**INTRODUCTION**

## Synopsis of the Project

**DATA VAULT**

## Proposed System

## Security is an essential aspect of the success of any technology. The high

## Interoperability of the popular collection of standards must be integrated with

## appropriate mechanisms to protect data.

## The proposed system focuses on protection of the data transferred between two devices using AES encryption algorithm and trying to secure the transferred data from various possible attacks such as eavesdropping, man in the middle attack, data modification, relay attack.

## Objective

The objective of this paper is to design and implement an application that encrypts and decrypts

Plain-text files using AES algorithm and utilizes digital signature technique to verify the integrity and authenticity of the message sent. The objective of this paper is to design and implement an application that encrypts and decrypts files using AES algorithm.

## Problem Definition

The underlying respective architecture of most IT systems, including the desktop computer and internet, does not guarantee security. Users with malicious intents have always found a way of exploiting one vulnerability or the other. An attack that affects the confidentiality of information often presents the platform for the integrity of such information to be compromised.

Intercepted information on transit would make little or no sense to an interceptor if he is not able to decipher the content of the information. This explains why it is very necessary to ensure that even when an intruder or un-authorized user successfully obtains access to some information the confidentiality and integrity of the information remain uncompromised.

## Theoretical Background

**Overview of Front End**

**Android**



Fig 1.2 Android Logo

Androidis an operating system for mobile devices such as smart phones and tablet computers. It is developed by the Open Handset Alliance led by Google.. Google released most of the Android code under the Apache License, a free software license. The Android Open Source Project (AOSP) is tasked with the maintenance and further development of Android. Android consists of a kernel based on the Linux kernel, with middleware, libraries and APIs written in C and application software running on an application framework which includes Java-compatible libraries based on Apache Harmony. Android uses the Dalvik virtual machine with just-in-time compilation to run compiled Java code. Android has a large community of developers writing applications (―apps‖) that extend the functionality of the devices. Developers write primarily in a customized version of Java. Apps can be downloaded from third-party sites or through online stores such as Android Market, the app store run by Google.

## ASP.NET



Fig 1.3 ASP.NET Logo

ASP.NET is more than the next version of Active Server Pages (ASP); it is a unified Web development platform that provides the services necessary for developers to build enterprise-class Web applications. While ASP.NET is largely syntax-compatible with ASP, it also provides a new programming model and infrastructure that enables a powerful new class of applications. You can migrate your existing ASP applications by incrementally adding ASP.NET functionality to them. ASP.NET is a compiled .NET Framework –based environment. You can author applications in any .NET Framework compatible language, including Visual Basic and Visual C#. Additionally, the entire .NET Framework platform is available to any ASP.NET application. Developers can easily access the benefits of the .NET Framework, which include a fully managed, protected, and feature-rich application execution environment, simplified development and deployment, and seamless integration with a wide variety of languages.

## Windows Communication Foundation (WCF)



Fig 1.4 WCF Logo

Windows Communication Foundation (WCF) is a framework for building service- oriented applications. Using WCF, you can send data as asynchronous messages from one service endpoint to another. A service endpoint can be part of a continuously available service hosted by IIS, or it can be a service hosted in an application. An endpoint can be a client of a service that requests data from a service endpoint. The messages can be as simple as a single character or word sent as XML, or as complex as a stream of binary data.

While creating such applications was possible prior to the existence of WCF, WCF makes the development of endpoints easier than ever. In summary, WCF is designed to offer a manageable approach to creating Web services and Web service clients.

#### Features of WCF

WCF includes the following set of features.

* + - * Service Orientation
      * Service Metadata
      * Data Contracts
      * AJAX and REST Support
      * Extensibility

## Overview of Back End

**SQL**

SQL tutorial provides basic and advanced concepts of SQL. Our SQL tutorial is designed for beginners and professionals.SQL (Structured Query Language) is used to perform operations on the records stored in the database such as updating records, deleting records, creating and modifying tables, views, etc.

SQL is just a query language; it is not a database. To perform SQL queries, you need to install any database, for example, Oracle, MySQL, MongoDB, PostGre SQL, SQL Server, DB2, etc.

SQL stands for Structured Query Language.It is designed for managing data in a relational database management system (RDBMS).It is pronounced as S-Q-L or sometime See-Qwell.SQL is a database language, it is used for database creation, deletion, fetching rows, and modifying rows, etc.SQL is based on relational algebra and tuple relational calculus.All DBMS like MySQL, Oracle, MS Access, Sybase, Informix, Postgres, and SQL Server use SQL as standard database language.

SQL is required:

* + - * To create new databases, tables and views
      * To insert records in a database
      * To update records in a database
      * To delete records from a database
      * To retrieve data from a database

# SYSTEM ANALYSIS

## Feasibility Study

A feasibility study is an analysis that takes all of a project‘s relevant factors into account—including economic, technical, legal, and scheduling considerations—to ascertain the likelihood of completing the project successfully. Project managers use feasibility studies to discern the pros and cons of undertaking a project before they invest a lot of time and money into it. Feasibility studies also can provide a company‘s management with crucial information that could prevent the company from entering blindly into risky businesses.

A feasibility study is simply an assessment of the practicality of a proposed plan or project. As the name implies, these studies ask: ―Is this project feasible? Do we have the people, tools, technology, and resources necessary for this project to succeed?‖ Also, ―Will the project get us the return on investment (ROI) that we need and expect?‖

The goal of a feasibility study is to thoroughly understand all aspects of a project, concept, or plan; become aware of any potential problems that could occur while implementing the project; and determine if, after considering all significant factors, the project is viable—that is, worth undertaking.

Feasibility studies are important to business development. They can allow a business to address where and how it will operate; identify potential obstacles that may impede its operations, and recognize the amount of funding it will need to get the business up and running. Feasibility studies also can lead to marketing strategies that could help convince investors or banks that investing in a particular project or business is a wise choice.

## Technical Feasibility

In technical feasibility the following issues are taken into consideration.

