CRYPTOLOCKER

A PROJECT REPORT

Submitted by

PARTHEEBAN D [REGISTER NO:211417104179]
RAHURAMAN N [REGISTER NO:211417104210]
RAM PRASAD S [REGISTER NO:211417104218]

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

PANIMALAR ENGINEERING COLLEGE, CHENNAI-600123.

ANNA UNIVERSITY: CHENNAI 600 025

APRIL 2021

BONAFIDE CERTIFICATE

Certified that this project report **“ CRYPTOLOCKER ”**
is the bonafide work of **“PARTHEEBAN D [REGISTERNO:211417104179], RAHURAMAN N [REGISTER NO:211417104210] , RAM PRASAD S [REGISTER NO:211417104218] ”** who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
Dr.S.MURUGAVALLI,M.E.,Ph.D.,	**Mr.M.SHANMUGANATHAN M.Tech**
HEAD OF THE DEPARTMENT	**ASSISTANT PROFESSOR**
DEPARTMENT OF CSE,	DEPARTMENT OF CSE,
PANIMALAR ENGINEERING COLLEGE,	PANIMALAR ENGINEERING COLLEGE,
NASARATHPETTAI,	NASARATHPETTAI,
POONAMALLEE,	POONAMALLEE,
CHENNAI-600 123.	CHENNAI-600 123.

Certified that the above candidate(s) was/ were examined in the Anna University Project Viva-Voce Examination held on...........................

INTERNAL EXAMINER	**EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

We express our deep gratitude to our respected Secretary and Correspondent **Dr.P.CHINNADURAI, M.A., Ph.D.** for his kind words and enthusiastic motivation, which inspired us a lot in completing this project.

We would like to extend our heartfelt and sincere thanks to our Directors **Tmt.C.VIJAYARAJESWARI**, **Thiru.C.SAKTHIKUMAR,M.E., Ph.D** and **Tmt. SARANYASREE SAKTHIKUMAR B.E.,M.B.A.,** for providing us with the necessary facilities for completion of this project.

We also express our gratitude to our Principal **Dr.K.Mani, M.E., Ph.D.** for his timely concern and encouragement provided to us throughout the course.

We thank the HOD of CSE Department, **Dr. S.MURUGAVALLI , M.E.,Ph.D.,** for the support extended throughout the project.

We would like to thank my **Project Guide Mr. M. SHANMUGANATHAN , M.Tech** and all the faculty members of the Department of CSE for their advice and suggestions for the successful completion of the project.

NAME OF THE STUDENTS

PARTHEEBAN D
RAHURAMAN N
RAM PRASAD S

ABSTRACT

Many systems rely on passwords for authentication. Due to numerous accounts for different services, users have to choose and remember a significant number of files and passwords. Crptolocker applications address this issue by storing user’s files and password. They are especially useful on mobile devices, because of the ubiquitous instant access. Crptolocker often use key derivations functions to convert a master password into cryptographic key suitable for encrypting the list of files, thus protecting the files and password against unauthorized, off-line access. Therefore, design and implementation problems in the key derivation function can render the encryption on the files is useless, by for example allowing efficient brute force attacks, or even worse- direct decryption of the stored files.

LIST OF FIGURES

LIST OF TABLES

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

Day to day, the usage of data in the computer has been increasing from common man to organization. The question arises where to store the important data, how to share the data, how to access the data globally, how to manage the data, how to make data available all the time, how can all these be achieved with reasonable cost? The answer to all these questions is cloud computing. NIST defines Cloud computing as a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction.

BENEFITS OF CLOUD COMPUTING

The factors that make more companies to move cloud are • Reduces the maintenance cost like no need of licensed software fee for each system, the purchase of new hardware and software is reduced. • Access to the application can be done anytime, anywhere provided that they should be connected to internet. • Scalable • Improves Flexibility • Disaster Recovery • As the services are based on ”Pay per use” ,capital expenditure can be reduced • User Friendly Environment • Quick Deployment • Less Energy Consumption

CLOUD SERVICES

The services of the cloud can be classified into the three categories namely Software as a Service, Platform as a Service, Infrastructure as a Service. All the services are based upon the “Pay-per-use” model. Software as Service: In SaaS, an application is hosted by service provider and then accessed via the world wide web by a client. These are mainly designed for end users. Customers need not install the application on the local computer there by eliminating installation and maintenance cost. The updating of software is taken care by the SaaS provider. Most of the SaaS solutions belong to multitenant architecture. As the software is managed at central location, customer can access to the application at any time and place, the only thing required is access to the web. Some of the SaaS providers are Google Apps, Quickbook overview, Microsoft Office live Business, Amazon, Linkedln, Workday,Netsuite. Usage of SaaS is beneficiary when there is significant need for mobile or web access like mobile sales management software, significant interplay between organization and outside world like email, applications like tax or billing software used once in a month. Platform as a Service: With this kind of servicing facilities, provided, one can deploy the application without installing the platform on the local system that is software can be deployed in cloud infrastructure. The main benefit of using PaaS is that developer need not worry about the platform updates, storage. These features are taken by PaaS providers. Some PaaS providers provide prebuilt functionality so that users can avoid building everything from the scratch. Some of the PaaS providers also provide online community where developers can share best practices can get ideas, seek advice from others. The implementation of PaaS is different from one provider to another provider. Amazon webservices, Appistry, Appscale, Google, OpenStack, Flexiscale, Long Jump are some of the PaaS providers. Infrastructure as a Service: Unlike SaaS and PaaS, IaaS provide hardware resources as service. The resources include memory, servers, networking devices, processing power. These are used to deploy the application. Multiple users can use infrastructure through the use of virtual machines. In order to manage these virtual machines, a governance framework is required, which helps in avoiding uncontrolled access to the users sensitive information. Utilization of this service will help in reducing the initial investment in company’s hardware. The service is based on “pay-peruse” model. Amazon Web Services EC2 and S3 are best examples for IaaS.

DEPLOYMENT MODEL.

The Cloud services can be deployed in any one of the four following ways depending on the customer requirement. Each model has its advantages and disadvantages 1. Public Cloud: In this model, general public can access the services, storage, application offered by the provider. Pubic clouds are owned and managed by the third- party service providers. Flexibility, elastic environment, freedom of self service, pay-per-use, availability, reliability are some of the characteristics of public cloud. The main drawback of this model is lack of high level security. Ex: Amazon Elastic Cloud Compute, Google App Engine, Blue Cloud by IBM. 2. Private Cloud: This model provides access to the systems and services within an organization. Data stored in private cloud can only be shared among the users of an organization. There are two types of private cloud namely, On-Premise Private Cloud, Externally-Hosted Private Cloud .The disadvantage of this model is ,it is difficult to deploy globally. Amazon Virtual Private Cloud, Microsoft Private Cloud are some of the examples of this model. 3. Hybrid Cloud: It is the combination of both public and private cloud. Scalability, cost efficiency, Security, Flexibility are the features of Hybrid cloud. 4. Community Cloud: Organizations with similar interest and requirements share the cloud infrastructure. It provides better security when compared to public cloud. This may be managed by either internally or third party

CLOUD STORAGE

Cloud storage is a service that maintains data, manage and backup remotely and made data available to users over the network. There are many cloud storage providers. Most of the providers provide free space up to certain gigabytes. For ex: DropBox provide free space up to 2GB, Google Drive, Box, Amazon, Apple Cloud provide free space up to 5GB, Microsoft SkyDrive provide free space up to 7GB.Customer have to pay amount according to the plan if they cross the free space limit. Features like maximum file size, auto backup, bandwidth, upgrade for limited space differ from one provider to another provider like maximum file size in DropBox is 300MB where as maximum file size in Google Drive is 1TB.

1.2 PROBLEM DEFINITION

Even though Security, Privacy and Trust issues exists since the evolution of Internet, the reason why they are widely spoken these days is because of the Cloud Computing scenario. Any client/small organization/enterprise that processes data in the cloud is subjected to an inherent level of risk because outsourced services bypass the "physical, logical and personnel controls" of the user . When storing data on cloud, one might want to make sure if the data is correctly stored and can be retrieved later. As the amount of data stored by the cloud for a client can be enormous, it is impractical (and might also be very costly) to retrieve all the data, if one’s purpose is just to make sure that it is stored correctly. Hence there is a need to provide such guarantees to a client. Hence, it is very important for both the cloud provider and the user to have mutual trust such that the cloud provider can be assured that the user is not some malicious hacker and the user can be assured of data consistency, data storage and the instance he/she is running is not malicious. Hence the necessity for developing trust models/protocols is demanding.

CHAPTER 2

LITERATURE SURVEY

1. M.Lakshmi Neelima et al, International Journal of Computer Science and Mobile Computing

This paper presents the key technologies and virtual storage architecture in cloud. Cloud storage is more advantageous than traditional storage because of its availability, scalability, performance, portability and its functional requirements. Implementing virtualization in the cloud storage improves the scalability, availability but at the same time providing security in the virtual environment is complex. So apart from virtualization, emphasis should be given regarding security in virtual storage.

2. Research on Cloud Data Storage Technology and Its Architecture Implementation Kun Liua, Long-jiang Dong 2012 International Workshop on Information and Electronics Engineering (IWIEE)

Cloud computing is the inevitable product with the development of the internet, and it also brings more rich applications to the internet. Cloud data storage technology is the core area in cloud computing and solves the data storage mode of cloud environment. In this paper, we introduce the related concepts of cloud computing and cloud storage. Then we pose a cloud storage architecture based on eyeOS web operating system in our computers. Experiments verified the system is well.

