

CAPSTONE PROJECT REPORT

(Project Term January-April, 2017)

The Pi Encryption Suite

Submitted by

B Sessa Sai Kiran
Srishti Singh
Rohit Yadav
Sandeep Kumar

Registration Number: 11205713
Registration Number: 11302280
Registration Number: 11304395
Registration Number: 11302813

Project Group Number: CSERGC0002

Course Code: CSE445

Under the Guidance of

Mr. Deepak Prashar
Assistant Professor.

School of Computer Science and Engineering



L OVELY
P ROFESSIONAL
U NIVERSITY

TOPIC APPROVAL PERFORMANCE

School of Computer Science and Engineering

Program : 1202::B.Tech.- CSE

COURSE CODE : CSE445

REGULAR/BACKLOG : Regular

GROUP NUMBER : CSERGC0002

Supervisor Name : Deepak Prashar

UID : 13897

Designation : Assistant Professor

Qualification : _____

Research Experience : _____

SR.NO.	NAME OF STUDENT	REGISTRATION NO	BATCH	SECTION	CONTACT NUMBER
1	Rohit Yadav	11304395	2013	K1317	09876996571
2	Vantipalli Sandeep Kumar	11302813	2013	K1302	9888772631
3	Srishti Singh	11302280	2013	K1315	09501905241
4	B Sesha Sai Kiran	11205713	2012	K1317	7696779690

SPECIALIZATION AREA : Networking and Security

Supervisor Signature: _____

PROPOSED TOPIC : The Pi-Encryption Suite

Qualitative Assessment of Proposed Topic by PAC		
Sr.No.	Parameter	Rating (out of 10)
1	Project Novelty: Potential of the project to create new knowledge	7.60
2	Project Feasibility: Project can be timely carried out in-house with low-cost and available resources in the University by the students.	7.40
3	Project Academic Inputs: Project topic is relevant and makes extensive use of academic inputs in UG program and serves as a culminating effort for core study area of the degree program.	7.40
4	Project Supervision: Project supervisor's is technically competent to guide students, resolve any issues, and impart necessary skills.	8.00
5	Social Applicability: Project work intends to solve a practical problem.	7.80
6	Future Scope: Project has potential to become basis of future research work, publication or patent.	7.40

PAC Committee Members		
PAC Member 1 Name: Prateek Agrawal	UID: 13714	Recommended (Y/N): NA
PAC Member 2 Name: Pushpendra Kumar Pateriya	UID: 14623	Recommended (Y/N): Yes
PAC Member 3 Name: Deepak Prashar	UID: 13897	Recommended (Y/N): Yes
PAC Member 4 Name: Kewal Krishan	UID: 11179	Recommended (Y/N): Yes
PAC Member 5 Name: Anupinder Singh	UID: 19385	Recommended (Y/N): Yes
DAA Nominee Name: Kanwar Preet Singh	UID: 15367	Recommended (Y/N): Yes

Final Topic Approved by PAC: The Pi-Encryption Suite

Overall Remarks: Approved

PAC CHAIRPERSON Name: 11024::Amandeep Nagpal

Approval Date: 25 Feb 2017

DECLARATION

We hereby declare that the project work entitled The Pi-Encryption Suite is an authentic record of our own work carried out as requirements of Capstone Project for the award of B-Tech degree in Computer Science and Engineering from Lovely Professional University, Phagwara, under the guidance of Mr. Deepak Prashar, during January to April 2017. All the information furnished in this capstone project report is based on our own intensive work and is genuine.

Project Group Number: CSERGC0002

Name of Student 1: B Sessa Sai Kiran

Registration Number: 11205713

Name of Student 2: Srishti Singh

Registration Number: 11302280

Name of Student 3: Rohit Yadav

Registration Number: 11304395

Name of Student 4: Sandeep Kumar

Registration Number: 11302813

B Sessa Sai Kiran

Date: _____

Srishti Singh

Date: _____

Rohit Yadav

Date: _____

Sandeep Kumar

Date: _____

CERTIFICATE

This is to certify that the declaration statement made by this group of students is correct to the best of my knowledge and belief. They have completed this Capstone Project under my guidance and supervision. The present work is the result of their original investigation, effort and study. No part of the work has ever been submitted for any other degree at any University. The Capstone Project is fit for the submission and partial fulfillment of the conditions for the award of B-Tech degree in Computer Science and Engineering from Lovely Professional University, Phagwara.

Mr. Deepak Prashar.

Asst. Professor.

School of Computer Science and Engineering,

Lovely Professional University,

Phagwara, Punjab.

Date:

ACKNOWLEDGEMENT

A formal thanksgiving and acknowledging the support is not only a part of formal protocol but it gives us a medium to elucidate the people responsible for our growth and development. First of all, we are grateful to Mr. Deepak Prashar for giving us the opportunity to undertake this project and provide us with all the facilities. Our mentor was extremely active in helping us throughout the semester, replying to our e-mails immediately with expert advice and guidance. Without his help and the arrangements, he made for us, this project would not have borne fruits as it did.

Our Parents, Friends, Faculty and Administration in college have always showered their blessings and provided guidance at each nook and corner of our learning curve.

TABLE OF CONTENTS

Inner first page.....	(i)
PAC form.....	(ii)
Declaration.....	(iii)
Certificate.....	(iv)
Acknowledgement.....	(v)
Table of Contents.....	(vi)

1. INTRODUCTION	1
1.1 MOTIVATION	
2. PROFILE OF THE PROBLEM	3
3. EXISTING SYSTEM	4
3.1 INTRODUCTION	
3.2 FLOWCHART FOR PRESENT SYSTEM	
3.3 DFD FOR PRESENT SYSTEM	
4. PROBLEM ANALYSIS	6
4.1 PRODUCT DEFINITION	
4.2 FEASIBILITY ANALYSIS	
4.3 PROJECT PLAN	
5. SOFTWARE REQUIREMENT ANALYSIS	12
5.1 INTRODUCTION	

5.2 GENERAL DESCRIPTION

5.2.1 PRODUCT PERSPECTIVE

5.2.2 USER CLASS AND CHARACTERISTICS

5.3 SPECIFIC REQUIREMENTS

5.3.1 OTHER NON-FUNCTIONAL REQUIREMENTS

6. DESIGN

13

6.1 SYSTEM DESIGN

6.1.1 USE CASE DIAGRAM

6.2 DESIGN NOTATIONS

6.2.1 DATA FLOW DIAGRAM

6.2.2 ENTITY RELATIONSHIP DIAGRAM

6.3 DETAILED DESIGN

6.3.1 CLASS DIAGRAM

6.4 FLOWCHART

6.5 PSEUDO CODE

6.5.1 PSEUDO CODE FOR APPLICATION

7. TESTING

21

7.1 FUNCTIONAL TESTING

7.2 STRUCTURAL TESTING

7.3 LEVELS OF TESTING	
7.4 TESTING THE PROJECT	
8 IMPLEMENTATION	25
8.1 IMPLEMENTATION OF PROJECT	
8.2 POST IMPLEMENTATION AND MAINTENANCE	
8.2.1 SOFTWARE MAINTENANCE	
8.2.2 WEBSITE'S MAINTENANCE	
9 PROJECT LEGACY	27
9.1 CURRENT STATUS OF PROJECT	
9.2 REMAINING AREAS OF CONCERN	
9.3 TECHNICAL AND MANAGERIAL LESSONS LEARNT	
10 USER MANUAL	29
11 SYSTEM SCREENSHOTS	30
12 BIBLIOGRAPHY	46

1. INTRODUCTION

The Pi-Encryption suite is a robust and a well grown cryptosystem which provides security to a datum. Like any, the most important aspect of a security system is the confidentiality of a given message. Now this message may be a private message, or a mail you have to send someone, or any password on a social networking site. This traditional goal of confidentiality is achieved through the process of encryption and decryption. Encryption is simply encoding the original message in such a way that it is unreadable to an unauthorized entity. The message thus, encrypted is called a ciphertext and the method is called ciphering. On the receiver's end there is an inverse cipher applied (deciphering) to get back the original intended message. This way only the authorized parties will have access to the entire communication. This brings us into a new territory, authorization. How do you determine whether the sender (or) the receiver is authorized? This challenge is solved using the technique involving signatures. These are the basic goals of any security system, but not limited to. There are many other important aspects to be taken care of like Integrity and Repudiation.

Our approach for confidential message exchange is achieved using a proprietary protocol developed especially for this purpose. The algorithm we have designed will work on the irrational number pi. The digits of pi after the decimal point are seemingly random. The flow of the said digits doesn't repeat and are non-ending. We have used this phenomenon as a vital part of our key generation process. The end result was a cryptographically valid and an efficient algorithm for encryption and decryption. It also accompanies some of the industrial standard algorithms like SHA512, MD5 and various Digital Signature schemes. Our motto was to deliver the complete package to the client and provide an all in one solution.

1.1 MOTIVATION

This project's main motto is to provide a customizable solution for security to an end user, let that be a corporate client or a personal home user. The motivation for this project was to make the concept of open source security a reality. The learning curve in this study is exponentially aspiring moreover, the start-up scene in our country is sky rocketing. This paper can also be used for further study in the same field and improving the efficiency of the existing concepts. The implementation of an encryption algorithm based on the irrational pi, is not a

completely new concept. However, the way we have done it is completely unique. This project was also motivating because we had a chance to learn new technologies and explore completely new things in cryptography. The implementation already gave us much hope. The project was massive and had to be done with extreme care. We also had the chance to learn new techniques of software management, parallel processing and high performance computing. The implementation of the above system was using a windows application for demonstrating the projects capabilities. We have also developed a password manager as a Google Chrome Extension, both of which gave us an opportunity to enter unexplored territories.

2. PROFILE OF THE PROBLEM

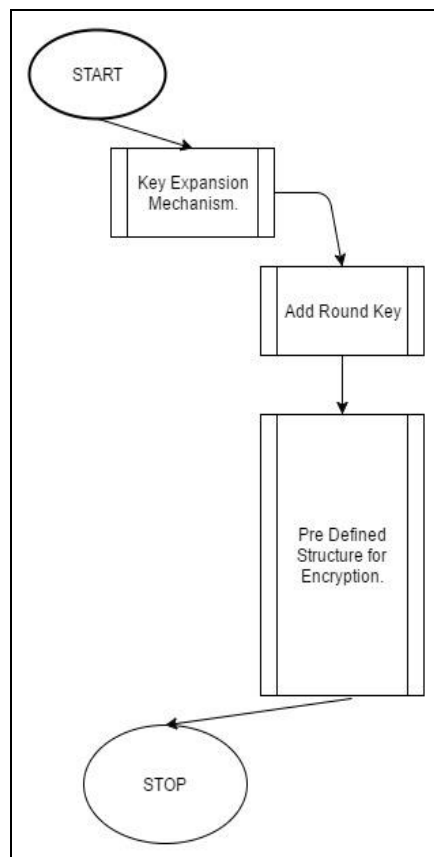
The profile of this problem was to implement an encryption suite on the basis of pi and make it open source for general public use under the Open Source Licenses. The cryptosystem had to have all the functionalities of a traditional cryptographic package and had to be efficient enough to be implemented on a sample data. The working of the cryptographic package was to be showcased using a windows application and a web browser extension. The project was to be completed within the prescribed time with all the required aspects and ready for corporate shipping.

3. EXISTING SYSTEM

The working area of the project is relatively new. So the existing system is quite small. There are some companies which provide network security via their own custom mechanism but are paid and abstracted. The existing system is also outdated and still use basic mechanisms like AES or DES. Now accepted these security measures are efficient but they also are relatively common. Our system is a brand new idea and it is implemented in its own unique way.

3.1 FLOWCHART (EXISTING SYSTEM)

This flowchart here (Fig 1) shows us the flow of data in the existing system, pictorially via a flow chart. The existing AES uses a formal structure for its encryption process and is the same for all the versions of the same. The only difference will be the key size and the number of blocks the key occupies in one go. The message goes through three phases basically. The encryption phase is done via adding different keys each time. There are multiple rounds as well. The number of rounds depends on the size of the key being used. Moreover, it is applied multiple times and is called multiple AES



The pre-defined block is kind of a feistel structure which will repeat itself when called similar to recursion. It has four main functions.

