Heart Disease Detection Using Iridology

| | |
|---|---|
| | |
| Submitted By | Azhar Ud Din |
| | |
| Roll no. | 011-BSCS-20 |
| | |
| Responsibility | Data Science |
| | |
| Submitted By | Shajar Abbas |
| | |

| | |
|---|---|
| Roll no. | 0304-BSCS-20 |
| | |
| Responsibility | Web Development |
| | |
| Supervised By | Asma Kanwal<br>Assistant Professor |
| | |

BS(HONS)

IN

COMPUTER SCIENCE


DEPARTMENT OF COMPUTER SCIENCE

GC UNIVERSITY LAHORE

# Teachers Evaluation System

Submitted to GC University Lahore in partial ful llment of the requirements for the award of degree of

BS(HONS)

IN

COMPUTER SCIENCE



| | |
|---|---|
| Submitted By | Azhar Ud Din |
| | |
| Roll no. | 011-BSCS-20 |
| | |
| Responsibility | Data Science |
| | |
| Submitted By | Shajar Abbas |
| | |
| Roll no. | 0304-BSCS-20 |
| | |
| Responsibility | Web Development |
| | |
| Supervised By | Asma Kanwal<br>Assistant Professor |
| | |

DEPARTMENT OF COMPUTER SCIENCE

GC UNIVERSITY LAHORE

# Declaration

We, Azhar Ud Din and Shajar Mehar, students of BS(Hons) in the subject of Computer Science session 2020 - 2024, hereby declare that the matter printed in this thesis titled,Heart Disease Detection using iridology is our own work and has not been printed, published and submitted as research work, thesis or publication in any form in any University, Research Institution etc in Pakistan or abroad.

Date: 08/09/2024 Signatures of Department

# Research Completion Certificate

It is certified that the research work contained in this thesis titled Heart disease detection using Iridology has been carried out by Azhar Ud Din and Shajar Abbas Roll. No 011-BSCS-20 and 0304-BSCS-20 under my supervision.

Dr. Muhammad Ilyas Fakhir

Assistant Professor

Date:

Submitted Through

Prof. Dr. Syed Asad Raza Kazmi
Chairperson
Department of Computer Science
GC University Lahore

Controller of Examination
GC University Lahore

# Abstract

Heart disease is the deadliest disease in the world. Using old methodology of heart disease detection in early stage is expensive, inconvenient, and slow. Mostly, testing of heart disease is performed after first heart attack or in severely pain condition as its expenses are too high which can not be afforded even by average income families in a country like Pakistan. Therefore, system is designed to automatically classify iris images into two classes normal and abnormal heart images. Each image is pre-processed for noise reduction, conversion into grey image, and for smoothing. Then, the Iris part is segmented using Doughman's algorithm [3]. The segmented image is normalized using Doughman's rubber sheet algorithm [3]. Features of heart between 2:00 to 3:15 clockwise part is extracted and features are reduced using PCA [4] from normalized image. Finally, the model is trained on CNN on PCA heart extracted features. This modern technology is inexpensive, and can be used for heart disease detection anytime.

# Contents

| | | | | |
|---|---|---|---|---|
| Declaration | | | | 3 |
| Research Completion Certificate | | | | 4 |
| Abstract | | | | 5 |
| Contents | | | | 6 |
| 1 | Introduction | | | 8 |
| | 1.1 | Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 8 |
| | | 1.1.1 Image preprocessing . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 9 |
| | | 1.1.2 Image Localization . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 10 |
| | | 1.1.3 Iris Segmentation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .<br>1.1.4 Iris Normalization . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 11 |
| | | 1.1.5 Feature Extraction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 12 |
| | | 1.1.6 Feature Reduction using PCA . . . . . . . . . . . . . . . . . . . . . . . . | | 12 |
| | | 1.1.7 Classification . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 13 |

| | | | | |
|---|---|---|---|---|
| | 1.2 | Literature Review . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 13 |
| 2 | Requirement Specification | | | 14 |
| | 2.1 | Functional Requirements . . . . . . . . . . . . . . . . . . . . . . . . | | 14 |
| | 2.2 | Non-functional Requirements . . . . . . . . . . . . . . . . . . . . . | | 156 |
| 3 | Project Design | | | 4 |
| | 3.1 | Methodology . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 4 |
| | 3.2 | Architecture Overview . . . . . . . . . . . . . . . . . . . . . . . . | | 4 |
| | 3.3 | Design Description . . . . . . . . . . . . . . . . . . . . . . . . . . | | 4 |
| 4 | Implementation and Evaluation | | | 5 |
| | 4.1 | Development Stages . . . . . . . . . . . . . . . . . . . . . . . . | | 5 |

| | | | |
|---|---|---|---|
| | | | |
| | 4.1.1 | Stage abc.. . . . . . . . . . . . . . . . . . . . . . . . . . . . | 5 |
| | 4.1.2 | Stage def.. . . . . . . . . . . . . . . . . . . . . . . . . . . . | 6 |
| 4.2 | System Integration . . . . . . . . . . . . . . . . . . . . . . . . . . | | 6 |
| 4.3 | User Interface . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 6 |
| 4.4 | Evaluation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 6 |
| 4.5 | Unit Testing . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 6 |
| 4.6 | Functional Testing . . . . . . . . . . . . . . . . . . . . . . . . . . | | 6 |
| | 4.6.1 | Testing Requirements . . . . . . . . . . . . . . . . . . . . . . | 6 |
| 4.7 | Requirements . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 6 |
| | 4.7.1 | Accurate Speci ed Run . . . . . . . . . . . . . . . . . . . . . | 8 |
| | 4.7.2 | Synchronized Speci cation . . . . . . . . . . . . . . . . . . . | 10 |
| 5 Conclusion & Future Work | | | 12 |
| 5.1 | Conclusion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 15 |
| A Appendix Title Here | | | 16 |
| Bibliography | | | 17 |

# Chapter 1

# Introduction

## 1.1 Introduction

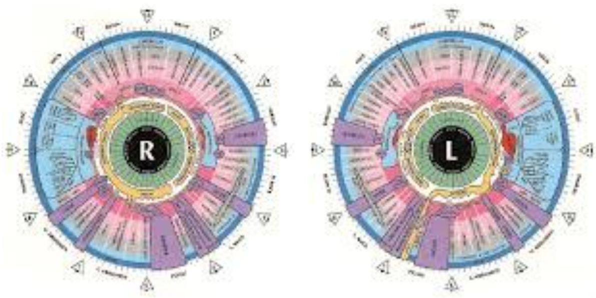
According to WHO (World Health Organization) report of (June 2011) the cardiovascular d-isease are the primary cause of death globally. (32%) of all fatalities worldwide in (2019) we-re attributed to CVDs, with an estimated (17.9) million deaths. Heart attacks and strokes were the cause of (85%) of these fatalities. More than (75%) of death from CVD occur in low and middle-income nations. In (2019), noncommunication illnesses accounted for (17) million pr-emature fatalities (deaths under 70 years of age) of which (38% ) were attributable to CVDs.

The age-standardized death rate was (357.88) per (100000) (global, 239.85 per 100000) and the projected age-standardized incidence of CVD was (918.18) per (100000) (global, 684.33

per (100000) in Pakistan, according to the (2019) Global Burden of Disease research.1. (18.9-%) of participants in the National Socioeconomic Registry Survey self-reported having CVD-s. The survey covers the demographic, socioeconomic, educational, health, and asset profiles of (34) million households in Pakistan. [2] (Samad and Hanif 2023)

As most people do not afford for checking CVD symptoms in early age due to high amount of money required for testing. The heart checking old methodology is also inconvenient and takes long time. Therefore, Iridology method can be used as a solution to all above highlight-ed main problems. According to the (Bernard Jenssen n.d.) Iris is a Window which shows w-hat does inside the body. The iris reveals the tissue conditions of each organism of the body. When the darkness found in any area of the Iris, it corresponds to pathology somewhere in b-ody. (pg67). That’s why the iridology is the priceless tool to primary health care professionals using wholistic approach. (pg 61).

According to The Dr. Bernard Jenson's charting of Iridology The heart part is positioned cloc-kwise direction between 2:00 to 3:15 o’clock of left iris. This is a cheap and an advanced me-thod of detecting heart abnormalities through the iris.



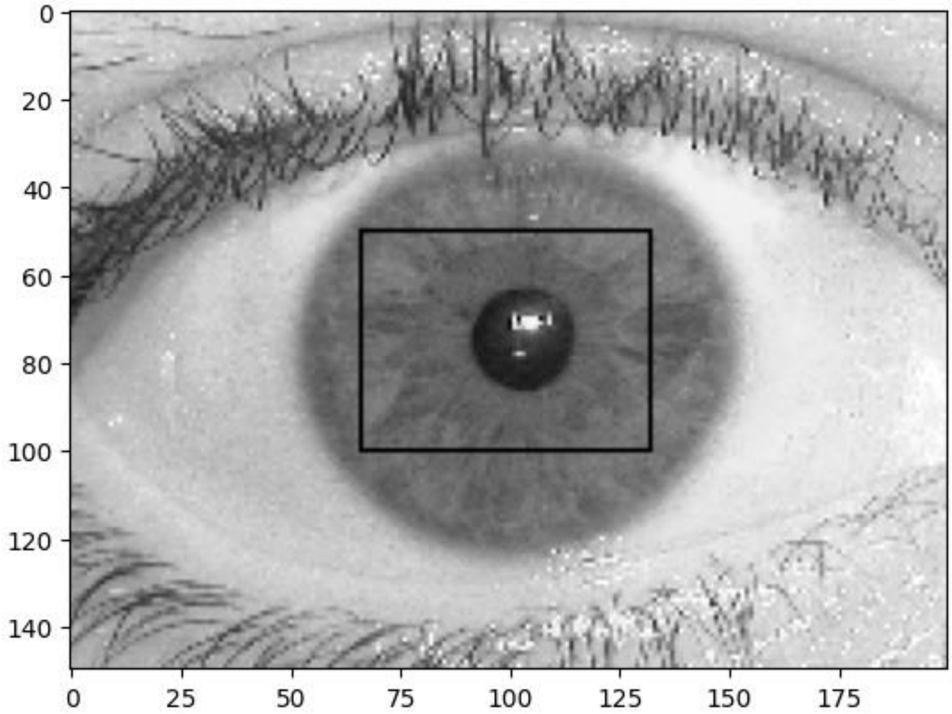
Therefore, system is designed to automatically classify iris images into two classes normal and abnormal heart images. Each image is pre-processed for noise reduction, conversion into grey image, and for smoothing applying histogram equalization. Then, the Iris part is segmented using Doughman’s algorithm [3]. The segmented image is normalized using Doughman’s rubber sheet algorithm [3]. Features of heart between 2:00 to 3:15 clockwise part is extracted and features are reduced using PCA [4] from normalized image. Finally, the model is trained on CNN on PCA heart extracted features.

### 1.1.1 Image preprocessing

The digital image captured through digital camera is in RGB (Red Green Blue) format. The RGB image is converted into Greyscale image. Greyscale images have intensity value between (0-255) 8bits. Images will be enhanced using histogram equalization to detect the iris and pupil boundaries correctly.
As light reflection effects the detection of iris images negatively using Daugman’s algorithm. Therefore, morphological operation using MATLAB tool ‘imfill’ is applied. Secondly, to make Daugman’s algorithm works faster in detection of iris, the image is scanned pixel by pixel to detect the local minimum in that particular pixel’s immediate 3-by-3 neighborhood. The pixel with lowest intensity value amongst nine by nine is used for further calculations. We will insinuate that, provided the eye is appropriately cropped, the iris cannot be greater than one-third of the image side in order to minimize the number of possible points. So that

we  every pixel will not be checked on the image, and only those, which are in the central part of 1/3 of the image:

| | | |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 0 | 0 | 0 |




# •	Image Localization

Daugman's operator algorithm proposed in (1993) will be used for localization of circular outer boundary of iris that separate the iris part from the sclera part of the eye image.

$$\max_{(r,x_0,y_0)} \left| G_\sigma(r) * \frac{\partial}{\partial r} \oint_{r,x_0,y_0} \frac{I(x,y)}{2\pi r} ds \right|$$

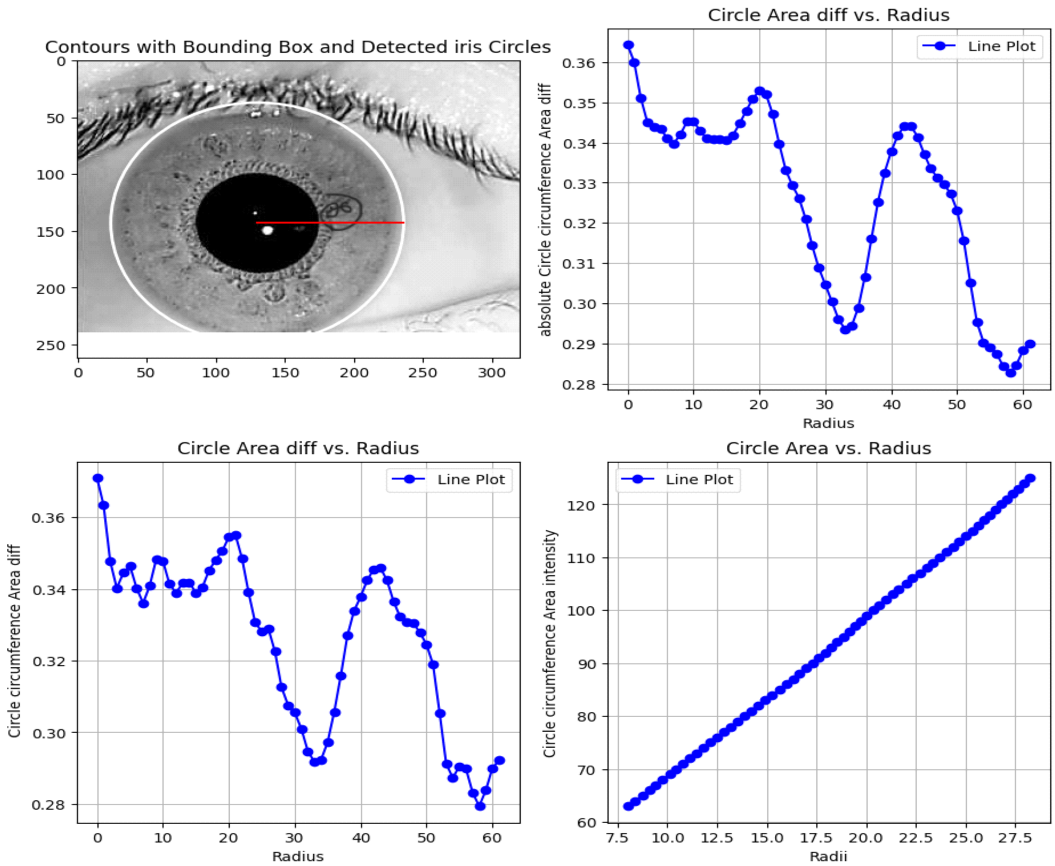
Here, r is radius of circle, 2*pi*r is the circumference of circle, and G sigma(r) is the Gaussian filter at sigma variation for noise reduction and smoothing of image. (ds) is change in contour at radius r. I(x,y) is the image intensity of pixel (x,y).