3. Cachin, C., Keidar, I., and Shraer , A. Trusting the cloud. ACM SIGACT News, 20:4 (2009).

Though clouds are becoming increasingly popular, we have seen that some things can “go wrong” when one trusts a cloud provider with his data. Providing defenses for these is an active area of research. We presented a brief survey of solutions being proposed in this context. Nevertheless, these solutions are, at this point in time, academic. There are still questions regarding how well these protections can work in practice, and moreover, how easy-to-use they can be. Finally, we have yet to see how popular storing data in clouds will become, and what protections users will choose to use, if any.

4. Security Issues In Cloud Computing, Kevin Curran

One of the biggest security worries with cloud computing model is sharing of resources. Cloud service providers need to inform their existing customers on the level of security that they provide on their cloud. The cloud service providers need to educate potential customers about the cloud deployment models such as public, private and hybrid along with the pros and cons of each. They need to show their customers that they are providing appropriate security measures that will protect their customer's data and build up confidence for their service. One way they can achieve this is through the use of third party auditors. New security techniques needed to be radically tweaked to be able to work with cloud architecture.

CHAPTER 3

SYSTEM ANALYSIS

3.1EXISTING SYSTEM

The existing system has around 100K downloads many people have found useful. The existing system similar but with some outdated features and outdated algorithm to upload and download the credentials. The existing system does not have one app for all system support.

3.2PROPOSED SYSTEM

This application will help the user who travel and need access to their files and password. The highly secured cloud storage ensures instant access to their files and password anywhere from the world. It makes user comfortable in remembering their passwords and storing important files without the knowledge of other user's.

3.3 FEASIBILITY STUDY

A feasibility study is carried out to select the best system that meets performance requirements. The main aim of the feasibility study activity is to determine that it would be financially and technically feasible to develop the product.

3.3.1 TECHNICAL FEASIBILITY

This is concerned with specifying the software will successfully satisfy the user requirement. Open source and business-friendly and it is truly cross platform, easily deployed and highly extensible.

3.3.2 ECONOMIC FEASIBILITY

Economic analysis is the most frequently used technique for evaluating the effectiveness of a proposed system. The enhancement of the existing system doesn't incur any kind of drastic increase in the expenses. Java is open source and ready available for all users. Since the project is run in Java and Firebase hence is cost efficient.

3.4 HARWARE REQUIREMENTS

Processor	: Intel Pentium Dual Core 2.00GHz
Hard disk	: 40 GB
RAM	: 2 GB (minimum)

3.5 SOFTWARE REQUIREMENTS

- Android Studio (IDE)
- Java
- FireBase

3.6 SOFTWARE SPECIFICATION

3.6.1 SYMMENTRIC KEY ENCRYPTION

Symmentric encryption is a type of encryption where only one key (a secret key) is used to both encrypt and decrypt electronic information. The entities communicating via symmetric encryption must exchange the key so that it can be used in the decryption process. This encryption method differs from asymmetric encryption where a pair of keys, one public and one private, is used to encrypt and decrypt messages.

By using symmetric encryption algorithms, data is converted to a form that cannot be understood by anyone who does not possess the secret key to decrypt it. Once the intended recipient who possesses the key has the message, the algorithm reverses its action so that the message is returned to its original and understandable form. The secret key that the sender and recipient both use could be a specific password/code or it can be random string of letters or numbers that have been generated by a secure random number generator (RNG). For banking-grade encryption, the symmetric keys must be created using an RNG that is certified according to industry standards, such as FIPS 140-2.

There are two types of symmetric encryption algorithms:

1. **Block algorithms.** Set lengths of bits are encrypted in blocks of electronic data with the use of a specific secret key. As the data is being encrypted, the system holds the data in its memory as it waits for complete blocks.

2. **Stream algorithms.** Data is encrypted as it streams instead of being retained in the system’s memory.

While symmetric encryption is an older method of encryption, it is faster and more efficient than asymmetric encryption, which takes a toll on networks due to performance issues with data size and heavy CPU use. Due to the better performance and faster speed of symmetric encryption (compared to asymmetric), symmetric cryptography is typically used for bulk encryption / encrypting large amounts of data, e.g. for database encryption. In the case of a database, the secret key might only be available to the database itself to encrypt or decrypt.

Some examples of where symmetric cryptography is used are:

- Payment applications, such as card transactions where PII needs to be protected to prevent identity theft or fraudulent charges
- Validations to confirm that the sender of a message is who he claims to be
- Random number generation or hashing

3.6.2 ADVANCED ENCRYPTION STANDARD

AES is a symmetric key cipher. This means the same secret key is used for both encryption and decryption, and both the sender and receiver of the data need a copy of the key. By contrast, asymmetric key systems use a different key for each of the two processes. Asymmetric keys are best for external file transfers, whereas symmetric keys are better suited to internal encryption. The advantage of symmetric systems like AES is their speed. Because a symmetric key algorithm requires less computational power than an asymmetric one, it’s faster and more efficient to run.

AES is also characterized as a block cipher. In this type of cipher, the information to be encrypted (known as plaintext) is divided into sections called blocks. AES uses a 128-bit block size, in which data is divided into a four-by-four array containing 16 bytes. Since there are eight bits per byte, the total in each block is 128 bits. The size of the encrypted data remains the same: 128 bits of plaintext yields 128 bits of ciphertext.

How does AES work? The basic principle of all encryption is that each unit of data is replaced by a different one according to the security key. More specifically, AES was designed as a substitution-permutation network. AES brings additional security because it uses a key expansion process in which the initial key is used to come up with a series of new keys called round keys. These round keys are generated over multiple rounds of modification, each of which makes it harder to break the encryption.

First, the initial key is added to the block using an XOR ("exclusive or") cipher, which is an operation built into processor hardware. Then each byte of data is substituted with another, following a predetermined table. Next, the rows of the 4x4 array are shifted: bytes in the second row are moved one space to the left, bytes in the third row are moved two spaces, and bytes in the fourth are moved three. The columns are then mixed—a mathematical operation combines the four bytes in each column. Finally, the round key is added to the block (much like the initial key was), and the process is repeated for each round. This yields ciphertext that is radically different from the plaintext. For AES decryption, the same process is carried out in reverse.

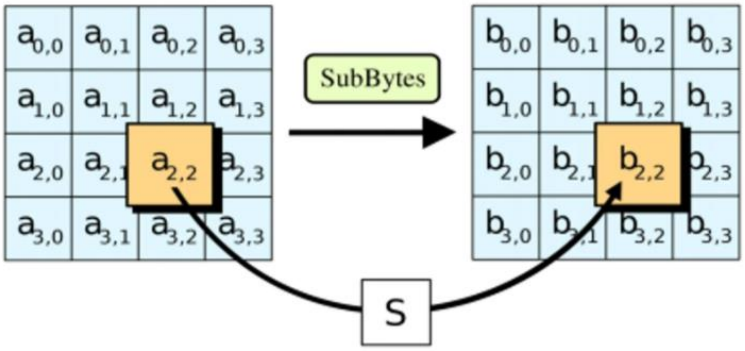
Each stage of the AES encryption algorithm serves an important function. Using a different key for each round provides a much more complex result. Byte substitution modifies the data in a nonlinear manner, obscuring the relationship between the original and encrypted content. Shifting the rows and mixing the columns diffuses the data, transposing bytes to further complicate the encryption. Shifting diffuses the data horizontally, while mixing does so vertically. The result is a tremendously sophisticated form of encryption.

3.6.3 STEPS IN EACH ROUND

Each round in the algorithm consists of four steps.

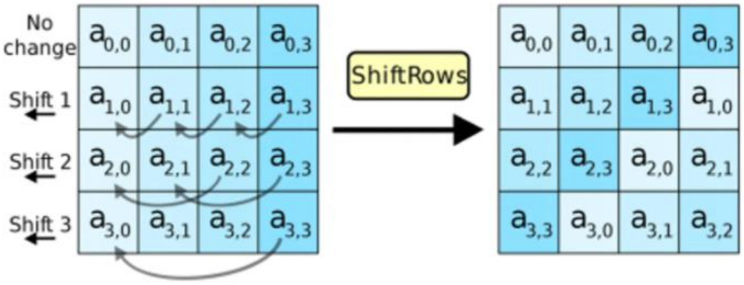
1. Substitution of Bytes

In the first step, the bytes of the block text are substituted based on rules dictated by predefined S-boxes (short for substitution boxes).


$a_{0,0}$ $a_{0,1}$ $a_{0,2}$ $a_{0,3}$
$a_{1,0}$ $a_{1,1}$ $a_{1,2}$ $a_{1,3}$
$a_{2,0}$ $a_{2,1}$ $a_{2,2}$ $a_{2,3}$
$a_{3,0}$ $a_{3,1}$ $a_{3,2}$ $a_{3,3}$
SubBytes
$b_{0,0}$ $b_{0,1}$ $b_{0,2}$ $b_{0,3}$
$b_{1,0}$ $b_{1,1}$ $b_{1,2}$ $b_{1,3}$
$b_{2,0}$ $b_{2,1}$ $b_{2,2}$ $b_{2,3}$
$b_{3,0}$ $b_{3,1}$ $b_{3,2}$ $b_{3,3}$
S

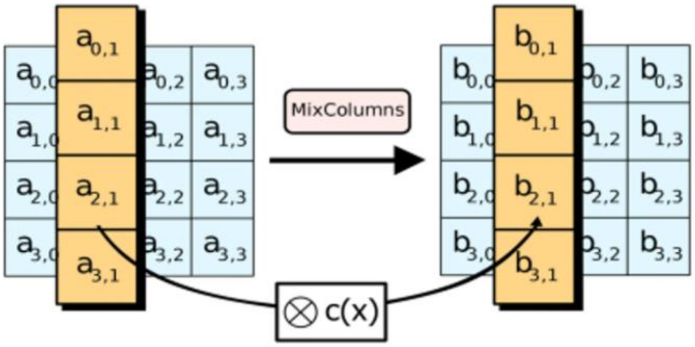
2. Shifting the rows

Next comes the permutation step. In this step, all rows except the first are shifted by one, as shown below.


