

**Android app for Disaster Management for disaster situations in
Hyderabad city**

A Project Report Submitted to

Jawaharlal Nehru Technological University Hyderabad

In partial fulfillment of the requirements

For the award of the degree of

**BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING**

By

**V.Amruth
K.Yeshwanth
A.Nikhil**

**16E11A05J0
16E11A05H8
16E11A05G7**

Under the guidance of

DR.R.Madana Mohan
Assistant Professor



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
BHARAT INSTITUTE OF ENGINEERING AND TECHNOLOGY**

(Affiliated to JNTUH Hyderabad, Approved by AICTE)

Ibrahimpattanam - 501 510, Hyderabad

2016 - 2020



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
BHARAT INSTITUTE OF ENGINEERING AND TECHNOLOGY
(Affiliated to JNTUH Hyderabad, Approved by AICTE)
Ibrahimpattanam - 501 510, Hyderabad

Certificate

This is to certify that the project work entitled "Project title" is the bonafide work done

	By	
V.Amruth		16E11A05J0
K.Yeshwanth		16E11A05H8
A.Nikhil		16E11A05G7

In the Department of Computer Science and Engineering, BHARAT INSTITUTE OF ENGINEERING AND TECHNOLOGY, Ibrahimpattanam is submitted to Jawaharlal Nehru Technological University, Hyderabad in partial fulfillment of the requirements for the award of B. Tech degree in Computer Science and Engineering during 2016-2020.

Guide:

Dr. R.Madana Mohan

Associate Professor
Dept of CSE,
Bharat Institute of Engineering and Technology,
Ibrahimpattanam – 501 510, Hyderabad.

Head of the Department:

Dr. R.Madana Mohan

Associate Professor
Dept of CSE
Bharat Institute of Engineering and Technology,
Ibrahimpattanam – 501 510, Hyderabad.

Viva-Voce held on.....

Internal Examiner

External Examiner

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of the task would be put incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crown all the efforts with success.

We avail this opportunity to express our deep sense of gratitude and hearty thanks to Sri CH.VenugopalReddy, Secretary of BIET, for providing congenial atmosphere and encouragement.

We would like to thank Prof. G. Kumaraswamy Rao, Director, Former Director & O.S. of DLRL Ministry of Defence, Dr. B. Prasada Rao, I.P.S. (Retd.), Director of Training & Placements, Industry Interface, Former Principal Secretary to Govt. of AP, DGP of ACB, Commissioner of Police, Hyderabad, Dr. M Lakshmi Narayana, Former Chairman IEEE and Dr. V.Rambabu, Principal for having provided all the facilities and support.

We would like to thank our Department coordinator .N.Aruna Jyothi Reddy ,Admin Incharge Dr.Delshi Howshalya Devi and Academic Incharge Ms.Y.Sailaja for their expert guidance and encouragement at various levels of our Project.

We are thankful to our Project Coordinator Mrs. V. M.Vijaya Bharathi, Assistant Professor, Computer Science and Engineering for her support and cooperation throughout the process of this project.

We are thankful to our Department R&D Incharge Dr.Anita Bai, Associate professor, Computer Science and Engineering for her support and cooperation throughout the process of this project.

We are thankful to our guide Ms.D.L.N.Prasunna, Assistant Professor and Computer Science and Engineering for her sustained inspiring Guidance and cooperation throughout the process of this project. Her wise counsel and suggestions were invaluable.

We express our deep sense of gratitude and thanks to all the Teaching and Non-Teaching Staff of our college who stood with us during the project and helped us to make it a successful venture.

We place highest regards to our Parent, our Friends and Well-wishers who helped a lot in making the report of this project

DECLARATION

We hereby declare that this project report is titled **Android App for Disaster Management for disaster situations in Hyderabad city** is a genuine mini project work carried out by us, in **B.Tech (Computer Science and Engineering)** degree course of **Jawaharlal Nehru Technology University Hyderabad, Hyderabad** and has not been submitted to any other course or university for the award of my degree by me.

Signature of the Student

- 1.
- 2.
- 3.

ABSTRACT

This project is about E-Public Problems. This E-Public Problems app provides an interface to the public to post their social complaints like water, drainage, municipality, roads problems faced by them and also police, fire, hospital, revenue, transport, revenue, excise related problems too. This app basically consists of all the fields discussed above. Public can enter into app and select the field they need and post their complaint over there. That complaint reaches to respective department and their problem can be solved as soon as possible without any inconvenience to them. In today's internet and busy days it is very hard to go to offices and complaint about any problem especially when it is a social problem. At present we already have few apps regarding few departments separately. This app consists of all the fields at one place so that people may not install many apps for different departments.

Table Of contents

CHAPTER NO.	TITLE	PAGE NO.
	Abstract	5
1	Introduction	7
	Requirements	
	- Functional requirements	
	- Non-functional requirements	
	System requirement specification	
	Use case diagram	
2	Design	11
	- Software Development Lifecycle	
	- Life cycle model	
	UML concepts	
3	Modules	22
4	Testing	24
	- Dimensions of testing	
	- Stages of testing	
	- Types of testing	
	- Test approach	
	Test cases	
5	User Manual including snapshots of your application usage	34
	- About java	
	- About SQ lite	
	- About android	
	Installing the Java Development Kit on Windows	
	Installation of android studio	
	Usage of our application	
6	Feasibility	54
7	Conclusion and future scope	56
8	Bibliography	57

1. INTRODUCTION

India is still a developing nation. So some things need to be worked until they are perfect. India has a total of about 2 million kilometers of roads out of which 960,000 kilometers are surfaced roads and about 1 million kilometers of roads in do not have the capacity to handle extreme weather and natural calamities. In order to notify people about the alerts like accidents on roads, climate like fog, heavy rainfall etc.

Disaster Management as a subject essentially deals with management of resources and information as far as a disastrous event is concerned and also how effectively and seamlessly one coordinates these resources.

Disaster management, at the individual and organizational level, deals with issues of planning, coordinating, communication and risk management. A book to teach disaster management would accomplish very little unless a broad spectrum of knowledge is imparted regarding effects of various types of disasters

- * To improve Human and Institutional Capacity.
- * To create awareness on disasters through intensive public education
- * To ensure disaster prevention, risk and vulnerability reduction, as a means of reducing the impact of disasters on society

Equally important is the knowledge about various international and national agencies involved in disaster relief and humanitarian assistance. This book covers all these subjects in order to enhance the ability of students and professionals for better disaster response.

Existing System:

In the present scenario, to file any complaint regarding the society the public has to go the particular office and must file a complaint. Otherwise each department has an application in which the public can upload their complaints. This is a time consuming process and much of manual work is needed to perform the operations. For the status of complaint the public has to visit the office again and again.

Proposed System:

The proposed system provides an interface to the public to post their social complaints directly through the application instead of going to the particular office.

- The users can select the department and can upload the complaint.
- The particular department officer will accept/reject the complaint.
- All the status regarding the complaint is tracked by the admin.
- The user can give feedback and rating after the complaint is resolved.

2. Requirements

2.1 Functional Requirements

2.1.1 Modules of the System

Admin

Admin can view all the complaints posted by the users. And the status of the complaints can be viewed.

Officer

The complaints posted by the user will be received by the particular officer. The respective officer will view the complaints and can accept or reject the complaints. After accepting if the problem is solved then the officer will update as resolved otherwise not resolved.

User

The user will register in order to upload the complaints. User will upload the complaint with an image. The status of the complaint can be viewed by the user.

2.1 Nonfunctional requirements

2.2.1 Performance Requirements

The buffer and cache memory used by sql server can be set to optimum for faster processing.

2.2.2 Safety Requirements

No harm is expected from the use of the product either to the OS or any data that resides on the client system.

2.2.3 Product Security Requirements

The product is protected from un-authorized users from using it. The system allows only authenticated users to work on the application. The user of the system is Admin and user.

2.2.4 Software Quality Attributes

The product is user friendly and its accessibility is from the clients. As it is developed in java it is highly interoperable with OS that have provided support for JVM (Server side). The system requires less maintenance as the data is stored in relational databases on the server side.

System Requirement Specifications

Software Requirements:

Operating System	:	Windows 7/10
Tool Kit	:	Android (API 23 or above)
Technology	:	Java
IDE	:	Studio
Database	:	SQLite
Frontend	:	XML

Hardware Requirements:

Hard Disk	:	500 GB (Min.)
RAM	:	512 MB (Min.)

Use case diagram:

Use case diagrams are a set of use cases, actors, and their relationships. They represent the use case view of a system. A use case represents a particular functionality of a system. Hence, use case diagram is used to describe the relationships among the functionalities and their internal/external controllers. These controllers are known as actors.

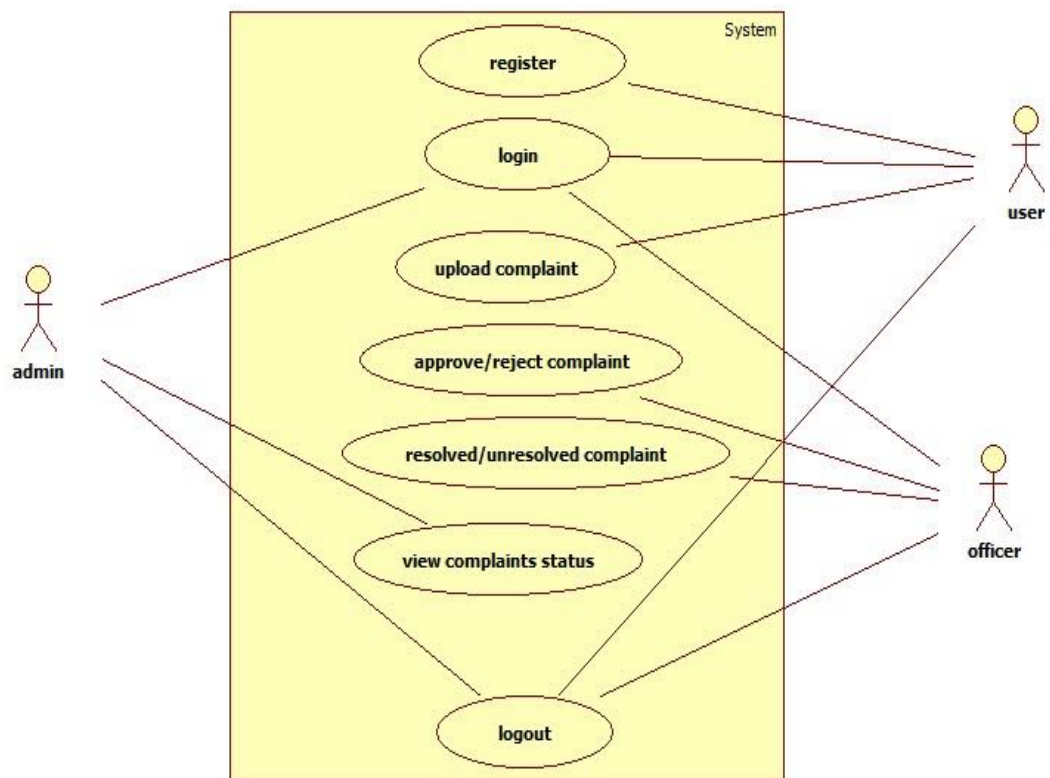
A use case diagram is usually simple. It does not show the detail of the use cases:

- It only summarizes some of the relationships between use cases, actors, and systems.
- It does not show the order in which steps are performed to achieve the goals of each use case.

Use cases share different kinds of relationships. Defining the relationship between two use cases is the decision of the software analysts of the use case diagram. A relationship between two use cases is basically modeling the dependency between the two use cases. The reuse of an existing use case by using different types of relationships reduces the overall effort required in developing a system.

Use cases are a set of actions, services, and functions that the system needs to perform. In this context, a "system" is something being developed or operated, such as a web site. The "actors" are people or entities operating under defined roles within the system.

Use case diagram for disaster management system:



2. DESIGN

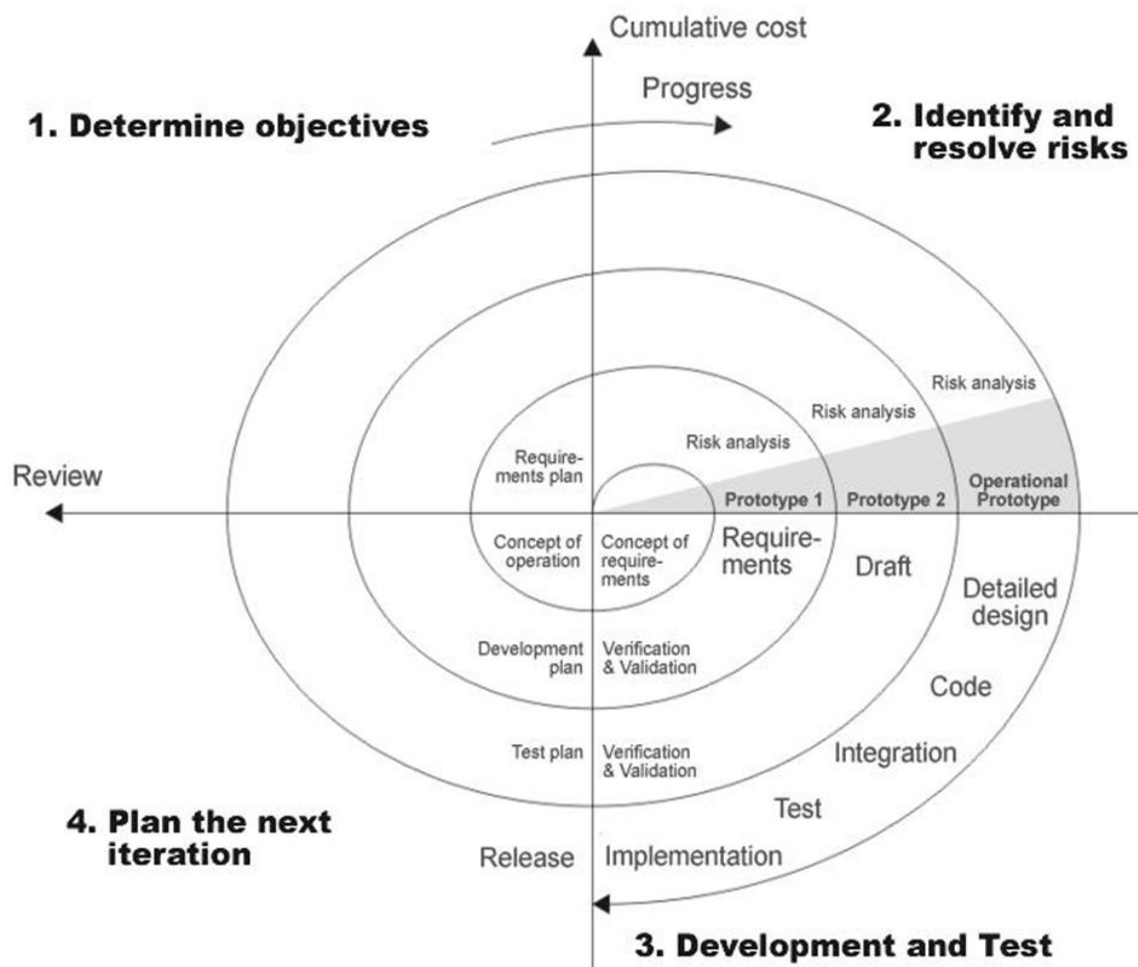
System can be defined as the “a set of fact, principles, rules etc., classified and arranged in an orderly form so as to show a logical plan linking the various parts” here the system design defines the computer based information system. The primary objective is to identify user requirements and to build a system that satisfies these requirements.

