Eventflow Project – Unit Test Report

Author: Dimitar Malamski

Date: May 2025

Project: Eventflow

Framework: ASP.NET MVC

Testing Framework: MSTest

Mocking Tool: Moq

Code Coverage: Coverlet + ReportGenerator

Contents

1. Introduction

This document summarizes the unit testing efforts for the Eventflow project. Unit testing was applied to ensure service-layer logic behaves correctly under both normal and edge-case conditions. The report includes the environment setup, test strategies, coverage metrics, and key outcomes.

2. Testing Environment

Component	Version / Details
.NET SDK	8.0
Testing Framework	MSTest 3.6.1
Mocking Library	Moq 4.20.72
Coverage Tool	Coverlet + ReportGenerator
OS	Windows 11

3. Testing Strategy

The unit tests were written for the service layer using MSTest. Repository dependencies were mocked using Moq, ensuring tests remained isolated and predictable. No database calls were executed directly. Each method was tested for both valid inputs and boundary conditions. Controllers and UI components were excluded from unit testing scope.

4. Code Coverage Summary

Metric	Value
Line Coverage	73 %
Branch Coverage	66 %

5. Test Cases Overview

5.1. PersonalEventService Tests

Test Case ID	Test Method Name	Description	Input(s)	Expected Outputs	Status
TC-PE-001	Constructor_Throws_WhenPersonalEventRepositoryIsNull	Throws if PersonalEventRepository is null	Constructor args (null, valid, valid, valid)	ArgumentNullException is thrown	Pass
TC-PE-002	Constructor_Throws_WhenCategoryRepositoryIsNull	Throws if CategoryRepository is null	Constructor args (valid, null, valid, valid)	ArgumentNullException is thrown	Pass
TC-PE-003	Constructor_Throws_WhenUserRepositoryIsNull	Throws if UserRepository is null	Constructor args (valid, null, valid, valid)	ArgumentNullException is thrown	Pass
TC-PE-004	Constructor_Throws_WhenInviteRepositoryIsNull	Throws if InviteRepository is null	Constructor args (valid, null, valid, valid)	ArgumentNullException is thrown	Pass
TC-PE-005	CreateAsync_CallsRepository_WithCorrectEvent	Creates a valid personal event	PersonalEvent with valid Title,	Repository CreateEventAsync is called with correct event	Pass

			Description, Date, UserId		
TC-PE-006	GetPersonalEventByIdAsync_ReturnsCorrectEvent	Verifies retrieval of a personal event by ID returns the correct event	Id = 5	A non-null Personal Event with Id = 5, Title = “Test”	Pass
TC-PE-007	GetPersonalEventsCountAsync_ReturnsCorrectCount	Ensures total personal event count is retrieved accurately.	(none)	Integer value 42	Pass
TC-PE-008	UpdatePersonalEventAsync_CallsRepository_WithCorrectEvent	Checks that update method passes correct event data to repository.	PersonalEvent object with details	Repository is called once with correct event data	Pass
TC-PE-009	GetRecentPersonalEventsAsync_ReturnsCorrectDtos	Verifies recent events are retrieved and usernames are mapped correctly.	limit = 2	List of 2 DTOs with correct usernames mapped	Pass
TC-PE-010	SoftDeleteEventAsync_ReturnsFalse_WhenEventIsNull	Confirms method returns false when event does not exist.	eventId = 99, userId = 1	false	Pass
TC-PE-011	SoftDeleteEventAsync_ReturnsFalse_WhenEventIsDeleted	Confirms soft delete returns false if event is already deleted.	eventId = 99, event is already deleted	false	Pass
TC-PE-012	SoftDeleteEventAsync_ReturnsFalse_WhenUserIsNotOwner	Ensures non-owners cannot soft delete the event.	eventId = 99, userId = 1, event.UserId = 2	false	Pass

TC-PE-013	SoftDeleteEventAsync_SoftDeletes_WhenConditionsAreMet	Tests valid deletion scenario and repository interactions.	eventId = 1, userId = 1, event is valid	True, verifies invite and event repository methods were called	Pass
TC-PE-014	GetEventsWithCategoryNamesAsync_FiltersOutPastEvents_And_MapsCorrectly	Verifies only future events are returned and mapped with categories/usernames.	userId = 1, mixed event dates	Returns only future event with correct CategoryName and CreatorUsername	Pass
TC-PE-015	GetAllManageEventsAsync_ReturnsMappedDtos	Checks complete mapping of events for admin management view.	(mocked lists of events, users, invites)	List with event DTO containing correct title, owner, category, and participant usernames	Pass
TC-PE-016	GetGlobalEventsWithCategoryAsync_MapsCorrectly	Verifies global events are mapped correctly with category and username.	year = 2025, month = 5	Returns 1 event with title "Global Event", category "Celebration", creator "global_creator"	Pass
TC-PE-017	UpdateEventFromAdminAsync_UpdatesAndReturnsMappedDto	Ensures admin update modifies fields and returns the updated DTO.	EditEventDto with new title/desc/date/category	Returns updated DTO with new title, updated category name, and owner username	Pass
TC-PE-018	GetAcceptedInvitedEventsAsync_ReturnsCorrectDtos	Ensures only accepted invited events are returned and mapped.	userId = 2, valid event with accepted invite	Returns event with title "Invited Event", category "Work", is invited, creator = "alice", 2 participants	Pass
TC-PE-019	GetAcceptedInvitedEventsAsync_FiltersOutWrongMonth	Verifies events in other months are not returned.	userId = 2, event from past month	Returns an empty list	Pass

