Synopsis

1. INTRODUCTION

1.1 Title of the Project

Crypto - Corner

1.2 Abstract

Business transactions involves large amount of documents, many of which are confidential and contains sensitive information. So, we need a secured way to store and share these information. We use concepts of cryptography to make the process more secure. Cryptography is a way in which we can protect the digital data, it is associated with converting the ordinary plain text into unreadable text and vice versa. It is considered more secure than physical documents as there is no risk of loss during transit or at it's rest. They are immediately available and more secure.

1.3 Objective of the Project

- To store and share the data in a more secure manner.
- To easily store, maintain and access the files.
- With the use of this project, clients have access to their files, anytime anywhere.
- As it is a burden to store the data physically, storing the data virtually will be more easy, effective and cost efficient.

1.4 Project Category

Web Based Relational Database Management System.

1.5 Language(s) to be used

1.5.1 Front End

HTML : HTML stands for Hyper Text Markup Language.HTML is the standard markup language for creating webpages. HTML describes the structure of a webpage. It consists of series of elements which tells the browser how to display the content.

CSS : CSS stands for Cascading Style Sheet.CSS is used to format the layout of webpages. They can be used to define the text styles and how HTML elements are to be displayed on the screen.

JavaScript: JavaScript is a programming language commonly used in web-development and is a client-side scripting language. It is developed to add dynamic and interactive elements to the website.

1.5.2 Back End

PHP: PHP stands for Hypertext Pre-processor that earlier stood for Personal Home Pages. It is used to develop Static websites or Dynamic Web applications. It is a server-side scripting language.

1.5.3 Database

MySQL: It's an open source, Relational Database Management System. It is based on Structured Query Language (SQL), which is used for adding, removing and modifying information in database.

1.6 Structure of the Proposed Project

- We are developing a website for Chartered Accountant (CA) firm, so that they could manage all the client's transaction details securely.
- CA is the one who performs major role, as he has all the controls on the particular client.
- What our website will do is when a client approaches CA, CA will appoint one of the Employee who works under him.
- Then, the CA i. e. the admin will register user and send their login credentials to that particular user.
- Once after a client is assigned with any particular employee, the client will tell all his requirements to the employee that is there will be a direct communication between client and appointed employee.
- The employee manages all the work assigned by the client i . e. his work transactions.
- The users will encrypt and upload the files and decrypt and download the files.
- After the audited files are ready, the employee will upload those files to admin for the verification.
- The admin will go through the report file and then will approve if the file is correct and will discard if there are any mistakes in the file.

- If the files audited by the employee is approved by the admin, then it is displayed to the client.
- If the audited file is discarded by the admin, then it will not be displayed to the client instead the employee has to work on the files again till the admin approves it.
- The main key factor which we use in this project is the concepts of Cryptography.
- The client can decrypt and view his files. The downloaded files will be saved in his device itself. If there are any other changes to be made, the client can request the employee to make the corrections through the chat box option provided in the website.
- All the users will be having their own to-do-list where they can keep any remainders for themselves or can list any task for themselves.
- If in case the client doesn't want to further work with CA, he can request for deletion of his account.
- The client can also give queries to the admin regarding the employee's work that is if there is any complaints or feedback to be given then the client can post them to the admin and these queries will not be displayed to the employee. The admin can also give replies to those queries.
- There is also an option for payment where the client can pay for all his works done and the admin can view those payment details.

1.7 Module Description

- **Authentication** – Employee and client profile both will be created by Admin by setting certain username and password
- **Manage users** – Admin (CA) can manage the employees who work under them and also the clients.
- **Encrypt/Decrypt files** – Here users can encrypt and upload the file and also can decrypt and download the files.
- **File category** – Employees offer services like Company Audit, Direct and Service Tax, International Tax, GST compliance etc.

- **File verification** – The admin will verify the files uploaded by the employee and approve if it is correct and discard if there is any mistake in the audited file.
- **View client** – CA and the assigned employee both can view the actions of client.
- **Settings** – Profile settings changes can be done.
- **Query** – Client can give the feedback to the employee based on his performance.
- **Chat Box** – Users can communicate with each other by using the Chat box option through the site
- **To-do-List** – Users and can list the tasks for themselves.
- **Payment** – Clients will do the payment and admin can view the payment details.

1.8 Any Other Information

1.8.1 Hardware Requirements

- Processor: Intel dual core or above
- Processor Speed: Minimum 2 GHz
- RAM: 2 GB or above
- Hard Disk: Minimum 40 GB

1.8.2 Software Requirements

- Operating System: Windows 7 or above
- Text editor: Sublime
- server: Apache
- Browsers: Chrome, Firefox or any other browsing application

1.9 Future Scope of the Project

- This website can also be used by the lawyers to secure the proofs related to their cases.

Software Requirement Specification (SRS)

2. SOFTWARE REQUIREMENTS SPECIFICATION(SRS)

2.1 Introduction

The introduction of the Software Requirements Specification (SRS) provides an overview of the entire system. It is a description of system to be developed. SRS document lists sufficient and necessary requirements for the development of the system. The detailed requirements of the Crypto-Corner are provided in this document. The aim of this document is to gather and analyze and give an in-depth insight of the complete Crypto-Corner.

2.1.1 Purpose

Clear requirements help development teams to create right product. Defining and describing the functions and specifications of the Crypto-Corner is the primary goal of this Software Requirements Specification (SRS). This Software Requirements Specification (SRS) illustrates, in clear terms, the system's primary uses and required functionality as specified by our client.

2.1.2 Scope

The system being developed is called Crypto-Corner, a website for Chartered Accountant (CA) firm. It is being developed to manage/maintain and keep to track of client's transactions via the internet. The system is designed to store and share the clients document in a secured manner.

2.1.3 Definitions, Acronyms, Abbreviations

SRS	Software Requirement Specifications Software Requirement Specifications (SRS) is a detailed description of a software system to be developed.
CFD	Context Flow Diagram Context Flow Diagram (CFD) is a diagram that defines boundaries

	between the system. It is called as the initial level or fundamental system model or a 0-level Data Flow Diagram.
DFD	Data Flow Diagram Data Flow Diagram (DFD) is a way of representing a flow of data through a process or a system.
CA	Chartered Accountant Chartered Accountant (CA) is a designation given to an accounting professional. The profession involves management of finances such as budgeting, auditing, business strategy and taxation
HTTPS	Hypertext Transfer Protocol Secure Hypertext Transfer Protocol Secure (HTTPS) is an extension of Hypertext Transfer Protocol (HTTP). It is used for secure communication over a computer network, and is widely used on the Internet

2.1.4 Overview

The remaining sections of this document provide a general description, including characteristics of the users of this system, the product's hardware, the functional and data requirements of the product. It also gives the constraints and assumptions made while designing the Crypto-Corner. It also gives the user viewpoint of product. it also gives the specific requirements of the product and also discusses the external interface requirements and gives detailed description of functional requirements.

2.2 Overall Description

This section will give a detailed information of the whole system. The system will be explained about how it interacts with other systems and also the basic functionality of it. It will also describe the user characteristics and what functionality is available for each.

2.2.1 Product Perspective

- The application will reduce the security risk.
- The application will improve the standard of data management and help the admin to maintain the data efficiently.
- The application enables easy storing and sharing of data.

2.2.2 Product Functions

- The project Crypto-Corner is developed to help Chartered Accountant (CA) as well as his clients to manage the data efficiently.
- The application will reduce the security risk by restricting the unauthorized users as Admin will have all controls of the system.
- The main key factor which we use in this project is the concepts of Cryptography.
- The users will encrypt and upload the files and decrypt and download the files.
- After the audited files are ready, the employee will upload those files to admin for the verification.
- The admin will go through the report file and then will approve if the file is correct and will discard if there are any mistakes in the file.
- The users can create several folders in their profile where they can upload the files related to it's category.
- All the users will be having their own to-do-list where they can keep any remainders for themselves or can list any task for themselves.
- Users can communicate with each other through the chat box feature provided in the website.
- Client can request for the deletion of his account.
- The client can also give queries to the admin based on employee's performance.
- There is also an option for payment where the client can pay for all his works done and the admin can view those payment details.

2.2.3 User Characteristics

2.2.3.1 Login as Admin

- Admin is the superior user of the system
- Admin will create profiles for both employee and client
- Admin (CA) can manage the employees who works under him.
- Admin will verify the file uploaded by employee, if the report is proper then it will be displayed to the client if there are any mistakes, admin can discard that file and inform the employee to work on that file once again.
- CA and the assigned employee both can view the actions of client.
- Only admin can add and remove the users, by which admin can restrict the other user accessing the system other than the client associated with them.
- Admin can chat with the client and employee through the chat box.
- Admin will receive the queries given by the client and he can also reply back.
- Admin can view the payments done by the client.

2.2.3.2 Login as Employee

- Employee can modify his profile.
- He has permission to access the files uploaded by the client
- Employee can create several folders in his profile where he can upload the files related to it's category
- After performing certain operations on client's data, employee will encrypt and upload the reports to admin. The uploaded reports will be first received by the Admin, Admin will go through the reports and if it is correct then it will be displayed to client or else the employee has to work on those files until the admin approves it.
- Employee can chat with the client and admin through the chat box.
- Employee can have his own to-do-list where he can keep any remainders or can list any task for himself.

2.2.3.3 Login as Client

- Client can browse the features available in the system.
- Client can modify his profile.
- He can encrypt and upload the documents of his daily transactions (purchase, sales).
- He can decrypt and download the reports uploaded by the employee which is already verified by the admin.
- Client can create several folders in his profile where he can upload the files related to it's category.
- Client has an option to chat with the employee and admin.
- Queries can be given to the admin based on employee's performance.
- Client can have his own to-do-list where he can keep any remainders or can list any task for himself.
- Client can make the payments for all his works done.
- Client can request for the deletion of his account.

2.2.4 General Constraints

- Only after admin approving the report uploaded by employee, it will be displayed to the client
- The encryption will work on several file types except video and audio files.

2.2.5 Assumptions and Dependencies

- It is assumed that the client will upload the documents regarding his transactions only.
- It is assumed that the employee will upload the particular report files to the particular client as he will be assigned to many clients.
- It is also assumed that the clients will give the queries based on the employee's performance.
- The audited files which will be displayed to the client depends on the admin's verification.

2.3Specific Requirements

2.3.1 External Interface Requirements

2.3.1.1 User Interfaces

- The system provides a user-friendly GUI to the users.
- HTML for developing the user layout for the system
- PHP for creating all the validation and client-side scripting functionality.
- CSS for designing web page of the system.
- JavaScript for designing dynamic page of the system

2.3.1.2 Hardware Interfaces

Not applicable

2.3.1.3 Software Interfaces

Not applicable

2.3.1.4 Communication Interfaces

The Crypto-Corner shall use the HTTPS protocol for communication over the internet. This project supports all types of the web browsers.

2.4 Functional Requirements

2.4.1 Registration

Functions **:** The admin can register clients and employees. The admin will provide login credentials for the clients and employees after the registration.

Employee Input **:** emp_name, phone_number, email, address, dob, image, password, roll, creation_date

Client Input **:** client_name, phone_number, email, address, dob, company_name, gst_num,

company_address, roll, image, password, creation_date.

Process **:** The system will receive and save the information of the user.

Output **:** The users will be registered.

2.4.2 Authentication

Functions **:** The admin can view the details about registered users. The admin is the superior user of the system. The whole system is controlled by the admin himself. The admin will provide login credentials for the registered client and employee where they enter the username and password to access the website**.**

Input **:** username, password

Process **:** The system checks for the correctness of the username and password.