Design is much more creative process than analysis. Design is the first step in the development of any system or product. Design can be defined as “the process of applying various techniques and principles for the purpose of defining a device, a process or a system in sufficient detail to permit its physical realization”.

It involves four major steps they are:

1. Understanding how the system is working now.
2. Finding out what the system does now.
3. Understanding what the new system will do and
4. Understanding how the new system will work.

SDLC (Spiral Model):



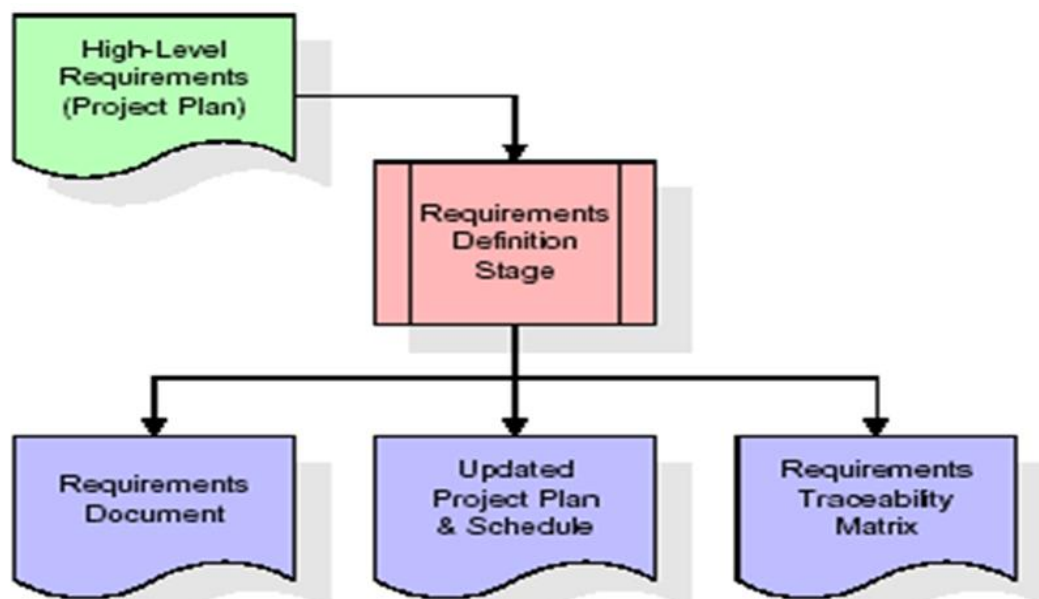
SDLC is nothing but Software Development Life Cycle. It is a standard which is used by software industry to develop good software.

Stages in SDLC:

- ◆ Requirement Gathering
- ◆ Analysis
- ◆ Designing
- ◆ Coding
- ◆ Testing
- ◆ Maintenance

Requirements Gathering stage:

The requirements gathering process takes as its input the goals identified in the high-level requirements section of the project plan. Each goal will be refined into a set of one or more requirements. These requirements define the major functions of the intended application, define Operational data areas and reference data areas, and define the initial data entities. Major functions include critical processes to be managed, as well as mission critical inputs, outputs and reports. A user class hierarchy is developed and associated with these major functions, data areas, and data entities. Each of these definitions is termed a Requirement. Requirements are identified by unique requirement identifiers and, at minimum, contain a requirement title and Textual description.



These requirements are fully described in the primary deliverables for this stage: the Requirements Document and the Requirements Traceability Matrix (RTM). The requirements document contains complete descriptions of each requirement, including diagrams and references to external documents as necessary. Note that detailed listings of database tables and fields are *not* included in the requirements document.

The title of each requirement is also placed into the first version of the RTM, along with the title of each goal from the project plan. The purpose of the RTM is to show that the product components developed during each stage of the software development lifecycle are formally connected to the components developed in prior stages.

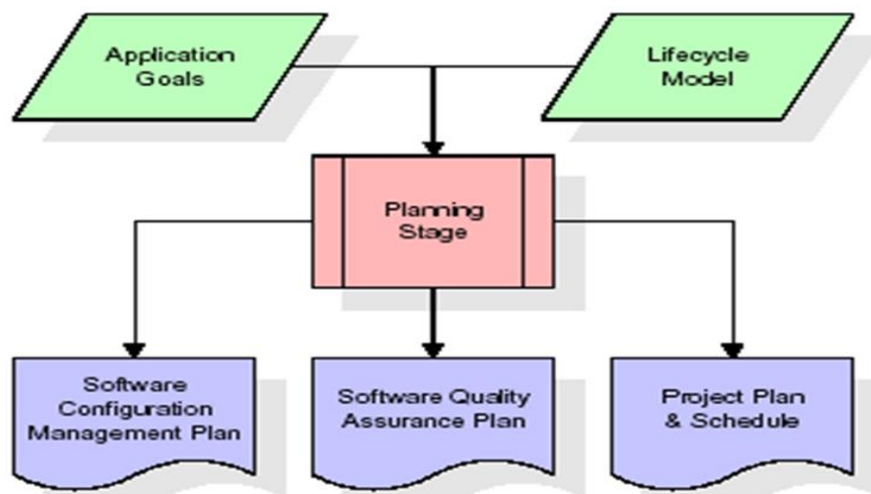
In the requirements stage, the RTM consists of a list of high-level requirements, or goals, by title, with a listing of associated requirements for each goal, listed by requirement title. In this hierarchical listing, the RTM shows that each requirement developed during this stage is formally linked to a specific product goal. In this format, each requirement can be traced to a specific product goal, hence the term *requirements traceability*.

The outputs of the requirements definition stage include the requirements document, the RTM, and an updated project plan.

- ◆ Feasibility study is all about identification of problems in a project.
- ◆ No. of staff required to handle a project is represented as Team Formation, in this case only modules are individual tasks will be assigned to employees who are working for that project.
- ◆ Project Specifications are all about representing of various possible inputs submitting to the server and corresponding outputs along with reports maintained by administrator

Analysis Stage:

The planning stage establishes a bird's eye view of the intended software product, and uses this to establish the basic project structure, evaluate feasibility and risks associated with the project, and describe appropriate management and technical approaches.



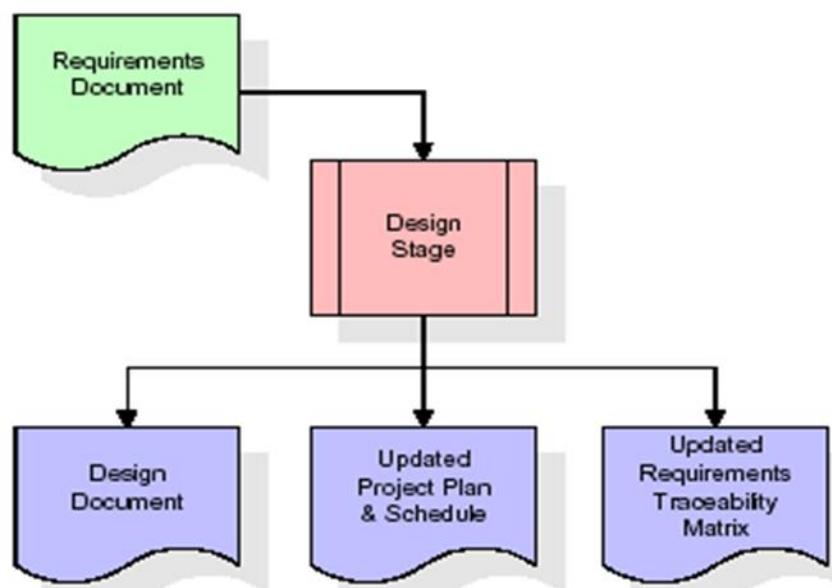
The most critical section of the project plan is a listing of high-level product requirements, also referred to as goals. All of the software product requirements to be developed during the requirements definition stage flow from one or more of these goals. The minimum information for each goal consists of a title and textual description, although additional information and references to external documents may be included. The outputs of the project planning stage are the configuration management plan, the quality assurance plan, and the project plan and schedule, with a detailed listing of scheduled activities for the upcoming Requirements stage, and high level estimates of effort for the out stages.

Designing:

The design stage takes as its initial input the requirements identified in the approved requirements document. For each requirement, a set of one or more design elements will be produced as a result of interviews, workshops, and/or prototype efforts.

Design elements describe the desired software features in detail, and generally include functional hierarchy diagrams, screen layout diagrams, tables of business rules, business process diagrams, pseudo code, and a complete entity-relationship diagram with a full data dictionary.

These design elements are intended to describe the software in sufficient detail that skilled programmers may develop the software with minimal additional input.

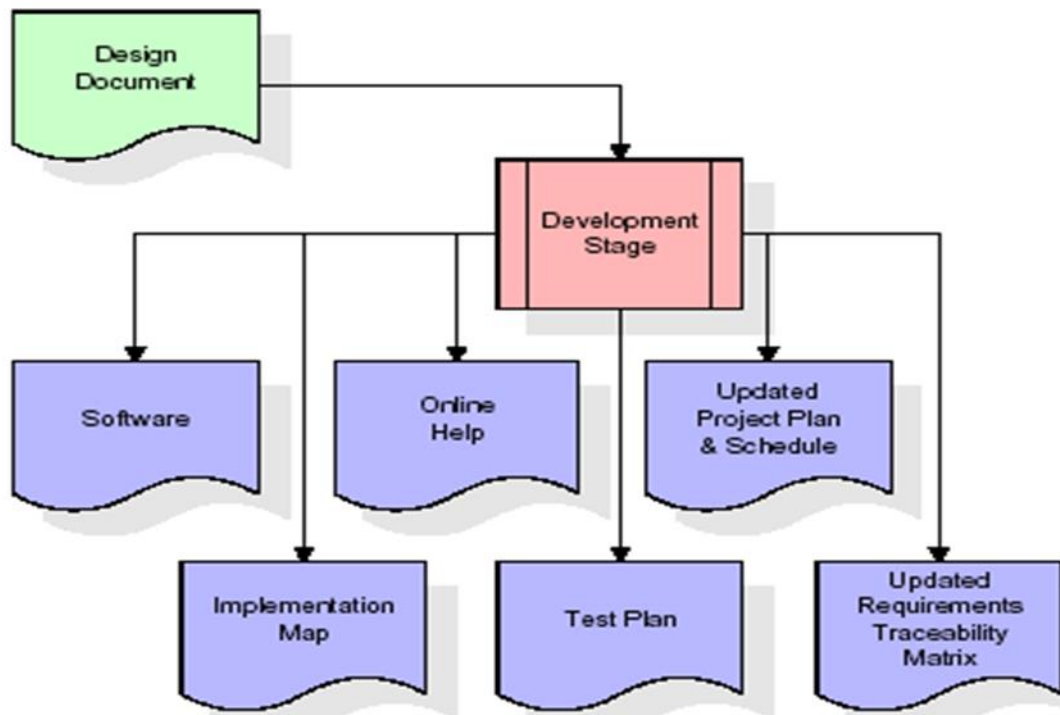


When the design document is finalized and accepted, the RTM is updated to show that each design element is formally associated with a specific requirement.

The outputs of the design stage are the design document, an updated RTM, and an updated project plan.

Development (Coding) Stage:

The development stage takes as its primary input the design elements described in the approved design document. For each design element, a set of one or more software artifacts will be produced. Software artifacts include but are not limited to menus, dialogs, and data management forms, data reporting formats, and specialized procedures and functions. Appropriate test cases will be developed for each set of functionally related software artifacts, and an online help system will be developed to guide users in their interactions with the software.

