**DATA SECURITY**

***A Mini Project-1 Report submitted***

***in partial fulfilment of the requirements***

***for the award of the degree of***

**BACHELOR OF TECHNOLOGY**

***In***

**COMPUTER SCIENCE & ENGINEERING**

***By***

1. K. V. Subbalakshmi 3. P. S. N. S. Sowmya

2. G. B. Vyshnavi 4. K. Jaya Kumari

***Under the esteemed guidance of***

**Mr. P. J. R. Shalem Raju** M.Tech (Ph.D)

**Associate Professor**



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**SHRI VISHNU ENGINEERING COLLEGE FOR WOMEN(A)**

**(Approved by AICTE, Accredited by NBA & NAAC, Affiliated to JNTU Kakinada)**

**BHIMAVARAM – 534 202**

**2019 – 2020**

**SHRI VISHNU ENGINEERING COLLEGE FOR WOMEN(A)**

**(Approved by AICTE, Accredited by NBA & NAAC, Affiliated to JNTU Kakinada)**

**BHIMAVARAM – 534 202**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**



**CERTIFICATE**

*This is to certify that the Mini Project-1 entitled “****Data Security****”, is being submitted by*

***K. V. Subbalakshmi,*** ***G. B. Vyshnavi, P. S. N. S. Sowmya, K. Jaya Kumari*** *bearing* *the* ***Regd. Numbers 18B01A0573*** *,* ***18B01A0569 , 18B01A05B1 , 19B05A0510*** *in partial fulfilment of the requirements for the award of the degree of “****Bachelor of Technology*** *in* ***Computer Science & Engineering****” is a record of bonafide work carried out by them under my guidance and supervision during the academic year 2019–2020 and it has been found worthy of acceptance according to the requirements of the university.*

**Internal Guide Head of the Department**

**Acknowledgements**

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose constant encouragement and guidance has been a source of inspiration throughout the course of this project. We take this opportunity to express our gratitude to all those who have helped us in this project.

We wish to place our deep sense of gratitude to **Sri K. V. Vishnu Raju, Chairman of SVES**, for his constant support on each and every progressive works of ours.

We wish to express our sincere thanks to **Dr. G. Srinivasa Rao, Principal of SVECW**, for being a source of inspiration and constant encouragement.

We wish to express our sincere thanks to **Dr. P. Srinivasa Raju, Vice Principal of SVECW**, for being a source of inspiration and constant encouragement.

On the successful completion of our project report, we convey our sincere thanks to

**Dr. V. Purushothama Raju**, **Head of the Department of Computer Science and Engineering**, for his valuable piece of advice in completing this project successfully.

We are really thankful to our internal guide Mr. P. J. R. Shalem Raju, who encouraged us a lot in the completion of the project work.

**Project Associates**

1. K. V. Subbalakshmi (18B01A0573)
2. G. B. Vyshnavi (18B01A0569)
3. P. S. N. S. Sowmya (18B01A05B1)
4. K. Jaya Kumari (19B05A0510)

**Abstract**

In our project, there is an Encryption and Decryption, a type of cryptography, refers to the process of scrambling information so that the observer cannot be detecting the data.  Data Encryption helps you to protect the privacy of your email messages, documents and sensitive files. Encryption works with both - text information and files. We just have to select what we want to encrypt, and encryption and decryption helps us keep documents, private information and files in a confidential way. Encryption is also used to ensure the confidentiality of the file and documents from the adversary so that the files and documents remain in a secure way.

Data encryption is also used to provide the security and safety of the files and other important documents from the opponent so that while sending the files or documents nobody else other than the recipient can see it.

Encryption: Encryption is the process of translating plain text data into something that appears to be random and meaningless

Decryption: Decryption is the process of converting ciphertext back to plain text.

 Ciphers: A cipher is a method of hiding words or text with encryption by replacing original letters with other letters, numbers and symbols through substitution or transposition. A combination of substitution and transposition is also often employed.

1. Reverse Cipher

2. Caesar Cipher (shift key)

3.Vigenere Cipher (Alphabet replacing)

4.Word length

5.Transposition Cipher

6. ROT13(Rotating by 13 places) process

Contents

  S.No.                    Topic                        Page No.

1. Introduction   1 - 2

2. System Analysis 3 - 4

2.1 Existing System

2.2 Proposed System

2.3 Feasibility Study

3. System Requirements Specification 5 - 6

3.1 Software Requirements

3.2 Hardware Requirements

3.3 Functional Requirements

4. System Design 7 - 19

4.1 Introduction

4.2 UML Diagrams

4.3 Database Design

4.3.1 ER Diagrams

4.3.2 Table Structures

5. System Implementation 20 - 22

5.1 Introduction

5.2 Project Modules

5.3 Algorithms

6. System Testing 23 - 24

6.1 Introduction

6.2 Testing Methods

6.3 Test Cases

7. Conclusion 25 - 25

8. Bibliography 25 - 25

9. Appendix 26 - 30

9.1 Introduction to Java

9.2 Introduction to HTML

9.3 Introduction to MySQL

**Introduction**

**Definition of Cryptography:**

Cryptography is associated with the process of converting ordinary plain text into unintelligible text and vice-versa. It is a method of storing and transmitting data in a particular form so that only those for whom it is intended can read and process it. Cryptography not only protects data from theft or alteration, but can also be used for user authentication.

**Terminology:**

**plain text:** Data that can be read and understood without any special measures.

**Cipher text:** Encrypting plaintext results in unreadable gibberish

You use encryption to make sure that information is hidden from anyone for whom it is not intended, even those who can see the encrypted data.

