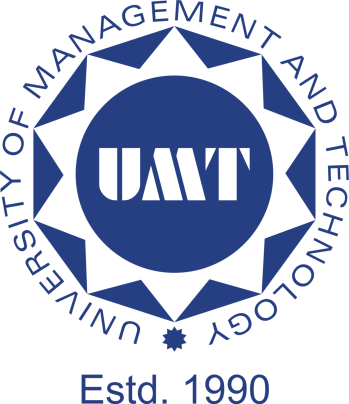
****

**Assignment no.2**

**Software Engineering**

**Topic:** Software process **Program:** BS(CS)

**Submitted to:** Sir Shabaz Qadeer **Semester:** 3rd

**Submitted by:**

Maria Haneef

Afra Fatima

Hifza Munawar

Hassan Ahmad

**Problem/proposal:**

**Smart Home Energy Consumption Monitor -**A real-time energy monitoring system that helps users identify areas for energy conservation and comprehend their consumption trends.

**Step 1: Planning and Requirements Collection:**

* **Identify Objectives:**

Your system need to monitor the amount of energy used by each device, examine use trends, and offer recommendations for ways to cut costs.

* **Investigation:**

Examine related energy-tracking apps or Internet of Things gadgets to learn how they work and what makes them easy to use.

**Functional requirement:**

* Real-Time Dashboard:
* Usage Reports and Trends:
* Energy Conservation Suggestions:
* Alert Notifications:
* Integration with IoT Devices (if feasible)

**Non-functional requirements:**

* Performance:
* Usability:
* Maintainability

**System Requirements:**

* **Hardware:** Smart meters for voltage, current, and power readings; energy metering nodes (wired/wireless); a central gateway; sensors like ACS712 for current measurement.
* **Software:** Android application for user interface, data visualization, and configuration settings.

**User Requirements:**

* Ability to monitor individual appliance energy consumption.
* Set thresholds for energy use (high/low cut-off).
* Access data remotely via smartphone.
* Share consumption data with third parties securely.
* This setup aims to optimize energy usage and reduce costs in smart homes.

**Set a Timeline:** Set reasonable due dates for each step of the work and divide it out over several months.

**Stakeholders**:

* Utilities
* Electric consumers (public)
* System integrator (Admin)
* IoT Technology providers(hardware, software)

**Step 2: Designing:**

* **Describe the primary elements:**

**Front-end:** Data visualization dashboard  
**Back-end:** Data processing server  
**Database:** retains usage history  
**Device API:** To interact with devices when working with the Internet of Things, you will require APIs.

**Step 3: Choosing a Technology**

**Front-end:** For a responsive interface, think about utilizing HTML, CSS, JavaScript, or a framework like React.  
**Back-end:** Python using Django or Flask to process and handle data requests.  
**Database:** MySQL or MongoDB for more reliable data processing; SQLite for local development.  
**IoT Integration:** If genuine IoT devices aren't available, you can utilize modeling. For real-time data, think about using MQTT or REST APIs to interface with devices.

**Step 4: Improvement:**

**Back-end development:** Create a database to hold information about energy usage.  
Provide APIs to update consumption records and get device use.  
**Front-end development:** Create the dashboard interface to provide device data (graphics, use insights) in an approachable way.  
Make a section or page with suggestions for energy conservation.  
**Source of IoT/Simulated Data:**  
Configure APIs to retrieve device data if you're using the IoT.  
If simulating, generate fictitious statistics to show the energy consumption of the gadget over time.

**Step 5: Evaluation:**

**Unit Testing:** Examine each of your code's distinct operations, including data storage, retrieval, and computation techniques.  
**Integration testing:** Verify that front-end elements accurately retrieve and present data from the back end.  
**User Testing:** To make sure your dashboard is clear and simple to use, have others utilize it. Get input regarding usability.

**Step 6: Records:**  
**Technological documentation: It i**ncludes any technological decisions you make as well as your codebase and APIs.  
**User Manual:** Compose a manual that explains how to utilize every dashboard function.  
**Upcoming Improvements:** Keep track of any future enhancements or new features that might be added.