Show/Hide](#)

Start date: YYYY-MM-DD End date: YYYY-MM-DD Status: Choose Status Circle: Choose Circle Division: Choose Circle First Sub Division: Choose Division First

Quarters: Select Sub Division First Building: Select Building [Search](#) [Refresh](#)

#SL	ID	Complainee	Address	Problems	Status	Submitted Date	Update
1	0716-001	Morshed	Building: (House-42, Tejkunipara, Tejgaon) / Type: D-Type ( 1000 Sft ) -01/A, Quarters:Walk-up Government Staff Quarter	Door / Window Problem	New	2016-07-13	<a href="#">Update</a>
2	0716-002	Abcd	Building: (House-42, Tejkunipara, Tejgaon) / Type: D-Type ( 1000 Sft ) -01/A, Quarters:Walk-up Government Staff Quarter	Door / Window Problem	New	2018-09-13	<a href="#">Update</a>
3	0716-003	MD Motakabbir Morshed	Building: (House-42, Tejkunipara, Tejgaon) / Type: D-Type ( 1000 Sft ) -01/A, Quarters:Walk-up Government Staff Quarter	Door / Window Problem	Completed	2018-09-13	

If you want to update the status of any complaint, then click the update option.

Complain Information

[Q Search Show/Hide](#)

Start date	End date	Status	Circle	Division	Sub Division
YYYY-MM-DD	YYYY-MM-DD	Choose Status	Choose Circle	Choose Circle First	Choose Division First

**Quarters**      **Building**

Select Sub Division First      Select Building      [Q Search](#)      [Refresh](#)

#SL	ID	Complainee	Address	Problems	Status	Submitted Date	Update
1	0716-001	Morshed	Building: (House-42, Tejkunipara, Tejgaon) / Type: D-Type ( 1000 Sft )-01/A, Quarters:Walk-up Government Staff Quarter	Door / Window Problem	New	2016-07-13	<a href="#">Update</a>
2	0716-002	Abcd	Building: (House-42, Tejkunipara, Tejgaon) / Type: D-Type ( 1000 Sft )-01/A, Quarters:Walk-up Government Staff Quarter	Door / Window Problem	New	2018-09-13	<a href="#">Update</a>
3	0716-003	MD Motakabbir Morshed	Building: (House-42, Tejkunipara, Tejgaon) / Type: D-Type ( 1000 Sft )-01/A, Quarters:Walk-up Government Staff Quarter	Door / Window Problem	Completed	2018-09-13	

A new box will be come. Here you will see all the information of the complaint. From here you can update the status of complaint. Select the status from the Status option, write any comment about the complaint and click update.

Complain details

Name	Allottee Name
Mobile No	01714522292
Address	Flat No:01/A, Building No: (House-42, Tejkunipara, Tejgaon), Building type: D-Type ( 1000 Sft ), Quarters Name: Walk-up Government Staff Quarter, District: Dhaka
Complain date	2018-10-03
Complain	gtgt
Status	New

SL	Problem Desc	Status	Comments
1	Door / Window Problem	<div style="border: 1px solid black; padding: 2px;"><ul style="list-style-type: none"><li>New</li><li><b>New</b></li><li>In Progress</li><li>Completed</li><li>Cancelled</li></ul></div> <div style="text-align: right;"><a href="#">Update</a>      <a href="#">Back</a></div>	<div style="border: 1px solid gray; height: 40px; width: 100%;"></div>

# All Reports

## Complaint Reports

From this section, you can check, download or print any kinds of complaints reports. Fill up all the boxes and click Show Report to check the reports, click Excel to download this reports as Excel file and click print to print this report.

Search Show /Hide

Start date	End date	Status	Circle	Division	Subdivision
YYYY-MM-DD	YYYY-MM-DD	Choose Status	Choose Circle	Choose Circle First	Choose Division First

Quarter	Building
Select Subdivision First	Select Building

Show Report Excel Print

**Government of the People's Republic of Bangladesh**  
**Office of the Executive Engineer**  
**Public Works Department**  
 Complaint Report Information  
 Print date :08 Dec 2018

SL	Task No	Complainer	Address	Problem Description	Status	Submitted Date	Completed Date	SDE Civil Mobile	SDE E/M Mobile
1	1118-001	Md. A. Al Faisal Mia	Building: B-1/ Type: E-101, Quarter: Motijheel colony (Ideal Zone ), Subdivision: MOTIJHIL SUBDIVISION 1, Division:	new problem	Completed	2018-11-04	2018-11-04	Mobile : 01517037914 Name : SDE, Abdush, Email : abc@gmail.com	01517037914 Name :Md. Abdush, Email : abc@gmail.com

**Print**

Total: 3 pages

Save Cancel

---

Destination Save as PDF

Change...

---

Pages All

12/8/2018

**Government of the People's Republic of Bangladesh**  
**Office of the Executive Engineer**  
**Public Works Department**  
 Complaint Report Information  
 Print date :08 Dec 2018

SL	Task No	Complainer	Address	Problem Description	Status	Submitted Date	Completed Date	SDE Civil Mobile	SDE E/M Mobile
1	1118-001	Md. A. Al Faisal Mia	Building: B-1/ Type: E-101, Quarter: Motijheel colony (Ideal Zone ), Subdivision: MOTIJHIL SUBDIVISION 1, Division:	new problem	Completed	2018-11-04	2018-11-04	Mobile : Name : SDE, Abdush, Email : abc@gmail.com	Name :Md. Abdush, Email : abc@gmail.com

## Report of User

From this section, you can check, download or print any kinds of users reports. Fill up all the boxes and click Submit to check the users reports, click Excel to download this reports as Excel file and click print to print this report.

### User Report Information

Name	Mobile	Username	User Role
			Choose Role
User Status	Department	Designation	Circle
Choose Status	Choose Dept.	Choose Designation	Choose Circle
Division	Subdivision		
Choose Circle First	Choose Division First	<input type="button" value="Show Report"/> <input type="button" value="Excel"/> <input type="button" value="Print"/>	

**Government of the People's Republic of Bangladesh**  
**Office of the Executive Engineer**  
**Public Works Department**  
 User Report Information  
 Print date :08 Dec 2018

SL	Name	Mobile	Email	Username	User Role	Status	Circle	Division	Subdivision	Designation	Department
1	Administrator		abcd@gmail.com	Admin	Administrator	Active				Admin	

Print

Total: 12 pages

Destination  Save as PDF

Pages

All

e.g. 1-5, 8, 11-13

Margins

Options

Headers and footers

Background graphics

12/8/2018

**Government of the People's Republic of Bangladesh**  
**Office of the Executive Engineer**  
**Public Works Department**  
 User Report Information  
 Print date :08 Dec 2018

SL	Name	Mobile	Email	Username	User Role	Status	Circle	Division	Subdivision	Designation	Department
1	Administrator	0181321	abcd@gmail.com	Admin	Administrator	Active				Admin	
2	MIS Cell	015585	abcd@gmail.com	mis	Administrator	Active				Admin	Electrical
3	Super Administrator Admin	0170000000	info@nextech.com.bd	Sadmin	Administrator	Active	Dhaka Circle-4			Admin	
4	SDE SDE	0000000000	sde	sde	Administrator	Active				Admin	Civil
5	SDE em	0000000000	sde_em	sde_em	Administrator	Active				Admin	Civil
6	kul verma	9218733084	prvzone@gmail.com	admin1	Administrator	Active				Admin	Civil
7	Naycem Sarker	0184253	support1@nextech.com.bd	NaycemQA	Administrator	Active				Admin	
8	SDE 1		abc@gmail.com	sde1	SDE	Active	Dhaka Motihil division		MOTIJHILL SUBDIVISION 1	SDE Civil	Civil
9	Mohammad Faysal Rahman		faysal_sahid21@yahoo.com	faysal	SDE	Active	Dhaka Circle-1		PWD, Dhaka Division-1		
10	Dhaka Sub division 7		sde7_mjhl@pwd.gov.bd	sde7	SDE	Active	Dhaka Circle-4	Dhaka Motihil division	Sub-Division 7	SDE Civil	Civil
11	sde civil civil		0000000000	sde_mis	SDE	Active	Dhaka Circle-1		PWD, Dhaka Division-1	SDE Civil	Civil
12	SDE 2	017		sde2	SDE	Active	Dhaka Circle-4	Dhaka Motihil division	MOTIJHILL SUBDIVISION 2	SDE Civil	Civil
13	Abu Shams Kaiser Chowdhury	0177	01882115129	sde_anz	SDE	Active	Dhaka Circle-4	Dhaka Azimpur PWD Division	Azimpur Sub-Division-1, Dhaka	SDE Civil	Civil
14	Nurul Hoque	0188		sde_pilkhana	SDE	Active	Dhaka Circle-4			SDE Civil	Civil
15	S.M. GOLAM QUADER	0171		sde_amz	SDE	Active	Dhaka Circle-4			SDE Civil	Civil

1/12

## Report of Quarter

From this section, you can check, download or print any kinds of quarters reports. Fill up all the boxes and click Submit to check the quarters reports, click Excel to download this reports as Excel file and click print to print this report.

Quarter Report Information

Circle	Division	Subdivision	Quarter
<input type="text" value="Choose Circle"/>	<input type="text" value="Choose Circle First"/>	<input type="text" value="Choose Division First"/>	<input type="text" value="Select Subdivision First"/>

**Building**

**Government of the People's Republic of Bangladesh**  
**Office of the Executive Engineer**  
**Public Works Department**  
**Quarter Report Information**  
**Print date :08 Dec 2018**

SL	Quarter Name	Division	Subdivision
1	Darussalam	Mirpur PWD Division, Dhaka	Mirpur PWD Sub-Division-3, Dhaka
2	Walk-up Government Staff Quarter	Mirpur PWD Division, Dhaka	Mirpur PWD Sub-Division-1, Dhaka
3	paikpara govt. wakap colony	Mirpur PWD Division, Dhaka	Mirpur PWD Sub-Division-2, Dhaka
4	Staf quarter	Mirpur PWD Division, Dhaka	Mirpur PWD Sub-Division-2, Dhaka
5	Tinshed quarter of Public Works employees behind D-Type 70 12 units -340 sqft	Mirpur PWD Division, Dhaka	Mirpur PWD Sub-Division-2, Dhaka
6		Dhaka PWD Division-2	Dhanmondi PWD Sub-Division, Dhaka

## Report of Building

From this section, you can check, download or print any kinds of buildings reports. Fill up all the boxes and click Submit to check the buildings reports, click Excel to download this reports as Excel file and click print to print this report.

### Building Information

🔍 Search Show /Hide

Division	Subdivision	Quarter	Building
Choose Division ▼	Choose Division First ▼	Select Subdivision First ▼	Select Quarter First ▼

Show Report
Excel
Print

**Government of the People's Republic of Bangladesh**  
**Office of the Executive Engineer**  
**Public Works Department**  
**Building Report Information**  
 Print date :08 Dec 2018

SL	Division Name	Subdivision	Quarter	Building	Building Type	Total Apartments
1	Dhaka PWD Division-2	Dhanmondi PWD Sub-Division, Dhaka	Tejgaon	(House-42, Tejgunipara, Tejgaon)	2-storied Building	2
2	Dhaka PWD Division-2	Dhanmondi PWD Sub-Division, Dhaka	Dhanmondi Area	(Road No-11/A, House No-90)		2
3	Dhaka PWD Division-2	Dhanmondi PWD Sub-Division, Dhaka	Dhanmondi Area	(Road No-2, House No-12)	Banglow	1
4	Dhaka PWD Division-2	Dhanmondi PWD Sub-Division,	Dhanmondi Area	(Road No-2, House No-14/A)	Banglow	1

## Report of Allottee

From this section, you can check, download or print any kinds of allottee reports. Fill up all the boxes and click Submit to check the allottee reports, click Excel to download this reports as Excel file and click print to print this report.

### Allottee Report Information

🔍 Search Show /Hide

Division	Subdivision	Quarter	Building
Dhaka City PWD Division ▼	Choose SubDivision ▼	Select Subdivision First ▼	Select Quarter First ▼

Search
Print
Excel

**Government of the People's Republic of Bangladesh**  
**Office of the Executive Engineer**  
**Public Works Department**  
**Allottee Report Information**  
 Division Name: Dhaka City PWD Division  
 Print date 08 Dec 2018 12:05 pm

Division Name: Dhaka City PWD Division,  
 Subdivision Name: Bangabhaban PWD Sub-Division,  
 Quarter Name: Bangabhaban PWD Sub-Division Quarter,  
 Building No: rosa

SL	Apartment No	Allottee Name	Allottee Mobile	Allottee Email
1	1	Md. Monir Hossain	01912578923	
2	2	Md. Saiful Islam	01550552230	

## Summary Report

From this section, you can check, download or print any kinds of summary report. Fill up all the boxes and click Submit to check the summary report, click Excel to download this reports as Excel file and click print to print this report.

