Software Requirements Specification

for

Smart Energy   
Consumption Planner

**Version 1.0**

**Prepared by:**

**Moeez Salman 23L-3000**

**Ali Hassan 22L-7973**

**Abdullah Salman 23L-3032**

**Eishah Iqbal 23L-2587**

**Kashaf-ul-Eman 23L-3064**

**FAST NUCES, LAHORE**

**29-10-2024**

**Table of Contents**

**Table of Contents [ii](#_gjdgxs)**

**Revision History [ii](#_30j0zll)**

**1. Introduction [1](#_1fob9te)**

1.1 Purpose [1](#_3znysh7)

1.2 Document Conventions [1](#_2et92p0)

1.3 Intended Audience and Reading Suggestions [1](#_tyjcwt)

1.4 Product Scope [1](#_3dy6vkm)

1.5 References [1](#_1t3h5sf)

**2. Overall Description [2](#_4d34og8)**

2.1 Product Perspective [2](#_2s8eyo1)

2.2 Product Functions [2](#_3rdcrjn)

2.3 User Classes and Characteristics [2](#_lnxbz9)

2.4 Operating Environment [2](#_35nkun2)

2.5 Design and Implementation Constraints [2](#_1ksv4uv)

2.6 User Documentation [2](#_44sinio)

2.7 Assumptions and Dependencies [3](#_2jxsxqh)

**3. External Interface Requirements [3](#_z337ya)**

3.1 User Interfaces [3](#_3j2qqm3)

3.2 Hardware Interfaces [3](#_1y810tw)

3.3 Software Interfaces [3](#_4i7ojhp)

3.4 Communications Interfaces [3](#_2xcytpi)

**4. System Features [4](#_1ci93xb)**

4.1 System Feature 1 [4](#_3whwml4)

4.2 System Feature 2 (and so on) [4](#_2bn6wsx)

**5. Other Nonfunctional Requirements [4](#_qsh70q)**

5.1 Performance Requirements [4](#_3as4poj)

5.2 Safety Requirements [5](#_1pxezwc)

5.3 Security Requirements [5](#_49x2ik5)

5.4 Software Quality Attributes [5](#_2p2csry)

5.5 Business Rules [5](#_147n2zr)

**6. Other Requirements [5](#_23ckvvd)**

**Appendix A: Glossary [5](#_ihv636)**

**Appendix B: Analysis Models [5](#_32hioqz)**

**Appendix C: To Be Determined List [6](#_1hmsyys)**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Purpose

The purpose of this Software Requirements Specification (SRS) document is to define the software requirements for the Smart Energy Consumption App, Version 1.0. This app is designed to assist users in managing their electricity budget by providing personalized schedules and recommendations based on their energy consumption patterns. The SRS covers the features, functionalities, and system requirements for the entire app, detailing the components needed for effective energy monitoring, user interaction, and budget management. It specifies the requirements for the core functionalities of the application, including energy usage tracking, budget alerts, and scheduling optimization.

## Document Conventions

This Software Requirements Specification (SRS) document follows specific standards and typographical conventions to ensure clarity and consistency. The following conventions are used throughout the document:

* **Bold Text** is used for section headers and key terms that require emphasis.
* *Italic Text* is used to highlight important notes or to define new concepts.
* Monospaced font is used for code snippets, system messages, and references to software components or variables.
* Each requirement is assigned a unique identifier (e.g., R-1, R-2) for easy tracking and reference.
* Priorities for higher-level requirements are inherited by their detailed sub-requirements, unless explicitly specified otherwise.
* Numbering is used for hierarchical organization of sections, starting from higher-level requirements down to detailed specifications.

These conventions help in maintaining a structured and uniform representation of the requirements across the document.