- Sub Byte Substitution
- Shift rows
- Mix Columns
- Add Round key

3.2 DFD EXISTING SYSTEM

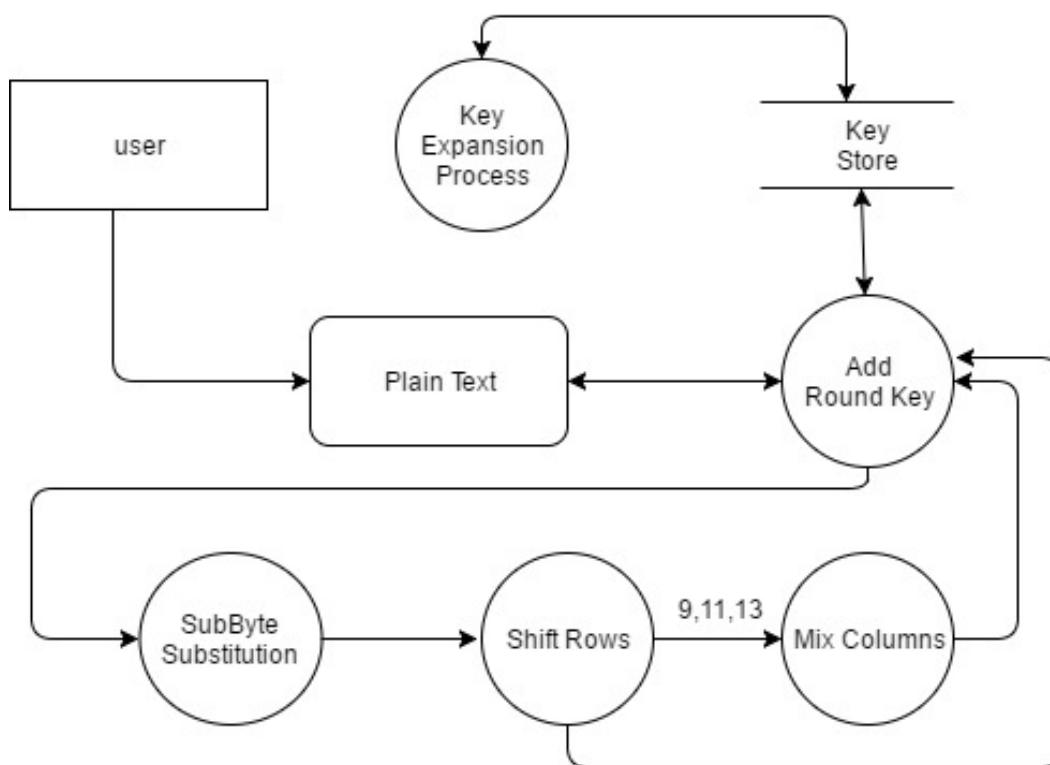


Fig 2

Here the numbers 9,11,13 represent round numbers, meaning the step is only executed if the round number is either 9 or 11, or 13. In all the other rounds the data flow happens through the arrow initiated on the other side of the circle. The step Mix Columns is not executed in that case.

4. PROBLEM ANALYSIS

4.1 PROBLEM DEFINITION

To develop an encryption suite which provides all the traditional goals of security over a network or as a standalone security application. The main goal is to overcome the basic problems with the existing system and build a new one which does all the tasks better. To implement the following security algorithm with the help of a chrome extension and a windows application which demonstrates the same.

4.2 FEASIBILITY ANALYSIS

The feasibility analysis is the study which determines the ‘do ‘ability of the project. The project proposed was a quite big undertaking for us as we had many new things to learn. Basically this phase emphasizes on the practicality of the project. It aims to uncover any weaknesses of an existing idea or the threats associated with it. There are basically two criteria when it comes to feasibility study, for judging the system, those are

- Cost Required
- Value to be Attained

In our case the cost of the system refers to the resource dependency. It is a vital check on the available resources at our disposal to attain the specified goal. Resources indicate mainly the memory and the processing architecture of our system. Given the field of network security and cryptography is highly resource exhausting, performing this study was very essential. We have determined that this project is very feasible after the first study.

The second aspect is the value to be attained. The value to be attained is the worthiness of the project after it is successfully completed. The value proposition of this system was considered based upon the efficiency of the existing system and the amount of improvement that can be brought upon. After extensive study for a week we have determined that the value proposition of this project is good and we can proceed ahead with it. The project was determined to be Very Feasible after the said study.

4.3 PROJECT PLAN

This phase emphasizes on the way the project is supposed to be completed. This phase is extremely important. Any mistake in this discipline can be lethal as far as the system completion is concerned. The main aspects to be taken care of during this phase are.

- Setting objectives (these should be measurable)
- Identifying deliverables
- Planning the schedule
- Making supporting plans

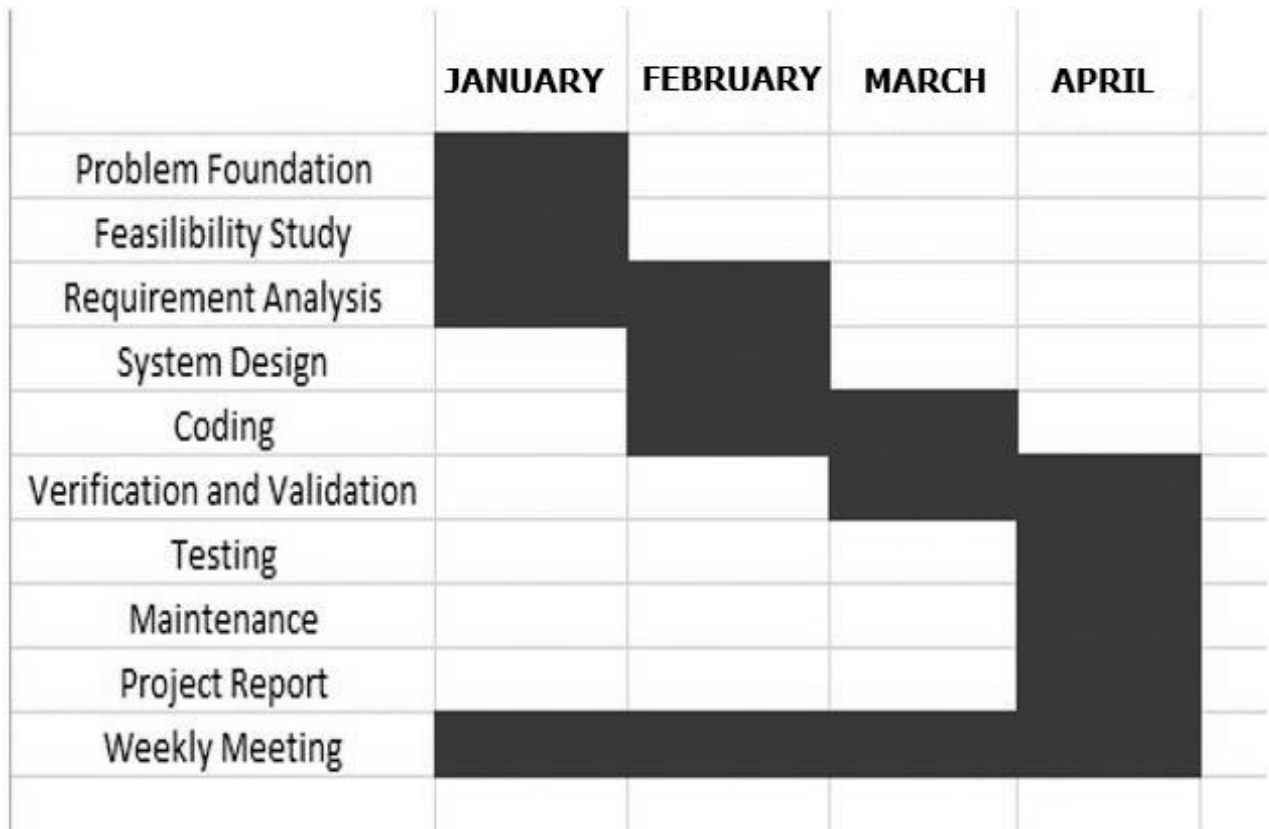


Fig 3.

Fig 3 describes a pictorial representation of the time division employed. This division helped us a lot as it made our tasks easier and more streamlined.

4.4 GANTT CHART

A Gantt chart is a graphical depiction of a project schedule. A Gantt chart is a type of bar chart that shows the start and finish dates of several elements of a project that include resources, milestones, tasks and dependencies. Henry Gantt, an American mechanical engineer, designed the Gantt chart.

Task name	January 2017	February 2017
	2W 3W 4W	1W 2W 3W 4W
Problem Foundation		
Feasibility Study		
Requirement analysis		
System Design		
Coding		
Verification and Validation		
Testing		
Maintenance		
Project Report		
Weekly meeting		

Task name	March 2017	April 2017
	1W 2W 3W 4W	1W 2W 3W 4W
Problem Foundation		
Feasibility Study		
Requirement analysis		
System Design		
Coding		
Verification and Validation		
Testing		
Maintenance		
Project Report		
Weekly meeting		

4.5 TECHNOLOGIES USED

4.5.1 C#

C# is an elegant and type-safe object-oriented language that enables developers to build a variety of secure and robust applications that run on the .NET Framework. You can use C# to create Windows client applications, XML Web services, distributed components, client-server applications, database applications, and much, much more. Visual C# provides an advanced code editor, convenient user interface designers, integrated debugger, and many other tools to make it easier to develop applications based on the C# language and the .NET Framework. As an object-oriented language, C# supports the concepts of encapsulation, inheritance, and polymorphism. All variables and methods, including the Main method, the application's entry point, are encapsulated within class definitions. A class may inherit directly from one parent class, but it may implement any number of interfaces. Methods that override virtual methods in a parent class require the override keyword as a way to avoid accidental redefinition. For example, in C#, a struct is like a lightweight class; it is a stack-allocated type that can implement interfaces but does not support inheritance. It is a robust and a strongly typed language and makes windows application development easier.

4.5.2 JAVA

Java is a programming language and a computing platform for application development. It was first released by Sun Microsystem in 1995 and later acquired by Oracle Corporation. A Java platform includes an execution engine, a compiler and a set of libraries. Java is not specific to any processor or operating system. Like C compiler, Java compiler does not produce native executable code for a particular machine. Instead, Java produces a unique format called bytecode. It executes according to the rules laid out in the virtual machine specification. The official website provides links to freely download the latest version of Java. You can use the Oracle Java website to learn more about downloading Java, verify Java is installed on your computer, remove older versions, troubleshoot Java or report an issue as well. Oracle also has complete documentation for java and it is one of the best documented languages in the world.

4.5.3 MATLAB

MATLAB (“Matrix Laboratory”) is a tool for numerical computation and visualization. The basic data element is a matrix, so if you need a program that manipulates array-based data it is generally fast to write and run in MATLAB (unless you have very large arrays or lots of computations, in which case you’re better off using C or Fortran). If you are doing a computation of any significant length in MATLAB, you will probably want to make an m-file. Anything that you would type at the command prompt you can put in the m-file (for example, “script.m”) and then run it all at once (by typing the name of the m-file, e.g. “script”, at the command prompt). You can even add comments to your m-file, by putting a “%” at the beginning of a comment line. Matlab also has support for almost every function that a computer programmer can think of. It has a wide selection starting from image processing to gaming and weather prediction etc. It is also a very highly documented language and is rated as the best documented language in the world.

4.5.4 C

A high-level programming language developed by Dennis Ritchie at Bell Labs in the mid-1970s. Although originally designed as a systems programming language, C has proved to be a powerful and flexible language that can be used for a variety of applications, from business programs to engineering. C is a particularly popular language for personal computer programmers because it is relatively small, it requires less memory than other languages. The first major program written in C was the UNIX operating system, and for many years C was considered to be inextricably linked with UNIX. Now, however, C is an important language independent of UNIX. Although it is a high-level language, C is much closer to assembly language than are most other high-level languages. This closeness to the underlying machine language allows C programmers to write very efficient code. The low-level nature of C, however, can make the language difficult to use for some types of applications.

4.5.5 C++

C++ is a general-purpose object-oriented programming (OOP) language, developed by Bjarne Stroustrup, and is an extension of the C language. It is therefore possible to code C++ in a "C style" or "object-oriented style." In certain scenarios, it can be coded in either way and is thus an

effective example of a hybrid language. C++ is considered to be an intermediate-level language, as it encapsulates both high- and low-level language features. Initially, the language was called "C with classes" as it had all the properties of the C language with an additional concept of "classes." However, it was renamed C++ in 1983. It is pronounced "see-plus-plus." The plus plus notation has a meaning and is there for a reason. The incremented operator in c is a ++ since c++ is an incremental update over c, the notation was given as C++. It was used as a precompiler for our c program as it supports backwards functionality as well.

4.5.6 SYSTEM CALLS

We have used system calls built into the Linux kernel and further included into the GCC library. These system calls helped us maintain parallel processing of the threads and also concurrent execution, the `execv ()` system call contributed the process image replacement and also in improving the modularity of the system.

4.5.7 MATLAB CONTROL LIBRARY BY GOOGLE

This library was essential for us in stitching the various technologies together. The API is provided by google and the documentation is provided by oracle. This library was once deprecated but is now alive, contributing to challenging projects like these.

5. SOFTWARE REQUIREMENTS

5.1 SOFTWARE REQUIREMENT ANALYSIS

5.1 INTRODUCTION

The purpose of this document is to provide the software requirement specification report for the “The Pi Encryption Suite”.

This is a graduate level project and is being implemented under the guidance of college professors. This would be useful to students as well as teachers to resolve their queries.

The purpose of the system is to provide all round security to the end user, let that end user be a corporate company or an end user. Above all, we hope to provide a comfortable user experience.

5.2 SPECIFIC REQUIREMENTS

These technologies were used for the development of the project are as listed.

C, C#

C++

JAVA

Matlab

MatlabCotrol API

Linux System Calls via GCC

Multi-threading concepts.

Microsoft Windows Operating System for the Windows Application.

Linux Operating System in general.

Basic Knowledge of scripting is required to understand the algorithm in detail.

Image processing and Optical character recognition were used as well.

Microsoft CryptgenNext library was used as well.

Java Cryptography.

6. SYSTEM DESIGN

6.1 DESIGN

System's design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. Object-oriented analysis and design methods are becoming the most widely used methods for computer systems design. The UML has become the standard language in object-oriented analysis and design. It is widely used for modelling software systems and is increasingly used for high designing non-software systems and organizations. The Unified Modelling Language (UML) is a general-purpose, developmental, modelling language in the field of software engineering that is intended to provide a standard way to visualize the design of a system.

The Unified Modelling Language (UML) offers a way to visualize a system's architectural blueprints in a diagram including elements such as:

1. Any activities (jobs)
2. Individual components of the system
3. How they can interact with other software components.
4. How the system will run
5. How entities interact with others (components and interfaces)
6. External user interface

Although originally intended solely for object-oriented design documentation, the Unified Modelling Language (UML) has been extended to cover a larger set of design documentation and been found useful in many contexts

6.1.1 USE CASE 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. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well.

