Software Design 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)

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 Design Consideration, Design Strategy
00.01	Talha, Hamza, Salar	Dec 1, 2024	Added System Architecture
00.01	Talha, Hamza, Salar	Dec 8, 2024	Added Detailed System Design

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	Project Role	Sign	Sign-off Date
00.01	Dr. Atif Tahir	Supervisor		December 11, 2024

Document Information

Category	Information
Customer	National University of Computer and Emerging Sciences
Project	PalmSecure
Document	Software Design Specification
Document Version	1.0
Status	Draft
Author(s)	Muhammad Hamza Talha Bilal Muhammad Salar
Approver(s)	Dr. Muhammad Atif Tahir
Issue Date	
Document Location	
Distribution	Supervisor Internal Jury Members FYP Coordinator

Definition of Terms, Acronyms and Abbreviations

Term	Description
CCNet	Comprehensive Competition Network is the model used for palmprint recognition.
CNN	Convolutional Neural Network, a class of deep neural networks commonly used in image processing tasks.
API	Application Programming Interface, a set of rules that allow different software entities to communicate with each other.
KPI	Key Performance Indicator, are metrics used to evaluate the success of an activity or project.
AI	Artificial Intelligence, the simulation of human intelligence processes by machines.
GPU	Graphics Processing Unit, used for high-performance computing tasks such as training deep learning models.
CA	Coordinate Attention, a mechanism to focus on specific regions of an image during feature extraction.
CRUD	Create, Read, Update, Delete, a common set of operations in database and application design.
PCA	Principal Component Analysis, a technique for reducing the dimensionality of data while retaining essential features.
CASIA	Chinese Academy of Sciences Institute of Automation, a source of public datasets used in biometrics.

Contents

1. Introduction

1.1. Purpose of Document

The purpose of this Software Design Specification (SDS) document is to provide a comprehensive, detailed, and structured blueprint for the development and implementation of the **PalmSecure - Revolutionizing Transport with Biometric Precision** system. This document serves as a critical reference for all stakeholders, ensuring alignment between the project's goals, technical requirements, and design considerations.

Specifically, this document is intended to:

1. **Articulate the Objectives and Vision of the System:**
   - To enhance the security and efficiency of transportation systems by utilizing advanced palmprint recognition technology.
   - To address and overcome the challenges of traditional biometric systems, including sensitivity to environmental conditions, hygiene concerns, and limited accuracy in high-traffic and dynamic environments.
2. **Provide a Unified Framework for Development:**
   - To define the architectural framework, design strategy, and detailed components necessary for the implementation of the Comprehensive Competition Network (CCNet) in the palmprint recognition system.
   - To guide the development team through a systematic design process, reducing ambiguity and ensuring consistency throughout the system's lifecycle.
3. **Facilitate Communication Across Stakeholders:**
   - To act as a bridge between technical teams, supervisors, and academic evaluators by clearly outlining the system's design, intended functionality, and implementation methodology.
   - To ensure that all stakeholders, including developers and end-users, have a shared understanding of the system's scope and objectives.
4. **Support the Validation and Testing Phases:**
   - To serve as a critical reference during the testing and validation phases, ensuring the system meets the specified functional and non-functional requirements.
   - To outline the mechanisms and methodologies for verifying the robustness, accuracy, and reliability of the system under diverse operating conditions.
5. **Future Adaptability:**
   - To lay a foundation for expanding the system's applications beyond the transport sector into domains such as banking, healthcare, and law enforcement.

This document is pivotal in ensuring that the *PalmSecure* system is developed to meet its intended objectives effectively and efficiently. By providing a clear and structured design roadmap, it minimizes risks, ensures resource optimization, and paves the way for the successful realization of a secure and innovative biometric verification system tailored to the transportation industry.

1.2. Intended Audience

This Software Design Specification (SDS) document is intended for a diverse group of stakeholders involved in the development and evaluation of the **PalmSecure - Revolutionizing Transport with Biometric Precision** system. Each audience group has specific roles and interests in the project, and this document ensures that their requirements and expectations are addressed.

1. **Development Team:**
   - Includes developers responsible for implementing the system.
   - This document provides a structured framework for the design and architecture of the system, ensuring consistency during the development lifecycle.
2. **Supervisors and Academic Evaluators:**
   - Supervisors, such as **Dr. Muhammad Atif Tahir**, and academic evaluators responsible for guiding and assessing the project's progress and outcomes.
   - This document serves as a reference to validate that the project meets academic standards, aligns with its objectives and follows a robust design methodology.
3. **End-Users:**
   - Includes transportation sector personnel such as operators, administrators, and passengers who will interact with the system.
   - This document outlines how the system will enhance operational efficiency, ensure passenger security, and provide a seamless user experience.
4. **Industry Stakeholders:**
   - Comprises transport companies and operators interested in adopting the system for improved security and identity verification.
   - This document highlights the technical capabilities, scalability, and benefits of the system in real-world scenarios.
5. **Regulatory Bodies:**
   - Regulatory authorities overseeing security, privacy, and compliance within the transportation sector.
   - This document ensures that the system adheres to industry standards and legal requirements for biometric data handling and privacy.
6. **Potential Industry Partners:**
   - Organizations exploring future scalability into domains such as banking, healthcare, and law enforcement.
   - This document provides a technical overview of the system's design, ensuring its adaptability for broader applications.

The intended audience collectively ensures that the system is designed, developed, and deployed to meet its goals effectively. This document bridges the gap between technical details, user requirements, and compliance standards, making it a comprehensive guide for all stakeholders involved in the *PalmSecure* project.

1.3. Document Convention

This Software Design Specification (SDS) document follows a set of structured conventions to ensure clarity, consistency, and ease of understanding for all stakeholders. The conventions adopted are aimed at providing a standardized approach to presenting information, making the document accessible to both technical and non-technical audiences.

1. **Font and Style:**
    - **Font Style:** Times New Roman is used throughout the document for both headings and body text.
    - **Font Size:**
        - Headings: 12-point, bold.
        - Body Text: 12-point, regular.
    - **Text Formatting:**
        - **Bold text** is used to highlight section titles, key terms, and important concepts.
        - *Italicized text* is used for technical terms, system names, and emphasis where necessary.
2. **Section Numbering:**
    - The document is structured hierarchically with numbered sections and subsections to provide a logical flow of information.
    - For example:
        - **Section 1:** Introduction
        - **Subsection 1.1:** Purpose of Document
        - **Subsection 1.2:** Intended Audience
3. **Figures:**
    - All figures are labelled with descriptive captions for easy identification and reference.
    - Figures are numbered sequentially (e.g., Figure 1, Figure 2) and referenced appropriately in the text.
4. **References and Citations:**
    - References are provided in a dedicated section at the end of the document.
    - Inline citations are included using a numerical format (e.g., [1], [2]) to link to the reference section.
    - Where applicable, citations adhere to academic standards and are formatted consistently.
5. **Terminology:**
    - Key technical terms, acronyms, and abbreviations are defined in the *Definitions of Terms, Acronyms, and Abbreviations* section.

- Specialized terms are explained in context where first introduced to ensure comprehension.

6. **Version Control:**
   - Each version of this document is tracked with a unique version number and revision history.
   - The document includes a *Document History* section outlining changes, the contributors responsible, and the dates of revisions.
7. **Document Structure:**
   - The document is organized into clearly defined sections and subsections to facilitate easy navigation.
   - A **Table of Contents** is provided at the beginning of the document for quick reference.

These conventions ensure that the document is professional, consistent, and user-friendly. By adhering to these standards, this SDS facilitates effective communication among all stakeholders and provides a reliable framework for the successful implementation of the *PalmSecure* system.

1.4. Project Overview

The **PalmSecure - Revolutionizing Transport with Biometric Precision** project aims to address critical security, efficiency, and reliability challenges in the transportation sector by leveraging advanced biometric technology. This project introduces a state-of-the-art palmprint recognition system, integrating deep learning techniques through the **Comprehensive Competition Network (CCNet)** to achieve unparalleled accuracy, scalability, and usability in identity verification processes.

