Development of Mobile Application for wireless SpO_2 Monitoring and Alert System

By

Shreenandan Sahu

Under the guidance of

Jithin Krishnan

September 2025

Biomedical Technology Wing

Sree Chitra Tirunal Institute for Medical Sciences and Technology, Trivandrum

Contents

Abstract 3

Introduction 4

  Clinical Background 4

  Problem Statement 4

  Objectives of the Project 5

Materials and Method 5

  System Components 5

  App Development Workflow 6

  System Architecture 7

Results 8

Discussion 11

References 12

Table of Figures

Figure 1. Basic architecture of the app and its components. 7

Figure 2. Anatomy of the Application and different screens. 8

Figure 3.Block level description of the landing, function options and archive section of the App along with their respective UI Developed using MIT app inventor. 9

Figure 4. Block level description of the Live Plotting and Recording section of the App Developed using MIT app inventor. 10

Figure 5. Different components of the CHITRA PMS app in action demonstrating the complete operational flow from device connection to live monitoring, data recording, & result sharing. 11

Abstract

Continuous monitoring of peripheral oxygen saturation (SpO_2) is essential for early detection of hypoxemia and timely clinical intervention. Conventional bedside monitoring systems provide accurate data but require continuous human supervision, which is not always feasible in busy hospital environments. This limitation increases the likelihood of unnoticed desaturation episodes, potentially compromising patient safety. To address this unmet clinical need, we developed a mobile application that interfaces wirelessly with an existing SpO_2 sensor module via Bluetooth. The application, implemented using MIT App Inventor, integrates real-time signal acquisition, normalization, and graphical display of SpO_2 trends. It incorporates customizable visual and auditory alarms, including screen color changes, sound cues, and vibration alerts, when oxygen saturation values fall below a user-defined threshold. Furthermore, the application supports data recording, secure local storage, and retrieval for subsequent review, thereby augmenting clinical documentation. Preliminary testing indicates that this solution improves efficiency by reducing the dependence on constant bedside observation, while simultaneously enhancing patient safety through proactive notification of critical events. The approach demonstrates the potential of mobile health technologies to provide cost-effective, scalable, and accessible monitoring solutions in diverse healthcare settings.

Keyword: Continuous Monitoring, Hypoxia, Oxygen Saturation, MIT App inventor, data logging

Introduction

Clinical Background

Monitoring of peripheral oxygen saturation (SpO_2) is one of the most essential and non-invasive methods for assessing a patient's respiratory and cardiovascular status. In ICUs, operating theatres, and high-dependency units, SpO_2 levels provide critical early warning of hypoxemia, enabling prompt interventions such as oxygen therapy, ventilation adjustment, or escalation to advanced care. Even in general wards and during post-operative recovery, unnoticed desaturation can have serious consequences, including organ damage, prolonged hospitalization, or even mortality.

In well-resourced healthcare systems, patient safety is maintained through a high nurse-to-patient ratio, ensuring continuous vigilance at the bedside. However, in many Indian hospitals, especially in government or high-volume centres, a single nurse may be responsible for 4–6 patients in the ICU and even more in general wards. This disproportionate ratio makes it practically impossible to provide uninterrupted observation for each patient. Although bedside pulse oximeters are widely available, they rely on the presence of staff to notice alarms, interpret data, and respond promptly. In situations of staff shortage, fatigue, or multitasking, critical episodes of hypoxemia may remain unnoticed for several minutes, leading to delayed intervention and avoidable complications.

Problem Statement

The core issue lies not in the absence of monitoring equipment but in the gap between continuous measurement and continuous supervision. Existing pulse oximeter modules measure SpO_2 reliably, but the data is confined to the bedside device. Nurses and clinicians cannot always remain near each patient, nor can they visually monitor multiple displays simultaneously. As a result, an alarming drop in oxygen saturation may occur without immediate detection, especially during nighttime shifts or in high-patient-load wards.