### Summary Report Information

**Circle** **Division** **Subdivision**

Choose Circle Choose Circle First Choose Division First

[Search](#) [Excel](#) [Print](#)

**Government of the People's Republic of Bangladesh**

**Office of the Executive Engineer**

**Public Works Department**

Summary Report of Quarter

Print date :08 Dec 2018 , 06:06:24 pm

Circle Name: Savar Circle				
Division Name: Mirpur PWD Division, Dhaka,				
Subdivision Name Name:Mirpur PWD Sub-Division-3, Dhaka				
SL	Quarter Name	Total Buildings	Total Apartments	Total Allottees
1	Darussalam	3	1098	1098
Circle Name: Savar Circle				
Division Name: Mirpur PWD Division, Dhaka,				
Subdivision Name Name:Mirpur PWD Sub-Division-1, Dhaka				
SL	Quarter Name	Total Buildings	Total Apartments	Total Allottees
1	Walk-up Government Staff Quarter	15	160	160
Circle Name: Savar Circle				

## Task Report

From this section, you can check, download or print any kinds of task report. Fill up all the boxes and click Submit to check the task report, click Excel to download this reports as Excel file and click print to print this report. This report show all information of Task.

### Task Report Information

[Search Show /Hide](#)

**Start date** **End date** **Status** **Circle** **Division** **Subdivision**

Choose Status Choose Circle Choose Circle First Choose Division First

[Show Report](#) [Excel](#) [Print](#)

**Government of the People's Republic of Bangladesh**

**Office of the Executive Engineer**

**Public Works Department**

Complaint Report Information

Print date :08 Dec 2018

SL	Task No	Circle	Division	Subdivision	District	Quarter	Building	Comment	Complainer	Mobile	Status	Department	Submitted Date	Updated Date	Updated By
1	1118-006	Dhaka Circle-4	Dhaka Motijhil division	MOTIJHIL SUBDIVISION 1	Dhaka	Motijheel colony (Ideal Zone )		Door / Window Problem	Md. A. Al Faisal Mia	01673548485	New	CIVIL	05-11-2018		
2	1118-007	Dhaka Circle-4	Dhaka Motijhil	MOTIJHIL SUBDIVISION 1	Dhaka	Motijheel colony (Ideal Zone )		Fan repair or	Md. A. Al Faisal Mia	01771882876	New	EM	05-11-2018		

## Profile:

This section is for only login user to change his profile as First Name, Last name, Email, mobile with picture and can change the current password.

User Profile



**Super Administrator**  
Admin

---

About Me

**Location**  
Dhaka Circle-4, ,

---

Member Since : 07/02/2017

Settings Picture

**First Name**

**Last Name**

**Email**

**Mobile**

**New Password**

**Confirm Password**






### Access to Information Idea( A2I)'s Page Layout [Web Page]

**Login:** This Login page is for A2I survey Login, To reach the Survey panel dashboard this page is used for user authentication.

## PWD -Survey

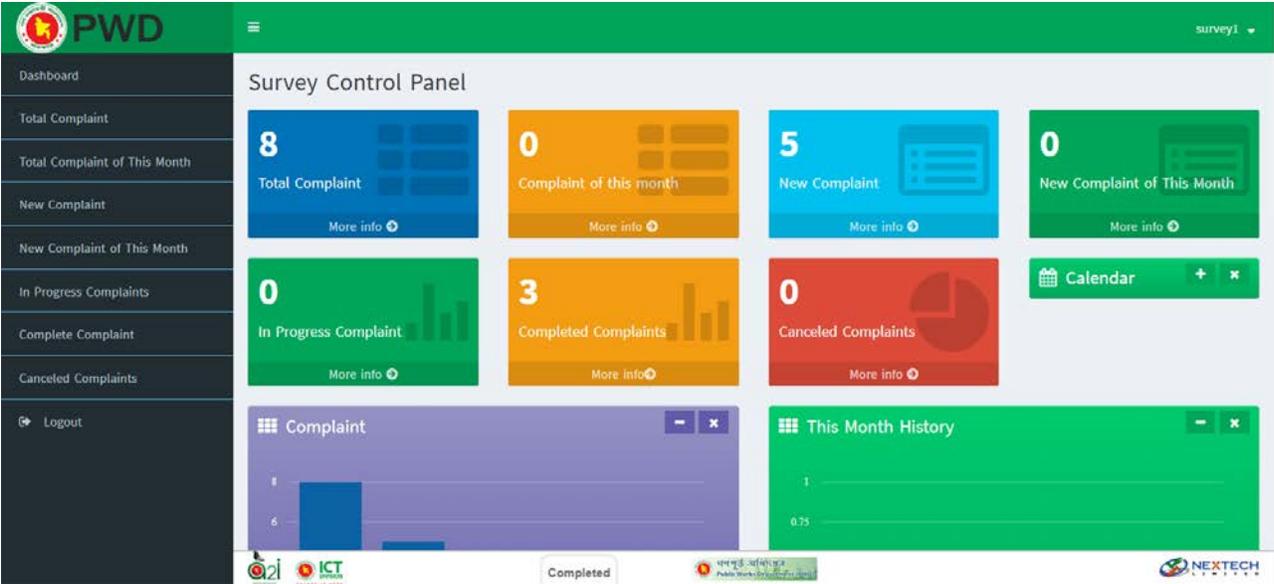
Survey Sign in Panel





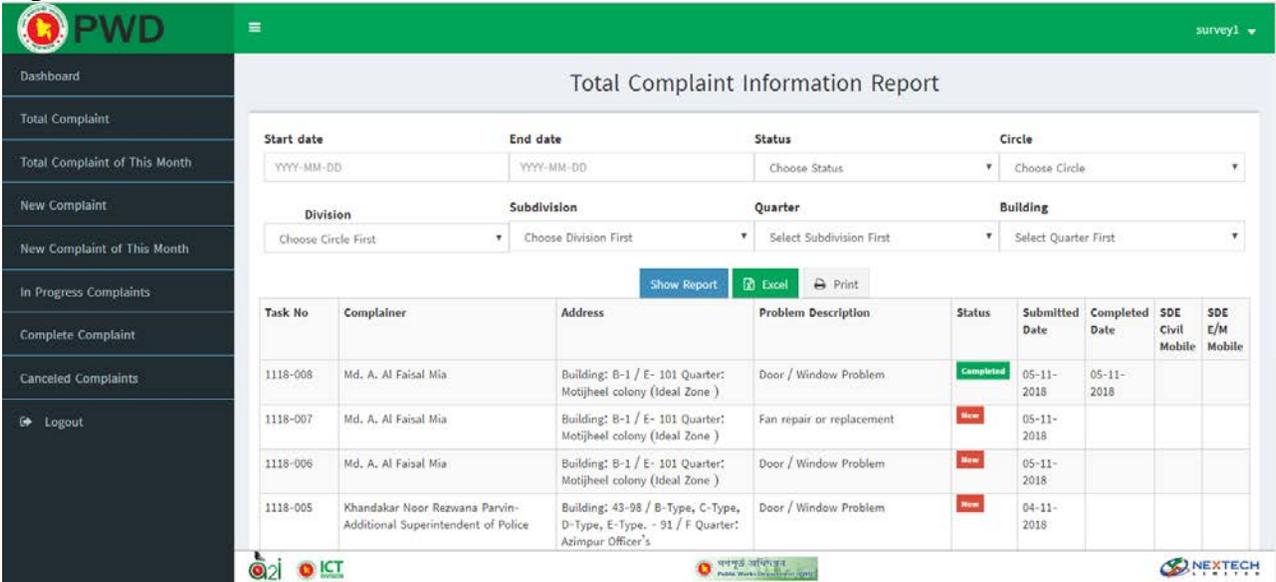
[Home](#)

**Dashboard:** This dashboard is for Survey Panel /A2I , This page is included Total Complaint, Complaint of this month, Total New Complaint, Total New complaint of this month, Total InProgress Complaint, Total completed complaint, Total Canceled complaints, Calendar, Complaint by chart, Complaint of this month chart, Complaint by month with chart and complaint by year with chart.



**Total Complaint Information Report:**

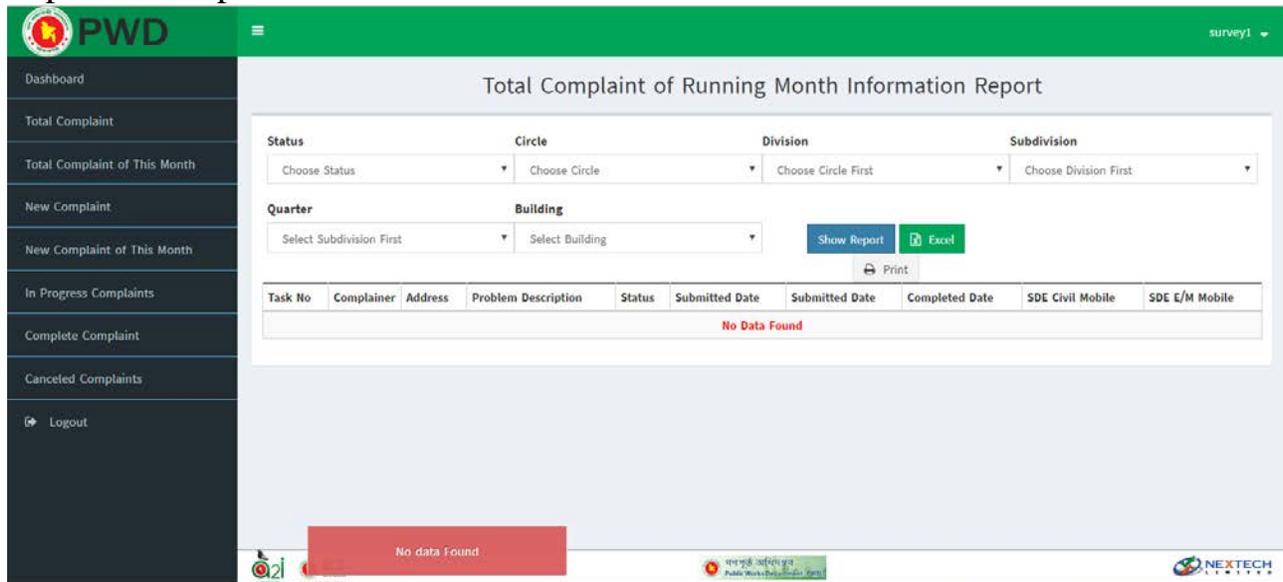
From this section, you can check, download or print any kinds of total complaint information reports. Fill up all the boxes and click Submit to check the following reports, click Excel to download this reports as Excel file and click print to print this report.



**Total Complaint of running month Information Report:**

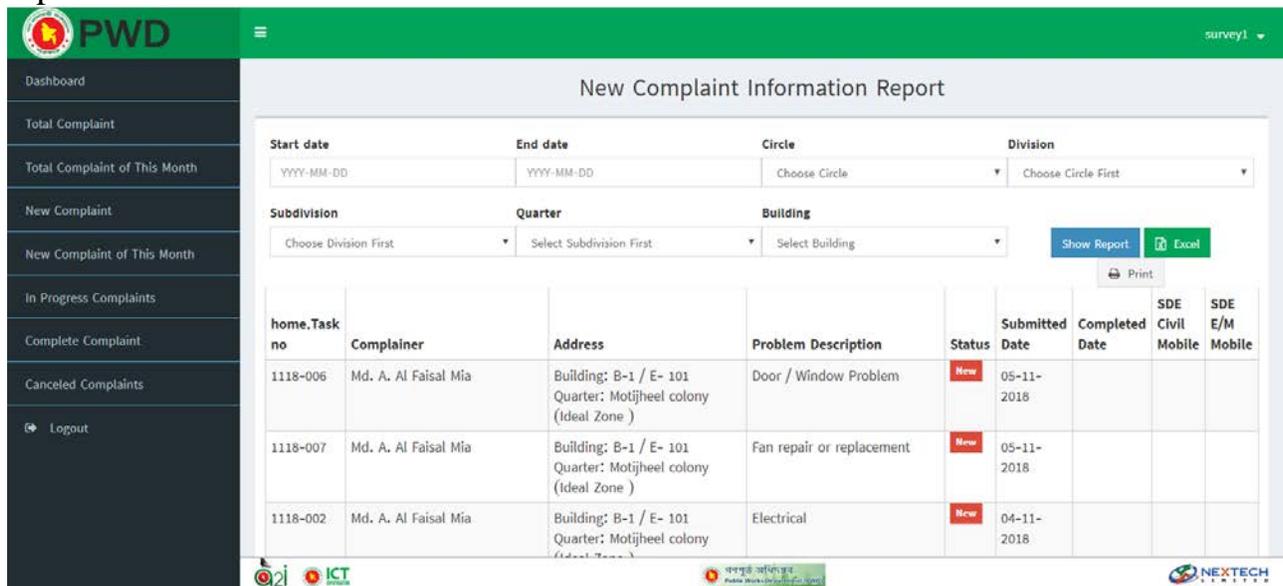
From this section, you can check, download or print any kinds of total complaint of

running month information reports. Fill up all the boxes and click Submit to check the following reports, click Excel to download this report as Excel file and click print to print this report.



