Testing Document

Course: SOEN341

Lab Section - FK-X

Presented by 404 Found:

Matthew Lunt - 40137949

Kruti Patel - 40259909

Sophie Hariri - 40281961

Pawan Kumar Gupta - 40254781

Zeynep Ozcan - 40309273

Shadman Chowdhury - 40286231

Gracie Idonije - 40279628

Joshua Vilda - 4082320

Date Submission: 17 November,2025

Test Summary:

In Sprint 4, we tested all the remaining features to make sure the platform works smoothly for Students, Organizers, and Admin. We used manual unit testing, integration testing, system testing, and user acceptance testing

1. Unit Testing

For unit testing, we tested small individual functions in the system to make sure each part worked correctly on its own. This included testing event filters, search bar logic, event creation validation, friend request functions, and small data-handling functions. We used different inputs and expected outputs to confirm that each function worked as intended before connecting everything together.

Example:

Unit Test 1: Event Category Filter
Function: `filterEvents(category)`
Purpose: Check if events filter correctly based on category.

Input	Expected Output	Actual	Result
Location = "Montreal"	Only Montreal events should show	Correct	Pass
Price = Min = "50" & Max = " 100"	Only this price range event should show	Correct	Pass
Organization= "xvy"	No event show	Correct	Pass

Result: PASS

Unit Test 2: Student Register for Event
Function: `registerForEvent(eventID, userID)`
Purpose: Check registration logic works properly.

Input	Expected Output	Actual	Result

No Registered Account	Firebase: Error (auth/invalid-login-credentials)	Correct	Pass
Already registered	Login Successfully	Correct	Pass
Wrong email address , example: without "@"	Pop up "please include "@""	Correct	Pass

Result: PASS

Unit Test 3: **Event Creation Field Validation**
Function: `validateEventForm(title, date,...)`
Purpose: Make sure the organizer cannot create invalid events.

Input	Expected Output	Actual	Result
Just Title add and the click publish event	Show "Title, location, and date are required."	Correct	Pass
Missing date	Show "Title, location, and date are required."	Correct	Pass
All fields correct	Preview Show	Correct	Pass

Result: PASS

2. Integration Testing

Integration testing was performed to check that different components of the platform work correctly together. We tested the connection between the front-end pages and Firestore, the flow from event creation to organizer analytics, the integration of the friend system with the recommendation engine, and the admin tools with the main database. All integrated parts worked smoothly without errors.

Examples:

Integration Test 1 – Event Creation
Purpose: Check if the event list on the Student Discovery Page loads correctly

Step	Expected Result	Actual Result
Student go to search	Event Discovery Page Appeared	As Expected
Student Filter the Events	Showing event according to the filter	As Expected
Click on "Buy Tickets"	QR Code Appeared	As Expected
Scan the QR Code using scanner or phone	Green "Ticket Verified" Page Appeared	As Expected

Result: PASS
Integrated components: Front-end UI + Firestore Database

Integration Test 2: Event Creation Field Validation + Search
Purpose: Verify that events created by the organizer appear correctly in the search

Step	Expected Result	Actual Result
Organizer creates a new event	Event created successfully	As Expected
Click on Publish Event	Event publish successfully	As Expected
Go to the Search page	Show new added event	As Expected

Result: PASS
Integrated components: Event Creation Form + Search Page

3. System Testing

System testing was done to verify the complete end-to-end flow of the platform. We tested full scenarios such as a student discovering and registering for events, an organizer creating events and checking analytics, and an admin managing users and platform data. The entire system worked correctly as a whole, and all main workflows were completed without issues.

Examples:

System Test 1 – Full Student Event Flow
Scenario: Student logs in → discovers events → views details → registers → sees friends attending.

Step	Expected Result	Actual Result
1.Student logs in	Dashboard loads successfully	As Expected
2. Student opens Event Discovery	All events load	As Expected
Student Filter the Events	Showing event according to the filter	As Expected
Click on "Buy Tickets"	QR Code Appeared	As Expected
Scan the QR Code using scanner or phone	Green "Ticket Verified" Page Appeared	As Expected

Result: PASS
Purpose: Confirms the entire student journey works properly.

System Test 2 – Organizer Event Management Flow
Scenario: Organizer logs in → creates an event → checks dashboard → views analytics → uses tools.

Step	Expected Result	Actual Result
1. Organizer logs in	Organizer Dashboard loads	As Expected

Step	Expected Result	Actual Result
2. Organizer creates event	Event Create successfully	As Expected
3. Organizer search shows new event	Event visible	As Expected
4. Organizer opens analytics	Correct charts and stats visible	As Expected
5. Organizer uses Tools (Export/Share)	Organizer uses Tools (Export/Share)	As Expected

Result: PASS
Purpose: Validates organizer workflow from event creation to analysis.

