ASHESI UNIVERSITY

ASHESI UNIVERSITY COLLEGE

PROJECT DESIGN DOCUMENT

SOLAR POTENTIAL CALCULATOR

OBJECT-ORIENTED PROGRAMMING: CS213

DR. DAVID EBO ADJEPON-YAMOAH

By

Afun Cal Senam, Tracy-Phyron Jinor, Nana Asantewaa Osei Bonsu, Tawanashe Makuvaza

DECEMBER 9, 2025

Table of Contents

SECTION 1: INTRODUCTION

Energy insecurity remains a major challenge in many African countries due to rising costs in the production of electricity (IEA, 2022). There are over 600 million people in sub-Saharan Africa who still lack access to proper electricity (World Bank, 2023). This is hindering Africa's economic growth and daily household activities.

Africa has received some of the world's highest levels of solar irradiation (IRENA,2021), making solar power a readily available resource. Though this has been established we have had a lot of households struggling to effectively determine and estimate just how much sunlight their rooftops receive and what size solar systems would meet their needs.

As a good solution, our project, the solar calculator, introduces a Solar Management System, an object-oriented Java application that calculates the rooftop solar capacity, supports accurate installation planning and helps technicians and administrators manage solar data more efficiently.

The significance of this solution is to provide a better and cheaper energy source for households, which can lead to an improvement in economic growth and household activities. In our project, we utilized a lot of OOP concepts such as Inheritance, Abstract, Encapsulation, Association, as well as file writing and reading.

SECTION 2: SYSTEM ARCHITECTURE

This section presents a clear structure of the Solar Calculators' design – the tools used and how a user would interact with the system.

Tools Used

We built the system using Java and the Java Swing Framework. We used Java to handle the backend logic of the system and Java Swing as well was used to create the Graphical User Interface (GUI), enabling the system to run as a desktop application while fully adhering to object-oriented programming principles.

System Flow

Our system was purposely designed for three main types of users:

- Homeowners
- Solar Technicians
- Adminstrators

When the system begins, Homeowners and Technicians are required to enter only their email address and password, which the system will compare with the data stored in users.txt in order to grant access. All passwords were hashed for security purposes.

However, the Administrators are required to enter a Company Verification Code first, which must be at least six characters long. After successful validation, the administrators are granted access to the administrator dashboard.

Once logged in, users are redirected to dashboards that match their role. The section below describes the functionality available to each user type.

Homeowners:

We assumed that homeowners do not have access to professional instruments for recording solar data, such as sunshine recorders or irradiance meters. Therefore, to process calculations, we used data uploaded from the administrator's page, such as the average solar hours respective to the town the user selected during signup.
Homeowners are required to enter:

- Roof length
- Roof width
- Estimated shading level (on a scale of 1 - 10)
- Daily energy demand (in kWh)

Output information from the system:

- Total roof area
- Amount of roof area that could be used after shading losses
- The number of panels needed that will ensure that they meet the daily energy demand
- The recommended number of panels for effective performance

All the assessment are saved as reports to ensure that the homeowners can either view previous records or export reports as txt file.

Solar Technicians

Technicians have almost the same functionality as homeowners, but with just few additional controls. However, unlike the homeowners' side, technicians are required to enter the measured average sun hours themselves because we assume they have access to the right equipment to measure such values. Additionally, technicians get more detailed reports with calculations. Below is a list of information contained in the technician's report.

- Formulas used for calculations
- Certain important assumptions made about system efficiency factors
- Clear outline of energy demand

Administrators

Administrators are responsible for managing and maintaining the system
Their roles include:

- Adding and updating towns and their average sun hours
- Registering and updating available solar panel types
- Resetting passwords for homeowners and technicians using user IDs
- Resetting the company verification code
- Administrators ensure that all users operate with accurate, up-to-date system data

Finally, administrators get a short analysis per town such as the total capacity and the number of installations.

SECTION 3: OUR UML CLASS DIAGRAM AND INTEGRATION DISCUSSION

This section shows the UML class diagram for the system and explains very well exactly how the major components interact to deliver the required functionality. Our design prioritised modularity (meaning the system is broken down into separate, independent parts (modules) that work together), clarity of responsibilities (each class having a clearly defined role) and proper relationship between classes(the connection between classes is logical and simple, making the system easier to maintain).

