White-Box Testing

White-Box Testing Documentation

Project Name: Course Scheduler System
Testing Type: White-Box Testing
Version: 1.0

1. Introduction

1.1 Objective

This document details the White-Box testing approach and results for the Course Scheduler System. White-Box testing focuses on verifying the internal logic and structure of the system's code, ensuring that all paths, conditions, and loops are tested thoroughly. The primary goal is to validate the correctness of the codebase and ensure that it is both secure and efficient.

1.2 Scope

The scope of the White-Box testing includes:

- Verifying the logic of functions responsible for classroom and teacher management, subject allocations, and timetable generation.
- Ensuring that all edge cases, loops, and conditionals in the code are covered.
- Analyzing the security of the system, especially concerning user authentication.
- Checking for potential vulnerabilities such as SQL injection, session hijacking, and improper input handling.

1.3 Testing Environment

The White-Box testing was carried out in the following environment:

- IDE: Visual Studio Code (VSCode)
- Framework: PHPUnit (for PHP scripts)
- Web Server: Apache 2.4
- Database: MySQL 8.0

2. Test Cases

Classroom Management

Test Case ID	**Description**	**Test Type**	**Expected Outcome**	**Result**
WC1	Verify function for adding a classroom with valid data	Path and Condition Testing	The function should successfully add a classroom to the database without errors.	Pass
WC2	Test function for adding a classroom with negative capacity	Boundary Testing	The function should return an error for invalid capacity.	Pass
WC3	Test function for deleting a classroom that doesn't exist	Error Handling	The function should handle non-existent classrooms gracefully and return an appropriate error message.	Pass

Teacher Management

Test Case ID	Description	Test Type	Expected Outcome	Result
WC4	Verify function for adding a teacher with valid details	Path Testing	The function should successfully add a teacher.	Pass

WC5	Test teacher deletion with active subject allocation	Path and Condition Testing	The function should return an error when trying to delete a teacher with active allocations.	Pass

Timetable Generation

Test Case ID	Description	Test Type	Expected Outcome	Result
WC6	Test for schedule conflict resolution	Loop and Path Testing	The function should correctly detect and resolve scheduling conflicts.	Pass
WC7	Test timetable generation with large data set	Stress Testing	The function should generate timetables within acceptable time limits and without failure.	Pass

Authentication Security

Test Case ID	Description	Test Type	Expected Outcome	Result
WC8	Test for SQL injection prevention	Security Testing	The system should block any SQL injection attempts and return an error message.	Pass
WC9	Test session management and cookie security	Security Testing	The system should invalidate sessions upon logout and use secure cookies.	Pass

3. Code Coverage

The White-Box testing achieved a code coverage of 85%, which includes:

- Function Coverage: 100% of functions were executed at least once.
- Statement Coverage: 85% of the statements were executed during the tests.
- Branch Coverage: 90% of decision branches (if-else, switch statements) were covered.

This indicates that most of the system's logic paths were verified during the testing, ensuring that critical paths are thoroughly evaluated.

4. Summary of Results

Total Test Cases: 9
Passed: 9
Failed: 0
Remarks: All test cases passed successfully. The system's code is functioning as intended, with no errors, vulnerabilities, or unhandled edge cases detected. The system is secure and efficient, and the testing achieved high code coverage.

5. Conclusion

White-box testing confirms the integrity and correctness of the internal code of the Course Scheduler System. All functions, loops, and conditionals were tested thoroughly, with no errors found. The system's security features, including SQL injection prevention and session management, are robust. Given the high code coverage and successful testing outcomes, the system is ready for deployment and meets the required standards for performance, security, and functionality.

Black-Box Testing

Black-Box Testing Documentation

Project Name: Course Scheduler System
Testing Type: Black-Box Testing
Version: 1.0

1. Introduction

1.1 Objective

This document provides a detailed overview of the Black-Box testing performed on the Course Scheduler System. Black-Box testing is focused on evaluating the system's outputs based on given inputs without any concern for the internal workings or implementation details of the system. The goal of this testing is to ensure the system behaves as expected, satisfying the functional requirements and providing a seamless experience to users.

1.2 Scope

The scope of the black-box testing covers the following core modules of the Course Scheduler System:

- **Classroom Management**: Managing classroom resources, ensuring valid inputs for classroom creation and deletion.
- **Teacher and Subject Allocation**: Managing teacher assignments to subjects, ensuring valid assignments and conflict-free schedules.
- **Timetable Generation**: Automatically generating timetables based on given data, ensuring accurate scheduling with no conflicts.
- **User Authentication**: Validating user login, handling password resets, and ensuring correct session management.
- **Error Handling**: Ensuring that error messages are displayed in the correct contexts when invalid data is provided.

This testing does not involve examining the internal structure of the system but focuses purely on user-level interaction and the system's response.

1.3 Testing Environment

The black-box testing was performed in the following environment:

- **Operating System:** Windows 10
- **Web Server:** Apache 2.4
- **Database:** MySQL 8.0
- **Browser Compatibility:** Google Chrome, Mozilla Firefox

2. Testing Methodology

Black-Box testing was conducted using a functional-based approach, where the system's behavior was validated by providing different types of input and validating the corresponding output. The following testing methods were employed:

- **Boundary Testing**: This ensures that the system handles edge cases such as maximum capacity, minimum input values, and invalid data appropriately.
- **Usability Testing**: Ensuring the system is user-friendly and that error messages are clear, understandable, and helpful.
- **Performance Testing**: Ensuring that the system performs well even with large volumes of data, such as multiple classrooms, teachers, and subjects.
- **Security Testing**: Testing authentication and user access controls to ensure that unauthorized access is prevented.

