

Test Report

🕒 Created	@October 22, 2022 6:27 PM
🕒 Last Edited Time	@February 4, 2023 10:59 AM
▼ Type	Technical Spec
▼ Status	Done
👤 Created By	
👤 Last Edited By	
👥 Stakeholders	
☰ Property	

1. Introduction

1.1 Goal

1.2 Content

1.3 Organization

2. Test Plan

2.1 Testing Strategy

Motivation

2.2 Test Subjects

2.3 Black Box Testing

2.4 White Box Testing

3. Additional Tests

3.1 Security Testing

3.2 Performance - Load/Stress Testing

3.3 Acceptance Testing

4. Test Results

4.1 Unit Test Report

4.2 Performance-Load/Stress Test Results

4.3 User Acceptance Tests Results

4.4 Security Tests Results

1. Introduction

1.1 Goal

The aim of this document is dedicated to explain the test plan and test results of the project.

1.2 Content

1. Introduction

1.1 Goal

1.2 Content

1.3 Organization

2. Test Plan

2.1 Testing Strategy

Motivation

2.2 Test Subjects

2.3 Black Box Testing

2.4 White Box Testing

3. Additional Tests

3.1	Security Testing
3.2	Performance - Load/Stress Testing
3.3	Acceptance Testing
4.	Test Results
4.1	Unit Test Report
4.2	Performance-Load/Stress Test Results
4.3	User Acceptance Tests Results
4.4	Security Tests Results

1.3 Organization

In the section "Test Plan", the test plan for this project is presented.

In the section "Additional Tests", all additional tests are listed.

In the section "Test Results", all test results of both unit tests and additional tests are reported.

2. Test Plan

2.1 Testing Strategy

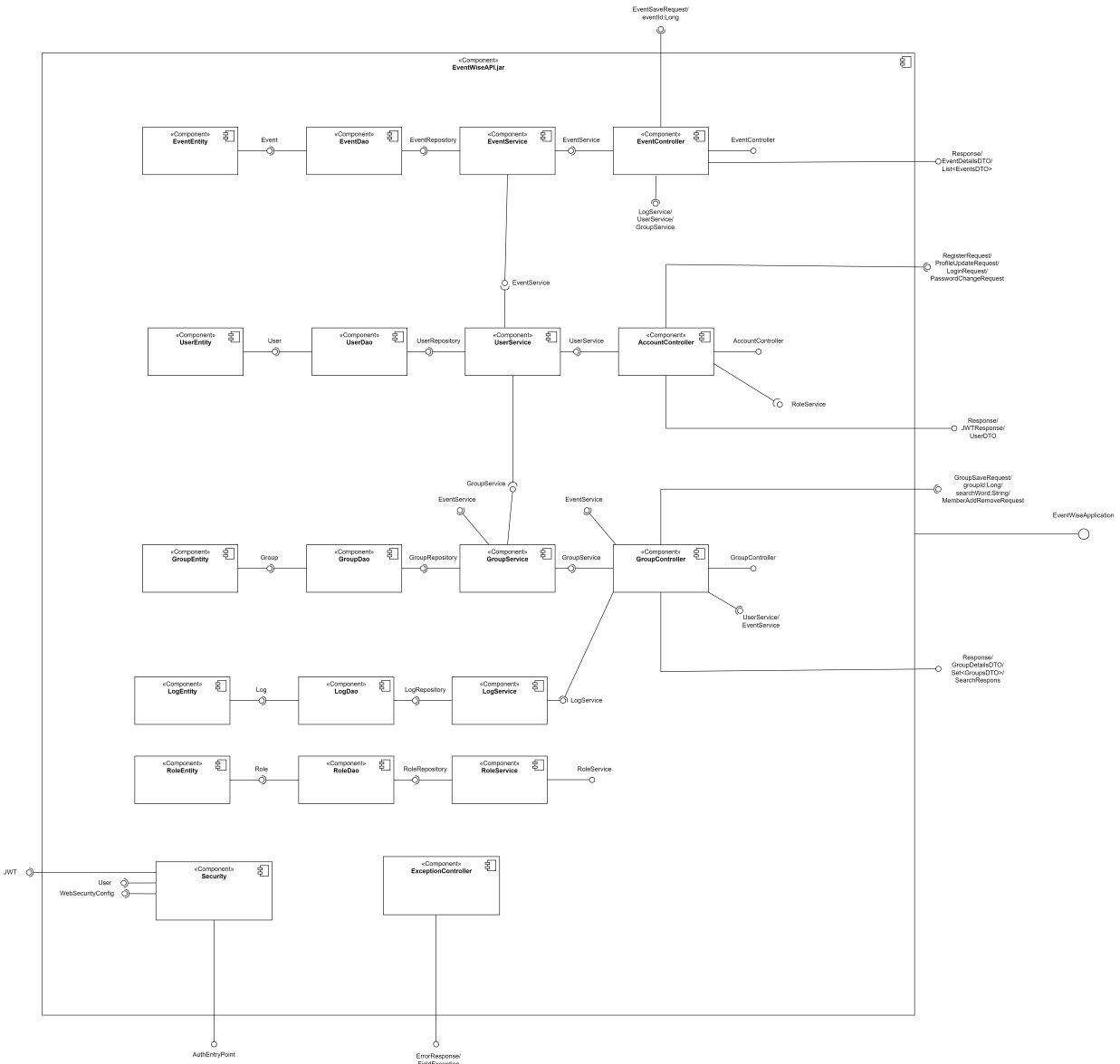
We have chosen the incremental approach (bottom-up) rather than the big-bang approach to test our main components.

