FACE RECOGNITION FOR SECURE ONLINE PAYMENT WITH PROXY DETECTION

A PROJECT REPORT

Submitted by

SARATH KUMAR S

SANTOSH LUITEL

SANTHOSH S

SANTHOSH S

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

PSNA COLLEGE OF ENGINEERING AND TECHNOLOGY,
DINDIGUL

ANNA UNIVERSITY: CHENNAI 600 025

MARCH 2021

BONAFIDE CERTIFICATE

Certified that this project report “**FACE RECOGNITION FOR SECURE ONLINE PAYMENT WITH PROXY DETECTION”** is the bonafide work of "**SARATH KUMAR S (921317104192), SANTOSH LUITEL (921317104191), SANTHOSH S (921317104186), SANTHOSH S (921317104187)"** who carried out the project under my supervision.

SIGNATURE	**SIGNATURE**
Dr. D. SHANTHI M.E., Ph.D.,	**Dr. R. KARTHIKEYAN**
HEAD OF THE DEPARTMENT	**SUPERVISOR**
PROFESSOR	**PROFESSOR**
COMPUTER SCIENCE AND	COMPUTER SCIENCE AND
ENGINEERING	ENGINEERING
PSNA COLLEGE OF	PSNA COLLEGE OF
ENGINEERING AND	ENGINEERING AND
TECHNOLOGY	TECHNOLOGY
DINDIGUL-624622	DINDIGUL-624622

Submitted for Viva-voce examination on....................................... 2021

INTERNAL EXAMINER	**EXTERNAL EXAMINER**

ABSTRACT

The purpose of our project is to develop a web based application which is used for online payment in a more secure way. Instead of OTP generation the online payment is enabled by face recognition of the user.

Online Secure Payment using Face Recognition is a web based application which enables a more secure online money transaction. The advancements in technology have led to a surge in online transactions via online shopping, internet banking, payment gateways, etc.

In the current world of advanced technology, it is easy for hackers to get personal details of users because of which some people are hesitant to use online transactions. This makes security an important factor at the time of making digital payments. For this reason, to make transaction more secure we use face recognition instead of OTP or MPIN. With face recognition, there are no passwords that hackers could compromise. Even if hackers stole your photo database, it would be of little use, as "liveness detection," prevent using them for impersonation purposes.

In Face recognition system, when a request is generated for transaction, the details of the payee are verified. If the details are legitimate i.e. payee is legitimate, then the facial details of payee are collected and simultaneously it is collected by the bank database and compared with each other. Positive response of comparison lead to successful transaction processing. Whereas, negative response will lead to termination of the transaction.

Facial recognition has improved dramatically in recent years and the best face identification algorithm has an error of just 0.08% making it more reliable and secure than OTP.

Keywords: Online Payment, Internet Banking, Liveness Detection.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

S.NO	ACRONYMS	ABBREVIATIONS
1	**ML**	**M**achine **L**earning
2	**MTCNN**	**M**ulti **T**ask **C**ascaded **C**onvolutional **N**eural **N**etwork
3	**SVM**	**S**upport **V**ector **M**achine
4	**ERD**	**E**ntity **R**elationship **D**iagram
5	**UML**	**U**nified **M**odelling **L**anguage
6	**DB**	**D**atabase

CHAPTER 1
INTRODUCTION

Face Recognition for Secure Online payment using proxy detection concerns on developing a Secured money transaction using face recognition thereby preventing prevailing hacking mechanism by hackers.

1.1 OVERVIEW OF PROJECT

The purpose of our project is to develop a web-based application which is used for online payment in a more secure way. Instead of OTP generation the online payment is enabled by face recognition of the user, the advancements in technology have led to a surge in online transactions via online shopping, internet banking, payment gateways, etc.

In the current world of advanced technology, it is easy for hackers to get personal details of users because of which some people are hesitant to use online transactions thus by making much more secured way oftransaction.

Since information of users can leak easily, many authentication approaches are proposed to secure data and systems. Authentication method should be able to “declare anyone that they are who they claim to be” and Moreover, they can be disclosed and stolen by intimates or colleagues. Some systems use an object such as passport or ID card to identify a user. It is practical in use but also is very easy to be stolen or copied.

1.2 AIM

This web application will get the account holders face by getting registered and transacting the money from the sender to recipient only by detecting and processing the owners original face thus providing much security than the prevailing methods.

Ensuring the safest transaction by the respective bank account holder as no proxy can be used to make unauthorized transaction and proving only secured and valid transaction.

Making the transaction more reliable than the other modes of transaction that ensures secure and uninterrupted transaction.

1.3 OBJECTIVE

- The main objective of the application is to complete online pay using face recognition which has secure payment. Proxy Detector was included in the system.
- Face recognition and identification using deep learning Face net much more reliable as it makes the user aware that it is much more secure to do transaction.
- It renders the access to the user at any time access with no requirement of OTP’s that could be hacked and thereby rendering complete safety over the transaction.
- The transaction permits in this study lie in the biometric face verification transaction permission and Mobile number to improve communication security and transaction data.
- Based on that cause the bank to need another solution for low-cost infrastructure. Technology developments and human dependence on technology and as well as technology have penetrated into various fields, such as industry, manufacturing, education, government, business, banking, and human daily life.
- To enhance the use of technology in banking sectors thus empowering the money transaction that are much more easy to access and less expensive.

CHAPTER 2
SYSTEM ANALYSIS

Systems are created to solve problems. One can think of the systems approach as an organized way of dealing with a problem. In this dynamic world, the subject system analysis and design, mainly deals with the software development activities. A collection of components that work together to realize some objectives forms a system. Basically, there are three major components in every system namely input, processing and output.

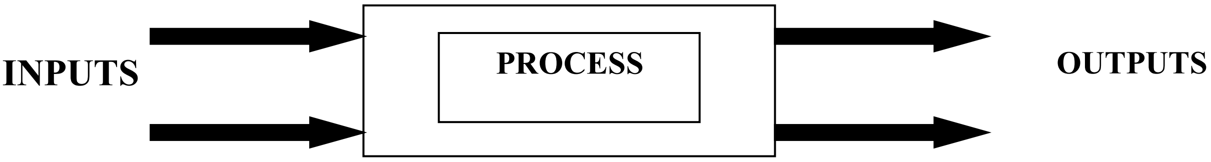


Fig 2.1 Components of the system

In a system, the different components are connected with each other and they are interdependent. The objective of the system demands that some output is produced as a result of processing the suitable inputs. A well-designed system also includes an additional element referred to as ‘control’ that provides a feedback to achieve desired objectives of the system. System Analysis is a process of collecting factual data, understand the process involved, identifying problems and recommending feasible suggestions for improving the system functioning. This involves studying the business process, gathering operational data, understand the information flow, finding out bottlenecks, and evolving solutions for overcoming the weakness of the system so as to achieve the organizational goals. System Analysis also includes subdividing of complex process involving entire system, identification of data store and manual process. The major objectives of system analysis are to find answers for each business process. What is being done, How is it being done, Who is doing it, When is he doing it, Why is it being done and How can it be improved? It is more of a thinking process and involves the creative skills of the system analyst.

It attempts to give birth to a new efficient system that satisfies the current needs of the user and has scope for future growth within the organizational constraints. The result of this process is a logical system design. System analysis is an iterative process that continues until a preferred and acceptable solution emerges.

2.1 SYSTEM LIFE CYCLE

System life cycle is an organizational process of developing and maintaining system. It helps in establishing system project plan, because it gives overall list of process and sub-process required for developing a system. System development life cycle means combination of various activities In the system Analysis and Design terminology, the system development life cycle also means software development life cycle. The different phases of system development life cycle is shown in this diagram.

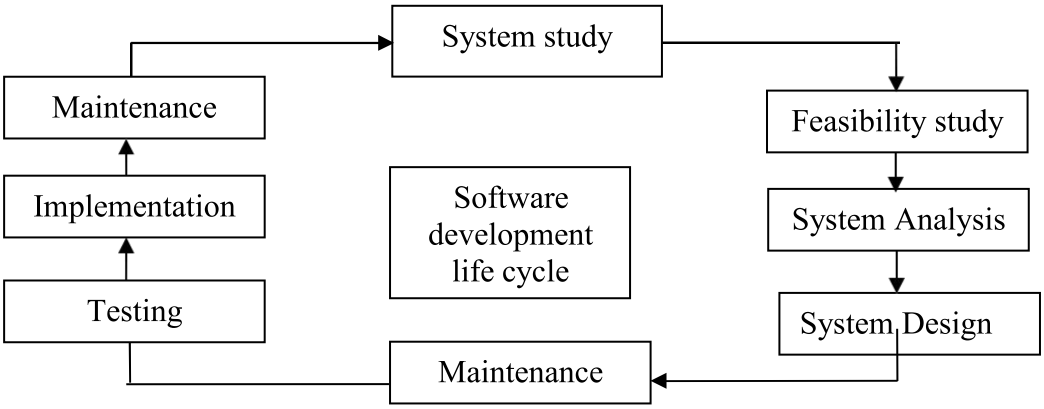


Fig 2.2 System life cycle

2.2 EXISTING SYSTEM

A simple multi-factor authentication setup involves asking a user for their username and password (something they know) as well as verifying their identity through a second factor such as an SMS message to their phone (something they have). That covers two factors of authentication, but adding in image recognition as well adds an extra layer of security to the login process without making it frustrating or overly complicated for authorized users.

Existing system was implemented using face encryption and decryption to avoid hackers while transaction. Existing System used Azure API face verification. This detection will provide the exact location of where faces appear in images, it will also provide meta data about the faces that it detects, such as age estimates, gender, and numerous face attributes such as facial hair.

Limitation of existing systems are proxy detectors was not implemented, even if people show their photo image it could able to recognize the face and completes the transaction. This leads to major thread of the process.

2.2.1 DRAWBACKS OF EXISTING SYSTEMS

1. In low lit places the face detection may not be accurate and may not run properly.
2. This system does not calculate pixel average value and thus does not identify the difference between live face and an image.
3. This system has larger code and complicated process thus wasting space and resources.

2.3 NEED FOR PROPOSED SYSTEM

1. In the proposed system, Face Verification will be implemented using Face Net Deep Learning Algorithm to embed with 128 feature points of the face. The FaceNet system can be used to extract high-quality features from faces, called face embeddings, that can then be used to train a face identification system.
2. Embed 128 features point of each person face will be calculated with triplet loss function. MTCNN is used for detecting the face captured and FaceNet is used to extract the features of the image.
3. An SVM based classifier is used in facial recognition to predict the similarity and dissimilarity between two images.
4. A Support vector machine (SVM) is a supervised machine learning model that uses classification algorithms for two-group classification problems. After giving an SVM model sets of labeled training data for each category, they're able to categorize new text.
5. Advantage of our proposed system is Proxy Detection will be implemented using image processing technique. This provided more secure to the system.

2.4 FEATURES

➢ In our proposed system for the payment process to complete successfully we can only use our live face. Images and photos cannot be used for verification process.

➢ Proxy Detector is included, if showing any photo in front the prediction system, it automatically detects the proxy and payment will not be completed**.**

- The pixel value average is calculated during the face verification which if exceeds a certain value the captured image is detected as a proxy image and the payment value is not completed.
- Any number of images can be used to train and test. The more the number of images given for training and testing the greater the accuracy of the detection and also the time complexity increases.
- No need for OTP/MPIN since we have face recognition.
- This allows the customers to carry out a more secure mode of payment.

