Shameer Mukhtar **(SP20-BSE-076)**

**Subject: Object Oriented Software Engineering**

Contents

[Chapter 1 Project Proposal 3](#_Toc83820206)

[Introduction 1](#_Toc83820207)

[Module and description 3](#_Toc83820208)

[Software project plan 9](#_Toc83820209)

[client requirement determination 10](#_Toc83820210)

[Use-Case diagram 13](#_Toc83820211)

[Brief cases and fully dressed 14](#_Toc83820212)

[Domain model and SSDs 18](#_Toc83820214)

[operation contracts and package diagram 22](#_Toc83820215)

[System design 23](#_Toc83820216)

[Screen shots 24](#_Toc83820219)

[Conclusion pending](#_Toc83820219)

INTRODUCTION

PROJECT DESCRIPTION

In **t**his project Encryption is the conversion of data into a form, called a cipher text that cannot be easily understood by unauthorized people. Decryption is the process of converting encrypted data back into its original form, so it can be understood.

The use of encryption/decryption is as old as the art of communication. In wartime, a cipher, often incorrectly called a "code," can be employed to keep the enemy from obtaining the contents of transmissions. (Technically, a code is a means of representing a signal without the intent of keeping it secret; examples are Morse code and ASCII.) Simple ciphers include the substitution of letters for numbers, the rotation of letters in the alphabet, and the "scrambling" of voice signals by inverting the sideband frequencies. More complex ciphers work according to sophisticated computer an algorithm that rearranges the data bits in digital signals.

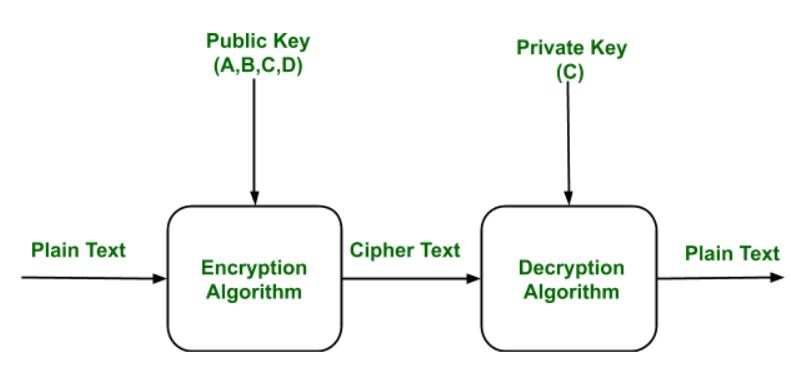
In order to easily recover the contents of an encrypted signal, the correct decryption key is required. The key is an algorithm that "undoes" the work of the encryption algorithm. Alternatively, a computer can be used in an attempt to "break" the cipher. The more complex the encryption algorithm, the more difficult it becomes to eavesdrop on the communications without access to the key.

Encryption/decryption is especially important in wireless communications. This is because wireless circuits are easier to "tap" than their hard-wired counterparts. Nevertheless, encryption/decryption is a good idea when carrying out any kind of sensitive transaction, such as a credit-card purchase online, or the discussion of a company secret between different departments in the organization. The stronger the cipher – that is, the harder it is for unauthorized people to break it – the better, in general. However, as the strength of encryption/decryption increases, so does the cost.

Crypto framework is answerable for encoding the client's information and give a protected system to store it in a virtual drive. This virtual drive will be made by the framework for the specific client for the absolute first while utilizing the framework. This framework will give restricted capacity region where the information can be saved. As we can say that it's a distributed storage medium where information can be gotten to from any area. This framework will likewise empower you to synchronize your work area or PC while utilizing this crypto framework.

For carrying out security instrument, every client's ought to have a legitimate login id and secret key and the confirmation of client's record will be done through their substantial email id. Upon getting to this crypto framework, clients can essentially move their archives from their framework to their virtual hard drive where it will require some investment for scrambling the reports and cycle of encryption and unscrambling will rely on the size of record and their sort. Easy to understand interface has been given, with the goal that clients can undoubtedly get to every one of the given highlights. It utilizes solid cycle encryption system and its virtual hard drive and handily stacked and dumped at any area according to the client's decision.

In network security, cryptography has a long history by provides a way to store sensitive information or transmit it across insecure networks



Decryption keys would be stored in a supposedly secure place, used only by authorities, and used only if backed up by a court order. Opponents of this scheme argue that criminals could hack into the key-escrow database and illegally obtain, steal, or alter the keys..

**Objection of the project:**

In Order to be able to define our system architecture, we must first dearly state what our objective that will deliver system behavior at the same one of our objective is to create an experience, which is not only unique to the (user) client, but also makes him feel that he has loyal attachment to the system and approaches us whenever he/she needs.

