



Software Requirement Specification

# PalmSecure - Revolutionizing Transport with Biometric Precision

**Version:** 00.01

**Project Code:** F24-10

**Project Team:**


- Muhammad Talha Bilal (K21-3349)
- Muhammad Hamza (K21-4579)
- Muhammad Salar (K21-4619)

**Internal Supervisor:** Dr. Muhammad Atif Tahir


**Submission Date:** December 11, 2024


Foundation of Advancement of Science and Technology
National University of Computer and Emerging Sciences
Department of Computer Science
Karachi, Pakistan

## Document History

| Version | Name of Person | Date | Description of Change |
|---|---|---|---|
| 00.01 | Talha, Hamza, Salar | Nov 17, 2024 | Added Introduction |
| 00.01 | Talha, Hamza, Salar | Nov 24, 2024 | Added Overall System Description |
| 00.01 | Talha, Hamza, Salar | Dec 1, 2024 | Added External Interface Requirements |
| 00.01 | Talha, Hamza, Salar | Dec 8, 2024 | Added Functional Requirements, Non Functional Requirements |

## Distribution List

| Name | Role |
|---|---|
| Dr. Muhammad Atif Tahir | Supervisor |
| Dr. Ghufran Ahmed | Internal Jury Member |
| Mr. Fahad Hussain | Internal Jury Member |
| Mr. Saad Manzoor | FYP Project Coordinator |

## Document Sign-Off

| Version | Sign-off Authority | Signature | Sign-off Date |
|---|---|---|---|
| 00.01 | Dr. Muhammad Atif Tahir | | December 11, 2024 |

# Contents

**1 Introduction** **3**
1.1 Purpose of Document . . . 3
1.2 Intended Audience . . . 3
1.3 Abbreviations . . . 3
1.4 Document Conventions . . . 4

**2 Overall System Description** **4**
2.1 Project Background . . . 4
2.2 Project Scope . . . 4
2.3 Not In Scope . . . 5
2.4 Project Objectives . . . 5
2.4.1 Comprehensive Competition Network: A Deep Learning Solution 6
2.4.2 Mobile Application: Secure and Efficient Biometric Solution . . . 6
2.4.3 Future Expansion . . . 6
2.4.4 Project Impact . . . 6
2.5 Stakeholders . . . 6
2.6 Operating Environment . . . 7
2.7 System Constraints . . . 8
2.8 Assumptions and Dependencies . . . 9

**3 External Interface Requirements** **10**
3.1 Hardware Interfaces . . . 10
3.2 Software Interfaces . . . 11
3.3 Communication Interfaces . . . 12

**4 Functional Requirements** **13**
4.1 Functional Hierarchy . . . 13
4.2 Use Cases . . . 14
4.2.1 Use Case Diagram . . . 14
4.2.2 Register Palm (Admin Version) . . . 15
4.2.3 Verify Palm . . . 16
4.2.4 Manage System . . . 17
4.2.5 Manage Profiles . . . 18
4.2.6 Delete Profile . . . 18
4.2.7 Reset Profile . . . 19

**5 Non-Functional Requirements** **20**
5.1 Performance Requirements . . . 20
5.2 Safety Requirements . . . 21
5.3 Security Requirements . . . 22
5.4 User Documentation . . . 23

**6 References** **24**

# 1. Introduction

## 1.1. Purpose of Document

The purpose of this document is to provide a detailed and structured outline of the functional and non-functional requirements for the system. It serves as a foundational guide for all stakeholders, including developers, designers, testers, and end-users, ensuring a common understanding of the project's goals, scope, and deliverables. Specifically, this document will:

- Clearly define the objectives and functionalities of the proposed system.
- Serve as a reference for the development, testing, and validation phases of the project.
- Act as a contract between the stakeholders and the development team to align expectations.

## 1.2. Intended Audience

This document is intended for the following stakeholders:

- **Project Team:** Developers, testers, and designers responsible for implementing the system.
- **Supervisors:** Dr. Muhammad Atif Tahir and relevant project reviewers for validation and feedback.
- **Jury:** Academic evaluators and mentors assessing the project's feasibility and compliance with academic standards.
- **Potential Users:** End-users, including employees, administrators, and other stakeholders who will use the system.

## 1.3. Abbreviations

- **KPI:** Key Performance Indicators
- **AI:** Artificial Intelligence
- **CRUD:** Create, Read, Update, Delete
- **API:** Application Programming Interface
- **CNN:** Convolutional Neural Network
- **CCNet:** Comprehensive Competition Network
- **CA:** Competition Attention

### 1.4. Document Conventions

This document follows specific conventions to maintain consistency and clarity:

- **Font Style:** Times New Roman
- **Font Sizes:**
  - **Headings:** 12-point bold
  - **Body Text:** 12-point regular
- **Figures and Tables:** All figures and tables are labeled with descriptive captions.
- **Terminology:** Technical terms and abbreviations are defined in the "Abbreviations" section.
- **Version Control:** Revision history is documented to track changes and updates.

## 2. Overall System Description

### 2.1. Project Background

The transportation industry is increasingly vulnerable to security breaches, unauthorized access, and identity-related crimes. These challenges compromise passenger safety, disrupt operations, and lead to significant financial losses. Traditional verification systems are often prone to inaccuracies, inefficiencies, and health concerns, especially those relying on contact-based methods.
To address these pressing concerns, this project introduces a novel solution: a palmprint verification system. By leveraging the unique features of palmprints—ridges, wrinkles, and minutiae—our approach offers a secure, hygienic, and reliable biometric alternative. The project employs advanced deep learning techniques, specifically the Comprehensive Competition Network approach, to extract multi-order texture and spatial features for unmatched recognition accuracy. This innovation is tailored for the transport sector but is scalable to other industries, paving the way for a secure, efficient, and future-proof identity verification landscape.