## Intended Audience and Reading Suggestions

This Software Requirements Specification (SRS) document is designed to serve multiple stakeholders who are involved in the development, deployment, marketing, and usage of the Smart Energy Consumption App. The intended audience includes:

* **Developers:** Responsible for implementing the application based on the requirements detailed in this document. They should focus on the functional and non-functional requirements sections to understand what features need to be developed and how the system should perform.
* **Project Managers:** Oversee the project scope, timeline, and resources. They should refer to the entire document to ensure all requirements are met and aligned with the project objectives, especially focusing on the purpose, scope, and assumptions sections.
* **Marketing Staff:** Gain insights into the app’s features, benefits, and target audience, which will help in creating marketing campaigns. They can concentrate on the overview and feature summary to understand the app's core value propositions.
* **End Users:** Understand the features and benefits of the app and how it can assist in managing their electricity budgets. Users may refer to sections that outline the user interface, core functionalities, and use case scenarios to see how the app will function.
* **Testers:** Design and execute test cases based on the functional and non-functional requirements. Testers should focus on these sections to ensure the app meets the defined specifications and quality standards through comprehensive testing.
* **Documentation Writers:** Develop user manuals, help guides, and other documentation materials. They should review the SRS thoroughly to ensure that all user-facing documents accurately reflect the app’s functionalities and features.

The SRS is organized as follows:

1. **Introduction:** Provides an overview of the purpose, scope, and structure of the document.
2. **Overall Description:** Describes the general factors that affect the product and its requirements, such as system environment, user needs, and constraints.
3. **System Features:** Details each functional requirement, outlining the specific features of the app, including energy usage tracking, budget management, and scheduling functionalities.
4. **Non-Functional Requirements:** Specifies performance standards, usability, security, and other attributes that ensure the app’s reliability and efficiency.
5. **External Interface Requirements:** Describes how the app interacts with external systems, devices, or users.
6. **System Architecture and Design:** Explains the underlying architecture and technical specifications of the app.
7. **Appendices and Glossary:** Includes definitions of terms, acronyms, and any supplementary information.

For a clear understanding, readers are recommended to start with the **Introduction** to get an overall view of the app’s purpose, scope, and structure. Depending on their role, they can then proceed to the **System Features** or **Non-Functional Requirements** for specific details. Project managers should read the entire document, while developers and testers should focus on the technical and functional sections. Marketing and documentation teams can prioritize sections that describe features and user interactions.

## Product Scope

The Smart Energy Consumption App is a software solution designed to help users efficiently manage their electricity consumption and budget. The app aims to empower users to monitor their energy usage in real time, set consumption targets, and receive personalized schedules and alerts to optimize energy use. By providing detailed insights and actionable recommendations, the app enables users to reduce their electricity bills and adopt energy-saving habits.

The key benefits of the app include improved energy efficiency, cost savings, and better awareness of consumption patterns. Users can schedule energy-intensive tasks during off-peak hours to take advantage of lower rates, set budget limits to avoid unexpected expenses, and receive notifications when consumption exceeds preset thresholds.

The development of this app aligns with broader corporate goals of promoting sustainable energy consumption and supporting users in their transition to more eco-friendly practices. It is part of a larger business strategy to innovate within the energy sector by providing smart, data-driven solutions that cater to both individual and commercial users. The app not only helps users save money but also contributes to overall energy conservation efforts, supporting green initiatives and responsible consumption.

For a detailed overview of the app's vision and long-term objectives, please refer to the Vision and Scope document associated with this project.

## References

This Software Requirements Specification (SRS) document refers to several other documents and resources that provide additional context and details necessary for understanding and developing the Smart Energy Consumption App. These references include guidelines, standards, and specifications that complement the information provided in this document. Below is a list of key references:

1. Vision and Scope Document for Smart Energy Consumption App  
   *Author:* Project Development Team  
   *Version:* 1.0  
   *Date:* October 2024
2. User Interface Style Guide  
   *Author:* UX Design Team  
   *Version:* 2.1  
   *Date:* September 2024
3. [IEEE Standard for Software Requirements Specifications](https://standards.ieee.org/standard/830-1998.html)  
   *Author:* Institute of Electrical and Electronics Engineers (IEEE)  
   *Version:* 1998
4. Use Case Document for Smart Energy Consumption App  
    *Author:* Business Analysis Team  
     
   *Version:* 1.2  
   *Date:* August 2024
5. External API Documentation  
   *Title:* API Reference for Smart Meter Integration  
   *Author:* Integration Team  
   *Version:* Latest (refer to API platform)

These references are critical for understanding the broader context, design standards, and integration requirements associated with the Smart Energy Consumption App. Readers should refer to these documents for more detailed information on specific aspects of the system.