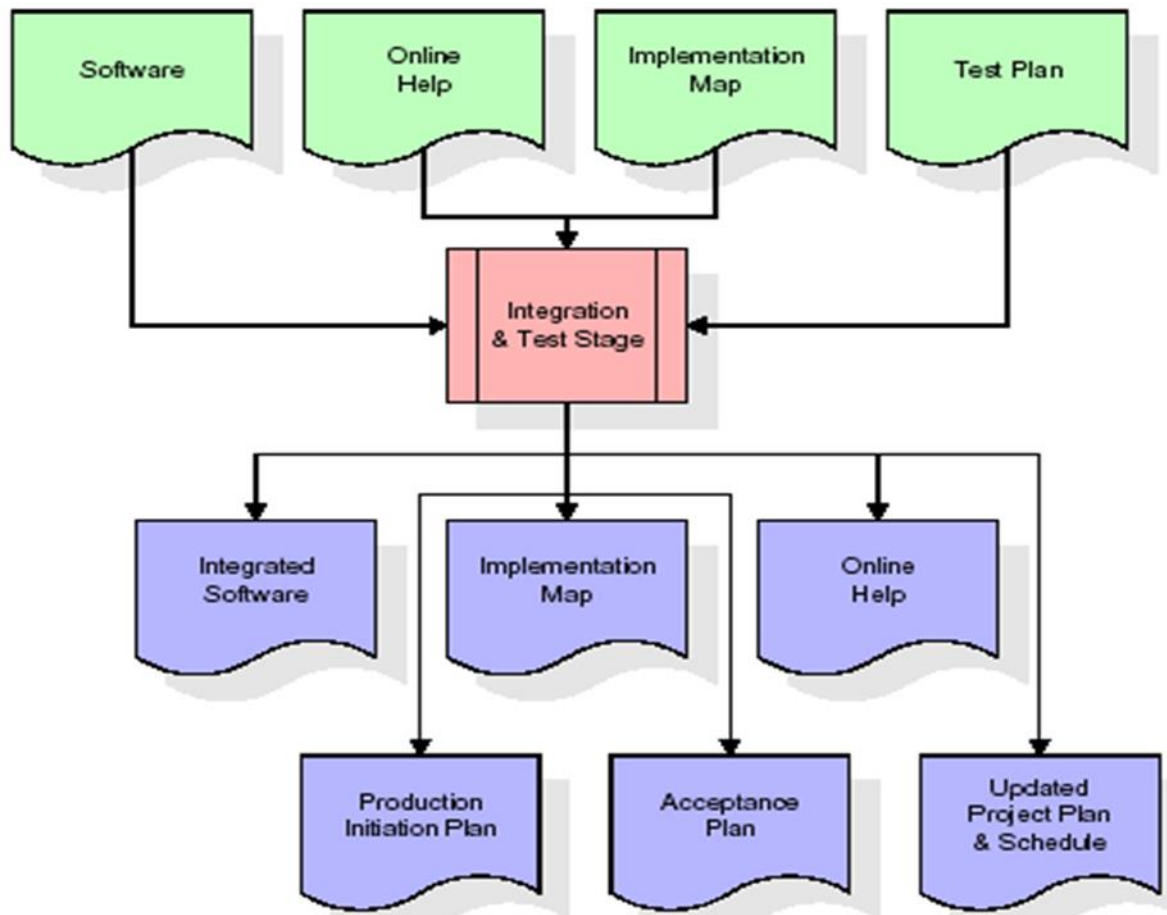


The RTM will be updated to show that each developed artifact is linked to a specific design element, and that each developed artifact has one or more corresponding test case items. At this point, the RTM is in its final configuration. The outputs of the development stage include a fully functional set of software that satisfies the requirements and design elements previously documented, an online help system that describes the operation of the software, an implementation map that identifies the primary code entry points for all major system functions, a test plan that describes the test cases to be used to validate the correctness and completeness of the software, an updated RTM, and an updated project plan.

Integration & Test Stage:

During the integration and test stage, the software artifacts, online help, and test data are migrated from the development environment to a separate test environment. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite confirms a robust and complete migration capability. During this stage, reference data is finalized for production use and production

Users are identified and linked to their appropriate roles. The final reference data (or links to reference data source files) and production user list are compiled into the Production Initiation Plan.



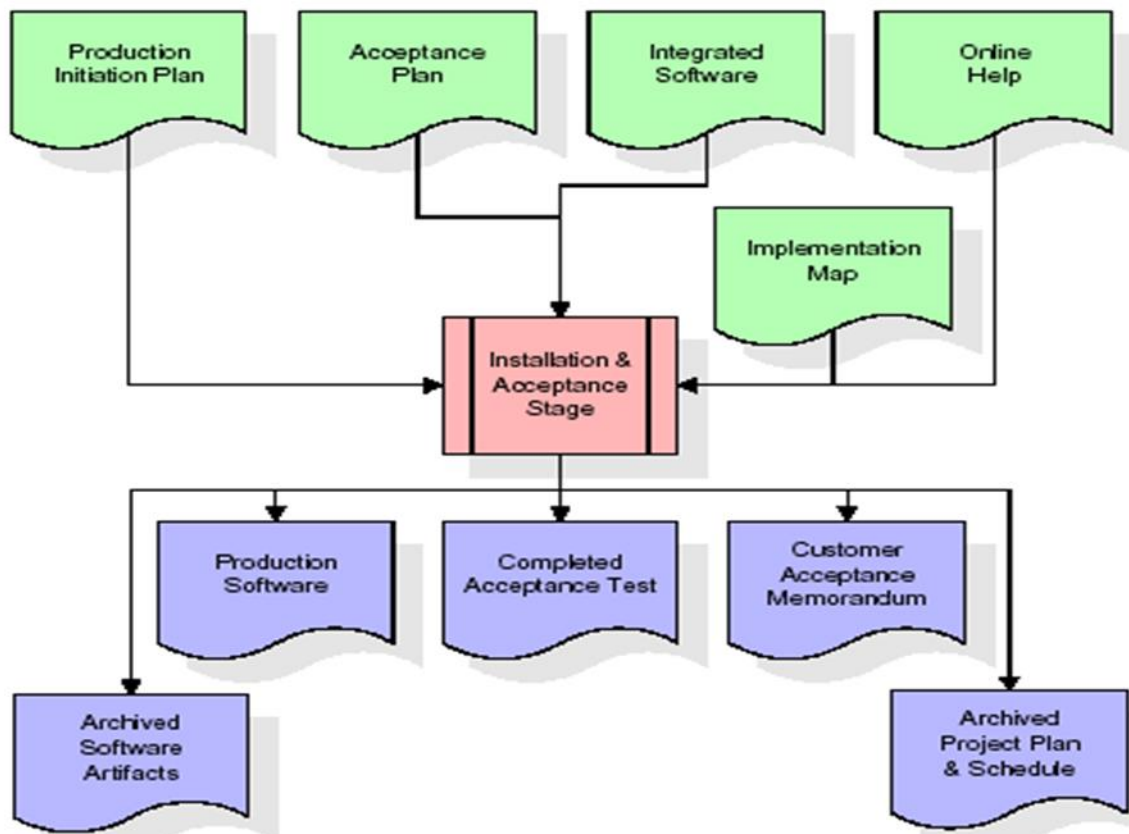
The outputs of the integration and test stage include an integrated set of software, an online help system, an implementation map, a production initiation plan that describes reference data and production users, an acceptance plan which contains the final suite of test cases, and an updated project plan.

Installation & Acceptance Test:

During the installation and acceptance stage, the software artifacts, online help, and initial production data are loaded onto the production server. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite is a prerequisite to acceptance of the software by the customer.

After customer personnel have verified that the initial production data load is correct and the test suite has been executed with satisfactory results, the customer formally accepts the delivery of the software.

The primary outputs of the installation and acceptance stage include a production application, a completed acceptance test suite, and a memorandum of customer acceptance of the software.



Finally, the PDR enters the last of the actual labor data into the project schedule and locks the project as a permanent project record. At this point the PDR "locks" the project by archiving all software items, the implementation map, the source code, and the documentation for future reference.

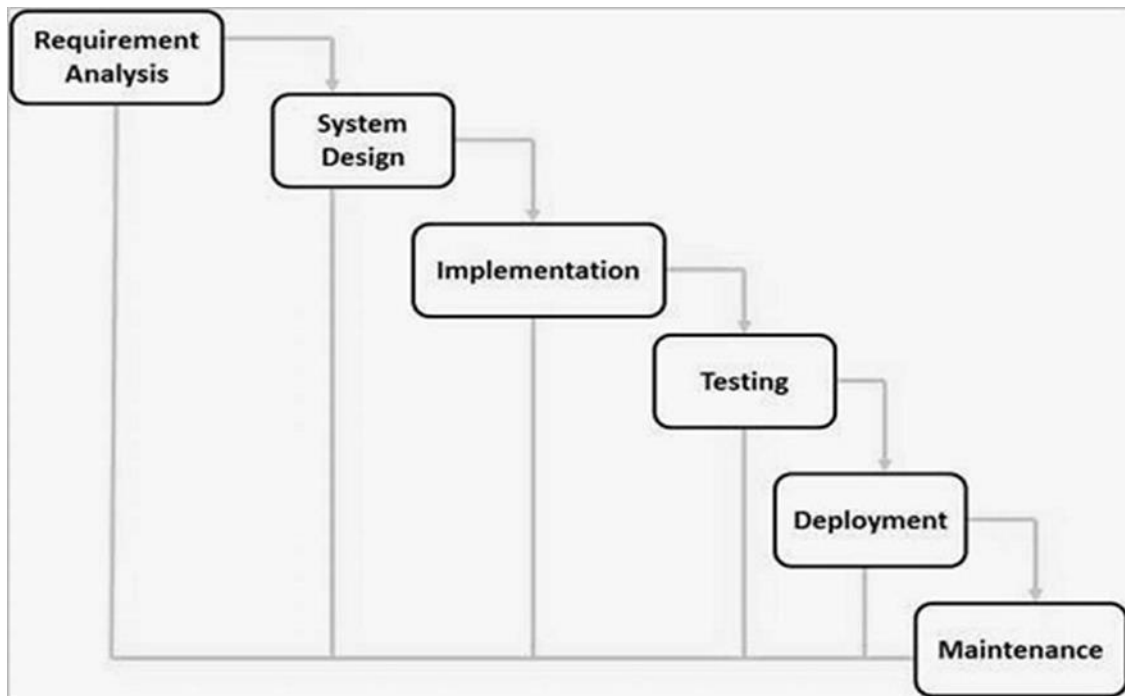
Maintenance:

Outer rectangle represents maintenance of a project, Maintenance team will start with requirement study, understanding of documentation later employees will be assigned work and they will undergo training on that particular assigned category.

For this life cycle there is no end, it will be continued so on like an umbrella (no ending point to umbrella sticks).

Life cycle model

It is also referred to as a linear-sequential life cycle model. It is very simple to understand and use. In a waterfall model, each phase must be completed fully before the next phase can begin. This type of software development model is basically used for the project which is small and there are no uncertain requirements. At the end of each phase, a review takes place to determine if the project is on the right path and whether to continue or discard the project. In this model software testing starts only after the development is complete. In waterfall model phases do not overlap.



Advantages of waterfall model:

- This model is simple and easy to understand and use.
- It is easy to manage due to the rigidity of the model – each phase has specific deliverables and a review process.
- In this model phases are processed and completed one at a time. Phases do not overlap.
- Waterfall model works well for smaller projects where requirements are very well understood.

UML Concepts:

UML diagrams are the ultimate output of the entire discussion. All the elements, relationships are used to make a complete UML diagram and the diagram represents a system.

The visual effect of the UML diagram is the most important part of the entire process. All the other elements are used to make it complete.

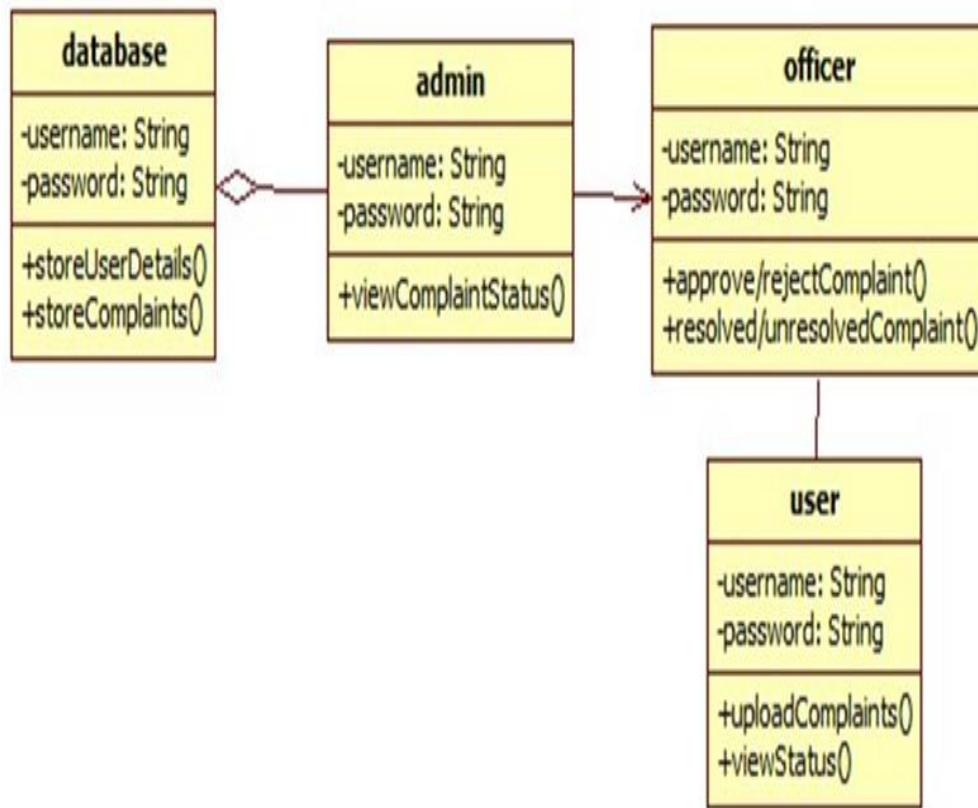
UML includes the following nine diagrams, the details of which are described in the subsequent chapters.

- Class diagram
- Use case diagram
- Sequence diagram
- Activity diagram

Class Diagram

Class diagrams are the most common diagrams used in UML. Class diagram consists of classes, interfaces, associations, and collaboration. Class diagrams basically represent the object-oriented view of a system, which is static in nature. Active class is used in a class diagram to represent the concurrency of the system.