Output **:** The users can access the profile.

2.4.3 Encrypt / Decrypt Files

Functions **:** AES encryption method is used to encrypt and decrypt the files. The client will encrypt and upload his daily transaction details in the system which is accessible by employee and admin. The employee will decrypt and download those files and then he'll perform certain operations and will again encrypt and upload the report

of the transactions of the particular user to the admin for the verification. The report files will be displayed to the Client only if the admin approves it.

Input **:** client_id, client_name, emp_id, emp_name, file_name, filesize, file, file_category, creation_date.

Process **:** The user can choose the file which they want to upload and download

Output **:** The files will be uploaded / downloaded.

2.4.4 Verify File

Functions **:** The admin will go through the reports uploaded by the employee and if the admin approves the reports then it will be displayed to the client, if the admin discards then employee has to work on the file once again.

Input **:** emp_id, emp_name, file_name, filesize, file, status, creation_date.

Process **:** The admin will check the report and will either approve or discard the reports that has to be displayed to the client.

Output **:** Client will get to see the verified file.

2.4.5 Settings

Functions **:** User of the system can change their personnel details such as setting up a new password

Input **:** old_password,new_password,new_profilepic, add_bio,

Process **:** The system will store all the changes made by the user

Output **:** The details will be updated

2.4.6 Manage Employee

Functions **:** Admin can add /remove employees

Input **:** emp_id, emp_name, phone_number, email, address, dob, image, password, roll, creation_date

Process : The admin will register/remove the employee.

Output **:** The registered employees can login to the system

2.4.7 Manage Clients

Functions **:** Admin can add/ remove clients

Input **:** client_id, client_name, phone_number, email, address, dob, company_name, gst_num, company_address, roll, image, password, creation_date.

Process : The admin will register/remove the clients.

Output **:** The registered clients can login to the system

2.4.8 Query

Functions **:** Client can evaluate the employees performance. Client can give complaints and feedback to the admin based on employee's work. The queries will be displayed to the

each employee working under him. The admin can reply to the query.

Input **:** query_id, q_replyid, subject, client_id, client_name, client_query, admin_id, admin_name, admin_reply, query_date, reply_date.

Process : Admin will receive the feedback.

Output **:** The queries and feedbacks will be displayed to the admin .

2.4.9 Chat Box

Functions **:** users will communicate with each other through the chat box.

Input **:** msg_id, reply_id, admin_id, admin_name, admin_msg, admin_reply, emp_id, emp_name, emp_msg, emp_reply, client_id, client_name, client_msg, client_reply, msg_date, reply_date.

Process : The system will update the new message to the user.

Output **:** The users can view and send the message to each other.

2.4.10 To-do-List

Functions **:** users will list the tasks for themselves.

Input **:** subject, description, date

Process : The system will add the remainder.

Output : The users will get the remainders they can also delete it

2.4.11 Payment

Functions **:** clients will make the payment for their works done and admin can view those payments.

Input **:** transcation_id, client_name, amount,date.

Process : The system will send the payment details to the admin

Output **:** The admin will get to see the payments done by the client

2.5 Performance Requirements

Information transmission should be securely transmitted to server without any changes in information. The interface will be visually simple, time to retrieve information and records will be based on the database queries, which keeps wait time to minimum. Performance also depends on the hardware components of the user. Software shall also support multiple user at a time. The performance of any web application should be as high as possible. The system must provide right tool for the user.

2.6 Design Constraints

- System shall be built using standard web page development tools.
- System must be stored in such a way that allows the users easy access to it.
- Response time for loading the system should take no longer than few minutes

2.7 System Attributes

- **Security:** login will be provided for the users with unique username and password to avoid usage of the system from other users. The password will be in hashed format to provide more security.
- **Maintainability:** during maintenance stage, the SRS can be referred for the validation.
- **Portability:** This application can be installed in any smart phones supporting operating system.
- **Flexibility:** The system keeps on updating the data according to the transactions that takes place.
- **Speed:** The system carries out all the operations with consumptions of very less time.
- **Accuracy:** good validation of user inputs will be done to avoid incorrect storage of records.

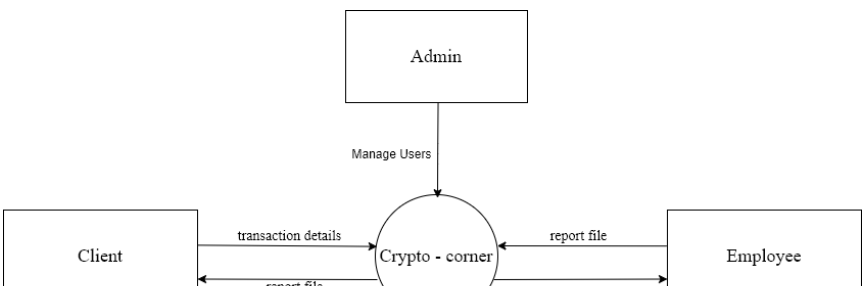
2.8 Other Requirements

2.8.1 Security Requirements

- The server on which the 'Crypto-corner' resides will have its own security to prevent unauthorized write/delete access. There is no restriction on read access.
- Only the admin will have physical access to the machine and the program on it.

2.9 Context Flow Diagram (CFD)

Context Flow Diagram (CFD) is a diagram that defines boundaries between the system. It is called as the fundamental system model or a 0-level Data Flow Diagram.



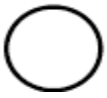
2.10 Data Flow Diagram

2.10.1 Introduction

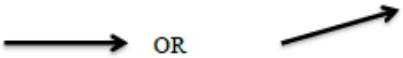
Data-flow diagrams (DFD) are also known as data-flow graphs or bubble charts. A DFD serves the purpose of clarifying system requirements and identifying major transformations. DFD's show the flow of data through a system.

2.10.2 Symbols Used for Constructing DFD's

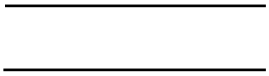
Function symbol : A function is represented using a circle. This symbol is called a process or a bubble and performs some processing of input data.



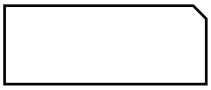
External entity : A square defines a source or destination of system data. External entities represent any entity that supplies or receives information from the system but is not a part of the system.

Data-flow symbol : A directed arc or arrow is used as a data-flow symbol. A data-flow symbol represents the data flow occurring between two processes or between an external entity and a process in the direction of the data flow arrow. 

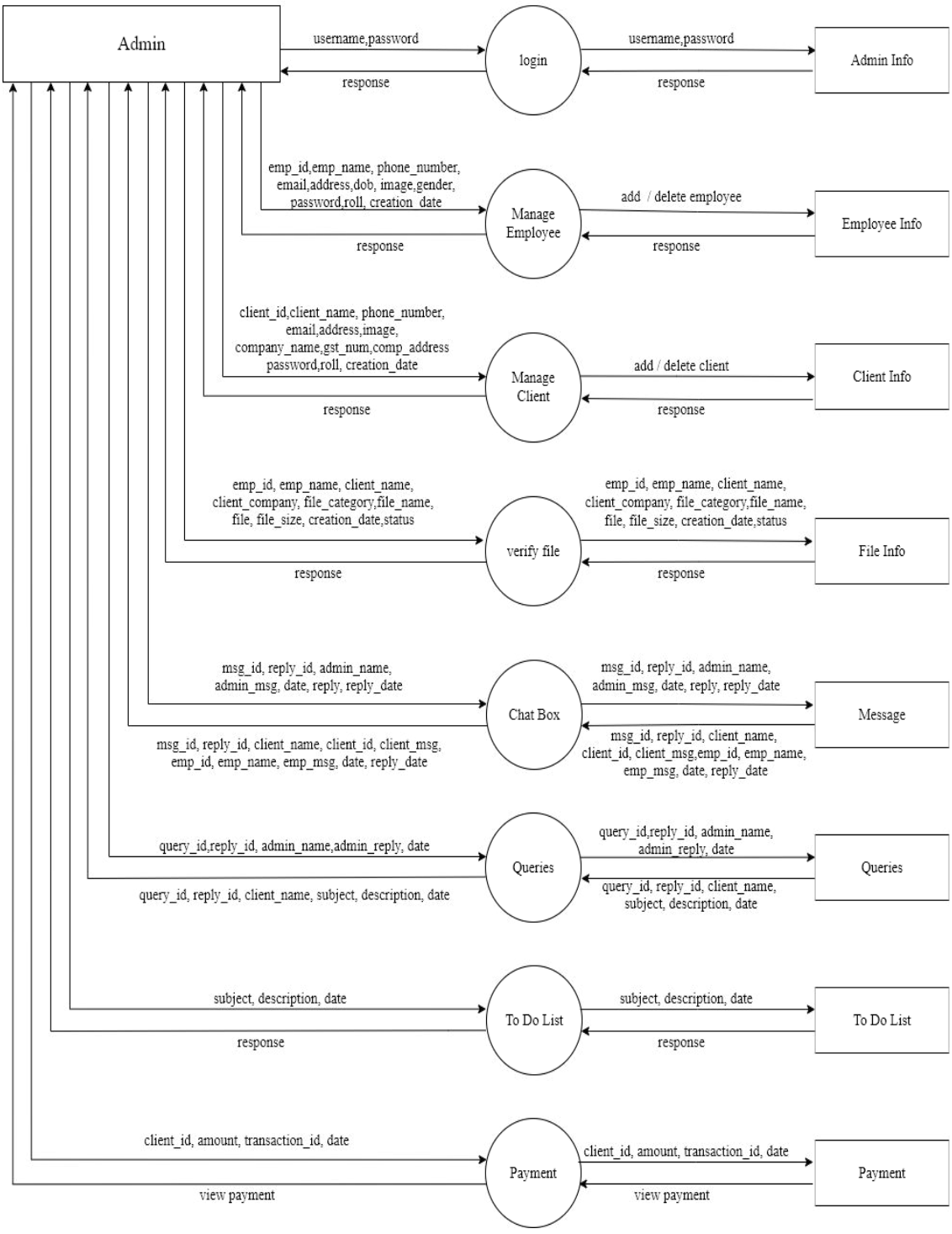
Data-store symbol : A data-store symbol is represented using two parallel lines. A logical file can represent either a data-store symbol, which can represent either a data structure, or a physical file on disk. Each data store is connected to a process by means of a data-flow symbol. The direction of the data-flow arrow shows whether data is being read from or written into a data store.