2.5 SYSTEM REQUIREMENTS

System requirements are the configuration that a system must have in order for a hardware or software application to run smoothly and efficiently. These requirements are known as (computer) system requirement. Most software defines two sets of system requirements: minimum and recommended. With increasing demand for higher processing power and resources in newer versions of software, system requirement tend to increase over time. Industry analysts suggest that this trend plays a bigger part in driving upgrades to existing computer systems than technological advancements.

2.5.1 SOFTWARE REQUIRMENTS

➢ Operating System	: Windows
➢ Front End	: Django
➢ Back End	: csv files
➢ Tools	: Spyder

2.5.2 HARDWARE REQUIREMENTS

➢ Processor	: Dual Core
➢ Ram	: 4 GB
➢ Speed	: 2.40GHz
➢ Camera	: External or inbuilt webcam enabled
➢ Type	: 32 or 64 bit Operating System

2.6 FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS

2.6.1 Functional Requirement

Functional Requirements as “the product capabilities, or things that a product must do for its users”. Functional Requirement defines how software behaves to meet user needs. A Functional Requirement, that when satisfied, will allow the user to perform some kind of function.

2.6.2 Non-Functional Requirement

Non-Functional Requirements as “the quality attributes, design and implementation constraints, and external interfaces which a product must have”. Examples of quality attributes include availability, maintainability, performance, portability, reliability, robustness, security, scalability, testability, usability, and others. Many non-functional requirements are global in nature; they apply to an entire system.

2.7 LANGUAGE SPECIFICATION

A programming language specification (or standard or definition) is a documentation artifact that defines a programming language so that the users and implementers can agree on what programs in that language mean.

2.7.1 DJANGO

Django is a high-level Python web framework that enables rapid development of secure and maintainable websites. Built by experienced developers, Django takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. It is free and open source, has a thriving and active community, great documentation, and many options for free and paid-for support.

It is often pointed to as the main competitor with:

- Flask
- web2py
- Tornado
- CherryPy
- TurboGears

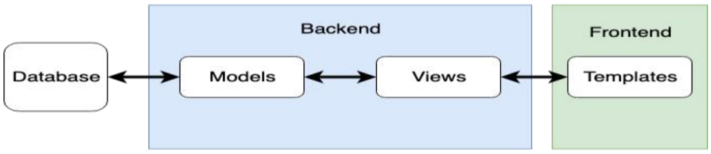


FIG 2.1 DJANGO architecture

Django provides a bridge between the data model and the database engine, and supports a large set of database systems including MySQL, Oracle, Postgres, etc. Django also supports NoSQL database through Django- nonrel fork. For now, the only NoSQL databases supported are MongoDB and google app engine.

Django provides a nice ready-to-use user interface for administrative activities. Django comes with a lightweight web server to facilitate end-to-end application development and testing.

2.7.2 Spyder

Spyder is a free and open source scientific environment written in Python, for Python, and designed by and for scientists, engineers and data analysts. It features a unique combination of the advanced editing, analysis, debugging, and profiling functionality of a comprehensive development tool with the data exploration, interactive execution, deep inspection, and beautiful visualization capabilities of a scientific package.

Beyond its many built-in features, its abilities can be extended even further via its plugin system and API. Furthermore, Spyder can also be used as a PyQt5 extension library, allowing you to build upon its functionality and embed its components, such as the interactive console, in your ownsoftware.

The components of spyder are Editor, iPython Console, Variable Explorer, Plots and Debugger.

Spyder’s multi-language Editor integrates a number of powerful tools right out of the box for an easy to use, efficient editing experience. The Editor’s key features include syntax highlighting (pygments): real-time code and style analysis (pyflakes and pycodestyle): on-demand completion, calltips and go-to-definition features (rope and jedi): a function/class browser, horizontal and vertical splitting, and much more.

The Variable Explorer allows you to interactively browse and manage the objects generated running your code.

The iPython Console allows you to execute commands and enter, interact with and visualize data inside any number of fully featured iPython interpreters. Each console is executed in a separate process, allowing you to run scripts, interrupt execution and restart or terminate a shell without affecting the others or Spyder itself, and easily test your code in a clean environment without disrupting your primary session.

2.7.3 Python

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

Python's large standard library, commonly cited as one of its greatest strengths, provides tools suited to many tasks. For Internet-facing applications, many standard formats and protocols such as MIME and HTTP are supported. It includes modules for creating graphical user interfaces, connecting to relational databases, generating pseudorandom numbers, arithmetic with arbitrary-precision decimals, manipulating regular expressions, and unit testing.

Python machine learning models used here are MTCNN(Multi Task Cascaded Convolution Neural Network), FaceNet and SVM (Support vector machine).

Python has a big list of good features, few are listed below −

- It supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.

2.7.4 MTCNN

Multi Task Cascaded Convolutional Neural Network is being used for facial recognition and authentication of the user. MTCNN is a deep learning algorithm and is found to be efficient in analysing images because they use relatively little pre-processing compared to other image classification algorithms. The network uses a cascade structure with three networks; first the image is rescaled to a range of different sizes (called an image pyramid), then the first model (Proposal Network or P-Net) proposes candidate facial regions, the second model (Refine Network or R-Net) filters the bounding boxes, and the third model (Output Network or O-Net) proposes facial landmarks.

2.7.5 FaceNet

FaceNet takes an image of the person's face as input and outputs a vector of 128 numbers which represent the most important features of a face. In machine learning, this vector is called embedding. Embedding is done because all the important information from an image is embedded into this vector. Basically, FaceNet takes a person's face and compresses it into a vector of 128 numbers. Ideally, embeddings of similar faces are also similar.

An embedding is a relatively low-dimensional space into which you can translate high-dimensional vectors. Embeddings make it easier to do machine learning on large inputs like sparse vectors representing words. Ideally, an embedding captures some of the semantics of the input by placing semantically similar inputs close together in the embedding space. An embedding can be learned and reused across models.

2.7.6 SVM CLASSIFICATION

SVM is a binary classifier based on supervised learning which gives better performance than other classifiers. SVM classifies between two classes by constructing a hyperplane in high-dimensional feature space which can be used for classification.

2.8 LITERATURE SURVEY

1. **Swing-Pay: One Card Meets All User Payment and Identity Needs (2017):**

   This paper, presents an approach and module by which one card can communicate with another using near-field communication (NFC) technology to digitally transfer money from the payer's bank to the payee's bank. This approach eliminates the need for physical cash and also serves all types of payment and identity needs. Embodiments of this approach furnish a medium for cashless card-to-card transactions.

2. **A Biometric based Payment System by using Payee and Payer Module (2017):**

   This paper provides an approach and module by which one payee module can communicate with the player module using Bluetooth for money transfer from the payer's to the payee's bank. The significance of this approach is that it eliminates the physical need of case cash and serves for all types of payment and identity needs.

3. **Surveying the Development of Biometric User Authentication on Mobile Phones Surveying the Development of Biometric User Authentication on Mobile Phones (2015):**

   We survey the development of existing biometric authentication techniques on mobile phones, especially on touch enabled devices, with reference to eleven biometric approaches (five physiological and six behavioral). We present a taxonomy of existing efforts regarding biometric authentication on mobile phones and analyze their feasibility of deployment on touch enabled mobile phones.

4. **FaceNet (2015 IEEE) A Unified Embedding for Face Recognition and Clustering:**

In this paper is presented a system, called FaceNet. FaceNet learns how to directly map face images to a compact Euclidean space. The distances between the generated vectors give the similarity between the faces. The created space can be used for different tasks such as face recognition, verification and clustering using standard techniques with FaceNet embeddings as feature vectors. We extend this concept to apply it to secure online transactions.

5. **When Face Recognition Meets with Deep Learning (2015 IEEE):**

The paper aims to provide a common ground to all students and researchers alike by conducting an evaluation of easily reproducible face recognition systems based on CNNs. It uses public database LFW (Labelled Faces in the Wild) to train CNNs instead of a personal database. It proposes three CNN architectures which are the first reported architectures trained using LFW data. We use the LFW dataset to train our network as well as a personal database to test it.

6. **Building Recognition System Based on Deep Learning (2016 IEEE):**

Deep learning architectures use a multiple convolution layers and activation functions which are cascaded. The most important aspect is the setup-the number of layers and the number of neurons in each layer, the selection of activation functions and optimization algorithm. It uses GPU implementation of CNN. The CNN is trained in a supervised way in order to achieve very good results. We extend this system in order to use it for the secure online transactions.

7. **An Efficient Scheme for Face Detection (2015 IEEE):**

This work is based on skin colour, contour drawing and feature extraction to provide an efficient and simple way to detect human faces in images. The features under consideration are mouth, eyes, and nose. The results are with good accuracy, great speed and simple computations.

8. **Gender and Age Classification of Human Faces (2017 IEEE):**

This paper introduces an approach to classify gender and age from images of human faces which is an essential part of our method for autonomous detection of anomalous human behaviour. This paper is a continuous study from previous research on heterogeneous data in which images as supporting evidence is used. A method for image classification based on a pre-trained deep model for feature extraction and representation followed by a Support Vector Machine classifier is presented. We use CNN in place of SVM.

9. **Credit Card Transaction Using Face Recognition Authentication (2015 ICIIBMS):**

This paper is based on a credit card transaction system which integrates face recognition and face detection technology using Haar Cascade and GLCM algorithms. The training data set includes the extracted features from the images which is used for authentication.

CHAPTER 3

SYSTEM DESIGN

System design is the process of defining the architecture, modules, interfaces, and data for a system to satisfy specified requirements.

3.1 UML DIAGRAMS

A UML diagram is a diagram based on the UML (Unified Modelling Language) with the purpose of visually representing a system along with its main *actors*, roles, actions, *artifacts* or classes, in order to better understand, alter, maintain, or *document* information about the system.

3.1.1 USE CASE DIAGRAM

A use case diagram is a graphic depiction of the interactions among the elements of a system. It is a methodology used in system analysis to identify, clarify, and organize system requirements.