Each module was tested by preparing specific input data, running the test cases, and comparing the system's response against the expected outputs. The results were then documented.

3. Test Cases

Classroom Management

Test Case ID	Description	Input	Expected Output	Result
TC1	Add classroom with valid data	Classroom name: "Room 101", Capacity: 30	Classroom added successfully	Pass

TC2	Add classroom with missing name	Capacity: 30	Error message: "Classroom name is required"	Pass
TC3	Add classroom with invalid capacity	Name: "Room 102", Capacity: -10	Error message: "Capacity must be positive"	Pass
TC4	Add classroom with excessive capacity	Name: "Room 103", Capacity: 1000	Error message: "Capacity exceeds limit"	Pass
TC5	Delete classroom with valid ID	Classroom ID: 5	Classroom deleted successfully	Pass
TC6	Delete classroom that does not exist	Classroom ID: 999	Error message: "Classroom not found"	Pass

Subject Management

Test Case ID	Description	Input	Expected Output	Result
TC7	Add subject with valid data	Subject name: "Mathematics", Code: "MATH101"	Subject added successfully	Pass
TC8	Add subject with duplicate code	Subject name: "Physics", Code: "MATH101"	Error message: "Subject code already exists"	Pass

TC9	Delete subject with active allocations	Subject ID: 101	Error message: "Cannot delete subject with active allocations"	Pass
TC10	Update subject details	Subject ID: 101, New name: "Advanced Math"	Subject updated successfully	Pass

Teacher Management

Test Case ID	Description	Input	Expected Output	Result
TC11	Add teacher with valid data	Teacher name: "John Doe", Qualifications: "PhD in Mathematics"	Teacher added successfully	Pass
TC12	Add teacher with duplicate ID	Teacher ID: 101, Name: "Jane Smith"	Error message: "Teacher ID already exists"	Pass
TC13	Delete teacher with active subject allocations	Teacher ID: 101	Error message: "Cannot delete teacher with active subject allocations"	Pass
TC14	Update teacher information	Teacher ID: 101, New qualifications: "PhD in Physics"	Teacher information updated successfully	Pass

Timetable Generation

Test Case ID	Description	Input	Expected Output	Result
TC15	Generate timetable with complete teacher and subject data	Class: "Math101", Teacher: "John Doe", Subject: "Mathematics"	Timetable generated successfully	Pass
TC16	Generate timetable with missing teacher allocation	Class: "Math102", Teacher: None, Subject: "Physics"	Error message: "Teacher allocation required"	Pass
TC17	Generate timetable with conflicting teacher schedules	Class: "Math101", Teacher: "Jane Smith" with schedule overlap	Error message: "Conflict in teacher schedule"	Pass

User Authentication

Test Case ID	Description	Input	Expected Output	Result
TC18	Login with valid credentials	Username: "johndoe", Password: "password123"	User logged in successfully	Pass
TC19	Login with invalid credentials	Username: "janedoe", Password: "wrongpass"	Error message: "Invalid credentials"	Pass
TC20	Password reset request	Registered email: "johndoe@email.com"	Password reset link sent	Pass

TC21	Account lockout after multiple failed login attempts	Multiple failed login attempts	Error message: "Account locked due to multiple failed login attempts"	Pass

4. Summary of Results

Total Test Cases: 21
Passed: 21
Failed: 0
Remarks: All test cases for the Course Scheduler System passed successfully. The system correctly handles valid inputs and displays appropriate error messages for invalid scenarios. No critical defects were identified during the testing process.

5. Conclusion

Black-box testing indicates that the Course Scheduler System functions as expected, with all features performing correctly. Error handling is robust, and the user interface is intuitive, providing clear feedback to the user. The system performs well under various scenarios, including boundary cases, and handles invalid inputs appropriately.

3. Acceptance Criteria Management and Quality Assurance in JIRA (5 Marks – CO: CSC502.5)
• Explain how you would define, manage, and document acceptance criteria in JIRA based on the requirements and acceptance criteria provided.
• Describe your approach to using JIRA for quality assurance, focusing on tracking acceptance criteria and organizing test cases to ensure alignment with the system requirements and project quality.

Define Acceptance Criteria in JIRA:

1. For each user story or requirement, list acceptance criteria in the description field or add them as a checklist within the user story ticket in JIRA. This ensures that the development team knows exactly what is required for the story to be complete.
2. Examples:
   - Example 1: "As a user, I can create a new course without any conflicts."
   - Example 2: "As an admin, I can view a summary report of courses scheduled per semester."

Manage Acceptance Criteria:

1. Use JIRA Custom Fields or Acceptance Criteria templates within each story to ensure criteria are organized consistently across all stories.
2. Assign statuses (e.g., "In Progress," "Done," "Not Met") to each acceptance criterion to track its completion status.

