**Course Learning Outcomes**

Upon completion of this assignment, you should be able to:

|  |  |
| --- | --- |
| CLO2 | Construct a programmable solution using appropriate problem-solving methods and programming concepts to the given scenario. (C3, PLO2) |

# 1.0 Group Assignment Description

Work with your group to develop a program based on the following guidelines:

1. You are required to employ a Python language only to create a program based on the case studies below.
2. You are advised to use your own logic flow to create the program.
3. You are required to form a team (in a group of **THREE (3)** to **FOUR (4)** persons)

Your group is required to choose the case study **either Case Study A or Case Study B.**

**Case Study A: Test Questions Management System**

A **Test Questions Management System (TQMS)** is a tool that can help in creating questions for test and exams, or assessments based on a given topic. A test questions management system (TQMS) can save time and effort by generating questions that are relevant, challenging, and tailored to test needs.

1) Academic admin

* Login – Admin should be able to login using username and password. Only three attempts are allowed otherwise the user will be terminated from the apps.
* Assign new lecturer and test/exam unit personnel with username and password
* Add lecturer profile (Nama, address, contact number etc)
* Modify/update/edit lecturer profile (dob, address, email address, age, citizenship, ID, contact number etc)
* Add subjects and topics. (Minimum 3 subjects with 3 topics needed)
* Delete existing lecturer and test/exam unit personnel

2) Registered lecturer

* Login - Lecturer should be able to login using username and password. Only three attempts are allowed.
* Change username and password
* Add new questions and answers based on the subject and topic.
* Modify/update/edit questions and answers
* View questions and answers based on subject and topic.

3) Exam unit personnel

* Login – Exam unit personnel should be able to login using username and password. Only three attempts are allowed.
* Change username and password
* Create two sets of exam papers: Set 1 and Set 2
* Add questions into Set 1 and Set 2 exam papers. (Each exam paper set consists of two sections, Section A have minimum 5 multiply choice questions, and Section B have 3 subjective questions)
* Modify/update/edit exam papers
* View Set 1 and Set 2 exam papers

**Case Study B: Clinic Management System**

**Clinic Management System** (CMS) covers the complete work flow a GP (General Practitioner). With this great tool, clinics can drastically reduce the administration and management hassle and dedicate their time to providing quality healthcare services. You are required to develop a Clinic Management Software (CMS) for Oceana Clinic.

**Patient Registration - Receptionist**

* Receptionist login – The receptionist should key in username and password.
* Register new patient information (identification number (ID) / passport number, name, address, birth of date, current age, contact number, nationality, gender)
* Register patient’s family members information – spouse, and dependencies (name, birth of date, gender, current age)
* Add in patient’s medication history (kidney disease, heart problem, high blood pressure, diabetes and other chronic diseases)
* Edit / modify / update existing any relevant patient information.
* View specific patient’s information, medical history, and family members information
* Delete / terminate patients that are not visiting the clinic more than one year after the last visit
* View patient medical certificate (MC)
* Patient search and add into waiting queue
* Clear indication patient status in the queue (waiting, dispensary, cashier, complete)

**Doctor**

* Login
* Search patient by ID / passport number
* View patients’ visit and medication history
* View previous drugs prescription
* Add new medical result (blood pressure, sickness, drug prescription, temperature, additional information etc)
* Add and view new medical certification (MC)

**Pharmacy / dispensary**

* Pharmacist login – Key in username and password only. Three attempts are allowed.
* Register new drug categories (general drug, vaccine, poisonous, psychotropic, others)
* Register drug information (date, quantity, supplier name, cost per unit, total cost, expiry dates)
* Update / edit / modify drugs information, cost price & selling price, default minimum, maximum and reorder level upon receiving from suppliers
* Delete drugs no longer being supplied
* Expires warning
* View stock balance based on drug categories
* View reports for drug which fall into reorder level

**Cashier**

* Cashier login – Key in username and password only. Three attempts are allowed.
* Add comprehensive data entry and cash collection
* Accept cash, cheque, credit card, debit card and payment mode from patient
* On demand receipt printing
* End of the day cash collection

***IMPORTANT INSTRUCTIONS:***

*This is a group assignment. Upon submission of your assignment, you are required to present your assignment at a date and time specified by your module lecturer.*

*Each team member is required to contribute towards some features in the system & documentation, present and explain his or her work accordingly. Each team member should be able to answer questions during presentation with regards to the overall systems project or specific question(s) related to the codes used in the development.*

# 2.0 Requirements

1. You are required to carry out research relevant to the existing program similar to your program going to be developed and document any logical assumptions in terms of functionalities can be embedded to your program.
2. Your program should use constants where appropriate. Validations need to be included to ensure the accuracy of the program. State any assumptions that you make under each function.