The UML Class Diagram

Our UML class diagram focuses on modelling the full Solar Management System by organizing the application into three main and effective layers:

- OurUser Interface Layer (Page Classes)
- Our Core System Layer (SolarManagementSystem and related data objects)
- Our Utility Layer (FileOperations and ReportGenerator)

Our UML diagram also captures the three user roles for our system. These roles are: Homeowner, Technician and Admin.

Fig 4.a (THE UML CLASS DIAGRAM)

The UML is available in the project document folder.

Description Of The System Components

The classes in the UML model could be grouped as follows:

- **User Roles and Page Classes:** At the top left of the UML diagram, we have the user page classes:
    i. **Login Page and Sign Up Page:** These pages are responsible for enabling the user to onboard and go through the authentication processes.
    ii. **HomeownerPage, TechnicianPage,** and **AdminPage** serve as the main dashboard for each user be it the homeowner, technician and admin.
    iii. Administrative Pages (AdminAddPage, AdminLookupPage, AdminSignupPage) support system maintenance and task

These classes represent the actual GUI (Graphical User Interface) portion of our project. They do not contain core logic, however they define exactly how the user will interact with the system.

- **User and User Type:** Our UML also consists of the User class with attributes: id, name, phone, and password hash. There is also the user type enum which defines these valid roles:
    i. **HOMEOWNER**
    ii. **TECHNICIAN**

iii. ADMIN

These roles are very important as they determine which UI page each user is directed to and what they can access inside the system.

- **The Core System: SolarManagementSystem:** This is the most significan and important part of the architecture of our project. It contains input attributes such as the roof dimensions, panel information, shading levels, system settings, it also has operational attributes such as the usage data, cost calculations, recommendations and the user-management attributes (records of the users and the systems).

- **Utility Classes:** For our project, we have two main important utility classes that are helping to support our entire Solar Potential Calculator system. These are the file operations (such as reading from a file and writing to a file). There is also the report generator that produces analytical outputs such as solar reports.

Person Abstract class
This is the blueprint for the user on the system. Attributes: name, email, and phone, along with their respective getter methods.
User Class
This class extends the Person class to provide the user with a more distinctive role within the system. In addition to the attributes obtained from the Persons, users are further defined with other attributes such as the usertype, their password, or their town of residence.
Report Generator Interface

We also have two interfaces for handling files. The first is the ReportGenerator, which assists in generating user reports. It contains the method generateUserReport, which assists in looking up user records by means of filtering records.txt by users' id.
The File Operations Interface:

The file operation interface consisted of two main methods, which were the saveToFile() method and the loadFromFile(). These methods allowed us to save sensitive user details such as the password, username and email into a file and read from them to log them back in.

The HomeownerRecord Class

The HomeownerRecord class was a concrete class with attributes such as userId, date, and time, among others, which allowed access to certain important information concerning the homeowner's results. We had methods such as the getDateTime(), getTown(), and getRoofArea() to simulate this.

The PanelSpec class

This class was inheriting from the SolarManagementSystem and enabled us determine the specific type of solar panel that could be used based on the calculated Solar Potential.

