**Software ideas for final project:**

Design, Build, Test and Deploy a Piece of Software for a Specific Purpose

**Description**

**The objective of this design, build, test and deploy project is to produce a piece of software for a specific purpose using a language of your choosing. The software development will be based on systematic and critical review of existing and similar (state of the art) software solutions as well as consultation with stakeholders including end users, identifying main issues, prospectss for improvement, and specifying the unique benefits of the newly proposed software. Using practices and philosophies of an agile approach to software development, your software will be designed and built to showcase new/improved features and functionalities following stakeholders needs and requirements. Experimental evaluation and testing of the software will be conducted, and user feedback will be collected for functionality refinement. A comprehensive documentation of the development processes (planning, research, flowchart, technologies involved, components built and how they were built, testing, deployment, etc.) for the software will be produced in fine grained details.**

**Requirements**

**Systematic and critical review of existing and similar (state of the art) software solutions  
System Models, Architectures, Prototypes, Algorithms, Software Tools  
Secure Software Design Principles**

**Smart security system with raspberry pi:**

* Facial recognition
* Motion detection
* Sends notification - secure networking
* Indoor wifi positioning ??

**1. Hardware Setup**:

* Set up Raspberry Pi boards with Raspberry Pi cameras in strategic locations around the home to capture video footage.

**2. Motion Detection:**

* Use motion detection algorithms (e.g., OpenCV) to detect any movement within the camera's field of view.

**3. Facial Recognition:**

* Implement facial recognition using libraries like OpenCV to recognize registered house members' faces.

**4. Database Integration:**

* Create a database to store facial recognition data and information about house members.

**5. Notification System:**

* Develop a notification system that sends alerts to users' smartphones or email addresses when motion is detected or a recognized face enters the home.

**6. User Interface:**

* Create a user-friendly interface for users to register their faces, manage notifications, and view security footage remotely.

**7. Testing and Iteration:**

* Test the system thoroughly to ensure accurate motion detection and facial recognition. Iterate on the system based on feedback and performance.

**8. Deployment:**

* Install the cameras and set up the system in the home, ensuring proper positioning and connectivity.

**9. Maintenance and Updates:**

* Regularly maintain and update the system to address any bugs, security vulnerabilities, or improvements.

This project requires proficiency in programming, hardware setup, and possibly networking, but it's a rewarding DIY home security solution.

Here are some addition

1**. Two-Way Communication:** Integrate microphones and speakers to allow users to communicate with visitors remotely.

2. **Intruder Deterrence**: Incorporate features like flashing lights or sounding alarms to deter intruders when unauthorized motion is detected.

3. **Remote Monitoring:** Enable users to access live video feeds and recorded footage from their smartphones or computers while away from home.

**4. Integration with Smart Home Devices**: Integrate with other smart home devices like smart locks or lights to automate actions based on detected motion or recognized faces.

**5. Multiple Camera Support:** Expand the system to support multiple cameras for comprehensive coverage of the entire property.

**6. Cloud Storage:** Offer the option to store video footage and facial recognition data securely in the cloud for easy access and backup.

**7. Activity Log:** Maintain a log of all detected events, including motion detection, face recognition, and user entries/exits, for historical reference.

**8. Customizable Alerts:** Allow users to customize notification preferences, such as specifying certain times of day when they want to receive alerts or adjusting sensitivity levels for motion detection.

**9. Integration with Emergency Services:** Implement a feature to automatically alert emergency services (e.g., police) in case of a security breach or emergency situation.

**10. Energy Efficiency:** Optimize the system to minimize power consumption, for example, by implementing sleep modes for inactive cameras or using low-power components.

Adding these features can further improve the functionality and effectiveness of your home security system, providing users with greater peace of mind and control over their home's security.

**Design Phase:**

**1. Requirements Gathering:**

* Define the functional requirements, such as motion detection, facial recognition, notification system, etc.

**2. System Architecture:**

* Design the overall system architecture, including hardware components (Raspberry Pi, cameras), software components (motion detection algorithm, facial recognition system), and data flow.

**3. User Interface Design:**

* Design the user interface for registering faces, managing notifications, and accessing security footage remotely.

**4.Database Schema Design:**

* Design the database schema for storing facial recognition data, user profiles, and activity logs.

