Palestine Polytechnic University
College of Information Technology and Computer Engineering

Graduation Project Title:

“BlurPic” A Mobile Application for encrypting and decrypting ROI in images

Team Members:

Hiba Badawi

Sara Taweel

Tala Abu Shameh

Supervisors:

Mohamad Abu Taha
Tareq Alajloni

2023

Acknowledgment

First and foremost, praises and thanks to God, the Almighty, for His showers of blessings throughout our project.

We are very grateful to our parents for their love, prayers, caring, and sacrifices for educating and preparing us for the future, and supporting us during this project.

We also extend our thanks and appreciation to Palestine Polytechnic University, which embraced us to complete our studies and obtain a bachelor's degree, and to the College of Information Technology and Computer Engineering staff. We want to express our deep and sincere gratitude to our project supervisor Dr. Mohammad Abu Taha for allowing us to do this project and providing us with invaluable guidance throughout this work.

Last but not least, we would also like to thank our families, friends, all the teachers, and everyone who supported us directly or indirectly.

الملخص

في ظل التقدم التكنولوجي السريع أصبحت وسائل التكنولوجيا الحديثة جزءاً أساسياً من يومنا العادي، فقد انعكس ذلك على طريقة التواصل فيما بيننا، حيث لا يكاد أن يمر يوم دون ارسال عشرات الصور عبر مختلف المواقع الاجتماعية، لا شك أنه هذه الصور قد تحتوي على محتوى هام.

في معظم الأحيان تحتوي الصور على معلومات حساسة كوجه واضح في الصورة والذي يحتاج الى اخفاءه لعدة أغراض، حيث الطريقة المعتادة لحماية هذه الصور قد تكون بالتشفير الكلي لإخفاء كل شيء لتحقيق الأمان الكامل، لكن هذه الطريقة قد تحمل الكثير من عمليات التشفير غير المرجوة، لذا هنا تظهر أهمية التشفير الجزئي للأجزاء المهمة فقط من إطار الصورة باستخدام خوارزميات تحقق الغاية.

أخيراً، هذا المشروع سيسمح لمختلف الأفراد و المنظمات تطبيق فكرة " منطقة الاهتمام" لضمان حماية المعلومات عبر الإنترنت باستخدام خوارزمية "معايير التشفير المتقدمة" التي تعتبر الأكثر أماناً وذلك بسبب طول مفتاح التشفير، و سيسمح بتبادل الصور المشفرة مع الاخرين مع اتاحة فك تشفير الصورة في حال الحصول على مفتاح التشفير.

Abstract

In light of the rapid technological progress, the means of modern technology have become an essential part of our ordinary day. This has been reflected in the way we communicate with each other, as hardly a day goes by without sending dozens of images through various social sites. There is no doubt that these images may contain important content.

Most of the time, the images contain sensitive information, such as a clear face in the image that needs protection, as the usual way to protect these images may be with total encryption to hide everything to achieve complete security, but this method may carry a lot of unwanted encryption operations, so here it appears The importance of partial encoding only the important parts of the image frame using algorithms that achieve the goal.

Finally, this project will allow various individuals and organizations to implement the idea of a ROI "area of interest" to ensure the protection of information over the Internet using the Advanced Encryption Standards ”AES” algorithm, which is considered the most secure way due to the length of the encryption key. And it will allow the exchange of encrypted images with others, with the possibility of decrypting the image in case of obtaining the decryption key which is the same encryption key.

Contents

Figures

Tables

Chapter 1: Introduction

1.1 Overview of the Project

Many changes have occurred in our society as a result of the development of information and communication technologies. The IoT (Internet of Things) and big data are recent examples of new communication technologies that have made it easier and more efficient to transfer and use different sorts of information. By utilizing computers and mobile devices, it is feasible to gather information fast and easily and increase social interaction opportunities via social networking platforms.

Additionally, there is a greater need for real-time data transmission in fields including healthcare, defense, finance, and education. Furthermore, the COVID-19 pandemic has caused numerous changes in our society's day-to-day operations, including an increase in the requirement for real-time multimedia security due to the rise in real-time non-face-to-face online meetings, education, telemedicine, and online collaboration.

The protection of personal information is a concern for multimedia traffic, which includes video, audio, and image content. Problems like personal information leakage and cyber terrorism also commonly arise. Many methods have been developed to safeguard images because image security is a crucial component of modern communication technologies and crucial for safe transmission.

The most user-friendly and efficient technology for transforming an image into an unrecognizably altered image is image encryption. By using this technique, it is easy to avoid image theft, unauthorized image reading, and the disclosure of personal information while exchanging photographs.

We suggest the biometric data-based region of interest (ROI) picture encryption architecture. The suggested architecture makes sure that, depending on their level of authority, various users are only permitted to access particular portions of an image.

1.2 Theoretical Background

This section provides some information about some technologies and algorithms that have been used in our project.

Face Detection:

Face detection is a computer technology that determines the location and size of a human face in a digital media “Video or image”. Face detection has been a standout among topics in computer vision literature.

Face detection is the stepping stone to all facial analysis algorithms, including face alignment, face modeling, face recognition, face verification/authentication, and many mor, so, computers can understand faces clearly, and then begin to understand people’s thoughts and intentions.

The primary goal of face detection is to determine whether or not there are any faces in the image. [1]

Image Processing:

These days, image processing has become widely used in many fields, including visualization, Pattern recognition, and image recognition. Image processing is a method that transforms an image into a digital form to perform some operations on it to get an enhanced result of the image, obtain specific models, or take some useful information from it. The input of this process is a video sector or image, and the output may be an enhanced image or characteristics/features associated with that image [2].

Image processing involves the following three steps [2].

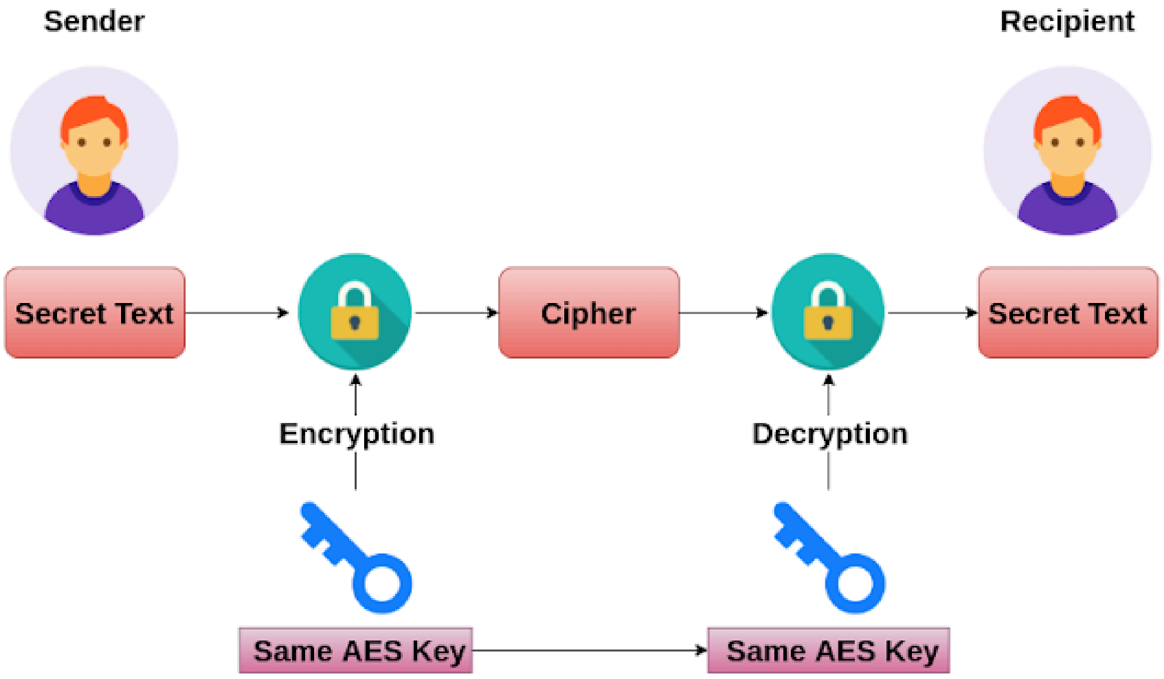
- Importing the image via image acquisition using image acquisition tools.
- Analyzing and manipulating the image
- Output in which results can be enhanced based on image analysis.

Advanced Encryption Standard "AES"

Advanced Encryption Standard (AES) is an encryption standard used for securing information which is defined by The National Institute of Standard and Technology (NIST) of the United States.[3]

The AES algorithm is a symmetric-key block cipher in which both the sender and receiver use a single key to encrypt and decrypt the information.

AES algorithm is used for data encryption that can process with the data block of 128 bits and cipher key length of 128 bits. The usage of 128-bit cipher key to achieve high security, because 128-bit cipher key is difficult to break.[4]



Region of Interest

A region of interest (ROI) is the meaningful and important regions in the images that want to filter or operate on in some way. The use of ROI can avoid the processing of irrelevant image points and accelerate the processing. [5]

You can represent an ROI as a binary mask image, where pixels that belong to the ROI are set to 1 and pixels outside the ROI are set to 0.

Many intelligence and surveillance applications need to detect potential targets or regions of interest (ROIs) in digital imagery. These ROIs can then be used to automatically call for further sensing or other action to direct further analysis for target identification and recognition. As in our project, we use the ROI concept to detect human faces in the image away from other objects in order to encrypt them.