Five key relationships in UML class diagram:

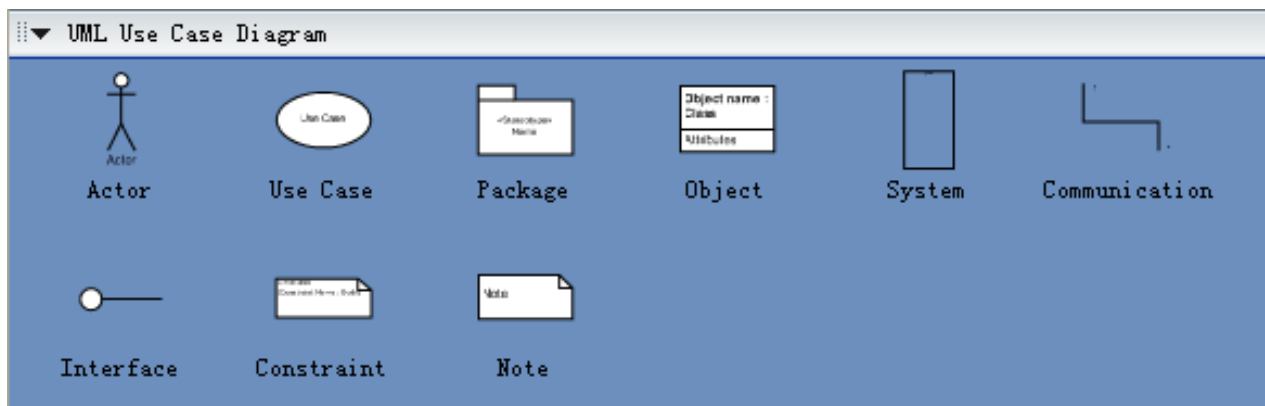
Aggregation is a special case of association. A directional association between objects. When an object 'has-a' another object. Direction between them specified which object contains the other object. Aggregation is also called a "Has-a" relationship.

Composition is a special case of aggregation; a restricted aggregation is called composition. When an object contains the other object, if the contained object cannot exist without the existence of container object, then it is called composition.

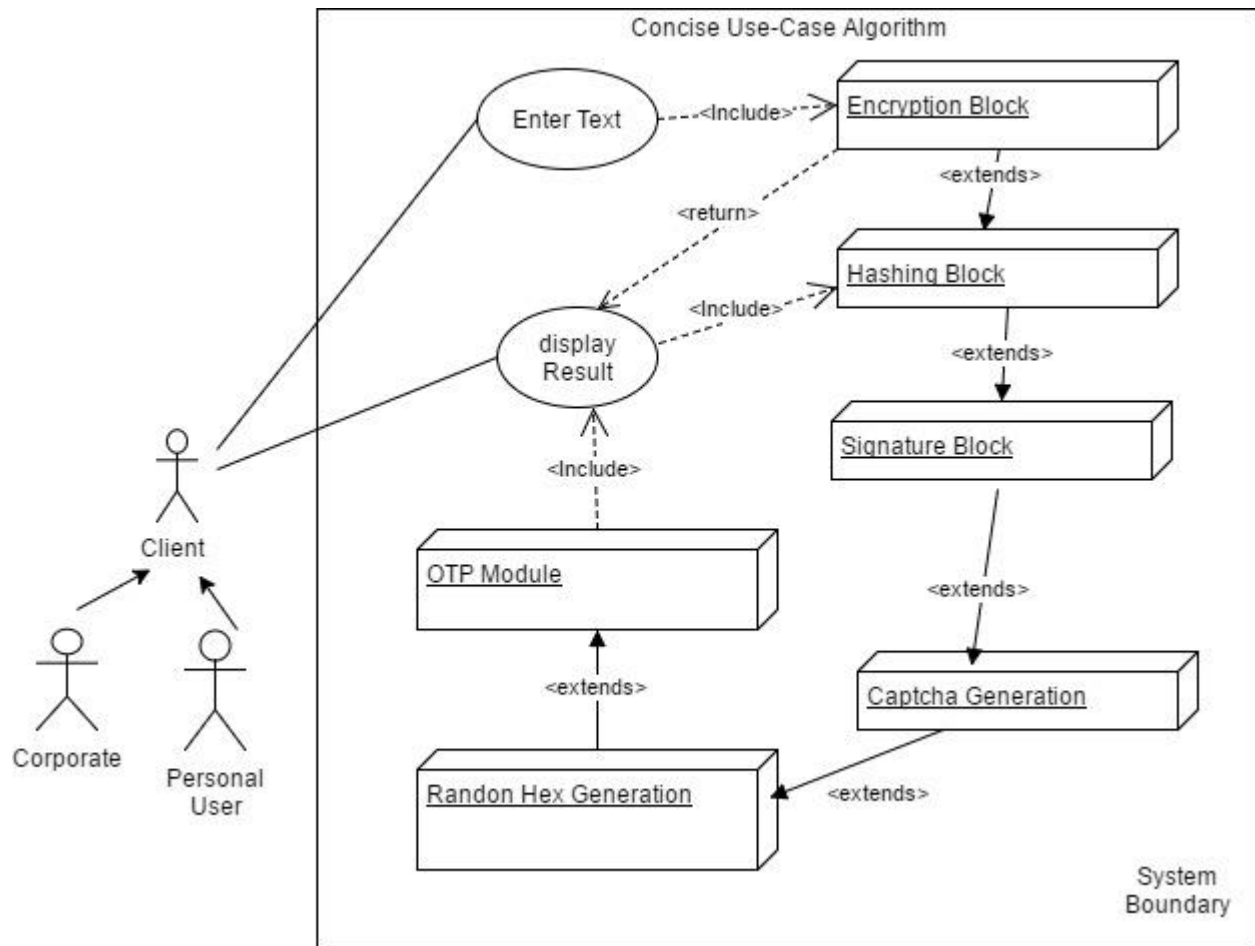
Realization is a relationship between the blueprint class and the object containing its respective implementation level details. This object is said to realize the blueprint class.

Inheritance shows a "is-a" relationship.

Dependency is an important aspect in UML elements. It describes the dependent elements and the direction of dependency. Dependency is represented by a dotted arrow.

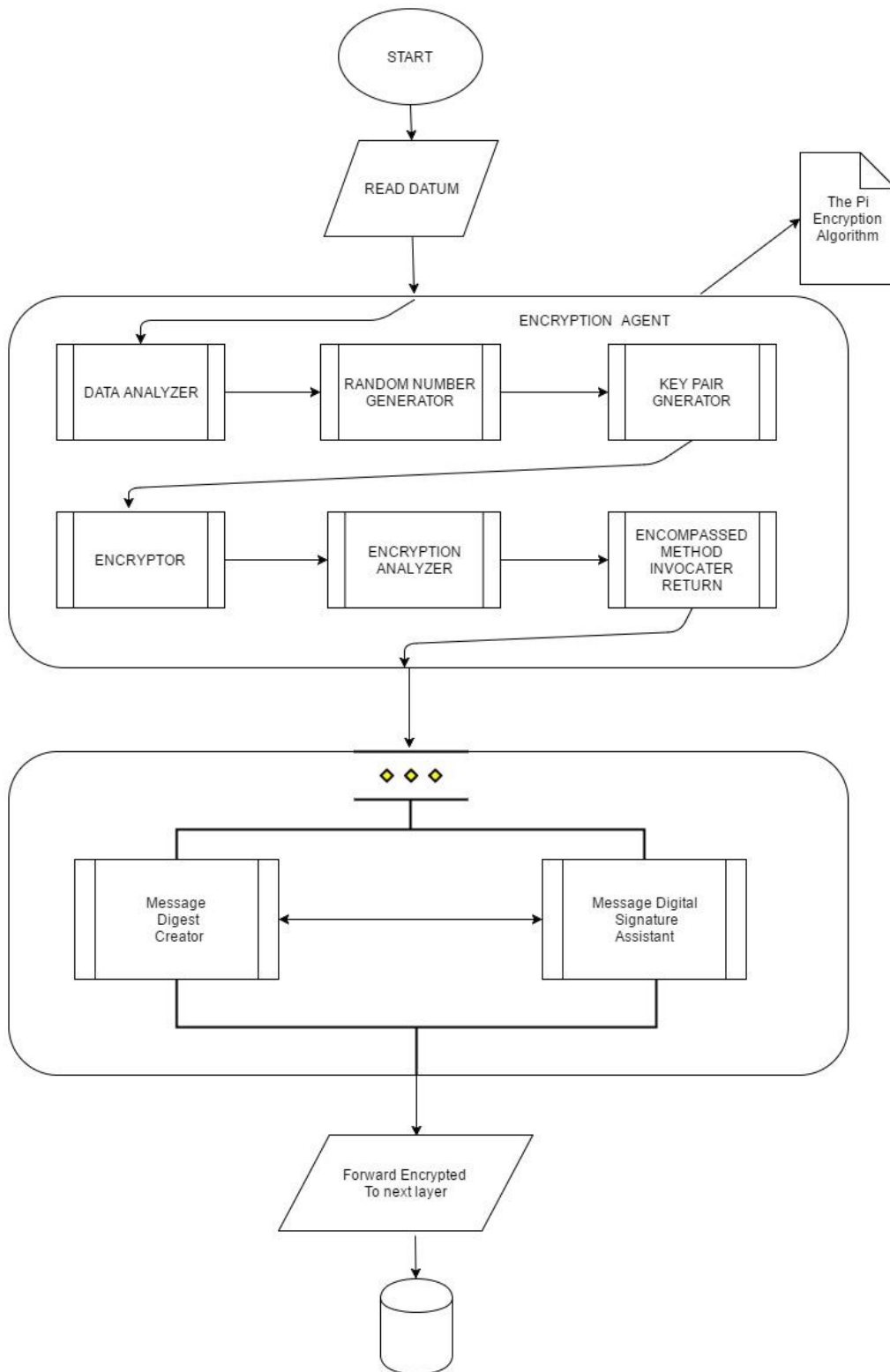


6.1.2 SYSTEM USE CASE DIAGRAM



The blocks in the above diagram represent an abstracted code segment which is a factory method. It is only invoked only when a requirement is met. Each function has a specific function and it is abstracted in a way that encapsulation is met. There are methods that are needed to be invoked to attain a certain function. The source code of that particular code segment is unknown to the end user.

6.1.3 FLOW DIAGRAM/ FLOWCHART



6.1.4 DFD OF SYSTEM

6.1.4.1 DESIGN NOTATION

Dynamic: Dynamic usually means capable of action and/or change.

1. Data flow diagrams (DFDs).
2. State transition diagrams (STDs).
3. State charts.
4. Structure diagrams.

Static: Static means fixed.

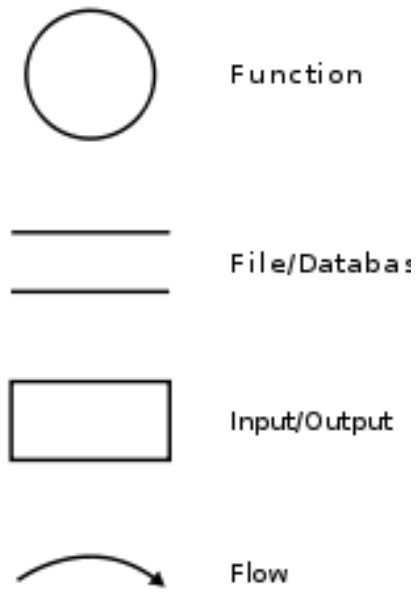
1. Entity Relationship Diagrams (ERDs).
2. Class diagrams.
3. Structure charts.
4. Object diagrams.

6.1.4.2 DFD INTRODUCTION

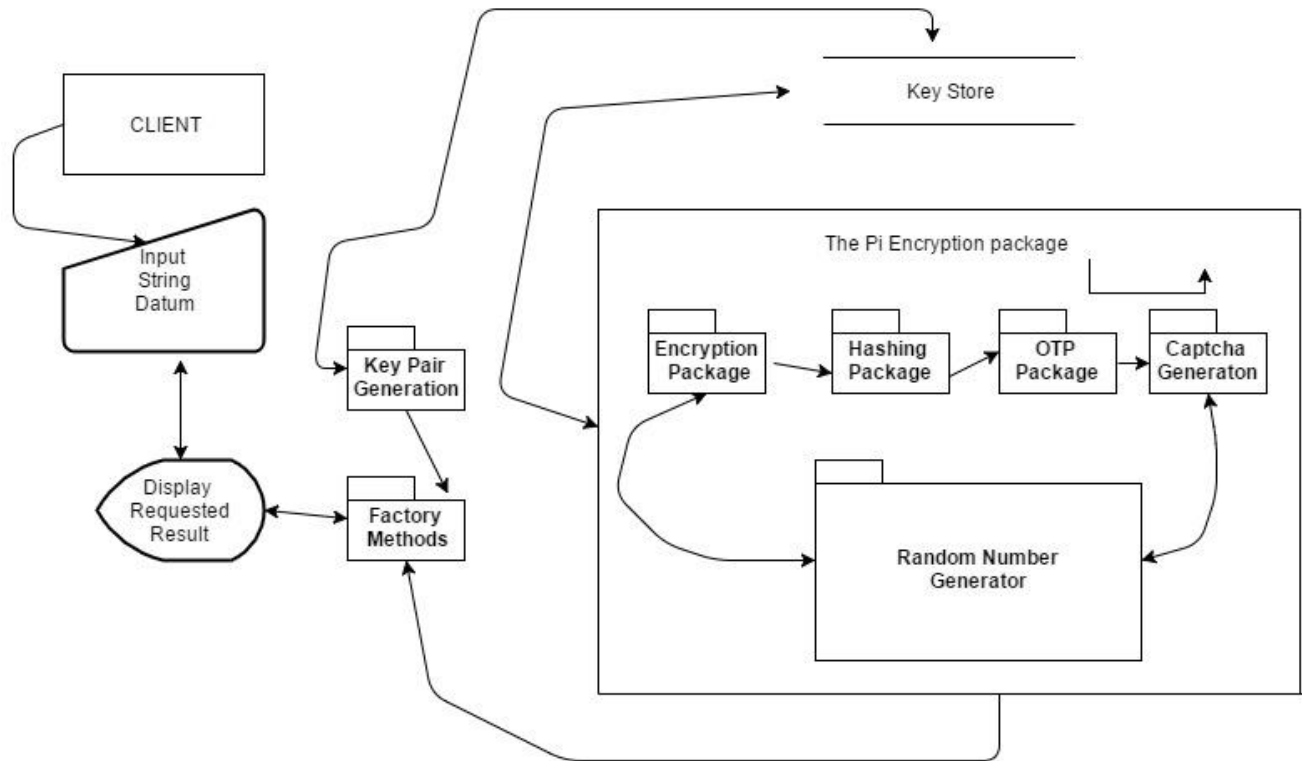
A data flow diagram (DFD) is a graphical representation of the “flow” of data through an information system, modelling its process aspects. A DFD is often used as a preliminary step to create an overview of the system, which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design). A DFD shows what kind of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of process or information about whether processes will operate in sequence or in parallel, which is shown on a flowchart.