We have determined to use breadth-first ordering while applying the bottom-up incremental strategy because the tested atomic parts do not depend on each other.

Motivation

- The test subjects (controllers) are independent of each other.
- Errors are easier to isolate and correct.
- Incremental approach provides more detailed observation

2.2 Test Subjects



The main subjects of the unit testing are UserController, GroupController, EventController and ExceptionController (indirectly).

Since we used a layered design for the purpose of encapsulation, all other components' functionality and correctness will be indirectly checked during the integration test.

All methods of the UserController, GroupController, and EventController are going to be tested. The exceptional inputs will trigger the ExceptionController.

2.3 Black Box Testing

Partition valid and invalid input values into equivalence classes (Equivalence Partitioning). List your test cases in a table with their description, test input, and expected output, and map them to your use cases. See course slides.

For Black Box Testing, we plan to use Espresso - Automated Testing Framework for the implementation and test report generation. With this way, we will be able to test the outputs of the system for certain inputs and see whether use cases are achieved or not.

Equivalence Partition

Aa Input	≡ Valid Equivalence Classes	≡ Invalid Equivalence Classes
<u>User - Username</u>	any unique string with size $L: 5 \leq L \leq 20$	any string with size $L: L < 5$ or $L > 20$ any string previously registered
<u>User - Email</u>	any string has Email address format with size $L: 1 \leq L \leq 50$	any string with size $L: L > 50$, any empty string or any string that has not Email format
<u>User - Displayed Name</u>	any string with size $L: 5 \leq L \leq 10$	any string with size $L: L < 5$ or $L > 10$
<u>User - Password</u>	any string with size $L: 5 \leq L$	any string with size $L: L < 5$
<u>User - Location</u>	any string with size $L: L \leq 20$	any string with size $L: L > 20$ or any empty string
<u>Group - Group Name</u>	any string with size $L: 1 \leq L \leq 20$	any string with size $L: L > 20$ or any empty string
<u>Group - Location</u>	any string with size $L: L \leq 20$	any string with size $L: L > 20$ or any empty string
<u>Group - Description</u>	any string with size $L: L \leq 500$	any string with size $L: L > 500$
<u>Event - Name</u>	any string with size $L: 1 \leq L \leq 50$	any string with size $L: L > 50$ or any empty string
<u>Event - Location</u>	any string with size $L: 1 \leq L \leq 20$	any string with size $L: L > 20$ or any empty string
<u>Event - Description</u>	any string with size $L: 1 \leq L \leq 500$	any string with size $L: L > 500$
<u>Event - Type</u>	any selected string in the dropdown list	any other string
<u>Event - DateTime</u>	any selected date-time using Date and Time Dialogs	any other method to try to choose date-time