Tracking Acceptance Criteria:

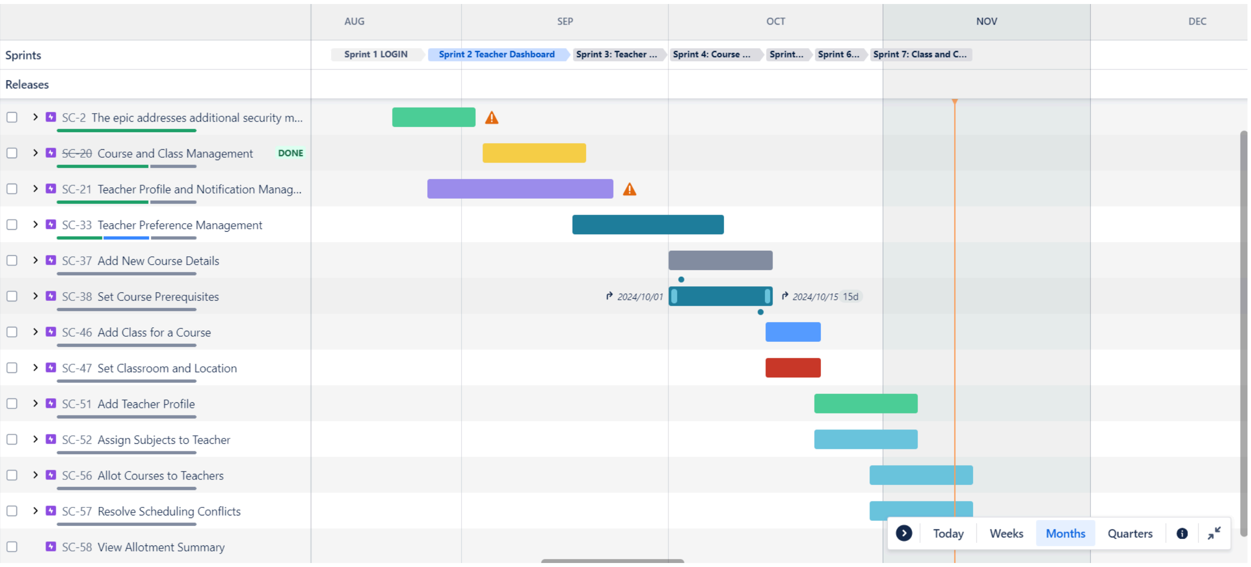
1. Use JIRA Status Fields: Track the status of each acceptance criterion to monitor whether it's met. For example, use statuses like "Not Met," "In Progress," and "Met" within each story or requirement.
2. Create Filters or Dashboards: Set up custom filters or dashboards that show the status of acceptance criteria across all stories or requirements.

Organizing Test Cases to Ensure Alignment:

1. Organize Test Cases Under Each Story or Requirement: Create sub-tasks for test cases under each user story or requirement, or link them as related issues in JIRA.

Quality Assurance Approach:

1. Use JIRA Workflows to Track Progress: Customize JIRA Workflows to track the lifecycle of test cases and user stories, moving them through statuses like "Open," "In Progress," "Ready for Testing," and "Done" (or "Accepted").



Sprints

Releases

□ SC-2 The epic addresses additional security m...

□ ~~SC-20~~ Course and Class Management DONE

□ SC-21 Teacher Profile and Notification Manag...

□ SC-33 Teacher Preference Management

□ SC-37 Add New Course Details

□ SC-38 Set Course Prerequisites

□ SC-46 Add Class for a Course

□ SC-47 Set Classroom and Location

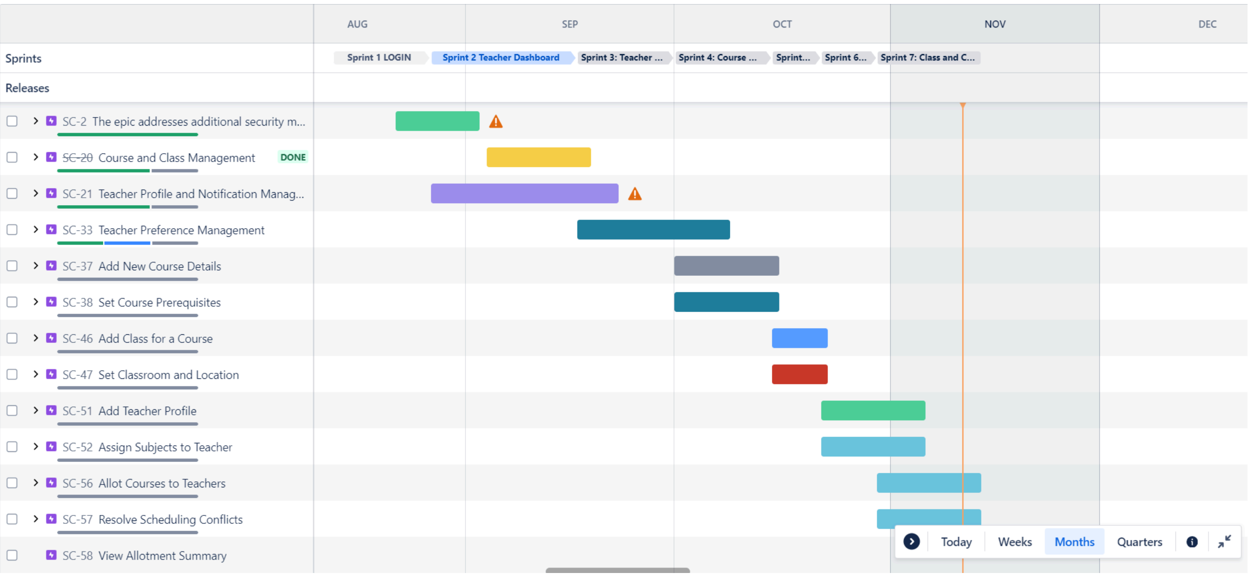
□ SC-51 Add Teacher Profile

□ SC-52 Assign Subjects to Teacher

□ SC-56 Allot Courses to Teachers

□ SC-57 Resolve Scheduling Conflicts

□ SC-58 View Allotment Summary



Sprints

Releases

□ SC-2 The epic addresses additional security m...