### 2.2. Project Scope

The project aims to design and develop an advanced palmprint verification system integrated into a user-friendly mobile application. The system's primary focus is to enhance transport security by enabling reliable, contactless biometric verification.

Key functionalities include:

- **Palmprint Verification System:**
  - Implementation of the Comprehensive Competition Network for palmprint recognition.
  - Utilization of local and public datasets for training and testing.
  - Continuous performance evaluation to refine system accuracy and robustness.

- **Mobile Application:**
  - Seamless interface for capturing and verifying palmprint data.
  - Real-time passenger identity verification to prevent identity theft and fraud.
  - Integration with transport infrastructure for enhanced operational efficiency.
- **Security and Scalability:**
  - Robust data handling to ensure privacy and accuracy.
  - Expandable design to support other sectors like banking, healthcare, and law enforcement.

Deliverables include a fully functional mobile application and backend system, setting a new standard for security in the transportation industry.

### 2.3. Not In Scope

Certain aspects of biometric and system functionalities are excluded from this project's scope due to time, resource, or feasibility constraints:

- **Alternative Biometric Methods:** The system will focus exclusively on palmprint recognition and will not include facial recognition, iris scanning, or other biometric approaches.
- **Hardware Development:** The project will not involve the development of custom hardware; it will rely on existing mobile device capabilities for palmprint scanning.
- **Comprehensive Multi-Modal Biometric Systems:** The system will not integrate multiple biometric methods simultaneously (e.g., combining fingerprints and palmprints).
- **Cross-Industry Implementations:** While the system is designed for scalability to other industries, direct deployment in sectors beyond transport (e.g., banking, healthcare) will not be part of the initial project phase.
- **AI Bias Mitigation:** Although the system aims for high accuracy, addressing broader ethical concerns around AI biases in datasets and algorithms is beyond the project's immediate focus.

### 2.4. Project Objectives

The primary goal of this project is to overcome the limitations of existing biometric verification systems in the transport industry by integrating cutting-edge palmprint recognition techniques. This innovative approach addresses the shortcomings of traditional biometric methods, including:

- Sensitivity to environmental factors (lighting, temperature, etc.).
- Health concerns associated with contact-based systems.
- Suboptimal accuracy due to limited feature extraction mechanisms.

### 2.4.1 Comprehensive Competition Network: A Deep Learning Solution

To overcome these challenges, our project leverages the Comprehensive Competition Network approach, utilizing deep learning techniques to:

- Extract multi-order texture and spatial features.
- Enhance recognition accuracy and robustness.
- Ensure high-performance palmprint verification.

### 2.4.2 Mobile Application: Secure and Efficient Biometric Solution

Our mobile app integration offers:

- Non-contact, hygienic biometric verification.
- Efficient passenger identity verification.
- Easy deployment and scalability for future transport systems.

### 2.4.3 Future Expansion

This innovative solution has potential applications beyond the transport industry, including:

- Banking and ATM verification systems.
- Access control systems for secure facilities.
- Identity verification for border control and law enforcement.

### 2.4.4 Project Impact

By developing an advanced palmprint recognition system, we aim to:

- Enhance transport security and passenger safety.
- Improve operational efficiency and reduce identity-related risks.
- Provide a scalable, future-proof biometric solution for various industries.

## 2.5. Stakeholders

This project involves multiple stakeholders, each playing a crucial role in its development and deployment:

- **Project Team:**
  - Developers, testers, and designers responsible for implementing and delivering the system.
  - Team members include Muhammad Talha Bilal (21K-3349), Muhammad Hamza (21K-4579), and Muhammad Salar (21K-4619).

- **Supervisors and Academic Advisors:**
  - Dr. Muhammad Atif Tahir (Supervisor and Head of School, Computing).
  - Responsible for providing guidance, feedback, and academic validation.
- **Transport Industry Stakeholders:**
  - Includes transport companies and operators who will deploy the system for enhanced passenger identity verification.
- **End-Users:**
  - Passengers utilizing the mobile application for identity verification.
  - Transport staff ensuring system integration with operational workflows.
- **Secondary Stakeholders:**
  - Regulatory bodies overseeing transport security compliance.
  - Potential industry partners exploring scalability for banking, healthcare, and law enforcement.

### 2.6. Operating Environment

The software for the palmprint recognition system is designed to function effectively in a variety of environments, including high-traffic and operationally critical areas. The key operational requirements include:

- **System Configuration:**
  - The system should operate on devices with access to stable network connectivity to ensure real-time verification.
  - A mobile application developed using React Native will interface with the recognition model hosted in a containerized environment (e.g., Docker).
- **Model Hosting Environment:**
  - The palmprint recognition model, developed using the Comprehensive Competition Network (CCNet), will be hosted on a local server or cloud platform with GPU capabilities to handle computationally intensive operations.
  - The environment must support frameworks such as PyTorch and TensorFlow.
- **Application Requirements:**
  - The application must have the capability to capture high-quality palmprint images using the device's camera.
  - It will preprocess the images and communicate with the backend for palmprint verification.