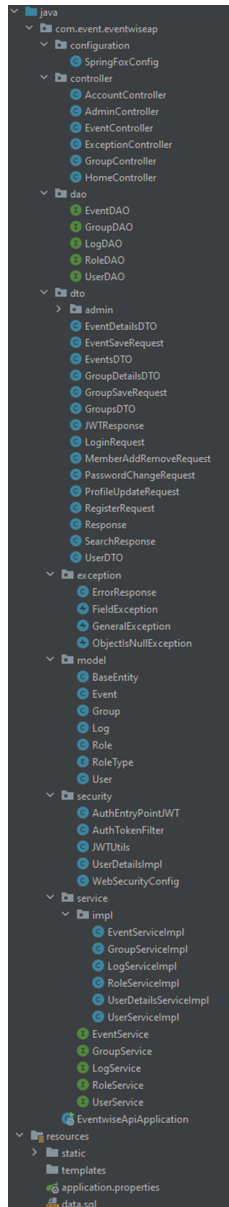
Test Cases

■ Test Number	≡ Test Name	≡ Description	≡ Input	≡ Output	≡ Related Use Case
<u>1</u>	checkSignup	Test create a new user profile	Username, Email, Displayed Name, Password, Location	Status Code, Message	Create an account
<u>2</u>	checkLogin	Test login to user profile to execute some operations	Username, Password	Status Code, Message, Credential Information (JWT)	Sign in to the System
<u>3</u>	checkCreateGroup	Test create a new group	JWT, Group Name, Location, Description	Status Code, Message	User creates a group
<u>4</u>	checkUpdateGroup	Test update group information	JWT, Group Id, Group Name, Location, Description	Status Code, Message	User updates a group information
<u>5</u>	checkCreateEvent	Test create a new event	JWT, Event Name, Location, Description, Type, DateTime	Status Code, Message	Creating an Event for a group
<u>6</u>	checkUpdateEvent	Test update event information	JWT, Event Id, Event Name, Location, Description, Type, DateTime	Status Code, Message	User updates an event information

Test Number	Test Name	Description	Input	Output	Related Use Case
<u>7</u>	checkRejectEvent	Test reject event invitation	JWT, Event Id	Status Code, Message	Provide response to an event invitation
<u>8</u>	checkAcceptEvent	Test accept event invitation	JWT, Event Id	Status Code, Message	Provide response to an event invitation
<u>9</u>	checkAddPeopleToGroup	Test add people to a group	JWT, User Id, Group Id	Status Code, Message	User adds another user to the group
<u>10</u>	checkRemovePeopleToGroup	Test remove people from a group	JWT, User Id, Group Id	Status Code, Message	User remove another user from the group
<u>11</u>	checkDeleteEvent	Test delete event	JWT, Event Id	Status Code, Message	User deletes an event
<u>12</u>	checkDeleteGroup	Test delete group	JWT, Group Id	Status Code, Message	User deletes a group
<u>13</u>	checkExitFromGroup	Test exit from a group	JWT, Group Id	Status Code, Message	User exits from a group
<u>14</u>	checkUpdateProfile	Test update user profile information	JWT, Displayed Name, Location	Status Code, Message	Edit user information
<u>15</u>	checkChangePasswordAndLogOut	Test change user password and log out from session	JWT, Password	Status Code, Message	Log Out from the System
<u>16</u>	checkDeleteAccount	Test delete user profile	JWT	Status Code, Message	User deletes associated account

2.4 White Box Testing

We used the IntelliJ IDE (by JetBrains) for the implementation and test report generation. We plan to use **WebMvcTest** to test the developed APIs and **Mockito** to create required mock dependencies.



Unit Tests

No	Name	Description/Sample Input	Required Mock	Expected Result
1	check_register_user_with_correct_user_information_should_return_OK	Correct user information given	UserService, RoleService	200, JWTResponse
2	check_register_user_with_incorrect_field_should_return_Bad_Request	One or more invalid fields/email: "not_a_proper_email@a"	UserService, RoleService	400, "Not a proper E-mail"
3	check_register_user_with_existing_username_email_should_return_Bad_Request	Create account with a username that already in use	UserService, RoleService	400, "Username is already taken"
4	check_login_with_nonexistent_account_should_return_bad_request	Try to log in with a nonexistent account	UserService	400, "There is no such an account"