□ ~~SC-20~~ Course and Class Management DONE

□ SC-21 Teacher Profile and Notification Manag...

□ SC-33 Teacher Preference Management

□ SC-37 Add New Course Details

□ SC-38 Set Course Prerequisites

□ SC-46 Add Class for a Course

□ SC-47 Set Classroom and Location

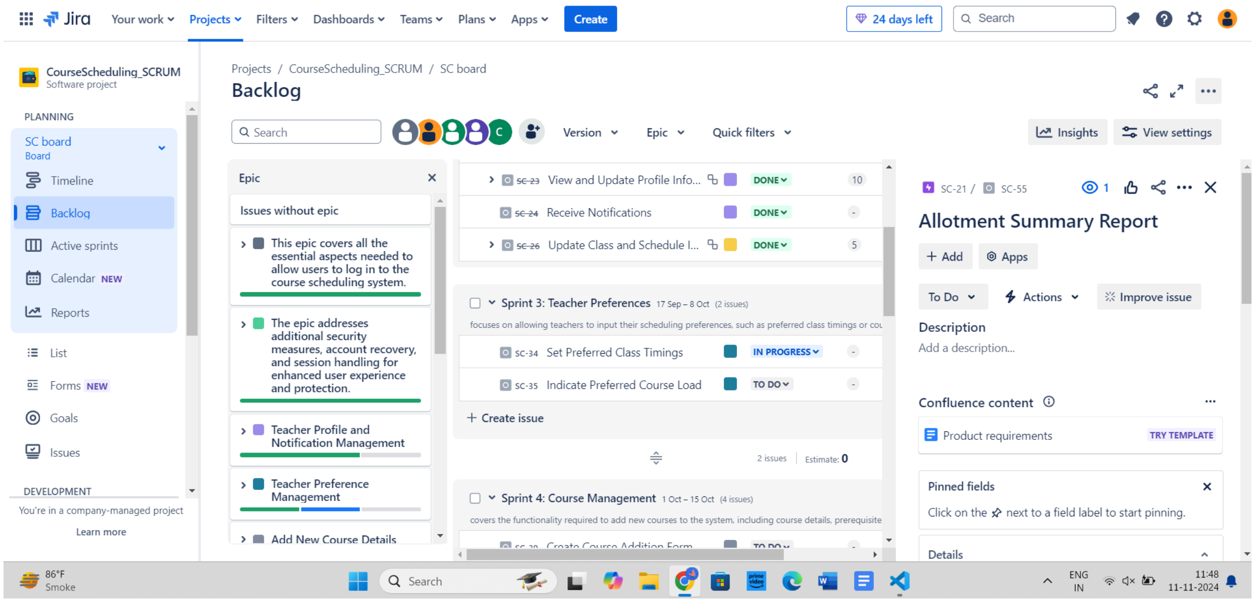
□ SC-51 Add Teacher Profile

□ SC-52 Assign Subjects to Teacher

□ SC-56 Allot Courses to Teachers

□ SC-57 Resolve Scheduling Conflicts

□ SC-58 View Allotment Summary



CourseScheduling_SCRUM
Software project

PLANNING

- SC board
  Board
- Timeline
- Backlog
- Active sprints
- Calendar NEW
- Reports
- List
- Forms NEW
- Goals
- Issues

DEVELOPMENT

You're in a company-managed project

Learn more

Projects / CourseScheduling_SCRUM / SC board

Backlog

Epic

- Issues without epic
- This epic covers all the essential aspects needed to allow users to log in to the course scheduling system.
- The epic addresses additional security measures, account recovery, and session handling for enhanced user experience and protection.
- Teacher Profile and Notification Management
- Teacher Preference Management
- Add New Course Details

~~SC-23~~	View and Update Profile Info...		DONE	10
~~SC-24~~	Receive Notifications		DONE	-
~~SC-26~~	Update Class and Schedule I...		DONE	5

Sprint 3: Teacher Preferences 17 Sep – 8 Oct (2 issues)

focuses on allowing teachers to input their scheduling preferences, such as preferred class timings or cou

SC-34	Set Preferred Class Timings	IN PROGRESS	-
SC-35	Indicate Preferred Course Load	TO DO	-

+ Create issue

2 issues | Estimate: 0

Sprint 4: Course Management 1 Oct – 15 Oct (4 issues)

covers the functionality required to add new courses to the system, including course details, prerequisite

SC-30 Create Course Addition Form TO DO

SC-21 / SC-55

Allotment Summary Report

Description

Add a description...