TC-PE-020	GetParticipantsByEventIdAsync_ReturnsFilteredParticipants	Ensures only participants with accepted status are returned.	eventId = 10, mixed invite statuses	Returns list with users Alice and Charlie (accepted invites only)	Pass
TC-PE-021	GetEventsWithCategoryNamesAsync_AssignsUncategorized_WhenCategoryIsMissing	Ensures missing categories are labeled as "Uncategorized".	CategoryId = 99 (no match in Db)	Returned event has CategoryName = "Uncategorized"	Pass
TC-PE-022	GetEventsWithCategoryNamesAsync_SkipsDeletedCreators	Verifies that events with deleted creators are excluded from the result.	UserId = 99, IsDeleted = true	No events returned	Pass
TC-PE-023	GetEventsWithCategoryNamesAsync_ShouldHandleNullCategoryId	Ensures events with null category are handled and assigned default name.	CategoryId = null	Returned event has CategoryName = "Uncategorized"	Pass
TC-PE-024	GetGlobalEventsWithCategoryAsync_ShouldHandleMissingCreator	Ensures events with unknown user ID assign "Unknown" as username.	UserId = 999, not in user list	Returned event has CreatorUsername = "Unknown"	Pass

5.2. PersonalEventReminderService Tests

Test Case ID	Test Method Name	Description	Input(s)	Expected Outputs	Status

TC-PER-001	Constructor_ThrowsException_WhenPersonalEventReminderRepositoryIsNull	Throws if PersonalEventReminderRepository is null	null, valid	ArgumentNullException is thrown	Pass
TC-PER-002	Constructor_ThrowsException_WhenPersonalEventRepositoryIsNull	Throws if PersonalEventRepository is null	valid, null	ArgumentNullException is thrown	Pass
TC-PER-003	CreatePersonalEventReminderAsync_Throws_WhenReminderIsNull	Throws if the reminder object is null	null reminder, valid userId	ArgumentNullException is thrown	Pass
TC-PER-004	CreatePersonalEventReminderAsync_Throws_WhenTitleIsEmpty	Throws if the reminder title is empty	reminder with empty title	InvalidReminderInputException is thrown	Pass
TC-PER-005	CreatePersonalEventReminderAsync_Throws_WhenDateIsInPast	Throws if the reminder date is in the past	reminder with past date	InvalidReminderInputException is thrown	Pass
TC-PER-006	CreatePersonalEventReminderAsync_Throws_WhenUserHasNoAccess	Throws if user has no access to event	userId = 1, access = false	UnauthorizedReminderAccessException is thrown	Pass
TC-PER-007	CreatePersonalEventReminderAsync_CallsRepository_WithCorrectReminder	Checks repository is called with correct reminder data	valid reminder, userId = 1	Repository method is called once	Pass

TC-PER-008	GetPaginatedFilteredPersonalRemindersAsync_Throws_WhenPageLessThanOne	Throws if page is less than 1	page = 0	ArgumentException is thrown	Pass
TC-PER-009	GetPaginatedFilteredPersonalRemindersAsync_FiltersBySearchTerm	Returns filtered reminders matching the search term	search = 'dentist'	1 reminder with title 'Dentist Appointment'	Pass
TC-PER-010	GetPaginatedFilteredPersonalRemindersAsync_SortsByDate	Sorts reminders by date ascending	sortBy = 'date'	Reminders sorted: A, B, C	Pass
TC-PER-011	GetPaginatedFilteredPersonalRemindersAsync_FallsBackToDefaultSort_WhenSortByInvalid	Falls back to default sort when sortBy is invalid	sortBy = 'fail', valid reminders	Sorted by ID: first ID = 1, last ID = 3	Pass
TC-PER-012	GetPaginatedFilteredPersonalRemindersAsync_ReturnsCorrectPageSlice	Returns reminders for the correct page slice	page = 2, pageSize = 1, 3 reminders	Title = 'Second', TotalPages = 3, CurrentPage = 2	Pass
TC-PER-013	GetPaginatedFilteredPersonalRemindersAsync_ReturnsEmpty_WhenNoReminders	Returns empty list when no reminders exist	empty reminder list	0 reminders, TotalPages = 1	Pass
TC-PER-014	GetPaginatedFilteredPersonalRemindersAsync_Throws_WhenPageSizeIsZero	Throws if page size is zero	pageSize = 0	ArgumentException is thrown	Pass