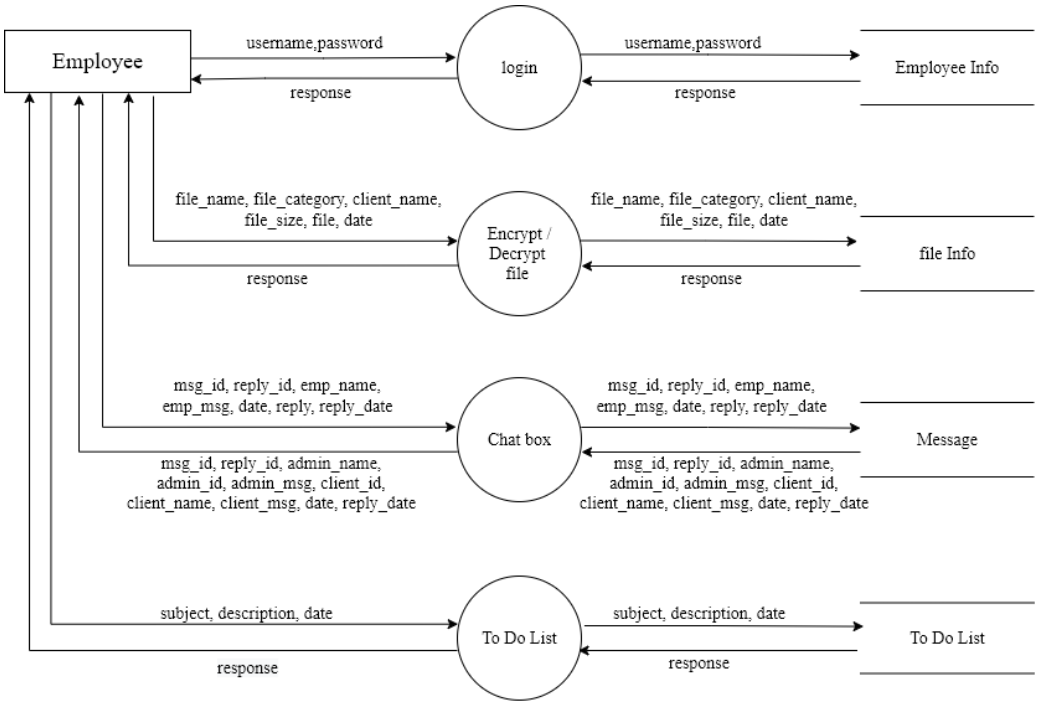
Output symbol : It is used to represent data acquisition and production during human-computer interaction.



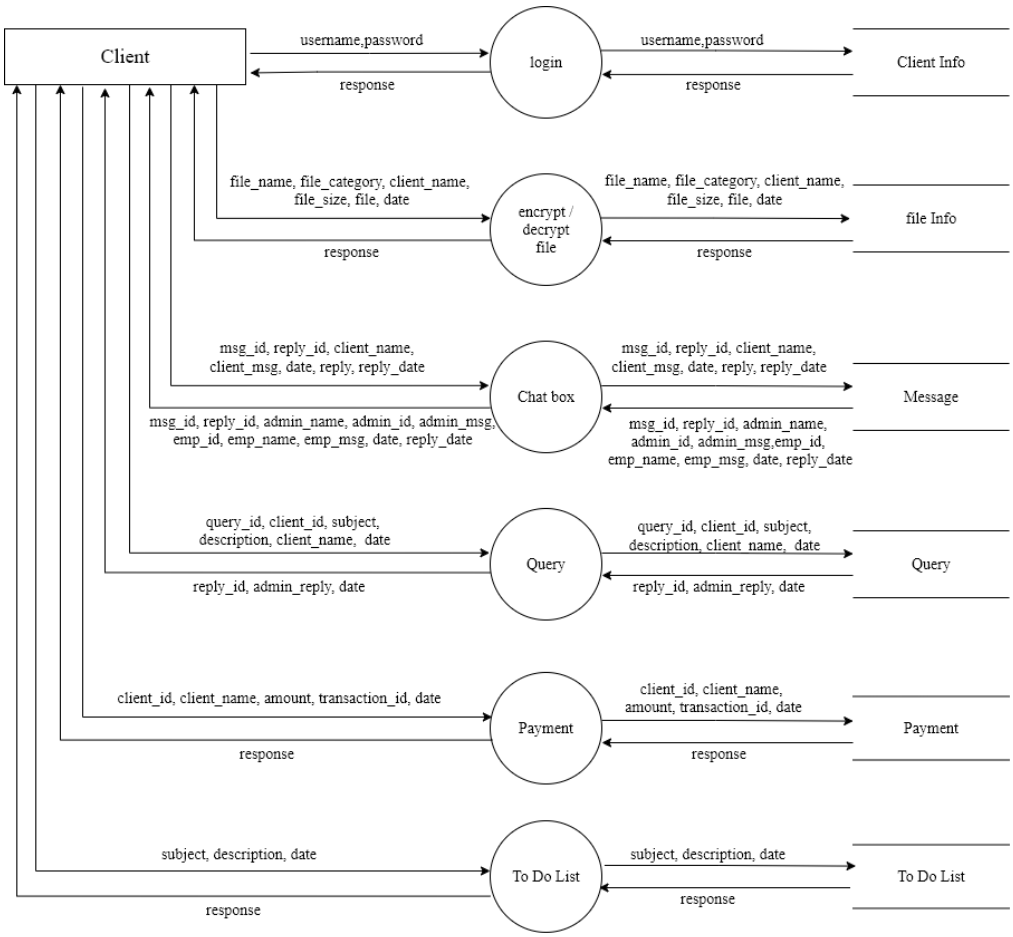
2.10.3 Data Flow Diagram for Admin



2.10.4 Data Flow Diagram for Employee



2.10.5 Data Flow Diagram for Client



System Modelling

3. SYSTEM MODELLING

3.1 Introduction

System Modelling is the process of developing abstract models of a system with model presenting a different view or perspective of that system. It represent aspects of a system and its environment. It is a mean of representing a world view a detailed view of the system using same kind of Graphical Notation. A common type of systems modeling is function modeling, with specific techniques such as the Data Flow Diagram etc. It defines the processes that serve the needs of the view under consideration. System Modeling represent the behaviour of the processes and the assumptions on which the behaviour is based. It also represents all linkages(input/output) that will enable engineer to better understand the view. System design is a process through which requirements are translated into a representation of a software. In detail design we specify how the modules in the system interact with each other. Detailed specification is given by explaining in natural language what the modules is supposed to do.

Detail design specification describes the features of the system. System design are created in order to meet the needs of the user. They are not only intended to solve the existing problems, but they also come up with acceptable solutions to the problems that may arise in the future. The whole process of system development, from blueprint to the actual product, involves considering all the relevant factors and taking the required specifications and creating a useful system based on strong technical, analytical and development skills of the professionals.

3.2 Functional Decomposition.

Functional decomposition breaks down a large, complex process into an array of smaller, simpler units or tasks for the better understanding of the overall process. Functional decomposition diagram contains the whole function or project along with all of the necessary sub-tasks needed to complete it. Functional decomposition is a problem-solving tool used in several contexts.

3.3 Class Diagram

Class diagrams provide a graphic notation for modeling classes and their relationships, thereby describing possible objects. Class diagram is a static diagram. It represents the static view of an application.

Class diagrams are the only diagrams which can be directly mapped with object-oriented languages and thus widely used at the time of construction. They are useful for abstract modelling and for designing actual programs.

A class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

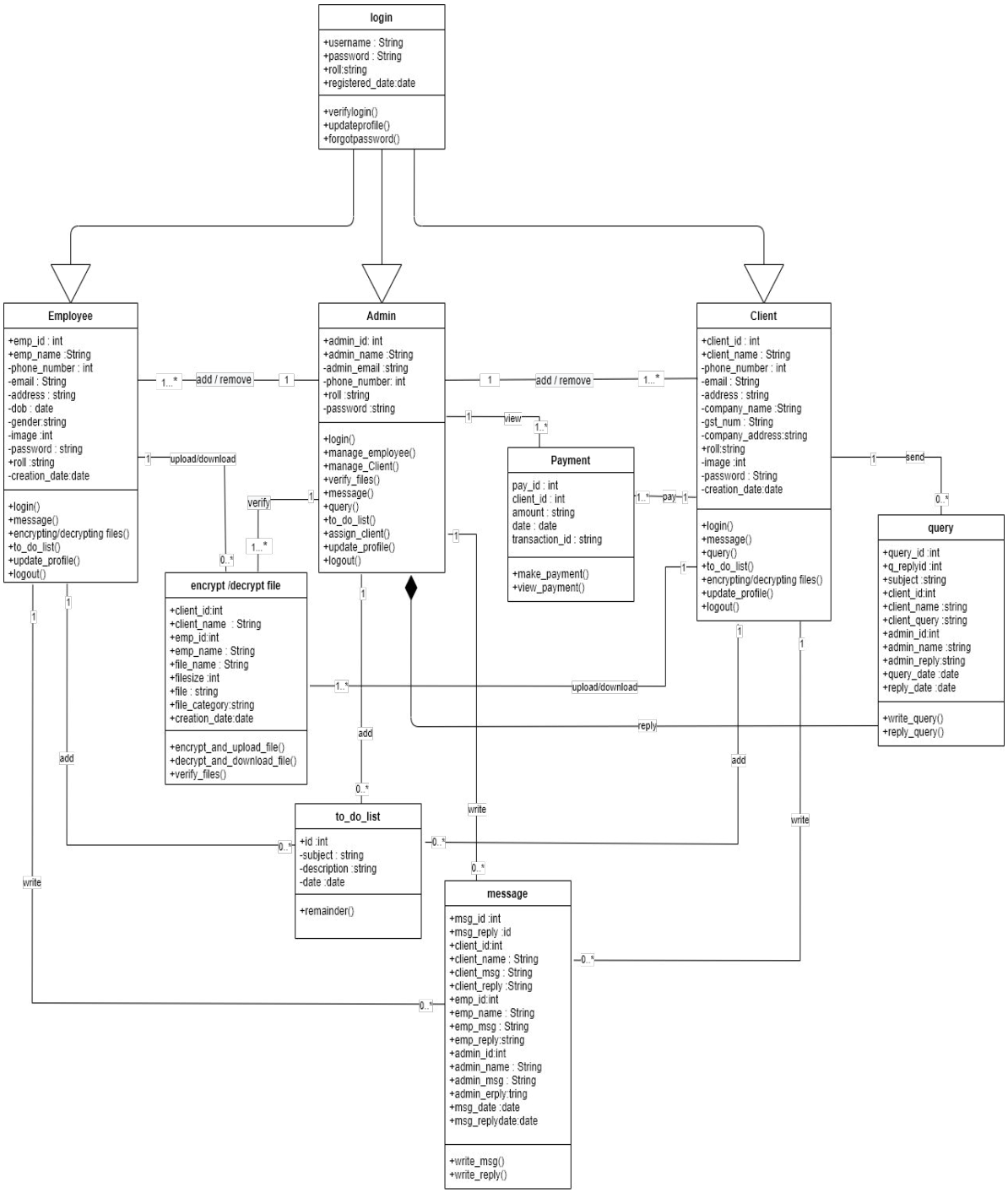
The UML symbol for class is box. Class name is written in boldface, center the name in the box, and capitalize the first letter

There are three types of modifiers that are used to decide the visibility of attributes and operations.

