SOFTWARE QUALITY

FINAL ASSIGNMENT

To make it feasible as an assessment for this course, the following scenario is formulated to ensure that students have achieved at least the minimum level of the course learning outcomes 1, 2, and 4, as defined in the course manual. Please note that this scenario might be very different in real world cases, which usually need other quality requirements. Normally such a system would involve many other requirements and components, but here you can limit yourself to only the given description.

Assignment Objectives

The learning objectives of the assignment and mapping to the intended learning outcome of the course are listed below:

1. To apply the knowledge of input validation for both user-generated and server-generated data (LO1, LO4).
2. To experience the common mistakes of coders in input validation (LO2, LO3).
3. To partially build a secure input validator by coding practice (LO4).

Scenario

This assignment consists of the design and implementation of a simple console-based interface in **Python 3** to store, retrieve, and modify information of clients of a house construction company in Netherlands. The system should employ a local data file to store the information of clients using a simple encryption technique of your choice.
Users are the employees of the company, which can be categorized as below:

1. Super Administrator (Hardcoded)
2. System Administrators (to be defined by the Super Administrator only)
3. Advisors (to be defined by the System Administrator only)

All Usernames and Passwords must follow the rules given below:

- Username:
  - must have a length of at least 5 characters
  - must be no longer than 20 characters
  - must be started with a letter
  - can contain letters (a-z), numbers (0-9), dashes (-), underscores (_), apostrophes ('), and periods (.)
  - no distinguish between lowercase or uppercase letters
- Password:
  - must have a length of at least 8 characters
  - must be no longer than 30 characters
  - can contain letters (a-z), (A-Z), numbers (0-9), Special characters such as ~!@#$%^&*_-+=`|\(){}[]:;'<>,.?/.
  - must have a combination of at least one lowercase letter, one uppercase letter, one digit, and one special character

When new clients request for a service, they should be registered in the system, first. A new user can be registered in the system by a system administrator only.
For a client, the following data are entered to the system:

- Full Name
- Address:
  - Street and House number
  - Zip Code (DDDDXX)
  - City (system should generate a list of 10 city names of your choice predefined in the system)
- Email Address
- Mobile Phone (+31-6-DDDD-DDDD)

The format of address and mobile phone should be according to the standard format of Netherlands.

Assignment Requirement

The delivered code must have the minimum requirements specified below:

1. Distinguish between different categories of users and their access level, with a proper authentication.
2. Input Validation: All User's inputs must be properly validated for the expected format.
3. Handling Invalid Input.

To evaluate the requirements of the assignment, the following criteria are made. The details of grading are provided in the Grading Table and Marking Scheme.

C1 Authentication for users are properly implemented.

- System must authenticate a user.
- Usernames and passwords can be stored in a local file, using a simple encryption method of your choice, such as Caesar sipher. The only security measure on username and password file is that the file should not be readable in text mode by a text editor. You do not need to implement complex mechanism, but you are free to choose your own option. As long as, the file is not readable by a text editor, the criterion is assessed as satisfactory.

C2 Users access level are implemented.

- Distinguish between different categories of users and their access level, as a result of authentication process.
- By this we mean for example, a user with advisor level should not be able to see the menu option for adding a new advisor user.

C3 All inputs are properly validated.

- All inputs, including both use-generated (e.g. client name, email, etc.) inputs and system-generated inputs (e.g. city) must be validated.

C4 Invalid inputs are properly handled.

- In case of invalid input by user, the system must take appropriate action. For example, it might only display a proper message to the user, or might ban the user for extra attempts, depends on the number of invalid inputs for a specific field.

C5 Suspicious activities are logged.

- In case of suspicious activities or realizing an attack; for example, a user is attacking the system by trying many passwords (brute force), or an open session is used by a stranger and entering suspicious characters in irrelevant fields of data (e.g. entering '/' in user name multiple times); then the system need to take proper action, and log the activity in a file. This file must be available only to Super Administrator or System Administrator in menu options.

Important Note: For development of your code, you can use built-in modules or standard library, such as math, string, sys, etc. However, any library, module, method, code, etc. which is **not coded by you** and **implement the requirements of the assignment** is **not allowed**. In general, the goal of the assignment is on the security aspects of Input Validation, not programming. It means you need to demonstrate that you understood the concept of Input Validation and you are able to implement it in your code. Using third party **Regular Expression** library is allowed (if needed), but the RE pattern must be defined by you.