- **Infrastructure Requirements:**
  - Local or cloud-based servers should have sufficient resources to deploy Dockerized environments.
  - Reliable API communication, such as through FastAPI, for secure and efficient data exchange.
- **Hardware Requirements:**
  - Devices must include high-resolution cameras for precise image capture.
  - Systems should feature basic biometric sensors for further scalability, though not mandatory for initial deployment.

The software must be robust enough to handle dynamic transport environments, ensuring reliable operation under diverse conditions such as variable lighting and weather. Scalability is prioritized to accommodate potential extensions into other industries, such as banking and healthcare.

### 2.7. System Constraints

The development and deployment of the palmprint recognition system are subject to the following constraints:

- **Computational Resource Limitations:**
  - The Comprehensive Competition Network (CCNet) requires high-performance GPU-enabled servers for training and inference.
  - Limited computational resources may impact the speed and efficiency of real-time recognition, especially in large-scale deployments.
- **Dataset Availability and Quality:**
  - Public datasets such as the Tongji Contactless Palmprint Dataset and CASIA Palmprint Image Database are used for training. However, these datasets may lack diversity in environmental and demographic conditions.
  - Locally collected datasets are subject to variability in image quality and consistency, potentially affecting model generalization.
- **Network Dependence:**
  - The system relies on stable network connectivity for real-time palmprint verification and backend communication. Any disruptions may result in delays or system downtime.
- **Mobile Application Constraints:**
  - The mobile application must be optimized for a range of devices with varying camera resolutions, processing power, and storage capacity.

- **Environmental Conditions:**
  - Variations in lighting, hand positioning, and environmental factors may affect the accuracy of palmprint image capture and recognition.
  - The system must be robust to these conditions, but extreme variations could still pose challenges.
- **Scalability Constraints:**
  - While designed for the transport sector, scaling the system to other industries such as banking and healthcare may require significant customization and additional testing.
- **Privacy and Security:**
  - Handling sensitive biometric data necessitates compliance with data protection regulations, such as local privacy laws.
  - Secure storage and transmission of data are critical, but implementing robust security measures increases system complexity and overhead.

These constraints will be carefully addressed during the design and implementation phases to ensure the successful deployment of the system while meeting project objectives.

### 2.8. Assumptions and Dependencies

The successful development and deployment of the palmprint recognition system are based on the following assumptions and dependencies:

- **Assumptions:**
  - The target transport organizations will have the required infrastructure, including internet connectivity and mobile devices, to support the system's deployment.
  - Users (passengers and staff) will have a basic level of familiarity with mobile applications and biometric systems, reducing the need for extensive training.
  - The datasets used for model training, including public datasets such as Tongji Contactless Palmprint Dataset and CASIA Palmprint Image Database, will sufficiently represent the operational conditions of the transport industry.
  - The collected local datasets will be of sufficient quality and quantity to fine-tune the model for specific deployment environments.
  - The system will primarily operate in environments with controlled lighting and standard mobile device capabilities, ensuring consistent performance.
- **Dependencies:**
  - **Hardware:** The system relies on mobile devices equipped with high-resolution cameras for accurate palmprint capture and backend servers with GPU capabilities for model inference.

- **Software:** Key frameworks and tools such as PyTorch, Docker, and React Native are essential for model development, deployment, and mobile application functionality.
- **Network Infrastructure:** Stable internet connectivity is required for real-time communication between the mobile application and the backend server.
- **APIs and Libraries:** The system depends on open-source libraries for preprocessing, image enhancement, and secure API communication (e.g., FastAPI).
- **Data Privacy Compliance:** The project must comply with relevant data privacy laws and regulations to ensure the secure handling of biometric data.

By adhering to these assumptions and dependencies, the system aims to achieve its objectives effectively within the defined operational and environmental constraints.

# 3. External Interface Requirements

## 3.1. Hardware Interfaces

The palmprint recognition system relies on the following hardware interfaces to ensure optimal performance and seamless integration:

- **Mobile Devices:**
  - Devices equipped with high-resolution cameras (minimum 12 MP) for accurate palmprint image capture.
  - Compatibility with Android platforms to ensure widespread accessibility.
  - Minimum processing capabilities equivalent to a quad-core processor and 4 GB RAM for smooth application performance.
- **Backend Servers:**
  - GPU-enabled servers (e.g., NVIDIA RTX series) for hosting the Comprehensive Competition Network (CCNet) model to handle computationally intensive tasks such as feature extraction and classification.
  - Sufficient storage capacity (minimum 256 GB) for storing palmprint datasets and logs.
  - Network connectivity with a minimum bandwidth of 10 Mbps for real-time communication between the mobile application and the server.
- **Development and Testing Equipment:**
  - High-performance workstations with GPU capabilities for model training and testing during the development phase.
  - Devices for field testing, including various mobile devices and external lighting equipment to simulate diverse environmental conditions.

These hardware interfaces are critical to the system's ability to perform efficiently and effectively in real-world operational settings, particularly in the transport industry.