1.3 Motivation and Importance

Threats like cyberterrorism and the exposure of personal information are commonly arising recently. Anyone who regularly and continuously uses multimedia, especially images, should be concerned about the safety of private details like faces.

Human faces encryption is a common use case to comply with privacy laws and protect the identity of individuals who don't want to be recognized in the media or anywhere else online with the possibility to send the encrypted image via Emails.

Here some cases encouraging blurring the faces in the images:

- Blurring faces on product photos in online marketplaces.
- Protecting the privacy of people in photojournalism and news reporting in case they don't prefer to be shown online.
- Protecting the privacy of people on social media platforms.

1.4 Objectives

The core goal of the project is to create a mobile application that can identify and encrypt the faces in the image and share it via emails.

The main objectives of the project are:

1. Identify the human faces in an image and blur it.
2. Allowing to decrypt the blurring image by using the same encryption key
3. Sending the encrypted image via emails.
4. Maintains a high quality of encrypted faces away from the background.

5. Implement a new way for detecting and encrypting faces by mobile cameras instead of using heavy techniques.

1.5 Project Scope

This mobile application seeks to encrypt the faces of the captured photographs in the business world, medical, military fields, and multimedia systems to prevent unauthorized users from accessing them. It is targeted at everyone who owns a portable device with a camera.

1.6 Methodology

The project methodology is based on the cycle life development software, which begins with planning, then requirements analysis, then system design, then system development, followed by the system examination stage, and finally the application and maintenance of the system by following the (Agile) methodology.

1.7 Problem Analysis and Definition

Image penetration has become a widespread problem due to technological advances and the development of image-capture devices used in various areas. This may lead to a lack of protection, mainly if these technologies are applied in critical places such as online networks or even universities, schools, and hospitals where they have millions of faces for people in their systems

Accordingly, it is necessary to protect enterprise data. By encrypting the faces in images, we protect corporate data from any incidents that might attack it, whether intentionally or unintentionally, and ensure that data is secure in the event that attackers bypass the firewall.

1.8 Description of the system

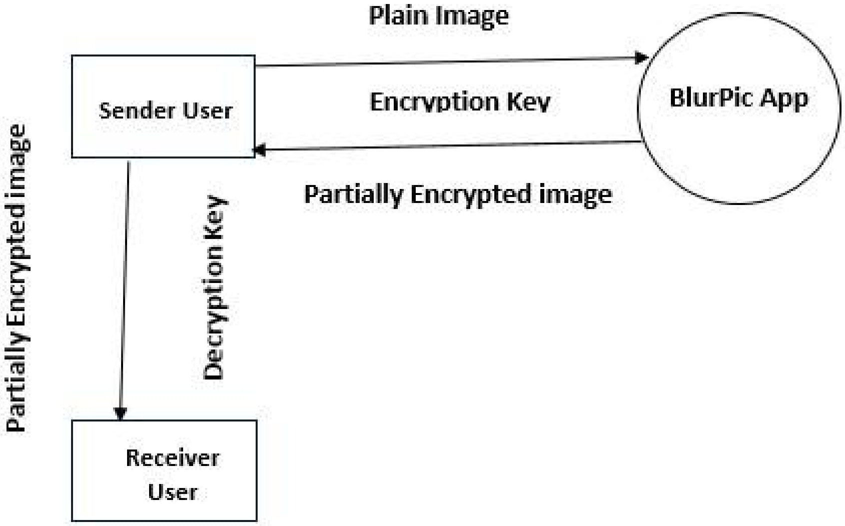
To help people protect their faces in the images, we built a mobile application that encrypts the region of interest (ROI) in the image, which is faces.

When a user takes a picture or picks a picture from a gallery, the Machine Learning “ML” kit checks if there are faces in the selected picture in order to detect them. If the result from machine learning indicates the faces existing, then we transfer the image to Base64 encoded text format. Then, we encrypt the text using the Advanced Encryption Standard "AES" principle. Accordingly, users can download, save and share the encrypted faces via emails with permission to decrypt the image in case they have the encryption key.

1.9 Expected Results

The Expected results upon completion of the project are encrypting the sensitive area of the captured image and preventing intruder users from accessing the personal image content and information.

1.10 Context Diagram



Chapter 2: Requirements Specification

2.1 Introduction

In this chapter, we will determine the functional and non-functional requirements of the proposed system in addition to a use case diagram, class diagram, and use case scenario tables.

2.2 Functional Requirements

The objective of these requirements is to define the functional aspects of the application, how it interacts with inputs and outputs, and how it behaves in some instances, which include these points:

1. For User
    - Sign Up
    - Log in
    - Insert an Image
    - Send the encrypted image to any receiver.
    - Decrypt the encrypted image.
    - Save or Delete the Decrypted image.

2.2. Nonfunctional Requirements

There are a set of agreed requirements by which image encryption can be developed:

1. Security: The application provides information security molecules via password to prevent unauthorized users from accessing the data, and encryption key to view the image's content
2. Ease of accessing the Application where the user can access the application using the mobile phone's camera.
3. Accuracy: the application performs its tasks at a high level of accuracy.
4. Ease of dealing with the application.

2.3. Describe and analyze the functional requirements tables for the system

Description of system requirements

Functional requirements for user:

Requirement Name	**Sign Up**
Actor	User
Description	The user has the ability to create account in the system
Inputs	Email, Password, Confirm password
Outputs	Have access to the Login and application features
Procedures	- Open the Login screen. - Write your Email, password, and confirm password - Click on the SignUp button.
Exceptions	- Internet disconnection. - The email already exists. - The password is weak.

Table 3: Explain how the user can Sign Up in the application

Requirement Name	**log in**
Actor	User
Description	The user has the ability to log in to the system
Inputs	Email and Password
Outputs	Have access to the application features
Procedures	- Open the application screen. - Write your email and password. - Click on the login button.
Exceptions	- Internet disconnection. - The email or password is incorrect.

Table 4:Explain how the user can Log in to the application

Requirement Name	**Insert an Image**
Actor	User
Description	The user can enter the needed image in which the face will be encrypted
Inputs	Image
Outputs	Encrypted face
Procedures	- Open the application using a phone connected to the Internet. - Login. - Click on the "pick image from gallery" option or "pick image from camera" from the main page. - Select the image you want to encrypt.
Exceptions	- Internet disconnection. - The image does not contain a face to Encrypt.

Table 5: Explain how the user can insert an image in the application

Requirement Name	**Send the encrypted image to any receiver**
Actor	User
Description	The user has the ability to send the encrypted image to the receiver.
Inputs	Image and the Encryption key.
Outputs	The encrypted image.
Procedures	- Open the application connected to the Internet. - Log in. - Insert the image you want to encrypt. - Enter the Encryption key. - Enter the receiver email
Exceptions	- Internet disconnection - The receiver's email is wrong.

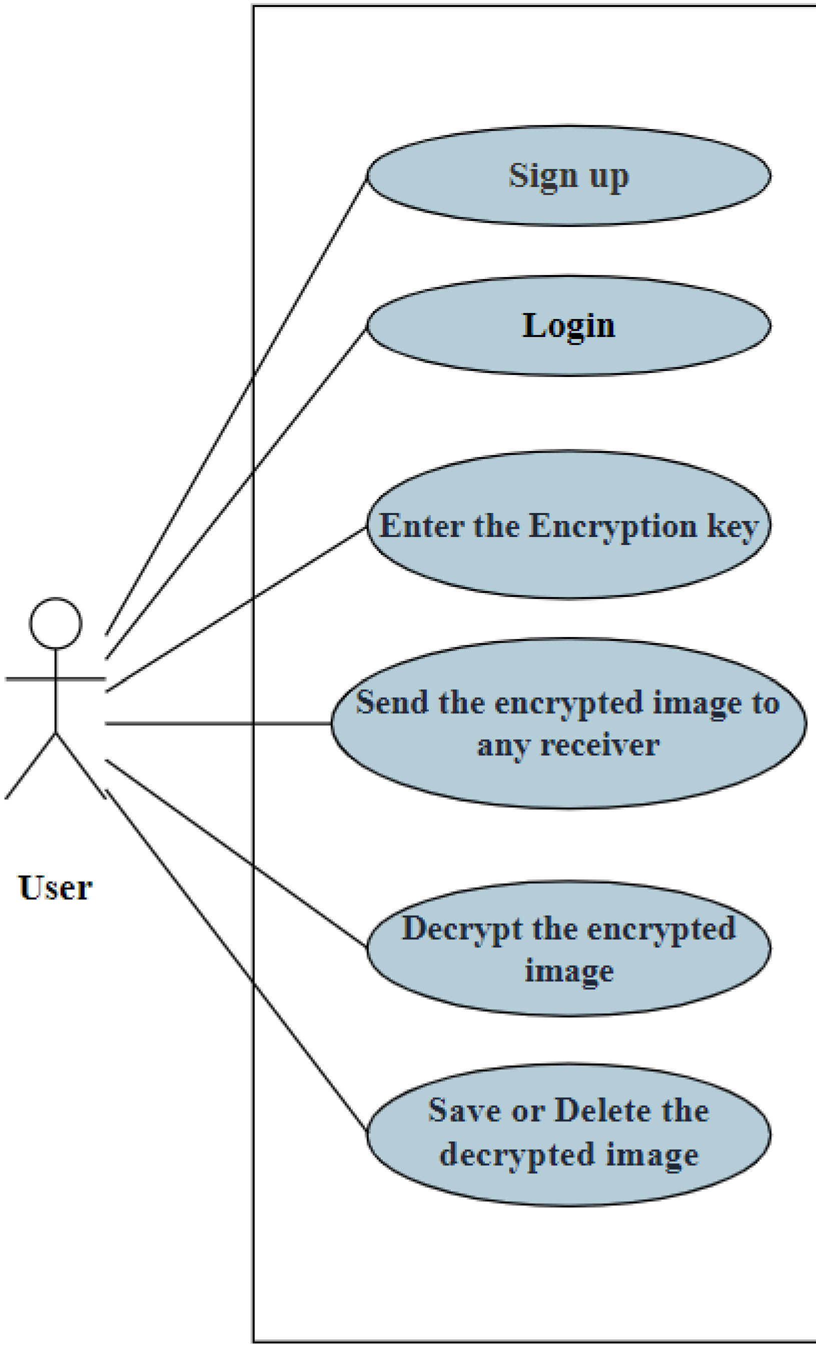
Table 6: Explain how the user can send the encrypted image to any receiver

Requirement Name	**Decrypt the Encrypted image**
Actor	User
Description	The user has the ability to decrypt the encrypted image.
Inputs	Encryption key
Outputs	Decrypted image
Procedures	- Open the application connected to the Internet. - Log in. - Open “Inbox”. - Enter the Encryption key to decrypt the image.
Exceptions	- Internet disconnection. - The Encryption key is less than 16 digits. - The entered Encryption key is wrong.