TC-PER-015	GetPaginatedFilteredPersonalRemindersAsync_SearchIsCaseInsensitive	Search should be case insensitive	search = 'DENTIST'	Returns reminder with title 'Dentist Appointment'	Pass
TC-PER-016	MarkPersonalEventReminderAsReadAsync_Throws_WhenReminderNotFound	Throws if reminder to mark as read is not found	invalid reminderId = 99	ReminderNotFoundException is thrown	Pass
TC-PER-017	MarkPersonalEventReminderAsReadAsync_Throws_WhenUserDoesNotOwnReminder	Throws if user does not own the reminder	reminder.UserId != userId	UnauthorizedReminderAccessException is thrown	Pass
TC-PER-018	MarkPersonalEventReminderAsReadAsync_Succeeds_WhenUserOwnsReminder	Successfully marks reminder as read	valid reminderId and userId	Repository method called once	Pass
TC-PER-019	ToggleLikeAsync_Throws_WhenReminderNotFound	Throws if reminder to like is not found	invalid reminderId = 100	ReminderNotFoundException is thrown	Pass
TC-PER-020	ToggleLikeAsync_Throws_WhenUserDoesNotOwnReminder	Throws if user does not own the reminder being liked	reminder.UserId != userId	UnauthorizedReminderAccessException is thrown	Pass
TC-PER-021	ToggleLikeAsync_Unlikes_WhenReminderIsAlreadyLiked	Unlikes reminder when it's already liked	IsLiked = true	Result = false, Unlike method called once	Pass

TC-PER-022	ToggleLikeAsync_Likes_WhenReminderIsNotLiked	Likes reminder when it's not liked	IsLiked = false	Result = true, Like method called once	Pass
TC-PER-023	HasUnreadRemindersForTodayAsync_ReturnsTrue_WhenRepositoryReturnsTrue	Returns true if repository returns true	userId = 1	Result = true	Pass
TC-PER-024	HasUnreadRemindersForTodayAsync_ReturnsFalse_WhenRepositoryReturnsFalse	Returns false if repository returns false	userId = 1	Result = false	Pass
TC-PER-025	GetLikedReminderCountAsync_ReturnsCorrectCount_WhenRemindersExist	Returns count of liked reminders when reminders exist	2 liked reminders	Count = 2	Pass
TC-PER-026	GetLikedReminderCountAsync_ReturnsZero_WhenNoReminders	Returns 0 when no liked reminders exist	Empty list	Count = 0	Pass
TC-PER-027	CountUnreadRemindersForTodayAsync_ReturnsCorrectCount_WhenUnreadRemindersExist	Returns count of unread reminders	2 unread reminders	Count = 2	Pass
TC-PER-028	CountUnreadRemindersForTodayAsync_ReturnsZero_WhenNoUnreadReminders	Returns 0 when no unread reminders exist	Empty list	Count = 0	Pass

TC-PER-029	GetPaginatedLikedRemindersAsync_ReturnsEmpty_WhenNoReminders	Returns empty list and 1 total page when no liked reminders exist	Empty list	0 reminders, TotalPages = 1	Pass
TC-PER-030	GetPaginatedLikedRemindersAsync_Throws_WhenPageIsLessThanOne	Throws when page number is less than one	page = 0	ArgumentException is thrown	Pass
TC-PER-031	GetPaginatedLikedRemindersAsync_Throws_WhenPageSizeIsZero	Throws when page size is zero	pageSize = 0	ArgumentException is thrown	Pass
TC-PER-032	GetPaginatedLikedRemindersAsync_AppliesSearchFilter	Filters liked reminders by search term	search = 'eggs'	1 result with title 'Cook Eggs'	Pass
TC-PR-033	GetPaginatedLikedRemindersAsync_SortsByDate	Sorts liked reminders by date ascending	sortBy = 'date'	First = 'A', Last = 'C'	Pass
TC-PR-034	GetPaginatedLikedRemindersAsync_ReturnsCorrectSlice	Returns reminders on the specified page	page = 2, pageSize = 1	Reminder title = '2', TotalPages = 3, CurrentPage = 2	Pass
TC-PR-035	GetPaginatedLikedRemindersAsync_FallsBackToDefaultSort_WhenSortByIsNull	Falls back to ID sort when sortBy is null	sortBy = null	First ID = 1	Pass

TC-PR-036	GetPaginatedLikedRemindersAsync_Filters_ByPersonalEventTitle	Search matches reminder’s related event title	search = 'yoga'	Reminder title = 'Reminder 1'	Pass
TC-PR-037	GetPaginatedLikedRemindersAsync_DoesNotThrow_WhenPersonalEventIsNull	Skips reminders with null PersonalEvent safely	reminder.PersonalEvent = null	No exception, result count = 0	Pass