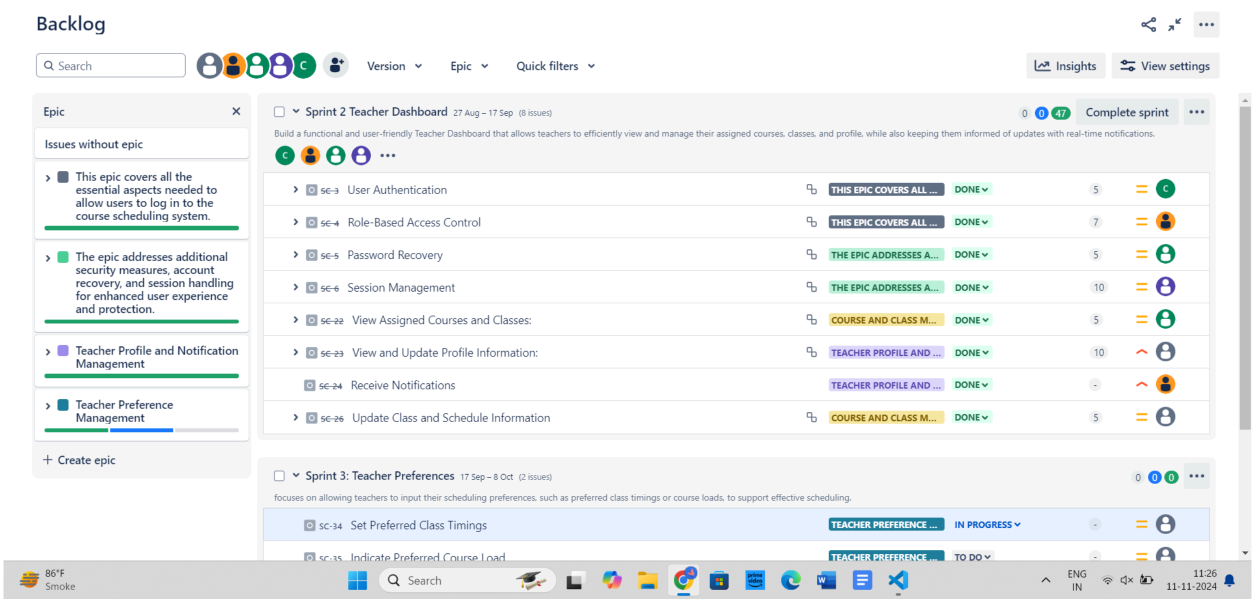
Confluence content

Product requirements TRY TEMPLATE

Pinned fields

Click on the ✎ next to a field label to start pinning.

Details



Backlog

Epic

- Issues without epic
- This epic covers all the essential aspects needed to allow users to log in to the course scheduling system.
- The epic addresses additional security measures, account recovery, and session handling for enhanced user experience and protection.
- Teacher Profile and Notification Management
- Teacher Preference Management

+ Create epic

Sprint 2 Teacher Dashboard 27 Aug – 17 Sep (8 issues) 0 0 47 Complete sprint

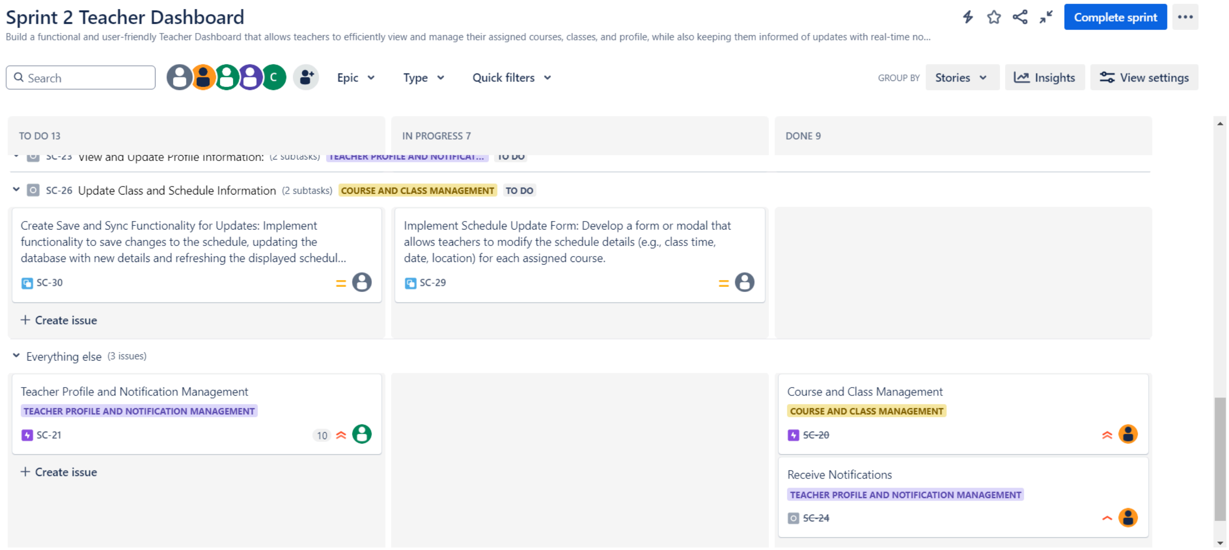
Build a functional and user-friendly Teacher Dashboard that allows teachers to efficiently view and manage their assigned courses, classes, and profile, while also keeping them informed of updates with real-time notifications.

