# CSEN4174: Software Engineering Lab

## Assignment 2

**Team number: \_\_\_\_\_\_\_\_\_\_4\_\_\_\_\_\_\_\_\_\_\_\_**

**Team name: \_\_\_\_\_\_\_\_\_Decrypt\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

*Before saving this file, please rename the file as 25-Odd-CSEN4174-Lab01-YYYYZZ-Name.docx, where “YYYY” is either AIML or DASC depending on whether you are AIML or Data Science students, “ZZ” is your team number and “Name” is the name of your team. Incorrectly named files will not be evaluated.*

### **Activity 2.1: Product backlog and sprint backlog**

Observe the following diagram carefully. Scrum is a lightweight, iterative method for managing software development in a complex and changing environment. It has a cyclical nature – the “sprint”, and processes within it, repeat over and over. The sprint is a time-boxed iteration of development where a specific set of functionalities are analysed, designed, implemented, tested, and released for use. The product “ backlog” is a collection of all the user stories that can be implemented in the product. And the sprint “backlog” is the collection of user stories that a team will develop in a particular sprint. The user stories that you developed in Lab 1 are part of your product backlog.



In this activity, identify a subset of user stories – at least five (5) – from the product backlog to include in your first sprint; and justify why you selected each user story. Your selection can not be random; it needs to capture a cohesive line of functionality of the product.

|  |  |  |
| --- | --- | --- |
| **SL No** | **User story title** | **Justification for including in the first sprint** |
| 1 | Student Registration & Verification | Essential to ensure only authorized students can use the system; forms the entry point of the application. |
| 2 | Student Login | Required for authenticated access; ensures secure and restricted access to resources. |
| 3 | Rent a Locker | Core functionality; enables students to reserve a locker ensuring no duplicates and liability compliance. |
| 4 | Access Locker | Provides a secure way to simulate locker access using OTP; key part of locker interaction. |
| 5 | Return Locker | Completes the locker usage lifecycle; handles penalties and updates availability. |

**Activity 2.2: Product backlog and sprint backlog**

Read Sections 15.1, 15.2, 15.3 15.4 of Software-Engg-Concepts-&-Applications-Chapter-15.pdf to understand how use cases are identified from the functionality of a software system. Using the template below, write use cases – at least five (5) -- for the functionality you will develop in your first sprint. Please be as detailed as possible, as the use cases will drive the following activities of the development life cycle. Please note that there *may or* *may not* be a one-to-one correspondence between the user stories you identified in Activity 2.1, and the use cases.

|  |  |
| --- | --- |
| **Use case ID** | UC1 |
| **Name** | Student Registration & Verification |
| **Objective** | Allow students to register and verify their email using OTP before accessing the system. |
| **Pre-conditions** | Student has not registered on the platform. |
| **Post-conditions** | Student's data is stored and verification status updated upon OTP confirmation. |
| **Actors** | Student |
| **Trigger** | Student submits registration form with ID, name, and email. |
| **Normal flow** | System checks for unique ID → Sends OTP using SMTP → Stores registration data. |
| **Alternate flow** | If ID is not unique → Registration is denied with appropriate message. |
| **Interacts with** | Auth DB, SMTP Server |
| **Open issues** | Handling SMTP delivery failures or delays. |

|  |  |
| --- | --- |
| **Use case ID** | UC2 |
| **Name** | Student Login |
| **Objective** | Allow verified students to log in and access dashboard securely. |
| **Pre-conditions** | Student is registered and verified. |
| **Post-conditions** | JWT token issued and user session initiated. |
| **Actors** | Student |
| **Trigger** | Student submits login credentials. |
| **Normal flow** | System validates credentials and verification → Issues JWT token. |
| **Alternate flow** | Unverified or incorrect credentials → Login is denied. |
| **Interacts with** | Auth DB, JWT library |
| **Open issues** | Handling brute-force attacks and rate limiting properly. |

|  |  |
| --- | --- |
| **Use case ID** | UC3 |
| **Name** | Rent a Locker |
| **Objective** | Allow eligible students to view available lockers and rent one with liability acceptance. |
| **Pre-conditions** | Student is logged in and currently has no locker. |
| **Post-conditions** | Locker assigned and marked as occupied in DB. |
| **Actors** | Student |
| **Trigger** | Student clicks 'Rent Locker' and accepts terms. |
| **Normal flow** | System verifies availability → Marks locker as occupied → Logs rental and mock payment. |
| **Alternate flow** | Student already has a locker → Rental denied. |
| **Interacts with** | Locker DB, Rental DB, Audit Log |
| **Open issues** | Preventing race conditions in locker assignment. |

|  |  |
| --- | --- |
| **Use case ID** | UC4 |
| **Name** | Access Locker |
| **Objective** | Simulate locker access using temporary OTP code generated on request. |
| **Pre-conditions** | Student has an active locker rental. |
| **Post-conditions** | OTP issued and access logged in system. |
| **Actors** | Student |
| **Trigger** | Student requests access from dashboard. |
| **Normal flow** | OTP is generated and shown → Logged in audit trail → OTP expires after use. |
| **Alternate flow** | No active locker rental → Access denied. |
| **Interacts with** | Locker Access DB, Audit Log DB |
| **Open issues** | Ensuring OTP security and expiration. |

|  |  |
| --- | --- |
| **Use case ID** | UC5 |
| **Name** | Return Locker |
| **Objective** | Allow student to return rented locker and calculate any applicable penalty. |
| **Pre-conditions** | Student has an active locker rental. |
| **Post-conditions** | Locker marked available, penalty logged, rental archived. |
| **Actors** | Student |
| **Trigger** | Student clicks 'Return Locker'. |
| **Normal flow** | System checks due date → Calculates penalty if overdue → Updates status and logs. |
| **Alternate flow** | Locker already returned or not found → Action denied. |
| **Interacts with** | Locker DB, Rental DB |
| **Open issues** | Properly defining penalty calculation logic. |