**Key**: Some critical information used by the cipher, known only to the sender & receiver

**Enciphering or encode or encryption**: The process of converting plaintext to cipher text using a cipher and a key

**Deciphering or decode or decryption**: the process of converting ciphertext back into plaintext using a cipher and a key

**Cryptanalysis:** The study of principles and methods of transforming an unintelligible message back into an intelligible message without knowledge of the key. Also called code breaking

**Cryptology:** Both cryptography and cryptanalysis

**Code:** An algorithm for transforming an intelligible message into an unintelligible one using a code-book C

**Ciphers**: A cipher is a method of hiding words or text with encryption by replacing original letters with other letters, numbers and symbols through substitution or transposition. A combination of substitution and transposition is also often employed.

For a Cipher to be Useful, several things must be known to both the Sending and Receiving ends.

1. Algorithm or Method Used to encipher the Plain text

2. The Key

3. Period or time during which the key is Valid

Ciphers are broken into two main Categories. They are:

1. Substitution Ciphers

2. Transposition Ciphers

**Substitution Ciphers**: These replace letters in the Plain text with other letters Or Symbols, keeping the order in which the symbol falls the same.

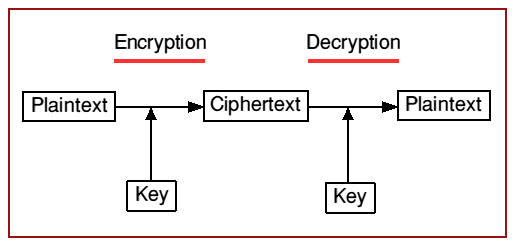
**Transposition Ciphers**: These keep all of the Original letters intact but Mix-up their order.

Encryption and Decryption is used for the One who wants to keep the data secure. Though someone opened and tried to see the data. He cannot understand as it is in encrypted form that is Non-readable language. Here the key to decrypt the data is given by the user. So only the user can decrypt his Personal data.

**Applications:**

Cryptography is used in many applications:

* banking transactions cards
* computer passwords
* e- commerce transactions
* encryption question papers
* social account password …...etc

**Process of encryption and decryption:**

**2. System Analysis**

**2.1 Existing System**

It completely depends upon the secret key. if you forget a key, we cannot recover the data and the data will be lost. Cryptography comes at cost in terms of money. It is a time taking process. It is always vulnerable for brute force attack.

**2.2 Proposed System**

Whenever you forgot the key, you can use the recover option, so that we can get your data back without losing it. It is an open source and can be used at free of cost i.e no charge. It is a handy process and can be done in less time when compared to other. By applying multiple ciphers, we can avoid the brute force attack up to some extends. We have additional security features like file lock to the selected files.

It can be used in real time applications like

• encryption and decryption in emails

• encrypting and decrypting question papers

• encrypting passwords

• encrypting and decrypting sensitive information.

**2.3 Feasibility Study**

Generally the feasibility study is used for determining the resource cost, benefits and

whether the proposed system is feasible with respect to the organization. The proposed

system feasibility could be as follows. There are six types of feasibility which are equally

important are:

* Technical feasibility
* Economic feasibility
* Behavioural feasibility

**Technical Feasibility**

Technical feasibility deals with the existing technology, software and hardware requirements

for the proposed system. The proposed system “Hand Gesture Detection” is planned to run

on phyton. Thus, the project is considered technically feasible for the development. The work for the project can be done with current equipment, existing software technology and

available personnel. Hence the proposed system is technically feasible.

**Economic Feasibility**

This method is most frequently used for evaluating the effectiveness of a python. It is also

called as benefit analysis. In this project “Hand Gesture Detection” is developed on current

equipment, existing software technology. Since the required hardware and software for

developing the system is already available in the organization, it does not cost must

developing the proposed system.

**Behavioural Feasibility**

This project has been implemented by phyton and it satisfies all conditions and norms of the

organization and the users. This proposed system “Hand Gesture Detection” Application has much behavioural feasibility because users are provided with a better facility.

**3. System Requirements Specification**

**3.1 Software Requirements**

* HTML (Hyper Text Mark-up Language)
* Java
* JSP (Java Server Page)
* JDBC (Java Database Connectivity)
* MySql

**3.2 Hardware Requirements**

Operating System : Windows 10 or Ubuntu

RAM : 4GB

Processor : Intel i3 or 15 or i7

**3.3 Functional Requirements**

Functional requirements are the requirements which deals with the operational

requirements of the system and the requirements that are requested by the user.

They are:

Functional requirements are the requirements which deals with the operational

requirements of the system and the requirements that are requested by the user.

They are:

**Confidentiality** - Information cannot be understood by anyone

**Integrity** - Information cannot be altered.

**Non-repudiation** - Sender cannot deny his/her intentions in the transmission of the information at a later stage

**Authentication** - Sender and receiver can confirm each

**Non – Functional Requirements**

A non-functional requirement specifies the process that can be used to check the

operations of a system. They are contrasted with the functional requirements that define

specific behaviour or functions. The plan for implementing non-functional requirements is

detailed in the system architecture, because they are usually architecturally significant

requirements.

Some of them are,

* Robustness to cope with errors during execution and cope with erroneous input.
* Performance which tells us whether the input is valid or not and obtained output.
* The cost should be less when compared to the existing system.
* The system acts as platform independent.
* The system is secured.

**4. System Design**

System design is the process of designing the elements of a system such as the architecture, modules and components, the different interfaces of those components and the data that goes through that system.

**System Analysis** is the process that decomposes a system into its component pieces for the purpose of defining how well those components interact to accomplish the set requirements. The purpose of the System Design process is to provide sufficient detailed data and information about the system and its system elements to enable the implementation consistent with architectural entities as defined in models and views of the system architecture.