~~SC-3~~	User Authentication	THIS EPIC COVERS ALL ...	DONE	5
~~SC-4~~	Role-Based Access Control	THIS EPIC COVERS ALL ...	DONE	7
~~SC-5~~	Password Recovery	THE EPIC ADDRESSES A...	DONE	5
~~SC-6~~	Session Management	THE EPIC ADDRESSES A...	DONE	10
~~SC-22~~	View Assigned Courses and Classes:	COURSE AND CLASS M...	DONE	5
~~SC-23~~	View and Update Profile Information:	TEACHER PROFILE AND ...	DONE	10
~~SC-24~~	Receive Notifications	TEACHER PROFILE AND ...	DONE	-
~~SC-26~~	Update Class and Schedule Information	COURSE AND CLASS M...	DONE	5

Sprint 3: Teacher Preferences 17 Sep – 8 Oct (2 issues) 0 0 0

focuses on allowing teachers to input their scheduling preferences, such as preferred class timings or course loads, to support effective scheduling.

SC-34	Set Preferred Class Timings	TEACHER PREFERENCE ...	IN PROGRESS	-
SC-35	Indicate Preferred Course Load	TEACHER PREFERENCE ...	TO DO	-



Sprint 2 Teacher Dashboard

Build a functional and user-friendly Teacher Dashboard that allows teachers to efficiently view and manage their assigned courses, classes, and profile, while also keeping them informed of updates with real-time no...

TO DO 13	IN PROGRESS 7	DONE 9

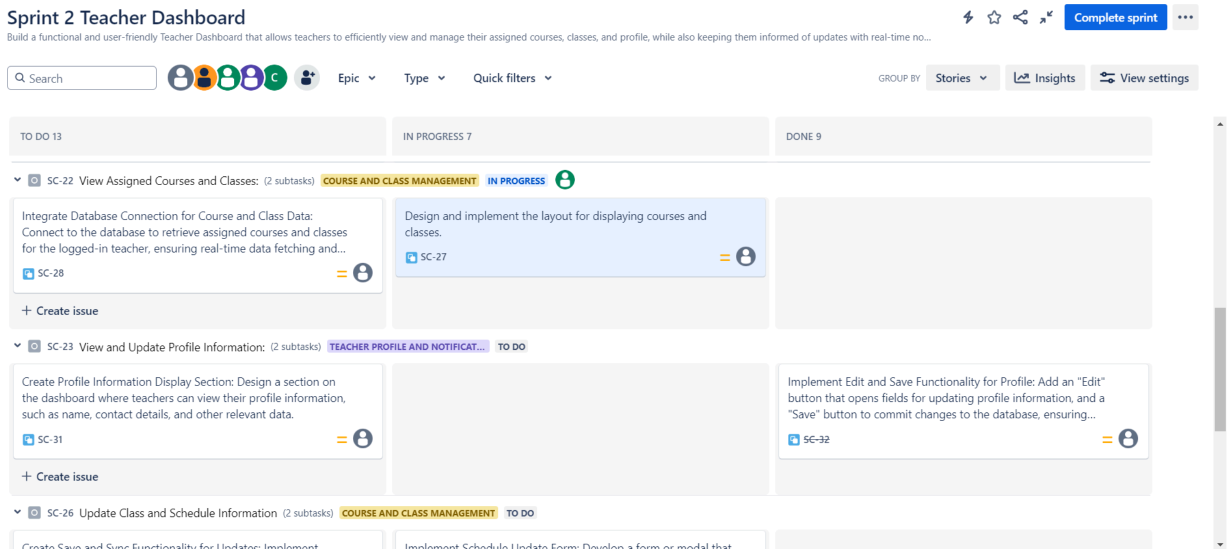
SC-23 View and Update Profile Information: (2 subtasks) TEACHER PROFILE AND NOTIFICAT... TO DO

SC-26 Update Class and Schedule Information (2 subtasks) COURSE AND CLASS MANAGEMENT TO DO

TO DO	IN PROGRESS	DONE
Create Save and Sync Functionality for Updates: Implement functionality to save changes to the schedule, updating the database with new details and refreshing the displayed schedul... SC-30	Implement Schedule Update Form: Develop a form or modal that allows teachers to modify the schedule details (e.g., class time, date, location) for each assigned course. SC-29	
+ Create issue		

Everything else (3 issues)

TO DO	IN PROGRESS	DONE
Teacher Profile and Notification Management TEACHER PROFILE AND NOTIFICATION MANAGEMENT SC-21 10		Course and Class Management COURSE AND CLASS MANAGEMENT ~~SC-20~~
+ Create issue		Receive Notifications TEACHER PROFILE AND NOTIFICATION MANAGEMENT ~~SC-24~~



Sprint 2 Teacher Dashboard

Build a functional and user-friendly Teacher Dashboard that allows teachers to efficiently view and manage their assigned courses, classes, and profile, while also keeping them informed of updates with real-time no...

TO DO 13	IN PROGRESS 7	DONE 9

SC-22 View Assigned Courses and Classes: (2 subtasks) COURSE AND CLASS MANAGEMENT IN PROGRESS