**Total New Complaint Information Report:**

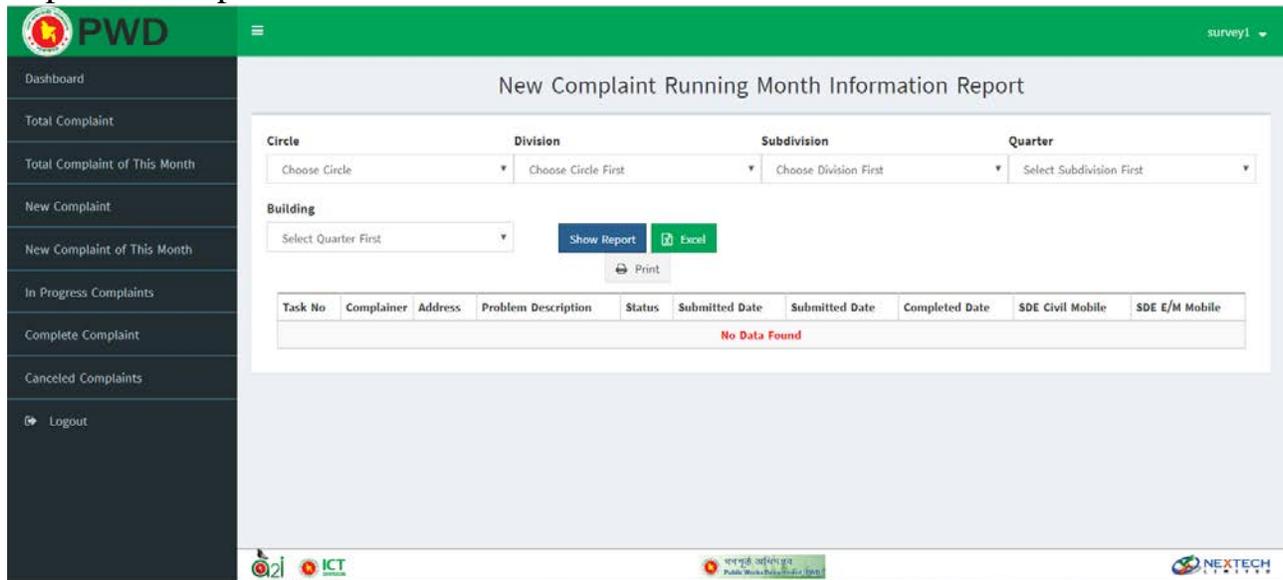
From this section, you can check, download or print any kinds of total new complaint information reports. Fill up all the boxes and click Submit to check the following reports, click Excel to download this report as Excel file and click print to print this report.



**Total New Complaint running month Information Report:**

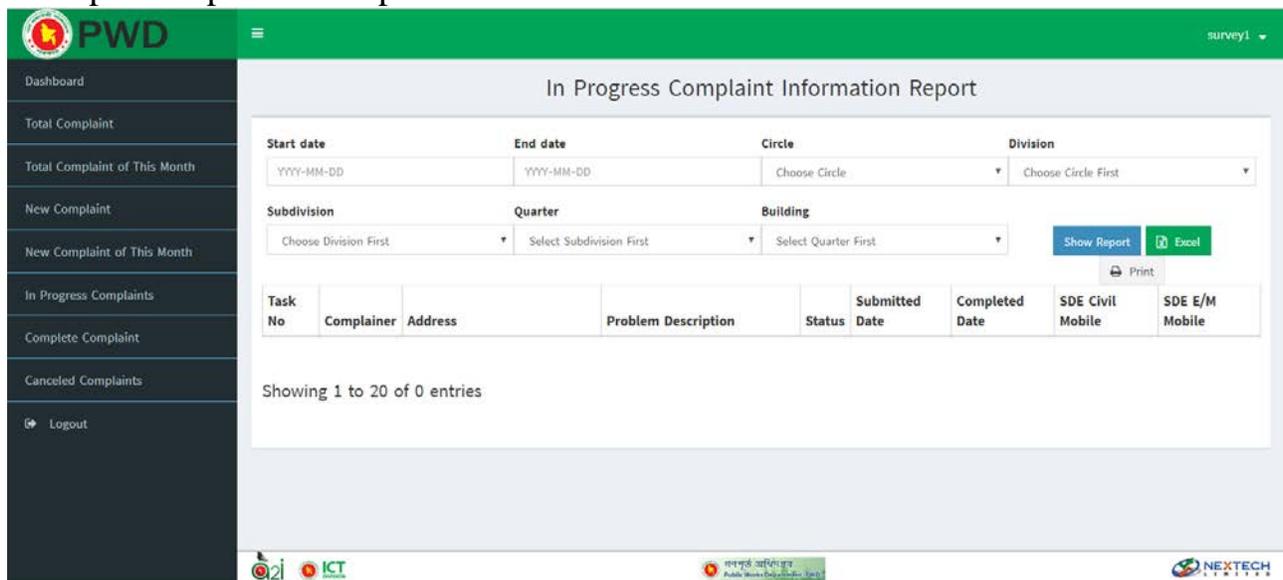
From this section, you can check, download or print any kinds of total new complaint

running month information reports. Fill up all the boxes and click Submit to check the following reports, click Excel to download this report as Excel file and click print to print this report.



**Total In Progress Complaint Information Report:**

From this section, you can check, download or print any kinds of total in progress complaint running month information reports. Fill up all the boxes and click Submit to check the following reports, click Excel to download this report as Excel file and click print to print this report.



**Total Completed Complaint Information Report:**

From this section, you can check, download or print any kinds of total completed

complaint information reports. Fill up all the boxes and click Submit to check the following reports, click Excel to download this report as Excel file and click print to print this report.

**Complete Complaint Information Report**

Start date: YYYY-MM-DD, End date: YYYY-MM-DD, Circle: Choose Circle, Division: Choose Circle First

Subdivision: Choose Division First, Quarter: Select Subdivision First, Building: Select Building

Buttons: Show Report, Excel, Print

Task No	Complainer	Address	Problem Description	Status	Submitted Date	Completed Date	SDE Civil Mobile	SDE E/M Mobile
1118-008	Md. A. Al Faisal Mia	Building: B-1 / E- 101 Quarter: Motijheel colony (Ideal Zone )	Door / Window Problem	Completed	05-11-2018	2018-11-05		
1118-001	Md. A. Al Faisal Mia	Building: B-1 / E- 101 Quarter: Motijheel colony (Ideal Zone )	new problem	Completed	04-11-2018	2018-11-04		
1118-004	Khandakar Noor Rezwana Parvin-Additional	Building: 43-98 / B-Type, C-Type, D-Type, E-Type. - 91 /	Door / Window Problem Electrical Switch and circuit	Completed	04-11-2018	2018-11-04		

**Total Canceled Complaint Information Report:**

From this section, you can check, download or print any kinds of total canceled complaint information reports. Fill up all the boxes and click Submit to check the following reports, click Excel to download this report as Excel file and click print to print this report.

**Cancel Complaint Information Report**

Start date: YYYY-MM-DD, End date: YYYY-MM-DD, Circle: Choose Circle, Division: Choose Circle First

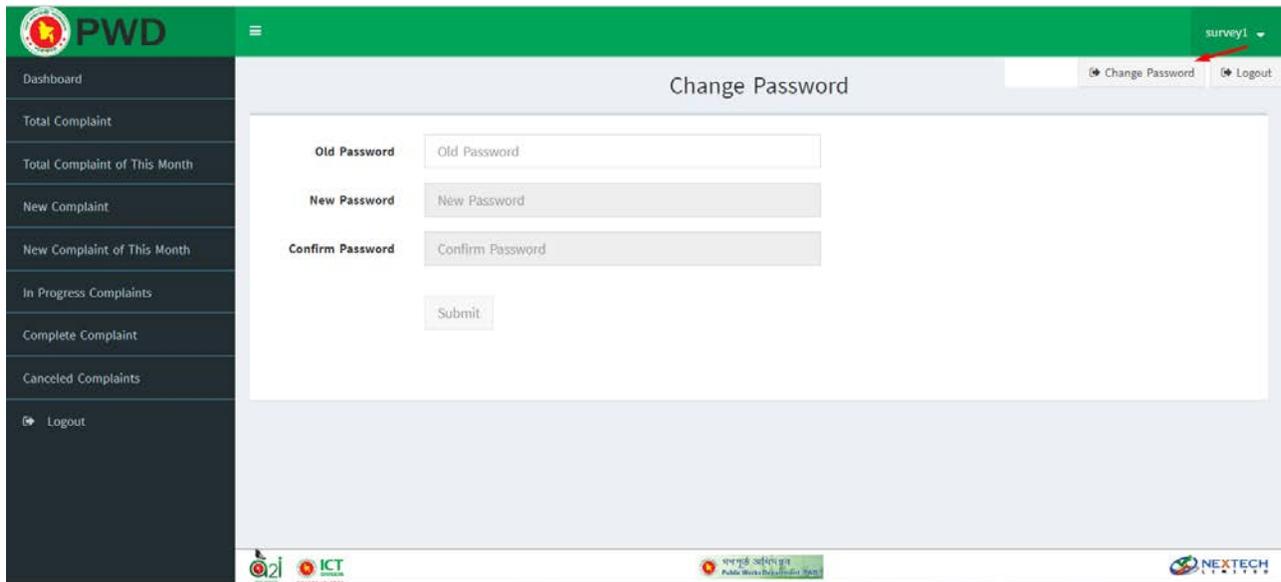
Subdivision: Choose Division First, Quarter: Select Subdivision First, Building: Select Building

Buttons: Show Report, Excel, Print

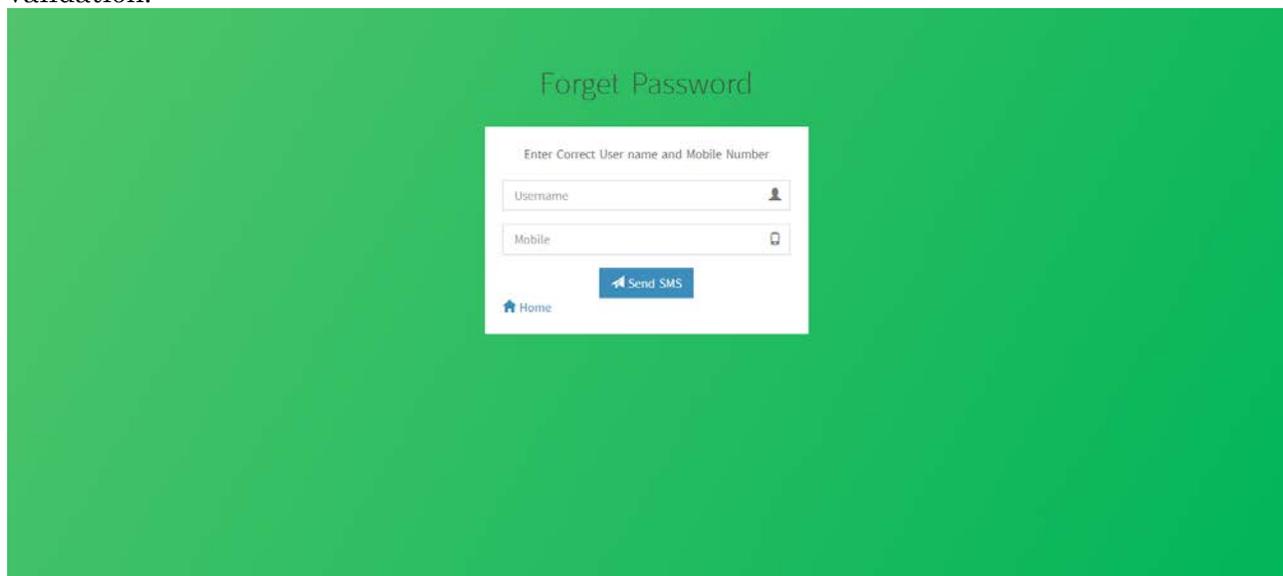
Task No	Complainer	Address	Problem Description	Status	Submitted Date	Completed Date	SDE Civil Mobile	SDE E/M Mobile
Showing 1 to 20 of 0 entries								

