# Chapter one: Requirement analysis

# Use case diagram

Use case diagrams are used to depict graphically the interactions between the system and external system and users. In other words, they graphically describe who will use the system and in what ways the user expects to textually describe the sequence of steps of each interaction.



Figure 3 use case diagram

**Description of use case**

|  |  |  |
| --- | --- | --- |
| Title | **Login** | |
| Requirement id | UC.1 | |
| Description | Its purpose is :  -provide proof of administrator.  -in order to enter the system. | |
| Actor | Administrator police | |
| Pre-condition | Administrator and police should have an account or id | |
| Post condition | Access the system or working place.  Authorize actor perform their own work. | |
| Course of action | **Actors action ፡**  **Step 1.**Click login button.  **Step 3.** Fill user name and password  **Step 5.**view the page and display the login page | **systems action**  **Step 2.**Request user name and password.  **Step.**4The system verifies whether the admin have a permission to login or not and displays the main page. |
| Alternate course of action | If there is unauthorized person tray to login to enter the page, the system check the permission of that person and display error message , and return to login form. | |

Table 3 Login use case documentation

|  |  |  |
| --- | --- | --- |
| Title | **View record detail** | |
| Requirement Id | UC.2 | |
| Description | This use case describes or manages the whole personal information that is recorded in the data base. | |
| Actors | Administrator | |
| Precondition | The administrator must be authorized to the system. | |
| Post condition | Manage details of personal information | |
| course of action | **Actors action**  **Step 2.** Admin enter user name and password correctly.  **Step 3.** After login the admin click the view detail button.  **Step 5.** then the actor performs different activities | **Systems action**  **Step 1.** Ask administrator password and user name  **Step 4.** The system displays the administrators need if the admin asks correct detail the record displays in the admin page if not display an error message. |
| Alternate course of action | If the administrator doesn’t the right to show details of personal files the page doesn’t open or display error message | |

Table 4 view record use case documentation

|  |  |  |
| --- | --- | --- |
| Title | **Record new crime** | |
| Required Id | UC.3 | |
| Actors | Administrator ,inspector | |
| Description | The administrator registers the new crime information in to the database. | |
| Precondition | The administrator must be authorized to the system. | |
| Post condition | Record details of personal information. | |
| course of action | **Actors action**  **Step 2.** Admin and police enter user name and password.  **Step 4.** The admin, police and the are entering to the system.  **Step 5.** Then click the register button.  **Step 6.** The record form displays and input the necessary data’s and submit. | **Systems action**  **Step 1.**ask user name and password  **Step 3.** Check the validity of data’s.  **Step 7.** After finishing the admin record the information the system say’s successfully recorded in the data base. |
| Alternate course of action | If the admin and police doesn’t fill the form correctly the system shows a message please fill the necessary data’s. | |

Table 5 view record use case documentation

Tabel.2.3.Record new crime use case documentation

|  |  |  |
| --- | --- | --- |
| Title | **Maintain crime data** | |
| Required ID | UC.5 | |
| Actors | Administrator | |
| Description | This use case contains or includes update, add data and delete if the data is unnecessary. The administrator controls the whole personal profiles’ database. If there is a need to change or delete any data’s the admin have the right to update, add and delete the records. | |
| Precondition | The admin must be sure to change the data and select specific records from the database. | |
| Post condition | After checking the selected profiles are the need to change or delete from the database the admin performs what he wants to. | |
| course of action | **Actors action**  **Step 1.** Admin fill the correct id  **Step 3.** Admin are login to the system and check his profile.  **Step 4.** Select any button to maintain.  **Step 5.** If the new message will find from another branch check the reality to update that information. | **Systems action**  **Step 2.** Request id to enter the system.  **Step 6.** If the admin selects update delete or add one or more records the system say’s the selected records are successfully updated ,delete or add to or from the database. |
| Alternate course of action | If there is no need to change the admin put as it is. | |