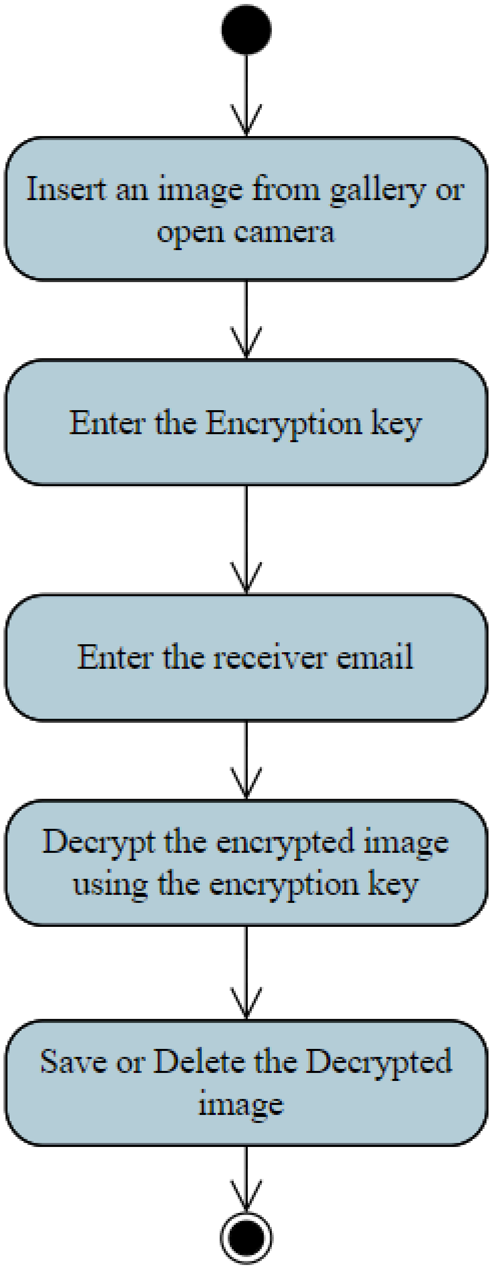
Requirement Name	**Save or Delete theDecrypted image**
Actor	User
Description	The user has the ability to save or delete the decrypted image.
Inputs	Encryption key
Outputs	Image.
Procedures	- Open the application connected to the Internet. - Log in. - Open “Inbox”. - Enter the Encryption key to decrypt the image. - Save or delete the decrypted image
Exceptions	Internet disconnection.

2.4. Data Analysis

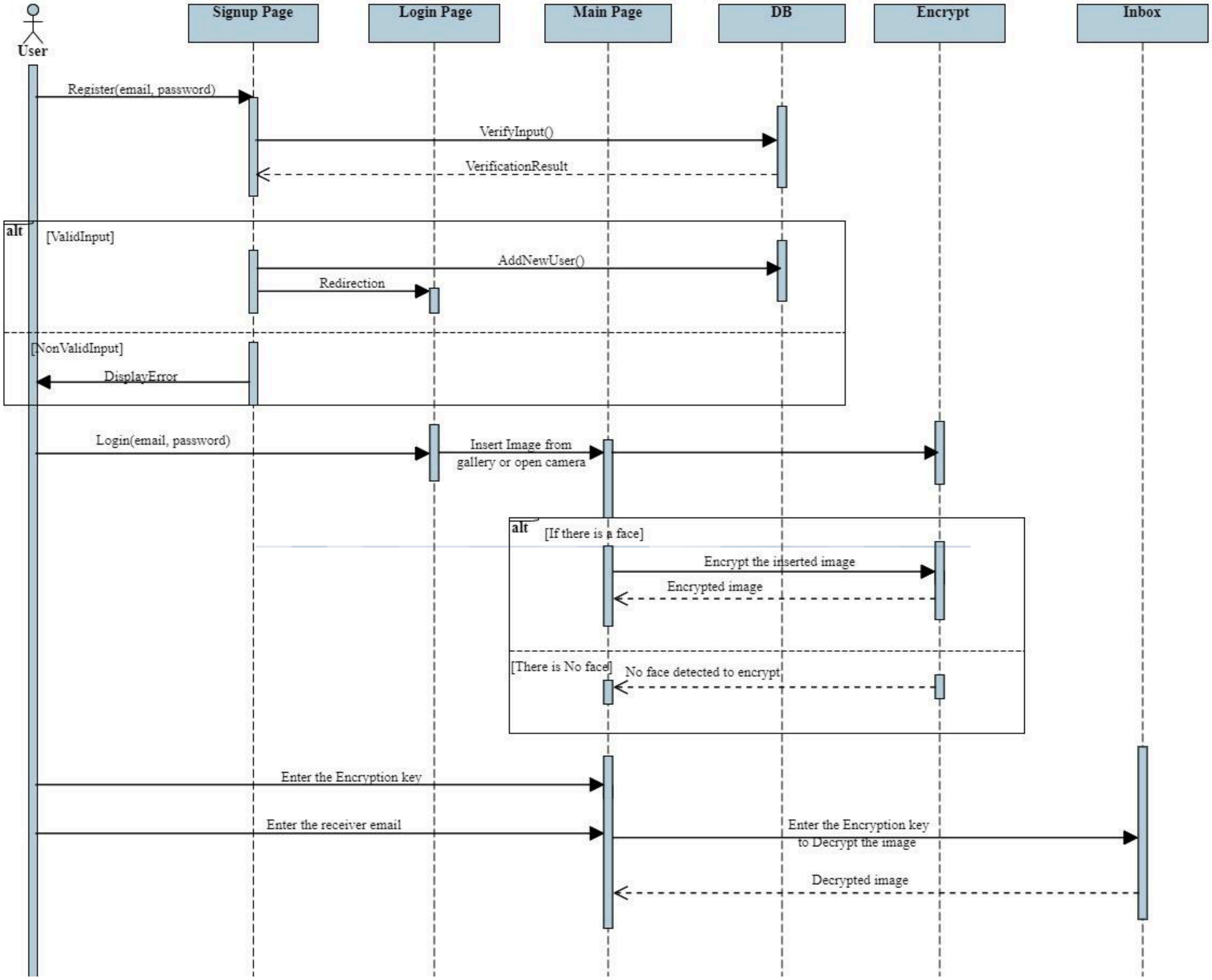
Use Case Diagram


Sign up
Login
Enter the Encryption key
Send the encrypted image to any receiver
User
Decrypt the encrypted image
Save or Delete the decrypted image

State Diagram


Insert an image from gallery or open camera
Enter the Encryption key
Enter the receiver email
Decrypt the encrypted image using the encryption key
Save or Delete the Decrypted image

Sequence Diagram



Chapter 3: Software Design

3.1 Introduction

In this chapter, we will talk about the application design, mobile application architecture, the database description, the database table's description, as well as the design of the application.

3.2 Design Decision

3.2.1 Architectural pattern

An architectural pattern is a description of a good design practice that has been tried and tested in different environments like MVC pattern, layered pattern, and client-server.

We chose this pattern because it is the most suitable one for the application. The MVC design pattern serves to separate the presentation layer from the business logic, and makes model classes reusable without modification.

1. mediator between every process between the user and the system.