* + - * Whether the required technology is available or not.
      * Whether the required resources are available.
        + Manpower- programmers, testers & debuggers.
        + Software and hardware

Once the technical feasibility is established, it is important to consider the monetary factors also. Since it might happen that developing a particular system may be technically possible but it may require huge investments and benefits may be less. For evaluating this, economic feasibility of the proposed system is carried out.

## Operational Feasibility

Operational feasibility is mainly concerned with issues like whether the system will be used if it is developed and implemented. Whether there will be resistance from users that will affect the possible application benefits? The essential questions that help in testing the operational feasibility of a system are following.

* + - * Does management support the project?
      * Are the users not happy with current business practices? Will it reduce the time (operation) considerably? If yes, then they will welcome the change and the new system.
      * Have the users been involved in the planning and development of the project? Early involvement reduces the probability of resistance towards the new system.
      * Will the proposed system really benefit the organization? Does the overall response increase? Will accessibility of information be lost? Will the system effect the customers in considerable way?

## System Planning and Schedule

**GANTT Chart**

Fig 2.1 GANTT chart

## PERT Table

Table 2.1 PERT Table

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Activity** | **Duration (Days)** | **Start**  **Date** | **Immediate Predecessor**  **Activities** | **Immediate Predecessor**  **Activities** |  |  |
| Requirement Analysis (A) | 10 | 15-Feb-20 |  |  |  |  |
| Design (B) | 15 | 01-March-20 | A | A |  |  |
| Coding and Debugging I | 40 | 14-March-20 | B | B |  |  |
| Testing and Implementation (D) | 10 | 06-April-20 | B and C | B and C |  |  |
| Documentation I | 7 | 20-April-20 | D | D |  |  |

