Software Engineering: Project

e-Val: An Online Exam and Quiz Management System

Development Methodology - Sprint 1

EST. 2025

PROCRAFTERS

Stop thinking. Start working.

Team Name: Procrafters

Members:

I220827 – Muhammad Ali

I220902 – Hussain Ali Zaidi

I221186 – Abdullah bin Azeem

Submission Date: 28/02/2025

Module to be developed: User Access (+ Pre-requisites)

Selected User Stories:

Authentication -

As a: User,
I want to: create an account and log in
so that I can: access my dashboard.

Sub-user stories:

As a: Student,
I want to: register an account
so that I can: access quizzes.

Acceptance Criteria: The student registers their account and is redirected to their dashboard.

As a: Teacher,
I want to: register an account
so that I can: create and manage quizzes.

Acceptance Criteria: The teacher register their account and is redirected to their dashboard.

As a: User,
I want to: reset my password
so that I can: regain access if I forget it.

Acceptance Criteria: The user is allowed to chose a new password and the password is updated in the database.

As a: User,
I want to: the system to verify my email upon registration
So I can: to ensure security.

Acceptance Criteria: The code sent for verification is verified with the one entered by user adn the email is linked to the registered account.

As a: User,
I want to: be able to log in
So that I can: access my registered account.

Acceptance Criteria: Users credentials are authenticated and they are redirected to their corresponding dashboard.

Classes Enrollment -

As a: Student,
I want to: enroll in an available class
So that I can: take exams by either opting for it or getting an invitation from a teacher.

Acceptance Criteria: Students can browse and request to join classes, while teachers receive notifications and can manually manage enrollments. Once enrolled, students gain access to quizzes and course materials.

Structured Specifications:

1. Student Registration

Input: Student enters name, email, password, and other required details.

Process: System checks if email is unique. Passwords are hashed and stored in the database. Student account is created, and a confirmation is displayed.

Output: If successful, redirect to the student dashboard. If email exists, display an error message.

Exceptions: If mandatory fields are missing, prompt the user to complete them.

2. Teacher Registration

Input: Teacher enters name, email, password, and other required details.

Process: System verifies if email is unique. Passwords are securely stored, and an account is created.

Output: If successful, redirect to the teacher dashboard. If email exists, show an error message.

Exceptions: If required fields are missing, prompt the user.

3. Password Reset

Input: User enters email to request a password reset. User provides a new password via reset link.

Process: System verifies email existence. Generates and sends a reset link. User sets a new password, which is hashed and updated.

Output: Success message upon password update. Error if email is unregistered.

Exceptions: If the reset link is expired or invalid, notify the user. If passwords do not match, prompt for re-entry.

4. Email Verification

Input: User provides an email during registration.

Process: System generates and emails a verification code/link. User enters the code or clicks the link. System validates the code and marks the email as verified.

Output: Confirmation message on success. Error message if verification fails.

Exceptions: Allow users to request a new code if needed.

5. Login Authentication

Input: User enters email and password.

Process: System validates credentials. If correct, retrieve the user role and redirect accordingly.

Output: Successful login redirects to the user dashboard. Failure prompts an "Invalid credentials" message.

Exceptions: Lockout after multiple failed attempts. Prevent login for unverified users.

6. Class Enrollment

Input: Student selects a class or accepts an invitation.