5.3. UserService Tests

Test Case ID	Test Method Name	Description	Input(s)	Expected Outputs	Status
TC-US-001	Constructor_ShouldThrowException_WhenUserRepositoryIsNull	Throws if UserRepository is null	UserRepository = null	ArgumentNullException is thrown	Pass
TC-US-002	Constructor_ShouldThrowException_WhenInviteRepositoryIsNull	Throws if InviteRepository is null	InviteRepository = null	ArgumentNullException is thrown	Pass
TC-US-003	Constructor_ShouldThrowException_WhenRoleRepositoryIsNull	Throws if RoleRepository is null	RoleRepository = null	ArgumentNullException is thrown	Pass

TC-US-004	LoginAsync_ShouldReturnUser_WhenPasswordIsCorrect	Returns user if credentials are valid	Valid username, correct password	User object is returned	Pass
TC-US-005	LoginAsync_ShouldReturnNull_WhenPasswordIsIncorrect	Returns null if password is wrong	Valid username, wrong password	Null is returned	Pass
TC-US-006	LoginAsync_ShouldReturnNull_WhenUserDoesNotExist	Throws if user does not exist	Non-existent username	InvalidLoginInputException is thrown	Pass
TC-US-007	LoginAsync_ShouldReturnNull_WhenPasswordIsEmpty	Throws if password is empty	Empty password	InvalidLoginInputException is thrown	Pass
TC-US-008	LoginAsync_ShouldReturnNull_WhenLoginInputIsEmpty	Throws if login input is empty	Empty login input	InvalidLoginInputException is thrown	Pass

TC-US-009	LoginAsync_ShouldReturnUser_WhenUsernameHasDifferentCasing	Returns user if username casing differs	Username: 'MITKO', correct password	User object is returned	Pass
TC-US-010	LoginAsync_ShouldReturnUser_WhenEmailHasDifferentCasing	Returns user if email casing differs	Email: 'MITKO@GMAIL.COM', correct password	User object is returned	Pass
TC-US-011	LoginAsync_ShouldThrowException_WhenInputIsNullOrEmpty	Throws when both login input and password are empty	login = '', password = ''	InvalidLoginInputException is thrown	Pass
TC-US-012	LoginAsync_ShouldThrowException_WhenPasswordIsInvalid	Throws when password is invalid (too short or weak)	password = 'short'	InvalidLoginInputException is thrown	Pass
TC-US-013	RegisterAsync_ShouldReturnFalse_WhenRequiredFieldsAreMissing	Throws when required registration fields are missing	All input fields empty	InvalidRegistrationInputException is thrown	Pass

TC-US-014	RegisterAsync_ShouldReturnFalse_WhenUserAlreadyExists	Throws when user already exists in DB	username = 'mitko', email = 'mitko@gmail.com'	InvalidRegistrationInputException is thrown	Pass
TC-US-015	RegisterAsync_ShouldReturnFalse_WhenRegisterFailsInRepository	Throws when repository fails to create new user	RegisterUserAsync returns 0	InvalidRegistrationInputException is thrown	Pass
TC-US-016	RegisterAsync_ShouldThrowException_WhenUsernameInvalid	Throws for invalid username (too short or too long)	username = 'a', 'aaa', very long string	InvalidRegistrationInputException is thrown	Pass
TC-US-017	RegisterAsync_ShouldSucceed_WhenUsernameIsValid	Succeeds when valid username is provided	username = 'abcd', 'validUsername'	User registered successfully	Pass
TC-US-018	RegisterAsync_ShouldThrowException_WhenFirstnameInvalid	Throws for invalid first names (too short or too long)	firstname = 'a', 'aaa', very long string	InvalidRegistrationInputException is thrown	Pass

TC-US-019	RegisterAsync_ShouldSucceed_WhenFirstnameIsValid	Succeeds when valid firstname is provided	firstname = 'abcd', 'validFirstname'	User registered successfully	Pass
TC-US-020	RegisterAsync_ShouldReturnFalse_WhenPasswordRegexFails	Throws when password does not meet regex rules	Various weak passwords	InvalidRegistrationInputException is thrown	Pass
TC-US-021	RegisterAsync_ShouldReturnTrue_ForValidPasswords	Succeeds for valid passwords	password = 'ValidPassword.123', 'Another$123', 'PassW0rd!'	User registered successfully	Pass
TC-US-022	RegisterAsync_ShouldThrowException_WhenUsernameIsEmpty	Throws when username is empty	username = ''	InvalidRegistrationInputException is thrown	Pass
TC-US-023	RegisterAsync_ShouldThrowException_WhenEmailInvalid	Throws when email is invalid format	email = 'incorrect-email'	InvalidRegistrationInputException is thrown	Pass