## PERT Chart

Start

A

B

C

D

E

Fig 2.2 PERT Chart

## Time Line

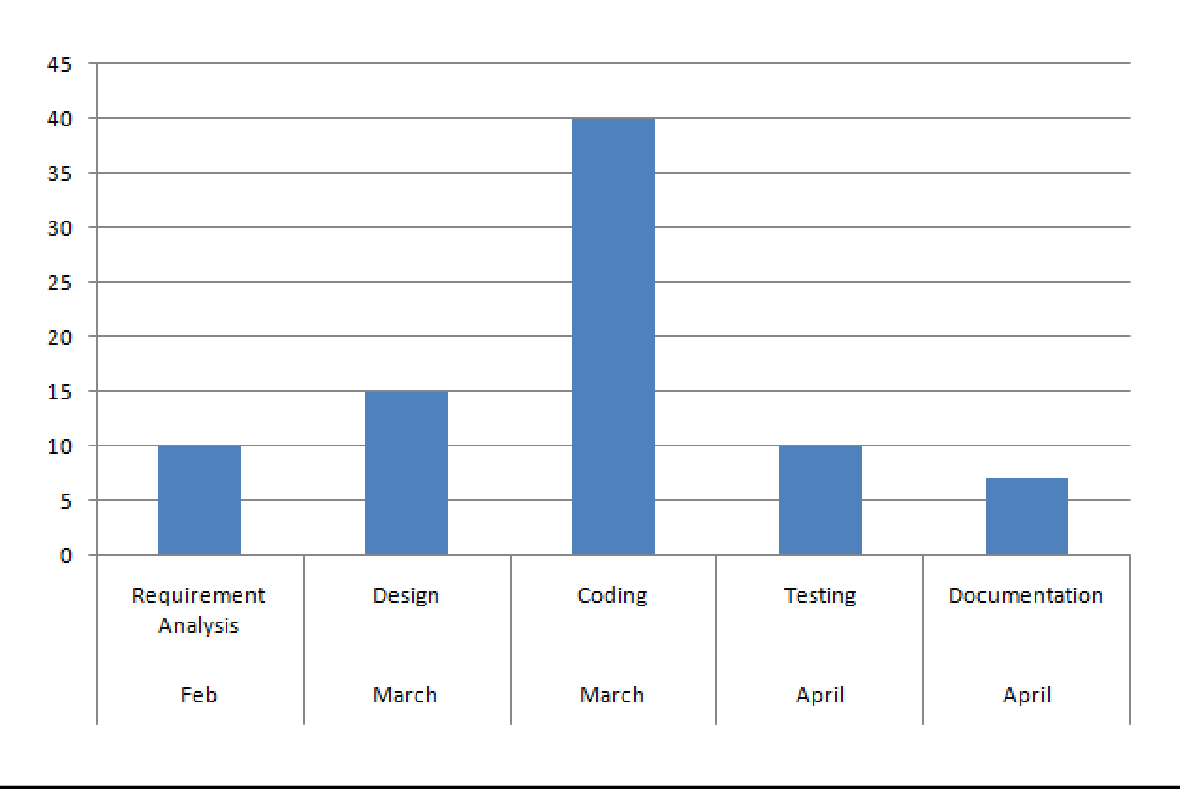


Fig 2.3 Timeline Chart

## Cause Effect Diagram

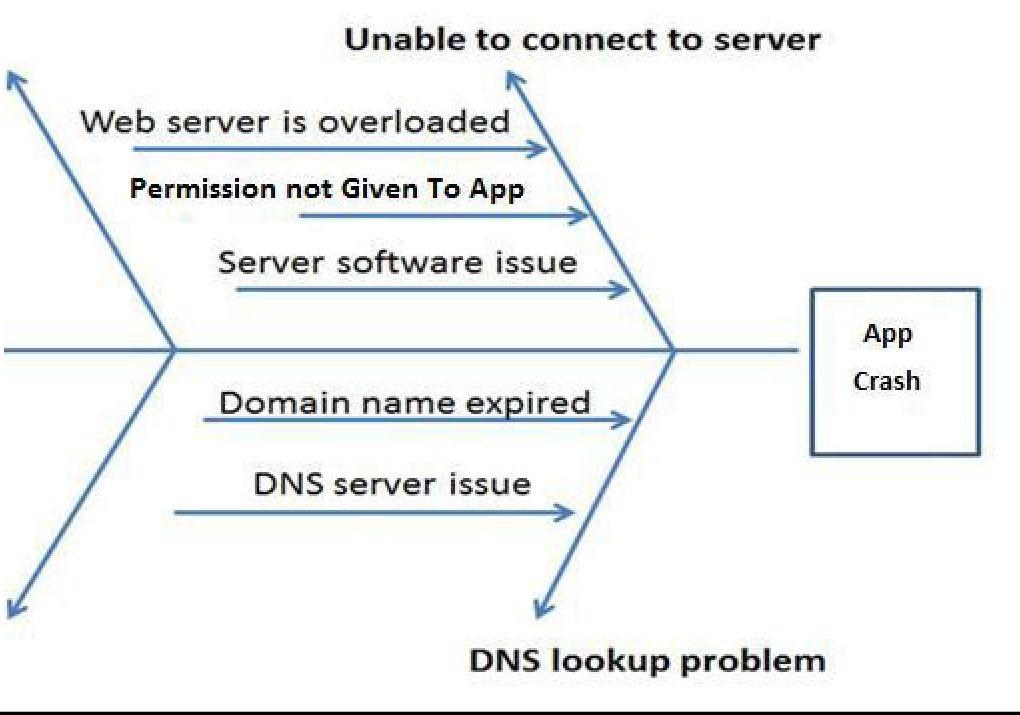


Fig 2.4 Cause Effect Diagram

# SYSTEM DESIGN

## Software Requirement Specification

A **Software Requirements Specification (SRS)** (also known as a System Requirements Specification) is a document or set of documentation that describes the features and behavior of a system or software application.

## Introduction

A good SRS defines the how Software System will interact with all internal modules, hardware, communication with other programs and human user interactions with wide range of real life scenarios. Using the Software requirements specification (SRS) document on QA lead, managers creates test plan. It is very important that testers must be cleared with every detail specified in this document in order to avoid faults in test cases and its expected results.

It is highly recommended to review or test SRS documents before start writing test cases and making any plan for testing. Most of the defects which we find during testing are because of either incomplete requirements or ambiguity in SRS. To avoid such defects it is very important to test software requirements specificationbefore writing the test cases.

## Selection of Technology/Specific Requirements

**Hardware to be used**

* + - * + Desktop / Laptop

Windows 7 or higher

Minimum 4 Gb RAM

Minimum 10 Gb Memory

* + - * + Smartphone

Android OS Jelly Bean(API level 16) or higher

Minimum 2 Gb RAM

10 Mb Storage

## Software to be used

**Android Studio**



Fig 3.1 Android Studio

Android Studio is the official integrated development environment (IDE) for Google‘s Android operating system, built on JetBrains IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems. It is a replacement for the Eclipse Android Development Tools (ADT) as the primary IDE for native Android application development.

Android Studio was announced on May 16, 2013 at the Google I/O conference. It was in early access preview stage starting from version 0.1 in May 2013, then entered beta stage starting from version 0.8 which was released in June 2014. The first stable build was released in December 2014, starting from version 1.0. Android Studio‘s latest stableversion is 3.4.1 released on May 15, 2019.

Since 7 May 2019, Kotlin is Google‘s preferred language for Android app development. Still, other languages are supported, including by Android Studio.

The following features are provided in the current stable version:

* + - * + Gradle-based build support
        + Android-specific refactoring and quick fixes
        + Lint tools to catch performance, usability, version compatibility and other problems
        + ProGuard integration and app-signing capabilities.
        + Template-based wizards to create common Android designs and components.
        + A rich layout editor that allows users to drag-and-drop UI components, option to preview layouts on multiple screen configurations.
        + Support for building Android Wear apps.
        + Built-in support for Google Cloud Platform, enabling integration with Firebase Cloud Messaging (Earlier ‗Google Cloud Messaging‘) and Google App Engine
        + Android Virtual Device (Emulator) to run and debug apps in the Android studio.

Android Studio supports all the same programming languages of IntelliJ (and Clion) Java, C++, and more with extensions, such as Go; and Android Studio 3.0 or later supports Kotlin and ―all Java 7 language features and a subset of Java 8 language features that vary by platform version.‖External projects backport some Java 9 features.While IntelliJ that Android Studio is built on supports all released Java versions, and Java 12, it‘s not clear to what level Android Studio supports Java versions up to Java 12 (the documentation mentions partial Java 8 support). At least some new language features up to Java 12 are usable in Android.

## Visual Studio



Fig 3.2 Visual Studio

The Visual Studio integrated development environment is a creative launching pad that you can use to edit, debug, and build code, and then publish an app. An integrated development environment (IDE) is a feature-rich program that can be used for many aspects of software development. Over and above the standard editor and debugger that most IDEs provide, Visual Studio includes compilers, code completion tools, graphical designers, and many more features to ease the software development process.

Visual Studio with an open project and several key tool windows you‘ll likely use:

* + - Solution Explorer (top right) lets you view, navigate, and manage your code files. **Solution Explorer** can help organize your code by grouping the files into solutions and projects.
    - The editor window (center), where you‘ll likely spend a majority of your time, displays file contents. This is where you can edit code or design a user interface such as a window with buttons and text boxes.
    - Team Explorer (bottom right) lets you track work items and share code with others using version control technologies such as Git and Team Foundation Version Control (TFVC).