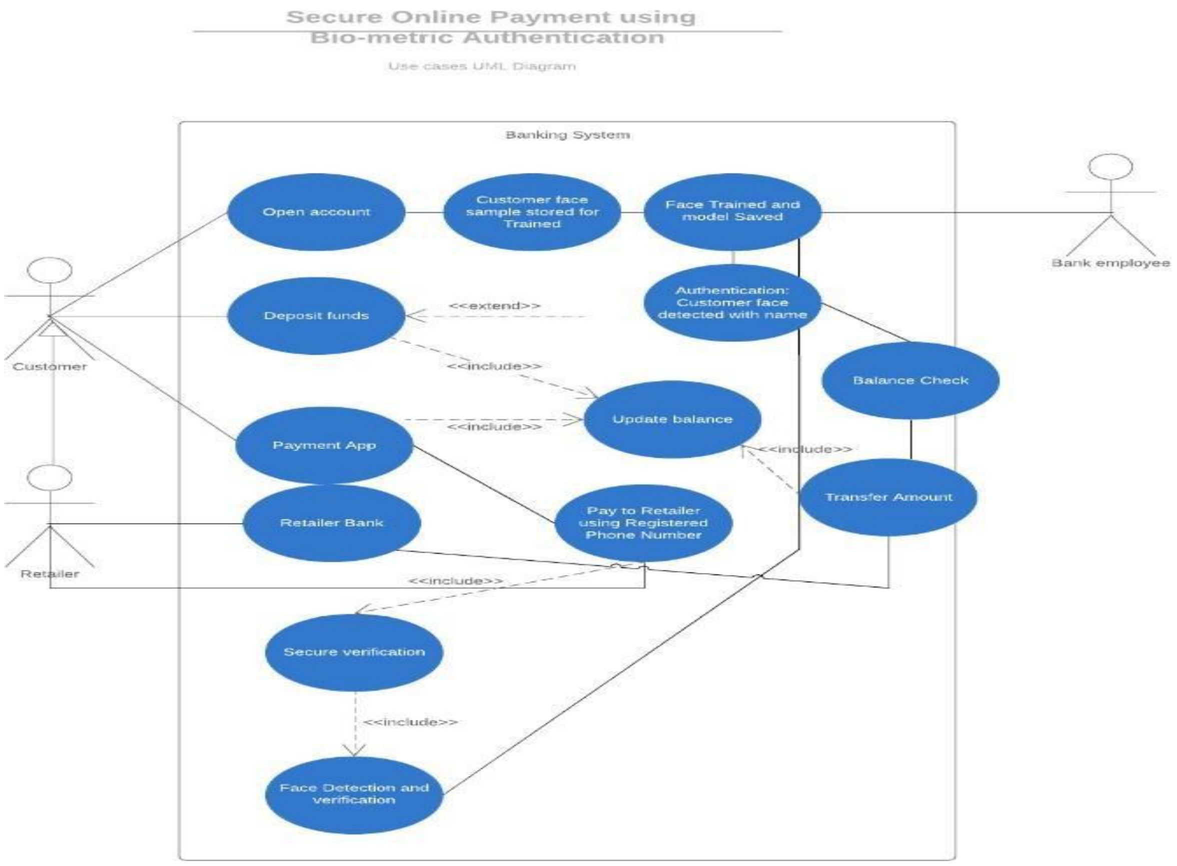

Secure Online Payment using
Bio-metric Authentication
Use cases UML Diagram
Banking System
Open account
Customer face sample stored for Trained
Face Trained and model Saved
Bank employee
Customer
Deposit funds
<<extend>>
Authentication: Customer face detected with name
<<include>>
Balance Check
Update balance
Payment App
<<include>>
<<include>>
Transfer Amount
Retailer
Retailer Bank
Pay to Retailer using Registered Phone Number
<<include>>
Secure verification
<<include>>
Face Detection and verification

FIG 3.1 Use case diagram

Due to their simplistic nature, use case diagrams can be a good communication tool for stakeholders. The drawings attempt to mimic the real world and provide a view for the stakeholder to understand how the system is going to be designed. The purpose of use case diagram is to capture the dynamic aspect of a system. Additional diagrams and documentation can be used to provide a complete functional and technical view of the system. Use case diagrams provide the simplified and graphical representation of what the system must actually do.

3.1.2 Class diagram

In Software engineering, a class diagram in the unified modelling language (UML) is a type of static structured diagram that describes the system by showing the system's classes, their Attributes, operations (or methods) and the relationship among the classes.

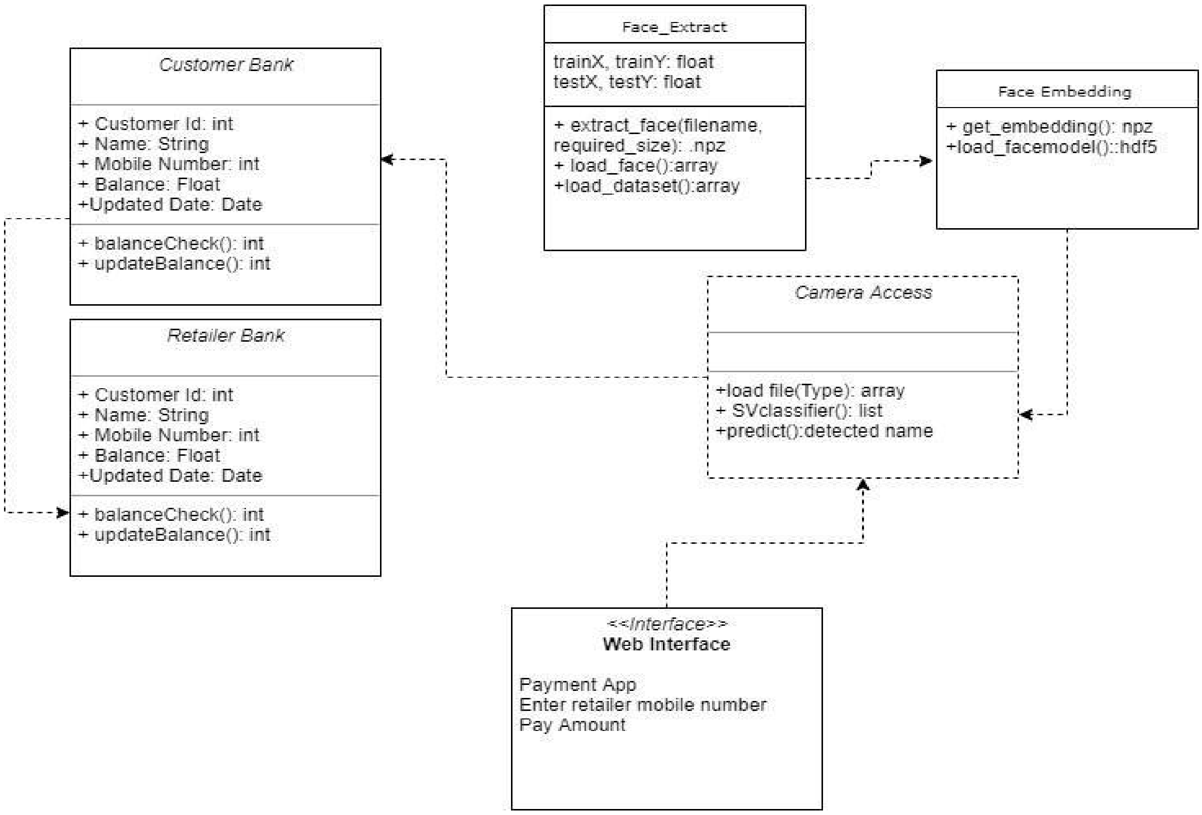


FIG 3.2 class diagram

In the diagram classes are represented with three boxes, as given below

- The top compartment contains the name of the class. It is printed in bold, and centered and the first letter is capitalized.
- The middle compartment contains the attribute of the class. They are left aligned and the first letter is lowercase.
- The bottom compartment contains the operations the class can execute and they are also left-aligned and the first letter is lowercase.

3.1.3 Activity diagram

Activity diagrams are graphical representation of workflows of stepwise activities and actions with support for choice, iteration and concurrency. Activity diagrams are intended to model both computational and organizational processes (i.e., workflows), as well as the data flows intersecting with the related activities.

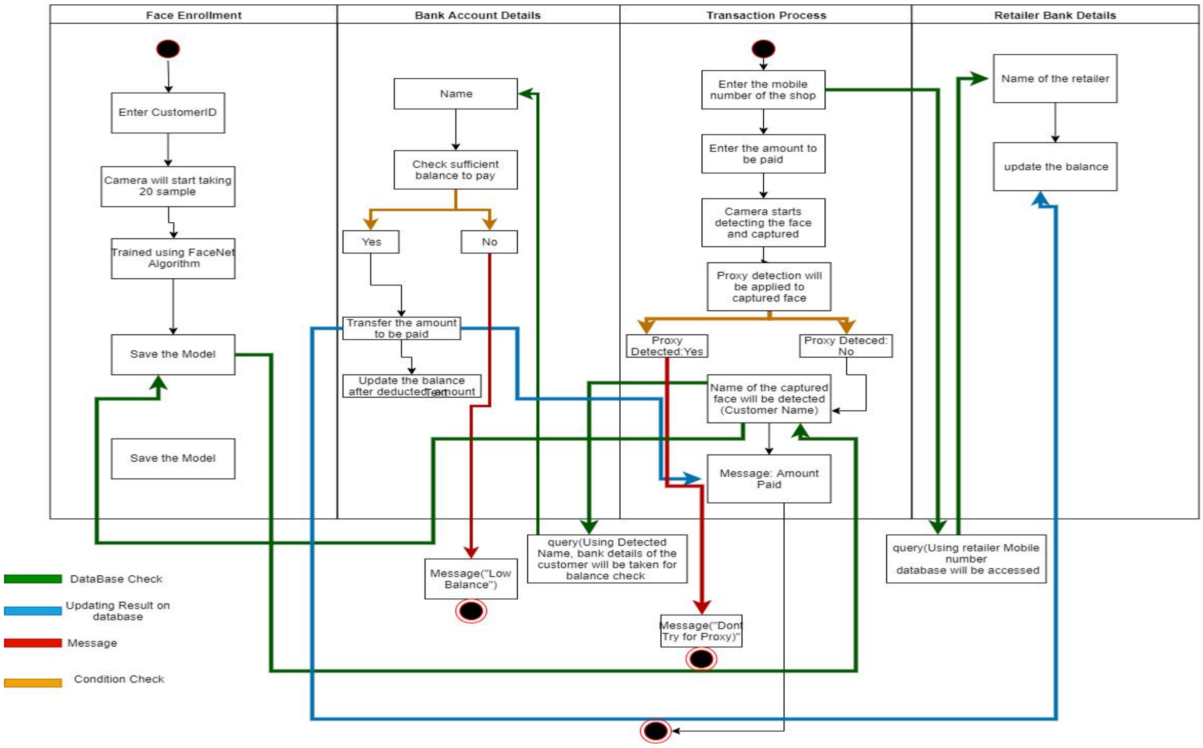

Face Enrollment
Bank Account Details
Transaction Process
Retailer Bank Details
Enter CustomerID
Camera will start taking 20 sample
Trained using FaceNet Algorithm
Save the Model
Save the Model
Name
Check sufficient balance to pay
Yes
No
Transfer the amount to be paid
Update the balance after deductedTextamount
Enter the mobile number of the shop
Enter the amount to be paid
Camera starts detecting the face and captured
Proxy detection will be applied to captured face
Proxy Detected:Yes
Proxy Deteced: No
Name of the captured face will be detected (Customer Name)
Message: Amount Paid
Name of the retailer
update the balance
query(Using Detected Name, bank details of the customer will be taken for balance check
Message("Low Balance")
Message("Dont Try for Proxy)"
query(Using retailer Mobile number database will be accessed
DataBase Check
Updating Result on database
Message
Condition Check

FIG 3.3 Activity diagram

Although activity diagrams primarily show the overall flow of control, they can also include elements showing the flow of data between activities through one or more data stores. Activity diagrams are mainly used as a flow chart consists of activities performed by the system. But activity diagram is not exactly a flow chart as they have some additional capabilities. These additional capabilities include branching, parallel flow, swim lane etc.

Activity diagrams can be regarded as a form of a structured flowchart combined with a traditional data flow diagram. Typical flowchart techniques lack constructs for expressing concurrency. However, the join and split symbols in activity diagrams only resolve this for simple cases; the meaning of the model is not clear when they are arbitrarily combined with decisions or loops.

