Czech University of Life Sciences Prague

Faculty of Economics and Management

Department of Information Engineering

Diploma Thesis

Information System Design in UML

Kyaw Pyae Phyo

CZECH UNIVERSITY OF LIFE SCIENCES PRAGUE

Faculty of Economics and Management

DIPLOMA THESIS ASSIGNMENT

Kyaw Pyae Phyo

Informatics

Thesis title

Information System Design in UML

Objectives of thesis

In this diploma thesis, the information system design in UML is covered by feature of web application which is Online Crime Reporting System with UML modeling and relative technologies for system and web development. The first part of the thesis contains mainly the information and facts for researching and studying for the depth theoretical part of UML language and the requirements which will be fulfill by the following web application system. The second part of the thesis will comprise with demonstration of the process and the final outcome of the web application development in UML modeling. The main idea of the second part is just to demonstrate the advantages by highlighting the particular features of the UML language in web application development.

The current crime reporting system is faced with several difficulties as there is no instant means of reporting crime other than telephone calls. By using this application, people who are afraid or don't have enough time to go police station for complaint about their personal legal issues here they can give their complaint through online to register any type of complaint.

Methodology

The methodology of this thesis is based on research and analysis of the information sources and to study and analyze the Information System Design in UML and to develop the web base crime reporting system and criminal record management system for both police force and citizens.

The second part which is the practical part, aims to develop web-based portal using HTML, CSS/Bootstrap, Javascript and JQuery technologies for front end and Java language along with Struts framework (OOP) for backend and with MySQL for relational databases.

The process fulfilled based on analyzing the specifications concluded in the previous research and study. Finally formulation for the conclusions are made based on theoretical and practical knowledge has done in diploma thesis.

The proposed extent of the thesis

60 – 80 pages

Keywords

Unified Modeling Language, Crime reporting, criminal records,OOP, Software Design, Web Application

Recommended information sources

James Martin, James J. Odell: Object oriented methods. Prentice-Hall, 1995.
Michael Blaha a James Rumbaugh: Object-oriented modeling and design with UML. Prentice-Hall. 2005.
Systems analysis and design /Alan Dennis, Barbara Haley Wixom, Roberta M. Roth.–5th ed. 2013.
Vrana, I. Projecting of Information System with UML, CULS Prague, 2009.

Expected date of thesis defence

2018/19 WS – FEM (February 2019)

The Diploma Thesis Supervisor

doc. RNDr. Dana Klimešová, CSc.

Supervising department

Department of Information Engineering

Electronic approval: 1. 12. 2017

Ing. Martin Pelikán, Ph.D.

Head of department

Electronic approval: 1. 12. 2017

Ing. Martin Pelikán, Ph.D.

Dean

Prague on 18. 11. 2018

Declaration

I declare that I have worked on my diploma thesis titled "Information System Design in UML" by myself and I have used only the sources mentioned at the end of the thesis. As the author of the diploma thesis, I declare that the thesis does not break copyrights of any third person.

In Prague on 2019

Kyaw Pyae Phyo

Acknowledgement

First and foremost, I would like to express my sincere gratitude to my thesis supervisor doc. RNDr. Dana Klimešová, CSc. of the Faculty of Economics and Management, Department of Information Engineering at Czech University of Life Sciences Prague for the continuous support of my thesis study and research. The door to my supervisor office was always open whenever I ran into a trouble spot or had a question about my research or writing.

Besides my advisor, I must express my very profound gratitude to all of my teachers and to my partner for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them.

Lastly, yet most importantly, I would like to thank my father, U Win Myaing and my mother, Daw San San Htay, for giving birth to me at the first place, for supporting me spiritually throughout my life and for giving me strength to chase my dreams.

Information System Design in UML

Návrh Informačního Systému v UML

Summary

Information technology has important effects on business operations. No matter the size of the enterprise, technology has both tangible and intangible benefits that will help organization make money and produce the results your customers demand. Technological infrastructure affects the culture, efficiency and relationships of a business. As an organization needs technology with application development, UML plays an important role in it. The UML notation is a language but not a methodology. It does contain a number of diagrams that are utilized in a given development process, what enables easier to understand essential properties of the application which is under development. Each diagram in UML is dedicated to provide a different perspective of the system under design. The essential and most useful, standard diagrams are use case diagram, class diagram, sequence diagram, state chart diagram, activity diagram, component diagram and deployment diagram. The main idea of the thesis second part is just to demonstrate the advantages by highlighting the particular features of the UML language in web application development. The current crime reporting system is faced with several difficulties as there is no instant means of reporting crime other than telephone calls. By using this application, people who are afraid or don't have enough time to go police station for complaint about their personal legal issues, can give their complaint through online to register any type of complaint. Moreover, police officer will be able to manage the reported crime records easily in term of reviewing, approving and assigning to investigating police officer for each case. The process fulfilled based on analyzing the specifications concluded in the previous research and study. Finally formulation for the conclusions are made based on theoretical and practical knowledge has done in diploma thesis.

Keywords: Unified Modeling Language, Crime Reporting, Criminal Records, Object-Oriented Programming, Software Design, Web Application

Souhrn

Informační technologie má významný dopad na obchodní operace. Bez ohledu na velikost podniku má technologie hmatatelné i nehmotné výhody, které organizaci pomohou vydělávat peníze a přinést výsledky, které zákazníci požadují. Technologická infrastruktura ovlivňuje kulturu, efektivitu a vztahy podniku. Protože organizace potřebuje technologii s vývojem aplikací, UML v ní hraje důležitou roli. Notace UML je jazyk, ale nikoli metodologie. Obsahuje řadu diagramů, které jsou využívány v daném vývojovém procesu, což umožňuje snadnější pochopení základních vlastností vyvíjené aplikace. Každý diagram v UML je vyhrazen pro poskytnutí jiné perspektivy systému ve vývoji. Základními a nejužitečnějšími standardními diagramy jsou diagram použití, diagram tříd, sekvenční diagram, diagram stavových diagramů, diagram aktivit, diagram komponent a diagram nasazení. Hlavní myšlenkou druhé části práce je pouze ukázat výhody zvýrazněním konkrétních vlastností jazyka UML ve vývoji webových aplikací. Stávající systém hlášení trestné činnosti čelí několika obtížím, protože neexistují žádné jiné prostředky okamžitého hlášení trestné činnosti než telefonní hovory. Pomocí této aplikace mohou lidé, kteří se bojí nebo nemají dostatek času jít na policejní stanici, aby si stěžovali na své osobní právní záležitosti, mohou podat stížnost online, aby zaregistrovali jakýkoli typ stížnosti. Kromě toho bude policejní důstojník schopen snadno nahlásit hlášené záznamy o trestné činnosti, pokud jde o přezkoumání, schválení a přiřazení vyšetřujícímu policistovi pro každý případ. Tento proces byl umoznen na základě analýzy specifikací uzavřených v předchozím výzkumu a studii. Nakonec jsou formulovány závěry na základě teoretických a praktických znalostí, které byly zpracovány v diplomové práci.

Klíčová slova: Unified Modeling Language, Reporting Crime, Crime Records, Bbjektově Orientovaný Programování, Softwarový design, Webová aplikace.

Table of content

Table of Figures

1. Introduction

Software design process a structure imposed on the development of a software product. To handle the software design process effectively, there is a need to know and understand software process modeling and in order to describe the relationship between the products, resources, activities, and other sort of data, which are involved in the software process.

A significant source of problems is the lack of semantics for the modeling notations used by these methods. A consequence of this is that understanding of models can be more apparent than real. In some cases, developers can waste considerable time resolving disputes over usage and interpretation of notations. While informal analysis, for example, requirements and design reviews, are possible, the lack of precise semantics for OO modeling makes it difficult to develop rigorous, tool-based validation and verification procedures.

Modeling is a technique very frequently used in research and practice related to software engineering. Software projects have high attention on models, which are commonly accepted as the main and the frequently the only type of artifacts that are considered and developed within the development process. UML stands for Unified Modeling Language. The architects of the UML have stated that precision of syntax and semantics is a major goal.

UML was approved as an industry standard for modeling software intensive systems by International Organization for standardization (ISO) [1] in the year 2000. The primary developers of UML were Jim Rumbaugh, Ivar Jacobson, and Grady Booch, who formerly had their own competing methods OMT, OOSE, and Booch. Finally, they have combined their methods to a singular unified notation and brought an open standard called UML [2].

2. Objectives and Methodology

Objectives of thesis: In this diploma thesis, the information system design in UML is covered by feature of web application which is Online Crime Reporting System with UML modeling and relative technologies for system and web development. The first part of the thesis contains mainly the information and facts for researching and studying for the depth theoretical part of UML language and the requirements which will be fulfill by the following web application system. The second part of the thesis will comprise with demonstration of the process and the final outcome of the web application development in UML modeling. The main idea of the second part is just to demonstrate the advantages by highlighting the particular features of the UML language in web application development.

The current crime reporting system is faced with several difficulties as there is no instant means of reporting crime other than telephone calls. By using this application, people who are afraid or don't have enough time to go police station for complaint about their personal legal issues here they can give their complaint through online to register any type of complaint.

Methodology:The methodology of this thesis is based on research and analysis of the information sources and to study and analyze the Information System Design in UML and to develop the web base crime reporting system and criminal record management system for both police force and citizens. The second part which is the practical part, aims to develop web-based portal using HTML, CSS/Bootstrap, Javascript and JQuery technologies for front end and Java language along with Struts framework (OOP) for backend and with MySQL for relational databases. The process fulfilled based on analyzing the specifications concluded in the previous research and study. Finally formulation for the conclusions are made based on theoretical and practical knowledge has done in diploma thesis.

3. Theory Background

Trends have shown in recent years, modeling becomes popular in area of software engineering. Models are usually graphical things. The success of a product depends on one of its most critical feature - the model. The planning involved in designing a software or product includes the same approach. The following figure 1.1 depicts the basic idea of process of development in every industry [3].

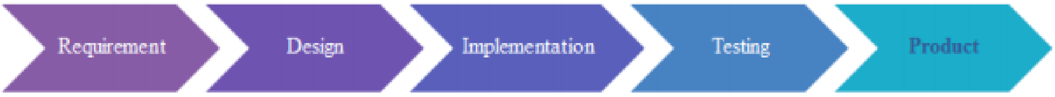


Figure 1: Process development - abstract view (Source: [3])

Demands of complex requirements are seeing an exponential increase, management of e-government sectors in particular. One of the main reasons for this growth is the rise in demand for safety features. And hence, interest was stirred to explore the roots for the same. Unified Modeling Language (UML) is a modeling language designed for general purposes in the software engineering. It conforms to a standard created and managed by the Object Management Group.

3.1 Unified Modeling Language (UML)

UML stands for Unified Modeling Language. The language allows a rich expression and is used to make software blueprints [4]. Software industries have considered UML as an accepted and commonly used language for object oriented language. To accomplish high quality end product, software developers should work accordingly to the selected and implemented software process model [4]. Modeling is a technique very frequently used in research and practice related to software engineering and also a rationale for Model driven development (MDD) paradigm.

UML syntax is a set of graphic notation techniques. It is used to specify, visualized, modify, construct and document details of an object-oriented system under development [4]. UML was approved as an industry standard for modeling software intensive systems by International Organization for standardization (ISO) [5] in the year 2000. The primary developers of UML were Jim Rumbaugh, Ivar Jacobson, and Grady Booch, who formerly had their own competing methods OMT, OOSE, and Booch. Finally, they have combined their methods to a singular unified notation and brought an open standard called UML [6].

Big software product with complicated and complex software designs are difficult to derive textually, but they can be carried through diagrams of UML. It provides three key attributes like visualization, complexity management, and clear communication, which help to describe and to design more easily complex software systems [7-8].

The UML notation is a language but not a methodology. It does contain a number of diagrams that are utilized in a given development process, what enables easier to understand essential properties of the application which is under development. UML includes a number of diagram. Each diagram is dedicated to provide a different loo or perspective of the system under design. The essential and most useful, standard diagrams are use case diagram, class diagram, sequence diagram, state chart diagram, activity diagram, component diagram, deployment diagram and object diagram [9-10]. While using UML modeling, the first step is to identify and describe the process requirements namely process analysis of system e.g. use case diagram. The second step is defining and describing important elements of process model e.g. class diagram, activity diagram. As the third step, we motivate to report the activities, execution of elements in the form of a state chart diagram, sequence diagram [9-10].

We use UML while we would like to model the software or product and not only for performing the programming modules. UML is available to everyone and software industries have freedom to use it. UML has range for the usage. It can be used in business modeling or software modeling [11]. One of the advantages of UML is that developers can learn it easily [11].

We, as human, may have a lot of views on multiple subjects. Due to this unique advantage of multiple views, there is a danger that different partial models produced during the development process can be inconsistent. According to the online dictionary [12], consistency means a harmonious uniformity or agreement among things or parts. Inconsistency in software design may leads the product to delays, bugs or even worse failure. The main advantage of UML is that it has the capability to describe the different kinds of design characteristics varying from structural diagrams to behavioral diagrams.

3.2 Evolution of UML

In the late 1950s, the first object orientated programming language, Simula was introduced, and with it came a powerful new combination of ideas into structuring computer programs, including instantiation of abstract data types, inheritance, and polymorphism [13]. To accompany this new idea of object orientated languages, methods for designing software in this object orientated way also started to emerge, and in time they were referred to as modeling languages. By the late 1980s there were more than fifty separate modeling language, each with their own syntax, structure and notation. There were many issues with this overwhelming variety of languages and it has been noted that such open-ended approaches affect and constrain the system in unexpected ways or even result in failure. This lack of standardization and communication was not only negatively affecting development projects but also limiting the potential of object-orientated technology in general.

During the 1990s many different methodologies, along with their own set of notations, were introduced to the market. Three of the most popular methods were OMT (Rumbaugh), Booch, and OOSE (Jacobson). Each method had its own value and emphasis. OMT was strong in analysis and weaker in the design area. Booch 1991 was strong in design and weaker in analysis. Jacobson was strong in behavior analysis and weaker in the other areas [14].

The end of the method wars as far as notation is concerned comes with the adoption of the Unified Modeling Language (UML). UML is a language used to specify, visualize, and document the artifacts of an object-oriented system under development. It represents the unification of the Booch, OMT, and Objectory notations, as well as the best ideas from a number of other methodologists as shown in Figure 2 [14].

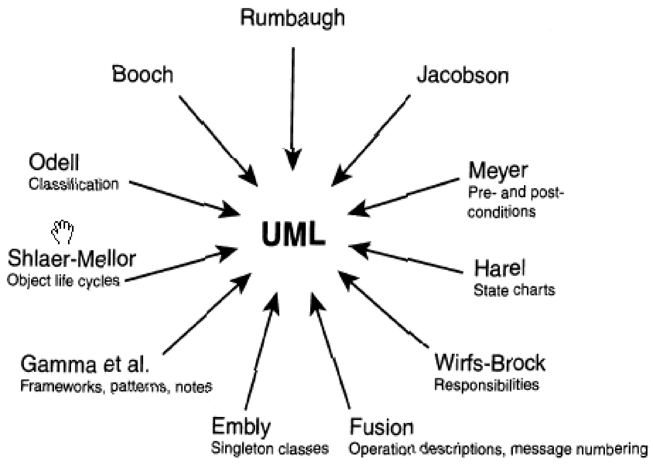


Figure 2: UML inputs (Source: [14])

The initial response after the release of the specification indicated that the Unified Modeling Language was very effective, once the personnel involved had made it past the difficult learning curve of training in a new modeling language. In fact there is speculation that the response towards UML was actually too great - for while it was proven to be much more effective than its predecessors, it still had issues. The rapid uptake and positive response meant that the uptake of UML ended up growing at an alarming rate before it had finished standardizing properly [15].

While Rational was bringing UML together, efforts were being made on achieving the broader goal of an industry standard modeling language. In early 1995, Ivar Jacobson (then Chief Technology Officer of Objectory) and Richard Soley (then Chief Technology Officer of OMG) decided to push harder to achieve standardization in the methods marketplace. In June 1995, an OMG-hosted meeting of all major methodologists (or their representatives) resulted in the first worldwide agreement to seek methodology standards, under the aegis of the OMG process [16].

A number of revisions have taken place to alter the notation in order to fix various shortcomings and to become more effective. For example, some of the is-sues that were resolved between UML 1.1 and UML 1.3 included the lack of integration between certain model types, the absence of certain modelers and that some of the standard elements were named and organized inconsistently.

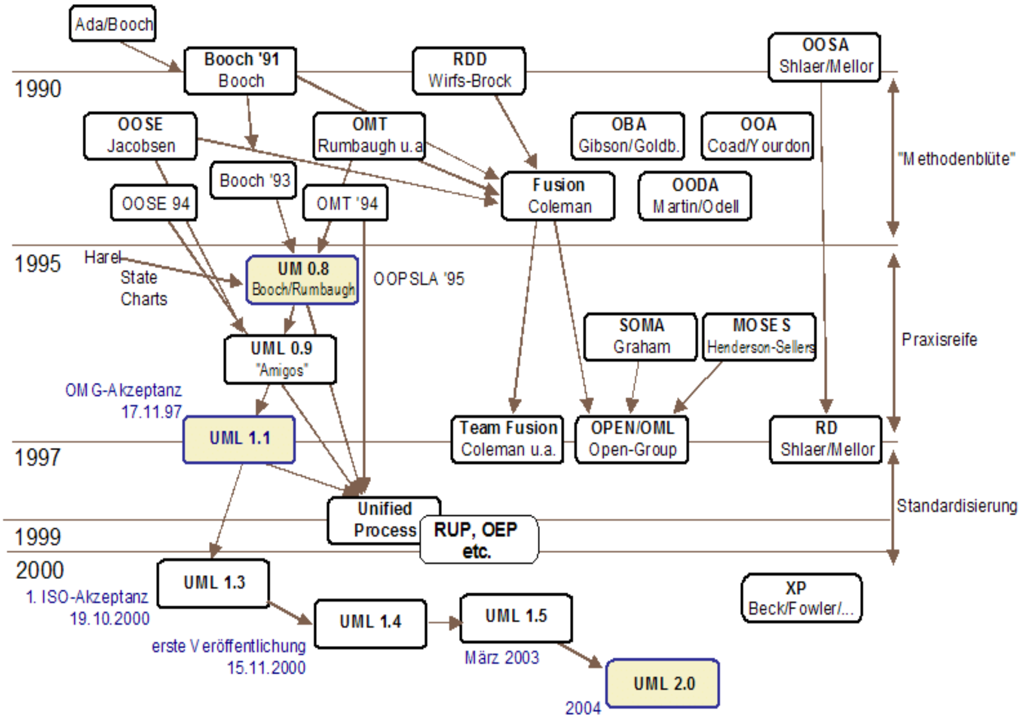


Figure 3: The history of UML and other Object-Oriented Methods and Notations (Source: [17])

The next specification for UML, UML 2.5.1, was officially published in December 2017 [18].

3.3 An Overview of UML

UML 2.0 defines thirteen types of diagrams, divided into three categories: Six diagram types represent static application structure; three represent general types of behavior; and four represent different aspects of interactions:

- Structure Diagrams include the Class Diagram, Object Diagram, Component Diagram, Composite Structure Diagram, Package Diagram, and Deployment Diagram.
- Behavior Diagrams include the Use Case Diagram (used by some methodologies during requirements gathering); Activity Diagram, and State Machine Diagram.
- Interaction Diagrams, all derived from the more general Behavior Diagram, include the Sequence Diagram, Communication Diagram, Timing Diagram, and Interaction Overview Diagram [18].

3.3.1 UML Models

Computers are an integral part of our society and economy and their software becomes larger and more complex every day. Software development is a difficult activity and many software projects take longer than expected, have large operational failures or are simply cancelled [19].

It is proposed that one way to reduce the complexity of software development is to build software models. A model is a simplification of the reality that still retains the elements relevant to our problem. The purpose of a software model is to represent and convey the most important requirements and design decisions of a software system. A good model should explain these concepts in an easy way and ignore all the unimportant details. A detail is unimportant if, for example, it is well known by all the members of the development team or if it can be easily inferred from the rest of the model. A software model is described using a modeling language. Currently, the most widespread software modeling language is the Unified Modeling Language (UML) [16].

A UML model is usually composed of several diagrams. This division of a model into different diagrams is an important mechanism to manage software complexity. Each diagram represents a different aspect or view of a system.

UML models are created and used during the whole software development process. We can start drawing UML diagrams during the initial planning and analysis stages, even when we still do not have a clear idea of how our system will look. UML models are created in the early stages of the development process and therefore the correction of mistakes during analysis and design phases can be achieved more cheaply and rapidly than during the implementation or when the software is already deployed. This principle still holds in evolutionary or spiral software processes, since design always precedes coding [20].

UML models are usually composed of different diagrams. Some UML diagrams show the static structure of a software system. For example, a class diagram shows a collection of classes, types and interfaces and their associations. A class diagram can show the operations of each class but it does not describe the actual behavior of these operations. Behavioral diagrams, such as sequence diagrams and state charts, can be used to describe the dynamics of a model. These diagrams can describe what happens when invoking an operation or the whole lifetime of an object. A complete UML model always contains structural and behavioral diagrams describing both the static and dynamic aspects of a system. However, this thesis is focused on the study of behavioral diagrams.

3.3.2 UML Semantics

UML definition includes a combination of semantic and syntactic rules that helps to understand the models. Some rules are formally stated [21] [22] to maintain the well- form of the models and the rest of the rules are informally stated to provide more flexibility to models at different levels of abstraction.

The current UML specification consists of two interrelated parts. They are UML Semantics and UML Notation. UML Semantics specifies the abstract syntax and semantics of UML object modeling concepts, since UML Notation is a graphic notation for the visual representation of the UML semantics. In addition, the UML Notation describes the mapping of the graphic notation to the underlying semantics. The two parts complement each other without duplicating functionality [23].

UML Semantics is divided into three layers; structural, behavioral and higher-level formalisms of UML: activities, state machines and interactions. Behavioral layer is caused by actions of structural entities from structural level. Communication between structural entities is called inter object behavior base, whereas, the intra-object behavior base, addresses behavior occurring within structural entities. Action sub layer itself defines the semantics of individual actions [24].

Semantics are categorized into three dimensions: internal, vertical and horizontal dimension. Internal deals with relationships between sub-models that coexist. For instance, an analysis model consists of an analysis class diagram, interaction diagram and collaboration diagrams. All the artifacts with a single model are related and must be compatible with each other. The vertical dimension considers relationship between models belonging to same iteration in different activities for example a design model realizing an analysis model. Whereas horizontal dimension considers relationships between artifacts belonging to the same activity in different iterations, for example a use case is extended by another use case [25-26].

3.3.3 Formal Semantics

To remove ambiguities in the language and to remedy the previous situations, many authors are working towards a formal semantics for UML [27]. Formal semantics give an unambiguous interpretation to UML using a mathematical formalism and enable static and behavioral analysis of UML models. We can give formal semantics to UML by introducing a mapping from UML diagrams to a well-established formal method for software development. Examples of this approach are the formalizations of UML into Z, Object-Z [28] or VDM [29]. Once a UML model is translated into a specification in a formal language, all the analysis and manipulations are performed on the formal specification. The clear advantage of this approach is that it is possible to reuse all the previous work on the formal method for analyzing UML models. Other authors work on the formalization of UML directly in a mathematical theory, like set theory or category theory. The advantage of this approach is that the underlying mathematical theory is usually more abstract and imposes fewer restrictions than a formal method that was not specifically designed for UML [30].

3.4 Use of UML

The UML is used to model systems and it can also be used in the different phases in the development of a system, from the requirements specification to the test of a finished system [31].

The goal if UML is to describe any type of system, in terms of object-oriented diagrams. Naturally, the most common use is to create models of software systems, but UML is also used to describe mechanical systems without any software, or the organization of a business [32].

The different types of systems which can be used to model with UML include:

- Information systems: Store, retrieve, transform, and present information to users. Handle large amounts of data with complex relationships, which are stored in relational or object databases.

- Technical systems: Handle and control technical equipment such as telecommunications, military systems, or industrial processes. They must handle the special interfaces of the equipment and have less standard software than information systems. Technical systems are often real-time systems.

- Embedded real-time systems: Execute on simple hardware embedded in some other equipment such as mobile phone, car, household appliance, etc. This is accomplished through low-level programming that requires real-time support. These systems often lack devices such as display, hard disk and so on [33].

- System software: Defines the technical infrastructure that other software uses. Operating systems, databases, and user interfaces perform low-level operations on the hardware, while presenting generic interfaces for other software to use.

It is important to emphasize that most systems don’t fit neatly into one of these categories, but belong to more than one of the system types or as a combination. For example, many of today’s information systems have both distributed and real-time requirements. UML has the capability to model all of these system types [31].

3.5 UML and System Development Phases

3.5.1 Phases of System Development

There are five phases of system development: requirement analysis, analysis, design, programming, and testing [34].