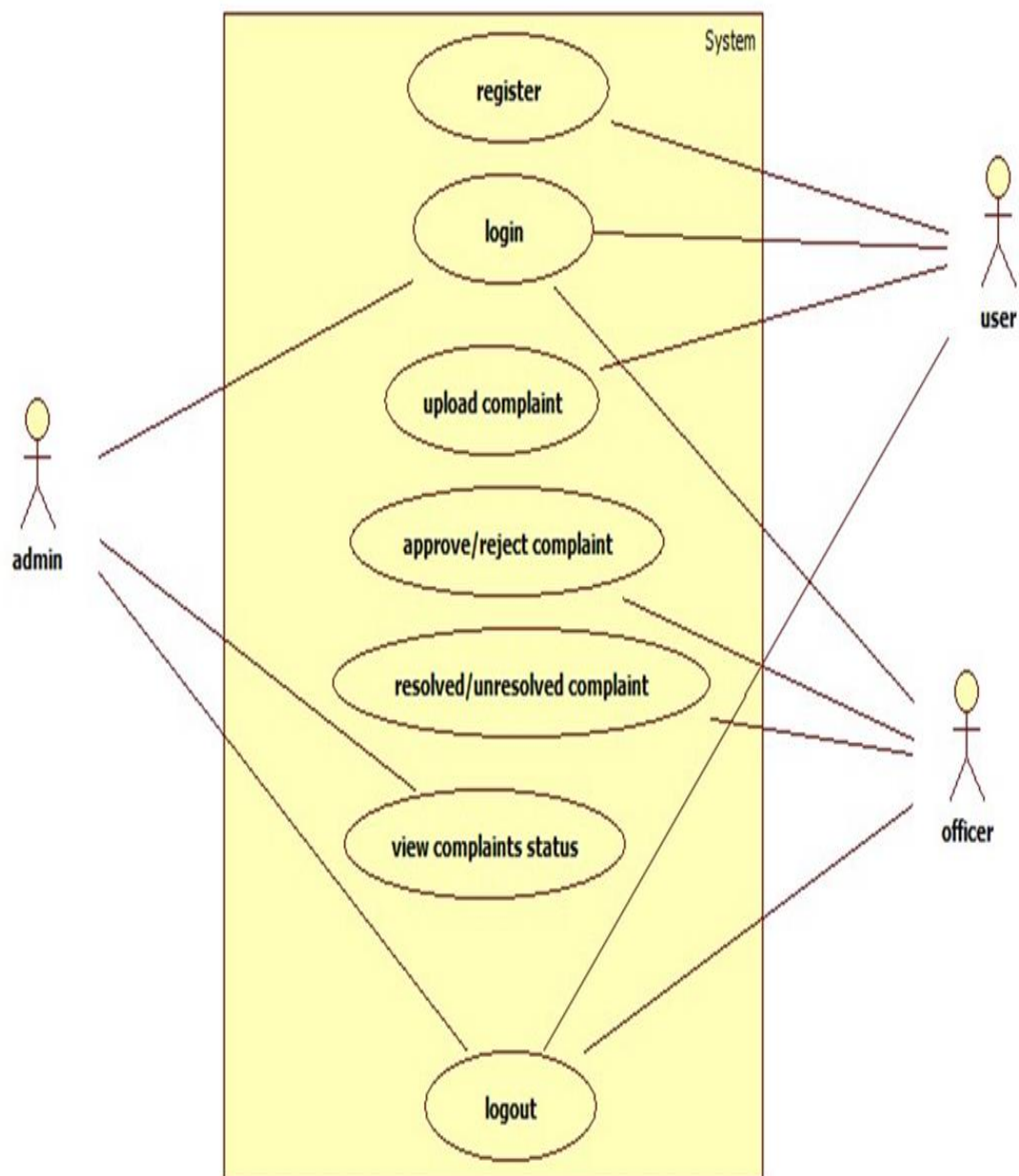
Class diagram represents the object orientation of a system. Hence, it is generally used for development purpose. This is the most widely used diagram at the time of system construction.



Use Case Diagram

Use case diagrams are a set of use cases, actors, and their relationships. They represent the use case view of a system.

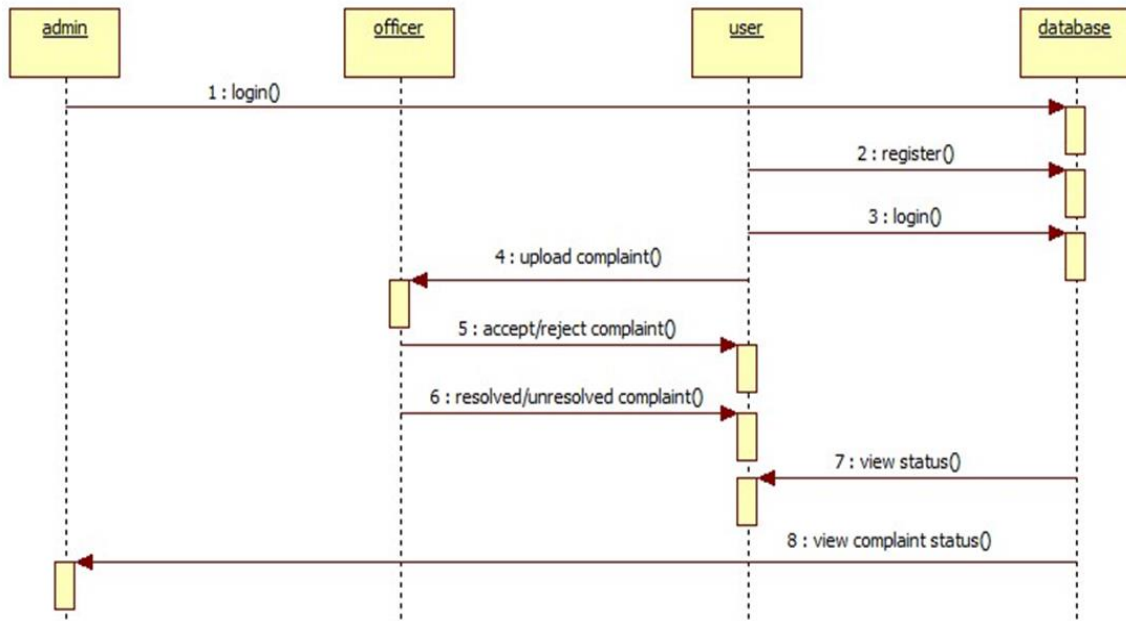
A use case represents a particular functionality of a system. Hence, use case diagram is used to describe the relationships among the functionalities and their internal/external controllers. These controllers are known as actors.



Sequence Diagram

A sequence diagram is an interaction diagram. From the name, it is clear that the diagram deals with some sequences, which are the sequence of messages flowing from one object to another.

Interaction among the components of a system is very important from implementation and execution perspective. Sequence diagram is used to visualize the sequence of calls in a system to perform a specific functionality.

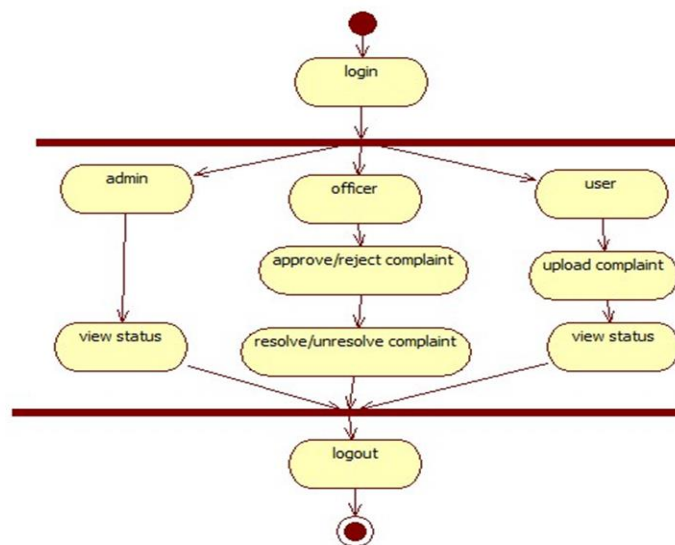


Activity Diagram

Activity diagram describes the flow of control in a system. It consists of activities and links. The flow can be sequential, concurrent, or branched.

Activities are nothing but the functions of a system. Numbers of activity diagrams are prepared to capture the entire flow in a system.

Activity diagrams are used to visualize the flow of controls in a system. This is prepared to have an idea of how the system will work when executed.



3. MODULES

Modules

- User
- Officer
- Admin

User:

The user will register in order to upload the complaints. User will upload the complaint with an image. The status of the complaint can be viewed by the user.

Complaint:

This is used to complaint about the natural disaster by uploading the photo of that particular disaster and giving address where it is been done, giving description about that disaster.

Status:

This is used to view our complaints whether the complaint is approved by admin or not.

Timeline:

This is used to view the timeline of complaint like when it is approved or rejected on a particular time.

Logout:

This is used to logout from the user login.

Officer:

The complaints posted by the user will be received by the particular officer. The respective officer will view the complaints and can accept or reject the complaints. After accepting if the problem is solved then the officer will update as resolved otherwise not resolved.

Complaints:

This is used to view the complaints of the particular officer from the particular department like Education, Health, Law and order, Transport and Others.

Approved:

This is used to approve the complaints or reject the complaints by a particular officer on which the complaint is based on the particular department.

Logout:

This is used to logout from the officer login.

Admin:

Admin can view all the complaints posted by the users. And the status of the complaints can be viewed.

Complaints:

This is used to view the complaints that number of accepted complaints and the number of rejected complaints.

Upload:

This is used to upload the newsfeed by the admin that where the alerts will be given for the user for the diversion from where they are travelling and showing them the shortest route to their destination.

Logout:

This is used to logout from the admin login.

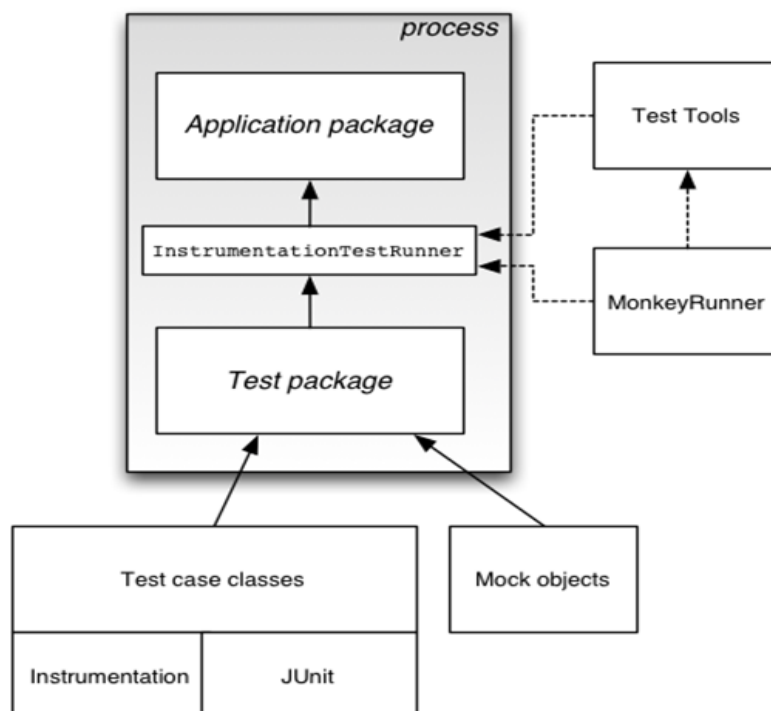
4. Testing

The Android testing framework, an integral part of the development environment, provides an architecture and powerful tools that help you test every aspect of your application at every level from unit to framework.

The testing framework has these key features:

- Test suites are contained in test packages that are similar to main application packages, so you don't need to learn a new set of tools or techniques for designing and building tests.
- The SDK tools for building and tests are available in Eclipse with ADT, and also in command-line form for use with other IDEs. These tools get information from the project of the application under test and use this information to automatically create the build files, manifest file, and directory structure for the test package.
- The SDK also provides monkey runner, an API testing devices with Python programs, and UI/Application Exerciser Monkey, a command-line tool for stress-testing UIs by sending pseudo-random events to a device.
- This document describes the fundamentals of the Android testing framework, including the structure of tests, the APIs that you use to develop tests, and the tools that you use to run tests and view results. The document assumes you have a basic knowledge of Android application programming and JUnit testing methodology.

The following diagram summarizes the testing framework:



Test Structure:

Android's build and test tools assume that test projects are organized into a standard structure of tests, test case classes, test packages, and test projects.

AndroidTestCase:

A useful general test case class, especially if you are just starting out with Android testing, is `AndroidTestCase`. It extends both `Test Case` and `Assert`. It provides the Java Unit-standard `setUp ()`

Component-specific test cases:

A key feature of the Android testing framework is its component-specific test case classes. These address specific component testing needs with methods for fixture setup and teardown and component lifecycle control. They also provide methods for setting up mock objects. These classes are described in the component-specific testing topics:

- Activity Testing
- Content Provider Testing
- Service Testing

Android does not provide a separate test case class for `Broadcast Receiver`. Instead, test a broadcast Receiver by testing the component that sends it `Intent` objects, to verify that the broadcast Receiver responds correctly.

ApplicationTestCase:

You use the `ApplicationTestCase` test case class to test the setup and teardown of `Application` objects. These objects maintain the global state of information that applies to all the components in an application package. The test case can be useful in verifying that the `<application>` element in the manifest file is correctly set up. Note, however, that this test case does not allow you to control testing of the components within your application package.

Instrumentation Test Case:

If you want to use instrumentation methods in a test case class, you must use `InstrumentationTestCase` or one of its subclasses. The Activity test cases extend this base class with other functionality that assists in Activity testing.

Assertion classes:

- `More Asserts` contains more powerful assertions such as `assertContainsRegex (String, String)`, which does regular expression matching.
- `View Asserts` contains useful assertions about Views. For example it contains `assert Has Screen Coordinates (View, View, int, int)` that tests if a View has a particular X and Y position on the visible screen. These assert simplify testing of geometry and alignment in the UI.

Mock object classes:

- To facilitate dependency injection in testing, Android provides classes that create mock system objects such as Context objects, Content Provider objects, Content Resolver objects, and Service objects. Some test cases also provide mock Intent objects. You use these mocks both to isolate tests from the rest of the system and to facilitate dependency injection for testing. These classes are found in the Java packages `android`, `Test` and `android.test.mock`.
- Mock objects isolate tests from a running system by stubbing out or overriding normal operations. For example, a `MockContentResolver` replaces the normal resolver framework with its own local framework, which is isolated from the rest of the system. `MockContentResolver` also stubs out the `notifyChange(Uri, Content Observer, Boolean)` method so that observer objects outside the test environment are not accidentally triggered.
- Mock object classes also facilitate dependency injection by providing a subclass of the normal object that is non-functional except for overrides you define. For example, the `Mock Resources` object provides a subclass of `Resources` in which all the methods throw `Exceptions` when called. To use it, you override only those methods that must provide information.
- These are the mock object classes available in Android.

Contexts for testing:

Android provides two Context classes that are useful for testing:

`Isolated Context` provides an isolated Context; File, directory, and database operations that use this Context take place in a test area. Though its functionality is limited, this Context has enough stub code to respond to system calls.

This class allows you to test an application's data operations without affecting real data that may be present on the device.

`RenamingDelegatingContext` provides a Context in which most functions are handled by an existing Context, but file and database operations are handled by an `Isolated Context`. The isolated part uses a test directory and creates special file and directory names. You can control the naming yourself, or let the constructor determine it automatically.

This object provides a quick way to set up an isolated area for data operations, while keeping normal functionality for all other Context operations.