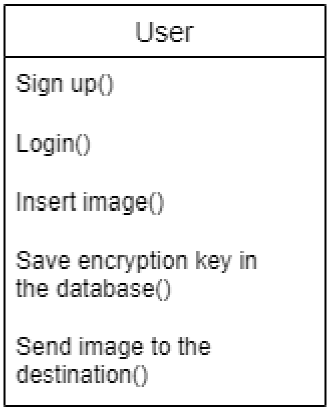

User
Sign up()
Login()
Insert image()
Save encryption key in the database()
Send image to the destination()

Figure 8: Controller

3.2.2 Architecture for mobile application

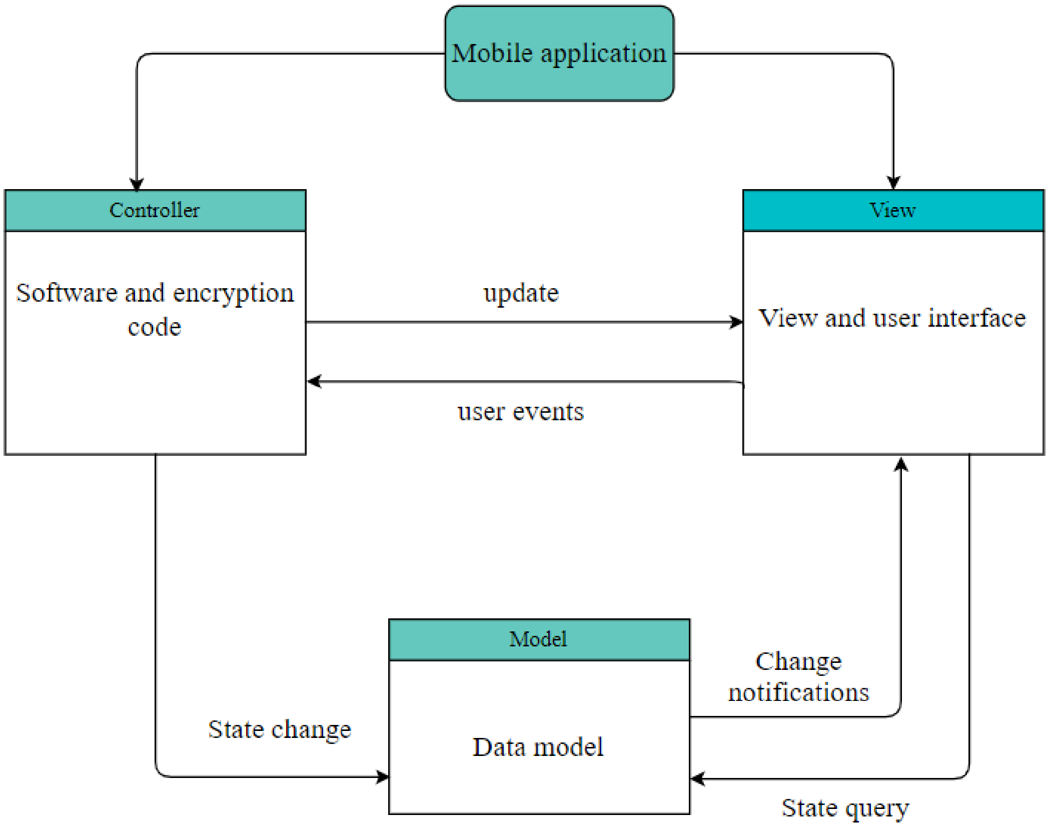


Figure 9: Architecture for the mobile application

3.3 Architecture

1. **Model:** It works to manage data and operations related to the application databases, this is the normalized database:

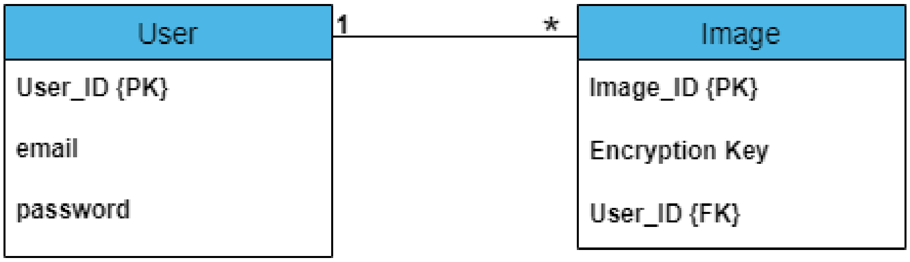


Figure 10: Database Mapping

2. **Description of the database(normalization):** the following figure shows the tables that will be created.

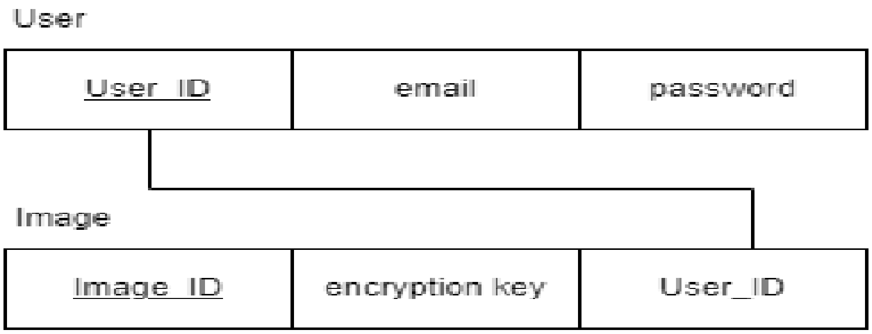


Figure 11: Normalized tables

3. **Database design:**

   The system is connected to a database that consists of two tables connected by a relationship.

A list of the tables that will be added to the database is provided below.

The name of the table in the database	Table description
User	Store data related to users
Image	Store data related to images that user add and key request

Table 9: Database table

Database Tables Description:

Field name	Data type	Size	Null	Unique	Description
user_ID	int	10	No	Yes	PK
Email	varchar	100	No	Yes	
password	varchar	100	No	No	

Table 10: User table

Field name	Data type	Size	Null	Unique	Description
image_ID	int	10	No	Yes	PK
key	varchar	100	No	Yes	
user_ID	int	10	No	Yes	FK

Table 11: Image table

3.4 Application Interfaces:

1. The main interface

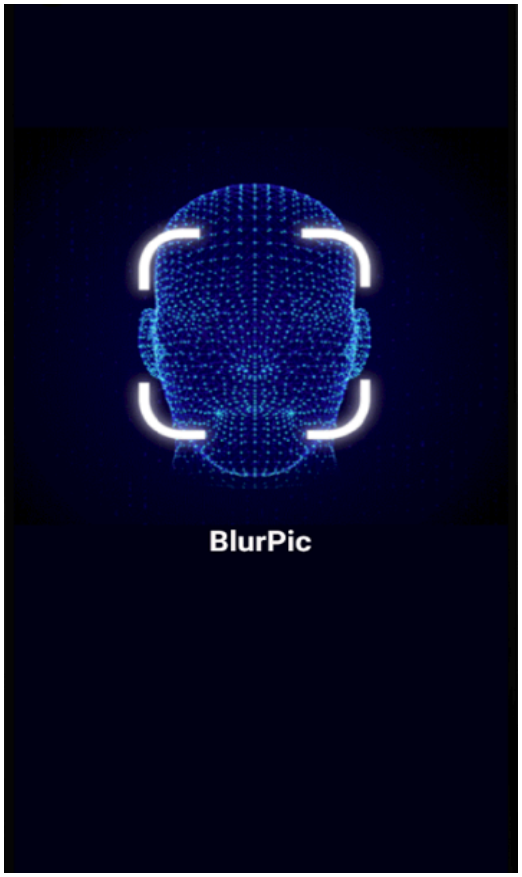


Figure 12: Splash screen

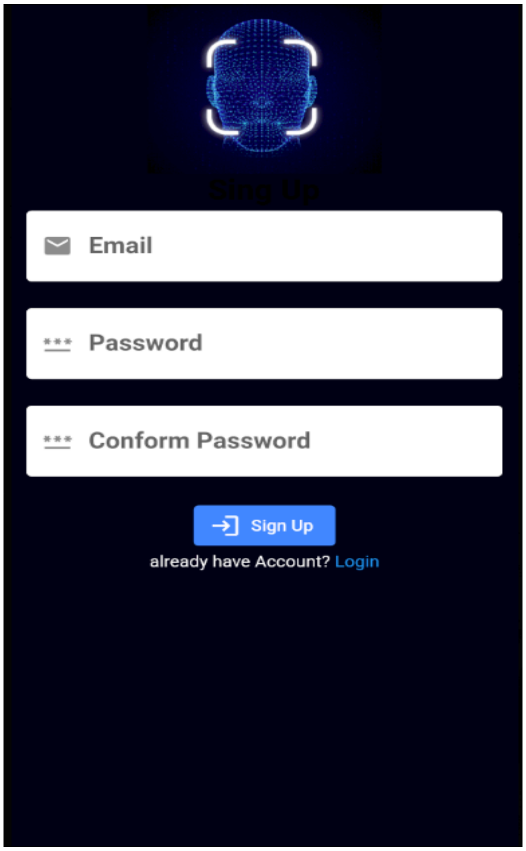


Figure 13: User’s Sign Up Page

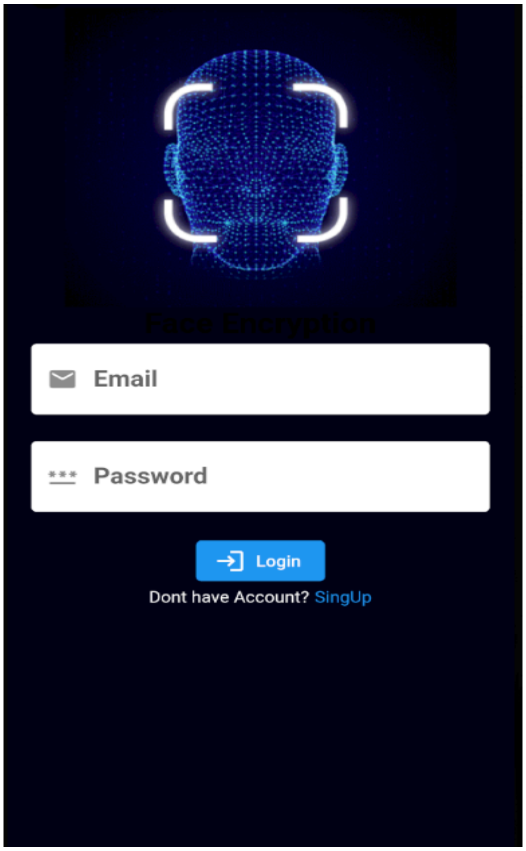


Figure 14: User’s login Page

2. The interface of inserting image

- You can snap a picture from the camera or select the image to be encrypted from the gallery using this interface.