Background and Motivation: Traditional biometric systems, such as fingerprint and facial recognition, have shown limitations in operational environments characterized by high traffic, diverse lighting, and hygiene concerns. The increasing need for secure, contactless, and efficient identity verification in transportation systems has paved the way for palmprint recognition. Palmprint biometrics offer unique advantages, including:

- **Hygienic and Non-contact Operation:** Ensures user safety and convenience, particularly in public and high-traffic settings.
- **Robustness to Environmental Variations:** Operates reliably under diverse lighting and weather conditions.
- **High Accuracy and Antispoof Capabilities:** Utilizes the unique patterns of ridges, wrinkles, and minutiae present in human palms.

Project Goals and Objectives: The primary goal of this project is to design, develop, and deploy a palmprint verification system tailored to the dynamic requirements of the transportation sector. Key objectives include:

- Implementing the **CCNet** architecture to extract multi-order texture and spatial features for enhanced recognition accuracy.
- Delivering a mobile application that provides real-time identity verification with a user-friendly interface.
- Demonstrating the effectiveness of the system through rigorous testing on both public datasets and locally collected palmprint data.
- Ensuring scalability and adaptability of the solution for future applications in sectors such as banking, healthcare, and law enforcement.

System Features: The system incorporates the following features:

- **Comprehensive Competition Network (CCNet):** Integrates channel, spatial, and multi-order competition mechanisms for robust feature extraction.
- **Mobile Application Integration:** Provides seamless palmprint image capture and verification capabilities through an intuitive and secure mobile interface.
- **Hygienic and Non-contact Verification:** Reduces health risks associated with traditional contact-based systems.
- **Scalability and Cross-industry Applicability:** Designed for transport systems but adaptable to other high-security industries.

Expected Outcomes: The project aspires to deliver a transformative solution that:

- Enhances security and passenger safety in the transport industry.
- Streamlines operational workflows by integrating biometric verification into existing systems.
- Sets a new standard for scalable, future-proof biometric solutions with the potential for adoption across multiple industries.

Significance and Impact: By integrating cutting-edge biometric recognition technology with a comprehensive design approach, the *PalmSecure* project addresses pressing security challenges while paving the way for innovation in identity verification systems. Its emphasis on accuracy, hygiene, and user experience ensures widespread applicability and establishes a benchmark for future biometric solutions.

1.5. Scope

The scope of the **PalmSecure - Revolutionizing Transport with Biometric Precision** project encompasses the design, development, and deployment of an advanced palmprint recognition system tailored for the transportation sector. By leveraging the **Comprehensive Competition Network (CCNet)** and state-of-the-art deep learning techniques, this project addresses critical security challenges and introduces a scalable, efficient, and user-friendly biometric verification solution.

Core Functionalities: The project focuses on delivering the following primary functionalities:

- **Palmprint Verification System:**
  - Implementation of CCNet to extract multi-order texture and spatial features for robust and accurate recognition.
  - Integration of non-contact palmprint verification for hygienic and user-friendly operation.
- **Mobile Application Development:**
  - A seamless interface for capturing and verifying palmprint images in real time.
  - Efficient identity verification to prevent fraud, unauthorized access, and identity theft.
  - Integration with existing transportation infrastructure to optimize operational workflows.
- **Data Handling and Security:**
  - Use of public datasets (e.g., Tongji, CASIA) and locally collected data for model training and testing.
  - Implementation of robust privacy and data security protocols to comply with industry standards.

In-scope Activities: The following activities are included within the scope of the project:

- Development of a fully functional CCNet-based palmprint recognition model.
- Deployment of the recognition model using a containerized environment (e.g., Docker) for scalability and efficiency.
- Design and implementation of a mobile application that interfaces with the recognition model to deliver real-time results.
- System testing and evaluation on both public and locally collected datasets to validate performance and accuracy.
- Documentation and presentation of findings, including potential areas for enhancement and future scalability.

Out-of-scope Activities: Certain activities are excluded from the project scope due to time, resource, or feasibility constraints:

- Development of custom hardware; the system will rely on existing mobile device capabilities for palmprint scanning.
- Integration of other biometric modalities, such as facial recognition or iris scanning.
- Deployment in sectors beyond transportation (e.g., banking, healthcare); however, scalability is considered for future phases.
- Comprehensive bias mitigation strategies in datasets; the focus remains on ensuring high accuracy and usability within the defined domain.

Future Scalability: While this project is primarily focused on the transportation sector, the system is designed to be scalable and adaptable to other industries. Potential applications include:

- **Banking and Financial Services:** Enhancing ATM security and customer identity verification.
- **Healthcare:** Streamlining patient check-ins and secure access to medical records.
- **Law Enforcement and Border Control:** Enabling rapid and secure identity verification in high-security environments.

Impact and Relevance: This project aims to set a new benchmark in biometric verification systems by addressing challenges such as environmental sensitivity, hygiene concerns, and limited accuracy. Its implementation in the transportation sector promises to:

- Enhance passenger safety and security.
- Improve operational efficiency by reducing reliance on manual verification methods.
- Provide a future-proof and scalable solution adaptable to evolving technological and industry demands.

By clearly defining its scope, the *PalmSecure* project ensures focused development, efficient resource utilization, and a robust foundation for achieving its goals.

2. Design Considerations

2.1. Assumptions and Dependencies

The development and implementation of the **PalmSecure - Revolutionizing Transport with Biometric Precision** system are based on a set of assumptions and dependencies that influence the design, functionality, and deployment of the project. These factors are critical to ensuring the success and feasibility of the system.

Assumptions: The following assumptions are made during the design and development of the system:

- **Availability of High-Quality Data:**
  - Public datasets such as the Tongji Contactless Palmprint Dataset and CASIA Palmprint Image Database are assumed to provide sufficient and diverse training and testing samples.
  - A locally collected dataset will complement public datasets to represent the target environment, ensuring the system is tailored to the transportation sector.
- **Hardware Compatibility:**
  - End-users will utilize modern mobile devices equipped with high-resolution cameras capable of capturing palmprint images.

  - The system assumes the availability of GPU-enabled servers or cloud platforms for training the recognition model.

- **Stable Network Connectivity:**
  - The system relies on stable network connections to ensure seamless communication between the mobile application and the backend model hosted on a server or cloud environment.

- **Compliance with Industry Standards:**
  - It is assumed that the system will operate in environments compliant with data security and privacy regulations.

- **User Cooperation:**
  - Users are assumed to position their palms correctly for image capture, following the guidance provided by the mobile application interface.

Dependencies: The successful implementation of the system depends on the following key factors:

- **Technological Dependencies:**
  - Frameworks such as PyTorch are required to develop and train the CCNet model.
  - Docker or a similar containerization tool is necessary for deploying the system in a scalable and portable environment.

- **Data Dependencies:**
  - Access to public and locally collected palmprint datasets is essential for model training, testing, and validation.
  - Data augmentation techniques (e.g., rotation, scaling, and illumination adjustments) are critical to enhancing the robustness of the model.

- **Infrastructure Dependencies:**
  - The system depends on the availability of backend servers with GPU capabilities to handle computationally intensive tasks.
  - Reliable API communication (e.g., through FastAPI) is necessary for efficient interaction between the mobile application and backend services.

- **Regulatory and Compliance Dependencies:**
  - The system requires adherence to privacy and security standards for biometric data handling and storage, ensuring legal compliance.

- **User Dependencies:**
  - Successful adoption of the system depends on its usability, requiring an intuitive mobile application interface to guide users during the verification process.

2.2. Risks and Volatile Areas

The development and deployment of the **PalmSecure - Revolutionizing Transport with Biometric Precision** system involves certain risks and volatile areas that could impact the project's success. Identifying and addressing these risks early ensures that mitigation strategies are implemented proactively, reducing the likelihood of project delays or failures. The following outlines the primary risks and volatile areas associated with the project:

1. Data Quality and Availability:

- **Risk:**
  - Public datasets, such as the Tongji Contactless Palmprint Dataset and CASIA Palmprint Database, may lack diversity in environmental conditions, demographics, and image quality.
  - Locally collected datasets may exhibit inconsistencies in image resolution, lighting, and hand positioning.
- **Impact:**
  - Limited data diversity could reduce the generalizability of the palmprint recognition model, leading to lower accuracy in real-world scenarios.
- **Mitigation Strategy:**
  - Employ data augmentation techniques (e.g., scaling, rotation, illumination adjustments) to enhance dataset robustness.
  - Collect a diverse range of local data to supplement public datasets and improve model adaptability.