Table 6 Maintain crime data use case documentation

|  |  |  |
| --- | --- | --- |
| Title | **View police station** | |
| Requirement ID | UC.8 | |
| Actors | User | |
| Description | The purpose of this use case is to view the police station details to report the crime by using the site contact domain or to get information about station. | |
| precondition | 1. User should know the site domain to report criminal action.  2. Selection of police station. | |
| Post condition | The user reports an crime by filling the proper personal detail. | |
| course of action | **Actor action**  **Step 1.** Users view the page.  **Step 3.**The page displays  **Step 4.** Then fill compliant detail and crime information properly. | **Systems action**  **Step 2.** The system doesn’t display if user do not enter correct domain name of police station site.  **Step 5.** If user doesn’t fill the necessary accident information error message will displayed. |
| Alternate course of action | If there is a problem to enter the domain name the page does not display and access. | |

Table 7 View police station use case documentation

**Chapter 2 high level design**

A high-level design provides an overview of a system, product, service or process.

Such an overview helps supporting components be compatible others.

The highest-level design should briefly describe all platforms, systems, products, services and processes that it depends on and include any important changes that need to be made to them.

Is abstract view of the system which does not show detail description about the system

**2.1 Sequence diagram**

Sequence diagrams describe interactions among classes in terms of an exchange of messages over time. They're also called event diagrams. A sequence diagram is a good way to visualize and validate various runtime scenarios. These can help to predict how a system will behave and to discover responsibilities a class may need to have in the process of modeling a new system