No	Name	Description/Sample Input	Required Mock	Expected Result
<u>5</u>	check_login_user_with_correct_information	Correct login information given	UserService	200, JWTResponse
<u>6</u>	check_logout_should_return_OK	Try to log out	UserService, Session	200
<u>7</u>	check_profile_request_should_return_OK	Try to see the user profile	UserService, Session	200
<u>8</u>	check_change_password_with_correct_information_should_return_OK	Correct password change request	UserService, Session	200, Password change successful
<u>9</u>	check_change_password_with_incorrect_information_should_return_Bad_Request	Current password is wrong or new and confirmation passwords do not match	UserService, Session	400, "Current password is wrong" or "New and confirm passwords do not match"
<u>10</u>	check_delete_account_should_return_OK	Try to delete an existing account	UserService, Session	200, "The account has been deleted"
<u>11</u>	check_delete_account_should_return_Bad_Request	Try to delete a non-existing account	UserService, Session	400, "There is no such a user"
<u>12</u>	check_update_profile_should_return_OK	Update user profile	UserService, Session	200, "Profile updated successfully"
<u>13</u>	check_createGroup_with_field_error_should_return_bad_request	One or more invalid fields/empty group name	UserService, GroupService, Session	400, "Group name cannot be empty"
<u>14</u>	check_createGroup_without_field_error_should_return_OK	All fields are valid	UserService, GroupService, Session	200, "Group has been created successfully,"
<u>15</u>	check_updateGroup_with_null_id_should_return_bad_request	Id field is null	Session	400, "Group ID cannot be null while updating"
<u>16</u>	check_updateGroup_with_nonexistent_group_should_return_bad_request	Try to update a nonexistent group	GroupService, Session	400, "This group does not exist"
<u>17</u>	check_updateGroup_with_not_owner_should_return_bad_request	A user (not the owner) tries to update the group	UserService, GroupService, Session	400, "Only group owner can update the group"
<u>18</u>	check_updateGroup_with_correct_info_should_return_OK	All fields are proper	UserService, GroupService, Session	200, "The group (group name: name, group id: id) has been updated successfully"
<u>19</u>	check_listUserGroups_with_correct_info_should_return_OK	Fetch groups of user	UserService, Session	200, List<GroupsDTO>
<u>20</u>	check_getGroupDetails_with_nonexistent_group_should_return_bad_request	Try to see details of a nonexistent group	UserService, GroupService, Session	400, "This group does not exist"
<u>21</u>	check_getGroupDetails_with_not_group_member_should_return_bad_request	A user (not a member) tries to see the group details	UserService, GroupService, Session	400, "You are not a member of this group"

No	Name	Description/Sample Input	Required Mock	Expected Result
<u>22</u>	<code>check_getGroupDetails_with_correct_info_should_return_OK</code>	A member tries to see the group details	UserService, GroupService, EventService, LogService, Session	200, GroupDetailsDTO
<u>23</u>	<code>check_deleteGroup_with_nonexistent_group_should_return_bad_request</code>	Delete a nonexistent group	UserService, GroupService, Session	400, "This group does not exist"
<u>24</u>	<code>check_deleteGroup_with_not_owner_should_return_bad_request</code>	A user (not the owner) tries to delete the group	UserService, GroupService, Session	400, "You are not the owner of this group"
<u>25</u>	<code>check_deleteGroup_with_correct_info_should_return_OK</code>	The owner deletes the group	UserService, GroupService, Session	200, "The group name has been deleted"
<u>26</u>	<code>check_searchMember_with_nonexistent_group_should_return_bad_request</code>	Search for a user in a nonexistent group	UserService, GroupService, Session	400, "This group does not exist"
<u>27</u>	<code>check_searchMember_with_not_member_should_return_bad_request</code>	A user (not a member) tries to search for another user	UserService, GroupService, Session	400, "You are not a member of this group"
<u>28</u>	<code>check_searchMember_with_search_null_should_return_OK</code>	A member searches for another user with no input	UserService, GroupService, Session	200, SearchResponse
<u>29</u>	<code>check_searchMember_with_search_should_return_OK</code>	A member searches for another user	UserService, GroupService, Session	200, SearchResponse
<u>30</u>	<code>check_add_remove_member_with_nonexistent_group_should_return_bad_request</code>	Toggle user membership in a nonexistent group	UserService, GroupService, Session	400, "This group does not exist"
<u>31</u>	<code>check_add_remove_member_with_not_member_group_should_return_bad_request</code>	A user (not a member) tries to toggle the user membership of another user	UserService, GroupService, Session	400, "You are not a member of this group"
<u>32</u>	<code>check_add_remove_member_remove_should_return_OK</code>	Toggle (remove) user membership with a proper input	UserService, GroupService, Session	200, "Member username removed from the group"
<u>33</u>	<code>check_add_remove_member_add_should_return_OK</code>	Toggle (add) user membership with a proper input	UserService, GroupService, Session	200, "Member username added to the group"
<u>34</u>	<code>check_add_remove_member_with_nonexistent_user_should_return_false</code>	Toggle user membership for a nonexistent user	UserService, GroupService, Session	200, "There is no such a user"
<u>35</u>	<code>check_exitFromGroup_with_non_existent_group_should_return_bad_request</code>	Exit from a nonexistent group	UserService, GroupService, Session	400, "This group does not exist"
<u>36</u>	<code>check_exitFromGroup_with_not_member_return_bad_request</code>	A user exits (not a member) from the group	UserService, GroupService, Session	400, "You are not a member of this group"