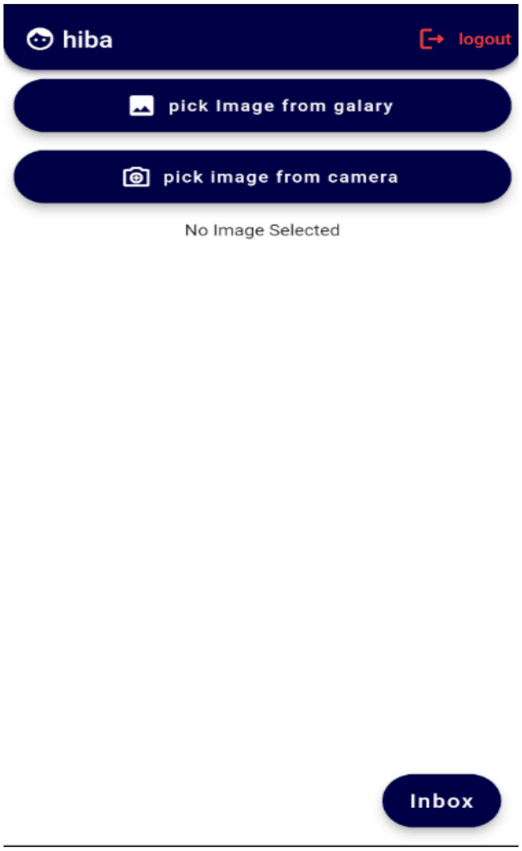


Figure 15: Insert image screen

3. When you want to encrypt faces in image

- **The first step:** type the encryption key of the image, which consists of 16 digits

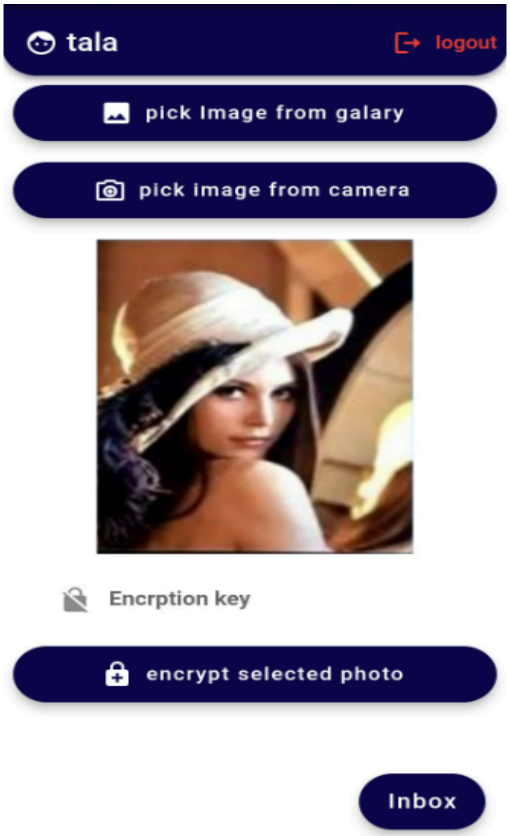


Figure 16: insert encryption key page

- **The second step:** type the recipient's email address here

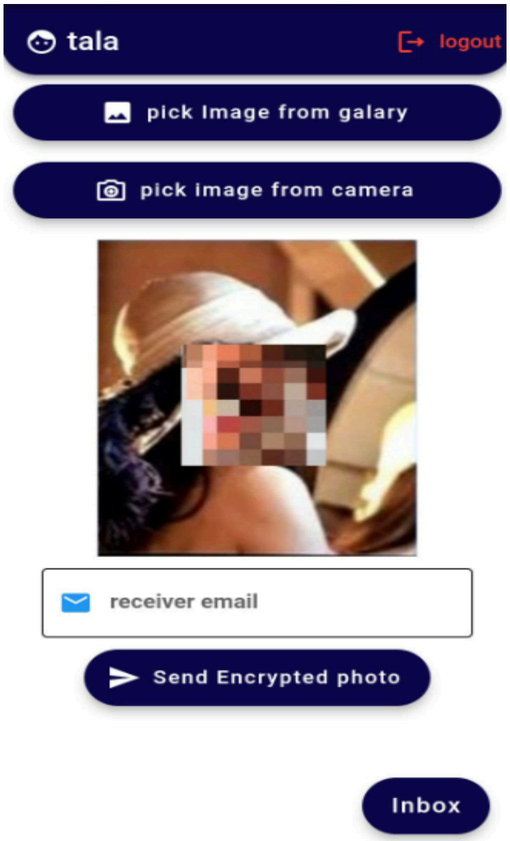


Figure 17: insert email page

4. Save and send encrypted images

- The email addresses of the recipients of your encrypted image transmissions as well as the senders of your encrypted image transmissions are displayed on this page.

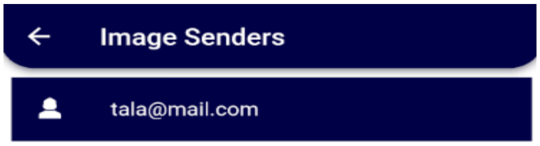


Image Senders

tala@mail.com

Figure 18: image senders page

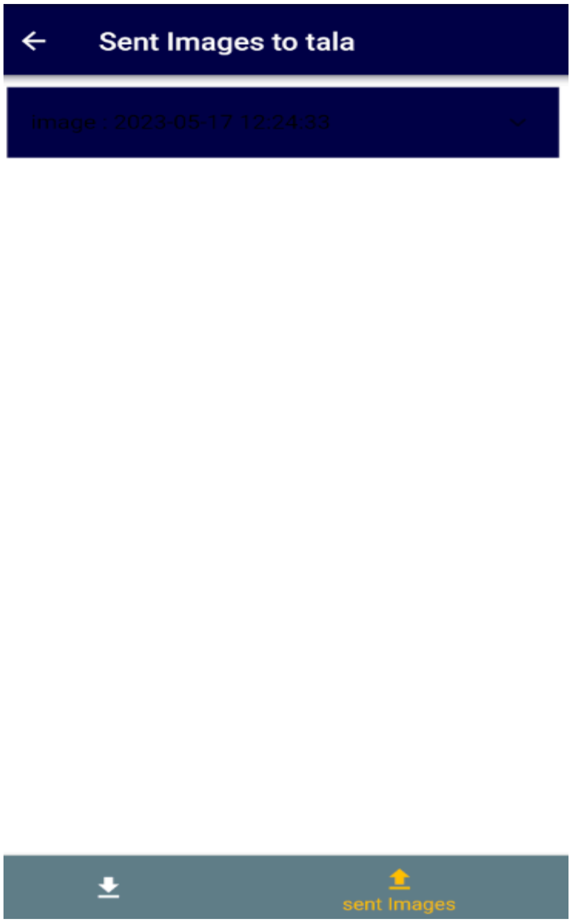
- Encrypted images obtained from another user and shown on this page after being decrypted using the specified encryption key

Recived Images from tala
Recieved Images

image : 2023-05-17 12:24:33

Figure 18: image senders page

- The encrypted images that have been encrypted and sent are stored on this page and the user can decrypt them using the unique encryption key that he entered



Sent Images to tala

image : 2023-05-17 12:24:33

Figure 18: image senders page

Chapter 4

Software Implementation

4.1 Introduction

In this chapter, we will discuss how the system was built. The system implementation stage is one of the most important stages, through which the transition from the theoretical stage, which is the preparatory stage of the system to the practical stage, and then start programming and building the system. We will learn about the tools and programs necessary to develop the system and operate it fully and effectively, and the software that was used.

4.2 System Software Resources

- **Visual studio code:** it is a source code editor developed by Microsoft for Windows Linux and macOS. It includes support for debugging, embedded Git control and GitHub, syntax.

- **Flutter:** Its an open-source UI software development kit developed by google facilities create cross-platform applications for Android, iOS, Linux, macOS, Windows, Google Fuchsia, and the web from a single codebase, The reason Flutter was chosen is that it uses the Dart programming language (Widget Base), which streamlines and organizes the application administration process.

- **Firebase:** Backend-as-a-Service (BaaS) app development platform called firebaseIt is instant By enabling secure access to the database directly from client-side code, database enables you to create rich, collaborative apps. It is a cloud-hosted NoSQL database that enables you to store and sync data between your users in real-time.

- **ML Kit:** ML Kit is a mobile SDK that brings Google's machine learning expertise to Android and iOS apps in a powerful yet easy-to-use package.

It allows new or experienced machine learning to implement the functionality in just a few lines of code, so there's no need to have deep knowledge of neural networks or model optimization to get started.

ML kit provides many capabilities including:

1- Recognize and locate facial features: Get the coordinates of the eyes, ears, cheeks, nose, and mouth of every face detected.

2-Recognize facial expressions Determine whether a person is smiling or has their eyes closed.