Moreover, conventional devices often lack intelligent alerting systems that extend beyond basic auditory alarms. In noisy clinical environments or when staff are occupied, such alarms can easily be overlooked. There is, therefore, a clear demand for a portable, automated, and scalable alert mechanism that ensures SpO_2 data is not only recorded but also actively communicated to caregivers in real time. A mobile-based platform can close this gap by wirelessly receiving SpO_2 values from the existing sensor module, analysing them continuously, and issuing unmistakable alerts via sound, vibration, and visual cues. This solution does not replace current infrastructure but instead complements it, enhancing patient safety without imposing additional hardware costs.

Objectives of the Project

The overarching objective is to improve patient safety by transforming passive bedside monitoring into active, real-time clinical alerting. Specifically, the project aims to:

- **Wirelessly acquire SpO_2 data** from an existing sensor module using Bluetooth.
- Provide **real-time display of SpO_2 values and graphical trends** on a mobile interface accessible to caregivers.
- Deliver **immediate alerts** in multiple modes (**auditory tones, vibration, and color-coded visual cues**) when SpO_2 falls below set thresholds.
- Enable basic **data logging and secure storage** to support review, clinical validation, and research applications.

By addressing the gap between data availability and actionable alerts, this project directly responds to the clinical demand for safer, more efficient, and resource-sensitive monitoring systems in Indian hospitals.

Materials and Method

System Components

The system was designed using a combination of hardware and software components, with the ESP32 microcontroller serving as the communication bridge and the MAX30102 optical sensor as the data source.

- **Data Source (Hardware):** The MAX30102 pulse oximeter sensor was used to capture raw photoplethysmography (PPG) signals. This sensor integrates red and infrared LEDs along with a photodetector, enabling measurement of blood oxygen saturation (SpO_2) and heart rate. The sensor was interfaced with an ESP32 microcontroller, which handled signal acquisition, preprocessing, and wireless transmission.

    - Wireless Transmission: The ESP32, with its built-in Bluetooth Low Energy (BLE) capability, transmitted SpO_2 values to the mobile application in real time. For preliminary testing, simulated data streams were also generated on the ESP32 to validate app performance under controlled conditions.
- **Mobile Application:** An Android application was developed using MIT App Inventor. The app provided a graphical interface for monitoring SpO_2, visual trends, and alarm notifications. MIT App Inventor was chosen for rapid prototyping, block-based programming, and ease of integration with Bluetooth communication.
- **Alert System:** The app incorporated a multi-modal alarm mechanism to ensure reliable notification. Alerts included:
    - Visual cues: Color-coded backgrounds (green for normal, red for hypoxemia).
    - Auditory signals: Beeping tones for critical alerts.
    - Vibration feedback: Triggered on threshold violation for handheld device notification.

App Development Workflow

The app was implemented in modular stages to ensure reliability and usability:

- **Frontend (User Interface):** The app has mainly 2 screen one is to display the live PPG signal and the other one is to display the recorded one. In the live PPG section, we have features to show the plot and buttons to save the
- **Backend (Data Processing & Storage):** The app's logic handled Bluetooth data reception, threshold analysis, and logging. A simple data buffer ensured smooth plotting and avoided overflow during continuous monitoring.
- **Alert Logic:** A threshold of $SpO_2 < 92\%$ was chosen as the clinical cutoff for hypoxemia. When this condition was met, the app simultaneously:
    - Turned the display background **red**,
    - Activated **vibration**, and
    - Triggered an **auditory beep alarm**.
      Under normal conditions, the background remained **green,**
    - and no alarms were active.
- **Testing & Validation:** To validate system performance, the app was first tested with **simulated SpO_2 data** from the ESP32 to confirm Bluetooth connectivity, alert

responsiveness, and UI behaviour. In parallel, actual sensor values from the MAX30102 were compared with reference device readings to verify accuracy.