System Test 3 – Admin Full Control Flow

Scenario: Login → Manage Accounts → Moderate Events → View Analytics → Generate Report

Step	Expected Result	Actual Result
1. Admin logs in	Admin Dashboard loads	As Expected
2. Admin opens Admin Management Page	Pending approvals, active organizers, and total organizers numbers load correctly	As Expected
3. Admin views organizer accounts list	Change applied globally	As Expected
4. Admin approves or rejects an organizer	Correct charts and stats visible	As Expected
5. Admin switches to Event Moderation tab	Organizer uses Tools (Export/Share)	As Expected
6. Admin switches to Reports & Issues	User reports are visible	As Expected
7. Admin opens Admin Analytics Page	Global analytics dashboard loads and shows total events,	As Expected

	tickets, revenue, active users, etc.	
8. Admin changes date range	Analytics refreshes and updates numbers	As Expected
9. Admin clicks Generate Report	System creates a downloadable report file	As Expected

Result: PASS
Purpose: Confirms the admin has full functional control over the platform.

System Test 4 – Friend System & Recommendations
Scenario: Add friend → accept request → see recommended events.

Step	Expected Result	Actual Result
1.User opens the Add Friend page	Page loads with search bar	As Expected
2. User searches for a friend by name or email	Correct user appears in search results	As Expected
3. User sends a friend request	Request is sent and status becomes "Pending"	As Expected
4. Friend accepts the request	Status updates to "Friends"	As Expected
5. User opens Recommended Events page	Events based on friends' attendance appear	As Expected
6. User opens an event	"Friends Attending" section loads	As Expected

Result: PASS
Purpose: Friends function works correctly with system

4. User Acceptance Testing

For User Acceptance Testing, we made one test case for each user story and also for our extra features like the friend system and event recommendations. Each test case included simple steps, what we expected to happen, and what actually happened. All 10 user stories and the extra features worked correctly, so every UAT test passed.

Each user story was tested using a simple acceptance test format with:

- Steps

- Expected results

- Actual results

All 10 user stories passed the tests.

Regression Testing:

After finishing the Sprint 4 features, we performed regression testing to make sure that none of the new updates affected the existing functions of the platform. We re-tested important features like student login, event discovery, event registration, organizer event creation, analytics loading, and admin management tools. All previously working features continued to function correctly, and no new bugs were introduced. This shows that the new Sprint 4 changes did not break any old functionality.

Acceptance Test Cases (US01 – US10):

Core Feature 1: Student Event Experience

US01: Student Event Discovery

Test Case: US01
Pre-Conditions: - Events exist in the database. - App backend and frontend are running.

Steps	Expected Results	Actual Results
1. Open the Event Discovery page	All events are displayed	As Expected
2. Apply a date/time filter	Only events matching the date/time appear	As Expected
2. Apply filters for price, location, organization, or type	Events update to match selected filters	As Expected
Result: Pass		

US02: Student Event Management

Test Case: US02		
Pre-Conditions: - Events exist in the database. - User is logged into the application.		
Steps	**Expected Results**	**Actual Results**
1. Open the Event Discovery page	Each event card displays a "Save" button	As Expected
2. Click the "Save" button on any event	The event is saved to the user's profile and synced to their calendar	As Expected
3. Click "Claim Ticket" on a saved event	A unique ticket ID is generated for that event	As Expected
4. Download or view the ticket	A digital ticket with a unique QR code is displayed or downloaded successfully	As Expected
Result: Pass		

Core Feature 2: Organizer Event Management

US03: Organizer Event Creation

Test Case: US03		
Pre-Conditions: - Organizer account exists and is logged in. - "Create Event" page is accessible. - Firebase backend is connected.		
Steps	**Expected Results**	**Actual Results**
1. Open the "Create Event" form.	The event creation form loads successfully	As Expected
2. Enter event title, description, date/time, location, and ticket details	All fields accept valid input without errors	As Expected
3. Ensure validation for required fields (title, date, location, etc.) works properl	Form prevents submission if any required field is missing	As Expected
4. Toggle between Free and Paid event options	The ticket type changes correctly and displays related input fields (e.g., price)	As Expected
5. Click "Preview" to review the event	The preview displays all entered details correctly	As Expected
6. Click "Publish"	Event data is saved in Firebase and visible in the database	As Expected
7. After publishing, check the Event Discovery/Search page	The newly created event appears in the public event list	As Expected
Result: Pass		