### 3.2. Software Interfaces

The palmprint recognition system interacts with various software components to ensure smooth operation, including:

- **Operating Systems:**
  - Mobile Application: Compatible with Android (version 8.0 or higher).
  - Backend Server: Runs on Linux-based operating systems (e.g., Ubuntu 20.04) for optimal performance and reliability.
- **Development Frameworks and Tools:**
  - **React Native:** Used to develop a mobile application for Android devices.
  - **PyTorch:** Deep learning framework for implementing and training the Comprehensive Competition Network (CCNet) model.
  - **FastAPI:** Provides the API layer for secure and efficient communication between the mobile application and backend server.
  - **Docker:** Containerization technology used to deploy the CCNet model and backend services in a scalable and portable manner.
- **Database Management System:**
  - **Firebase and MongoDb:** Stores metadata, logs, and user information securely and efficiently.
  - **Cloud Storage Integration:** Optional integration with platforms such as AWS S3 or Azure for scalable and secure data storage.
- **Data Processing and Preprocessing Libraries:**
  - **PIL:** Used for preprocessing palmprint images, including resizing, normalization, and noise reduction.
  - **NumPy and SciPy:** Provide numerical operations and scientific computing functionalities for model implementation and optimization.
- **Security and Encryption:**
  - **TLS/SSL Protocols:** Ensure secure data transmission between the mobile application and the backend server.
- **Third-Party APIs:**
  - APIs for camera access, image processing, and notification services to enhance application functionality.

These software interfaces form the backbone of the palmprint recognition system, enabling efficient data processing, communication, and deployment in a real-world environment.

### 3.3. Communication Interfaces

The palmprint recognition system relies on various communication interfaces to facilitate seamless data exchange and interaction between its components. These include:

- **Mobile Application to Backend Server:**
  - **Protocol:** Communication between the mobile application and backend server will use HTTPS to ensure secure data transmission.
  - **API Layer:** RESTful APIs developed using FastAPI will handle all client-server interactions, including image uploads, authentication requests, and response retrieval.
  - **Data Format:** JSON format will be used for all data exchanges to ensure lightweight and standardized communication.
- **Backend Server to Database:**
  - **Protocol:** Secure connections (e.g., SSL) will be established between the backend server and the database for querying and data storage.
  - **Database Interaction:** MySQL queries or ORM-based interactions will facilitate efficient data retrieval and storage.
- **Model Hosting and Deployment:**
  - **Container Communication:** Docker containers hosting the CCNet model will use internal APIs to communicate with the backend services.
  - **Inference Requests:** The backend server will send palmprint image data to the CCNet model for real-time processing and receive results via inter-container API calls.
- **Network Requirements:**
  - A stable internet connection with a minimum bandwidth of 10 Mbps is required for real-time operations.
  - Support for LAN and WAN connectivity for flexibility in deployment environments.
- **Notification Services:**
  - Notifications, such as verification results or alerts, will be sent to end-users using push notification services.
- **Security and Encryption:**
  - Data exchanged between all system components will be encrypted using TLS 1.2 or higher.

These communication interfaces ensure secure, efficient, and reliable data flow across the system, facilitating seamless interaction between the mobile application, backend server, database, and model.

# 4. Functional Requirements

## 4.1. Functional Hierarchy

The functional requirements of the palmprint recognition system are organized in a hierarchical structure to represent the core functionalities and their subcomponents:

1. **Palmprint Verification System**
   (a) **Image Capture:**
      i. Capture high-resolution palmprint images using the device camera.
      ii. Ensure proper alignment and focus during image capture.
   (b) **Image Preprocessing:**
      i. Perform noise reduction, resizing, and normalization of captured images.
      ii. Apply data augmentation techniques, such as rotation and scaling, for improved recognition accuracy.
   (c) **Palmprint Recognition:**
      i. Use the Comprehensive Competition Network (CCNet) model for feature extraction and classification.
      ii. Ensure high accuracy and robustness in diverse environmental conditions.
   (d) **Authentication and Verification:**
      i. Compare extracted features with stored templates for identity verification.
      ii. Return verification results to the mobile application in real time.
2. **Mobile Application**
   (a) **User Interface:**
      i. Provide a user-friendly interface for image capture and verification feedback.
   (b) **Real-Time Communication:**
      i. Send captured palmprint images to the backend server for processing.
      ii. Display verification results or error messages to the user.
   (c) **Notifications:**
      i. Notify users of successful or failed verification attempts.
      ii. Provide alerts for suspicious activity or system errors.
3. **Backend System**
   (a) **Model Integration:**
      i. Host the CCNet model for processing palmprint images.
      ii. Ensure seamless integration with the mobile application and database.
   (b) **API Services:**
      i. Provide RESTful APIs for communication with the mobile application.
      ii. Handle authentication requests and data transmission securely.

(c) **Data Management:**

i. Store and manage user information and biometric templates securely.

ii. Log system activities and verification attempts for auditing purposes.

4. **Security and Privacy**

(a) **Access Control:**

i. Restrict system access to authorized users only.

(b) **Compliance:**

i. Ensure compliance with relevant privacy regulations.

This hierarchy outlines the critical functions of the palmprint recognition system, ensuring clarity and traceability during the development process.