TC-US-024	RegisterAsync_ShouldThrowException_WhenUserAlreadyExists	Throws when user already exists	username = 'mitko', email = 'mitko@email.com'	InvalidRegistrationInputException is thrown	Pass
TC-US-025	RegisterAsync_ShouldReturnFalse_WhenPasswordTooLong	Throws when password exceeds max length	password length > userPasswordMaxLength	InvalidRegistrationInputException is thrown	Pass
TC-US-026	RegisterAsync_ShouldAssignDefaultUserRole	Assigns default User role on successful registration	Valid user info	RoleId = Role.User, result = true	Pass
TC-US-027	RegisterAsync_ShouldReturnFalse_WhenEmailTooLong	Throws when email exceeds max length	email length > userEmailMaxLength	InvalidRegistrationInputException is thrown	Pass
TC-US-028	RegisterAsync_ShouldReturnTrue_WhenUserIsSuccessfullyCreated	Verifies full user data and registration process	Valid input, lastname = null	User registered with correct data	Pass

TC-US-029	RegisterAsync_ShouldReturnTrue_WhenLastnameIsNull	Allows null lastname on registration	lastname = null	User registered successfully	Pass
TC-US-030	RegisterAsync_ShouldLowercaseUsernameAndEmail	Ensures username and email are stored lowercase	username = 'MiTkO', email = 'Mitko@Email.com'	Username and Email lowercased in DB	Pass
TC-US-031	GetUserByIdAsync_ShouldReturnUser_WhenExists	Returns user if found by ID	id = 1	User object with Username = 'mitko'	Pass
TC-US-032	GetUserByIdAsync_ShouldReturnNull_WhenUserNotFound	Returns null if user not found by ID	id = 99	Null is returned	Pass
TC-US-033	GetUserByIdAsync_ShouldThrowException_WhenIdIsInvalid	Throws for invalid user ID (0 or negative)	id = 0 or -1	ArgumentOutOfRangeException is thrown	Pass

TC-US-034	GetUserByUsernameAsync_ShouldReturnNull_WhenUserNotFound	Returns null if username not found	username = 'mitko'	Null is returned	Pass
TC-US-035	GetUserByUsernameAsync_ShouldReturnUser_WhenExists	Returns user if username found	username = 'mitko'	User object with Username = 'mitko'	Pass
TC-US-036	GetUserByUsernameAsync_ShouldThrowException_WhenUsernameInvalid	Throws when username is empty	username = ''	ArgumentException is thrown	Pass
TC-US-037	SoftDeleteUserAsync_ShouldCallAutoDecline_WhenDeletionSucceeds	Calls AutoDecline when soft delete is successful	SoftDeleteUserAsync returns true	AutoDeclineInvites called once, result = true	Pass
TC-US-038	SoftDeleteUserAsync_ShouldNotCallAutoDecline_WhenDeletionFails	Does not call AutoDecline when soft delete fails	SoftDeleteUserAsync returns false	AutoDeclineInvites not called, result = false	Pass

TC-US-039	GetAllUsersAsync_ShouldReturnMappedDtos	Returns list of mapped User DTOs with role and status	GetAllUsersAsync returns 2 users	Mapped DTOs with correct RoleName and Status	Pass
TC-US-040	GetRecentUsersAsync_ShouldReturnRecentDtos	Returns recent user DTOs	count = 2	DTOs with Username 'a' and 'b'	Pass
TC-US-041	GetUserCountAsync_ShouldReturnCorrectCount	Returns total number of users	None	count = 42	Pass
TC-US-042	UpdateUserAsync_ShouldPassCorrectUserToRepository	Updates user and verifies correct data passed to repository	UserEditDto with Id=1, Username='mitko', Role='User', Status='Banned'	UpdateAsync called with correct User and role, result = true	Pass
TC-US-043	UpdateUserBanStatusAsync_ShouldCallRepositoryWithCorrectValues	Updates user ban status via repository	userId = 1, isBanned = true	UpdateUserBanStatusAsync called once with correct params	Pass

TC-US-044	GetFilteredAdminUsersAsync_ShouldReturnCorrectlyFilteredUsers	Returns filtered users based on search, role, and status	searchTerm = 'mi', role = 'User', status = 'Active'	List with 1 user 'mitko'	Pass

5.4. InviteService Tests

Test Case ID	Test Method Name	Description	Input(s)	Expected Outputs	Status
TC-IS-001	Constructor_Throws_WhenInviteRepositoryIsNull	Throws exception if InviteRepository is null	null, valid UserRepository	ArgumentNullException	Pass
TC-IS-002	Constructor_Throws_WhenUserRepositoryIsNull	Throws exception if UserRepository is null	valid InviteRepository, null	ArgumentNullException	Pass
TC-IS-003	CountPendingInvitesAsync_ReturnsCorrectCount	Returns count of pending invites for user	userId = 5	2	Pass