#### Popular productivity features

* + - * Squiggles and Quick Actions
      * Code Cleanup
      * Refactoring
      * IntelliSense
      * Search box
      * Live Share
      * Call Hierarchy
      * CodeLens
      * Go To Definition
      * Peek Definition

## SQL Server Management Studio (SSMS)

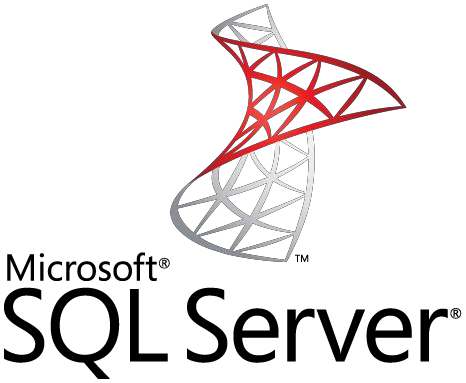


Fig 3.3 SQL Server

SQL Server Management Studio (SSMS) is an integrated environment for managing any SQL infrastructure. Use SSMS to access, configure, manage, administer, and develop all components of SQL Server, Azure SQL Database, and SQL Data Warehouse. SSMS provides a single comprehensive utility that combines a broad group of graphical tools with a number of rich script editors to provide access to SQL Server for developers and database administrators of all skill levels.

SQL Server Management Studio (SSMS) is an integrated environment for managing any SQL infrastructure, from SQL Server to Azure SQL Database. SSMS provides tools to configure, monitor, and administer instances of SQL Server and databases. Use SSMS to deploy, monitor, and upgrade the data-tier components used by your applications, as well as build queries and scripts.

Use SSMS to query, design, and manage your databases and data warehouses, wherever they are – on your local computer, or in the cloud.

## Postman



Fig 3.4 Postman

The Postman API has several endpoints to help you integrate Postman with your development toolchain.You can add new collections, update existing collections, update environments, or add and run monitors directly through the API. This API enables you to programmatically access your data stored in your Postman account with ease.The easiest way to start with the API is to click the **Run in Postman** button at the top of the Postman API documentation page and use the Postman app to send requests.

You need to know three important points about the Postman API:

1. You need a valid API Key to send requests to the API endpoints. Postman users can get a key in the Integrations page.
2. The API is rate limited.
3. Using the API, you can add and update collections, environments, and users. You can also run monitors, create a mock server, and so much more.

## Methodologies Adapted

**System Development Life Cycle:**

The System Development Life Cycle is the process of developing information systems through investigation, analysis, design, implementation, and maintenance. The System Development Life Cycle (SDLC) is also known as Information Systems Development or Application Development.

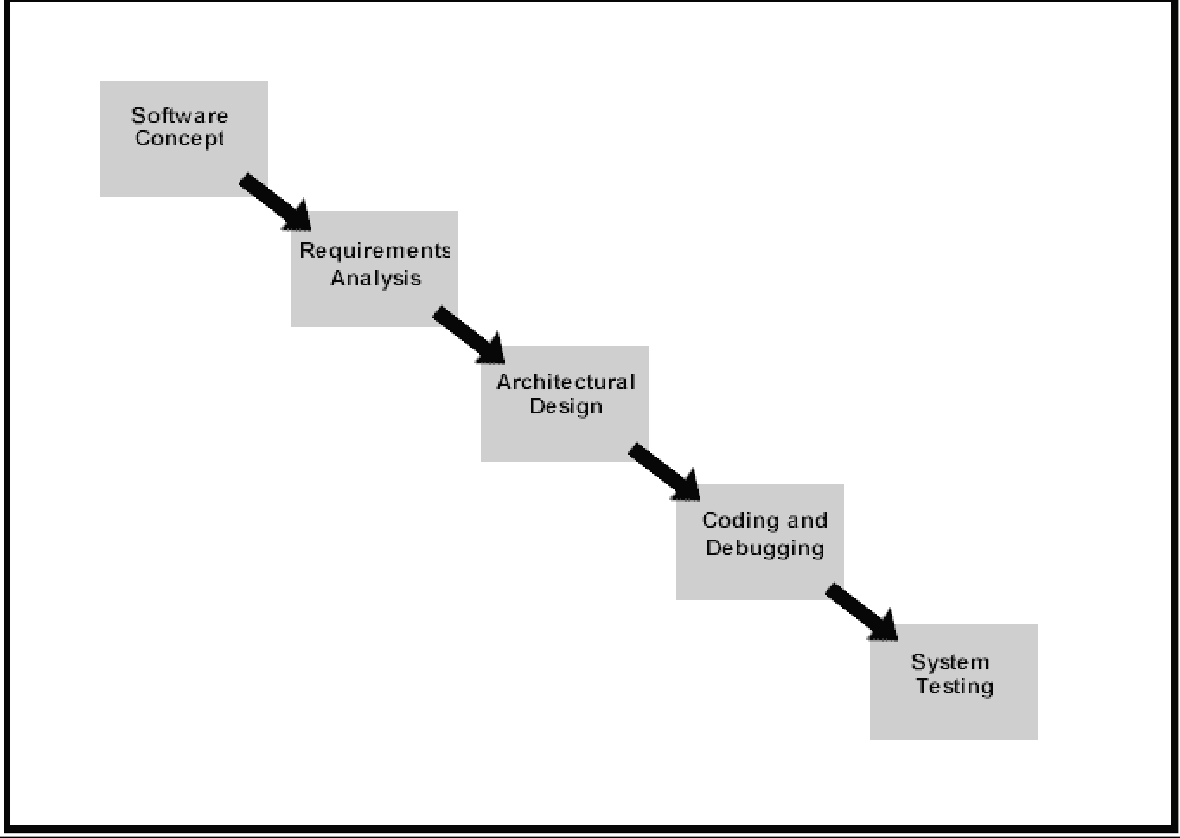


Fig 3.5 SDLC

Steps involved in the System Development Life Cycle :

Below are the steps involved in the System Development Life Cycle. Each phase within the overall cycle may be made up of several steps.

#### Step 1: Software Concept

The first step is to identify a need for the new system. This will include determining whether a business problem or opportunity exists, conducting a feasibility study to determine if the proposed solution is cost effective, and developing a project plan.

This process may involve end users who come up with an idea for improving their work. Ideally, the process occurs in tandem with a review of the organization‘s strategic plan to ensure that IT is being used to help the organization achieve its strategic objectives. Management may need to approve concept ideas before any money is budgeted for its development.