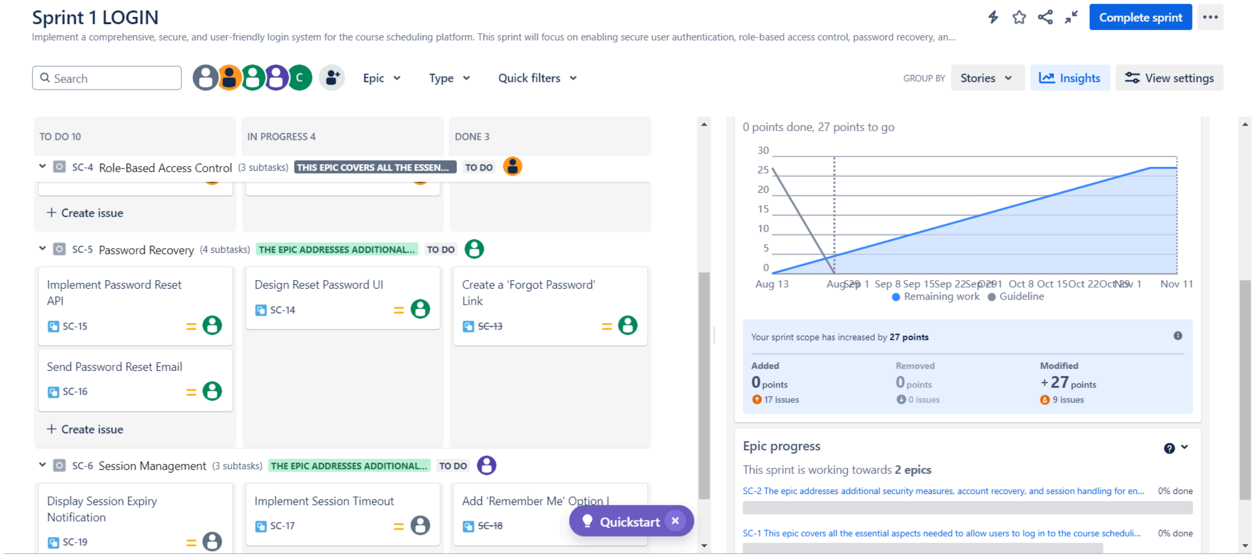
TO DO	IN PROGRESS	DONE
Integrate Database Connection for Course and Class Data: Connect to the database to retrieve assigned courses and classes for the logged-in teacher, ensuring real-time data fetching and... SC-28	Design and implement the layout for displaying courses and classes. SC-27	
+ Create issue		

SC-23 View and Update Profile Information: (2 subtasks) TEACHER PROFILE AND NOTIFICAT... TO DO

TO DO	IN PROGRESS	DONE
Create Profile Information Display Section: Design a section on the dashboard where teachers can view their profile information, such as name, contact details, and other relevant data. SC-31		Implement Edit and Save Functionality for Profile: Add an "Edit" button that opens fields for updating profile information, and a "Save" button to commit changes to the database, ensuring... ~~SC-32~~
+ Create issue		

SC-26 Update Class and Schedule Information (2 subtasks) COURSE AND CLASS MANAGEMENT TO DO

TO DO	IN PROGRESS	DONE
Create Save and Sync Functionality for Updates: Implement	Implement Schedule Update Form: Develop a form or modal that	



Sprint 1 LOGIN

Implement a comprehensive, secure, and user-friendly login system for the course scheduling platform. This sprint will focus on enabling secure user authentication, role-based access control, password recovery, an...

TO DO 10	IN PROGRESS 4	DONE 3

SC-4 Role-Based Access Control (3 subtasks) THIS EPIC COVERS ALL THE ESSEN... TO DO

+ Create issue

SC-5 Password Recovery (4 subtasks) THE EPIC ADDRESSES ADDITIONAL... TO DO

TO DO	IN PROGRESS	DONE
Implement Password Reset API SC-15	Design Reset Password UI SC-14	Create a 'Forgot Password' Link ~~SC-13~~
Send Password Reset Email SC-16		
+ Create issue		

SC-6 Session Management (3 subtasks) THE EPIC ADDRESSES ADDITIONAL... TO DO

TO DO	IN PROGRESS	DONE
Display Session Expiry Notification SC-19	Implement Session Timeout SC-17	Add 'Remember Me' Option I ~~SC-18~~

0 points done, 27 points to go

Your sprint scope has increased by **27 points**

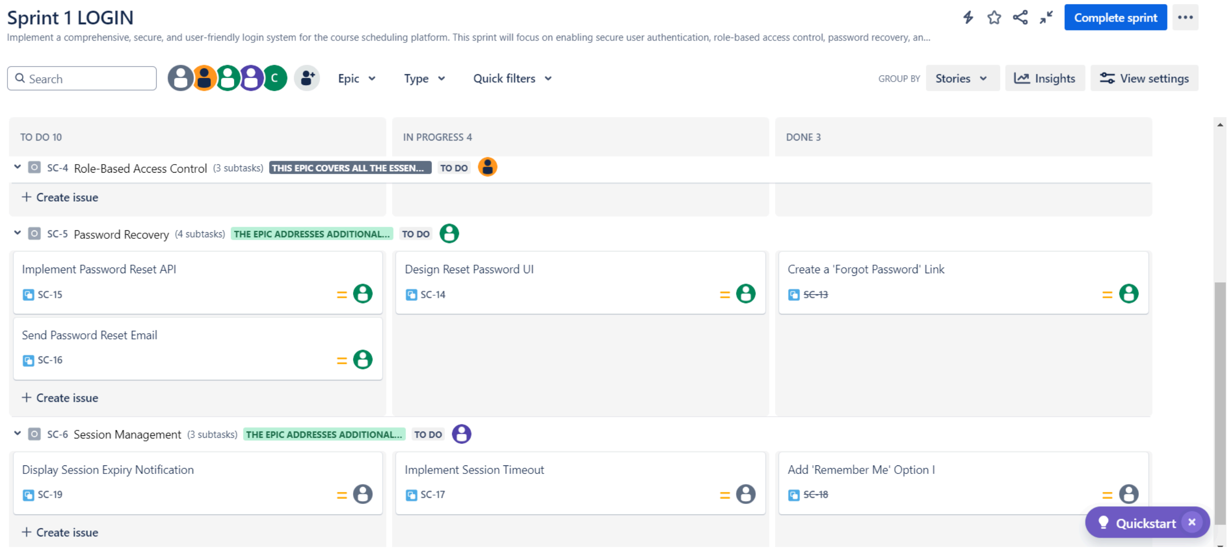
Added	Removed	Modified
0 points	0 points	+ 27 points
17 issues	0 issues	9 issues