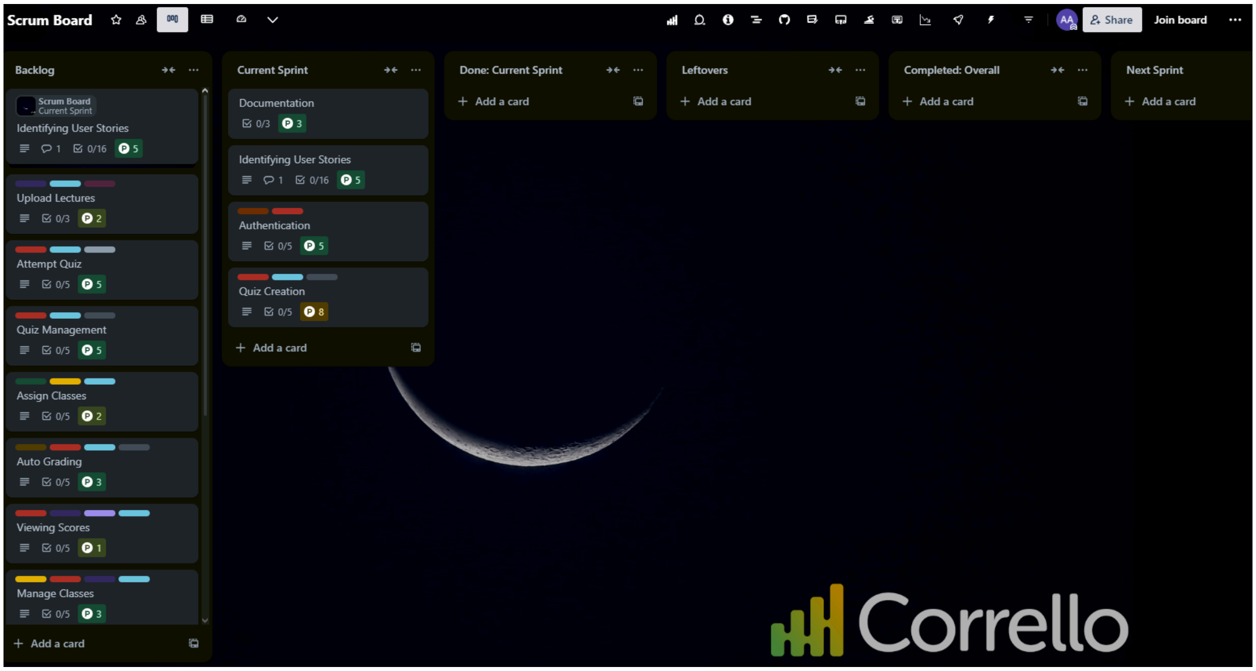
Process: System checks enrollment status. Updates student-class association in the database.

Output: Confirmation message on successful enrollment. Error if enrollment fails.

Exceptions: If class is full, deny enrollment and notify the student. Prevent duplicate enrollments.

SCRUM Board Updates:

Initial Board:



Backlog

- Scrum Board / Current Sprint — Identifying User Stories — 1 — 0/16 — 5
- Upload Lectures — 0/3 — 2
- Attempt Quiz — 0/5 — 5
- Quiz Management — 0/5 — 5
- Assign Classes — 0/5 — 2
- Auto Grading — 0/5 — 3
- Viewing Scores — 0/5 — 1
- Manage Classes — 0/5 — 3
- + Add a card

Current Sprint

- Documentation — 0/3 — 3
- Identifying User Stories — 1 — 0/16 — 5
- Authentication — 0/5 — 5
- Quiz Creation — 0/5 — 8
- + Add a card