The purpose of the design phase is to plan a solution of the problem specified by the requirement document. This phase is the first step in moving from problem domain to the solution domain. The design of a system is perhaps the most critical factor affecting the quality of the software, and has a major impact on the later phases, particularly testing and maintenance. The output of this phase is the design document. This document is similar to a blue print or plan for the solution, and is used later during implementation, testing and maintenance.

    The design activity is often divided into two separate phase-system design and detailed design. System design, which is sometimes also called top-level design, aims to identify the modules that should be in the system, the specifications of these modules, and how they interact with each other to produce the desired results. At the end of system design all the major data structures, file formats, output formats, as well as the major modules in the system and their specifications are decided.

    A design methodology is a systematic approach to creating a design by application of set of techniques and guidelines. Most methodologies focus on system design. The two basic principles used in any design methodology are problem partitioning and abstraction. A large system cannot be handled as a whole, and so for design it’s partitioned into smaller systems. Abstraction is a concept related to problem partitioning. When partitioning is used during design, the design activity focuses on one part of the system at a time. Since the part being designed interacts with other parts of the system, a clear understanding of the interaction is essential for property designing the part.

**4.2 UML Diagrams**

**Use Case diagrams :-**

The purpose of use case diagram is to capture the dynamic aspect of a system. However, this definition is too generic to describe the purpose, as other four diagrams (activity, sequence, collaboration, and State chart) also have the same purpose. We will look into some specific purpose, which will distinguish it from other four diagrams. Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. Hence, when a system is analysed to gather its functionalities, use cases are prepared and actors are identified. When the initial task is complete, use case diagrams are modelled to present the outside view.

In brief, the purposes of use case diagrams can be said to be as follows

Used to gather the requirements of a system.

* Used to get an outside view of a system.
* Identify the external and internal factors influencing the system.
* Show the interaction among the requirements are actors.

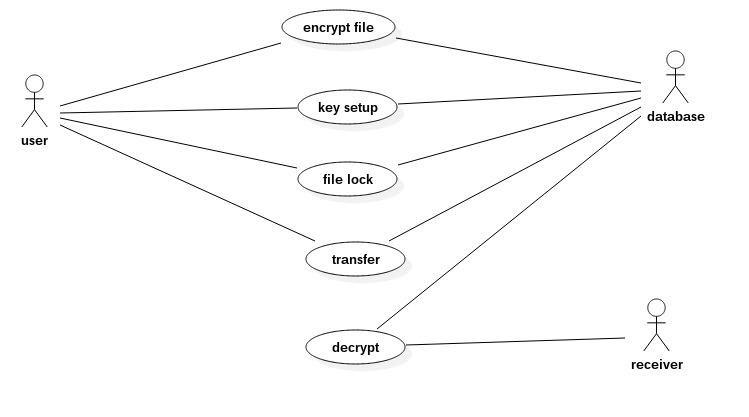
How to Draw a Use Case Diagram?

Use case diagrams are considered for high level requirement analysis of a system. When the requirements of a system are analysed, the functionalities are captured in use cases. We can say that use cases are nothing but the system functionalities written in an organized manner. The second thing which is relevant to use cases are the actors. Actors can be defined as something that interacts with the system. Actors can be a human user, some internal applications, or may be some external applications. When we are planning to draw a use case diagram, we should have the following items identified.

* Functionalities to be represented as use case
* Actors
* Relationships among the use cases and actors.

Use case diagrams are drawn to capture the functional requirements of a system. After identifying the above items, we have to use the following guidelines to draw an efficient use case diagram

* The name of a use case is very important. The name should be chosen in such a way so that it can identify the functionalities performed.
* Give a suitable name for actors.
* Show relationships and dependencies clearly in the diagram.
* Do not try to include all types of relationships, as the main purpose of the diagram is to identify the requirements.
* Use notes whenever required to clarify some important points.



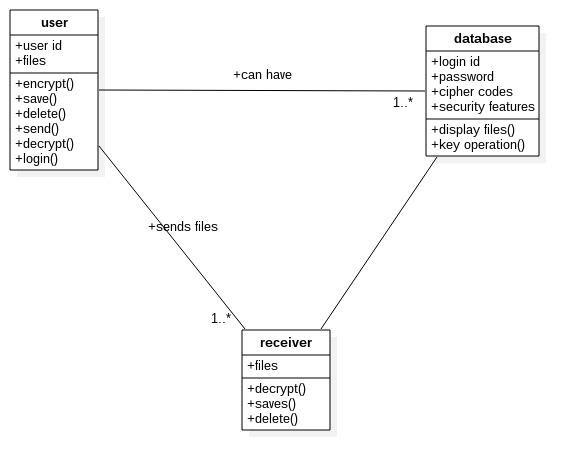
**CLASS DIAGRAMS:-**

Class diagrams are one of the most useful types of diagrams in UML as they clearly map out the structure of a particular system by modelling its classes, attributes, operations, and relationships between objects. With our UML diagramming software, creating these diagrams is not as overwhelming as it might appear. This guide will show you how to understand, plan, and create your own class diagrams. The Unified Modelling Language (UML) can help you model systems in various ways. One of the more popular types in UML is the class diagram. Popular among software engineers to document software architecture, class diagrams are a type of structure diagram because they describe what must be present in the system being modelled. No matter your level of familiarity with UML or class diagrams, our UML software is designed to be simple and easy to use. UML was set up as a standardized model to describe an object-oriented programming approach. Since classes are the building block of objects, class diagrams are the building blocks of UML. The various components in a class diagram can represent the classes that will actually be programmed, the main objects, or the interactions between classes and objects. The class shape itself consists of a rectangle with three rows. The top row contains the name of the class, the middle row contains the attributes of the class, and the bottom section expresses the methods or operations that the class may use. Classes and subclasses are grouped together to show the static relationship between each object.