Symbols used in a data flow diagram are called the notations used in the same. The data flow diagram uses many aspects of conventional programming like functions, methods, packages, invokers, connectors, components etc. The only advantage of a dfd is that it shows the data flow.

The data flow is also generalized for optimal flexibility and improves the dynamic nature of the same.



6.1.5 DFD OF THE SYSTEM



6.2 PSEUDO CODE

Pseudocode is an informal high-level description of the operating principle of a computer program or other algorithm.

It uses the structural conventions of a programming language, but is intended for human reading rather than machine reading. Pseudocode typically omits details that are essential for machine understanding of the algorithm, such as variable declarations, system-specific code and some subroutines. The programming language is augmented with natural description details, where convenient, or with compact mathematical notation. The purpose of using pseudocode is that it is easier for people to understand than conventional programming language code, and that it is an efficient and environment-independent description of the key principles of an algorithm. It is commonly used in textbooks and scientific publications that are documenting various algorithms, and also in planning of computer program development, for sketching out the structure of the program before the actual coding takes place.

No standard for pseudocode syntax exists, as a program in pseudocode is not an executable program. Pseudocode resembles, but should not be confused with skeleton programs which can be compiled without errors.

6.2.1 PSEUDOCODE OF THE ALGORITHM

```
Start Pi Encryption Suite;  
  
If(user wants encryption)  
  
Enter Text; (The text that the user wants to be ciphered)  
  
Generate System entropic values;  
  
Generate key via KeyGenerator;  
  
Generate keypair;  
  
Generate encryption function;  
  
Use encryption function via keyPair;  
  
If(user wants additional services)  
  
Pipeline the data towards the signature suite;
```

Pipeline the output to the hashing algorithm;
Generate the end message digest;
Else()
Forward the ciphered text to the display dialog;
Else()
Display the end result to the user.
End;

6.2.2 PSEUDOCODE FOR THE APPLICATION

Login
if (registered user)
login
proceed to the pi encryption suite.
Else
Redirect to signup page;
Start pi encryption suite;
Enter text;
Cipher text;
or
Enter Cipher text;
Enter Key;
The real message;

7. TESTING

Testing is the method in which the code is executed expecting to find an error. It is done so maximum number of errors can be found and removed before finally presenting the system to the customer. The process ensures smoothly working, efficient system with minimal failure probability. A piece of information is not considered valid if not tested properly. The phase ensures that there is no complaint from customer's side when the project is finally submitted as the final product.

7.1 NEED FOR TESTING

To ensure the success of the project, testing has always been considered, one of the most important phases. Like one does not buy a car before test drive to ensure its smooth functionality, just like that, one does not just simply submit a project before testing it properly. No testing or wrong testing lead to odd and wrong results that turn into a much bigger issue later in time that may not even be fixed. It may have its cost but it always prevents long run costs of maintenance and service. Just like they say, prevention is better than cure.

7.2 TYPES OF TESTING

7.2.1 FUNCTIONAL TESTING

As the name suggests, functional testing requires the tester to test the project to meet all the requirement expectations of the end user and the customer. After the completion of functional testing the project is expected to have all the functional requirements mentioned in the SRS document working smoothly and properly. The whole project is tested with the use cases that are provided by the team of design, so that the project can be tested for all the requirements in different scenarios. For example, the system can be tested to make sure, that in the encryption module, the message given as an input is successfully encoded into a cipher text; now this is called testing against a requirement.

The importance of functional testing cannot be underrated. It plays a vital role in ensuring that the system properly works, every customer expectation is met and is suitable for release. It

also ensures that the system will prove itself to be useful after release. Basically, what's made sure is that nothing is left in the SRS document that's not implemented in the proposed system. Function testing expects one to meet all the customer's requirements while non- functional testing expects one to meet all the customer's expectations. A log in page is a requirement, hence functional testing will test the system against it but good performance is a non- functional requirement. Functional testing is a process of assuring quality of the software product. It is also called black box testing which includes test cases on the basis of specification of the product components.

Here different modules are tested by feeding them input and then further examining the output. Hence internal procedure of the functioning is not considered to be checked. It usually describes the work which system is supposed to perform. Functional testing tests a slice of functionality of the whole system.

Functional testing typically involves five steps:

1. The identification of functions that the software is expected to perform.
2. The creation of input data based on the function's specifications.
3. The determination of output based on the function's specifications.
4. The execution of the test case.
5. The comparison of actual and expected outputs.

7.2.2 STRUCTURAL TESTING

The structural testing is the testing of the structure of the system. The testers should have the knowledge of internal implementations and the code of the internal system. They should be known to the complete working of the software and all of the ways in which code is been implemented. In white-box testing an internal perspective of the system, as well as programming skills, are used to design test cases. The tester chooses inputs to exercise paths through the code and determine the appropriate outputs. This is analogous to testing nodes in a circuit, e.g. in-circuit testing.

Structural testing emphasizes on:

1. Logic coverage statement

2. Branch coverage
3. Condition coverage
4. Dataflow coverage
5. Path conditions and symbolic evaluation etc.

7.2.2.1 STATEMENT TESTING

The technique involves testing the whole code statement wise, every statement is tested individually which helps to eliminate minute errors syntax or logic wise. Minimum number of test cases is used to avoid confusion.

7.2.2.2 BRANCH TESTING

A code can have branches and statement coverage can cover the statements but miss small modules which are further covered in branch coverage. It is made sure that all the branches are tested at least once and no branch is left untested doomed to cause trouble later.

7.2.2.3 PATH COVERAGE

What is left untested after branch and statement coverage is covered here. Every path possible is found and tested leaving no branch, no statement behind.

7.3 LEVELS OF TESTING

7.3.1 UNIT TESTING

In unit testing, the smallest part of the code called a unit is tested individually. A unit can be a statement or even a module. This type of testing is done to check whether every part of the code is functioning properly. To verify the proper working of code in phase of coding, unit testing is believed to be essential.

The main idea behind unit testing is to divide the code into small parts called units and ensure that each one is working properly and no unit or module has a bug of its own.

7.3.2 INTEGRATION TESTING

Unit testing is the first level of testing and integration testing is the next level. In this testing the whole code is divided into different modules and each of them is tested separately. What makes it different from unit testing is that in integration testing, initial module is taken into account and tested and the testing succeeds, the next module is tested and added to the previous one and then both of them are tested together. Similarly, more and more modules are added and at last system is tested as a whole constituting the next type of testing.

7.3.3 SYSTEM TESTING

After every unit is tested, modules are tested and integration testing is finished, one can move to the next step which is system testing. In system testing, the whole system is tested as a whole. The main goal of this testing to check whether the requirements of the customer have been met. This type of testing is concerned with input and output and not with how the system is working. This testing is done in reference to SRS document. This type of testing is believed to be essential for validation.

7.3.4 ACCEPTANCE TESTING

This testing is last level of testing. After testing, the whole project is believed to be properly working and worthy of final presentation to the client. The tester uses realistic data which is provided by the client in order to prove that the project is working satisfactorily, efficiency and up to customer's expectations. Acceptance testing is known to be a type of black box testing where the tester is only concerned with input and output of the system and not the working or implementation or coding for that matter.

8. IMPLEMENTATION

As a simple entrepreneur goes from planning to production, an engineer goes from development to implementation. This is the main step, the one all the work is done for. When a product is properly tested and believed to be fit for use, it is released in the market for the users to handle and manage, hoping that there will be a positive response and its acceptance in the society. This phase will start when the designing phase is complete and one finally has a system to work on. The project is finally run into motion. The satisfaction of the completed work runs in the organization. The system is new and is being implemented just before maintenance starts; every user is enjoying the new technology.

The organization may get feedbacks after the implementation of the system, about whether it was liked or not or whether it could have been better, suggestions on improving may flood the organization or maybe even praise will take over. The next step would be maintenance.

8.1 IMPLEMENTATION OF PROJECT

The project as of now is live on Git and can be forked by anyone who needs security on the go. The project is completely open-source and is fully functional. The further users have to quote this capstone team as the lead developers and can use them for future security measures. It provides all the functionality it has deemed it would and it is performing them with ease.

8.2 POST IMPLEMENTATION AND MAINTENANCE

Post implementation review is what comes after implementation. It is the feedback that is taken after the project is released to work with users. The main goal of this post implementation review or the feedback survey is the confirmation that the project has met all the expected requirements and has met the expectations of the customer and the users. If that's not the case, then the question that arises is that if the project was successful. From the beginning, the organization was looking forward to a successful project and naturally the disappointment after the failure would not be a good sign for the reputation of the organization. Usually it is the perspective of the people that systems are most efficient when they work their best on the minimal cost. Success of a project is not just measured in terms of quality but also in terms of cost, time and effort. The following points are also to be considered:

How does the project affect the project budget and organization funds?

Whether the ways in which the organization works are changed by the system.

Whether the system has had any effect on the time and effort constraint of the organization.

How is this project holding up against the other similar projects?

How efficient is our algorithm really?

How well does it perform the aspects mentioned in the SRS?

Post deployment, the scope of the project to be a company.

The start-up opportunities and employment opportunities it provides.

9. PROJECT LEGACIES

9.1 CURRENT STATUS OF THE PROJECT

As per the current scenario, our project The Pi Encryption Suite is available online and currently the user client can use it to implement a cheap and open standard of security for their company.

The client can also access the source code and edit the same. The key Variations can also be done on the same level allowing modularity of the code.

Crypto random number Generation systems are also available online for free and unrestricted use.

9.2 REMAINING AREAS OF CONCERN

The system is currently completely functional but takes a toll on the maintainability. There must be people controlling the backend code optimization and we will be pretty much constantly be working on it.

9.3 LESSONS LEARNT

Technical lessons: We learnt more extensive usage of a framework ASP.NET and to work on Visual Studio IDE. We have used software like Magic Draw UML and Smart Draw and hence gained more knowledge about it.

Managerial lessons: We learnt how to manage time and at the same time, coordinate with the fellow team members.

1. Create Structure:

Specifically, regular meetings and avoid wastage of time. Ambitious deadlines push everyone beyond their comfort zones and compensate for unexpected delays.

Don't Just Listen - Understand

There should be an understanding among the team members.

Don't Underestimate Inspiration

The easiest way to inspire your team is perhaps the most obvious.

Communicate.

Structure, understanding, and inspiration depend on the one irreplaceable management skill: communication.

10. USER MANUAL

For the encryption suite.

The text you that is supposed to be applied on to the cipher will be divided into blocks and the key size can be decided later on.

The algorithm can work on multiple key sizes as well.

Matlab must be pre-installed on the user's system.

On the windows application enter your credentials provided to you.

Select what all security measures are to be employed.

Paste the text and enter Start.

The text will be ciphered accordingly and displayed on the screen beside it.

The chrome extension as long as it is enabled will autofill automatically.

More over the algorithm can be forked from GitHub and be used as an external api.

The function definitions and declaration types can be found in the package downloaded.

The application interface can be configured accordingly.