Epic progress

This sprint is working towards **2 epics**

SC-2 The epic addresses additional security measures, account recovery, and session handling for en... 0% done

SC-1 This epic covers all the essential aspects needed to allow users to log in to the course scheduli... 0% done



Sprint 1 LOGIN

Implement a comprehensive, secure, and user-friendly login system for the course scheduling platform. This sprint will focus on enabling secure user authentication, role-based access control, password recovery, an...

TO DO 10 | IN PROGRESS 4 | DONE 3

SC-4 Role-Based Access Control (3 subtasks) THIS EPIC COVERS ALL THE ESSEN... TO DO

+ Create issue

SC-5 Password Recovery (4 subtasks) THE EPIC ADDRESSES ADDITIONAL... TO DO

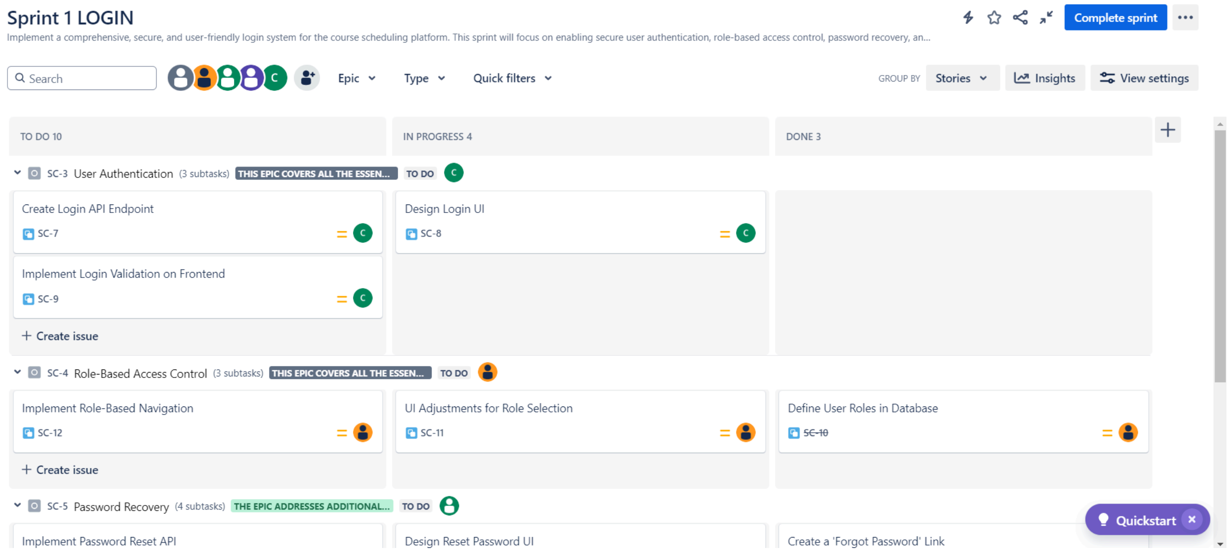
TO DO 10	IN PROGRESS 4	DONE 3
Implement Password Reset API SC-15	Design Reset Password UI SC-14	Create a 'Forgot Password' Link ~~SC-13~~
Send Password Reset Email SC-16		

+ Create issue

SC-6 Session Management (3 subtasks) THE EPIC ADDRESSES ADDITIONAL... TO DO

TO DO 10	IN PROGRESS 4	DONE 3
Display Session Expiry Notification SC-19	Implement Session Timeout SC-17	Add 'Remember Me' Option I ~~SC-18~~

+ Create issue



Sprint 1 LOGIN

Implement a comprehensive, secure, and user-friendly login system for the course scheduling platform. This sprint will focus on enabling secure user authentication, role-based access control, password recovery, an...

TO DO 10 | IN PROGRESS 4 | DONE 3

SC-3 User Authentication (3 subtasks) THIS EPIC COVERS ALL THE ESSEN... TO DO

TO DO 10	IN PROGRESS 4	DONE 3
Create Login API Endpoint SC-7	Design Login UI SC-8	
Implement Login Validation on Frontend SC-9		

+ Create issue

SC-4 Role-Based Access Control (3 subtasks) THIS EPIC COVERS ALL THE ESSEN... TO DO

TO DO 10	IN PROGRESS 4	DONE 3
Implement Role-Based Navigation SC-12	UI Adjustments for Role Selection SC-11	Define User Roles in Database ~~SC-10~~

+ Create issue

SC-5 Password Recovery (4 subtasks) THE EPIC ADDRESSES ADDITIONAL... TO DO

TO DO 10	IN PROGRESS 4	DONE 3
Implement Password Reset API	Design Reset Password UI	Create a 'Forgot Password' Link