#### Step 2: Requirements Analysis

Requirements analysis is the process of 29nalysing the information needs of the end users, the organizational environment, and any system presently being used, developing the functional requirements of a system that can meet the needs of the users. Also, the requirements should be recorded in a document, email, user interface storyboard, executable prototype, or some other form. The requirements documentation should be referred to throughout the rest of the system development process to ensure the developing project aligns with user needs and requirements. Professionals must involve end users in this process to ensure that the new system will function adequately and meets their needs and expectations.

#### Step 3: Architectural Design

After the requirements have been determined, the necessary specifications for the hardware, software, people, and data resources, and the information products that will satisfy the functional requirements of the proposed system can be determined. The design will serve as a blueprint for the system and helps detect problems before these errors or

problems are built into the final system. Professionals create the system design, but must review their work with the users to ensure the design meets users‘ needs.

#### Step 4: Coding and Debugging

Coding and debugging is the act of creating the final system. This step is done by software developer.

#### Step 5: System Testing

The system must be tested to evaluate its actual functionality in relation to expected or intended functionality. Some other issues to consider during this stage would be converting old data into the new system and training employees to use the new system. End users will be key in determining whether the developed system meets the intended requirements, and the extent to which the system is actually used.

#### Step 6: Maintenance

Inevitably the system will need maintenance. Software will definitely undergo change once it is delivered to the customer. There are many reasons for the change. Change could happen because of some unexpected input values into the system. In addition, the changes in the system could directly affect the software operations. The software should be developed to accommodate changes that could happen during the post implementation period.

There are various software process models like:-

* + - Prototyping Model
    - RAD Model
    - The Spiral Model
    - The Waterfall Model
    - The Iterative Model

Of all these process models we‘ve used the Iterative model(The Linear Sequential Model) for the development of our project.

#### The Iterative model

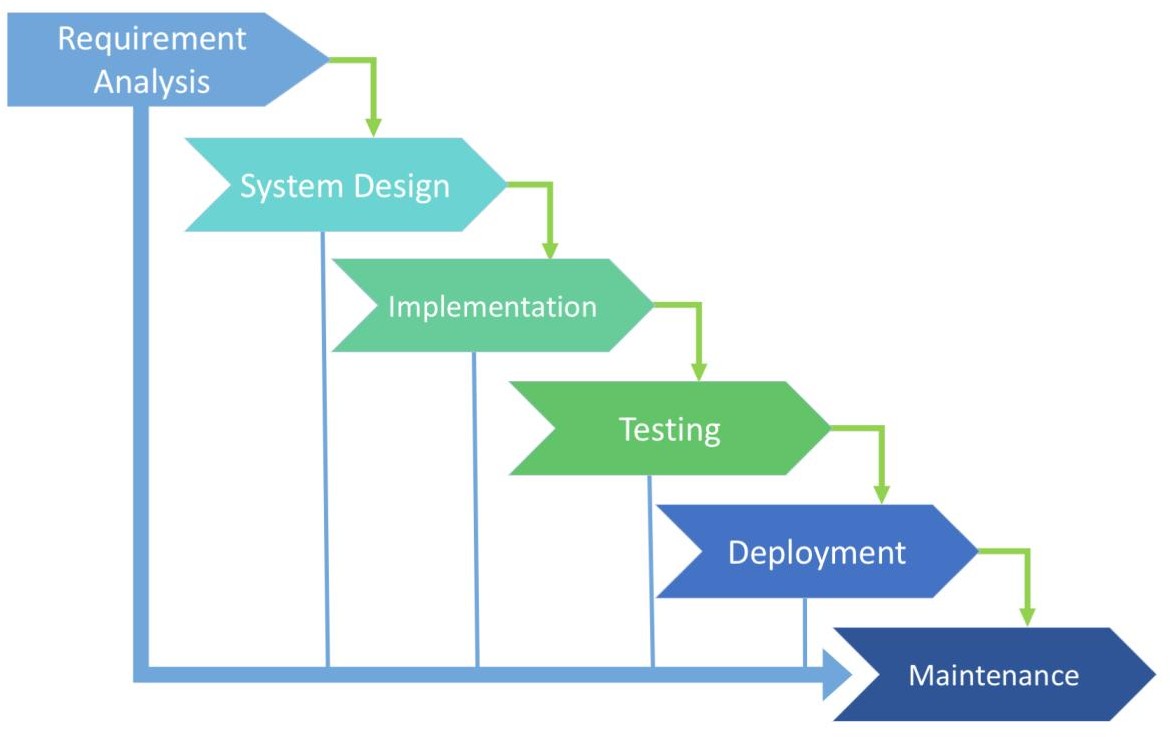


Fig 3.6 Iterative Model

The waterfall model derives its name due to the cascading effect from one phase to the other as is illustrated in Figure1.1. In this model each phase well defined starting and ending point, with identifiable deliveries to the next phase.

This model is sometimes referred to as the linear sequential model or the software life cycle.

The model consists of six distinct stages, namely:

* + - In the requirements analysis phase
      * The problem is specified along with the desired service objectives (goals).
      * The constraints are identified.
    - In the specification phase the system specification is produced from the detailed definitions of (a) and (b) above. This document should clearly define the product function.
    - In the system and software design phase, the system specifications are translated into a software representation. The software engineer at this stage is concerned with:
    - Data structure
    - Software architecture
    - Algorithmic detail
    - Interface representations

The hardware requirements are also determined at this stage along with a picture of the overall system architecture. By the end of this stage should the software engineer should be able to identify the relationship between the hardware, software and the associated interfaces. Any faults in the specification should

ideally not be passed down stream.

