SWEN-261 - Practice Midterm - Society of Software Engineers - **KEY**

1. What is the definition of Software Engineering? What is the objective of software Engineering?
    - A systematic approach to defining requirements, design, development, deployment, and management of software systems
    - Objective: On time delivery of high quality software that satisfies and delights the customer.

2. Label each of the requirements below as N for non functional or F for functional:
    - Users should be able to turn lights on and off with the remote control.
        - Functional
    - The system should be able to scale with the current number of users up to 1,000 concurrent users.
        - Non-functional
    - Users shall be able to change the color of their background.
        - Functional
    - The software must run on both Windows 10 and macOS Big Sur.
        - Non-functional
    - The system shall allow only authorized users to rent a car.
        - Functional
    - The website should let users unsubscribe from product updates
        - Functional

3. Provide two methods that a team can use in order to resolve conflicts between team members.
    - Collaborative: examining all perspectives as a team
    - Competitive:
    - Accommodating: willingness to meet the needs of others at expense of one's own needs
    - Avoiding: Ignoring issues
    - Compromising: giving up something in order to get something else in order to reach an agreement
4. What is the purpose of Sprint Planning?
    - A sprint plan is the team's goal/commitment to finish by the end of that sprint. No stories should be added during the middle of a sprint.

5. List and describe the four phases of OpenUP?
    - Inception Phase
        - Identify requirements, scope, stakeholders
        - Domain analysis
        - Possible high-level architecture
    - Elaboration Phase

        - Address architecturally significant risks
        - Further explore requirements
    - Construction Phase
        - Designing, Implementation and Testing of system
        - Iterative
    - Transition Phase
        - Deployment of the full working system to product owner.
        - Ensure the product meets all expectations
        - May be bug fixes needed or clean up for team to continue working on
        - Reflection on teams performance

6. Define Project Stakeholder.
    - Any one who is directly impacted by the success or failure of of the project
    - Although "product owner" or "customer" ars stakeholders, answers of that nature do not actually answer the question

7. What is coupling? What is cohesion? In software, we want ___ coupling and ___ cohesion.

    Coupling is the relationship/dependencies between classes.
        - The more a class knows (is dependent on) other classes the higher the coupling.

    Cohesion is a measure of how related a class's attributes and methods are related to itself.
        - A class with high cohesion will have responsibilities that are tightly focused on (ideally) one single responsibility (SOLID Single Responsibility principle, GRASP High Cohesion).
    - We want low coupling and high cohesion

8. Match the five stages that user stories move through to the criteria that must be met for the story to be in that stage. **Note** there may be more than one criteria for any given stage

Stage	**Criteria**
A. Product Backlog	1. All acceptance criteria met and verified
B. Sprint Backlog	2. Testing Started
C. In Development	3. User Stories created

D. Ready for Testing

E. In Test

F. Done

4. Effort Estimated(Story points assigned)
5. Acceptance Criteria defined
6. Solution task completed
7. Solution Tasks defined
8. Development Started

- Product Backlog → User Stories Created
- Sprint Backlog → Effort Estimated(Story points assigned), Acceptance Criteria defined, Solution tasks defined
- In Development → Development started
- Ready for Testing → Solution task completed
- In testing → Testing started
- Done → All acceptance criteria is met and verified

9. Rewrite these poorly written user stories so that they avoid common mistakes:
    - I want to resign from my checkers game so that I can end the game early.
        - Actor is ambiguous
        - As a User, I want to resign from a game so I can end the game early
    - As an SE student, I want to go to the SSE weekly meeting.
        - No benefit given
        - As an SE student, I want to go to the SSE weekly meeting so I can stay up to date with SSE events.
    - As a programmer, I want to save my file by clicking the save button at the top of the screen so that I can save my progress.
        - Dictates the user interface
        - As a programmer, I want to be able to save my file so that I can save my progress.

10. You are creating a desktop music player that allows music listeners to create and keep track of their music with playlists. Define an epic that captures some aspect of the feature below. For that epic, define two user stories that would be part of the epic, though not necessarily defining all of the requirements of the epic.

Feature: “Music listeners should be able to create, update and remove music playlists, which are composed of one or more songs”.

Epic: User-created Playlists
Epic User Story: As a user, I want to be able to keep track of my favorite music in playlists so I can listen to my favorite songs more easily.