**Algorithm working explanation:**

With the change of radius r, the contour is changed. And the pixels of contour is summed up. Contour is the circumference of circle with radius r. means, only the circumference pixels at radius are summed up from center at pixel (Xo,Yo). Which the sum result is normalized by 2*pi*r.

After finding the summed normalized area of each contour, the Gaussian filter with sigma=1 is applied. Then, max change between each two consecutive contours is considered as the boundary point of iris circle at radius r from center at pixel (Xo,Yo).

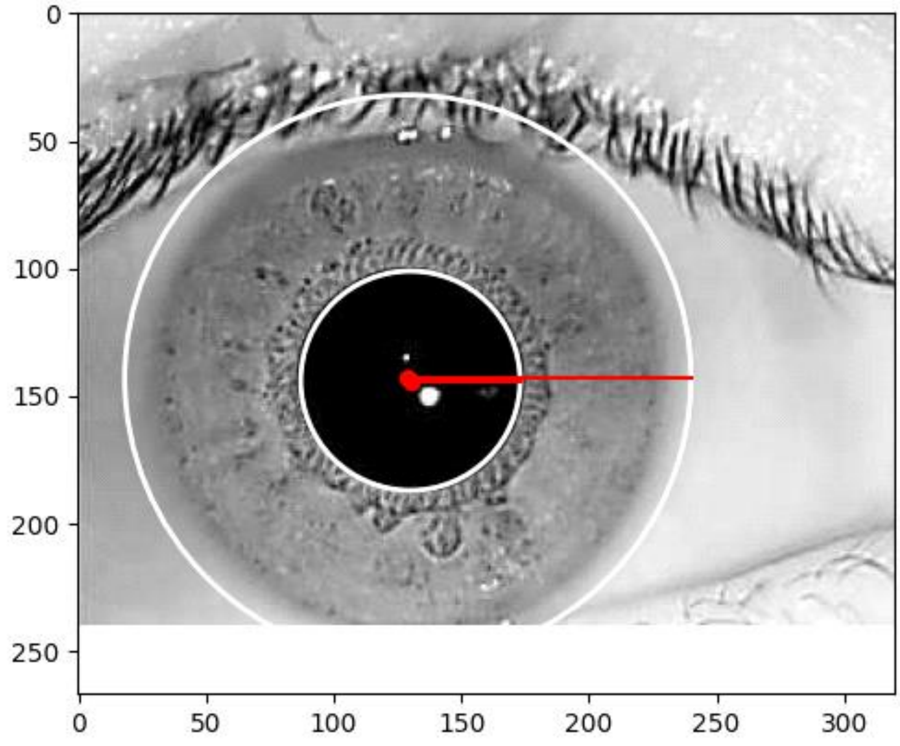
'10_L.bmp', 001 folder image IIT delhi v.10, heart part 2:00 to 3:00 highlighted with dark lines.



- **Iris segmentation**

In Iris segmented from  the image between the iris boundary, called lambic boundary separating sclera from iris part, and pupil boundary, which is 0.2 to 0.5 times the radius of iris boundary radius.

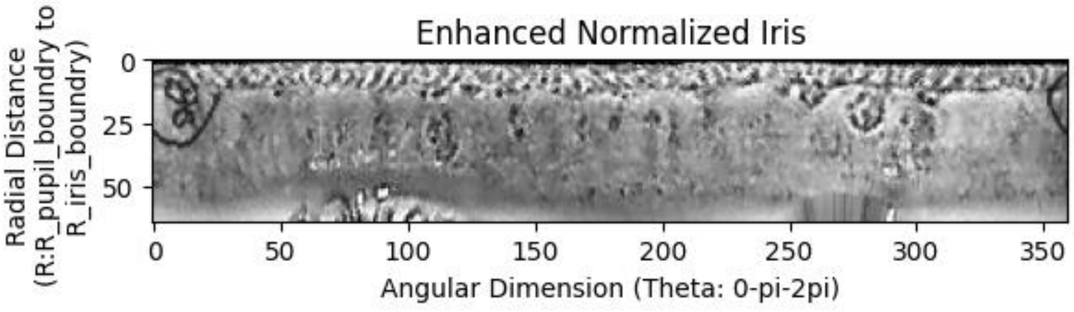
Circle Center pupil: (130, 144), Radius: 43
Best circle found at (x, y) = (129, 143) with radius r = 111
Circle Center: (129, 143), Radius: 111




'10_L.bmp', 001 folder image IIT delhi v.10

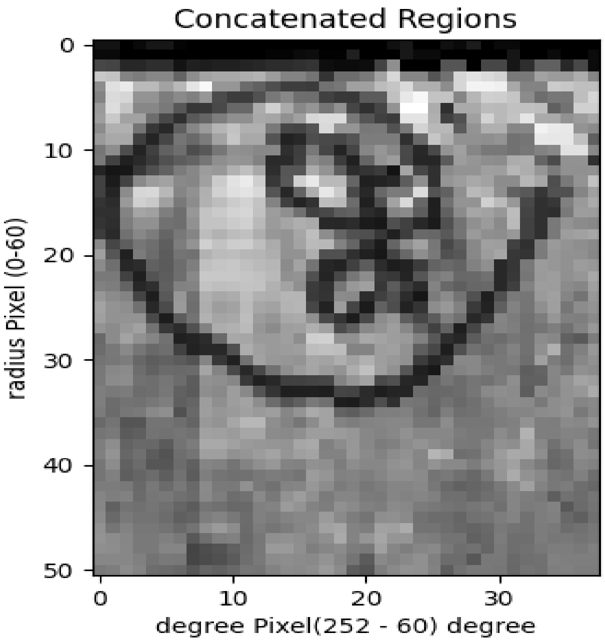
- **Iris Normalization**

Daugman’s rubber sheet algorithm is used for the conversion of iris circular part into rectangle.  In rectangle iris the horizontal lines represents radial distance and vertical columns represents the angular dimension theta. The normalized image enhanced using CLAHE (Contrast Limited Adaptive Histogram Equalization) to improve image quality by increasing the intensity of pixel so that a lot of detail can be seen on image.




## • Feature extraction

The heart region in Iris is 2:00 to 3:15 o'clock in clock wise direction. Which is our Region Of Interest (ROI). From normalized image the ROI is extracted and used only for further processing.




## • Feature Reduction

**PCA**(Principal Component Analysis is used to reduce the features of ROI image. The working of PCA on ROI image is showed below:

```python
from numpy.linalg import eig
#1st )centric to mean for variation stablity in data variables
mean_bycolumn = np.mean(enhanced_iris.T,axis=1)
#print(mean_bycolumn.shape)  #each column mean found

#2nd)  standardazing data
standardized_iris_heart = enhanced_iris - mean_bycolumn
#print(standardized_iris_heart)

#3rd) finding coverence matrix of standardized matrix between each two features to find relations
coverence_matrix = np.cov(standardized_iris_heart.T)
#print(coverence_matrix)
#4th) find eigen vectors and values of coverence matrix,
#eigen vectors will be new dimentions on which data will be floted,
#eigen values will be percentage variation in data by eigen vector
eigen_values, eigen_vectors = eig(coverence_matrix)
#print(f"eigen_values: {eigen_values} \n eigen vectors:  {eigen_vectors}")

#5th) projection of data on new eigen vectors
projected_data = eigen_vectors.T.dot(standardized_iris_heart.T)
print(projected_data.shape)
print(projected_data)
```

```
(38, 50)
```

The data of ROI image is projected on new eigen vectors having sum of eigen values or variation greater than (90%) ninety percent.

```python
In [63]:
#above pca work of  steps in pca library
from sklearn.decomposition import PCA as pca
pcs = pca(n_components = 20)
pcs_vectors = pcs.fit_transform(enhanced_iris)
#print(pcs_vectors.shape)
#print(pcs_vectors)
# Get the eigenvalues (explained variance)
eigenvalues = pcs.explained_variance_
# Get the explained variance ratio and round it to 2 decimal places
explained_variance_ratio = pcs.explained_variance_ratio_
rounded_variance_ratio = [round(val, 5) for val in sorted(explained_variance_ratio, reverse = True )]
if np.sum(rounded_variance_ratio)>0.9:
    print("ok: 90%")
print("Loaded ratio:", rounded_variance_ratio)
print("Eigenvalues (Lambda values):", eigenvalues)
```

```
ok: 90%
Loaded ratio: [0.41087, 0.10283, 0.08076, 0.06335, 0.04735, 0.04273, 0.04149, 0.02925, 0.02661, 0.02153, 0.01973, 0.01707,
0.01386, 0.01231, 0.01168, 0.0096, 0.00797, 0.00687, 0.00525, 0.00427]
Eigenvalues (Lambda values): [53802.19261803 13465.02630725 10575.36590663  8295.37064946
  6200.56604913  5595.17839107  5432.95779553  3830.10119699
  3484.76418326  2819.80360941  2584.13487901  2235.35386117
  1814.49863895  1612.2776585   1530.04366606  1256.86386904
  1043.63038076   899.77617581   687.32741344   559.70241036]
```

- ## Classification

The CNN(Convolutional Neural Network) Model is trained on the feature reduced images to classify images into two groups of normal and abnormal. The system outputs 1 or 0 representing abnormal or normal respectively.[4]

- ## Result Comparison on increase PCA and hidden Layers of Model.

The accuracy rate of model varies with PCAs and Hidden Layers. With 10,20,30 PCAs the accuracy rate is 68, 80,95 percent respectively.  With the increase of Hidden Layers 10,20,30 the accuracy rate is 77.81,95 percent. Thus, the model accuracy is directly proportional to the increase of PCAs and Hidden layers of Neural Network.[4]

## • Literature Review

The perception was done by Ignaz Von Peczely on the iris of an owl. Within the iris of the owl, there are dull spots which already did not exist. After the owl was cured, the dull spots are gone. (Frank, Ferreira and & Pellow 2013).
A study by Nils Lilijequist was done on a patient with lymph disorder. The patient was observed before and after treatment many changes were found on the iris of the patient. (Frank, Ferreira and & Pellow 2013).
Research was also done on iris dataset using CLAHE for contrast enhancement, PCA and CNN for classification with highest rate of success (92.5%) (Putra, et al. 2020).
Research was also done on iris dataset. Where ROI (Region Of Interest) is extracted, equalized and CNN model is trained. The accuracy rate achieved was (81%).(P.K and P.K 2020).
Research was also done on iris dataset. Where ROI region is extracted, equalized, and Canny edge detection is applied. PCAs is applied on canny edge images, and CNN model is trained. The accuracy rate achieved was (95%).
Research was also done on iris dataset, Where on ROI images the classification of images to be normal and abnormal heart based on white and black ratio. The training process concluded that if the image had ratio range less than (0.25) white ratio and more than (0.725) black ratio. It is considered abnormal.[]

# Chapter 2

# Requirement Specification

## 2.1Functional Requirements

1**. Data Preprocessing**
Loading Data: The system needs to load the iris images from IIT Delhi v1.0 dataset.

**2. Image segmentation**

Implementation of Daugman Algorithm: Image normalization: The system needs to normalize the iris image using a Daugman rubber sheet algorithm and display a rainbow shell in a fixed coordinate system. The horizontal line indicates the radius of the iris, and a part of the iris is displayed in the vertical line. The radius of the rainbow shell, the border of the pupil, and the limit of the rainbow shell at the square corner.

**3 Feature Extraction**

Region of Interest (ROI) Extraction: The system should extract features only from the region from 2:00 to 3:15 of the normalized iris image, which is the heart part of the iris.

Feature Representation: The features extracted from the specified iris ROI should be represented in a format suitable for further processing.

**4. Reduce Symptoms Principal Component Analysis (PCA):**

In PCA, the system needs to reduce the dimensionality of the extracted features by retaining only the most significant components to reduce computational complexity and improve classification accuracy.

**5. Classification**

CNN -based classification: The system uses a neural network in adulthood (CNN) to classify reduced signs for each image in abnormal and normal classes to detect the presence of cardiovascular disease.

Threshold: The system needs to apply a threshold on the CNN output to make a final binary decision on the presence or absence of heart disease.

**6. Data Processing and Storage**

Data Management: The system must process the dataset and correctly apply all steps of processing for each image.
Results Storage: The system must store the segmented images, normalized images, extracted features, PCA transformed features, and classification results for further analysis.

**7. Usability of the system**

Interface (optional): If a user interface is included, it should be easy to interact with the system, including loading images, viewing processing, and displaying classification results.
Scalability: The system must be designed to handle an ever-increasing number of images and advanced features such as detection of additional diseases or regions of interest.

**8. Testing and testing**

Cross-validation: Cross-validation techniques are implemented to verify the performance of the CNN model and ensure reliable classification.
Performance Metrics: The system should evaluate the model's accuracy, precision, recall, F1 score, and other relevant metrics to assess its effectiveness in detecting heart disease.

**9. Documents and reports**

Record: The system must record all stages of preliminary processing, functions and classifications.
Reporting: The system should generate detailed reports including graphical representations of segmented images, normalized images, feature space, and classification results.

# 2.2 Non functional Requirements

**1. Performance**
Processing time: The system should be able to perform pre-processing, segmentation, normalization, feature extraction and classification of an iris image within a certain period of time, ideally less than 2 seconds per image. Scalability: The system must be able to handle growing image volumes and support datasets up to 10,000 images without significant performance degradation.

**2. Accuracy**
Classification accuracy: The CNN model must achieve a minimum of 90% accuracy in detecting heart disease from iris images.
Segmentation accuracy: Daugman's algorithm must accurately segment the iris with at least 95% accuracy in detecting the iris border.

**3. Ease of use**
Ease of use: The system interface, if any, should be intuitive and require minimal user training to use. Command line operations should have clear instructions and examples. Error handling: If an image cannot be processed or the system crashes, the system should provide a clear error message and recommended solution.

**4. Reliability**
System accessibility: Systems often need to access and use because of high availability of 99.5 % or more. Data Consistency: The system must ensure that no data is lost during processing, and all image data and results must be recorded.

**5 Security**
Data Security: The system must ensure that all imaging data and results are stored and transmitted securely, using the necessary encryption protocols to protect sensitive health information. Access Control: Role-Based Access Control (RBAC) should be implemented to ensure that only authorized personnel have access to the system and the data it processes.

**6. Support Modularity**
The system should be modular, allowing individual components (e.g., segmentation, feature extraction, classification) to be updated or replaced without affecting the entire system. Documentation: The system should be well documented, with clear instructions on how to maintain, troubleshoot, and update it.