# Overall Description

## Product Perspective

“Smart Energy Consumption Planner” is an independent software system whose main aim is to help the user to minimize the cost, manage demand and improve efficiency by providing or generating a schedule for the user.  
  
**Product Type and Scope:**

The” Smart Energy Consumption Planner” is a new, standalone product whose main aim is to provide aid to the users to manage the cost of their electricity bills. This app in future can further be modified to function as a module with a smart home or energy management ecosystem.  
  
**Purpose of SECP in Overall system:**

* It predicts the electricity rates depend upon the data provided by the user and generates a schedule according to the given information.
* It generates a schedule on a weekly basis according to the budget provided by the user.

It warns the peak hours to the user to reduce energy consumption.  
  
  
**External Systems:**

It includes a secure database which stores the data of the user securely including their schedule history and appliance power consumption to maintain privacy.

## Product Functions

* **User Sign-Up and Profile creation:**The system allows the user to sign up to the system.
* **User Login:**  
  The system allows the user to log in by entering valid credentials.
* **Password Management:**The system shall remember the user’s password for future login sessions.
* **Plan Generation:**  
    
  The system shall generate a new plan for energy consumption.
* **Device Management:**The system shall allow the user to add new devices.
* **View Past Data:**

The system shall allow the user to view past schedules.

* **Feedback**The system shall allow the user to submit feedback at the each of week.
* **Energy Tracking:**

The app shall track electricity consumption and allow users to modify schedule weekly.

* **Peak Hour Notification:**The system shall notify users about peak hours. User Classes and Characteristics:  
    
  The Smart Energy Consumption Planner (SECP) is designed to cater to various user types based on their roles, frequency of use, and specific needs within the system. Below are the primary user classes and their pertinent characteristics:  
    
  1.General User:  
    
  General users are homeowners, renters or small business owners who want to manage and reduce their energy consumption.  
     
  Characteristics:  
    
  . Technical Expertise: Basic to intermediate technical skills, typically comfortable with common apps and interfaces.  
    
  .Usage Frequently: Regularly uses the app to check schedules and manage appliances, view or add appliances.  
    
  .Key Functions: Profile management, viewing the central dashboard, appliance management, schedule generation, receiving peak hour alerts, and accessing energy-saving tips.  
    
  2.Admin User:  
    
  The admin user has the sole responsibility of managing peak hour settings within the SECP to ensure that all users receive accurate notifications and energy-saving suggestions.  
    
  Characteristics:  
    
  . Technical Expertise: Basic familiarity with peak hour settings and system configuration, no advanced technical skills required.  
    
  .Usage Frequency: Accesses the system as needed to adjust peak hours.  
    
  .Key Function: Changing and updating peak hours as necessary.

## Operating Environment SECP is a desktop application and is designed to operate in a moderate, flexible environment, ensuring compatibility with common hardware and software configurations. Below are some minimum and recommended requirements: .Minimum Requirements: 1.Processor:Dual Core processor (Minimum intel Core i3 or equivalent) 2.Memory:Minimum 2 GB of RAM for optimal performance. 3.Storage: Requires at least 500 MB of free disk space for installing the application and MySQL Server for data storage*.* 4.Oprating System: Windows 7 or later .Recommended Requirements: Desktop or laptop with 8GB RAM or higher, quad-core CPU, and a stable internet connection for real-time data access.

## Design and Implementation Constraints The Smart Energy Consumption Planner (SECP) must adhere to several design and implementation constraints that could impact its development, deployment, and overall functionality. 1.Hardware Limitation: The software must be optimized for performance on low-end devices to accommodate users with minimal hardware resources, ensuring the application remains responsive and functional across a range of devices. 2.Database Constraints: The system will use a secure database (SQL) which may impose limits on data storage, retrieval speed, and connection management. The design must accommodate potential limitations on concurrent connections and data access. 3.Security Considerations: Security must be prioritized throughout the development process, including . Implementing encryption for data at rest and in transit. . Ensuring secure user authentication and authorization mechanisms. 4.Design Conventions: The system design will ensure to be user-friendly by using eye catching color schemes and accessibility considerations for users with disabilities. 5.Languages The SECP should be developed using supported programming languages that align with the team's expertise (C++) to facilitate easier maintenance and future updates.