## 4.2. Use Cases

### 4.2.1 Use Case Diagram

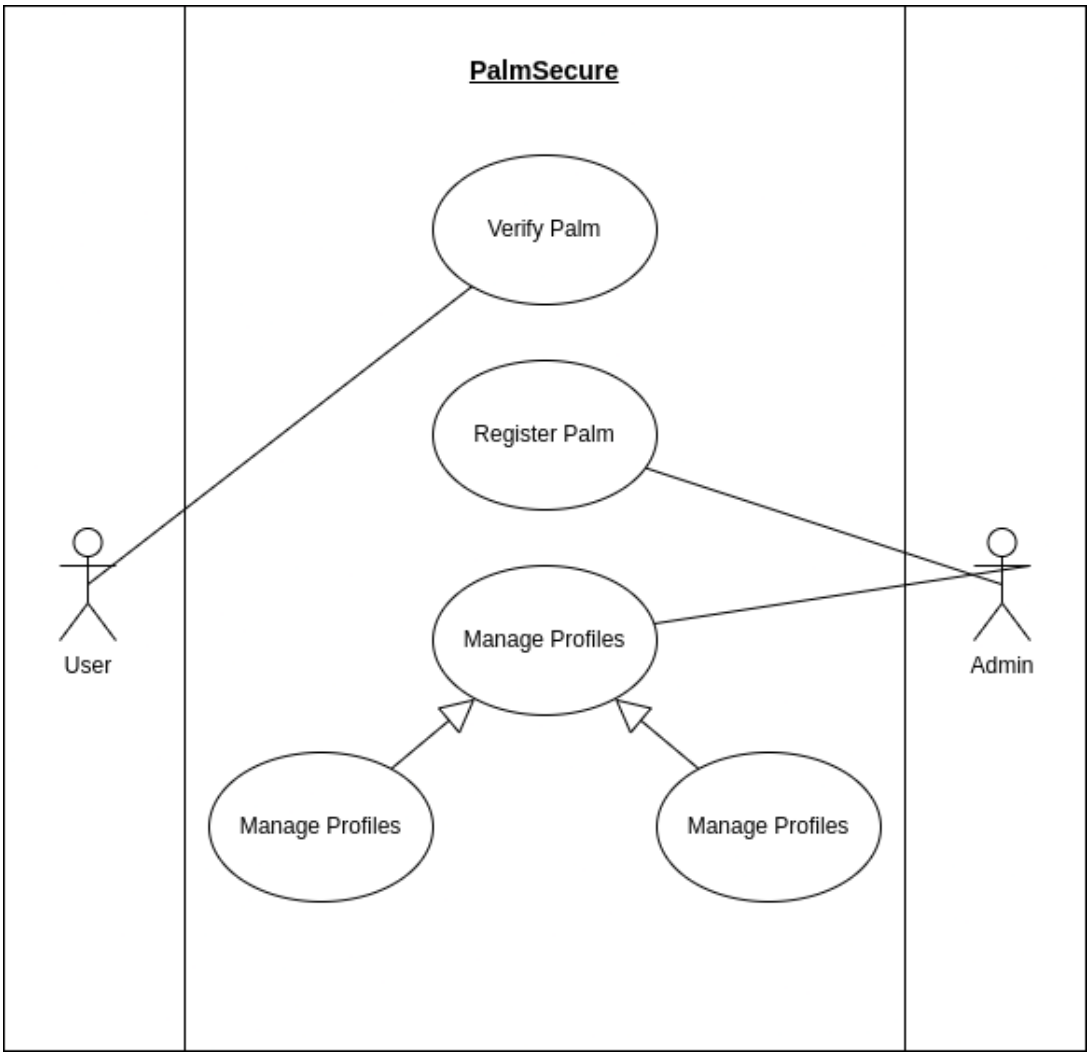


Figure 1: Use Case Diagram for PalmSecure System

### 4.2.2 Register Palm (Admin Version)

- **Use Case Name:** Register Palm (Admin Version)
- **ID:** 1
- **Importance Level:** Critical
- **Primary Actor:** Admin
- **Use Case Type:** Detail, Essential
- **Stakeholders and Interests:**
  - Admin – for registering a user's palm into the system for future authentications.
  - Organization – to ensure secure and streamlined user registration.
- **Brief Description:** This use case describes the process for registering a palm for a user profile by the admin.
- **Trigger:** Admin registers a user's palm to the system.
- **Type:** Internal
- **Relationships:**
  - **Association:** Admin
  - **Include:** Validate User Information
  - **Extend:** None
  - **Generalization:** None
- **Normal Flow of Events:**
  1. Admin selects the "Register Palm" option from the admin dashboard.
  2. Admin enters the user's details, including their unique ID and name.
  3. System creates a new user profile with the entered details.
  4. Admin aligns the user's palm in front of the camera device.
  5. Admin previews the captured ROI on the interface and confirms.
  6. System extracts features from the ROI using the model.
  7. System appends the embedding into the user's profile.
  8. System notifies the admin of successful registration.
- **Sub Flows:** None
- **Alternate/Exceptional Flows:**
  - 8a. If the system fails to register the palm:
    - System displays an error message with troubleshooting steps.

### 4.2.3 Verify Palm

- **Use Case Name:** Verify Palm
- **ID:** 2
- **Importance Level:** Critical
- **Primary Actor:** Admin/User
- **Use Case Type:** Detail, Essential
- **Stakeholders and Interests:**
  - User – To verify their palm for profile retrieval or clock-in.
  - Admin – To ensure the system processes palm verification accurately for users.
- **Brief Description:** This use case describes the real-time process for verifying a palmprint and associating it with a stored profile.
- **Trigger:** A user/admin initiates palm verification.
- **Type:** External
- **Relationships:**
  - **Association:** Admin/User
  - **Include:** Validate Profile
  - **Extend:** None
  - **Generalization:** None
- **Normal Flow of Events:**
  1. User/Admin aligns the user's palm in front of the camera device.
  2. System captures the palm image and extracts the ROI.
  3. System generates the embedding vector for the captured palm image.
  4. System compares the embedding vector with stored profiles.
  5. System finds the best match based on detection thresholds and confidence levels.
  6. System displays the result:
     - If matched, the system shows "Welcome user name" or other relevant profile details.
     - If unmatched, the system shows "Unrecognized Palm".
- **Sub Flows:** None
- **Alternate/Exceptional Flows:**
  - 6a. If the palmprint matches but with low confidence:
    - System prompts admin to validate manually.
  - 6b. If the palmprint fails to match:
    - System redirects back to the verification screen.