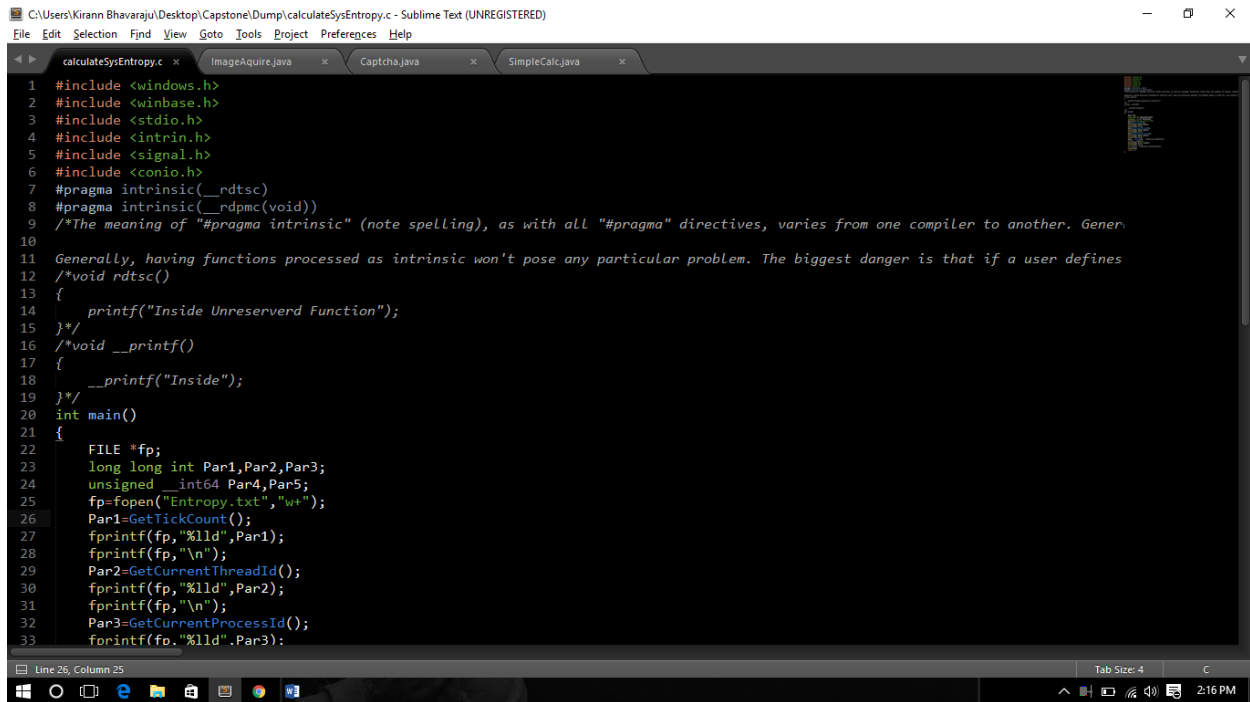
For implementing the algorithm as an application interface some level of coding expertise is required.

The languages supported are Java (or) C.

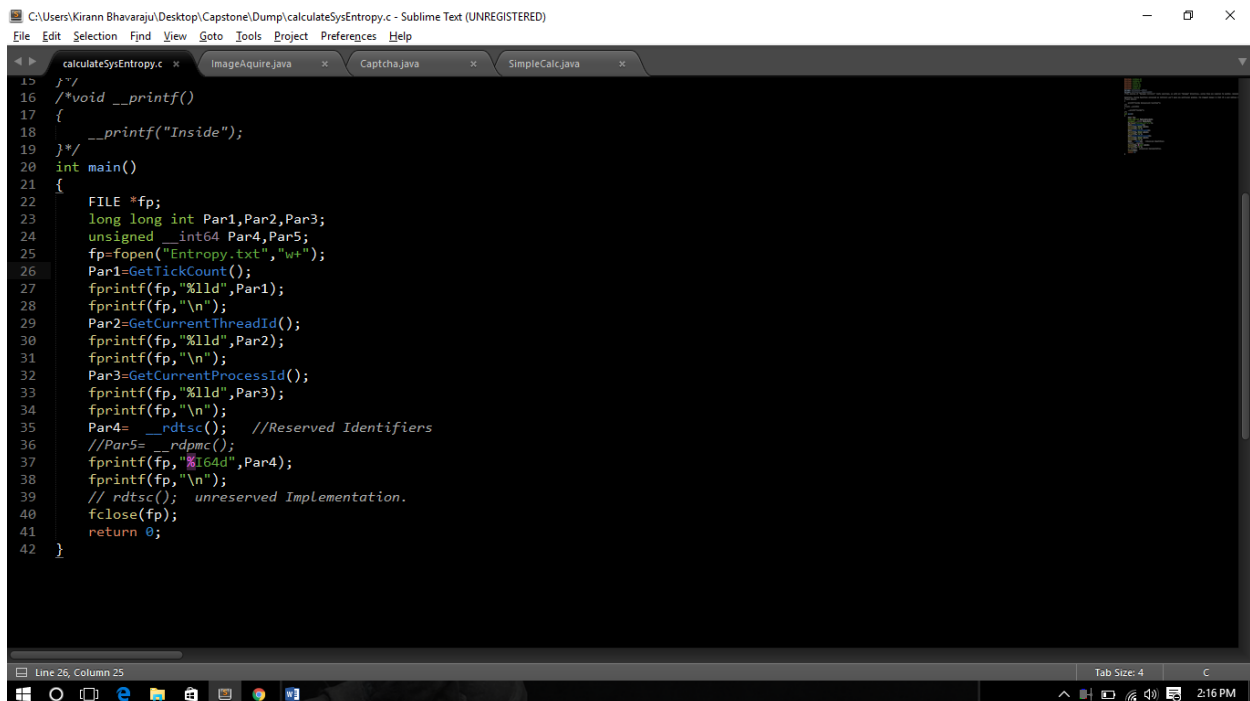
Knowledge of matlab along with its connection to java is also required.

Minimal network security skills are a must.

11. SCREENSHOTS AND CODE SNIPPETS



```
1 #include <windows.h>
2 #include <winbase.h>
3 #include <stdio.h>
4 #include <intrin.h>
5 #include <signal.h>
6 #include <conio.h>
7 #pragma intrinsic(__rdtsc)
8 #pragma intrinsic(__rdpmc(void))
9 /*The meaning of "#pragma intrinsic" (note spelling), as with all "#pragma" directives, varies from one compiler to another. Generally, having functions processed as intrinsic won't pose any particular problem. The biggest danger is that if a user defines
10
11 Generally, having functions processed as intrinsic won't pose any particular problem. The biggest danger is that if a user defines
12 */void rdtsc()
13 {
14     printf("Inside Unreserverd Function");
15 }*/
16 /*void __printf()
17 {
18     __printf("Inside");
19 }*/
20 int main()
21 {
22     FILE *fp;
23     long long int Par1,Par2,Par3;
24     unsigned __int64 Par4,Par5;
25     fp=fopen("Entropy.txt","wt");
26     Par1=GetTickCount();
27     fprintf(fp,"%lld",Par1);
28     fprintf(fp,"\n");
29     Par2=GetCurrentThreadId();
30     fprintf(fp,"%lld",Par2);
31     fprintf(fp,"\n");
32     Par3=GetCurrentProcessId();
33     fprintf(fp,"%lld",Par3);
```



```
15 */
16 /*void __printf()
17 {
18     __printf("Inside");
19 }*/
20 int main()
21 {
22     FILE *fp;
23     long long int Par1,Par2,Par3;
24     unsigned __int64 Par4,Par5;
25     fp=fopen("Entropy.txt","wt");
26     Par1=GetTickCount();
27     fprintf(fp,"%lld",Par1);
28     fprintf(fp,"\n");
29     Par2=GetCurrentThreadId();
30     fprintf(fp,"%lld",Par2);
31     fprintf(fp,"\n");
32     Par3=GetCurrentProcessId();
33     fprintf(fp,"%lld",Par3);
34     fprintf(fp,"\n");
35     Par4= __rdtsc(); //Reserved Identifiers
36     //Par5= __rdpmc();
37     fprintf(fp,"%i64d",Par4);
38     fprintf(fp,"\n");
39     // rdtsc(); unreserved Implementation.
40     fclose(fp);
41     return 0;
42 }
```

```
C:\Users\Kirann Bhavaraju\Desktop\Capstone\Dump\ImageAquire.java - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

calculateSysEntropy.c x ImageAquire.java x Captcha.java x SimpleCalc.java x

32     Long upt= 0L;
33     Long rdtscno=0L;
34     String temp="";
35     String uptime="";
36     String threadid="";
37     String prid="";
38     String rdtsc="";
39     //boolean 1;
40     //boolean 2;
41     //1= TRUE;
42     //2= FALSE;
43     FileReader input=new FileReader("Entropy.txt");
44     BufferedReader breader=new BufferedReader(input);
45     s=0;
46     while((temp=breader.readLine())!=null)
47     {
48         if(s==0)
49         {
50             uptime=uptime+temp;
51             //uptime=uptime+'\n';
52             s++;
53         }
54         else if(s==1)
55         {
56             threadid=threadid+temp;
57             s++;
58         }
59         else if(s==2)
60         {
61             prid=prid+temp;
62             s++;
63         }
64         else if(s==3)
65         {

```

```
C:\Users\Kirann Bhavaraju\Desktop\Capstone\Dump\ImageAquire.java - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

calculateSysEntropy.c x ImageAquire.java x Captcha.java x SimpleCalc.java x

1  import java.lang.*;
2  import java.awt.Robot;
3  import java.awt.Rectangle;
4  import java.awt.Toolkit;
5  import java.awt.AWTException;
6  import java.awt.image.BufferedImage;
7  import java.io.File;
8  import java.io.FileInputStream;
9  import java.io.IOException;
10 import java.io.FileReader;
11 import java.io.Reader;
12 import java.io.BufferedReader;
13 import java.io.InputStreamReader;
14 import javax.imageio.ImageIO;
15 import java.awt.Graphics;
16 import java.awt.Color;
17 import java.awt.Dimension;
18 import java.awt.Font;
19 import java.awt.AWTEventMulticaster;
20 import java.awt.event.*;
21 class CropImage
22 {
23     public void cropBufferedImage(BufferedImage image)
24     {
25     try{
26         System.out.println("Running Entropy Generation");
27         Process probject= Runtime.getRuntime().exec("a.exe");
28         System.out.println("Entropy Generation Complete");
29         System.out.println("File Entropy.txt Created and can be used for reference");
30         int s,t;
31         int thid=0,pid=0;
32         Long upt= 0L;
33         Long rdtscno=0L;
34         String temp="";