3- Human face detection, because ML Kit can perform face detection in real-time, it's possible to use it in applications like video chat or games.[8]

With ML Kit's face detection API, it's easy to detect faces in an image, identify key facial features, and get the contours of detected faces but not recognize people. [9]

4.3 Software Implementation

The software of the system has been developed mainly in three stages:

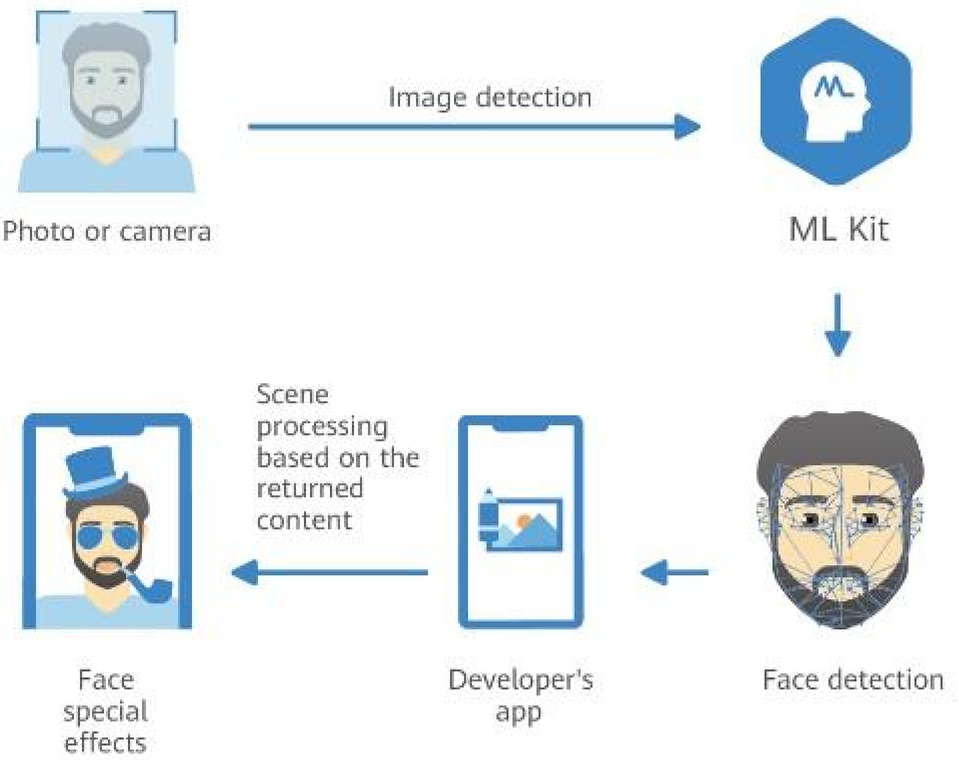
1. Human faces detection using ML KIT tool: this stage goes through several steps to generate the function of face detection.
2. Faces Encryption using AES Algorithm: this stage describes the encryption steps from transforming image pixels to text.
3. Sending the partially encrypted image to other users via email: this stage will let users see the encrypted image with the availability to decrypt it in case they have the encryption key.

Here’s more details about each stage:

1- Human faces detection using ML KIT

We suggest using ML Kit to accurately detect faces, where input images must contain faces that are represented by sufficient pixel data. In general, each face want to detect in an image should be at least 100x100 pixels. [6]

Face detection in an image starts by creating an InputImage object from either a Bitmap or media.Image such as when you capture an image from a device's camera or a file on the device. Then, pass the InputImage object to the FaceDetector's process method.[6].



Picture of face detection using ML KIT

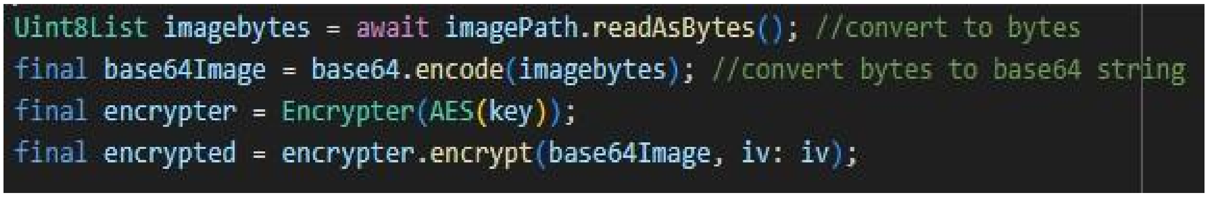
2-Face encryption using AES Algorithm

AES takes a plain text in blocks of 128 bits and converts them to ciphertext using keys of 128, 192, and 256 bits. As long As our project is to deal with digital images, we used a base64 algorithm to generate the base64 string from the image, so we can use the encryption algorithm seamlessly on the mobile environment without the need for servers.

So, The idea behind using base64 is that we can encode binary data into text and then decode that text data to a byte array.

To implement the string encrypting process we follow these steps:

- Encode the string to binary data, e.g. using UTF-8 (text.getBytes("UTF-8"))
- Encrypt the binary data using AES
- Encode the cyphertext using Base64 to get the text.



```
Uint8List imagebytes = await imagePath.readAsBytes(); //convert to bytes
final base64Image = base64.encode(imagebytes); //convert bytes to base64 string
final encrypter = Encrypter(AES(key));
final encrypted = encrypter.encrypt(base64Image, iv: iv);
```

3- Send the encrypted image and decrypt it.

This stage is used to allow all users to send the partially encrypted image to other users who have accounts on the same application firebase where we create and establish the connection.

The application forces users to sign up to get benefits from the provided encryption services with the availability to download it on their device or send it to other parties who already registered using the same firebase. So, it's not acceptable to send the encrypted face to your Gmail, Yahoo, or Outlook accounts.

Once the other intended parties receive the image, they will have it as an encrypted version, not the original one, So, in order to have the actual content of the image, the application asks to insert the decryption key contains of 16 characters which is the same encryption key that shared by the owner of the actual image.

4.4 Mobile Application Implementation

We developed an Android application that is implemented using the Flutter framework as it is an excellent tool to implement mobile applications for either Android or IOS. We also link Firebase databases in real time to the application. In the Android application, we used many plugins and already built classes to support achieving the goals of project:

1- image_picker: A Flutter plugin for iOS and Android for picking images from the image library, and taking new pictures with the camera. [7]

2- google_mlkit_face_detection: A Flutter plugin to detect faces in an image, identify key facial features, and get the contours of detected faces. [7]

3- Cloud_firestore: Flutter plugin for Cloud Firestore, a cloud-hosted, NoSQL database with live synchronization and offline support on Android and iOS. [7]

4- Encrypt package: A set of high-level APIs over PointyCastle for two-way cryptography. [7]

5- Firebase_auth: a Flutter plugin for enabling Android and iOS authentication using passwords and phone numbers.

6- Firebase_core: a Flutter plugin for enabling connecting to multiple Firebase apps.

7- FaceDetector: a flutter class that returns a list of face classes which contains the rect coordinates. Rect holds four integer coordinates for a rectangle. The rectangle is represented by the coordinates of its 4 edges (left, top, right bottom).

8- CustomPainter: a flutter class that uses the rect coordinates to draw a rectangular box around the faces.

Chapter 5:

Software Testing

5.1 Introduction

In the chapter of testing the system, we make sure that the system works correctly without any problems and discuss the unit testing for backend APIs and functional requirements testing, to ensure that our flutter app is working as expected and helps manage new changes in specification or implementation and that the system works with accuracy and high speed in completing tasks and displaying information, the stage of testing comes after the design and implementation of the system.

5.2 Validation

Every piece of data submitted into every field in the application is checked to see if it satisfies the following criteria:

- Adjust the field so that it corresponds to the entry.
- If incorrect data is entered, the process won't run.
- If the encryption key is not entered, the operation will not be carried out.
- If the decryption key doesn't match the encryption key, the decryption operation will fail.
- If the user enters an image without a face, the encryption process will not be carried out.
- If the user blurs a face in an image, it's not possible to re-encrypt the same image again.

5.3 Mobile Application Testing:

1- Application interfaces:

We ensured all text fields worked well by allowing users to insert the right data and alert them when entering a wrong formula or already existing account and direct them to the right solutions.

In addition to that, we paid attention to afford straightforward actions, easy clicks, and attractive buttons to streamline the decryption process.

2. Testing Firebase:

We checked that the connection between the Flutter application and Firebase is established successfully, where new users can create a new account and log in to the application easily.