- + is used for public visibility (for everyone)
- # is used for protected visibility (for friend and derived)
- – is used for private visibility (for only me)

The standard class diagram is composed of three sections:

- **Upper section:** Contains the name of the class.
- **Middle section:** Contains the attributes of the class. This is only required when describing a specific instance of a class.
- **Bottom section:** Includes class operations (methods). The operations describe how a class interacts with data.


login
+username : String
+password : String
+roll:string
+registered_date:date
+verifylogin()
+updateprofile()
+forgotpassword()
Employee
+emp_id : int
+emp_name :String
-phone_number : int
-email : String
-address : string
-dob : date
-gender:string
-image :int
-password : string
+roll :string
-creation_date:date
+login()
+message()
+encrypting/decrypting files()
+to_do_list()
+update_profile()
+logout()
Admin
+admin_id: int
+admin_name :String
-admin_email :string
-phone_number: int
+roll :string
-password :string
+login()
+manage_employee()
+manage_Client()
+verify_files()
+message()
+query()
+to_do_list()
+assign_client()
+update_profile()
+logout()
Client
+client_id : int
+client_name : String
-phone_number : int
-email : String
-address : string
-company_name :String
-gst_num : String
-company_address:string
+roll:string
-image :int
-password : String
-creation_date:date
+login()
+message()
+query()
+to_do_list()
+encrypting/decrypting files()
+update_profile()
+logout()
Payment
pay_id : int
client_id : int
amount : string
date : date
transaction_id : string
+make_payment()
+view_payment()
query
+query_id :int
+q_replyid :int
+subject :string
+client_id:int
+client_name :string
+client_query :string
+admin_id:int
+admin_name :string
+admin_reply:string
+query_date :date
+reply_date :date
+write_query()
+reply_query()
encrypt /decrypt file
+client_id:int
+client_name : String
+emp_id:int
+emp_name : String
+file_name : String
+filesize :int
+file : string
+file_category:string
+creation_date:date
+encrypt_and_upload_file()
+decrypt_and_download_file()
+verify_files()
to_do_list
+id :int
-subject : string
-description :string
-date :date
+remainder()
message
+msg_id :int
+msg_reply :id
+client_id:int
+client_name : String
+client_msg : String
+client_reply :String
+emp_id:int
+emp_name : String
+emp_msg : String
+emp_reply:string
+admin_id:int
+admin_name : String
+admin_msg : String
+admin_erply:tring
+msg_date :date
+msg_replydate:date
+write_msg()
+write_reply()
add / remove
upload/download
verify
view
pay
send
reply
add
write

3.4 Use case Diagram

A use case diagram is a graphical depiction of a user's possible interactions with a system. A use case diagram shows various use cases and different types of users the system has. The purpose of a use case diagram in UML is to demonstrate the different ways that a user might interact with a system. A use case diagram is a dynamic or behaviour diagram in UML. Use case diagrams model the functionality of a system using actors and use cases. Use cases are a set of actions, services, and functions that the system needs to perform.

A system involves a set of use cases and a set of actors. Each use case represents a slice of the functionality the system provides. Set of use cases shows the complete functionality of the system at some level of detail. Each actor represents one kind of object for which the system can perform behaviour. The set of actors represents the complete set of objects that the system can serve.

3.4.1 UML notations

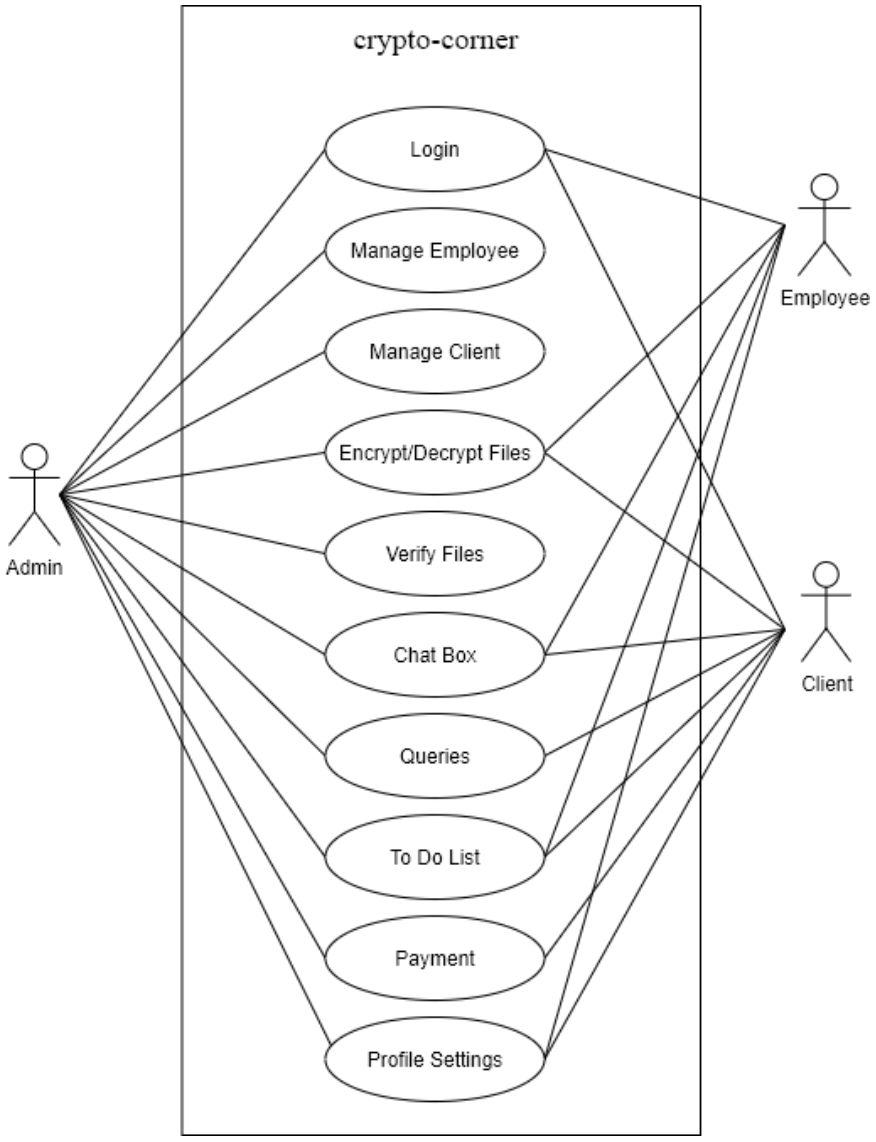
Rectangle contains the use cases for a system with actors listed on the outside.

Name of the system may be written near a side of the rectangle.

Use case represented by using **ellipse**- name should be written within it.

A **stick man** icon denotes an actor.

Solid lines connect use cases with an actor.



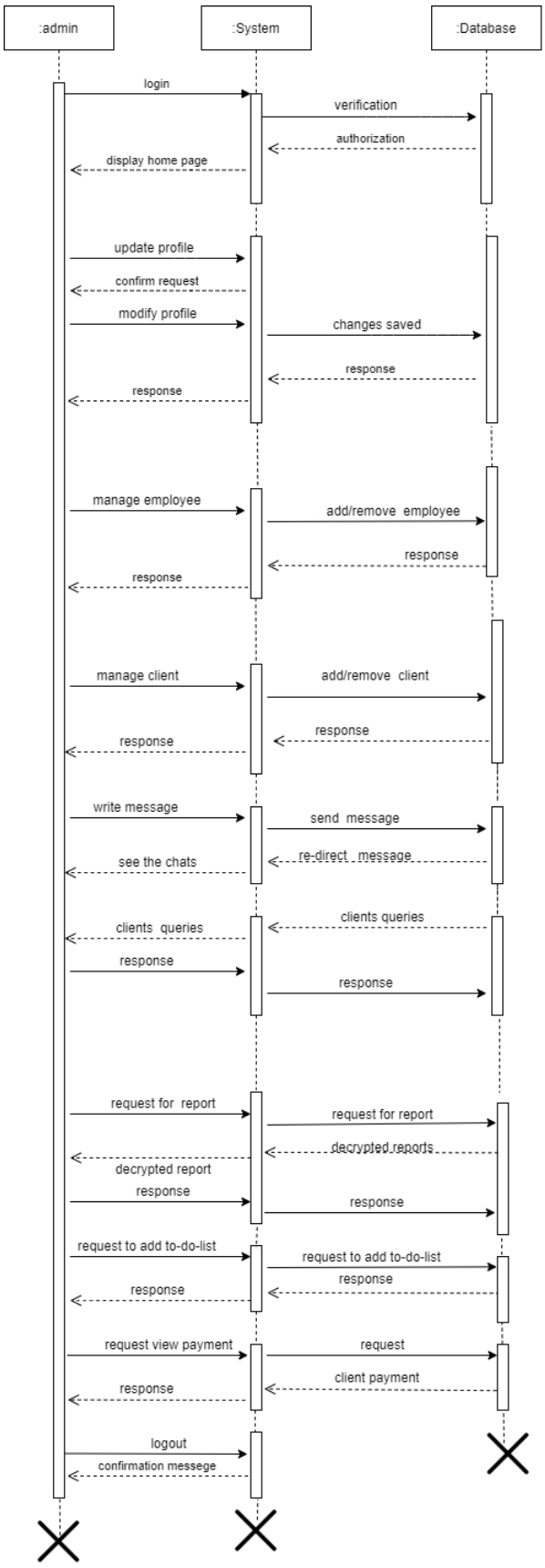
3.5 Sequence diagram

A sequence Diagram shows the participants in an interaction and the sequence of messages among them. Sequence diagram shows the interaction of a system with its actors to perform all or part of a use case.

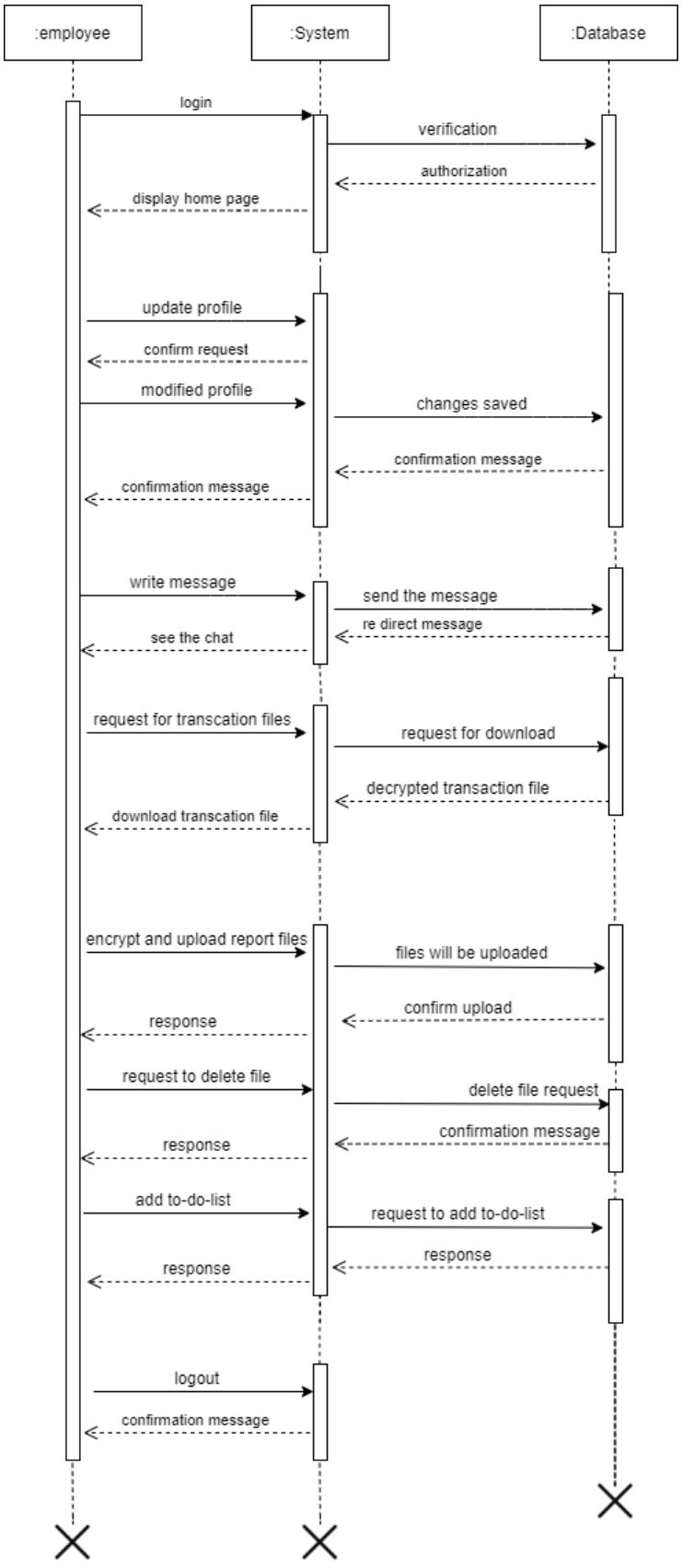
Each use case requires one or more sequence diagrams to describe its behaviour. Each sequence diagram shows a particular behavior sequence of the use case.