3.1.4 Sequence diagram

A Sequence diagram is an interaction diagram that shows how processes operate with one another and in what order. It is a construct of a message sequence chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios. A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

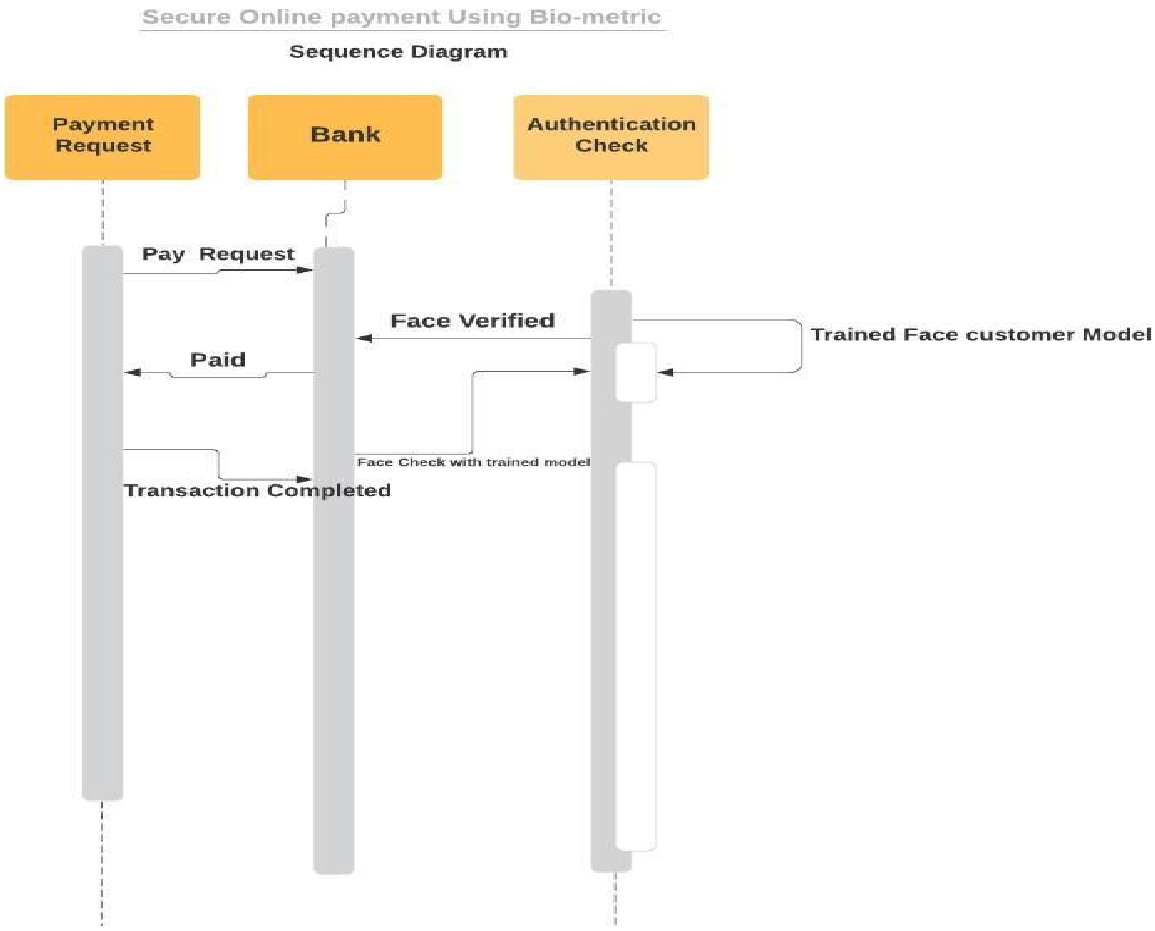


FIG 3.4 Sequence diagram

3.1.5 COMPONENT DIAGRAM

Component diagram depicts how components are wired together to form larger components or software systems. They are used to illustrate the structure of arbitrarily complex systems. A component diagram allows verification that a system's required functionality is acceptable. These diagrams are also used as a communication tool between the developer and stakeholders of the system. Programmers and developers use the diagrams to formalize a roadmap for the implementation, allowing for better decision-making about task assignment or needed skill improvements. System administrators can use component diagrams to plan ahead, using the view of the logical software components and their relationships on the system. The component diagram extends the information given in a component notation element.

The component diagram for Face Enrolment module is shown below.

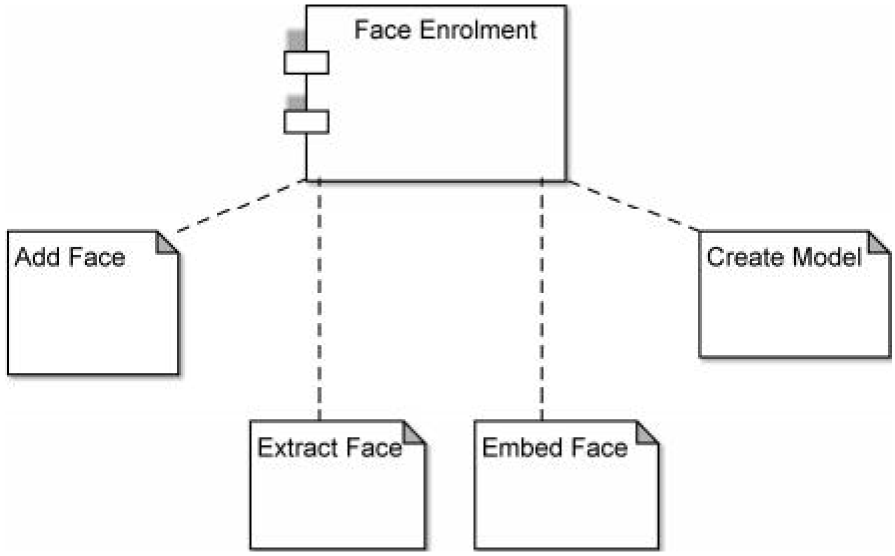


FIG 3.5 Component diagram for Face Enrolment module

The component diagram for Payment module is shown below.

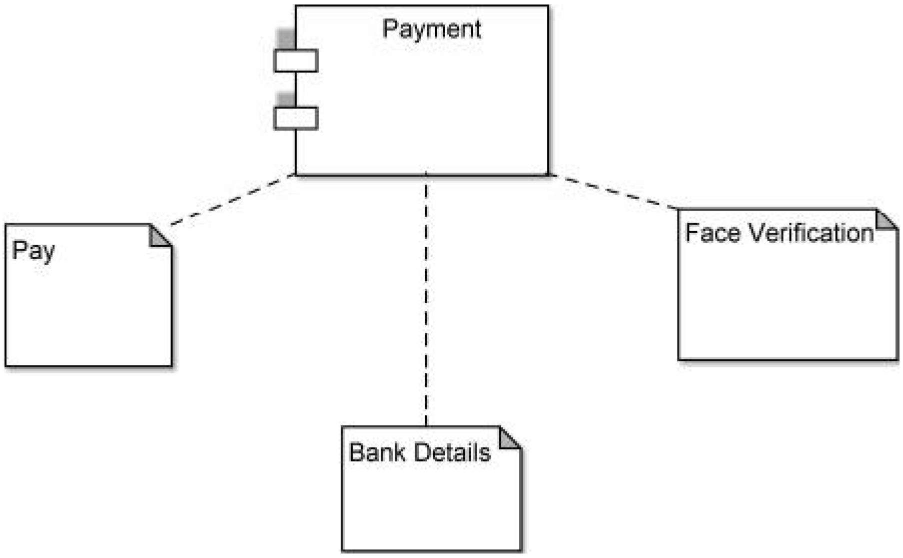


FIG 3.6 Component diagram for Payment module

3.1.6 COLLABORATION DIAGRAM

A collaboration diagram, also known as a communication diagram, is an illustration of the relationships and interactions among software objects in the Unified Modelling Language (UML). These diagrams can be used to portray the dynamic behaviour of a particular use case and define the role of each object.

Collaboration diagrams are created by first identifying the structural elements required to carry out the functionality of an interaction. A model is then built using the relationships between those elements. Several vendors offer software for creating and editing collaboration diagrams. A collaboration diagram resembles a flowchart that portrays the roles, functionality and behaviour of individual objects as well as the overall operation of the system in real time.

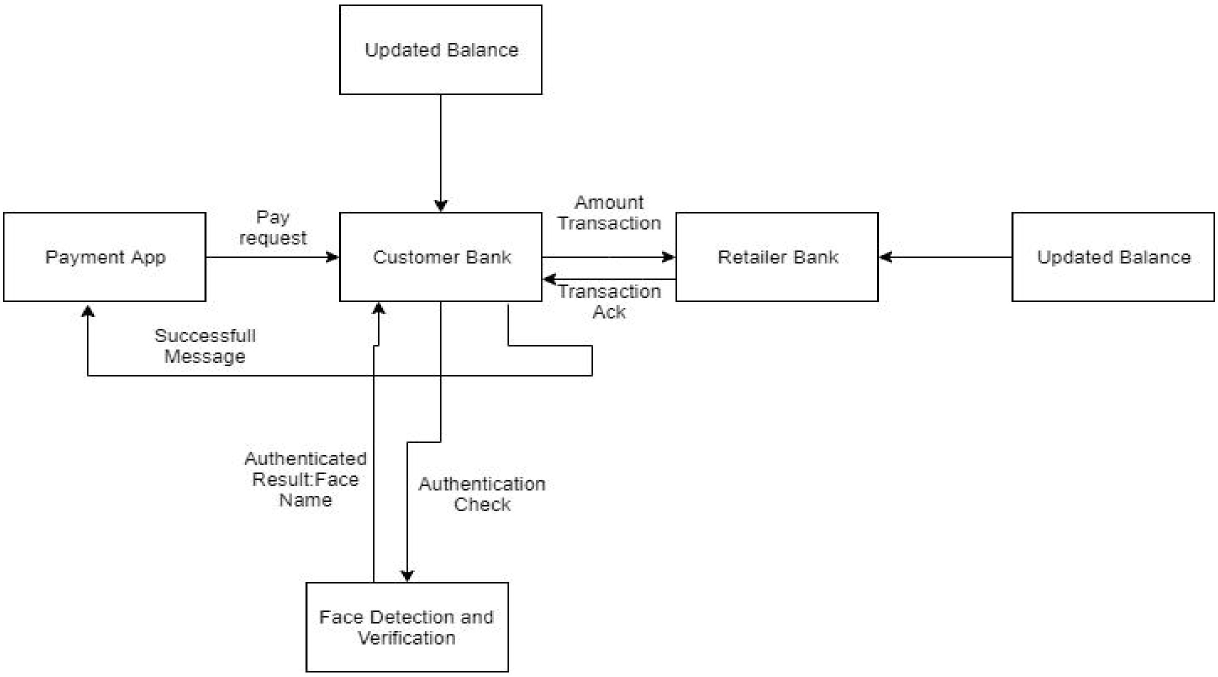


FIG 3.7 Collaboration diagram

3.1.7 Entity Relationship Diagram

Entity Relationship Diagram, also known as ERD is a diagram that displays the relationship of entity sets stored in a database. In other words, Entity Relationship diagrams help to explain the logical structure of databases. Entity Relationship diagrams are created based on three basic concepts: entities, attributes and relationships.

