# Facial Recognition for safer homes

**R1: Raspberry Pi Setup**

**R1.1: Gather components (Raspberry Pi, case, power supply, SD card, monitor, keyboard, mouse).**

**R1.2: Assemble the Raspberry Pi hardware (insert SD card, attach peripherals).**

**R1.3: Download Raspberry Pi OS from the official site.**

**R1.4: Flash the OS onto the SD card.**

**R1.5: Boot the Raspberry Pi and complete the initial setup.**

**R1.6: Verify network functionality via Wi-Fi/Ethernet.**

**R2: Camera Integration**

**R2.1: Attach the camera to the Raspberry Pi using the CSI interface.**

**R2.2: Install camera software libraries (OpenCV).**

**R2.3: Write a simple script to test image capture.**

**R2.4: Capture video stream and verify camera alignment.**

**R3: Facial Recognition Software**

**R3.1: Research available libraries OpenCV.**

**R3.2: Install necessary dependencies and test sample scripts.**

**R3.3: Collect facial images of users and organize datasets.**

**R3.4: Preprocess the images (resize, grayscale, normalize).**

**R3.5: Train the facial recognition model using collected data.**

**R3.6: Test facial recognition accuracy on live stream.**

**R3.7: Fine-tune model parameters for better accuracy.**

**R4: Smart Lock Integration**

**R4.1: Research and purchase a compatible smart lock (GPIO, Wi-Fi, or Bluetooth-enabled).**

**R4.2: Connect the smart lock to the Raspberry Pi and ensure power compatibility.**

**R4.3: Write a script for basic lock control (lock/unlock).**

**R4.4: Test lock response to Raspberry Pi commands.**

**R5: System Integration**

**R5.1: Combine camera and lock scripts into a single program.**

**R5.2: Implement logic for facial recognition to trigger lock/unlock events.**

**R5.3: Test the integrated system under different scenarios (e.g., valid/invalid users).**

**R6: User Management (Faces that will be used will be from a public database)**

**R6.1: Write code for adding new users (capture and save facial data).**

**R6.2: Implement a function for deleting or updating user data.**

**R6.3: Test user management system and ensure data is securely stored.**

**R7: User Interface Development**

**R7.1: Design a simple web or mobile interface using frameworks like React.**

**R7.2: Include features like user management, system status, and access logs.**

**R7.3: Connect the interface to the Raspberry Pi for real-time control.**

**R7.4: Test interface usability with various devices.**

**R8: Documentation**

**R8.1: Document step-by-step assembly and installation instructions.**

**R8.2: Write a guide for system setup and usage.**

**R9: Feedback and Iteration**

**R9.1: Deploy the system in a real environment and gather user feedback.**

**R9.2: Address reported issues and improve usability.**

**R9.3: Update the system based on iterative testing results.**

**R10: Advanced Features**

**R10.1: Integrate a backup unlocking method (e.g., keypad, RFID).**

**R10.2: Add notifications for lock/unlock events and security breaches (Have sensors that detect if the door is open without approving a face).**

**R10.3: Add/Remove Faces from the database**

**\*\*CAN ACCIDENTALLY GRAB REAL FACE (think of a way to prevent this) \*\***

**R10.4: Speaker/Screen for aesthetics**

**R11: Finalization and Presentation**

**R11.1: Prepare a detailed project portfolio (photos, videos, reports).**

**R11.2: Create a final demo showcasing all features.**