COMP0066 Introductory Programming Coursework 2024/25

The coursework will assess your ability to demonstrate your programming knowledge in python. You will work in groups on a programming project.

Breeze - A Mental Health Management System

The purpose of Breeze is to provide a mental health communication, support and management platform for both Mental health and wellbeing practitioners (MHWPs) and patients. It also provides essential tools and resources to improve patients' well-being.

Task

The NHS has appointed several developers (your group) to implement a mental health application. A list of required features is described below but it is encouraged to add novel useful features.

Features

The system accepts 3 x types of users. The application administrator, the Mental health and wellbeing practitioner (MHWP) and the patient. Consider that all users have been registered already. All user types should be able to login/logout using a username and password pair (see note 1 below).

(a) The admin:

a. Allocates registered patients to registered MHWPs. One MHWP per patient.
b. Edit MHWPs and patients' information.
c. Delete a user (MHWP or patient) record.
d. Disable a user (MHWP or patient) record. Once disabled, the user is still in the system, can login/logout but no change can be made. All features are disabled.
e. Display a summary of all related details, including, patients, MHWPs, their allocations, Number of confirmed bookings per MHWP for the current week, etc.

(b) **The patient**:

a. Edits their own personal information. This includes name, email, and emergency contact email.
b. Describe their mood of the day using a colour code and add comments. Refer to Figure 1.
c. Enter their journaling text. Each entry should be saved with latest date/time.
d. Access meditation and relaxation exercises based on a keyword search. You can search and fetch results from existing resources such as https://www.freemindfulness.org/download or https://insighttimer.com/guided-meditations. You can either point to the URLs of the results (audios or videos) or embed the results into your application for the user to play directly.
e. Book/cancel an appointment with their MHWP. The selection should be done on the MHWP's calendar (either text-based if you are using a command line application or on a calendar widget if using a graphical user interface). The booking needs to be confirmed by the MHWP.

f. Receive an email for every booking confirmation by the MHWP or cancellation (either done by the patient or the MHWP).

(c) The MHWP:

a. Displays his/her calendar with requested and confirmed appointments.
b. Confirm or cancel an appointment made by a patient (email sent to both his/her and the patient email addresses).
c. Add information to a patient's record, such as condition, notes, etc
d. To make things easier, you can let the MHWP select a set of mental conditions from a predefined list such as: anxiety, autism, etc.
e. Display on a dashboard a summary of all his/her patients data and a chart per patient with their mood tracking information.

Note 1: to simplify the testing process and assessment of your application, make sure the usernames are *patient1*, *patient2*, *mhwp1*, *mhwp2*, etc, and all passwords set to the empty string.

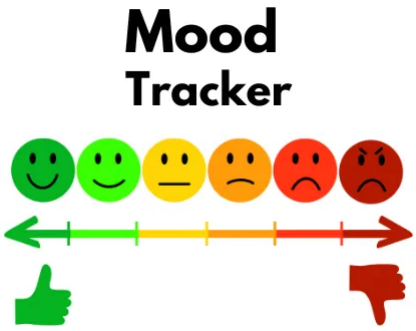

Figure 1: Mood tracker colour code (Source: Medium.com)

Persistence

All information edited in the system, should be persistent across sessions; meaning that if you make a change and close your application, the last changes should be shown when you restart your application again.

Data Integrity

All information entered into the system, should be realistic. i.e. the system should not accept made up mental conditions, impossible dates, infeasible names. Your system should prompt the user to enter information which is correct.

Implementation

You can use object-oriented programming if you wish. It's preferable but not mandatory.

The management system should be implemented as a command-line application, but you can have a GUI -based application instead. Both options are ok.
It is possible to use third-party libraries as long as the source is mentioned.

It is allowed to use of generative AI for code snippets generation and code debugging.

The aim of the exercise is to practice your knowledge in core python. The use of ready-made frameworks such as ***Django, Flask, CherryPy or equivalent, is not allowed***.

Deliverables

The grade for your COMP0066 coursework will depend on the quality and correctness of your programming implementation, but also on the peer-evaluation of your group mates. You are required to submit two deliverables:

Deliverable (a) - one submission per group

1. Use Moodle link to submit your single .zip file containing all your code source files (.py files).

2. The **link** to your UCL MediaCentral (https://mediacentral.ucl.ac.uk/) video.
   The video should contain 2 x parts.
   (a) The first part showing in detail how to install your application.
   (b) The second part demonstrating all the features of your application "not the code". You can add voice-over or text comments.