An entity in this context is an object, a component of data. An entity set is a collection of similar entities. These entities can have attributes that define its properties. By defining the entities, their attributes, and showing the relationships between them, an ER diagram illustrates the logical structure of databases.ER diagrams are used to sketch out the design of a database.

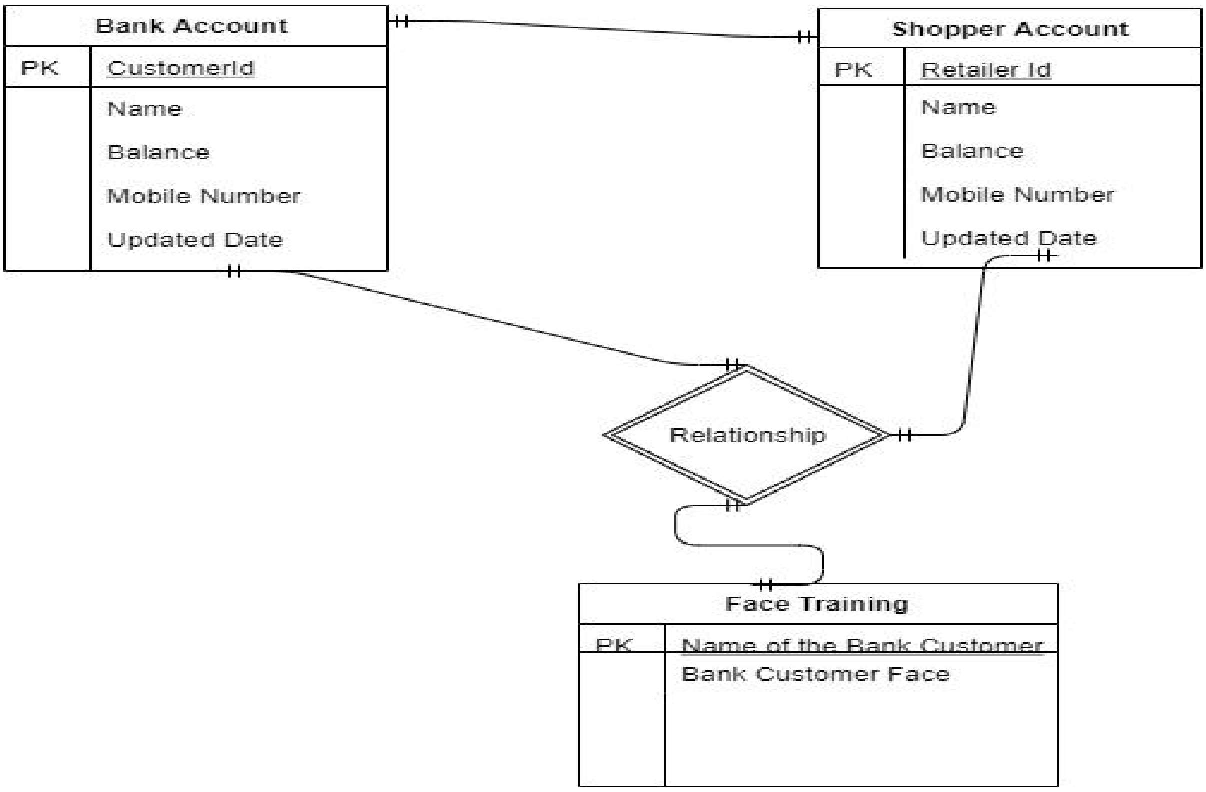


FIG 3.8 Entity Relationship diagram

3.1.8 System Architecture Diagram

An architectural diagram is a diagram of a system that is used to abstract the overall outline of the software system and the relationships, constraints, and boundaries between components. It is an important tool as it provides an overall view of the physical deployment of the software system and its evolution roadmap.

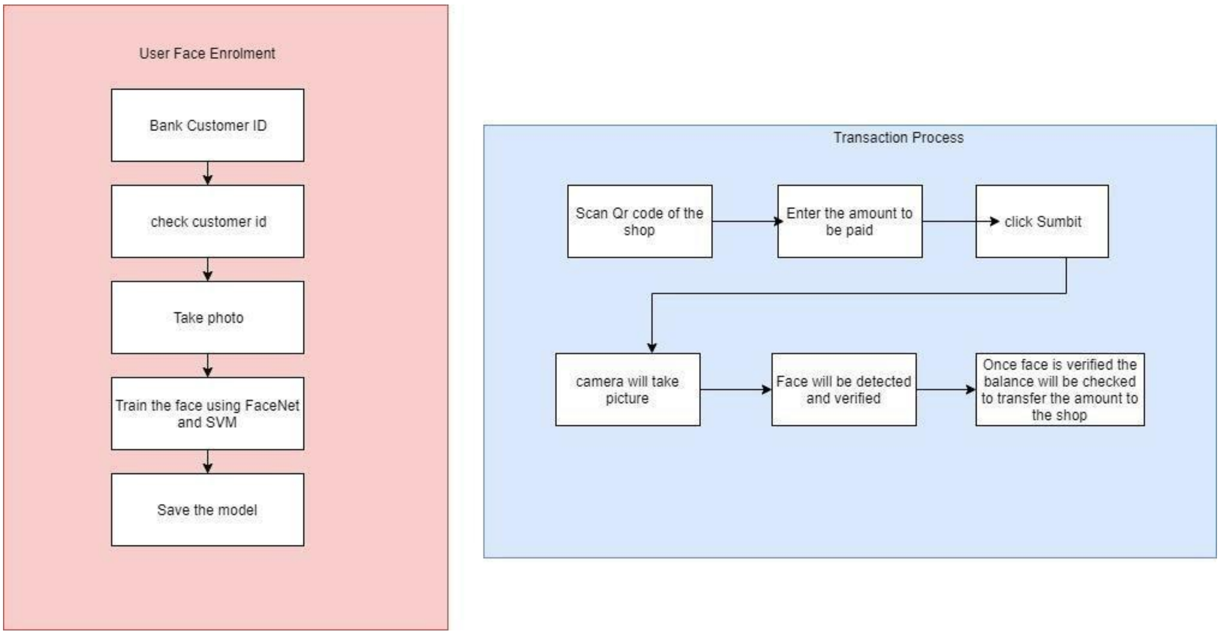

User Face Enrolment
Bank Customer ID
check customer id
Take photo
Train the face using FaceNet and SVM
Save the model
Transaction Process
Scan Qr code of the shop
Enter the amount to be paid
click Sumbit
camera will take picture
Face will be detected and verified
Once face is verified the balance will be checked to transfer the amount to the shop

FIG 3.9 System architecture diagram

3.2 DATABASE DESIGN

Backend refers to the separation of concerns between the presentation layer (front end), and the data access layer (back end) of a piece of software, or the physical infrastructure or hardware. Database Design is a collection of processes that facilitate the designing, development, implementation and maintenance of enterprise data management systems

3.2.1 CUSTOMER BANK DETAILS

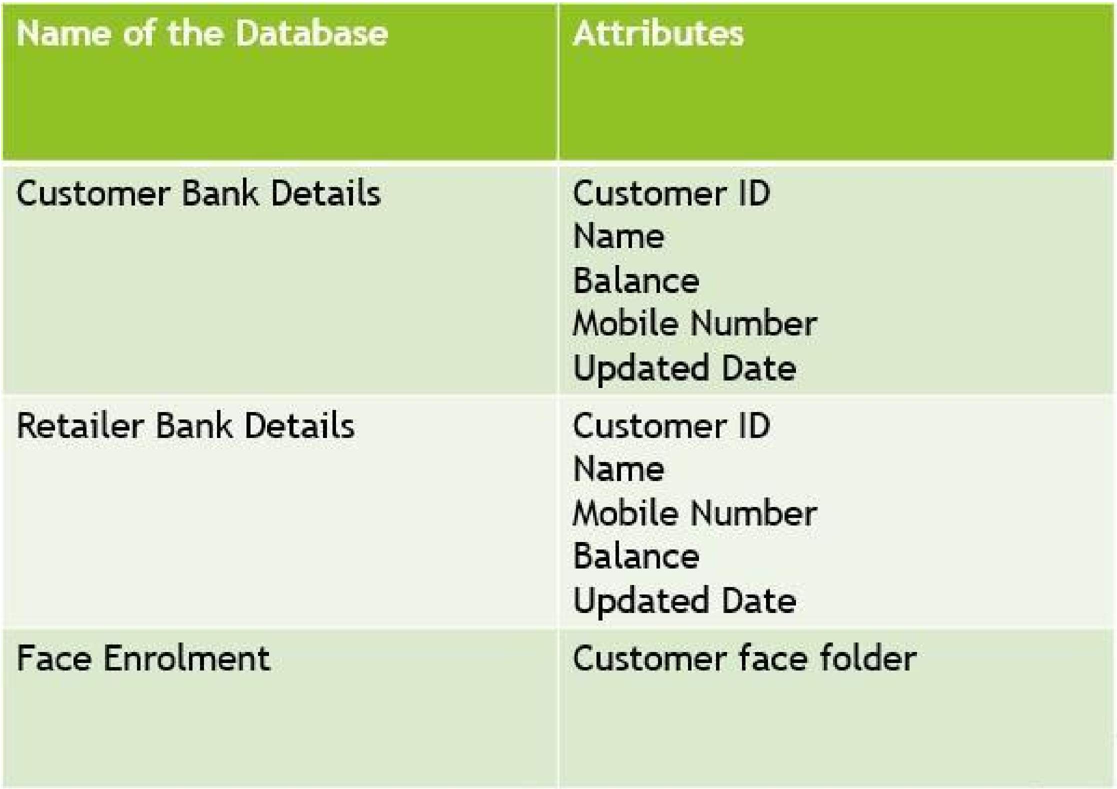
S.NO.	FIELDS	DATATYPE
1	Customer ID	INT
2	Name	STRING
3	Mobile Number	INT
4	Balance	INT
5	Updated Date	DATE

3.2.2 RETAILER BANK DETAILS

S.NO.	FIELDS	DATATYPE
1	Customer ID	INT
2	Name	STRING
3	Mobile Number	INT
4	Balance	INT
5	Updated Date	DATE

3.2.3 FACE ENROLMENT

The photos taken during face enrolment are stored in the static folder under test and train folders. The photo taken during face verification is stored in static folder under input and output folder.



Name of the Database	Attributes
Customer Bank Details	Customer ID Name Balance Mobile Number Updated Date
Retailer Bank Details	Customer ID Name Mobile Number Balance Updated Date
Face Enrolment	Customer face folder

FIG 3.10 Database Design

CHAPTER 4

SYSTEM IMPLEMENTATION AND TESTING

After having the user acceptance of the new system developed, the implementation phase begins. Implementation is the stage of a project during which theory is turned into practice. The major steps involved in this phase are:

- Acquisition and Installation of hardware and software
- Conversion
- User Training
- Documentation

The hardware and the relevant software required for running the system must be made fully operational before implementation. The conversion is also one of the most critical and expensive activities in the system development life cycle. The data from the old system needs to be converted to operate in the new format of the new system. The database needs to be setup with security and recovery procedures which is fully defined.