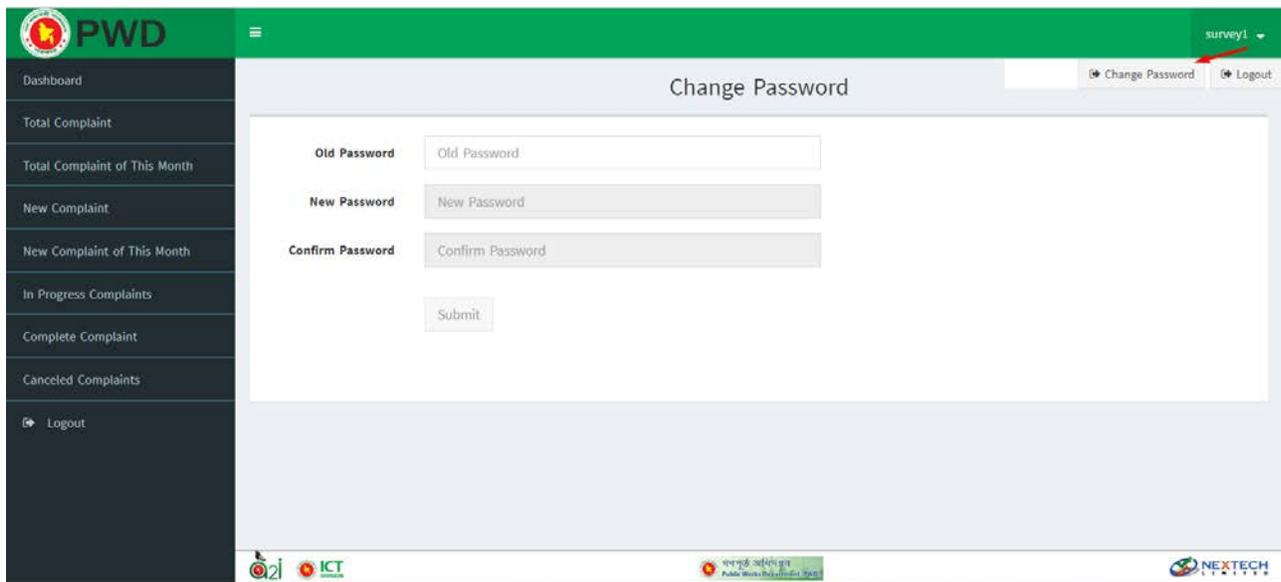
Change Password:

This section is for survey / a2i panel. If the survey user want to change the password , he/she can easily change the password by providing existing password for security reason.



**Forget Password Page:** This page is for getting the existing password recovery using proper validation.

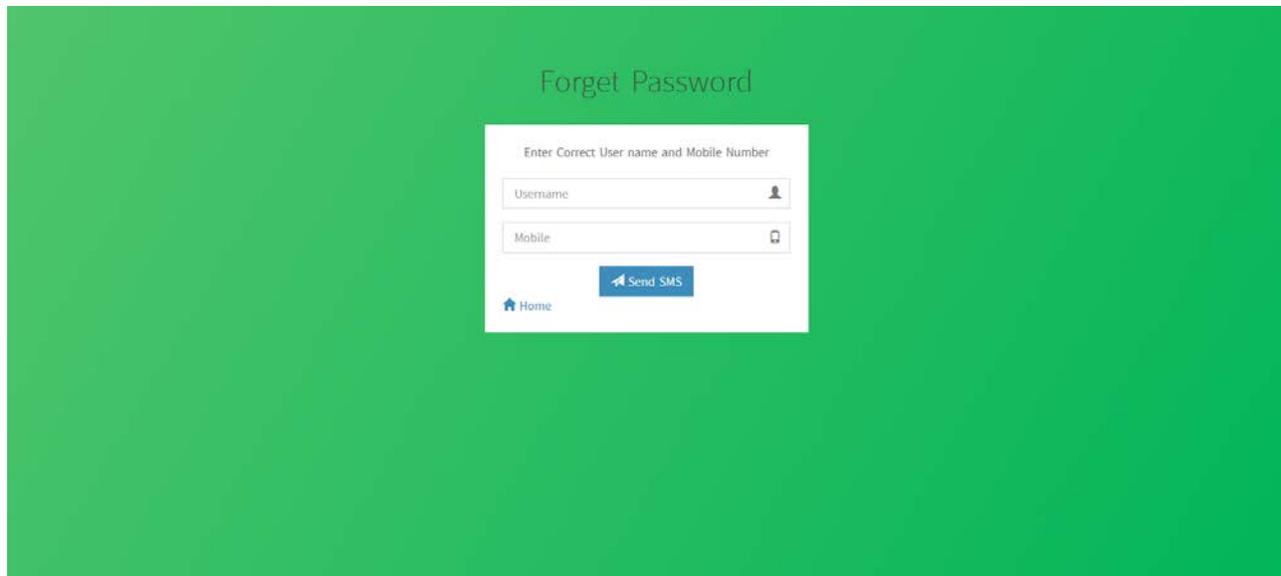




OOOPS!  
404 - THE PAGE CAN'T BE FOUND

GO TO HOMEPAGE

**Forget Password Page:** This page is for getting the existing password recovery using proper validation.



## 6.2. Module interface design

Module design maintains MVC (Model – View - Controller) architecture. View is a UI. Through UI user inputs data which goes to Controller. Controller transfers data into Model. If data is incorrect Model shows error message. Otherwise it processes the request, prepares the result and sends it to the Controller. Finally, Controller transfers generated code into View. The user views the result.

General principle of MVC (Model – View - Controller) architecture can be described with sequential diagram.

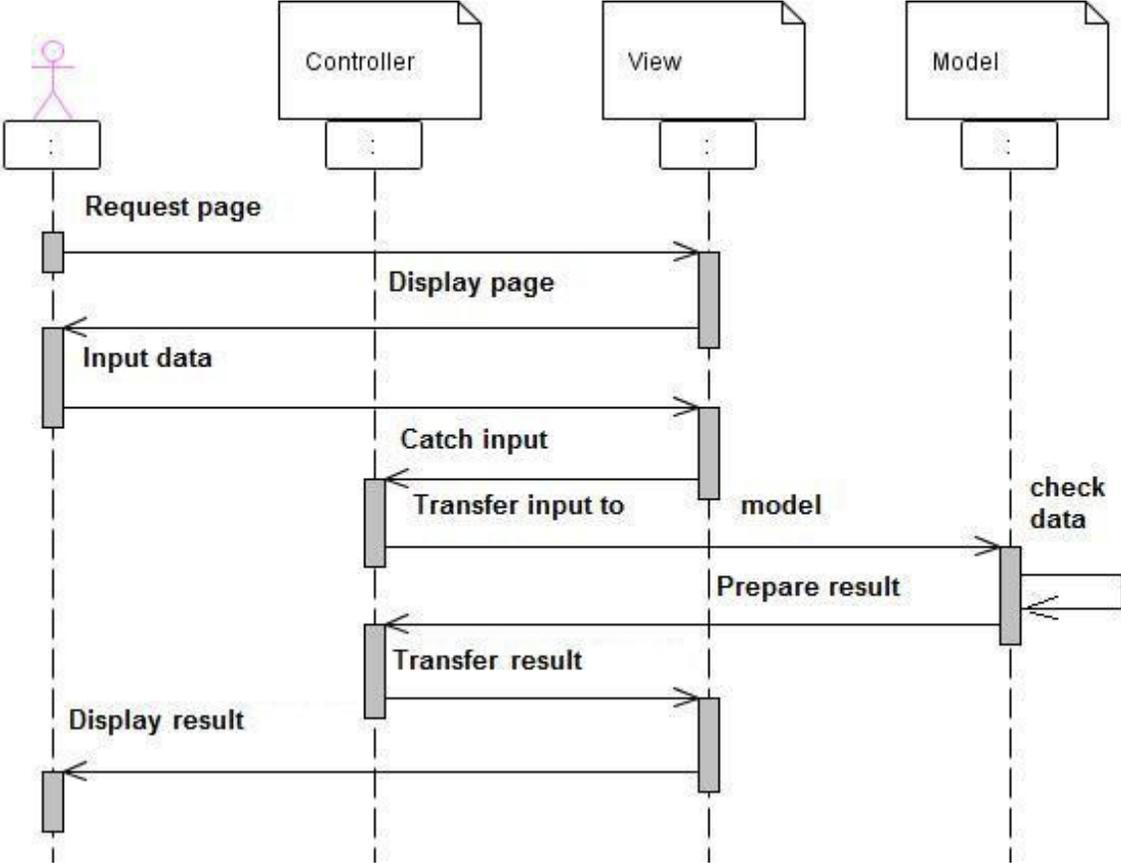


Figure - Sequence diagram of MVC (Model – View - Controller) architecture

The Laravel class have two types of class one is Model Class and the other is Controller Class.

**The Model Class interact with the database table**

**class Allottee extends Model { }**: Used the protected variable of “tbl\_allottee”, used public variable timestamps, used the protected fillable field of selected table variable also used protected hidden field(if any).

**class Apartment extends Model { }**: Used the protected variable of “tbl\_apartment”, used public variable timestamps, used the protected fillable field of selected table variable also used protected hidden field(if any).

**class Building extends Model { }**: Used the protected variable of “tbl\_building”, used public variable timestamps, used the protected fillable field of selected table variable also used protected hidden field(if any).

**class BuildingType extends Model { }**: Used the protected variable of “tbl\_buildingtype”, used public variable timestamps, used the protected fillable field of selected table variable also used protected hidden field(if any).

**class Circle extends Model { }**: Used the protected variable of “tbl\_circle”, used public variable timestamps, used the protected fillable field of selected table variable also used protected hidden field(if any).

**class Department extends Model { }**: Used the protected variable of “tbl\_department”, used public variable timestamps, used the protected fillable field of selected table variable also used protected hidden field(if any).

**class District extends Model { }**: Used the protected variable of “tbl\_district”, used public variable timestamps, used the protected fillable field of selected table variable also used protected hidden field(if any).

**class DistrictGEO extends Model { }**: Used the protected variable of “tbl\_districtgeo”, used public variable timestamps, used the protected fillable field of selected table variable also used protected hidden field(if any).

**class Division extends Model { }**: Used the protected variable of “tbl\_division”, used public variable timestamps, used the protected fillable field of selected table variable also used protected hidden field(if any).

**class DivisionGEO extends Model { }**: Used the protected variable of “tbl\_divisiongeo”, used public variable timestamps, used the protected fillable field of selected table variable also used protected hidden field(if any).

**class GovernmentDepartment extends Model { }**: Used the protected variable of “tbl\_govt\_department”, used public variable timestamps, used the protected fillable field of selected table variable also used protected hidden field(if any).

**class Problem extends Model { }**: Used the protected variable of “tbl\_problemtyp”, used public variable timestamps, used the protected fillable field of selected table variable also used protected hidden field(if any).

**class Quarter extends Model { }**: Used the protected variable of “tbl\_quarters”, used public variable timestamps, used the protected fillable field of selected table variable also used protected hidden field(if any).

**class Rank extends Model{ }**: Used the protected variable of “tbl\_rank”, used public variable timestamps, used the protected fillable field of selected table variable also used protected hidden field(if any).

**class Status extends Model{ }**: Used the protected variable of “tbl\_status”, used public variable timestamps, used the protected fillable field of selected table variable also used protected hidden field(if any).