Done: Current Sprint

- + Add a card

Leftovers

- + Add a card

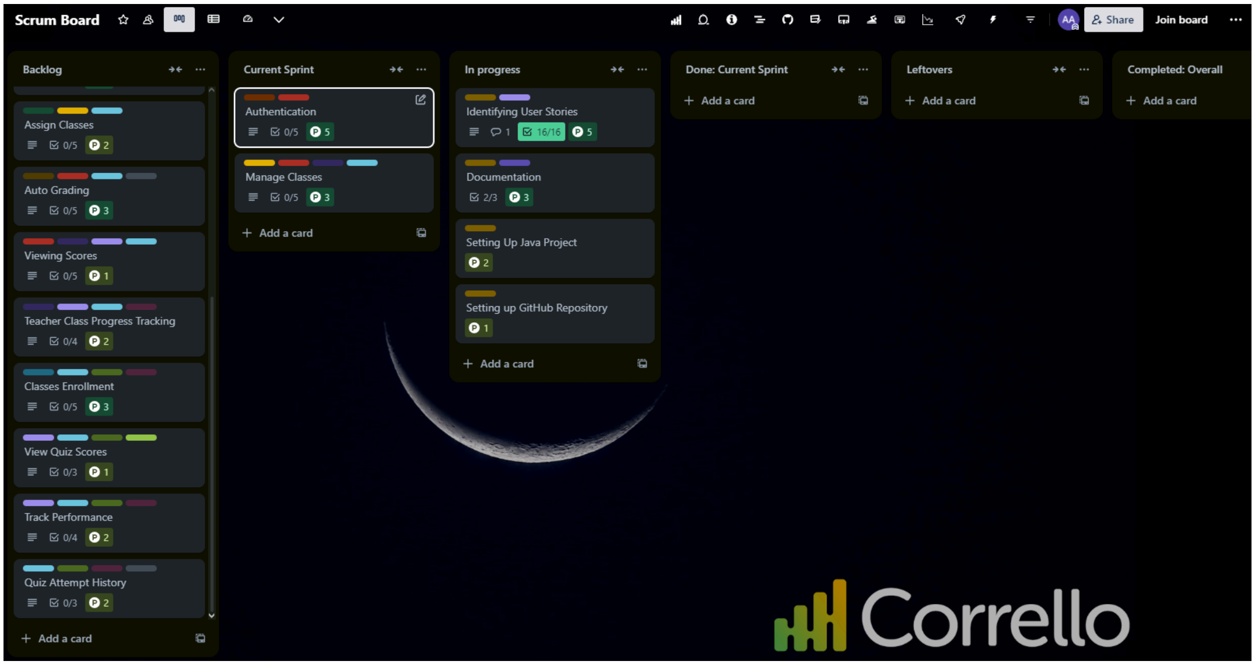
Completed: Overall

- + Add a card

Next Sprint

- + Add a card

In progress Board:



Backlog

- Assign Classes — 0/5 — 2
- Auto Grading — 0/5 — 3
- Viewing Scores — 0/5 — 1
- Teacher Class Progress Tracking — 0/4 — 2
- Classes Enrollment — 0/5 — 3
- View Quiz Scores — 0/3 — 1
- Track Performance — 0/4 — 2
- Quiz Attempt History — 0/3 — 2
- + Add a card

Current Sprint

- Authentication — 0/5 — 5
- Manage Classes — 0/5 — 3
- + Add a card

In progress

- Identifying User Stories — 1 — 16/16 — 5
- Documentation — 2/3 — 3
- Setting Up Java Project — 2
- Setting up GitHub Repository — 1
- + Add a card

Done: Current Sprint

- + Add a card

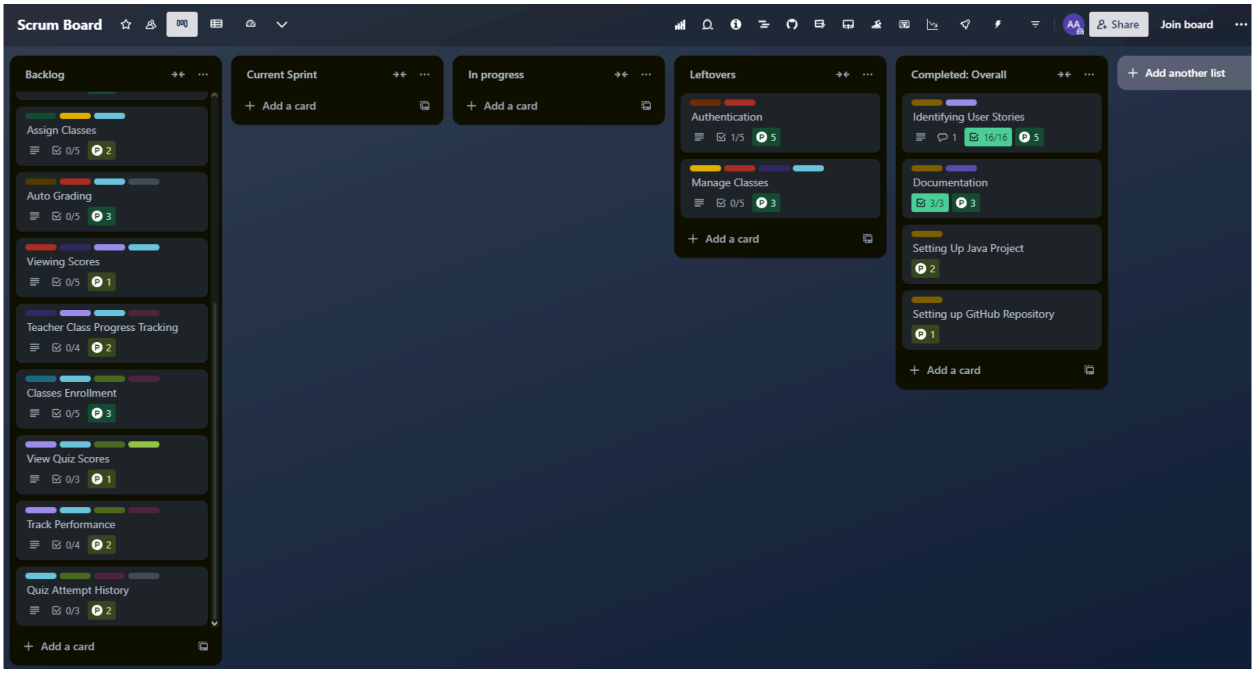
Leftovers

- + Add a card

Completed: Overall

- + Add a card

Ending Board:



Scrum Board

Share

Join board

Backlog

Assign Classes
0/5 P 2

Auto Grading
0/5 P 3

Viewing Scores
0/5 P 1

Teacher Class Progress Tracking
0/4 P 2

Classes Enrollment
0/5 P 3

View Quiz Scores
0/3 P 1

Track Performance
0/4 P 2

Quiz Attempt History
0/3 P 2

+ Add a card

Current Sprint

+ Add a card

In progress

+ Add a card

Leftovers

Authentication
1/5 P 5

Manage Classes
0/5 P 3

+ Add a card

Completed: Overall

Identifying User Stories
1 16/16 P 5

Documentation
3/3 P 3

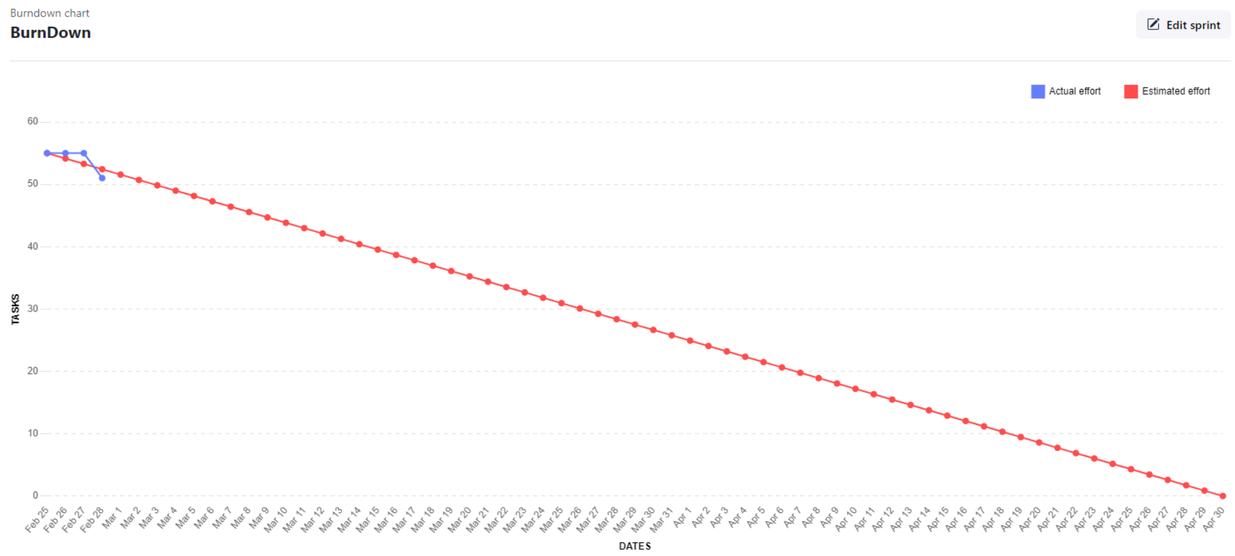
Setting Up Java Project
P 2

Setting up GitHub Repository
P 1

+ Add a card

+ Add another list

Burndown Chart:


Burndown chart
BurnDown
Edit sprint
Actual effort
Estimated effort
TASKS
Feb 25
Feb 26
Feb 27
Feb 28
Mar 1
Mar 2
Mar 3
Mar 4
Mar 5
Mar 6
Mar 7
Mar 8
Mar 9
Mar 10
Mar 11
Mar 12
Mar 13
Mar 14
Mar 15
Mar 16
Mar 17
Mar 18
Mar 19
Mar 20
Mar 21
Mar 22
Mar 23
Mar 24
Mar 25
Mar 26
Mar 27
Mar 28
Mar 29
Mar 30
Mar 31
Apr 1
Apr 2
Apr 3
Apr 4
Apr 5
Apr 6
Apr 7
Apr 8
Apr 9
Apr 10
Apr 11
Apr 12
Apr 13
Apr 14
Apr 15
Apr 16
Apr 17
Apr 18
Apr 19
Apr 20
Apr 21
Apr 22
Apr 23
Apr 24
Apr 25
Apr 26
Apr 27
Apr 28
Apr 29
Apr 30
DATES