Afshan Parkar

SWE 321 – Final Project - Fall 2021

Project Objective and Guidelines:
This project integrates what the student has learnt through the semester to analyze a given system requirement using Object Oriented (OO) techniques, design using UML class diagram, extract class relationships, and implement the system in Python.

Project submission guidelines:

1. The project team will comprise of a maximum of 3 students.
2. Submit final Python code (SWE 321), UML and other documents to Black Board within the given deadline.
3. Weight: 30% of total course grade.

Project Description:
This assessment is a group assessment. The students will be working in a group of **two-three** to complete the task.
The student group should choose one of the real-life business cases, analyze the problem and develop a Python application for the same.
The main aim of this assessment is to test the student's skill on analyzing the chosen business case and develop a GUI design for the application that covers all the requirements.

The assessment carries 30% for the final grade. The 30% is split as 5% for individual work and 25% for group work. The individual work will be assessed through online presentation of the project.

Part 1: UML Report

1. Write a well-defined paragraph on the business case chosen, stating the various requirements.
2. Identify the associated classes and list 5- 7 classes for the chosen business case.
3. Depict various associations between the classes (It should have inheritance, aggregation, composition, binary association)
4. Construct a UML class diagram using a Case tool (Lucidchart.com or Umlet)

Part 2: Implementation of the system.

1. Accept all the inputs and display the outputs.
2. Implement the code to complete the process requirements.
3. All **inputs should be validated by testing the code using the try-except clause.**
4. Validation should include: not empty, data type validation, data range, no selection, etc. Use masked text box if applicable.
5. Give proper error messages, in case there is some error.
6. Implement a **File Manager** class which manages various text files, where it reads and stores data into various text files. It also has the main module (test code) of the system.
7. The code must be well organized and documented. The use of good documentation, proper naming convention for files, classes, and variables is essential to integrate the modules of the project.
8. It is important to make sure that each team implements all the requirements agreed in the business case report. (Check with your instructor what needs to be implemented if you have any difficulty in identifying the process).

9. The output should display complete set of information.

Part 3: GUI Interface Design.

1. Create a Python window form application and give proper name to the form/application.
2. Using proper controls, design the form/window for the application.
3. Customize the form/window color, background, font etc.,
4. Give proper names to the controls (Labels,Textbox etc.)
5. The window/form design should be clear and complete.
6. Your application should have at least four windows/forms with good navigation.
7. Make sure that your application has consistent design, e.g. colors, fonts, font size, images used, forms location, etc.
8. Customize the window/form properties and give proper title to the windows forms to make the application look & feel more professional.
9. Customize the buttons with proper names.
10. Have a proper reset button on each form and close button for the application.

Part 4: Student Group Presentation- Individual part/Question answer

Students will be displaying the project in a group, online during the Presentation week. The team members have to prepare themselves to explain clearly about their project, how they managed to complete such a challenging task by following learning by doing methodology.

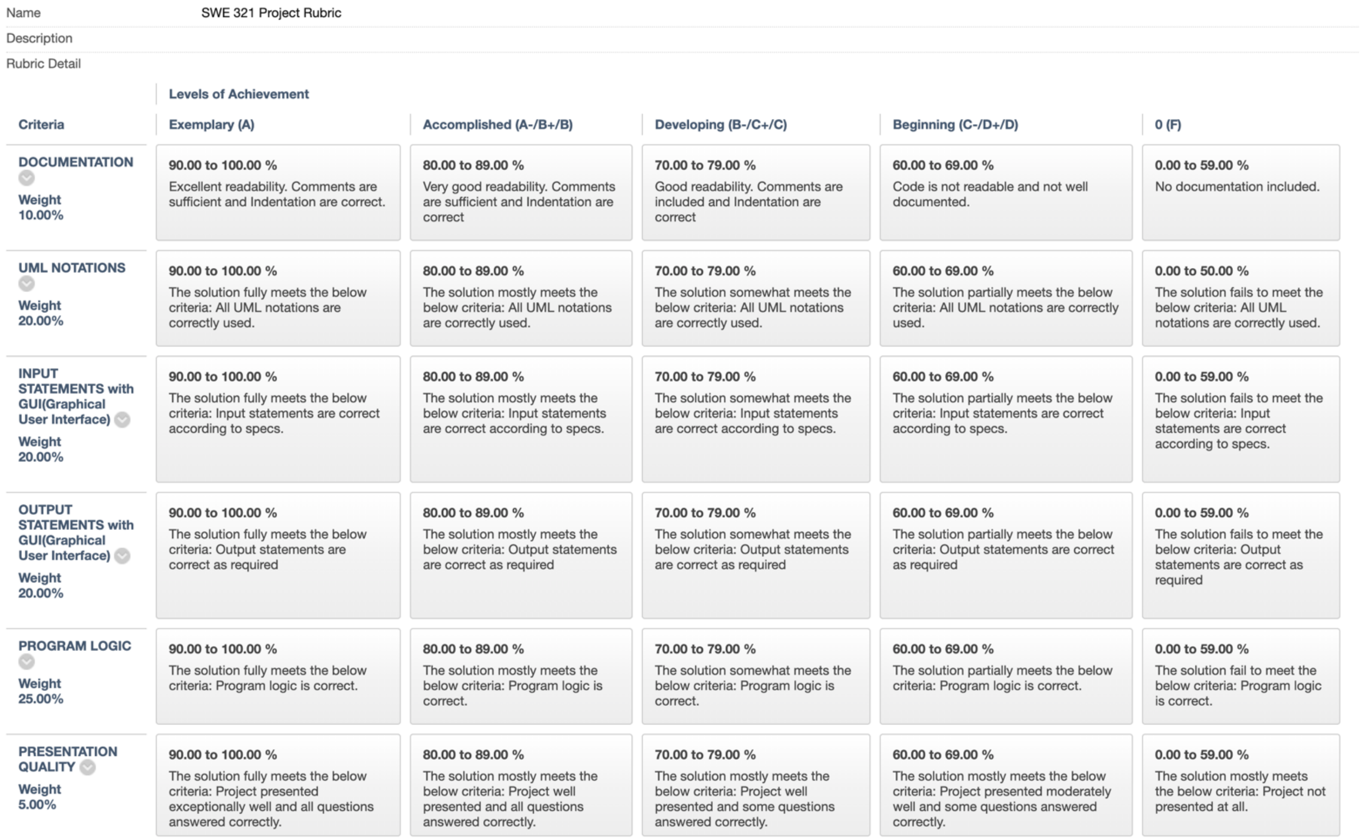
The team should explain clearly the functionality of the project and answer all the questions raised.