**class Subdivision extends Model{ }**: Used the protected variable of “tbl\_subdivision”, used public variable timestamps, used the protected fillable field of selected table variable also used protected hidden field(if any).

**class Survey extends Model{ }**: Used the protected variable of “tbl\_survey\_user”, used public variable timestamps, used the protected fillable field of selected table variable also used protected hidden field(if any).

**class TaskReport extends Model{ }**: Used the protected variable of “tbl\_task”, used public variable timestamps, used the protected fillable field of selected table variable also used protected hidden field(if any).

**class TblTask extends Model{ }**: Used the protected variable of “tbl\_task”, used public variable timestamps, used the protected fillable field of selected table variable also used protected hidden field(if any).

**class Thana extends Model { }**: Used the protected variable of “tbl\_thana”, used public variable timestamps, used the protected fillable field of selected table variable also used protected hidden field(if any).

**class User extends Authenticatable { }**: Used the protected variable of “users”, used public variable timestamps, used the protected fillable field of selected table variable also used protected hidden field(if any).

**class designation extends Model{ }**: Used the protected variable of “tbl\_designation”, used public variable timestamps, used the protected fillable field of selected table variable also used protected hidden field(if any).

**class tbl\_trees extends Model{ }**: Used the protected variable of “\_nisl\_tree\_entries”, used public variable timestamps, used the protected fillable field of selected table variable also used protected hidden field(if any).

### The Controller Class interact with the various action using functions

**class AllotteeController extends Controller { }** : Controllers are typically stored in app/controllers directory. This contains several methods to act several actions The methods are given bellow with the functionality.

- public function \_\_construct (): This construct method is used for using several middleware for accessing / permission of various methods in this class.
- index() : This method returns various columns of data from different tables using leftjoin query and redirect the allottee view page.
- insertAllottee () : This method used for insert the allottees' information in the database table.
- updateAllottee (): This method used for update the allottees' information in the database table.
- getAllotteeModal (): This method used for opeing the allottees' form with Modal popup.

**class AllotteeReportController extends Controller { }** : This class Controller is typically used for allottee report related functionality .This contains several methods to act several actions The methods are given bellow with the functionality.

- public function \_\_construct (): This construct method is used for using several middleware for accessing / permission of various methods in this class.
- Index () : This method (default) returns redirect to the view page.
- AllotteeSearch () : This method used for searching the allottees' information from the database table using mysql query.

**class ApartmentController extends Controller { }** : This class Controller is typically used for apratment related functionality .This contains several methods to act several actions The methods are given bellow with the functionality.

- public function \_\_construct (): This construct method is used for using several middleware for accessing / permission of various methods in this class.
- Index () : This method (default) returns redirect to the view page.
- getFlat () : This method used for getting the flat from the flat table information from the database table using mysql query.
- getApartmentModal (): This method used for opeing the apartment form with Modal popup.
- insertApartment () : This method used for insert the apartment information in the database table.

- `updateApartment ( )` : This method used for update the apartment information in the database table.
- `ApartmentSearch ( )` : This method used for searching the apartment information from the database table using MySQL query.

**class BuildingController extends Controller { }** : This class Controller is typically used for Building related functionality .This contains several methods to act several actions The methods are given bellow with the functionality.

- public function `__construct ()`: This construct method is used for using several middleware for accessing / permission of various methods in this class.
- `Index ( )` : This method (default) returns redirect to the view page.
- `getBuildingName ( )`: This method used for getting the building name from the building name table information from the database table using MySQL query.
- `getBuildingType ( )`: This method used for get building type from building type table.
- `getBuilding ( )` : This method used getting the building name from the building name table information from the database table using MySQL query.
- `getBuildingModal ( )` : This method used for opeing the building form with Modal popup.
- `insertBuilding ( )` : This method used for insert the building information in the database table.
- `updateBuilding ( )` : This method used for update the building information in the database table.
- `getBuildingSearch ( )`: This method used for searching the building information from the database table using MySQL query.
- `getBuildingReport ( )`: This method used for report the building information from the database table using MySQL query.

**class BuildingTypeController extends Controller { }** : This class Controller is typically used for BuildingType related functionality .This contains several methods to act several actions The methods are given bellow with the functionality.

- public function `__construct ()`: This construct method is used for using several middleware for accessing / permission of various methods in this class.
- `Index ( )` : This method (default) returns redirect to the view page.
- `getBuildingType ( )`: This method used for getting the building name from the building type name table information from the database table using MySQL query.

- `getBuildingtypeforintro ( )`: This method used for get building type from building type table.
- `getBulildingTypeModal ( )` This method used for opeing the building type form with Modal popup.
- `insertBuildingtype ( )` : This method used for insert the building type information in the database table.
- `updateBuildingtype ( )` : This method used for update the building type information in the database table.

**class CircleController extends Controller { }** : This class Controller is typically used for Circle related functionality .This contains several methods to act several actions The methods are given bellow with the functionality.

- public function `__construct ()`: This construct method is used for using several middleware for accessing / permission of various methods in this class.
- `Index ( )` : This method (default) returns redirect to the view page.
- `getCircleDropdown ( )`: This method used for getting the building name from the building type name table information from the database table using MySQL query.
- `getCircleModal ( )` This method used for opeing the circle form with Modal popup.
- `insertCircle ( )` : This method used for insert the circle information in the database table.
- `updateCircle ( )` : This method used for update the circle information in the database table.

**class ComplaintController Extends Controller { }** : This class Controller is typically used for **Complain** related functionality .This contains several methods to act several actions The methods are given bellow with the functionality.

- public function `__construct ()`: This construct method is used for using several middleware for accessing / permission of various methods in this class.
- `Index ( )` : This method (default) returns redirect to the view page.
- `getComplainSearch ( )`: This method used for searching the building information from the database table using MySQL query
- `getComplainModal ( )` This method used for opeing the complaint form with Modal popup.
- `getTaskShowModal ( )` : This method used for getting the task information from the database table.

- `updateComplain ()` : This method used for update the complain information in the database table.

**class DepartmentController Extends Controller { }** : This class Controller is typically used for Department related functionality .This contains several methods to act several actions The methods are given bellow with the functionality.

- `public function __construct ()`: This construct method is used for using several middleware for accessing / permission of various methods in this class.
- `getDepartment ()` : This method (default) returns getting the department to the view page.

**class DesignationController Extends Controller { }** : This class Controller is typically used for Designation related functionality .This contains several methods to act several actions The methods are given bellow with the functionality.

- `public function __construct ()`: This construct method is used for using several middleware for accessing / permission of various methods in this class.
- `Index ()` : This method (default) returns redirect to the view page.
- `insertDesignation ()`: This method used for insert the designation information in the database table.
- `getDesignationModal ()` This method used for opeing the designation form with Modal popup.
- `updateDesignation ( )` : This method used for update the designation information in the database table.

**class DistrictControllerExtends Controller { }** : This class Controller is typically used for District related functionality .This contains several methods to act several actions The methods are given bellow with the functionality.

- `public function __construct ()`: This construct method is used for using several middleware for accessing / permission of various methods in this class.
- `Index ()` : This method (default) returns redirect to the view page.
- `insertDistrict ()`: This method used for insert the district information in the database table.
- `getDistrictModal ()` : This method used for opeing the district form with Modal popup.
- `updateDistrict ( )` : This method used for update the district information in the database table.
- `getDistrict ( )` : This method used for getting the district for other related task.

**class DistrictGEOController Controller { }** : This class Controller is typically used for DistrictGEO related functionality .This contains several methods to act several actions The methods are given bellow with the functionality.

- public function `__construct ()`: This construct method is used for using several middleware for accessing / permission of various methods in this class.
- `Index ()` : This method (default) returns redirect to the view page.
- `insertDistrictGEO ()`: This method used for insert the districtGEO information in the database table.
- `districtGEOModal ()` : This method used for opeing the districtGEO form with Modal popup.
- `updateDistrictGEO ( )` : This method used for update the districtGEO information in the database table.
- `findDistrictGEO ( )` : This method used for getting the districtGEO for other related task.
- `findDistrictGEOforallottee ( )` : This method used for getting the districtGEO for allottee related task.

**class DivisionController extends Controller { }** : This class Controller is typically used for Division related functionality .This contains several methods to act several actions The methods are given bellow with the functionality.

- public function `__construct ()`: This construct method is used for using several middleware for accessing / permission of various methods in this class.
- `Index ()` : This method (default) returns redirect to the view page.
- `insertDivision ( )`: This method used for insert the division information in the database table.
- `getDivisionModal ()` : This method used for opeing the division form with Modal popup.
- `updateDivision ( )` : This method used for update the division information in the database table.

**class DivisionGEOController extends Controller { }** : This class Controller is typically used for DivisionGEO related functionality .This contains several methods to act several actions The methods are given bellow with the functionality.

- public function `__construct ()`: This construct method is used for using several middleware for accessing / permission of various methods in this class.

- `Index ()` : This method (default) returns redirect to the view page.
- `insertDivisionGEO ( )`: This method used for insert the DivisionGEO information in the database table.
- `divisionGEOModal ( )` : This method used for opening the DivisionGEO form with Modal popup.
- `updateDivisionGEO ( )` : This method used for update the DivisionGEO information in the database table.

**class GovernmentDepartmentController extends Controller { }** : This class Controller is typically used for government department related functionality .This contains several methods to act several actions The methods are given bellow with the functionality.

- public function `__construct ()`: This construct method is used for using several middleware for accessing / permission of various methods in this class.
- `Index ()` : This method (default) returns redirect to the view page.
- `insetGovtDepartment ( )`: This method used for insert the government department information in the database table.
- `govtDepAddModal ( )` : This method used for opening the government department form with Modal popup.
- `updateGovtDept ( )` : This method used for update the government department information in the database table.

**class ProblemController extends Controller { }** : This class Controller is typically used for problem related functionality .This contains several methods to act several actions The methods are given bellow with the functionality.

- public function `__construct ()`: This construct method is used for using several middleware for accessing / permission of various methods in this class.
- `Index ()` : This method (default) returns redirect to the view page.
- `insertProblem ( )`: This method used for insert the problem information in the database table.
- `getProblemTypeModal ( )` : This method used for opening the problem form with Modal popup.
- `updateProblem ( )` : This method used for update the problem information in the database table.
- `getProblemType ( )` : This method used for update the problem information in the database table.

**class ProfileController extends Controller { }** : This class Controller is typically used for problem related functionality .This contains several methods to act several actions The methods are given bellow with the functionality.

- public function `__construct ()`: This construct method is used for using several middleware for accessing / permission of various methods in this class.

- `Index ()` : This method (default) returns redirect to the view page.
- `store ()` : This method used for insert the Profile image information in the database table.
- `insertProfileChange ()` : This method used for insert the Profile information in the database table.

**class QuarterController extends Controller { }** : This class Controller is typically used for Quarter related functionality .This contains several methods to act several actions The methods are given bellow with the functionality.

- public function `__construct ()`: This construct method is used for using several middleware for accessing / permission of various methods in this class.
- `Index ()` : This method (default) returns redirect to the view page.
- `insertQuarter ()` : This method used for insert the quarter information in the database table.
- `insertDesignation ()`: This method used for insert the designation information in the database table.
- `getQuaterModal ()` : This method used for opeing the quarter form with Modal popup.
- `updateQuarter ()` : This method used for update the quarter information in the database table.
- `getQuarters ()` : This method used for update the quarter information in the database table.
- `getQuarterfor ()`: This method used for retrieving the quarter list for other controller purposes.
- `Quarterreport()` : This method used for getting the quarter report .
- `getQuarterSearch ()` : This method used for getting quarter search.
- `QuarterSearch ()` : This method used for getting quarter search for a2i.
- `insertquarterSup ()` : This method used for insert quarter supervisor or SDE for assigned employee.
- `deleteEmp ()` : This method is used for delete SDE form quarter list.

**class RankController extends Controller { }** : This class Controller is typically used for rank related functionality .This contains several methods to act several actions The methods are given bellow with the functionality.

- public function `__construct ()`: This construct method is used for using several middleware for accessing / permission of various methods in this class.
- `Index ()` : This method (default) returns redirect to the view page.
- `insertRank ()` : This method used for insert the rank information in the

database table.

- `getRankModal ()` : This method used for opening the rank form with Modal popup.
- `updateRank ()` : This method used for update the rank information in the database table.

**class ReportController extends Controller {}** : This class Controller is typically used for rank related functionality .This contains several methods to act several actions The methods are given bellow with the functionality.