TC-IS-004	AutoDeclineExpiredInvitesAsync_UpdatesAllStatuses	Auto-declines all expired pending invites	expired invites with Ids 1 and 2	UpdateInviteStatusAsync called with (1, 3) and (2, 3)	Pass
TC-IS-005	CreateOrResetInviteAsync_Throws_WhenUserIsNullOrDeleted	Throws if invited user does not exist or is deleted	invite.InvitedUserId = 100 (non-existent)	InvalidOperationException	Pass
TC-IS-006	CreateOrResetInviteAsync_ReturnsExpectedResult_WhenUserExists	Creates or resets invite if user exists	invite.InvitedUserId = 42 (exists)	Returns InviteActionResult.Created	Pass
TC-IS-007	GetPaginatedFilteredInvitesAsync_ReturnsCorrectPaginationAndFiltersKickedOut	Returns paginated and filtered invites, excludes kicked out	userId = 1, statusId = 1, page = 1, pageSize = 2	2 invites returned, kicked out invite excluded	Pass
TC-IS-008	GetInviteAsync_ReturnsNull_WhenInviteNotFound	Returns null if invite not found	eventId = 1, invitedUserId = 2 (non-existent)	null	Pass

TC-IS-009	GetInviteAsync_ReturnsDto_WhenInviteExists	Returns InviteDto if invite and user exist	eventId = 1, invitedUserId = 2	InviteDto with Id = 10, Status = 'Pending'	Pass
TC-IS-010	DeleteInviteAsync_CallsRepositoryWithCorrectArguments	Deletes invite by calling repository with correct params	eventId = 5, invitedUserId = 12	SoftDeleteInviteAsync called once with correct params	Pass
TC-IS-011	LeaveEventAsync_CallsRepositoryWithCorrectArguments	Calls repository to mark invite as left	userId = 7, eventId = 21	MarkInviteAsLeftAsync called once with (7, 21)	Pass
TC-IS-012	InviteExistsAsync_ReturnsTrue_WhenInviteExists	Returns true if invite exists	eventId = 3, invitedUserId = 14	true	Pass
TC-IS-013	InviteExistsAsync_ReturnsFalse_WhenInviteDoesNotExist	Returns false if invite doesn't exist	eventId = 3, invitedUserId = 14	false	Pass

TC-IS-014	HasUserAcceptedInviteAsync_ReturnsTrue_WhenAccepted	Returns true if user has accepted invite	userId = 5, eventId = 10	true	Pass
TC-IS-015	HasUserAcceptedInviteAsync_ReturnsFalse_WhenNotAccepted	Returns false if user hasn't accepted invite	userId = 5, eventId = 10	false	Pass
TC-IS-016	HasPendingInvitesAsync_ReturnsTrue_WhenInvitesExist	Returns true if pending invites exist	userId = 8	true	Pass
TC-IS-017	HasPendingInvitesAsync_ReturnsFalse_WhenNoInvites	Returns false if no pending invites	userId = 8	false	Pass
TC-IS-018	GetAllInvitesByUserIdAsync_ReturnsExpectedInvites	Returns all invites for a user	userId = 1	2 invites returned	Pass

TC-IS-019	GetInvitesByUserAndStatusAsync_ReturnsFilteredList	Returns filtered invites based on status	userId = 1, statusId = 2	List of invites with StatusId = 2	Pass
TC-IS-020	UpdateInviteStatusAsync_CallsRepository	Updates invite status via repository	inviteId = 99, statusId = 3	UpdateInviteStatusAsync called once with (99, 3)	Pass
TC-IS-021	GetPaginatedFilteredInvitesAsync_Throws_WhenPageIsInvalid	Throws ArgumentException for invalid page number	page = 0	ArgumentException	Pass

5.5. StatusService Tests

Test Case ID	Test Method Name	Description	Input(s)	Expected Outputs	Status
TC-IS-001	Constructor_ShouldThrowException_WhenRepositoryIsNull	Throws ArgumentNullException if StatusRepository is null	null	ArgumentNullException	Pass

TC-SS-002	GetAllStatusOptionsAsync_ShouldReturnList_WhenRepositoryReturnsOptions	Returns list of status options	Repository returns 3 options	List with 3 status options (Accepted, Declined, Pending)	Pass
TC-SS-003	GetAllStatusOptionsAsync_ShouldReturnEmptyList_WhenRepositoryReturnsEmpty	Returns empty list if repository returns none	Repository returns empty list	Empty list	Pass
TC-SS-004	GetAllStatusOptionsAsync_ShouldThrowException_WhenRepositoryThrows	Throws StatusRetrievalException if repository fails	Repository throws exception	StatusRetrievalException	Pass

5.6. CountryService Tests

Test Case ID	Test Method Name	Description	Input(s)	Expected Outputs	Status
TC-CS-001	Constructor_ShouldThrowException_WhenRepositoryIsNull	Throws ArgumentNullException if CountryRepository is null	null	ArgumentNullException	Pass