System Architecture

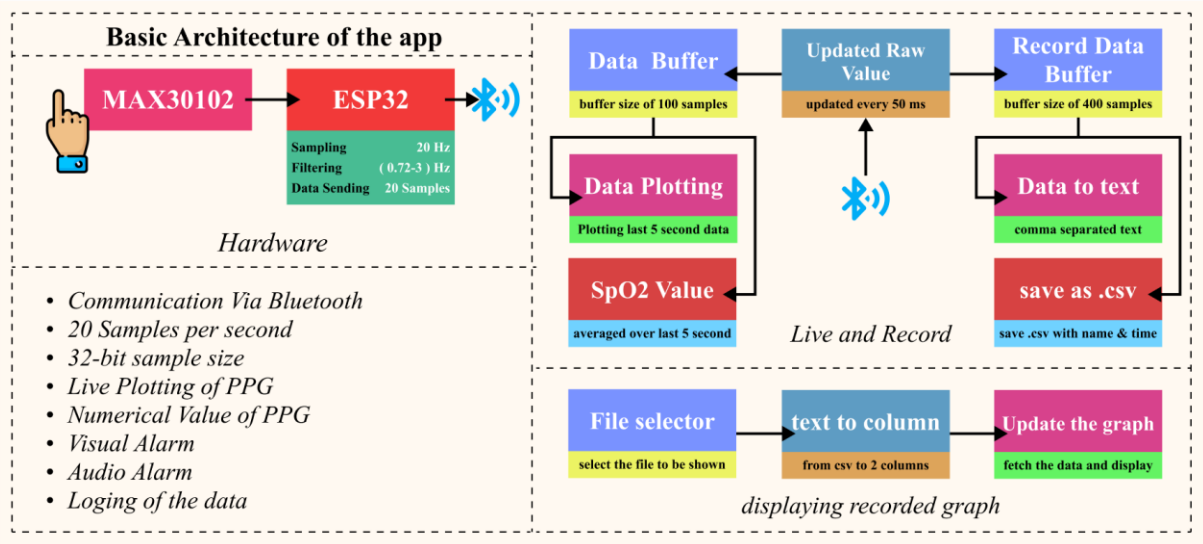


Figure 1. Basic architecture of the app and its components.

The system architecture consisted of four primary layers:

1. **Signal Acquisition:** The MAX30102 sensor collected PPG signals, which were processed by the ESP32 to compute SpO_2 values.
2. **Communication Layer:** The ESP32 sent data packets wirelessly via Bluetooth to the mobile application.
3. **Application Backend:** The app continuously received and decoded incoming data, checked values against defined thresholds, and stored basic logs.
4. **User Interface & Alerts:** The app displayed real-time SpO_2 values numerically and graphically. If SpO_2 dropped below 92% (or another preset threshold), the app triggered auditory, visual, and vibration-based alerts.

Results

User Interface

The UI was developed in MIT APP Inventor’s designer section where we mainly used image, text level, buttons, list pickers, 2D graph plot and text box for the visual components of the app. We have used clock, Bluetooth client, files, sound, and notifier as the hidden components. Once the UI is ready, we then used the block section of the MIT app Inventor to give the logic to the UI and logically connect the different sections and functions. The following figure describes the entire mobile app and its anatomy.