- public function `__construct ()`: This construct method is used for using several middleware for accessing / permission of various methods in this class.
- `Index ()` : This method (default) returns redirect to the view page.
- `getReportSearch ()`: This method used for searching the report information.

**class StatusController extends Controller {}** : This class Controller is typically used for status related functionality .This contains several methods to act several actions The methods are given bellow with the functionality.

- public function `__construct ()`: This construct method is used for using several middleware for accessing / permission of various methods in this class.
- `Index ()` : This method (default) returns redirect to the view page.
- `insertStatus ()`: This method used for insert the status information in the database table.
- `getStatusModal ()` : This method used for opening the status form with Modal popup.
- `updateStatus ()` : This method used for update the status information in the database table.

**class SubdivisionController extends Controller {}** : This class Controller is typically used for subdivision related functionality .This contains several methods to act several actions The methods are given bellow with the functionality.

- public function `__construct ()`: This construct method is used for using several middleware for accessing / permission of various methods in this class.
- `Index ()` : This method (default) returns redirect to the view page.
- `insertSubdivision ()`: This method used for insert the subdivision information in the database table.
- `getSubdivisionModal ()` : This method used for opening the subdivision form with Modal popup.
- `updateSubdivision ()` : This method used for update the subdivision information in the database table.

**class SummaryReportController extends Controller { }** : This class Controller is typically used for summaryreport related functionality .This contains several methods to act several actions The methods are given bellow with the functionality.

- public function \_\_construct (): This construct method is used for using several middleware for accessing / permission of various methods in this class.
- Index () : This method (default) returns redirect to the view page.
- SummaryReport ( ) : This method used for summary report of various table using MYSQL query.

**class TaskReportController extends Controller { }** : This class Controller is typically used for taskfg related functionality .This contains several methods to act several actions The methods are given bellow with the functionality.

- public function \_\_construct (): This construct method is used for using several middleware for accessing / permission of various methods in this class.
- Index () : This method (default) returns redirect to the view page.
- TaskReportSearch ( ) : This method used for searching task related all information using MYSQL query.

**class ThanaController extends Controller { }** : This class Controller is typically used for thana related functionality .This contains several methods to act several actions The methods are given bellow with the functionality.

- public function \_\_construct (): This construct method is used for using several middleware for accessing / permission of various methods in this class.
- Index () : This method (default) returns redirect to the view page.
- insertThana ( ) : This method used for insert the thana information in the database table.
- getThanaModal ( ) : This method used for opeing the thana form with Modal popup.
- updateThana ( ) : This method used for update the thana information in the database table.
- findThanaforallottee ( ) : This method used for find thana for the allottee information .

**class UserController extends Controller { }** : This class Controller is typically used for user related functionality .This contains several methods to act several actions The methods are given bellow with the functionality.

- public function \_\_construct (): This construct method is used for using several

middleware for accessing / permission of various methods in this class.

- Index () : This method (default) returns redirect to the view page.
- userpermission () : This method used for send the data of users to the user view .
- circleview () : This method used for getting the circle for the user entry.
- findDivisionName () : This method used for getting the circle for the user entry.
- findSDEcivil () : This method used for getting the SDE for the user entry..
- findSDEcv () : This method used for getting the SDE civil for the user entry.
- findEECivil () : This method used for getting the-EE civil for the user entry.
- findEEem () : This method used for getting the SDE -em for the user entry.
- getSECivil () : This method used for getting the SE Civil for the user entry.
- findSDEem () : This method used for getting the SDE -EE for the user entry.
- getSECivilMobile () : This method used for getting the SE-civil mobile for the user entry.
- getSDEemMobile () : This method used for getting the SDE -Em mobile for the user entry.
- getSEem () : This method used for getting the SE -Em for the user entry.
- getSEemMobile () : This method used for getting the SE -Em mobile for the user entry.
- getSDECivilMobile () : This method used for getting the SDECivilMobile for the user entry.
- getEEemMobile () : This method used for getting the EE Mobile for the user entry.
- getEEcvilMobile () : This method used for getting the EE Civil Mobile for the user entry.
- findsubDivisionName () : This method used for getting the Subdivision name for the user entry.
- userInsert () : This method used for insert the user information in the database table.
- findUserForUpdate () : This method used for update the user information in the database table.
- getUserModal () : This method used for opeing the user form with Modal popup.
- UserSearch () : This method used for searching the user.
- userReport () : This method used for report of the users.
- UserSearchwithEdit () : This method used for user search for edit of the users.

- `checkUserName ( )`: This method is used for check User Name that username availability.

**class UserPermissionController extends Controller { }** : This class Controller is typically used for users permission related functionality .This contains several methods to act several actions The methods are given bellow with the functionality.

- `public function __construct ()`: This construct method is used for using several middleware for accessing / permission of various methods in this class.
- `Index ( )` : This method (default) returns redirect to the view page.
- `insertPermission ( )`: This method used for insert the users permission information in the database table.
- `getuserpermissionmodal ( )` : This method used for opeing the users permission form with Modal popup.

**class AjaxController extends Controller { }** : This class Controller is typically used for ajax related functionality .This contains several methods to act several actions The methods are given bellow with the functionality.

- `getAllotteeModal ( )`: This method used for opeing the allottee form with Modal popup.
- `getEmployee ( )` : This method used for getting the employees/users .
- `findDivisionNamee ( )` : This method used for getting the division.
- `findsubDivisionNamee ( )` : This method used for getting the sub division.
- `findQuarters ()`:This method used for getting the quarters.
- `findBuildings ( )` : This method used for getting the buildings.
- `getReportSearchs ( )`: This method used for getting the Report search.
- `findDistrictGEO ( )` : This method used for getting the Report search.
- `divSearchFunction ( )`: This method is used for division search .
- `subDivSearch ( )`: This method is used for sub-division search .
- `findemployee ( )` : This method used for getting the employees/users .
- `findGEOdistrict ( )`: This method used for getting the district geo .
- `findGEOthana ( )` : This method used for getting the thana.
- `getSdeinfo ( )` : This method used for getting the sde info of employees/users .

**class CustomAuthController extends Controller { }** : This class Controller is typically used for ajax related functionality .This contains several methods to act several actions The methods are given bellow with the functionality.

- `login ( )`: This method used for login for the users .

- Logout () :This method used for logout/ session destroy task.

**class CustomHomeController extends Controller {}** : This class Controller is typically used for ajax related functionality .This contains several methods to act several actions The methods are given bellow with the functionality.

- public function \_\_construct (): This construct method is used for using several middleware for accessing / permission of various methods in this class.
- index (): This method (default) returns redirect to the view page.
- back\_top () :This method returns redirect to admin login panel.
- Home () : This method returns redirect to admin dashboard.

**class ForgetController extends Controller {}** : This class Controller is typically used for ajax related functionality .This contains several methods to act several actions The methods are given bellow with the functionality.

- index () : This method (default) returns redirect to the forget password view page.
- checking () :This method returns redirect to forget password page if name and password checking wrong.
- confirmchecking () : This method returns redirect to confirm checking page .
- Checkpin () : This method checks the SMS pin verification for forget password.
- Checkingconfirm () : This method checks after giving the pin number is correct or wrong.

**class HomeController extends Controller {}** : This class Controller is typically used for ajax related functionality .This contains several methods to act several actions The methods are given bellow with the functionality.

- index (): This method (default) returns redirect to landing page.

**class ItemController extends Controller {}** : This class Controller is typically used for ajax related functionality .This contains several methods to act several actions The methods are given bellow with the functionality.

- pdfview (): This method (default) returns to generate to pdf page.

**class LanguageController extends Controller {}** : This class Controller is typically used for ajax related functionality .This contains several methods to act several actions The methods are given bellow with the functionality.

- `changeLanguage ( )`: This method return to change the language English to Bangla or vice versa.

**class SubmitController extends Controller { }** : This class Controller is typically used for ajax related functionality .This contains several methods to act several actions The methods are given bellow with the functionality.

- `index ( )`: This method used for sending data to review page in confirm submit page.
- `getQuartersforDistrict ( )` : This method is used for get quarter for district wise.
- `getBuildingforintro ( )` : This method is used for getting building according to quarter wise.
- `getFlatforIntro ( )` : This method is used for getting flat according to building wise.
- `getBuildingTypeforintro ( )`:: This method is used for getting building type according to building.
- `getAllotteeinfo ( )` : This method is used for getting allottee according to apartment.
- `checkdata ( )`: This method used for checking the complaint generated pin.
- `InsertComplainforintro ( )` : After successfully complaint form this method is used to insert the complaint and send the sms api to SDE and allottees for final confirmation and generating the ticket.
- `ticketChecking ( )` : This method is used for ticket checking of allottees' complint with getting the send ticket no.

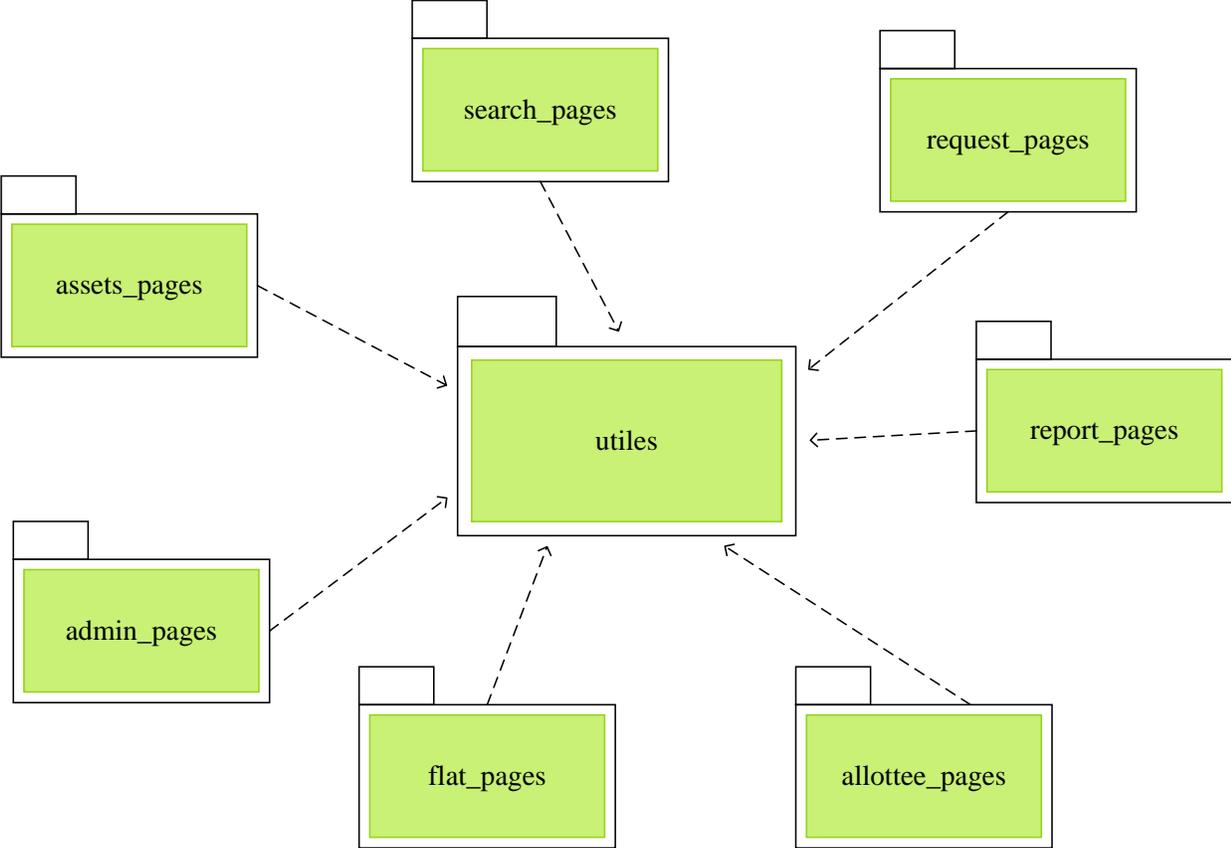
**class SurveyController extends Controller { }** : This class Controller is typically used for ajax related functionality .This contains several methods to act several actions The methods are given bellow with the functionality.