Grading

The assignment will be evaluated as either PASS or FAIL. To successfully pass the course, students must pass the assignment together with passing the exam.

Students will receive feedback from the teachers via Google Classroom, if needed.

Your assignment will be assessed according to the following marking Scheme. To successfully pass the assignment you need to meet the following assessment criteria:

- You must get C3 as Satisfactory (L2 or L3), and
- You must get a minimum of 10 points in total.

Grading Table

Criteria and Points		Unsatisfactory		Satisfactory	
		L0 (0 point)	L1 (1 point)	L2 (2 point)	L3 (3 points)
C1	Authentication for users are properly implemented.				
C2	Users access level are implemented.				
C3	All inputs are properly validated.				
C4	Invalid inputs are properly handled.				
C5	Suspicious activities are logged.				

- L0: Not implemented or Very basic attempts
- L1: Poor implementation or Major problems
- L2: Minimum requirements are implemented or Minor problems
- L3: Meet the requirements or Good implementation

Marking Scheme

Assignment will be evaluated according to the marking scheme, below.

Criteria	Unsatisfactory		Satisfactory	
	L0 (0 point)	L1 (1 point)	L2 (2 point)	L3 (3 points)
C1	Authenticating does not exist, or it is not working properly.	Authentication is based on username and passwords. PWs are longer than 8 characters and are hashed. There are some bugs of major problems.	Authentication has proper error messages. Authentication data are stored in an unreadable file by a text editor. There is no bug or major issue.	Authentication has a secure recovery mechanism. Authentication is protected against multiple wrong authentication.
C2	Authorization is not implemented or at very basic level.	Application code has hard-coded role checks. Lack of centralized access control logic. There are some bugs.	Authorization is implemented based on user roles and is centralized. No bugs or major problem.	Authentication is implemented and authorization is implemented based on user's actions.
C3	Input Validation is not implemented or at very trivial level. There are frequent bugs and errors, which let IV to be bypassed easily.	Input Validation is implemented, but not for all input types, or contains few bugs and errors. IV can be still bypassed.	Input Validation is complete for all input types, and does not allow to bypass. Whitelisting is used. There is no bug or error.	Input Validation is fully implemented and there are sign of following good practices in IV, such as checking for NULL-Byte, range and length of input, Validation Functions, etc.
C4	Invalid inputs are not handled, or very basic level, with many bugs and errors.	There are some attempts of invalid input handling, but not correctly implemented. The reaction to different types of inputs are not suitable.	Invalid inputs are properly handled, without bugs or major problems. However, there might be very few improper reactions or minor improvements needed.	Invalid inputs are very well handled, and there are evidences of following good practices in response to different types of inputs.
C5	There Logging is not implemented.	Logging is partially implemented. There some bugs.	Logging is fully implemented, and all suspicious incidents are logged. However, it could be still improved. E.g. there might be some minor mistakes in categorization of incidents.	Logging is complete, and there are evidences of good practices in logging.

Group Submission

The assignment can be completed by a student alone, or in a group of maximum two students.
In this case, the tasks should not be divided into two separate sections, each by one student, but we expect both students work together on all parts of assignment. If needed, we ask students to have a presentation for their teacher and explain their contribution.

Deliverable and Submission

Two components must be delivered:

- The implantation of the code in Python (*.py). Please do not include any bulky Python system files in the delivery.
  The code must run error-free. If needed, the code should only write to temporary storage subfolder of the current folder, on the local machine.
- A short report to briefly explain the functions of the code and evidence of the achieved level regarding the criteria. The report should not exceed 2500 words.

You can submit your final product in Google Classroom. Please do not send your assignment via email.
If working in a group, only one group member submits the assignment.

Use a standard zip software to create a *.zip file containing both component of your deliverables. Choose your Student ID as file name, e.g. “0987654.zip” for individual, “0987654_0987653.zip” for group submission.

- Please do not change the file extension. Do not choose other format for file names, such as your name, name of the course, title of assignment, etc.

Deadline

The last date allowed for submission is **Wednesday 8 July 2020**.
There will be a second occasion (re-sit) to submit in the next period. The deadline for re-submission will be announced on the Google Classroom.