Requirement Analysis

UML has use cases to capture the requirements of the customer. Through the use-case modelling, the external actor that have interest in the system are modelled along with functionality they require from the system (the use cases). The actors and use cases are modelled with relationships, and have communication associations with each other or are broken down into hierarchies. The actors and use cases are described in a UML use-case diagram. Each use case is described in text, and that specifies the requirements of the customer: What he or she expects of the system without considering how the functionality will be implemented. A requirement analysis can also be done with business processes, not only for software systems.

Analysis

The analysis phase is concerned with the primary abstractions (classes and objects) and mechanisms that are present in the problem domain. The classes that model these are identified, along with their relationships to each other, and described in a UML class diagram. Collaborations between classes to perform the use cases are also described, via any of the dynamic models in UML. In the analysis, only classes that are in the problem domain (real world concepts) are modelled – not technical classes that define details and solutions in the software system, such as classes for user interface, databases, communication, concurrency, and so on.

Design

In the design phase, the result of the analysis is expanded into a technical solution. New classes are added to provide the technical infrastructure: the user interface, database handling to store objects in a database, communication with other systems, interfacing to devices in the system, and others. The domain problem classes from analysis are "embedded" into this technical infrastructure, making it possible to change both the problem domain and the infrastructure. The design results in detailed specifications for the construction phase.

Programming

In the programming, or construction, phase, the classes from the design phase are converted to actual code in an objected-oriented programming language (using a procedural language is not recommended). Depending on the capability of the language used, this can either be a difficult or an easy task. When creating analysis and design models in UML, it is best to avoid trying to mentally translate the models into code. In the early phases, the models are a means to understand and structure a system; thus, jumping to early conclusions about the code can be counterproductive to creating simple and correct models. The programming is a separate phase during which the models are converted into code.

Test

A system is normally tried in unit tests, integration tests, system tests, and acceptance tests. The unit tests are of individual classes or a group of classes, and are typically performed by the programmer. The integration test integrates components and classes in order to verify that they cooperate as specified. The system test views the system as a "black box" and validates that the system has the end functionality expected by the end user. The acceptance test conducted by the customer to verify that the system satisfies the requirements is similar to the system test. The different test teams use UML diagrams as the basis for their work: unit tests use class diagrams and class specifications, integration test typically use component diagrams and collaboration diagrams, and the system tests implement user-case diagrams to validate that the system behaves as initially defined in these diagrams.

3.6 Software Testing

Software testing determines whether or not a product works correctly and efficiently according to the requirements of end-users [35]. Software testing subjects a piece of code to both controlled and uncontrolled operating conditions in order to verify the output which should be in accordance with pre-defined conditions.

Through software testing, we are able to find problems in the system so that high quality software can be provided. It has been suggested that software testing should occupy 40% of a software development budget [35].

During the software development life cycle, software testing starts with unit testing, following by function testing/integration testing and ending with acceptance testing to build high quality software [36]. The different software testing types are defined below:

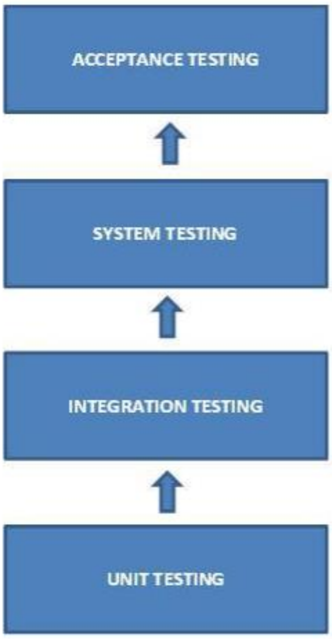


Figure 4: Software testing building quality to completion (Source: [37])

3.6.1 Testing Methods

Software testing methods are traditionally divided into black box testing and white box testing [36]. These two methods are used to describe the point of view that a test engineer takes when designing test cases.

Black Box Testing

Black box testing treats the software as a black box without any knowledge of internal implementation. It includes model-based testing, traceability matrix and specification-based testing [36].

Specification-Based Testing

Specification-based testing aims to test the functionality according to the requirements. Therefore the tester inputs data and only sees the output from the test object. This level of testing usually requires thorough test cases to be provided to the tester who then can simply verify that for a given input, the output value (or behaviour), is the same as the expected value specified in the test case [36].

White Box Testing

White box testing, by contrast to black box testing, is when the tester has access to the internal data structures and algorithms (and the code that implement these).

The following are the types of white box testing exists [36]:

- Code Coverage: It creates tests to satisfy some criteria of code coverage,
- Mutation Testing methods ,
- Fault Injection Methods,
- Static Testing: White box testing includes all static testing.

3.7 UML Diagrams

The Unified Modeling Language (UML) is the de-facto standard object-oriented modeling language for modeling software systems [38]. UML specifications are developed and maintained by the Object Management Group (OMG). UML is a set of modeling notations for describing static structures and behaviors of software systems. This thesis uses UML v2.5. The UML 2.5 specification defines seven structural diagrams: class, object, composite structure, profile, package, and component and deployment diagrams. It also defines four kinds of behavioral diagrams: user case, activity, and state machine and interaction diagrams.

UML specification defines two major kinds of UML diagram: structure diagrams and behavior diagrams [38].

Structure diagrams show the static structure of the system and its parts on different abstraction and implementation levels and how they are related to each other. The elements in a structure diagram represent the meaningful concepts of a system, and may include abstract, real world and implementation concepts.

Behavior diagrams show the dynamic behavior of the objects in a system, which can be described as a series of changes to the system over time.

Making the decision to use UML in the process of system development does not mean that all UML diagrams must be used. It is up to the decision of the modeler, which diagrams to use in the process of development. The consideration is usually based on the nature of the system, the domain, the functionalities, as well as the preferences of the modeler, since some UML diagrams might in some cases replace each other.

Diagrams Overview

UML2.5 has 14 types of diagrams divided into multiple categories as in the following figure 5.

Structure Diagram

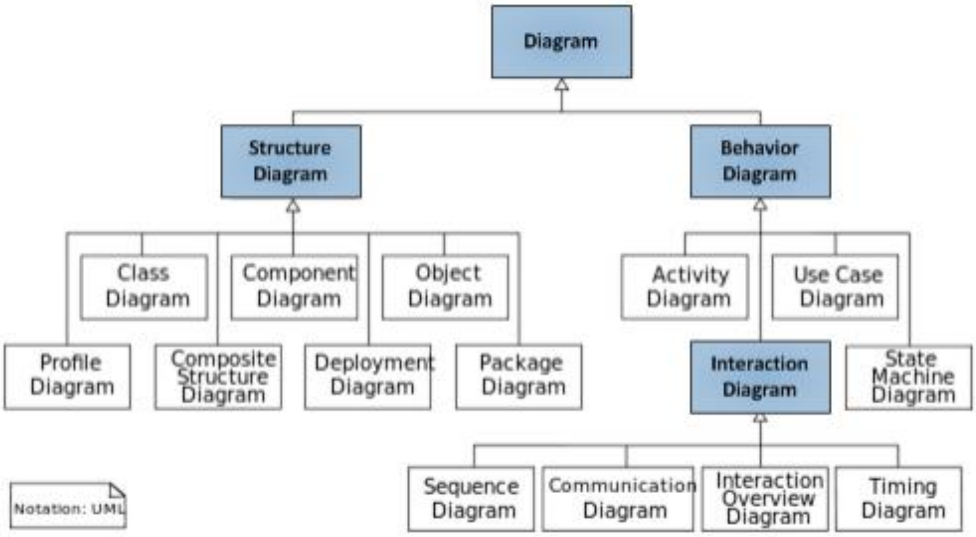


Figure 5: Diagrams Overview of UML (Source: [38])

Structure diagrams describes the static structure of system and its parts on different abstraction and implementation levels and how they are related to one another. The elements in a structure diagram represent the meaningful concepts of a system, and may include abstract, real world and implementation concepts.

These diagrams emphasize the things that must be present in the system being modeled. Since they represent the structure, they are used extensively in documenting the software architecture of software systems. Important features of UML include the unification of these diagrammatic notations, the specification of a standard instead of having several incompatible notations and an approach to support the entire software development process. UML therefore provides [38]:

- **Class diagram** which shows structure of the designed system, subsystem or component as related classes and interfaces, with their features, constraints and relationships - associations, generalizations, dependencies, etc.

- **Component diagram** which shows components and dependencies between them. This type of diagram is used for Component-Based Development (CBD), to describe systems with Service-Oriented Architecture (SOA).

- **Object diagram** was defined in now obsolete UML 1.4.2 Specification as "a graph of instances, including objects and data values. A static object diagram is an instance of a class diagram; it shows a snapshot of the detailed state of a system at a point in time." It also stated that object diagram is "a class diagram with objects and no classes. UML 2.5 specification simply provides no definition of object diagram.

- **Composite structure diagram** is a diagram which could be used to show the internal structure of a classifier and a behavior of a collaboration.

- **Package diagram** which shows packages and relationships between the packages.

- **Profile diagram**s are auxiliary UML diagram which allows to define custom stereotypes, tagged values, and constraints as a lightweight extension mechanism to the UML standard. Profiles allow to adapt the UML meta-model for different platforms (such as J2EE or .NET), or domains (such as real-time or business process modeling). Profile diagrams were first introduced in UML 2.0.

- **Deployment diagram** which shows architecture of the system as deployment (distribution) of software artifacts to deployment targets.

Class Diagram

A class diagram is a diagram that shows a set of classes, interfaces, and collaborations and their relationships; class diagrams address the static design view of a system; a diagram that shows a collection of declarative (static) behavior [1] define the class as the descriptor of a set of objects that share the same attributes, operations, methods, relationships, and behavior.

UML class diagrams allow for modeling, in a declarative way, the static structure of an application domain, in terms of concepts and relations between them. We concentrate on UML class diagrams for the conceptual perspective. In particular, we do not deal with those features that are relevant for the implementation perspective, such as public, protected, and private qualifiers for operations and attributes. We describe the semantics of each construct of UML class diagrams in terms of first order logic (FOL) [39].

The Basics

As mentioned earlier, the purpose of the class diagram is to show the types being modeled within the system. Four fundamental aspects of UML class diagrams are:

- Classes, which describe the different kinds of objects that can exist in a system;
- Associations, which describe what kinds of objects can be linked together;
- Multiplicity constraints, which state how many objects can be related to each other;
- Generalization, which is used to classify objects, and therefore simplify the overall structure of a design.

UML uses a special name for these types: "classifiers." Generally, a classifier behave as a class, but technically a classifier is a more general term that refers to the other three types above as well.

Classes

A class in a UML class diagram denotes a set of objects with common features. A class is graphically rendered as a rectangle divided into three parts as we can see in following figure 6 [1].

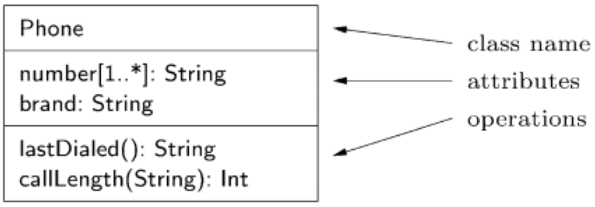


Figure 6: Sample Class Diagram (Source: [1])

The first part contains the name of the class, which has to be unique in the whole diagram. The second part contains the attributes of the class, each denoted by a name, possibly followed by the multiplicity, and with an associated type, for the attribute values. The third part contains the operations of the class, i.e., the operations associated to the objects of the class. Note that both the second and the third part are optional.

The purpose of class diagram is to graphically depict the relationships holding among objects manipulated by a system. As an example, a typical class diagram is shown below, which depicts the relationship between a university and its students:

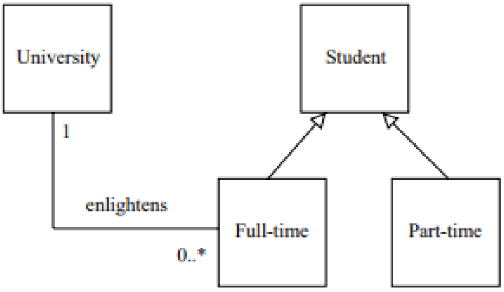


Figure 7: Overview of sample Class Diagram (Source: [1])

As this example in figure 7 shows, a UML class diagram provides a visually expressive and intuitive model of a system. However, it is less effective when it comes to answering important questions about the system it represents. In particular, it is not possible to reason (in a precise manner) with the diagram or deduce properties about it.

The components of UML class diagrams

A class diagram is composed of a number of primitive diagrammatic objects as follows:

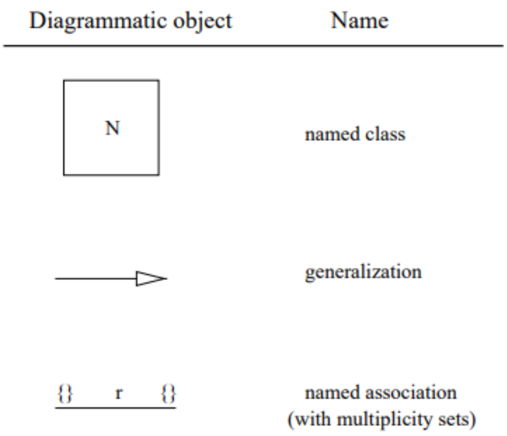


Figure 8: Number of diagrammatic objects with class diagram (Source: [1])

In general, a class diagram consists of a number of classes that are related (linked) by associations and generalization hierarchies. Generalization is a relationship between classes in which one class is identified as the general class and the others as specializations of it.

Relationships in Class Diagrams

Classes are interrelated to each other in specific ways. In particular, relationships in class diagrams include different types of logical connections. The following are such types of logical connections that are possible in UML [40]:

- Association
- Directed Association
- Reflexive Association
- Multiplicity
- Aggregation
- Composition
- Inheritance/Generalization
- Realization

Association

Association is a broad term that encompasses just about any logical connection or relationship between classes. For example, passenger and airline may be linked as below figure:

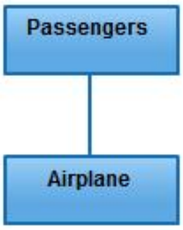


Figure 9: Sample association in class diagram (Source: [40])

Directed Association

Directed association refers to a directional relationship represented by a line with an arrowhead. The arrowhead depicts a container-contained directional flow.

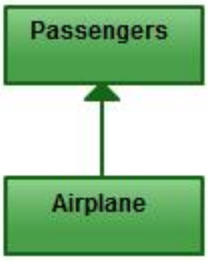


Figure 10: Sample directed association in class diagram (Source: [40])

Reflexive Association

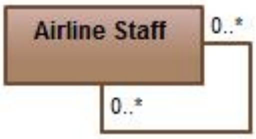


Figure 11: Reflexive Association (Source: [40])

This occurs when a class may have multiple functions or responsibilities. For example, a staff member working in an airport may be a pilot, aviation engineer, a ticket dispatcher, a guard, or a maintenance crew member. If the maintenance crew member is managed by the aviation engineer there could be a managed by relationship in two instances of the same class.

Multiplicity

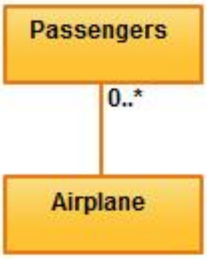
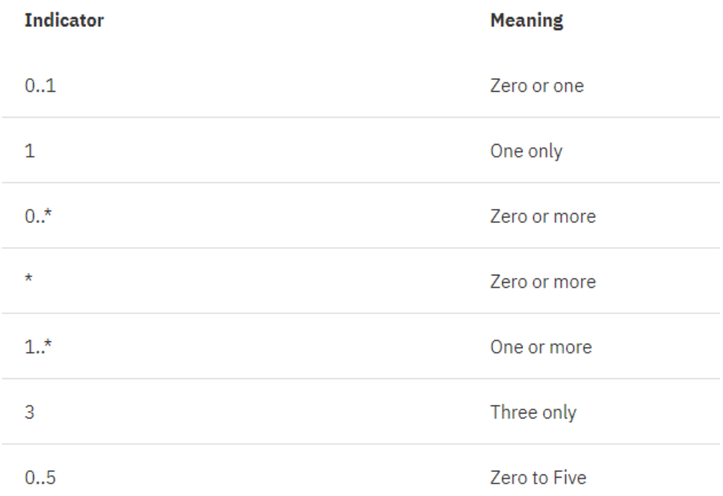


Figure 12: Multiplicity (Source: [40])

Multiplicity is the active logical association when the cardinality of a class in relation to another is being depicted. For example, one fleet may include multiple airplanes, while one commercial airplane may contain zero to many passengers. The notation 0..* in the diagram means “zero to many”.



Indicator	Meaning
0..1	Zero or one
1	One only
0..*	Zero or more
*	Zero or more
1..*	One or more
3	Three only
0..5	Zero to Five

Table 1: Multiplicity values and their indicators (Source: [41])

Aggregation

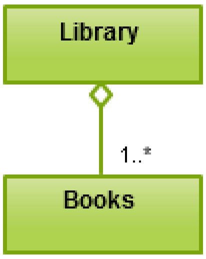


Figure 13: Aggregation (Source: [40])

Aggregation refers to the formation of a particular class as a result of one class being aggregated or built as a collection. In aggregation, the contained classes are not strongly dependent on the lifecycle of the container. In the same example, books will remain so even when the library is dissolved. To show aggregation in a diagram, draw a line from the parent class to the child class with a diamond shape near the parent class.

Composition

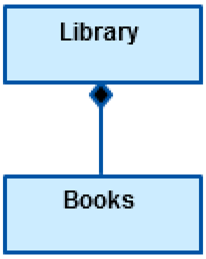


Figure 14: Composition (Source: [40])

The composition relationship is very similar to the aggregation relationship with the only difference being its key purpose of emphasizing the dependence of the contained class to the life cycle of the container class. To show a composition relationship in a UML diagram, use a directional line connecting the two classes, with a filled diamond shape adjacent to the container class and the directional arrow to the contained class.

Inheritance / Generalization

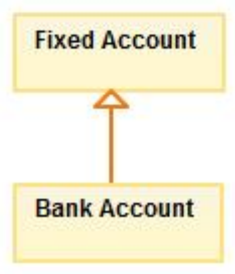


Figure 15: Inheritance (Source: [40])

Inheritance/ Generalization refers to a type of relationship wherein one associated class is a child of another by virtue of assuming the same functionalities of the parent class. In other words, the child class is a specific type of the parent class. To show inheritance in a UML diagram, a solid line from the child class to the parent class is drawn using an unfilled arrowhead.

Realization

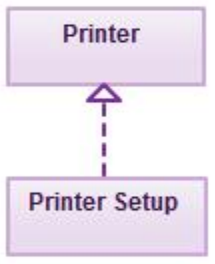


Figure 16: Realization (Source: [40])

It denotes the implementation of the functionality defined in one class by another class. To show the relationship in UML, a broken line with an unfilled solid arrowhead is drawn from the class that defines the functionality of the class that implements the function. In the example, the printing preferences that are set using the printer setup interface are being implemented by the printer.

Behavior Diagrams

These diagrams emphasize what must happen in the system being modeled. Since they illustrate the behavior of a system, they are used extensively to describe the functionality of software systems.

Use Case Diagram

A use case diagram is a diagram that shows a set of use cases and actors and their relationships; use case diagrams address the static use-case view of a system [1]. Use case diagrams are composed of three main elements, which have to be identified before creating them.

- **System Boundary**: represents the boundaries that distinguish the system from the rest of the world.
- **Actors**: are located outside the system boundary and communicate directly with the system by either sending or receiving data to/from the system, or both. Identifying the actors is very important to develop the system effectively. In order to identify them, it is necessary to specify who interacts with the system, what uses the system, and what is the system using [42] ?
- **Use cases**: are defined in as a specification of sequences or actions, including variant sequences and error sequences, that a system, subsystem, or class can perform by interacting with outside actors. The use case is initiated by the actor and written from the actor's point of view. Identification of the use cases is also important and they can be recognized by deciding what the system is supposed to do for each actor, and the way, in which each actor will use the system [42].

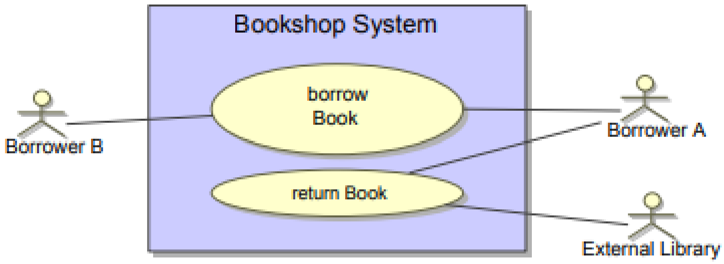


Figure 17: Sample Use Case diagram (Source: [42])

The above figure 17 presents a sample diagram that shows the main elements of use case diagram. The diagram shows a Bookshop System that offers two functionalities represented as use cases borrow book and return book. The use cases are associated with three actors and surrounded with a rectangle that specifies the boundaries.

Use Cases are a means to capture the requirements of systems, i.e., what systems are supposed to do. The key concepts specified in this clause are Actors, Use Cases, and subjects. Each Use Case's subject represents a system under consideration to which the Use Case applies. An instance of a Use Case refers to an occurrence of the emergent behavior that conforms to the corresponding Use Case. Such instances are often described by Interactions [42].

In this kind of process, use cases are a primary artifact for capturing functional requirements and establishing the desired external behavior of the system, for the verification and validation of the system design and implementation and as a communication device among the different stakeholders of the project. The Rational Unified Process and Catalysis [43] are two examples of use case driven software engineering processes based on UML. At the time of their introduction they were considered too informal and simple. Yet, it is this simplicity that makes them popular and accessible.

Why Use Cases?

Constructing use cases has many advantages. When we create use cases, we:

- Identify the different goals for individual actors.
- Capture the context that the system is supposed to live in and also the functionality required by the external actors.
- Understand the problem domain, the candidate objects for the problem domain, and also of the proposed solution, and
- Provide a model against which the final system design can be checked.

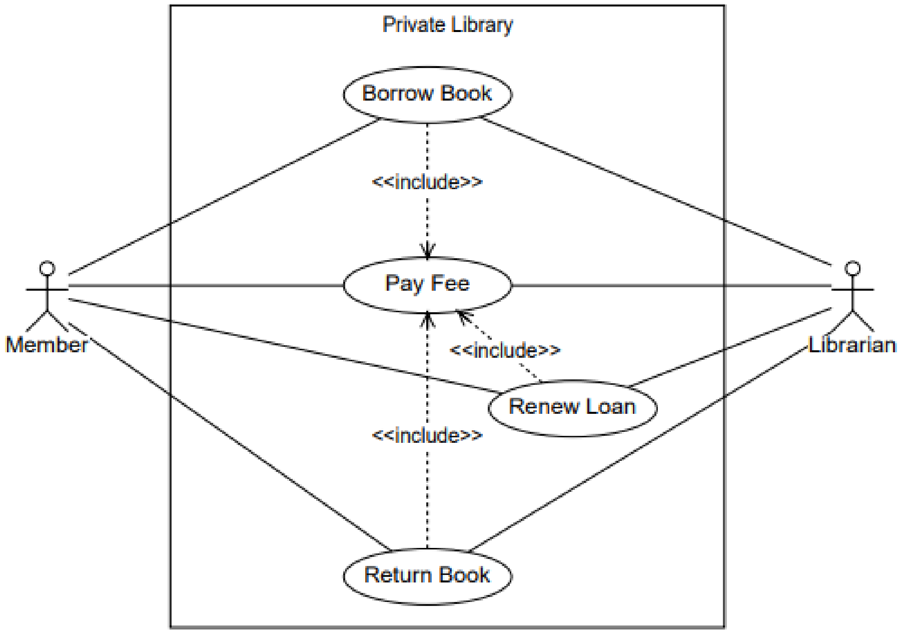


Figure 18: Sample overview of Use Case diagram (Source: [42])

However, use cases have also a number of shortcomings:

- It is difficult to check whether the system provides the functionality expected by the actors, that is, it is difficult to ensure that the actors can achieve their goals by using the system and informal requirements can be easily misinterpreted.
- They are essentially functional in character, even though they are used in UML to develop object-oriented systems. There is a missing connection between the control-oriented use case diagrams and object-oriented class diagrams.

State Diagram

A state machine diagram is a diagram that shows a state machine, with emphasis on the flow of control between states. State machine diagram is a behavior diagram which shows discrete behavior of a part of designed system through finite state transitions. State machine diagrams can also be used to express the usage protocol of part of a system. Two kinds of state machines defined in UML are:

- Behavioral State Machine, and
- Protocol State Machine [41].

An event is the specification of a noteworthy occurrence that has location in time and space. A transition is the movement from one state to another as a result of an event occurrence [42]. Each state machine should have a starting state (filled circle), from which the transition(s) begin(s), and may also have a final state (bull's eye).

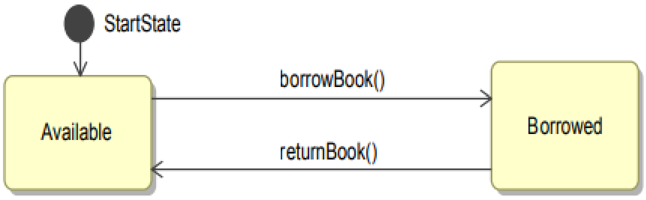


Figure 19: Sample State Machine Diagram (Source: [42])

The above figure 19 shows a sample state machine diagram for a Book object. The state machine diagram contains two states, i.e. Available and Borrowed that can be triggered by the operations, i.e. returnBook() and borrowBook() respectively to change the state of the Book object.