2. Network Dependence:

- **Risk:**
  - The system relies on stable network connectivity for communication between the mobile application and backend servers.
  - Network disruptions or latency issues could hinder real-time palmprint verification.
- **Impact:**
  - Delays or failures in verification processes could lead to user dissatisfaction and reduced system reliability.
- **Mitigation Strategy:**
  - Implement an offline mode for temporary local verification, with periodic synchronization to the server when connectivity is restored.
  - Optimize network communication protocols to minimize latency and improve resilience against connectivity issues.

3. **User-related Issues:**

   - **Risk:**
     - Incorrect hand positioning, poor lighting, or user errors during palmprint image capture could affect image quality.
   - **Impact:**
     - Poor-quality images could lead to reduced recognition accuracy and false negatives, affecting user trust in the system.
   - **Mitigation Strategy:**
     - Design an intuitive mobile application interface with real-time feedback to guide users in proper palm positioning and image capture.
     - Incorporate image preprocessing techniques to enhance the quality of captured images.

4. **Regulatory and Compliance Challenges:**

   - **Risk:**
     - Biometric data handling involves strict privacy and security regulations.
   - **Impact:**
     - Non-compliance with data protection laws could result in legal penalties, project delays, or loss of stakeholder trust.
   - **Mitigation Strategy:**
     - Implement robust data encryption, secure storage, and access control mechanisms.
     - Regularly review and update the system to comply with evolving regulatory requirements.

5. **Scalability and Performance:**

   - **Risk:**
     - The system must handle a large number of users and high traffic volumes in transportation settings.
     - Insufficient computational resources or poorly optimized models could lead to performance bottlenecks.
   - **Impact:**
     - Scalability issues could hinder system adoption in larger deployments and compromise user experience.

- **Mitigation Strategy:**
  - Use containerized environments, such as Docker, to enable efficient scaling of the system.
  - Optimize the CCNet model for real-time performance and minimize computational overhead.

6. Technological Dependencies:

- **Risk:**
  - Reliance on third-party frameworks (e.g., PyTorch) and tools (e.g., Docker) introduces risks of software bugs, updates, or deprecations.
- **Impact:**
  - Disruptions in dependencies could delay system development or affect reliability.
- **Mitigation Strategy:**
  - Regularly monitor updates and changes to third-party tools and frameworks.
  - Maintain backward compatibility and plan for alternate tools to mitigate disruptions.

Conclusion: By identifying and addressing these risks and volatile areas, the project ensures a robust and adaptable design strategy for the successful implementation of the *PalmSecure* system. Regular risk assessments and iterative development processes further reduce uncertainties and enhance the system's reliability and performance.

3. System Architecture

3.1. System Level Architecture

The system-level architecture of the **PalmSecure - Revolutionizing Transport with Biometric Precision** project provides a high-level view of the system's components, their interactions, and the flow of data. This architecture ensures seamless integration between the mobile application, backend server, and other external interfaces to deliver efficient palmprint-based biometric verification.

Overview: The system consists of the following major components:

- **Mobile Application:**
  - Acts as the primary interface for users to capture palmprint images and view verification results.
  - Communicates with the backend server for processing and verification tasks.

- **Backend Server:**
  - Hosts the palmprint recognition model and handles preprocessing, feature extraction, and verification.
  - Manages API endpoints for seamless communication with the mobile application.
- **Recognition Model (CCNet):**
  - Implements the Comprehensive Competition Network (CCNet) for multi-order texture and spatial feature extraction.
  - Performs identity verification based on captured palmprint images.
- **Database:**
  - Stores user data, extracted palmprint features, and system logs for future reference and analysis.
  - Ensures secure and efficient data management.

System-Level Diagram:

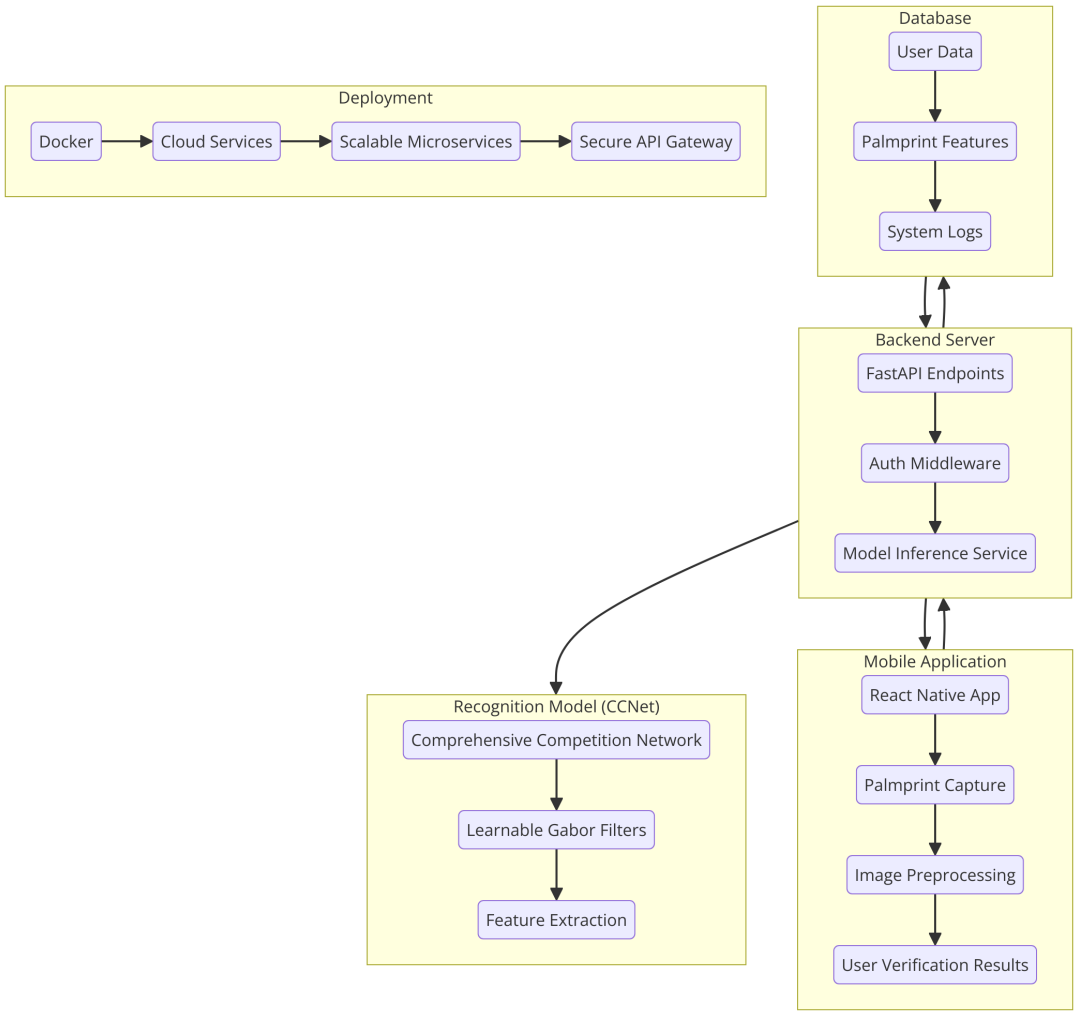


Figure 1: System Architecture for PalmSecure

This diagram illustrates the interaction between the mobile application, backend server, database, and recognition model.

Data Flow: The flow of data within the system involves the following steps:

1. The user captures a palmprint image via the mobile application.
2. The image is sent to the backend server for preprocessing and feature extraction.
3. The recognition model (CCNet) processes the image and performs verification.
4. The verification result is sent back to the mobile application and displayed to the user.
5. All relevant data is logged in the database for audit and analysis purposes.