**7. Compatibility**
Platform compatibility: The system must be compatible with major operating systems such as Windows, Linux, and macOS. Software Dependencies: The system must have clearly defined software dependencies and be able to work with common versions of those dependencies (Python, TensorFlow, etc.).

**8. Portability**
Deployment: The system should be easily deployed on different machines or servers with minimal configuration changes.
Cloud Compatibility: The system should be designed to be deployable on cloud platforms if required.

**9. Auditability**
Logging: The system should log all significant actions, including image processing steps, classification results, and user actions, to facilitate auditing and debugging. Traceability: All system operations must be traceable so that processing steps can be reconstructed from raw data to final results.

**10. User experience**
Response time: The system should provide feedback to the user within a second of an interaction, such as pressing a button or submitting an image for processing. Visual feedback: If you are using an interface, you should provide visual feedback during processing steps, such as a loading indicator or progress bar.

# Chapter 3

# Project Design

The subsequent sections of the Software Requirements Specifications (SRS) doc- ument offer a thorough summary of the Heart Disease Detection Uisng Iriodology. This section provides guidance and valuable information about the system and the SRS document to help the reader understand and navigate through both. These subsections contain precise software requirements that are adequate for designers to construct a system that fulfils those requirements. Moreover, testers can utilise this information to verify that the system adequately meets the given standards. The integration of the system context diagram, use cases, and use case descriptions improves comprehension of the project's requirements and facilitates the effective design and testing of the Heart Disease Detection Uisng Iriodology.

- ***Methodology***

The Agile methodology will be used for the development of the Heart Disease Detection using Iriodolgy.We want to make our project continuously evolvable and we want to make it reusable ,flexible and changeable so that we can add more functionalities and change the existing one and also add more web and machine learning model functionalities . The project is ideally suited for Agile methodology because of its inherent flexibility, adaptability, and capacity to satisfy the dynamic requirements of the online auction sector. The Agile methodology prioritizes iterative and in- cremental development, enabling the timely delivery of valuable features in short cycles or sprints. This approach fosters cooperation, openness, and continuous 12

input from all parties involved, including users and administrators, to ensure that the end result aligns with their changing requirements.

- **Agile Methodology**

The Agile methodology employs an iterative and collaborative approach to soft- ware development, in which requirements and solutions evolve through the joint effort of self-organizing and cross-functional teams. The Agile principles place customer satisfaction, the continuous delivery of valuable software, and the ca- pacity to successfully adjust to changes in requirements as top priorities. The Heart Disease Detection using iriodolgy project aims to incorporate the fundamental tenets of Ag- ile methodology to improve communication, flexibility, and the overall quality of the product.

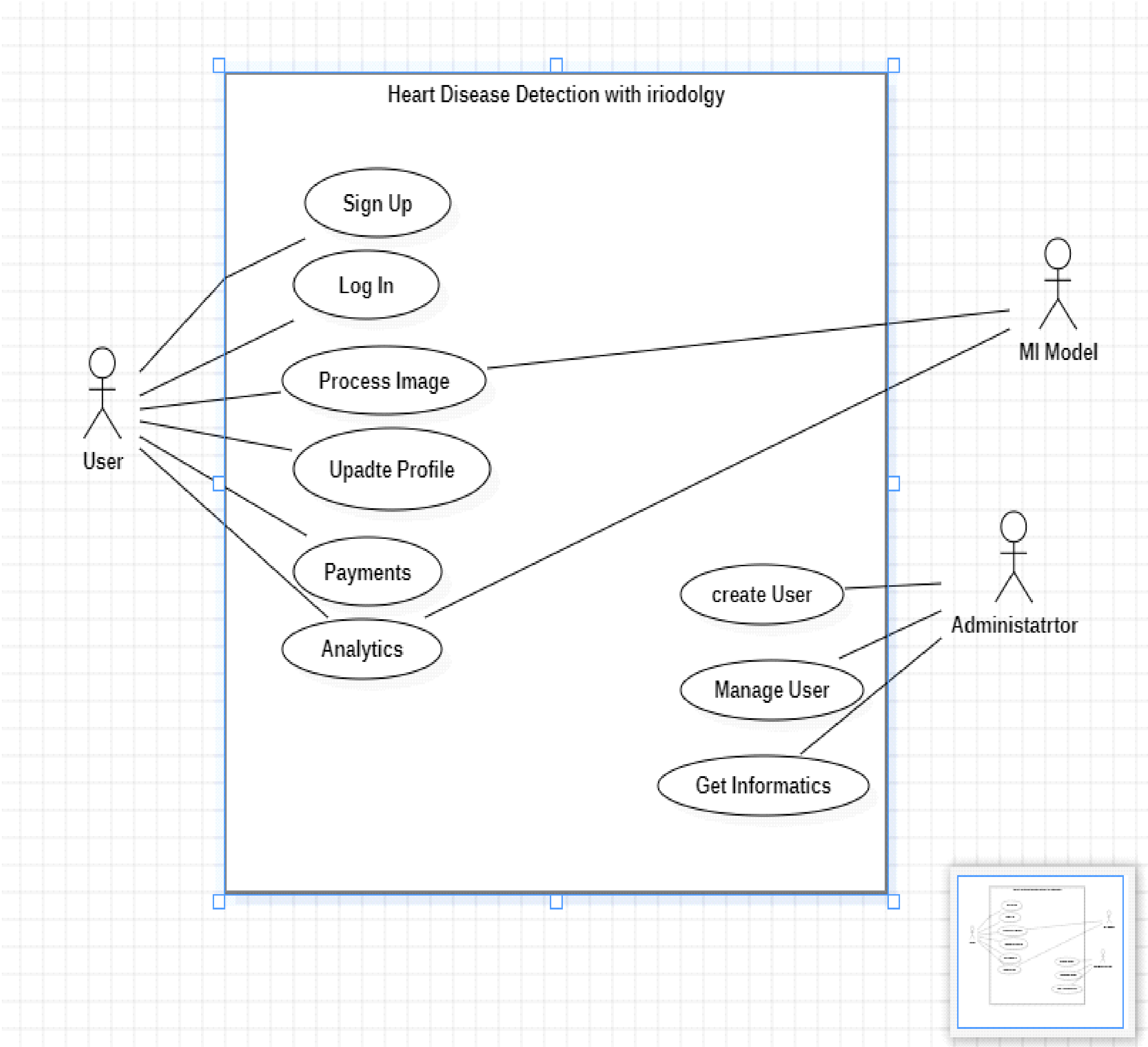
- **Why Agile Model?**

The decision to adopt Agile for the Online Bidding System project is driven by several factors:

- **Flexibility:** The project is will be used for all people from different socities and we think it subject to changes in regulations, market demands, and user preferences. The Agile model allows the project team to adapt to these changes seamlessly and efficiently.
- **User-Centric Approach:** Agile encourages frequent feedback from users and stakeholders, ensuring that the herat disease detection system meets their specific needs and expectations.
- **Continuous Improvement:** By delivering working increments at the end of each sprint, the project team can continuously improve and refine the application based on feedback and user interactions.
- **Iterative Development:** The Agile Model employs an iterative approach to development, wherein requirements are divided into more manageable

  pieces that are produced and delivered over the course of several sprints. This enables the team to prioritize the most crucial features first and make adjustments as necessary in response to user and stakeholder input.

- ## Use Case Diagram


Heart Disease Detection with iriodolgy
Sign Up
Log In
Process Image
Upadte Profile
Payments
Analytics
create User
Manage User
Get Informatics
User
Ml Model
Administatrtor

## Image process use case:

| | |
|---|---|
| **Use-case ID** | Heart Disease Detection using Iriodology |
| **Use-case Name** | Process image |
| **Description** | User select the image and upload the image for processing by clicking button |
| **Primary Actor** | User |
| **Pre-condition** | User must be logged into the system. |
| **Input** | Image |
| **Post-condition** | Image Process result should display |
| **Normal Flow** | • User select image.<br>• Upload it<br>• Press the Process button. |

| | |
|---|---|
| | • Result came back from model.<br>• Result displayed on user interface. |
| **Alternative Flow** | • Invalid Image.<br>• Reattempt the user to upload image. |

## Use case sign up

| | |
|---|---|
| **Use-case ID** | Heart Disease Detection using Iriodology |
| **Use-case Name** | Sign Up |
| **Description** | User registers for a new account to participate in the online bidding system. |
| **Primary Actor** | User |
| **Pre-condition** | User must not already have an account in the system. |
| **Input** | Press the "Sign Up" button |
| **Post-condition** | User account is successfully created. |
| **Normal Flow** | • Open sign up page.<br>• Enter valid email.<br>• Enter valid password.<br>• Confirm password.<br>• Click the "Sign Up" button.<br>• Account created successfully.<br>• Redirects to login page. |

## Use case Log in:

| | |
|---|---|
| **Use-case ID** | Heart Disease Detection using Iriodology |
| **Use-case Name** | Log In |
| **Description** | User logs into the system to access their account and to process image. |
| **Primary Actor** | User |
| **Pre-condition** | User must already have an account in the system. |
| **Input** | Press the ”Log In” button |
| **Post-condition** | User is successfully logged in. |
| **Normal Flow** | • Open login page.<br>• Enter valid email.<br>• Enter correct password.<br>• Click the ”Log In” button.<br>• Logged in successfully.<br>• Redirects to user dashboard. |
| **Alternative Flow** | • User enters invalid email or incorrect password.<br>• Re-enter valid email and password. |

## Use case Log out

| | |
|---|---|
| **Use-case ID** | Heart Disease Detection using Iriodology |
| **Use-case Name** | Log Out |
| **Description** | User logs out of the system to secure their account. |
| **Primary Actor** | User |
| **Pre-condition** | User must be logged into the system. |
| **Input** | Press the ”Log Out” button |
| **Post-condition** | User is successfully logged out. |

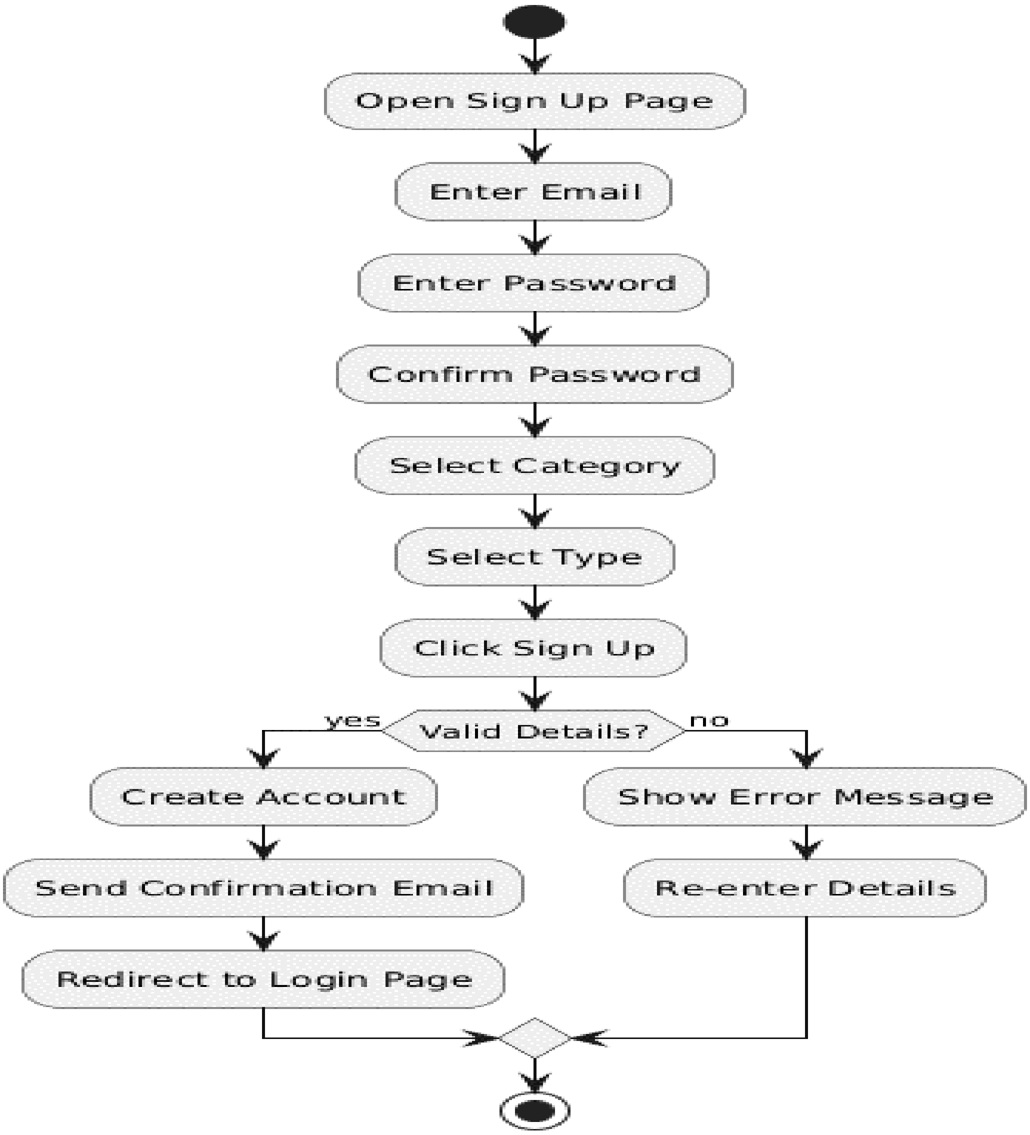
| | |
|---|---|
| **Normal Flow** | • Locate the ”Log Out” button in the user interface.<br>• Click the ”Log Out” button.<br>• Logged out successfully.<br>• Redirects to the home page. |