```

```
C:\Users\Kirann Bhavaraju\Desktop\Capstone\Dump\ImageAquire.java - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

calculateSysEntropy.c x ImageAquire.java x Captcha.java x SimpleCalc.java x
65 imageio.write(croppedImage, "jpg", new File("Crop1.jpg"));
66 System.out.println("Image Cropped");
67 }
68 catch(IOException exceptionobject)
69 {
70     System.out.println(exceptionobject);
71 }
72 }
73 }
74 }
75 }
76 class ImageAquire
77 {
78     public static void main(String[] args) {
79         //screencap snippet.
80         /*try
81         {
82             Robot robot=new Robot();
83             Rectangle screenRectangle=new Rectangle(Toolkit.getDefaultToolkit().getScreenSize());
84             BufferedImage fullScreenImage= robot.createScreenCapture(screenRectangle);
85             ImageIO.write(fullScreenImage,"jpeg", new File("TEST_0.jpg"));
86             System.out.println("1");
87         }
88         catch (AWTException | IOException exceptionobject)
89         {
90             System.err.println(exceptionobject);
91         }
92         */
93         // writing to an image.
94         try
95         {
96             String sample="";
97             FileInputStream in=new FileInputStream("C:/Users/Kirann Bhavaraju/Desktop/Capstone/piraw.txt");
98             int c;
99         }
100     }
101 }
```

```
C:\Users\Kirann Bhavaraju\Desktop\Capstone\Dump\ImageAquire.java - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

calculateSysEntropy.c x ImageAquire.java x Captcha.java x SimpleCalc.java x
63 }
64 else if(s==3)
65 {
66     rdtsc=rdtsc+temp;
67     s++;
68 }
69 }
70 Dimension dsize=Toolkit.getDefaultToolkit().getScreenSize();
71 double dwidth=dsize.getWidth();
72 double dheight=dsize.getHeight();
73 System.out.println(uptime);
74 System.out.println(threadid);
75 System.out.println(pid);
76 System.out.println(rdtsc);
77 breader.close();
78 upt=Long.valueOf(uptime);
79 thid=Integer.parseInt(threadid);
80 pid=Integer.parseInt(pid);
81 rdtscno=Long.valueOf(rdtsc);
82 int max1=(int)(dwidth/2);
83 int max2=(int)(dheight/2);
84 BufferedImage croppedImage= image.getSubimage((thid%max1),(pid%max2),(int)(upt%max1),(int)(rdtscno%max2));
85 ImageIO.write(croppedImage,"jpg",new File("Crop1.jpg"));
86 System.out.println("Image Cropped");
87 }
88 catch(IOException exceptionobject)
89 {
90     System.out.println(exceptionobject);
91 }
92 }
93 }
94 }
95 }
```



```
C:\Users\Kirann Bhavaraju\Desktop\Capstone\Dump\ImageAquire.java - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

calculateSysEntropy.c x ImageAquire.java x Captcha.java x SimpleCalc.java x

113
114     try
115     {
116         String sample="";
117         FileInputStream in=new FileInputStream("C:/Users/Kirann Bhavaraju/Desktop/Capstone/piraw.txt");
118         int c;
119         while((c=in.read())!= -1)
120         {
121             sample=sample+c;
122         }
123         in.close();
124         //System.out.println(sample);
125         Font piFont=new Font("Trebuchet_MS",Font.BOLD,15);
126         Dimension disp_size=Toolkit.getDefaultToolkit().getScreenSize();
127         double disp_width=disp_size.getWidth();
128         double disp_height=disp_size.getHeight();
129         BufferedImage bufferedImage=new BufferedImage((int)disp_width,(int)disp_height,BufferedImage.TYPE_INT_RGB);
130         Graphics graphics=bufferedImage.getGraphics();
131         graphics.setColor(Color.WHITE);
132         graphics.fillRect(0,0,(int)disp_width,(int)disp_height);
133         graphics.setFont(piFont);
134         graphics.setColor(Color.BLACK);
135         int x=15;
136         char [] digits=sample.toCharArray();
137         //System.out.println(digits);
138         //graphics.drawString(sample,0,15);
139         int lines=0;
140         for(int k=0;k<((int)disp_height/17;k++)
141         {
142             graphics.drawChars(digits,lines,(int)(disp_width/8),0,x);
143             x+=17;
144             lines=lines+(int)(disp_width/8);
145         }
146         ImageIO.write(bufferedImage,"jpg",new File("TEST_0.jpg"));
147     }
148 }
149
150
151
152
153
154
155
156
157

Line 24, Column 6
Tab Size: 4
Java
2:16 PM
```

```
C:\Users\Kirann Bhavaraju\Desktop\Capstone\Dump\ImageAquire.java - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

calculateSysEntropy.c x ImageAquire.java x Captcha.java x SimpleCalc.java x

137         //System.out.println(digits);
138         //graphics.drawString(sample,0,15);
139         int lines=0;
140         for(int k=0;k<((int)disp_height/17;k++)
141         {
142             graphics.drawChars(digits,lines,(int)(disp_width/8),0,x);
143             x+=17;
144             lines=lines+(int)(disp_width/8);
145         }
146         ImageIO.write(bufferedImage,"jpg",new File("TEST_0.jpg"));
147         System.out.println("Main Image Created");
148         CropImage cropper= new CropImage();
149         cropper.cropBufferedImage(bufferedImage);
150     }
151     catch(IOException exceptionobject)
152     {
153         System.err.println(exceptionobject);
154     }
155 }
156
157

Line 24, Column 6
Tab Size: 4
Java
2:16 PM
```

```
C:\Users\Kirann Bhavaraju\Desktop\Capstone\Dump\Captcha.java - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

calculateSysEntropy.c x ImageAquire.java x Captcha.java x SimpleCalc.java x

1 import java.awt.*;
2 import java.awt.event.*;
3 import java.io.*;
4 //import java.swing.*;
5 class Captcha extends JFrame implements ActionListener
6 {
7     int flag=0;
8     String captcha="";
9     String captchaFin="";
10    Button b1;
11    TextField tf=new TextField();
12    Captcha()
13    {
14        add(tf,BorderLayout.SOUTH);
15        Panel p=new Panel(new GridLayout(4,4));
16        setSize(200,400);
17        setTitle("Random Captcha Generation");
18        b1=new Button("GENERATE");
19        p.add(b1);
20        add(p);
21        b1.addActionListener(this);
22        setVisible(true);
23    }
24    public void actionPerformed(ActionEvent ae)
25    {
26        if(flag==1)
27        {
28            tf.setText("");
29        }
30        if(ae.getSource()==b1)
31        {
32            captchaFin=this.randomCaptcha();
33            tf.setText(captchaFin);
34            captchaFin="a";
35        }
36    }
37    }
38    public String randomCaptcha()
39    {
40        try{
41            FileInputStream input=new FileInputStream("text.txt");
42            int c;
43            int counter=0;
44            while((c=input.read())!=-1)
45            {
46                if(counter<5)
47                {
48                    captcha=captcha+c;
49                    counter++;
50                }
51                else
52                {
53                    input.close();
54                    break;
55                }
56            }
57        }
58        catch(FileNotFoundException fe)
59        {
60            System.out.println("FileNotFoundException");
61        }
62        catch(IOException ae)
63        {
64            System.out.println("IOException");
65        }
66        return captcha;
67    }
68    }
```

```
C:\Users\Kirann Bhavaraju\Desktop\Capstone\Dump\Captcha.java - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

calculateSysEntropy.c x ImageAquire.java x Captcha.java x SimpleCalc.java x

35         flag=1;
36     }
37 }
38 public String randomCaptcha()
39 {
40     try{
41         FileInputStream input=new FileInputStream("text.txt");
42         int c;
43         int counter=0;
44         while((c=input.read())!=-1)
45         {
46             if(counter<5)
47             {
48                 captcha=captcha+c;
49                 counter++;
50             }
51             else
52             {
53                 input.close();
54                 break;
55             }
56         }
57     }
58     catch(FileNotFoundException fe)
59     {
60         System.out.println("FileNotFoundException");
61     }
62     catch(IOException ae)
63     {
64         System.out.println("IOException");
65     }
66     return captcha;
67 }
68 }
```

```
C:\Users\Kiran Bhavaraju\Desktop\Capstone\Dump\Captcha.java - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

calculateSysEntropy.c x ImageAquire.java x Captcha.java x SimpleCalc.java x

56     }
57     }
58     catch(FileNotFoundException fe)
59     {
60         System.out.println("FileNotFoundException");
61     }
62     catch(IOException ae)
63     {
64         System.out.println("IOException");
65     }
66     return captcha;
67 }
68
69 public static void main(String[] args)
70 {
71     new Captcha();
72 }
73 }
```

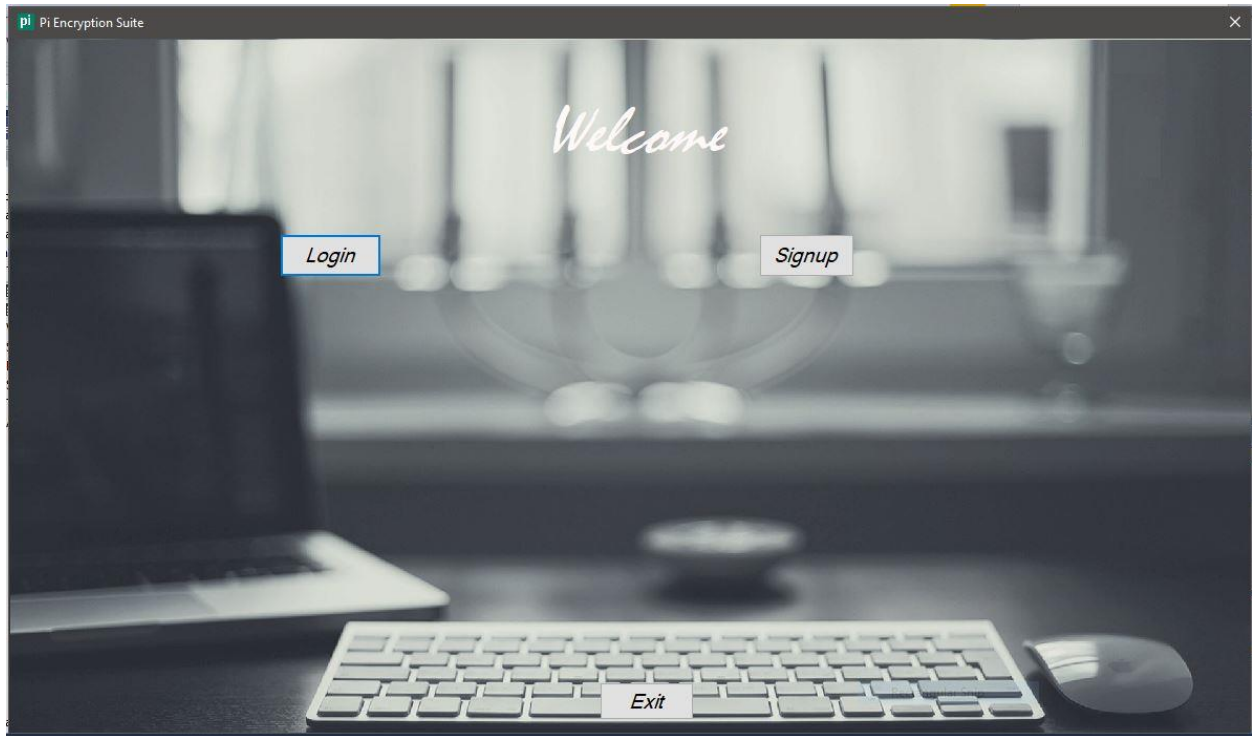
Line 11, Column 34 Tab Size: 4 Java 2:16 PM

```
Editor - C:\Users\Kiran Bhavaraju\Desktop\Capstone\Dump\OCR.m
File Edit Text Go Cell Tools Debug Desktop Window Help

1 warning off
2 clc, close all, clear all
3 imagen=imread('Crop1.jpg');
4 imshow(imagen);
5 title('INPUT IMAGE WITH NOISE')
6 if size(imagen,3)==3 %RGB image
7     imagen=rgb2gray(imagen);
8 end
9 threshold = graythresh(imagen); %convert to BW/GrayScale
10 imagen = ~im2bw(imagen,threshold);
11 imagen = bwareaopen(imagen,25); %less than 25 pixels will not be considered.
12 word=[];
13 re=imagen;
14 fid = fopen('text.txt','wt');
15 load templates
16 global templates
17 num_letras=size(templates,2);
18 while 1
19     [f1 re]=lines(re);
20     imgn=f1; imshow(f1);pause(0.5); % use this for displaying the images one by one.
21     [L Ne] = bwlabel(imgn);
22     for n=1:Ne
23         [r,c] = find(L==n);
24         nl=imgn(min(r):max(r),min(c):max(c)); % Extract letter
25         img_r=imresize(nl,[42 24]); %imshow(img_r);pause(0.5) %to show one by one.
26         letter=read_letter(img_r,num_letras); % Calling fcn to convert image to text
27         word=[word letter];
28     end
29     fprintf(fid,'%s',word); %Write 'word' in text file (upper) for lower case fprintf(fid,'%s\n',lower(word));
30     word=[];
31     if isempty(re)
32         break
33     end
34 end
```

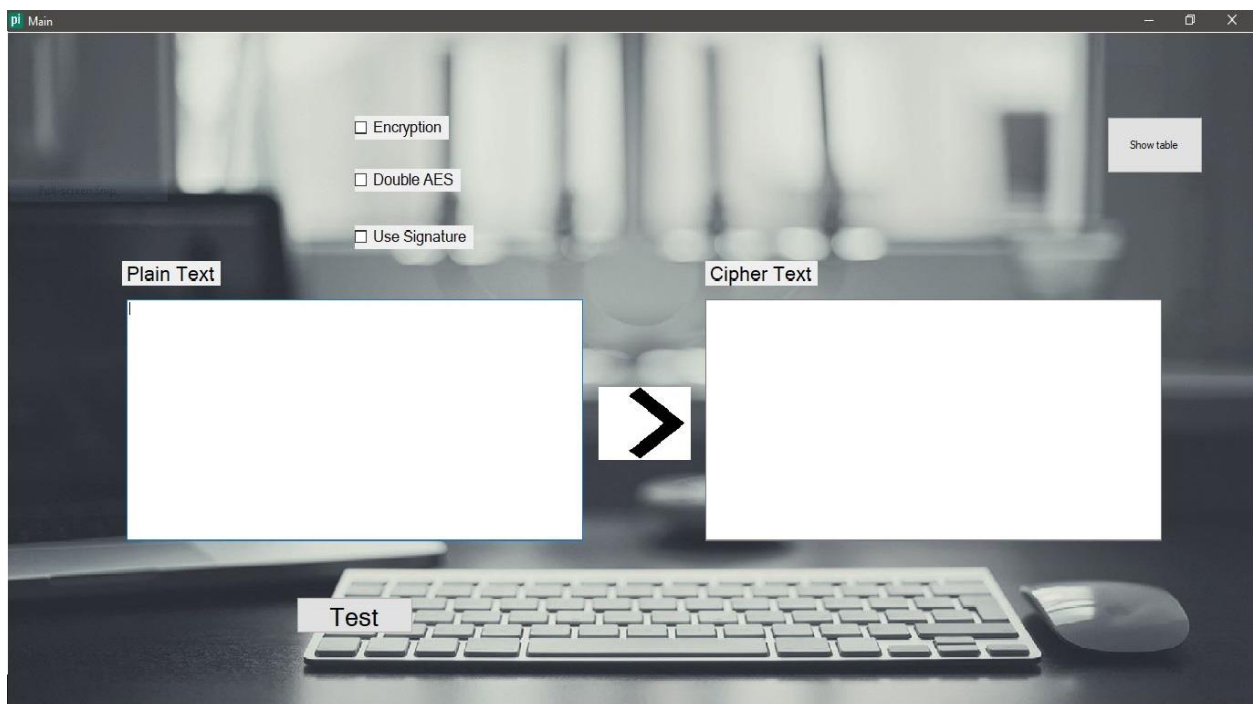
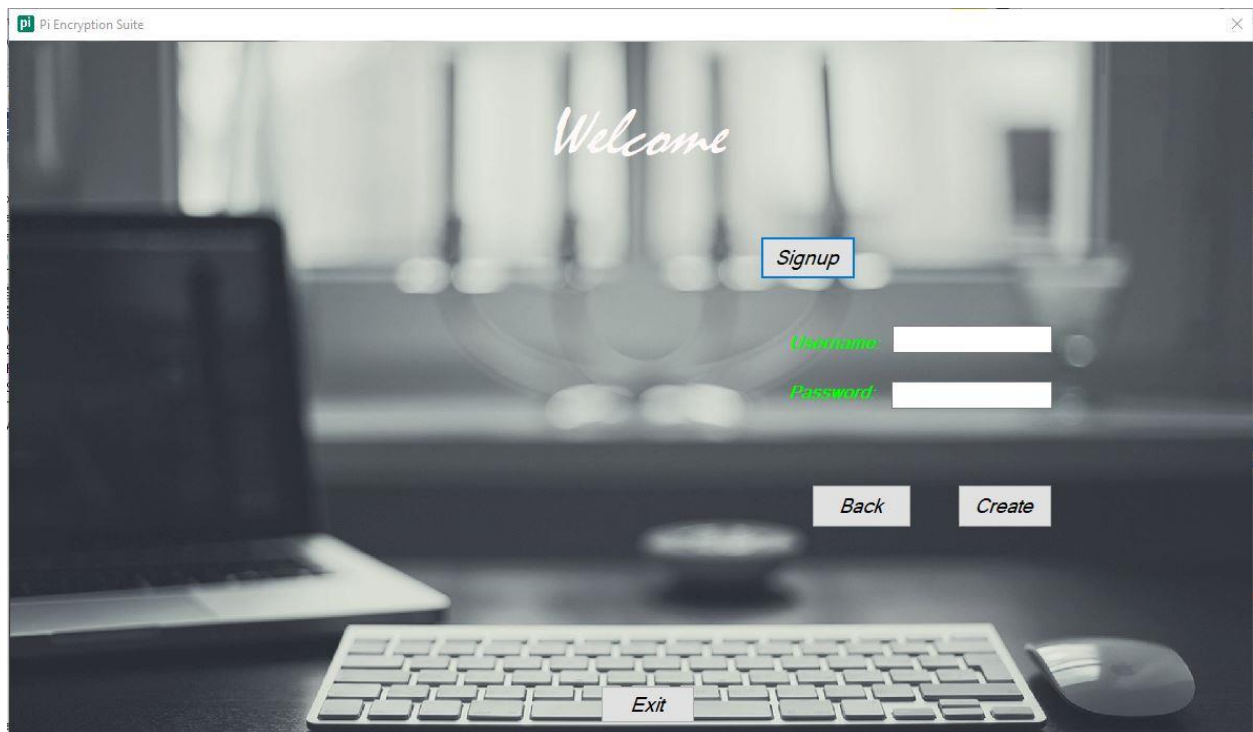
script Ln 1 Col 1 OVR 2:26 PM