3.2. Software Architecture

The software architecture describes the internal structure of the system, focusing on the software modules and their interactions. This architecture ensures modularity, scalability, and maintainability of the *PalmSecure* system.

Design Paradigm: The software architecture adopts a modular design paradigm to ensure:

- Scalability: Modules can be updated or extended independently.
- Maintainability: Changes in one module do not significantly impact others.
- Reusability: Components can be reused in future applications or extensions.

Key Software Modules: The architecture comprises the following software modules:

- **User Interface Module:**
  - Provides the mobile application's interface for capturing palmprint images and displaying results.
  - Ensures a user-friendly experience with real-time feedback.
- **API Layer:**
  - Facilitates communication between the mobile application and the backend server.
  - Uses secure protocols (e.g., HTTPS) for data transmission.
- **Processing Module:**
  - Handles image preprocessing tasks such as normalization, noise reduction, and feature extraction.
  - Prepares data for the recognition model.

- **Recognition Module (CCNet):**
  - Implements the Comprehensive Competition Network for palmprint recognition.
  - Extracts multi-order features and performs identity verification.
- **Database Management Module:**
  - Manages storage, retrieval, and updating of user data, palmprint features, and logs.
  - Ensures compliance with security and privacy regulations.
- **Logging and Monitoring Module:**
  - Tracks system performance and logs errors for debugging and analysis.
  - Provides insights into system usage and potential bottlenecks.

Integration of Modules: The modules are integrated to ensure seamless operation:

- The User Interface Module interacts with the API Layer for data exchange.
- The API Layer routes requests to the Processing Module and Recognition Module.
- The Database Management Module stores and retrieves data for all modules.

Software Architecture Diagram:

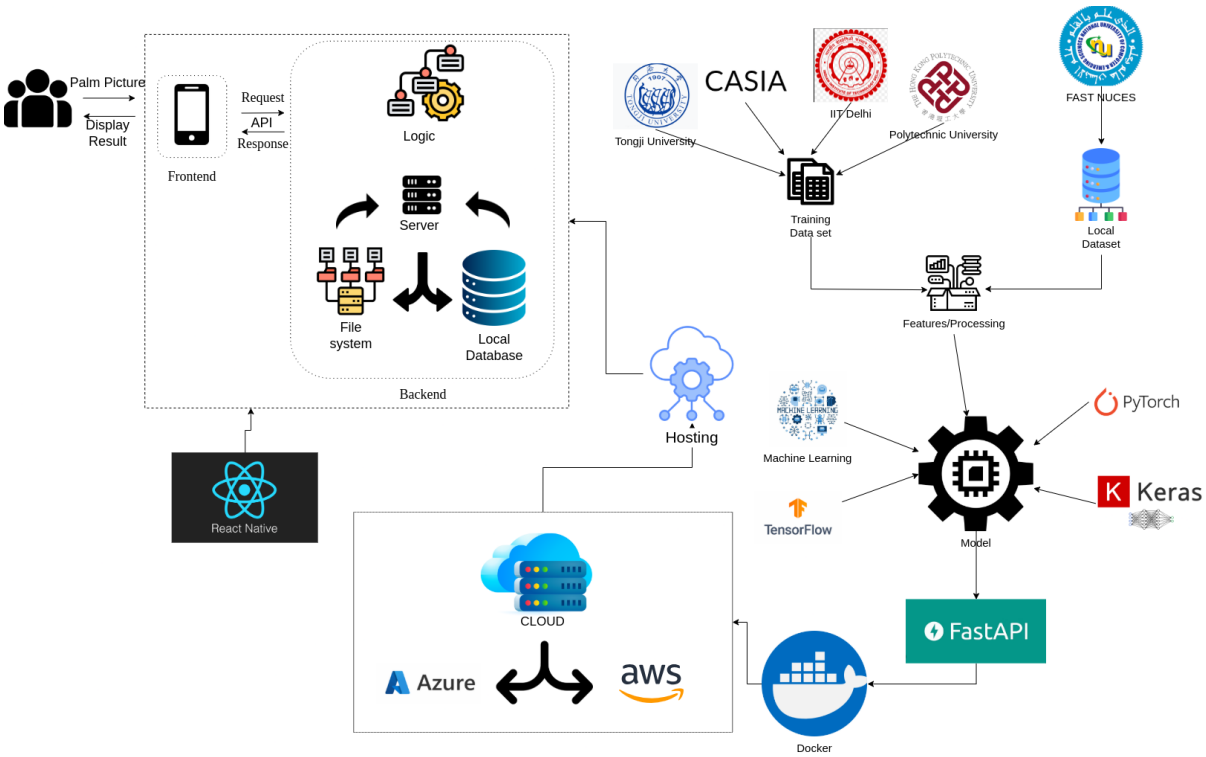


Figure 2: Software Architecture for PalmSecure

This diagram illustrates the relationships and data flow between the software modules, ensuring a clear understanding of the system's internal structure.

Key Technologies:

- **Frontend:** React Native for mobile application development.
- **Backend:** FastAPI for server-side development and communication.
- **Model Development:** PyTorch for training and deploying the CCNet model.
- **Database:** Firebase or MongoDB for secure data storage.
- **Containerization:** Docker for scalable deployment of backend and recognition modules.

This architecture ensures the system's reliability, performance, and adaptability, laying the foundation for successful implementation and scalability into future applications.

4. Design Strategy

The **Design Strategy** outlines the approach, principles, and methodologies adopted to design the *PalmSecure - Revolutionizing Transport with Biometric Precision* system. It ensures that the system's design aligns with the project's objectives, functional requirements, and constraints while addressing scalability, maintainability, and security.

Introduction: The design strategy focuses on creating a robust, modular, and efficient system architecture that supports real-time palmprint verification. By adhering to established methodologies and principles, the strategy ensures the system's scalability, maintainability, and adaptability for future applications in other industries such as healthcare and banking.

Design Objectives: The design is guided by the following objectives:

- **Scalability:** Ensure the system can handle an increasing number of users and datasets without significant performance degradation.
- **Maintainability:** Design components that are easy to update, debug, and extend.
- **Security:** Protect user data and ensure compliance with industry standards for biometric data privacy.
- **Efficiency:** Optimize the system for real-time operations, minimizing latency and resource usage.
- **User Experience:** Create an intuitive and user-friendly interface for seamless interaction.

Design Methodology: The following methodologies are employed in the design process:

- **Modular Design:** The system is divided into independent modules (e.g., mobile application, backend, recognition model) to enhance maintainability and scalability.
- **Agile Practices:** The development process follows agile principles, allowing flexibility in incorporating changes during the project lifecycle.

Design Principles: The design adheres to these guiding principles:

- **Separation of Concerns:** Each module is responsible for specific functionality (e.g., preprocessing, feature extraction, API communication).
- **Reusability:** Components are designed to be reusable in future applications.
- **Scalability:** The architecture supports seamless scaling to accommodate higher user loads and extended features.
- **Performance Optimization:** Real-time operations are prioritized to ensure user satisfaction.

Tools and Technologies: The following tools and technologies are used in the design and implementation:

- **Frontend:** React Native for mobile application development.
- **Backend:** FastAPI for server-side communication.
- **Machine Learning:** PyTorch for implementing and deploying the CCNet model.
- **Database:** Firebase or MongoDB for secure and efficient data storage.
- **Containerization:** Docker for deploying backend and recognition model components.
- **Version Control:** Git for tracking code changes and collaboration.
- **Visualization:** UML diagrams and flowcharts for design representation.

System Constraints: The design is influenced by the following constraints:

- **Hardware Limitations:** Dependence on GPU-enabled servers for model training and deployment.
- **Network Dependency:** Stable internet connectivity is required for real-time palmprint verification.
- **Compliance Requirements:** The system must adhere to biometric data privacy regulations.