5.4 Functional Requirements Testing

The system components are tested to assure their functionality. The examination's outcome was successful. The tests we conducted are summarized in the following tables:

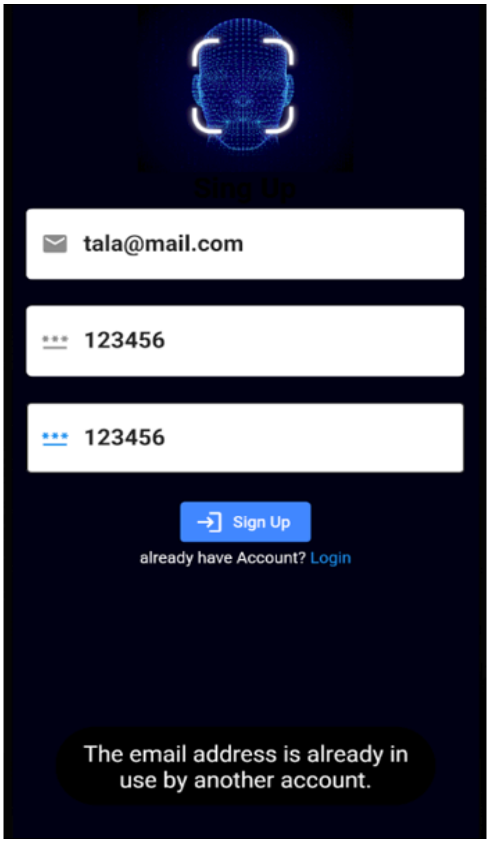
Test Case	Scenario	Input data	Expected Output	Actual Output	Test result
Sign up into the BlurPic application	user enters his email and password to create account	email password confirm password	create account successfully in firebase	create account successfully in firebase	pass
	user enters an email that already existed in firebase	email password confirm password	the email address is already in use by another account	the email address is already in use by another account	pass

Login into the account that has been created	user enters his email and password correctly	email password	user logged in his account successfully	user logged in his account successfully	pass
	user enters his email and password wrong	email password	there is no user record corresponding to this identifier- the password is invalid or the user does not have account- email cannot be empty- password cannot be empty	there is no user record corresponding to this identifier- the password is invalid or the user does not have account- email cannot be empty- password cannot be empty	pass
Choose an image to be encrypted	user enters a picture containing faces either from the gallery or takes a picture with the camera	image	Enter the image's encryption key	Enter the image's encryption key	pass
	The user inserts a photo that does not contain faces either from the gallery or takes a photo with the camera	image	No faces detected	No faces detected	pass
Enter the encryption key	user enters the encryption key, which consists of 16	encryption key	user encrypted his image successfully	user encrypted his image successfully	pass

	digits				
Enter the receiver email	user enters the receiver email	receiver email	The encrypted image was sent successfully	The encrypted image was sent successfully	pass
Decrypt the image by the receiver	receiver enters the encryption key of the image that the user entered when encrypting it	decryption key	The image has been successfully decrypted	The image has been successfully decrypted	pass
Delete the image	user can delete the image from the list	-	image deleted successfully	image deleted successfully	pass

5.4.1 API testing:

- **SignUp validation:** To make an account, the user must enter his information there is no way to establish an account with an empty field. The user must use a singular email address.



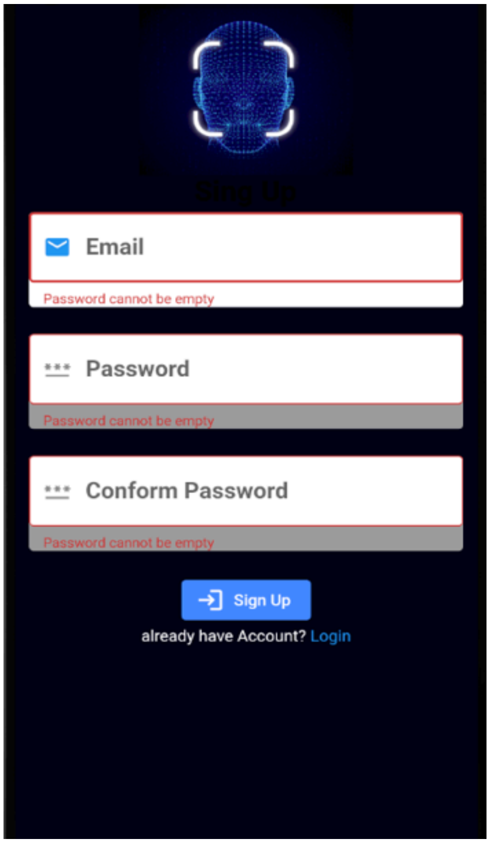
tala@mail.com

123456

123456

The email address is already in use by another account.

Figure 20: SignUp(invalid email)



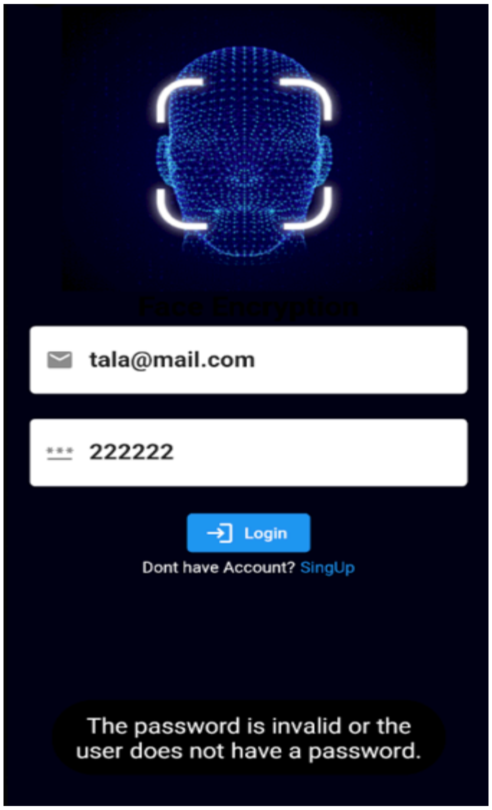
Password cannot be empty

Password cannot be empty

Password cannot be empty

Figure 21:SignUp(required fields)

- **Login validation:** user must provide login details in order to access the application. The email and password must match the email and password used to create the account. All fields must be filled in

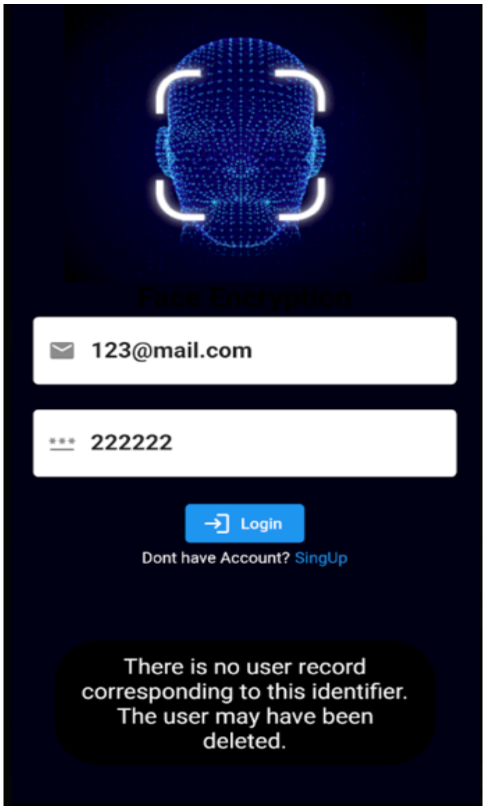


tala@mail.com

222222

The password is invalid or the user does not have a password.

Figure 22: Login(invalid password)

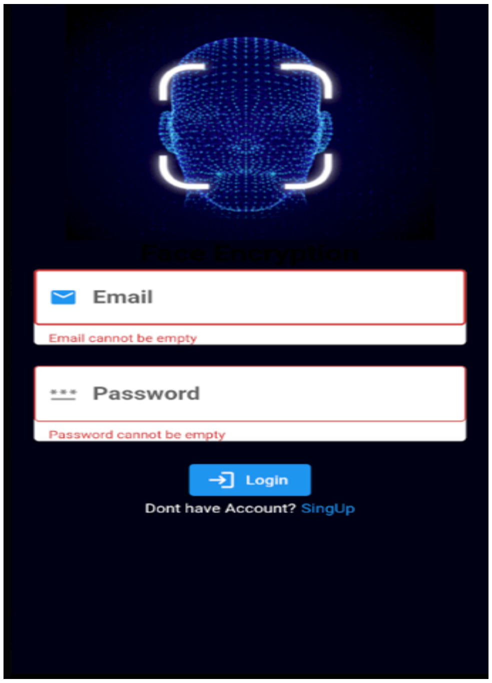


123@mail.com

222222

There is no user record corresponding to this identifier. The user may have been deleted.

Figure 23: Login(invalid email)

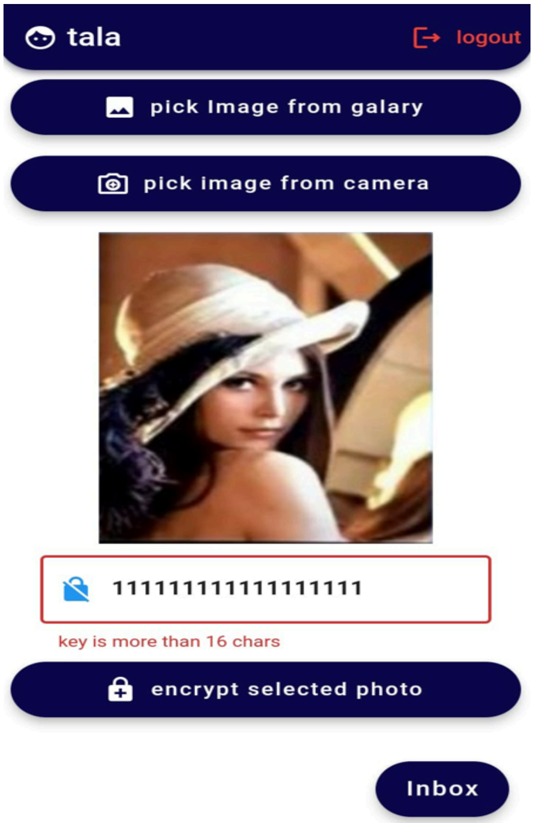


Email cannot be empty

Password cannot be empty

Figure 24: Login(required fields)

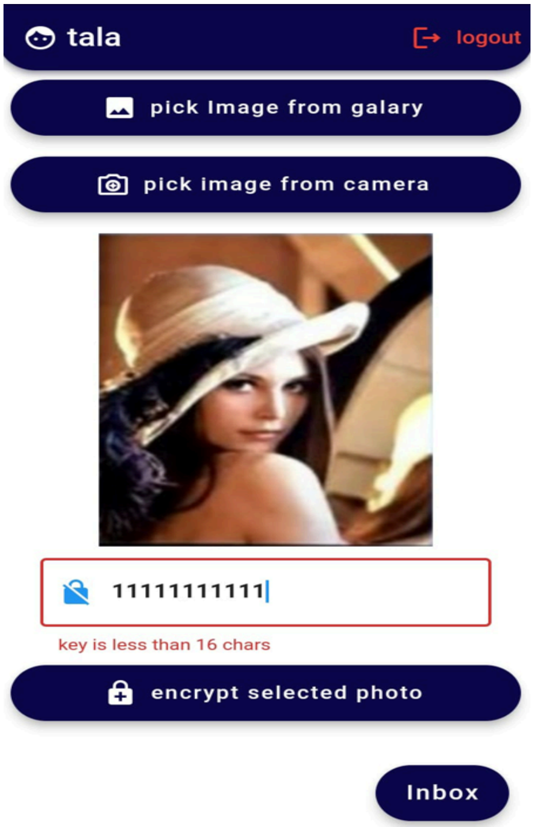
- **Encryption key validation:** The AES algorithm dictates that the encryption key must have 16 digits, hence the system forbids any value greater or less than the minimum needed to encrypt the image.