* + - In the implementation and testing phase stage the designs are translated into the software domain
      * Detailed documentation from the design phase can significantly reduce the coding effort.
      * Testing at this stage focuses on making sure that any errors are identified and that the software meets its required specification.
    - In the integration and system testing phase all the program units are integrated and tested to ensure that the complete system meets the software requirements. After this stage the software is delivered to the customer [Deliverable – The software product is delivered to the client for acceptance testing.]
    - The maintenance phase the usually the longest stage of the software. In this phase the software is updated to:
      * Meet the changing customer needs
      * Adapted to accommodate changes in the external environment
      * Correct errors and oversights previously undetected in the testing phases
      * Enhancing the efficiency of the software

Observe that feed back loops allow for corrections to be incorporated into the model. For example a problem/update in the design phase requires a ‗revisit‘ to the specifications phase. When changes are made at any phase, the relevant documentation should be updated to reflect that change.

#### Advantages of the Iterative Model

* + - Testing is inherent to every phase of the Iterative model
    - It is an enforced disciplined approach
    - It is documentation driven, that is, documentation is produced at every stage

#### Disadvantages of the Iterative Model

The waterfall model is the oldest and the most widely used paradigm. However, many projects rarely follow its sequential flow. This is due to the inherent problems associated with its rigid format. Namely:

* + - It only incorporates iteration indirectly, thus changes may cause considerable confusion as the project progresses.
    - As The client usually only has a vague idea of exactly what is required from the software product, this IM has difficulty accommodating the natural uncertainty that exists at the beginning of the project.
    - The customer only sees a working version of the product after it has been coded.

This may result in disaster any undetected problems are precipitated to this stage.