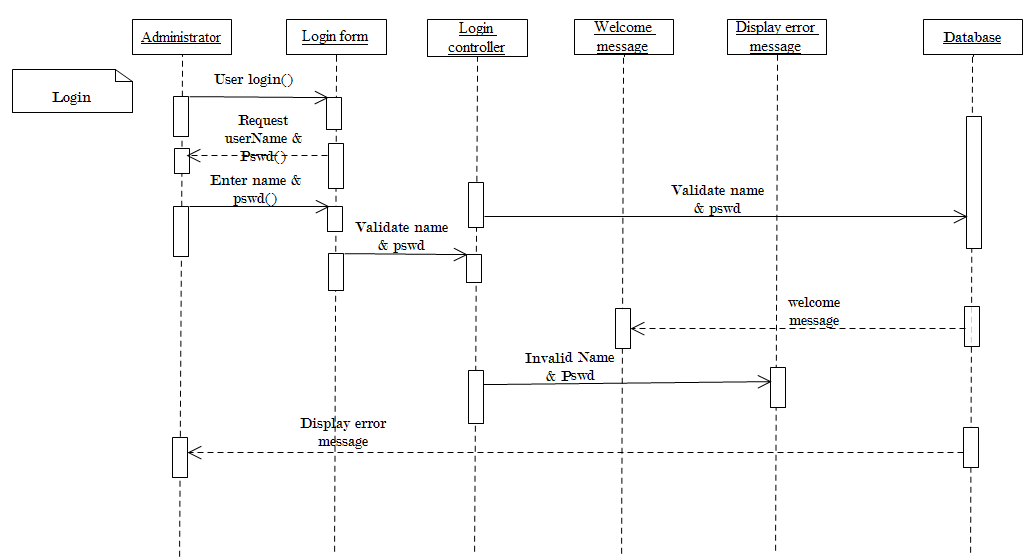


Figure 5 Login sequence diagram

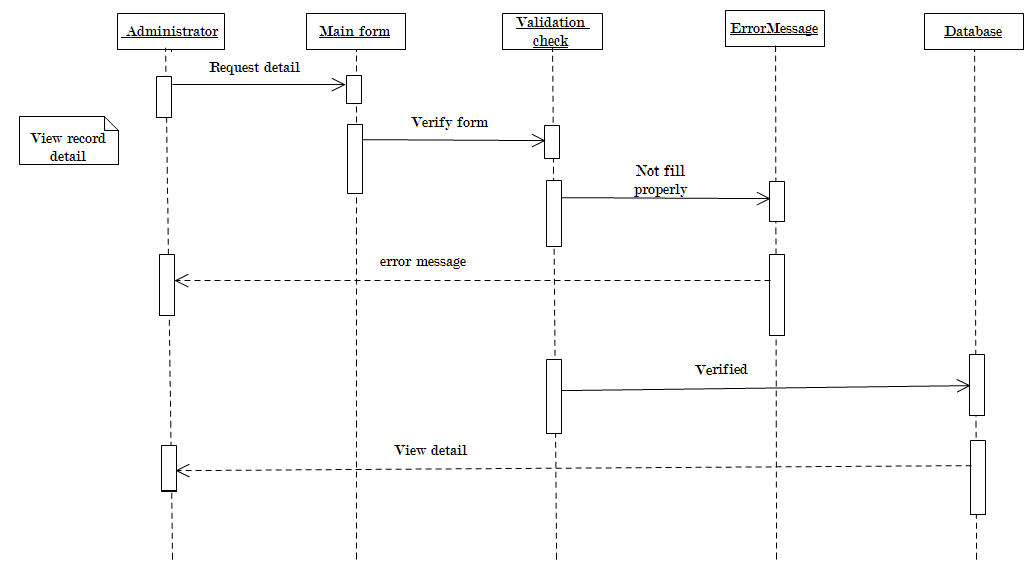


Figure 6 View record sequence diagram

Figure 7 Record new crime

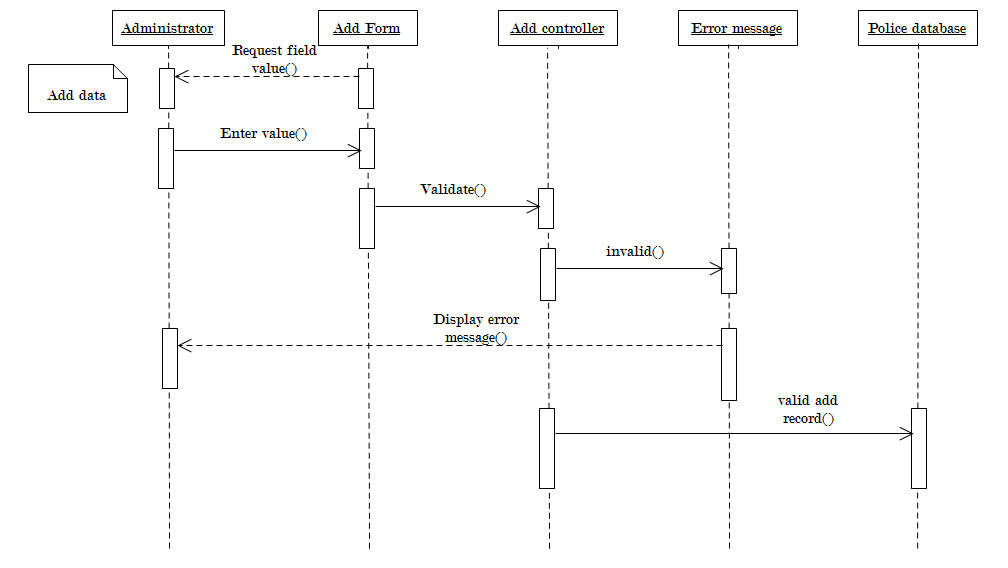


Figure 8 Add additional data to the data base sequence diagram

**Chapter 3 Low level design**

Low level design uses class diagram with more detail description.

**Low-level design** (LLD) is a component-level design process that follows a step-by-step refinement process. This process can be used for designing data structures, required software architecture, source code and ultimately, performance algorithms. Overall, the data organization may be defined during requirement analysis and then refined during data design work. Post-build, each component is specified in detail

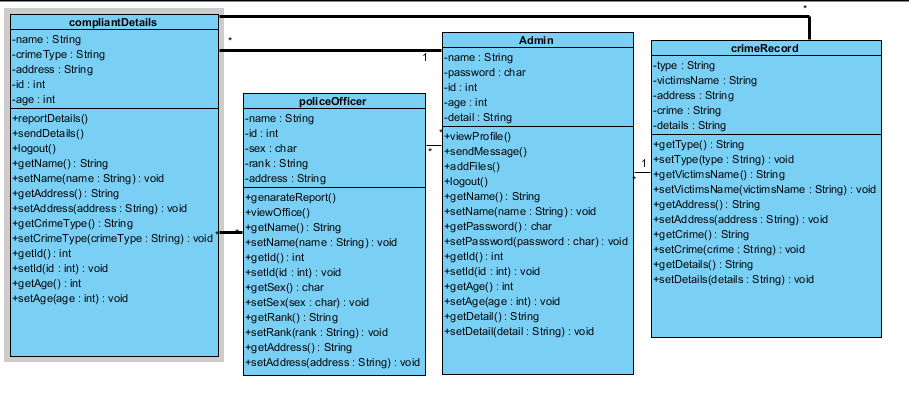
**3.1 class diagram**

 Is a type of static structure diagram that describes the structure of a system by showing the system's [classes](https://en.wikipedia.org/wiki/Class_(computer_science)), their attributes, operations (or methods), and the relationships among objects.