No change
Shift 1
Shift 2
Shift 3
$a_{0,0}$ $a_{0,1}$ $a_{0,2}$ $a_{0,3}$
$a_{1,0}$ $a_{1,1}$ $a_{1,2}$ $a_{1,3}$
$a_{2,0}$ $a_{2,1}$ $a_{2,2}$ $a_{2,3}$
$a_{3,0}$ $a_{3,1}$ $a_{3,2}$ $a_{3,3}$
ShiftRows
$a_{0,0}$ $a_{0,1}$ $a_{0,2}$ $a_{0,3}$
$a_{1,1}$ $a_{1,2}$ $a_{1,3}$ $a_{1,0}$
$a_{2,2}$ $a_{2,3}$ $a_{2,0}$ $a_{2,1}$
$a_{3,3}$ $a_{3,0}$ $a_{3,1}$ $a_{3,2}$

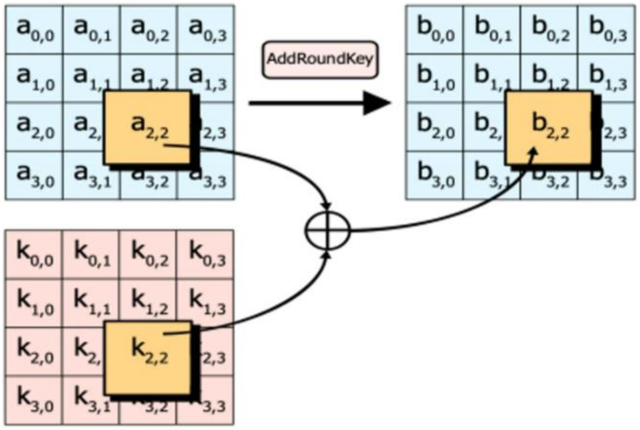
3. Mixing the columns

In the third step, the Hill cipher is used to jumble up the message more by mixing the block’s columns.



4. Adding the round key

In the final step, the message is XORed with the respective round key.



3.6.4 IN TERMS OF SECURITY

The National Institute of Standards and Technology selected three “flavors” of AES: 128-bit, 192-bit, and 256-bit. Each type uses 128-bit blocks. The difference lies in the length of the key. As the longest, the 256-bit key provides the strongest level of encryption. With a 256-bit key, a hacker would need to try 2^{256} different combinations to ensure the right one is included. This number is astronomically large, landing at 78 digits total. It is exponentially greater than the number of atoms in the observable universe. Understandably, the US government requires 128- or 256-bit encryption for sensitive data.

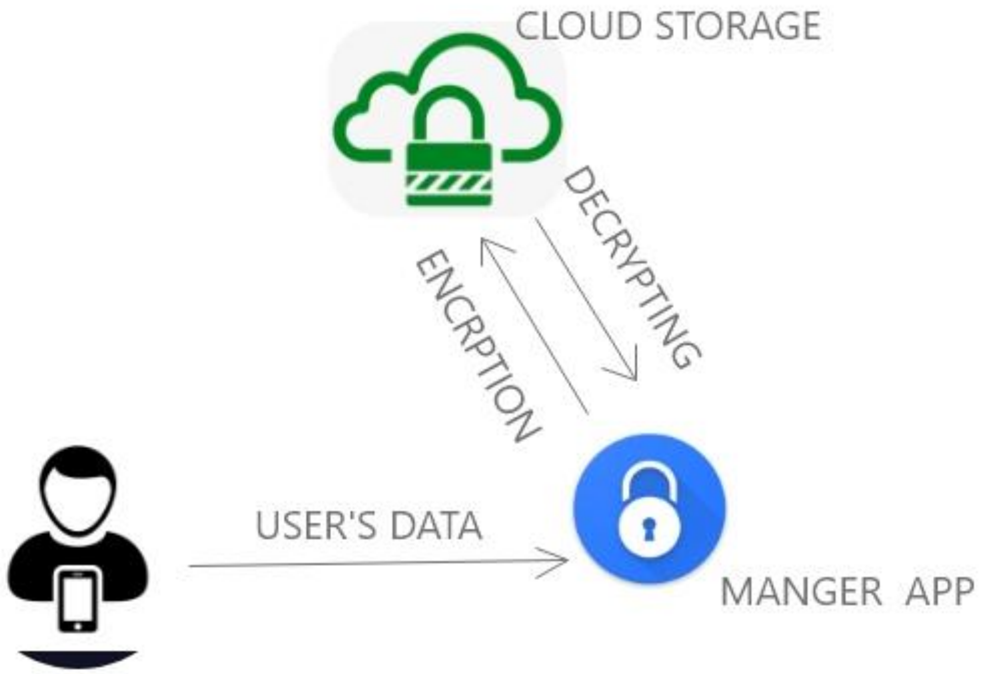
The three AES varieties are also distinguished by the number of rounds of encryption. AES 128 uses 10 rounds, AES 192 uses 12 rounds, and AES 256 uses 14 rounds. The more rounds, the more complex the encryption, making AES 256 the most secure AES implementation. It should be noted that with a longer key and more rounds comes higher performance requirements. AES 256 uses 40% more system resources than AES 192, and is therefore best suited to high sensitivity environments where security is more important than speed.

CHAPTER 4

ARCHITECTURE

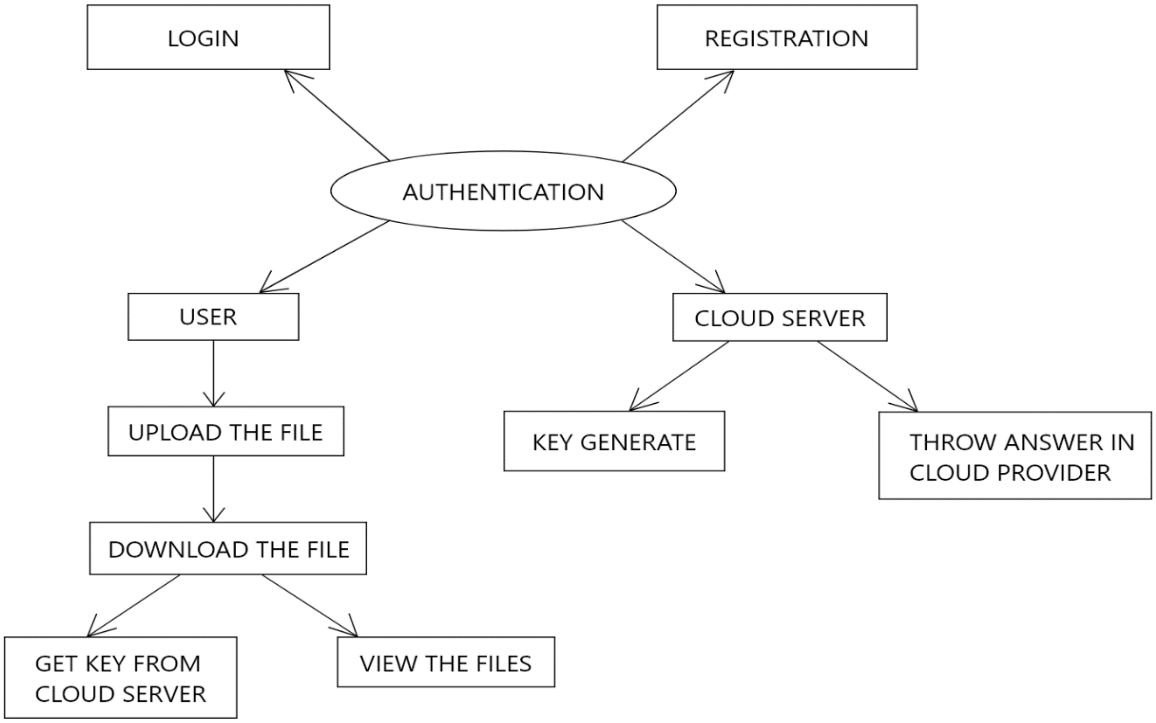
4.1 SYSTEM ARCHITECTURE

System architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal encryption and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system.