During this phase, all the programs of the system are loaded onto the user's computer. After loading the system, training of the user starts. Main topics of such type of training are:

- How to execute the package
- How to enter the data
- How to process the data
- How to take out the reports
- After the users are trained about the computerized system, working has to shift from manual to computerized working. The process is called 'Changeover'. The following strategies are followed for changeover of the system

The user documentation is a complete description of the system from the user's point of view detailing how to use or operate the system. It also includes the major error messages likely to be encountered by the users. The system documentation contains the details of system designs, programs, their coding, system flow, data dictionary, process description, etc

1. **Direct Changeover**

This is the complete replacement of the old system by the new system. It is a risky approach and requires comprehensive system testing and training

2.Parallel run

In parallel run both the systems, i.e., computerized and manual, are executed simultaneously for certain defined period. The same data is processed by both the systems. This strategy is less risky but more expensive because of the following:

- Manual results can be compared with the results of the computerized system.
- The operational work is doubled.

Failure of the computerized system at the early sage does not affect the working of the organization, because the, annual system continues to work, as it used to do

3. **Pilot run**

In this type of run, the new system is run with the data from one or more of the previous periods for the whole or part of the system. The results are compared with the old system results. It is less expensive and risky than parallel run approach. This strategy builds the confidence and the errors are traced easily without affecting the operations. The documentation of the system is also one of the most important activity in the system development life cycle. This ensures the continuity of the system. There are generally two types of documentation prepared for any system. These are:

- User or Operator Documentation
- System Documentation

4. Maintenance

Maintenance is necessary to eliminate errors in the system during its working life and to tune the system to any variation in its working environments. It has been seen that there are always some errors found in the systems that must be noted and corrected. It also means the review of the system from time to time. The review of the system is done for:

- Knowing the full capabilities of the system
- Knowing the required changes or the additional requirements
- Studying the performance

Project implementation (or project execution) is the phase where visions and plans become reality. It is the process of defining how the information system should be built. It includes the construction of new systems and modules.

4.1 MODULE DESCRIPTION

A module is a software component or part of a program that contains one or more routine. One or more independently developed modules make up a program.

The modules used in this project is face enrollment module and transaction process module.

4.1.1 FACE ENROLMENT MODULE

In this module, separate database from bank side will be maintained for face detection and verification. Here the details and the face of user will be stored in bank database.

4.1.1.1 INPUT DATA COLLECTION

Each bank customer will have 20 sample photos for training the face. Photo will be taken at the time of when customer opened the account. All the details of the customer is also put in the database along with the sample training face.

4.1.1.2 TRAINING THE COLLECTED INPUT DATA

Face will be trained using FaceNet Algorithm which works on triplet loss. FaceNet is a deep neural network used for extracting features from an image of person's face.

4.1.1.3 CLASSIFICATION

Extracted face will be classified using SVM classifier.

4.1.2 TRANSACTION PROCESS MODULE

This module contains the transaction procedures and transaction details of the user. Throughout the system, the transaction module manages the collection, verification, and provision of secure transaction data should be done by this module.

4.1.2.1 PAYMENT PROCESS INPUT

When customer have to pay, customer will enter the registered mobile number and the mobile number of the retailer shop, and the enter the amount to be paid.

4.1.2.2 CROSS CHECKING THE CAPTURED FACE

Camera will be starting detecting the face of the bank customer and verify with trained face. Detected face will be compared with trained model.

4.1.2.3 PROXY DETECTION

In between the verification process, detected face is proxy or real face will be detected. If proxy is detected, then transaction will not be completed.

Detected name will be compared with entered customer phone number, for confirmation.

4.1.2.4 BALANCE CHECK

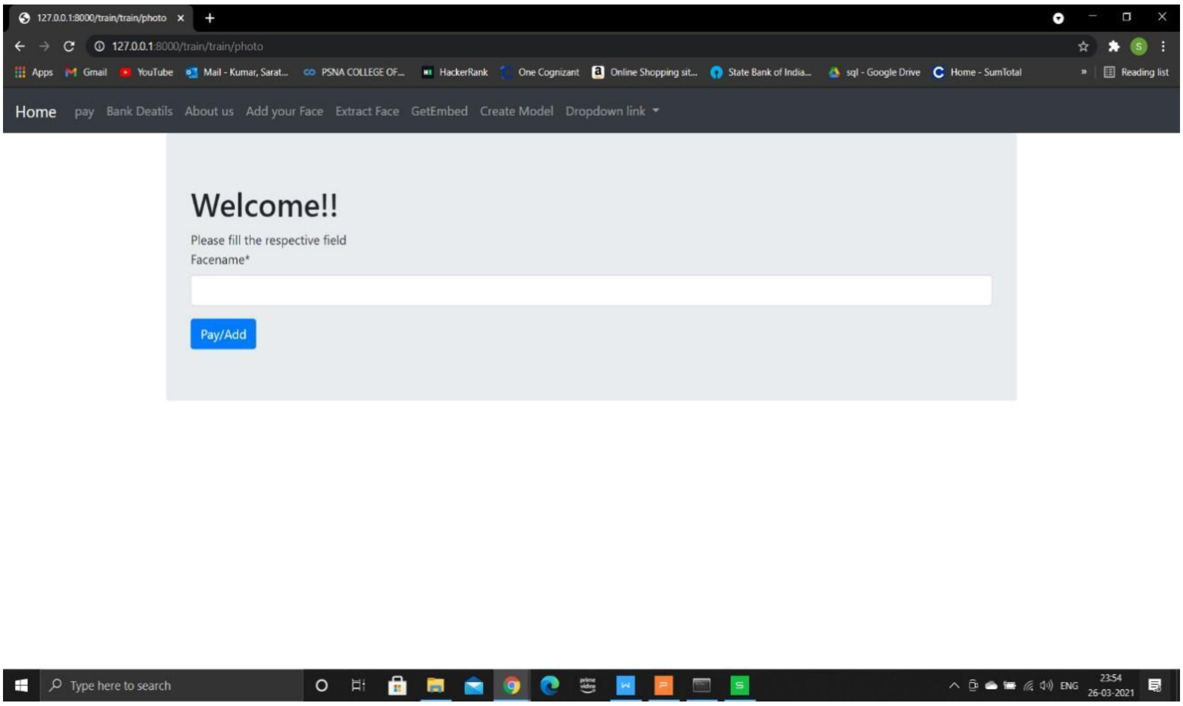
Once it confirmed, sufficient balance will be checked to transfer the amount. Finally, it will transfer the money to retailer shop. After the successful transaction the user can check the balance through their bank database.

4.2 PROCESS FOR TRANSACTION

Below are some steps for the implementation of the transaction of the user.

STEP 1: ADDING NEW FACE

The first step is to enter our name and capture our face by clicking pay/add.



Welcome!!

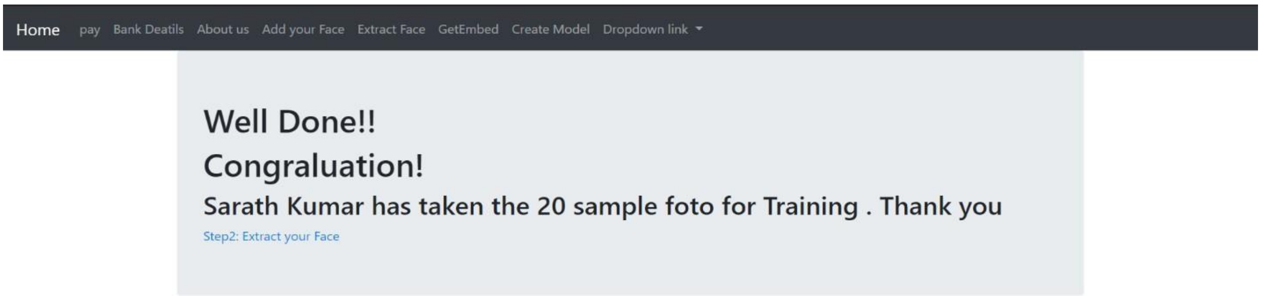
Please fill the respective field

Facename*

FIG 4.1 Add new face

STEP 2: TRAIN THE SAMPLE PHOTO

Next step the sample photos are taken and the photos are ready to get extracted which is done using MTCNN.



Well Done!!

Congraluation!

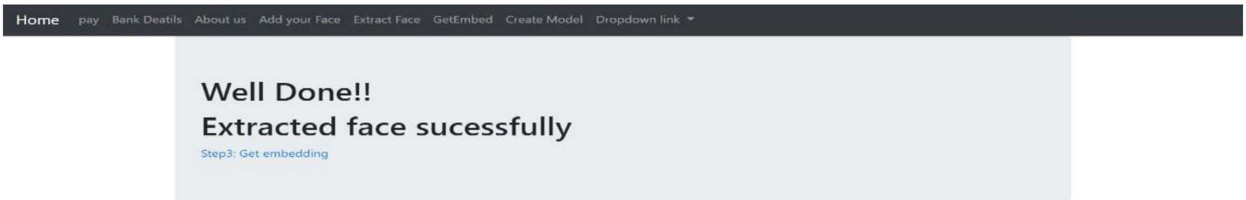
Sarath Kumar has taken the 20 sample foto for Training . Thank you

Step2: Extract your Face

FIG 4.2 Extract face

STEP 3: EMBED THE FACE

The features of the extracted faces from the photos are embedded using FaceNet.



Well Done!!

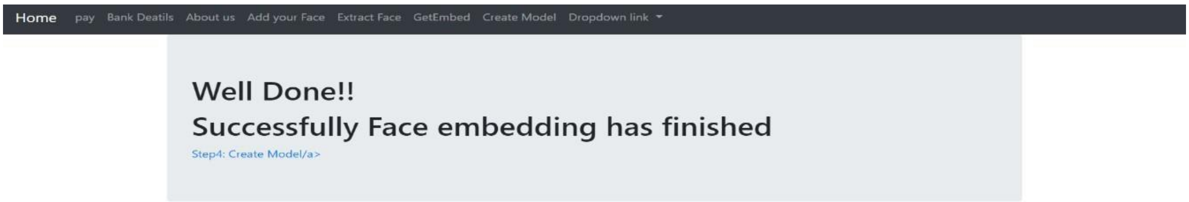
Extracted face sucessfully

Step3: Get embedding

FIG 4.3 Embed face

STEP 4: CREATE MODEL

Finally a model of the image is created after embedding is completed.

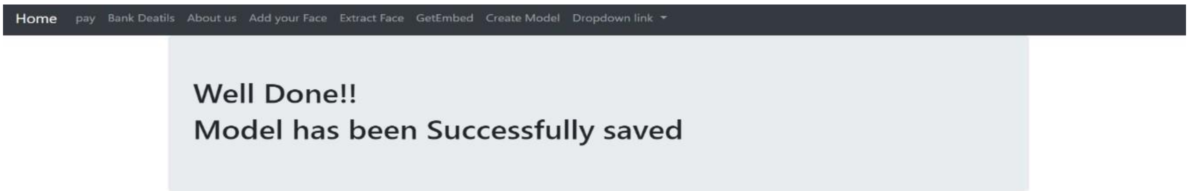


Well Done!!
Successfully Face embedding has finished
Step4: Create Model/a>

FIG 4.4 Create Model

STEP 5: SAVING THE MODEL

The trained model is saved successfully in the train folder in static.

