HOMEWORK WEEK 5-6

TASK 1 (Agile Techniques)

Question 1

SCRUM CEREMONIES

Product backlog refinement

In scrum, product backlog refinement is when you add details or modify items in the product backlog in order to make sure that the backlog is up to date and provides relevant information.

Sprint planning.

Sprint planning is a sprint kick-starter event where those who take part in the sprint discuss and decide on the contents of the sprint, the goals and how they can be achieved.

Daily scrum.

Daily scrum is a stand-up meeting that usually lasts for a short time (most commonly 15 minutes) and is done to inform everyone of updates and progress. Daily scrums are usually performed with the employees/participants standing up and not siting which is done so the meetings can be kept short.

Sprint review.

Sprint review happens at the end of a sprint and is when everyone taking part in the sprint gathers to discuss the completed work of the sprint and decide on future modifications and future directions etc.

Sprint retrospective

The sprint retrospective also happens at the end of a sprint and it is where the participants of the sprint discuss what went well during the sprint and what didn’t and what can be done to improve future sprints.

SCRUM ROLES

Scrum Master

A scrum master helps in the sprint planning and organizing and can be thought of as a facilitator. He oversees that everything goes smoothly and also checks that the backlog is up to date and useful.

Product Owner

The product owner is the one who owns the product in development and the one who makes decisions such as how the product should look like and its functionalities. He/she is also contributing to the product backlog and takes part during the sprints. The product owner is also the one who communicates with the stakeholders and customers.

Development Team.

The development team consists of the software engineers/programmers/developers etc., who are developing the product and are responsible for implementing the ideas using programming and various software tools.

Question 2

You are leading a development team that was given a task to create a new yoga booking system.

High level description of the system is as follows:

It has a very simple interface to accept user input (bookings) and display classes information

All bookings, appointments, schedules etc should be stored in a SQL database.

There is a ‘backend’ system that should be written in Python to handle the logic and manage the data flow.

Your team has two weeks to build a simple prototype that will be shown to the client to seek their feedback and discuss further enhancements.

TASK

Break this task into smaller stories (chunks of work) for the team to work on.

Assume that one person works on one task.

Mark tasks that can be worked on in parallel and perhaps those that need to be worked on in particular order.

Answer:

Tasks:

For creating the interface:

* Create a login page that will accept the login credentials of the users (needs to be done first)
* Ensure that the password is encrypted for protection and security (can be done in parallel)
* Accept user input for the booking after the login: date of the yoga class or booking number

Create the display classes page that will come after the login page:

* The page will show the yoga class for the specific booking that the user has inputted (can be done in parallel with the rest of the interface but after the yoga class information has been added in sql)

Creating the DB and storing the data:

* Store the login credentials of the users in one table
* For user creds table example columns: name, email, password, appointments/bookings (needs to be done after the creation of the interface since it will be needed first to gather the user information)
* Store the yoga class information in another table (task needs to be done first)
* For the yoga class: columns should look something like: class title, class date, time, location (for date and time built-in function will be used) etc. (creation of all columns can be done in parallel)

Python back-end:

* Manage the data flow and ensure that the requested data information is retrieved for the SQL DB when users input their booking (needs to be done after the interface has been created)

TASK 2

Question 1

Design a cinema booking system.

Think how you would approach the problem and what are potential ways of solving it?

You do not need to write actual code, but describe the high-level approach:

Draw a list of key requirements

Requirements:

* Accept a city input from the user and a date
* Display all the movies on the requested city and dates and provide a booking option next to the movie
* Display movie information (duration, description, genre, actors)
* Accept the bank credentials of the user, along with their email to send them the booking confirmation
* Provide seating information and ask user to choose a seat

What are your main considerations?

Considerations:

* How to display the correct cinema schedule depending on the city the user is in considering the number of cities
* Provide various payment options like PayPal, Visa, Mastercard, ApplePay etc 9will probably require an agreement/partnership with the banks)

What would be your common or biggest problems?

Problems:

* How to ensure that the seating information is up to date and that seats don’t show as vacant when they are not
* Ensure the security of the bank details inputted by the user for a safe transaction
* Keep the movie information up to date

What components or tools would you potentially use?

Tools:

* React/ Javascript for front-end
* Python for back-end
* SQL for storing the user information
* Potentially also consider a cloud storage for the movies for every different city as there might be a big amount of information
* An API for the movie information that are in cinemas right now and all the cities that they play in (for example <https://developer.movieglu.com/> or <https://developers.themoviedb.org/3/movies/get-now-playing> )