

## Testing Summary

<i>Nagha Abdoul-Aziz Kone - 40307505</i>	<i>Abderraouf El Khalil Karoun - 40315753</i>	<i>Etienne Vorms - 40286823</i>	<i>Ibrahim Ahmad - 40279224</i>
<i>Wendy Mbog- 40259455</i>	<i>Saeed Ramez Fakhoury - 40308146</i>	<i>Madona Beshara - 40316641</i>	<i>Ryan Abdalla- 40243555</i>

### Project Information

*CampX is a Campus Events & Ticketing Web Application designed to help students discover, organize, and attend events on campus. The system enables students to browse events, save them, claim free or paid tickets, and check in using QR codes. Organizers can create and manage events, track attendance, and access analytics through dashboards, while administrators moderate content and oversee organizations.*

### Test Objective

*The objective of the tests we have conducted throughout the progression of our Web Application's development was to verify that the functional requirements defined in the User Stories were defined, but also to ensure that the Backend API was interacting correctly with our database.*

*We performed three types of testing:*

- **Acceptance Testing** – 15 acceptance tests were administered, one for each User Story, and each of them allowed us to verify that the implemented features fulfilled the requirements and satisfied their expected behavior.
- **Integration Testing** – 3 integration tests were carried out to verify that the backend components were working together correctly according to the requirement of the corresponding sub-features.
- **Unit testing** – 5 components were tested individually to ensure that they function as expected.

## **Test Summary**

Test Type	Tools Used	Planned	Executed	Passed	Failed
Acceptance Tests	Web Browser, Application Under Test	15	15	15	0
Integration Tests	Vitest test framework	3	3	3	0
Unit Tests	Vitest test framework	5	5	5	0

**Results** - All planned tests were executed flawlessly, for a combined total of 23 successful Acceptance, Integration Tests and Unit Tests.

## **Acceptance Test Overview**

- **US.01 Browsing upcoming events**
  - Verify that logged in student can browse upcoming events.
- **US.02 Find events using filters**
  - Verify that logged in student can search for events using filters.
- **US.03 Save events to calendar**
  - Verify that logged in student can save an event.
- **US.04 Claiming a ticket**
  - Verify that logged in student can claim a ticket for guaranteed access to a certain event.
- **US.05 Receiving digital ticket with QR code**
  - Verify that logged in student can receive a digital ticket with QR code for an event.
- **US.06 Create Events**
  - Verify that logged in organizer can create events by entering their details, so that students can access them.
- **US.07 Monitoring event engagement**
  - Verify that logged in organizer can monitor event engagement.
- **US.08 Scan a ticket to check validity**
  - Verify that logged in organizer can validate ticket.
- **US.09 Export attendees list in CSV**
  - Verify that logged in organizer can export attendees list in CSV format.

- **US.10 Create events**
  - Verify that logged in administrator can approve/disapprove organizer accounts.
- **US.11 Create Admin view of list of events**
  - Verify that logged in administrator can see a list of all events and moderate event listings.
- **US.12 View Global Statistics**
  - Verify that logged in administrator can view global statistics like total events, tickets issued, etc.
- **US.13 Manage organizations and assign roles**
  - Verify that logged in administrator can manage organizations and assign roles to organization members.
- **US.14 Receive reminders about events**
  - Verify that logged in student can receive notifications for specific events.

### **Integration Test Overview**

- **events.test.js**
  - Verify that event routes work as expected for creating events and reading events data
- **login.test.js**
  - Verify that login routes work as expected when a user attempts to login
- **register.test.js**
  - Verify that register routes work as expected when a user attempts to register

### **Unit Test Overview**

- **getEvents.test.js**
  - Verify that the reading event helper function works as expected
- **getOrganizations.test.js**
  - Verify that the reading organization helper function works as expected
- **handleLogin.test.js**
  - Verify that the login helper function works as expected
- **handleRegister.test.js**
  - Verify that the register helper function works as expected
- **handleCreateEvent.test.js**
  - Verify that the create event helper function works as expected

### **Defects**

*All 23 tests passed with no outstanding issues, and no defects were recorded.*