Test Cases:

Test cases are run by a test runner class that loads the test case class, set ups, runs, and tears down each test. An Android test runner must also be instrumented, so that the system utility for starting

applications can control how the test package loads test cases and the application under test. You tell the Android platform which instrumented test runner to use by setting a value in the test package's manifest file. **Seeing Test Results:**

The Android testing framework returns test results back to the tool that started the test. If you run a test in Eclipse with ADT, the results are displayed in a new JUnit view pane. If you run a test from the command line, the results are displayed in STDOUT. In both cases, you see a test summary that displays the name of each test case and method that was run. You also see all the assertion failures that occurred. These include pointers to the line in the test code where the failure occurred. Assertion failures also list the expected value and actual value.

Working with Package names:

In the test environment, you work with both Android application package names and Java package identifiers. Both use the same naming format, but they represent substantially different entities. You need to know the difference to set up your tests correctly.

An Android package name is a unique system name for a .apk file, set by the "android: package" attribute of the <manifest> element in the package's manifest. The Android package name of your test package must be different from the Android package name of the application under test. By default, Android tools create the test package name by appending ".test" to the package name of the application under test.

The test package also uses an Android package name to target the application package it tests. This is set in the "android: target Package" attribute of the <instrumentation> element in the test package's manifest.

A Java package identifier applies to a source file. This package name reflects the directory path of the source file. It also affects the visibility of classes and members to each other.

What to Test:

The topic What to Test describes the key functionality you should test in an Android application, and the key situations that might affect that functionality.

Most unit testing is specific to the Android component you are testing. The topics Activity Testing, Content Provider Testing, and Service Testing each have a section entitled "What To Test" that lists possible testing areas.

When possible, you should run these tests on an actual device. If this is not possible, you can use the Android Emulator with Android Virtual Devices configured for the hardware, screens, and versions you want to test.

Overview of testing:

Testing is a process, which reveals errors in the program. It is the major quality measure employed during software development. During software development. During testing, the program is executed with a set of test cases and the output of the program for the test cases is evaluated to determine if the program is performing as it is expected to perform.

Testing objectives:

To ensure that during operation the system will perform as per specification. To make sure that system meets the user requirements during operation. To make sure that during the operation, incorrect input, processing and output will be detected. To see that when correct inputs are fed to the system the outputs are correct. To verify that the controls incorporated in the same system as intended. Testing is a process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding a yet undiscovered error.

The software developed has been tested successfully using the following testing strategies and any errors that are encountered are corrected and again the part of the program or the procedure or function is put to testing until all the errors are removed. A successful test is one that uncovers a yet undiscovered error.

Note that the result of the system testing will prove that the system is working correctly. It will give confidence to system designer, users of the system, prevent frustration during implementation process etc.

Dimensions of testing:

To assess product quality, different kinds of tests, each one with a different focus, are needed. These tests can be categorized by several dimensions:

Quality dimension: The major quality characteristic or attribute that is the focus of test.

Stage of testing: The point in the lifecycle at which the test, usually limited to a single quality dimension.

Type of testing: The specific test objective for an individual test, usually limited to a single quality dimension.

Stages of testing

Testing is not a single activity, executed all at once. Testing is executed against different types of targets in different stages of the software development. Test stages progress from testing small elements of the system, such as components (unit testing), to testing completed systems (system testing).

The four stages have the following purposes

Unit Testing:

The smallest testable elements of the system are tested individually, typically at the same time those elements are implemented.

Integration Testing:

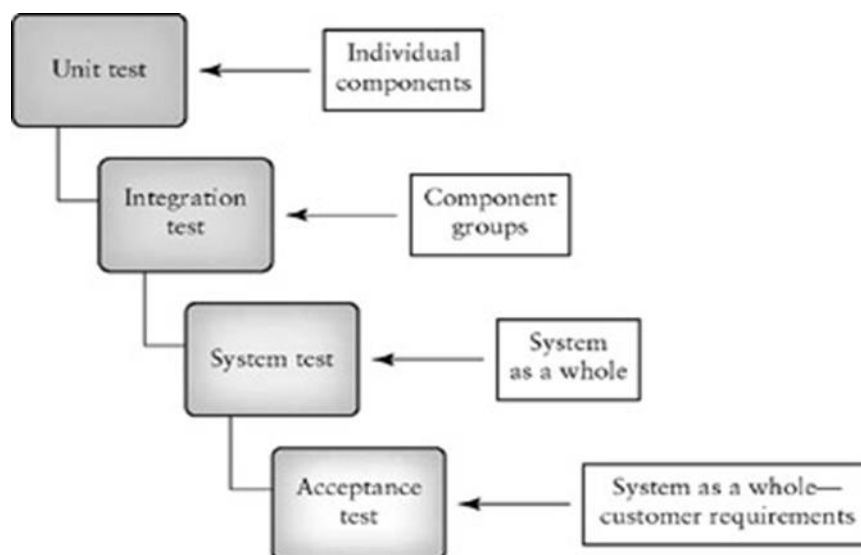
The integrated units (or components or subsystems) are tested.

System Testing:

The complete application is tested by end users for determining readiness for deployment. These stages occur throughout the life cycle, with varying emphasis. An early conceptual prototype user in the inception phase to assess the viability of the product vision will be subjected to acceptance tests. Architecture prototype developed during the elaboration phase be performance of key architectural element.

Acceptance testing:

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.



Types of testing

To make sure that the system does not have errors, the different levels of testing strategies that are applied at differing phases of software development are:

Unit testing:

Unit Testing is done on individual modules as they are completed and become executable. It is confined only to the designer's requirements.

Each module can be tested using the following two Strategies:

Black Box testing:

In this testing, some test cases are generated as input conditions that fully execute all functional requirements for the program. This testing has been used to find errors in the following categories:

- Incorrect or missing functions.
- Interface errors
- Errors in data structure or external database access
- Performance errors.
- Initialization and termination errors.

In this testing only the output is checked for correctness. The logical flow of the data is not checked.

White Box testing:

In this the test cases are generated on the logic of each module by drawing flow graphs of that module and logical decisions are tested on all the cases. It has been used to generate the test cases in the following cases:

- Guarantee that all independent paths have been executed.
- Execute all logical decisions on their true and false sides.
- Execute all loops at their boundaries and within their operational bounds.
- Execute internal data structures to ensure their validity.

Integration testing:

Integration testing ensures that software and subsystems work together a whole. It tests the interface of all the modules to make sure that the modules behave properly when integrated together.

System testing:

Involves in-house testing of the entire system before delivery to the user. Its aim is to satisfy the user the system meets all requirements of the client's specifications.

Functional testing:

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

Acceptance testing:

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test approach

Testing can be done in two ways:

- Bottom up approach
- Top down approach

Bottom up approach:

Testing can be performed starting from smallest and lowest level modules and proceeding one at a time. For each module in bottom up testing a short program executes the module and provides the needed data so that the module is asked to perform the way it will when embedded within the larger system. When bottom level modules are tested attention turns to those on the next level that use the lower level ones they are tested individually and then linked with the previously examined lower level modules.

Top down approach:

This type of testing starts from upper level modules. Since the detailed activities usually performed in the lower level routines are not provided stubs are written. A stub is a module shell called by upper level module and that when reached properly will return a message to the calling module indicating that proper interaction occurred. No attempt is made to verify the correctness of the lower level modules.

TEST CASES

Module name: Admin

S. No	Test Case	Input	Expected Output	Actual Output	Pass/Fail
1.	Check admin login functionality	Admin username and password	Login must be successful	Login success	Pass
2.	View complaint status	Request for complaint status	Must be viewed successfully	View success	Pass

Module name: Officer

S. No	Test Case	Input	Expected Output	Actual Output	Pass/Fail
1.	Check officer login functionality	Officer username and password	Login must be successful	Login success	Pass
2.	Accept/Reject complaints	Request for complaints from user	Must be accepted/rejected successfully	Accept/reject success	Pass
3.	Resolve/Unresolved complaints	Complaints which are accepted	Must be resolved/unresolved successfully	Resolved/unresolved successfully	Pass

Module name: User

S. No	Test Case	Input	Expected Output	Actual Output	Pass/Fail
1.	Check User registration	Details of User	Registration must be successful	Registration success	Pass
2.	Check user login functionality	User username and password	Login must be successful	Login success	Pass
3.	Upload complaints	Details regarding complaints	Upload must be successful	Upload success	Pass
4.	Give feedback	Feedback about the resolved complaints	Feedback must be given successfully	Feedback given successfully	Pass

5. User Manual including snapshots of your application usage

About Java:

Initially the language was called as “oak” but it was renamed as “java” in 1995. The primary motivation of this language was the need for a platform-independent (i.e. architecture neutral) language that could be used to create software to be embedded in various consumer electronic devices.

- Java is a programmer’s language
- Java is cohesive and consistent
- Except for those constraint imposed by the Internet environment. Java gives the programmer, full control
- Finally Java is to Internet Programming where c was to System Programming.

Importance of Java to the Internet:

Java has had a profound effect on the Internet. This is because; java expands the Universe of objects that can move about freely in Cyberspace. In a network, two categories of objects are transmitted between the server and the personal computer. They are passive information and Dynamic active programs. in the areas of Security and probability. But Java addresses these concerns and by doing so, has opened the door to an exciting new form of program called the Applet.

Applications and applets: An application is a program that runs on our Computer under the operating system of that computer. It is more or less like one creating using C or C++ .Java’s ability to create Applets makes it important. An Applet I san application, designed to be transmitted over the Internet and executed by a Java-compatible web browser. An applet I actually a tiny Java program, dynamically downloaded across the network, just like an image. But the difference is, it is an intelligent program, not just a media file. It can be react to the user input and dynamically change.

Java Architecture:

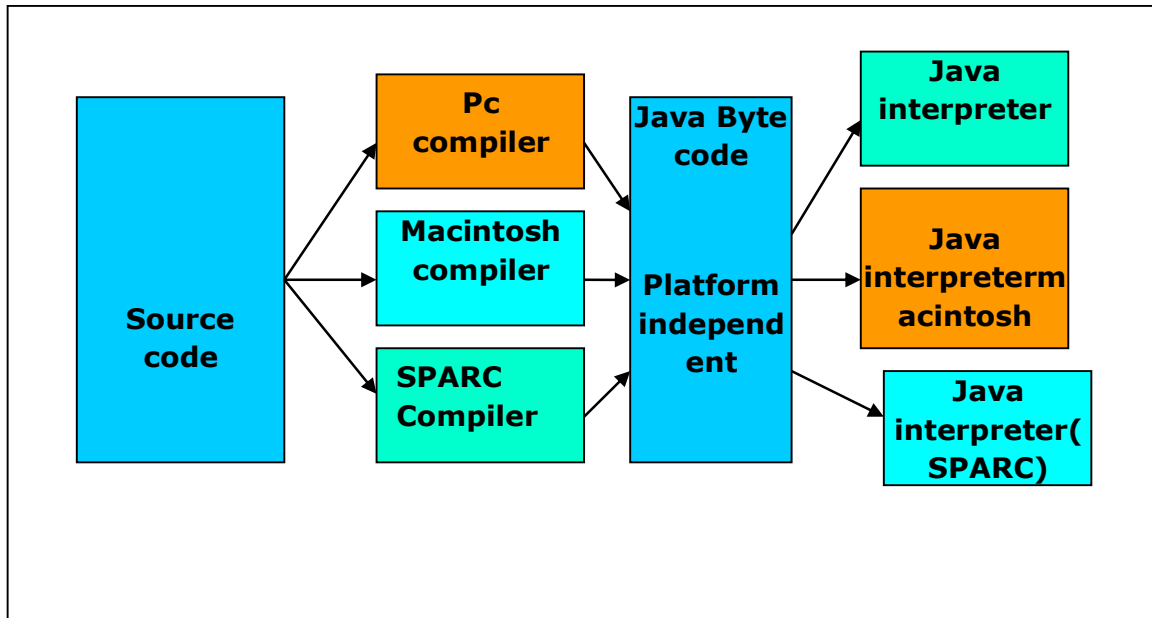
Java architecture provides a portable, robust, high performing environment for development. Java provides portability by compiling the byte codes for the Java Virtual Machine, which is then interpreted on each platform by the run-time environment. Java is a dynamic system, able to load code when needed from a machine in the same room or across the planet.

Compilation of code:

When you compile the code, the Java compiler creates machine code (called byte code) for a hypothetical machine called Java Virtual Machine (JVM). The JVM is supposed to executed the byte code. The JVM is

created for the overcoming the issue of probability. The code is written and compiled for one machine and interpreted on all machines .This machine is called Java Virtual Machine.

Compiling and interpreting java source code.



During run-time the Java interpreter tricks the byte code file into thinking that it is running on a Java Virtual Machine. In reality this could be an Intel Pentium windows 95 or sun SPARCstation running Solaris or Apple Macintosh running system and all could receive code from any computer through internet and run the Applets.

Simple:

Java was designed to be easy for the Professional programmer to learn and to use effectively. If you are an experienced C++ Programmer. Learning Java will oriented features of C++. Most of the confusing concepts from C++ are either left out of Java or implemented in a cleaner, more approachable manner. In Java there are a small number of clearly defined ways to accomplish a given task.

Object oriented:

Java was not designed to be source-code compatible with any other language. This allowed the Java team the freedom to design with a blank state. One outcome of this was a clean usable, pragmatic approach to objects. The object model in Java is simple and easy to extend, while simple types, such as integers, are kept as high-performance non-objects.

Robust:

The multi-platform environment of the web places extraordinary demands on a program, because the program must execute reliably in a variety of systems. The ability to create robust programs. Was given a high priority in the design of Java. Java is strictly typed language; it checks your code at compile time and runtime. Java virtually eliminates the problems of memory management and deal location, which is completely automatic. In a well-written Java program, all run-time errors can and should be managed by your program.

SQLite

- SQLite is an Open Source Database which is embedded into Android. SQLite supports standard relational database features like SQL syntax, transactions and prepared statements. In addition it requires only little memory at runtime (approx. 250 Kbyte).
- SQLite supports the data types TEXT (similar to String in Java), INTEGER (similar to long in Java) and REAL (similar to double in Java). All other types must be converted into one of these fields before saving them in the database.
- Note: By default, SQLite on Android does not have a management interface or an application to create and manage databases from, so we're going to create the database ourselves by code.
- SQLite is available on every Android device. Using an SQLite database in Android. does not require any database setup or administration.
- You only have to define the SQL statements for creating and updating the database. Afterwards the database is automatically managed for you by the Android platform.
- First, we will create a class that handles all the operations required to deal with the database such as creating the database, creating tables, inserting and deleting records and so on.

