Project Title: Wristforge

Date: 13th October 2024

Supervisor: Mr. QASIM RIAZ

Group Members: ALI NAUFIL AND M. SAADULLAH

Co-Supervisor: Dr. WAQAR Ahmad

Project Title

Revision History:

Revision History	*Date*	*Comments*
1.00		
2.00		

Document Approval:
The following document has been accepted and approved by the following:

Signature	*Date*	*Name*

List of Contents

1. INTRODUCTION 6

1.1. PURPOSE 6
1.2. PRODUCT SCOPE 6
1.3 OVERVIEW 8

2. THE OVERALL DESCRIPTION 8

2.1. PRODUCT PERSPECTIVE 8

3. WORK BREAKDOWN STRUCTURE 9

Work Breakdown Diagram: 9
.... 9
3.1. Planning Phase 11
3.2. Design Phase 11
3.3. Development Phase 12
3.4. Testing Phase 12
3.5. Deployment Phase 13
3.6. Maintenance and Updates 13

4. DESIGN 14

4.2. Why We Chose This Architecture *........ 30*
4.3. MODULE IDENTIFICATION 31

5. 4+1 ARCHITECTURE VIEW MODEL 34

USE CASE VIEW 34
LOGICAL VIEW: 35
DEVELOPMENT VIEW 36
PROCESS VIEW 37
PHYSICAL VIEW 38
USER INTERFACE DESIGN 39

List of Figures

Figure 1: Work Breakdown Structure 10
Figure 2: Use Case Diagram 16
Figure 3: Class Diagram 17
Figure 4: Activity Diagram 18
Figure 5: Interaction Diagram 19
Figure 6: Component Diagram 20
Figure 7: Communication Diagram 21
Figure 8: Deployment Diagram 22
Figure 9: Package Diagram 23
Figure 10: Object Diagram 24
Figure 11: Composite Structure Diagram 25
Figure 12: Profile Diagram 26
Figure 13: State Machine Diagram 27
Figure 14: Sequence Diagram 28
Figure 15: Timing Diagram 29
Figure 16: Use Case View 34
Figure 17: Logical View 35
Figure 18: Logical View 35
Figure 19: Development View 36
Figure 20: Process View 37
Figure 21: Physical View 38
Figure 22: User Registration 39
Figure 23: Smartwatch Connection 39
Figure 24:Home Screen 40
Figure 25: User Authorization 40
Figure 26: OTP 40
Figure 27: Login 41
Figure 28: Heart Rate 41
Figure 29: Step Count 41
Figure 30: LED Control 42
Figure 31: De-Authentication Attack 42
Figure 32: Packet Tracing 42
Figure 33: Beacon Attack 43
Figure 34: Probe Attack 43
Figure 35: Attack Confirmation 43
Figure 36: Settings 44
Figure 37: Display Preferences 44
Figure 38: Security Settings 44
Figure 39: Permission Management 45
Figure 40: Other Preferences 45
Figure 41: Notification Preferences 45
Figure 42: Vibration Preferences 46

Figure 43: Watch Customization 46
Figure 44: Display Brightness 46
Figure 45: Serial number to connect to app 47
Figure 46: Home Screen 47
Figure 47: Menu screen 48
Figure 48: User Verification 48
Figure 49: OTP 48
Figure 50: Heart Rate 49
Figure 51: Step Count 49
Figure 52: De-Authentication Attack 49
Figure 53: Packet Tracing 50
Figure 54: Beacon Attack 50
Figure 55: Probe Attack 50
Figure 56: Attack Confirmation 51

1. INTRODUCTION

1.1. PURPOSE

The purpose of this Software Requirements Specification (SRS) document is to outline the functional and non-functional requirements for the development of **Wrist Forge**, a multifunctional smartwatch. The document will serve as a guide for developers and designers to ensure that the project meets the desired objectives and provides clear instructions on features, functionalities, and constraints.

Wrist Forge is intended to combine traditional fitness tracking features such as heart rate monitoring and step counting with advanced cybersecurity tools. These include network packet tracing, de-Authentication attacks, and other security monitoring capabilities. This document will provide detailed requirements for both the smartwatch and its companion mobile application, ensuring proper integration between the two systems and delivering a seamless user experience for both fitness and cybersecurity enthusiasts.

1.2. PRODUCT SCOPE

Wrist Forge aims to create a unique smartwatch that addresses the needs of two primary target audiences: **fitness enthusiasts** and **security-conscious users**. The product will offer a comprehensive set of fitness tracking features alongside advanced network security tools, giving users the ability to monitor their health and their digital environment simultaneously.

Wrist Forge is designed to fill a gap in the market by offering advanced security features alongside traditional fitness functionalities, making it a versatile device for users who value both personal health and digital safety. The product will be built using an **ESP-8266** board for its network capabilities, with the core focus on providing an intuitive and efficient user interface for switching between fitness and security modes

Table 1: Terms used in this document and their description

Name	Description
Wrist Forge	The smartwatch being developed for fitness tracking and cybersecurity monitoring, designed with advanced features for network security.
ESP-8266	A low-cost Wi-Fi microcontroller used in the smartwatch to enable network communication and security operations.
De-Authentication Attack	A type of cyberattack where the smartwatch disrupts Wi-Fi connections by sending de-authentication frames, part of the device's network security features.
Packet Tracing	A network security feature that monitors and captures data packets sent over Wi-Fi networks, allowing users to track network activity.
Companion App	The mobile application that pairs with the smartwatch to display health data, network logs, and allows users to configure settings for both fitness and security features.
Step Counting	A fitness feature of the smartwatch that tracks the number of steps taken by the user during their daily activities.
Heart Rate Monitor	A sensor in the smartwatch that tracks the user's heart rate in real time and provides alerts for anomalies.
Probe Attack	A network scanning method used by the smartwatch to detect available Wi-Fi networks in the vicinity.
LED Indicator	A light on the smartwatch used for notifications and alerts, including health alerts and network activity signals.
Beacon Flooding Attack	A type of network attack where the smartwatch sends fake Wi-Fi beacons to create illegitimate networks and flood the local area.
Data Synchronization	The process of ensuring that health and network data collected by the smartwatch is updated in real-time or periodically in the companion app.
Firmware Update	Software updates delivered to the smartwatch that enhance functionality, improve performance, and maintain security features.
Firebase	A cloud-based platform by Google that provides real-time database services and authentication for mobile and web applications, used to store and sync data between the smartwatch and the companion app in this project.

1.3 OVERVIEW

WristForge is a cutting-edge smartwatch that combines fitness tracking and cybersecurity testing in a single microcontroller**(ESP-8266)** powered wearable. It features essential health monitoring tools such as heart rate sensing and step counting, alongside powerful cybersecurity capabilities. The device allows for network packet sniffing, deauthentication attacks, probe attacks, and beacon flooding, specifically targeting the 2.4 GHz Wi-Fi spectrum to test vulnerabilities in wireless networks. These features make WristForge an ideal tool for both fitness enthusiasts and cybersecurity professionals, offering a seamless integration of health management and network security assessment.

2. THE OVERALL DESCRIPTION

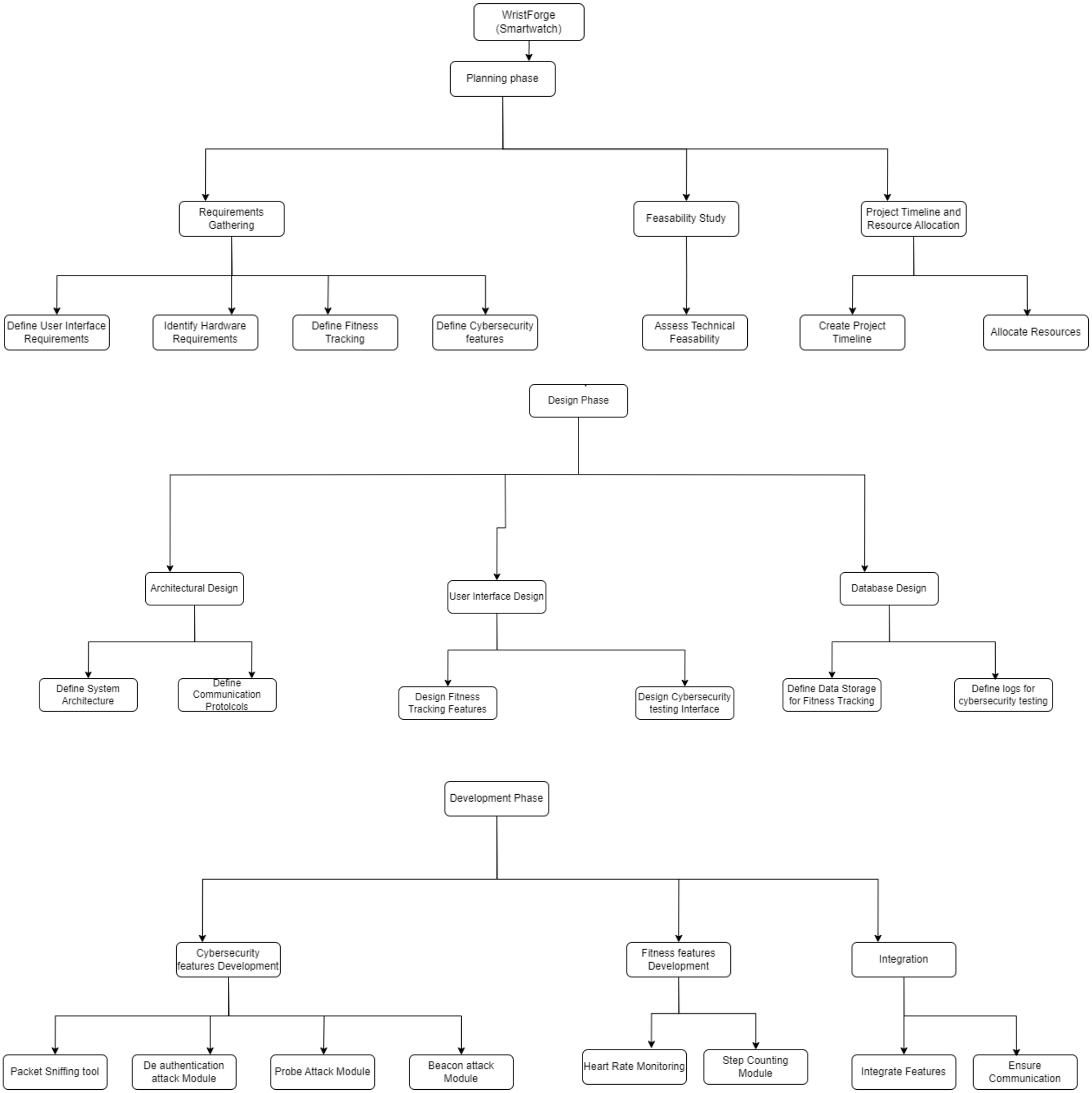
Wrist Forge is an advanced smartwatch developed as part of a final year project that integrates health tracking features with network security tools. It combines typical fitness functionalities, such as heart rate monitoring and step counting, with sophisticated cybersecurity operations like packet tracing and De-Authentication attacks. The device is designed for both fitness enthusiasts and users interested in network security, providing real-time monitoring and user-friendly controls via a companion mobile app.

2.1. Product Perspective

Wrist Forge is a unique combination of fitness tracking and network security features. Unlike traditional smartwatches, which mainly focus on health monitoring, Wrist Forge integrates cybersecurity capabilities such as Wi-Fi packet monitoring, probe attacks, and beacon flooding. These features are accessed and controlled through a mobile companion app, which also serves as the central interface for syncing health data and configuring security operations. This product fits into the niche of wearable technology for tech-savvy users who are interested in personal fitness and network security.

3. WORK BREAKDOWN STRUCTURE

Work Breakdown Diagram:


WristForge (Smartwatch)
Planning phase
Requirements Gathering
Feasability Study
Project Timeline and Resource Allocation
Define User Interface Requirements
Identify Hardware Requirements
Define Fitness Tracking
Define Cybersecurity features
Assess Technical Feasability
Create Project Timeline
Allocate Resources
Design Phase
Architectural Design
User Interface Design
Database Design
Define System Architecture
Define Communication Protolcols
Design Fitness Tracking Features
Design Cybersecurity testing Interface
Define Data Storage for Fitness Tracking
Define logs for cybersecurity testing
Development Phase
Cybersecurity features Development
Fitness features Development
Integration
Packet Sniffing tool
De authentication attack Module
Probe Attack Module
Beacon attack Module
Heart Rate Monitoring
Step Counting Module
Integrate Features
Ensure Communication

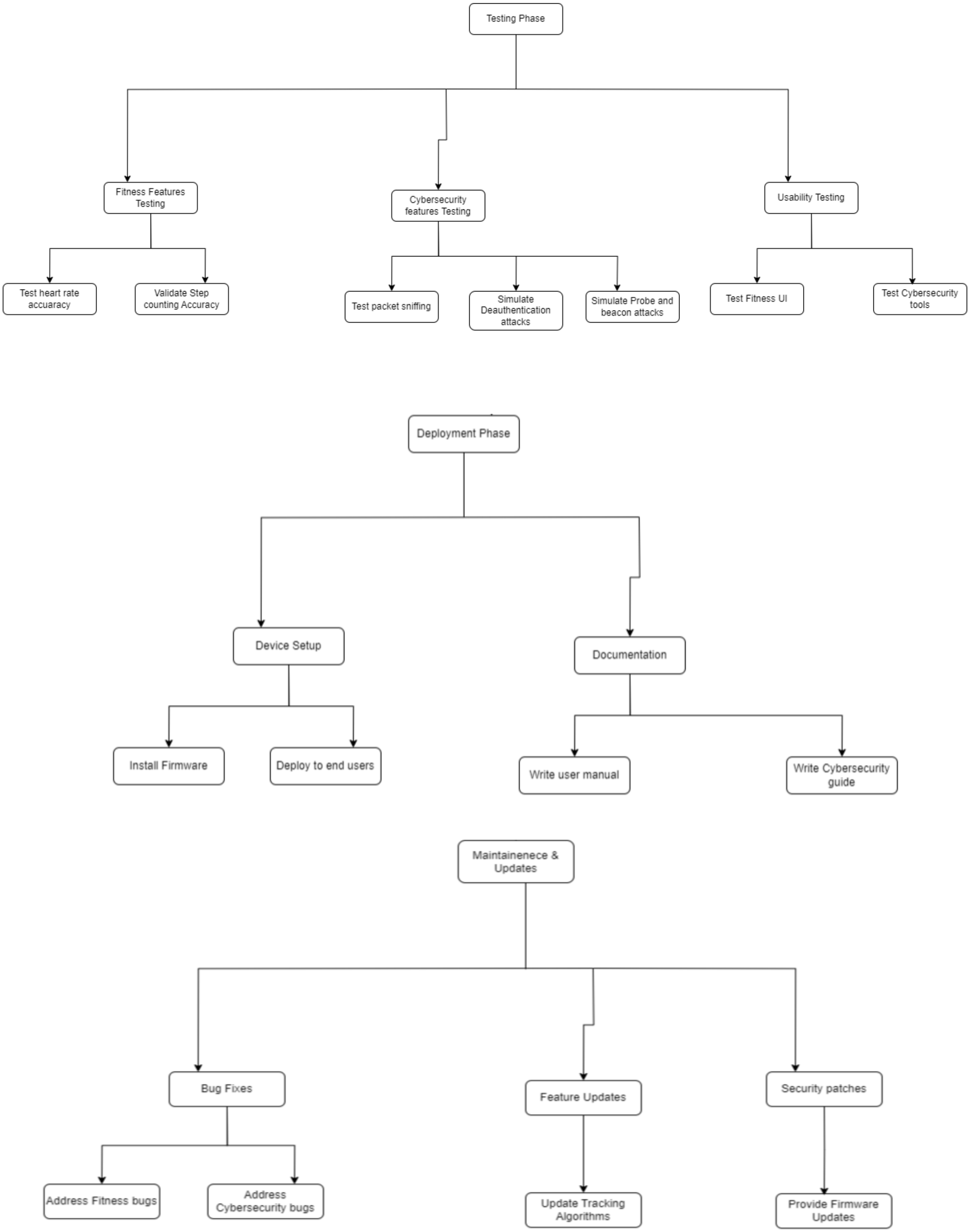


Figure 1: Work Breakdown Structure

3.1. Planning Phase

3.1.1. Requirements Gathering

- Define Fitness Tracking Features (Heart Rate, Step Counting)
- Define Cybersecurity Features (Packet Sniffing, Deauthentication, Probe Attack, Beacon Flooding)
- Identify Hardware Requirements (Microcontroller, Sensors)
- Define User Interface Requirements.