To achieve better results and success by implement computerized process instead of manual process.

**Modules and their Description**

**Admin**

**The login module consists of username and password. This process is for authentication .The username and password is correct it is link into next page. This process is done in login.**

**Symmetric Key/Algorithms**

* **AES Algorithm**
* **RC4 Algorithm**
* **Triple DES Algorithm**
* **RSA Algorithm**

**AES (Advanced Encryption Standard**)

In cryptography, the Advanced Encryption Standard (AES) is a symmetric-key encryption standard adopted by the U.S. government. The standard comprises three block ciphers, AES-128, AES-192 and AES-256, adopted from a larger collection originally published as Rijndael. Each of these ciphers has a 128-bit block size, with key sizes of 128, 192 and 256 bits, respectively.

AES was announced by National Institute of Standards and Technology (NIST) as U.S. FIPS PUB 197 (FIPS 197) on November 26, 2001 after a 5-year standardization process in which fifteen competing designs were presented and evaluated before Rijndael was selected as the most suitable (see Advanced Encryption Standard process for more details). It became effective as a Federal government standard on May 26, 2002 after approval by the Secretary of Commerce.

The Rijndael cipher was developed by two Belgian cryptographers, Joan Daemen and Vincent Rijmen, and submitted by them to the AES selection process.[4] Rijndael (Dutch pronunciation: [ˈrɛindaːl][5]) is a wordplay based upon the names of the two inventors.

**RC4 Encryption Algorithm**

The security of data has become a recurrent topic in computer science. I think all software developers in their careers have to study that topic. I always keep informed about that, and I apply various kinds of algorithms into the several applications customers ask me to develop.

RC4 is a stream cipher symmetric key algorithm. It was developed in 1987 by Ronald Rivest and kept as a trade secret by RSA Data Security. On September 9, 1994, the RC4 algorithm

RC4 uses a variable length key from 1 to 256 bytes to initialize a 256-byte state table. The state table is used for subsequent generation of pseudo-random bytes and then to generate a pseudo-random stream which is XOR-end with the plaintext to give the cipher text. Each element in the state table is swapped at least once.

The RC4 key is often limited to 40 bits, because of export restrictions but it is sometimes used as a 128 bit key. It has the capability of using keys between 1 and 2048 bits. RC4 is used in many commercial software packages such as Lotus Notes and Oracle Secure SQL. It is also part of the Cellular Specification.

## ALGORITHM DESCRIPTION

The RC4 algorithm works in two phases:

1. Key Setup
2. Ciphering.

### Key setup

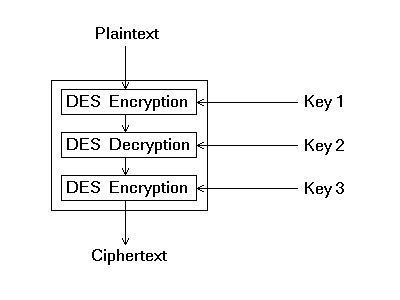
Key setup is the first and most difficult phase of this algorithm. During a N-bit key setup (N being your key length), the encryption key is used to generate an encrypting variable using two arrays, state and key, and N-number of mixing operations.

In the attached project you can see how I do it in the EncryptionKey set property Once the encrypting variable is produced from the key setup, it enters the ciphering phase, where it is XOR-ed with the plain text message to create an encrypted message. If the bits are the same, the result is 0. Once the receiver gets the encrypted message, he decrypts it by XOR-ing the encrypted message with the same encrypting variable. In the attached project you can see how I do it in the RC4Engine class:

* Encrypt: encript method
* Decrypt: decript method

I want to remark that the crypted message comes decrypted using the algorithm used in the encryption phase.

**Triple DES:**

Triple DES is simply another mode of DES operation. It takes three 64-bit keys, for an overall key length of 192 bits. In Private Encryptor, you simply type in the entire 192-bit (24 character) . The Triple DES DLL then breaks the user provided key into three subkeys, padding the keys if necessary so they are each 64 bits long. The procedure for encryption is exactly the same as regular DES, but it is repeated three times key.

Consequently, Triple DES runs three times slower than standard DES, but is much more secure if used properly. The procedure for decrypting something is the same as the procedure for encryption, except it is executed in reverse. Like DES, data is encrypted and decrypted in 64-bit chunks. Unfortunately, there are some weak keys that one should be aware of: if all three keys, the first and second keys, or the second and third keys are the same, then the encryption procedure is essentially the same as standard DES.. Note that although the input key for DES is 64 bits long, the actual key used by DES is only 56 bits in length. These parity bits are ignored, so only the seven most significant bits of each byte are used, resulting in a key length of 56 bits. This means that the effective key strength for Triple DES is actually 168 bits because each of the three keys contains 8 parity bits that are not used during the encryption process.

