## **A close-up of a logo Description automatically generatedComputer Systems Department**

## **Pervasive Computing Project**

The pervasive computing project aims to develop a solution for remote smart control. The purpose is to control embedded electronic circuits using a mobile application. The main objectives are **Seamless Connectivity, Remote Control, Real-time Monitoring.**



**System description**

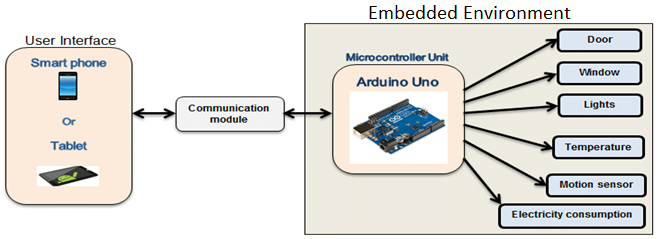
The remote smart control project consists of a digitally controlled interface between an embedded system comprising a microcontroller, wireless integrated module, environmental sensors. This system should make a connection over Wifi and send and receive data to a monitoring system on a mobile application. Also, it has a feature to control some parts of the system remotely from the mobile application. The data is stored in cloud (firebase) and cached by both the microcontroller and mobile application through SQL lite. You need to design and implement one of the following three embedded systems and integrate it to the mobile application:

1. **Smart home system** - Enables users to remotely control and manage various devices and systems within their homes using smartphone:
   1. Feature1: Temperature: Measure the room temperature through temperature sensor, uploads the value on the cloud (firebase database), then view it on the mobile application through an activity (e.g. Temperature Activity).
   2. Feature2: Password: Create a password for your home through the mobile application, store it on the cloud. It is used by the microcontroller’s connected keypad to open the home lock (door) if it is entered correctly.
   3. Feature 3: Light: A Boolean value is sent through the mobile application to the microcontroller to control and turn on/off a LED. The Boolean value is captured through a button on the mobile application’s interface and stored on the cloud.
   4. Feature 4: Fan: A Boolean value is sent through the mobile application to the microcontroller to control and turn on/off a fan motor. The Boolean value is captured through a button on the mobile application’s interface and stored on the cloud.
   5. Feature 5: Entry Attack: Detect a person passing through the home border (ultrasonic sensor connected to microcontroller), set a variable called “alert” by true on the cloud, on the Arduino side you implement value change listener on that variable if it’s true you send notification alert to android. The value of variable is changed to false only when password is entered correctly. A “We are Safe”, or “We are at Risk!” message is displayed on the mobile application’ interface based on the value of the variable.
   6. Feature 6 (Bonus): Message: A message is entered by the mobile application (e.g. Message Activity), stored on the cloud, and cashed by the microcontroller from the cloud to be displayed on LCD, for example “Welcome Mohamed!”.
2. **Garage system** - Connect the garage door opener to a mobile application allowing users to open, close, and monitor the status of their garage door from anywhere:
   1. Feature1: Control the garage door (open/close) through a DC motor. The order is taken from the mobile application.
   2. …..

You need to discuss and confirm with your T.A. on your suggestions for the other features before implementing the project.

1. **Agriculture watering system** -Integrate various sensors, actuators, and communication technologies to automate and optimize the irrigation process, improving water efficiency and crop yield:
   1. Feature1: Control water pump (on/off), through a DC motor pump. The order is taken from the mobile application.
   2. …..

You need to discuss and confirm with your T.A. on your suggestions for the other features before implementing the project.

****

**Software Requirements**

Use android studio IDE for developing the mobile application. **All** the following features should be implemented:

1. **Registration Activity**: name, username, password, confirm password, profile, picture, email, birthdate (calendar) [stored on firebase, then cached SQLite].
2. **Login Activity**:

1- Login using normal method or firebase authentication.(Implement both).

2- Implement remember me, using checkbox.

3- Implement forget password.

1. **Main Activity** (Home Activity): which contains a listview of all actions.
2. **Listview** or Recycler view: which contains an item for each action, each item contains (image/title), if item clicked it goes to the specified action activity.

Ex: Temperature action, the list item will contain: a title is called Temperature, and an image of a “thermometer.png”, if clicked it will go to Temperature Action activity.

1. **Search facility** in the application bar, where you filter by action[title].

Ex: you want to search for temperature action, you write “Temperature” in search bar, so it appears without the rest of actions.

1. **Options menu**:
2. Activity Log (item1): click, then go to Activity Log activity.
3. Profile (item2): go to Profile Activity.
4. Logout (item3): where you return back to login form.
5. **Activity Log**: for every action is made from your application, it stores the action with time stamp into the cloud using Firebase Realtime database, in case of disconnection, it caches the last version of the cloud using SQLite.
6. **Profile Activity**: Contains User Profile Picture, and Username and button logout.
7. **Action\_Activity**: for each action, implement its activity, for example, Temperature\_Activity, which displays the current temperature from the electronic circuit via cloud (firebase).

**Notes**:

* All actions are Realtime, but if network is disconnected, it will focus on cached SQLite, and when you click on any action from action ListView, it displays a toast message “Please Check your Network Connection!”.
* Actions are the features of the embedded system.

**Hardware Requirements:**

Your system must perform at least (4) actions / features of the embedded system.

**Note**: If the network is disconnected, we want to disconnect the circuit due to power consumption, for example: turn of led, fan, sensor reading.

Hardware Common Components:

1. Breadboard.
2. Jump wires.
3. LEDs .
4. ESP8266NodeMCU (recommended) or (ArduinoUno+ ESP8266 WiFiModule). Note: in case of ArduinoUno, (you might need a different library than described in laboratory).

Application Specific Components:

1. Temperature Sensor (LM35).
2. Ultra Sonic Sensor.
3. Motor.
4. LCD.
5. Keypad.

You need to discuss and confirm with your T.A. on your suggestions for the other components before implementing the project.

**Milestones:**

1. Front-End: Design and implement all requirements described in the software requirements section. (Deadline 12th April 2024).
2. Back-End: Apply database connection to cloud (Firebase) and caching (SQLite). (Deadline 26th April 2024).
3. Circuit Desing: Connect, implement hardware circuit, and integrate it to your mobile application. (Deadline 17th May 2024).

* A detailed per-person discussion will be held when delivering the project milestones that will account for the final mark of the project.
* Make sure to record a video of the project running.