   Use video submission link on Moodle to submit a .txt file with the link to your video. (One submission per group)

   Note: The video is not marked but used to promote your application to the examiners.

3. Name your zip file *groupNN.zip*, where *NN* is the group number. For example, if your group number is 7, your file should be named group07.zip

Notes: Before submission, make sure the maximum size of your zip file doesn't exceed 50MB.

Warning: Email submissions are equivalent to a non-submission. It's the responsibility of the group members to ensure submissions are made on Moodle by a member of their group, well before the deadline, to avoid last minute technical glitches.

Deliverable (b) - **individual submission**

4. A completed online IPAC form that will be available on Moodle when groups are finalised. The form takes 10 minutes to complete.

Note: The IPAC form is mandatory and should be completed by the deadline on Moodle. No IPAC form, no mark!

Marking scheme

You will be assessed clearly on the following, which must be shown in a useful context.

1. Implemented a reasonably complete application.
2. Showed that you can put in practice what we have covered in lectures and labs.
3. Make sure your code is robust enough by testing it before submission, as you may lose marks if your software application raises errors or behaves strangely.
4. The submitted code should be self-contained. It should install and work on any machine with core python, without any extra configuration.
5. A detailed coursework marking criteria is attached at the end to the current document. Your final mark is devised by considering your individual contribution to the project (using IPAC scores).

Group work

It is highly recommended that you select a project leader that will organise the group and supervise the allocation of tasks and progress of the implementation. All members of the group should contribute to the implementation of the project. Poor engagement with the project will decrease your individual mark.

Plagiarism

UCL enforces a firm policy regarding plagiarised content. Any attempt to include existing code (from former students work or other sources) into your application "could be penalised for Academic Misconduct, which is defined as any action or attempted action that may result in you obtaining an unfair academic advantage."
Refer to UCL academic integrity page.

Common questions and their answers

Q: Can I implement the application as a GUI instead of a command line application?
A: Yes, please see section 'Implementation above'

Q: Should the implementation follow an object-oriented approach?
A: Not necessarily, see section 'implementation above'.

Q: Can I include third party libraries?
A: You can use any third-party open-source python library **if you can't avoid it**. Please limit their use to situations where you really can't just use core python, NumPy or Pandas. Also, you should make sure that your application is self-contained, so the installation is automatic on the assessor's machine. This means that you should find a way to include or automatically fetch the libraries when your application is being installed.

Q: What is the configuration of the assessor's machine
A: Core Python (used in lab sessions), NumPy and Pandas only.

Q: Can we use databases to implement persistence?
A: Yes, as long as the installation of your application on any other machine doesn't require extra configuration apart from core python, NumPy and Pandas.

Extenuating circumstances & late submissions

Please check out your academic manual available here.

UCL Computer Science: COMP0066 Programming project marking criteria and grade descriptors.

Fail		Pass (2:2)		Merit (2:1)		Distinction (1st)		
Inadequate	Weak	Satisfactory		Good		Excellent	Outstanding	Exceptional
Below 40: Fail	40-49: Fail	50-54: Low pass	55-59: High pass	60-64: Low merit	65-69: High merit	70-79	80-89	90+
Either no solution or solution provided is inappropriate and irrelevant.	Rudimentary coding, significant omissions in list of implemented requirements.	A reasonable attempt at providing a software solution with limited features and code containing bugs limiting the use of the solution.	A sound solution with a reasonable list of implemented required features. Code containing several bugs and requiring improvements.	The solution implements all or the majority of required features, contains few bugs and is subject to some improvements and fixes.	The solution implements all required features, contains no or minor bugs and is subject to minor improvements.	The solution implements major extra features related to application domain in addition to the required ones. Research has been done and the resulting software is of the quality of commercial applications, competing in terms of features and robustness.	The solution is innovative, provides a major addition to the application domain. The solution can be published in a conference or journal.	The solution is exceptional in terms of algorithms, performance, and features. The solution provides a major contribution to the domain of software development.