4. Design of Online Crime Reporting System

4.1 Overview of the System

4.1.1 Introduction

Online Crime Reporting System initiates the objective of providing the user with customized and powerful online crime reporting feature and crime management system application. The web application is built with all options such as online crime reporting, checking reported crime status, member registration for user's side as well as adding new crime records, manage crime records, manage news, manage wanted person information for admin's side. The interface provided is very user friendly and flexible for all times.

Crime, an offence or criminal offence is an act harmful not only to some individuals but also to the community or the state (Aquinas, 1988). It is also an act or the omission of an act that is hidden or the omission of a duty that is commanded by a public law and that makes the offender liable to punishment by that law (Merriam, nd). A criminal is a law breaker, convict, offender, villain, culprit, sinner, felon (Collins, 2002). A criminal is convicted when he/she is found guilty by the laws of the place where crime is committed. In Myanmar, a criminal is convicted when he/she is found guilty under the constitution which is the supreme law of the Federal Republic of Myanmar.

The crime is growing at a really fast pace in all around the world. There is large numbers of criminal activities in the society nowadays. Armed robberies, ritual killings, kidnapping, murder, assassinations, terrorisms are few of the cases of crime, which are prevalent in many countries of the world and reported in the print and electronic media.

The prevalence of crimes in the society can be traced to several factors. Primarily is the issue of unemployment, which has assumed a frightening and alarming dimension in many Countries as well as in my country, Myanmar.

The crime management system is an application for reporting crime easily by users and registering crime records and managing a criminal record in a police station. This thesis application work is focused on the design and implementation of web based Online Crime Reporting and Crime Records management System.

4.1.2 Requirement Gathering Phase

Case Study

Online Crime Reporting System is a system which can be used to report crime by normal users as well as managing crime records from admin side. This web based portal application project will be done using HTML, CSS/Bootstrap, Javascript and JQuery technologies for front end side and Java technology along with Struts framework for backend with MySQL for relational database. It can be used to report crime, manage crime by users and police officers (admin). This project is useful for normal citizens who are afraid or don't have enough time to go to police station will be able to report their legal issue or crimes easily. This system will also help to manage all the activities in a police station using computers such as managing crime records, assigning police officers to crime records, managing news and wanted person, etc.

Currently in Myanmar Police Force, all the works are done manually and by computerizing all the activities inside a police station can be managed easily and effectively.

A significant part concerning Crime reporting and crime management system is reporting the crime easily and storing the record of criminal very efficiently. A police station or station house is a building which serves to accommodate police officers and other members of staff. These buildings often contain offices and accommodation for personnel and vehicles, along with locker rooms, temporary holding cells and interview/interrogation room. In most cases it simply provides a work place.

There are mainly two entities in this thesis project,

- Citizens (Members or Visitors) who can easily report the crime and check the status of reported crimes if he/she is a registered user of the system.
- Police Officers (Admin or Investigating Officers) who can manage the reported crime records, create new information such as wanted person and news.

Why Automation

It's really useful to have a system automated for the following reasons:

- Reporting a crime in an easy and timely manner.

- Provide fast and easier methods for recording information about crime.
- Offer flexibility in updating.
- Making timely schemes changes.
- Reduce complexity.
- Improve services provided to the people.

Problem Domain

The areas that the current system needs to be examined are the difficulty in reporting crimes by citizens where citizens doesn’t want or afraid to go to the police station to report the crime and also the difficulty in traditional reporting system which often leads to time consuming for both citizens as well as police officers in managing reported crime records and it also consumes large volume of pare work and need manual calculations. There is also a concern on lack of security of data in the current system.

Solution Domain

The current crime reporting system is faced with several difficulties as there is no instant means of reporting crimes other than telephone phone calls. By using this application, people who are afraid or doesn’t have enough time to go to police station for complaint about their legal issues can give their complaint through online to register any type of complaint.

The aim of proposed system is to develop a system of improved facilities for both citizens and managing police officers. The proposed system can overcome all the limitations of the existing system. The system provides proper security and reduces the manual work such as-

- Security of data.
- Ensure data accuracy.
- Proper control of the officials related with the crime records.
- Minimize manual data entry on paper.
- Minimum time needed for the various processing.
- Greater efficiency.
- Better service.
- User friendliness and interactive.

Application Domain

The main advantage of online web based application is that, a person can report the crime anytime from anywhere with only an access to internet which nowadays everyone is having. Users can view the progress status of their reported record online. By the future work intended to develop after this thesis, user can view the case details and progress of the complaints on their mobile phones.

When citizens reported the crime, it will firstly become the FIR (First Information Record). FIR is the record which is normally prepared when the police first receive information about the commission of a crime. FIR contains the details of the informant, the details of the crime, and the date and time it was committed according to the informant. An FIR is the first step towards any sort of criminal activity. It triggers the criminal justice machinery of the State into motion. The Police officers (Admin) can then review the FIR and can approve or decline the record accordingly.

4.2 Software Requirement Specification (SRS)

The purpose of this SRS (Software Requirements Specification) document is to provide a detailed description of our software project, specifications, and goals. This document describes the features included that are included in our project, its user interface, and its functionality. The document describes how our team sees and understands the requirements of the software.

In the fast moving world, if people lack something, it is time. All are busy in their world. It will be welcomed if services are provided at their will. So the main objective of our product is better communication, better leadership, reducing crime and disorder etc. The product provides a framework within which a user can easily work with. That was our next objective. We know users are of many categories, like users from who know working with computers very well to users who didn't know about computers. So all the category can use the software. So it should be user friendly.

Scope

The scope of a single software product is defined by the bounds of the capabilities provided in that product. Similarly, the scope of a software product line is determined by the bounds of the capabilities provided by the collection of products in the product line.

4.2.1 Functional Requirements

In software engineering, a functional requirement defines a function of a software system or its component. Functional requirements are product features or functions that developers must implement to enable users to accomplish their tasks. So, it's important to make them clear both for the development team and the stakeholders. Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish. Generally, functional requirements describe system behavior under specific conditions.

The functional requirements of the project Online Crime Reporting System contains the following:

- User Registration
- Update profile information by user
- Login authentication
- Reporting Crime (FIR record) by user
- View status of reported crime record by member
- Create and manage new members by admin
- Review and approve FIR record by admin
- Create and manage crime record by admin
- Create and manage wanted person list by admin
- Manage Investigation records by police officers
- View wanted person and news by users

4.2.2 General Description of Functional Requirements

We will see some of the key features of the proposed Online Crime Reporting System in this documents.

User Registration

This includes the registration of new users to the system and new user will need to enter their personal information such as National ID number, phone number in order to register into the system.

Update Profile by User

This module allows users to update their information as needed such as name, date of birth, address, etc. When user sign up for an account, our system provides user with a randomly generated "user ID" which is identical to each user and user will not be able to change that ID.

Login Authentication

User authentication module allows the system to verify the identity of members and admin who connects with a network resource. This system will use the cookie based authentication where the client posts the login credential to the server, server verifies the credential and creates session id which is stored in server (state-full) and returned to client via set-cookie. Upon logout session id will be cleared from both client cookie and server.

Create and Manage New Members by Admin

This module allows the system admin to be able to create new members who can be police officers or another office with admin right. Admin can choose and grant the role for the new user accordingly. Admin will be able to edit the current information of existing users such as address, position, etc. whereas members can also edit those information by themselves.

Reporting Crime (FIR Records) by Users

This module will allow the user to be able to report the new crime record. This is done by entering details of the crime record, such as crime incident type, incident details, victim details, date and location of incident, etc. After FIR was reported, system will generate the unique FIR ID and user will be able to use that FIR ID in order to track the status of the reported crime.

Firstly, all newly reported records will become FIR records (First Information Report). Each FIR is important as it sets the process of criminal justice in motion. It is only after the FIR is registered in the police station that the police take up investigation of most types of cases. After the admin review and approve the FIR record, it will becomes the crime record handling by the police officer who was assigned by the admin while approving the record.

View Status of Reported Crime by Users

This module will allow the registered members to view the status of the reported crime. However, in order to use this functionality, user needed to be register in the system. Status of the record will be “Registered”, “Suspended”, “Initial Investigation by Police”, “Further Investigation”, “Suspect and Evidential Evaluation”, “Charged” and “Closed”.

When the user reported the crime record, it will recorded as “Registered” status. After admin has review and approve, then the status will become “New Crime Record” and it can be the initially investigation record when the assignee police officer has been assigned by admin. If admin reject the record, the status will become “Suspended” and it will become “Closed” when admin close the request. When the police officer start investigating the record, he/she can change the status of the record to “Investigation Record in Progress” if necessary. As soon as the police officer has investigate the record with suspect and evidence, status can be updated to “Suspect and Evidential Evaluation”. After the suspect has been charged by officer, status can then be updated eventually to “Closed” by providing closure note by assignee police officer.

Review and Approve FIR Record by Admin

This module allows the admin to review and approve the reported FIR records reported by users. Admin can also to edit the records if necessary during the approval process. After FIR has been approved, it will become the crime records and admin can then assign the police office as an assignee to handle the ongoing investigation of that record.

Create and Manage Crime Record by Admin

This module allows the admin to register the new crime records by entering details of the crime record, such as crime incident type, incident details, victim details, date and location of incident as well as assigning the police officer for ongoing investigation. Admin can also edit and update the records as necessary.

Create and Manage Wanted Person List by Admin

This module allows the admin to create the wanted person which will allow the visitor of the website to see those created most wanted person information for the public knowledge. Admin can also update and edit those records.

Manage Investigation Records by Police Officers

This module allows the police officer to be able to update the status of assigned crime records. Assignee police officer will also be able to put the information of suspects, evidences details, progress of investigation and work note by using this functionality.

4.2.3 Non-functional Requirements

Non-functional requirements describe how a system must behave and establish constraints of its functionality. This type of requirements is also known as the system's quality attributes.

The non-functional requirements from the system will be: -

- Secure access of confidential data (crime records details, police officer details, etc.).
- 24 x 7 availability in a networked environment.
- Better component design to get better performance at peak time.
- System will be able to handle the large volume of users that keeps increasing for a better scalability.

4.3 Analysis Model

4.3.1 Data Dictionary

- **Visitor:** A person who is just visiting the website in order to report crimes or view information on the website.
- **Member:** A person who is registered in the system with all the citizens information such as name, national ID number, phone number, address, etc.
- **Admin:** A person who is registered with a police administrative officer role in the system who can manage crime records, approve FIR records, manage users of the system.
- **Police Officer:** A person who has been created and assigned to each specific role by the police administrative officer in order to manage the assigned crime records by updating the investigation record.
- **FIR Record:** A crime record details which has been reported by the users (visitor, member) of the system.
- **FIR Dismissed:** A crime record which has been declined by the police administrative officer with the given explanation reason.
- **Crime Record:** A crime record which has been created by the police administrative officer or an approved FIR record.
- **Witness:** A record of person which has been entered into the system as a witness by the assignee police officer of the specific crime record.
- **Suspect:** A record of person which has been entered into the system as a suspect by the assignee police officer of the specific crime record.
- **Wanted Person:** A record of person which has been entered into the system as wanted person by the police administrative officer.

4.3.2 Class Model

In class modeling, the static structure of the system that will be built. In particular, it illustrates the things such as classes, types, their internal structure and their relationships to each other which will illustrate and create a functional diagram of the system classes and serves as a system development resource within the software development life cycle.

4.2.2.1 Class Diagram

The following figure 20 shows the classes define the methods and variables in an object, which is a specific entity in a program or the unit of code representing that entity and those are describes as mentioned in the Online Crime Reporting System requirements section.

As we can see in the following figure 20, the classes are arranged in groups that share common characteristics. We can see the classes of Admin, Member, Visitor and Police Officer who are the main actors of this OCRS system. The relationships and multiplicity between those classes were also illustrated in the diagram. As we can also see from this figure, there is an association class between the Admin and Crime Record classes and it also reflected the requirements mentioned in the OCRS system requirements section where admin need to approve the FIR records to become the crime records. As illustrated in the figure, Crime Record is a class which has aggregation relationship with the Investigation Record class and that Investigation Record class is related with Assignee, Witness and Suspect classes with composition relationship.

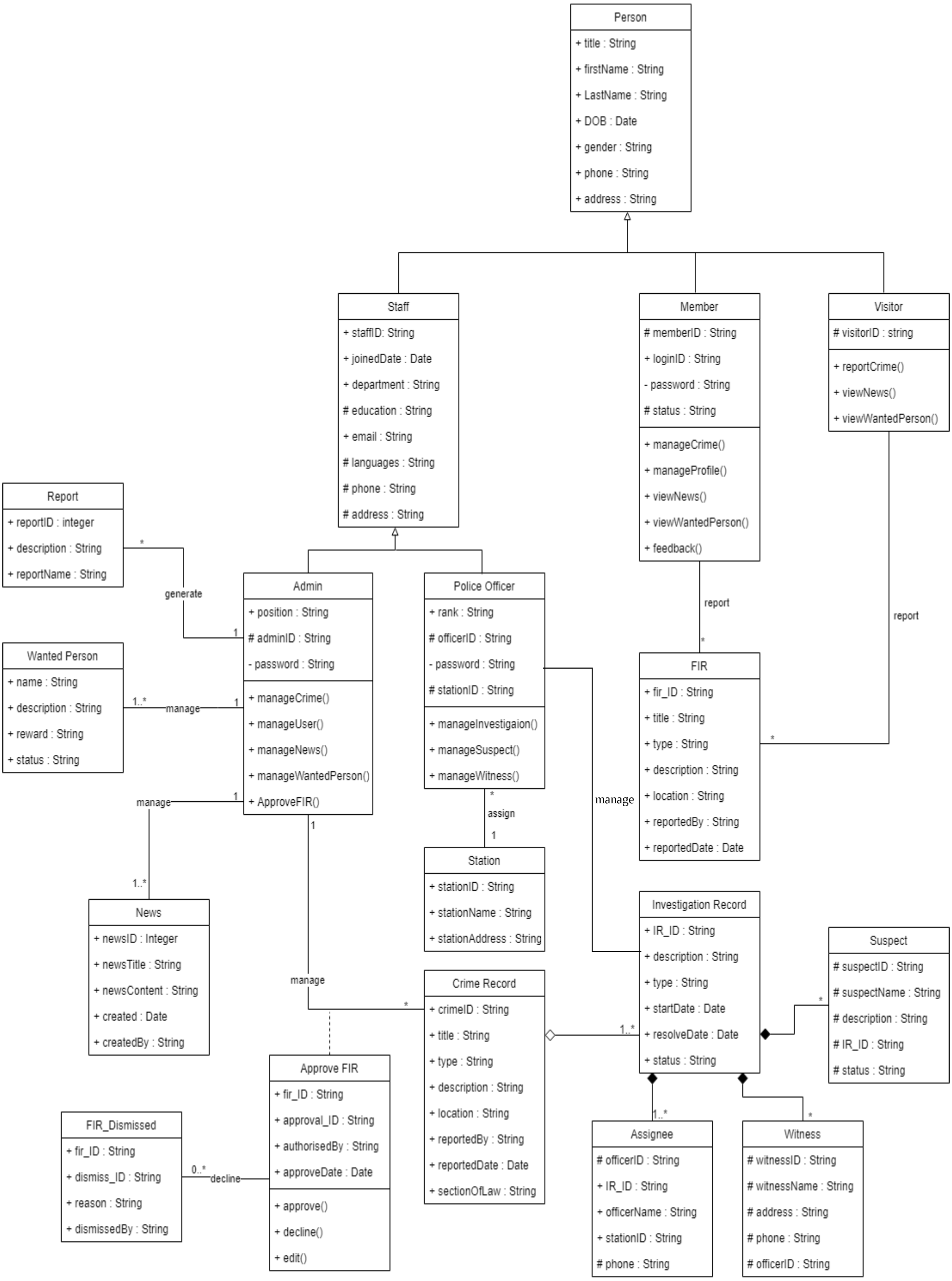


Figure 20: Class diagram of OCRS system (Source: [author])

4.3.3 Interaction Model

4.3.3.1 Use Case Diagrams

There will be two separate use case diagrams for the OCRS system; one is use case diagram for the admin and one will be the use case diagram for the user (citizen). A use case diagram for the admin will show the interaction between admin, police officers and a system where use case diagram for the user will show the interaction between members, visitors and a system. Those diagrams will also capture the goal of the users and the responsibility of the system to its users.

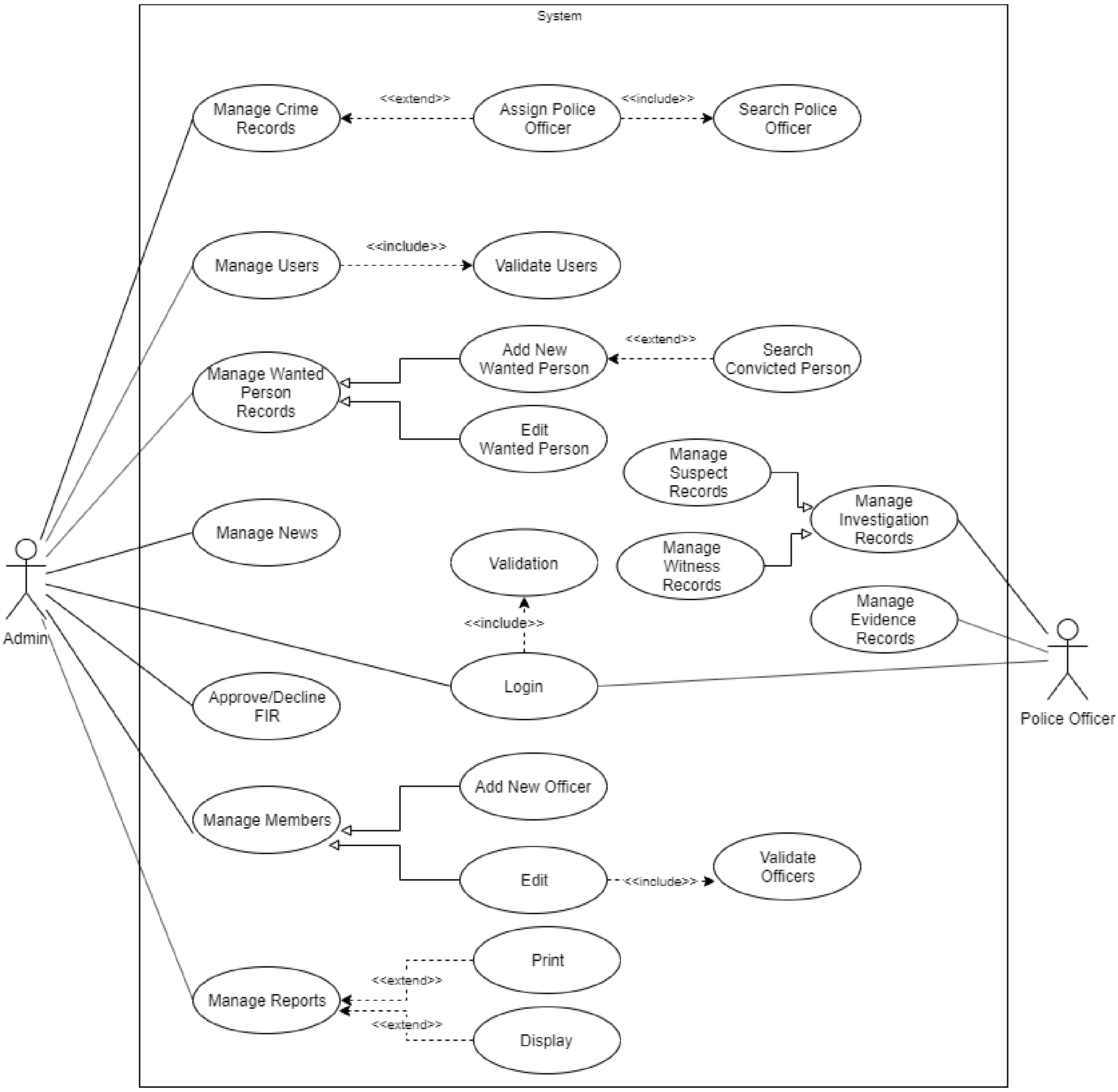


Figure 21: Use Case Diagram for the admin of OCRS System (Source: [author])

As we can see in above figure 21, admin can manage the crime records by adding new records or assigning the police officer to the specific crime record. Admin can also manage users, manage news and manage wanted person records accordingly by adding and updating the records. Police officer can manage evidence records and investigation records as well as manage suspect records and witness records.

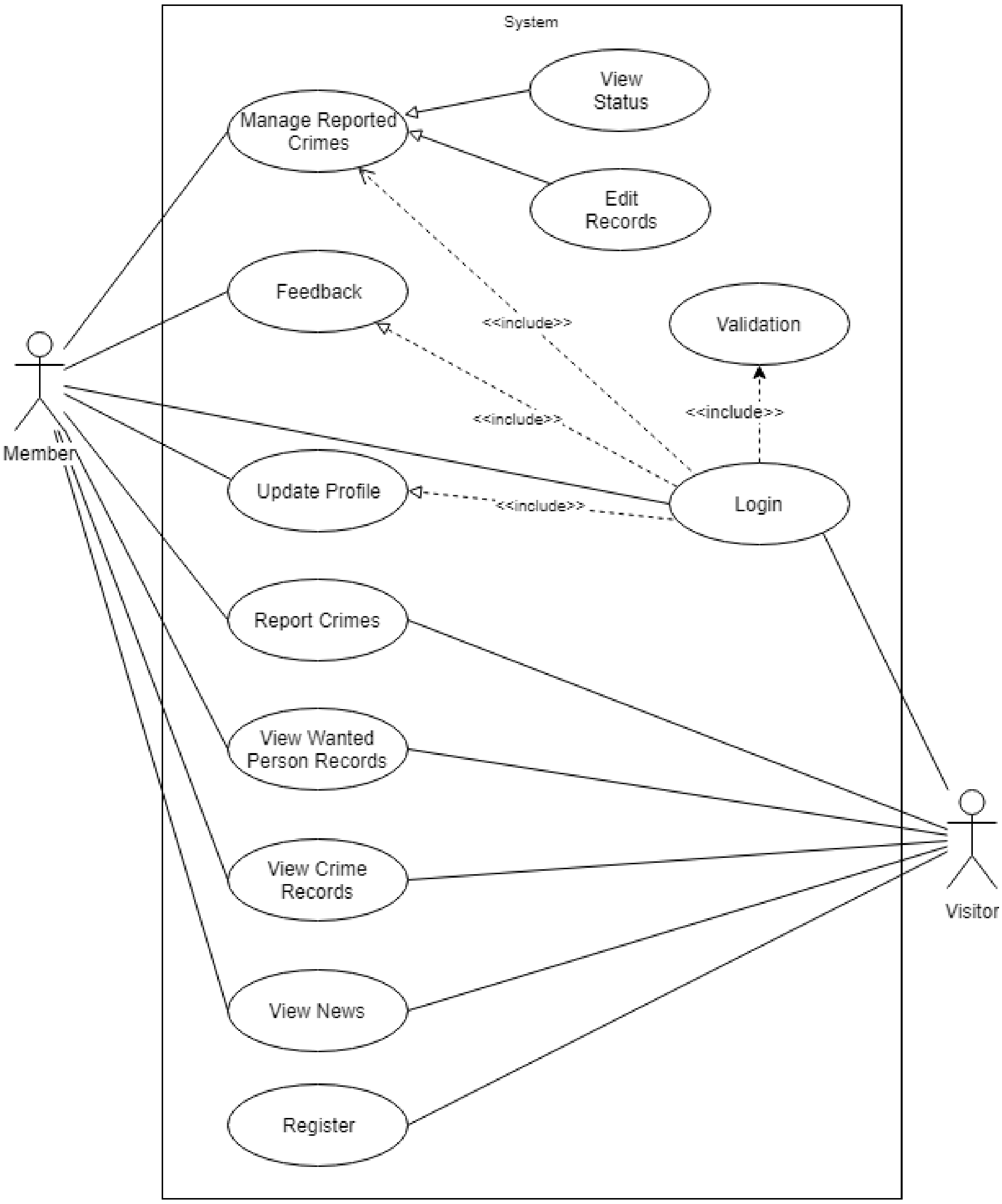


Figure 22: Use Case Diagram for the user of OCRS System (Source: [author])

As we can see in above figure 22, visitor of the OCRS system will be able to register as a member and they can also view the crime records, news and wanted person information available on the system. Visitor can also able to report the crime which will become the FIR record to be approved by the admin. Unlike the visitor, registered member of the OCRS system will be able to check the status of their reported crime records by logging into the system.

4.3.3.2 Sequence Diagrams

Sequence diagrams are used to model the logic of the scenarios or the description of the potential way of the system used. It will describe interactions among classes in terms of an exchange of messages over time.