## User Documentation The following documentation will accompany the Smart Energy Consumption Planner (SECP) to assist users in effectively utilizing the software and understanding its features: 1.InApp Help: The user can get help via feedback option in the app to get further details and assistance from the development team or the company.

**2.Expected Formats:**The tutorial videos of this application will be on powerful social media platforms like YouTube.

## Assumptions and Dependencies

**Assumptions:**1.**User Access and Familiarity:**

It is assumed that users of the Smart Energy Consumption Planner will have basic computer skills for app usage, as well as familiarity with typical household appliances and their energy consumption.  
  
2. **Availability of Data**:  
  
The project assumes that users will provide accurate information about their appliances, including power ratings and usage patterns. Inaccurate data could lead to ineffective energy consumption optimization.  
  
3. **Admin User Role**:  
  
It is assumed that the admin user will have a clear understanding of the system and the authority to make changes, such as modifying peak hours. Changes made by the admin will be communicated effectively to all users.  
  
4. **User Feedback and Iteration**:  
  
The project assumes that user feedback will be collected and analyzed continuously to improve the application's functionality and user experience. Users are expected to engage with the feedback process.  
  
 **Dependencies:**1. **Database Management System**:  
  
The project will depend on a database system SQL for storing user data, appliance information, and scheduling details. The choice of Data Base Management Studio will impact data retrieval speed and reliability.

**2. Frameworks and Libraries:**The project will depend on specific frameworks and libraries, such as:  
  
**.** C++/CLI: For developing the Windows Forms application.

# External Interface Requirements

## User Interfaces

The Smart Electricity Consumption Planner (SECP) application provides an intuitive, light-themed graphical user interface (GUI) that allows users to manage their electricity consumption and scheduling with ease. The main dashboard offers four primary buttons: **Add Appliance**, **Review Previous Schedule**, **Generate New Plan**, **View Appliances** and **View Profile**, each represented by large, visually distinct icons for simplified navigation.

1. **Layout and Visual Elements**:
   * The application has a consistent layout across all screens, with a light blue header that displays the SECP logo, and a user profile icon.
   * The default font size is standardized across the app, with no customization options available.
   * All primary buttons are styled with light blue borders, enhancing visual coherence throughout the interface.
2. **Standard Screens and Functions**:
   * **Login and Sign-Up Screens**:
     + Users can log in or create an account by entering personal details like email or phone number and password.
     + Standard navigation buttons such as **Sign in** and **Create Account** are provided, along with a “Remember me” option to retain login information.
     + Error messages display in cases of invalid input, such as incorrect passwords or improperly formatted email addresses.
   * **Main Dashboard**:
     + The main screen allows users to input their city name, electricity rates, and budget. Fields like "Enter your city name" and "Enter your budget for electricity bill" guide the user in setting up energy preferences.
     + A “Confirm” button lets users submit data to progress to the next stage, while validation checks ensure all required fields are properly filled.
   * **Device Management Screen**:
     + Users can add appliance details like name, watt usage, and usage duration, with clear labels (e.g., “Enter appliance name,” “Enter watt usage”) for each field. Once entered, users can click “Register” to add the appliance to their energy plan.
   * **Feedback and Notifications**:
     + Pop-up notifications inform users about updated tariff rates, and weekly feedback option let users share their experience with the app.
3. **Error Messages and Notifications**:
   * Error messages appear as pop-ups at the bottom of the screen (e.g., "Incomplete fields" or "Invalid entry") and can be dismissed manually or automatically disappear after a few seconds.
   * Timely notifications alert users to important updates, like new tariff rates, without interrupting the user experience.
4. **Screen Images**:
   * Sample screen images of the app are provided in Appendix A, serving as a visual reference for development.
5. **Keyboard Shortcuts**:
   * No keyboard shortcuts are included, as all interactions are performed through clickable buttons to simplify navigation.