- `index ( )`: This method used to return redirect to survey login panel.
- `login ( )` : This method is used to return redirect to successful login.
- `servey_home ( )` : This method is used to redirect to survey dashboard.
- `logout ( )`: This method is used for logout and destroy the session.
- `totalcomplain ( )` : This method is used for total complaint report for survey.
- `totalcomplainthismonth ( )` : This method is used for total complaint report of this current month for survey.
- `newComplain()`: This method is used for new complaint report for survey.
- `newComplainThisMonth ( )` : This method is used for new complaint report of this current month for survey.
- `InProgressProblem ( )`: This method is used for in progress complaint report for survey.

- CompleteComplains ( ) : This method is used for completed complaint report for survey.
- CancelComplains ( ) : This method is used for canceled complaint report for survey.
- Changepasssword ( ) : This method is used for change password for the survey.
- Checkpassword ( ) : This method is used to change password for the survey.
- updatePassword ( ) : This method is used to update password for the survey.

# 7. Class diagram

## 7.1. Basic folders for class diagram

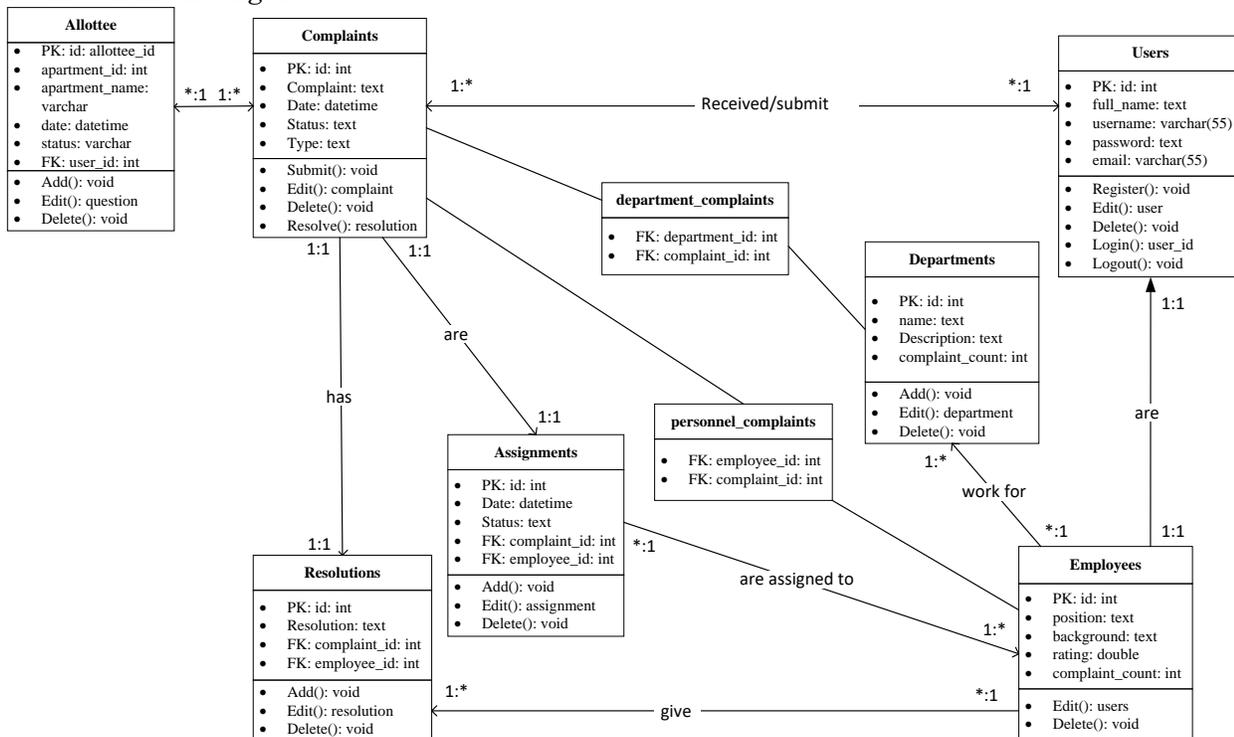


## 7.2 Basic class diagram

Classes are used to capture the vocabulary of a system. They represent software things, hardware things, and even things that are purely conceptual. The classes for the complaint management system were identified through iterative brainstorming with the users. Table shows the complaint management system’s classes identified with their respective descriptions.

No	Class name	Description
1	Allottee	A class to hold details of allottee.
2	Complaints	A class to hold details about complaints.
3	Users	A class to hold username, password and other user account details.
4	Employees	A class to contain employee information, including their backgrounds.
5	Assignments	A class to hold complaints’ assignment to employees, in order to provide resolutions.
6	Resolutions	A class to hold resolutions for complaints.
7	Departments	A class to hold details about departments.

A class diagram of the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, their methods, and the relationships among them. The class diagram is the main building block of the object-oriented modeling. It is used both for general conceptual modeling of the systematic of the application. Class diagrams are used for data modeling. The class diagram of the system is shown in the figure



## 8. Implement and Testing

### A. Setup and Configuration

Installation:

1. Install Linux Centos 7.2.
2. Include MySQL database in installer.
3. After installation of OS following components have to be installed:
  1. phpMyAdmin (ver.3.2.3) graphical tools to manage MySQL database can be downloaded from:

<http://downloads.sourceforge.net/project/phpmyadmin/phpMyAdmin>

After installation of all components some configuration has to be done:

1. MySQL database configuration:
  - To start mySQL automatically, run in shell `chkconfig --add mysql`
  - Create a database with name “pwdgov\_cms” and Collation utf-8-
  - general `CREATE DATABASE project`
  - `CHARACTER SET utf8`
  - `COLLATE utf8_bin;`
  - Set password for root user: `SET PASSWORD FOR 'root'@'localhost' = PASSWORD('yourPassword');`
  - Create user in DB : `CREATE USER project IDENTIFIED BY PASSWORD 'project12345'`
  - Grand privileges to user: `GRANT ALL ON project.* TO 'project'@'%';`
  - Run script “project.sql” to populate database.

#### **Deployment of the CMS on the server:**

On the server CMS has following structure:

-  cms-demo.pwa.gov.ua
- +  .well-known
- +  10\_10\_2018
- +  22\_10\_2018
- +  app
- +  bootstrap
  -  config
- +  database
- +  public
- +  resources
  -  routes
- +  storage
- +  tests
-  vendor
  - +  barryvdh
    -  bin
    -  composer
  - +  dnoegel
  - +  doctrine
  - +  dompdf
  - +  egulias
  - +  erusev
  - +  fideloper
  - +  filp
  - +  fzaninotto

## Tool set and environment

**Table 3.** Tools and Environment

Netbeans version 8.2	Programming, testing, and making UML diagrams
Microsoft Visio	UML diagrams: block diagram, DT, Sequence diagrams, Activity diagrams
Xampp Server v7.0	Admin MySQL through PHP MyAdmin
Microsoft Office Word	Documentation
Microsoft Office Power Point	Presentation
Microsoft Office Excel	Log sheets, cost estimation
MySQL	For database management
e-mail	Communication
CVS integrated into NetBeans	Checkout, update code
JUnit	Unit test
Dreamweaver	Welcome pages/Landing Page
Apache	As HTTP web server

## **B. Implementation list**

The following table shows the list of requirements and indicates whether the requirement was fulfilled in CMS V1.0.

**Table 4.** Implementation list

Name	Implemented (+) / not implemented (-)	Note
1. Dashboard 1.1. Dashboard 1.1.1. View of today's complaint 1.1.2. View of current month complint 1.1.3. View of quarter, Apartment 1.1.4. View of allottee. 1.1.5. Graphical interface report 1.1.6. Graphical interface report 1.2. My profile 1.2.1. View of user information 1.2.2. Change information 1.2.3. Change password and picture 1.3. Logout 1.3.1. Logout from the page.	+ + + + + + - + + + + - +	with restrictions
2. User Control 2.1. User 2.1.1. Add new 2.1.2. View users 2.1.3. Search 2.1.4. Edit 2.2. User permission 2.2.1. View permission 2.2.2. Update permission 2.3. Assign to 2.3.1. Assign to asset	+ + + + + + + +	
3. Setup 3.1. Designation 3.1.1. Add new 3.1.2. View 3.1.3. Edit 3.2. Status 3.2.1. Add new 3.2.2. View 3.2.3. Edit 3.3. Problem type 3.3.1. Add new 3.3.2. View 3.3.3. Edit 3.4. Government Department 3.4.1. Add new 3.4.2. View 3.4.3. Edit 3.5. Division GEO 3.5.1. Add new	+ + + + + + + + + + + + + + -	with restrictions

	3.5.2. View	+	with restrictions
	3.5.3. Edit	+	
3.6.	District GEO	+	
	3.6.1. Add new		
	3.6.2. View	+	
	3.6.3. Edit	+	
3.7.	Building	+	
	3.7.1. Add new		
	3.7.2. View	+	
	3.7.3. Edit	+	
3.8.	Building Type	+	
	3.8.1. Add new		
	3.8.2. View	+	
	3.8.3. Edit	+	
4.	Entry Form	+	
4.1.	Circle		
	4.1.1. View		
	4.1.2. Add new	+	
	4.1.3. Edit	+	
4.2.	Division	+	
	4.2.1. Add new		
	4.2.2. View	+	
	4.2.3. Edit	+	
	4.2.4. Search	+	
4.3.	Subdivision	+	
	4.3.1. Add new		
	4.3.2. View	+	
	4.3.3. Edit	+	
	4.3.4. Search	+	
4.4.	Quarter	+	
	4.4.1. Add new		
	4.4.2. View	+	
	4.4.3. Edit	+	
	4.4.4. Search	+	
4.5.	Building	+	
	4.5.1. Add new		
	4.5.2. View		
	4.5.3. Edit	+	
	4.5.4. Search	+	
4.6.	Allottee	+	
	4.6.1. Add new	+	
	4.6.2. View	-	
	4.6.3. Edit	+	
	4.6.4. Search	+	
4.7.	Apartment	+	
	4.7.1. Add new		
	4.7.2. View	+	
	4.7.3. Edit	+	
	4.7.4. Search	+	
4.8.	Complaint	+	
	4.8.1. Add new	+	

4.8.2.	View		
4.8.3.	Edit	+	
4.8.4.	Search	+	
5.	All Reports	+	
5.1.	Complaint Report	+	
5.1.1.	View		
5.1.2.	Search	+	
5.1.3.	Print	+	
5.1.4.	Export in excel.	+	
5.2.	User Report	+	
5.2.1.	View	+	
5.2.2.	Search	+	
5.2.3.	Print		
5.2.4.	Export in excel.	+	
5.3.	Quarter Report	+	
5.3.1.	View	+	
5.3.2.	Search		
5.3.3.	Print	+	
5.3.4.	Export in excel.	+	
5.4.	Building Report	+	
5.4.1.	View	+	
5.4.2.	Search	+	
5.4.3.	Print	+	
5.4.4.	Export in excel.	+	
5.5.	Allottee Report	+	
5.5.1.	View	+	
5.5.2.	Search	+	
5.5.3.	Print	+	
5.5.4.	Export in excel.	+	
5.6.	Summary Report	+	
5.6.1.	View	+	
5.6.2.	Search	+	
5.6.3.	Print	+	
5.6.4.	Export in excel.	+	
5.7.	Task Report	+	
5.7.1.	View	+	
5.7.2.	Search	+	
5.7.3.	Print	+	
5.7.4.	Export in excel.	+	
6.	Logout	+	
		+	
		+	
		+	
		+	

Dashboard of Access to Information idea (Survey)	-	
1. Dashboard	+	
1.1. Dashboard	+	
1.1.1. View of Total complaint	+	
1.1.2. View of current month complaint	+	
1.1.3. View of new complaint.	+	
1.1.4. New complaint of current month.	+	
1.1.5. View of inprogress Complaint.	+	
1.1.6. View of Completed Complaints	+	
1.1.7. View of canceled Complaints	+	
1.1.8. View of calendar.	+	
1.1.9. Graphical interface report of complaint	+	
1.1.10. Graphical interface report of this month report	+	
1.1.11. Graphical interface of Complaints by month	+	
1.1.12. Graphical interface of complaints by year.	+	
2. Total Complaint	+	
2.1.1. View	+	
2.1.2. Search	+	
2.1.3. Print	+	
2.1.4. Export in excel.	+	
3. Total Complaint of this month	+	
3.1.1. View	+	
3.1.2. Search	+	
3.1.3. Print	+	
3.1.4. Export in excel.	+	
4. New Complaint	+	
4.1.1. View	+	
4.1.2. Search	+	
4.1.3. Print	+	
4.1.4. Export in excel.	+	
5. New Complaint of this month	+	
5.1.1. View	+	
5.1.2. Search	+	
5.1.3. Print	+	
5.1.4. Export in excel.	+	
6. In progress Complaint	+	
6.1.1. View	+	
6.1.2. Search	+	
6.1.3. Print	+	
6.1.4. Export in excel.	+	
7. Completed Complaint	+	
7.1.1. View	+	
7.1.2. Search	+	
7.1.3. Print	+	
7.1.4. Export in excel.	+	
8. Canceled Complaint	+	
8.1.1. View	+	