Main classes to create SQLite database:-**SQLiteOpenHelper:**

To create and upgrade a database in your Android application we usually subclass SQLite Open Helper.

SQLite Database:

SQLite Database is the base class for working with a SQLite database in Android and provides methods to open, query, update and close the database.

Android

If you attempt to develop Android apps in Android Studio without first having a good understanding of Java, you will be frustrated. Java is an extremely useful and popular programming language for many reasons. Perhaps the most important reason for Java's popularity is that Java is memory managed. Memory managed means that the programmer does not need to be concerned with de-allocating memory off the heap, nor with worrying about memory leaks. Programmers developing in a memory-managed environment tend to be more productive, and their programs tend to have fewer runtime errors. Like Java, Android is a memory-managed programming environment. Managing memory turns out to be such a good idea that both Microsoft and Apple have adopted this model for their mobile development platforms.^{ix}

Switching from ADT/Eclipse

If you are an experienced Android developer and are used to programming with ADT, you are in for a pleasant surprise. Thankfully, all the SDK tools such as DDMS and Hierarchy Viewer are still available, and you will find them easily accessible from within Android Studio. If you're an ADT user, you probably find yourself continuously cleaning and rebuilding your projects in order to synchronize your resources with your source code (the dreaded R.java synchronization error). In the months that we have been using Android Studio, we have never been troubled with this problem. If you're an experienced ADT user, then in order to get up to speed with Android Studio, you will need to learn a few keyboard shortcuts, familiarize yourself with Gradle, and reorient yourself to Android Studio's presentation logic. Altogether, this is a small price to pay for the power and pleasure of Android Studio.

Installing the Java Development Kit on Windows

This section pertains to Windows users. If you're a Mac user, skip ahead to the section titled "Installing the Java Development Kit on Mac." Android Studio uses the Java tool chain to build, so you need to make sure that you have the Java Development Kit (JDK) installed on your computer before you start using Android Studio.

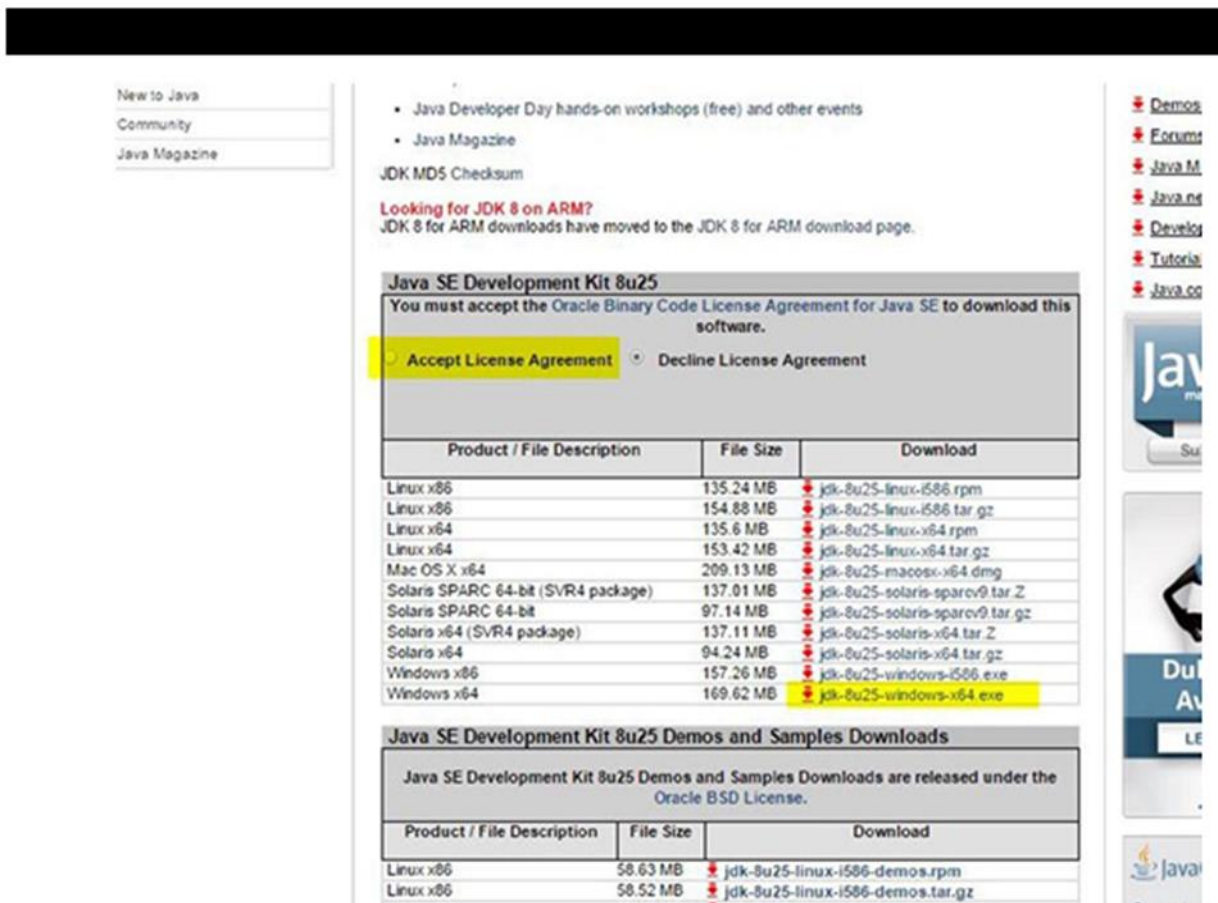
It's quite possible that you already have the JDK installed on your computer, particularly if you're a seasoned Android or Java developer. If you already have the JDK installed on your computer, and you're running JDK version 1.6 or higher, then you can skip this section. However, you may want to download, install, and configure the latest JDK anyway. You can download the JDK from the following Oracle site:

www.oracle.com/technetwork/java/javase/downloads/index.html



Downloading the JDK on Windows

The next step in the installation, shown in Figure 1-2, requires that you accept a license agreement by clicking the Accept License Agreement radio button. Then you must choose the appropriate JDK for your operating system. If you're running Windows 7 or Windows 8, you should click the file link to the right of the Windows x64 label, also shown in Figure 1-2. Oracle makes frequent release updates to the JDK. By the time this book goes to press, a newer version of the JDK will almost certainly be available, so please be sure to download the latest version. Wait for the installation file to download. This file is usually around 125MB, so the download shouldn't take long.



Executing the JDK Wizard on Windows

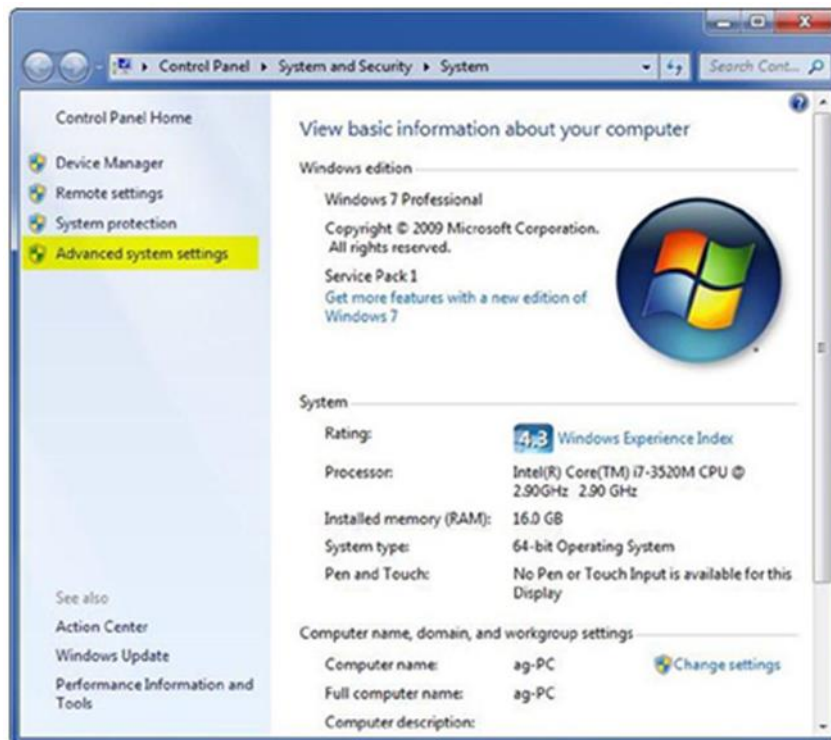
Before you install the JDK, create a directory in the root of your C: drive called Java. The name of this directory is arbitrary, though we call it Java because many of the tools we are going to install here are related to Java, including the JDK, Android Studio, and the Android SDK. Consistently installing the tools related to Android Studio in the C:\Java directory also keeps your development environment organized.

Navigate to the location where your browser downloaded the installation file and execute that file by double-clicking it. Once the installation begins, you will be presented with the Installation Wizard, shown in Figure 1-3. In Windows, the JDK installer defaults to C:\Program Files\Java\. To change the installation directory location, click the Change button. We recommend installing your JDK in the C:\Java directory because it contains no spaces in the path name and it's easy to remember. See Figure 1-4.

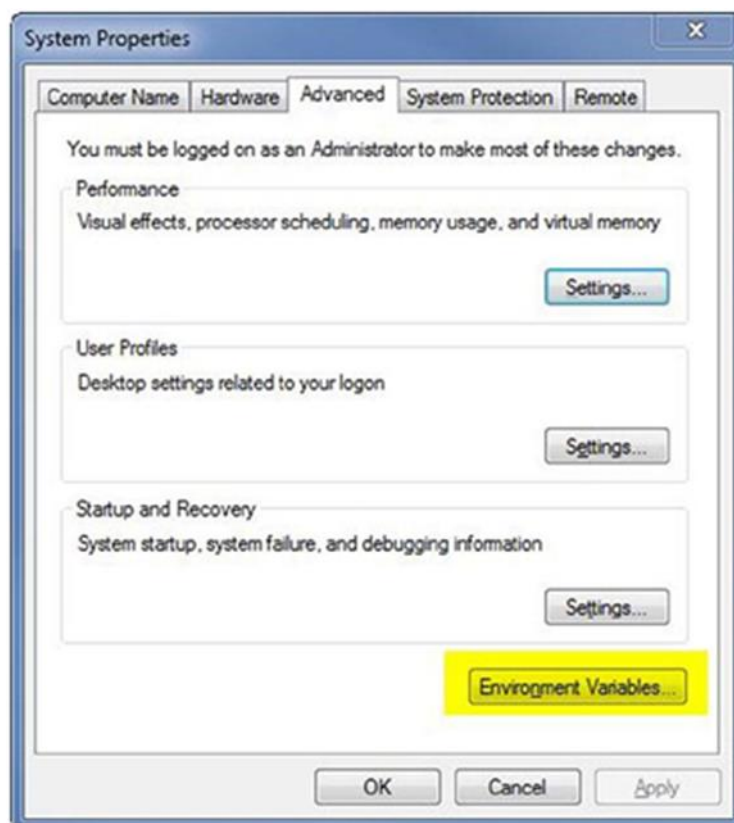


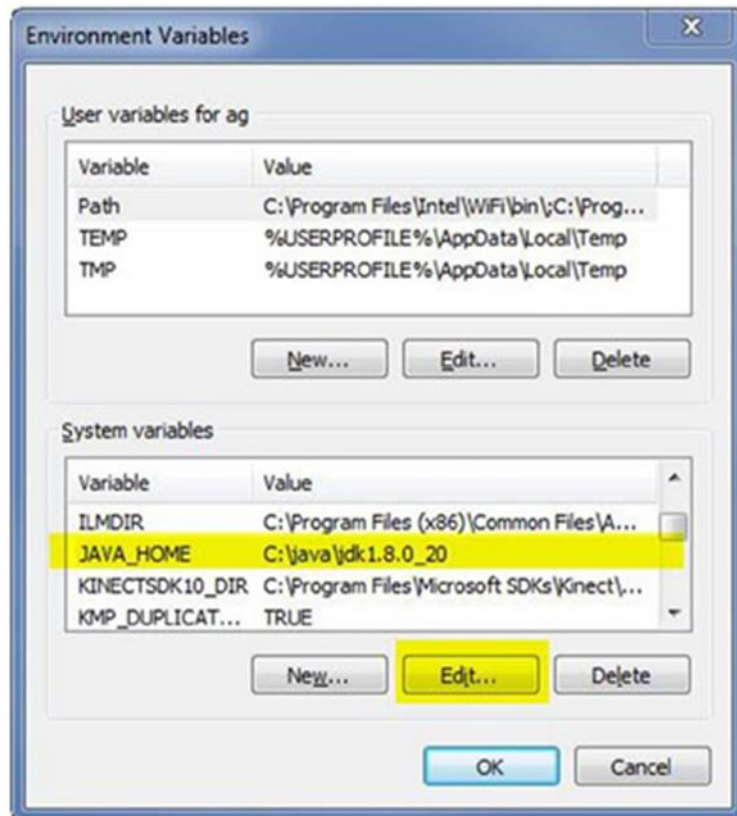
Configuring Environmental Variables on Windows

This section shows you how to configure Windows so that the JDK is found by Android Studio. On a computer running Windows, hold down the Windows key and press the Pause key to open the System window. Click the Advanced System Settings option, shown in Figure 1-5.



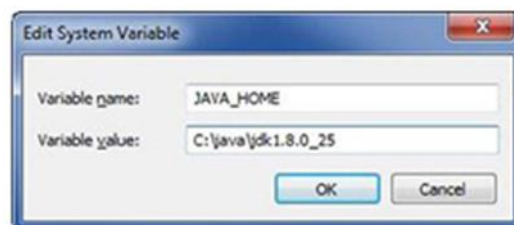
Click the Environmental Variables button, shown in Figure 1-6. In the System Variables list along the bottom, shown in Figure 1-7, navigate to the JAVA_HOME item. If the JAVA_HOME item does not exist, click New to create it. Otherwise, click Edit.





Clicking either New or Edit displays a dialog box similar to Figure 1-8. Be sure to type

JAVA_HOME in the Variable Name field. In the Variable Value field, type the location where you installed the JDK earlier (less any trailing slashes), as shown in Figure 1-4. Now click OK.