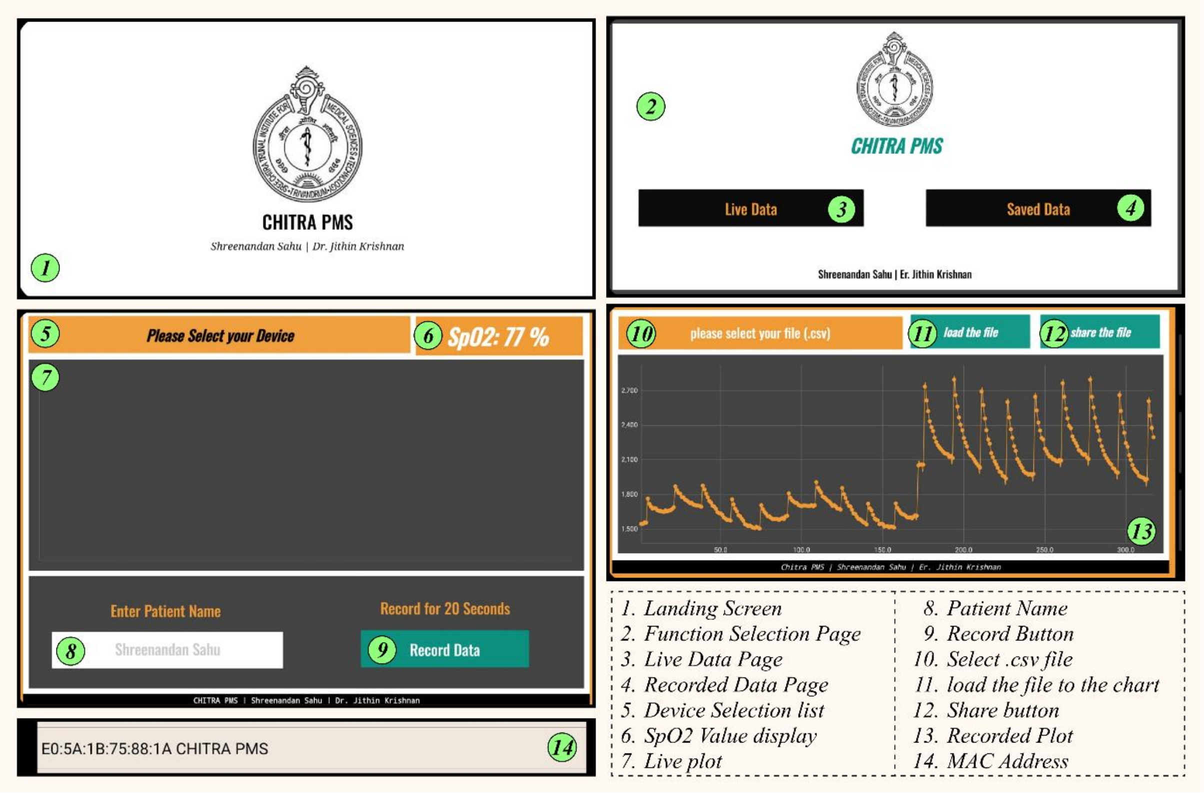
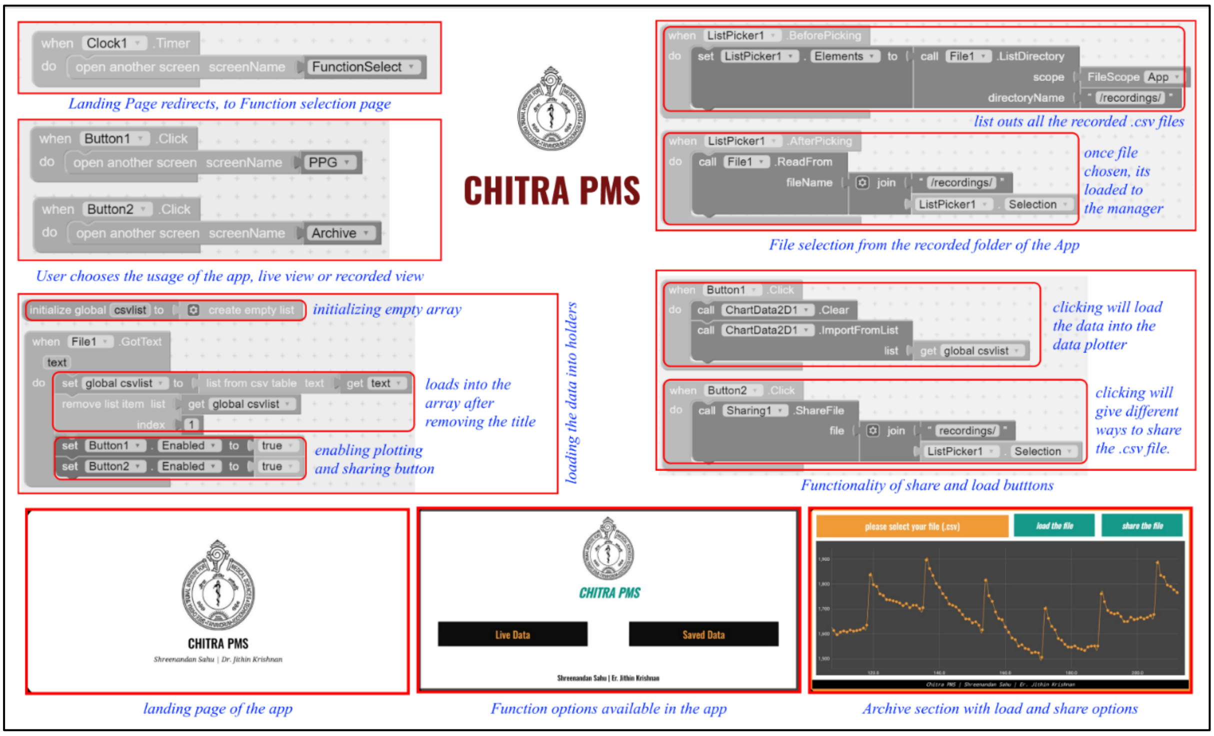


Figure 2. Anatomy of the Application and different screens.