Story One: Creating a playlist
As a listener, I would like to be able to create a playlist with a name I choose so that I can have a place to store my songs.

Story Two: Adding songs to a playlist
As a listener, I want to be able to add songs to my created playlists so that I can use them to organize my songs.

11. What is version control and why is it important?

Version control is the process for tracking changes to any project artifacts (source code, documents, plans, …) which provides details of the changes that were made and who made them. It also provides a mechanism for retrieving previous versions of artifacts. It is important because without it the team members will have difficulty staying in sync with each other in terms of the latest artifacts

12. Which of the following should be version controlled?
    - Source Control
    - Documentation
    - Requirements

    All of them!

13. Why is it important to understand the architecture of your software early on?
    - Helps inform program design and class structure from the beginning
    - Enables more accurate cost and schedule estimation
    - Helps mitigate risk of choosing a poor architecture. Making changes to architecture is very difficult later on

14. Your software team has just finished its **fourth** sprint. Here’s a breakdown of how many user stories you completed over the last three sprints. What is your team’s velocity for the upcoming (fifth) sprint?
    - First: 50 points
    - Second: 67 points
    - Third: 58 points
    - Fourth: 55 points

    Answer: Average of the last three sprints (67+58+55)/3 ⇒ 180 / 3 = 60

15. Describe the five stages that a team goes through, from initial creation to being an effective team (according to Tuckman's Model).
    - Forming
        - The team is formed and begins to work with one another
        - Team members are mostly reserved and formal in their interactions
    - Storming
        - The team members become more assertive and alliances may form between team members.
        - There may be opposed positions
        - This is where most conflicts arise
    - Norming
        - Commitment and unity between team members begins
        - There may still be some confrontation
    - Performing
        - This is where team harmony occurs
        - Team members are at their highest performance levels
        - High trust
    - Adjourning
        - The group's task is complete, and they break up
16. The most commonly used HTTP methods represent CRUD operations. What are the names of these requests for each operation?
    - Create
        - POST
    - Read
        - GET
    - Update
        - PUT
    - Delete
        - DELETE

17. Describe the difference between a module and a component.
    - A module is a collection of components, services, directives, pipes and so on, while a component is a building block of the application with an associated template.

18. Describe the **four** types of data binding in Angular.
    - Event binding
        - This data binding type is when information flows from the view to the component when an event is triggered
        - <button (click)="doSomething()">
    - Interpolation

        - Text representing variables in components are placed in between double curly braces in the template
        - <h1>{{Some Text}}</h1>
    - Two-way data binding
        - Two-way binding is a mechanism where data flows both ways from the component to the view and back
        - <input [(ngModel)]=”someVariable” />
    - Property binding
        - Property binding is a one-way mechanism that lets you set the property of a view element
        - <div [prop]=”123”>

19. Explain the two parts of a CI/CD pipeline.
    - Continuous Integration is when developers integrate all code into a shared repository, frequently. Once code is merged, the automated build is ready to be verified and tested.
    - Continuous Delivery executes after Continuous Integration. Application is built and deployed to a pre-production environment where additional automated testing and/or Acceptance Testing is conducted before deploying to production.

Long Answer Questions:

For the rest of the questions, consider the following problem description:

PROBLEM STATEMENT OPTION #1:

The software engineering department has a collection of 11 team rooms. Traditionally there has not been a way for a student or faculty member to know how many rooms are available at any given time. The department would like a way for students and faculty to reserve these rooms for a block of time no more than 3 hours in length.

You have been contracted to create an application which meets this need. The application will be web based, and must support both mobile and desktop computing environments. Students, faculty, and system administrators will be able to log in with their RIT credentials. A student must be able to reserve a room for either themselves or for their entire team. A faculty member will be able to reserve any number of rooms for classroom use regardless of existing student reservations. Students whose reservations have been overridden will receive an email indicating which reservation has been removed. Students may not reserve a team more than thirty days in advance, however a faculty member should be able to reserve team rooms for any time in the current academic period. Since team rooms contain different equipment, the application should list what amenities each team room offers. An administrator should be able to make changes to the list of amenities available in each team room. Additionally they should be able to mark a

room as unavailable for an indefinite amount of time. In the event that users are found to be abusing the team rooms, an administrator will be able to issue a ban from the system for a configurable amount of time.