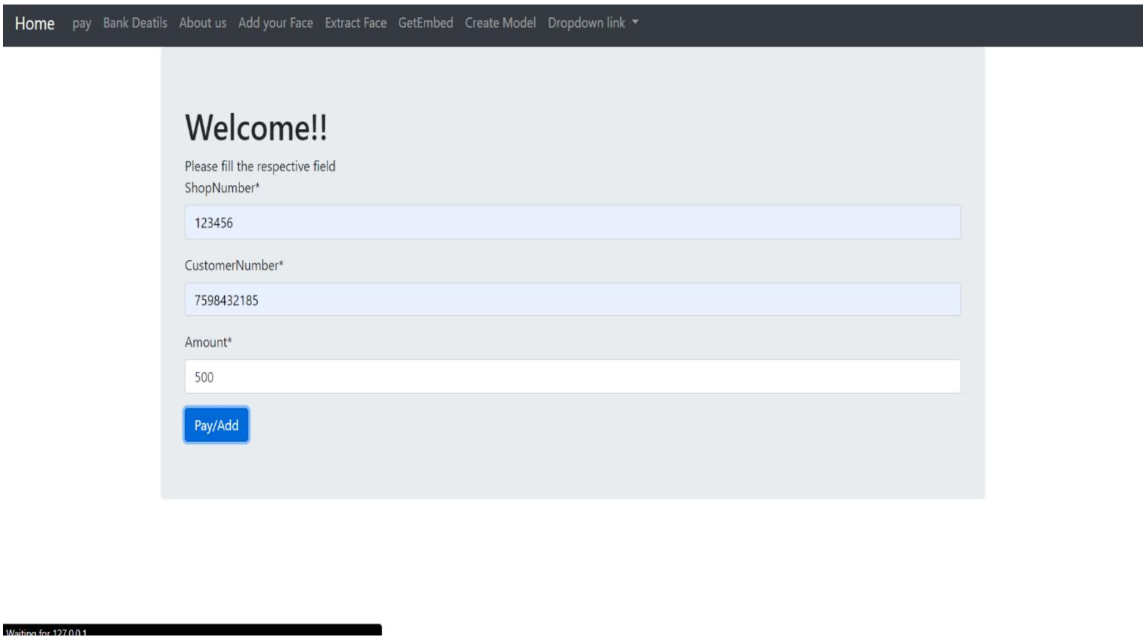


Well Done!!
Model has been Successfully saved

FIG 4.5 Save model

STEP 6: MAKING THE PAYMENT

Once face enrolment process is completed payment details are entered.



Welcome!!

Please fill the respective field

ShopNumber*

123456

CustomerNumber*

7598432185

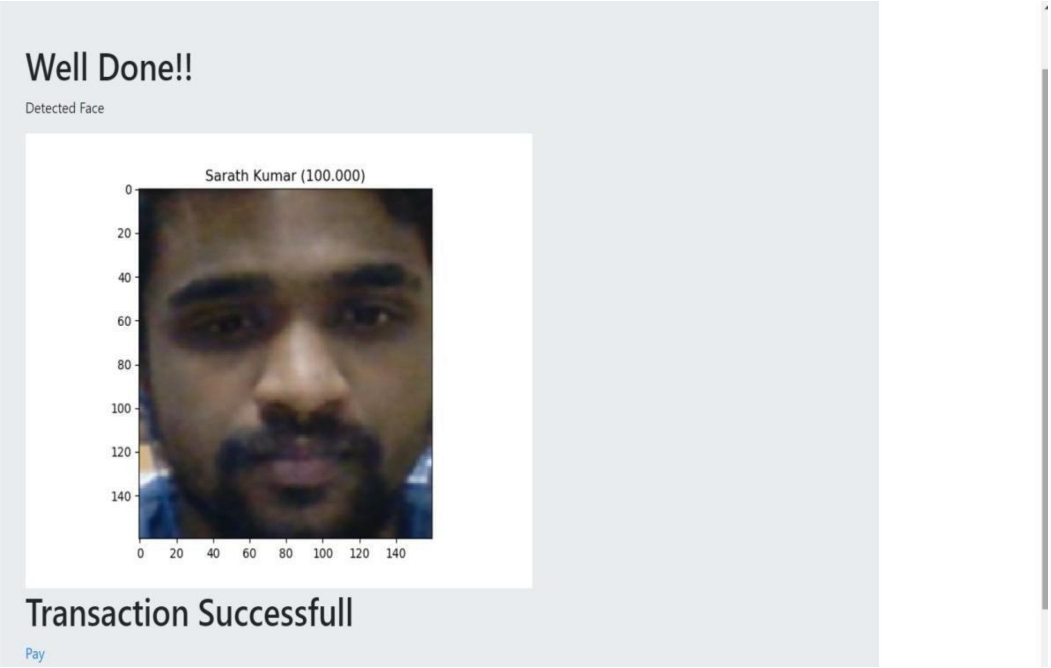
Amount*

500

FIG 4.6 Make Payment

STEP 7: TRANSACTION SUCCESSFUL

On clicking the pay/add button face verification process is done and the face if identified transaction gets completed and if not an error message is displayed.



Well Done!!

Detected Face

Transaction Successfull

Pay

FIG 4.7 Output

INPUT DESIGN

The input design for the payment process is illustrated below.

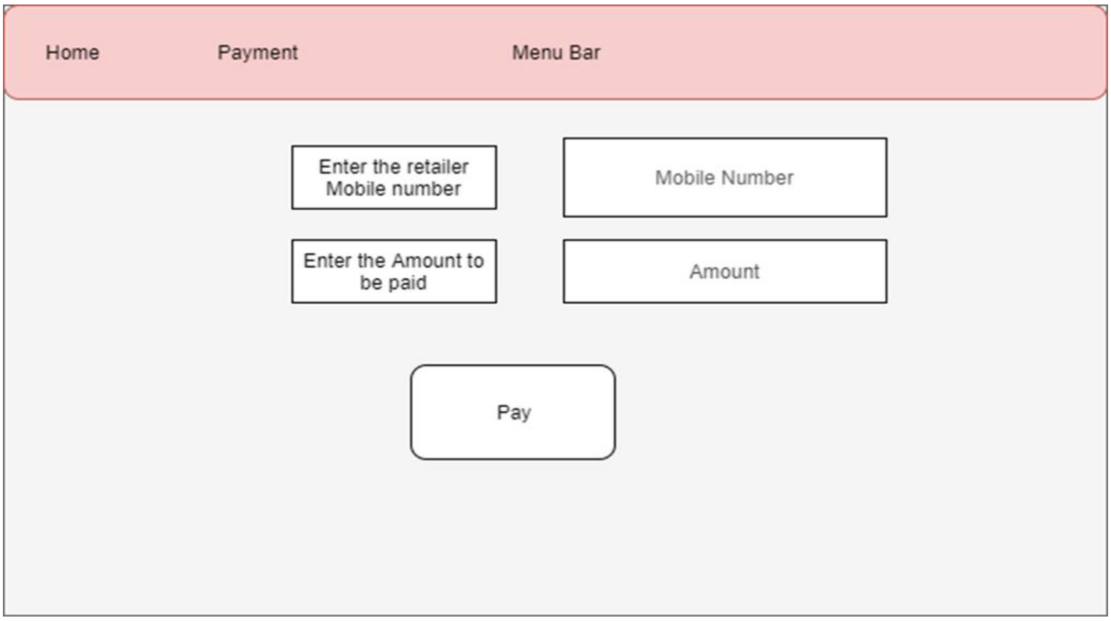


FIG 4.8 Input Design

OUTPUT DESIGN

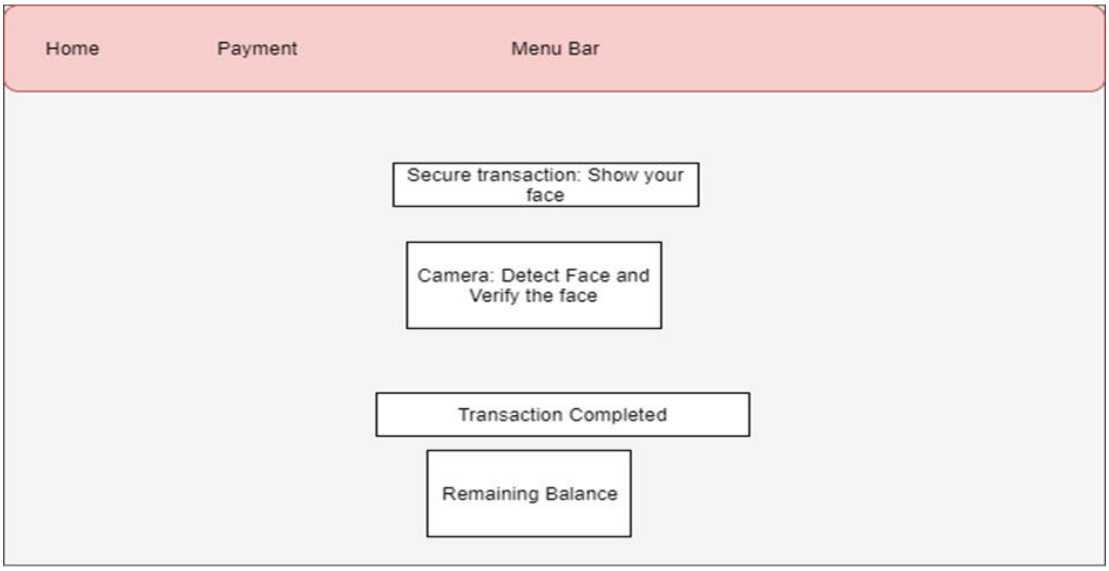
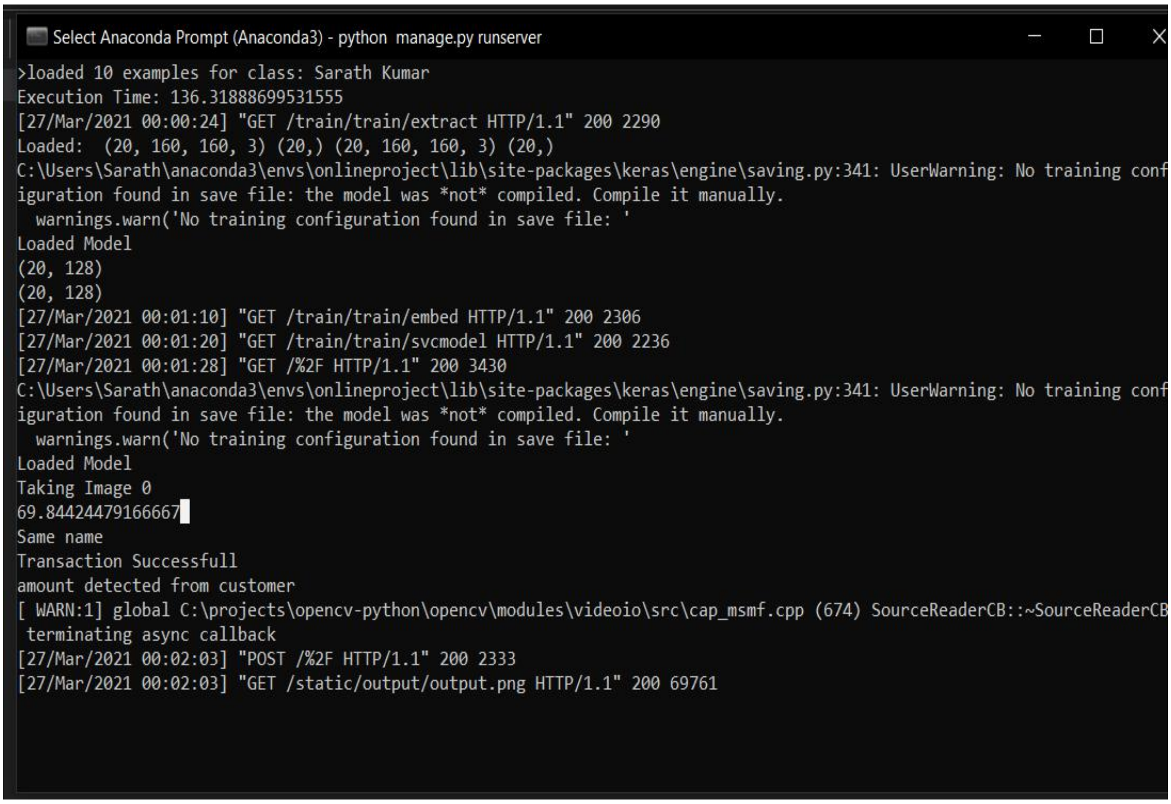