### 4.2.4 Manage System

- **Use Case Name:** Manage System
- **ID:** 3
- **Importance Level:** Critical
- **Primary Actor:** Admin
- **Use Case Type:** Detail, Essential
- **Stakeholders and Interests:**
  - Admin – To perform system management tasks such as editing settings or managing profiles.
- **Brief Description:** This use case describes how an admin accesses the system console to perform administrative tasks.
- **Trigger:** Admin initiates the "Admin Console" from the system menu.
- **Type:** External
- **Relationships:**
  - **Association:** Admin
  - **Include:** Authenticate Admin
  - **Extend:** None
  - **Generalization:** None
- **Normal Flow of Events:**
  1. Admin selects the "Admin Console" option from the main menu.
  2. System prompts the admin to enter their password.
  3. Admin enters the password, and the system validates it.
  4. System grants access to the admin console, displaying all settings and management options.
- **Sub Flows:** None
- **Alternate/Exceptional Flows:**
  - 3a. If the entered password is incorrect:
    - System denies access and provides an option to retry.
    - Admin may contact support if they cannot access the console.

### 4.2.5 Manage Profiles

- **Use Case Name:** Manage Profiles
- **ID:** 4
- **Importance Level:** High
- **Primary Actor:** Admin
- **Use Case Type:** Detail, Essential
- **Stakeholders and Interests:**
  - Admin – To manage user profiles in the system.
- **Brief Description:** This use case describes how an admin manages profiles, including viewing, resetting, and deleting user profiles.
- **Trigger:** Admin selects the "Manage Profiles" option from the admin console.
- **Type:** External
- **Relationships:**
  - **Association:** Admin
  - **Include:** View Profiles
  - **Extend:** None
  - **Generalization:** None
- **Normal Flow of Events:**
  1. Admin selects "Manage Registered Profiles" from the admin console.
  2. System displays all user profiles, with options for reset and delete actions.
- **Sub Flows:** None
- **Alternate/Exceptional Flows:** None

### 4.2.6 Delete Profile

- **Use Case Name:** Delete Profile
- **ID:** 5
- **Importance Level:** High
- **Primary Actor:** Admin
- **Use Case Type:** Detail, Essential
- **Stakeholders and Interests:**
  - Admin – To remove inactive or invalid user profiles.

- **Brief Description:** This use case describes how an admin deletes a user profile.
- **Trigger:** Admin selects the delete option for a user profile.
- **Type:** External
- **Relationships:**
  - **Association:** Admin
  - **Include:** Manage Profiles
  - **Extend:** None
  - **Generalization:** None
- **Normal Flow of Events:**
  1. Admin clicks the "Delete" button associated with a user profile.
  2. System prompts the admin to confirm the deletion.
  3. Admin confirms the action, and the system removes the profile from the database.
  4. System updates the view with the remaining profiles.
- **Sub Flows:** None
- **Alternate/Exceptional Flows:**
  - 2a. Admin cancels the action:
    - System redirects to the view without making changes.

#### 4.2.7 Reset Profile

- **Use Case Name:** Reset Profile
- **ID:** 6
- **Importance Level:** High
- **Primary Actor:** Admin
- **Use Case Type:** Detail, Essential
- **Stakeholders and Interests:**
  - Admin – To reset a user profile's embeddings in the system for re-registration.
- **Brief Description:** This use case describes how an admin resets a user profile.
- **Trigger:** Admin selects the reset option for a user profile.
- **Type:** External

- **Relationships:**
  - **Association:** Admin
  - **Include:** Manage Profiles
  - **Extend:** None
  - **Generalization:** None
- **Normal Flow of Events:**
  1. Admin clicks the "Reset" button associated with a user profile.
  2. System prompts the admin for confirmation.
  3. Admin confirms, and the system resets the embeddings associated with the profile.
  4. System updates the table view with the changes.
- **Sub Flows:** None
- **Alternate/Exceptional Flows:**
  - 2a. Admin cancels the action:
    - System redirects to the table view without making changes.

# 5. Non-Functional Requirements

## 5.1. Performance Requirements

The palmprint recognition system must meet the following performance requirements to ensure reliability and efficiency:

- **Accuracy:**
  - The system should achieve a minimum recognition accuracy of 90% in controlled environments.
  - The Equal Error Rate (EER) should not exceed 5%, ensuring a balance between False Acceptance Rate (FAR) and False Rejection Rate (FRR).
- **Response Time:**
  - he system must provide verification results with minimal latency following the submission of a palmprint image.
  - API response times for backend communication should not exceed 500 milliseconds under normal operating conditions.
- **Scalability:**
  - The backend infrastructure should handle concurrent verification requests without significant performance degradation.
  - The system should support future expansion to accommodate additional users and increased data volume.