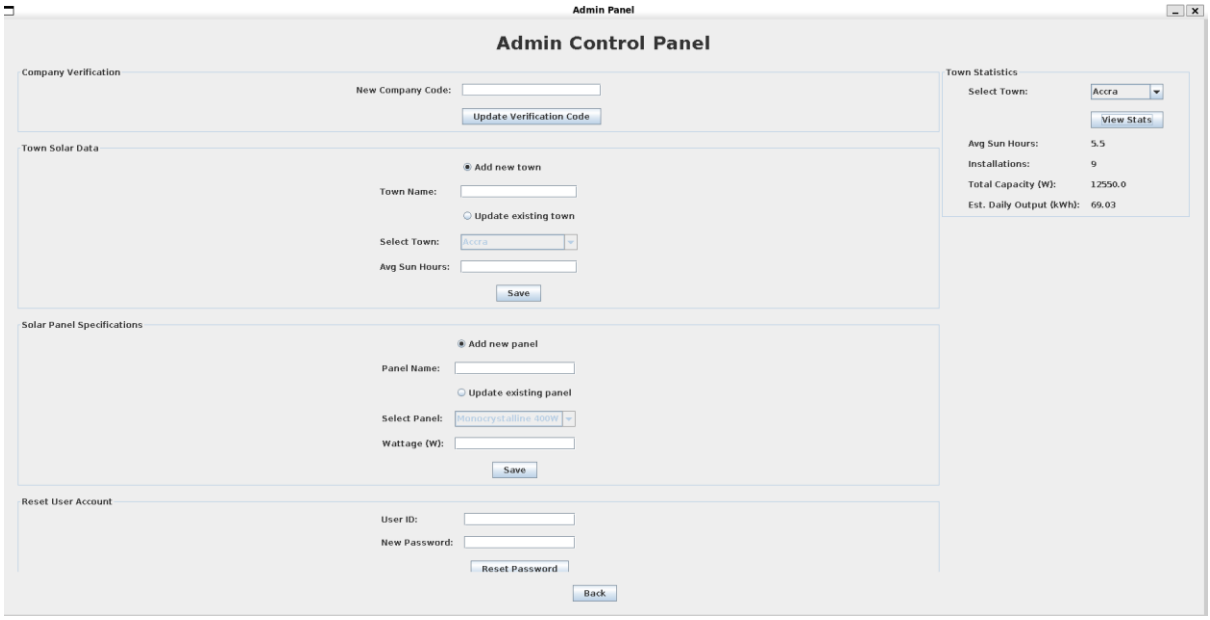
Explanation Of Relationships Between Classes

The UML diagram consist of several relationship types:

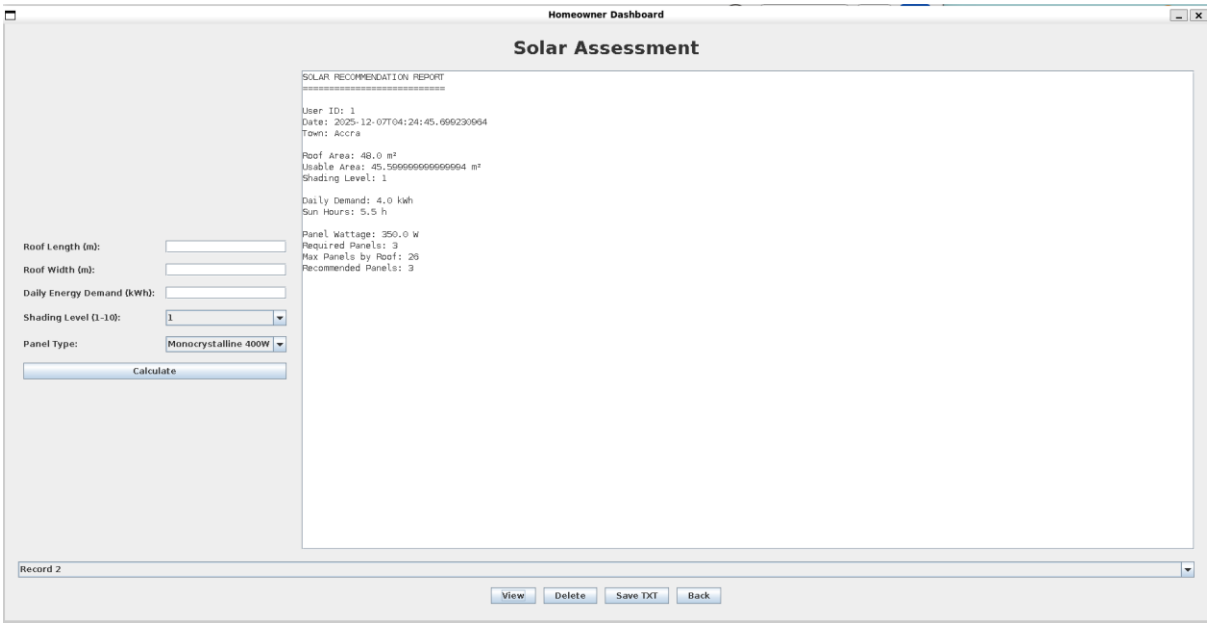
- The association between the SolarManagementSystem class and its related data classes such as; Homeowner Record, Panel Spec and Solar Cellar indicate that the system operates on instances of these classes for its core functionality.
- The dependency relationships from SolarManagementSystem to both FileOperations and ReportGenerator demonstrate that the system relies on these external utility components to perform file-handling and reporting tasks. Because they are dependencies rather than associations, they are used temporarily and are not held as long-term attributes within the system
- The relations between the roles of the user and the classes of the Page which are page classes a reflect the access paths. These associations determine the relation between the various types of users (Homeowner, Technician, Admin) and particular system interfaces, which is the access-control structure of the system and the flow of interaction.

