

ATHLONE INSTITUTE OF TECHNOLOGY

SCHOOL OF ENGINEERING

SEMESTER 2 EXAMINATIONS 2016

Summer Session



BACHELOR OF ENGINEERING IN SOFTWARE ENGINEERING

YEAR 4

AGILE METHODOLOGIES 4.2

**External Examiner(s): Dr Norah Power
 Mr Pdraig Brennan**

Internal Examiner(s): Mr Michael P. Russell

Instructions to candidates: Read
all questions carefully.
All questions carry equal marks.
Answer **ALL** questions.

Time Allowed: 2 Hrs

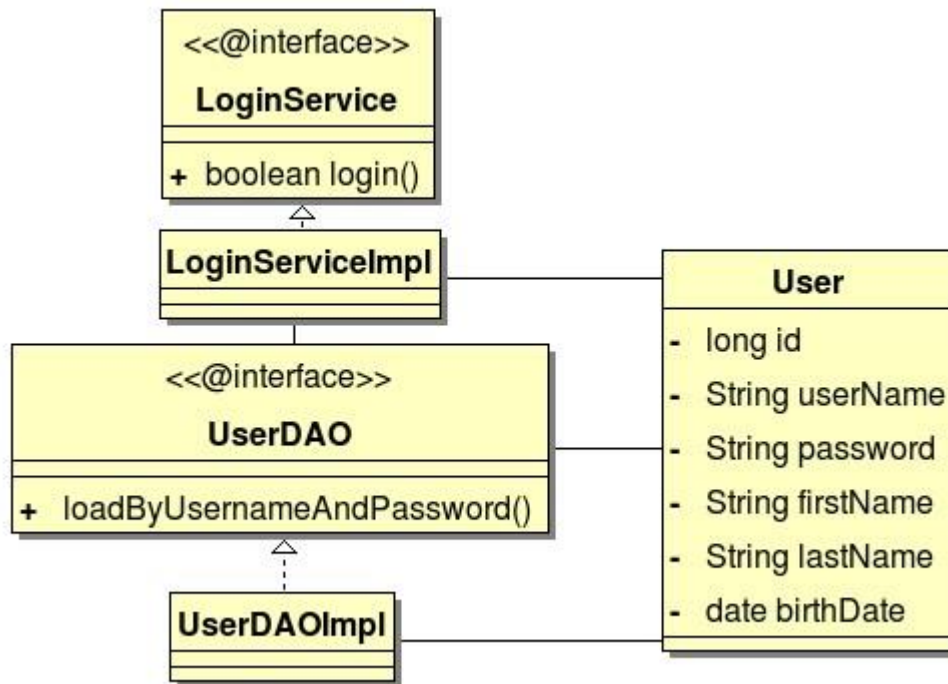
No. of pages including cover sheet: 4

Q.1

(a) Describe the Proxy Pattern. Support your answer with an appropriate UML diagram.

(2 marks)

(b) Consider the following UML diagram and associated Java Interface Specifications:



```
public interface LoginService {

    /**
     * Handles a request to login.
     *
     * If a user with the same userName and password are found,
     *
     * true is returned, else false is returned.
     *
     * @parameter userName
     * @parameter password      * @return      boolean
     */
    boolean login(String userName, String password);
}
```

```

/** Provides database access for login related functions
 */
public interface UserDao {

    /**
     * Loads a User object for the record that
     * is returned with the same userName and password.
     *
     * @parameter userName
     * @parameter password
     * @return User
     */
    User loadByUsernameAndPassword(String userName, String password);
}

```

You have been tasked with developing and unit testing the *LoginServiceImpl* class and in particular the *login()* method. You are required to employ the mock testing framework, EasyMock, to test the *LoginServiceImpl* class in isolation.

Using Java, JUnit, and EasyMock, briefly sketch a solution which details both the product and test code required to demonstrate the correct functioning of the *login()* method when it is passed both a valid username and a valid password.

(8 marks)

(c) The following *switch* statement violates both the Single Responsibility Principle and Open Closed Principle. Rewrite the *switch* statement to conform to both of these principles.

```

public Money calculatePay(Employee e) throws InvalidEmployeeType {
    switch (e.type) {
        case COMMISSIONED:
            return calculateCommissionedPay(e);
        case HOURLY:
            return calculateHourlyPay(e);
        case SALARIED:
            return calculateSalariedPay(e);
        default:
            throw new InvalidEmployeeType(e.type);
    }
}

```

(5 marks)

(d) For the following user story employ a Decision Table to identify the different combinations of inputs to tested. Reduce or simplify the Decision Table as much as possible. Clearly demonstrate the steps involved in reducing the Decision Table.