The class diagram is the main building block of [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming) modelling. It is used for general [conceptual modelling](https://en.wikipedia.org/wiki/Conceptual_model) of the systematic of the application, and for detailed modelling translating the models into [programming code](https://en.wikipedia.org/wiki/Programming_code) Class diagrams can also be used for [data modeling](https://en.wikipedia.org/wiki/Data_modeling).[[1]](https://en.wikipedia.org/wiki/Class_diagram#cite_note-1) The classes in a class diagram represent both the main elements, interactions in the application, and the classes to be programmed.

In the diagram, classes are represented with boxes that contain three compartments:

* The top compartment contains the name of the class. It is printed in bold and centered, and the first letter is capitalized.
* The middle compartment contains the attributes of the class. They are left-aligned and the first letter is lowercase.
* The bottom compartment contains the operations the class can execute. They are also left-aligned and the first letter is lowercase.



**Chapter 4 implementation**

**4.1 export class diagram to code: code generation**

|  |  |
| --- | --- |
|  |  |

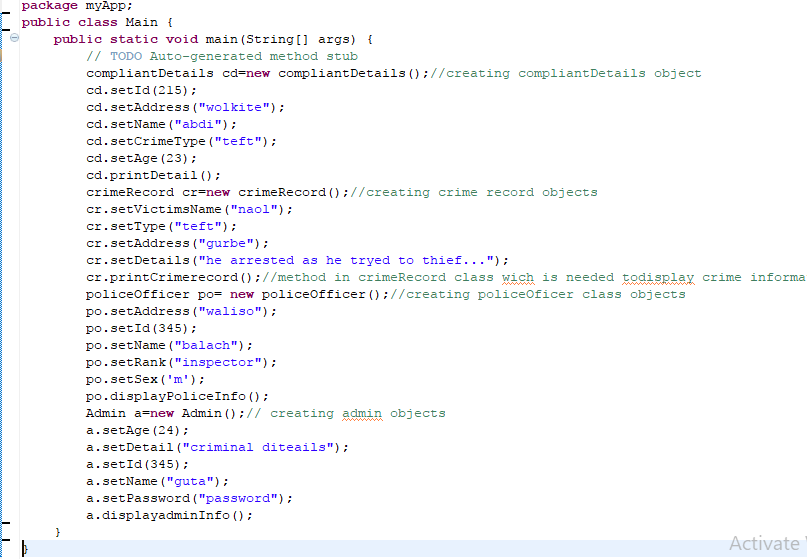
|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
|  |  |

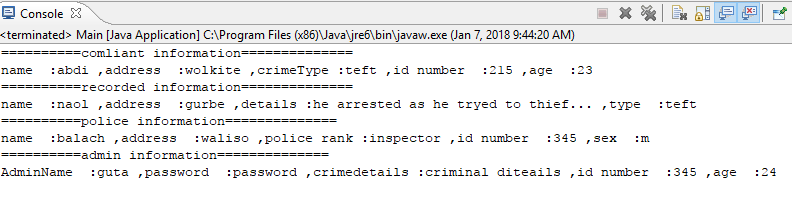
**4.2 Perform coding**

In this section, you are going to build an executable application with each and every classes.



* Add a method for each and every classes and call the in main class

|  |  |
| --- | --- |
| **Method** | **Collee object in main class** |
|  |  |
|  |  |
|  |  |
|  |  |

**4.3 out put of crime record information management system****4.4 update code and diagram**

This is the way for seeing change in each and every classes on what we add a method in the class. For instance the method **printCrimerecord()** is added to **crimeRecord** class to display all recorded criminal information. The method **displayPoliceInfo()** added to **policeOficer** class to prints all police officer information. When we press the updateUml button on the bar; immediately it generates all methods we add in the class. The following change occur on every classes after we update the diagram.

