

CSIT121 Assignment 2

Report Generation System for Green Star

Due Date: Week 8
Marks: 10

1. Objective

The objective of Assignment 2 is to apply the objective-oriented design and programming knowledge gained from Week 1 to Week 8 lectures. In this assignment, students will design a Python program using UML class diagrams to aid the Green Star in generating reports for projects with different ratings, scores, locations, companies, and/or other special requirements (see Fig. 1).

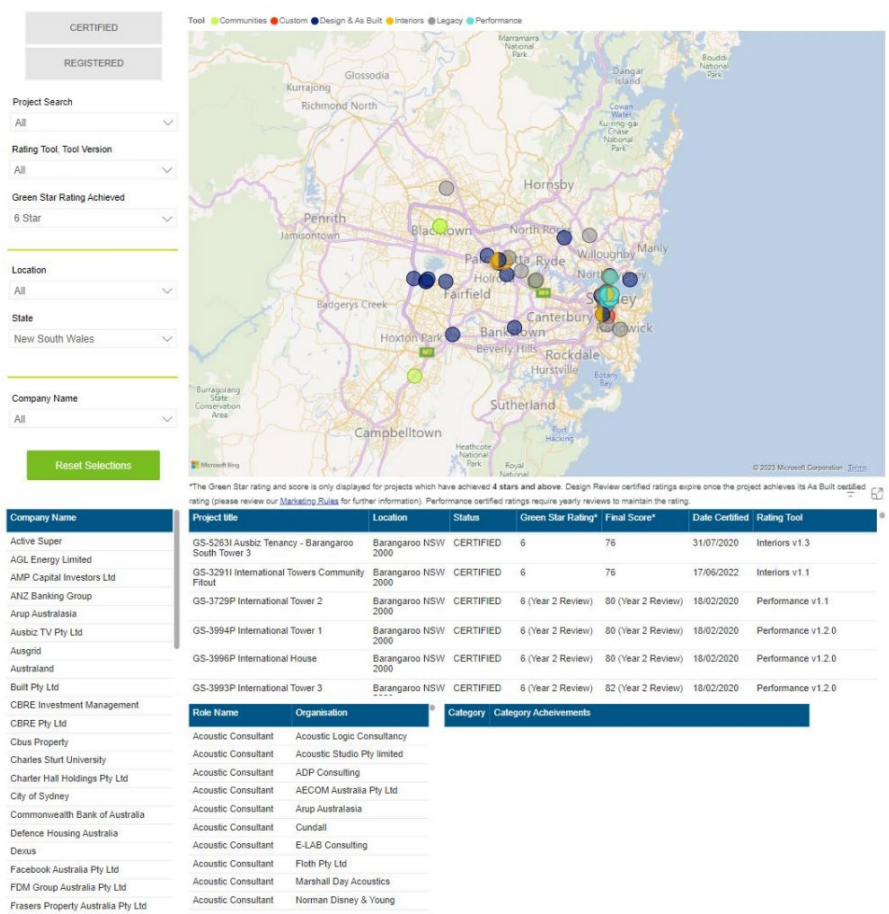


Fig. 1: Summary of projects rated level 6 in NSW

Fig. 1 shows a summary of projects, which rated at level 6 in New South Wales. The proposed Python program should collect user requirements (e.g., location, ratings, etc.), through user inputs and generate a comprehensive text report for projects that meet user requirements. Also, the proposed Python program shall be robust to the user's inputs and be able to handle unexpected/inappropriate usage to avoid the program crash by using the Python exception handling mechanism. Besides, the Python program should allow users to export and import the generated text report using File I/O, for printing and documentation purposes. In Assignment 2, students are asked to design program tests using the Python unittest module and generate the coverage report using the coverage module.

2. Project Description

In this assignment, students are tasked with designing a Python program that accomplishes the following:

2.1 Design

1. Analyze the Green Star scenario using OOA and design proper classes for the report generation system using OOD.
2. Implement appropriate class attributes and methods to support the functionalities (e.g., user inputs, report generation, File I/O, unittest, etc) of the report generation system.
3. Create a UML class diagram representing the object-oriented design of the Python program.

2.2 Program Implementation

The general steps for this project are as follows:

1. Implement classes, attributes, and methods based on the UML class diagram.
2. Implement methods to collect user requirements through user inputs, e.g., project rating level, score, location, etc.
3. The Python program should generate a report that includes all projects' information satisfying the user inputs.
- ✗ 4. The generated report should be text outputs. The Python program should enable users to import and export the report using File I/O for printing, reuse, and documentation purposes.
5. The Python program should be able to handle unexpected and inappropriate user inputs using the Python exception handling mechanism, so as to ensure the program's stability.

2.3 Testing and Analysis

1. Design comprehensive program tests using the Python unittest module, to verify the correctness of the implemented functionality. Each class and important static and non-static methods should be tested using the unittest module.
2. Analyze the test results using the code coverage module, and generate the coverage report for both Python code and tests.

3. Tasks

This assignment consists of three tasks:

1. Students must examine the Green Star projects on the Green Star project directory (<https://www.gbca.org.au/project-directory/>) using OOA and complete the class design using OOD and UML class diagrams.
2. Students must implement Python code based on their design and evaluate their programs with multiple projects. The Python program should be capable of collecting user inputs, generating text reports for Green Star projects based on user inputs, and handling unexpected user inputs. Unittest is required to test the correctness of the Python program.
3. Students need to describe their object-oriented design, including class design, attributes and methods, exception handling mechanism, unittest, and coverage analysis, in a PDF report. Also, the report is expected to include screenshots of the running results to verify the correctness of functionalities.

4. Submission

Both the report and source code must be submitted to Moodle Assignment 2. Email submissions will NOT be accepted.

4.1 Report Submission (3 marks)

1. Provide a PDF file (named report_generation.pdf) describing the program's functionalities, design decisions, and the use of the exception handling mechanism.
2. A UML class diagram must be included to show the object-oriented design.
3. The report must provide clear instructions on running the program, and it must include screenshots of the running results, to verify the correctness of the functionalities.
4. The report must explain the unittest and code coverage in detail.

4.2 Source Code Submission (7 marks)

1. Python program: Submit the complete Python program (named A2.py) implementing the report generation system for Green Star.
2. Test cases and Analysis: Submit the source code of unittest, along with the code coverage report.
2. Demonstrate the solution to the lab demonstrator in Week 10's lab.

5. Appendix

- A. The Australian Sustainable Development Goals (SDGs) website: <https://sdgs.org.au>
- B. Green Star website: <https://new.gbca.org.au>
- C. Green Star Project Directory: <https://www.gbca.org.au/project-directory/>