UML Notations for sequence diagram : Actor and system is represented by a vertical line called a lifeline. Message transfer is represented by horizontal

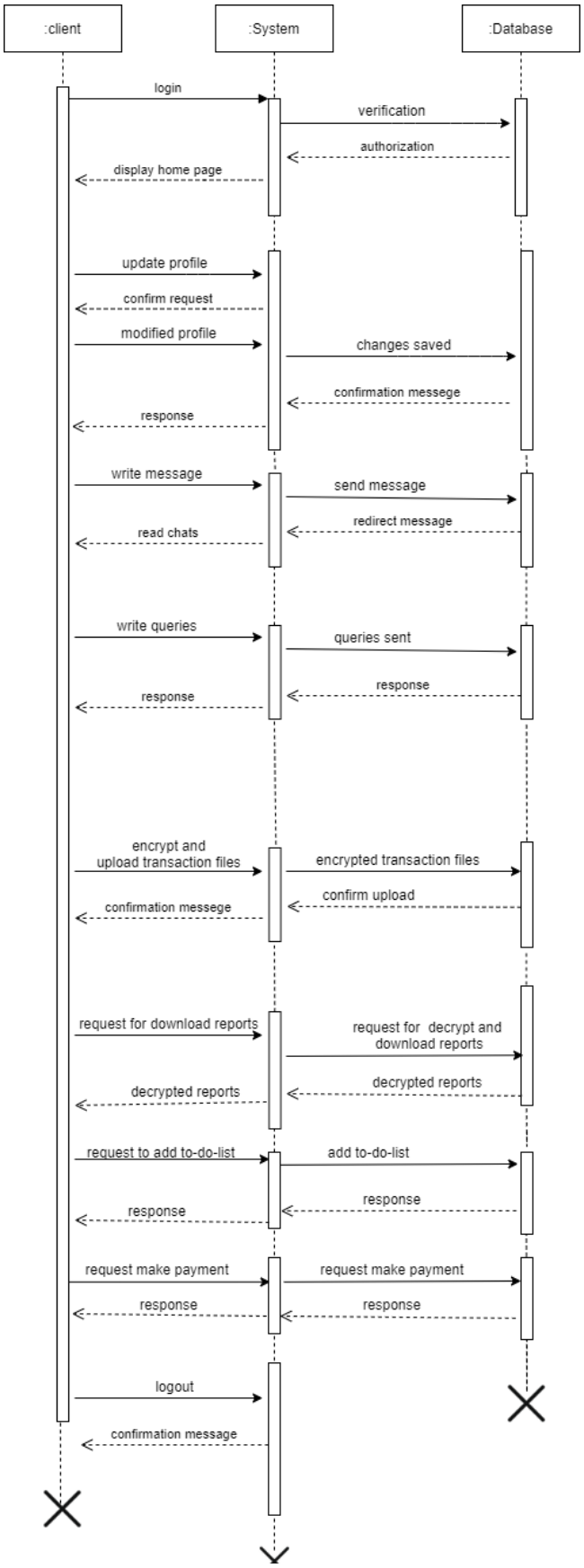
3.5.1 Sequence Diagram for Admin



3.5.2 Sequence Diagram for Employee



3.5.3 Sequence Diagram for Client



Database Modelling

4. DATABASE MODELLING

4.1 ER Diagram

The entity-relationship (E-R) data model was developed to facilitate database design by allowing specification of an enterprise schema that represents the overall logical structure of a database. It is a data-oriented model of a system.

It has three main components: **data entities,** their **relationships,** and their **associated attributes**.

Entity : It is the most elementary process of an organization about which data is to be maintained. Every entity has a unique identity, which distinguishes it from other entities. An entity type is the description of all entities to which a common definition and common relationships and attributes apply.
It is represented by a rectangular box with the name of the entity written inside.

Relationship : Entities are connected to each other by relationships. It indicates how two entities are associated.
A diamond notation with the name of the relationship is represented as written inside.

The number of entity types that participate in the relationship is called the degree of the relationship.

The three most common relationships in the E-R diagram are unary, i.e., Degree one, binary, i.e., Degree two, and ternary, i.e., Degree three.

Attribute : Each entity type has a set of attributes associated with it. An attribute is a property or characteristic of an entity that is of interest to the organization.
It is represented by an oval-shaped box with the name of the attribute written inside it.

4.1.1 Types of Attributes

Simple attribute : Simple attributes can't be divided any further.

Composite attribute : composite attribute can be divided into sub-parts

Single valued attribute : Single-valued attribute is an attribute that can have only a single value. .

Multivalued attribute : Multivalued attributes can have more than one values.

Derived attribute : This type of attribute does not include in the physical database. However, their values are derived from other attributes present in the database.

4.1.2 Main components and its symbols in ER Diagrams:

Rectangles: This Entity Relationship Diagram symbol represents entity types

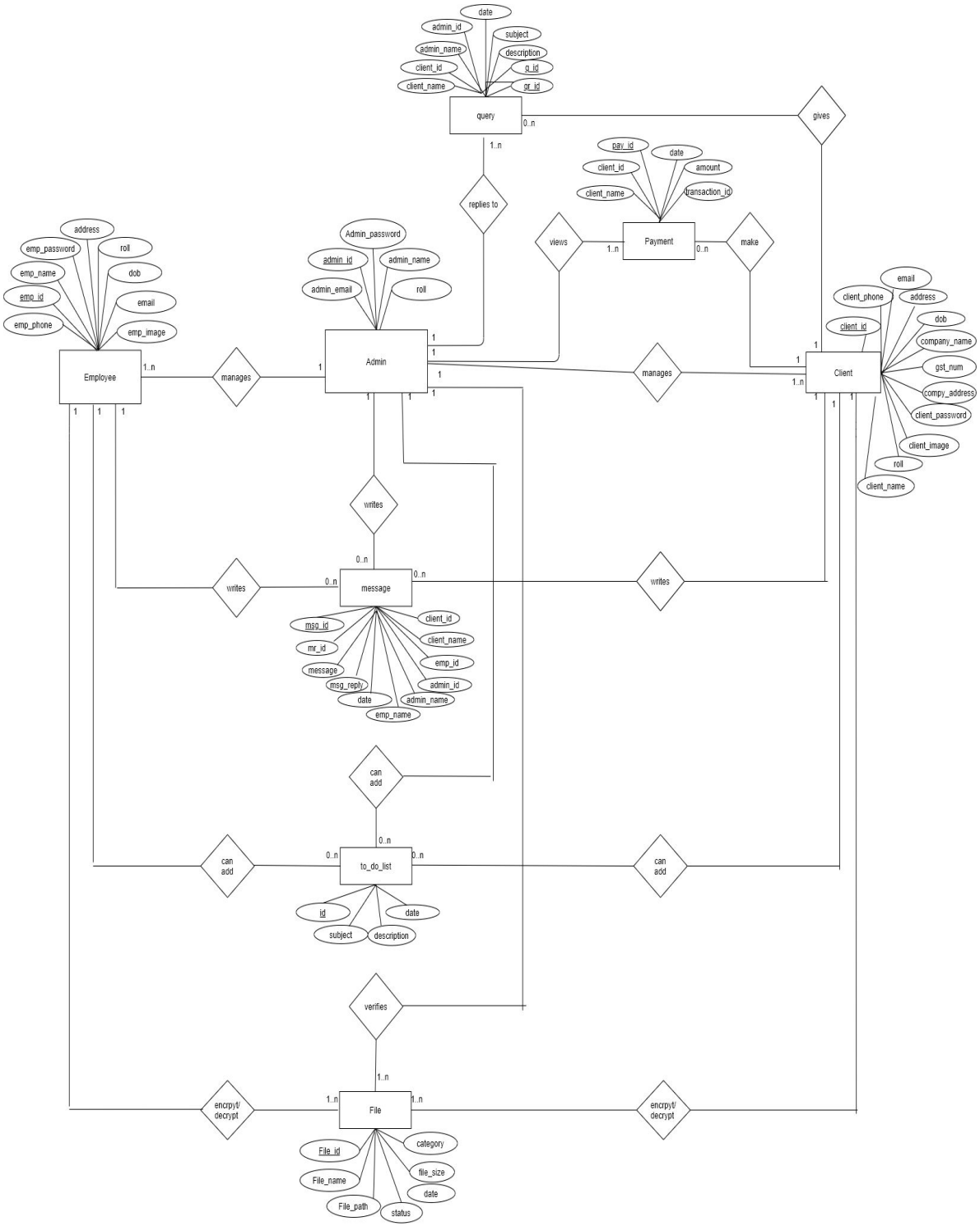
Ellipses: This Symbol represent attributes

Diamonds: This symbol represents relationship types

Lines: It links attributes to entity types and entity types with other relationship types

Primary key: attributes are underlined

Double Ellipses: Represent multi-valued attributes


date
admin_id
subject
admin_name
description
client_id
g_id
client_name
gr_id
query
gives
0..n
1..n
replies to
pay_id
date
client_id
amount
client_name
transaction_id
views
Payment
make
1..n
0..n
address
emp_password
roll
emp_name
dob
emp_id
email
emp_phone
emp_image
Admin_password
admin_id
admin_name
admin_email
roll
Employee
1..n
manages
1
Admin
manages
1
1..n
email
client_phone
address
client_id
dob
company_name
gst_num
compy_address
client_password
client_image
roll
client_name
Client
writes
0..n
writes
0..n
message
0..n
writes
msg_id
client_id
client_name
mr_id
emp_id
message
admin_id
msg_reply
date
admin_name
emp_name
can add
0..n
can add
0..n
to_do_list
0..n
can add
id
date
subject
description
verifies
1..n
encrpyt/ decrypt
1..n
File
1..n
encrpyt/ decrypt
File_id
category
File_name
file_size
date
File_path
status

4.2 Table Description

4.2.1 Table Name : Admin

Column_name	Type	size	notnull	key
a_id	INTEGER	10	Notnull	Primarykey
admin_name	VARCHAR	30	Notnull	
admin_email	VARCHAR	20	Notnull	
Phone_number	BIGINT	10	Notnull	
admin_img	VARCHAR	200	Notnull	
roll	VARCHAR	5	Notnull	
admin_password	VARCHAR	200	Notnull	

4.2.2 Table Name : employee

Column_name	type	size	notnull	key
e_id	INTEGER	10	Notnull	Primarykey
emp_name	VARCHAR	20	Notnull	
emp_phone	BIGINT	10	Notnull	
emp_email	VARCHAR	20	Notnull	
emp_address	VARCHAR	30	Notnull	
emp_dob	DATE		Notnull	
gender	VARCHAR	10	Notnull	
emp_img	VARCHAR	200	Notnull	
roll	VARCHAR	5	Notnull	
creation_date	DATE & TIME		Notnull	
emp_password	VARCHAR	200	Notnull	

4.2.3 Table Name : client

Column_name	Type	size	Notnull	key
c_id	INTEGER	10	Notnull	Primarykey
client_name	VARCHAR	20	Notnull	
client_phone	BIGINT	10	Notnull	
client_email	VARCHAR	20	Notnull	
client_address	VARCHAR	30	Notnull	
company_name	VARCHAR	30	Notnull	
gst_num	VARCHAR	20	Notnull	
company_address	VARCHAR	30	Notnull	
client_img	VARCHAR	200	Notnull	
roll	VARCHAR	5	Notnull	
client_password	VARCHAR	200	Notnull	
creation_date	DATE & TIME		Notnull	