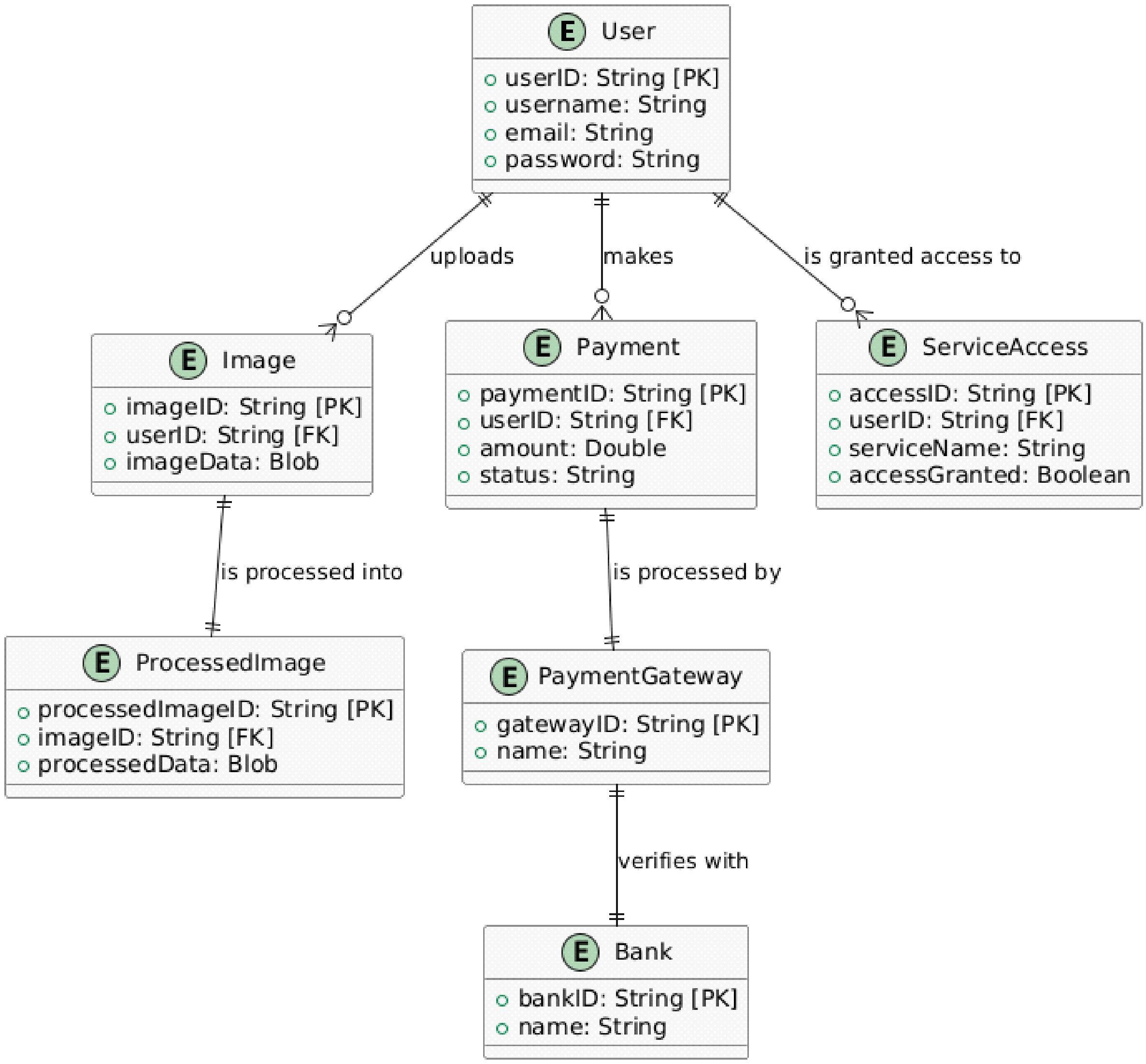
Use case Payment Processing

| | |
|---|---|
| **Use-case ID** | Heart Disease Detection using Iriodology |
| **Use-case Name** | Paymets Processing |
| **Description** | Users will have to pay for services through checkout card |
| **Primary Actor** | User |
| **Pre-condition** | User must be logged into the system . |
| **Input** | Enter credit card number and press the checkout button |
| **Post-condition** | User payment record should be stored into database |
| **Normal Flow** | • User log in.<br>• Click on button to go Premium.<br>• Enter the credit card Details.<br>• Press the checekout button.<br>• User promoted to premium level. |
| **Alternative Flow** | • User enter invalid credit card number p.<br>• Enter a valid credit card number. |

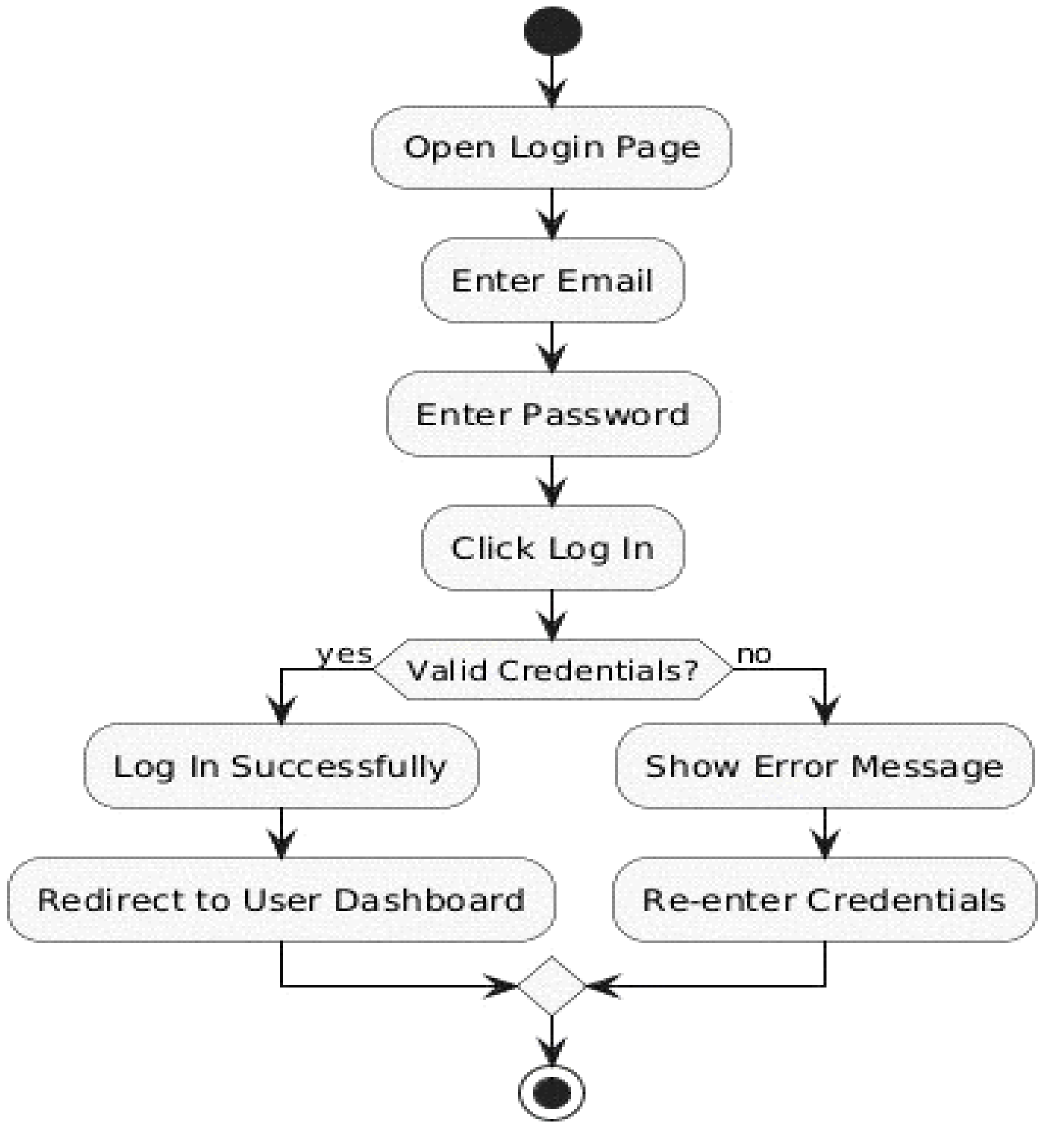
Activity diagrams:

Activity diagram for Sign Up use case:

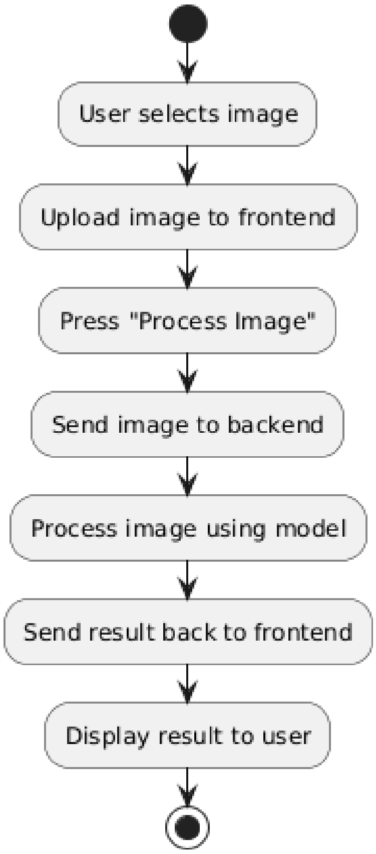


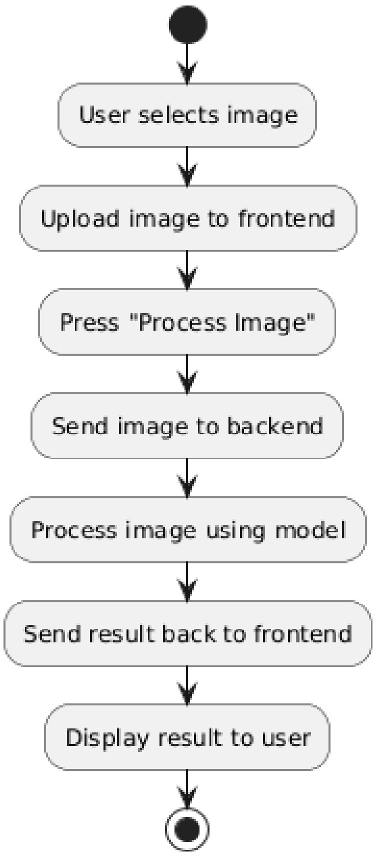

User
userID: String [PK]
username: String
email: String
password: String
uploads
makes
is granted access to
Image
imageID: String [PK]
userID: String [FK]
imageData: Blob
Payment
paymentID: String [PK]
userID: String [FK]
amount: Double
status: String
ServiceAccess
accessID: String [PK]
userID: String [FK]
serviceName: String
accessGranted: Boolean
is processed into
is processed by
ProcessedImage
processedImageID: String [PK]
imageID: String [FK]
processedData: Blob
PaymentGateway
gatewayID: String [PK]
name: String
verifies with
Bank
bankID: String [PK]
name: String

Activity Diagram for Log In use case:

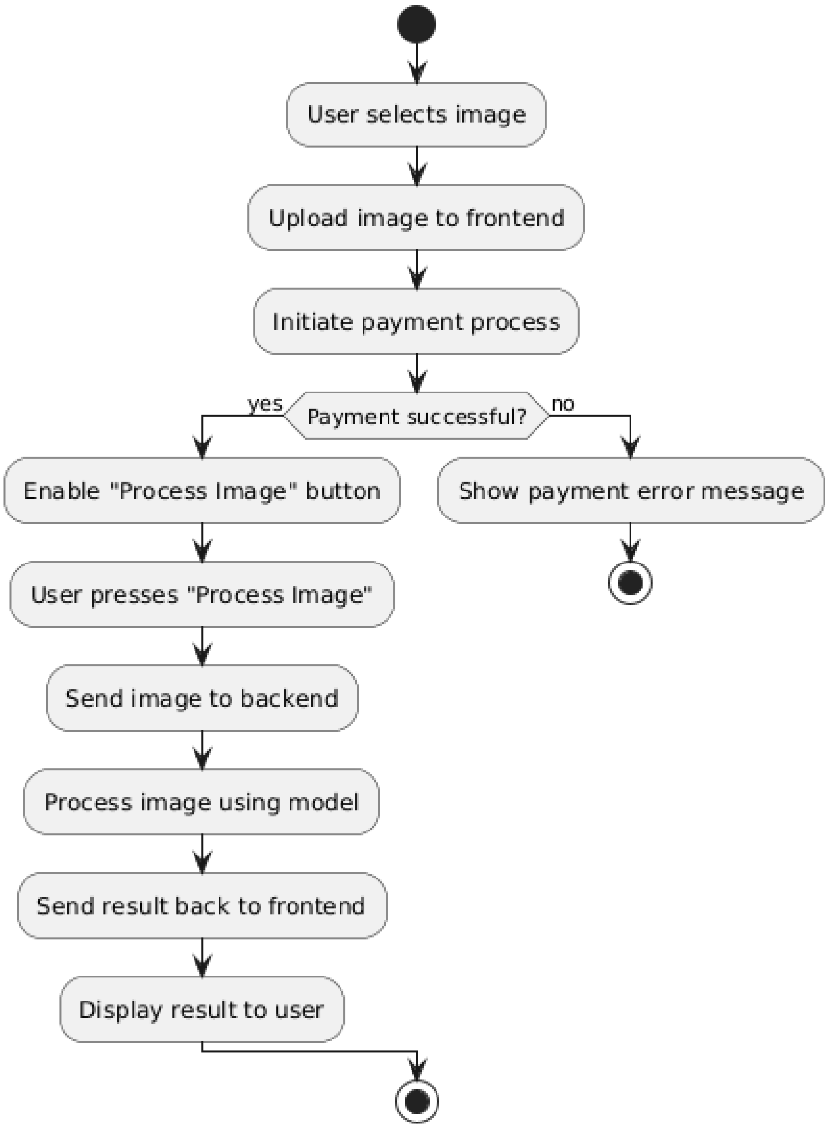



Activity Diagram for Image Prrocessing use case:

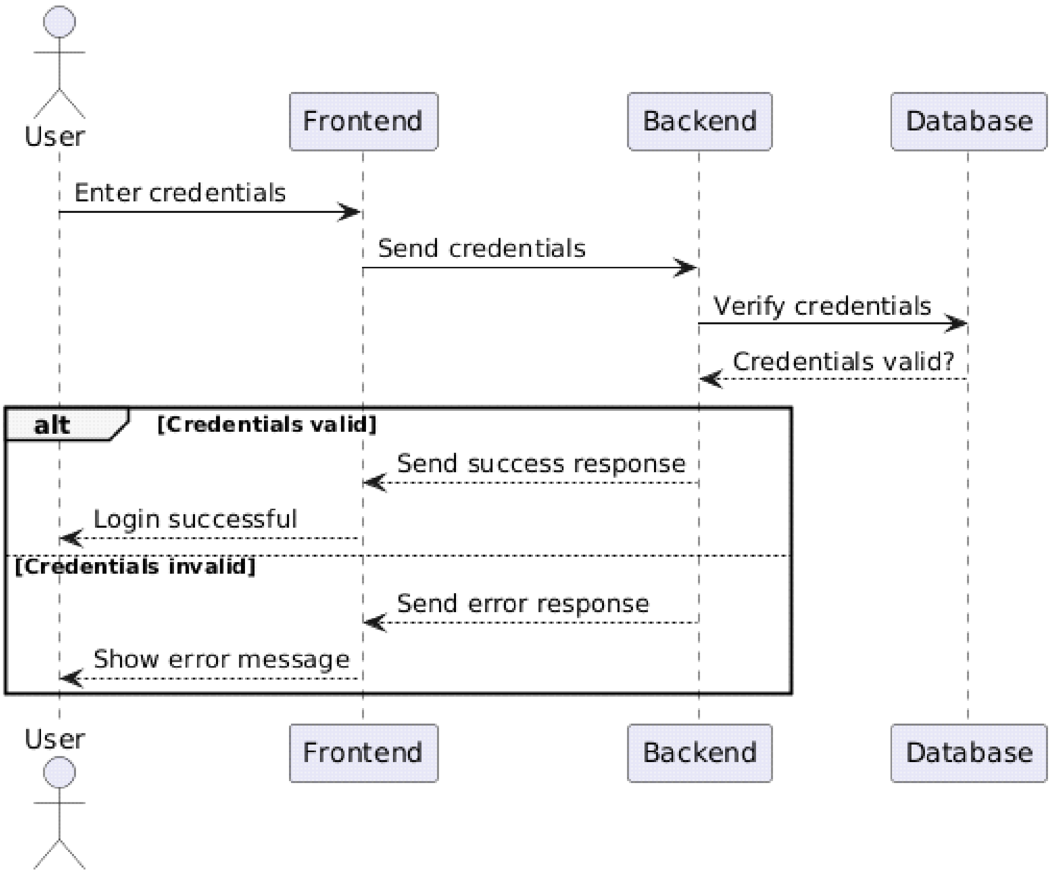

User selects image
Upload image to frontend
Press "Process Image"
Send image to backend
Process image using model
Send result back to frontend
Display result to user