Just as you did with the JAVA_HOME environmental variable, you will need to edit the PATH environmental variable. See Figure 1-9. Place your cursor at the end of the Variable Value field and type the following:

%JAVA_HOME%\bin



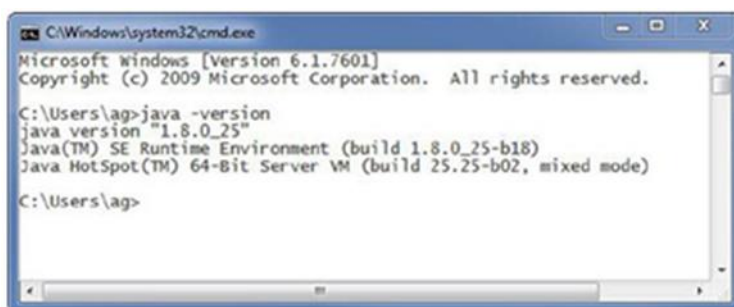
Now click OK, OK, OK to accept these changes and back out of the system properties.

To test that the new JDK has been installed properly, pull up a command line by clicking the Start button, typing **cmd**, and then pressing Enter.

In the command-line window, issue the following command and press Enter:

```
java -version
```

If you get a response like the one shown in Figure 1-10, congratulations. You just installed the JDK properly.



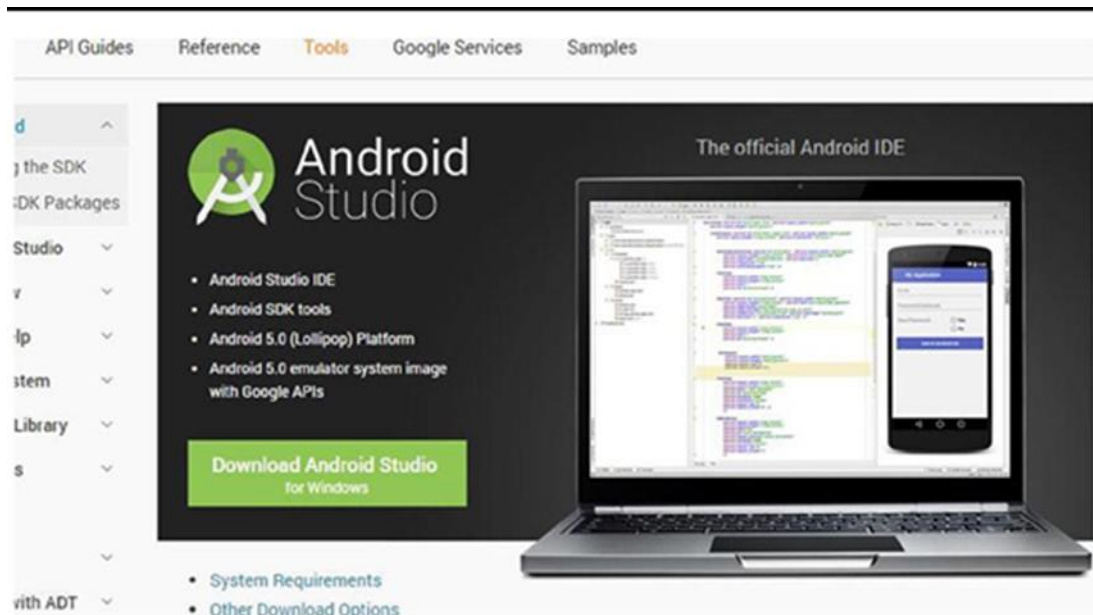
Installing Android Studio

Before you begin downloading Android Studio, create a labs parent directory for the labs you will create in this book. We use C:\androidBook\ as our labs' parent directory throughout the book, but you may choose or create whatever directory you see fit. For that reason, we simply call it the labs parent directory.

Downloading Android Studio is straightforward. Point your browser to this site:

developer.android.com/sdk/installing/studio.html

Now click the large green Download Android Studio for your OS button, shown in Figure 1-16. Next, select the check box labeled I Have Read and Agree with the Above Terms and Conditions. Click Download Android Studio for your OS again, and your installation file should begin downloading. Once the download is complete, execute the file you just downloaded.



After the Installation Wizard begins, move through its screens by clicking the Next buttons until you reach the Choose Components screen. There, select all the component check boxes, shown in Figure 1-17. Then click next. Agree to the terms and conditions once again. When you reach the Configuration Settings: Install Locations screen, shown in Figure 1-18, select the locations for Android Studio and the Android SDK. To be consistent, we chose to install Android Studio in C:\Java\astudio\ and the Android SDK in C:\Java\asdk\.

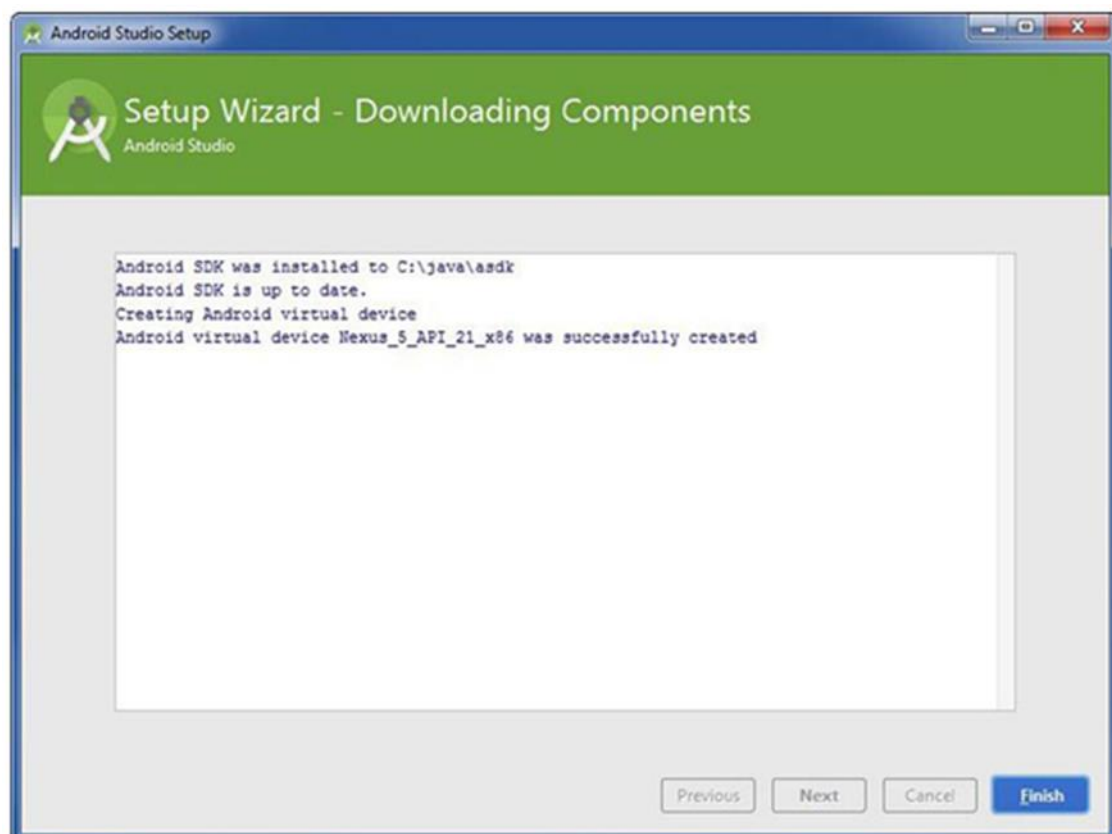




Click through several Next buttons as you install both Android Studio and the Android SDK. You should eventually arrive at the Completing the Android Studio Setup screen, shown in Figure 1-19. The Start Android Studio check box enables Android Studio to launch after you click Finish. Make sure the check box is selected, and then go ahead and click Finish, and Android Studio will launch. Please note that from here on out, you will need to navigate to either the desktop icon or the Start menu to launch Android Studio.

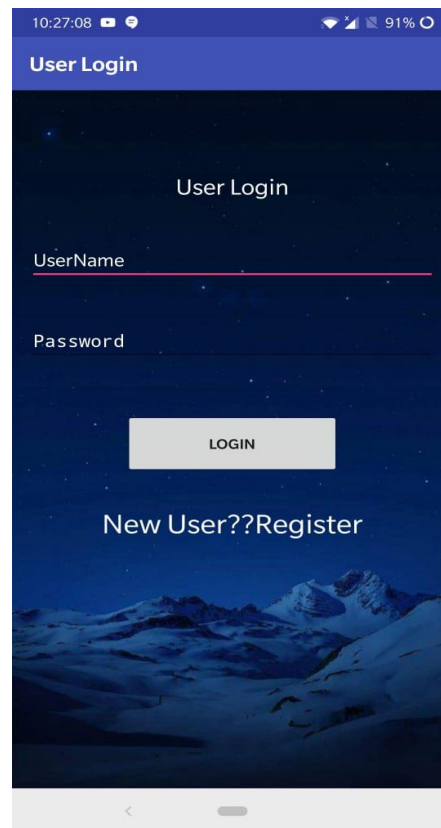


When Android Studio starts for the very first time, the Setup Wizard, shown in Figure 1-20, will analyze your system looking for an existing JDK (such as the one you installed earlier), as well as the location of the Android SDK. The Setup Wizard should download everything you need to begin developing apps in Android Studio. Click the Finish button to dismiss the Setup Wizard.



Usage of our application

Click on user



10:27:08 91%

User Login

User Login

UserName

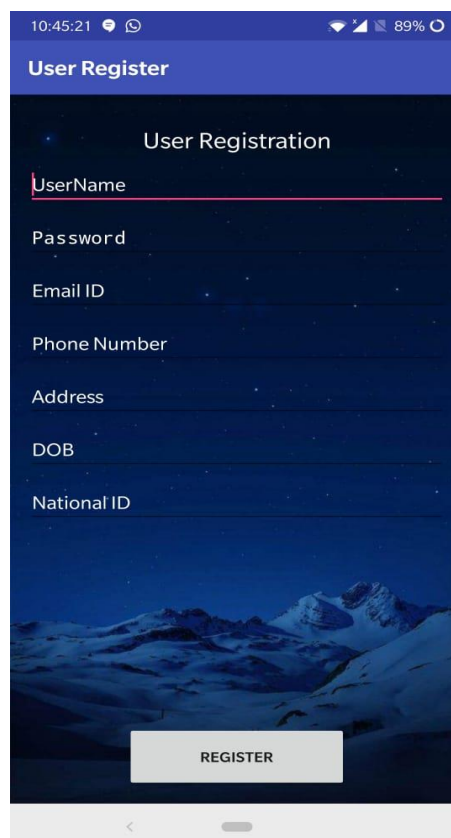
Password

LOGIN

New User??Register

The screenshot shows a mobile application interface for user login. At the top, there is a status bar with the time 10:27:08 and battery level at 91%. Below this is a blue header with the text 'User Login'. The main content area has a dark blue background with a subtle star pattern. It features a title 'User Login' in white, followed by two input fields labeled 'UserName' and 'Password'. A white 'LOGIN' button is centered below the fields. At the bottom, there is a link 'New User??Register' in white. The background image shows a snowy mountain range under a dark sky.

Before login into user we should register



10:45:21 89%

User Register

User Registration

UserName

Password

Email ID

Phone Number

Address

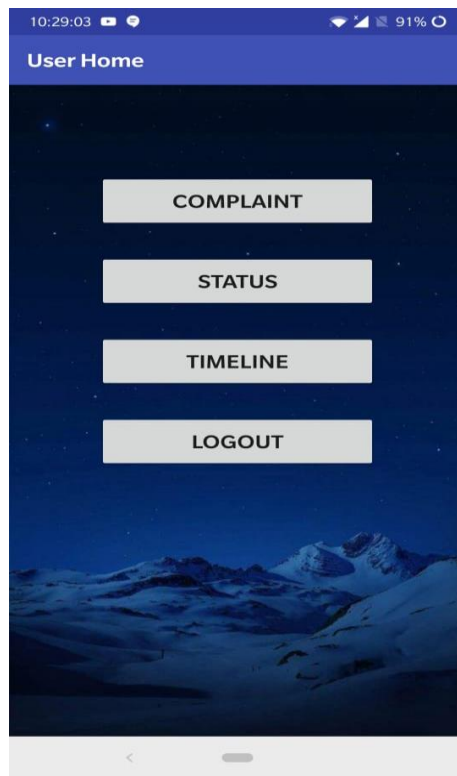
DOB

National ID

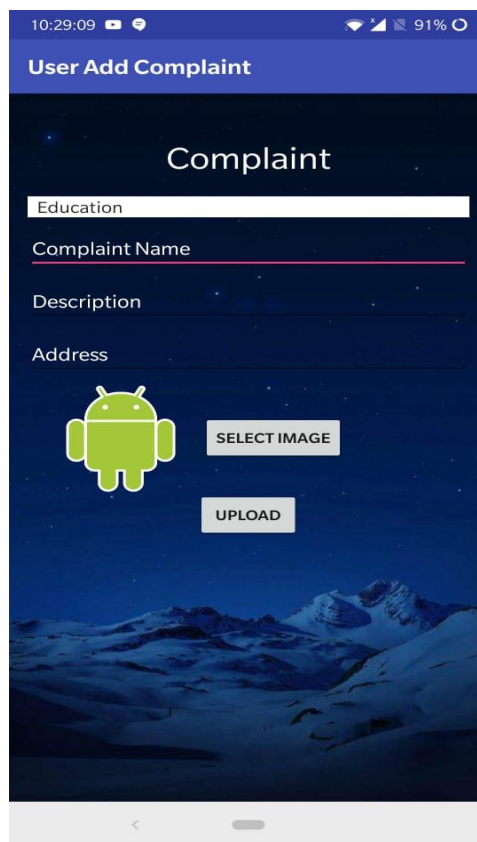
REGISTER

The screenshot shows a mobile application interface for user registration. At the top, there is a status bar with the time 10:45:21 and battery level at 89%. Below this is a blue header with the text 'User Register'. The main content area has a dark blue background with a subtle star pattern. It features a title 'User Registration' in white, followed by seven input fields labeled 'UserName', 'Password', 'Email ID', 'Phone Number', 'Address', 'DOB', and 'National ID'. A white 'REGISTER' button is centered at the bottom. The background image shows a snowy mountain range under a dark sky.