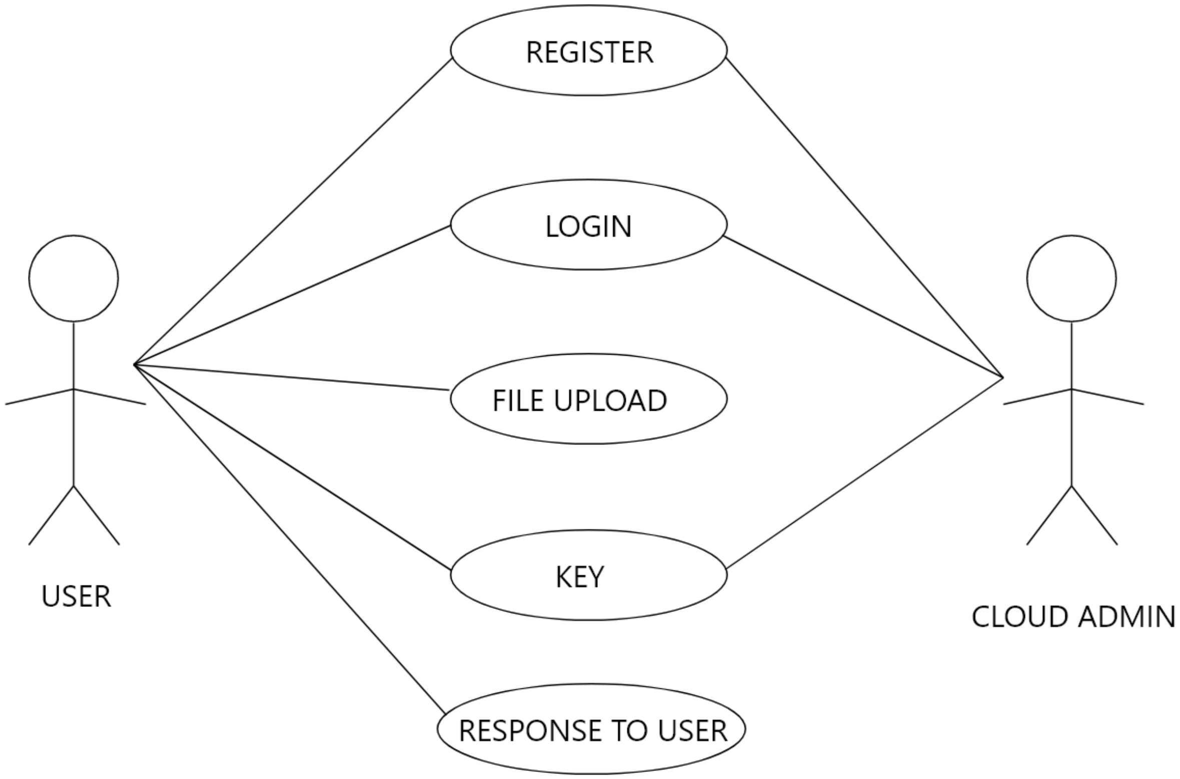
4.2 ER DIAGRAM

An entity–relationship model describes interrelated things of interest in a specific domain of knowledge. A basic ER model is composed of entity types and specifies relationships that can exist between entities.



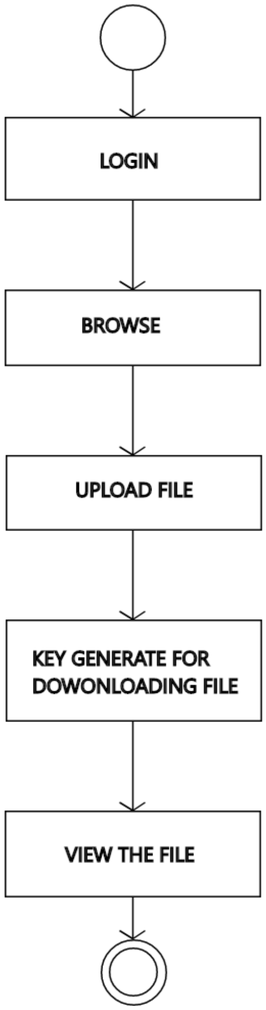
4.3 USECASE DIAGRAM

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved.



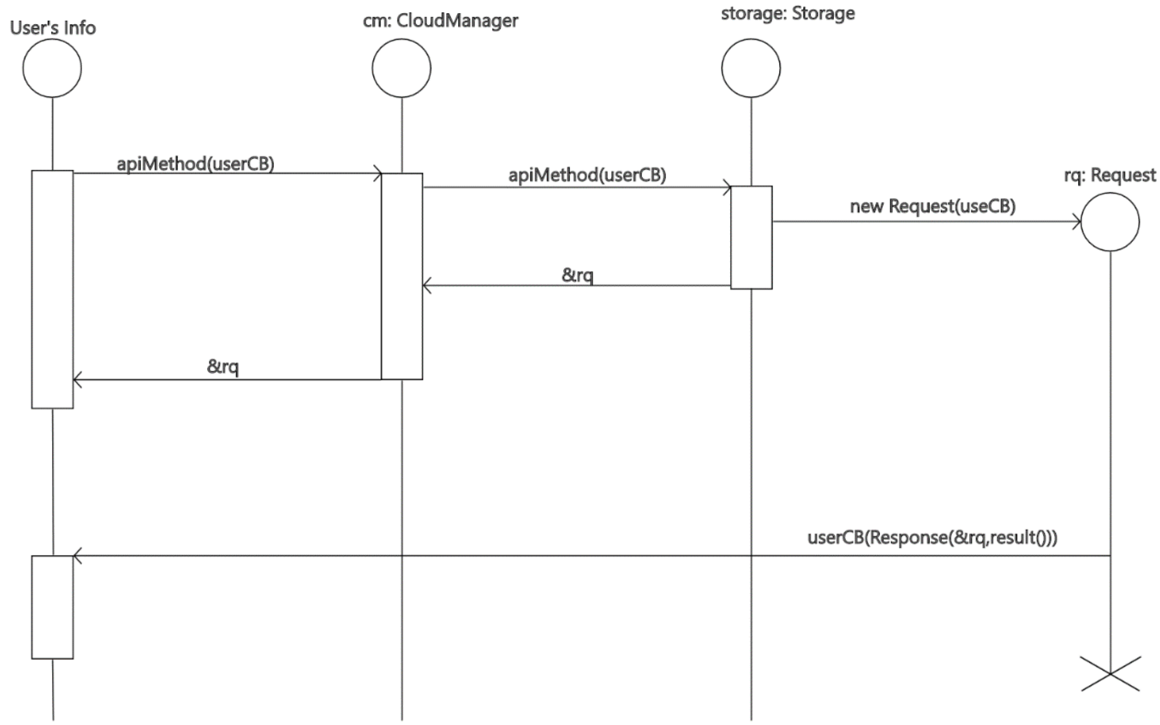
4.4 ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency.



4.5 SEQUENCE DIAGRAM

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.



4.6 DATA DICTIONARY

A data dictionary, or metadata repository, as defined in the IBM Dictionary of Computing, is a "centralized repository of information about data such as meaning, relationships to other data, origin, usage, and format". Oracle defines it as a collection of tables with metadata. The term can have one of several closely related meanings pertaining to databases and database management systems (DBMS):

4.6.1 USER REGISTRATION

COLUMN NAME	**DATA TYPE**
E-MAIL ID	VARCHAR(100)
USER NAME	VARCHAR(100)
PASSWORD	VARCHAR(100)
CONFIRM PASSWORD	VARCHAR(100)
ADDRESS	VARCHAR(100)
ZIP CODE	VARCHAR(100)
MOBILE NO	VARCHAR(100)

4.6.2 FILES

COLUMN NAME	**DATA TYPE**
ID	INT
USER NAME	VARCHAR(50)
NAME	VARCHAR(100)
CONTENT TYPE	VARCHAR(50)
SIZE	INT
DATA	VARBINARY(MAX)

CHAPTER 5

SYSTEM MODULE

5.1 MODULE

- Data Processing
- Encryption of Data
- Secure Retrieval of Encrypted data

5.2 MODULE DESCRIPTION

Data Processing

Processing of data based on the user’s input (i.e. numerical, text) are categorized using this module and encrypted by the Encryption Of Data module and then stored in the cloud.

Encryption of Data

This step involves: The data entered by the user is now encrypted by the algorithm which is highly secure and encrypted as soon as the user save the credentials from his side, it is then stored in the cloud in the encrypted form and it can only retrieved and decrypted only when the user decides to view his saved credential from the cloud.

Secure Retrieval of Encrypted data

This process is a continuation of the previous process, this involves decryption of the data that had been stored by the user in the cloud. Since the process being an decryption of the highly secure data of user it asks for the key from the user before the decryption of the data from the cloud. Once the authentication process is over the data is retrieved from the cloud and the displayed to the user.

CHAPTER 6

SYSTEM DESIGN

6.1 Algorithm:

1. Initialize the state array with the block data (plaintext).
2. Add the initial round key to the starting state array.
3. Perform nine rounds of state manipulation.
4. Perform the tenth and final round of state manipulation.
5. Copy the final state array out as the encrypted data

Working Of AES 256

AES uses symmetric key encryption, which involves the use of only one secret key to cipher and decipher information. AES 256 which has a key length of 256 bits, supports the largest bit size and is practically unbreakable by brute force based on current computing power, making it the strongest encryption standard.