Trade-offs and Design Decisions: Key trade-offs and decisions made during the design process include:

- **CCNet Implementation:** Prioritizing high accuracy and robustness over computational complexity.
- **Non-contact Biometric System:** Choosing palmprint recognition for its hygiene and reliability over other modalities like fingerprint or facial recognition.
- **Cloud vs. On-premises Deployment:** Opting for containerized solutions (Docker) to support scalability and flexibility.

Testing and Validation Strategy: The design is validated through rigorous testing:

- **Unit Testing:** Individual modules (e.g., preprocessing, API layer) are tested for functionality.
- **Integration Testing:** Interactions between modules (e.g., mobile app and backend) are validated.
- **Performance Testing:** The system is tested for real-time processing under varying loads.
- **User Acceptance Testing (UAT):** End-user feedback is gathered to refine the design and ensure usability.

Future-Proofing the Design: The design is future-proofed to support scalability and adaptability:

- **Modular Architecture:** Enables easy addition of new features or integration with other systems.
- **Scalable Deployment:** Containerization ensures the system can handle increased user loads.
- **Cross-industry Applications:** Designed to be extendable for use in healthcare, banking, and law enforcement.

Design Diagrams: These diagrams visually represent the system's design, showing the interactions between components and the overall data flow.

The following UML diagrams illustrate the design strategy for the *PalmSecure* system, highlighting the interactions between modules and the system's physical deployment.

The high-level design flowchart illustrates the overall architecture and workflow of the *PalmSecure* system, highlighting interactions between its key components.

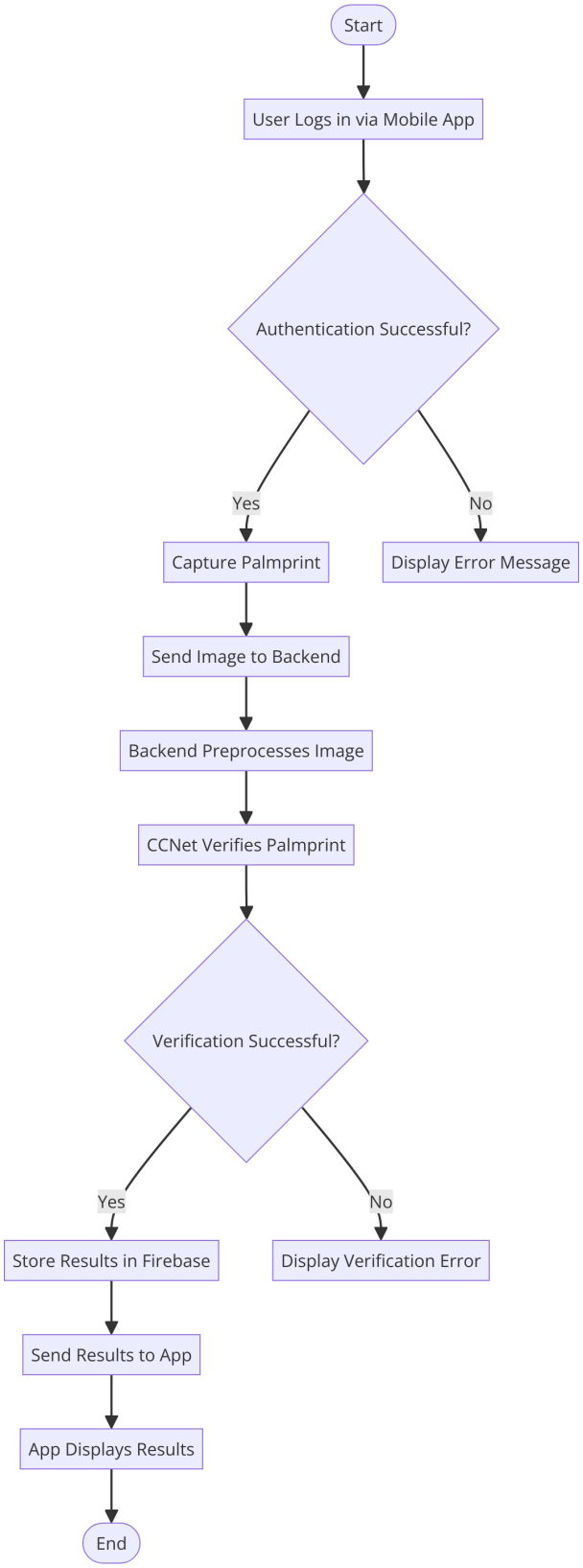


Figure 3: High-Level Design Flowchart for PalmSecure System

The class diagram describes the system's static structure, including core classes and their relationships.

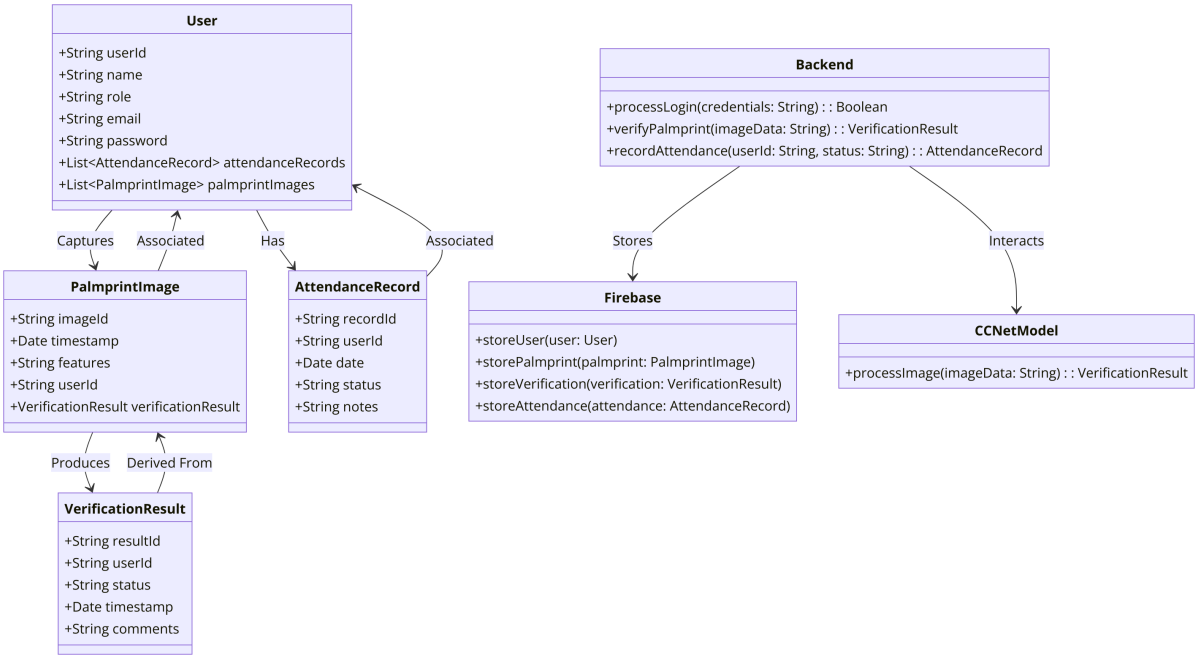


Figure 4: Class Diagram for PalmSecure System

The component diagram provides an overview of the system's modular architecture and interdependencies.

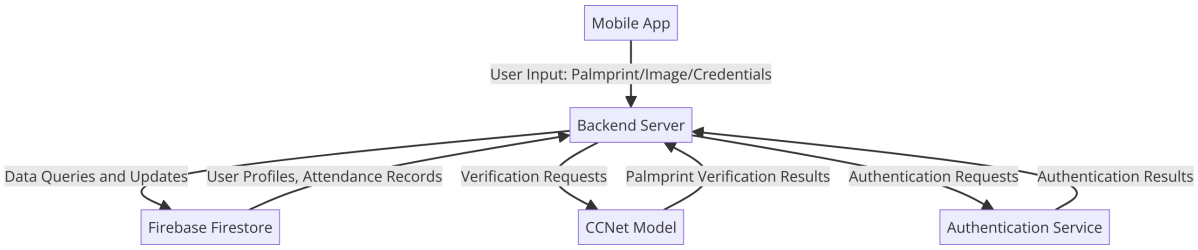


Figure 5: Component Diagram for PalmSecure System

The deployment diagram represents the physical deployment of software components across nodes.