**Benefits of class diagrams:**

Class diagrams offer a number of benefits for any organization. Use UML class diagrams to:

* Illustrate data models for information systems, no matter how simple or complex.
* Better understand the general overview of the schematics of an application.
* Visually express any specific needs of a system and disseminate that information throughout the business.
* Create detailed charts that highlight any specific code needed to be programmed and implemented to the described structure.
* Provide an implementation-independent description of types used in a system that are later passed between its components.

**Sequence diagrams:-**

UML Sequence Diagrams are interaction diagrams that detail how operations are carried out. They capture the interaction between objects in the context of a collaboration. Sequence Diagrams are time focus and they show the order of the interaction visually by using the vertical axis of the diagram to represent time what messages are sent and when.

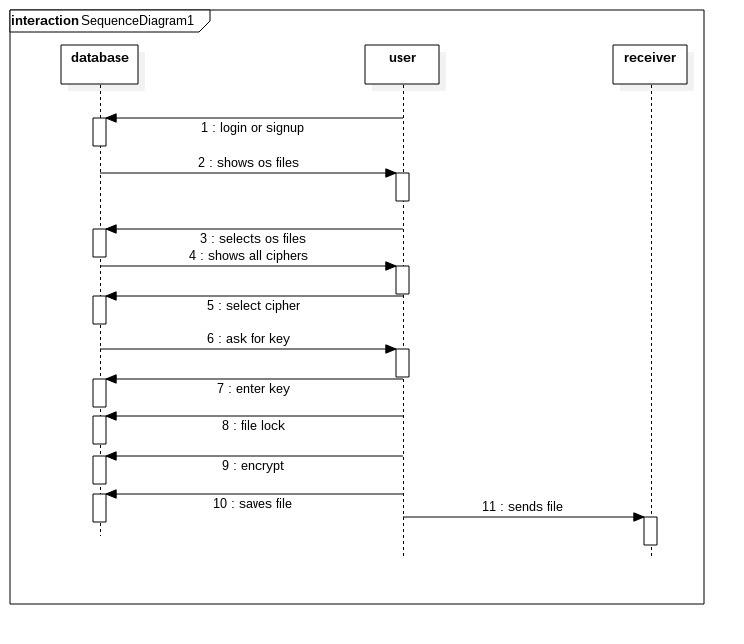
Sequence Diagrams captures:

* The interaction that takes place in a collaboration that either realizes a use case or an operation (instance diagrams or generic diagrams)
* high-level interactions between user of the system and the system, between the system and other systems, or between subsystems (sometimes known as system sequence diagrams)

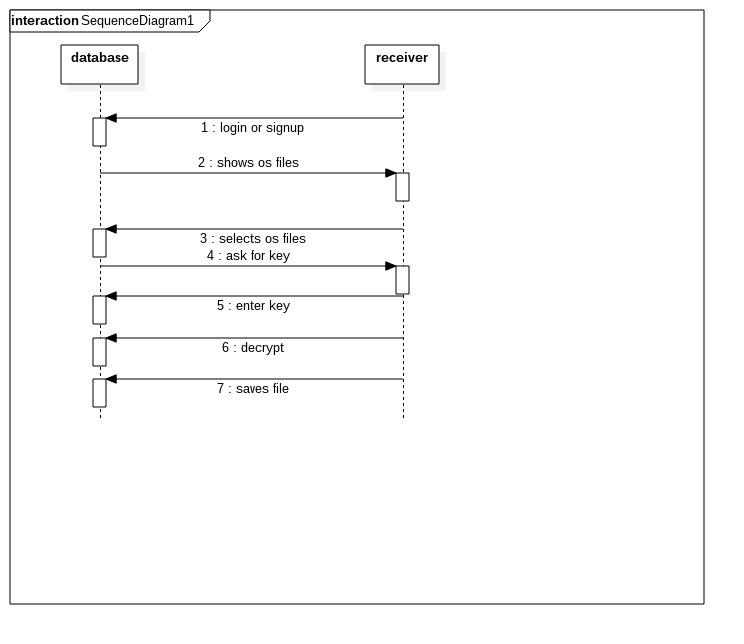
Purpose of Sequence Diagram:

* Model high-level interaction between active objects in a system
* Model the interaction between object instances within a collaboration that realizes a use case
* Model the interaction between objects within a collaboration that realizes an operation
* Either model generic interactions (showing all possible paths through the interaction) or specific instances of a interaction (showing just one path through the interaction)

Sequence diagram for encryption :-



Sequence diagram for decryption :-



**ACTIVITY DIAGRAMS:-**

Activity diagram is another important behavioural diagram in UML diagram to describe dynamic aspects of the system. Activity diagram is essentially an advanced version of flowchart that model the flow from one activity to another activity. Activity Diagrams describe how activities are coordinated to provide a service which can be at different levels of abstraction. Typically, an event needs to be achieved by some operations, particularly where the operation is intended to achieve a number of different things that require coordination, or how the events in a single use case relate to one another, in particular, use cases where activities may overlap and require coordination. It is also suitable for model how a collection of use cases co-ordinate to represent business workflows

1. Identify candidate use cases, through the examination of business workflows

2. Identify pre- and post-conditions (the context) for use cases

3. Model workflows between/within use cases

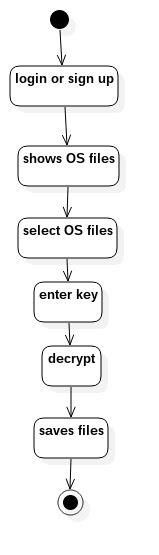
4. Model complex workflows in operations on objects

5. Model in detail complex activities in a high-level activity Diagram

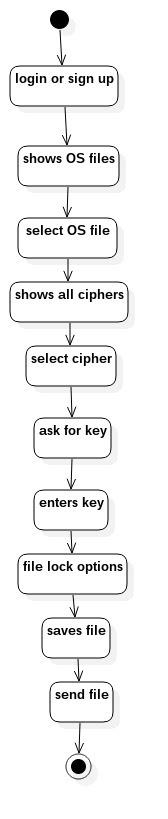
**Benefits of activity diagrams**

* Demonstrate the logic of an algorithm.
* Describe the steps performed in a UML use case.
* Illustrate a business process or workflow between users and the system
* Simplify and improve any process by clarifying complicated use cases.
* Model software architecture elements, such as method, function, and operation.