**RSA Algorithm**

The RSA algorithm is named after Ron Rivest, Adi Shamir and Len Adleman, who invented it in 1977 [RIVE78]. The basic technique was first discovered in 1973 by Clifford Cocks [COCK73] of CESG (part of the British GCHQ) but this was a secret until 1997. The patent taken out by RSA Labs has expired.

The RSA algorithm can be used for both public key encryption and digital signatures. Its security is based on the difficulty of factoring large integers.

Key Generation Algorithm

1. Generate two large random primes, p and q, of approximately equal size such that their product n = pq is of the required bit length, e.g. 1024 bits. [See note 1].q

2. Compute n = pq and (φ) phi = (p-1)(q-1).

3. Choose an integer e, 1 < e < phi, such that gcd (e, phi) = 1. [See note 2].

4. Compute the secret exponent d, 1 < d < phi, such that Ed ≡ 1 (mod phi). [See note 3].

5. The public key is (n, e) and the private key is (n, d). Keep all the values d, p, q and phi secret.

* n is known as the modulus.
* e is known as the public exponent or encryption exponent or just the exponent.
* d is known as the secret exponent or decryption exponent.

**Encryption**

Sender A does the following:-

* Obtains the recipient B's public key (n, e).
* Represents the plaintext message as a positive integer m [see note 4].
* Computes the ciphertext c = me mod n.
* Sends the ciphertext c to B.

**Decryption**

Recipient B does the following:-

* Uses his private key (n, d) to compute m = cd mod n.
* Extracts the plaintext from the message representative m.

**Signature verification**

Recipient B does the following:-

* Uses sender A's public key (n, e) to compute integer v = se mod n.
* Extracts the message digest from this integer.
* Independently computes the message digest of the information that has been signed.
* If both message digests are identical, the signature is valid.

**SOFTWARE PROJECT PLAN**

This part discusses about that time schedule for the project and it contains the various phases of the project.

**The Various Phases of the Project:**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **TASK** | **DURATION** |
| **1** | **Requirement Specification** | **7 Day’s** |
| **2** | **Use cases** | **5 Day’s** |
| **3** | **System sequence diagram/domain model** | **5 Day’s** |
| **4** | **Package diagram/analysis** | **2 Day’s** |
| **5** | **Documentation/ Review** | **4 Day’s** |
| **6** | **Coding** | **In process** |
|  | **Total** | **pending** |

**CLIENT REQUIREMENTS DETERMINATION**

**ABOUT SYSTEM**

To overcome all the problems in privacy, we develop an “**Encryption -Secure Communication”** to ease the operation.

A system is required which is being capable of elimination all the problems and become useful to users and thus the new system is derived. Here, User can set the byte of key manually.

**Benefits**

1. Security is enhanced in well manner.
2. Users set the byte key manually.

**Security Requirements**

Web application will be available via network access, it is difficult. If not possible, to limit the population of the end-user who may access the applications? In order to product sensitive connect and provide secure mode be implemented throughout the infrastructure that the supports web application and within the application itself.

Web Application have become heavy integrated with critical corporate and database.

E-commerce application extracts and then store sensitive customer information.

**Design Requirements**

To create project, add base masters and masters to the project, assign behaviors to the master, create and assign behavior sets, and then apply, test and validate those behaviors. It also shows how to create and build a stencil to hold the shapes.