TC-CS-002	GetCountriesByContinentIdAsync_ShouldReturnSortedList_WhenRepositoryReturnsUnsorted	Returns sorted list of countries by name	ContinentId = 1, unsorted list	Sorted list: Albania, Brazil, Zimbabwe	Pass
TC-CS-003	GetCountriesByContinentIdAsync_ShouldReturnEmptyList_WhenRepositoryReturnsEmpty	Returns empty list if no countries found	ContinentId = 99	Empty list	Pass
TC-CS-004	GetCountriesByContinentIdAsync_ShouldThrowException_WhenRepositoryThrows	Throws CountryRetrievalException when repository fails	ContinentId = 1, repository throws exception	CountryRetrievalException	Pass
TC-CS-005	GetCountriesByContinentIdAsync_ShouldThrow_WhenContinentIdIsZeroOrNegative	Throws ArgumentOutOfRangeException for invalid continent ID	ContinentId = -1, -15, 0	ArgumentOutOfRangeException	Pass
TC-CS-006	GetCountriesByContinentIdAsync_ShouldIgnoreNullOrWhitespaceNames	Ignores countries with null or whitespace names	ContinentId = 1, list with some invalid names	List: France, Germany	Pass

5.7. ContinentService Tests

Test Case ID	Test Method Name	Description	Input(s)	Expected Outputs	Status
TC-CONS-001	Constructor_ShouldThrowException_WhenRepositoryIsNull	Throws ArgumentNullException if ContinentRepository is null	null	ArgumentNullException	Pass
TC-CONS-002	OrderContinentByNameAsync_ShouldReturnSortedList_WhenRepositoryReturnsUnsorted	Returns sorted list of continents by name	Unsorted list of 3 continents	Sorted list: Africa, Asia, Europe	Pass
TC-CONS-003	OrderContinentByNameAsync_ShouldReturnEmptyList_WhenRepositoryReturnsEmpty	Returns empty list if no continents found	Empty list	Empty list	Pass
TC-CONS-004	OrderContinentByNameAsync_ShouldThrowException_WhenRepositoryThrows	Throws ContinentRetrievalException when repository fails	Repository throws exception	ContinentRetrievalException	Pass
TC-CONS-005	OrderContinentByNameAsync_ShouldIgnoreNullOrWhitespaceNames	Ignores continents with null or whitespace names	List with some null/whitespace names	List: Africa, Europe	Pass

5.8. CategoryService Tests

Test Case ID	Test Method Name	Description	Input(s)	Expected Outputs	Status
TC-CATS-001	Constructor_ShouldThrowException_WhenRepositoryIsNull	Throws ArgumentNullException if CategoryRepository is null	null	ArgumentNullException	Pass
TC-CATS-002	GetAllCategoriesAsync_ShouldReturnSortedList_WhenRepositoryReturnsUnsorted	Returns sorted list of categories by name	Unsorted list of 3 categories	Sorted list: Outside, Personal, Work	Pass
TC-CATS-003	GetAllCategoriesAsync_ShouldReturnEmptyList_WhenRepositoryReturnsEmpty	Returns empty list if no categories found	Empty list	Empty list	Pass
TC-CATS-004	GetAllCategoriesAsync_ShouldThrowException_WhenRepositoryFails	Throws CategoryRetrievalException when repository fails	Repository throws exception	CategoryRetrievalException	Pass
TC-CATS-005	GetAllCategoriesAsync_ShouldIgnoreNullOrWhitespaceNames	Ignores categories with null or whitespace names	List with some null/whitespace names	List: Family, Work	Pass

5.9. CalendarService Tests

Test Case ID	Test Method Name	Description	Input(s)	Expected Outputs	Status
TC-CALS-001	GenerateCalendar_WithPersonalEvents_ReturnsCorrectStructure	Maps personal events correctly onto calendar days	year = 2025, month = 5, 1 personal event on 15th	1 personal event added to day 15	Pass
TC-CALS-002	GenerateCalendar_WithNationalEvents_ReturnsCorrectStructure	Maps national events correctly onto calendar days	year = 2025, month = 5, 1 national event on 1st	1 national event added to day 1	Pass
TC-CALS-003	GenerateCalendar_WithoutEvents_FillsCalendarCorrectly	Creates a complete calendar structure with no events	year = 2025, month = 1	Calendar with correct year/month and complete weekly grid	Pass
TC-CALS-004	GenerateEmptyCalendar_ReturnsAllDaysWithoutEvents	Generates calendar with all days empty	year = 2025, month = 1	All days have no personal or national events	Pass
TC-CALS-005	GenerateNationalHolidayCalendarAsync_ReturnsPopulatedCalendar	Returns calendar with fetched national holidays	year = 2025, month = 5, countryId = 1	Mock Holiday shown on correct day	Pass