After logging into the user

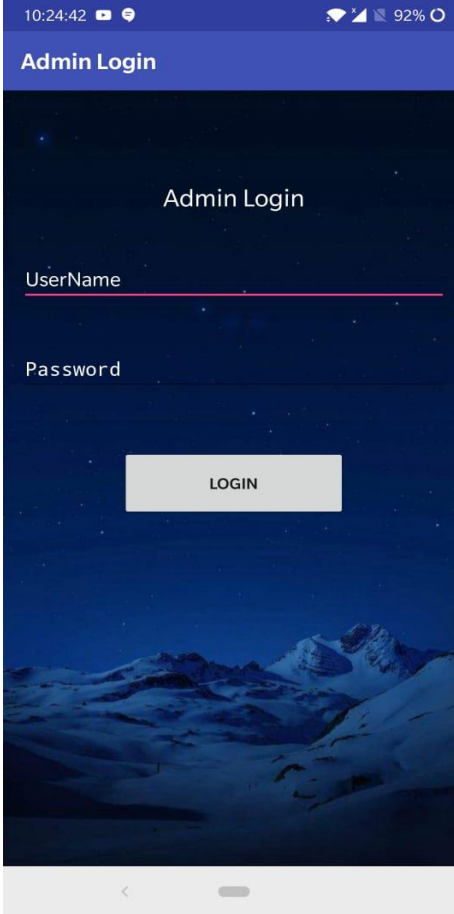


For the registration of the complaint, click ON complaint



Fill the details of the complaint, after filling click ON upload for the complaint registration

Logging into the admin



10:24:42 92%

Admin Login

Admin Login

UserName

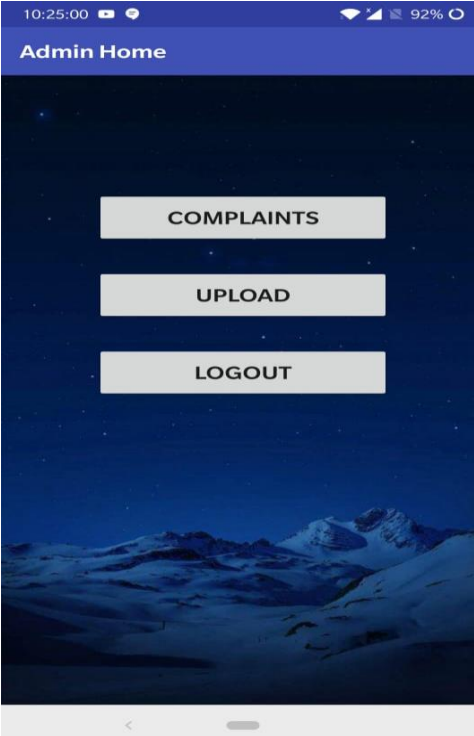
Password

LOGIN

The screenshot shows a mobile application interface for admin login. At the top, a status bar displays the time 10:24:42 and battery level 92%. Below this is a blue header bar with the text 'Admin Login'. The main area has a dark blue background with a faint image of snow-capped mountains. It contains the title 'Admin Login', two input fields labeled 'UserName' and 'Password', and a grey 'LOGIN' button. A white navigation bar at the bottom shows a back arrow and a home indicator.

For the logging into the admin give username as admin and password as admin

After logging into the admin



10:25:00 92%

Admin Home

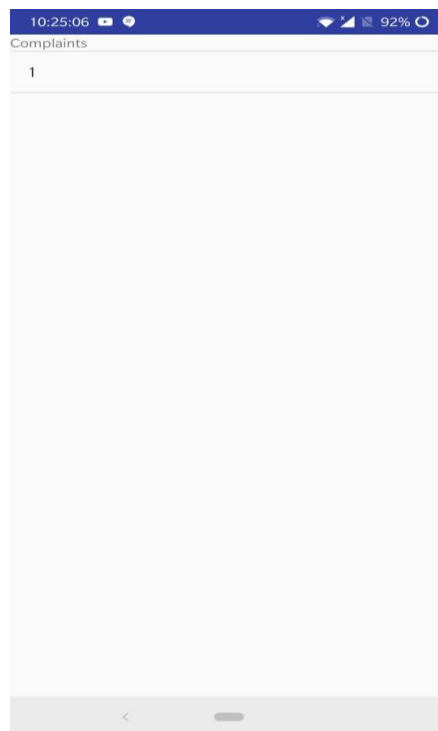
COMPLAINTS

UPLOAD

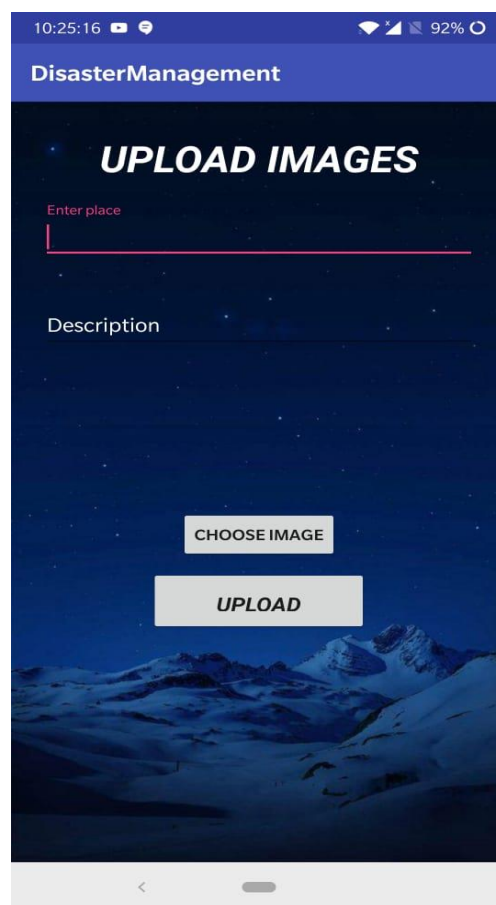
LOGOUT

The screenshot shows the 'Admin Home' screen. The status bar at the top shows the time 10:25:00 and battery level 92%. A blue header bar contains the text 'Admin Home'. The main area has the same dark blue background with mountain imagery. It features three stacked grey buttons labeled 'COMPLAINTS', 'UPLOAD', and 'LOGOUT'. A white navigation bar at the bottom shows a back arrow and a home indicator.

Click ON complaints, to see how many complaints have been approached

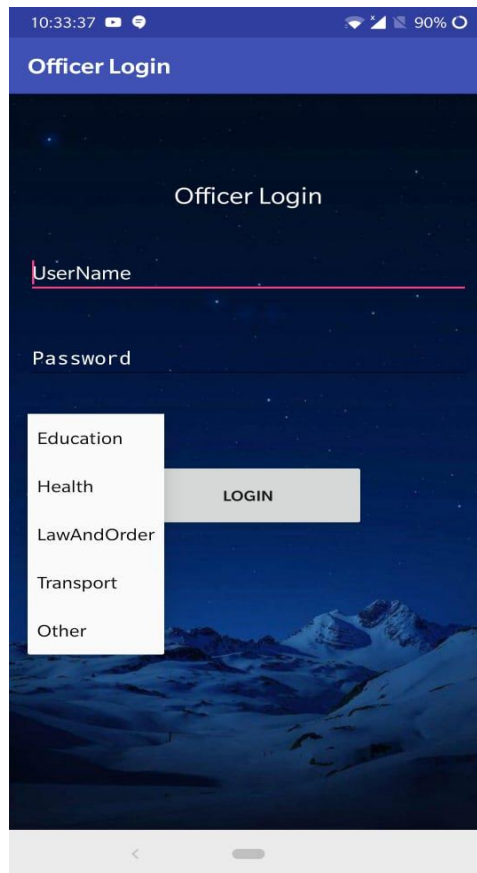


Click ON upload, for uploading alerts for the user and the newsfeed to the user



Click ON logout, for logging out from admin

Click ON officer

A mobile application screenshot of the 'Officer Login' screen. The title bar is blue with 'Officer Login' in white. The background is a dark blue night sky with a mountain range at the bottom. The screen contains two input fields: 'UserName' and 'Password'. A dropdown menu is open below the 'Password' field, showing a list of departments: Education, Health, LawAndOrder, Transport, and Other. A grey 'LOGIN' button is positioned to the right of the dropdown menu. The status bar at the top shows the time 10:33:37, signal strength, and 90% battery.

For logging into the officer, select the particular department and give the username and password as selected department example username: education, password: education

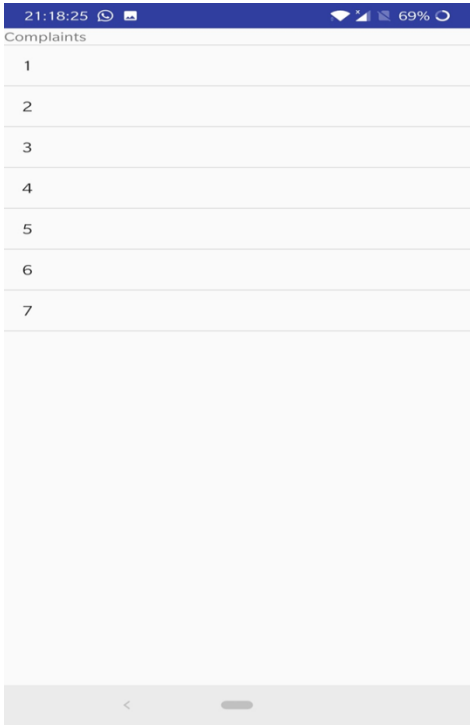
After logging into the officer

A mobile application screenshot of the 'Education Home' screen. The title bar is blue with 'Education Home' in white. The background is the same dark blue night sky with a mountain range. The screen displays three large, grey, rectangular buttons stacked vertically: 'COMPLAINTS', 'APPROVED', and 'LOGOUT'. The status bar at the top shows the time 10:34:21, signal strength, and 90% battery.

Click ON complaints, for viewing how many complaints have been registered for education



Click ON that complaint, for accepting or rejecting the complaint
Click ON approved, accepted complaints will be shown here



Approved complaint can be viewed like this from admin

21:18:30 69%

Admin Complaint Details

Complaint Details

User Name	Amruth
Department	Education
Name	y
Address	g
Status	Waiting
Description	g
Registered Time	2019-08-13 16:08:02
Approval Time	Waiting
Finished Time	Waiting
Task Duration(in Min.)	
Feedback	

Rating Waiting

6. Feasibility

Feasibility Study

Preliminary investigation examines project feasibility; the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All systems are feasible if they are given unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

- Technical Feasibility
- Operation Feasibility
- Economic Feasibility

Technical Feasibility:

The technical issue usually raised during the feasibility stage of the investigation includes the following:

- Does the necessary technology exist to do what is suggested?
- Do the proposed equipment's have the technical capacity to hold the data required to use the new system?
- Will the proposed system provide adequate response to inquiries, regardless of the number or location of users?
- Can the system be upgraded if developed?

Are there technical guarantees of accuracy, reliability, ease of access and data security?

Operation Feasibility:

User-friendly

Customer will use the forms for their various transactions i.e. for adding new routes, viewing the routes details. Also the Customer wants the reports to view the various transactions based on the constraints. These forms and reports are generated as user-friendly to the Client.

Reliability

The package will pick-up current transactions on line. Regarding the old transactions, User will enter them in to the system.

Security

The web server and database server should be protected from hacking, virus etc.

Portability

The application will be developed using standard open source software (Except Oracle) like Java, tomcat web server, Internet Explorer Browser etc. these software will work both on Windows and Linux o/s. Hence portability problems will not arise.

Availability

This software will be available always.

Maintainability

The system called the ewheelz uses the 2-tier architecture. The 1st tier is the GUI, which is said to be front-end and the 2nd tier is the database, which uses My-Sql, which is the back-end.

The front-end can be run on different systems (clients). The database will be running at the server. Users access these forms by using the user-ids and the passwords.

Economic Feasibility

The computerized system takes care of the present existing system's data flow and procedures completely and should generate all the reports of the manual system besides a host of other management reports.

It should be built as a web based application with separate web server and database server. This is required as the activities are spread throughout the organization customer wants a centralized database. Further some of the linked transactions take place in different locations.

Open source software like TOMCAT, JAVA, MySQL and Linux is used to minimize the cost for the Customer.

7. Conclusions and Future scope

- e- Public problems android application overcomes the problem of installing of many applications. By this one application we can complaint to any of the Government services.
- The department related to that complaint can View it and reply to that complaint.
- This android application makes it easy for the users to access all sort of problems in every department.
- In today's busy world nobody has the time to call or go to an organization and file their complaint.
- Even people want to help the needy they don't find time. So they can install this app and solve their problems at any time.
- e- Public problems app contains a select option for the users to choose any one based on their complaint.
- Public can enter into app and select the field they need and post their complaint over there.
- Complaint reaches to respective department and their problem can be solved as soon as possible without any inconvenience to them.

Future scope:

We can extend this application by adding more departments in near future. We can also add Camera options to click photos and upload as a proof.

8. Bibliography

References for the project development were taken from the following books and websites.

- [1] *"Android Programming for Beginners"* by John Horton
- [2] *Web technologies- part 1-* SubbaRaju
- [3] *Core java and integrated approach-* Dr. R. Nageshwar Rao
- [4] www.developers.android.com
- [5] B.Y. Ricardo and R.N. Berthier, *Modern Information Retrieval*. Addison Wesley Longman, 1999
- [6] Y. Yang And J.O. Pedersen, "A Comparative Study On Feature Selection In Text Categorization," *Proc.14th Int'l Conf. Machine Learning*, Pp. 412-420, 1997.
- [7] D.D. Lewis, "Feature Selection and Feature Extraction For Text Categorization," *Proc. Workshop Speech and Natural Language*, Pp. 212-217, 1992
- [8] E.F. Combarro, E. MontanˆE'S, I. Di'Az, J. Ranilla, And R. Mones , "Introducing A Family Of Linear Measures For Feature Selection In Text Categorization," *Ieee Trans. Knowledge And Data Eng.*, Vol. 17, No. 9, Pp. 1223-1232, Sept. 2005.
- [9] www.w3schools.com
- [10] *UML 2 Toolkit*, by Hans-Erik Eriksson, Magnus Penker, Brian Lyons, David Fado:
WILEY'-Dreamtech India Pvt. Lid
- [11] www.androidtools.com