No	Name	Description/Sample Input	Required Mock	Expected Result
<u>37</u>	<code>check_exitFromGroup_with_member_return_OK</code>	A member exits from the group	UserService, GroupService, Session	200, "You left the group groupname"
<u>38</u>	<code>check_createEvent_with_field_error_should_return_bad_request</code>	Create an event with an invalid field/empty event name	UserService, GroupService, Session	400, "Event name cannot be empty"
<u>39</u>	<code>check_createEvent_with_nonexistent_group_should_return_bad_request</code>	Create an event in a nonexistent group	UserService, GroupService, Session	400, "This group does not exist"
<u>40</u>	<code>check_createEvent_with_correct_info_should_return_OK</code>	Create an event in a group	UserService, GroupService, EventService, Session	200, "The event created by the user username"
<u>41</u>	<code>check_updateEvent_with_nonexistent_group_should_return_bad_request</code>	Update a nonexistent group	UserService, GroupService, Session	400, "This group does not exist"
<u>42</u>	<code>check_createEvent_with_not_member_should_return_bad_request</code>	A user (not a member) updates a group	UserService, GroupService, Session	400, "You are not a member of this group"
<u>43</u>	<code>check_createEvent_with_nonexistent_event_should_return_bad_request</code>	A member updates a nonexistent event	UserService, GroupService, EventService, Session	400, "This event does not exist"
<u>44</u>	<code>check_createEvent_with_not_organizer_should_return_bad_request</code>	A member (not the organizer) updates the event	UserService, GroupService, EventService, Session	400, "Only the organizer can update the event"
<u>45</u>	<code>check_createEvent_with_organizer_should_return_OK</code>	The organizer updates the event	UserService, GroupService, EventService, Session	200, "The event updated by the user username"
<u>46</u>	<code>check_listUserEvents_should_return_OK</code>	Fetch events of the user	UserService, GroupService, EventService, Session	200, List<EventsDTO>
<u>47</u>	<code>check_eventDetails_with_nonexistent_event_should_return_bad_request</code>	Fetch event details of a nonexistent event	UserService, EventService, Session	400, "This event does not exist"
<u>48</u>	<code>check_eventDetails_with_not_member_should_return_bad_request</code>	A user (not a member) tries to see event details	UserService, EventService, Session	400, "You are not a member of this group"
<u>49</u>	<code>check_eventDetails_with_member_should_return_OK</code>	A member tries to see event details	UserService, EventService, Session	200, EventDetailsDTO
<u>50</u>	<code>check_deleteEvent_with_nonexistent_event_should_return_bad_request</code>	Delete a nonexistent event	UserService, EventService, Session	400, "This event does not exist"
<u>51</u>	<code>check_deleteEvent_with_not_organizer_should_return_bad_request</code>	A member (not the organizer) tries to delete the event	UserService, EventService, Session	400, "Only the organizer or the group owner can delete the event"

No	Name	Description/Sample Input	Required Mock	Expected Result
52	check_deleteEvent_with_organizer_should_return_OK	The organizer deletes the event	UserService, EventService, Session	200, "The event deleted by the user username"
53	check_acceptEvent_with_nonexistent_event_should_return_bad_request	Accept a nonexistent event	UserService, EventService, Session	400, "This event does not exist"
54	check_rejectEvent_with_nonexistent_event_should_return_bad_request	Reject a nonexistent event	UserService, EventService, Session	400, "This event does not exist"
55	check_acceptEvent_with_not_member_should_return_bad_request	A user (not a member) tries to accept the event	UserService, EventService, Session	400, "You are not a member of this group"
56	check_rejectEvent_with_not_member_should_return_bad_request	A user (not a member) tries to reject the event	UserService, EventService, Session	400, "You are not a member of this group"
57	check_acceptEvent_already_accepted_should_return_bad_request	A member accepts an already accepted event	UserService, EventService, Session	400, "This event has already been accepted"
58	check_rejectEvent_already_rejected_should_return_bad_request	A member rejects an already rejected event	UserService, EventService, Session	400, "This event has already been rejected"
59	check_acceptEvent_not_accepted_should_return_OK	A member accepts an event	UserService, EventService, Session	200, "username accepted eventname"
60	check_rejectEvent_rejected_should_return_bad_request	A member rejects an event	UserService, EventService, Session	200, "username rejected eventname"