111111111111111111

key is more than 16 chars

Figure 25:encryption key(invalid key)

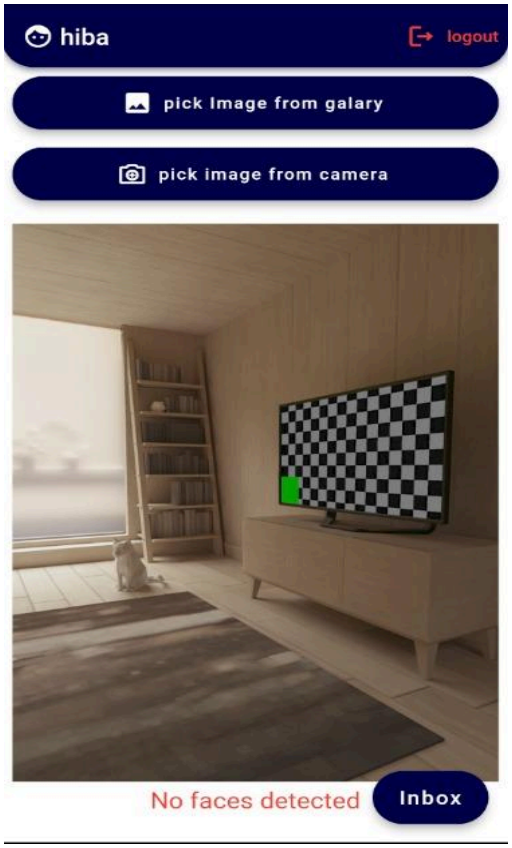


11111111111

key is less than 16 chars

Figure 26:encryption key(invalid key)

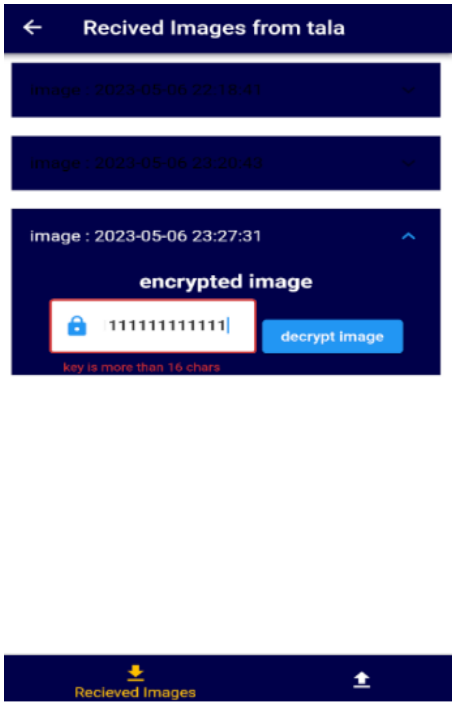
- **face detection validation:** The MLkit library is used to detect the faces in the image when it is input into the system. The system will notify you that the image doesn't contain a face if no faces are found in it.



No faces detected

Figure 27: face detection(face does not exist)

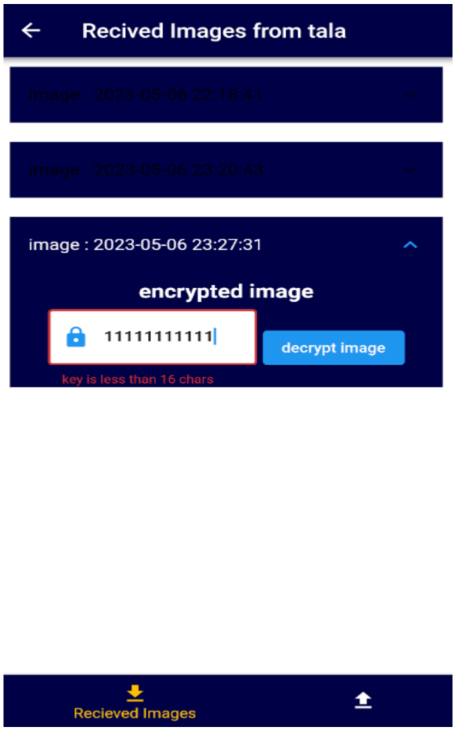
- **decryption key validation:** The user receives ownership of the image after they are given the encryption key with which the image was encrypted, allowing them to view the content of the image.


Recived Images from tala
image : 2023-05-06 22:18:41
image : 2023-05-06 23:20:43
image : 2023-05-06 23:27:31
encrypted image
decrypt image
Recieved Images

111111111111

key is more than 16 chars

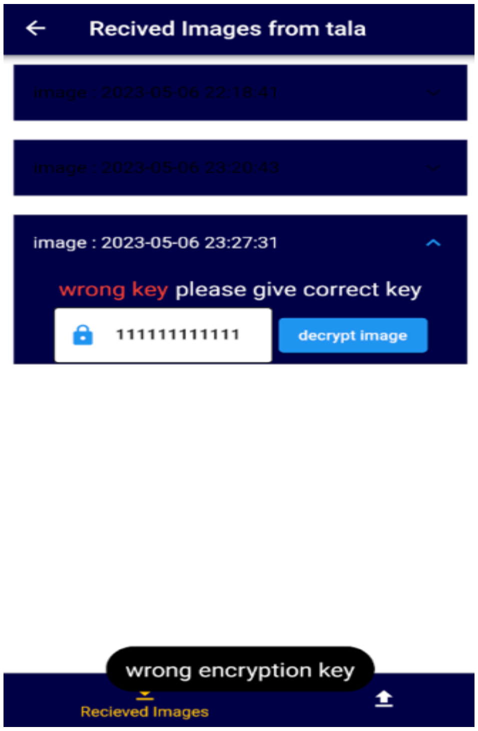
Figure 28:decryption key(invalid key)


Recived Images from tala
image : 2023-05-06 22:18:41
image : 2023-05-06 23:20:43
image : 2023-05-06 23:27:31
encrypted image
decrypt image
Recieved Images

11111111111

key is less than 16 chars

Figure 29:decryption key(invalid key)


Recived Images from tala
image : 2023-05-06 22:18:41
image : 2023-05-06 23:20:43
image : 2023-05-06 23:27:31
decrypt image
Recieved Images

wrong key please give correct key

111111111111

wrong encryption key

Figure 30:decryption key(wrong key)

Chapter 6

Conclusion and future recommendations

6.1 Conclusion

After our hard work on this project, the application successfully achieves the main reasons behind building it, where the application allows users who aim to online blur or hide a face or many faces to create an account, pick an image from a gallery, or catch one from the camera, and then encrypt them using with a special key, send them to others with an option to decrypt them just in case having the decryption key from the image owner.

6.2 Future work

As a project team sees that the following points may be a good improvement in the future to make more reliable and effective real-life applications:

- Detect and encrypt car plates, places and sensitive texts like credit cards.
- Providing an option for users to non-detect specific faces in images to avoid encrypting them.
- Image labeling by Identifying objects, locations, animal species, products, and more.
- Recognize the faces of people wanted by the law, and automatically send them to the police body.

Reference Page:

1. A.Kumar, A. Kor, "Face detection techniques" 2019, [Online; accessed 14-May-2023].

2. R.C.Gonzalez and R.E.Woods, Digital Image Processing, 3rd ed. 2002.

3. X.Zhang, K.Parhi, "Implementation approaches for the Advanced Encryption Standard algorithm", 2002.

4. K.Mahanta, An Enhanced Advanced Encryption Standard Algorithm, 2015, p 1-2, IEEE

5.Q. Zhang , H.Xiao, "Extracting Regions of Interest in Biomedical Images", 2008, International Seminar on Future BioMedical Information Engineering.

6. ML KIT Android, "Detect faces with ML Kit on Android", https://developers.google.com/ml-kit/vision/face-detection/android [Online; accessed 14-May-2023].

[7] Flutter packages, https://pub.dev/packages [Online; accessed 18-May-2023].

8.Machine learning for mobile applications, https://developers.google.com/ml-kit. [Online; accessed 18-May-2023]

9. ML kit face detection, https://developers.google.com/ml-kit/vision/face-detection, [Online; accessed 18-May-2023].