## Detailed Life Cycle of the Project

**3.3.1 Object Oriented Analysis and Design Diagrams**

* + - 1. **Use Case Diagram**

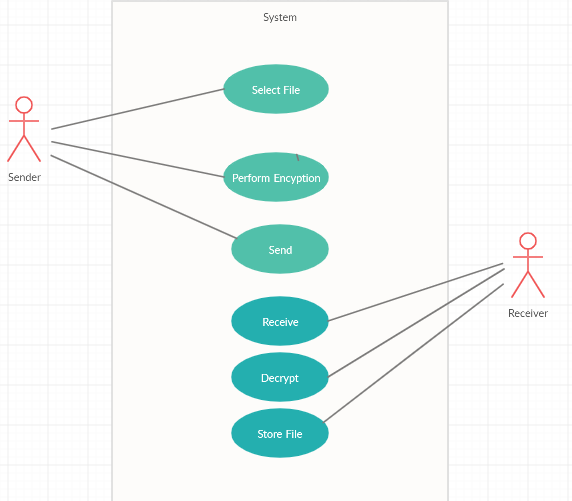


Fig 3.7 Use Case



Fig 3.8 Sender Use Case

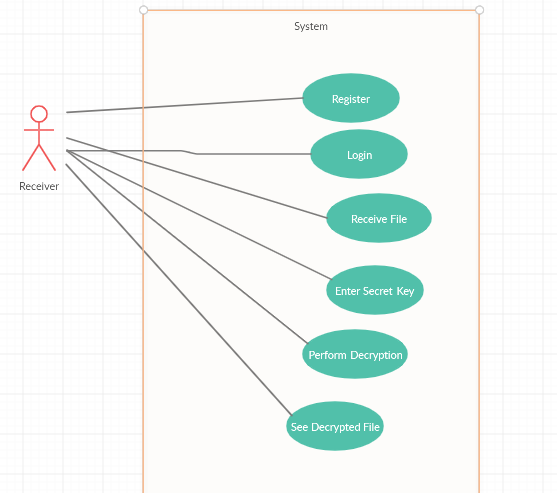


Fig 3.9 Receiver Admin Use Case

## Activity Diagram

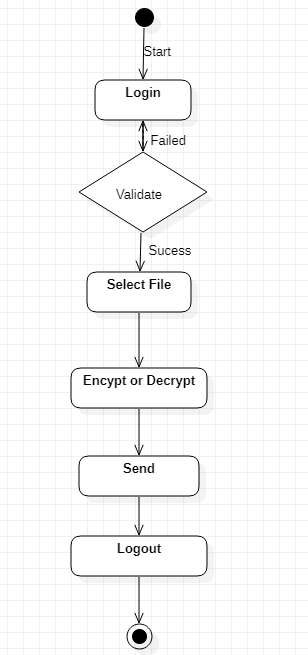
****

Fig 3.10 Activity Diagram

## Sequence Diagram

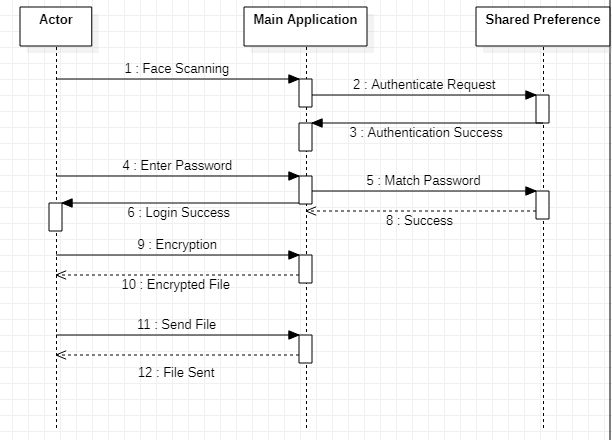
****

Fig 3.11 Sequence Diagram

## Flow Chart

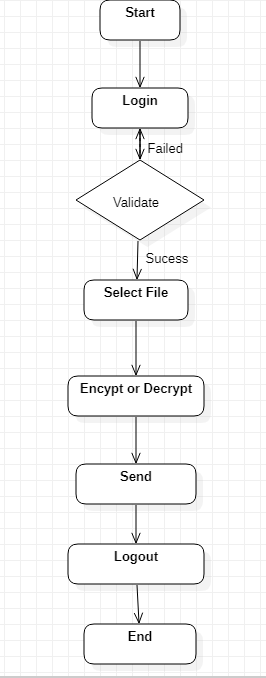


Fig 3.13 Flow Chart

## Data Flow Diagram

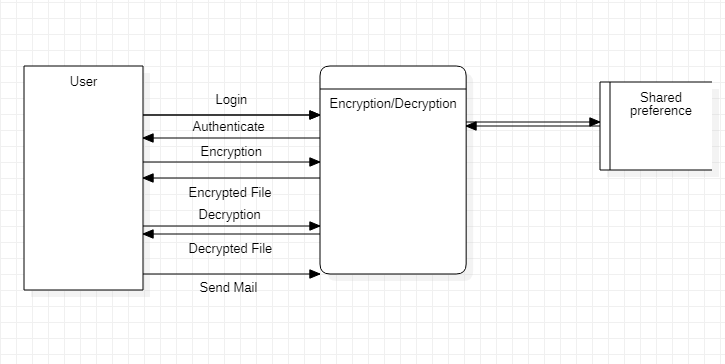
****

Fig 3.14 DFD Diagram

# TESTING

## Methodologies used for testing

Software testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. Software testing also provides an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include, but are not limited to, the process of executing a program or application with the intent of finding software bugs (errors or other defects).

Software testing can also be stated as the process of validating and verifying that a software program/application/product:

* + - Meets the business and technical requirements that guided its design and development
    - Works as expected; and
    - Can be implemented with the same characteristics.

#### Primary purpose

Testing is to detect software failures so that defects may be discovered and corrected. This is a non-trivial pursuit. Testing cannot establish that a product functions properly under all conditions but can only establish that it does not function properly under specific conditions.

#### Scope

The scope of software testing often includes examination of code as well as execution of that code in various environments and conditions as well as examining the aspects of code: does it do what it is supposed to do and do what it needs to do. In the current culture of software development, a testing organization may be separate from the development team.

#### Implementation

Software testing, depending on the testing method employed, can be implemented at any time in the development process. However, most of the test effort occurs after the requirements have been defined and the coding process has been completed. As such, the methodology of the test is governed by the software development methodology adopted.

#### Software Testing Model

**V Model**

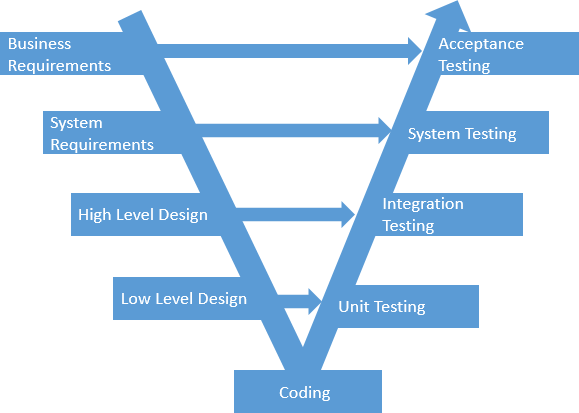


Fig 4.1 V-Model

The V-model involves building a logical V shape sequence where the testing techniques associated with the design are reflected as descending and are applied for the

―verification‖ and connected to the requirements or specifications parts are reflected as ascending and are applied for ―validation‖.

The V-model ordains that the code testing documentation is written in tandem with the development phases that means, for instance, the integration tests should be documented as and when the high level design is finalized and the unit tests should be ready as and when the detailed specifications are laid down.

The idea of the V-model is to have a implementation plan for the software testing at each level namely component, interface, system, acceptance and release of the software project which need to be adhered to eliminate discrepancies in the software simultaneously rather than waiting for the software development process to complete before handling it to the software testing professionals.

## Types Of Testing

**White Box Testing**

White box testing is performed based on the knowledge of how the system is implemented. White box testing includes 48nalysing data flow, control flow, information flow, coding practices, and exception and error handling within the system, to test the intended and unintended software behaviour. White box testing can be performed to validate whether code implementation follows intended design, to validate implemented security functionality, and to uncover exploitable vulnerabilities.

White box testing is used to test areas that cannot be reached from a black box level. White box testing uses an internal perspective of the system to design test cases based on internal structure. It requires programming skills to identify all paths through the

software. The tester chooses test case inputs to exercise paths through the code and determines the appropriate outputs.

#### Branch coverage

Branch coverage exclusively considers the logical value of the result of a condition (true or false). ‗Social Networking App‘ contains many conditions which has effect on overall working of the application. Selection of date range, category has overall effect on expense. Such cases are tested in branch coverage.

#### Path coverage

Path coverage requires the execution of all different paths through the test object. This is important with respect to mobile application. Application should execute all the paths and should not crash in between.

## Black Box Testing

Black-box testing is a method of software testing that tests the functionality of an application as opposed to its internal structures or workings. Specific knowledge of the application‘s code/internal structure and programming knowledge in general is not required.

Test cases are built around specifications and requirements, i.e. what the application is supposed to do. It uses external descriptions of the software, including specifications, requirements, and designs to derive test cases. These tests can be functional or non- functional, though usually functional. The test designer selects valid and invalid inputs and determines the correct output. There is no knowledge of the test object‘s internal structure. This method of test can be applied to all levels of software testing: unit, integration, functional, system and acceptance.

Black-Box testing helps to find errors such as:

* + - * Incorrect or missing functions
      * Interface errors
      * Errors in data structures

#### Equivalence class partitioning

Test case for input fields such as username, password, etc. are prepared and tested. Equivalence class partitioning helps to reduce total time in testing by dividing valid and invalid test cases.

#### Boundary value analysis

Faults often occur at the boundary of equivalence classes, because boundaries are not often defined clearly or misunderstood by programmers. Application having range fields such as date ranges are tested using this technique.

#### State transition testing

In many cases, not only current input, but also the history of execution or events or inputs, influences the outputs. Application has many fields such as date, category which has different effects on overall application. Minor change in each of them triggers changes in lot of input methods. These fields are tested under state transition testing.

## Test Report

**Testing and Result**

This is overall test usually done by people other than developer to find out if the project worked efficiently and if all the function in the software is functional and if there is any problem is the software .Then find out where the problem occurred and solved it.