**Activity diagram for decryption :-**



**Activity diagram for encryption :-**



**Collaboration diagrams:-**

A collaboration diagram, also known as a communication diagram, is an illustration of the relationships and interactions among software objects in the Unified Modelling Language (UML). These diagrams can be used to portray the dynamic behaviour of a particular use case and define the role of each object. Collaboration diagrams are created by first identifying the structural elements required to carry out the functionality of an interaction. A model is then built using the relationships between those elements. Several vendors offer software for creating and editing collaboration diagrams. Notations of a collaboration diagram:- A collaboration diagram resembles a flowchart that portrays the roles, functionality and behaviour of individual objects as well as the overall operation of the system in real time. The four major components of a collaboration diagram are:

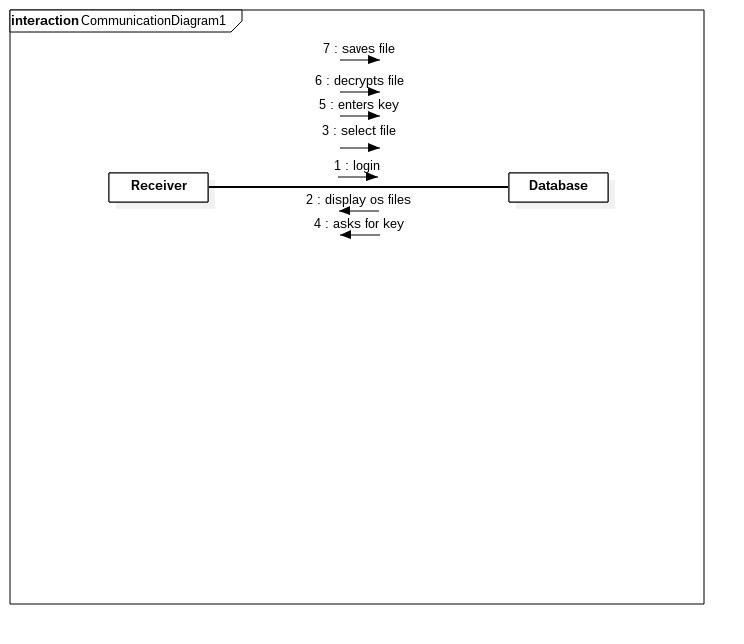
1. Objects- Objects are shown as rectangles with naming labels inside. The naming label follows the convention of object name: class name. If an object has a property or state that specifically influences the collaboration, this should also be noted.

2. Actors- Actors are instances that invoke the interaction in the diagram. Each actor has a name and a role, with one actor initiating the entire use case.

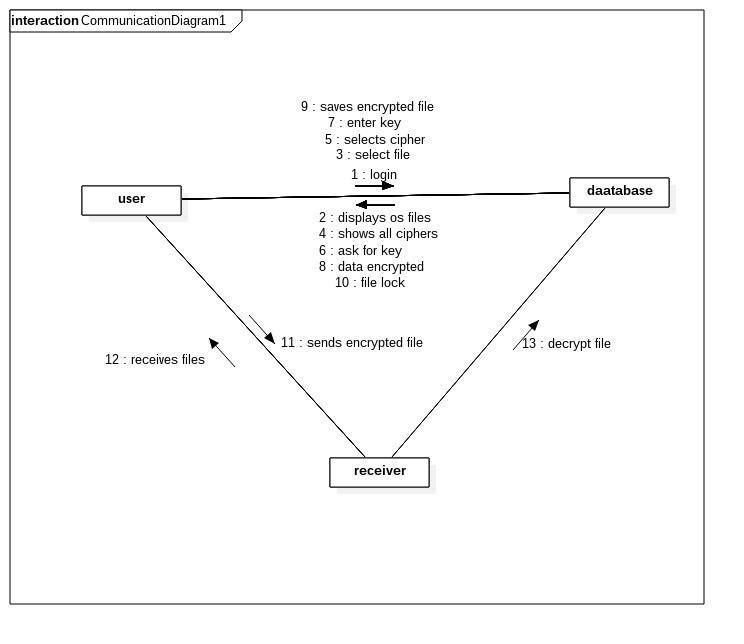
3. Links- Links connect objects with actors and are depicted using a solid line between two elements. Each link is an instance where messages can be sent.

4. Messages- Messages between objects are shown as a labelled arrow placed near a link. These messages are communications between objects that convey information about the activity and can include the sequence number.

