**HOMEWORK WEEK 5-6**

**TASK-1**

**Question 1**

Complete definitions for Scrum-related key terminology are provided below.

**SCRUM CEREMONIES**:

· Product backlog refinement- Product Backlog refinement is the act of adding detail, estimates, and orders to items in the Product Backlog. This is an ongoing process in which the Product Owner and the Development Team collaborate on the details of Product Backlog items. During Product Backlog refinement, items are reviewed and revised.

· Sprint planning- This ceremony helps to set up the entire team for the coming sprint, creating a smooth pathway for a successful sprint. Sprint planning requires the participation of all the scrum roles: the development team, scrum master and the product owner. The planning, of course, is prior to the sprint. It typically lasts for an hour or two.

· Daily scrum- The daily scrum is, as it says, a daily occurrence, which usually takes place each morning with the development team, scrum master, and product owner. The ceremony is short, usually 15 minutes, which is why it’s also called a standup meeting. That will make sure it doesn’t drag on.

· Sprint review- The sprint review is a meeting that takes place at the end of the sprint. The sprint review ceremony has three main objectives: Team demonstrates their finished work to the product owner and stakeholders. The team discusses any feedback and business context for this project.

· Sprint retrospective- The last scrum ceremony is called the sprint retrospective. It occurs at the end of a sprint, after the review, and is usually an hour in duration. The retrospective includes the development team, scrum master, and product owner.

**SCRUM ROLES:**

· Scrum Master- A Scrum Master is a facilitator who ensures that the Scrum team follows the processes that they agreed to follow. The Scrum master skillfully removes obstacles and distractions that may impede the team from meeting goals. This individual is the liaison between the Scrum team and people or teams outside the Scrum team.

· Product Owner- The Scrum Product Owner prioritizes work during the sprint planning meeting and motivates the team with clear goals, answering any questions. The development team decides how much work they can do, taking the items from the top of the product backlog list.

· Development Team- The development team creates products that offer a great user experience and have the right quality.

**Question 2**

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

A 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**:**

To create a yoga booking system we need front end page, back end database, and also people who should promote the page or system

Here I am dividing the required work into tasks so that my team understands and there is no confusion to work.

I have taken 3 teams i.e. TEAM ALPHA, TEAM BETA, TEAM GAMA

Team alpha does the database work by collecting classes, setting up prices, and adding payment-related setups. They will also work on collecting people’s information who wants to enroll in the yoga program.

Example database for taking personal information of customers is given below

Create database yoga;

Create table cust\_info (Name, Age, Phone number, Email, Enrolled yoga, city, class);

Insert into cust\_info ( x, 29, 7896541230, [m12bg@gmail.com](mailto:m12bg@gmail.com), yin yoga, London, afternoon);

Team beta will write the coding required for creating a booking system that also connects with our database and gives an API needed.

Team gama will work on the user interface (UI) page and create a user-friendly web page that allows customers to easily enroll for their required program.

* Now each person from the above teams will have an update of completed tasks.
* Python coding and user interface page are managed at a time and both team beta and gama will continue to work together.
* SQL database and API are managed at the same time.

Hence this way, we will manage to create our booking system on time by teamwork.

**TASK -2 (SQL)**

**Question 1**

Design a cinema booking system.

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

You don’t need to write actual code, but describe the high-level approach:

· Draw a list of key requirements

· What are your main considerations?

· What would be your common or biggest problems?

· What components or tools would you potentially use?

· You are welcome to draw a diagram (a very simple one) for the process flow to explain how it is going to work.

Answer- Diagram showing process of ticket booking

User logs into the booking system

User selects System shows User selects System shows YES

Location available movies a movie available times

User picks

a show

YES Are seats

User selects System displays Available?

available seats availability

NO

Notify customers if choose

seats are not available a different

show

NO

after selection of Is payment NO System notifies customer

seats, the system forwards successful? failure of payment and

user to the payment page releases seats after 5 minutes

YES

System books System generates Tickets confirmation

END

Ticket to user tickets sent to user

* Problems with the above cinema booking system are:
* Connectivity issues
* User login issues
* Main considerations are:
* User should be able to select among different cities to help find movies or shows in areas.
* When the user selects a movie, the system should display all different movies airing with their showtimes.
* The user can also make searches based on genre, name and release date, etc.
* The system should display seat availability and users can choose preferred seats.
* After booking a ticket, the system will allow payment within a predefined time.
* The system sends notifications of ticket booking, payments, and cancellations.
* The system must make sure that no two customers book the same seat.
* Components and tools used are:
* HTML
* CSS
* MySQL
* User interface of Cinema booking system
* JavaScript
* Java server page (JSP)