WINDOWS APPLICATION SCREENSHOTS

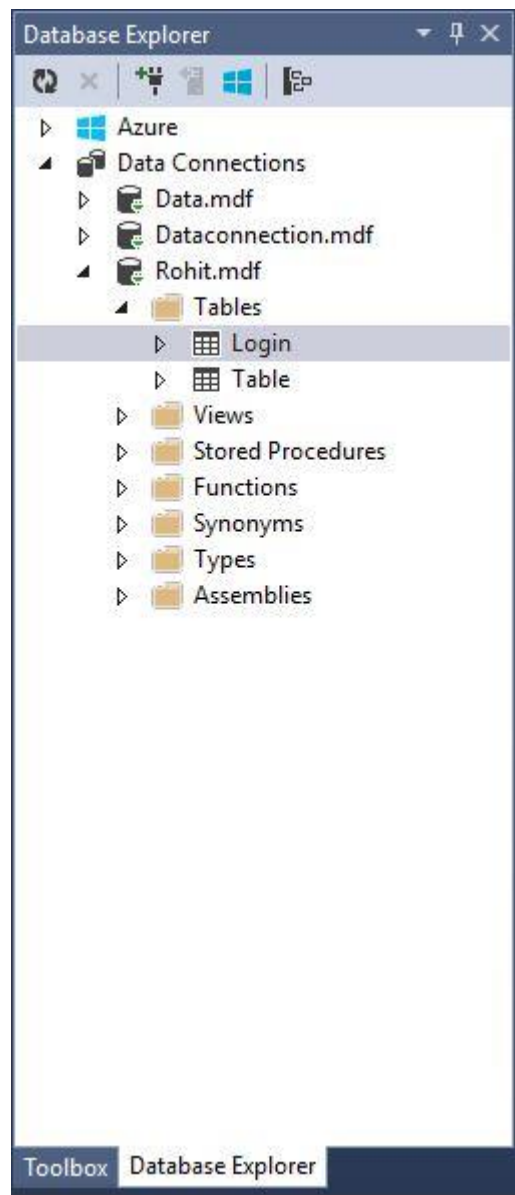


Login page





Database tables



Main.cs [Design] Form1.cs [Design] Work

Work Work.Form1 button7_Click(object sender, EventArgs e)

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace Work
{
    public partial class Form1 : Form
    {
        SqlConnection conn = new SqlConnection(@"Data Source=(LocalDB)\v11.0;AttachDbFilename=C:\Users\Rohit\Documents\Rohit.mdf;Integrated Security=SSPI;");

        public Form1()
        {
            InitializeComponent();
            panel1.Hide();
            panel2.Hide();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            panel1.Show();
            button2.Hide();
            button1.Hide();
        }

        private void panel1_Paint(object sender, PaintEventArgs e)
        {
        }

        private void button5_Click(object sender, EventArgs e)
        {
            this.Close();
        }

        private void button3_Click(object sender, EventArgs e)
        {
            button1.Show();
            button2.Show();
            panel1.Hide();
        }

        private void button4_Click(object sender, EventArgs e)
        {
            SqlConnection conn = new SqlConnection(@"Data Source=(LocalDB)\v11.0;AttachDbFilename=C:\Users\Rohit\Documents\Rohit.mdf;Integrated Security=SSPI;");
            SqlDataAdapter sda = new SqlDataAdapter("Select Count(*) from Login where Username = '" + textBox2.Text + "' and Password = '" + textBox3.Text + "'", conn);
            DataTable dt = new DataTable();
            sda.Fill(dt);
            if (dt.Rows[0][0].ToString() == "1")
            {
                this.Hide();
                Main ss = new Main();
                ss.Show();
            }
            else
            {
                MessageBox.Show("Invalid Login Credentials");
                textBox2.Clear();
                textBox3.Clear();
            }
        }

        private void button2_Click(object sender, EventArgs e)
        {
            button1.Hide();
            panel1.Hide();
            panel2.Show();
        }

        private void button6_Click(object sender, EventArgs e)
        {
            button1.Show();
            panel2.Hide();
        }

        private void button7_Click(object sender, EventArgs e)
        {
            conn.Open();
            String query = "Insert INTO Login (Username,Password) VALUES ('" + textBox4.Text + "','" + textBox1.Text + "')";
            SqlDataAdapter SDA = new SqlDataAdapter(query, conn);
            SDA.SelectCommand.ExecuteNonQuery();
            conn.Close();
            MessageBox.Show("Account Successfully Created.");
        }
    }
}

```



```
1 //successful
2 //pi generator
3 #include<stdio.h>
4 #include<stdlib.h>
5
6 #define SCALE 10000
7 #define ARRINIT 2000
8
9 void pi_digits(int digits) {
10     int carry = 0;
11     int count=0;
12     int arr[digits + 1];
13     for (int i = 0; i <= digits; ++i)
14         arr[i] = ARRINIT;
15     for (int i = digits; i > 0; i-= 14) {
16         int sum = 0;
17         for (int j = i; j > 0; --j) {
18             sum = sum * j + SCALE * arr[j];
19             arr[j] = sum % (j * 2 - 1);
20             sum /= j * 2 - 1;
21         }
22         printf("%0.4d", carry + sum / SCALE);
23         count++;
24         carry = sum % SCALE;
25     }
26     count=count*4;
27     printf("\n");
28     printf("%d", count);
29 }
30
31 int main(int argc, char** argv) {
32     int n = argc == 2 ? atoi(argv[1]) : 100;
33     pi_digits(400000);
34
35     return 0;
36 }
```

piraw - Notepad
File Edit Format View Help
3141
5926
5358
9793
2384
6264
3383
2795
0288
4197
1693
9937
5105
8209
7494
4592
3078
1640
6286
2089
9862
8034
8253
4211
7067
9821
4808
6513
2823
0664
7093
8446
0955
0582
2317
2535
9408