Backend Logic

The application begins with a landing page that automatically redirects the user to the function selection interface, where the choice between live monitoring (PPG/SpO_2 acquisition) or reviewing previously recorded data is provided. When the live mode is selected, the system first initializes global variables for data storage and buffer handling, then lists all paired Bluetooth devices of the phone. Once a device is selected, the application attempts connection and updates the interface to indicate the status. Upon successful connection, incoming data packets from the Bluetooth client are received, stored in a buffer, and plotted in real time. A first-in-first-out (FIFO) mechanism is employed to clear older entries and maintain a windowed plot. Periodically, the application computes the average, performs normalization, and updates the SpO_2 user interface. Alarm levels are continuously checked, and multimodal alerts are generated through color-coded visual cues, auditory beeps, and vibration to ensure reliable detection of critical conditions.



Landing Page redirects, to Function selection page

User chooses the usage of the app, live view or recorded view

initializing empty array

loads into the array after removing the title

enabling plotting and sharing button

loading the data into holders

list outs all the recorded .csv files

once file chosen, its loaded to the manager

File selection from the recorded folder of the App

clicking will load the data into the data plotter

clicking will give different ways to share the .csv file.

Functionality of share and load butttons

landing page of the app

Function options available in the app

Archive section with load and share options

Figure 3.Block level description of the landing, function options and archive section of the App along with their respective UI Developed using MIT app inventor.

For recording, a dedicated trigger activates timers to define total duration and sampling intervals. The application sequentially indexes and appends acquired data into structured storage, while a countdown mechanism update elapsed time in the user interface. Data files are automatically named either through user input or timestamp generation, and after completion, the recorded dataset is saved into the app’s private directory, accessible via a connected PC. On successful save, timers and variables are reset, buffers cleared, and confirmation messages displayed to the user.

In archive mode, the application initializes an empty array to hold stored values and provides a file manager for selecting previously recorded .csv files from the designated directory. Once a file is chosen, its contents are loaded into memory (with header rows removed), enabling both plotting and sharing functionalities. Users may visualize stored datasets on a plotting interface or share the corresponding .csv file through multiple system-defined sharing options. This dual-mode structure—live acquisition with recording and retrospective review with sharing—ensures the application provides both real-time monitoring and reliable long-term data management.