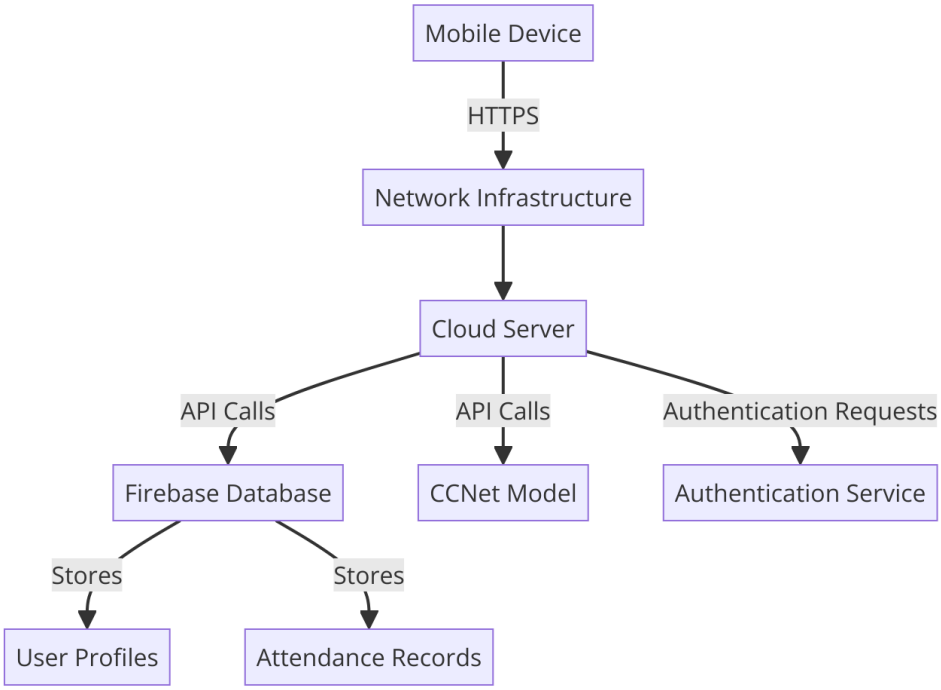


Figure 6: Deployment Diagram for PalmSecure System

By adhering to these design strategies, the *PalmSecure* project ensures a robust, efficient, and scalable system that meets user needs and aligns with project objectives.

5. Detailed System Design

5.1. Database Design

The database design for the *PalmSecure* system leverages Firebase Firestore, a NoSQL cloud-hosted database, to provide real-time data synchronization, flexible schema management, and seamless integration with the overall architecture. This section outlines the key principles, data schema and database rules necessary for implementing an efficient and secure database.

Firestore Design Principles: Unlike traditional relational databases, Firebase Firestore organizes data hierarchically into collections and documents. Each document contains fields that can store various data types, including strings, numbers, booleans, and nested objects, as well as references to sub-collections. This structure is particularly suited for dynamic and scalable applications like *PalmSecure*.

5.1.1 Data Schema Design

The Firebase Firestore database is designed with the following collections and sub-collections to manage the system's core functionalities:

- **Users Collection:**
  - Each document represents a unique user, identified by an auto-generated document ID.
  - Fields include:
    - **firstName:** *String* — The first name of the user.
    - **lastName:** *String* — The last name of the user.
    - **email:** *String* — The user's email address.
    - **role:** *String* — The role of the user (e.g., Admin, Student).
    - **profilePictureURL:** *String* — The URL of the user's profile picture.
    - **userEmbedding:** *String* — The feature vector of the user's biometric.
  - Sub-collections include:
    - **Attendance:**
      - **date:** *Timestamp* — The date of attendance.
      - **status:** *String* — The attendance status (Present/Absent).
- **Logs Collection:**
  - Documents store activity logs such as login times, failed authentication attempts, and system usage metrics.
  - Fields include:
    - **userId:** *String* — Reference to the user's document in the *Users* collection.
    - **action:** *String* — Description of the activity (e.g., login, logout).
    - **timestamp:** *Timestamp* — Time of the logged activity.

Database Rules: Firebase Security Rules are implemented to enforce access control and ensure data security:

- **Role-Based Access:** Users with the role 'Admin' have read/write permissions on all collections, while 'Student' users can only access their documents.
- **Field-Level Validation:** Rules validate input types and ranges to prevent unauthorized data manipulation.

5.1.2 Data Dictionary

The following outlines the key fields and their descriptions:

Users Collection

- **firstName:** User's first name (*String*).
- **lastName:** User's last name (*String*).
- **email:** User's email address (*String*).
- **role:** User's role (*String*, e.g., Admin, Student).
- **profilePictureURL:** URL to the user's profile picture (*String*).
- **userEmbedding:** *String* — The feature vector of the user's biometric.

Attendance Sub-collection

- **date:** Date of attendance (*Timestamp*).
- **status:** Attendance status (*String*, Present/Absent).

Diagrams for Database Design

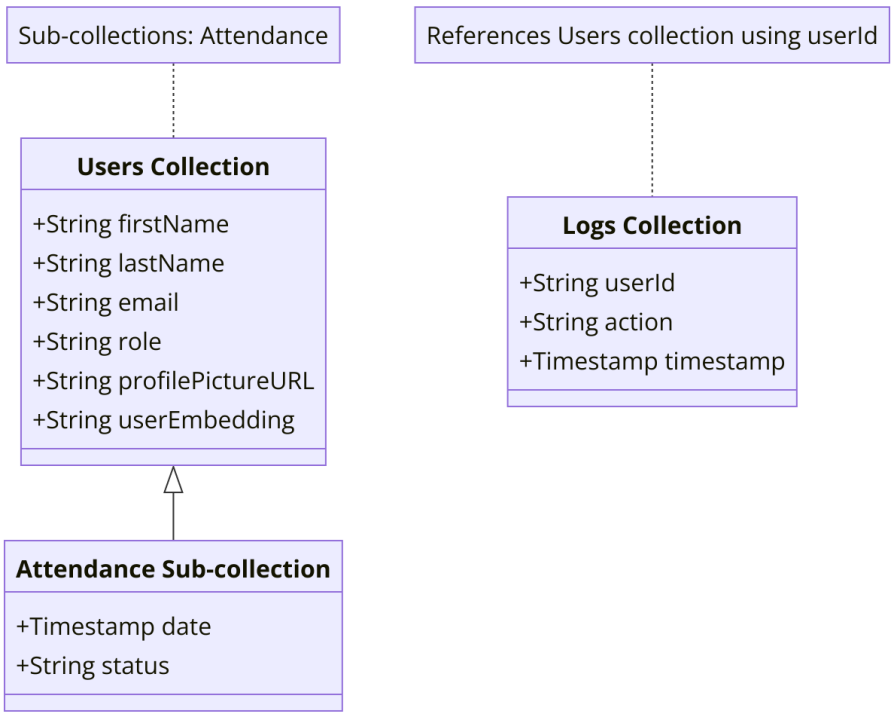


Figure 7: Firebase Database Design Diagram for PalmSecure

Why Firebase?

Firebase was chosen for its:

- **Real-time Synchronization:** Ensures data is updated in real-time across all clients.
- **Flexible Schema:** Supports dynamic data structures without requiring predefined schemas.
- **Security and Scalability:** Provides built-in security rules and seamless scaling for large user bases.

5.2. Application Design

5.2.1 Sequence Diagrams

User Login and Authentication: This sequence diagram illustrates the process of user login and authentication through the mobile application. It highlights the interaction between the user, the mobile app, and Firebase Authentication.

- **Actors:** User (Admin), Mobile Application, Firebase Authentication.
- **Flow:**
  1. The user opens the mobile application and enters login credentials.
  2. The app sends the credentials to Firebase Authentication.
  3. Firebase verifies the credentials and returns a success or failure response.
  4. On success, the app retrieves the user's profile from the 'Users' collection in Firestore.
  5. The app displays the user dashboard.
  6. On failure, an error message is displayed, and the user is prompted to retry.