**Quality and Reliability Requirements**

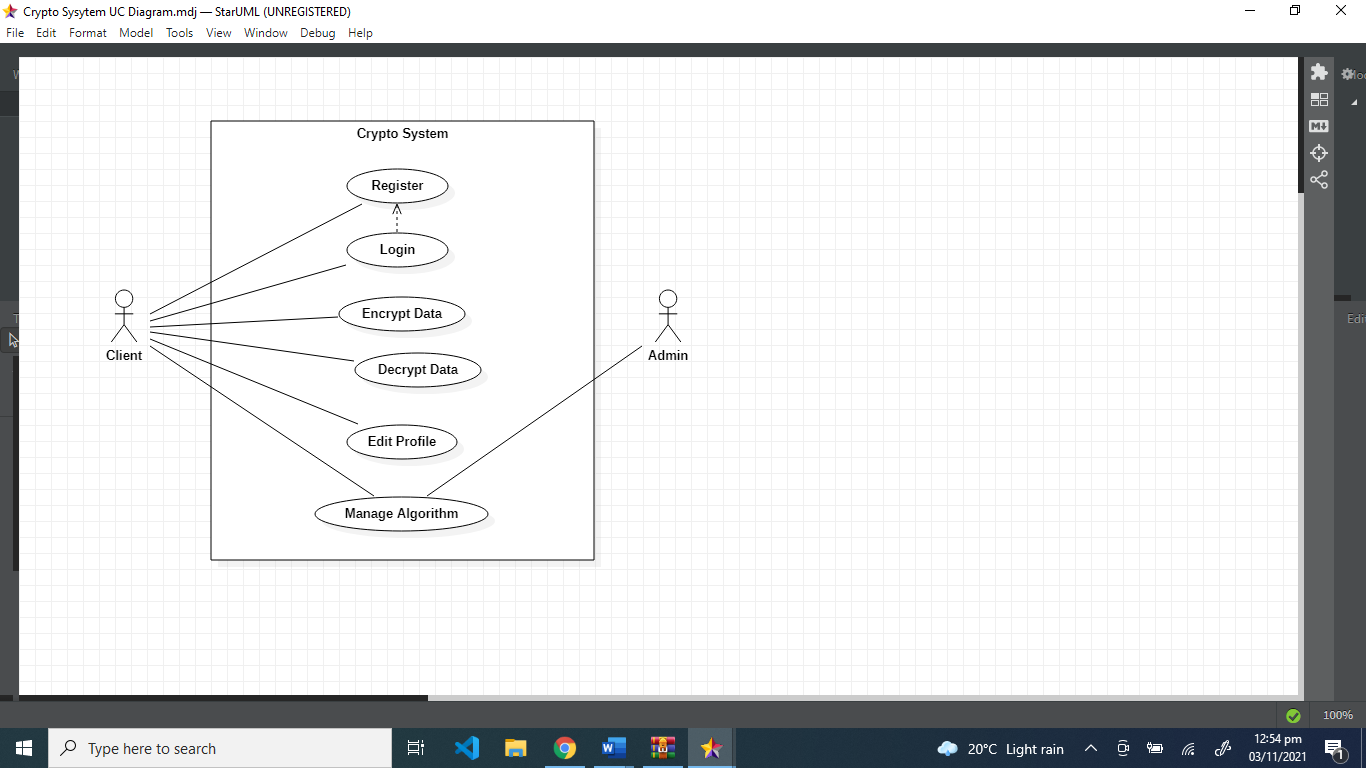
A software component that is developed for reuse would be correct and contain no defects. Formal verification is not carried out routinely, and defects can add to occur. However, with each reuse, defects are found eliminated, and a component qualify improve as a result. Over time the components virtually defect free.

Software reliability is defined in statical term as” the probability of faultier-free operation of a computer program in a specified environment for specified tine”. The software quality and reliability, failure is nonconformance to software requirements. Failure can be only anything or catastrophic. One failure can be corrected within seconds while another requirements week even mouths to correct. Complicating the issue even further, the correction of the one failure may in fact result in the introduction of the errors that ultimately result in other failure.

Web Correct link processing

Application Reliability Error recovery

Quality Input validation and recovery



**Use Case Diagram:**

**Login**

**Brief Case:**

The System prompts the user for a username and password or register new account. The user enters his/her username and password. The system validates the entered username and password, making sure that the entered username is a valid username in the System, and that the required password is entered for the entered username. The user is signed in and returned to the home page as a Logged In User and the use case ends

**Sign Up**

**Brief Case:**

The system prompts the user to select a buyer or a seller account. After selecting the type of account, the System request the user for a username and password or register new account. The user selects registration option. The System prompts user for registration information, Username, password, etc*.* The user enters in their information. System verifies information and creates account and the use case ends.

**Encrypt**

**Brief Case:**

The selected data of client will be encrypted using the specified algorithm.

**Decrypt**

**Briefcase:**

The selected data of client will be decrypted using the specified algorithm.

**Manage algorithms**

**Brief Case:**

The client will select the algorithm of his choice and will select the file to encrypt or decrypt, the client can select some specific algorithms which are free to use however if some algorithm is paid the client will have to pay the required amount.

**Edit profile**

**Brief Case:**

The client will be able to edit and manage his profile and select the settings of his choice also the admin will have the access to these settings.