## Hardware Interfaces

The Smart Electricity Consumption Planner application is a desktop application optimized for offline functionality and is compatible with standard desktop or laptop computers. It does not require specialized hardware beyond basic computer peripherals and runs solely on Windows environments.

**Supported Device Types**:

* **Processor**: Dual-core processor (minimum Intel Core i3 or equivalent).
* **Memory**: Minimum 2 GB of RAM for optimal performance.
* **Storage**: Requires at least 500 MB of free disk space for installing the application and MySQL Server for data storage.
* **Operating System**: Windows 7 or later.

The app does not interface with IoT smart meters or external devices at this stage, though future updates may integrate with smart home devices for real-time data exchange and automated energy scheduling.

## Software Interfaces

The SECP application interacts with a local database system and standard Windows libraries to manage data storage and application functions. It is designed to operate independently of external software components, though certain core interfaces enable data handling and user management.

**Key Software Components**:

1. **Database**:
   * **MySQL Server Management Studio** is used as the primary database for storing user profiles, device details, energy plans, and consumption history.
   * The database allows for reliable and consistent data transactions using SQL queries, which support user data input, schedule generation, and historical tracking of electricity usage.
2. **Operating System**:
   * The application is compatible with Windows-based operating systems and utilizes native Windows libraries to manage GUI and local storage functions.
3. **Future Extensions**:
   * While the app currently operates offline, future versions may include integration with APIs to fetch real-time tariff data or offer cloud storage options for user data backup. Currently, only the admin can update the tariff rate.

No additional libraries or external software are integrated, adhering to the system’s design constraints and the user’s offline use case.

## Communications Interfaces

Though the SECP application is primarily offline, it provides in-app notifications to enhance user engagement and ensure timely information delivery.

1. **Pop-up Notifications**:
   * SECP sends pop-up notifications about tariff updates. These notifications appear within the app, particularly when the user opens the application after manual tariff data updates.
   * No specific message formatting is applied, as notifications are generated directly within the app’s UI, ensuring readability.
2. **Future Communication Protocols**:
   * If cloud-based data storage or tariff APIs are integrated, HTTPS protocols will be implemented to secure data transmission, aligning with industry standards for communication security.

The SECP app does not currently rely on protocols like FTP or HTTP due to its offline design. Future versions with cloud integration will include relevant communication standards for secure data exchanges and synchronization mechanisms.

# System Features

**4.1 User account management:**

**4.1.1 Description and priority:**

This feature allows the user to log in and view his/her profile if they already have an account and allows to sign up for a new user. The user can also view his/ger profile and edit the neseccary details.This is a high priority feature.

**4.1.2 Stimulus/Responses**

The user enters his/her user name and password if they already have an account and clicks “log in” upon which the main dashboard of the app will be shown to them.The app does not allow any unregisered user to log in to the system.If any unregistered user will try to log in, an error message will be displayed and the user will be asked to sign up.

A new user clicks “sign up” to register himself/herself, thereafter, the log in page will automatically shows up in front of him/her. The clicks on “profile icon” on the profile icon, upon which his/her profile will be displayed. The user can make necessary changes to his/her credentials by clicking the “edit” icon.

**4.1.3 Functional Requirements:**

**4.1.3.1** The system shall allow the user to enter his/her username and password.

**4.1.3.2** The system shall not log in any unregistered user and shall display an error message.

**4.1.3.3** The system shall allow a new user to sign up with the necessary credentials.

**4.1.34** The system shall allow the user to view his/her profile.

**4.1.3.5** The system shall allow the user to view his/her profile.

**4.1.3.6** The system shall allow the user to edit his/her profile.

**4.2 Appliances Management:**

**4.2.1 Description and priority:**

This feature allows the user to add, view, and edit the appliance's description (wattage and duration of use). This is a high priority feature.