Source Code:

AES 256

```
package com.example.cryptolocker;

import android.util.Base64;

import javax.crypto.Cipher;
import javax.crypto.SecretKey;
import javax.crypto.SecretKeyFactory;
import javax.crypto.spec.IvParameterSpec;
import javax.crypto.spec.PBEKeySpec;
import javax.crypto.spec.SecretKeySpec;

import java.nio.charset.StandardCharsets;
import java.security.spec.KeySpec;
import java.util.ArrayList;

public class Aes256 {

    private static final String SALT = "mpQnAhyeeaPFijbG1tkJzTukvtUVfq1W";


    public static ArrayList<String> encrypt(String strToEncrypt1, String strToEncrypt2, String
strToEncrypt3, String strToEncrypt4, String SECRET_KEY) {
        try {
            ArrayList<String> encryptionData = new ArrayList<>();
            byte[] iv = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
            IvParameterSpec ivspec = new IvParameterSpec(iv);

            SecretKeyFactory factory =
SecretKeyFactory.getInstance("PBKDF2WithHmacSHA256");
            KeySpec spec = new PBEKeySpec(SECRET_KEY.toCharArray(), SALT.getBytes(),
65536, 256);
            SecretKey tmp = factory.generateSecret(spec);
            SecretKeySpec secretKey = new SecretKeySpec(tmp.getEncoded(), "AES");

            Cipher cipher = Cipher.getInstance("AES/CBC/PKCS5Padding");
            cipher.init(Cipher.ENCRYPT_MODE, secretKey, ivspec);


encryptionData.add(Base64.encodeToString(cipher.doFinal(strToEncrypt1.getBytes(StandardCh
arsets.UTF_8)), Base64.DEFAULT));

encryptionData.add(Base64.encodeToString(cipher.doFinal(strToEncrypt2.getBytes(StandardCh
arsets.UTF_8)), Base64.DEFAULT));

encryptionData.add(Base64.encodeToString(cipher.doFinal(strToEncrypt3.getBytes(StandardCh
arsets.UTF_8)), Base64.DEFAULT));

encryptionData.add(Base64.encodeToString(cipher.doFinal(strToEncrypt4.getBytes(StandardCh
arsets.UTF_8)), Base64.DEFAULT));
            return encryptionData;

        } catch (Exception e) {
            System.out.println("Error while encrypting: " + e.toString());
        }
        return null;
    }

    public static ArrayList<String> decrypt(String strToDecrypt1, String strToDecrypt2, String
strToDecrypt3, String strToDecrypt4, String SECRET_KEY) {
        try {

            ArrayList<String> decryptionData = new ArrayList<>();
            byte[] iv = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
            IvParameterSpec ivspec = new IvParameterSpec(iv);

            SecretKeyFactory factory =
SecretKeyFactory.getInstance("PBKDF2WithHmacSHA256");
            KeySpec spec = new PBEKeySpec(SECRET_KEY.toCharArray(), SALT.getBytes(),
65536, 256);
            SecretKey tmp = factory.generateSecret(spec);
            SecretKeySpec secretKey = new SecretKeySpec(tmp.getEncoded(), "AES");

            Cipher cipher = Cipher.getInstance("AES/CBC/PKCS5PADDING");
            cipher.init(Cipher.DECRYPT_MODE, secretKey, ivspec);

            decryptionData.add(new String(cipher.doFinal(Base64.decode(strToDecrypt1,
Base64.DEFAULT))));
            decryptionData.add(new String(cipher.doFinal(Base64.decode(strToDecrypt2,
Base64.DEFAULT))));
            decryptionData.add(new String(cipher.doFinal(Base64.decode(strToDecrypt3,
Base64.DEFAULT))));
            decryptionData.add(new String(cipher.doFinal(Base64.decode(strToDecrypt4,
Base64.DEFAULT))));

            return decryptionData;
        } catch (Exception e) {
            System.out.println("Error while decrypting: " + e.toString());
        }
        return null;
    }
}
```

6.2 Client-side Coding

Create Activity

Fragment_Create.xml

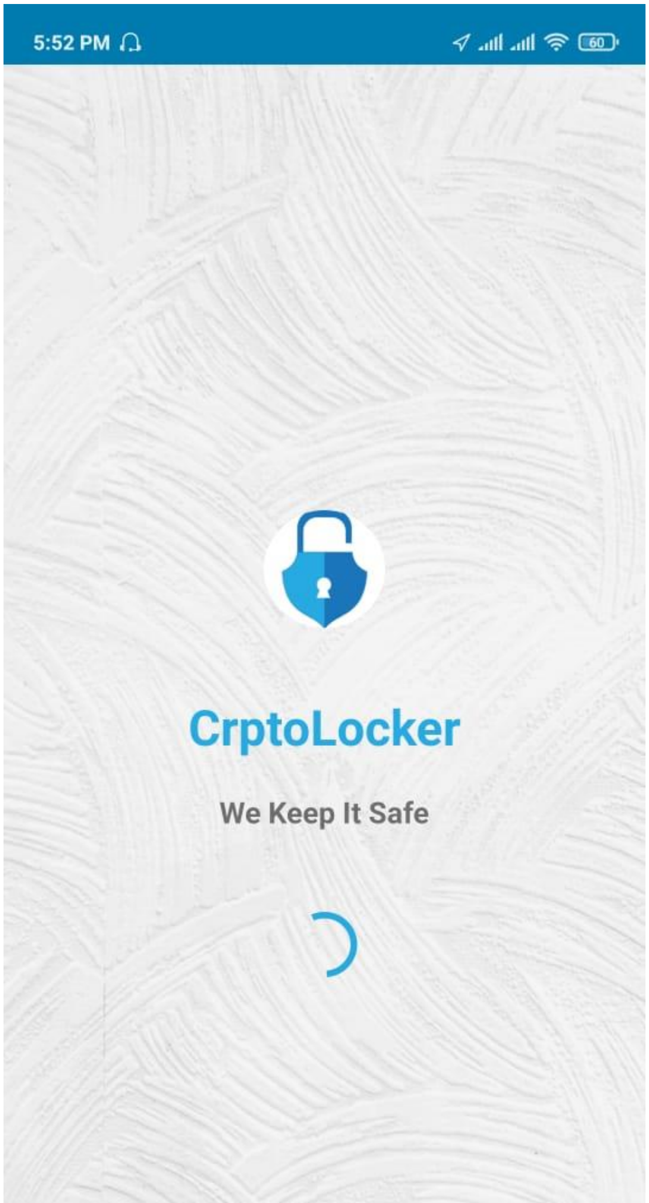
```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".ui.create.CreateFragment">

    <ImageView
        android:id="@+id/imageViewCreate"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="30dp"
        android:layout_marginTop="40dp"
        android:src="@drawable/icons8_create" />

    <TextView
        android:id="@+id/textview_create"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginTop="40dp"
        android:layout_marginEnd="8dp"
        android:layout_toEndOf="@+id/imageViewCreate"
        android:textSize="20dp"
        android:text="Create Record"
        android:textStyle="bold"/>

    <Spinner
        android:id="@+id/spinnerCategory"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="50dp"
        android:layout_marginHorizontal="20dp"
        android:background="@drawable/spinner_bg_layout"
        android:layout_below="@+id/textview_create"

        />
    <Button
        android:id="@+id/buttonSave"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="35dp"
        android:layout_marginHorizontal="40dp"
        android:layout_alignParentEnd="true"
        android:layout_toEndOf="@id/textview_create"
        android:text="Save"/>

    <EditText
        android:id="@+id/editTextTitle"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="50dp"
        android:layout_below="@+id/spinnerCategory"
        android:layout_marginHorizontal="20dp"
        android:hint="Title"
        android:inputType="textPersonName"
        android:drawableLeft="@drawable/icons8_edit_96"
        android:drawablePadding="30dp"/>
    <EditText
        android:id="@+id/editTextSubTitle1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="50dp"
        android:layout_below="@+id/editTextTitle"
        android:layout_marginHorizontal="20dp"
        android:ems="10"
        android:hint="SubTitle1"
        android:inputType="textPersonName"
        android:drawableLeft="@drawable/icons8_user_96"
        android:drawablePadding="30dp"/>

    <EditText
        android:id="@+id/editTextSubTitle2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="50dp"
        android:layout_below="@+id/editTextSubTitle1"
        android:layout_marginHorizontal="20dp"
        android:ems="10"
        android:hint="SubTitle2"
        android:inputType="textPassword"
        android:drawableLeft="@drawable/icons8_lock"
        android:drawableRight="@drawable/eye"
        android:drawablePadding="30dp" />
</RelativeLayout>
```