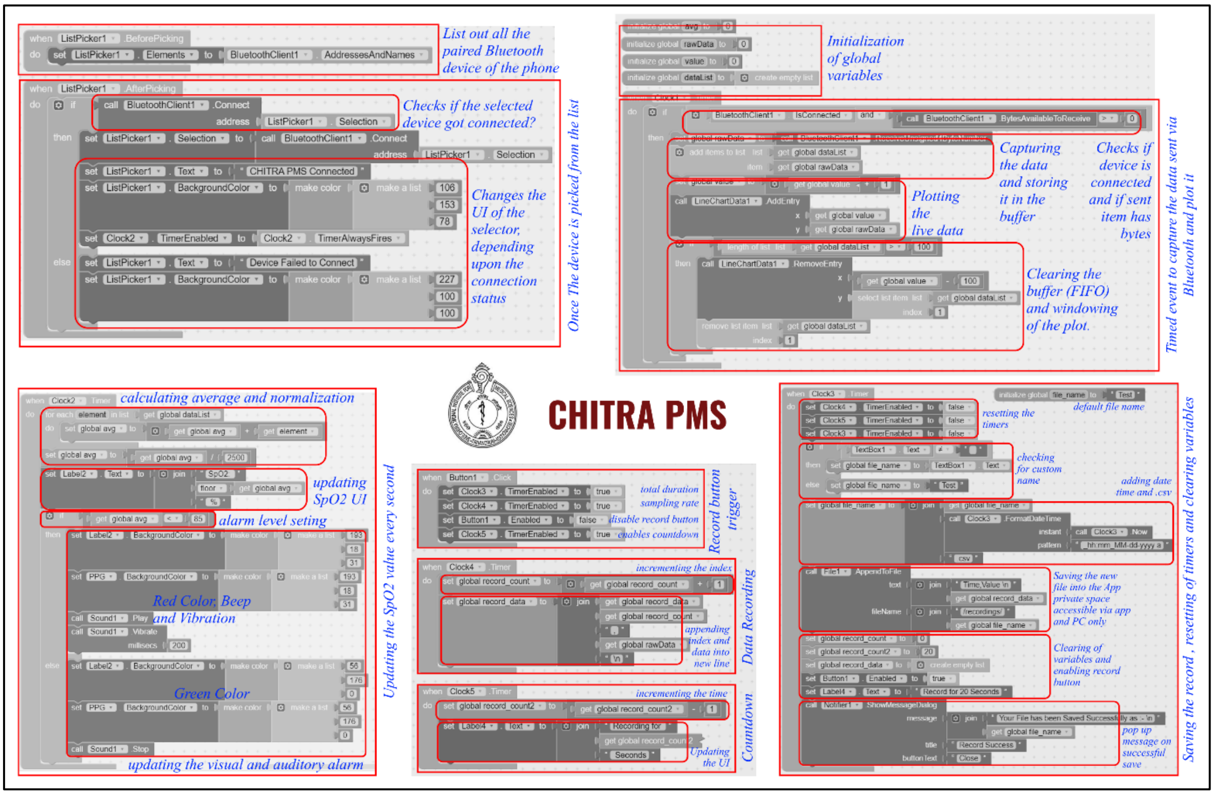
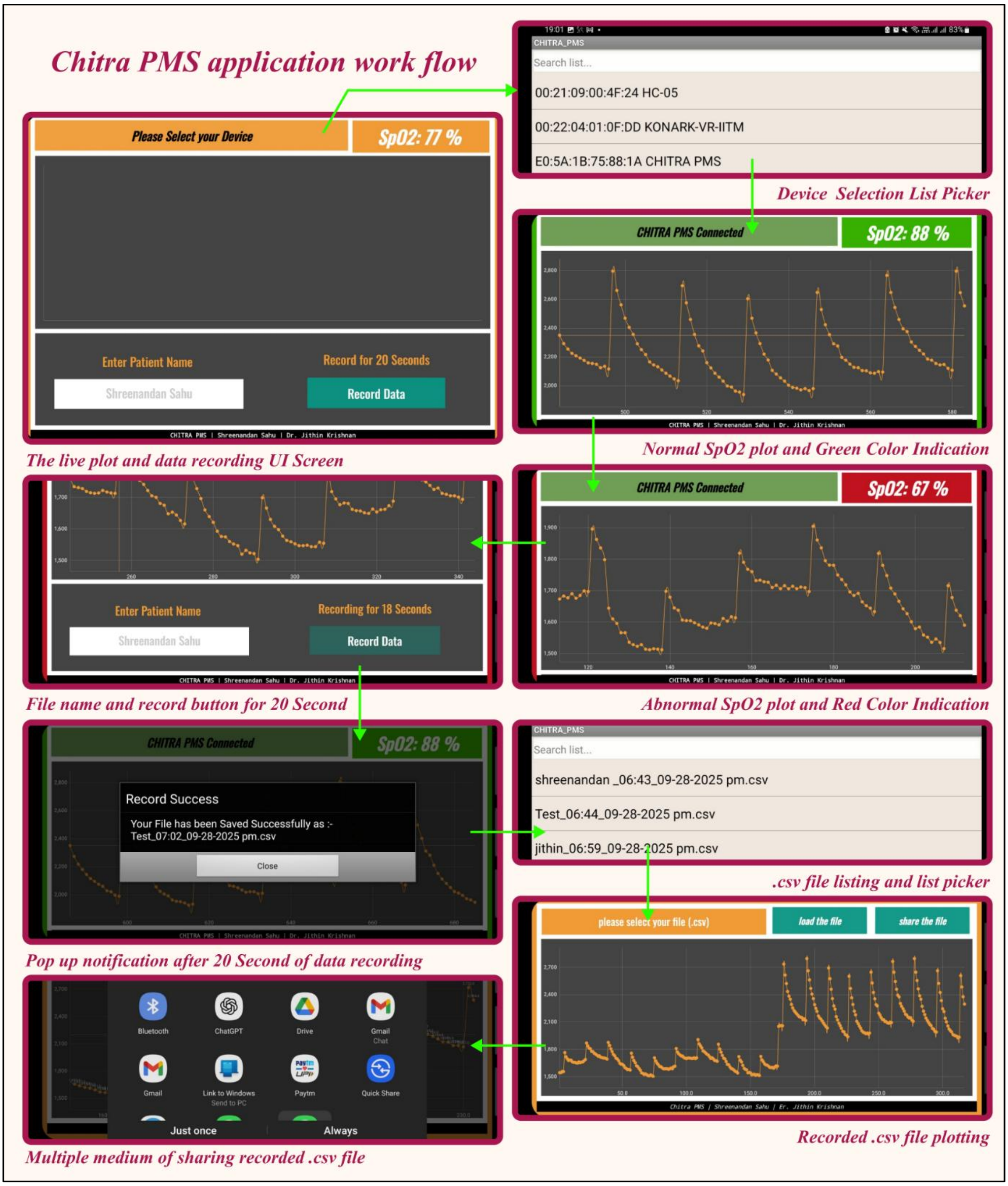