**4.2.2 Stimulus/Responses:**

User clicks the “add appliance” option from the main dashboard. The user then enters the fields and clicks “add”. The new appliance along with its details is added to the database. The same interface appears before the user.

The user can view the existing appliances by clicking the view appliances option from the dashboard. The user is shown a list of appliances.

The user can edit the appliances by clicking the “edit icon” in the view appliances interface. The new changes are saved on the data base.

**4.2.3 Functional Requirements:**

**4.2.3.1** The system shall allow the user to enter the necessary details of the electronic appliance to be added.

**4.2.3.2** The system shall allow the user to view the appliances’ details.

**4.2.3.3** The system shall allow the user to make changes in the appliances’ details.

**4.3 Schedule management:**

**4.3.1 Description and priority:**

This is the main feature of the app which allows the user to generate a schedule based on the existing list of appliances and their details. This is a high priority feature.

**4.1.2 Stimulus/Responses**

The user clicks the “Generate schedule” option from the main dashboard. Thereafter, a schedule is generated for him along with the expected bill after 30 days. The schedule is not generated if the list of appliances is empty.

If any appliance description is changed, a new schedule will be generated for the remaining days.

The user can also view the current schedule by clicking the “current schedule” option on the main dashboard. The system shows the schedule to the user. This option appears only after a schedule is generated.

**4.3.3 Functional Requirements:**

**4.3.3.1** The system shall allow the user to generate a new schedule.

**4.3.3.2** The system shall allow the user to view the current schedule.

**4.3.3.3** The system shall allow the user to customize the schedule by changing the appliance(s)’ details.

**4.4 User support:**

**4.4.1 Description and priority:**

This feature provides support to the user by providing a feedback option, allowing to view the past records of the bill, and by notifying the change from the admin side.

This is a medium priority feature.

**4.4.2 Stimulus/Responses**

The user can click the “support icon” from any interface and provide feedback anytime. The feedback option is also provided daily by the app automatically. The feedback is sent to the admin side via email.

The user is notified if any change is made from the admin on the admin side.

The user clicks the “view past record” option on the main dashboard, and a list of past records will be displayed before him/her. The record covers a year (12 months) from the current month.

**4.4.3 Functional Requirements:**

**4.1.4.1** The system shall allow the user to provide feedback on a daily basis.

**4.1.4.2** The system shall allow the user to provide feedback after a month

**4.1.4.3** The system shall allow the user to customize the current schedule if the feedback given is bad.

**4.1.4.4** The system shall notify the user about the changes made on the admin side.

**4.1.4.5** The system shall allow the user to view the past records.

# Other Nonfunctional Requirements

## Performance Requirements

∙ **System Response Time:** The system should respond to user interactions (e.g., entering appliance details, refreshing data) within 2 seconds and generate an appliance usage schedule within 5 seconds for an average household setup with 10-15 appliances.

∙ **Capacity:** The system should support up to 1,000 simultaneous users without performance degradation, ensuring consistent response times and accurate computation.

## Safety Requirements

∙ **Data Protection:** Measures should ensure no unauthorized access to user data (e.g., appliance information, consumption patterns) to prevent misuse.

∙ **System Safeguards:** The system should incorporate features that prevent data loss.

∙ **Regulatory Compliance:** The design should align with any applicable safety regulations, especially concerning electrical appliance scheduling, to avoid overloading circuits or increasing energy consumption risks

## Security Requirements

∙ **User Authentication:** The system should protect sensitive information, such as consumption history and personal account details.

​

∙ **Data Encryption:** All stored and transmitted user data should be encrypted, especially for sensitive information like login credentials and personal usage data.

∙ **Compliance with Privacy Standards:** Ensure adherence to external privacy regulations, such as GDPR or local privacy standards, relevant to data handling within the application.

## Software Quality Attributes

∙ **Usability:** The interface should be intuitive, with 80% of non-technical users able to perform key tasks (like schedule generation) within 3 minutes unaided.

∙ **Reliability:** Ensure the system’s accuracy, aiming for 95% accuracy in consumption data and tariff calculations, as well as 99% accuracy for computational tasks.