Sequence Diagram of Login Validation

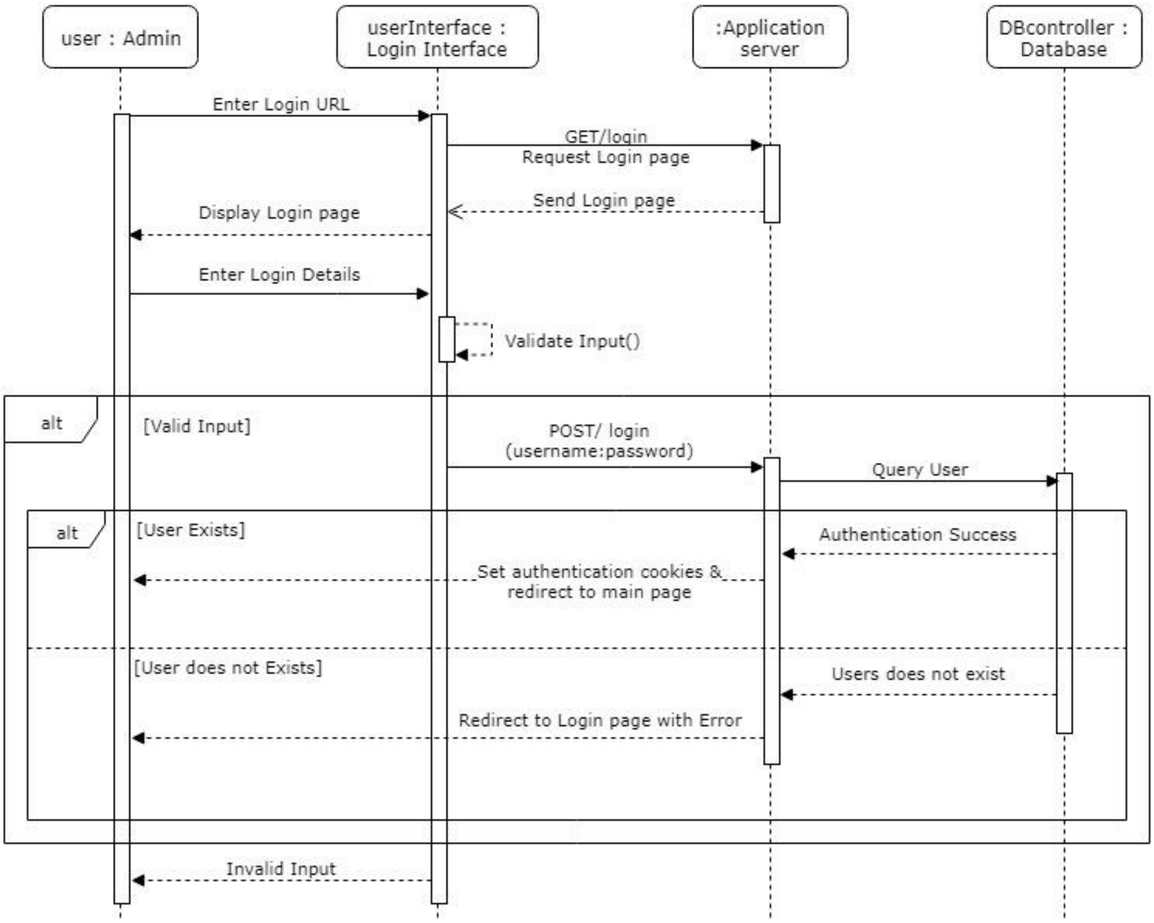


Figure 23: Sequence Diagram of Login Validation (Source: [author])

The above figure 23 shows the sequence diagram flow of the login validation.

Scenario: Admin try to login to the system.

Admin as a user will connect to the system by typing the URL in the system or by clicking the login button on the main landing page. The interface will then request login page from application server with the GET method and display the login form in order to fill the login credentials by user. Entered credentials will also validate by interface in order to check the empty field. The login credentials entered by user will then be send to application server by

POST method and application will validate against the information stored in database server. If the login authentication is successful, system creates session id which is stored in server (state-full) and returned to client via set-cookie. Upon logout session id will be cleared from both client cookie and server.

Sequence Diagram of Manage User by Admin

The following figure 24 shows the sequence diagram of manage user by admin.

Scenario: Admin will manage user by adding, editing and deleting user in the system.

When the admin click the manage user button which appeared on the admin interface, admin will then able to add, edit and delete the users in the system. If the admin choose the add new user option, system will redirect to the add new user page with the new user form which will allows admin to create the new users (members, police officers) into the system. Admin can also choose the specific role for the new user accordingly and after validation of the input form, inserted data will be saved into the database table.

If the admin choose the edit user option, system will redirect to the edit user page and also display the existing user list which has been loaded from the database table. Admin can then choose the specific user and edit the information of the user. After the input validation has been done successfully, newly edited data will be updated into the database table with specific user ID. Likewise for the delete option, admin will be redirected to the delete user page which has been displayed with existing user lists from the database table. Admin can choose the specific user from the list and delete the user.

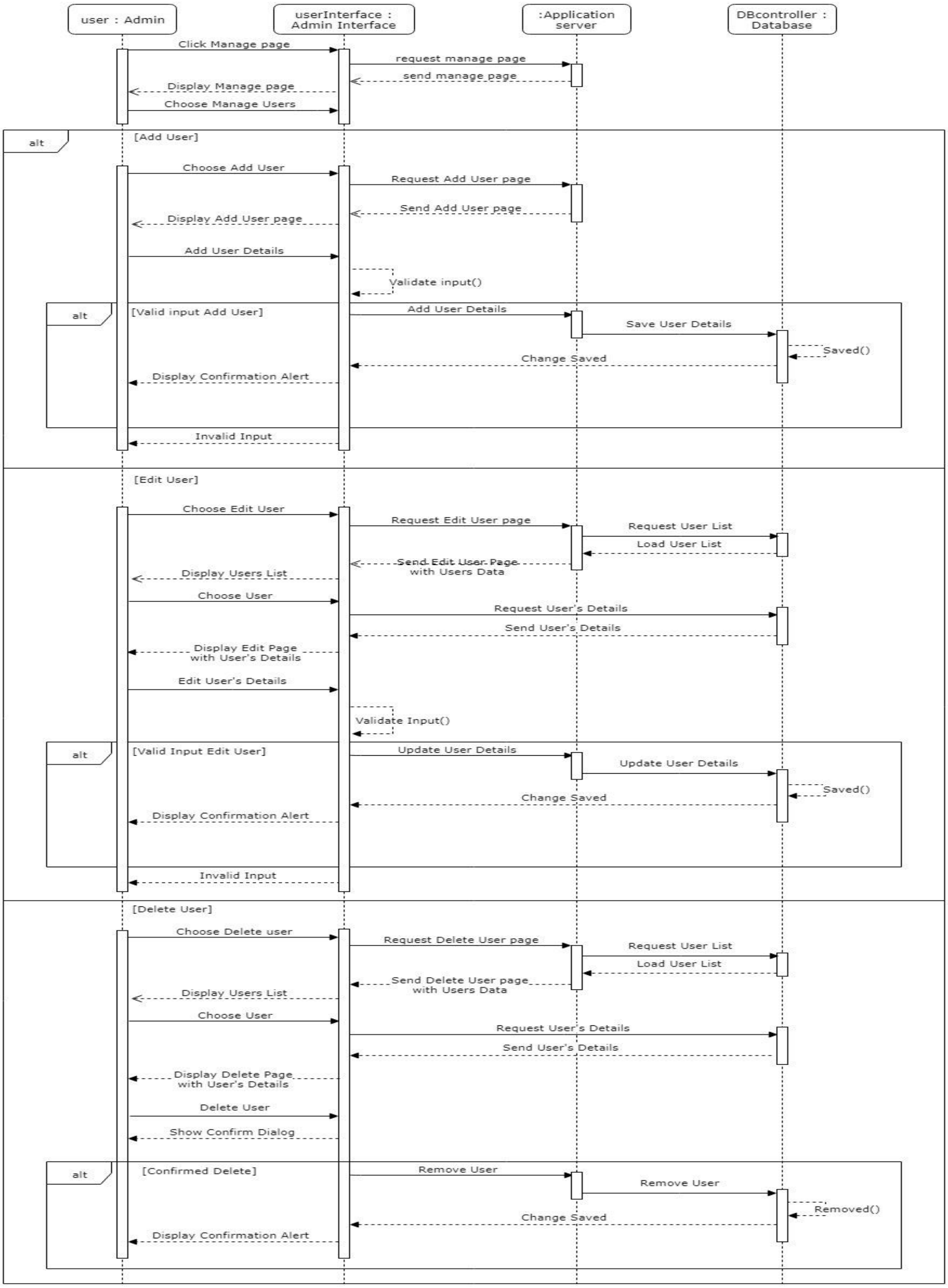


Figure 24: Sequence Diagram of Manage User by Admin (Source: [author])

Sequence Diagram of Report Crime Record by Users

The following diagram 25 shows the sequence diagram of report crime record (FIR record) by users of the system.

Scenario: User will report the crime by clicking report crime button in interface.

When users including visitors and members click the report crime button on the landing page of the system, it will be redirected to the report crime page which included the report crime form. Users will then need to fill up all the mandatory field in the form and those fields will also be validated when user click submit button. After successful input validation, system will create those newly reported record as the FIR record and saved into the database. System will also display the FIR ID after the successful submission of the record to user, so that user can able to check the reported record in the system as well as in the actual police station in the future.

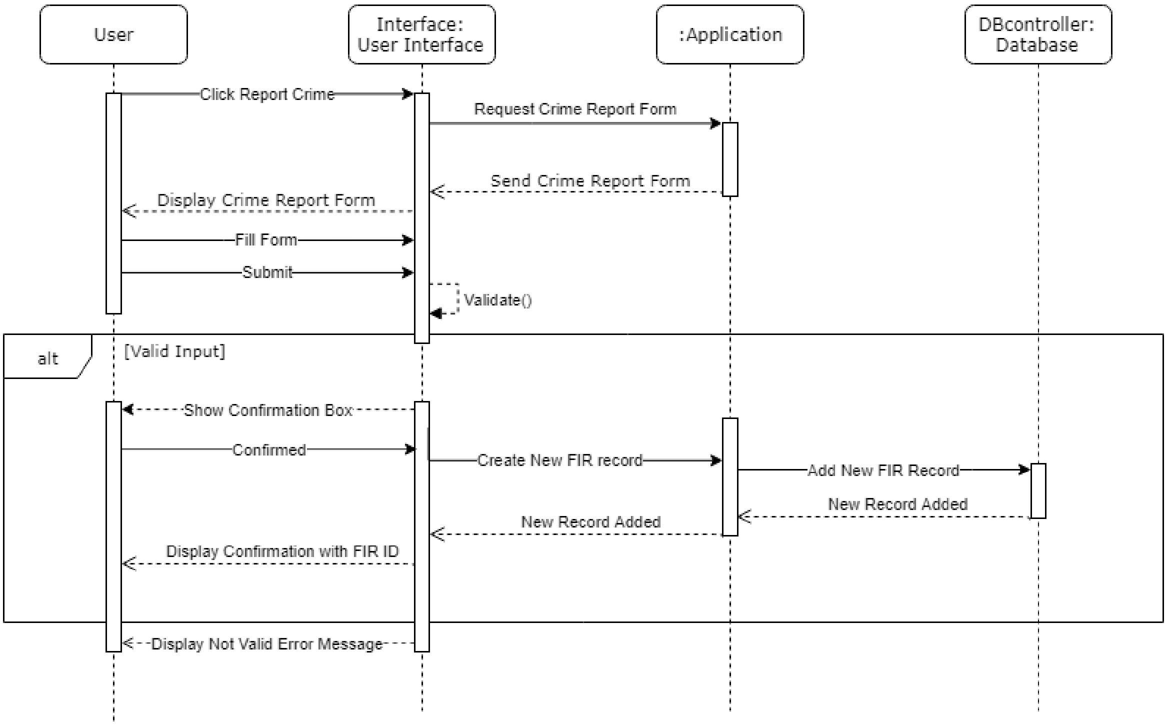


Figure 25: Sequence Diagram of Report Crime by Users (Source: [author])

Sequence Diagram of Manage Crime Records by Members

The following diagram 26 shows the sequence diagram of manage crime records by registered user (member) of the system.

Scenario: Member will manage the reported crime records by checking the current status, editing the deleting the records in the system.

When the registered user (member) of the system has been successfully login to the system, they are allow to manage their own reported crime records by clicking manage reported crime records button. System will then display the user's reported crime records by requesting data from database with the specific logged in user ID. Member will be able to check the status of the reported crime by clicking the check status button and will be able to see the specific reported crime details. Member can also edit their reported crime records (FIR) information before it has been approved by the admin. If the reported FIR record has been approved by admin, member will not be able to edit the record anymore. They can only see them in the check status page. This system also allows the member to delete their reported crime (FIR) records and it can also be done only before the reported record has been processed and approved by the admin.

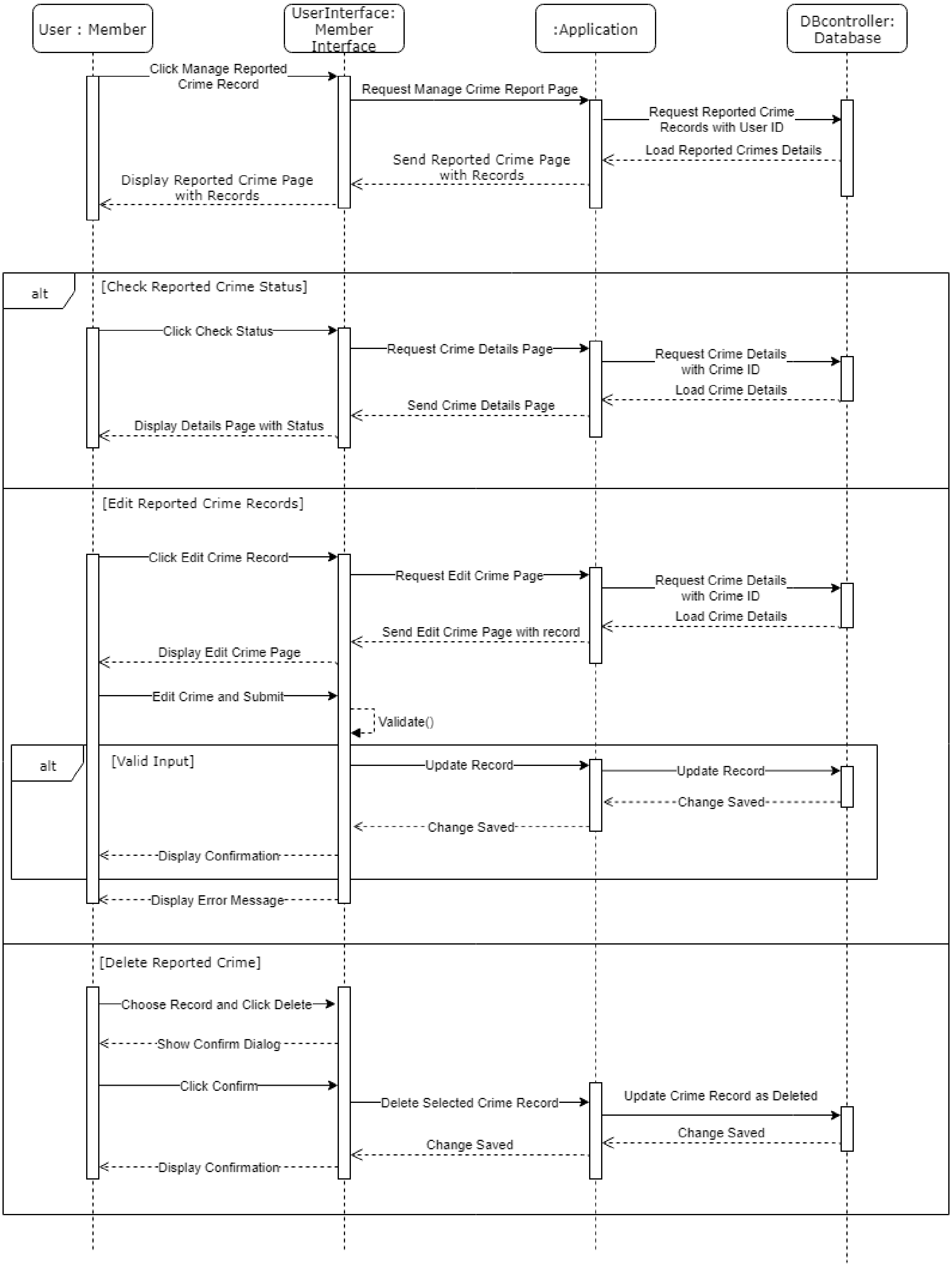

User : Member
UserInterface: Member Interface
:Application
DBcontroller: Database
Click Manage Reported Crime Record
Request Manage Crime Report Page
Request Reported Crime Records with User ID
Load Reported Crimes Details
Send Reported Crime Page with Records
Display Reported Crime Page with Records
alt
[Check Reported Crime Status]
Click Check Status
Request Crime Details Page
Request Crime Details with Crime ID
Load Crime Details
Send Crime Details Page
Display Details Page with Status
[Edit Reported Crime Records]
Click Edit Crime Record
Request Edit Crime Page
Request Crime Details with Crime ID
Load Crime Details
Send Edit Crime Page with record
Display Edit Crime Page
Edit Crime and Submit
Validate()
alt
[Valid Input]
Update Record
Update Record
Change Saved
Change Saved
Display Confirmation
Display Error Message
[Delete Reported Crime]
Choose Record and Click Delete
Show Confirm Dialog
Click Confirm
Delete Selected Crime Record
Update Crime Record as Deleted
Change Saved
Change Saved
Display Confirmation

Figure 26: Sequence Diagram of Manage Crime Records by Member (Source: [author])

Sequence Diagram of Approve FIR record by Admin

The following diagram 27 shows the sequence diagram of approve FIR (First Information Report) records by admin.

Scenario: Admin will review, edit if necessary and approve the newly reported FIR records.

When the admin click the approve FIR record button in the landing page of the system, it will be redirected to the approve FIR record page which will display the newly reported FIR records which need to be approve by admin in order to become the active crime records. Admin can then review each record by clicking on each record. Admin can also edit the reported FIR records as necessary and approve them. After approving the record, system will show the police office list to the interface where admin need to choose and assign to the recently approved records since every crime records need to have an active police officer. Admin can also reject the FIR record if needed and in that case, admin will need provide the reason of rejection by filling out the reject form which will be provided by the system.

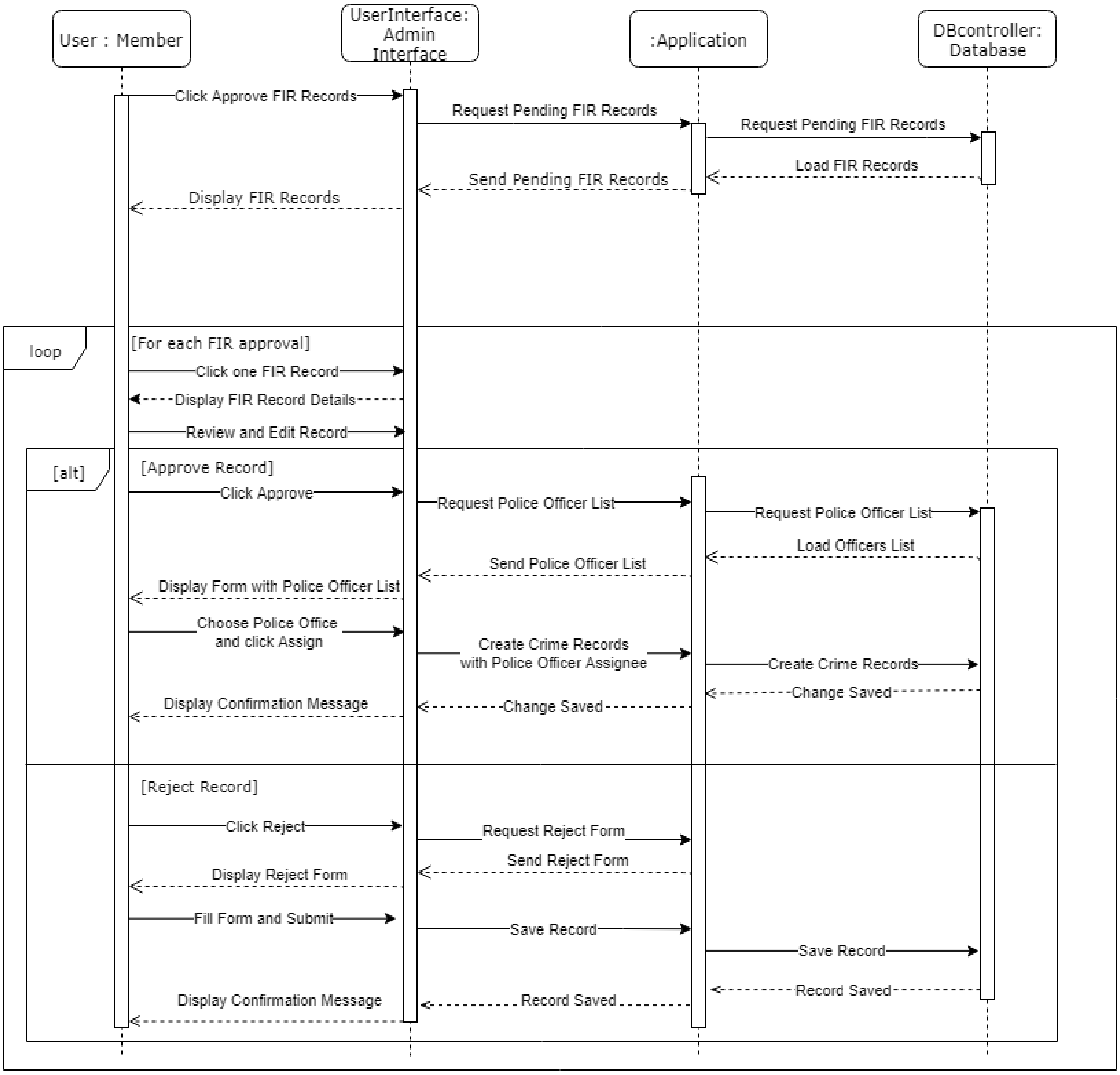


Figure 27: Sequence Diagram of Approve FIR record by Admin (Source: [author])

4.3.3.2 Sequence Diagrams

Activity Diagram of Login Validation

The following figure 28 shows the login validation activity and admin or registered user of the system will be able to enter the login credentials via login page and it will verify against the database. If the validation is successful, system will redirect the user to home page and it will show the error pop up in case of unsuccessful validation.

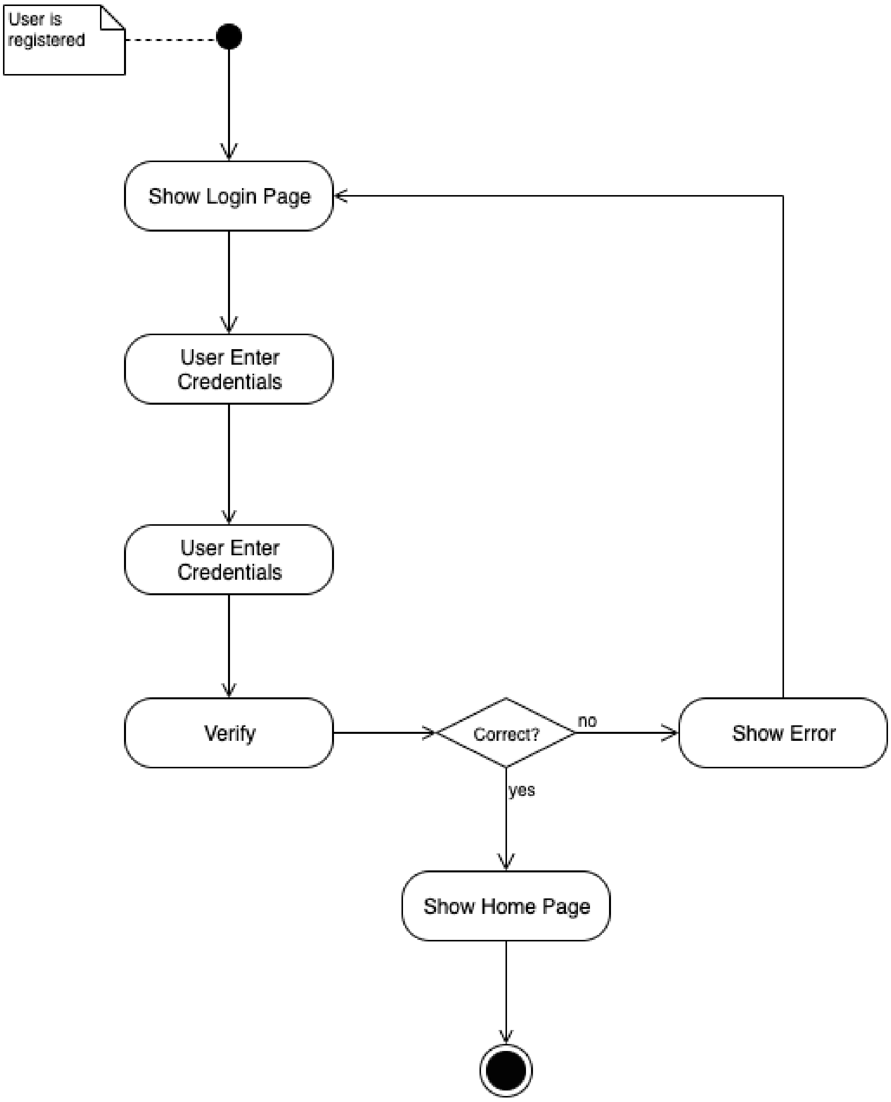


Figure 28: Activity Diagram of Login Validation (Source: [author])

Activity Diagram of Report Crime by User

The following figure 29 shows the activity diagram of the report crime by user. User will be able to report crime just by clicking the Report Crime button and system will ask the user to login or not. If user decided to login, then system will validate the user and after successful validation, user will be able to report crime by filling out the necessary form. After that, filled information in the form will be validated and system will create the FIR (First Information Report) record only after the successful form validation. Admin can then approve the FIR record in order the record to become the crime record.