4.2.4 Table Name : login

Column_name	Type	size	Notnull	key
id	INTEGER	10	Notnull	Primarykey
name	VARCHAR	20	Notnull	
email	VARCHAR	20	Notnull	
roll	VARCHAR	5	Notnull	
password	VARCHAR	200	Notnull	
creation_date	DATE & TIME		Notnull	

4.2.5 Table Name : assigned_to_emp

Column_name	type	Size	notnull	key
asc_id	INTEGER	10	Notnull	Primarykey
e_id	INTEGER	10	Notnull	Foriegnkey
c_id	INTEGER	10	Notnull	Foriegnkey

4.2.6 Table Name : audited_files

Column_name	Type	size	Notnull	key
audit_id	INTEGER	10	Notnull	Primarykey
e_id	INTEGER	10	Notnull	Foriegnkey
c_id	INTEGER	10	Notnull	Foriegnkey
category_id	INTEGER	10	Notnull	Foriegnkey
audit_filename	VARCHAR	60	Notnull	
audit_fsize	VARCHAR	20	Notnull	
audit_file	VARCHAR	200	Notnull	
audit_date	DATE & TIME		Notnull	
status	VARCHAR	10	Notnull	

4.2.7 Table Name : file_category

Column_name	type	Size	notnull	Key
category_id	INTEGER	10	Notnull	Primarykey
category_name	VARCHAR	30	Notnull	
description	VARCHAR	100	Notnull	
creation_date	DATE & TIME		Notnull	

4.2.8 Table Name : files

Column_name	Type	Size	notnull	Key
file_id	INTEGER	10	Notnull	Primarykey
category_id	INTEGER	10	Notnull	Foriegnkey
c_id	INTEGER	10	Notnull	Foriegnkey
file_name	VARCHAR	70	Notnull	
file_size	VARCHAR	20	Notnull	
file	VARCHAR	200	Notnull	
status	VARCHAR	10	Notnull	
fdate	DATE & TIME		Notnull	

4.2.9 Table Name : admin_msg_to_client

Column_name	type	Size	notnull	key
m_id	INTEGER	10	Notnull	Primarykey
c_id	INTEGER	10	Notnull	Foriegnkey
message	VARCHAR	300	Notnull	
msg_from	VARCHAR	20	Notnull	
m_date	DATE & TIME		Notnull	

4.2.10 Table Name : admin_msg_to_emp

Column_name	type	Size	notnull	key
m_id	INTEGER	10	Notnull	Foriegnkey
e_id	INTEGER	10	Notnull	Foriegnkey
message	VARCHAR	300	Notnull	
msg_from	VARCHAR	20	Notnull	
date	DATE & TIME		Notnull	

4.2.11 Table Name : emp_message_to_client

Column_name	type	Size	notnull	key
m_id	INTEGER	10	Notnull	Foriegnkey
e_id	INTEGER	10	Notnull	Foriegnkey
c_id	INTEGER	10	Notnull	Foriegnkey
message	VARCHAR	300	Notnull	
msg_from	VARCHAR	20	Notnull	
date	DATE & TIME		Notnull	

4.2.12 Table Name : query

Column_name	type	Size	notnull	key
q_id	INTEGER	10	Notnull	Primarykey
c_id	INTEGER	10	Notnull	Foriegnkey
Subject	VARCHAR	20	Notnull	
Description	VARCHAR	250	Notnull	
creation_date	DATE & TIME		Notnull	

4.2.13 Table Name : query_reply

Column_name	type	Size	notnull	key
qr_id	INTEGER	10	Notnull	Primarykey
q_id	INTEGER	10	Notnull	Foriegnkey
c_id	INTEGER	10	Notnull	Foriegnkey
Message	VARCHAR	250	Notnull	
r_date	DATE & TIME		Notnull	

4.2.14 Table Name : admin_todolist

Column_name	type	Size	notnull	key
id	INTEGER	10	Notnull	Primarykey
title	VARCHAR	20	Notnull	
description	VARCHAR	250	Notnull	
date	DATE & TIME		Notnull	

4.2.15 Table Name : emp_todolist

Column_name	type	Size	notnull	key
id	INTEGER	10	Notnull	Primarykey
e_id	INTEGER	10	Notnull	Foriegnkey
title	VARCHAR	20	Notnull	
description	VARCHAR	250	Notnull	
date	DATE & TIME		Notnull	

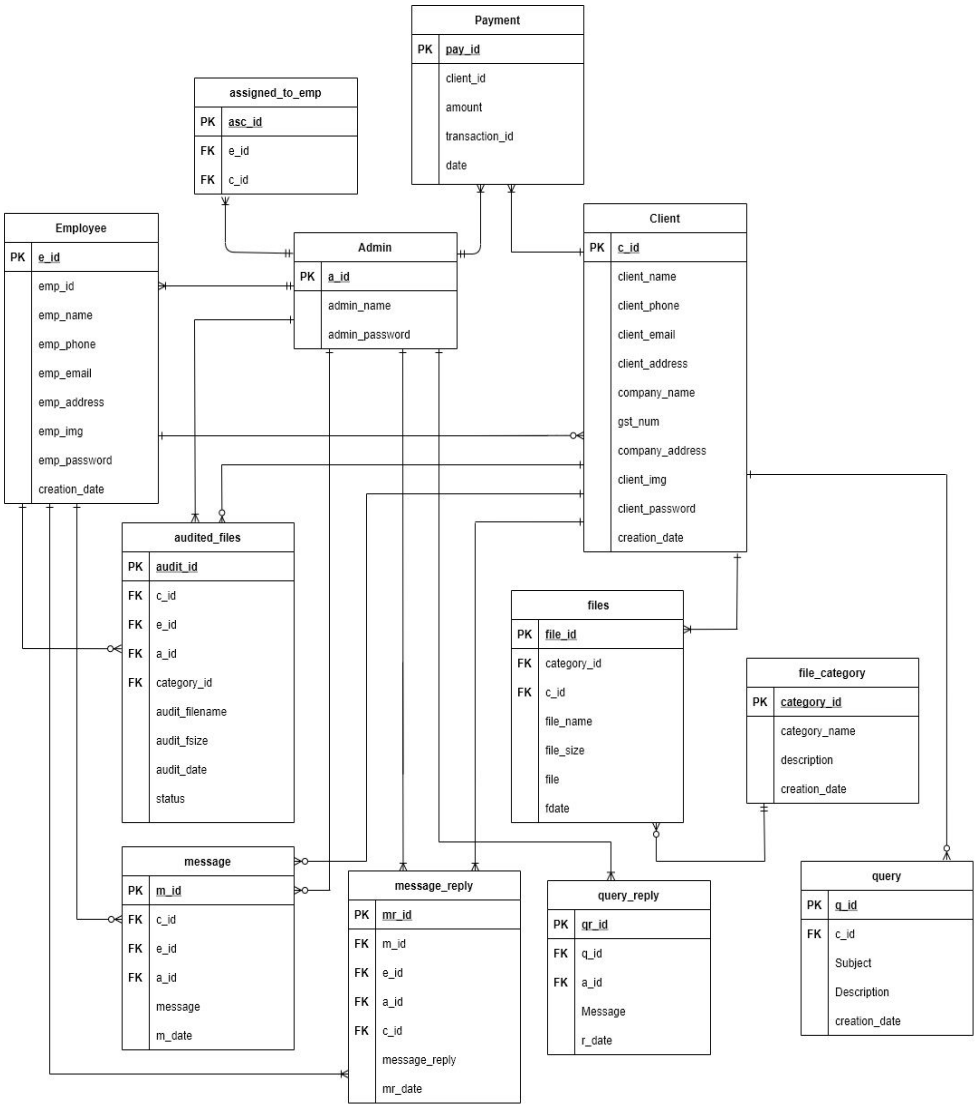
4.2.16 Table Name : cl_todolist

Column_name	type	Size	notnull	key
id	INTEGER	10	Notnull	Primarykey
c_id	INTEGER	10	Notnull	Foriegnkey
title	VARCHAR	20	Notnull	
description	VARCHAR	250	Notnull	
date	DATE & TIME		Notnull	

4.2.17 Table Name : Payment

Column_name	type	Size	notnull	key
pay_id	INTEGER	10	Notnull	Primarykey
c_id	INTEGER	10	Notnull	Foriegnkey
amount	VARCHAR	50	Notnull	
transaction_id	VARCHAR	90	Notnull	
date	DATE & TIME		Notnull	

4.3Table Relationship



Testing and Validation

6. TESTING AND VALIDATION

6.1 Introduction

Software testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test.Testing has been defined as the process of analysing a software item to detect the differences between existing and required conditions and to evaluate the features of the software item. Software testing is the process used to assess the quality of computer software.

It involves operation of a system or application under controlled conditions and evaluating the results. The controller conditions should include both normal and abnormal conditions. Testing should intentionally attempt to make things go wrong to determine if things happen when they should. It is oriented to ‘detection’.

Software testing has three main purposes

- The verification process confirms that the software meets its technical specifications. A “specification” is a description of a function in terms of a measurable output value given a specific input value under specific preconditions.
- The validation process confirms that the software meets the business requirements.
- A defect is a variance between the expected and actual result. The defect’s ultimate source may be traced to a fault introduced in the specification, design, or development phases. Not all the defect will necessarily result in failures.

There are two types of software testing:

- Black box testing - internal system design is not considered in this type of testing. Tests are based on requirement and functionality.
- White box testing - Thus testing is based on knowledge of the internal logic of an application's code. Also known as glass box testing. Internal software and code working should be known for this type of testing. Test are based on coverage of code statements, branches, paths and conditions.

A test case is a software testing document, which consist of event, action, input, output, expected result and actual result. Clinically defined a test case is an input and an expected result. This can be a pragmatic as ‘for condition x your derived result y, whereas other test cases described in more detail the input scenario and what results might be expected. It can occasionally be a series of steps but one with expected results or expected outcome. A test case should also contain a place for the actual result.

White box testing is applicable at the unit, integration and system levels of the software testing access. We have performed two of these, explained below:

6.2 Testing Objective

- Finding defects which may get created by the programmer while developing the software
- Gaining confidence in and providing information about the level of quality
- To prevent defects
- To make sure that the end results meets the business and user requirements.
- To ensure that is satisfies the Business Requirement Specification and System Requirement Specification.

6.2.1 Unit Testing

Unit tests are written from a programmer’s perspective. They ensure that a particular method of a class successfully performs a set of specific tasks. Each test confirms that a method produces the expected output when given a known input. A series of stand-alone tests are conducted during unit testing. Each test examines an individual component that is new or has been modified. Unit tests focus on functionality and reliability. Unit testing is done in a test environment prior to system integration. If a defect is discovered at unit test, the severity of the defect will dictate whether or not it will be fixed before the module is approved.