3. Additional Tests

3.1 Security Testing

We used Swagger UI for manual security tests. The objective user has following details:

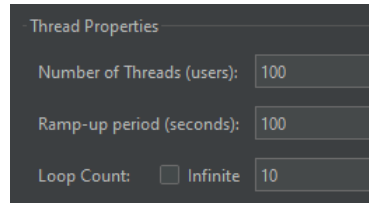
- username: securityTester
- password: security
- role: USER

SQL injection and authorizations (manual)

3.2 Performance - Load/Stress Testing

For performance testing, we preferred to use JMeter, which is an open-source desktop application based on Java. JMeter Performance Testing is a technique used to assess the capabilities of an application in terms of handling various levels of user traffic. The testing includes load testing, which simulates a high volume of user traffic, and stress testing, which involves subjecting the application to extreme levels of user traffic to identify any potential performance issues.

Our app was subjected to 100 users and to 10 loops for each user.



Tests were done for the following endpoints, which provide the most resource-demanding functionalities of our app:

- List user's event groups: GET
- Get group details: GET
- Create event: POST
- Accept event: PUT

3.3 Acceptance Testing

We use the Espresso - Automation Testing Framework to test user interactions. These interactions cover account operations, group operations, and event operations (including create, update, delete and read operations). Furthermore, we combined Black Box testing and Acceptance Testing to test effectively.

Espresso is officially supported by the official maintainer of Android and provides easy implementation on testing for user clicks to buttons and labels, keyboard typings to text fields and swipe operations on the screen. These functionalities make our job easier to react as normal users.

4. Test Results

4.1 Unit Test Report

- Account Controller: 12 test cases (12 passes) and 100% coverage

AccountController 100% methods, 100% lines covered

AccountControllerTest: 12 total, 12 passed			914 ms
			Collapse Expand
AccountControllerTest			914 ms
check_change_password_with_correct_information_should_return_OK()	passed	538 ms	
check_register_user_with_existing_username_email_should_return_Bad_Request()	passed	72 ms	
check_logout_should_return_OK()	passed	6 ms	
check_update_profile_should_return_OK()	passed	15 ms	
check_login_with_nonexistent_account_should_return_bad_request()	passed	10 ms	
check_register_user_with_correct_user_information_should_return_OK()	passed	78 ms	
check_profile_request_should_return_OK()	passed	11 ms	
check_change_password_with_incorrect_information_should_return_Bad_Request()	passed	139 ms	
check_login_user_with_correct_information()	passed	21 ms	
check_register_user_with_incorrect_field_should_return_Bad_Request()	passed	11 ms	
check_delete_account_should_return_OK()	passed	7 ms	
check_delete_account_should_return_Bad_Request()	passed	6 ms	

- GroupController: 25 test cases (25 passes) and 100% coverage

GroupController 100% methods, 100% lines covered

GroupControllerTest: 25 total, 25 passed 765 ms

GroupControllerTest		765 ms
check_updateGroup_with_null_id_should_return_bad_request()	passed	401 ms
check_deleteGroup_with_not_owner_should_return_bad_request()	passed	27 ms
check_add_remove_member_with_nonexistent_group_should_return_bad_request()	passed	19 ms
check_exitFromGroup_with_non_existent_group_should_return_bad_request()	passed	12 ms
check_getGroupDetails_with_correct_info_should_return_OK()	passed	37 ms
check_deleteGroup_with_correct_info_should_return_OK()	passed	17 ms
check_createGroup_with_field_error_should_return_bad_request()	passed	30 ms
check_searchMember_with_not_member_should_return_bad_request()	passed	23 ms
check_getGroupDetails_with_nonexistent_group_should_return_bad_request()	passed	13 ms
check_searchMember_with_nonexistent_group_should_return_bad_request()	passed	12 ms
check_updateGroup_with_not_owner_should_return_bad_request()	passed	11 ms
check_deleteGroup_with_nonexistent_group_should_return_bad_request()	passed	9 ms
check_searchMember_with_search_null_should_return_OK()	passed	10 ms
check_updateGroup_with_correct_info_should_return_OK()	passed	12 ms
check_searchMember_with_search_should_return_OK()	passed	12 ms
check_add_remove_member_add_should_return_OK()	passed	23 ms
check_add_remove_member_remove_should_return_OK()	passed	10 ms
check_updateGroup_with_nonexistent_group_should_return_bad_request()	passed	8 ms
check_listUserGroups_with_correct_info_should_return_OK()	passed	12 ms
check_add_remove_member_with_nonexistent_user_should_return_false()	passed	11 ms
check_createGroup_without_field_error_should_return_OK()	passed	13 ms
check_add_remove_member_with_not_member_group_should_return_bad_request()	passed	11 ms
check_exitFromGroup_with_not_member_return_bad_request()	passed	10 ms
check_exitFromGroup_with_member_return_OK()	passed	11 ms
check_getGroupDetails_with_not_group_member_should_return_bad_request()	passed	11 ms