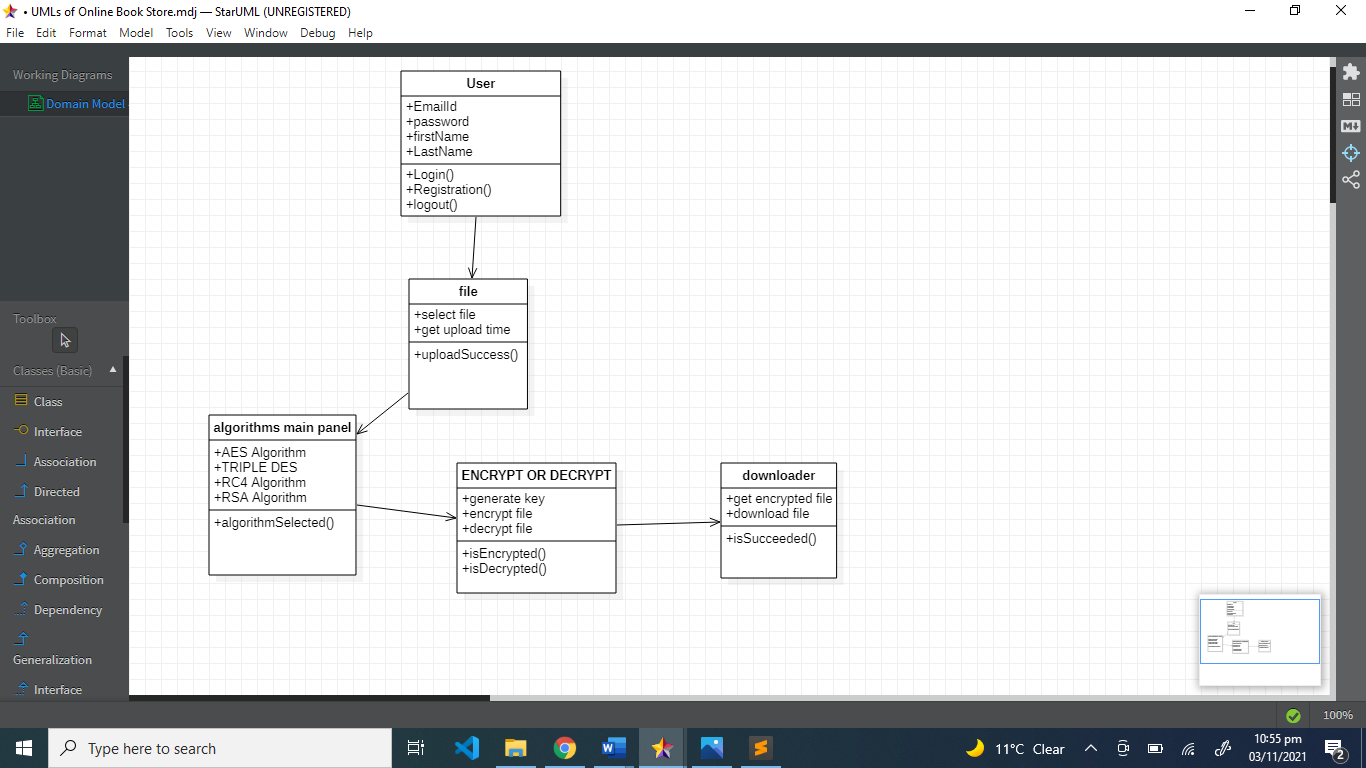
**FULLY DRESSED USE CASES:**

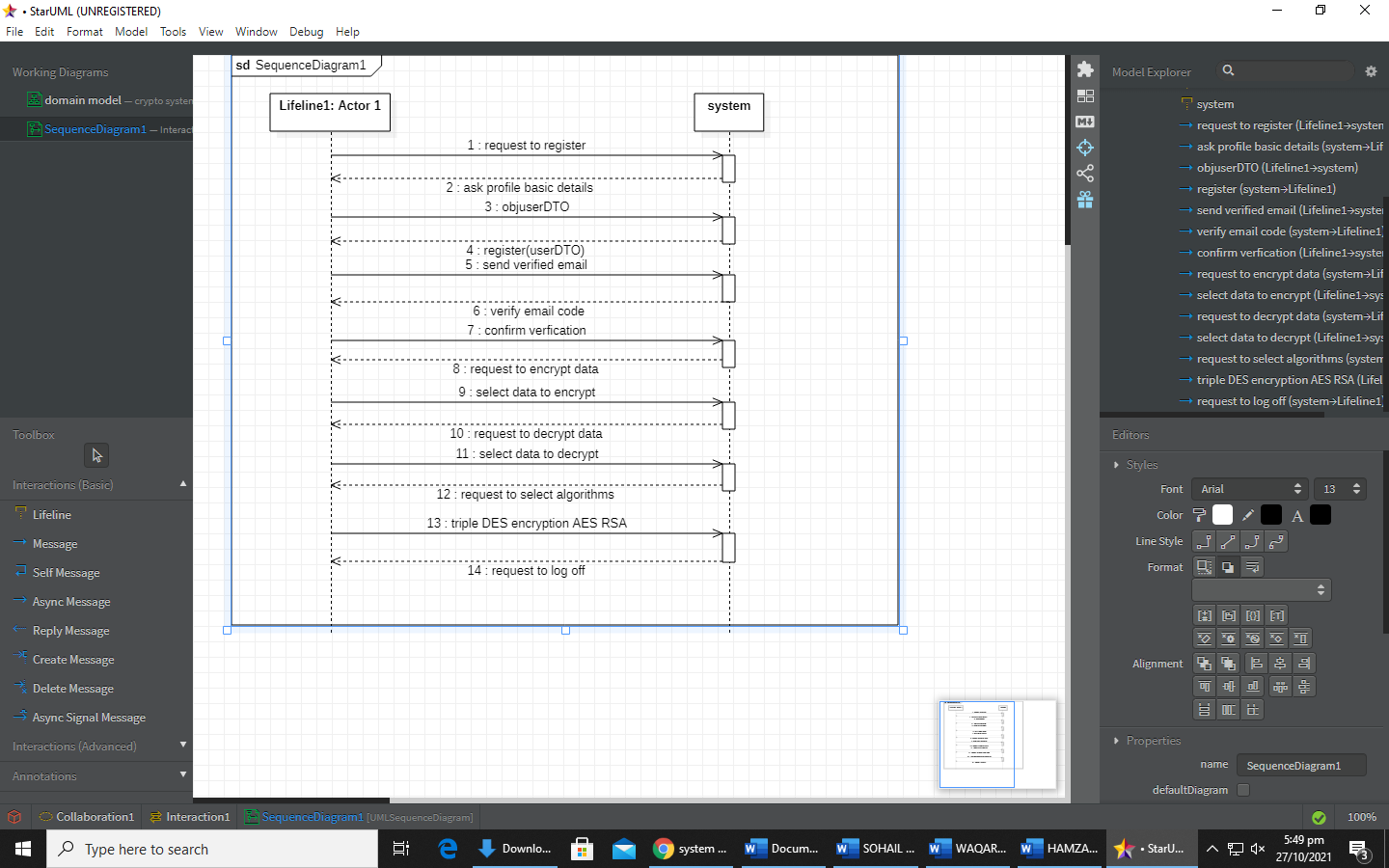
|  |  |
| --- | --- |
| USE CASES | COMMENTS |
| USE CASE NAME | LOGIN/REGISTER |
| SCOPE | CRYPTOSYSTEM |
| LEVEL | USER GOAL |
| PRIMARY ACTOR | CLIENT |
| STAKE HOLDER AND INTEREST | CLIENT: CLIENT WANT TO REGISTER HIMSELF  ADMIN: CONTROLS THE CLIENTS LOGIN INFO |
| PRECONDITIONS | THE CLIENT MUST REGISTER AND LOG IN |
| SUCCESS GURANTEE | THE CLIENTS LOGIN INFO WILL BE SECURED SUCESSUFULLY |
| MAIN SUCCESS SCENARIO | THE CLIENT WILL REGISTER HIMSELF IN ORDER TO MOVE ON |
| EXTENSIONS | IF THE LOGIN DETAILS IN NOT APPROPRIATE THAN LOGIN WILL BE FAILED |
| SPECIAL REQUIRMENTS | USER FRIENDLY, RELIABILITY, USABILITY |

|  |  |
| --- | --- |
| USE CASES | COMMENTS |
| USE CASE NAME | ENCRYPT/DECRYPT DATA |
| SCOPE | CRYPTOSYSTEM |
| LEVEL | USER GOAL |
| PRIMARY ACTOR | CLIENT |
| STAKE HOLDER AND INTEREST | CLIENT: CLIENT WANT TO SECURE DATA  ADMIN: CONTROLS THE CLIENTS DATA |
| PRECONDITIONS | THE CLIENT MUST BE REGISTER AND LOGGED IN |
| SUCCESS GURANTEE | THE CLIENTS DATA WILL BE SECURED SUCESSUFULLY |
| MAIN SUCCESS SCENARIO | THE CLIENT WILL PROVIDE THE DETAILS OF DATA TO BE ENCRYPTED |
| EXTENSIONS | IF THE LOGIN DETAILS IN NOT APPROPRIATE THAN LOGIN WILL BE FAILED |
| SPECIAL REQUIRMENTS | USER FRIENDLY, RELIABILITY, USABILITY |