**Methodology: SCRUM (agile)**

**Build Phase:**

**1. Hardware Setup:**

* Purchase and assemble the required hardware components, including Raspberry Pi boards, cameras, microphones, and speakers.

**2. Software Development:**

* Develop the software components, including motion detection algorithms, facial recognition system, notification system, and user interface.

**3. Database Implementation:**

* Set up the database system and implement the database schema designed in the design phase.

**4. Integration:**

* Integrate the hardware components with the software system, ensuring proper communication and functionality.

**Test Phase:**

**1. Unit Testing:**

* Test each individual component (motion detection, facial recognition, notification system) in isolation to ensure they work as expected.

**2. Integration Testing:**

* Test the integrated system as a whole to ensure all components work together seamlessly.

**3. User Acceptance Testing (UAT):**

* Involve end-users to test the system's usability, functionality, and performance against the defined requirements.

**4. Bug Fixing and Optimization:**

* Identify and fix any bugs or performance issues discovered during testing. Optimize the system for better performance and efficiency.

**Deploy Phase:**

**1. Installation:**

* Install the cameras and set up the system in the home, ensuring proper positioning, connectivity, and power supply.

**2. Configuration:**

* Configure the system settings, including user profiles, notification preferences, and security settings.

**3. Training:**

* Train users on how to use the system, including registering their faces, managing notifications, and accessing security footage remotely.

**4. Documentation:**

* Provide comprehensive documentation, including user manuals, troubleshooting guides, and maintenance procedures, to assist users in operating and maintaining the system effectively.

**Social media app with tracking and leaderboards:**

* Fitness verison
* Developer version

**Developer social media app:**

* profile -> people enter how many projects or hours they have worked on during the day
* Tracks the progress
* Compares to others
* Switch between leaderboards
* Some sort of social media element to it
* view  other profiles and follow people
* Have a feed
* Have a weekly challenge question ?

**1.  Design phase**

**User interface design:**

* sketch wireframes
* Focus on intuitive navigation and visually appealing design elements.
* Making it accessible

**Database schema design:**

* Define the database schema to store user profiles, posts, coding challenge submissions, leaderboard data, etc.
* Database technologies
* Establish relationships between different entities

**System architecture design:**

* Determine the overall system architecture, including frontend and backend components
* Decide on the technology stack: front end ( react ) backend ( nodejs), database (mongoDB)

**2. Development phase  (Build)**

**Methodology: SCRUM (agile)**

**Front End:**

* Choose the frontend technologies
* Implement UI components based on the wireframes

**Backend:**

* Build authentication mechanisms for user registration, user login, and session management
* Develop APIS to handle CRUD operations for user profiles, posts and coding challenges, etc
* Implement algorithms for calculating leaderboard rankings based on specific criteria.

**Database**

* Create the database tables based on schema.
* Set up indexes and relationships to optimize data retrieval and querying performance.

**3.Testing**

**Unit testing:**

* Write unit tests to verify the functionality of of the individual components and modules
* Juse for fontment
* Mocha for backend

**Integration testing:**

* Testing the interaction between frontend and backend components to ensure seamless communication
* User acceptance testing:
* Provide feedback on the apps features and functionality
* Address issues or bugs identified and make necessary adjustments

**4. Deployment**

**Deployment to staging environment:**

* Deploy the application to a staging environment for final testing in a production-like setting
* Perform additional rounds of testing to ensure stability and performance

**Deployment to production:**

* Deploy it to the production environment
* Monitor the app's performance and address any issues that arise during the initial rollout.

**5. Maintenance:**

**Ongoing support and maintenance:**

* Provide ongoing support to address user inquiries, bug reports and feature requests.

**Continuous improvement:**

* Collect user feedback and analytics data to identify areas for improvement
* Release regular updates and new feature to enhance the apps functionality and user experience

**Fitness social media app:**

* Basically the same, but instead of coding components
* Implement tools for users to log workouts,
* Track progress e.g. weightlifting stats, body measurements) , set goals
* Enables users to create profiles, follow others, like/comment on posts ans send messages.
* Integrate adding friends, group chats
* Feed: develop an algorithm to display relevant content , friends, popular posts and recommended workouts
* Gym based features like local gym checkin and gc for the specific gyms
* Leader board for each workout