US04: Organizer Event Analytics

Test Case: US04		
Pre-Conditions: - Organizer account exists and is logged in. - Events have been created with ticket and attendance data. - Analytics dashboard page is accessible.		
Steps	**Expected Results**	**Actual Results**
1. Open the Organizer Analytics Dashboard	Dashboard loads successfully and displays event summary data	As Expected
2.Check event data for tickets sold, attendance, and total revenue	Correct values are displayed for each metric	As Expected
3. Use filters to sort by date or event	The dashboard updates to show analytics for the selected filter	As Expected
Result: Pass		

US05: Organizer Tools

Test Case: US05		
Pre-Conditions: - Organizer is logged in. - Organizer dashboard is accessible. - Demo data.		
Steps	**Expected Results**	**Actual Results**
1. Open the Organizer	Tools for CSV export and QR	As Expected

Dashboard	scanning are visible	
2. Click "Download CSV"	A sample CSV file is generated or downloaded	As Expected
3. Open QR Scanner	QR scanner page opens and camera activates	As Expected
4. Scan a test QR code	System shows "Valid" or "Invalid" based on demo data	As Expected
5. Try a few scans to check feedback	Valid and invalid messages display correctly	As Expected
Result: Pass		

Core Feature 3: Administrator Dashboard & Moderation

US06: Administrator Platform Oversight

Test Case: US06		
Pre-Conditions: - Admin is logged in. - Admin dashboard is working with demo data. - There are pending account requests and existing user accounts.		
Steps	**Expected Results**	**Actual Results**
1. Open the Admin Dashboard	Dashboard loads with account management options	As Expected
2. View pending account requests	All pending requests are shown	As Expected
3. Approve or reject a request	Approved accounts are added to the database; rejected ones are removed	As Expected

4. Open the Manage Accounts page	All user accounts are listed	As Expected
5. Refresh the dashboard	Updated account list shows correct statuses	As Expected

Result: Pass

US07: Administrator Analytics

Test Case: US07		
Pre-Conditions: - Admin account is logged in. - Analytics dashboard is accessible. - Demo data for events and participation is available		
Steps	**Expected Results**	**Actual Results**
1. Open the Admin Analytics Dashboard	Dashboard loads showing demo statistics and charts	As Expected
2. Check platform usage logs (events, tickets, participation)	Sample data appears correctly for each category	As Expected
3. View analytics dashboard for feature usage	Graphs and counts display as expected using demo data	As Expected
4. Generate a report for review	A summary report is displayed or downloaded with the same demo values	As Expected

Result: Pass

US08: Administrator Management

Test Case: US08		
Pre-Conditions: - Admin account is logged in. - Admin dashboard is accessible. - Local or demo data is available for testing		
Steps	**Expected Results**	**Actual Results**
1. Open the Admin Dashboard	Dashboard loads with organization and role management options	As Expected
2. Open the Organization Management section	List of sample organizations appears	As Expected
3. Add a new organization	Organization appears in the local list	As Expected
4. Assign roles (e.g., Organizer, User) to usersreview	Roles update visually in the UI and reflect correct assignments temporarily demo values	As Expected
5. Edit or remove an organization or role	Changes appear immediately on screen	As Expected
Result: Pass		

Core Feature 4: Friends' Event Attendance

US09: Add Friends & Event Recommendation

Test Case: US09		
Pre-Conditions: - User is logged in - Friend account exists		
Steps	**Expected Results**	**Actual Results**
1. Open "Add Friend" page	Page loads with search bar	As Expected
2. Search for a user	Matching user appears	As Expected
3. Send friend request	Status becomes "Pending"	As Expected
4. Friend accepts request	Status becomes "Friends"	As Expected
5. Open Recommended Events page	Recommended events show based on friend's attendance	As Expected
Result: Pass		

US10: View Friends Attending Event

Test Case: US10		
Pre-Conditions: - User is logged in - Friends exist - Some friends are attending the event		
Steps	**Expected Results**	**Actual Results**
1. Open event details page	Event information loads	As Expected
2. Scroll to "Friends	List of friends is displayed	As Expected

Attending" section		
3. Click on a friend's name	Friend profile opens	As Expected
Result: Pass		

Testing Conclusion:

All the testing activities in Sprint 4 were completed successfully. We performed unit testing, integration testing, system testing, regression testing, and user acceptance testing for all 10 user stories and extra features. All tests passed, and the platform worked smoothly without major issues. The main flows for students, organizers, and admins functioned correctly, and the new friend and recommendation features also worked as expected. Overall, the system is stable, meets the requirements.