6.3 Server-side Coding

CreateFragment.java

```
package com.example.cryptolocker.ui.create;

import android.os.Bundle;
import android.text.TextUtils;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Spinner;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.fragment.app.Fragment;

import com.example.cryptolocker.Aes256;
import com.example.cryptolocker.Home;
import com.example.cryptolocker.R;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;

import java.util.ArrayList;
import java.util.Calendar;
import java.util.Date;

public class CreateFragment extends Fragment {
    EditText editTextTitle,editTextSubTitle1,editTextSubTitle2;
    Button buttonSave;
    Spinner spinnerCategory;
    String category,title, subtitle1, subtitle2;
    int spinner_category_position;

    FirebaseAuth firebaseAuth;
    FirebaseUser user;
    DatabaseReference databaseReference;

    Date currentTime;
    ArrayList<String> encryptedData;
    public View onCreateView(@NonNull LayoutInflater inflater,
                             ViewGroup container, Bundle savedInstanceState) {
        View root = inflater.inflate(R.layout.fragment_create, container, false);
        editTextTitle = root.findViewById(R.id.editTextTitle);
        editTextSubTitle1 = root.findViewById(R.id.editTextSubTitle1);
        editTextSubTitle2 = root.findViewById(R.id.editTextSubTitle2);

        buttonSave = root.findViewById(R.id.buttonSave);

        spinnerCategory=root.findViewById(R.id.spinnerCategory);

        String[] categoryArray={"Select Category","Social Media","Bank","Custom"};
        ArrayAdapter<String> categoryAdapter=new
ArrayAdapter<String>(getActivity(),R.layout.spinner_item_category,R.id.spinner_textview_category,categoryArray);
        spinnerCategory.setAdapter(categoryAdapter);
        firebaseAuth= FirebaseAuth.getInstance();
        user=firebaseAuth.getCurrentUser();
        databaseReference= FirebaseDatabase.getInstance().getReference();

        Toast.makeText(getActivity(), String.valueOf(currentTime),
Toast.LENGTH_SHORT).show();

        spinnerCategory.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {
            @Override
            public void onItemSelected(AdapterView<?> adapterView, View view, int i, long l) {
                spinner_category_position= spinnerCategory.getSelectedItemPosition();
                if (spinner_category_position==1){
                    editTextTitle.setHint("Account Type");
                    editTextSubTitle1.setHint("UserName");
                    editTextSubTitle2.setHint("Password");
                }
                else if (spinner_category_position==2){
                    editTextTitle.setHint("Bank Name");
                    editTextSubTitle1.setHint("Account Number");
                    editTextSubTitle2.setHint("Pin");
                }
                else
                {
                    editTextTitle.setHint("Title");
                    editTextSubTitle1.setHint("SubTitle1");
                    editTextSubTitle2.setHint("SubTitle2");
                }

            }

            @Override
            public void onNothingSelected(AdapterView<?> adapterView) {

            }
        });

        buttonSave.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                currentTime = Calendar.getInstance().getTime();
                saveData_To_Firebase();
            }
        });

        return root;
    }
    private void saveData_To_Firebase() {
        title=String.valueOf(editTextTitle.getText());
        subtitle1 =String.valueOf(editTextSubTitle1.getText());
        subtitle2 =String.valueOf(editTextSubTitle2.getText());

        int spinner_category_position = spinnerCategory.getSelectedItemPosition();

        if (spinner_category_position == 0)
        {
            Toast.makeText(getActivity(), "Please select category", Toast.LENGTH_SHORT).show();
            return;
        }
        else
        {
            category=computeCategory(spinner_category_position);
        }
        encryptedData=Aes256.encrypt(title,subtitle1,subtitle2,category,user.getUid());

        title=encryptedData.get(0);
        subtitle1=encryptedData.get(1);
        subtitle2=encryptedData.get(2);
        category=encryptedData.get(3);

        if (TextUtils.isEmpty(title)||TextUtils.isEmpty(subtitle1)||TextUtils.isEmpty(subtitle2))
        {
            Toast.makeText(getActivity(), "Please enter credentials",
Toast.LENGTH_SHORT).show();
            return;
        }

        Home home=new Home(title, subtitle1, subtitle2,category);

databaseReference.child("Home").child(user.getUid()).child(String.valueOf(currentTime)).setVa
lue(home);

        Toast.makeText(getActivity(), "Data saved to firebase", Toast.LENGTH_SHORT).show();
    }

    private String computeCategory(int spinner_category_position) {
        String category="";
        if(spinner_category_position==1)
        {
            category="social media";
        }
        else if (spinner_category_position==2)
        {
            category="bank";
        }
        else
        {
            category="custom";
        }
        return category;
    }
}
```

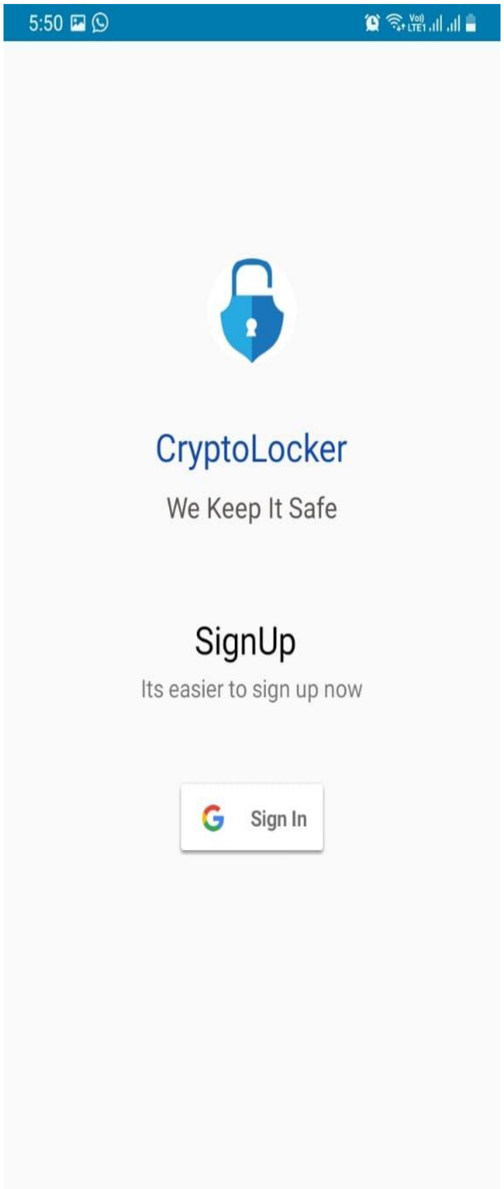
6.4 SCREENSHOTS AND OUTPUTS



5:52 PM

CrptoLocker

We Keep It Safe



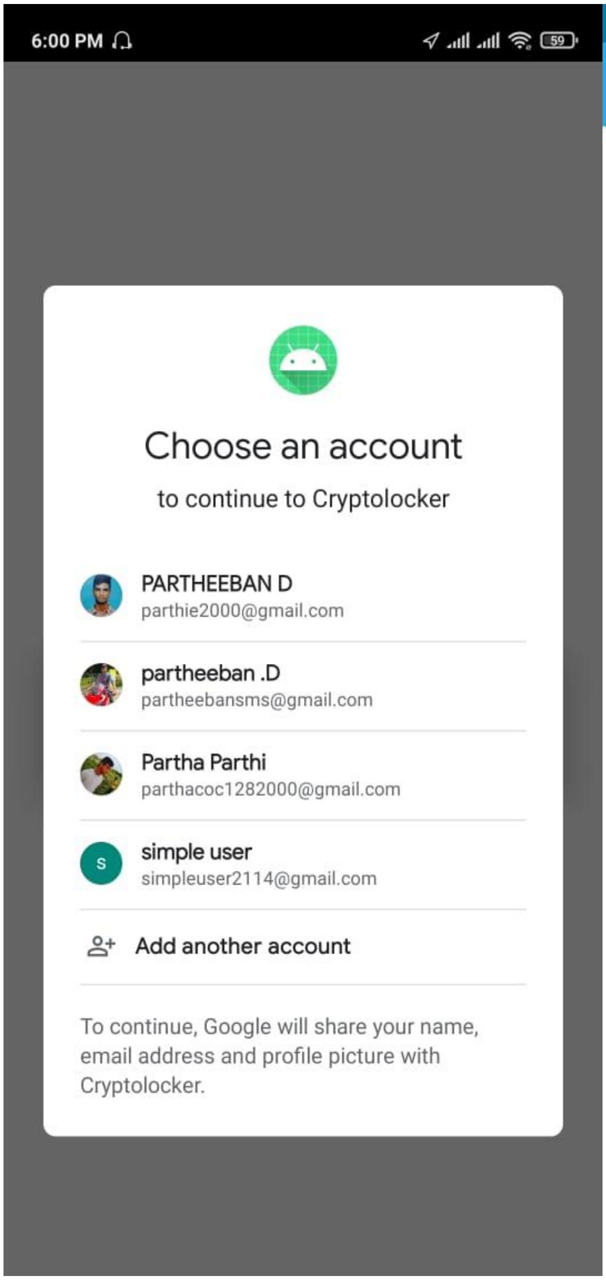
5:50

CryptoLocker

We Keep It Safe

SignUp

Its easier to sign up now



Choose an account

to continue to Cryptolocker

PARTHEEBAN D
parthie2000@gmail.com

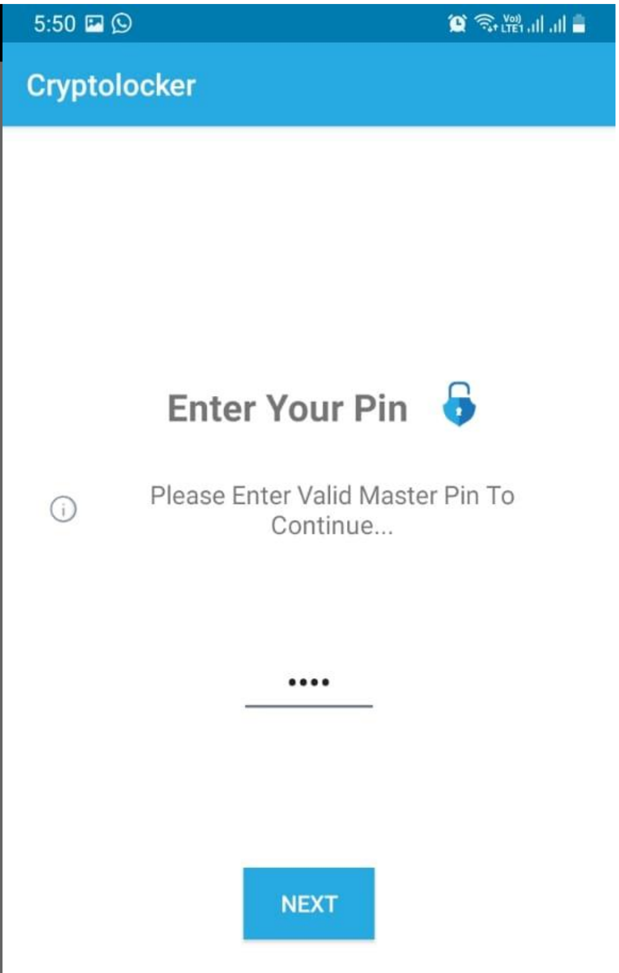
partheeban .D
partheebansms@gmail.com

Partha Parthi
parthacoc1282000@gmail.com

simple user
simpleuser2114@gmail.com

Add another account

To continue, Google will share your name, email address and profile picture with Cryptolocker.