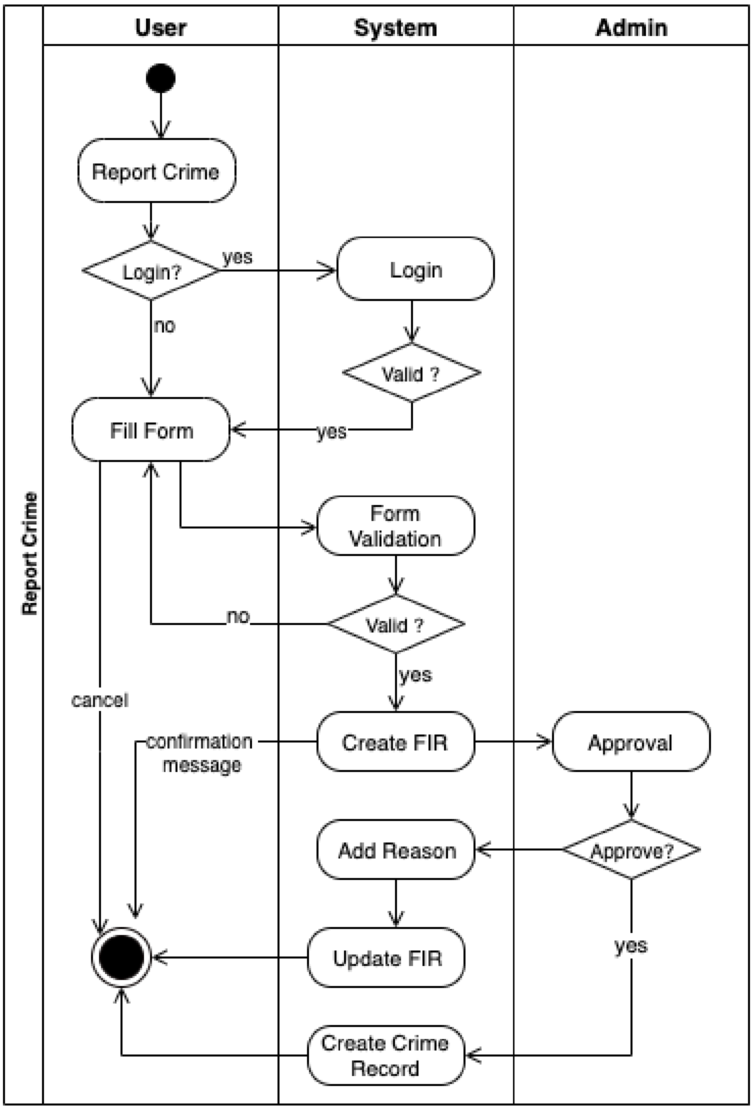


Figure 29: Activity Diagram of Report Crime by User (Source: [author])

Activity Diagram of Manage Crime Records by Members

The following figure 30 shows the activity diagram of managing the reported crime records by members.

After successful login process by user, system will retrieve the reported crime records by logged in user’s ID and display it to user. User can then choose the different options in the home page; including check status, edit record and delete record. If user chooses the check status option for specific reported FIR record, system will retrieve the selected FIR record details information and show it to user. If user chooses edit record option, system will then validate the input of the user and saved it to the database only after successful validation. If user chooses the delete record option, system will show the confirmation pop up message box and will delete the record from database after user confirm it.

Activity Diagram of Manage Crime Record by Admin

The following figure 31 shows the activity diagram of manage crime record by admin.

After the successful login process by admin, system will redirect to the admin home page. After clicking the manage crime record button by admin, system will show the two different options where admin can choose from add new crime records to edit crime records. If admin choose add new crime records option, system will validate the input form filled by admin and after successful validation, it will create the new crime record in the system. Admin can also edit the crime records option and system will retrieve the existing crime records from the database and display them on the interface where admin can choose and edit the record accordingly.

Activity Diagram of Manage Crime Records by Members

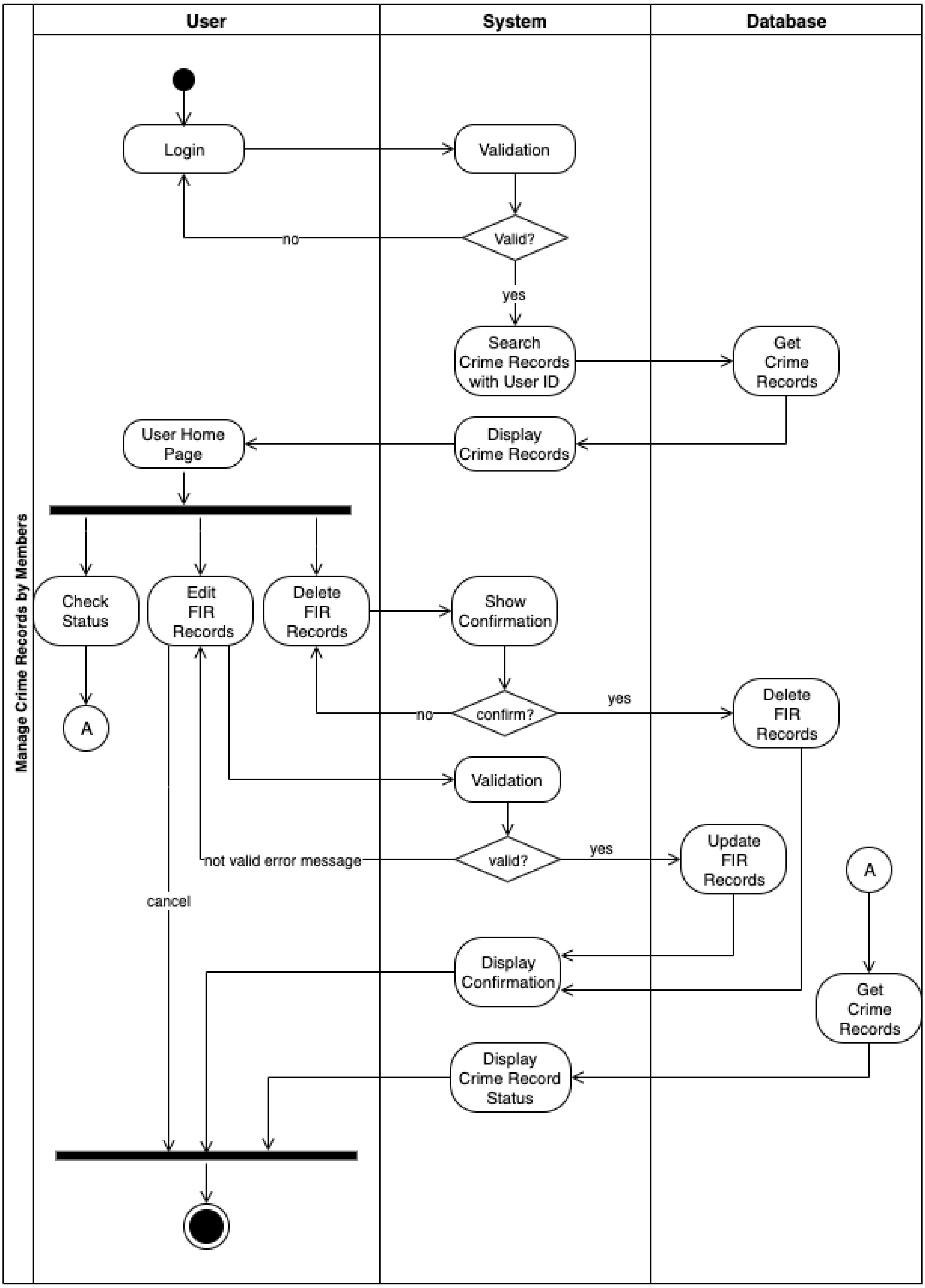


Figure 30: Activity Diagram of Manage Crime Records by Members (Source: [author])

Activity Diagram of Manage Crime Record by Admin

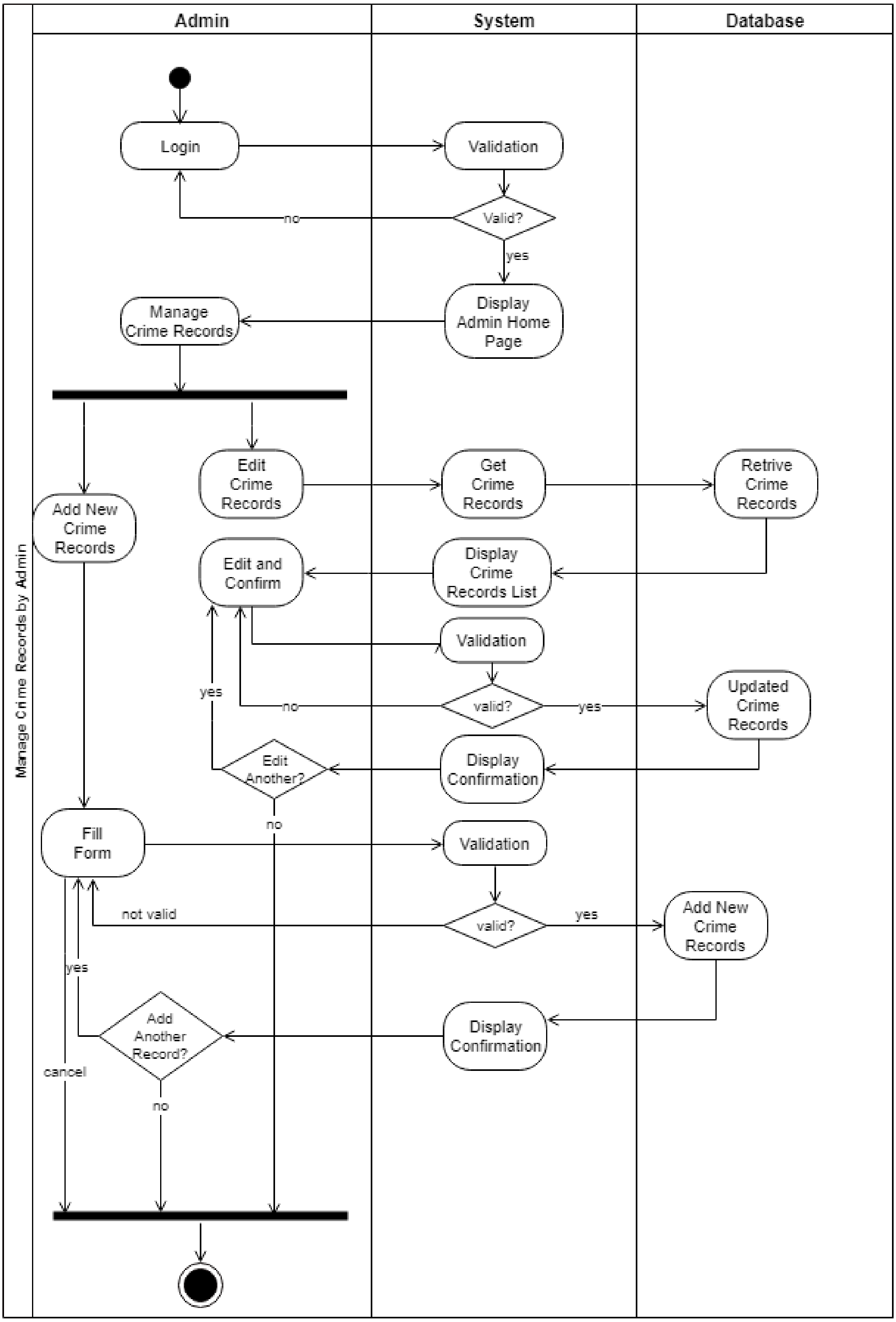


Figure 31: Activity Diagram of Manage Crime Record by Admin (Source: [author])

Activity Diagram of Approve FIR Record by Admin

The following diagram 32 shows the approve FIR (First Information Report) record by admin.

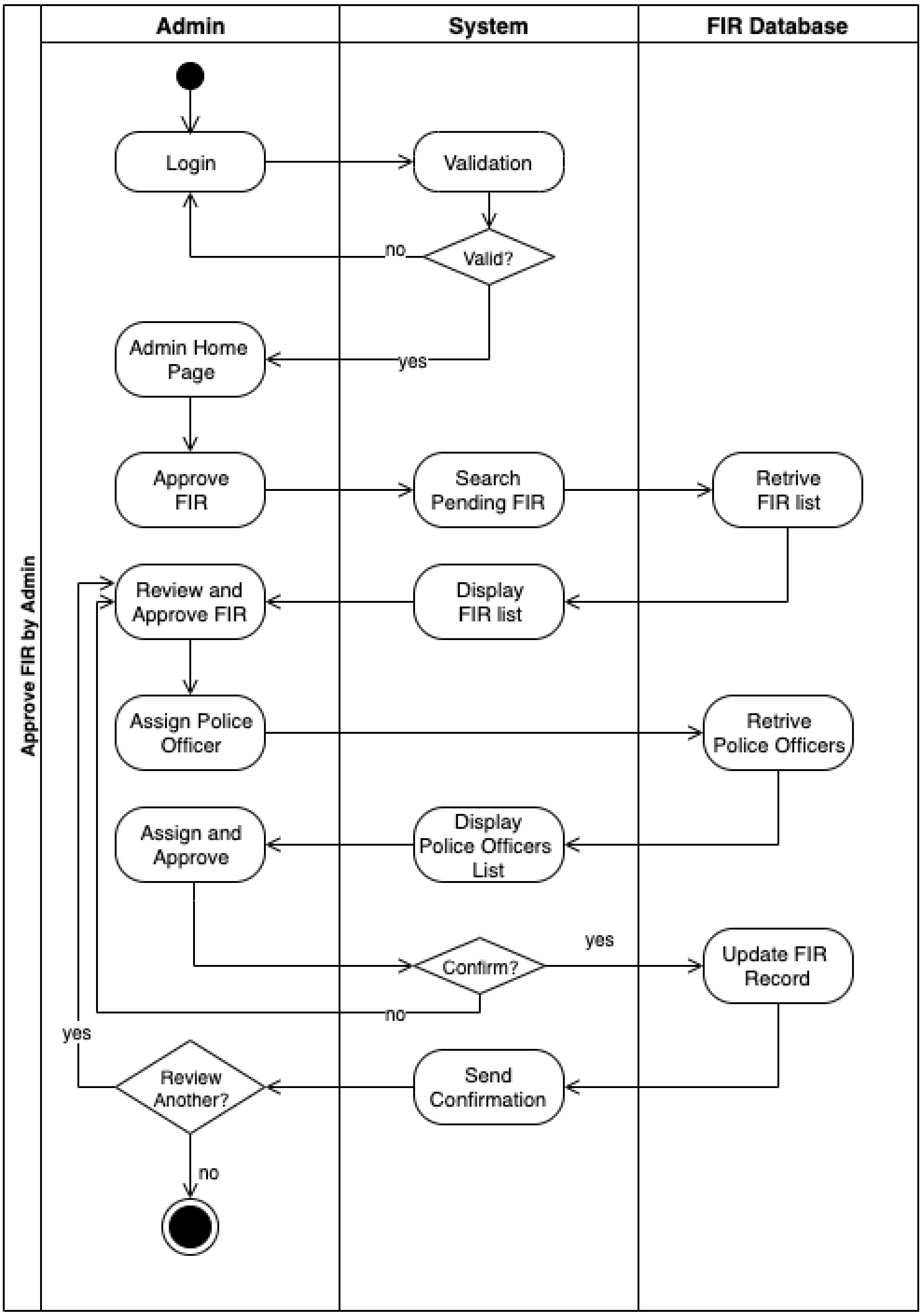


Figure 32: Activity Diagram of Approve FIR Record by Admin (Source: [author])

4.3.4 State Model

A state diagram is a graphic representation of a state machine. It illustrates a behavioral model consisting of states, transitions, and actions, as well as the events that affect these.

State Diagram

State Diagram of Report Crime by User

The following figure 33 shows the state diagram of report crime by user of the system. The state starts with the home page of the user. User report crime and then system will display the report form that user need to fill. After user submitted and successful verification, newly FIR (First Information Report) record will be created and user will see the confirmation message along with FIR ID in order to check it in the future.

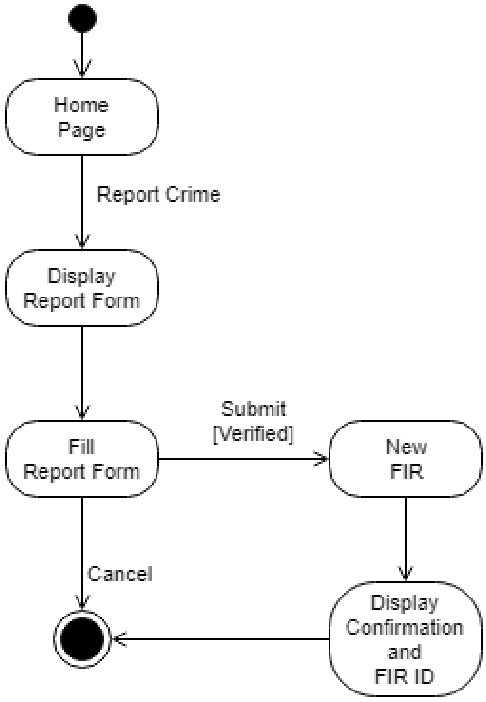


Figure 33: State Diagram of Report Crime by User (Source: [author])

State Diagram of Approve FIR record by Admin

The following figure 34 shows the state diagram of approve FIR record by admin. The state starts with the new FIR record which was initially created by user. Admin will then validate the FIR record and approve if it is valid. Approved record will then need to assign the police officer. After assigning the police officer, crime record can become the investigation record handled by the assignee police officer.

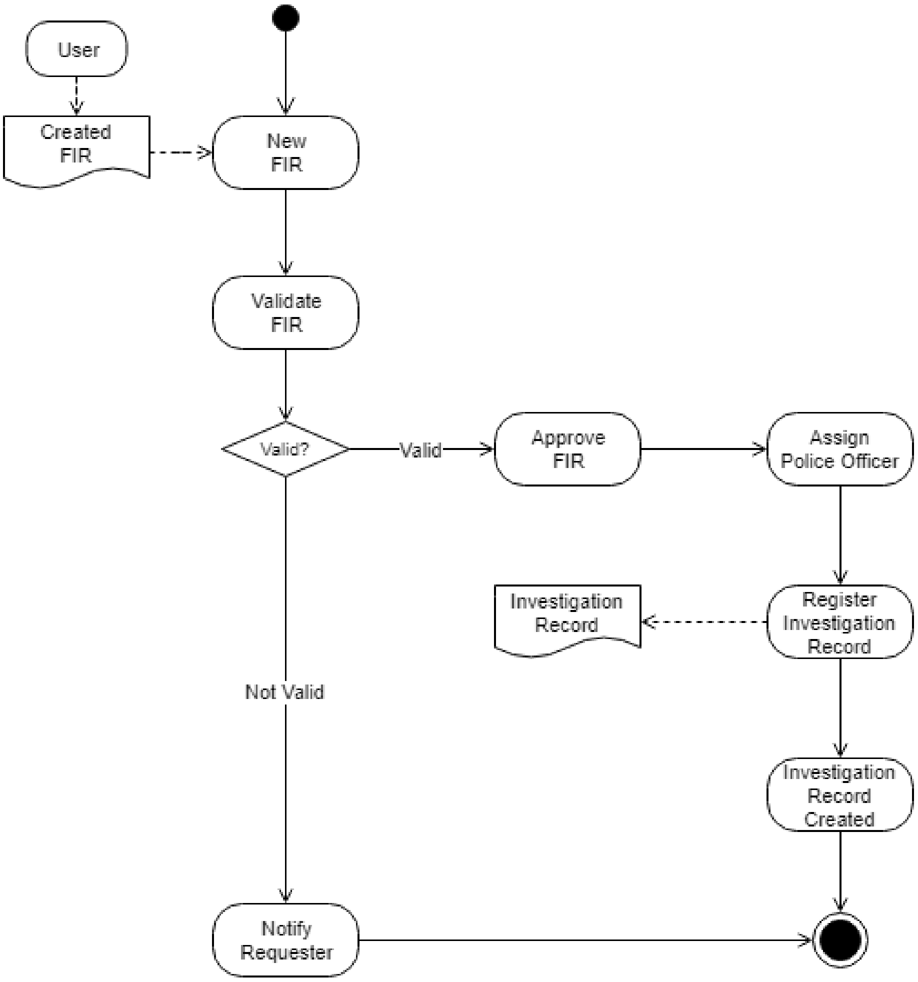


Figure 34: State Diagram of Approve FIR Record by Admin (Source: [author])

Crime Record Status State Diagram

The following figure 35 shows the state diagram for different state life cycle of crime record. When the user reported the new crime, it will first become the new FIR (First Information Report) record. After verification and approval of the FIR record, it will become the new Crime Record which can then assign to specific police officer and it will become the new Investigation Record. Investigation Record can be updated by assignee police officer as Investigation Record in Progress as well as Closed Investigation Record. If the new FIR record has not been approved, then it will become the Suspended Crime Record.

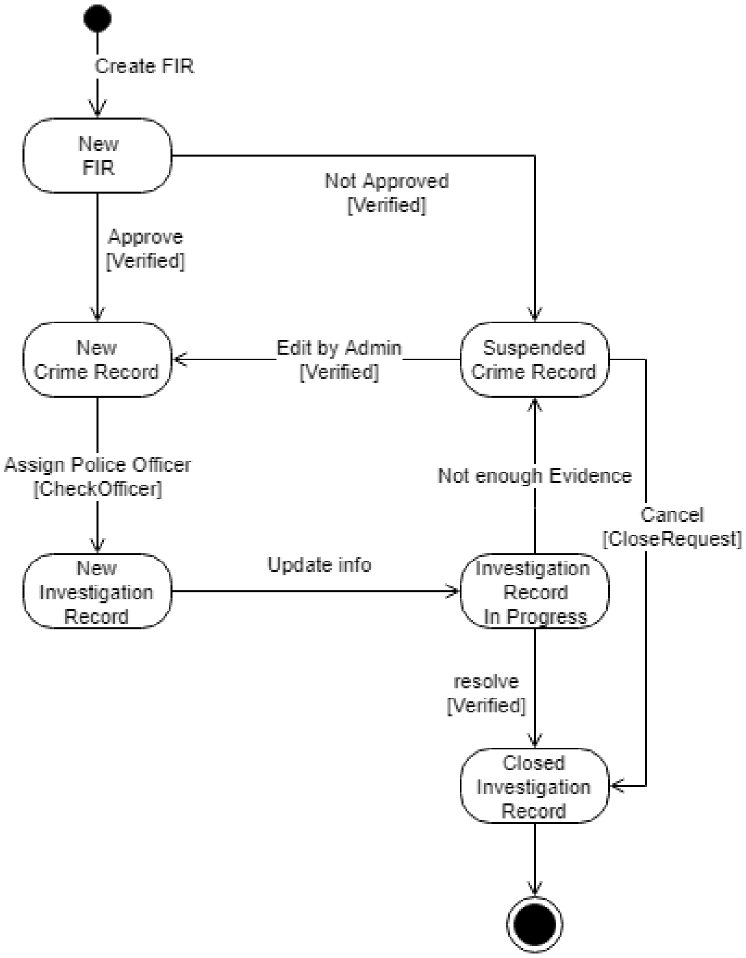


Figure 35: Crime Record Status State Diagram (Source: [author])

4.4 Architecture of the System

The architecture of the system is the general approach given to develop the application to address the issue of the given problem statement. The purpose is to determine how to build the system and to obtain information required to drive the necessary implementation of the system. The focus is mainly on the solution domain rather than on the problem domain. The proposed architecture and proposed system design will contain the object oriented design consisting transformation of the analysis model into the design model.