3.1.2. Feasibility Study

- Assess Technical Feasibility (Cybersecurity and Fitness Features)
- Estimate Cost and Resources

3.1.3. Project Timeline and Resource Allocation

- 1.3.1. Create Project Timeline
- 1.3.2. Allocate Resources (Hardware, Software, Development Team)

3.2. Design Phase

3.2.1. Architecture Design

- 2.1.1. Design System Architecture (Hardware, Software Integration)
- 2.1.2. Define Communication Protocols (Wi-Fi Module, Sensor Communication)

3.2.2. User Interface Design

- 2.2.1. Design Fitness Tracking Interface
- 2.2.2. Design Cybersecurity Testing Interface (Packet Sniffing, Attack Options)

3.2.3. Database Design

- 2.3.1. Define Data Storage for Fitness Tracking
- 2.3.2. Define Logs for Cybersecurity Testing (Attack History, Network Data)

3.3. Development Phase

3.3.1. Fitness Features Development

- 3.1.1. Heart Rate Monitoring Module
- 3.1.2. Step Counting Module

3.3.2. Cybersecurity Features Development

- 3.2.1. Packet Sniffing Tool
- 3.2.2. Deauthentication Attack Module
- 3.2.3. Probe Attack Module

- 3.2.4. Beacon Flooding Attack Module

3.3.3. Integration

- Integrate Fitness and Cybersecurity Features
- Ensure Communication Between Components (Sensors, Microcontroller)

3.4. Testing Phase

3.4.1. Fitness Features Testing

- 4.1.1. Test Accuracy of Heart Rate Monitoring
- 4.1.2. Validate Step Counting Accuracy

3.4.2. Cybersecurity Features Testing

- 4.2.1. Test Packet Sniffing on Various Networks
- 4.2.2. Simulate and Test Deauthentication Attacks
- 4.2.3. Simulate Probe and Beacon Flooding Attacks

3.4.3. Usability Testing

- 4.3.1. Test User Interface for Fitness Tracking
- 4.3.2. Test Usability of Cybersecurity Tools

3.5. Deployment Phase

3.5.1. Device Setup

- 5.1.1. Install Firmware on Devices
- 5.1.2. Deploy to End Users for Beta Testing

3.5.2. Documentation

- 5.2.1. Write User Manual for Fitness Features
- 5.2.2. Write Cybersecurity Testing Guide

3.5.3. Release and Support

- 5.3.1. Provide Ongoing Technical Support
- 5.3.2. Implement Post-Launch Updates

3.6. Maintenance and Updates

3.6.1. Bug Fixes

- 6.1.1. Address Bugs from Fitness Features
- 6.1.2. Address Bugs from Cybersecurity Tools

3.6.2. Feature Updates

- 6.2.1. Update Fitness Tracking Algorithms
- 6.2.2. Introduce New Cybersecurity Testing Tools

3.6.3. Security Patches

- 6.3.1. Provide Regular Firmware Updates

4. Design

4.1 Architectural Design

The architectural design of **WristForge** integrates both fitness tracking and cybersecurity functionalities into a single, cohesive system. The architecture is structured to ensure efficient processing of sensor data, secure communication, and seamless integration of both fitness and network testing features. Below is an overview of the proposed architecture:

4.1.1. Layered Architecture

The system follows a **layered architecture** model to separate concerns and facilitate modular development. Each layer has a specific function and interacts with other layers through well-defined interfaces.

4.1.2. Components of WristForge Architecture:

4.1.2.1 Hardware Layer

ESP8266 Microcontroller: Acts as the brain of the device, managing communication, computation, and data processing.

Fitness Sensors: Includes sensors for heart rate monitoring and step counting.

Wi-Fi Module: Handles network communications for cybersecurity testing (packet sniffing, deauthentication, probe attacks, beacon flooding).

4.1.2.2 Device Abstraction Layer

Sensor Data Collection Module: Gathers real-time data from fitness sensors (heart rate, step counter).

Network Interaction Module: Interfaces with the Wi-Fi module for cybersecurity features, capturing and sending network packets or performing attacks.

4.1.2.3. Application Logic Layer

Fitness Tracking Service: Processes and analyzes the data from fitness sensors to display heart rate and step counts.

Cybersecurity Testing Service: Provides functionality to perform deauthentication attacks, probe attacks, beacon flooding, and packet sniffing. Manages attack initiation and data collection.

4.1.2.4. User Interface Layer

Display Module: Shows the current heart rate, step count, and status of cybersecurity tests (e.g., ongoing attacks, network scanning results).

Control Interface: Allows users to switch between fitness and cybersecurity modes, start tests, and view results.

4.1.2.5. Data Storage and Management

Fitness Data Storage: Locally stores data on heart rate and step count, which can be synced to an external app or device.

Cybersecurity Logs: Keeps track of network test results and captured packets for later analysis.

4.1.2.6. Communication Layer

Wi-Fi Communication Protocol: Handles data transfer between the smartwatch and external devices, such as a companion app or cloud service for updates, or network testing results.

USE CASE DIAGRAM:

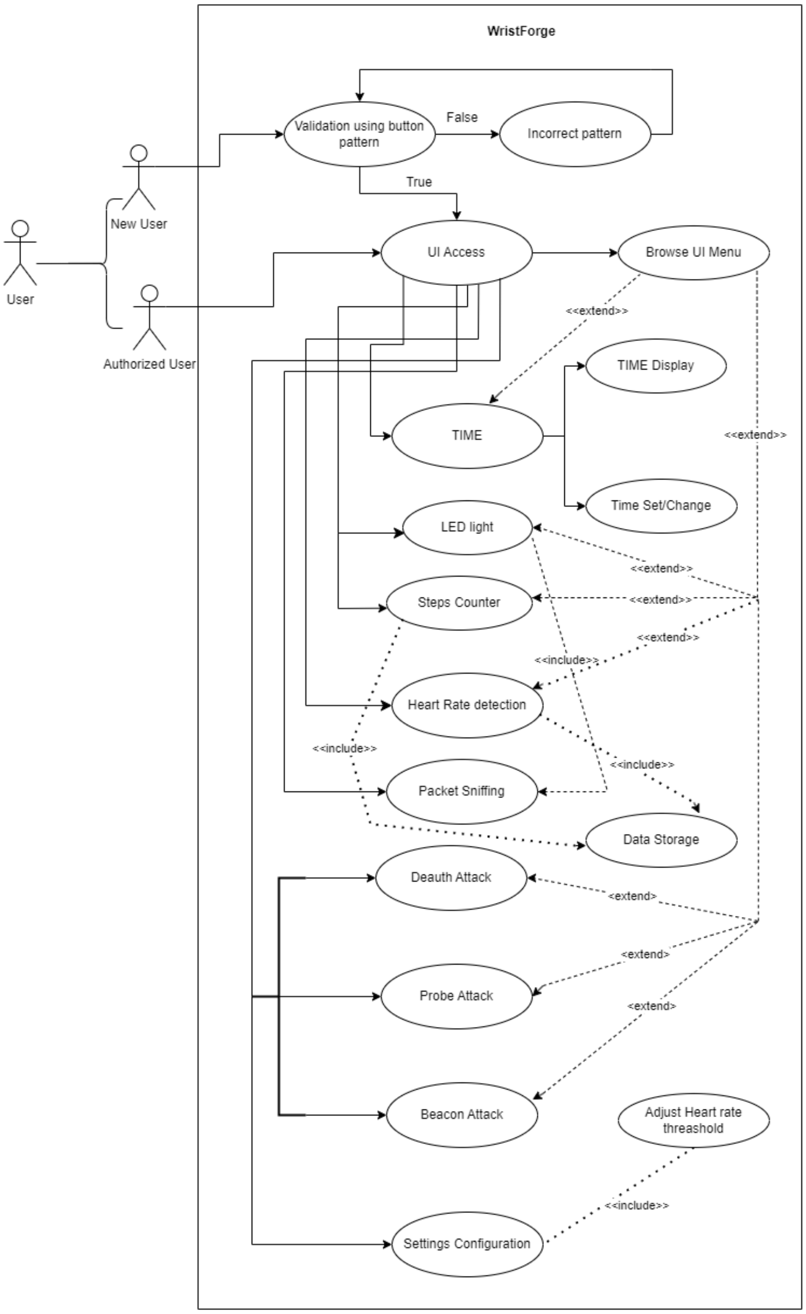


Figure 2: Use Case Diagram

Project Title

CLASS DIAGRAM:

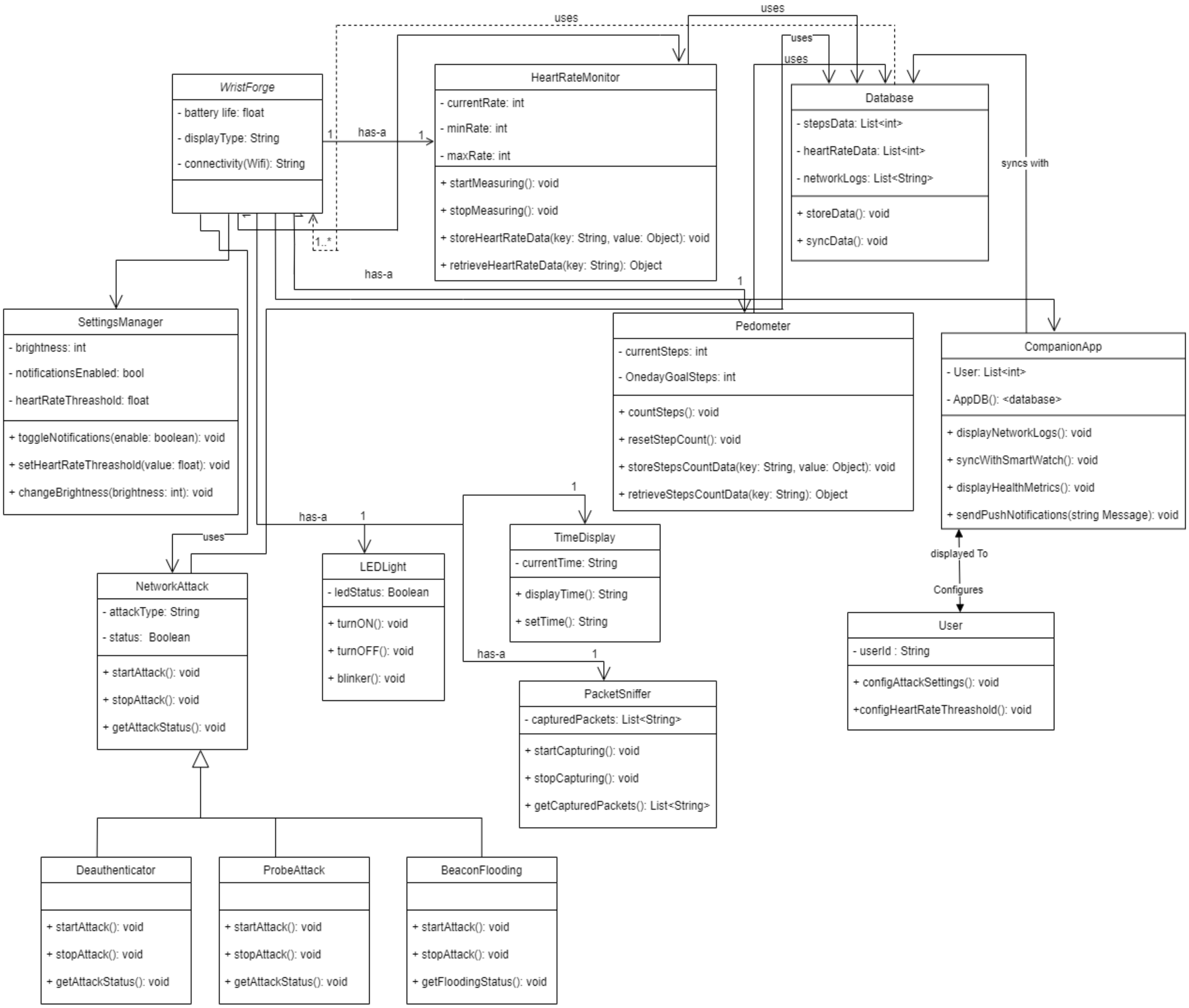


Figure 3: Class Diagram

Project Title

ACTIVITY DIAGRAM:

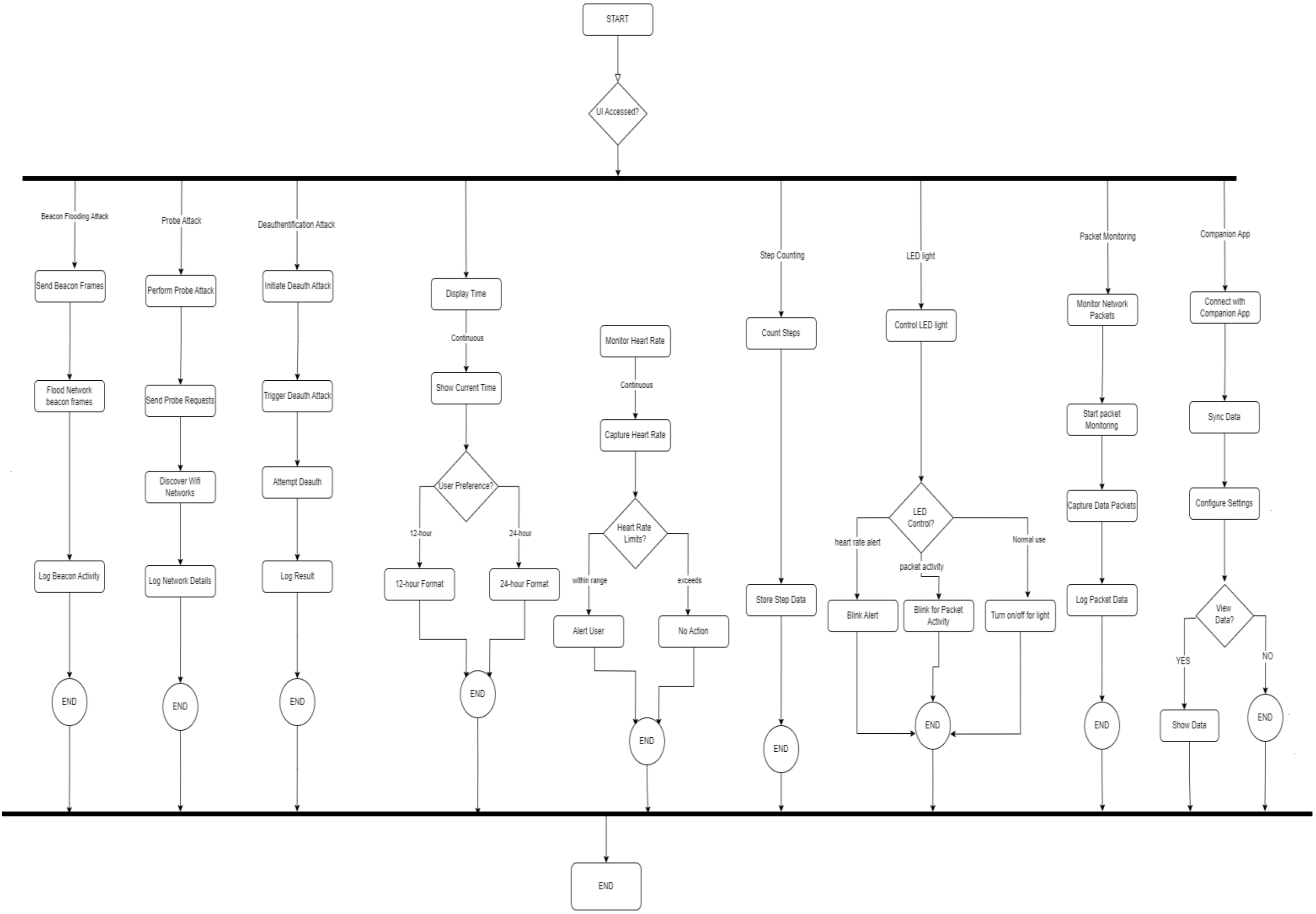


Figure 4: Activity Diagram

INTERACTION DIAGRAM:

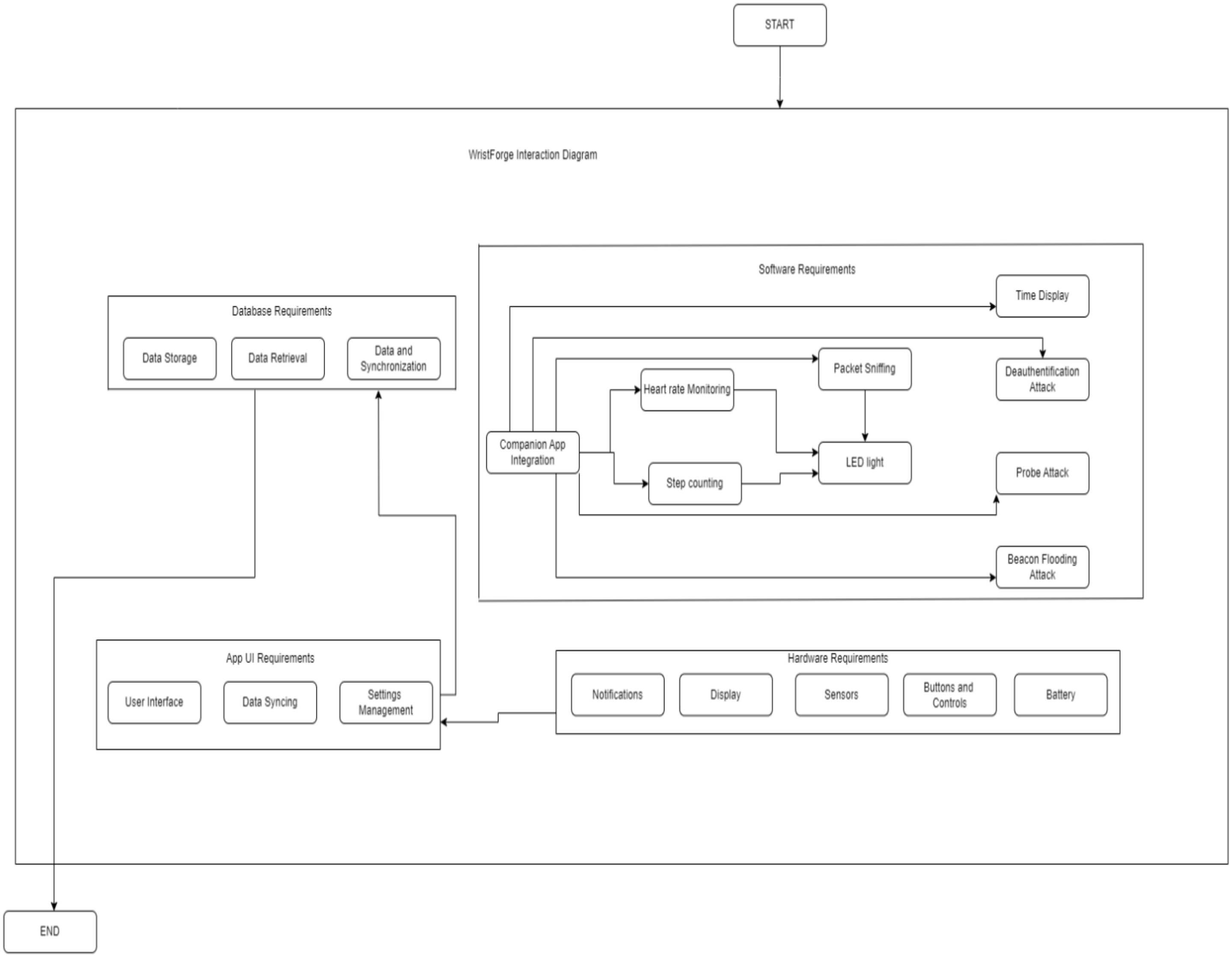


Figure 5: Interaction Diagram

COMPONENT DIAGRAM:

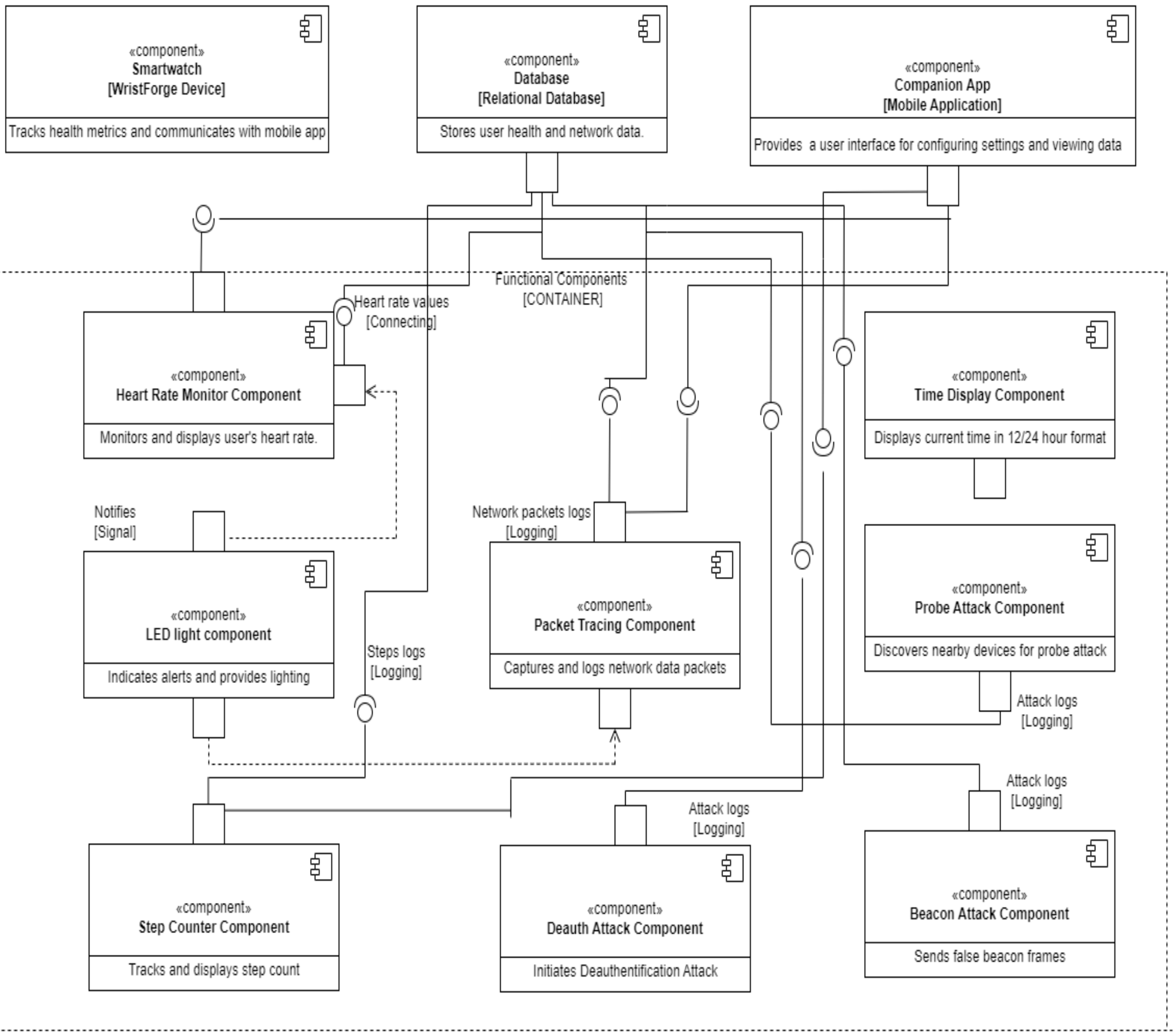


Figure 6: Component Diagram

COMMUNICATION DIAGRAM:

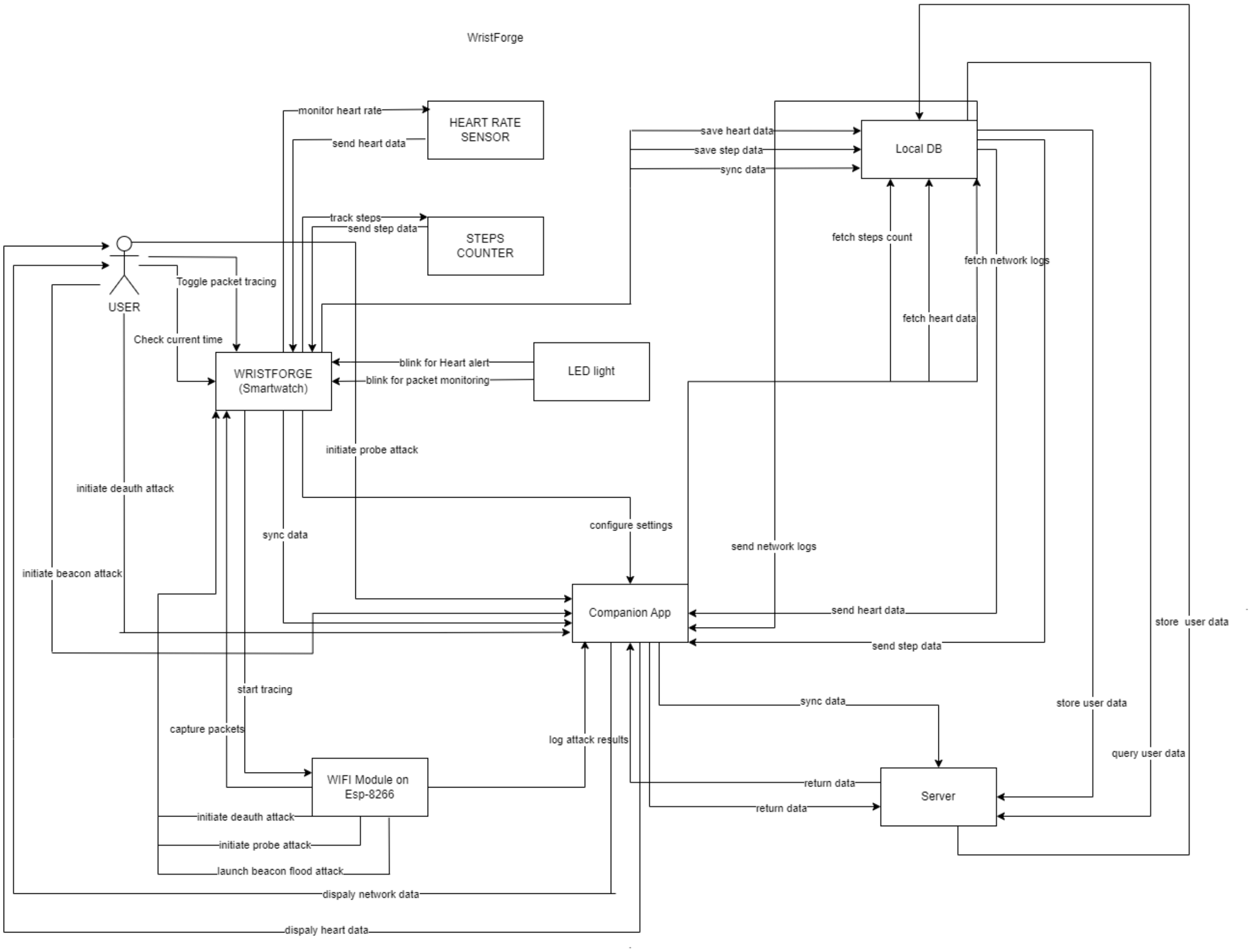


Figure 7: Communication Diagram

DEPLOYMENT DIAGRAM:

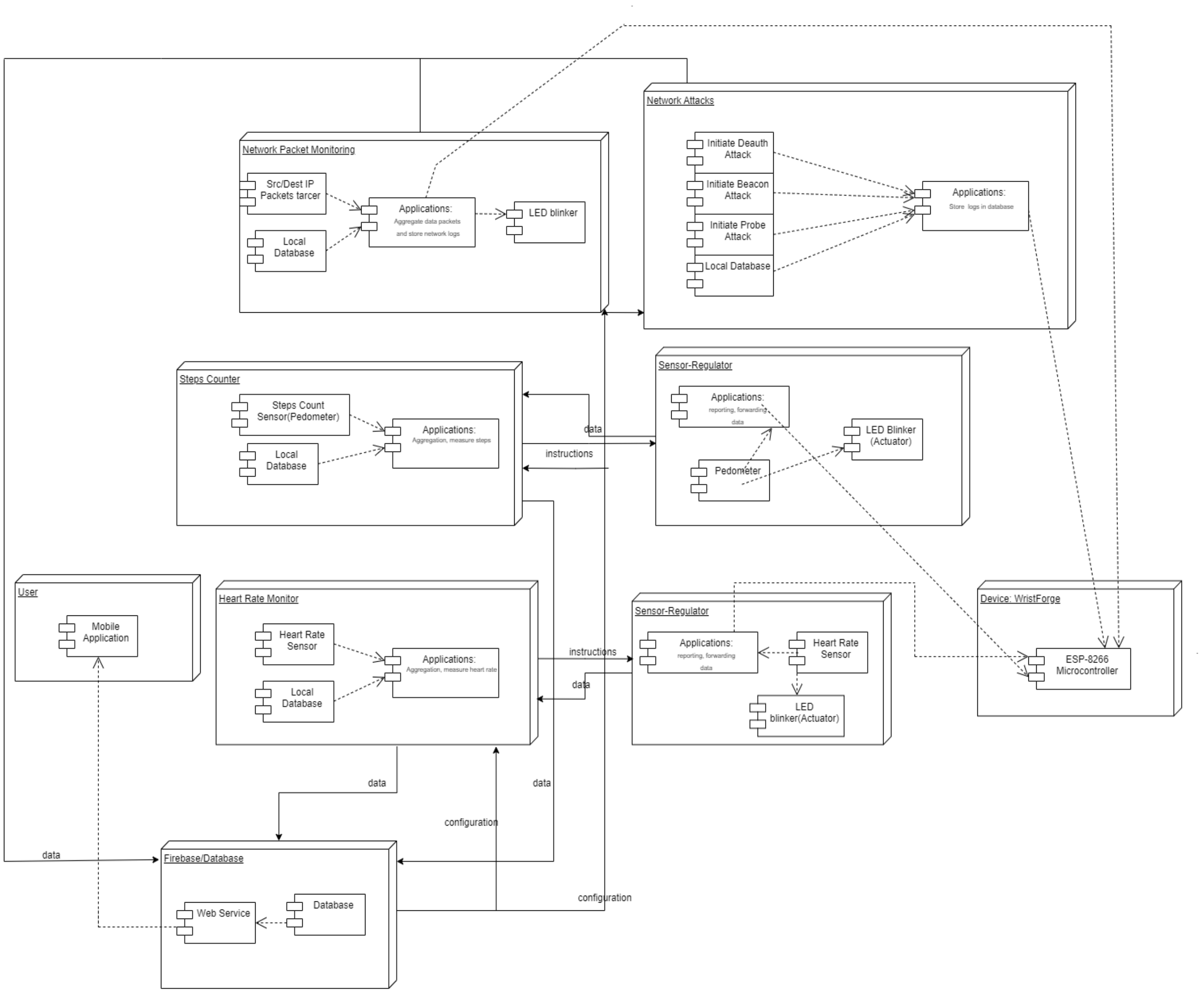


Figure 8: Deployment Diagram

PACKAGE DIAGRAM:

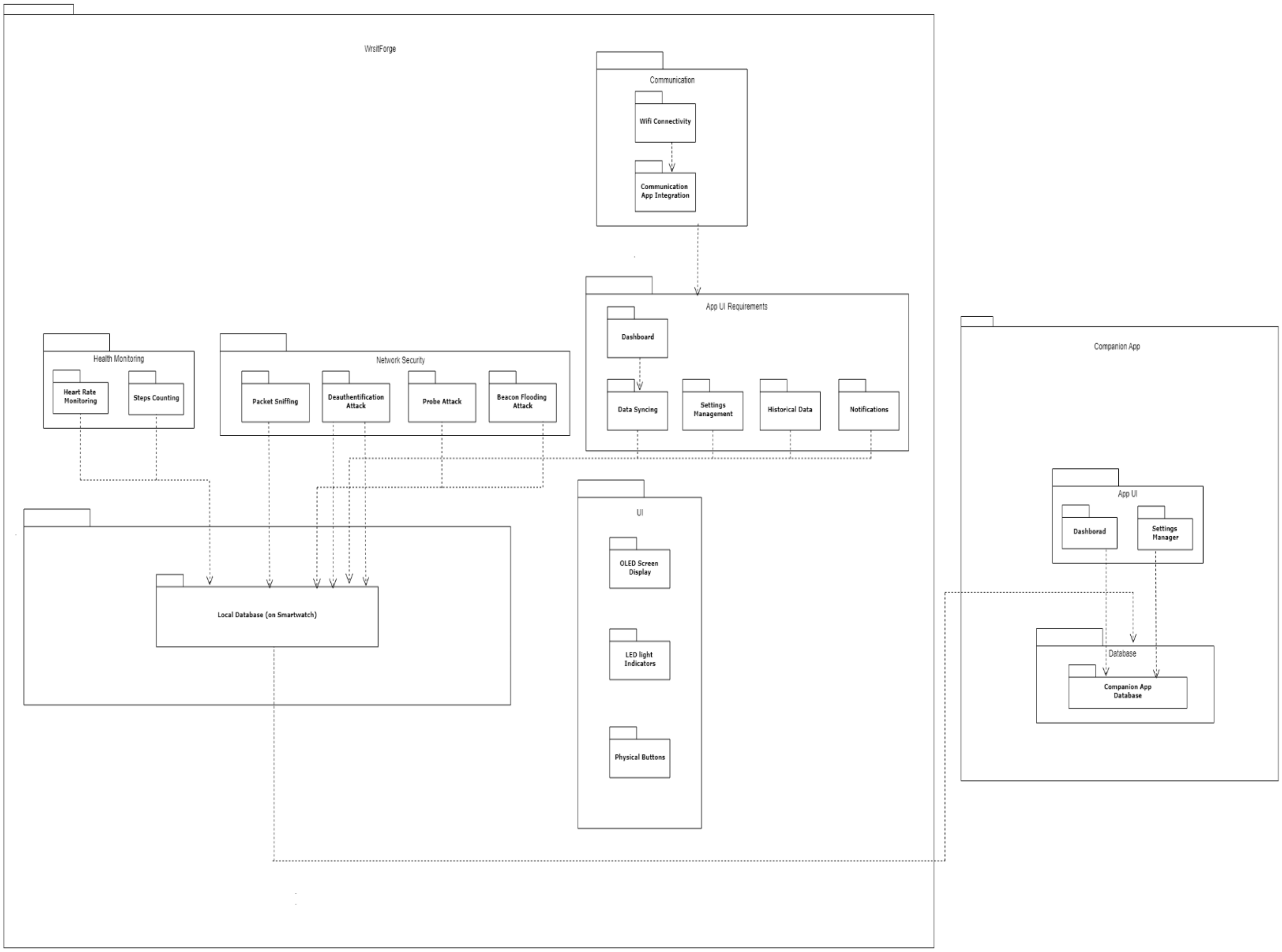


Figure 9: Package Diagram

OBJECT DIAGRAM:

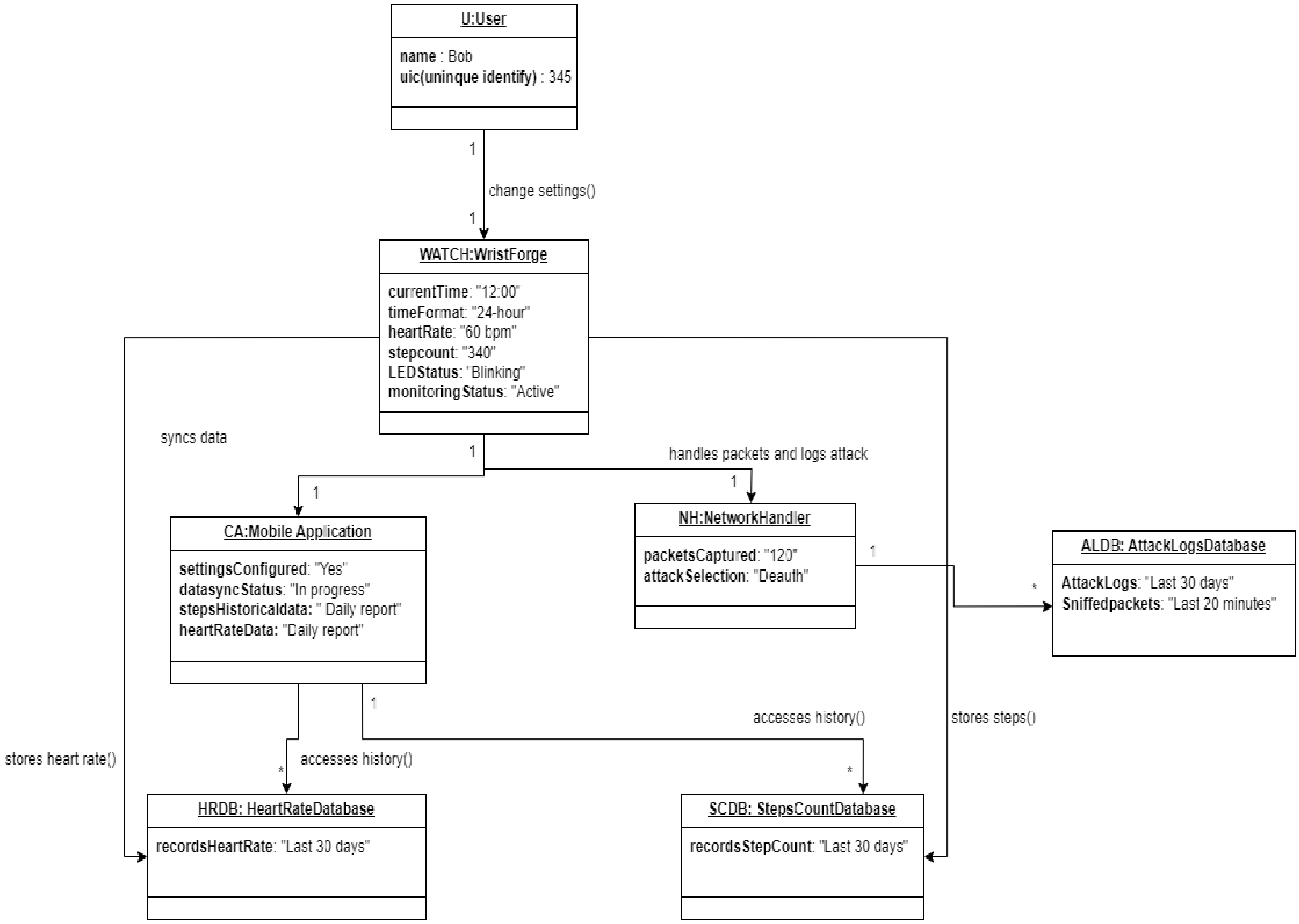


Figure 10: Object Diagram

Project Title

COMPOSITE STRUCTURE DIAGRAM:

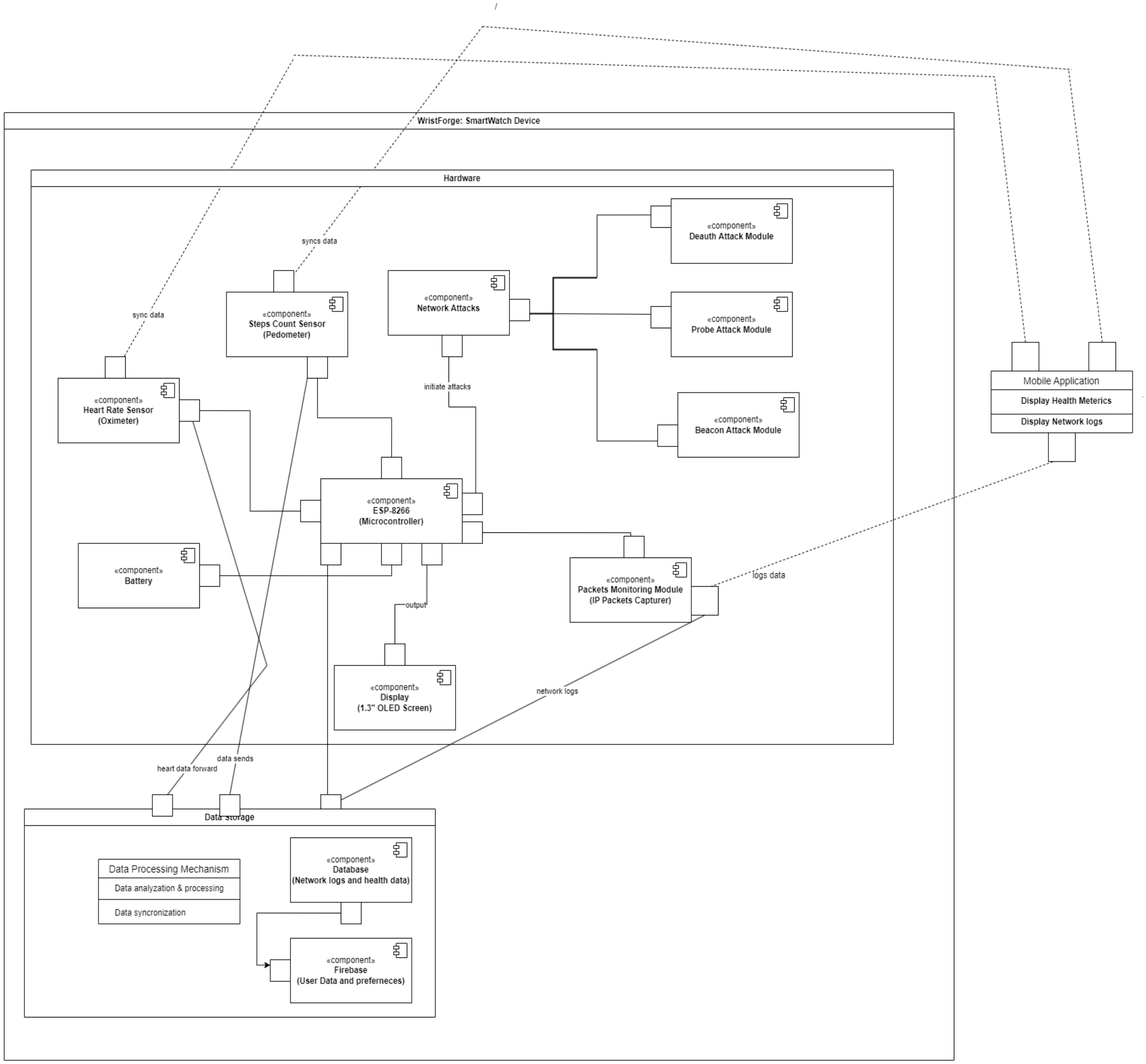


Figure 11: Composite Structure Diagram

PROFILE DIAGRAM:

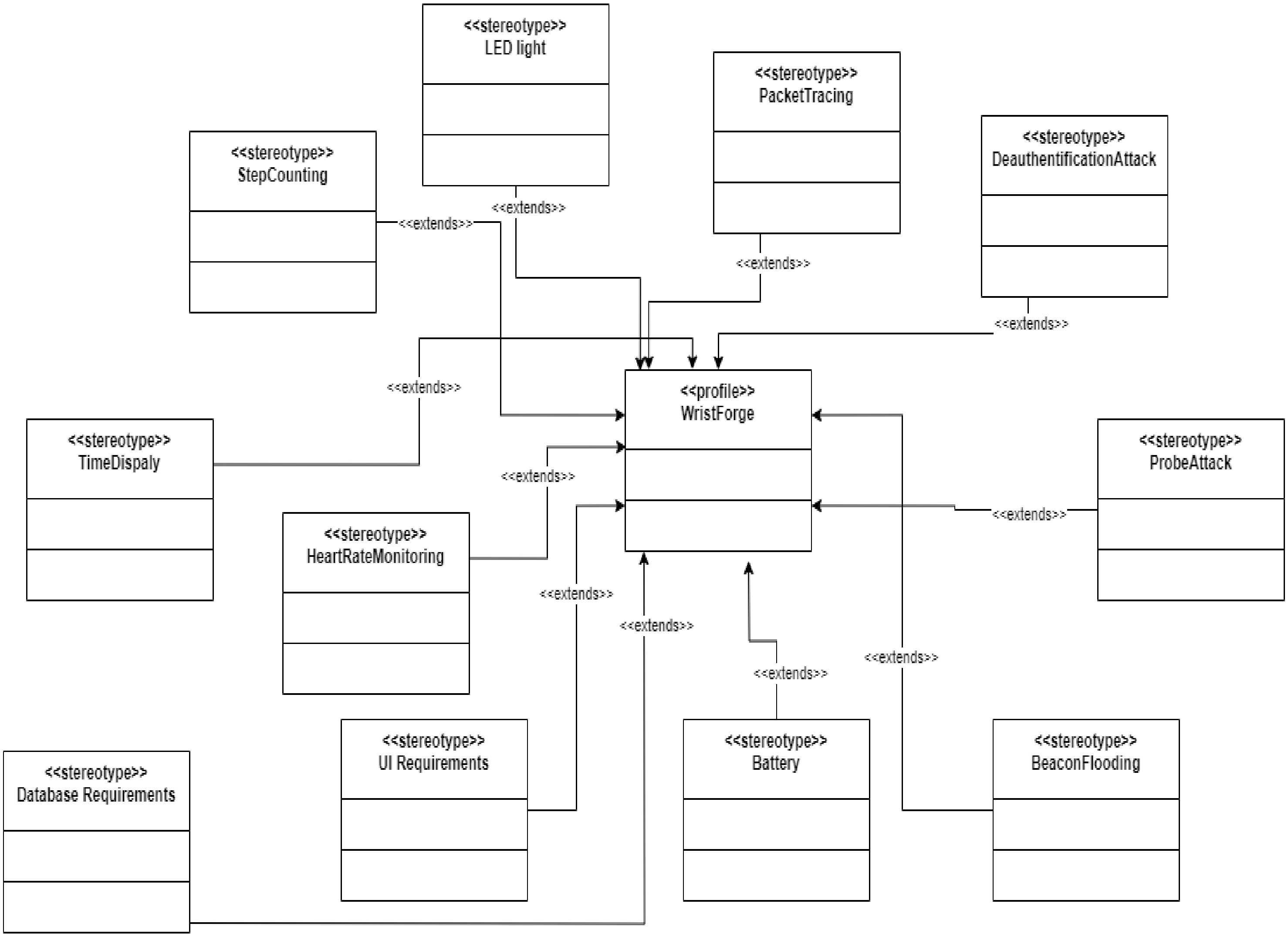


Figure 12: Profile Diagram

STATE MACHINE DIAGRAM:

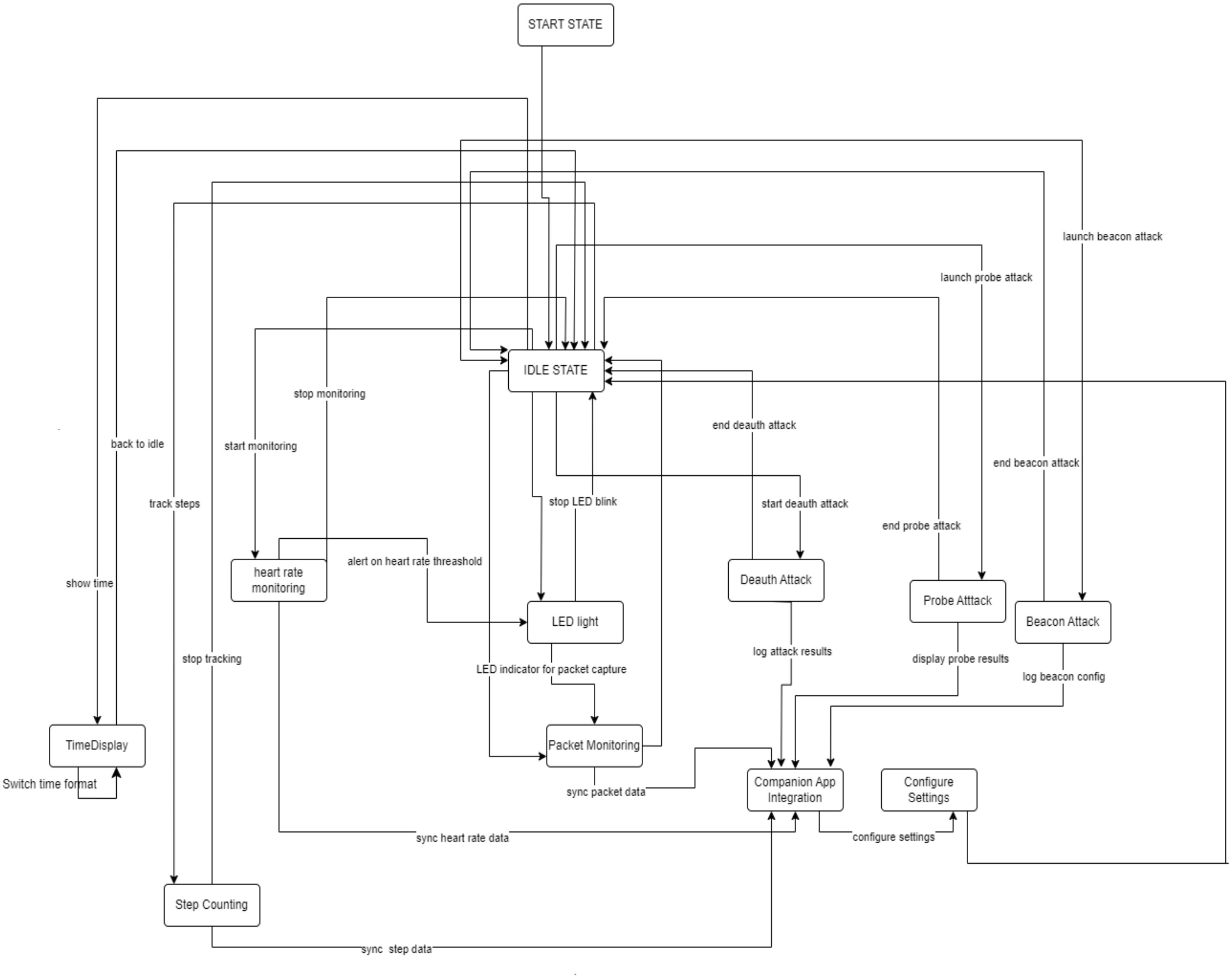


Figure 13: State Machine Diagram

SEQUENCE DIAGRAM:

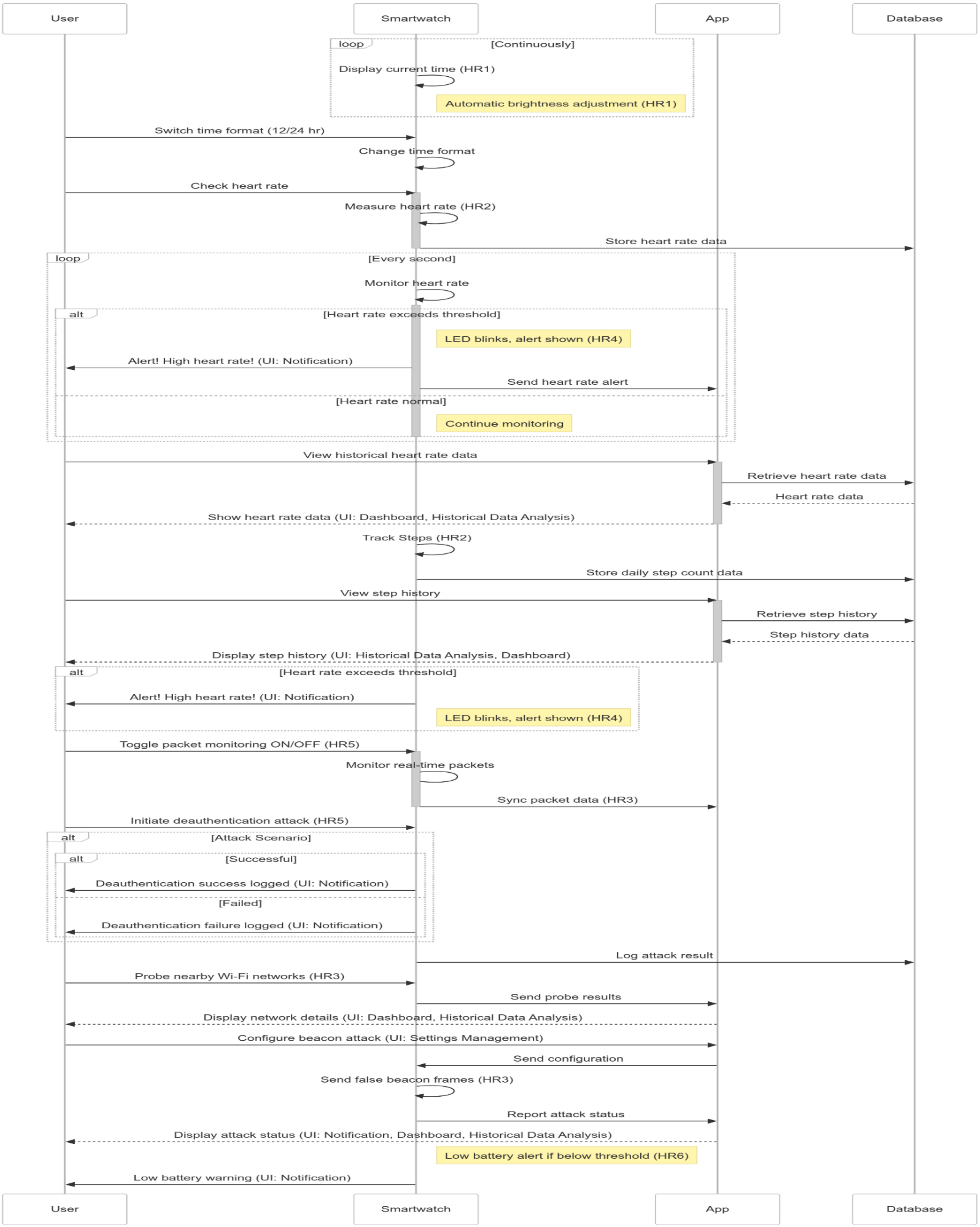

User
Smartwatch
App
Database
loop
[Continuously]
Display current time (HR1)
Automatic brightness adjustment (HR1)
Switch time format (12/24 hr)
Change time format
Check heart rate
Measure heart rate (HR2)
Store heart rate data
loop
[Every second]
Monitor heart rate
alt
[Heart rate exceeds threshold]
LED blinks, alert shown (HR4)
Alert! High heart rate! (UI: Notification)
Send heart rate alert
[Heart rate normal]
Continue monitoring
View historical heart rate data
Retrieve heart rate data
Heart rate data
Show heart rate data (UI: Dashboard, Historical Data Analysis)
Track Steps (HR2)
Store daily step count data
View step history
Retrieve step history
Step history data
Display step history (UI: Historical Data Analysis, Dashboard)
alt
[Heart rate exceeds threshold]
Alert! High heart rate! (UI: Notification)
LED blinks, alert shown (HR4)
Toggle packet monitoring ON/OFF (HR5)
Monitor real-time packets
Sync packet data (HR3)
Initiate deauthentication attack (HR5)
alt
[Attack Scenario]
alt
[Successful]
Deauthentication success logged (UI: Notification)
[Failed]
Deauthentication failure logged (UI: Notification)
Log attack result
Probe nearby Wi-Fi networks (HR3)
Send probe results
Display network details (UI: Dashboard, Historical Data Analysis)
Configure beacon attack (UI: Settings Management)
Send configuration
Send false beacon frames (HR3)
Report attack status
Display attack status (UI: Notification, Dashboard, Historical Data Analysis)
Low battery alert if below threshold (HR6)
Low battery warning (UI: Notification)
User
Smartwatch
App
Database

Figure 14: Sequence Diagram

TIMING DIAGRAM:

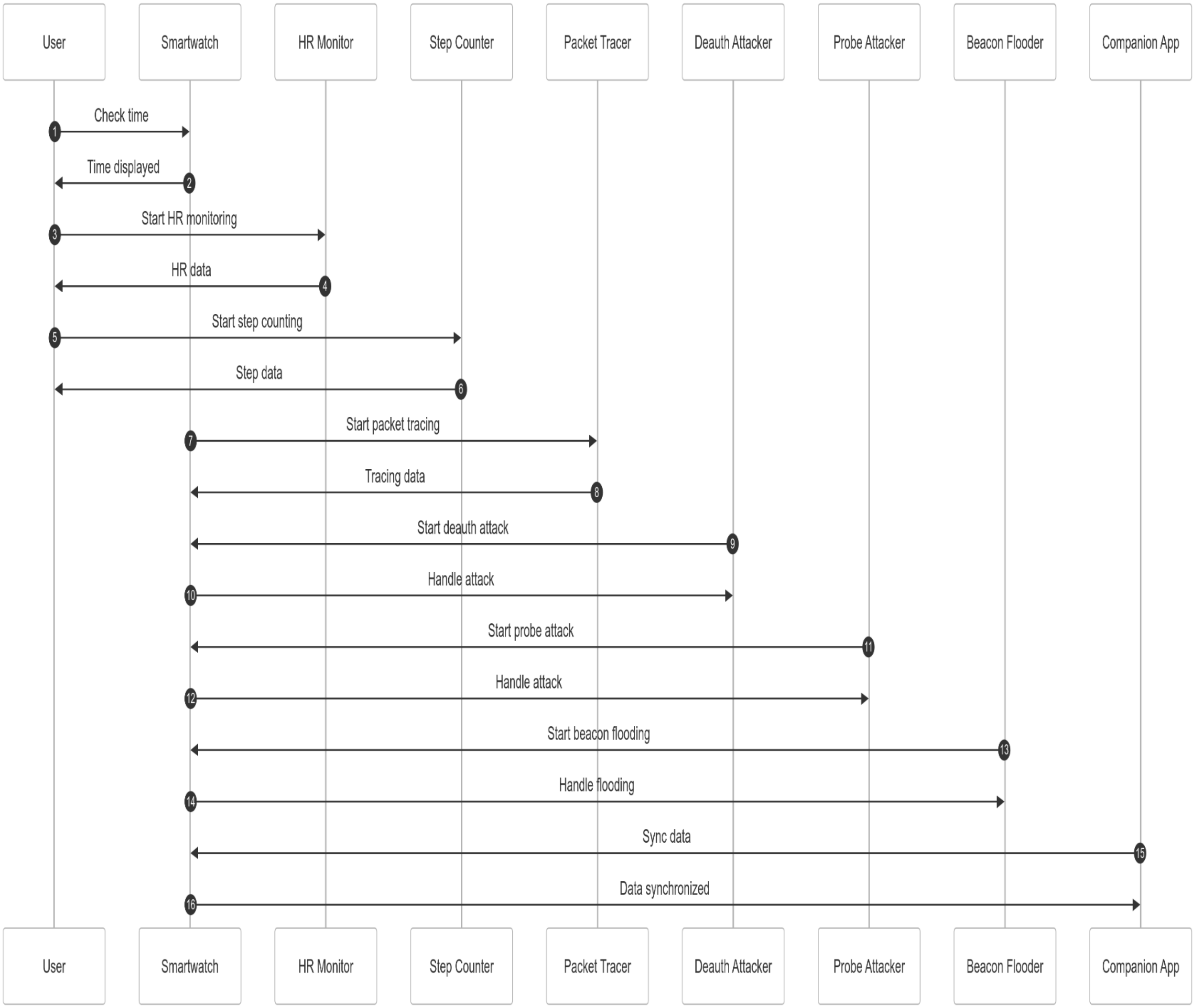


Figure 15: Timing Diagram

4.2. Why We Chose This Architecture

4.2.1. Modularity:

Layered architecture allows us to separate different functionalities of the WristForge. The fitness tracking and cybersecurity testing systems are kept distinct, simplifying both development and maintenance. This separation ensures that fitness features won’t interfere with the more complex cybersecurity tools.

4.2.2. Scalability and Extendibility:

The modular design ensures that new features (e.g., additional fitness metrics or new cybersecurity testing techniques) can be added easily without disrupting the existing architecture. For instance, we can introduce new network attacks or fitness sensors without reworking the entire system.

4.2.3. Real-Time Performance:

By using the ESP8266 microcontroller, which is designed for efficient, low-power Wi-Fi applications, WristForge can perform real-time network monitoring and attacks while simultaneously collecting fitness data. The architecture supports quick data processing, which is essential for both fitness metrics and cybersecurity activities.

4.2.4. Security:

The layered approach allows us to implement security measures at various stages. For example, we can ensure that Wi-Fi communication is secure and encrypted, while also keeping logs of network testing results, which are important for forensic analysis and security audits.

4.2.5. Ease of Use:

The separation of the user interface layer ensures that users have a simple, intuitive interface to switch between fitness and cybersecurity modes, view data, and manage tests. The design is user-centric, ensuring that even non-technical users can benefit from its functionalities.

4.2.6. Low Power Consumption:

Given that this is a wearable device, power efficiency is critical. The layered architecture, combined with the efficient use of ESP8266 and fitness sensors, ensures minimal energy consumption while still providing the necessary functionality.

4.2.7. Data Integrity and Syncing:

The data storage layer allows the fitness data and cybersecurity logs to be securely stored and potentially synced with external applications. This allows users to track fitness progress and cybersecurity test results over time, even when not actively using the smartwatch.

4.2.8. Support for Future Integration:

The architecture makes it easy to integrate with external apps, cloud storage, or machine learning models in the future, either for advanced fitness analytics or cybersecurity anomaly detection.

4.3. MODULE IDENTIFICATION

4.2.1. Hardware Interaction Module

- **Description**: This module manages direct interaction with the physical components of WristForge, such as the microcontroller, sensors, and Wi-Fi module.
- **Responsibilities**:
  - Communicate with the **ESP8266 microcontroller** to control the device's operations.
  - Interface with fitness sensors (e.g., heart rate, step counter) for data collection.
  - Manage Wi-Fi operations for cybersecurity features.
- **Key Components**:
  - Microcontroller Driver
  - Sensor Drivers (Heart Rate, Accelerometer)
  - Wi-Fi Communication Controller

4.2.2. Fitness Data Processing Module

- **Description**: This module handles the real-time collection and processing of fitness-related data from sensors, such as heart rate and step count.
- **Responsibilities**:
  - Gather raw data from heart rate and accelerometer sensors.

  - Process and analyse the data to derive meaningful metrics (e.g., step count, heart rate trends).
  - Send processed data to the user interface for display.
- **Key Components**:
  - Heart Rate Data Processor
  - Step Counting Algorithm
  - Fitness Data Aggregator

4.2.3. Cybersecurity Testing Module

- **Description**: This module provides all the tools needed to perform cybersecurity tests, such as network packet sniffing and Wi-Fi attacks. It is the core of the device's security testing capabilities.
- **Responsibilities**:
  - Execute **packet sniffing** to capture nearby Wi-Fi traffic.
  - Perform **deauthentication attacks**, **probe attacks**, and **beacon flooding** to assess network vulnerabilities.
  - Log network traffic and test results for analysis.
- **Key Components**:
  - Packet Sniffer Engine
  - Deauthentication Attack Engine
  - Probe Request Simulation
  - Beacon Flooding Engine
  - Attack Results Logger

4.2.4. User Interface (UI) Module

- **Description**: This module is responsible for rendering the user interface, displaying fitness metrics, and providing control over cybersecurity functions.
- **Responsibilities**:
  - Display real-time fitness data (heart rate, steps) to the user.
  - Allow users to switch between fitness and cybersecurity modes.
  - Enable interaction with cybersecurity tools (start/stop attacks, view results).
- **Key Components**:
  - Fitness Dashboard
  - Cybersecurity Tools Dashboard
  - Mode Switching Interface

4.2.5. Data Storage and Sync Module

- **Description**: This module stores data locally on the device, including fitness tracking data and logs from cybersecurity tests. It may also sync data with external apps or cloud services.
- **Responsibilities**:
  - Save heart rate, step count, and other fitness data for long-term tracking.
  - Store logs of network tests and captured packets for analysis.
  - Sync data with external devices or apps, if required (e.g., mobile companion app).
- **Key Components**:
  - Fitness Data Storage
  - Cybersecurity Logs Storage
  - Sync Manager

4.2.6. Communication and Networking Module

- **Description**: This module manages all networking functions, enabling communication between WristForge and other devices or services via Wi-Fi.
- **Responsibilities**:
  - Manage Wi-Fi connection for both network testing (sniffing, attacks) and data syncing.
  - Ensure secure communication between WristForge and external devices (e.g., mobile app, cloud services).
- **Key Components**:
  - Wi-Fi Connectivity Manager
  - Data Sync Handler
  - Secure Communication Protocol (for secure data exchange)

4.2.7. Power Management Module

- **Description**: This module optimizes the power consumption of the device, ensuring that it can run efficiently without draining the battery quickly.
- **Responsibilities**:
  - Monitor power consumption across different modules.
  - Activate low-power modes when certain functions are not in use (e.g., during idle periods).
  - Manage battery life and power-saving settings.
- **Key Components**:
  - Power Usage Monitor
  - Low Power Mode Controller
  - Battery Status Manager

5. 4+1 ARCHITECTURE VIEW MODEL

Use Case View

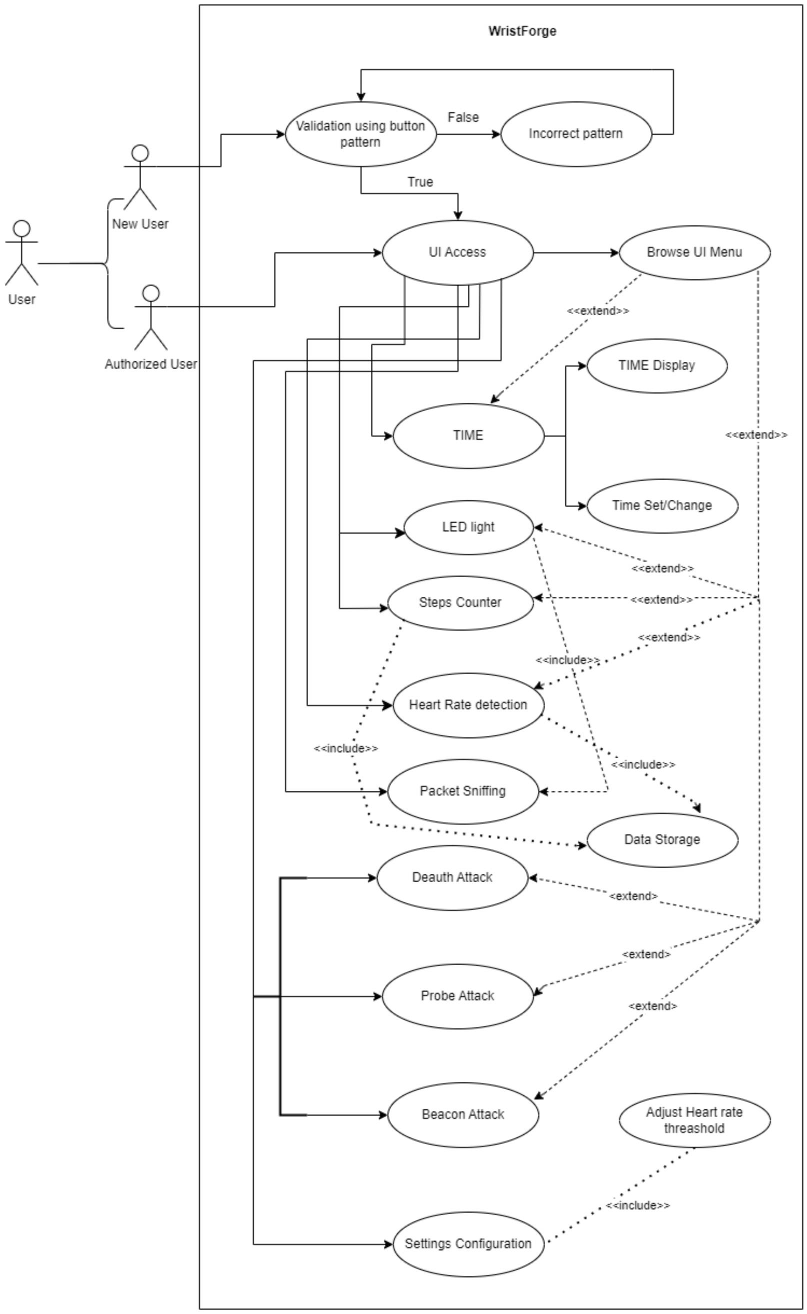


Figure 16: Use Case View

Logical View:

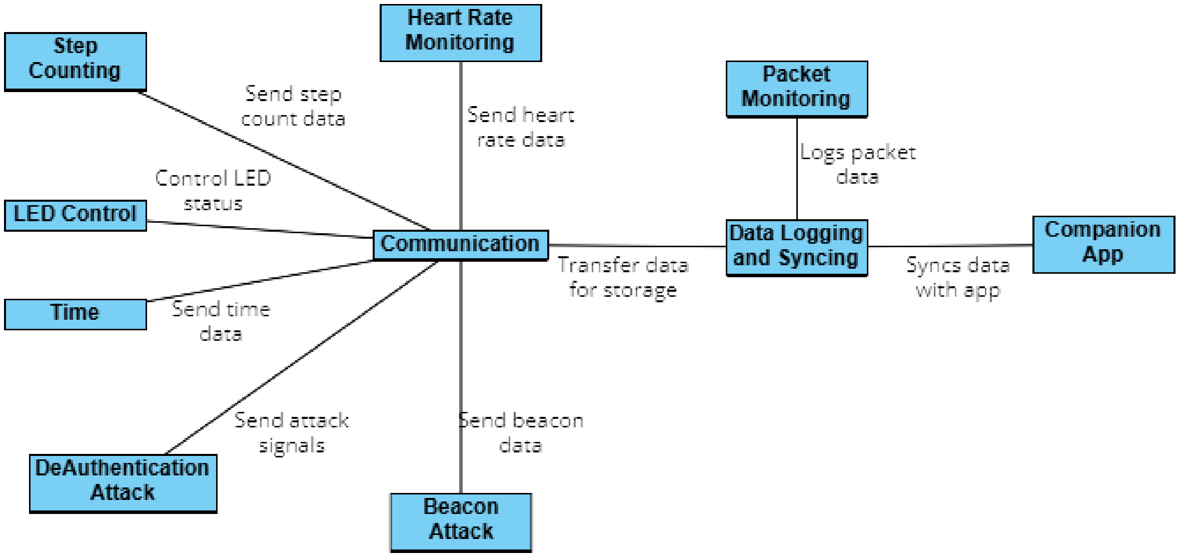


Figure 18: Logical View

Development View

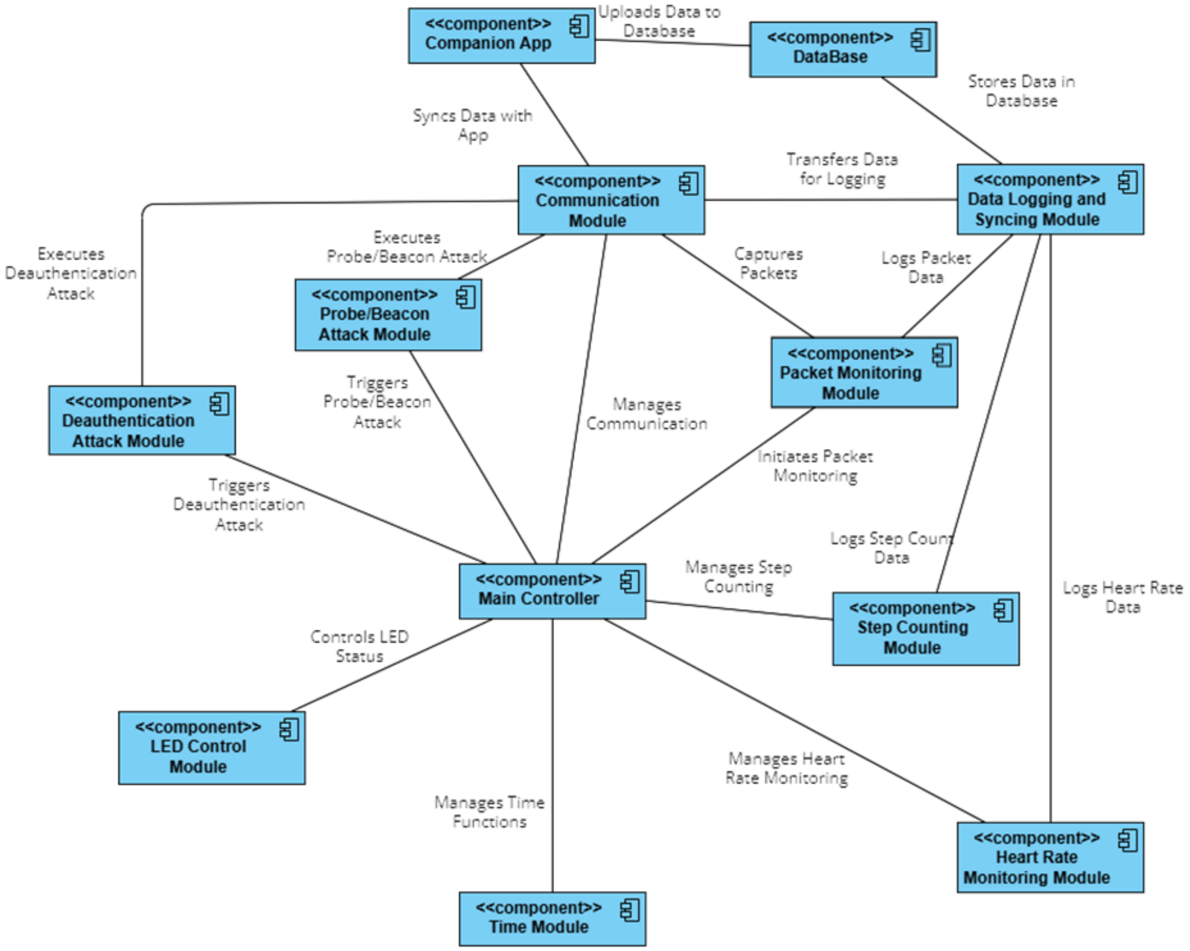


Figure 19: Development View

Process View

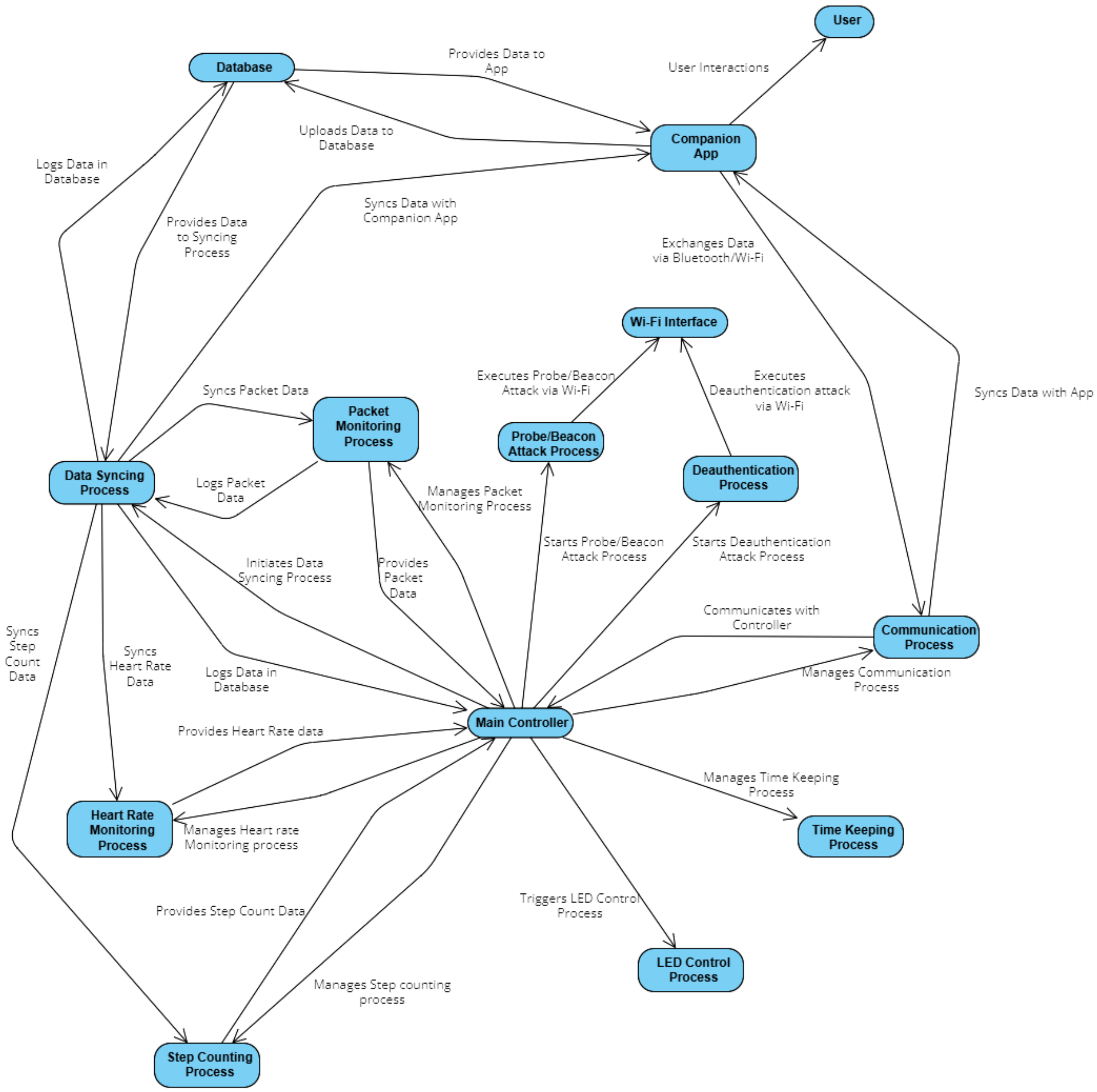

User
Database
Provides Data to App
User Interactions
Uploads Data to Database
Companion App
Logs Data in Database
Syncs Data with Companion App
Provides Data to Syncing Process
Exchanges Data via Bluetooth/Wi-Fi
Wi-Fi Interface
Executes Probe/Beacon Attack via Wi-Fi
Executes Deauthentication attack via Wi-Fi
Syncs Data with App
Syncs Packet Data
Packet Monitoring Process
Probe/Beacon Attack Process
Deauthentication Process
Data Syncing Process
Logs Packet Data
Manages Packet Monitoring Process
Starts Probe/Beacon Attack Process
Starts Deauthentication Attack Process
Initiates Data Syncing Process
Provides Packet Data
Communicates with Controller
Communication Process
Syncs Step Count Data
Syncs Heart Rate Data
Logs Data in Database
Manages Communication Process
Main Controller
Provides Heart Rate data
Manages Time Keeping Process
Heart Rate Monitoring Process
Manages Heart rate Monitoring process
Time Keeping Process
Triggers LED Control Process
Provides Step Count Data
LED Control Process
Manages Step counting process
Step Counting Process

Figure 20: Process View

Physical View

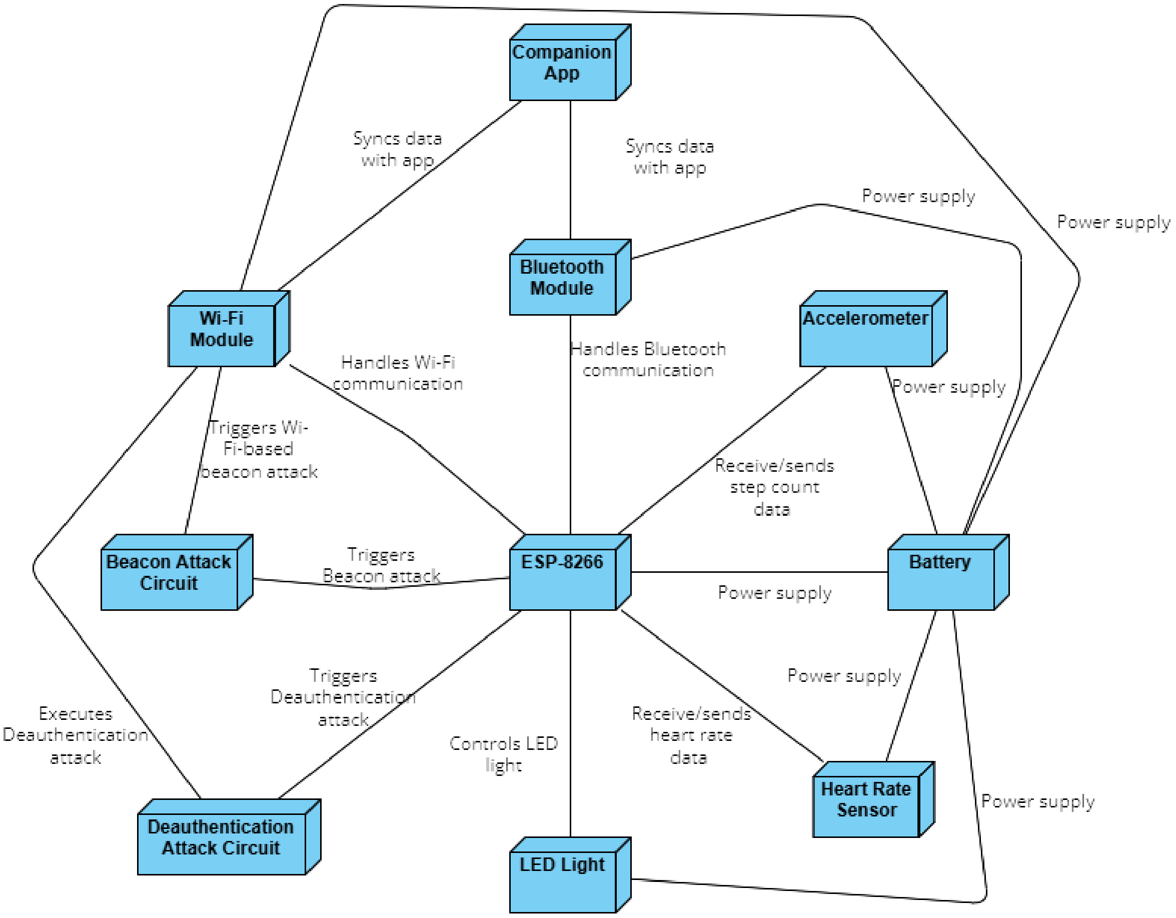


Figure 21: Physical View

User Interface Design

- User Interface Designs
    - App Wireframes

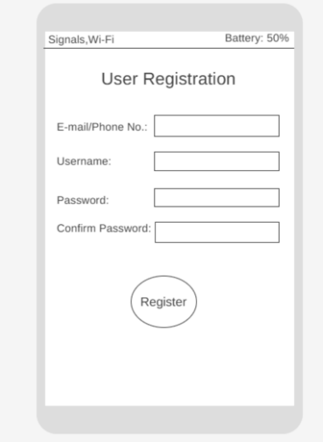
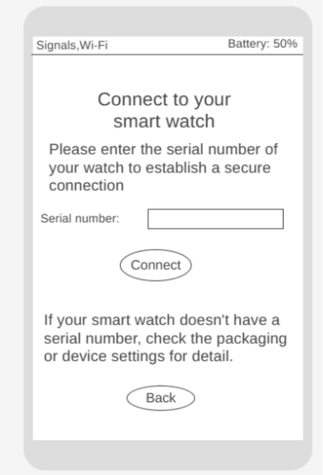


Figure 22: User Registration



Please enter the serial number of your watch to establish a secure connection

If your smart watch doesn't have a serial number, check the packaging or device settings for detail.

Figure 23: Smartwatch Connection

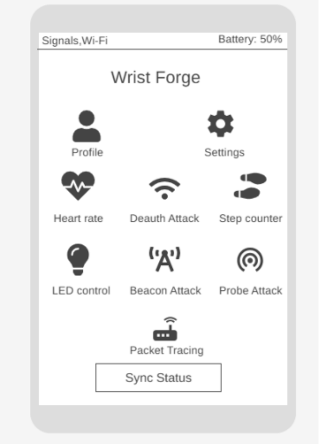
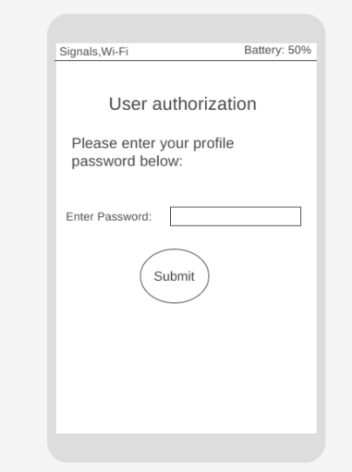


Figure 24:Home Screen

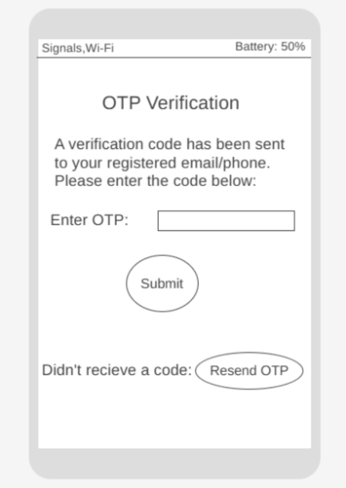


User authorization

Please enter your profile password below:

Enter Password:

Figure 25: User Authorization



OTP Verification

A verification code has been sent to your registered email/phone. Please enter the code below:

Enter OTP:

Didn't recieve a code:

Figure 26: OTP

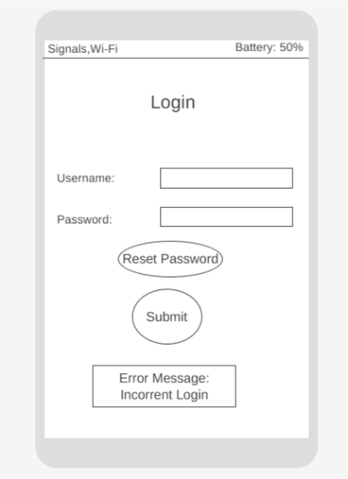


Figure 27: Login

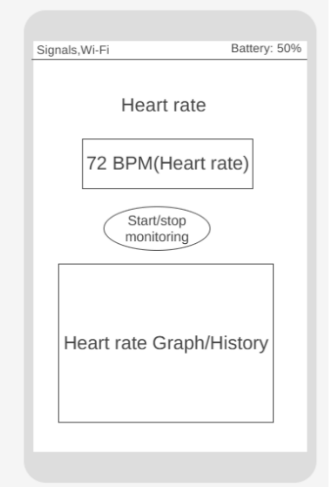


Figure 28: Heart Rate

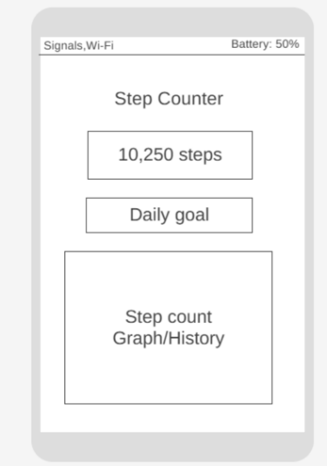


Figure 29: Step Count

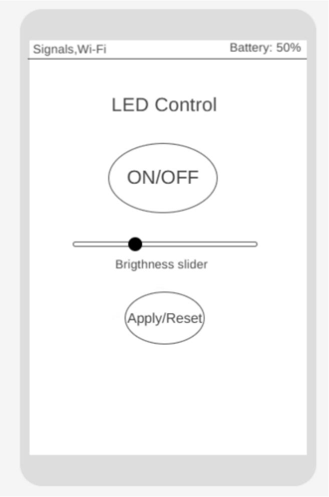


Figure 30: LED Control

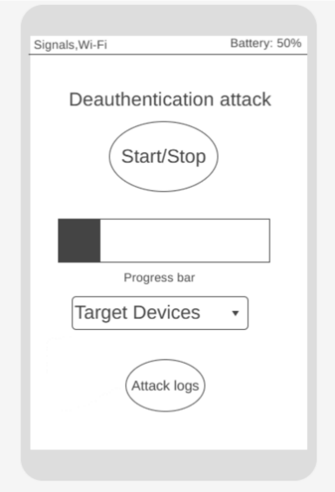


Figure 31: De-Authentication Attack

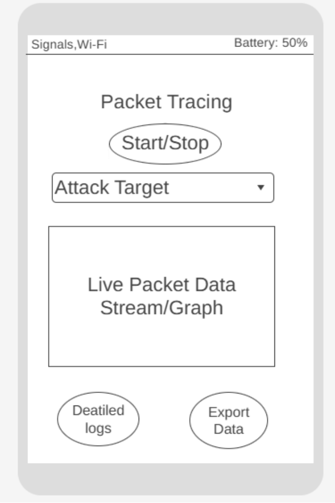


Figure 32: Packet Tracing

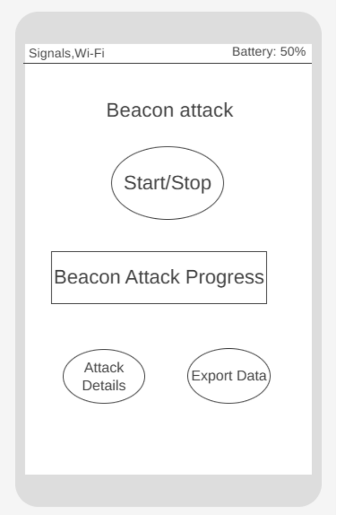


Figure 33: Beacon Attack

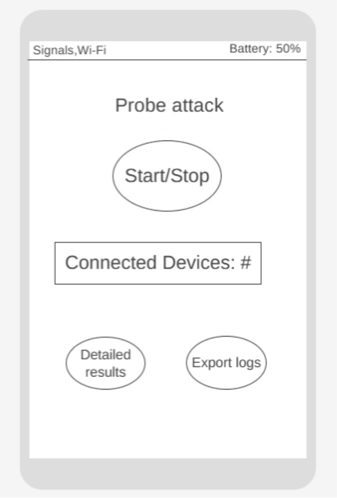
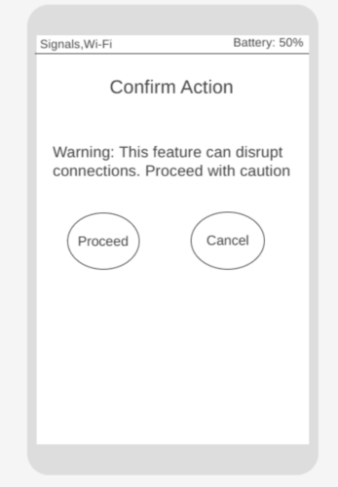


Figure 34: Probe Attack



Warning: This feature can disrupt connections. Proceed with caution

Figure 35: Attack Confirmation

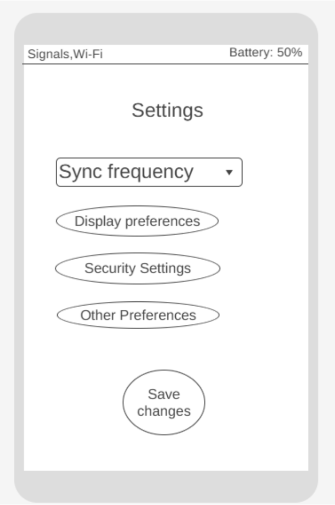


Figure 36: Settings

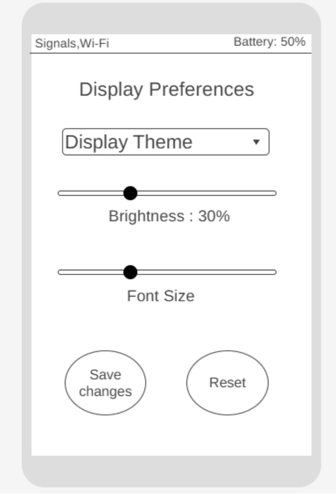


Figure 37: Display Preferences

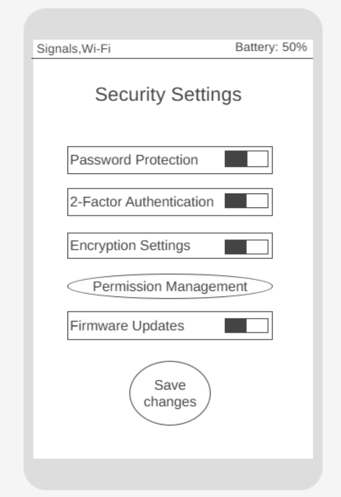


Figure 38: Security Settings

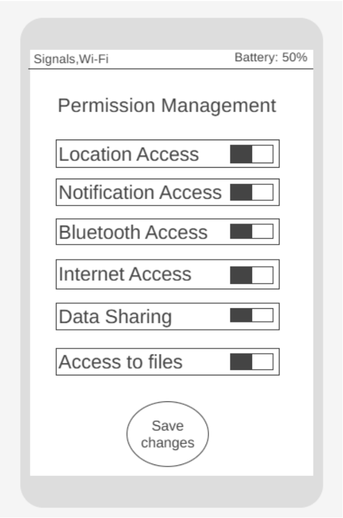


Figure 39: Permission Management

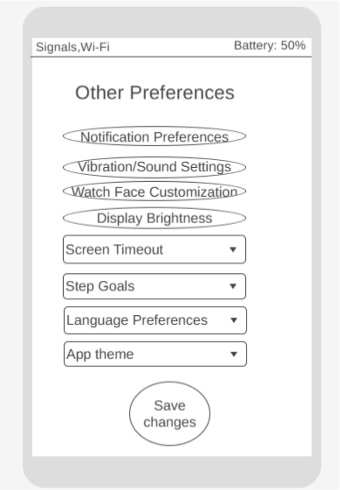


Figure 40: Other Preferences

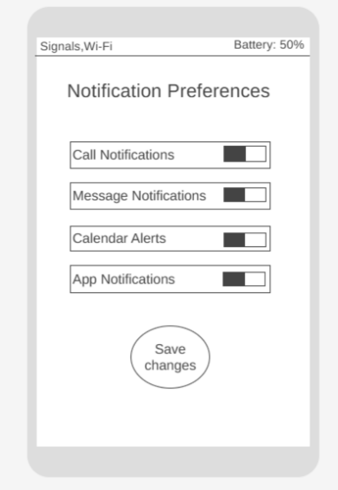


Figure 41: Notification Preferences

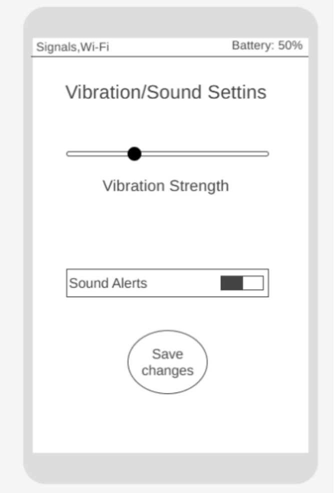


Figure 42: Vibration Preferences

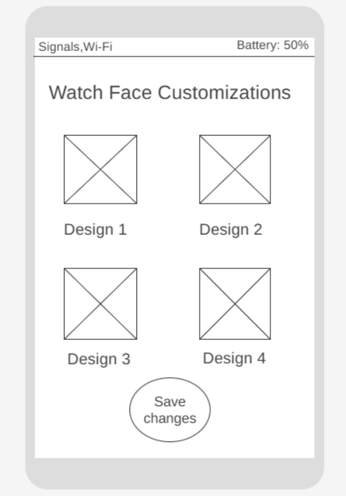


Figure 43: Watch Customization

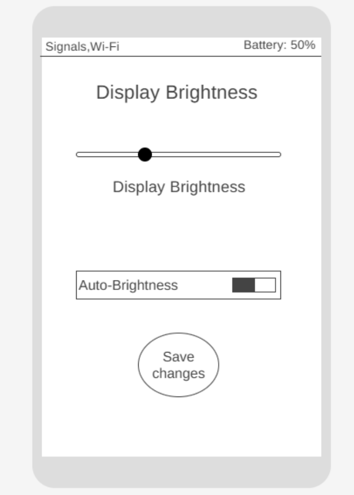


Figure 44: Display Brightness

○ Watch Wireframes

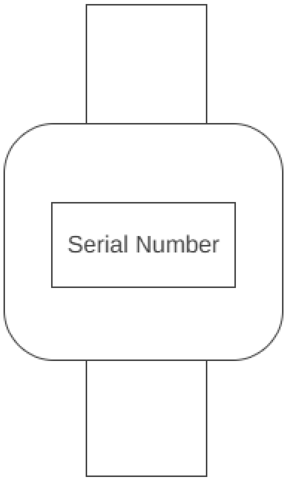


Figure 45: Serial number to connect to app

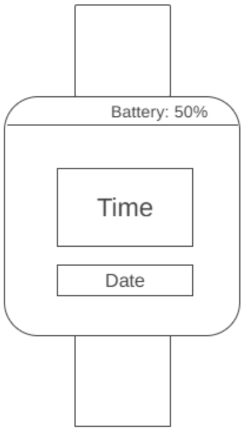


Figure 46: Home Screen

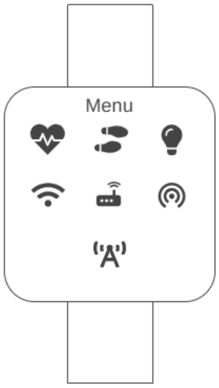
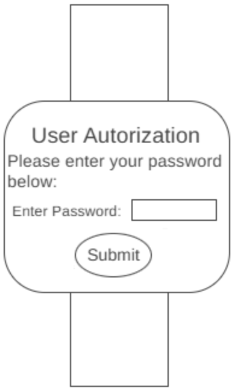
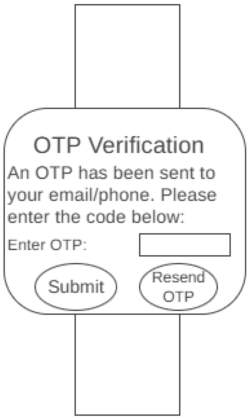


Figure 47: Menu screen



Please enter your password below:

Figure 48: User Verification



An OTP has been sent to your email/phone. Please enter the code below:

Figure 49: OTP

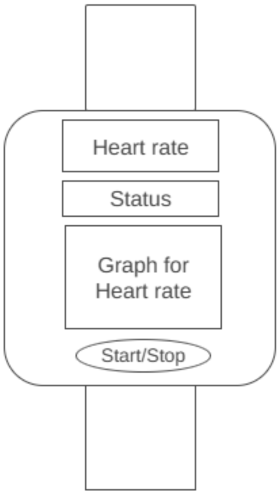


Figure 50: Heart Rate

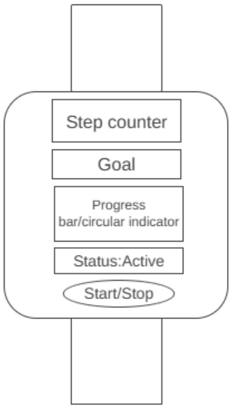


Figure 51: Step Count

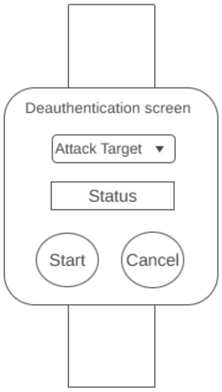


Figure 52: De-Authentication Attack

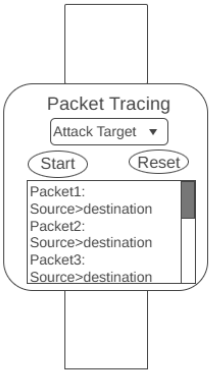


Figure 53: Packet Tracing

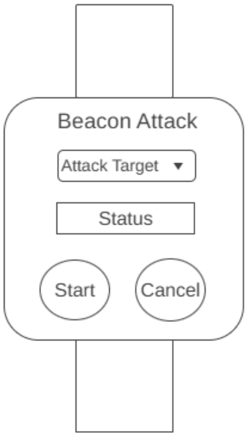


Figure 54: Beacon Attack

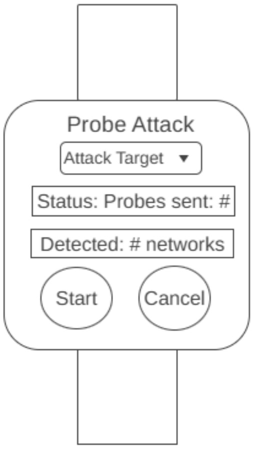
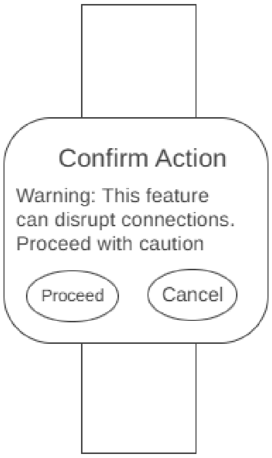


Figure 55: Probe Attack



Warning: This feature can disrupt connections. Proceed with caution

Figure 56: Attack Confirmation