```

1 #include <stdio.h> //this program generates numbered key and
2 #include <stdlib.h>
3
4 int main()
5 {
6
7     FILE *myFile;
8     myFile = fopen("piraw.txt", "r");
9     FILE *fp = fopen("piraw-fin.txt", "w");
10
11     //read file into array
12     int numberArray[35715];
13     int i,a,b,c,d,value;
14
15     if (myFile == NULL)
16     {
17         printf("Error Reading File\n");
18         exit (0);
19     }
20     for (i = 0; i < 35715; i++)
21     {
22         fscanf(myFile, "%d", &numberArray[i] );
23     }
24
25     for (i = 0; i < 35715; i++)
26     {
27         value= numberArray[i];
28         a=numberArray[i]%10;
29         numberArray[i]=numberArray[i]-a;
30         numberArray[i]=numberArray[i]/10;
31         b=numberArray[i]%10;
32         numberArray[i]=numberArray[i]-b;
33         numberArray[i]=numberArray[i]/10;
34         c=numberArray[i]%10;
35         numberArray[i]=numberArray[i]-c;
36         numberArray[i]=numberArray[i]/10;
37         d=numberArray[i]%10;
38         numberArray[i]=numberArray[i]-d;
39
40         printf("Number is: %d\n", value);
41         fprintf(fp,"%d",d);
42         fputc("\n",fp);
43         fprintf(fp,"%d",c);
44         fputc("\n",fp);
45         fprintf(fp,"%d",b);
46         fputc("\n",fp);
47         fprintf(fp,"%d",a);
48         fputc("\n",fp);
49     }
50
51     fclose(myFile);
52     fclose(fp);
53
54     return 0;
55 }
56

```

piraw-fin - Notepad

File Edit Format View Help

b

1

4

1

5

9

2

6

5

3

5

8

9

7

9

3

2

3

8

4

6

2

6

4

3

3

8

3

2

7

9

5

0

2

8

8

4

1

9

7

1

6

9

3

9

9

3

7

5

1

0

5

8

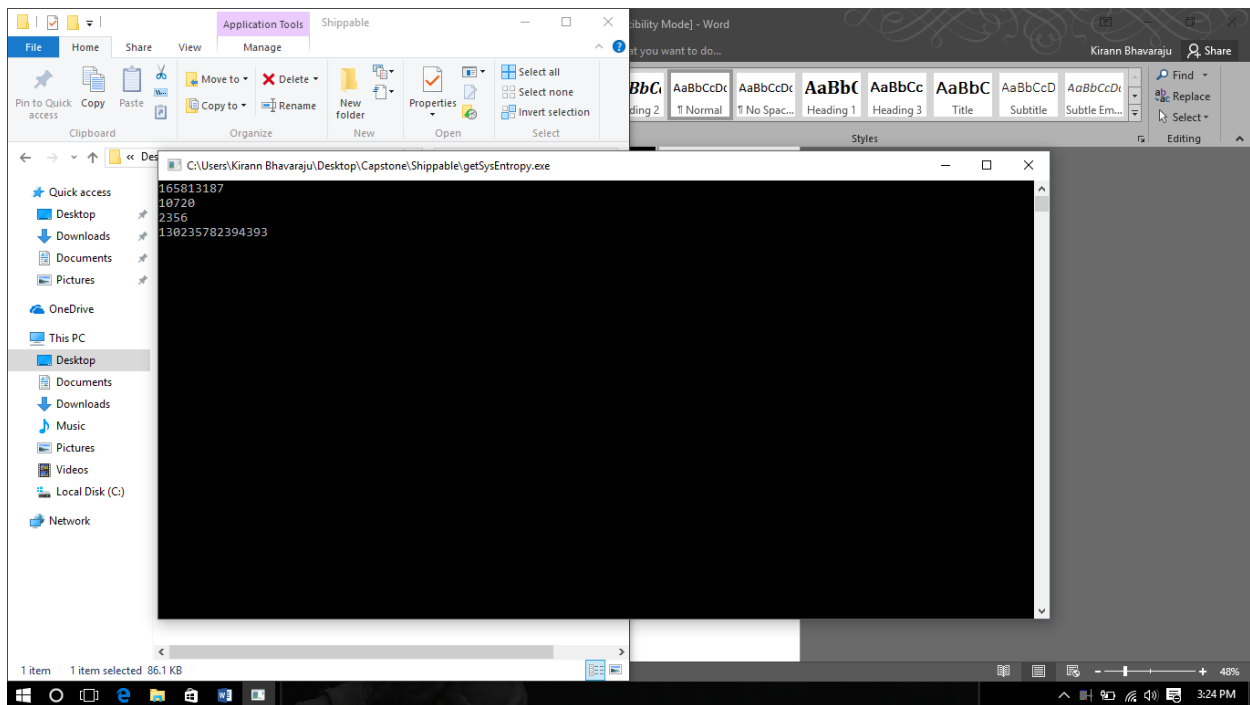
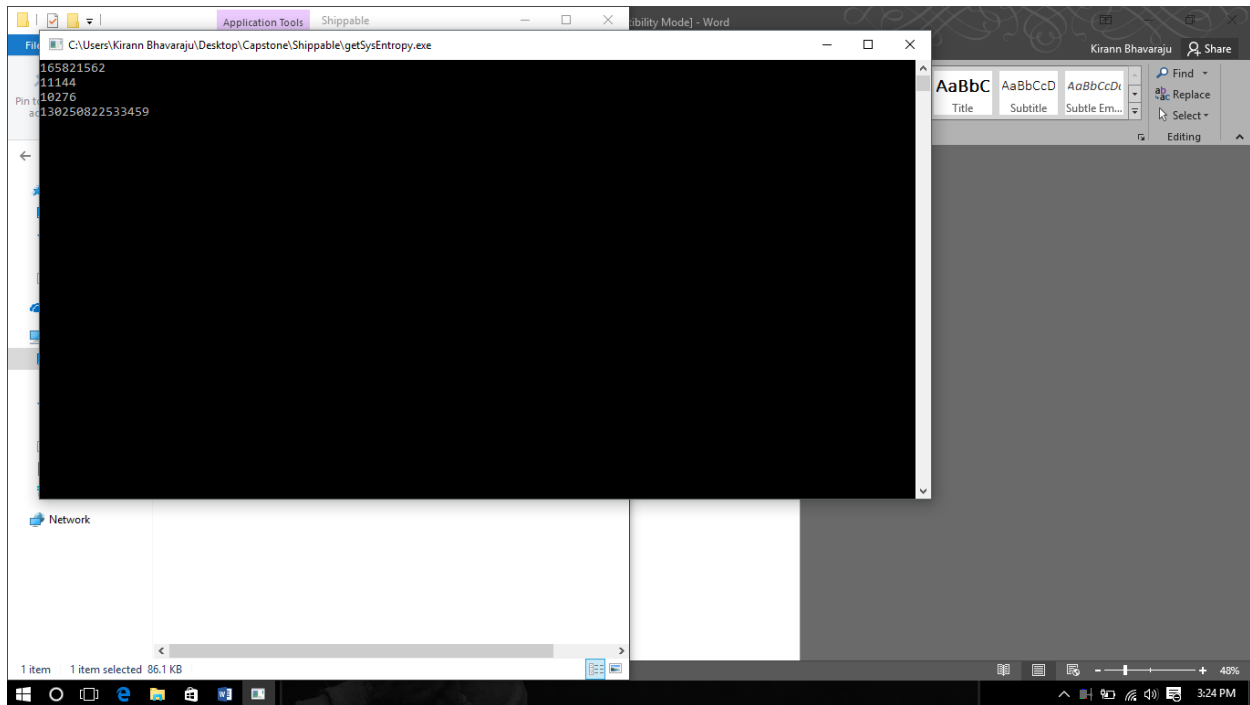
2

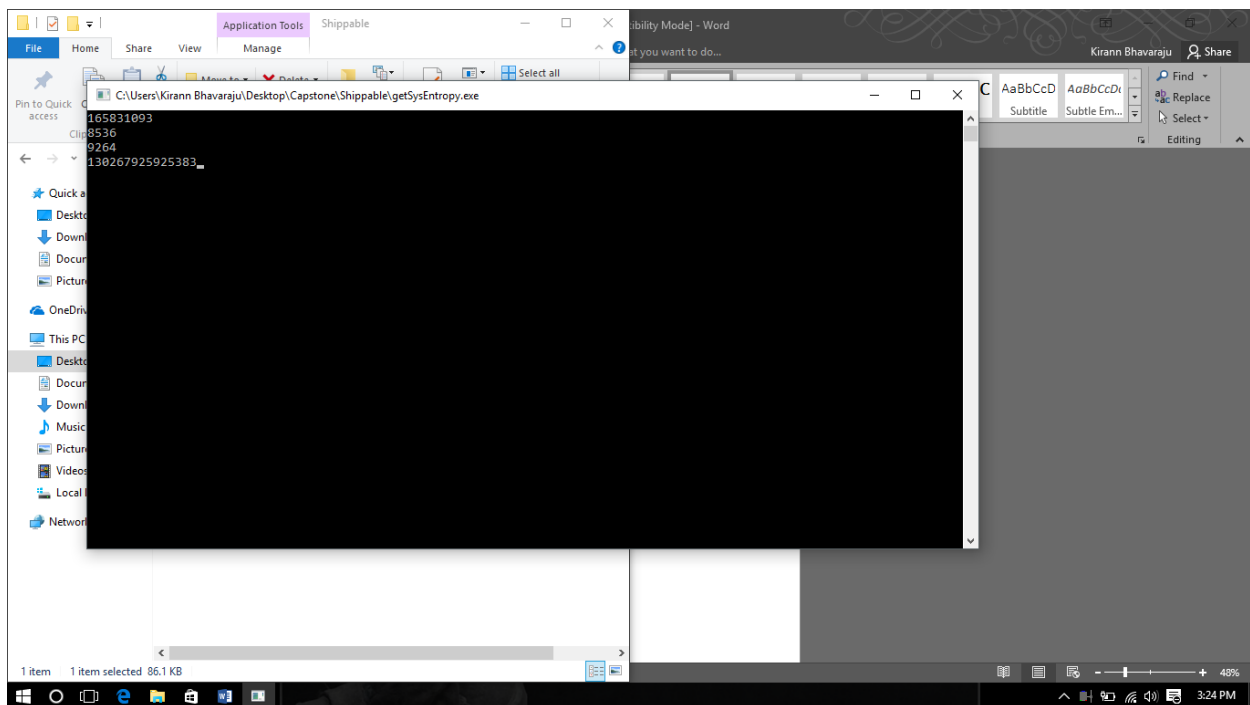
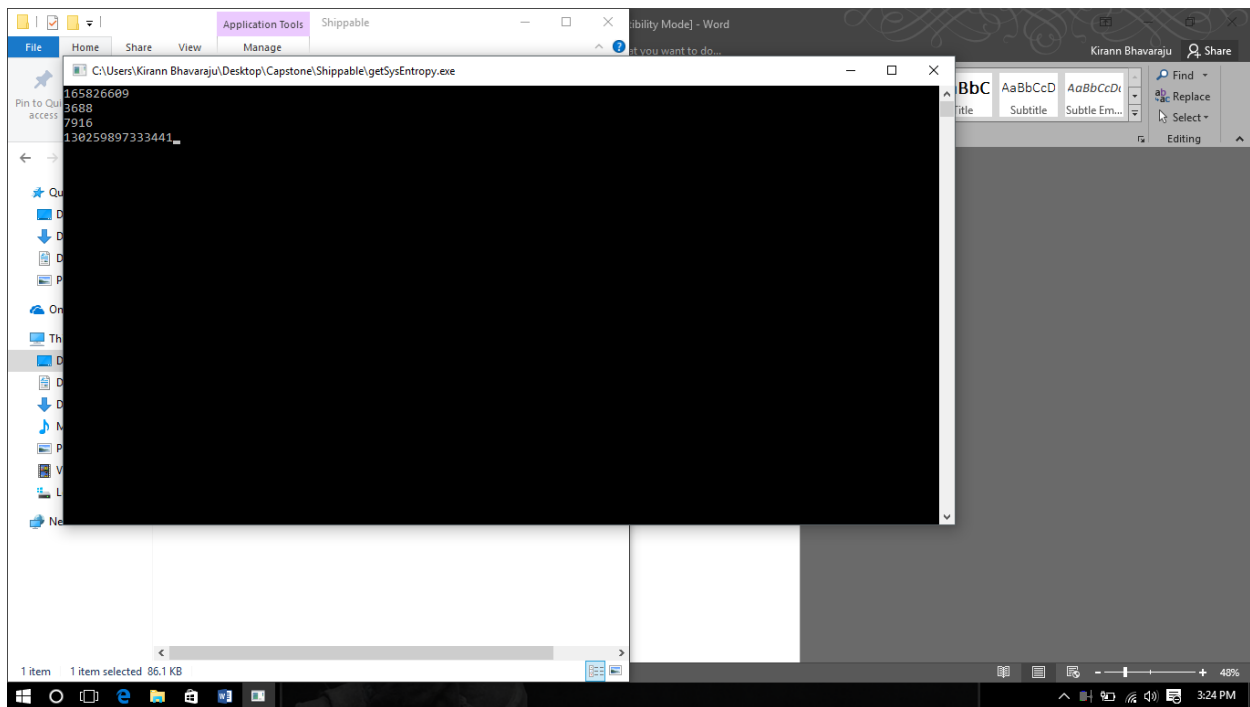
0

9

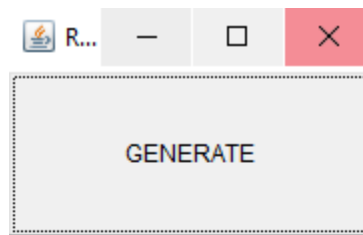
7

Random number generation screenshots

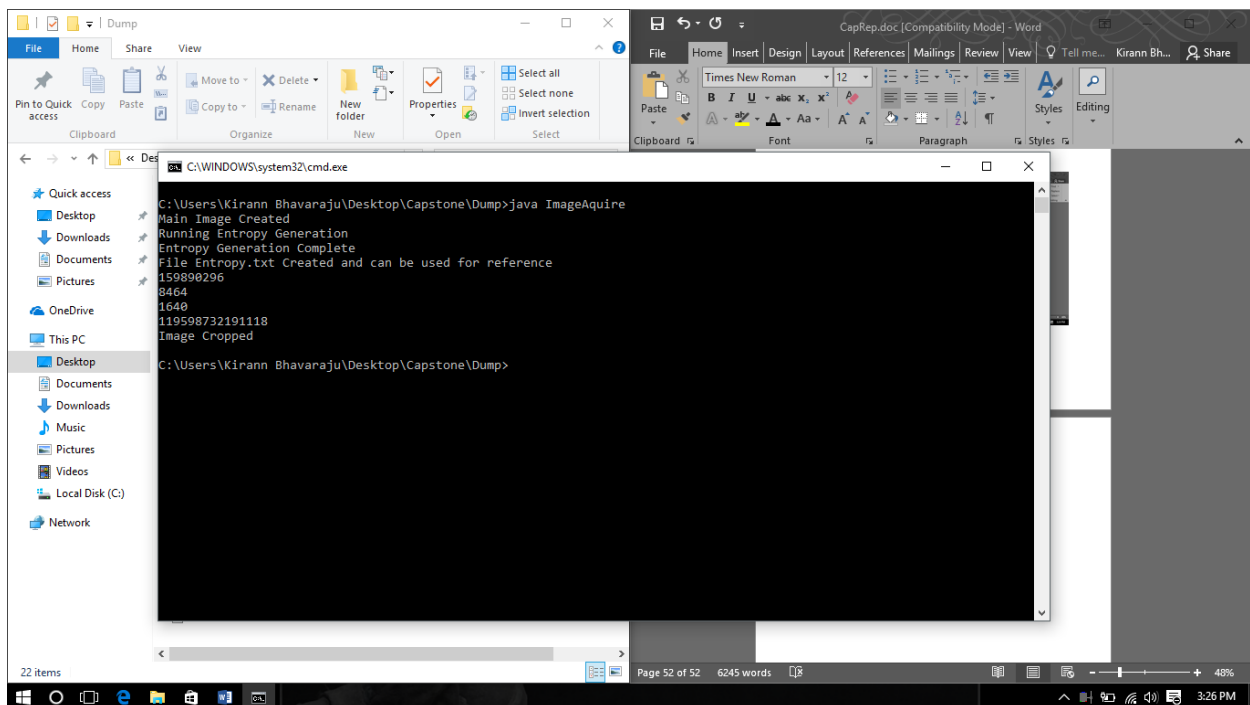


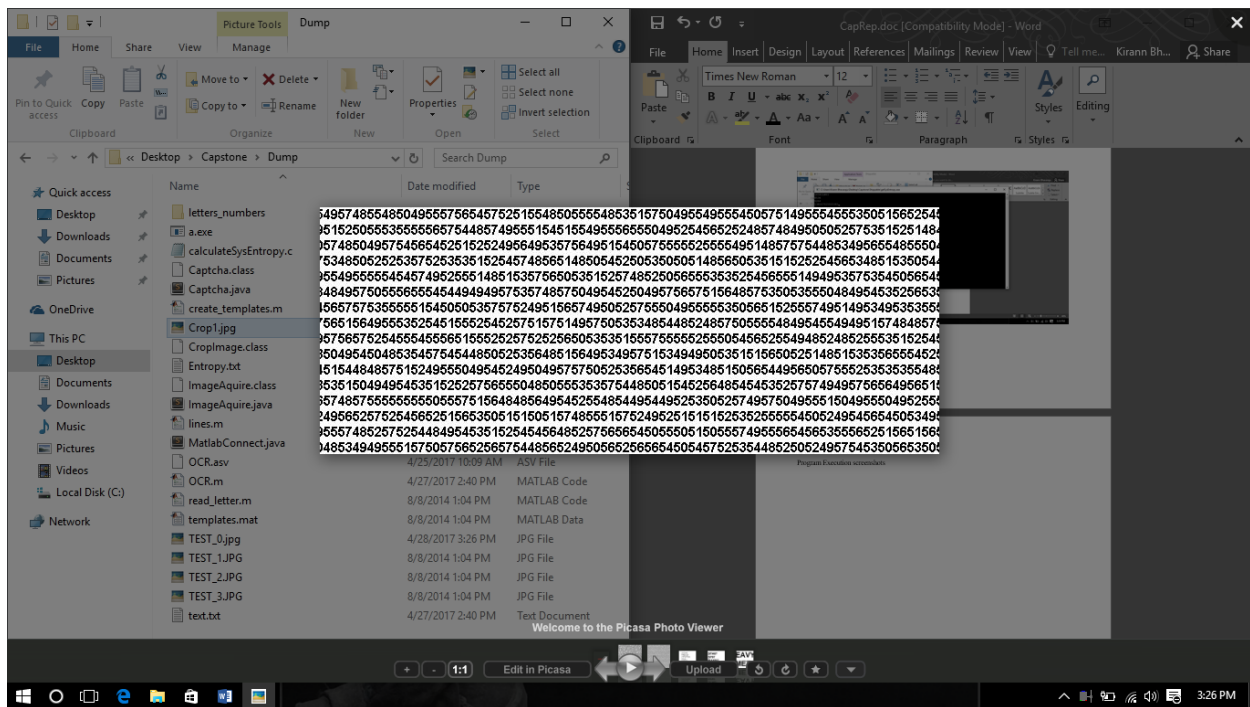


Program Execution screenshots



8585824882





12. BIBLIOGRAPHY

[https://msdn.microsoft.com/en-us/library/windows/hardware/ff553516\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/hardware/ff553516(v=vs.85).aspx)

<http://www.unixwiz.net/techtips/gnu-c-attributes.html>

[https://msdn.microsoft.com/en-us/library/ms724408\(VS.85\).aspx](https://msdn.microsoft.com/en-us/library/ms724408(VS.85).aspx)

<https://code.google.com/archive/p/matlabcontrol/>

<http://lxr.free-electrons.com/source/tools/perf/arch/x86/tests/rdpmc.c>

<https://octave.sourceforge.io/image/index.html>