1. You are required to store all data in text files. There is no limit on the number of text files that can be used but they should be kept minimum.

1. You are expected to use list and functions in your program. Your program must embrace modular programming technique and should be menu driven. Your program also should provide a termination function.

1. You may include any extra features which you may feel relevant and that add value to the system.

1. There should be no need for graphics in your program, as what is being assessed, is your programming skill not the interface design. The marking scheme for the assignment has been provided so that you clearly know how the assessment for this assignment would be done.

1. You should include the good programming practice such as comments, variable naming conventions and indentation**.**

1. In a situation where a student:
   * ***Failed to attempt the assignment demonstration, overall marks awarded for the assignment will be adjusted to 50% of the overall existing marks.***
   * ***Found to be involved plagiarism, the offence and will be dealt in accordance to APU regulations on plagiarism.***

1. You are required to use Python programming language to implement the solution. Use of any other language like C/C++/Java is **not allowed**.

# 3.0 Deliverables

You are required to submit a softcopy of:

1. Program coded in Python – submitted as .py file.
2. Text files created through testing data – submitted as .txt files.
3. A documentation of the system – submitted as pdf file - that incorporates basic documentation standards such as header and footer, page numbering and includes:
   * Cover page – include all group member names & TP numbers
   * Table of contents
   * Introduction and assumptions of your system developed.
   * Design of the whole program by using **flowcharts**.
   * **Explanation of programming concepts** applied with **sample of source code** based-on the system developed.
     + Application of storage types (list, tuple, dictionary, text files)
     + Application of control structure
       - Selective: if.. else, nested if
       - Iteration: for loop, while loop
     + Application of Try … Except
     + Application of validation
     + The flow of add function
     + The flow of update/modify/edit function
     + The flow of delete function
     + The flow of generating reports
   * Additional features source code with explanation (if any)
   * Screenshots of sample input/output with explanation
   * Conclusion
     + The advantages of your system developed
     + Any limitation discovered
     + Suggestion to overcome the limitation for further enhancement
   * References using APA referencing format
   * Appendix
     + Workload Matrix
     + Screen shot of every text file contents
4. Only the submission through Moodle will be graded.
5. Submission due date:  **(6th August 2024 (Tuesday) before 11:59 p.m.)**

# 4.0 Performance Criteria

## Distinction (80% and above)

This grade will be assigned to work which meets all the requirements stated in the question. The program runs smoothly when executed. There is clear evidence and application of Python concepts up to advanced level. The program solution is unique with excellent coding styles and validation. The program implemented maps completely against the design (pseudocode and flowchart) as seen in the documentation. The design of the solution varies in styles and has unique logic with hardly any errors / omissions. The documentation does not have any missing components. Sample inputs/outputs documented have clear explanation. Student must be able to provide excellent explanation of the codes and work done, show additional concepts / new ideas used in the solution, able to answer all questions posed with accurate / logical answers / explanation provided with sound arguments and clear discussion. The overall an excellent piece of work submitted.

## Credit (65%-74%)

This grade will be assigned to work which is considered to be a good standard and meets most of the requirements stated in the question. The program runs smoothly when executed. There is clear evidence and application of Python concepts up to at least intermediate level. The program solution is unique with good coding styles and validation. The program implemented maps well against the design (pseudocode and flowchart) as seen in the documentation. The design of the solution varies in styles and has unique logic with minor errors / omissions. The documentation does not have any missing components. Sample inputs/outputs documented with some explanation. Student must be able to provide good explanation of the codes and work done, answer most questions posed with mostly accurate / logical answers / explanation.

The overall a good assignment submitted.

## Pass (50%-64%)

This grade will be assigned to work which meets at least half of the basic requirements (approximately 50%) stated in the questions. The program runs smoothly when executed. There is clear evidence and application of Python concepts at basic level. The program solution is common with basic coding styles and validation. The program implemented somewhat maps with the design (pseudocode and flowchart) as seen in the documentation. The design of the solution is average in terms of logic and style with some errors / omissions. The documentation has some missing components. Sample inputs/outputs documented but without any explanation. Student must be able to explain some codes and work done and able to answer some questions posed with some accurate / logical answers / explanation. The overall an average piece of work submitted.

## Fail (Below 50%)

This grade will be assigned to work which achieved less than half of the requirements stated in the question. The program is able to compile but not able to execute or with major errors. The program solution has only basic coding styles with no validation. The program solution has little or no mapping with the design. The design of the solution has major / obvious errors / omissions. The documentation has some missing essential components. Student is barely able to explain the codes / work done and answer given on the questions posed but with mostly inaccurate / illogical answers / explanation. The overall poor piece of work submitted.