6.2.2 Integration Testing

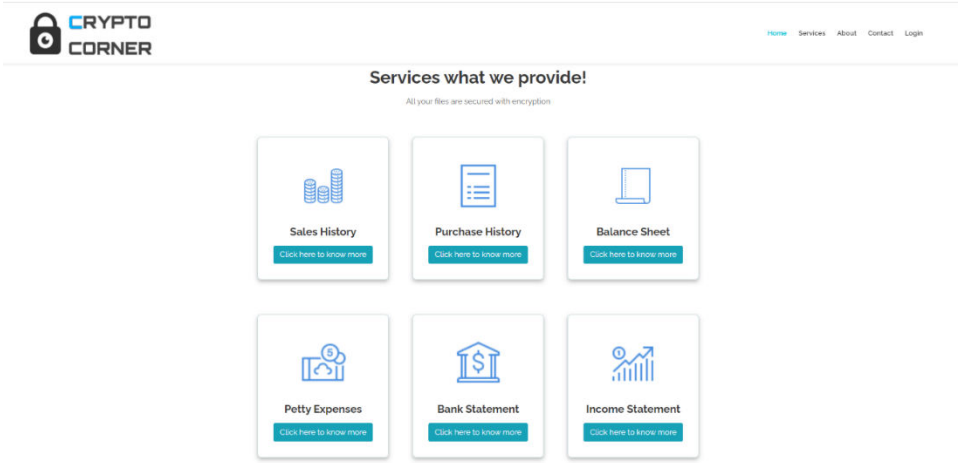
Integration testing examines all the component and modules that are new. changed. affected by a change, or needed to form a complete system. Integration testing requires involvement of other systems and interfaces with other applications, including those owned by an outside vendor, external partners or the customer.

It is the phase of software testing in which individual software modules are combined and tested as a group. It follows unit testing and precedes system testing, integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing.

Therefore without adequate testing, there is greater risk that an application will inadequately deliver what was expected by the users or that the final product will have problems such that the users will eventually abandon it out of frustration. in either case, time and money are lost and the credibility and reputation of both the developers and the software is damaged. more formal, rigorous testing will go far to reducing the risk that either of these scenarios occurs.

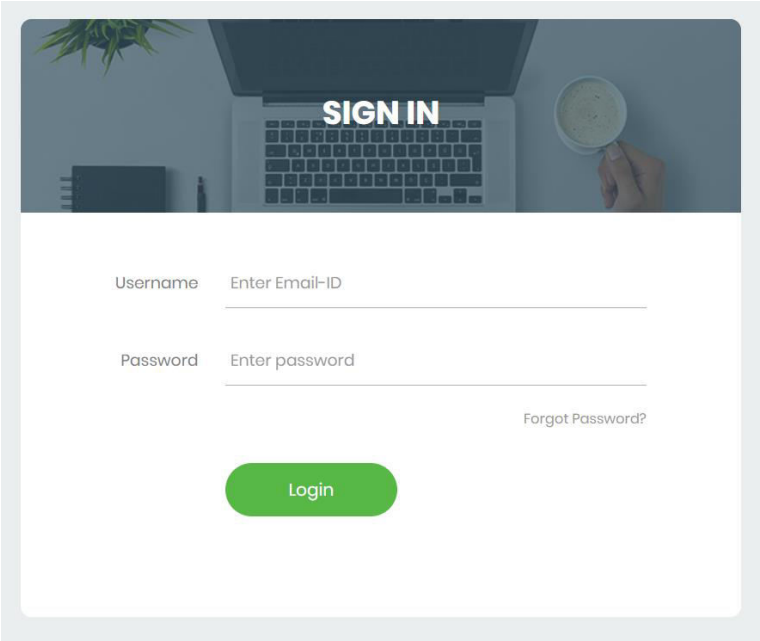
6.3 Unit Testing Plan

6.3.1 User Interface



Test Case ID	User event	User Input	Expected output	Result
1	Page (onLoad())	Loads the page	Displays the user interface page	Success
2	Home (onClick())	Loads the homepage	Displays user homepage	Success
3	Services (onClick())	Loads the services	Displays the services provided by the company	Success
4	About (onClick())	Loads the user page	Displays the information about the company and the CEO	Success
5	Contact (onClick())	Loads the user page	Displays the contact information of the admin	Success
6	Login (onClick())	Loads login page	User can enter login credentials to get logged in	Success

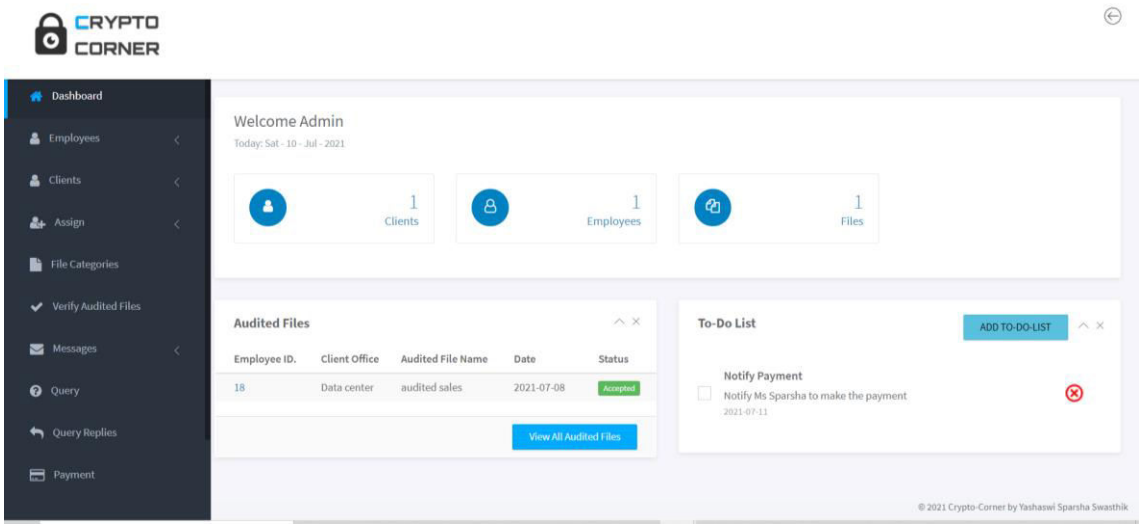
6.3.2 Login



Test Case ID	User event	User Input	Expected output	Result
1	Page (onLoad())	Loads the page	Displays the form	Success
2	User (onClick())	Loads the user login form	Users logging form must appear	Success
3	User (onKeyPress())	User enters username	Values are accepted	Success
4	Password (onKeyPress())	User enters password	Values are accepted	Success
5	Login (onClick())	User has entered invalid values	Prompt: “you are not a valid user”	Success
6	Login (onClick())	User has entered wrong password	Prompt: “you have entered wrong password”	Success
7	Login (onClick())	User has entered valid values and clicks the button	User panel is displayed	Success
8	Login (onClick())	If user submits blank form	“username is required” “password is required”	Success
9	forgotPassword (onClick())	Loads the forgot password page	Displays the form	Success
10	Submit (onClick())	If user submits blank form	“email ID is required” “registration date is required”	Success
11	Submit (onClick())	User has entered invalid values	“you have entered wrong details”	Success
12	Submit (onClick())	User has entered valid values and clicks the button	‘Set new password’ page is displayed	Success

13	Update (onClick())	User has to enter new password and confirm it	Prompt: “password changed successfully” Redirects to login page	Success

6.3.3 Admin Panel

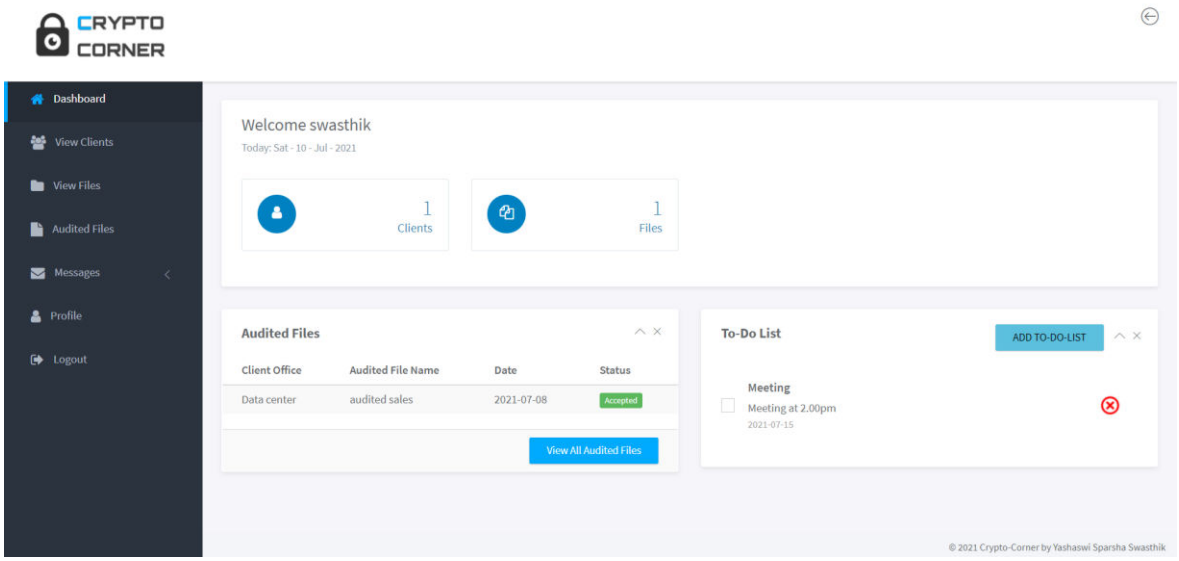


Test Case ID	User event	User Input	Expected output	Result
1	Page (onLoad())	Loads the page	Admin Dashboard appears	Success
2	Employees (onClick())	Displays dropdown with two options “add employee” and “view employee”	Admin can register and view the employees	Success
3	Add employee (onClick())	Loads the employee registration form	Admin can register the employee	Success

4	View employee (onClick())	Loads the employees list	Admin can view the employees working under him	Success
5	Clients (onClick())	Displays dropdown with two options “add client” and “view client”	Admin can register and view the clients	Success
6	Add client (onClick())	Loads the client registration form	Admin can register the client	Success
7	View clients (onClick())	Loads the clients list	Admin can view the clients and their files	Success
8	Assign (onClick())	Loads the assigning form	Admin will assign one employee to his client	Success
9	File categories (onClick())	Loads the form to add categories	Admin can add the file categories (services what he provides)	Success
10	Verify Audited File(onClick( ))	Loads the audited files form	Admin can accept or reject the audited files	Success
11	Accept (onClick())	Updates the status	Displays the files to client and displays the status to employee	Success
12	Reject (onClick())	Updates the status	Displays the status “rejected” to employee	Success
13	Messages (onClick())	Displays dropdown button with two options “employee messages” and “client messages”	Admin can choose to message the users	Success
14	employee messages (onClick())	Displays the list of employees working under the admin	Admin can choose to message any of the employee	Success

15	client messages (onClick())	Displays the list of clients	Admin can choose to message any of the client	Success
16	Query (onClick())	Loads the queries	Displays the queries given by the clients	Success
17	Reply (onClick())	Loads the query reply form	Admin can give replies to the queries	Success
18	Delete (onClick())	Admin clicks the button	Deletes the query	Success
19	Query replies (onClick())	Loads the queries	Displays the queries given by the clients	Success
20	Payment (onClick())	Loads the Payments page	Displays all the payments made by the client	Success
21	Add to-do-list (onClick())	Displays new to-do-list form	Admin can create tasks for himself	Success
22	Add (onClick())	Clicks the button	New To-do-list is added successfully	Success
23	Close (onClick())	Clicks the button	Closes the to-do-list form	Success
24	Change (onClick())	Loads profile picture changing form	Admin can change the profile picture	Success
25	Change password (onClick())	Loads the change password page. admin has to enter new password and confirm it	Admin can change the password	Success
26	Update (onClick())	Admin can update his personal details	“profile updated successfully”	Success
27	Logout (onClick())	Clicks the button	Admin exits from his panel and gets back to login page	Success