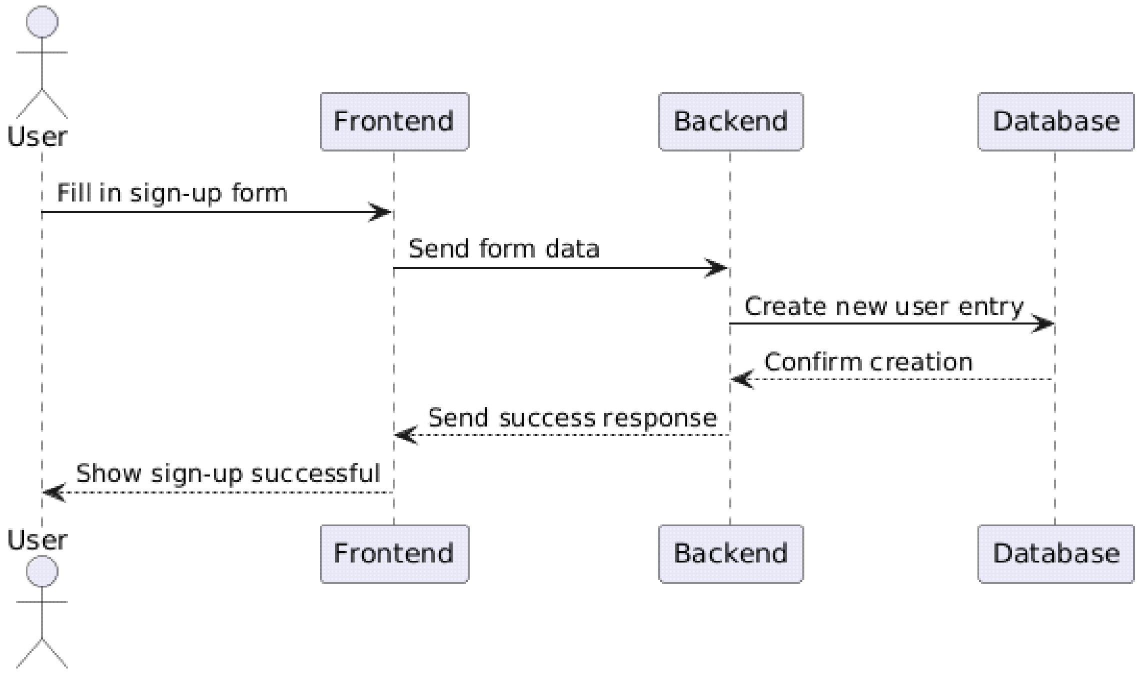

Activity Diagram for Payment Porcessing:



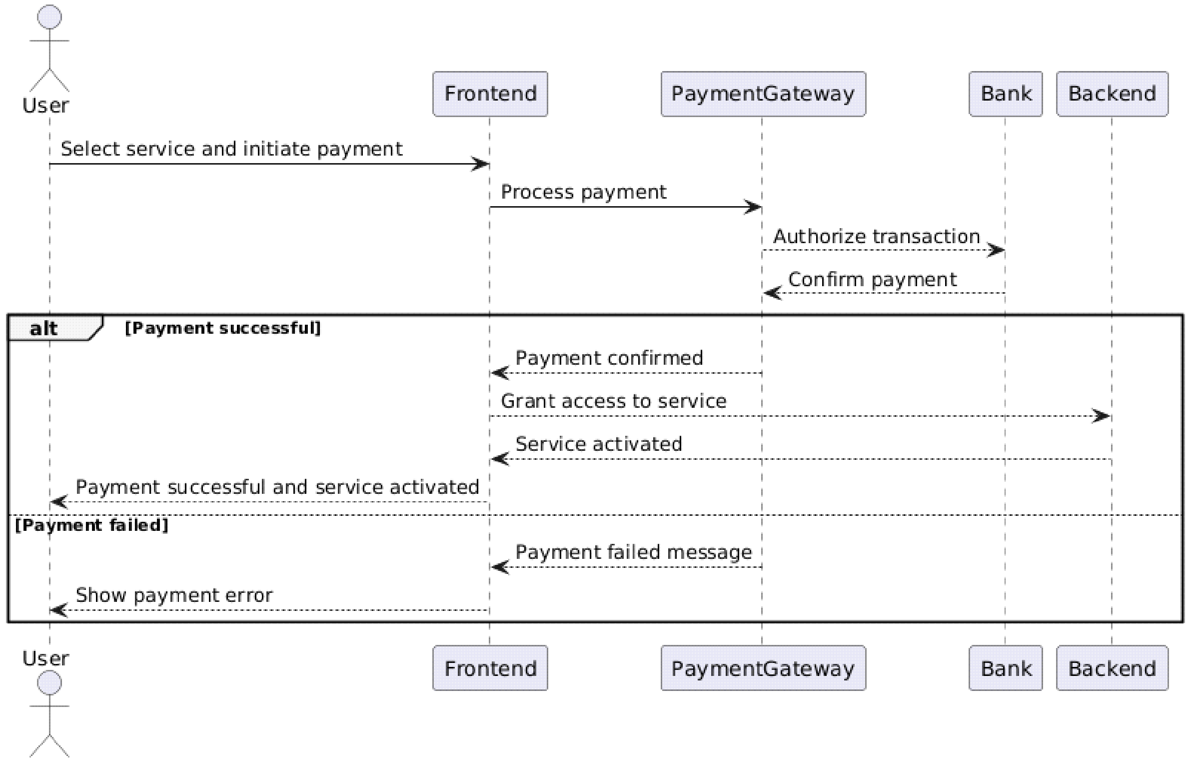

Sequence diagram for Log In:



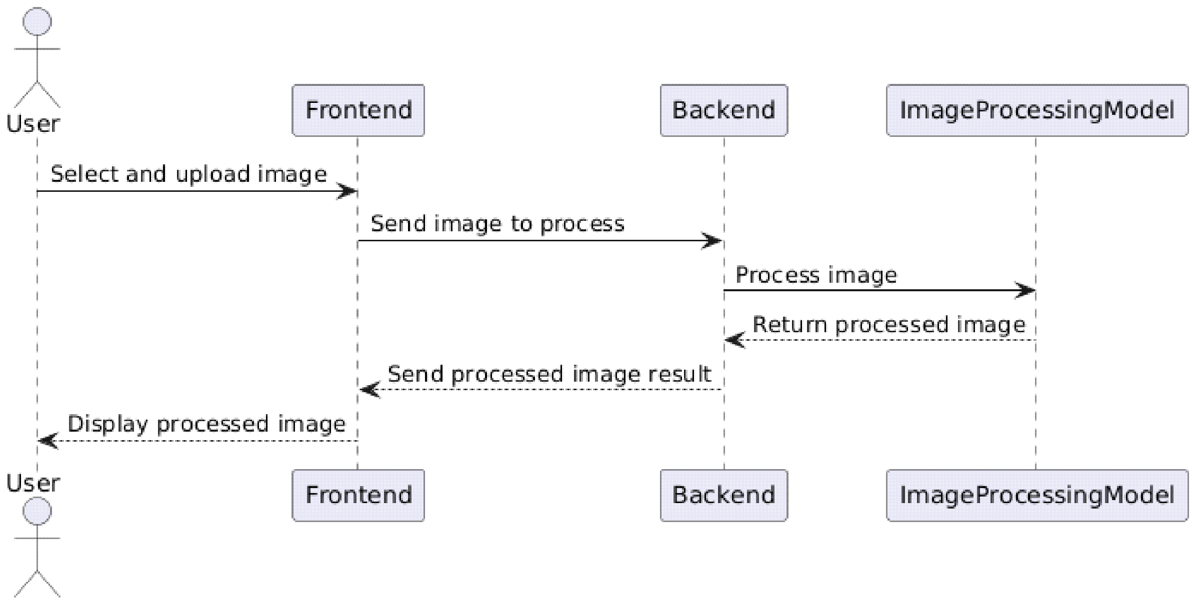

Sequence Diagram for Sign Up use case:



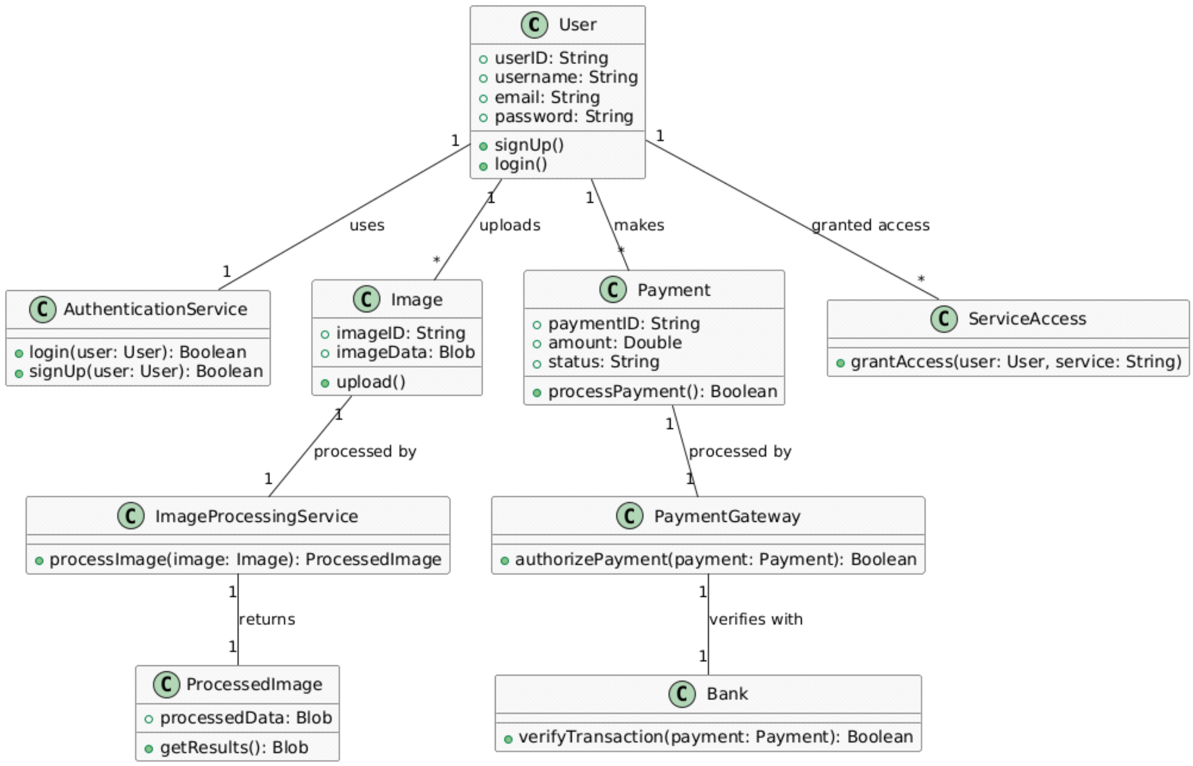

Sequence Diagram for Payment Processing use case:



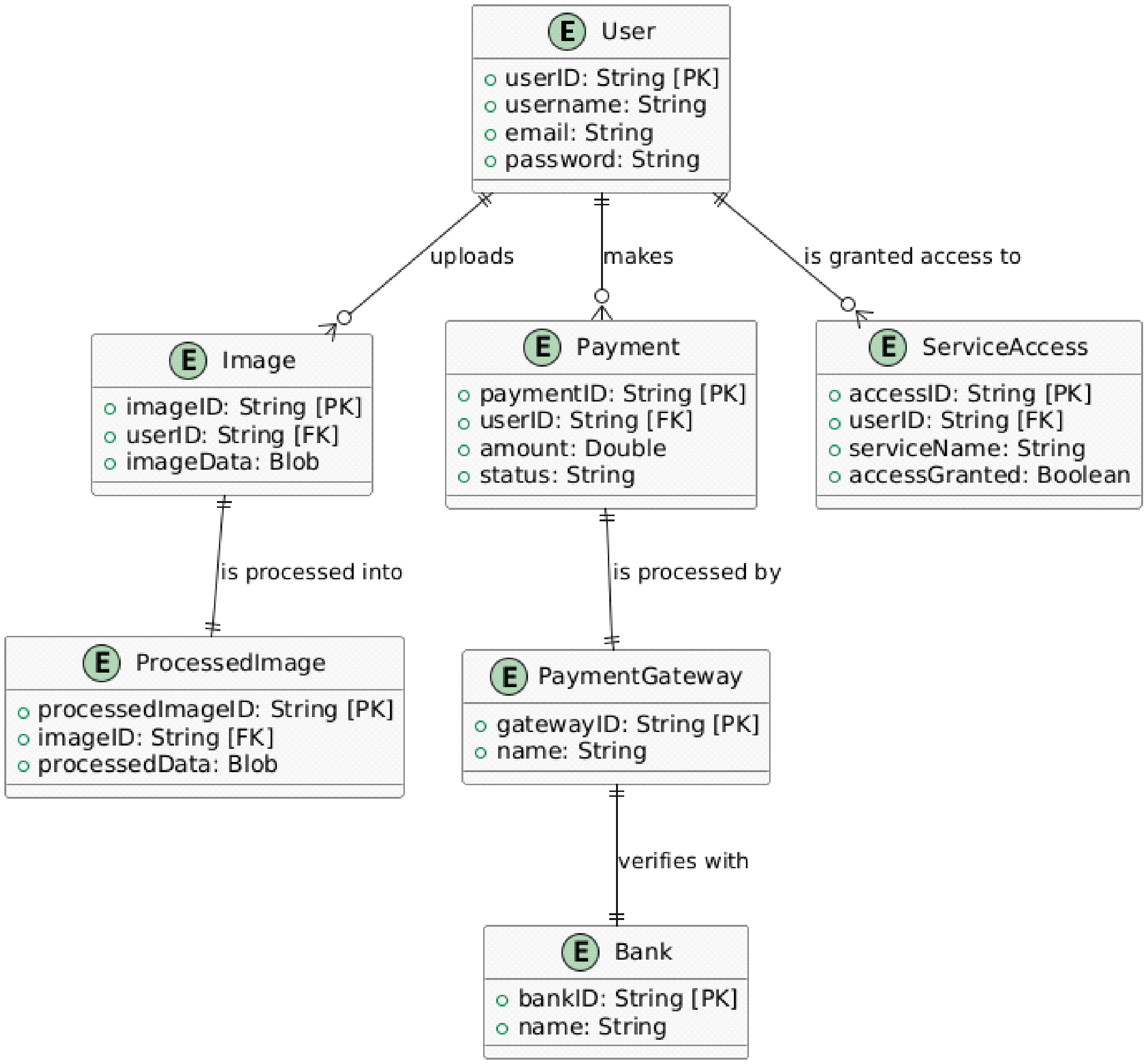

Sequence diagram for image processing:



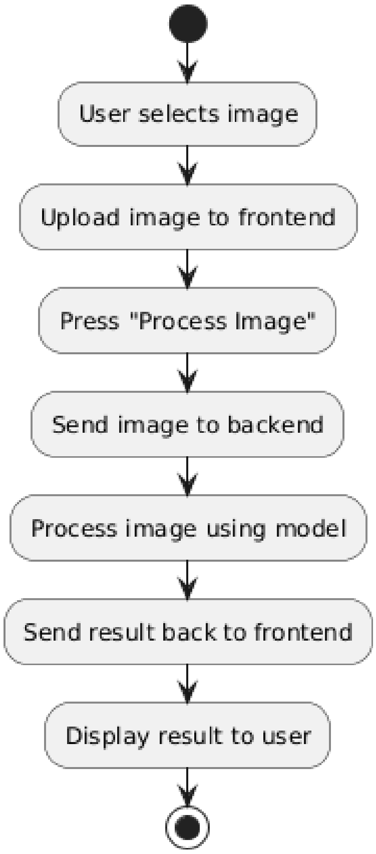

# Class Diagram:

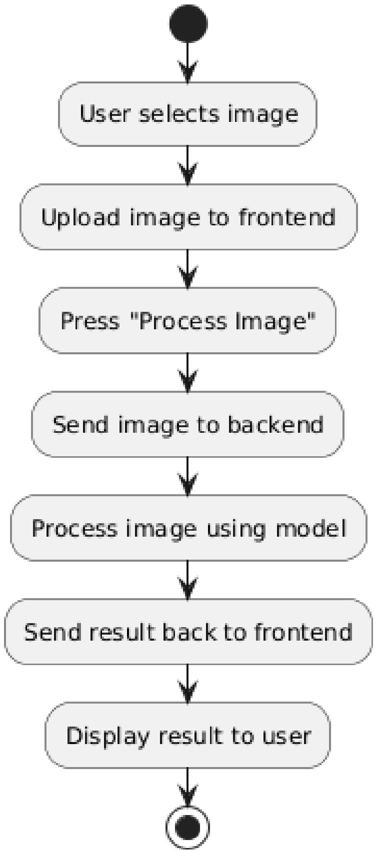

User
userID: String
username: String
email: String
password: String
signUp()
login()
uses
uploads
makes
granted access
AuthenticationService
login(user: User): Boolean
signUp(user: User): Boolean
Image
imageID: String
imageData: Blob
upload()
Payment
paymentID: String
amount: Double
status: String
processPayment(): Boolean
ServiceAccess
grantAccess(user: User, service: String)
processed by
processed by
ImageProcessingService
processImage(image: Image): ProcessedImage
PaymentGateway
authorizePayment(payment: Payment): Boolean
returns
verifies with
ProcessedImage
processedData: Blob
getResults(): Blob
Bank
verifyTransaction(payment: Payment): Boolean

# ER diagram :



User
userID: String [PK]
username: String
email: String
password: String
uploads
makes
is granted access to
Image
imageID: String [PK]
userID: String [FK]
imageData: Blob
Payment
paymentID: String [PK]
userID: String [FK]
amount: Double
status: String
ServiceAccess
accessID: String [PK]
userID: String [FK]
serviceName: String
accessGranted: Boolean
is processed into
is processed by
ProcessedImage
processedImageID: String [PK]
imageID: String [FK]
processedData: Blob
PaymentGateway
gatewayID: String [PK]
name: String
verifies with
Bank
bankID: String [PK]
name: String






# Chapter 4

## Implementation and Evaluation

- ### Development Stages

- ### Technology Stack

The Heart Disease Detection Uisng Iriodolgy makes use of a modern technological stack to guarantee performance, scalability, and maintainability. The employed technologies are:

- *Frontend:*

**React.js:** A JavaScript library for building user interfaces.
**HTML5/CSS3:** For structuring and styling web pages.
**JavaScript/ES6:** For client-side scripting.

- *Backend:*

**Node.js:** A JavaScript runtime for server-side development. **Express.js:** A web framework for building the backend API. **MongoDB:** A NoSQL database for data storage.

- *Development Tools:*

**Postman:** API development and testing.



**VS Code:** Integrated Development Environment (IDE).

## • System Architecture

The Heart Disease Detection Using Iriodology makes use of a modern technological stack to guarantee performance, scalability, and maintainability. The employed technologies are:

- *Model:*

Represents the data structure and business logic, implemented using Mon- goDB.

- *View:*

The user interface, built with React.js components.

- *Controller:*

Handles the application's business logic and user requests, implemented using Express.js.

## • Database Design

The database design employs MongoDB to manage the application's data. Key collections include:

- *Users:*

Stores user information, such as userId, email, password, and role.

- *Monthly payemsts:*

Contains details of users who have have buyed monthly services.

## • API Endpoints

The system provides a RESTful API to facilitate communication between the frontend and backend. Key endpoints include:

- *User Routes:*

**POST /api/users/register:** Registers a new user.

**POST /api/users/login:** Authenticates a user
.
**POST /api/users/logout:** Logs out a user.

**GET /api/users/:userId:** Retrieves user details.

**PUT /api/users/:userId:** Updates user profile information.

- *Image Processing:*

**POST /api/imageprocessing:** process image for processing.

- *Payment processing:*

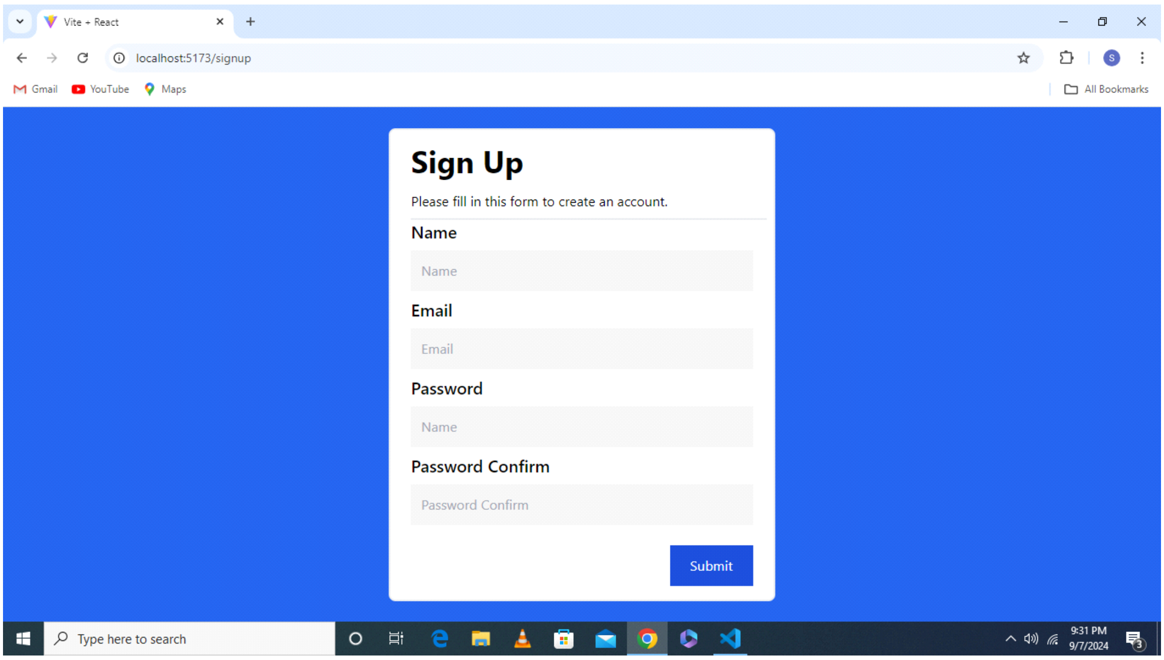
**POST /api/paymentProcessing:** handle payment processing.

- *Analytics page:*

**GET /api/analytics:** get analytics about image process data.

## • User Interface

Sign Up page:



Vite + React
localhost:5173/signup
Gmail
YouTube
Maps
All Bookmarks
Sign Up
Please fill in this form to create an account.
Name
Name
Email
Email
Password
Name
Password Confirm
Password Confirm
Submit
Type here to search
9:31 PM
9/7/2024

Log In page

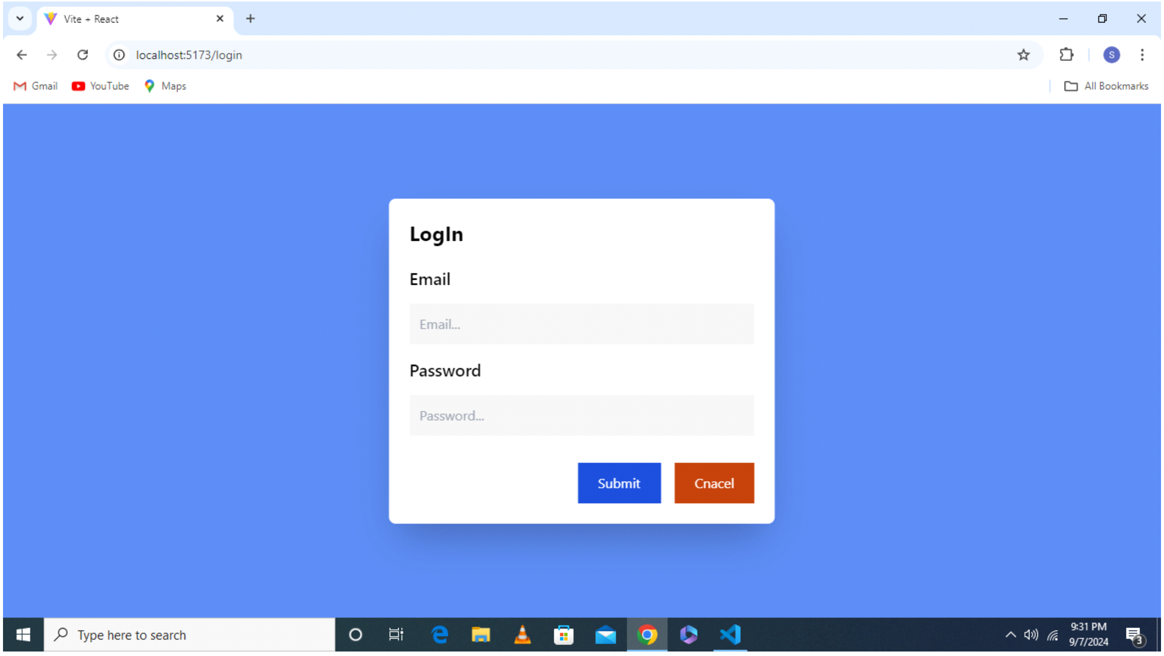
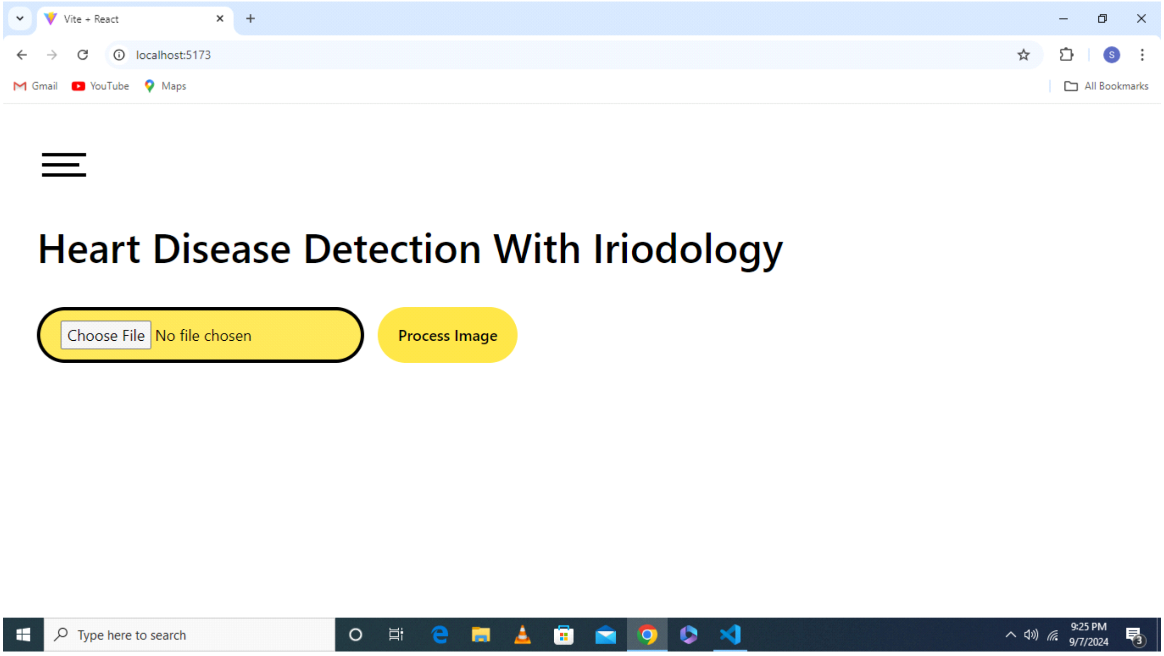
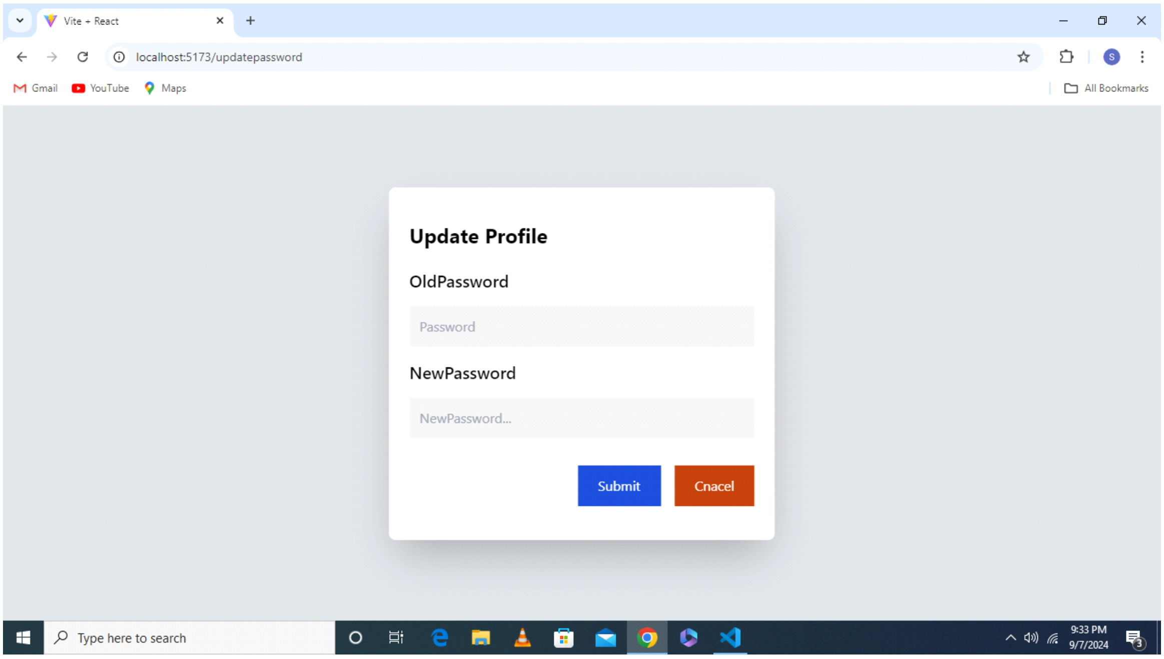