Final Submission:

As part of the course completion requirement of SWE 321, the above completed Python project is submitted on Black Board with all the required files in a compressed format.

The overall marking scheme rubric for the Python code is given below, which will be used to evaluate a fully functional error free code.

Marking Scheme:



Name SWE 321 Project Rubric

Description

Rubric Detail

	Levels of Achievement				
Criteria	Exemplary (A)	Accomplished (A-/B+/B)	Developing (B-/C+/C)	Beginning (C-/D+/D)	0 (F)
DOCUMENTATION Weight 10.00%	90.00 to 100.00 % Excellent readability. Comments are sufficient and Indentation are correct.	80.00 to 89.00 % Very good readability. Comments are sufficient and Indentation are correct	70.00 to 79.00 % Good readability. Comments are included and Indentation are correct	60.00 to 69.00 % Code is not readable and not well documented.	0.00 to 59.00 % No documentation included.
UML NOTATIONS Weight 20.00%	90.00 to 100.00 % The solution fully meets the below criteria: All UML notations are correctly used.	80.00 to 89.00 % The solution mostly meets the below criteria: All UML notations are correctly used.	70.00 to 79.00 % The solution somewhat meets the below criteria: All UML notations are correctly used.	60.00 to 69.00 % The solution partially meets the below criteria: All UML notations are correctly used.	0.00 to 50.00 % The solution fails to meet the below criteria: All UML notations are correctly used.
INPUT STATEMENTS with GUI(Graphical User Interface) Weight 20.00%	90.00 to 100.00 % The solution fully meets the below criteria: Input statements are correct according to specs.	80.00 to 89.00 % The solution mostly meets the below criteria: Input statements are correct according to specs.	70.00 to 79.00 % The solution somewhat meets the below criteria: Input statements are correct according to specs.	60.00 to 69.00 % The solution partially meets the below criteria: Input statements are correct according to specs.	0.00 to 59.00 % The solution fails to meet the below criteria: Input statements are correct according to specs.
OUTPUT STATEMENTS with GUI(Graphical User Interface) Weight 20.00%	90.00 to 100.00 % The solution fully meets the below criteria: Output statements are correct as required	80.00 to 89.00 % The solution mostly meets the below criteria: Output statements are correct as required	70.00 to 79.00 % The solution somewhat meets the below criteria: Output statements are correct as required	60.00 to 69.00 % The solution partially meets the below criteria: Output statements are correct as required	0.00 to 59.00 % The solution fails to meet the below criteria: Output statements are correct as required
PROGRAM LOGIC Weight 25.00%	90.00 to 100.00 % The solution fully meets the below criteria: Program logic is correct.	80.00 to 89.00 % The solution mostly meets the below criteria: Program logic is correct.	70.00 to 79.00 % The solution somewhat meets the below criteria: Program logic is correct.	60.00 to 69.00 % The solution partially meets the below criteria: Program logic is correct.	0.00 to 59.00 % The solution fail to meet the below criteria: Program logic is correct.
PRESENTATION QUALITY Weight 5.00%	90.00 to 100.00 % The solution fully meets the below criteria: Project presented exceptionally well and all questions answered correctly.	80.00 to 89.00 % The solution mostly meets the below criteria: Project well presented and all questions answered correctly.	70.00 to 79.00 % The solution mostly meets the below criteria: Project well presented and some questions answered correctly.	60.00 to 69.00 % The solution mostly meets the below criteria: Project presented moderately well and some questions answered correctly.	0.00 to 59.00 % The solution mostly meets the below criteria: Project not presented at all.