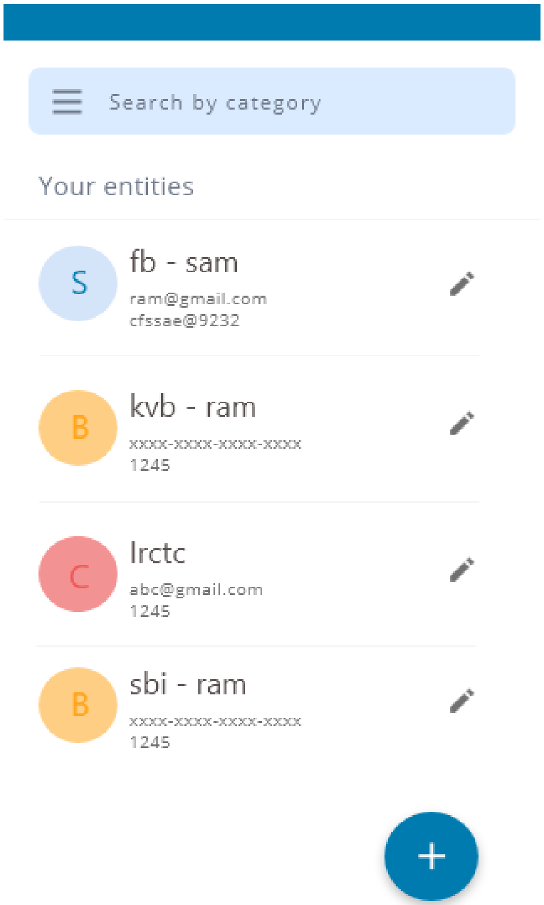


Cryptolocker

Enter Your Pin

Please Enter Valid Master Pin To Continue...