4.4.1 Proposed System Architecture

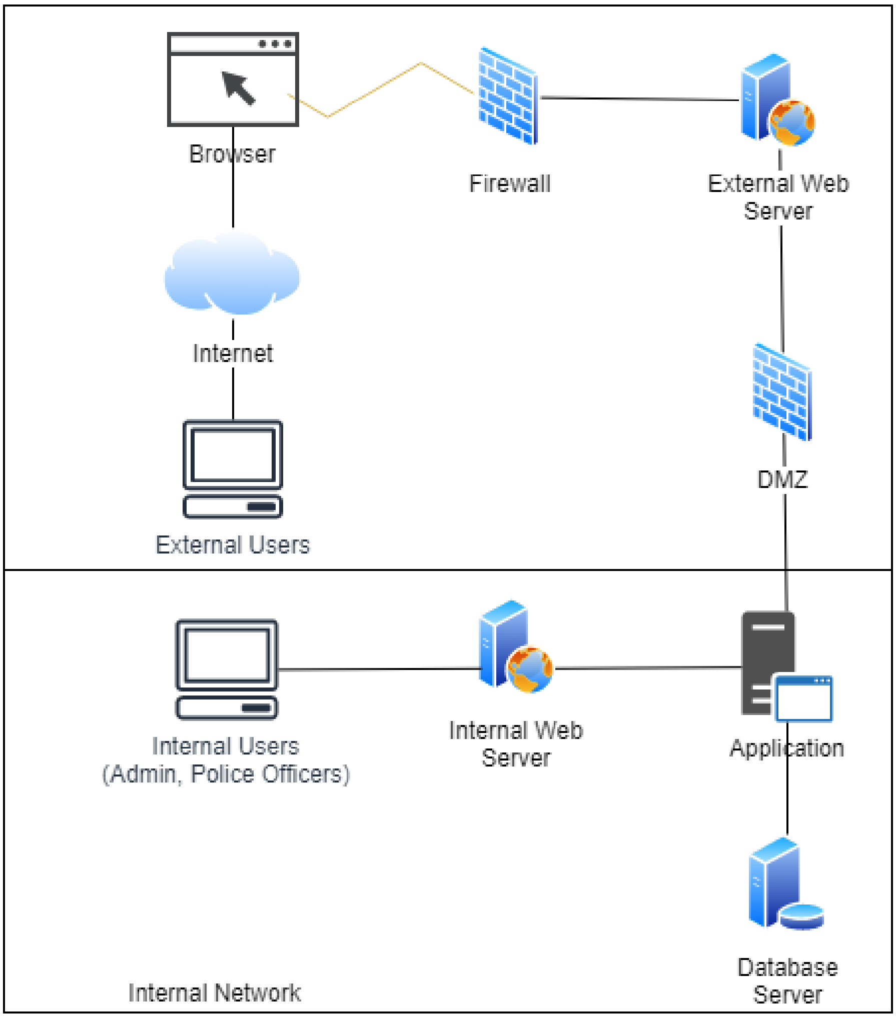


Figure 36: System Architecture Diagram of OCRS (Source: [author])

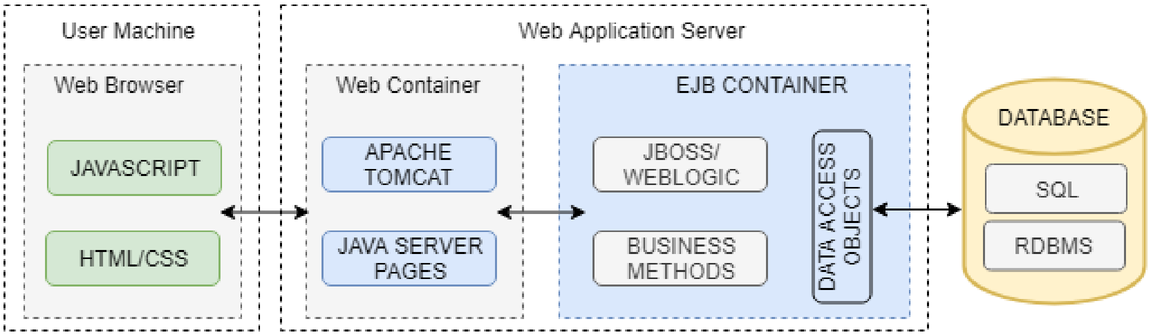


Figure 37: Overview System Architecture Diagram of OCRS (Source: [author])

4.4.2 Layered Class Type Architecture

Since our proposed system modeling approach is object oriented, we have to describe the system in term of its architecture including its hardware and software mapping, persistency management and control flow of the system. This layered class type architecture is the general approach to development and design part of the Online Crime Reporting System. This architecture will cover the different kinds of class type architectures, such as system decomposition, user interface layer, business layer, domain layer, user interface layer, persistent layer and database layer.

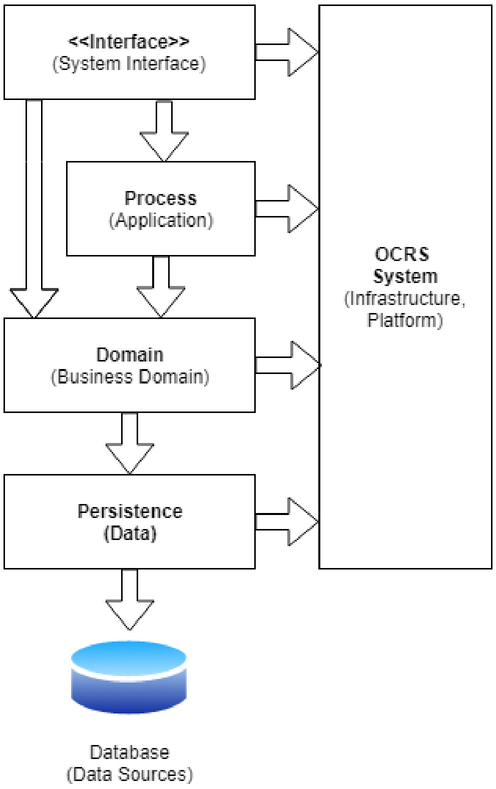


Figure 38: Layered Class Type Architecture (Source: [author])

4.4.3 Deployment Diagram

A deployment diagram depicts the configuration of run time processing nodes and the components include in our proposed system. It will show the structure diagram used in modeling the physical aspects of an object-oriented system and capture the hardware that will be used to implement the system and the connections between those different kinds of hardware components containing in this proposed system. In other words, deployment diagrams shows the hardware of the proposed system, the software that is implemented on its hardware and the middleware component used to connect those different machines on each other.

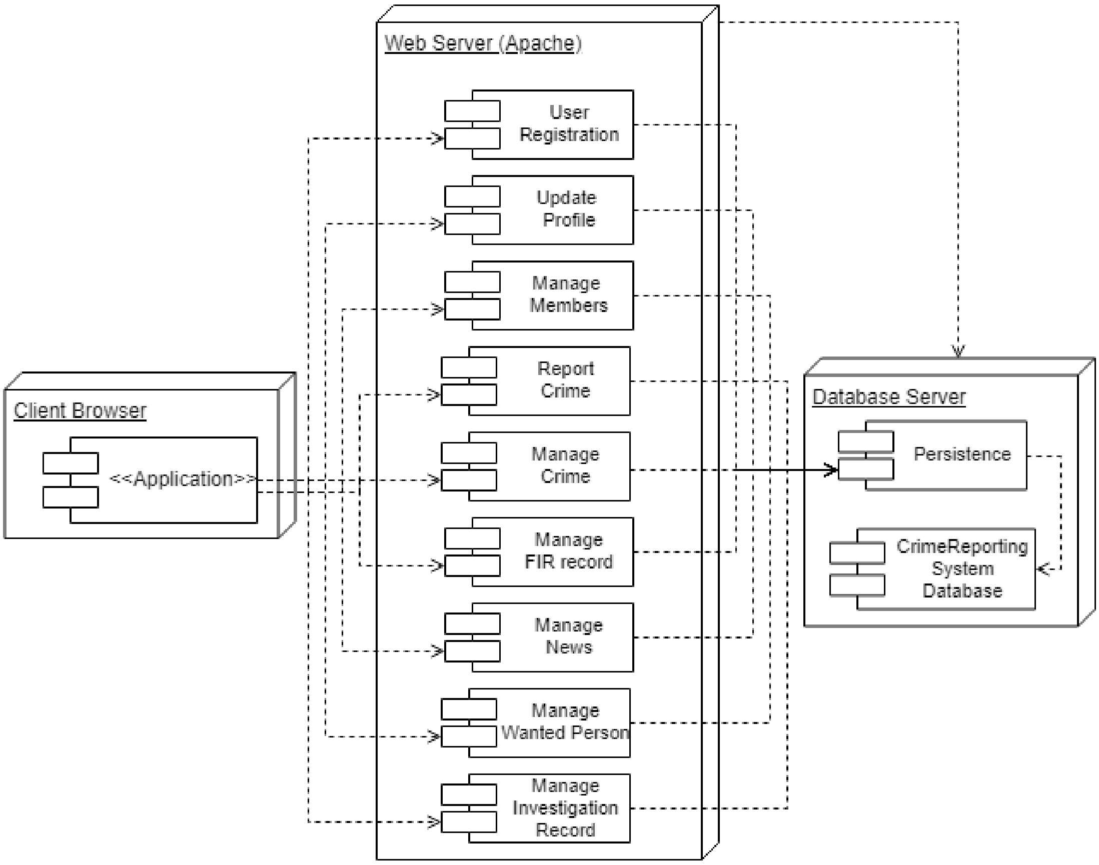


Figure 39: Deployment Modeling Diagram of OCRS (Source: [author])

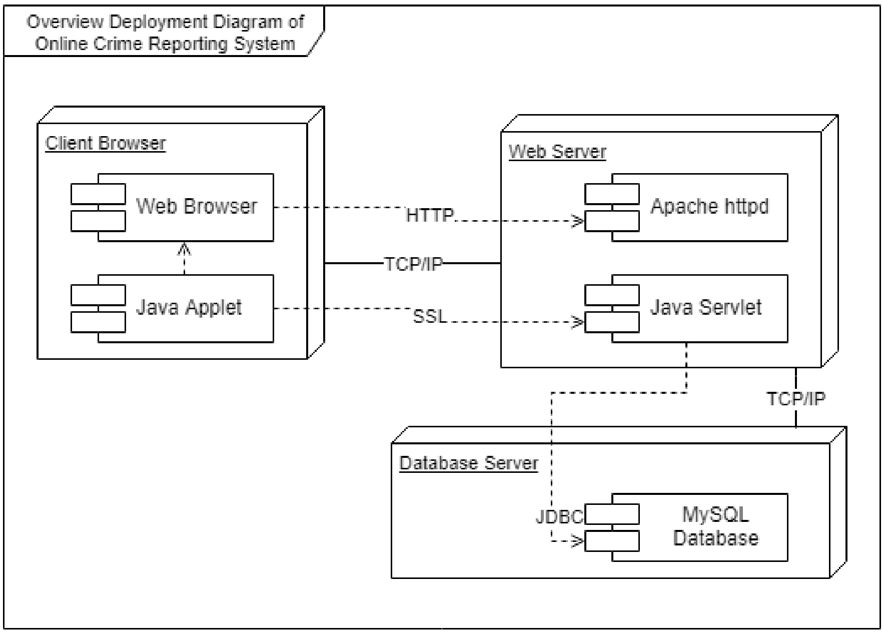


Figure 40: Overview Deployment Diagram of OCRS (Source: [author])

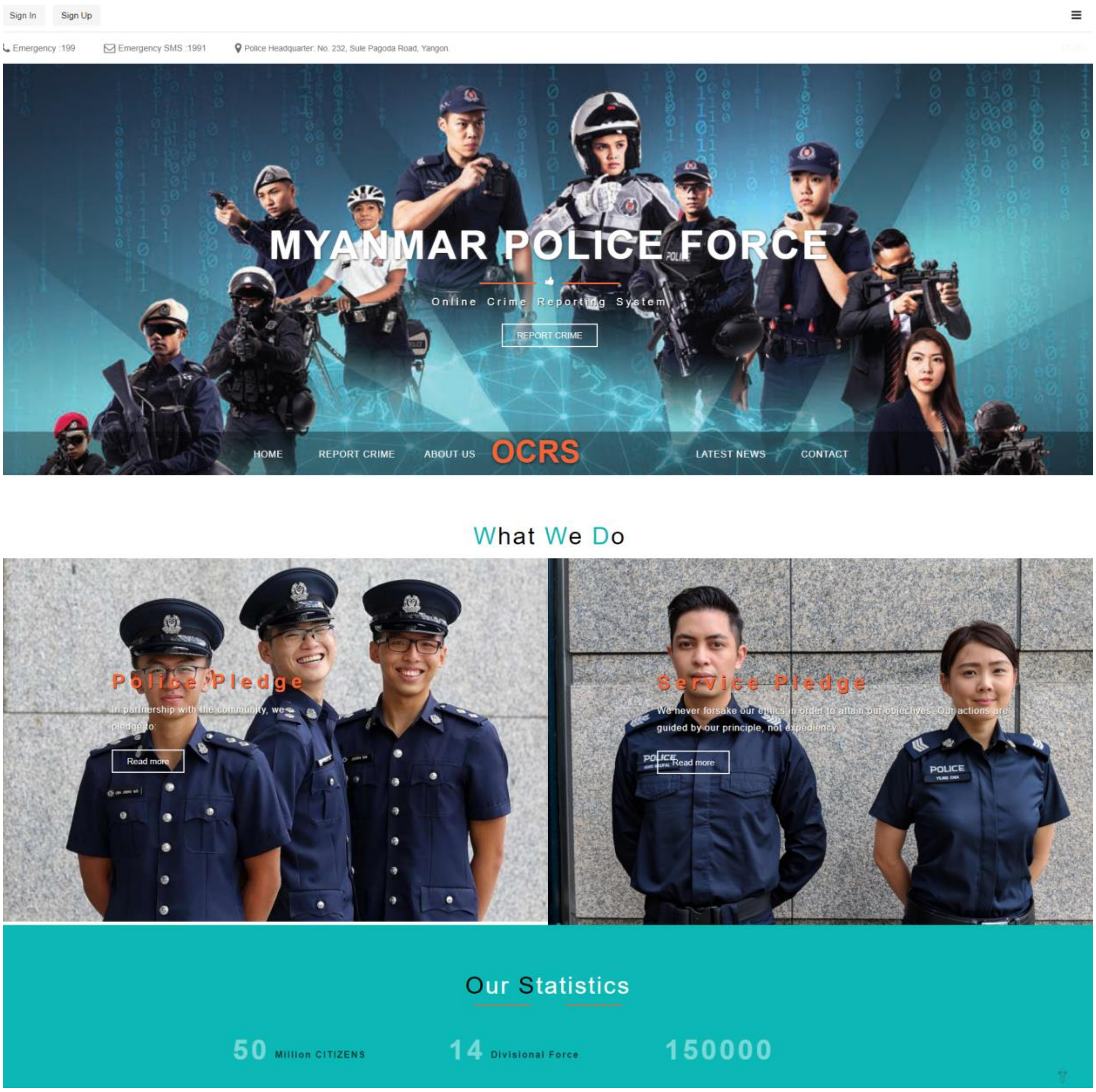
5. Implementation of Online Crime Reporting System

A prototype web-based application for Online Crime Reporting System has been implemented by using Java technology along with HTML, CSS, Javascript and JQuery technologies. MySQL database server has used for relational database management and data storage.

There will be two parts of the implementation for this prototype web-based application of OCRS. One part is intended for the citizens who are visitors or registered member of the system, to be able to report crime by using the OCRS and see the information of Myanmar Police Force along with news and wanted person information on the website. The second part of the implementation will be for the police admin who are police administrative officer and police officer where they can manage crime records, create new crime records, manage users, manage wanted person information and approve FIR records. The Online Crime Reporting System has many different modules, however, in this document some of the key important part of the system such as reporting crime by user, approving FIR records, managing crime records and managing users by police administrative officer have been designed and implemented in order to describe the use of UML in the process of development.

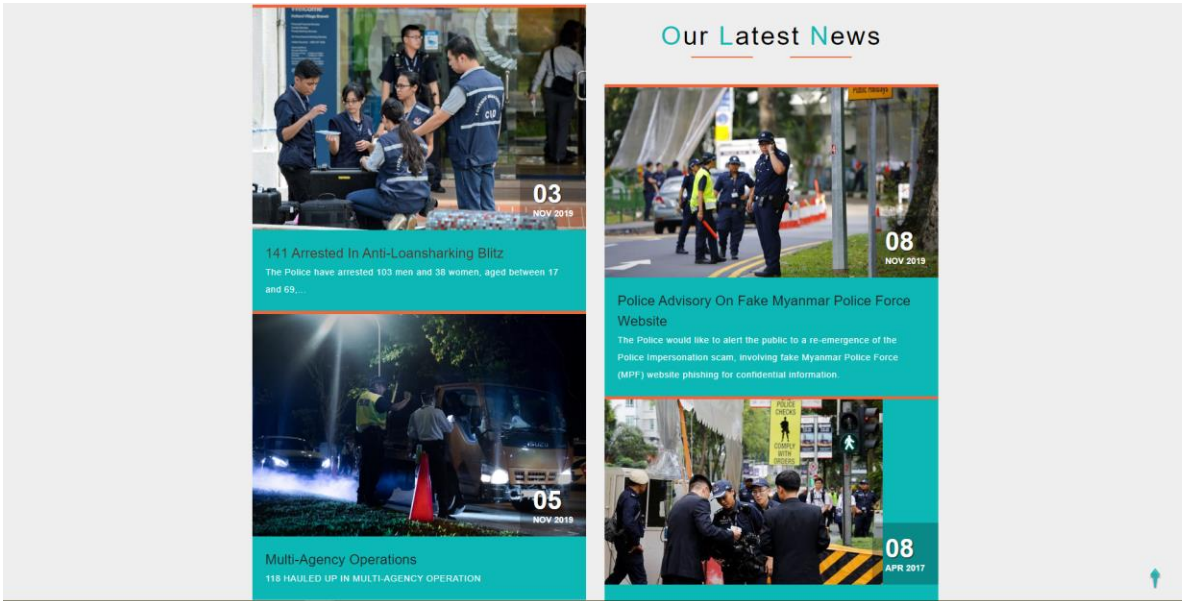
Screenshots of Online Crime Reporting System

As described previously, there are two parts in the implementation of OCRS and the first part is intended for the citizens who are visitors or registered members of the system. The visitors of the system will see landing page of the system and the following figure 41 shows the home page of the Online Crime Reporting System.



In partnership with the community, we pledge to:

We never forsake our ethics in order to attain our objectives. Our actions are guided by our principle, not expediency


Our Latest News

03 NOV 2019

141 Arrested In Anti-Loansharking Blitz

The Police have arrested 103 men and 38 women, aged between 17 and 69,...

08 NOV 2019

Police Advisory On Fake Myanmar Police Force Website

The Police would like to alert the public to a re-emergence of the Police Impersonation scam, involving fake Myanmar Police Force (MPF) website phishing for confidential information.

05 NOV 2019

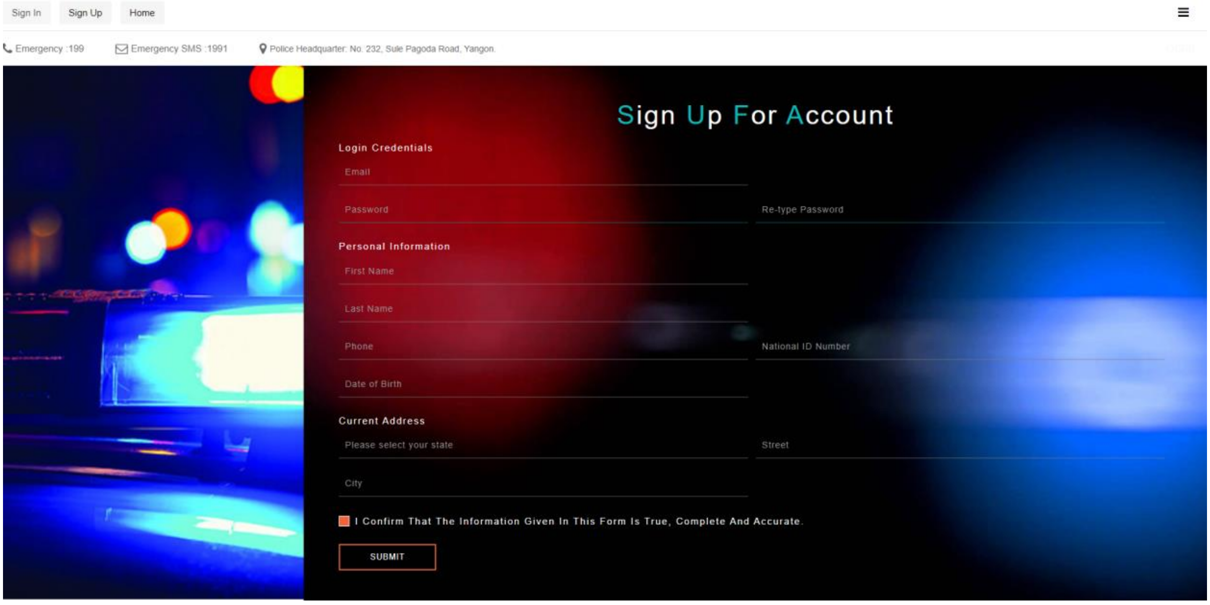
Multi-Agency Operations

118 HAULED UP IN MULTI-AGENCY OPERATION

08 APR 2017

Figure 41: Landing Home Page of OCRS (Source: [author])

In order to access to the system as a member, one must register to the system by using the following figure 42. After clicking the Sign Up button on home page, system will redirected to the page as we can see in following figure 42. User must entered all the necessary mandatory personal information such as national ID number, name, phone number and current address information, etc. User must also enter the login credentials details such as user name and password.



Emergency :199 Emergency SMS :1991 Police Headquarter: No. 232, Sule Pagoda Road, Yangon.

Sign Up For Account

Login Credentials

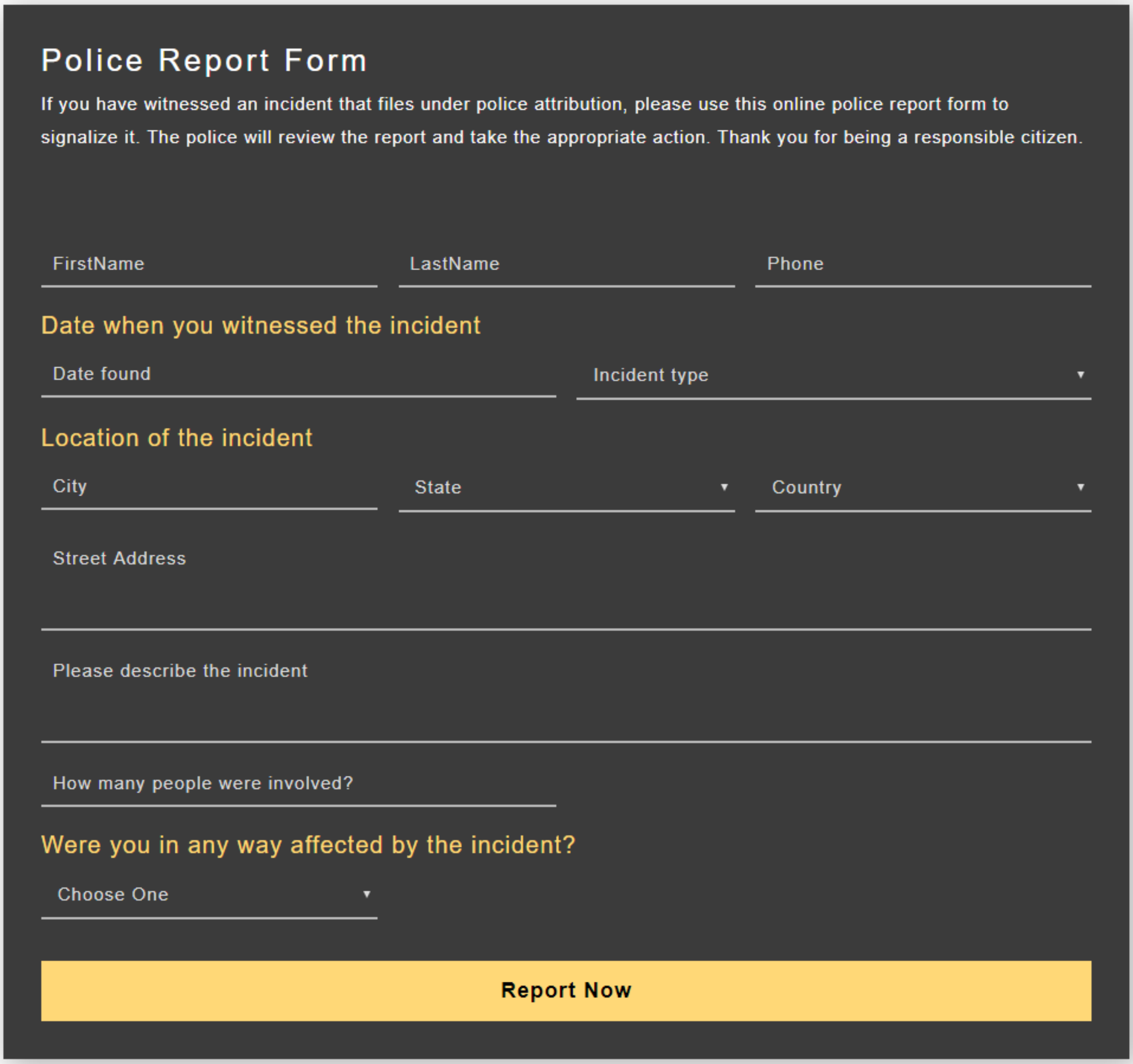
Personal Information

Current Address

☐ I Confirm That The Information Given In This Form Is True, Complete And Accurate.