## Business Rules

∙ **Role-Based Access:** Differentiate access rights for system users, limiting critical operations (e.g., data export or user management) to authorized personnel.

∙ **Scheduling Constraints:** Allow users to set personalized schedules within safe operational limits to prevent unintentional energy overconsumption.

∙ **Feedback Collection Rules:** Collect feedback at a designated frequency (e.g., monthly or weekly), and ensure that survey results inform future updates in line with stakeholder preferences.

# Other Requirements

Everything has been covered in SRS There is nothing left to right here*.*

**Glossary  
  
1. Admin User:**A user with elevated privileges to manage system-wide settings, such as peak hour configurations and tariff updates.

**2. API (Application Programming Interface):**A set of protocols and tools that allow different software applications to communicate with each other. In this context, it may refer to future integration with smart meters or real-time tariff services.

**3. CLI (Command-Line Interface):**A text-based interface used for interacting with software by typing commands directly, though not typically used by end users of this application.

**4. Consumption History:**A record of past energy usage tracked and stored in the system database to provide insights and help generate future energy consumption plans.

**5. Device Management:**The process of adding, updating, and tracking information about appliances used by the user, including their power consumption details.

**6. Encryption:**A security measure that encodes data to prevent unauthorized access, ensuring that only authorized users can read the information.

**7. GDPR (General Data Protection Regulation):**A European Union law on data privacy and protection, which may influence how the application handles user data, especially if the app is distributed internationally**.**

**8. GUI (Graphical User Interface):**The visual interface through which users interact with the software, including buttons, menus, and icons.

**9. MySQL Server:**A relational database management system used to store and manage application data, such as user profiles, energy consumption logs, and appliance details.

**10. Peak Hours:**Periods of the day when energy consumption is typically higher, often resulting in higher electricity rates.

**11. Role-Based Access:  
A security feature that provides different levels of access and functionality based on a user’s role, such as admin or general user.**

**12. Scheduling Constraints:**Limits imposed on appliance usage schedules to ensure safe and efficient energy consumption.

**13. SECP (Smart Energy Consumption Planner):**The name of the desktop application described in this SRS, which helps users manage their energy consumption and budget.

**14. SQL (Structured Query Language):**A programming language used for managing and querying data in the MySQL database.

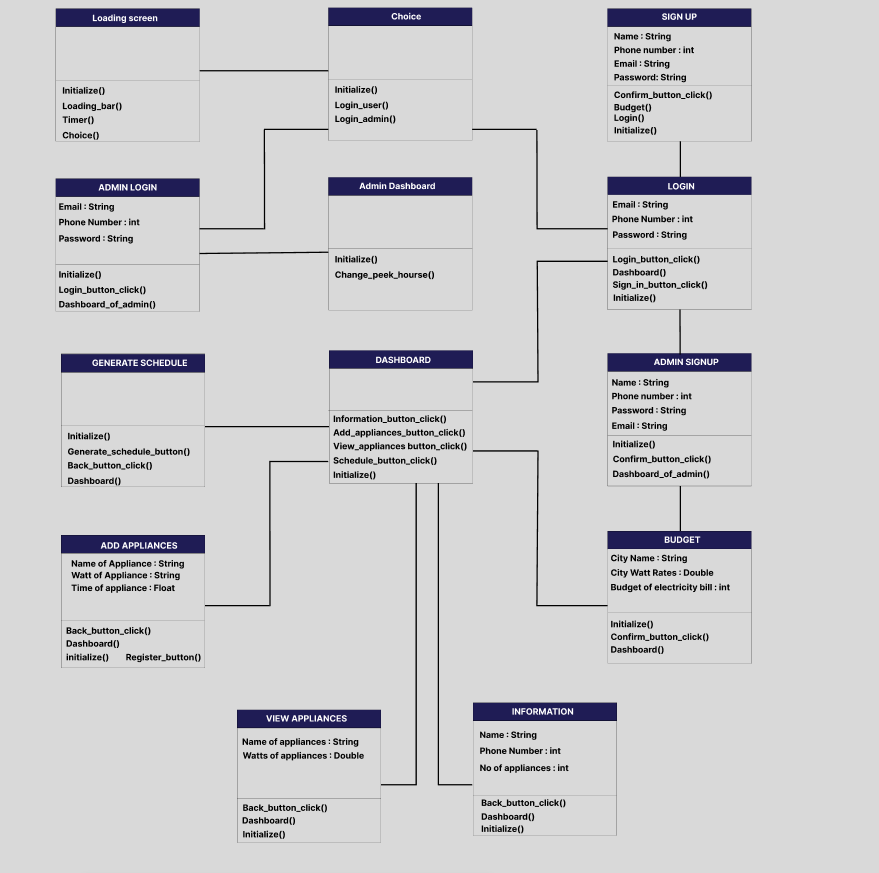
**15. Tariff Rate:**The cost of electricity per unit, which may vary based on peak and off-peak hours, affecting the user’s energy consumption plan.