**Collaboration diagram for Decryption:-**



**Collaboration Diagram for Encryption:-**

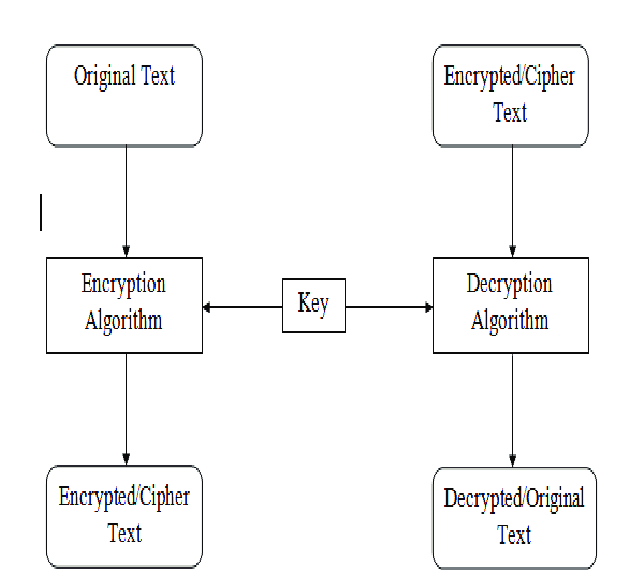


### 

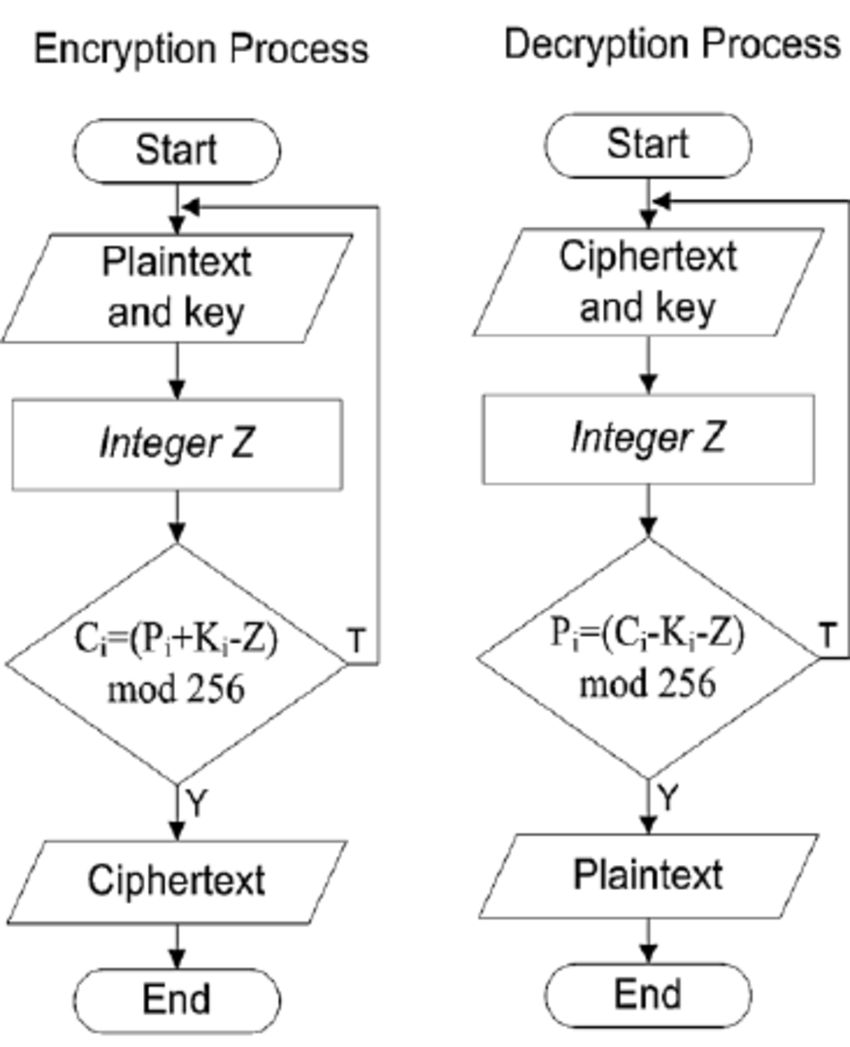
### 4.3 Database Design:-

**4.3.1 ER diagrams**

**ENTITY RELATIONAL (ER) MODEL** is a high-level conceptual data model diagram. ER modelling helps you to analyse data requirements systematically to produce a well-designed database. The Entity-Relation model represents real-world entities and the relationship between them. It is considered a best practice to complete ER modelling before implementing your database.



**4.3.2 Table Structures:-**



## 

## 

## 

## 5.System Implementation

### 5.1 Introduction

In this project, we encrypt data by using Cipher Codes and then we decrypt the data which is encrypted. Some of the six ciphers which we work on are as shown below

1. Reverse Cipher

2. Caesar Cipher (shift key)

3.Vigenere Cipher (Alphabet replacing)

4.Word length

5.Transposition Cipher

6. ROT13(Rotating by 13 places) process

**Reverse cipher:**

Reverse cipher uses a pattern to reverse the string to convert the plain text into cipher text.it is as shown below

## 

## 

#### Caesar Cipher:

The Caesar cipher is one of the earliest known and simplest ciphers. It is a type of substitution cipher in which each letter in the plaintext is 'shifted' a certain number of places down the alphabet

example: caesar cipher with key = 1

plaintext: defend the east wall of the castle

ciphertext: efgfoe uif fbtu xbmm pg uif dbtumf

#### Vigenere Cipher:

The **Vigenere Cipher** is a method of [encrypting](https://en.wikipedia.org/wiki/Encryption) [alphabeti](https://en.wikipedia.org/wiki/Alphabetic)c text by using a series of interwoven [Caesar ciphers](https://en.wikipedia.org/wiki/Caesar_cipher), based on the letters of a keyword. It employs a form of [polyalphabetic substitution](https://en.wikipedia.org/wiki/Polyalphabetic_cipher).

#### Word length:

The word length divides the plain text into parts having certain length

example:

plain text: cryptography

cipher text: cry pto gra phy (if word length =3)

#### Transposition cipher:

[cryptography](https://en.wikipedia.org/wiki/Cryptography), a **transposition cipher** is a method of encryption by which the positions held by units of [plaintext](https://en.wikipedia.org/wiki/Plaintext) (which are commonly characters or groups of characters) are shifted according to a regular system, so that the [ciphertext](https://en.wikipedia.org/wiki/Ciphertext) constitutes a [permutation](https://en.wikipedia.org/wiki/Permutation) of the plaintext

#### ROT13 CIPHER:

**ROT13** ("**rotate by 13 places**", sometimes hyphenated **ROT-13**) is a simple letter [substitution cipher](https://en.wikipedia.org/wiki/Substitution_cipher) that replaces a letter with the 13th letter after it, in the alphabet. ROT13 is a special case of the [Caesar cipher](https://en.wikipedia.org/wiki/Caesar_cipher) which was developed in ancient Rome.