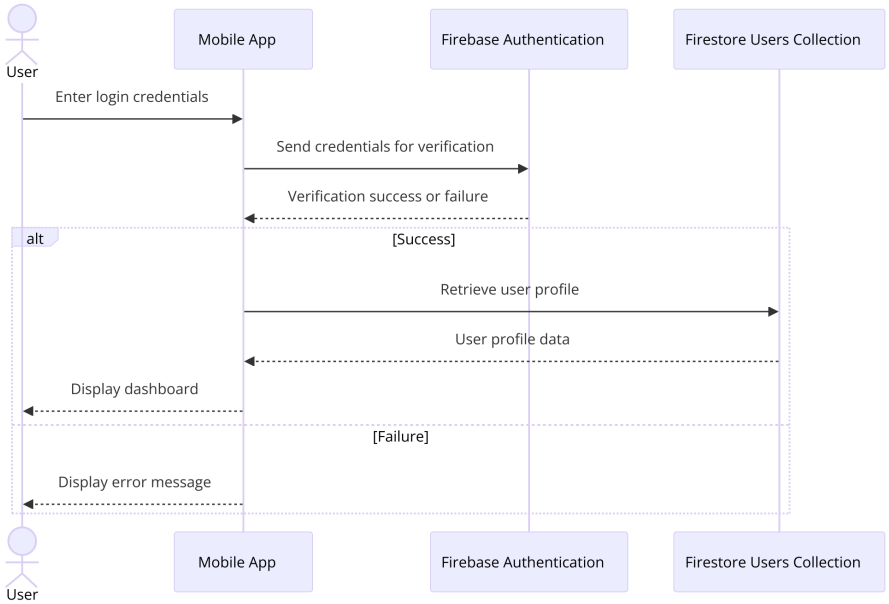


Figure 8: Sequence Diagram for User Login and Authentication

Palmprint Capture and Verification: This sequence diagram demonstrates how the mobile app captures the user's palmprint and verifies it with the backend system.

- **Actors:** User(Student), Mobile Application, Backend API, CCNet Model, Firebase Firestore.
- **Flow:**
  1. The user initiates palmprint verification through the app.
  2. The app captures the palmprint image using the device camera.
  3. The image is sent to the backend via an API call.
  4. The backend preprocesses the image and forwards it to the CCNet recognition model.
  5. The model performs feature extraction and comparison against stored data.
  6. The result (success/failure) is sent back to the backend, logged in Firebase, and returned to the app.
  7. The app displays the verification result to the user.

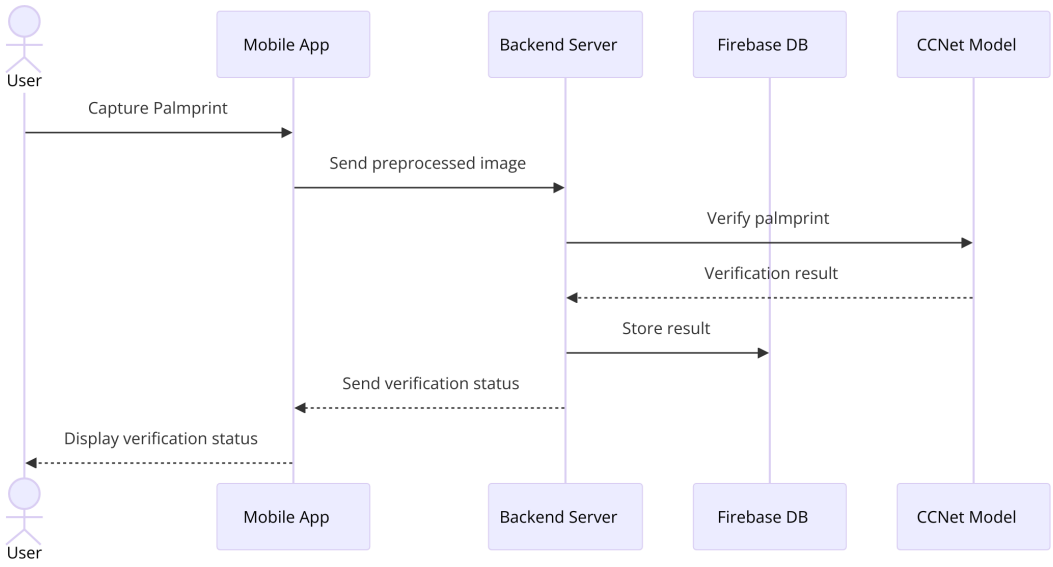


Figure 9: Sequence Diagram for Palmprint Capture and Verification

Attendance Management (Admin Functionality): This sequence diagram illustrates how an admin manages user attendance records in the system.

- **Actors:** Admin, Mobile Application, Firebase Firestore.

- **Flow:**
  1. The admin opens the attendance management module in the app.
  2. The app fetches attendance records from the ‘Attendance‘ sub-collection in Firestore.
  3. The admin modifies or updates specific attendance records.
  4. The app validates the changes and sends them to Firestore for storage.
  5. A confirmation message is displayed to the admin.

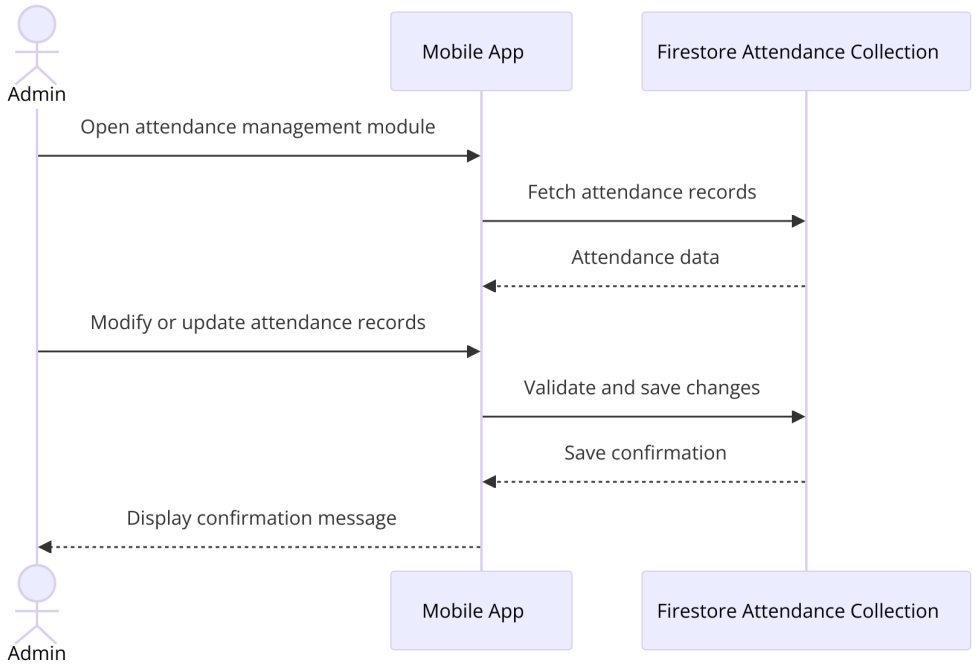


Figure 10: Sequence Diagram for Attendance Management

5.2.2 State Diagrams

Mobile Application States This state diagram describes the various states of the mobile application during its operation.

- **States:**
  - Initializing
  - Idle
  - Awaiting Input
  - Capturing Palmprint
  - Sending Data
  - Waiting for Response
  - Displaying Result
  - Idle

- **Transitions:**
  - Login success leads to *Idle*.
  - User action leads to *Awaiting Input*.
  - Palmprint capture starts the transition to *Capturing Palmprint*, followed by *Sending Data*.
  - Backend response triggers the transition to *Displaying Result*.