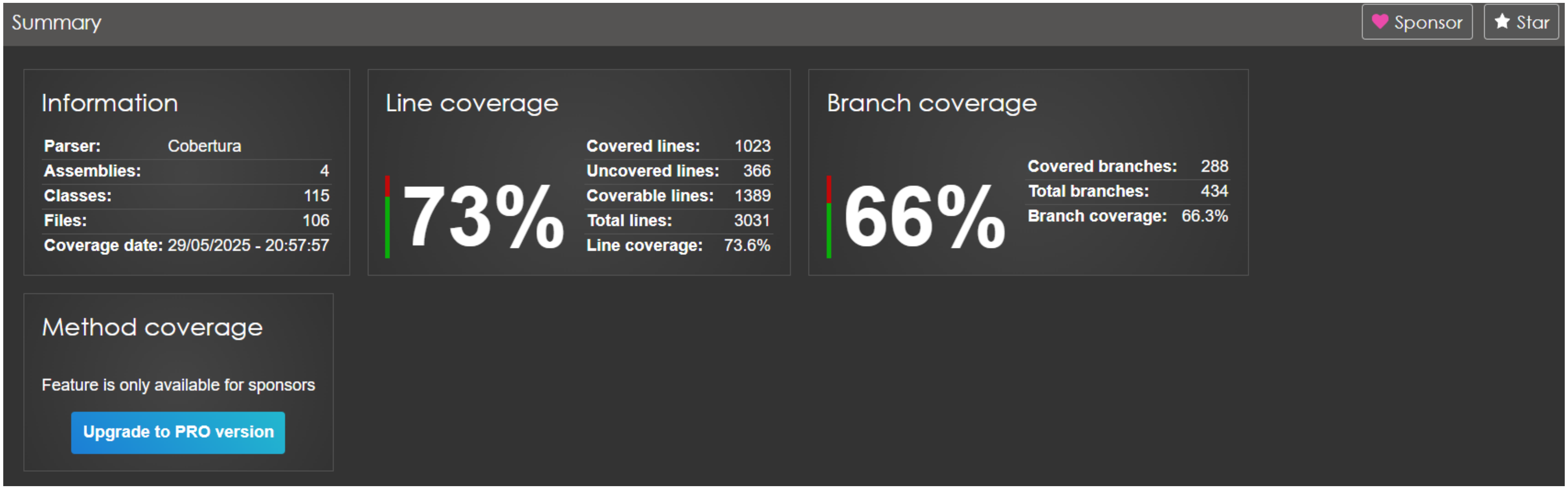
TC-CALS-006	GenerateUserCalendarAsync_ReturnsCombinedEventCalendar	Returns calendar with own, invited, and global events	year = 2025, month = 5, userId = 7	3 events appear on same day	Pass
TC-CALS-007	GenerateCalendar_AllowsMultipleEventsOnSameDay	Supports multiple personal events on a single date	2 events on June 15, 2025	2 personal events shown for day 15	Pass

5.10. CalendarNavigationService Tests

Test Case ID	Test Method Name	Description	Input(s)	Expected Outputs	Status
TC-CNAVS-001	Normalize_ReturnsCurrentYearAndMonth_WhenInputsAreNull	Returns current year and month if inputs are null	year = null, month = null	Current year and month	Pass
TC-CNAVS-002	Normalize_HandlesMonthLessThanOne_ByWrappingToPreviousYear	Wraps month 0 to previous year December	year = 2025, month = 0	year = 2024, month = 12	Pass

TC-CNAVS-003	Normalize_HandlesMonthGreaterThanTwelve_ByWrappingToNextYear	Wraps month 13 to next year January	year = 2025, month = 13	year = 2026, month = 1	Pass
TC-CNAVS-004	Normalize_ReturnsSameYearAndMonth_WhenWithinBounds	Returns same values if month is valid	year = 2025, month = 7	year = 2025, month = 7	Pass

6. Screenshots / Evidence



Information

Parser:	Cobertura
Assemblies:	4
Classes:	115
Files:	106
Coverage date:	29/05/2025 - 20:57:57

Line coverage

73%

Covered lines:	1023
Uncovered lines:	366
Coverable lines:	1389
Total lines:	3031
Line coverage:	73.6%

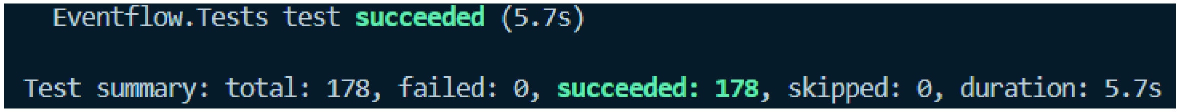
Branch coverage

66%

Covered branches:	288
Total branches:	434
Branch coverage:	66.3%

Method coverage

Feature is only available for sponsors



```
  Eventflow.Tests test succeeded (5.7s)

Test summary: total: 178, failed: 0, succeeded: 178, skipped: 0, duration: 5.7s
```

7. Conclusion

All major services were successfully tested with a high degree of code coverage. The test strategy focused on business logic validation through clean separation of concerns and mocking. The Eventflow backend is robust and ready for integration with frontend components.