Figure 42: User Registration Page of OCRS (Source: [author])

As described in the previous section, users of the Online Crime Reporting System will be able to report the crime incident by clicking the report crime button in the home page. After clicking the report crime button, terms and conditions information will be appeared on the pop up alert box describing some mandatory important information. Once user confirmed reading information and proceed to report crime, the police report form page will be redirected by the system as we can see from the following figure 43.



Police Report Form

If you have witnessed an incident that files under police attribution, please use this online police report form to signalize it. The police will review the report and take the appropriate action. Thank you for being a responsible citizen.

FirstName
LastName
Phone
Date when you witnessed the incident
Date found
Incident type
Location of the incident
City
State
Country
Street Address
Please describe the incident
How many people were involved?
Were you in any way affected by the incident?
Choose One
Report Now

Figure 43: Police Report Form Page of OCRS (Source: [author])

The second part of the implementation of OCRS will be for the police administrative officer and police officer as described previously in this thesis document.

To access the OCRS system as the police, user must login with their username and password. After the successful login process is done by entering the correct username and password, the system will display the main menu page of the user according to the role. The following figure 44 shows the main dashboard page of OCRS for the police administrative officer.

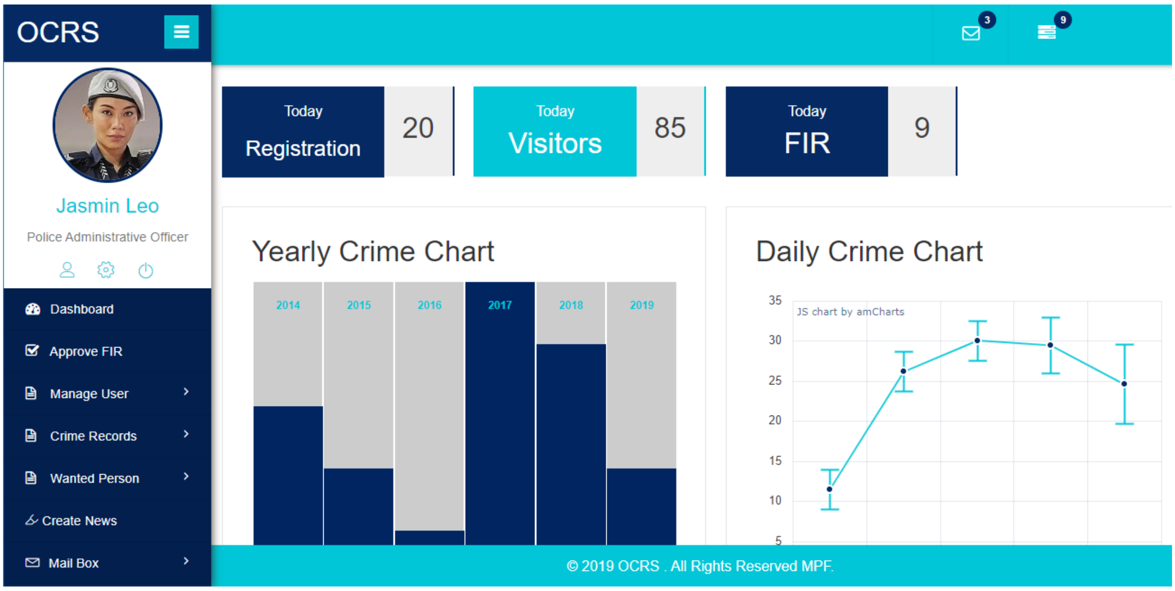
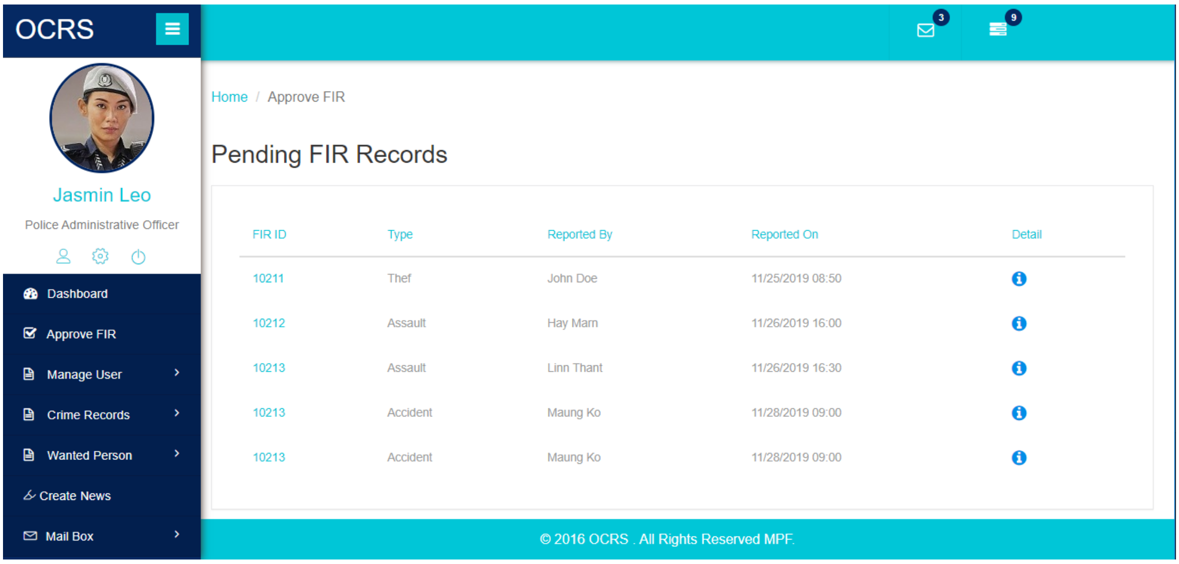


Figure 44: Main Dashboard Page of OCRS for admin (Source: [author])



Pending FIR Records

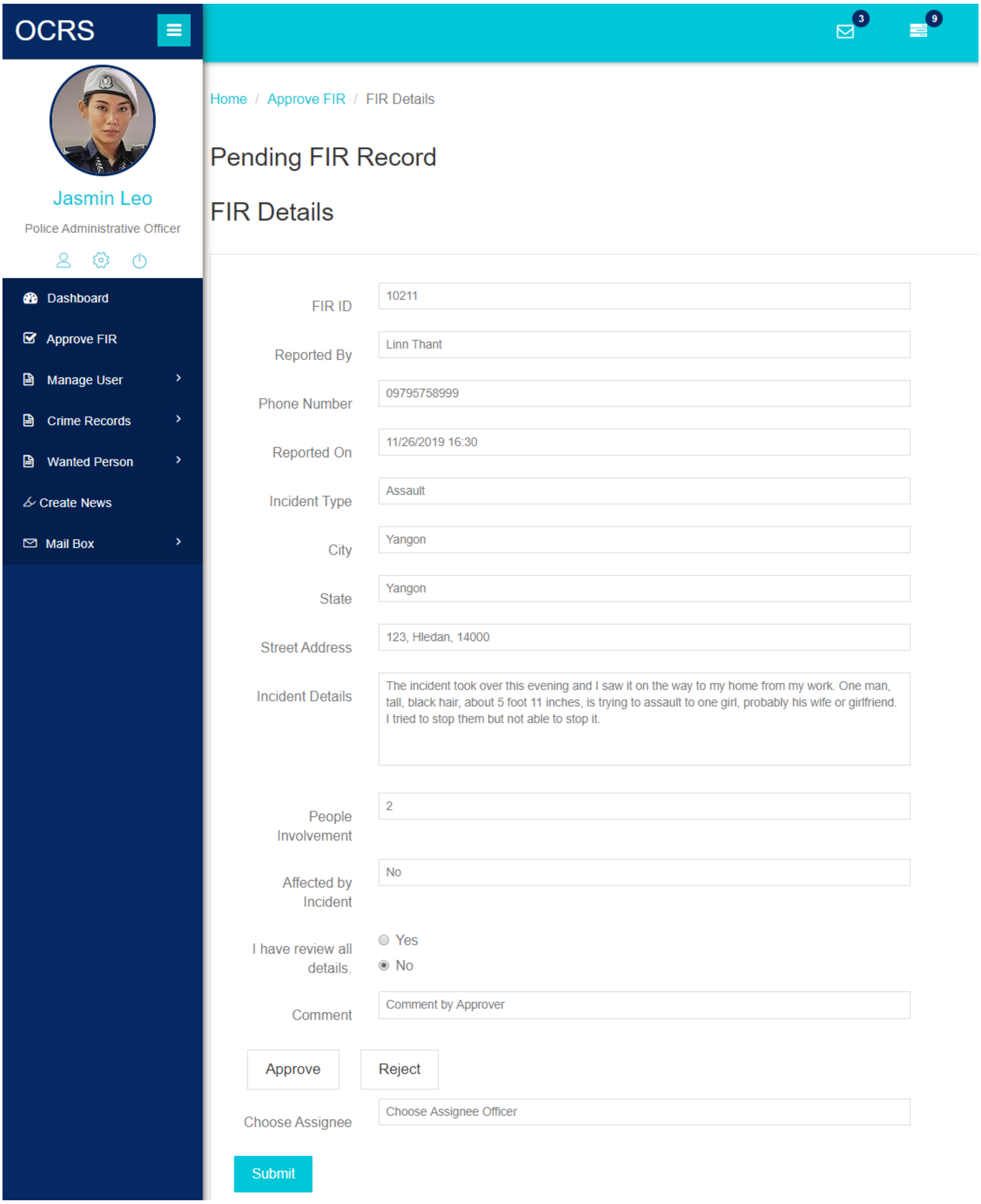
FIR ID	Type	Reported By	Reported On	Detail
10211	Thef	John Doe	11/25/2019 08:50	ℹ
10212	Assault	Hay Marn	11/26/2019 16:00	ℹ
10213	Assault	Linn Thant	11/26/2019 16:30	ℹ
10213	Accident	Maung Ko	11/28/2019 09:00	ℹ
10213	Accident	Maung Ko	11/28/2019 09:00	ℹ

© 2016 OCRS . All Rights Reserved MPF.

Figure 45: Approve FIR Record Page of OCRS for admin (Source: [author])

The above figure 45 shows the approve FIR record page and it will be redirected by the system once user click approve FIR link on the menu. As we can see in following figure 45, the list of pending FIR records will be displayed and police administrative officer can review the details of them just by clicking the information icon under Detail column tab.

Once police administrative officer click the information icon for specific pending FIR record, system will redirect to details FIR record page as we can see in following figure 46, where all the details of the reported FIR (First Information Report) records will be displayed. Police administrative officer can then review, edit as necessary and approve or reject the record accordingly. After reviewing of the record, police administrative officer can click approve or reject button accordingly. The new field will appear once police administrative officer click the button according to which button has been clicked. If the approve button has been clicked, then assignee police officer drop down box will appear with the police officer name information. If reject button has been clicked, then reject reason text box will appear and police administrative officer has to fill out the field.



Pending FIR Record

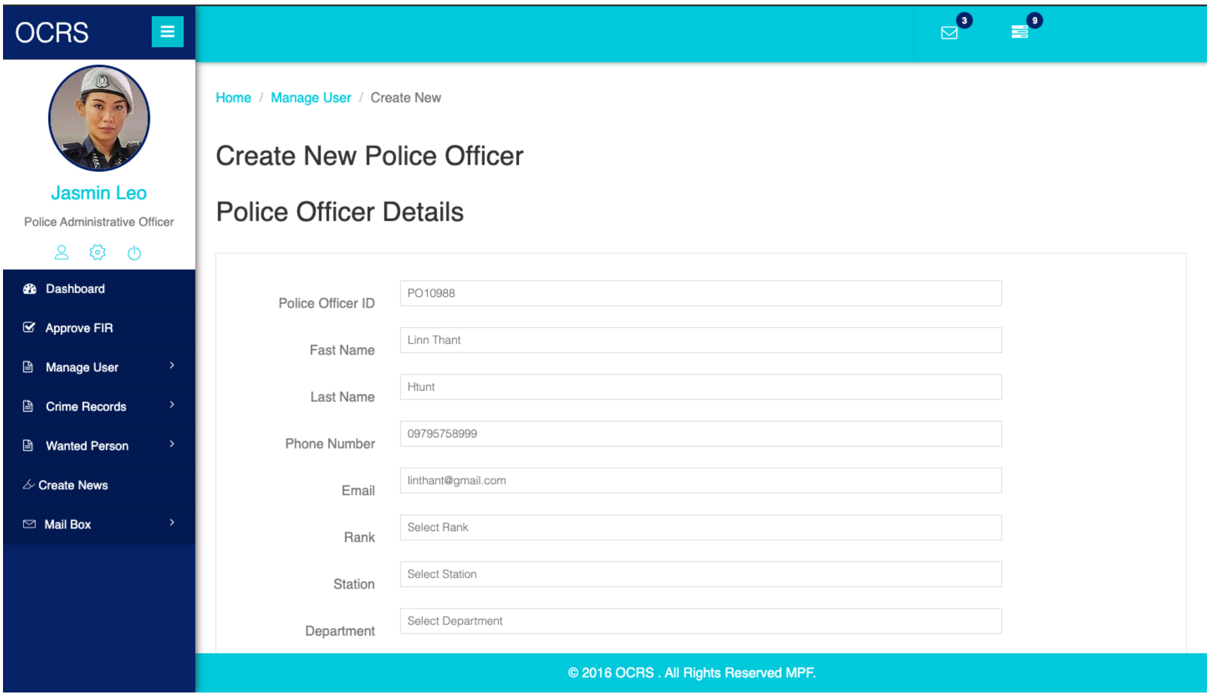
FIR Details

FIR ID	10211
Reported By	Linn Thant
Phone Number	09795758999
Reported On	11/26/2019 16:30
Incident Type	Assault
City	Yangon
State	Yangon
Street Address	123, Hledan, 14000
Incident Details	The incident took over this evening and I saw it on the way to my home from my work. One man, tall, black hair, about 5 foot 11 inches, is trying to assault to one girl, probably his wife or girlfriend. I tried to stop them but not able to stop it.
People Involvement	2
Affected by Incident	No
I have review all details.	○ Yes ● No
Comment	Comment by Approver

Choose Assignee: Choose Assignee Officer

Figure 46: Pending FIR Record Details Page of OCRS for admin (Source: [author])

The following figure 47 shows the creating of new user (police officer) by the police administrative officer. The system will redirect to that page when user click manage user and then create new user option in navigation bar.



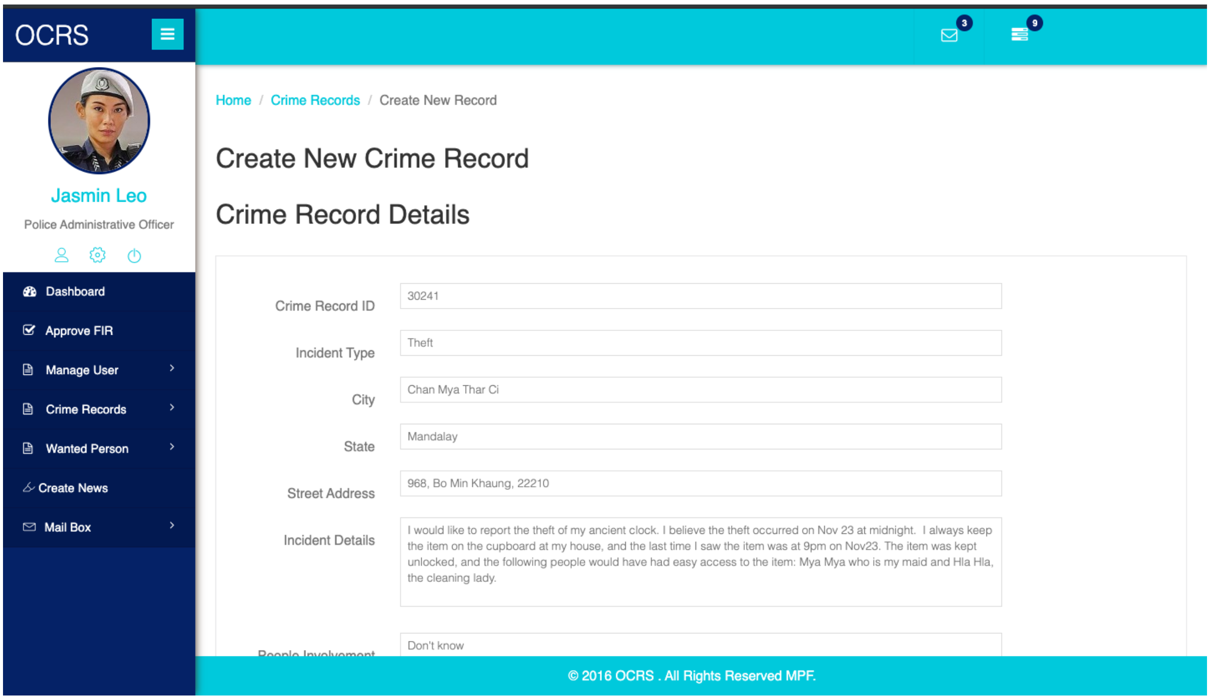
Home / Manage User / Create New

Create New Police Officer

Police Officer Details

Police Officer ID	PO10988
Fast Name	Linn Thant
Last Name	Htunt
Phone Number	09795758999
Email	linthant@gmail.com
Rank	Select Rank
Station	Select Station
Department	Select Department

Figure 47: Create New Police Officer Page of OCRS for admin (Source: [author])



Home / Crime Records / Create New Record

Create New Crime Record

Crime Record Details

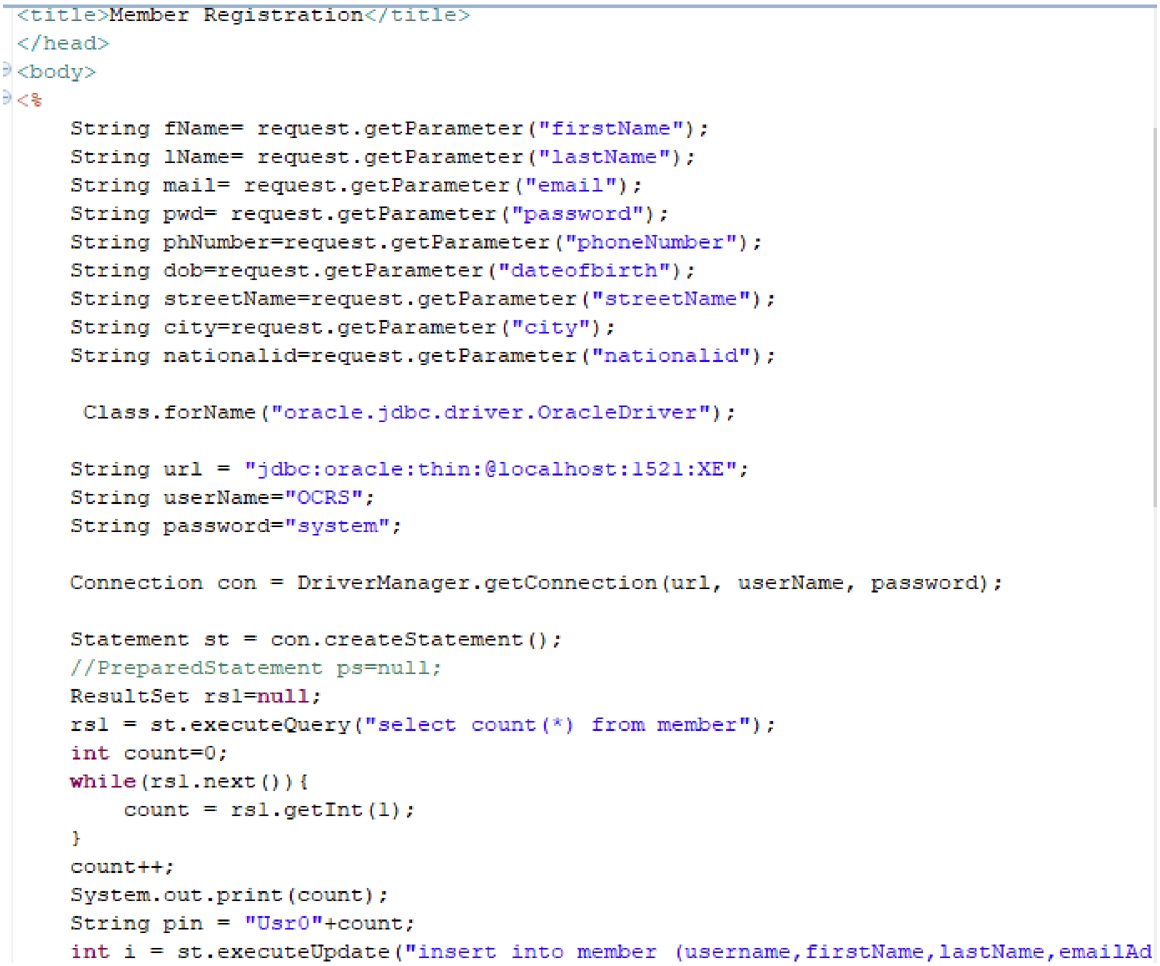
Crime Record ID	30241
Incident Type	Theft
City	Chan Mya Thar Ci
State	Mandalay
Street Address	968, Bo Min Khaung, 22210
Incident Details	I would like to report the theft of my ancient clock. I believe the theft occurred on Nov 23 at midnight. I always keep the item on the cupboard at my house, and the last time I saw the item was at 9pm on Nov23. The item was kept unlocked, and the following people would have had easy access to the item: Mya Mya who is my maid and Hla Hla, the cleaning lady.
People Involvement	Don't know

Figure 48: Create New Crime Record Page of OCRS for admin (Source: [author])

Once police administrative officer click crime records and then create new record, the system will redirect to create new crime record page where officer can create new record and assign to police officer accordingly as we can see in above figure 48.

We can also see the coding snippets for some implementation components of Online Crime Reporting System in the following figures. As described before in this thesis document, OCRS has been implemented by using HTML, CSS, Javascript and JQuery as front-end technology and those pages have been integrated as .jsp pages since Java technology was used as a back-end development. However, only a few segment of Java coding will be described in this document as putting all of the coding together would be occupied a lot of pages.