- **Robustness:**
  - The system should maintain consistent performance under varying environmental conditions, such as different lighting levels and hand orientations.
  - It must recover gracefully from minor hardware or network failures without data loss or corruption.
- **Resource Utilization:**
  - The mobile application should be optimized for efficient performance, consuming minimal storage space and utilizing minimal CPU and memory resources during operation.
  - The backend server should utilize resources efficiently, ensuring optimal processing of recognition tasks.
- **Compatibility:**
  - The system should be compatible with devices running Android (8.0 or higher).
  - It should integrate seamlessly with existing transport infrastructure for real-time operation.
- **Availability:**
  - The system must maintain uptime in accordance with the Service Level Agreement (SLA) of the cloud provider, ensuring high availability for end-users and transport operators.

These performance requirements are critical to the success of the palmprint recognition system, ensuring it meets the expectations of users and stakeholders in the transport industry.

### 5.2. Safety Requirements

The palmprint recognition system must adhere to the following safety requirements to ensure secure and reliable operation, especially in critical environments such as the transport industry:

- **Data Integrity:**
  - Biometric data and user credentials must be protected against corruption, unauthorized modification, or loss during transmission and storage.
  - Regular backups of the database must be maintained to prevent data loss in the event of a system failure.
- **System Resilience:**
  - The system must handle unexpected hardware or software failures gracefully, with minimal disruption to ongoing operations.
  - Automatic failover mechanisms should be implemented to maintain system functionality during server or network outages.

- **User Safety:**
  - The mobile application must ensure that the process of capturing palmprint images is non-intrusive and does not cause discomfort to users.
- **Access Control:**
  - Only authorized personnel and users should have access to sensitive system features and data.
  - Multi-factor authentication should be implemented for administrative access to the backend server.
- **Environmental Safety:**
  - The system must function safely in diverse environmental conditions, including varying lighting and temperature levels, without compromising user data or system performance.
- **Regulatory Compliance:**
  - The system must comply with relevant safety standards and regulations, including data protection laws, to ensure secure handling of biometric data.

These safety requirements ensure the secure, reliable, and user-friendly operation of the palmprint recognition system in both routine and critical scenarios.

### 5.3. Security Requirements

The palmprint recognition system must meet the following security requirements to ensure the confidentiality, integrity, and availability of sensitive data and system operations:

- **Data Security:**
  - All biometric data, user credentials, and communication logs must be encrypted using Advanced Encryption Standard (AES-256) for data at rest and Transport Layer Security (TLS 1.2 or higher) for data in transit.
  - Biometric templates must be stored in a secure, hashed, and non-reversible format to prevent misuse in the event of a breach.
- **Authentication and Authorization:**
  - The system must implement multi-factor authentication (MFA) for administrative and backend access to enhance security.
  - Role-based access control (RBAC) should be enforced to ensure that users only access features and data relevant to their roles.
- **Application Security:**
  - Input validation mechanisms must be implemented to prevent injection attacks and unauthorized data manipulation.
  - The mobile application must securely handle session tokens to prevent replay attacks and session hijacking.

- **Network Security:**
  - All communication between the mobile application, backend server, and database must occur over secure channels using HTTPS.
- **System Monitoring and Logging:**
  - The system must log all user activities, authentication attempts, and backend interactions for auditing purposes.
  - Logs must be stored securely and monitored for anomalous behaviour to detect potential security breaches.
- **Disaster Recovery and Backup:**
  - Regular backups of critical data must be performed and stored securely to ensure recovery in case of a cyberattack or system failure.
  - A disaster recovery plan must be in place to minimize downtime and data loss in the event of a security incident.
- **Regulatory Compliance:**
  - The system must adhere to global and local data protection regulations to ensure the secure handling of biometric data.
  - Regular security audits must be conducted to verify compliance and identify vulnerabilities.

These security requirements are designed to safeguard sensitive biometric data, maintain system integrity, and ensure trustworthiness in critical operational environments.

### 5.4. User Documentation

Comprehensive user documentation will be provided to ensure ease of use and efficient operation of the palmprint recognition system. The documentation will include the following components:

- **User Manual:**
  - Detailed instructions on installing and setting up the mobile application on Android devices.
  - Step-by-step guide for capturing palmprint images, ensuring proper hand placement and alignment.
  - Explanation of the verification process, including how to interpret results and handle common errors.
- **Administrative Guide:**
  - Guidelines for managing user accounts, roles, and access control.
  - Procedures for monitoring system logs and troubleshooting common issues.

- **Developer Documentation:**
  - API reference documentation for integration with third-party systems and transport infrastructure.
  - Details on the software architecture, including the structure of the Comprehensive Competition Network (CCNet) model.
  - Instructions for extending the system to other domains, such as banking or healthcare.
- **Training Materials:**
  - Sample datasets and testing guidelines to help users understand the system's capabilities.
- **Frequently Asked Questions (FAQ):**
  - Answers to common user queries about the system's functionality and performance.
  - Troubleshooting tips for resolving basic issues without technical support.

The user documentation will be made available in digital format, and accessible through the mobile application ensuring users and administrators have easy access to relevant information at all times.