Table 4.1 Testing and Result

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Condition** | **Input Data** | **Expected Output** | **Actual Output** | **Result** |
| Face Scanning | Low Light while scanning | Error message | App Crashes | **Pass** |
|  |  |  |  |  |
| Face Scanning | proper Light while scanning | Should save face | Display success message | **Pass** |
| Login | Invalid Password | Authentication Message | Error | **Pass** |
| Login | Valid Password | Authentication Message | Success | **Pass** |
| Encryption | Secret Key not of 16 characters | File not encrypted | Error Message | **Pass** |
| Encryption | Correct Secret Key | File encrypted | Encryption Completed | **Pass** |
| Decryption | Secret Key does not match | File not decypted | Error Message | **Pass** |
| Decryption | Correct Secret Key | File decrypted | Decryption Completed | **Pass** |
| Send Mail | File not selected | Mail not sent | Error | **Pass** |
| Send Mail | File selected | Mail sent | Success | **Pass** |

# SYSTEM IMPLEMENTATION

## Hardware Required

This Application requires user to have Android Smartphone which has active internet connection.

* + - Desktop / Laptop
      * Windows 7 or higher
      * Minimum 4 Gb RAM
      * Minimum 10 Gb Memory
    - Smartphone
      * Android OS Jelly Bean(API level 16) or higher
      * Minimum 2 Gb RAM
      * 10 Mb Storage

## Software Required

* + - Visual Studio
    - Android Studio
    - SQL Server Management Studio
    - Postman
  1. Languages Used
     + C#
     + JAVA
     + SQL
     + JSON

# SYSTEM MAINTENANCE AND

**EVALUATION**

## Maintenance

Maintenance is an enigma of the system development. It holds the software industry captive. Analysts spend more time in maintaining programs than coding them. Software maintenance denotes any changes made to the software product after it has been delivered to the customer. Most products need maintenance due to the wear and tear of the product. Software Maintenance can be divided into following types:

* + - **Corrective Maintenance:** It is necessary to rectify the bugs observed while the system is in use. **Data Vault** needs this maintenance for any removing flaws that can arise while sending the data or for correcting the logical bugs that might have been left unchecked as they appear only in real time like empty database.
    - **Perfective Maintenance:** Software product might need maintenance to support the new features that users want it to support, to change different functionalities of the system according to the customer demands, or to enhance the performance of the system.

Software Maintenance is essential as initial stages of any software developed are always unstable. Over the time it achieves stability as bugs are fixed and faults are removed to make the system accurate. System Maintenance is often termed as the task of doing repairs to the developed system. When websites are inaccessible due to attacks from hackers, server problems or for updating and repair, the

administrators of the website will often display an image apologizing for System Maintenance and Website downtime. This allows the user to understand that the website cannot be used and that the administrators are aware of the issue.

## Evaluation

System Evaluation is termed as the task of evaluating the success and failure of the system. It is performed with the help of following two V‘s:

#### Verification:

Verification determines whether the system is built correctly and does not contain technical errors. It also involves the review of the requirements, to verify that the right problem is being solved. Verification also ensures that the system is syntactically and logically correct and performs functionally as being specified. It is a static practice of verifying documents, design, code and program.

As verification relates to the humanized effort of checking the documents and files, we have taken utmost care to see to it that the application conforms to specifications. Reviews and inspections were carried out periodically. The web based application has been put through the process of Verification successfully.

#### Validation:

Validation on the other hand is a difficult task of insuring the meaning and content of the rules meet some carefully defined criteria of adequacy. Defining such criteria is the key to successfully conduct Validation procedure and demonstrating the level of acceptability of the system.

As Validation is a dynamic mechanism of validating and testing the actual product; we have implemented the process of validation by executing the code thoroughly. By

performing White Box as well as Black Box testing; along with Acceptance Testing, we have made sure that the application adheres to customer‘s expectation and requirements.

The target for validation was actual product-a unit, a module, a bent of integrated modules, and effective final product.

**Verification** process describes whether the outputs are according to inputs or not.

**Validation** process describes whether the software is accepted by the user or not.

# USER/OPERATIONAL MANUAL

## Security Aspects

#### What is Authentication?

Authentication is the process of identifying who users are when they visit a Application

#### What is Authorization?

Authentication is typically used in combination with authorization. Authorization is the process of determining whether a user has permissions to access a particular resource or to perform some action.

## Screenshots

# FUTURE ENHANCEMENT

This project has a wide scope for future development, as the user‘s requirement is always going to be changed which is not static and this needs are dynamic. The technology which is famous today becomes outdated in very next day. To keep abstract of technical improvements, the system may be further refined. So such type of system is improved in further future development. This enhancement is done in an efficient and effective manner. We can thus update the same with further modification establishment and can be integrated with minimal modification. Thus the project is extendable and can be developed in anytime with more advanced features.

#### Some advance future enhancements:

* Choice of encryption methods

# LIMITATIONS

* + - Android Smartphone is must.
    - Phone must have camera.

.

# CONCLUSION

Data encryption and decryption systems are used to improve information security to secure data that, thereby providing enhanced level of assurance such that the data that are encrypted cannot be viewed by unauthorized parties in the event of theft, loss or interception. This system replaces the existing data encryption and decryption system by adding some functionality such as digital signature

# APPENDIX

|  |  |  |
| --- | --- | --- |
| WCF | Windows Communication Foundation | Windows Communication Foundation is a framework for building service-oriented applications. Using WCF, you can send data as asynchronous messages from one service endpoint to  another. |

**BIBLIOGRAPHY**

### <https://msdn.microsoft.com/>

* + - [*https://www.codeproject.com/*](https://www.codeproject.com/)
    - [*https://github.com/*](https://github.com/)
    - [*https://stackoverflow.com/*](https://stackoverflow.com/)
    - [*https://www.tutlane.com/tutorial/android*](https://www.tutlane.com/tutorial/android)
    - [*https://www.androidhive.info/*](https://www.androidhive.info/)

.