- Event Controller: 23 test cases (23 passes) and 99% coverage

EventController 100% methods, 99% lines covered

EventControllerTest: 23 total, 23 passed 644 ms

EventControllerTest		644 ms
check_createEvent_with_nonexistent_group_should_return_bad_request()	passed	381 ms
check_eventDetails_with_not_member_should_return_bad_request()	passed	14 ms
check_deleteEvent_with_nonexistent_event_should_return_bad_request()	passed	10 ms
check_rejectEvent_with_not_member_should_return_bad_request()	passed	10 ms
check_createEvent_with_field_error_should_return_bad_request()	passed	28 ms
check_listUserEvents_should_return_OK()	passed	17 ms
check_eventDetails_with_nonexistent_event_should_return_bad_request()	passed	12 ms
check_deleteEvent_with_organizer_should_return_OK()	passed	15 ms
check_createEvent_with_correct_info_should_return_OK()	passed	12 ms
check_rejectEvent_already_rejected_should_return_bad_request()	passed	10 ms
check_acceptEvent_with_nonexistent_event_should_return_bad_request()	passed	13 ms
check_deleteEvent_with_not_organizer_should_return_bad_request()	passed	14 ms
check_acceptEvent_not_accepted_should_return_OK()	passed	9 ms
check_eventDetails_with_member_should_return_OK()	passed	13 ms
check_createEvent_with_nonexistent_event_should_return_bad_request()	passed	11 ms
check_acceptEvent_with_not_member_should_return_bad_request()	passed	11 ms
check_createEvent_with_organizer_should_return_OK()	passed	11 ms
check_createEvent_with_not_organizer_should_return_bad_request()	passed	9 ms
check_rejectEvent_rejected_should_return_bad_request()	passed	10 ms
check_createEvent_with_not_member_should_return_bad_request()	passed	8 ms
check_rejectEvent_with_nonexistent_event_should_return_bad_request()	passed	8 ms
check_updateEvent_with_nonexistent_group_should_return_bad_request()	passed	9 ms
check_acceptEvent_already_accepted_should_return_bad_request()	passed	9 ms

- Exception Controller: 0 test case (since it just controls the thrown exceptions by other controllers) and 100% coverage

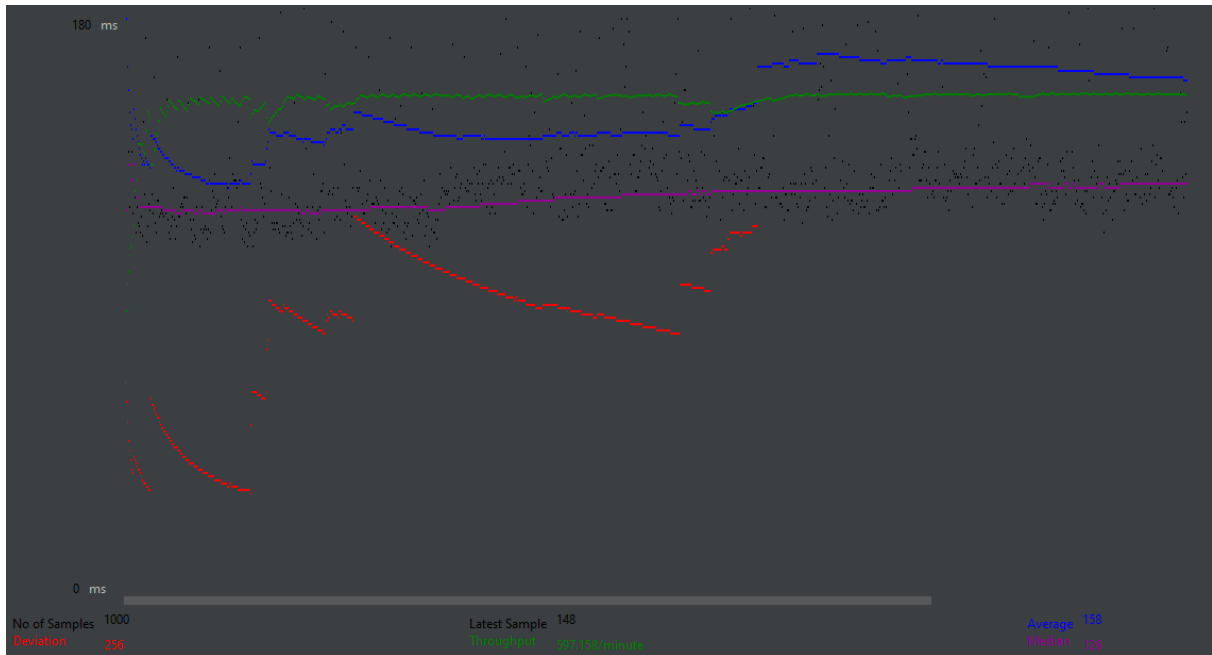
ExceptionHandler 100% methods, 100% lines covered