8.1.2. Search	+	with restrictions
8.1.3. Print	+	
8.1.4. Export in excel.	+	
9. Logout	+	
10. Change Password	+	



## F. Test Report

### F.1. Operational Testing

**Table 1.** Test case 1.1

Test Case Number	1.1
Test Case Name	<i>Login</i>
Test Case Description	This test case verifies if a user can properly log into the system and see the menu.
Preconditions	N/A
Test Case Input	name: string password: string
Test Case Expected Output	Display Home page & List of menu
Test Case Steps:	
Login	
<p>1: Input:</p> <p style="padding-left: 40px;">name: string</p> <p style="padding-left: 40px;">password: string</p> <p>Output:</p> <p style="padding-left: 40px;">Permission: string</p>	
Show Menu	
<p>2: Input:</p> <p style="padding-left: 40px;">username: string</p> <p style="padding-left: 40px;">permission: string</p> <p>Output:</p> <p style="padding-left: 40px;">List of menu</p>	

**Table 2.** Test case 1.2

Test Case Number	1.2
Test Case Name	<i>Report</i>
Test Case Description	This test case
Preconditions	User logged in. Report page.
Test Case Input	Username Permissions Name of report to be built
Test Case Expected Output	Report as a HTML table
Test Case Steps:	
Report	
	<p>1: Input:</p> <p style="padding-left: 40px;">Username</p> <p style="padding-left: 40px;">Permissions</p> <p style="padding-left: 40px;">Name of report to be built</p> <p>Output:</p> <p style="padding-left: 40px;">Report as a HTML table</p>

**Table 3.** Test case 1.3

Test Case Number	1.3
Test Case Name	<i>Advanced Search Model</i>
Test Case Description	This test case verifies if a user searches for a string the system shows all the items in Database including that string.
Preconditions	User logged in. Advanced search page.
Test Case Input	String to be found

	List of tables & List of columns
Test Case Expected Output	HTML table
Test Case Steps:	
Query	
<p>1: Input:</p> <p>String to be found</p> <p>List of tables with</p> <p>List of columns</p> <p>Output:</p> <p>SQL query: string</p>	
Connection	
<p>2: Input:</p> <p>address of SQL server: string</p> <p>Output:</p> <p>TRUE if connected</p>	
Retrieve Data	
<p>3: Input:</p> <p>Data from SQL</p> <p>Output:</p> <p>Array of string</p>	
Display HTML Table	
<p>4: Input:</p> <p>Array of string</p> <p>Output:</p> <p>HTML table</p>	

## F.2. Data Integrity Testing

**Table 4.** Test case 2.1

Test Case Number	2.1
Test Case Name	<i>Saved Data Integrity</i>
Test Case Description	This test case verifies if the data saved when the window is closed by double clicking on the close box?
Preconditions	Logged in user.
Test Case Input	Closing a window.
Test Case Expected Output	Saving data in that window
Test Case Steps:	
Saved Data Integrity	
<p>1: Input:</p> <p style="padding-left: 40px;">Closing a window.</p> <p>Output:</p> <p style="padding-left: 40px;">Saving data in that window.</p>	

**Table 5.** Test case 2.2

Test Case Number	2.2
Test Case Name	<i>Maximum character length</i>
Test Case Description	This test case checks the maximum field lengths to ensure that there are no truncated characters.
Preconditions	Logged in user needed to enter a string.
Test Case Input	Enter a large-length string.
Test Case Expected Output	Temporarily saved large-length string for further actions.

Test Case Steps:	
Maximum Character Length	
	<p>1: Input:</p> <p>Enter a large-length string in the search text box.</p> <p>Output:</p> <p>System could keep it in a variable to search for it without losing any substring.</p>

**Table 6.** Test case 2.3

Test Case Number	2.3
Test Case Name	<i>Default value preserving</i>
Test Case Description	This test case verifies where the database requires a value (other than null) then this should be defaulted into fields. The user must either enter an alternative valid value or leave the default value intact.
Preconditions	User logged in.
Test Case Input	Leave a field with default value.
Test Case Expected Output	Display the proper default value.
Test Case Steps:	
Default Value Preserving	
	<p>1: Input:</p> <p>User does not enter any value for a field which is supposed to load with a default value.</p> <p>Output:</p> <p>Field is displayed with its default value.</p>

**Table 7.** Test case 2.4.1

Test Case Number	2.4.1
Test Case Name	<i>Numeric value range</i>
Test Case Description	This test case checks <u>maximum</u> and minimum field values for numeric fields.
Preconditions	User logged in.
Test Case Input	Enter a numeric value not greater than the maximum limit.
Test Case Expected Output	System can handle the value without any problem.
Test Case Steps:	
Maximum Numeric value	
	<p>1: Input:</p> <p>Enter a large numeric value not greater than maximum limit</p> <p>Output:</p> <p>System can handle the value (Variable type is defined properly).</p>

**Table 8.** Test case 2.4.2

Test Case Number	2.4.2
Test Case Name	<i>Numeric value range</i>
Test Case Description	This test case checks maximum and <u>minimum</u> field values for numeric fields.
Preconditions	User logged in.
Test Case Input	Enter a numeric value not less than the minimum limit.

Test Case Expected Output	System can handle the value without any problem.
Test Case Steps:	
Minimum Numeric value	
<p>1: Input:</p> <p>Enter a small numeric value not less than the minimum value.</p> <p>Output:</p> <p>System can handle the value (variable type is defined properly).</p>	

**Table 9.** Test case 2.5.1

Test Case Number	2.5.1
Test Case Name	<i>Avoid truncation of string and rounding of numeric value.</i>
Test Case Description	This test case verifies If a particular set of data is saved to the database check that each value gets saved fully to the database. i.e. Beware of <u>truncation (of strings)</u> and rounding of numeric values.
Preconditions	User logged in.
Test Case Input	Enter an asset attributes.
Test Case Expected Output	Displays the modifications in the Database.
Test Case Steps:	
Avoid truncation of string values	
<p>1: Input:           Enter an asset attributes.</p> <p>Output:</p> <p>Displays the modifications in the database without losing any character.</p>	

**Table 10.** Test case 2.5.2

Test Case Number	2.5.2
Test Case Name	<i>Avoid truncation of string and rounding of numeric value.</i>
Test Case Description	This test case verifies If a particular set of data is saved to the database check that each value gets saved fully to the database. i.e. Beware of truncation (of strings) and <u>rounding of numeric values</u> .
Preconditions	User logged in.
Test Case Input	Enter a room size
Test Case Expected Output	Displays the modification in the Databse.
Test Case Steps:	
Avoid rounding of numeric values	
<p>1: Input:</p> <p>Enter a set of numeric values for a room size.</p> <p>Output:</p> <p>Displays all the actual values in the Database without losing anything.</p>	

### F.3. Graphical User Interface Testing

**Table 11.** Test case 3.1

Test Case Number	3.1
Test Case Name	<i>Application Compliance Standard</i>
Test Case Description	Closing the application should result in an "Are you Sure" message box
Preconditions	User logged in.
Test Case Status	Failed for main page.

Test Case Attempt:	A user logged in as a test user. By clicking on the close box, the application terminates without any notice!
--------------------	---

**Table 12.** Test case 3.2

Test Case Number	3.2
Test Case Name	<i>Help Menu</i>
Test Case Description	All screens should have a Help button, F1 should work doing the same.
Preconditions	User logged in.
Test Case Status	Failed.
Test Case Attempt:	Pressing F1 in all the pages has no result! Moreover, there is no Help button.

**Table 13.** Test case 3.3

Test Case Number	3.3
Test Case Name	<i>List Box Color</i>
Test Case Description	List boxes are always white background with black text whether they are disabled or not. All others are grey.
Preconditions	User logged in.
Test Case Status	Passed.
Test Case Attempt:	In the search menu, there are list boxes with the correct color. Moreover, in the main page database attributes of each entity are written in black color with white background.

License Pages							
<input type="text"/>		Search	All Licenses	Pages: 1			
Row/Page 15 50 100 150 250							
Name	Name	Total_Quantity	Type	Price	Term	Licence_company	Creation_date
<input type="checkbox"/> 0 <a href="#">More Info</a>	Adobe 9.0	6	Teaching License	200	null	Nosso	2010-04-04 17:54:02.0
<input type="checkbox"/> 1 <a href="#">More Info</a>	MS Office2003	8	Static License	100	null	Microsoft	2010-04-04 17:43:45.0
<input type="checkbox"/> 2 <a href="#">More Info</a>	Photoshop	1	Static License	0			2010-04-11 23:24:03.0
<input type="checkbox"/> 3 <a href="#">More Info</a>	Visio-6	11	Research License	20	2Years	Microsoft	2010-04-04 17:51:39.0
<input type="checkbox"/> 4 <a href="#">More Info</a>	WindowsXP	2	Research License	1000	5 years	Microsoft	2010-04-11 22:53:32.0
All Licenses Pages: 1							

Table 14. Test case 3.4

Test Case Number	3.4
Test Case Name	<i>Curser over an enterable text box</i>
Test Case Description	Move the Mouse Cursor over all Enterable Text Boxes. Cursor should change from arrow to Insert Bar. If it doesn't then the text in the box should be grey or non-updateable. Refer to previous page.
Preconditions	User logged in.
Test Case Status	Passed.
Test Case Attempt:	All over the search text box, curser always changes from arrow to insert bar. 

	<h2 style="text-align: center;">Basic search</h2> <p>Search String: <input type="text"/></p> <p style="text-align: center;"><input type="button" value="Search"/></p>
--	---

**Table 15.** Test case 3.5

Test Case Number	3.5
Test Case Name	<i>Text Box Characters</i>
Test Case Description	<p>Try to overflow the text by typing to many characters - should be stopped Check the field width with capitals W.</p> <p>Enter invalid characters - Letters in amount fields, try strange characters like + , - * etc. in All fields.</p> <p>SHIFT and Arrow should Select Characters. Selection should also be possible with mouse. Double Click should select all text in box.</p>
Preconditions	User logged in.
Test Case Status	Passed, Failed, Passed.
Test Case Attempt:	In some of the fields as attributes of entities in the database, strange characters are accepted.

**Table 16.** Test case 3.6

Test Case Number	3.6
Test Case Name	<i>Color validation</i>

Test Case Description	<p>Is the general screen background the correct colour?</p> <p>Are the field prompts the correct colour?</p> <p>Are the field backgrounds the correct color?</p> <p>In read-only mode, are the field prompts the correct color?</p> <p>In read-only mode, are the field backgrounds the correct color?</p>
Preconditions	User logged in.
Test Case Status	Passed.
Test Case Attempt:	There is no strange color used in the screen, background and text. Read-only and deactive colors are from less sharpness.

**Table 17.** Test case 3.7

Test Case Number	3.7
Test Case Name	<i>Text validation</i>
Test Case Description	<p>Are all the field prompts spelt correctly?</p> <p>Are all character or alpha-numeric fields left justified? This is the default unless otherwise specified.</p> <p>Are all numeric fields right justified? This is the default unless otherwise specified.</p> <p>Is all the microhelp text spelt correctly on this screen?</p> <p>Is all the error message text spelt correctly on this screen?</p> <p>Is all user input captured in UPPER case or lower case</p>

	consistently?
Preconditions	User logged in.
Test Case Status	Passed.
Test Case Attempt:	All the prompts are spelt correctly. Alpha-numeric characters are left justified. Numeric fields are right justified. User inputs are captured in lower case.

**Table 18.** Test case 3.8

Test Case Number	3.8
Test Case Name	<i>Alignment</i>
Test Case Description	Are all the field prompts aligned perfectly on the screen? Are all the field edit boxes aligned perfectly on the screen? Are all group boxes aligned correctly on the screen?
Preconditions	User logged in.
Test Case Status	Passed.
Test Case Attempt:	Very likely all the users enjoy the simple and user-friendly interface of the application along with the right placement of options, buttons and boxes.