PROBLEM STATEMENT OPTION #2:

People in a neighborhood often have different tools available, which they do not always need to use. It would be nice if they had a way to share their tools with each other in a reservation based system. You have been contracted to create an application that will make this possible.

The application will be web based and must support both mobile and desktop environments. It will provide an easy way for anyone in a community to create an account and start sharing and borrowing items. Users must be able to join a community which will act as a hub that tools can be shared from. Users should be able to create a reservation in which they will take possession of the tool and then return it at the end of use. Users should be able to rate the experience they have had with other users to discourage people from damaging tools. There should be a way to ban users who abuse the system. There should be a way for users to communicate with each other through this application to set up meeting times during which tools can be exchanged. Owners of tools should be able to override a future reservation of tools they own in case they need to use it for something.

1. Identify 3 stakeholders in the problem description:

OPTION 1

- Software Engineering Department
- Faculty
- Students

OPTION 2

- Neighborhood residents
- Tool owners
- Tool borrowers

2. Identify 2 non-functional requirements from the description.

OPTION 1

- Application must be desktop compatible
- Application must be mobile compatible

OPTION 2

- Desktop and mobile platform compatible
- Web based platform

3. Write 3 user stories for the given problem. For each story, write one acceptance criterion.

OPTION 1

As a user, I want to reserve a room so that I can have a team meeting in a team room.
Given that I am a user and logged in, when I click reserve, a team room is reserved for me.

