## Assignment 2

### Yasmina Habchi

Comments in Red – Alexander Puttré

(Quang comments in Green).

## Waterfall:

I believe that it would be helpful to have projected dates for each phase to be completed by. That way, we can make sure the project is on track to be completed by the due date.

### 1- Requirements Definition:

Interview users/ clients and see what specific requirements and constraints are needed to successfully achieve what the client needs. Check out the number of people that will be using the software and the functionality that each user will have access to.

How should we aggregate the users’ specifications? Should we produce a design document to use as a reference for the rest of the development process?

Interviewing client is a good way to see what specific requirements we need, but how to we start off our design from what we already know

### 2- System and Software Design:

This phase is concerned with developing and designing the software, security, database as well as ensuring that the university’s logo and information are included.

User interface design goes here as well.

Appling PyQt5 is a good Gui to have.

### 3- Implementation and unit testing:

This phase brings together the previous 2 steps by combining all the decisions and features and manufacturing the website to test and see any malfunctions.

### 4- Integration and system testing:

Integrate the software into the system and testing it under various conditions such as multiple users logging in and accessing the courses. Evaluate the performance, and the time taken to complete tasks.

We should also gauge the user experience and make sure the UI is intuitive and functional.

We can also stress test our websites here. Maybe set a timer to reserve a spot in the class for a few minutes.

### 5- Operation and Maintenance:

For the last step, we should fix any issues we encountered while using the software and more specifically, interview again the users/ clients after testing it to see if all the requirements were met. The users in this method are the most important part for this process and project.

How often should we interview the users? Should we have some kind of built-in feedback functionality whereby users can report issues as they encounter them?

## Incremental:

The incremental model allows the users to test the initial implementation and develop other iterations until the final product satisfies the clients.

We start off by implementing the basic necessities to have a successful code running, in our case, we provide the students, instructors and administrators access to search courses, print their schedules. We then allow the students to add/ drop courses, the instructor to print their class lists and admin to add and remove courses to the system as well as students from a course… Step by step and with each increment, we will be adding more functionality while also paying more attention to the customer feedback on the development process.

How many increments should there be in total before the software is fully featured, and what functions should be present in which increments?

In what phases do we prioritize in each step.. front end or back end?

## Integration and Configuration:

### 1- Requirements Specification:

I already mentioned the specific requirements needed to ensure that the client is satisfied with the end result.

### 2- Component Analysis:

For the database, I would suggest starting with SQLite, it is a C-language library that implements a small, fast, full-featured, SQL database engine. As for the website, it is known that Python is the easiest language to use in this case. I would suggest starting with: PySimpleGUI (<https://www.pysimplegui.org/en/latest/>). It is a GUI, the program's front-end and it is designed to have a single user connect and interact with the GUI.

For the GUI, we should also consider alternatives like tkinter and Qt.

PyQt5 has more than 35 extensions we can use for a GUI.

### 3- Requirements Modification:

PySImpleGUI is a simple way to create a GUI and also allow us to change and evolve the code to our likings. We could start off by updating it for more than 1 person to use. Revise original specs and update it to the standard functionality. Apply a user authentication system to ensure that only registered users can access the software.

What about SQLite? Does it need to be modified in any way before it can be used for this project?

A user authorization system is a good idea too. Let’s also add in 2FA.

### 4- System Design with Reuse:

We design the whole system and integrate SQLite by storing the necessary information for the website, such as displaying the courses, add and remove, the instructors and time, schedules, allow the students to view available courses….

We incorporate whatever GUI library we choose to use in this step, too.

### 5- Development and Integration:

Develop and find errors within the code before it is handed over to the users (check for duplicate enrollments, server errors, create error messages…)

Beta testing will help us root out any bugs we or they might encounter.

### 6- System Validation:

Ensure all features work as expected and interview the users and get feedback, make improvements based on their suggestions.

How should we make sure the features work? Should we have pre-release testers who look for errors in the system? Should we write test cases and make sure the software performs as expected?