**16. User Feedback:**Comments, suggestions, and complaints collected from users to improve the app’s features and user experience.

**17. User Profile:**The personal information and preferences stored for each user, including login credentials and appliance data.

**18. Usability:**A measure of how easy and efficient it is for users to perform tasks within the app without external assistance.

**Appendix B: Analysis Models**

****

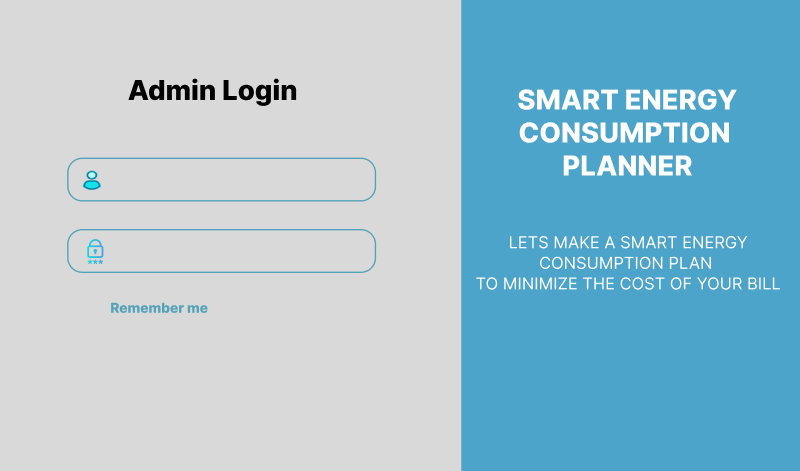
**A white sheet with blue squares and black lines

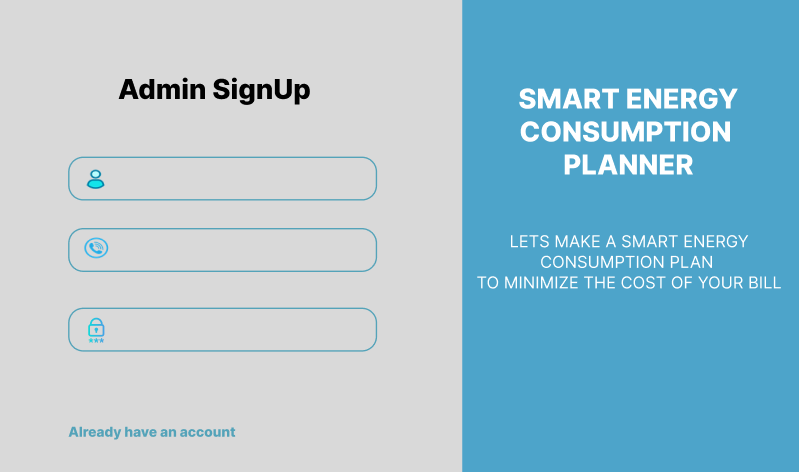
Description automatically generated**

**A diagram of a social network

Description automatically generated**

**Appendix A: GLOSSARY**

****

**A screenshot of a computer screen

Description automatically generated**

**A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generatedA screenshot of a login form

Description automatically generatedA screenshot of a login screen

Description automatically generated**

**A screenshot of a login form

Description automatically generatedA screenshot of a home appliance

Description automatically generated**