Figure 4. Block level description of the Live Plotting and Recording section of the App Developed using MIT app inventor.

Final Output of APP

The screenshots shown in the following Figure illustrate the functional output of the developed CHITRA PMS mobile application. The app successfully displays live SpO_2 readings along with the real-time PPG waveform, visual alarms through color-coded indicators, and recording progress with time updates. Upon completion of data recording, a confirmation message indicates successful file saving with automatically generated filenames. The archive section allows users to view the list of recorded CSV files, reload them for waveform visualization, and share them through various platforms. These result screens validate the complete working of live monitoring, recording, and data retrieval features within the app.



00:21:09:00:4F:24 HC-05
00:22:04:01:0F:DD KONARK-VR-IITM
E0:5A:1B:75:88:1A CHITRA PMS

Your File has been Saved Successfully as :-
Test_07:02_09-28-2025 pm.csv

shreenandan _06:43_09-28-2025 pm.csv
Test_06:44_09-28-2025 pm.csv
jithin_06:59_09-28-2025 pm.csv

Figure 5. Different components of the CHITRA PMS app in action demonstrating the complete operational flow from device connection to live monitoring, data recording, & result sharing.

Discussion

The developed CHITRA PMS mobile application demonstrates reliable and efficient performance for real-time SpO_2 monitoring and data recording using Bluetooth communication. During testing, the system achieved a stable data acquisition rate of approximately 20 samples per second, with each sample occupying 4 bytes. This rate ensured smooth and continuous waveform rendering without noticeable latency or data loss, providing a visually stable PPG signal. The live plotting interface displayed a dynamic, windowed view of the most recent 5 seconds of data, enabling focused real-time observation while maintaining computational efficiency and avoiding unnecessary memory usage.

Each recording session was configured to last for 20 seconds, after which the acquired data was automatically processed and stored. The application's file management system utilized the app's scoped storage for secure data handling, preventing access by other applications and thereby ensuring data privacy. When no filename was specified by the user, an automatic naming convention was applied using a timestamp that included the date and time of acquisition, facilitating easy record identification. The stored files could be later accessed within the app's archive section, where they could be visualized as complete plots with zoom functionality for detailed analysis.

The built-in sharing option enabled direct transmission of recorded data to healthcare providers via multiple supported platforms such as email or cloud-based services, thus bridging the gap between patient-side monitoring and clinical supervision. The application was thoroughly validated using simulated SpO_2 and PPG data transmitted from the ESP32 microcontroller, confirming consistent Bluetooth connectivity, correct alert functionality, and accurate data plotting. The system performed without frame drops or communication interruptions, validating its reliability for extended use. Overall, the app successfully meets its design objectives of providing a lightweight, portable, and secure real-time SpO_2 monitoring solution that can operate effectively in resource-limited healthcare settings.

References