Because there are 26 letters (2×13) in the [basic Latin alphabet](https://en.wikipedia.org/wiki/ISO_basic_Latin_alphabet), ROT13 is its own [inverse](https://en.wikipedia.org/wiki/Inverse_function); that is, to undo ROT13, the same [algorithm](https://en.wikipedia.org/wiki/Algorithm) is applied, so the same action can be used for encoding and decoding. The algorithm provides virtually no [cryptographic](https://en.wikipedia.org/wiki/Cryptography) security

#### 5.2 Project modules

#### 5.3 Algorithms

#### Algorithm for Reverse Cipher

## Reverse Cipher uses a pattern of reversing the string of plain text to convert as cipher text.

## The process of encryption and decryption is same.

## To decrypt cipher text, the user simply needs to reverse the cipher text to get the plain text.

## Algorithm for Caesar Cipher:

## Traverse the given text one character at a time.

## For each character, transform the given character as per the rule, depending on whether we’re encrypting or decrypting the text the new string generated.

* return to cipher text generated

## 

## 

## 

## 6.System Testing

**6.1. INTRODUCTION:**

Software Testing is an important element of the software quality assurance and represents the ultimate review of specification, design and coding. The increasing feasibility of software as a system and the cost associated with the software failures are motivated forces for III planned through testing.

**TESTING OBJECTIVES:**

These are several rules that can save as testing objectives:

* Testing is a process of executing program with the intent of finding an error.
* A good testcase is one that has a high probability of finding an undiscovered error.

**Test Levels:**

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or darkness in a work product. It provides a way to check the functionality of components, subassemblies, assemblies and/or a finished product. Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

**6.2. TESTING METHODS**

**6.2.1 Unit Testing**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application.

**6.2.2 Integration Testing**

Integration tests are designed to test integrated software components to determine if they run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields.

**6.2.3 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. Organization and preparation of functional tests is focused on requirements, key functions, or special test cases.

**6.2.4 System Testing**

System testing ensures that the entire integrated software system meet requirements. It tests a configuration to ensure known and predictable results. An exampleof system testing is the configuration oriented system integration test.

**6.2.5 White Box Test**

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

**6.2.6 Black Box Test**

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document.

**6.2.7 Unit Testing**

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

**6.2.8 Integration Testing**

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

**6.2.9 Acceptance Testing**

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user.

## 7.Conclusion

Electronic networks for banking, shopping, inventory control, benefit and service delivery, information storage and retrieval, distributed processing, and government applications will need improved methods for access control and data security. Information security can be easily achieved by using Cryptography technique. D E S (data encryption standard) is now considered to be insecure for some applications like banking system.

Cryptography is used to achieve few goals like confidentiality, data integrity, authentication etc. of the data Now, in order to achieve these goals various cryptographic algorithms are developed by various people. For a very minimal amount of data those algorithms wouldn’t be cost effective since those are not designed for small amount of data. The aim of this work was to design and implement a new algorithm to address this issue so that we don’t have to apply those algorithms (which are not cost-effective) to encrypt a small amount of data.

**8.Bibliography**

* Cryptography and Network Security Seventh Edition by William Stallings
* Cryptography Theory & practice by Douglas Stinson

## 

## 9. Appendix

**9.1** [**Introduction of Java**](https://www.vogella.com/tutorials/JavaIntroduction/article.html#a-small-history-of-java)

**Platform independent:** Java programs use the Java virtual machine as abstraction and do not access the operating system directly. This makes Java programs highly portable. A Java program (which is standard-compliant and follows certain rules) can run unmodified on all supported platforms, e.g., Windows or Linux.

* **Object-orientated programming language:** Except the primitive data types, all elements in Java are objects.
* **Strongly-typed programming language:** Java is strongly-typed, e.g., the types of the used variables must be pre-defined and conversion to other objects is relatively strict, e.g., must be done in most cases by the programmer.
* **Interpreted and compiled language:** Java source code is transferred into the bytecode format which does not depend on the target platform. These bytecode instructions will be interpreted by the Java Virtual machine (JVM). The JVM contains a so-called Hotspot-Compiler which translates performance critical bytecode instructions into native code instructions.
* **Automatic memory management:** Java manages the memory allocation and de-allocation for creating new objects. The program does not have direct access to the memory. The so-called garbage collector automatically deletes objects to which no active pointer exists.

**Java is used for:**

* Desktop applications
* Mobile applications (especially Android apps)
* Web applications
* Web servers and application servers
* Games
* Database connection
* And much, much more!
* **Why Use Java?**
* Java on different platforms (Windows, Mac, Linux, Raspberry Pi, etc.)
* It is one of the most popular programming languages in the world
* It is easy to learn and simple to use
* It is open-source and free
* It is secure, fast and powerful
* It has a huge community support (tens of millions of developers)
* Java is an object-oriented language which gives a clear structure to programs and allows code to be reused, lowering development costs
* As Java is close to C++ and C#, it makes it easy for programmers to switch to
* Java or vice versa.

**Features of Java**

The primary objective of [Java programming](https://www.javatpoint.com/java-tutorial) language creation was to make it portable, simple and secure programming language. Apart from this, there are also some excellent features which play an important role in the popularity of this language. The features of Java are also known as java *buzzwords*.