Interfaces

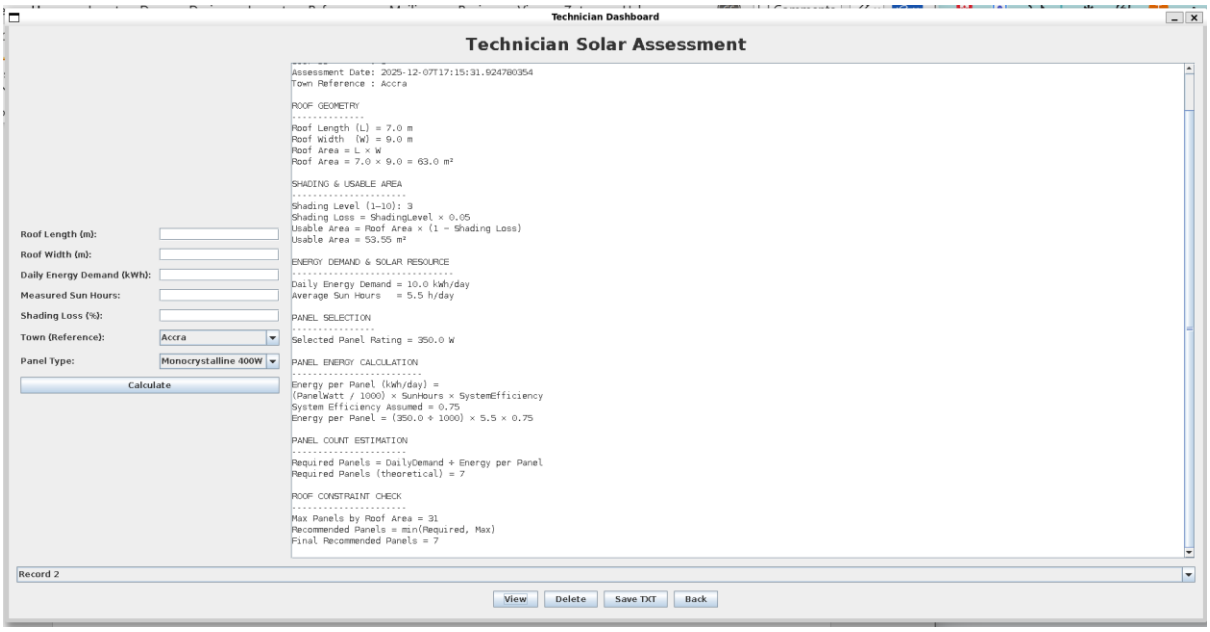
Admin Dashboard



Homeowner Interface



Technician Dashboard



References

Africa. “Africa Energy Outlook 2022 – Analysis - IEA.” *IEA*, 20 June www.iea.org/reports/africa-energy-outlook-2022?utm_source=chatgpt.com.

Group, World Bank. “Basic Energy Access Lags amid Renewable Opportunities, New Report Shows.” *World Bank*, World Bank Group, 6 June 2023, www.worldbank.org/en/news/press-release/2023/06/06/basic-energy-access-lags-amid-renewable-opportunities-new-report-shows?utm_source=chatgpt.com.

International Renewable Energy Agency (IRENA) & African Development Bank (AfDB). (2022). Renewable Energy Market Analysis: Africa and Its Regions – Summary for Policy-Makers. IRENA, Abu Dhabi & AfDB, Abidjan.

GROUP MEMBERS

1. Afun Cal Senam
2. Tracy-phyron Jinor
3. Nana Asantewaa Osei Bonsu
4. Tawanashe Makuvaza