As a user I want to be able to join a community hub where I can share tools

As a user I want to be able to create reservation where I can get a tool to rent

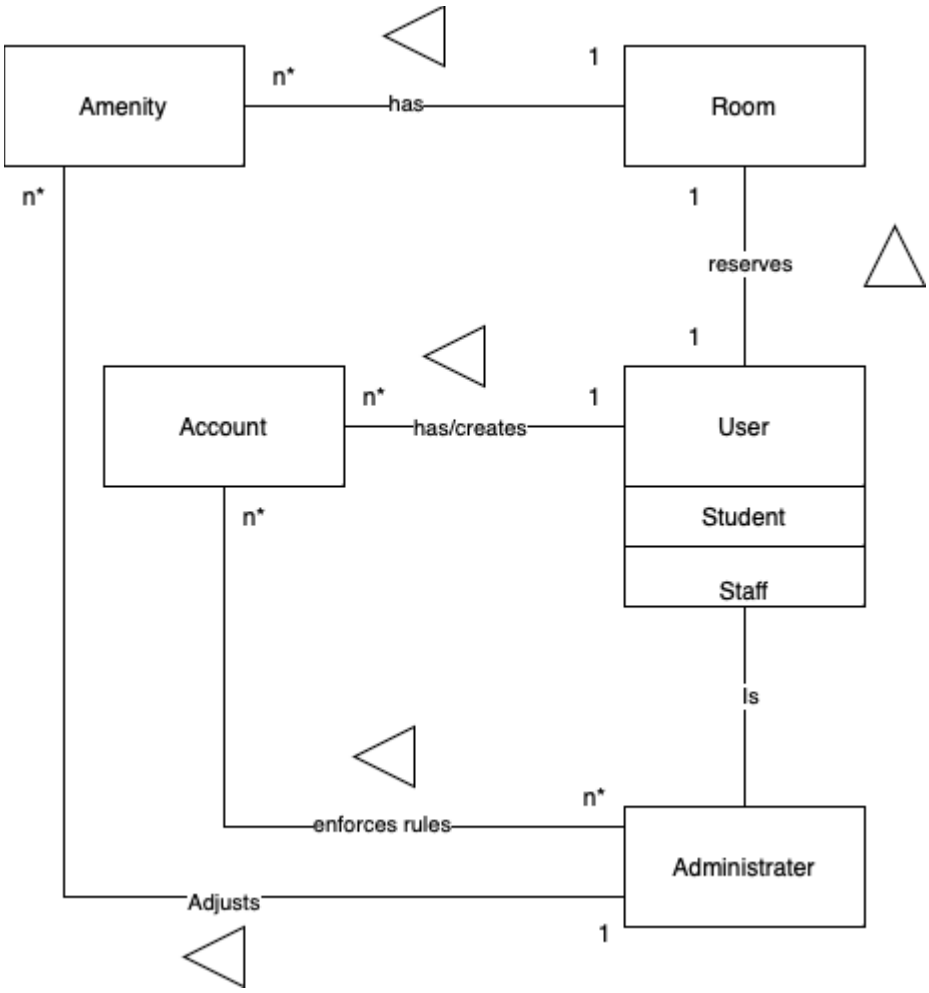
OPTION 2

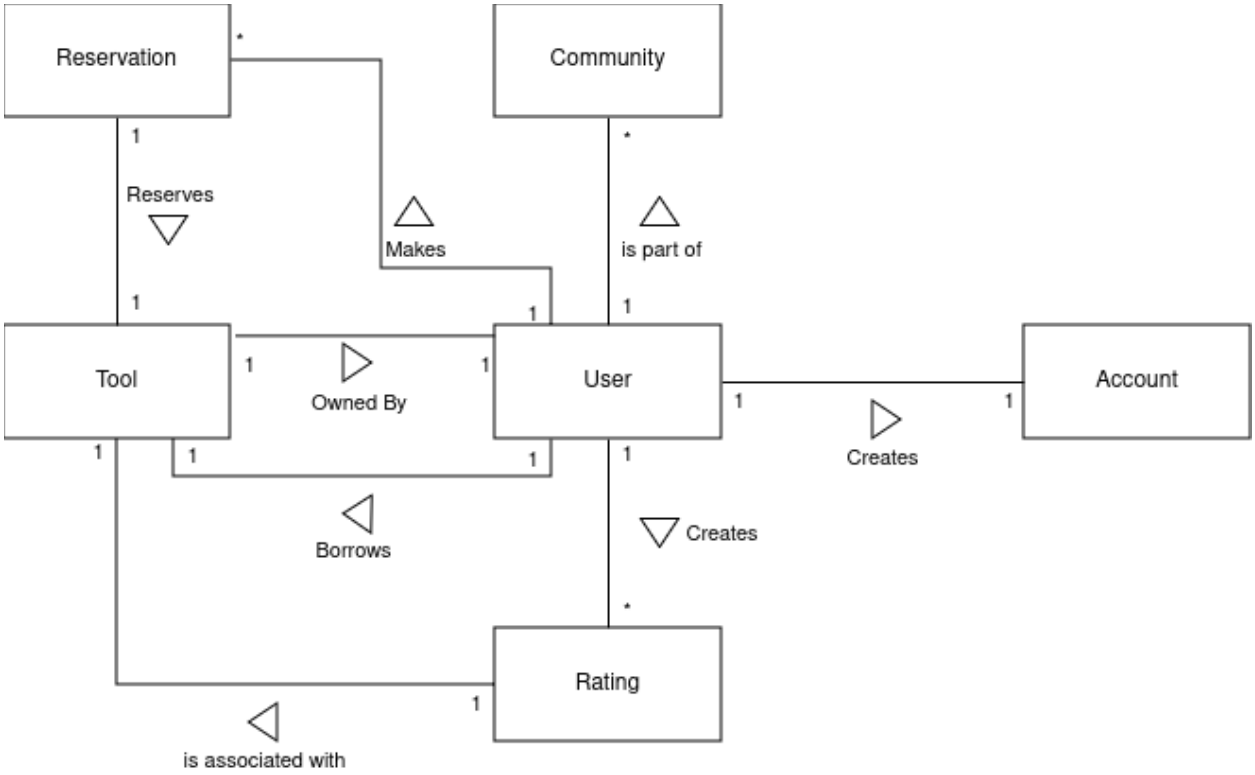
As a tool owner, I want to lend my tools so that I can help my neighbors
Given that I am a tool owner when I list my tools on the website the tools appear on the website

As a user I want the ability to return my tools once I am done using it

As a user I want to be able to rate my experience with other users based on tool quality.

4. Draw a domain model for the problem:


Amenity
n*
has
1
Room
n*
1
reserves
1
Account
n*
has/creates
1
User
Student
Staff
n*
Is
n*
enforces rules
Administrater
Adjusts
1


Reservation
Community
Reserves
Makes
is part of
Tool
Owned By
User
Account
Creates
Borrows
Creates
Rating
is associated with