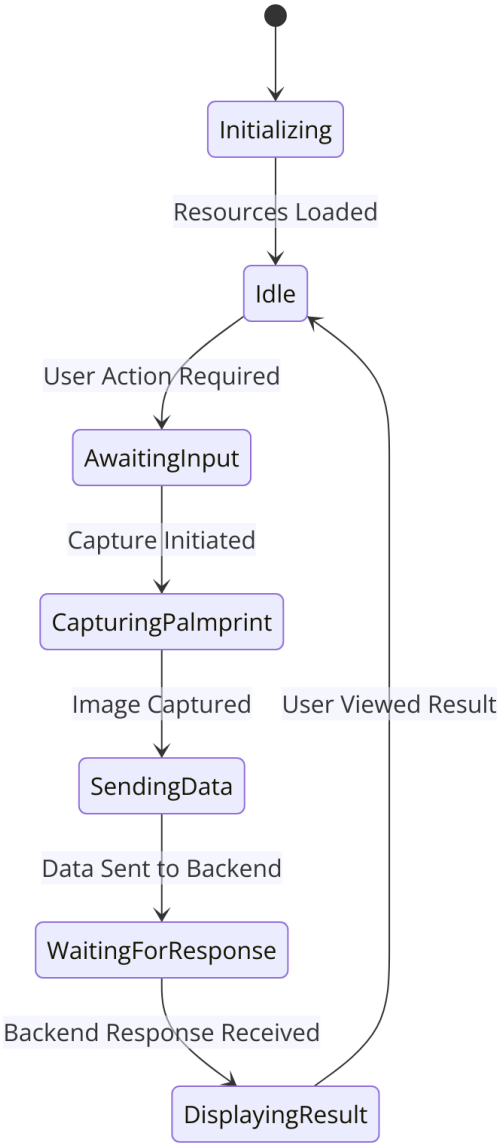


Figure 11: State Diagram for Mobile Application States

Palmprint Verification Process States This state diagram shows the states of the backend system during the palmprint verification process.

- **States:**
  - Idle
  - Receiving Data
  - Preprocessing Data
  - Verifying Palmprint
  - Storing Results
  - Returning Response
- **Transitions:**
  - Data reception triggers the transition to *Preprocessing Data*.
  - Preprocessing completion leads to *Verifying Palmprint*.
  - Verification completion triggers *Storing Results* and *Returning Response*.

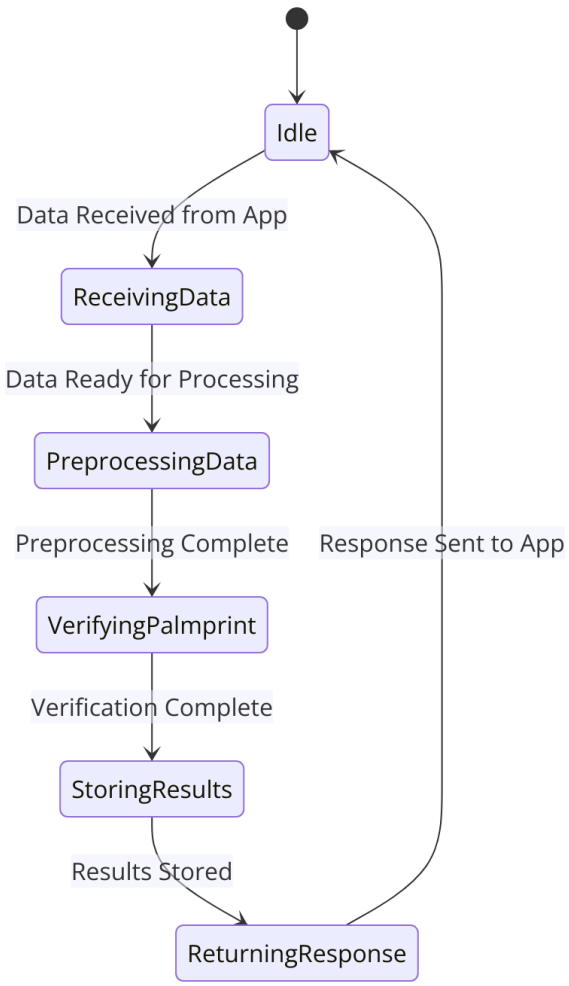


Figure 12: State Diagram for Palmprint Verification Process

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: https://doi.org/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.
   - Google Firebase Documentation. Available at: https://firebase.google.com/docs.

4. **Software Standards:**

   - IEEE Standard 1016-2009, IEEE Recommended Practice for Software Design Descriptions. Available at: https://ieeexplore.ieee.org/document/5290673.

Appendices

Glossary of Terms and Acronyms

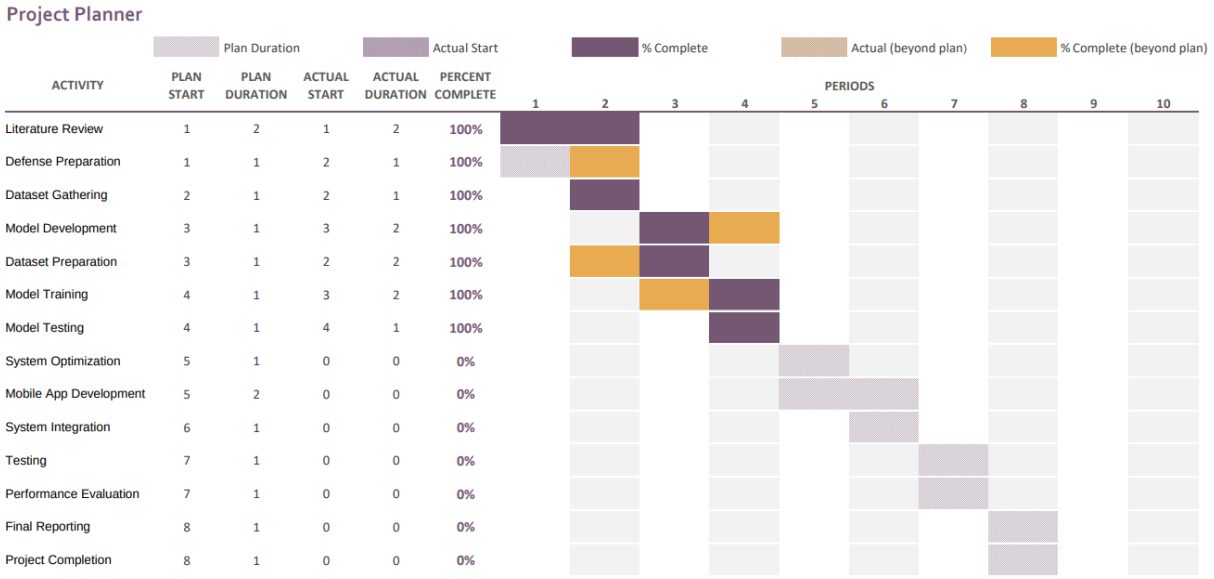
- **Firebase:** A NoSQL cloud-hosted database service provided by Google.
- **UML:** Unified Modeling Language, used for software design visualization.

Test Cases

- **Test Case 1: Palmprint Verification**
  - **Input:** Palmprint image.
  - **Expected Output:** Verification success or failure.
  - **Result:** Pass/Fail.
- **Test Case 2: User Login**
  - **Input:** Valid/Invalid login credentials.
  - **Expected Output:** Access to the system or error message.
  - **Result:** Pass/Fail.

Project Timeline

The project timeline outlines the key phases and milestones for the development of the *PalmSecure* system. The Gantt chart provides a visual representation of the project's schedule, including deliverables, deadlines, and dependencies.



ACTIVITY	PLAN START	PLAN DURATION	ACTUAL START	ACTUAL DURATION	PERCENT COMPLETE
Literature Review	1	2	1	2	100%
Defense Preparation	1	1	2	1	100%
Dataset Gathering	2	1	2	1	100%
Model Development	3	1	3	2	100%
Dataset Preparation	3	1	2	2	100%
Model Training	4	1	3	2	100%
Model Testing	4	1	4	1	100%
System Optimization	5	1	0	0	0%
Mobile App Development	5	2	0	0	0%
System Integration	6	1	0	0	0%
Testing	7	1	0	0	0%
Performance Evaluation	7	1	0	0	0%
Final Reporting	8	1	0	0	0%
Project Completion	8	1	0	0	0%

Figure 13: Gantt Chart for PalmSecure Project Timeline