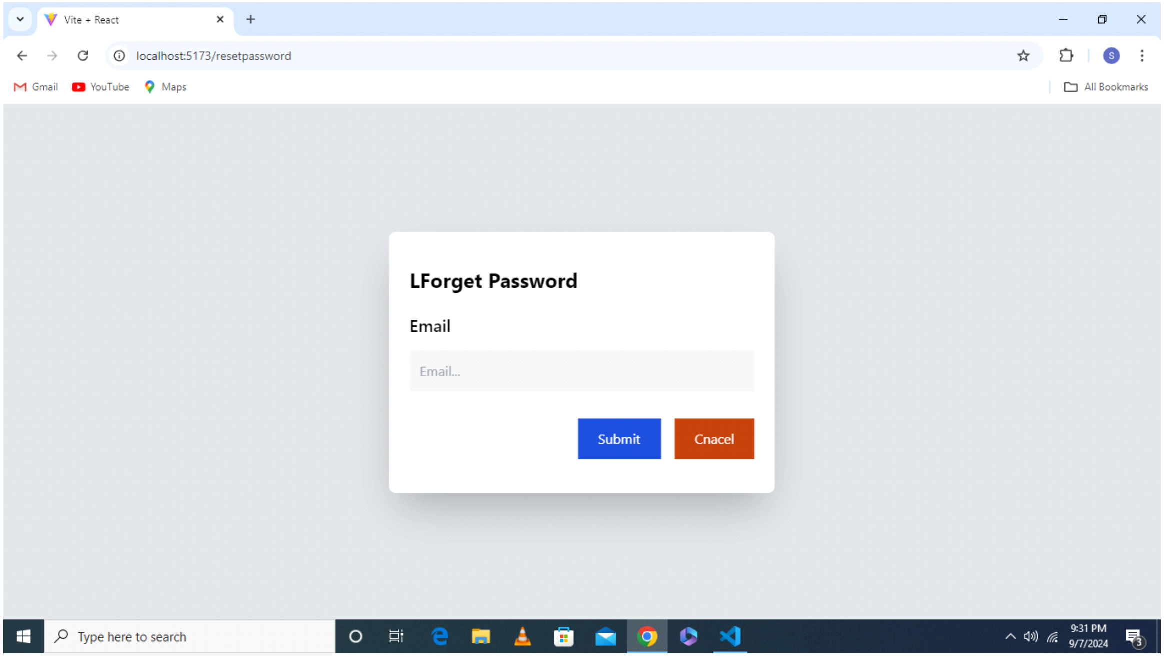

Image processing page



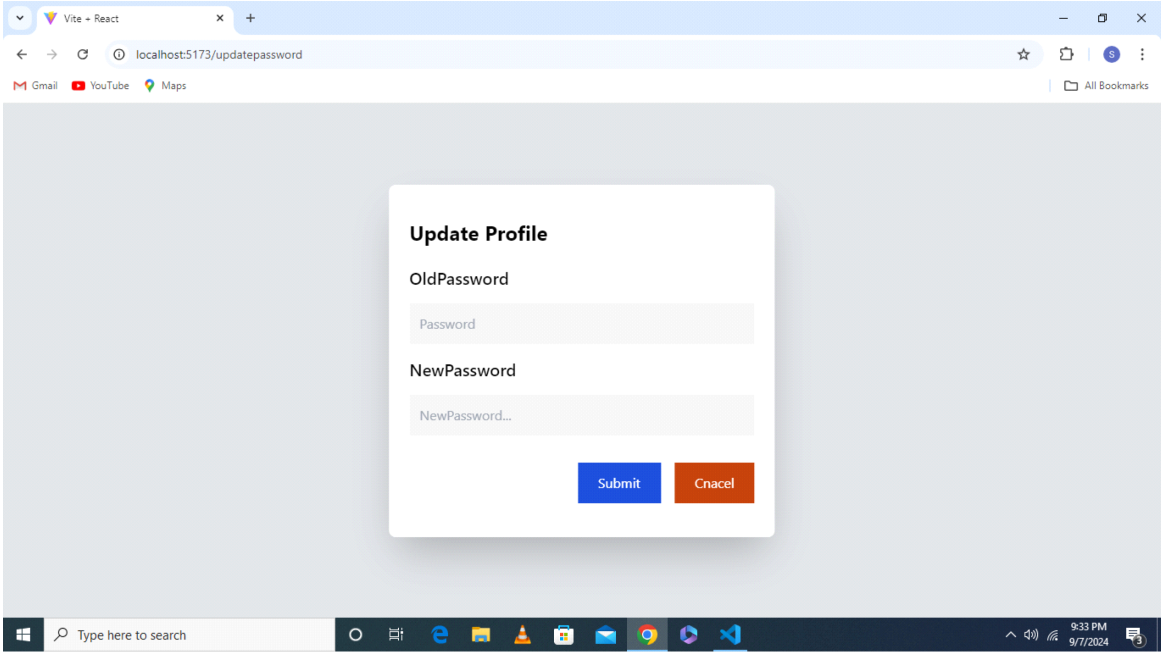

# Update profile



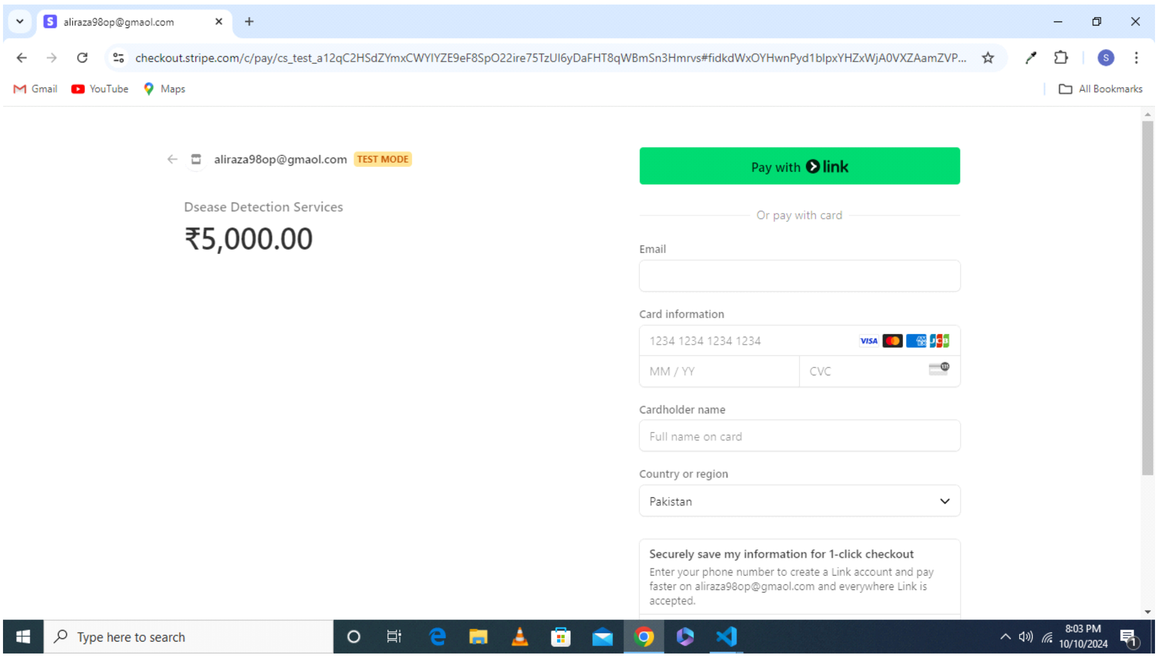
Forget password



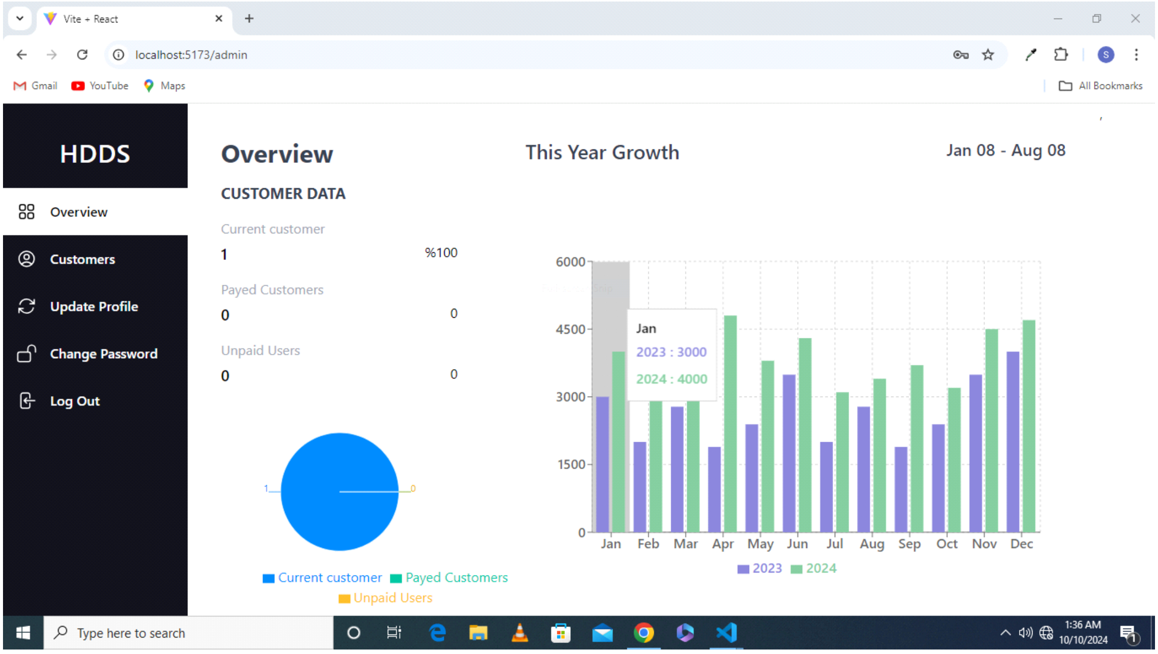

Update user information



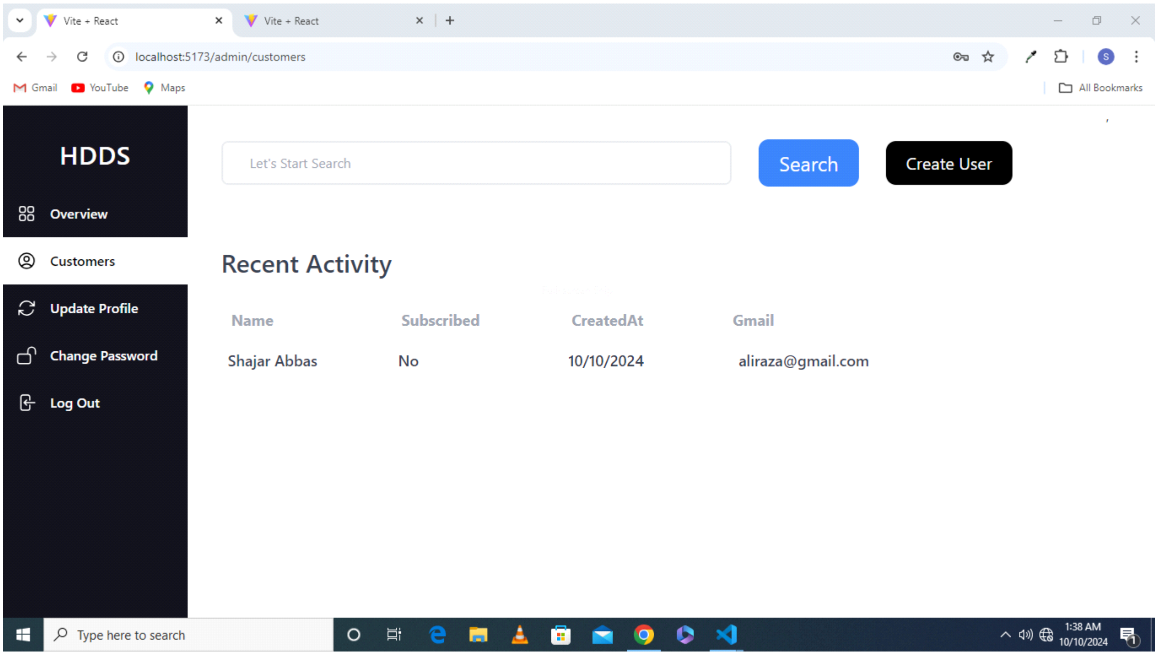

# Stripe Checkout Page:



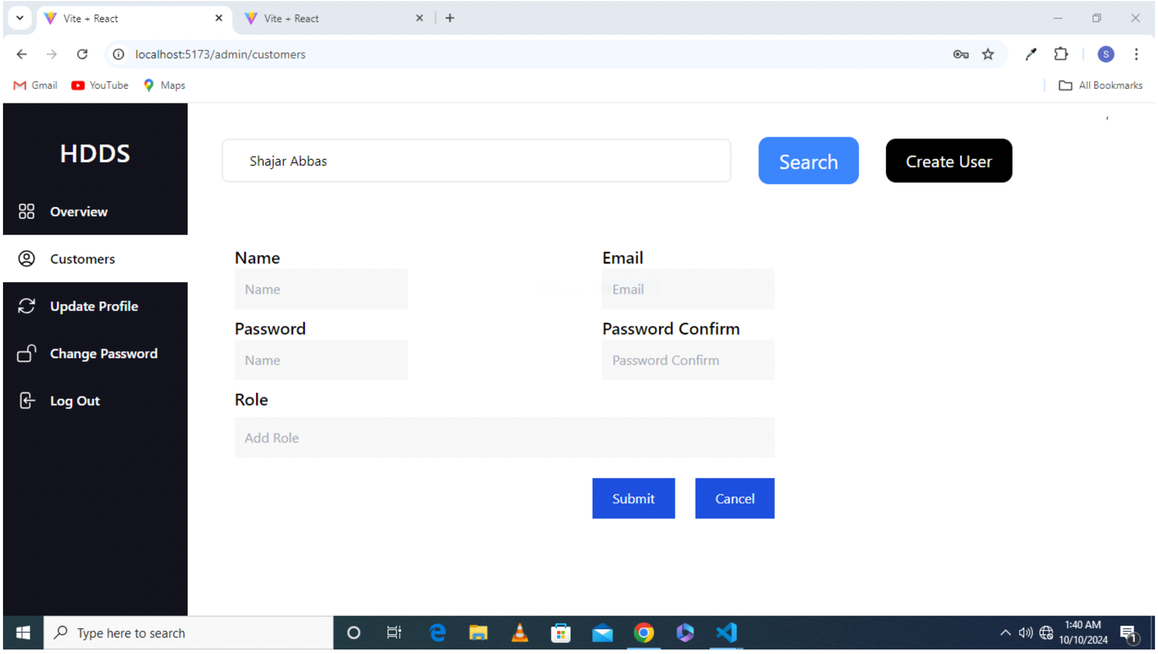
# Admin Overview page:




# Recent logged :




# Create and update customers by Admin:

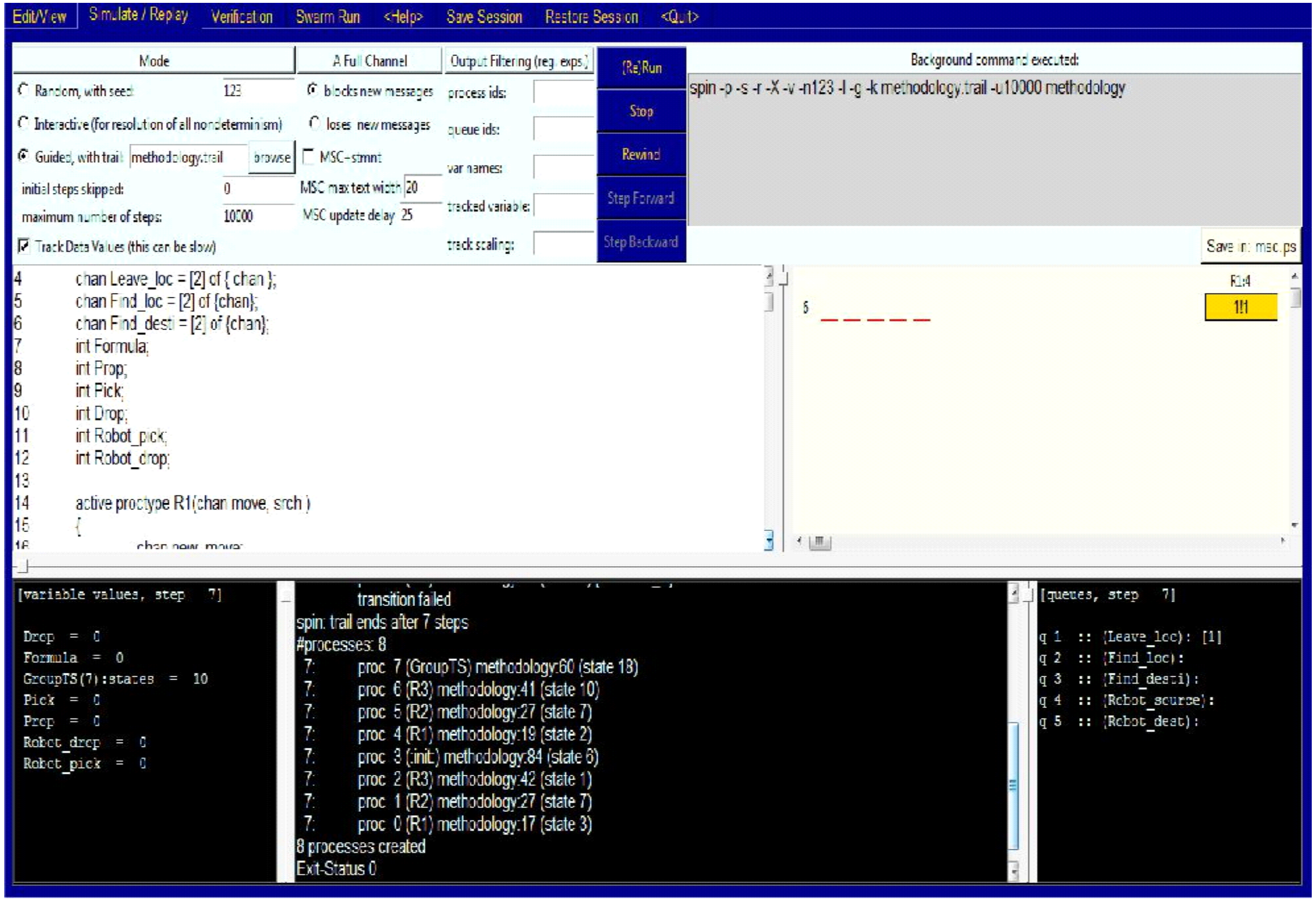



# Chapter 5

# Conclusion & Future Work

12

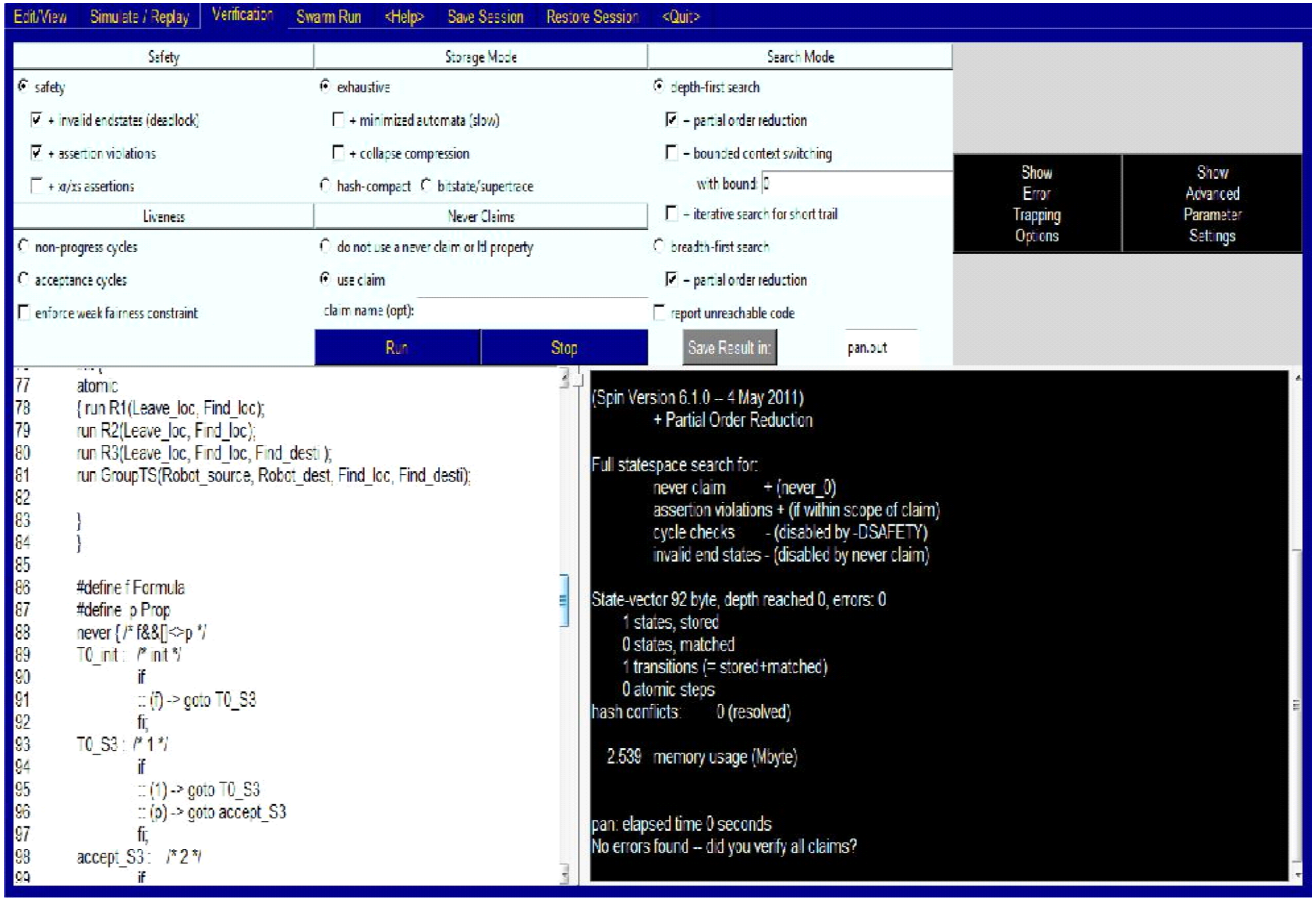

| | |
|---|---|
| Chapter 5. Conclusion & Future Work | 13 |
| | |



Edit/View
Simulate / Replay
Verification
Swarm Run
<Help>
Save Session
Restore Session
<Quit>
Mode
Random, with seed: 123
Interactive (for resolution of all nondeterminism)
Guided, with trail: methodology.trail
browse
initial steps skipped: 0
maximum number of steps: 10000
Track Data Values (this can be slow)
A Full Channel
blocks new messages
loses new messages
MSC-stmnt
MSC max text width 20
MSC update delay 25
Output Filtering (reg. exps.)
process ids:
queue ids:
var names:
tracked variable:
track scaling:
(Re)Run
Stop
Rewind
Step Forward
Step Backward
Background command executed:
spin -p -s -r -X -v -n123 -l -g -k methodology.trail -u10000 methodology
Save in: msc.ps
chan Leave_loc = [2] of { chan };
chan Find_loc = [2] of {chan};
chan Find_desti = [2] of {chan};
int Formula;
int Prop;
int Pick;
int Drop;
int Robot_pick;
int Robot_drop;
active proctype R1(chan move, srch )
{
[variable values, step 7]
Drop = 0
Formula = 0
GroupTS(7):states = 10
Pick = 0
Prop = 0
Robot_drop = 0
Robot_pick = 0
transition failed
spin: trail ends after 7 steps
#processes: 8
7: proc 7 (GroupTS) methodology:60 (state 13)
7: proc 6 (R3) methodology:41 (state 10)
7: proc 5 (R2) methodology:27 (state 7)
7: proc 4 (R1) methodology:19 (state 2)
7: proc 3 (:init:) methodology:84 (state 6)
7: proc 2 (R3) methodology:42 (state 1)
7: proc 1 (R2) methodology:27 (state 7)
7: proc 0 (R1) methodology:17 (state 3)
8 processes created
Exit-Status 0
[queues, step 7]
q 1 :: (Leave_loc): [1]
q 2 :: (Find_loc):
q 3 :: (Find_desti):
q 4 :: (Robot_source):
q 5 :: (Robot_dest):

Figure 5.1: Result after simulation

| | |
|---|---|
| Chapter 5. Conclusion & Future Work | 14 |
| | |



Edit/View
Simulate / Replay
Verification
Swarm Run
<Help>
Save Session
Restore Session
<Quit>
Safety
Storage Mode
Search Mode
safety
+ invalid endstates (deadlock)
+ assertion violations
+ xr/xs assertions
exhaustive
+ minimized automata (slow)
+ collapse compression
hash-compact
bitstate/supertrace
depth-first search
+ partial order reduction
+ bounded context switching
with bound: 0
+ iterative search for short trail
Show Error Trapping Options
Show Advanced Parameter Settings
Liveness
Never Claims
non-progress cycles
acceptance cycles
enforce weak fairness constraint
do not use a never claim or ltl property
use claim
claim name (opt):
breadth-first search
+ partial order reduction
report unreachable code
Run
Stop
Save Result in:
pan.out
atomic
{ run R1(Leave_loc, Find_loc);
run R2(Leave_loc, Find_loc);
run R3(Leave_loc, Find_loc, Find_desti );
run GroupTS(Robot_source, Robot_dest, Find_loc, Find_desti);
}
}
#define f Formula
#define p Prop
never { /* f&&[]<>p */
T0_init : /* init */
if
:: (f) -> goto T0_S3
fi;
T0_S3 : /* 1 */
if
:: (1) -> goto T0_S3
:: (p) -> goto accept_S3
fi;
accept_S3 : /* 2 */
if
(Spin Version 6.1.0 -- 4 May 2011)
+ Partial Order Reduction
Full statespace search for:
never claim + (never_0)
assertion violations + (if within scope of claim)
cycle checks - (disabled by -DSAFETY)
invalid end states - (disabled by never claim)
State-vector 92 byte, depth reached 0, errors: 0
1 states, stored
0 states, matched
1 transitions (= stored+matched)
0 atomic steps
hash conflicts: 0 (resolved)
2.539 memory usage (Mbyte)
pan: elapsed time 0 seconds
No errors found -- did you verify all claims?

Figure 5.2: Result after verication



## 5.1 Conclusion

A method is presented in this work for modeling the concurrent activities of a group of robots using temporal logics. The speci cations are expressed in LTL formula and an algorithm is provided to model the transition system for the group of robots. Our method is optimal in a computational way as compared to previous methods, in which they constructed a model that captures all group members and their mission speci cation. The main drawback is the complexity of previous models that are time consuming processes.

Our approach is optimal to handle such cases where robots can take an action after con rmation of path availability according to the plan, and in some applications they has practical value where a series of di erent tasks performed by multiple robots in an environment. For some applications including new states and corresponding transitions to the structure of the robotic system may indicate to introducing advance stages or motion commands at some lower level. So the proper way in which the changes of these models are strictly application speci c and we do not consider such details in our work. Assuming that these changes can be implemented in future.

# Appendix A

# Appendix Title Here

Write your Appendix content here.

# Bibliography

[1] http://scribd.com.

# Index

Computer Science, i

sample, 1