As a User I want to login using my UserName and Password so that I can access my account which may or may not be activated.

(5 marks)
[20 marks]

Q.2

(a) How is Kanban different from the other Agile methodologies, specifically Scrum?

Kanban is a methodology that we have not focused on because it is fundamentally different from the others that we have discussed.

- Most of the Agile methodologies deliver working software in a time-boxed fashion, meaning that the work proceeds in iterations, or sprints.
- Conversely, Kanban operates in a “continuous flow” model, meaning that there is no time box for development; tasks are continuously added to the backlog and removed as they are completed

Kanban has three primary characteristics: –

Visualize the workflow • A Kanban board maps the steps of the workflow in columns across a board. • Items, or tasks, are represented on cards or sticky notes and are tracked as they move through the workflow steps.

Limit work in progress (WIP) – Assign limits to each column, or workflow step, so that no one group or person can be overloaded with work. – The WIP limits naturally identify bottlenecks in the process so they can be addressed.

Measure the lead time – By understanding the workflow and removing the bottlenecks, the teams will discover the time it takes for a task to move from creation to resolution, thus providing the organization with metrics for lead time

Kanban vs Scrum • How does Kanban compare with Scrum?

- Scrum is a great fit for products/projects with a time-boxed workflow, such as product development efforts, that can progress in a series of sprints or iterations.
- On the other hand, Kanban is a better fit for an unpredictable workflow, unplanned work, and development tasks that require deployment in less time than a typical iteration.
- For example, Kanban is a great fit for helpdesk tickets and custom software support where small tickets are continually created and require speedy resolution.
- This is where Kanban is a great option because it shares all of the Agile elements that make this work so well—collaboration, clear prioritization, small increments of work, fast delivery of working software—and moves it to a continuous workflow.
- It helps to separate new product development from support of the current product.
- Teams often use white boards and sticky notes to create the Kanban board, but any visual tool can be used, from a chalkboard to a large sheet of paper and markers.
- Looking at a specific example, we have five columns on our Kanban board: “backlog,” “not started,” “in progress,” “testing,” and “completed.”
- Since workflow stages can vary based on the nature of the work, the columns will change to match the needs of the particular team

(b) What do the columns on a typical Kanban board represent?

Backlog—The product owner puts all of the requirements (stories) on sticky notes in priority order. The estimated effort of the requirement is also noted. This list could be quite lengthy, depending on the size of the backlog.

- Not started—These stories have been selected by a developer and are therefore assigned, but the development work has yet to begin. This is equivalent to a “pending” queue, or the next items to be worked.
- In progress—These are the requirements that are currently being coded by a developer.
- Testing—This is when the testing is done to ensure that the code is working as intended.
- Completed—The requirements in this column are considered ready to be delivered to a customer.

(4 marks)

(c) What is the difference between a burn-up chart and a burn-down chart? How do you know from a burn-down chart if you are behind schedule?

(4 marks)

(d) What is the purpose of the stand-up meeting? What three key questions are answered by team members during a daily stand-up meeting? Who typically gets assigned action items during a daily stand-up meeting?

(4 marks)

(d) Product backlogs vary in their depth, breadth, and quality. Explain how the DEEP acronym may be employed to improve product backlog quality.

Backlogs vary in their depth, breadth, and quality.

- The acronym DEEP, which stands for Detailed appropriately, Estimated, Emergent, and Prioritized provides sound guidance on how to improve backlog quality.
- Detailed appropriately means that the highest priority stories contain sufficient detail for the development teams to deliver them: Questions are answered, and the necessary clarifications are included in the story description. Acceptance criteria are a component of the essential detail for the story.
- Estimated means that the team understands the stories and believes there is adequate information to estimate the level of effort or amount of time required to deliver the story.
- Emergent refers to the backlog’s constant evolution: As new information is learned about the marketplace, a competitor, or a technological advancement, the backlog is modified, reprioritized, or clarified.
- Prioritized means exactly that—the user stories are in priority order, with the highest priority items that will deliver the most business value at the top of the backlog for immediate development.

(4 marks)

[20 marks]