6.3.4 Employee Panel

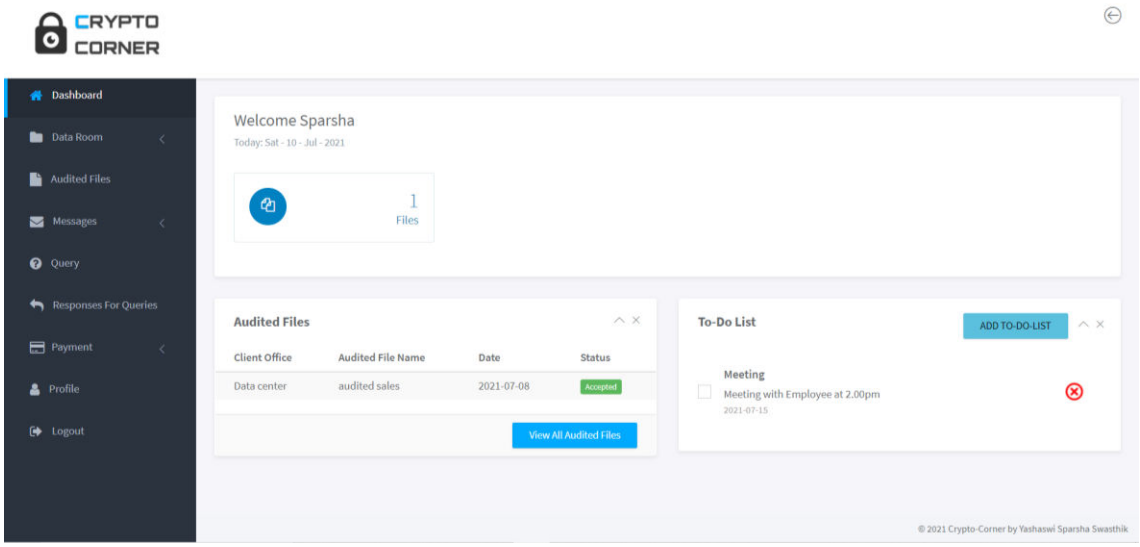


Test Case ID	User event	User Input	Expected output	Result
1	Page (onLoad())	Loads the page	Employee Dashboard appears	Success
2	View clients (onClick())	Loads the clients list	Employee can view the clients and their files	Success
3	Clients (onClick())	Loads the client list	Employee view the clients	Success
4	View files (onClick())	Loads the list of client files	Employee can view the files uploaded by all his clients	Success
5	Download (onClick())	Clicks the button	Downloads the particular file	success
6	Audited File (onClick())	Loads the list of audited files	Displays all the files audited by employee	Success
7	Upload audited files (onClick())	Loads audited file uploading form	Employee can choose particular client and file	Success

			category and then upload the audit file	
8	Upload file (onClick())	Clicks the button	Uploads file to admin for verification	Success
9	Close (onClick())	Clicks the button	Closes file uploading form	Success
10	Messages (onClick())	Displays dropdown button with two options “admin messages” and “client messages”	Employee can choose the users to message	Success
11	Admin messages (onClick())	Loads admin chat option	Employee can chat with admin	Success
12	client messages (onClick())	Displays the list of clients	Employee can choose to message any of his client	Success
13	Add to-do-list (onClick())	Displays new to-do-list form	Employee can create tasks for himself	Success
	Add (onClick())	Clicks the button	New To-do-list is added successfully	Success
14	Close (onClick())	Clicks the button	Closes the to-do-list form	Success
15	Update profile (onClick())	Clicks the button	Employee can update his details	success
16	Change (onClick())	Loads profile picture changing form	employee can change the profile picture	Success
17	Change password (onClick())	Loads the change password page. employee has to	employee can change the password	Success

		enter new password and confirm it		
18	Update (onClick())	employee can update his personal details	“profile updated successfully”	Success
19	Logout (onClick())	Clicks the button	Employee exits from his panel and gets back to login page	Success

6.3.5 Client Panel



Test Case ID	User event	User Input	Expected output	Result
1	Page (onLoad())	Loads the page	Client Dashboard appears	Success
2	Data room (onLoad())	Displays dropdown button with two options “upload files” and “view files”	Client can either choose to upload file or to view file	Success
3	Upload File (onClick())	Loads the files uploading form	Client can choose the particular category and	Success

			upload the files to employee	
4	Upload (onClick())	Clicks the button	Files uploaded successfully	Success
5	View files (onClick())	Loads file categories	Client can view files based on categories	Success
6	Audited Files (onClick())	Loads the list of audited files	Displays all the files audited by employee	Success
7	Messages (onClick())	Displays dropdown button with two options “employee messages” and “admin messages”	client can choose to message the other users	Success
8	employee messages (onClick())	Loads employee name	client can message the employee	Success
9	Admin messages (onClick())	Displays the admin name	Client can message the admin	Success
10	Query (onClick())	Loads the query form	Client can give queries or feedbacks to admin based on employee’s performance	Success
11	Responses for queries (onClick())	Loads the query reply form	Clients can view the query replies given by the admin	Success
12	Delete (onClick())	clicks the button	Deletes the query	Success
13	Payment (onClick())	Displays dropdown button with two options “make payment” and “view payment”	Client can make payments and also can view the payments that are already made	Success

14	Make payment (onClick())	Loads the payment form	Client can make payment by scaning the QR code	Success
15	Pay (onClick())	Clicks the button	Payment is sent and transaction details are displayed to admin	Success
16	View payment (onClick())	Loads the page of payment that are already made	Client can view all the payments he made	Success
17	Send message (onClick())	Loads admin chat box	Client can inform admin about the payments made	Success
18	Add to-do-list (onClick())	Loads new to-do-list form	client can create tasks for himself	Success
19	Add (onClick())	Clicks the button	New To-do-list is added successfully	Success
20	Close (onClick())	Clicks the button	Closes the to-do-list form	Success
21	Profile (onClick())	Loads the client information form	Displays client profile	Success
22	Update profile (onClick())	Loads client profile form to update	Client can modify the details	Success
23	Change (onClick())	Loads profile picture changing form	Client can change the profile picture	Success
24	Change password (onClick())	Loads the change password page. client has to enter new password and confirm it	Client can change the password	Success
25	Update (onClick())	Client can update his personal details	“profile updated successfully”	Success

26	Back (onClick())	Loads the client information form	Client profile is displayed where information cannot be edited	
27	Logout (onClick())	Clicks the button	client exits from his panel and gets back to login page	Success

BIBLIOGRAPHY

- Daemen, J., & Rijmen, V. (2013). *The Design of Rijndael : AES – The Advanced Encryption Standard.* New York. Springer
- Denis. T. (2007). *Cryptography for Developers*. Amsterdam. Elsevier Science
- Menezes. A. J., Vanstone. S. A., & van Oorschot. P. C. (2018). *Handbook of Applied Cryptography*. United States. CRC Press.
- Pilone. D., & Pitman. N. (2009). *UML 2.0 in a Nutshell.* Sebastopol, California. O'Reilly Media.
- Pelzl. J., Preneel. B., & Paar C. (2009). *Understanding Cryptography: A Textbook for Students and Practitioners*. New York. Springer
- Stinson. D. R., & Paterson. M. (2018). *Cryptography: Theory and Practice*. United States. CRC Press.
- Seidl. M., Scholz. M., Huemer. C., & Kappel. G. (2015). *The Use Case Diagram*. Switzerland. Springer International Publishing.
- Knudsen. L. R., Robshaw. M. J. (2011). *The Block Cipher Companion*. Berlin. Springer.
- Barbudhe. V. K., Zanjat. S. N., & Karmore. B. S. (2020). *Cryptography and Network Security Operation*. Chennai. Notion press media Pvt. ltd.
- Hathaway. A., & Hathaway. T. (2016). *Data Flow Diagrams – Simply Put!: Process Modelling Techniques for Requirements Elicitation and Workflow Analysis.* Scotts Valley. CreateSpace Independent Publishing Platform.
- Buchmann. J. (2004). *Introduction to Cryptography.* New York. Springer.
- Bagui. S., & Earp. R. (2003). Database Design Using Entity-Relationship Diagrams. Boca Raton. Auerbach Publications(CRC Press).
- Grover. D. (2012). *Object Oriented Analysis and Design with UML.* Delhi. I K International Publishing House Pvt. Ltd
- Trappe W., & Washington. L. C. (2006). *Introduction to Cryptography: With Coding Theory.* New York. Pearson Prentice Hall.
- Sridhar. S. (2013). Cryptography & Network Security . Chennai. Charulatha Publications.
- Roff. J. T. (2002). *UML: A Beginner's Guide.* Berkeley. Mcgraw-Hill Osborne Media.

- Jackson. M., & Jackson. J. (1995). *Software Requirements & Specifications: A Lexicon of Practice, and Prejudices.* New York. ACM Press.
- Rumpe. B. (2016). *Modelling with UML.* Switzerland. Springer International Publishing
- K. K. Aggarwal., &Singh. Y. (2007). *Software Engineering*. Delhi. New Age International Publishers.
- Blaha. M. R., & Rumbaugh. J. R. (2004). *Object Oriented Modelling and Design with UML*. New York. Pearson
- https://www.ijecs.in/index.php/ijecs/article/download/3630/3378/
- https://www.youtube.com/watch?v=gP4PqVGudtg
- https://www.tutorialspoint.com/cryptography/advanced_encryption_standard.htm
- https://www.youtube.com/watch?v=YVT4fcW7sI8&t=1392s
- https://www.tutorialspoint.com/cryptography/block_cipher_modes_of_operation.htm
- https://searchsecurity.techtarget.com/definition/Advanced-Encryption-Standard
- https://www.youtube.com/watch?v=X8whYEWoDSI&t=796s
- https://www.n-able.com/blog/aes-256-encryption-algorithm
- https://www.devglan.com/online-tools/aes-encryption-decryption
- https://www.youtube.com/watch?v=DMtFhACPnTY
- https://www.educative.io/edpresso/what-is-the-aes-algorithm
- https://www.youtube.com/watch?v=O4xNJsjtN6E
- https://krazytech.com/projects/sample-software-requirements-specificationsrs-report-airline-database
- https://www.reqview.com/doc/iso-iec-ieee-29148-srs-example
- https://personal.utdallas.edu/~chung/RE/Presentations07S/Team_1_Doc/Documents/SRS4.0.doc
- http://www.learnerswindow.com/software-requirement-specification-srs-on-e-commerce/
- https://www.studocu.com/in/document/rashtrasant-tukadoji-maharaj-nagpur-university/book-keeping-and-accountancy/tutorial-work/system-requirement-specification-for-ecommerce-web-store/5353426/view
- https://www.edrawmax.com/context-diagram/
- https://www.visual-paradigm.com/support/documents/vpuserguide/94/2577/7025_drawingseque.html
- https://developer.ibm.com/technologies/web-development/articles/the-sequence-diagram/

CONCLUSION

The project **" Crypto - Corner "** is successfully tested. This has been developed to fulfil the requirements of **Blue Bird Data Solutions Pvt. Ltd**. This system is developed to be user-friendly and interactive. The website is found to be beneficial for the concerned aspects. The website is responsive and can be used on any device. This software has been tested with all possible sample data and was found to function efficiently.

The system provides security to the users data by using AES encryption, also efficient insertion and retrieval of data. The users can easily add data, chat, make payment, give queries, edit profile, change password if required, view reports. The paper work and manual work is reduced as the system itself holds the record. It is helpful to maintain the data security of users.

Although it's a successfully running application, it can be still modified if required without disrupting the working part of the application.

ANNEXURES

No Annexures