## 6. References

1. Ziyuan Yang, Huijie Huangfu, Lu Leng, Bob Zhang, Andrew Beng Jin Teoh, and Yi Zhang. "Comprehensive Competition Mechanism in Palmprint Recognition." *IEEE Transactions on Information Forensics and Security*, vol. 18, 2023. DOI: 10.1109/TIFS.2023.3306104.
2. **Datasets:**
   - Tongji Contactless Palmprint Dataset. Available at: `https://cslinzhang.github.io/ContactlessPalm`.
   - CASIA Palmprint Image Database. Available at: `http://biometrics.idealtest.org`.
   - COEP Palmprint Dataset. Available at: `https://www.coeptech.ac.in/`
3. **Tool Documentation:**
   - PyTorch Documentation. Available at: `https://pytorch.org/docs`.
   - FastAPI Documentation. Available at: `https://fastapi.tiangolo.com`.
   - React Native Documentation. Available at: `https://reactnative.dev`.
   - Docker Documentation. Available at: `https://docs.docker.com`.
4. **Software Standards:**
   - IEEE 830-1998, IEEE Recommended Practice for Software Requirement Specification. Available at: `https://ieeexplore.ieee.org/document/720574`.

# Appendices

## Glossary

- **ROI:** Region of Interest – The specific area of an image used for processing.
- **CCNet:** Comprehensive Competition Network – A deep learning model for palmprint recognition.
- **API:** Application Programming Interface – Enables communication between software components.

## Datasets Used

- **Tongji Contactless Palmprint Dataset:** Contains 12,000 images captured from 300 individuals, suitable for contactless palmprint recognition applications.
- **CASIA Palmprint Image Database:** Includes 5,502 images from 312 individuals, widely used for benchmarking palmprint recognition systems.
- **COEP Palmprint Dataset:** It consists of 1,344 palmprint images collected from 168 individuals, with each participant contributing 8 images.

## System Architecture Diagram

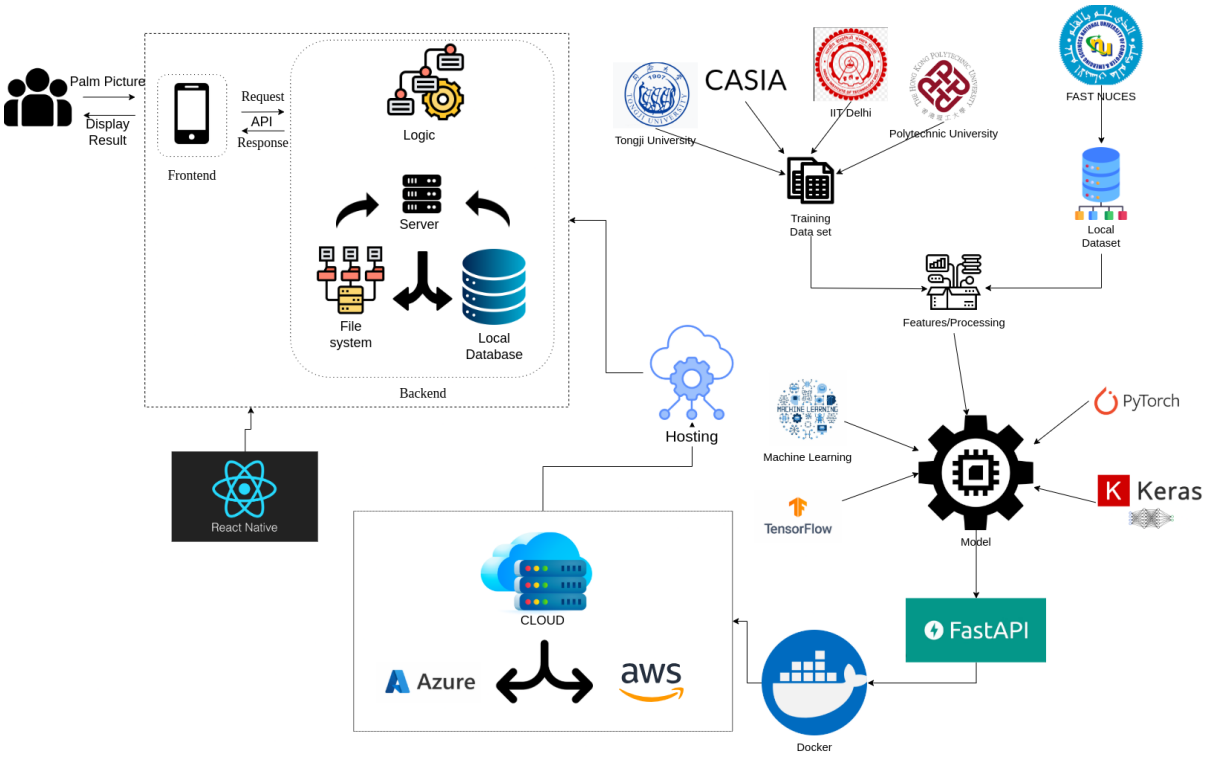


Figure 2: System Architecture for PalmSecure

### Sample Code Snippet

```
# Sample Python Code for Model Training
import torch
from torch import nn
from torchvision import transforms

# Define the CCNet model
class CCNet(nn.Module):
    def __init__(self):
        super(CCNet, self).__init__()
        # Add layers here...

    def forward(self, x):
        # Forward pass logic
        return x

model = CCNet()
print(model)
```

### Technical Specifications

- **Development Tools:**
  - PyTorch for model implementation.
  - React Native for mobile application development.
  - Docker for containerized deployment.
- **Hardware Requirements:**
  - High-resolution camera for palmprint capture.
  - GPU-enabled servers for training and inference.

### Future Work

- Explore the scalability of CCNet in domains like healthcare and banking.
- Optimize CCNet for resource-constrained devices.
- Investigate cross-domain generalization techniques.