Brief piece of Java code for user registration of OCRS

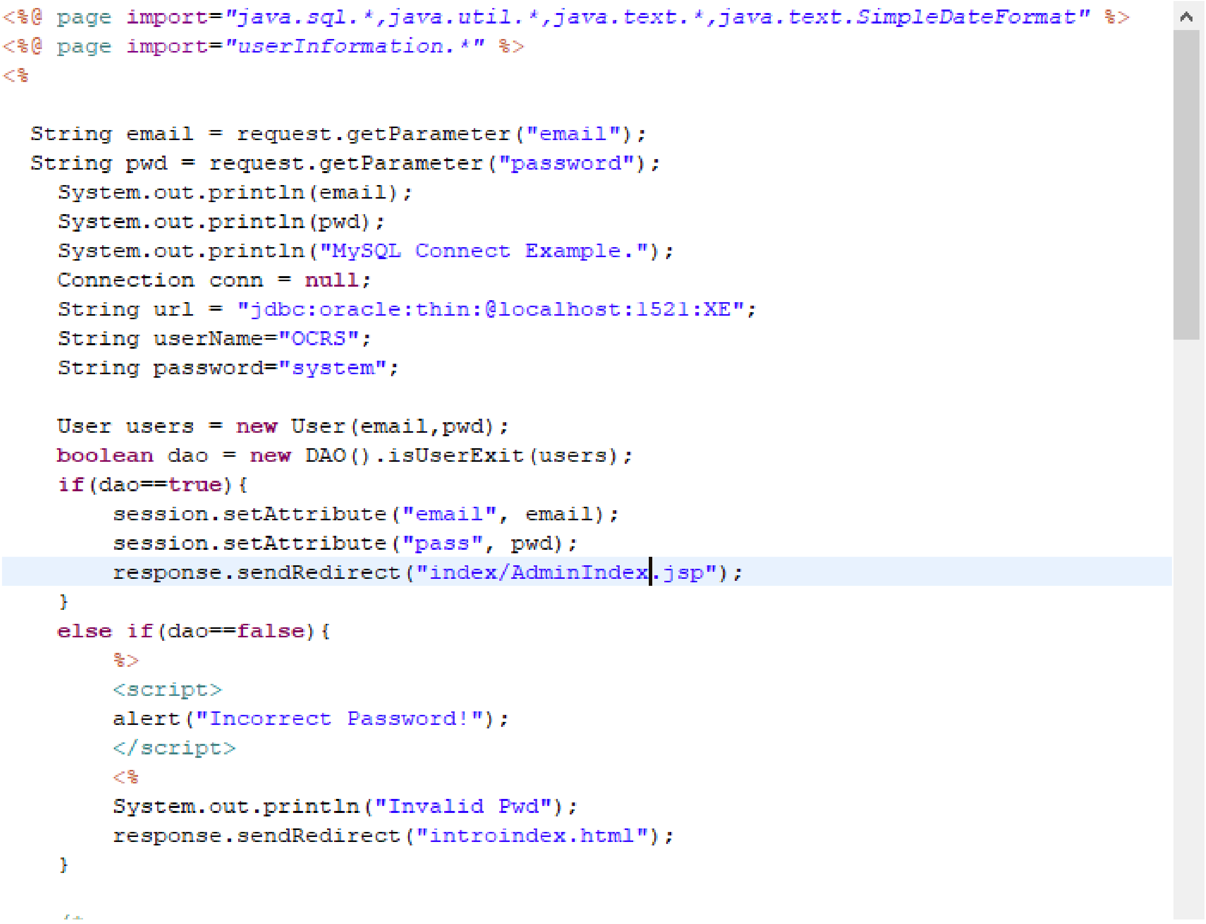


```
<title>Member Registration</title>
</head>
<body>
<%
    String fName= request.getParameter("firstName");
    String lName= request.getParameter("lastName");
    String mail= request.getParameter("email");
    String pwd= request.getParameter("password");
    String phNumber=request.getParameter("phoneNumber");
    String dob=request.getParameter("dateofbirth");
    String streetName=request.getParameter("streetName");
    String city=request.getParameter("city");
    String nationalid=request.getParameter("nationalid");

     Class.forName("oracle.jdbc.driver.OracleDriver");

    String url = "jdbc:oracle:thin:@localhost:1521:XE";
    String userName="OCRS";
    String password="system";

    Connection con = DriverManager.getConnection(url, userName, password);

    Statement st = con.createStatement();
    //PreparedStatement ps=null;
    ResultSet rs1=null;
    rs1 = st.executeQuery("select count(*) from member");
    int count=0;
    while(rs1.next()){
        count = rs1.getInt(1);
    }
    count++;
    System.out.print(count);
    String pin = "Usr0"+count;
    int i = st.executeUpdate("insert into member (username,firstName,lastName,emailAd
```

Figure 49: Java code for user registration of OCRS (Source: [author])

The above figure 49 shows the small piece of coding for user registration process of OCRS along with the database driver connection to the table where registration details will be stored.

Java code for login validation of OCRS



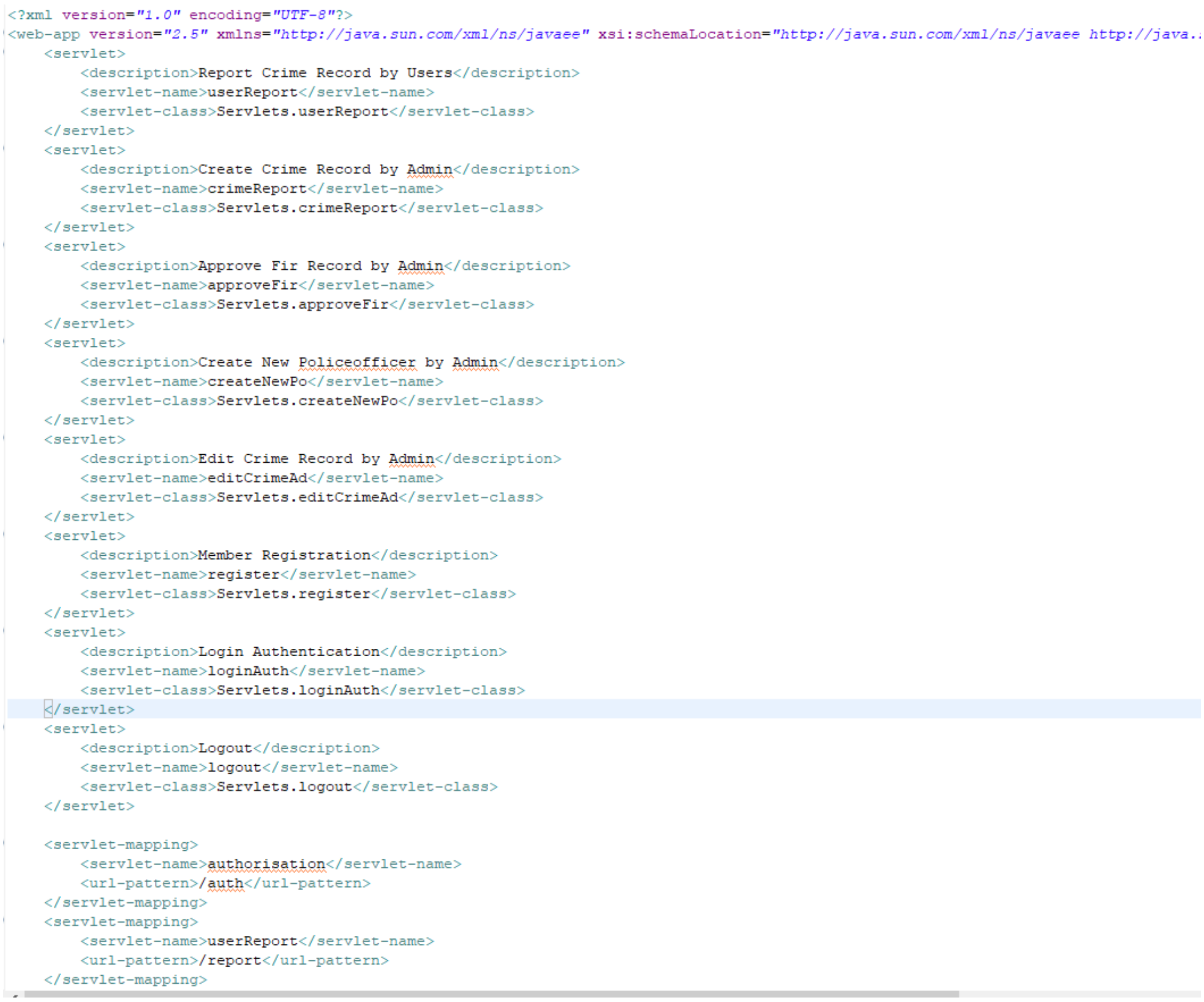
```
<%@ page import="java.sql.*,java.util.*,java.text.*,java.text.SimpleDateFormat" %>
<%@ page import="userInformation.*" %>
<%

  String email = request.getParameter("email");
  String pwd = request.getParameter("password");
    System.out.println(email);
    System.out.println(pwd);
    System.out.println("MySQL Connect Example.");
    Connection conn = null;
    String url = "jdbc:oracle:thin:@localhost:1521:XE";
    String userName="OCRS";
    String password="system";

    User users = new User(email,pwd);
    boolean dao = new DAO().isUserExit(users);
    if(dao==true){
        session.setAttribute("email", email);
        session.setAttribute("pass", pwd);
        response.sendRedirect("index/AdminIndex.jsp");
    }
    else if(dao==false){
        %>
        <script>
        alert("Incorrect Password!");
        </script>
        <%
        System.out.println("Invalid Pwd");
        response.sendRedirect("introindex.html");
    }
```

Figure 50: Java code for login validation of OCRS (Source: [author])

The above figure 50 shows the login validation of OCRS whereas all login credentials will be validated against the database by calling the created method of the DAO (data access object) which allows to isolate the application layer from the persistence layer in relational database.

Java code for web.xml file of OCRS

Since OCRS is the java web application, it uses web.xml file as a deployment descriptor file to determine how URLs map to servlets, and also describes the classes, resources and configuration of the application on how the web server uses them to map the web requests. We can see the web.xml file of OCRS in the following figure 51.

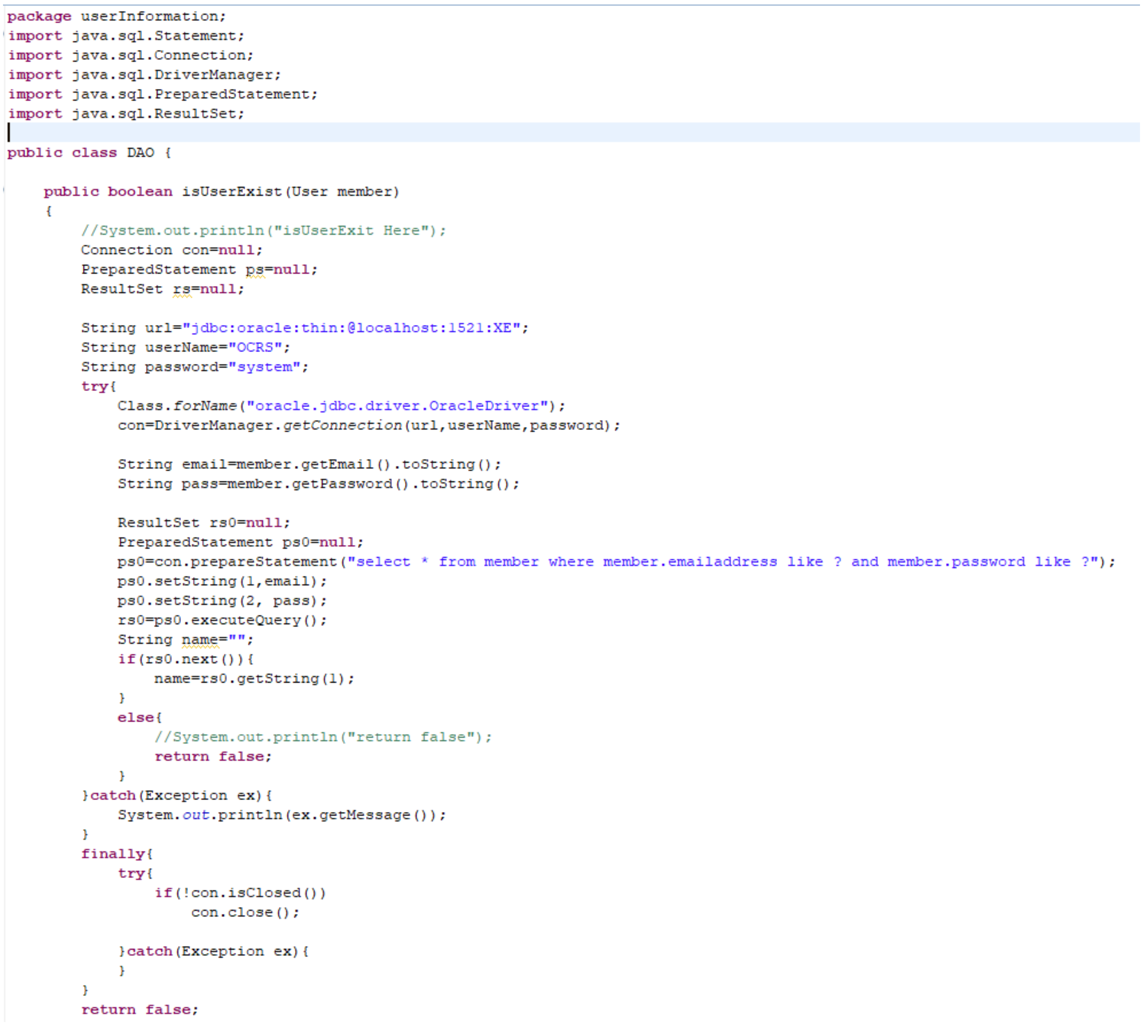


```xml
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="2.5" xmlns="http://java.sun.com/xml/ns/javaee" xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.
    <servlet>
        <description>Report Crime Record by Users</description>
        <servlet-name>userReport</servlet-name>
        <servlet-class>Servlets.userReport</servlet-class>
    </servlet>
    <servlet>
        <description>Create Crime Record by Admin</description>
        <servlet-name>crimeReport</servlet-name>
        <servlet-class>Servlets.crimeReport</servlet-class>
    </servlet>
    <servlet>
        <description>Approve Fir Record by Admin</description>
        <servlet-name>approveFir</servlet-name>
        <servlet-class>Servlets.approveFir</servlet-class>
    </servlet>
    <servlet>
        <description>Create New Policeofficer by Admin</description>
        <servlet-name>createNewPo</servlet-name>
        <servlet-class>Servlets.createNewPo</servlet-class>
    </servlet>
    <servlet>
        <description>Edit Crime Record by Admin</description>
        <servlet-name>editCrimeAd</servlet-name>
        <servlet-class>Servlets.editCrimeAd</servlet-class>
    </servlet>
    <servlet>
        <description>Member Registration</description>
        <servlet-name>register</servlet-name>
        <servlet-class>Servlets.register</servlet-class>
    </servlet>
    <servlet>
        <description>Login Authentication</description>
        <servlet-name>loginAuth</servlet-name>
        <servlet-class>Servlets.loginAuth</servlet-class>
    </servlet>
    <servlet>
        <description>Logout</description>
        <servlet-name>logout</servlet-name>
        <servlet-class>Servlets.logout</servlet-class>
    </servlet>

    <servlet-mapping>
        <servlet-name>authorisation</servlet-name>
        <url-pattern>/auth</url-pattern>
    </servlet-mapping>
    <servlet-mapping>
        <servlet-name>userReport</servlet-name>
        <url-pattern>/report</url-pattern>
    </servlet-mapping>
```

Figure 51: Java code for web.xml file of OCRS (Source: [author])

Java Code for DAO class of OCRS

The following figure 52 shows the java code snippets of DAO class for validating user credential details from login validation function along with database connection.



```
package userInformation;
import java.sql.Statement;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;

public class DAO {

    public boolean isUserExist(User member)
    {
        //System.out.println("isUserExit Here");
        Connection con=null;
        PreparedStatement ps=null;
        ResultSet rs=null;

        String url="jdbc:oracle:thin:@localhost:1521:XE";
        String userName="OCRS";
        String password="system";
        try{
            Class.forName("oracle.jdbc.driver.OracleDriver");
            con=DriverManager.getConnection(url,userName,password);

            String email=member.getEmail().toString();
            String pass=member.getPassword().toString();

            ResultSet rs0=null;
            PreparedStatement ps0=null;
            ps0=con.prepareStatement("select * from member where member.emailaddress like ? and member.password like ?");
            ps0.setString(1,email);
            ps0.setString(2, pass);
            rs0=ps0.executeQuery();
            String name="";
            if(rs0.next()){
                name=rs0.getString(1);
            }
            else{
                //System.out.println("return false");
                return false;
            }
        }catch(Exception ex){
            System.out.println(ex.getMessage());
        }
        finally{
            try{
                if(!con.isClosed())
                    con.close();

            }catch(Exception ex){
            }
        }
        return false;
```

Figure 52: DAO class for login validation of OCRS (Source: [author])

In this section, the implementation of the Online Crime Reporting System has been discussed. All details coding of every classes are omitted to conserve space of this document. A prototype web-based application has been implemented for Online Crime Reporting System by using Java technology along with HTML, CSS, Javascript and JQuery technologies. MySQL database server has used for relational database management and data storage. As described, we can see that the use of UML makes easier implementation process along with the coding process. We will discuss the deployment and hosting of this java-based web application in the following chapter.

6. Discussion

In this study, the focus is on the implementation process of Online Crime Reporting System, which is approached by means of systems analysis, with its strong organizational emphasis. The intent is to create theoretical and conceptual research models, as well as practical solutions. The intent is to implement prototype web-based application which includes a crime reporting module for the user as well as assist the management of crime records for the police administrative office and police officers.

This study document describes the information system design in UML and the major components of this proposed system has been described briefly in UML illustrations. This document has covered the creation and usage of UML models during the whole software development process from drawing of UML diagrams during the initial planning and analysis stages. UML gives a standard way to design an information system model and covering. With an understanding of modeling process, the use and application of UML can demonstrate the advantages by highlighting the particular features of the UML language in web application development.

A web-based application has been implemented for Online Crime Reporting System in different modules which will help the user to report the crime incident information easily by clicking the report crime button. The implemented system also allow the user to register into the system as a member, so that they will be able to check the status of their reported crime and manage their reported crime records easily. A police administrative officer is a person who can manage users of the system and can also add the new users into the system with their respective roles such as new administrative officer or new police officer. The system allows the police administrative officer to review, edit if necessary and approve the newly reported FIR (First Information Report) records which have been reported by citizens. A police administrative officer can add and manage the crime records, manage the wanted person information whereas police officer can manage the respective crime records which has been assigned to them.

Each user of the system has a main display with the menus based on their respective roles and duties such as registered member of the system can be able to report crimes and check the status of reported crime record while visitor of the system can only see report crimes and see the basic public information of the website.

User who serves as the police administrative officer role will be able to manage crime records, approve FIR records, manage users and manage wanted person information while user with police officer role can only be able to manage the crime records which has been assigned to them.

Lastly, in this thesis document, the open source Java technology along with HTML, CSS, Javascript and JQuery technologies have been used for implementation of the proposed Online Crime Reporting System. The docker java web application archive (.war file) can be extracted and it will aid the easy deployment of this web application to any existing server. Since OCRS is the java-based web application, it would require to have a separate java hosting solution such as PaaS (Platform-as-a-Service) solution which delivers the infrastructure to deploy and manage the web applications. Since it has provides the fundamental benefits of cloud computing, organization does not need to spend excess amount of resources. Another advantage of PaaS solution is that is support the multi-cloud portability which can provides the easy workload portability across databases, containers and open source technology such as Java that has been used in implementation of OCRS system in this document. It can even be hosted on VPS (Virtual Private Server) where we will get the better hosting resources along with the affordable price.

7. Conclusion

Lastly, based on the results of implementation and discussion in the OCRS system design, it can be concluded that the problems which were described in the problem domain section have been addressed in this thesis document. User will be easily report their crime incidents and can easily check the status just by registering in the system. Police administrative officer and police officer can also manage the crime records easily which can help them to avoid large volume of pare work consumption and manual calculations.

This thesis document addressed and described the important components of UML modeling along description of UML models with UML semantics. This thesis document also described the use of UML in different phases in the development of the system from the requirements specification to the test of a finished system. The major components of the proposed system has been illustrated by using UML in this document.

The study in this thesis document delineates the need of basic functionalities feature of basic crime reporting system and crime records management system.

However, further features can also be added to the implementation of this system such as dynamically updating current news by police administrative officer, so that all the updated crime news will be accessible by the visitors (citizens). The implementation reporting module should include in the further development feature, so that police officer can easily generate the report chart and export them into editable format such as excel or csv file. As all the technologies in the IT world are moving faster than ever before, the further research is also required on the path models and the mutual interdependencies between police force and citizens. By means of these models, one can manage and direct development efforts for the implementation of information systems design with UML, as well as proceeding to accomplish the aimed goals. Further, these models described in this thesis document aid the growth of the government organization like police force, as well as the development of competencies in the organization and easy accessibility by the public.

8. Bibliography

[1] Rumbaugh, J., Booch, G. and Jacobson, I. *The unified modeling language reference manual*. Boston: Addison-Wesley. 2010.

[2] OMG Unified Modeling Language specification, Needham, Mass. Object Management Group, Inc. 2001.

[3] Addison-Wesley. Unified Modeling Language User Guide, the (2 ed.), 2005. ISBN: 0321267974

[4] Bremen, U. and Technologies, T.UML 2001 — The Unified Modeling Language. Modeling Languages, Concepts, and Tools. 2019.

[5] C. Larman, Applying UML and patterns: an introduction to object-oriented analysis and design, Prentice Hall PTR, 1998.

[6] P. Stevens and R. Pooley, Using Uml: Software Engineering with Objects and Components, 1st ed. Addison-Wesley Longman Publishing Co., Boston, MA, USA, 1999.

[7] M. Szlenk, "UML Static Models in Formal Approach," Balancing Agility and Formalism in Software Engineering, pp. 129–142, 2008.

[8] Q. Chunyan, UML-based software process modeling, In International Conference on Computer, Mechatronics, Control and Electronic Engineering, pp. 247– 250, 2010.

[9] C. Larman, Applying UML and patterns: an introduction to object-oriented analysis and design, Prentice Hall PTR, 1998.

[10] UML basics: Introduction to the Unified Modeling Language. [Online]. Available: http://www.ibm.com/developerwork/library/769.html. [Accessed: 08-July-2019].

[11] Eriksson, H.-E. and M. Penker, UML toolkit. 1998: John Wiley & Sons, Inc. 397.

[12] Find out how strong your vocabulary is and learn new words at Vocabulary.com, [Online]. Available: http://www.vocabulary.com/. [Accessed: 09-July-2019].

[13] Cook, S. Looking Back at UML Software & System Modeling, pp. 471-480, 2012.

[14] Terry Quatrani, Visual Modeling with Rational Rose and UML, 3rd Edition, 2002.

[15] Thompson, N &amp; Platt, R. The Evolution of UML, pp. 348-353, 2015.

[16] OMG. Unified Modeling Language Specification, Object Management Group, Inc., Framingham, Mass., Available: https://www.omg.org, 2010.

[17] Buchmann, Alejandro & Geoff Coulson. A Model Driven Architecture for Adaptable Overlay Networks, 2018.

[18] Uml.org. What is UML | Unified Modeling Language. [Online] Available at: https://www.uml.org/what-is-uml.htm [Accessed 25-July-2019].

[19] W. Wayt Gibbs. Software's chronic crisis. Scientific American, pp. 72–81, September 1994.

[20] B. W. Boehm. A spiral model of software development and enhancement. IEEE Computer, pp. 61–72, 1998.

[21] Object Management Group – UML 2.0 Infrastructure Specification. Internet: http://www.omg.org /docs/ptc/03- 09- 15.pdf, Sept 15, 2003[Accessed: 25-July-2019].

[22] Object Management Group – OCL Specification. [Online]. Available: http://www.omg.org/docs/ptc/03- 10- 14.pdf, Oct 14, 2003[Accessed: 25-July-2019].

[23] UML Semantics, Object Management Group, 1997.

[24] B. V. Selic, “On the Semantic Foundations of Standard UML 2.0,” in Formal Methods for the Design of Real-Time Systems, Springer Berlin, 2004, pp. 181-199, 2004.

[25] L. C. Briand, Y. Labiche and T. Yue, “Automated traceability analysis for UML model refinement,” Journal Information and Software Technology, vol. 51, issue 2, pp. 512-517, 2009.

[26] C. Pons, R. Giandini, G. Baum, J. L. Garbi, P. Mercado, “Specification and Checking Dependency Relations between UML Models,” in UML and the Unified Process. Hershey: IGI Publishing, pp 237-253, 2003.

[27] S. Kent, A. Evans, and B. Rumpe (Eds.). UML semantics FAQ. [Online]. Available: http://www.univ-pau.fr/OOPSLA99/samplewr99.pdf, [Accessed: 25-July-2019].

[28] R. France, J.M. Bruel, M. Larrondo-Petrie, and M. Shroff. Exploring the semantics of UML type structures with Z. In H. Bowman and J. Derrick, editors, Proc. 2nd IFIP Conf. Formal Methods for Open Object-Based Distributed Systems, 1997.

[29] M. van den Berg, M. Verhoef, and M. Wigmans. Formal specication of an auctioning system using VDM++ and UML. In J. Fitzgerald and P. G. Larsen, editors, VDM in Practice, 1999.

[30] J. Smith. UML Formalization and Transformation. PhD thesis, Northeastern University College of Engineering, USA, December 1998.

[31] Jim A., Ila N., UML 2 AND THE UNIFIED PROCESS Practical Object-Oriented Analysis and Design Second Edition Pearson, 2006.

[32] Dobing, B., Parsons, J. How UML is Used, Communications of the ACM, 2006.

[33] Miro S., Practical Statecharts in C/C++ Quantum Programming for Embedded Systems CMP Books, 2002.

[34] Holt, N., A Systematic Review of Case Studies in Software Engineering, University of Oslo, 2006.

[35] Goutam Kumar Saha. Understanding software testing concepts, 2008.

[36] Edward Kit, “Software testing in the real world”, 1995.

[37] Software Testing Fundamentals: Software Testing Levels 2012. [Online]. Available: http://softwaretestingfundamental.com/softwaretesting-level, [Accessed: 26-July-2019].

[38] Fakhroutdinov. UML 2.5 Diagrams Overview. 2019. [Online] Uml-diagrams.org. Available: https://www.uml-diagram.org/uml-diagram.html [Accessed 26-July-2019].

[39] M. Fowler, K. Scott, UML Distilled—Applying the Standard Object Modeling Language, Addison-Wesley, Reading, MA, 1997.

[40] V. (2019). Class Diagram Relationships in UML Explained with Examples. [Online] Creately. Available: https://creately.com/blog/diagrams/class-diagram-relationships/ [Accessed 26-July-2019].

[41] Uml-diagrams.org. (2019). UML State Machine Diagrams - Overview of Graphical Notation. [Online] Available: https://www.uml-diagrams.org/state-machine-diagrams.html [Accessed 26-July-2019].

[42] Vrana, I., 2016. Projecting of Information Systems with UML, Prague, Czechia: Faculty of Economics & Management, CULS Prague.

[43] P. Kruchten. Rational Unified Process. Addison-Wesley, 1998.