••••



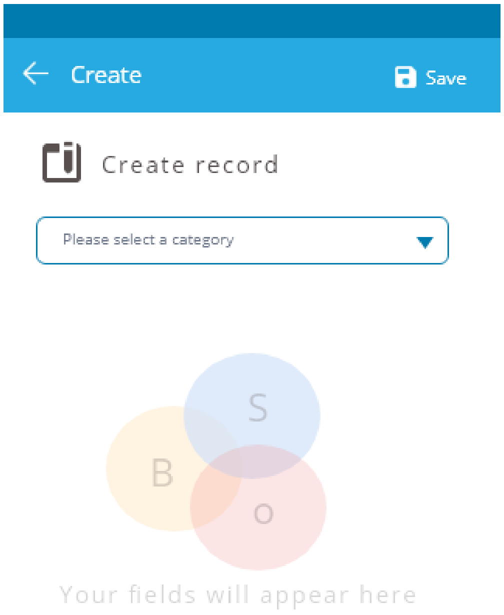
Your entities

fb - sam
ram@gmail.com
cfssae@9232

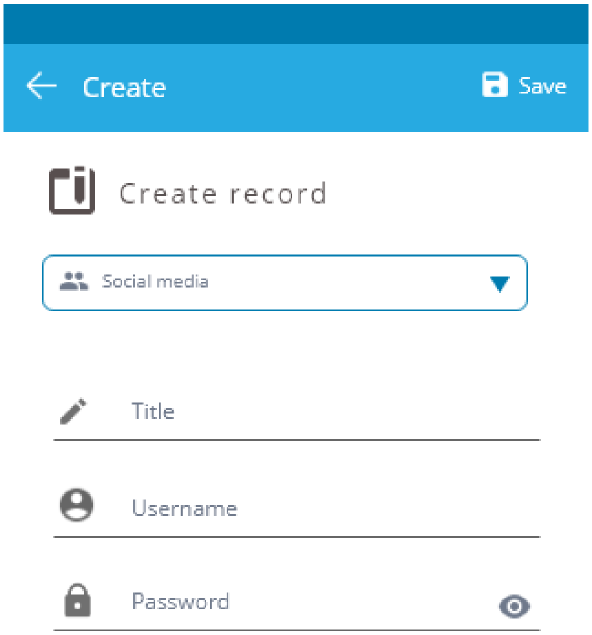
kvb - ram
xxxx-xxxx-xxxx-xxxx
1245

Irctc
abc@gmail.com
1245

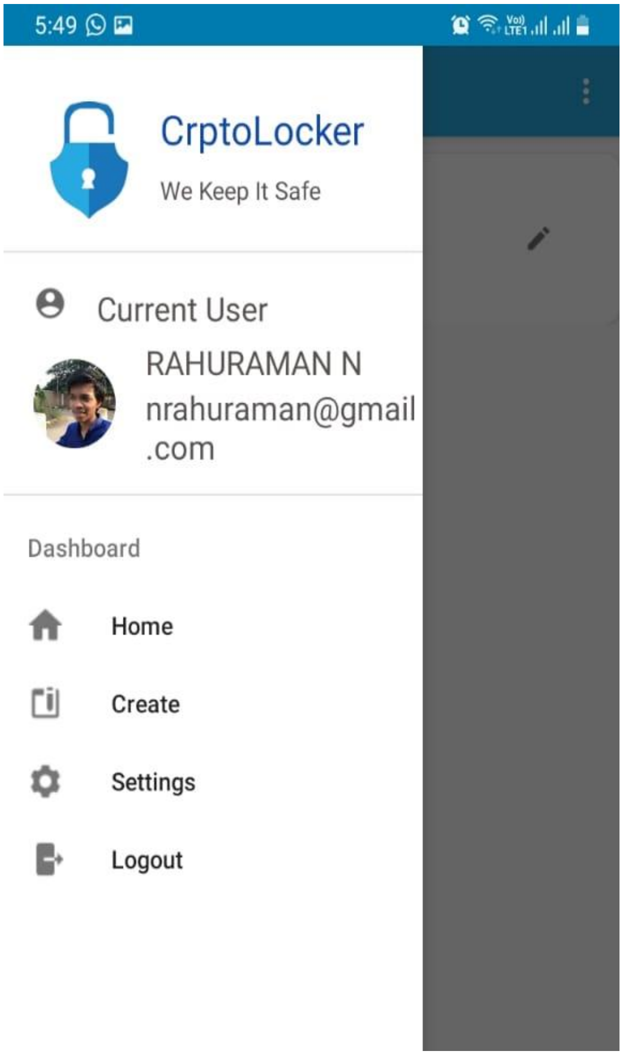
sbi - ram
xxxx-xxxx-xxxx-xxxx
1245



Your fields will appear here

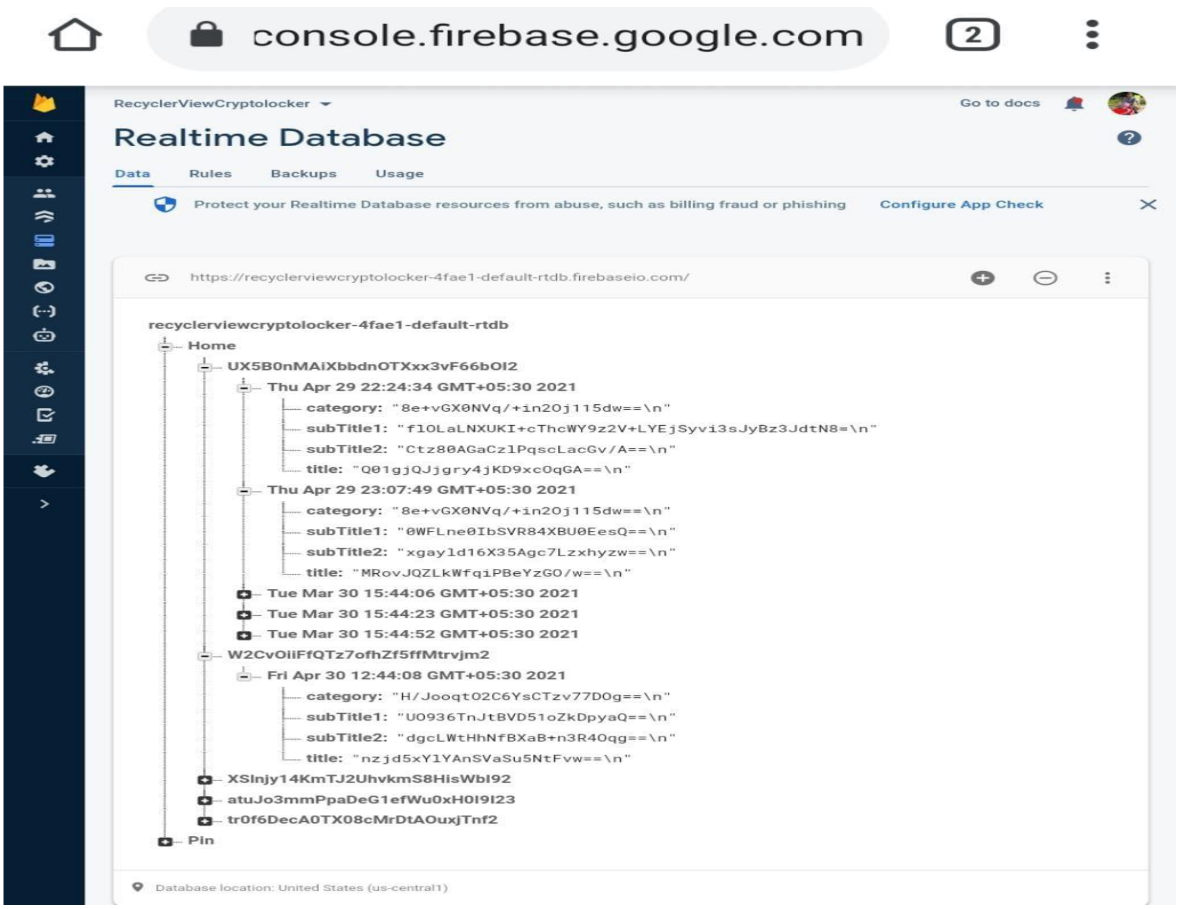


Social media



RAHURAMAN N
nrahuraman@gmail.com

DATABASE IMAGES



console.firebase.google.com
2
RecyclerViewCryptolocker
Go to docs

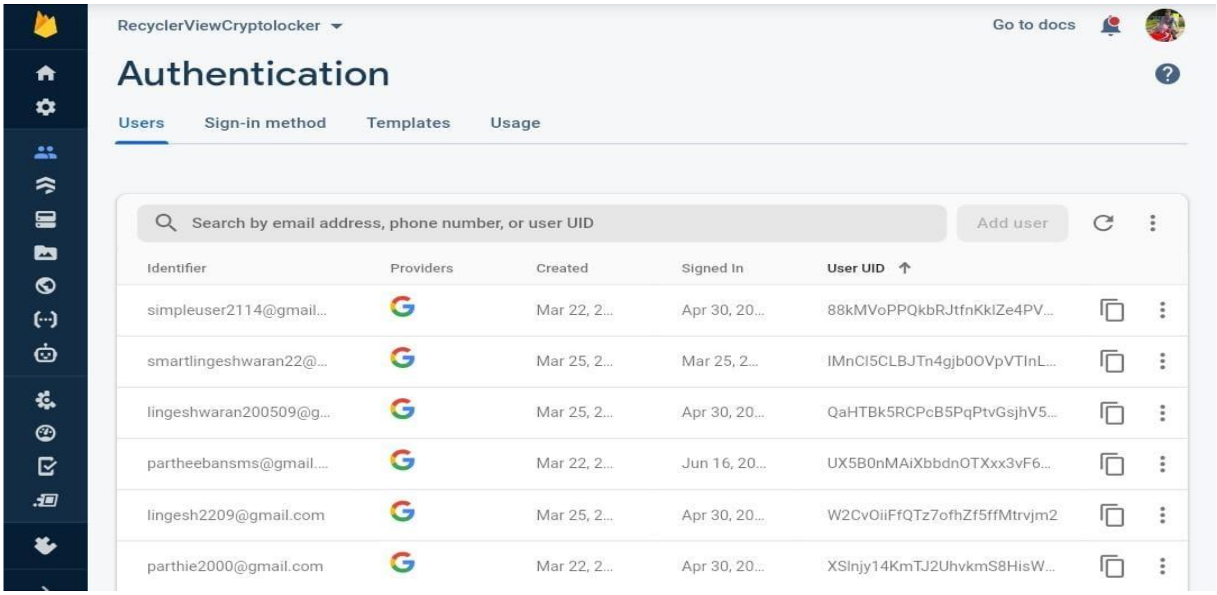
Realtime Database

Data
Rules
Backups
Usage
Protect your Realtime Database resources from abuse, such as billing fraud or phishing
Configure App Check

https://recyclerviewcryptolocker-4fae1-default-rtdb.firebaseio.com/

```
recyclerviewcryptolocker-4fae1-default-rtdb
  Home
    UX5B0nMAiXbbdnOTXxx3vF66bOI2
      Thu Apr 29 22:24:34 GMT+05:30 2021
        category: "8e+vGX0NVq/+in2Oj115dw==\n"
        subTitle1: "f1OLaLNXUKI+cThcWY9z2V+LYEjSyvi3sJyBz3JdtN8=\n"
        subTitle2: "Ctz80AGaCzlPqscLacGv/A==\n"
        title: "Q01gjQJjgry4jKD9xcOqGA==\n"
      Thu Apr 29 23:07:49 GMT+05:30 2021
        category: "8e+vGX0NVq/+in2Oj115dw==\n"
        subTitle1: "0WFLne0IbSVR84XBU0EesQ==\n"
        subTitle2: "xgayld16X35Agc7Lzxhyzw==\n"
        title: "MRovJQZLkWfqiPBeYzGO/w==\n"
      Tue Mar 30 15:44:06 GMT+05:30 2021
      Tue Mar 30 15:44:23 GMT+05:30 2021
      Tue Mar 30 15:44:52 GMT+05:30 2021
    W2CvOiiFfQTz7ofhZf5ffMtrvjm2
      Fri Apr 30 12:44:08 GMT+05:30 2021
        category: "H/Jooqt02C6YsCTzv77DOg==\n"
        subTitle1: "UO936TnJtBVD51oZkDpyaQ==\n"
        subTitle2: "dgcLWtHhNfBXaB+n3R4Oqg==\n"
        title: "nzjd5xYlYAnSVaSu5NtFvw==\n"
    XSlnjy14KmTJ2UhvkmS8HisWbI92
    atuJo3mmPpaDeG1efWu0xH0I9I23
    tr0f6DecA0TX08cMrDtAOuxjTnf2
  Pin
```

Database location: United States (us-central1)



Authentication

Identifier	Providers	Created	Signed In	User UID ↑
simpleuser2114@gmail...	G	Mar 22, 2...	Apr 30, 20...	88kMVoPPQkbRJtfnKkIZe4PV...
smartlingeshwaran22@...	G	Mar 25, 2...	Mar 25, 2...	IMnCI5CLBJTn4gjb0OVpVTlnL...
lingeshwaran200509@g...	G	Mar 25, 2...	Apr 30, 20...	QaHTBk5RCPcB5PqPtvGsjhV5...
partheebansms@gmail....	G	Mar 22, 2...	Jun 16, 20...	UX5B0nMAiXbbdnOTXxx3vF6...
lingesh2209@gmail.com	G	Mar 25, 2...	Apr 30, 20...	W2CvOiiFfQTz7ofhZf5ffMtrvjm2
parthie2000@gmail.com	G	Mar 22, 2...	Apr 30, 20...	XSlnjy14KmTJ2UhvkmS8HisW...

CHAPTER 7

TESTING

7.1 TESTING TECHNIQUES

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 an as-yet –undiscovered error. A successful test is one that uncovers an as-yet- undiscovered error. System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently as expected before live operation commences. It verifies that the whole set of programs hang together. System testing requires a test consists of several key activities and steps for run program, string, system and is important in adopting a successful new system. This is the last chance to detect and correct errors before the system is installed for user acceptance testing.

The software testing process commences once the program is created and the documentation and related data structures are designed. Software testing is essential for correcting errors. Otherwise the program or the project is not said to be complete. Software testing is the critical element of software quality assurance and represents the ultimate the review of specification design and coding. Testing is the process of executing the program with the intent of finding the error. A good test case design is one that as a probability of finding an yet undiscovered error. A successful test is one that uncovers an yet undiscovered error. Any engineering product can be tested in one of the two ways:

7.2 WHITE BOX TESTING

This testing is also called as Glass box testing. In this testing, by knowing the specific functions that a product has been design to perform test can be conducted that demonstrate each function is fully operational at the same time searching for errors in each function. It is a test case design method that uses the control structure of the procedural design to derive test cases. Basis path testing is a white box testing.

Basis path testing:

- Flow graph notation
- Kilometric complexity
- Deriving test cases
- Graph matrices Control

7.3 BLACK BOX TESTING

In this testing by knowing the internal operation of a product, test can be conducted to ensure that “all gears mesh”, that is the internal operation performs according to specification and all internal components have been adequately exercised. It fundamentally focuses on the functional requirements of the software.

The steps involved in black box test case design are:

SOFTWARE TESTING STRATEGIES:

A software testing strategy provides a road map for the software developer. Testing is a set activity that can be planned in advance and conducted systematically. For this reason a template for software testing a set of steps into which we can place specific test case design methods should be strategy should have the following characteristics:

Testing begins at the module level and works “outward” toward the integration of the entire computer based System.

7.4 Test Cases

Test Case Id	**Test Cases**	**Input Test Data**	**Test Case Description**	**Expected Results**	**Actual Results**	**Pass/Fail**
TU01	User Registration	Enter Emailed and Password	Checkuser with the database	User Available	User Available	Pass
TU02	Upload Credentials	Enter the Credentials to be uploaded	To encrypt the entered credential and upload to cloud	Credentials Uploaded	Credentials Uploaded	Pass
TU03	Download Credentials	Select the Credentials To be downloaded	To decrypt the selected credential from the cloud and display	Credentials Displayed	Credentials Displayed	Pass

CHAPTER 8

CONCLUSION AND FUTURE ENHANCEMENT

CONCLUSION

The proposed android application is simple, fast, secure and user friendly to manage user’s Personnel data and securely store them in cloud for instant access. Crptolocker is simple and easy to use app that helps in managing your passwords and files of different account. Cloud data storage technology is the core area in cloud computing and solves the data storage mode of cloud environment. This will be a best app under 15mb to manage your files and password.

FUTURE ENHANCEMENT

Cross platform support for the app will be released in next cycle of update. Auto Fill feature across several android version will be released in the upcoming update. Better and faster UI will be released.

REFERENCES

1. M.Lakshmi Neelima et al, International Journal of Computer Science and Mobile Computing, Vol.3 Issue.5, May- 2014
2. Kun Liua, Long-jiang Donga College of Oriental Application & Technology Beijing Union University, Beijing 102200, China
   Software Development Department Adobe (Beijing) Corporation, Beijing 100085, China
3. Cachin, C., Keidar, I., and Shraer , A. Trusting the cloud. ACM SIGACT News, 20:4 (2009).
4. B. Krebs. Amazon: Hey spammers, get off my cloud, July 2008.