Home
Payment
Menu Bar
Secure transaction: Show your face
Camera: Detect Face and Verify the face
Transaction Completed
Remaining Balance

FIG 4.9 Output Design

CONSOLE OUTPUT

The console output displays the pixel value average to determine if the image is fake or live and also it displays if transaction gets completed.

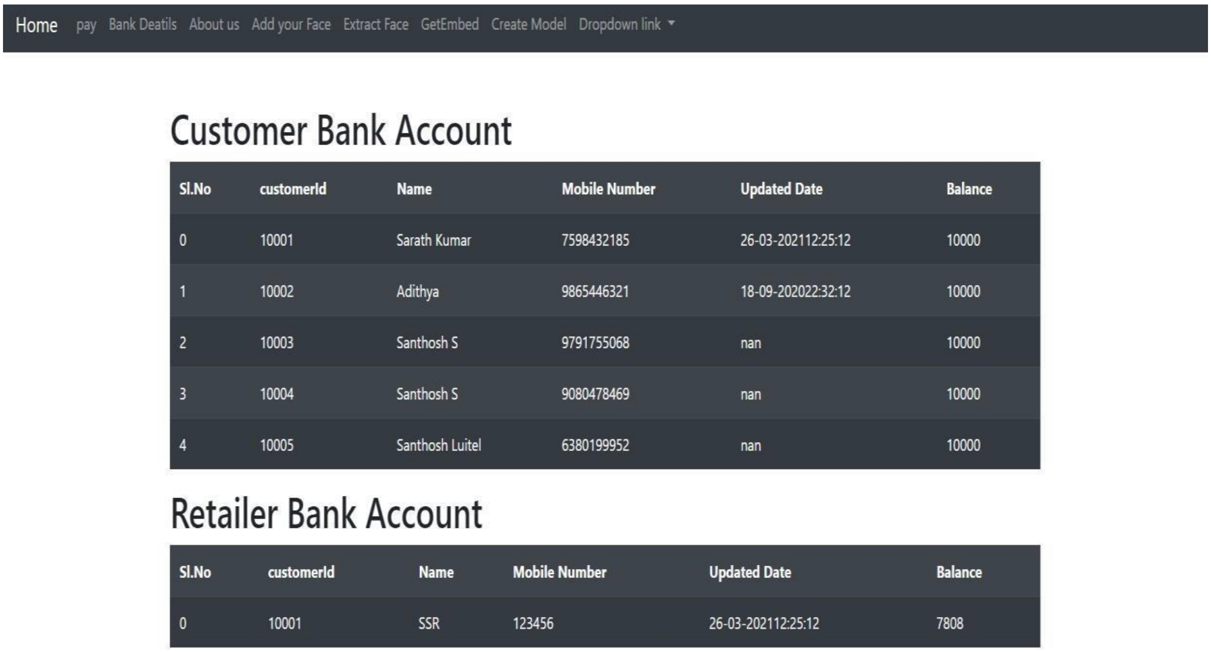

Select Anaconda Prompt (Anaconda3) - python manage.py runserver

```
>loaded 10 examples for class: Sarath Kumar
Execution Time: 136.31888699531555
[27/Mar/2021 00:00:24] "GET /train/train/extract HTTP/1.1" 200 2290
Loaded:  (20, 160, 160, 3) (20,) (20, 160, 160, 3) (20,)
C:\Users\Sarath\anaconda3\envs\onlineproject\lib\site-packages\keras\engine\saving.py:341: UserWarning: No training conf
iguration found in save file: the model was *not* compiled. Compile it manually.
  warnings.warn('No training configuration found in save file: '
Loaded Model
(20, 128)
(20, 128)
[27/Mar/2021 00:01:10] "GET /train/train/embed HTTP/1.1" 200 2306
[27/Mar/2021 00:01:20] "GET /train/train/svcmodel HTTP/1.1" 200 2236
[27/Mar/2021 00:01:28] "GET /%2F HTTP/1.1" 200 3430
C:\Users\Sarath\anaconda3\envs\onlineproject\lib\site-packages\keras\engine\saving.py:341: UserWarning: No training conf
iguration found in save file: the model was *not* compiled. Compile it manually.
  warnings.warn('No training configuration found in save file: '
Loaded Model
Taking Image 0
69.84424479166667
Same name
Transaction Successfull
amount detected from customer
[ WARN:1] global C:\projects\opencv-python\opencv\modules\videoio\src\cap_msmf.cpp (674) SourceReaderCB::~SourceReaderCB
 terminating async callback
[27/Mar/2021 00:02:03] "POST /%2F HTTP/1.1" 200 2333
[27/Mar/2021 00:02:03] "GET /static/output/output.png HTTP/1.1" 200 69761
```

FIG 4.10 Console Output

DATEBASE: CUSTOMER AND RETAILER

The details of the customer and the customer bank account is stored in the database. Using this reference of database, the system will process the transaction for the payment.



Customer Bank Account

Sl.No	customerId	Name	Mobile Number	Updated Date	Balance
0	10001	Sarath Kumar	7598432185	26-03-202112:25:12	10000
1	10002	Adithya	9865446321	18-09-202022:32:12	10000
2	10003	Santhosh S	9791755068	nan	10000
3	10004	Santhosh S	9080478469	nan	10000
4	10005	Santhosh Luitel	6380199952	nan	10000

Retailer Bank Account

Sl.No	customerId	Name	Mobile Number	Updated Date	Balance
0	10001	SSR	123456	26-03-202112:25:12	7808

FIG 4.11 Database

4.3 TESTING

4.3.1 SYSTEM TESTING

Software testing is an investigation conducted to provide stack holder with information about the quality of the product or service under test. It is a level of testing that validates the complete and fully integrated software product. Software testing can also provide an objective, in dependent view of the software to allow the business to appreciate and understand the risk of software implementation.

Test techniques include but are not limited to the process of executing a program or application with the intent of finding Software bugs (errors or other software defects). The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in the work product.

It provides a way to check the functionality of components, subassemblies, assemblies and/or a finished product it is the process of executing software with the intent of ensuring the software system meets its requirements and user expectation and does not fail in an unacceptable manner.

4.3.2 UNIT TESTING

This type of testing is performed by developers before the setup is handed over to the testing team to formally execute the test case. Unit testing is performed by respective developers on the individual unit of source code. The developers use test data that is separate from the test data of the quality assurance team. The goal of unit testing is to isolate each part of the program and show the individual parts are correct in terms of requirements and functionality. Unit testing is important because software developers sometimes try saving time doing minimal unit testing and this is myth because inappropriate unit testing leads to high cost defect fixing during System testing and Integration testing even Beta Testing after application is built. If proper unit testing is done in early development, then it saves time and money in the end.

4.3.3 SOFTWARE TESTING

This is the next level in testing and test the system as a whole. Once all the components are integrated, the application as a whole is tested rigorously to see that it meets Quality Standards. This type of testing is performed by specialized testing team. Software testing is important because if there are any bugs or errors in the software, it can be identified early and can be solved before delivery of the software product. Properly tested software product ensures reliability, security and high performance which further results in time saving, cost effectiveness and customer satisfaction. Testing is important because software bugs could be expensive or even dangerous. Software bugs can potentially cause monetary and human loss, and history is full of such examples.

CHAPTER 5

CONCLUSION AND FUTURE ENHANCEMENT

ADVANTAGES

- Only with live face payment will be completed.
- No need to share the OTP for friends and family.
- Proxy Detector is included, if showing any photo in front the prediction system, it automatically detects the proxy and placement will not be completed.
- Secure payment is established**.**

DISADVANTAGES

- Implementation in the real time, to change entire process may be difficult.
- Training time may take longer for the large dataset.
- Dataset of the customer face at least need 20 for each**.**

5.1 CONCLUSION

This system can convert the less security into more security transaction. This system creates a more reliable communication means user and bank system. Also, it generates face at the end of transaction which is verify by various algorithm.

Payment system for online transaction is based on face recognition provides authenticate user data privacy and prevents misuse of data in world.

- This technique identify theft and prevent customer data which improve security.
- This system can convert the less security into more security transaction. This system creates a more reliable communication means for user and bank system.
- It will also go a long way in ensuring that innocent people are not wrongly arrested based on previous crimes

5.2 FUTURE ENHANCEMENT

The proposed system gives a method for online transaction security using facial recognition. The system can be developed to automate the process. Some future enhancements for the proposed system can be:

- Improving the system for banking transactions.
- To add some more feature to make the transaction secure.
- More efficient Object Detection Neural network.
- To implement the system in a banking and online transaction section web is mot more convenient so develop the app for this application.

The proposed system can be evolved to meet several other operations which are not included in this project. Expanding the system will result in more efficient and hassle free operations.

REFERENCES

- Ahmad Amran, Surya Michrandi Nasution, Fairuz Azmi," Implementation of Cryptography Algorithm For Biometric Payment". E-proceeding of engineering, Vol.3, No.1 April 2016.
- Phil Tresandern and Timothy F.Cootes," Mobile Biometrics: Combined Face and Voice Verification for the mobile platform", IEEE, Books, 2013.
- Selvia Rahmawati1, Ichsan Taufik, Gitarja Sandi," Implementasi Algoritma AES(Advanced Encryption Standard)256Bit Dan Kompresi Menggunakan Algoritma Huffman Pada Aplikasi Voice Recorder", SENTER 2017, 15-16 December 2017.
- Microsoft Azure, "Face API Azure Cloud documentation" access February 2019. Available at https://www.azure.microsoft.com.
- J.Zhou, X.G.Lu, D. Zhang, C. Wu, Orientation analysis for rotated human face detection, Image and Vision Computing, 20, 2002, 257264.
- Weizhi Meng; Wong, D.S.; Furnell, S.; Jianying Zhou,"Surveying the Development of Biometric User Authentication on Mobile Phones," in Communications Surveys & Tutorials, IEEE, vol.17, no.3, pp.12681293, third quarter 2015.