A list of most important features of Java language is given below.

1. Simple
2. Object-Oriented
3. Portable
4. Platform independent
5. Secured
6. Robust
7. Architecture neutral
8. Interpreted
9. High Performance
10. Multithreaded
11. Distributed
12. Dynamic

**Java is an Object-Oriented language**

Object oriented programming is a way of organizing programs as collection of objects, each of which represents an instance of a class.

4 main concepts of Object-Oriented programming are:

1. [Abstraction](https://beginnersbook.com/2013/03/oops-in-java-encapsulation-inheritance-polymorphism-abstraction/)
2. [Encapsulation](https://beginnersbook.com/2013/05/encapsulation-in-java/)
3. [Inheritance](https://beginnersbook.com/2013/05/java-inheritance-types/)
4. [Polymorphism](https://beginnersbook.com/2013/03/polymorphism-in-java/)

**9.2 Introduction to HTML:-**

HTML stands for Hyper Text Mark-up Language. It is used to design web pages using mark-up language. HTML is the combination of Hypertext and Mark-up language. Hypertext defines the link between the web pages. Mark-up language is used to define the text document within tag which defines the structure of web pages. This language is used to annotate (make notes for the computer) text so that a machine can understand it and manipulate text accordingly. Most of mark-up (e.g. HTML) languages are human readable. Language uses tags to define what manipulation has to be done on the text.  
HTML is a mark-up language which is used by the browser to manipulate text, images and other content to display it in required format. HTML was created by Tim Berners-Lee in 1991. The first ever version of HTML was HTML 1.0 but the first standard version was HTML 2.0 which was published in 1999.

**Elements and Tag:** HTML uses predefined tags and elements which tells the browser about content display property. If a tag is not closed then browser applies that effect till end of page.

**HTML page structure:** The Basic structure of HTML page is given below. It contains some elements like head, title, body, … etc. These elements are used to build the blocks of web pages.

**<DOCTYPE! html>:** This tag is used to tells the HTML version. This currently tells that the version is HTML 5.

**<html>:** This is called HTML root element and used to wrap all the code.

**<head>:** Head tag contains metadata, title, page CSS etc. All the HTML elements that can be used inside the <head> element are:

* <style>
* <title>
* <base>
* <script>
* <meta>
* <title>

**<body>:** Body tag is used to enclosed all the data which a web page has from texts to links. All of the content that you see rendered in the browser is contained within this element.

**Example:** HTML page can be created using any text editor (notepad). Then save that file using .htm or .html extension and open that file in browser. It will get the HTML page response.

**Features of HTML:**

* It is easy to learn and easy to use.
* It is platform independent.
* Images, video and audio can be added to a web page.
* Hypertext can be added to text.
* It is a mark-up language.

**Why to learn HTML?**

* It is a simple mark-up language. Its implementation is easy.
* It is used to create a website.
* Helps in developing fundamentals about web programming.
* Boost professional career.

**Advantages:**

* HTML is used to build a website.
* It is supported by all browsers.
* It can be integrated with other languages like CSS, JavaScript etc.

**Disadvantages:**

* HTML can create only static webpages so for dynamic web page other languages have to be used.
* Large amount of code has to be written to create a simple web page.
* Security feature is not good.

**9.3 MySQL Introduction:-**

MySQL is an open-source, fast reliable, and flexible relational database

management system, typically used with PHP.

What is a Database?

A database is a separate application that stores a collection of data. Each database has

one or more distinct APIs for creating, accessing, managing, searching and replicating

the data it holds.

Other kinds of data stores can also be used, such as files on the file system or large

hash tables in memory but data fetching and writing would not be so fast and easy with

those type of systems.

Nowadays, we use relational database management systems (RDBMS) to store and

manage huge volume of data. This is called relational database because all the data is

stored into different tables and relations are established using primary keys or other

keys known as Foreign Keys.

A Relational Database Management System (RDBMS) is a software that −

* Enables you to implement a database with tables, columns and indexes.
* Guarantees the Referential Integrity between rows of various tables.
* Updates the indexes automatically.
* Interprets an SQL query and combines information from various tables.

**RDBMS Terminology :-**

Before we proceed to explain the MySQL database system, let us revise a few

definitions related to the database.

* Database − A database is a collection of tables, with related data.
* Table − A table is a matrix with data. A table in a database looks like a simple

spreadsheet.

* Column − One column (data element) contains data of one and the same kind,

for example the column postcode.

* Row − A row (= tuple, entry or record) is a group of related data, for example

the data of one subscription.

* Redundancy − Storing data twice, redundantly to make the system faster.
* Primary Key − A primary key is unique. A key value cannot occur twice in one

table. With a key, you can only find one row.

* Foreign Key − A foreign key is the linking pin between two tables.
* Compound Key − A compound key (composite key) is a key that consists of

multiple columns, because one column is not sufficiently unique.

* Index − An index in a database resembles an index at the back of a book.
* Referential Integrity − Referential Integrity makes sure that a foreign key value always points to an existing row.

What is MySQL?

MySQL is a database system used for developing web-based software applications.

* MySQL used for both small and large applications.
* MySQL is a relational database management system.
* MySQL is fast, reliable, and flexible and easy to use.
* MySQL supports standard SQL.
* MySQL is free to download and use.
* MySQL was developed by Michael Widenius and David Axmark in 1994.
* MySQL is presently developed, distributed, and supported by Oracle Corporation.
* MySQL Written in C, C++.
* Main Features of MySQL
* MySQL server design is multi-layered with independent modules.
* MySQL is fully multithreaded by using kernel threads. It can handle multiple CPUs if

they are available.

* MySQL provides transactional and non-transactional storage engines.
* MySQL has a high-speed thread-based memory allocation system.
* MySQL supports in-memory heap table.