|  |  |
| --- | --- |
| USE CASES | COMMENTS |
| USE CASE NAME | EDIT PROFILE/MANAGE ALGORITHMS |
| SCOPE | CRYPTOSYSTEM |
| LEVEL | USER GOAL |
| PRIMARY ACTOR | ADMIN/CLIENT |
| STAKE HOLDER AND INTEREST | CLIENT: CLIENT WANT TO EDIT HIS PROFILE  ADMIN: MANAGES ALGORITHMS |
| PRECONDITIONS | THE CLIENT MUST REGISTERED AND LOGGED IN TO EDIT HIS PROFILE  ADMIN: CAN MANAGE ALGORITHMS |
| SUCCESS GURANTEE | ADMIN CAN SELECT APPROPRIATE ALGORITHMS ACCORDING TO WILL OF CLIENT |
| MAIN SUCCESS SCENARIO | THE CLIENT WILL BE ABLE TO MANAGE HIS PROFILE AND THE ADMIN CAN MANAGE THE ALGORITHMS |
| EXTENSIONS | IF THE CLIENT IS NOT LOGGED IN HE CANNOT EDIT HIS PROFILE |
| SPECIAL REQUIRMENTS | USER FRIENDLY, RELIABILITY, USABILITY |

**Domain Model:**

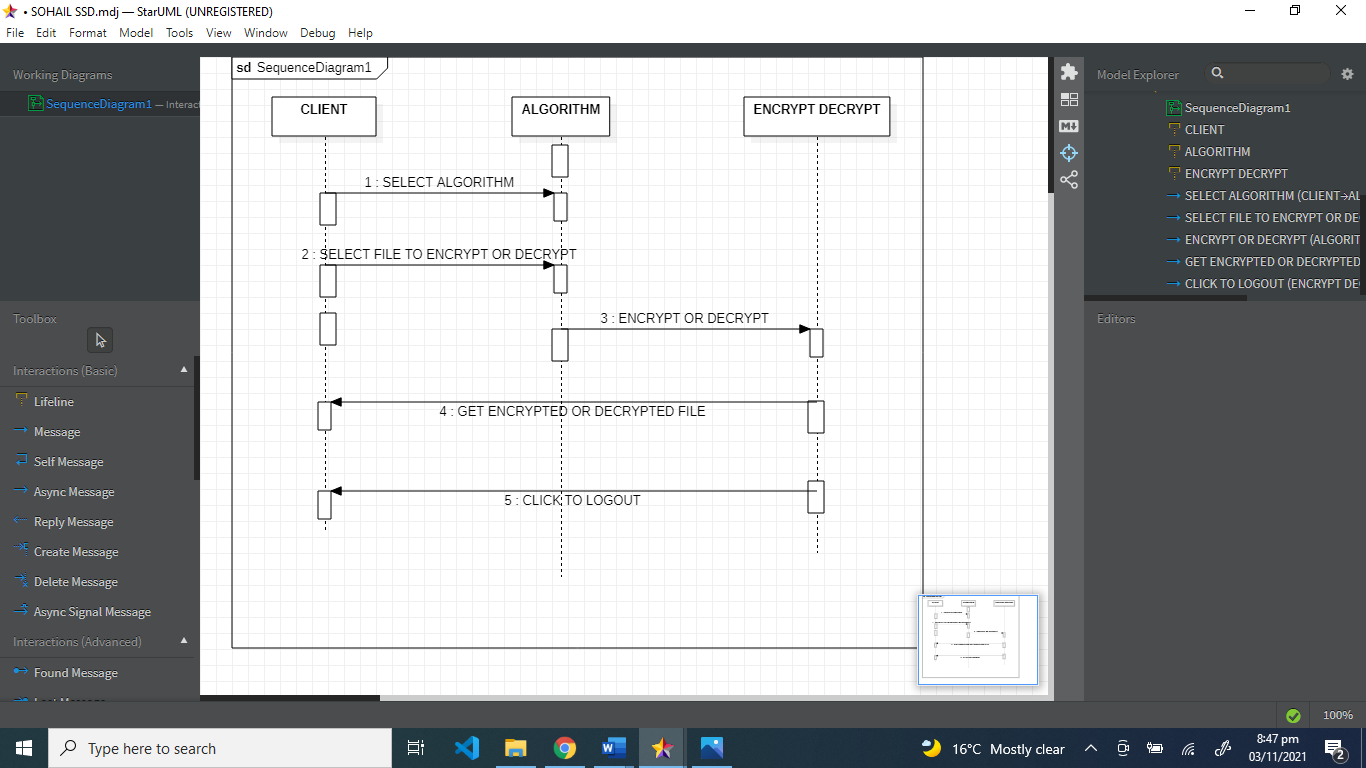


System Sequence Diagram:

A picture containing text, indoor, monitor, electronics

Description automatically generated

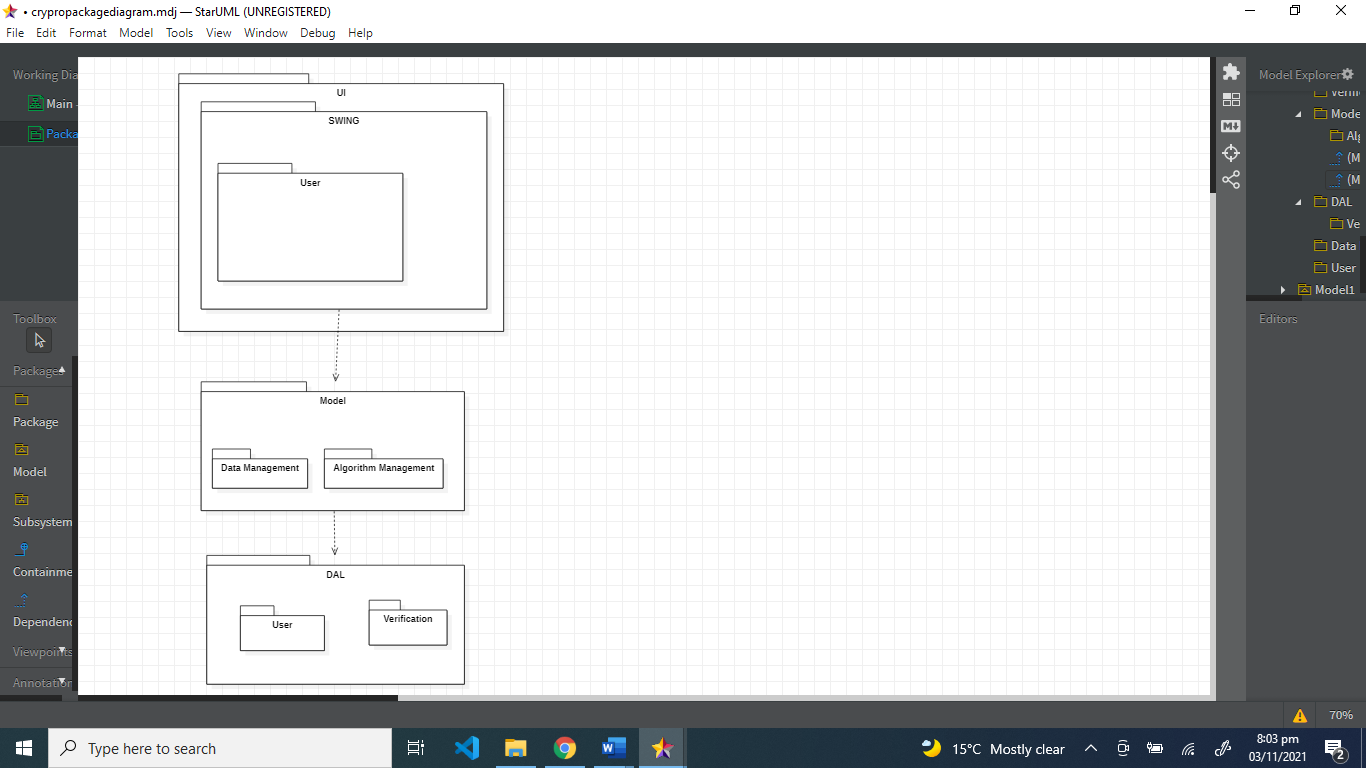
SSD DIAGRAM:



**Operation Contracts:**

|  |  |
| --- | --- |
| **Operation** | Edit profile, Manage algorithms |
| **Cross References** | Use Case: Manage algorithms and edit or set profile |
| **Preconditions:** | The client or admin can edit profile and can select or manage algorithms. |
| **Postconditions:** | The admin will validate the details of client and can manage his profile and the client can manage algorithms according to the data he/she wants to encrypt,decrypt. |

Package Diagram:



SYSTEM DESIGN

INPUT DESIGN

Input design is the process of converting user-originated inputs to a computer-based format. Input design is one of the most expensive phases of the operation of computerized system and is often the major problem of a system.

In the project, the input design is made in various window forms with various methods. This project consist of Encryption is the conversion of data into a form, called a cipher text, that cannot be easily understood by unauthorized people. Decryption is the process of converting encrypted data back into its original form, so it can be understood.

OUTPUT DESIGN

Output design generally refers to the results and information that are generated by the system for many end-users; output is the main reason for developing the system and the basis on which they evaluate the usefulness of the application. In any system, the output design determines the input to be given to the application.

**INTERFACE DESIGN**

The ODBC (Open Database Connectivity) interface is a pure .NET to execute SQl statement. The ODBC provides a set classes and interfaces that can be used by developers to write database applications. Basic ODBC interactions in its simplest form, can be broken down into four steps:

1. Open a connection to the database.

2. Execute a SQL statement

3. Process the result

4. Close the connection to the database

**Screenshots of project:**