4.2 Performance-Load/Stress Test Results

To interpret results, one should focus on Throughput(green) and Deviation(red). More throughput and less deviation indicate better performance.

- List user's event groups: GET

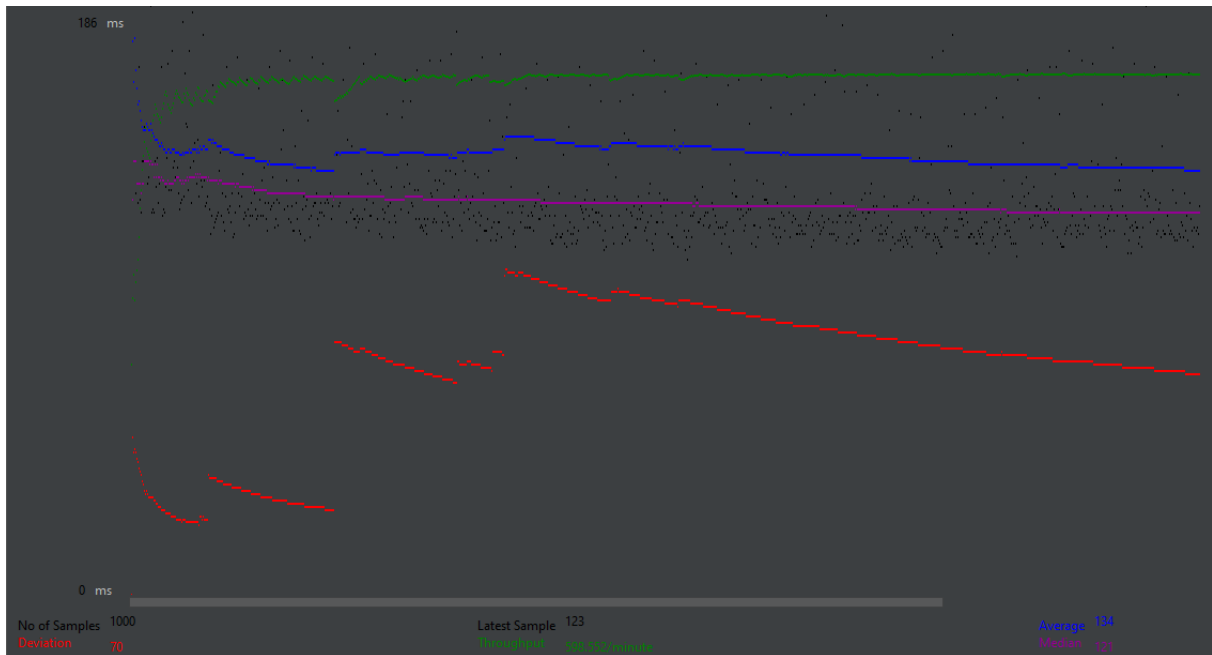
Web Server	
Protocol [http]:	Server Name or IP: eventwise-env.eba-ycrptzz8.eu-central-1.elasticbeanstalk.com
HTTP Request	
GET	Path: /api/group/list-user-groups



Here the latency was calculated as 180ms which is double our 1st non-functional requirement ("The application should not require unnecessary computing power and should respond to the actions of users fastly (60-90ms)").

- Get group details: GET

Web Server	
Protocol [http]:	Server Name or IP: eventwise-env.eba-ycrptzz8.eu-central-1.elasticbeanstalk.com
HTTP Request	
GET	Path: /api/group/group-details?groupId=43



- Create event: POST

This was the most critical and stressful test for our application

Protocol [http]: Server Name or IP: eventwise-env.eba-ycrptzz8.eu-central-1.elasticbeanstalk.com

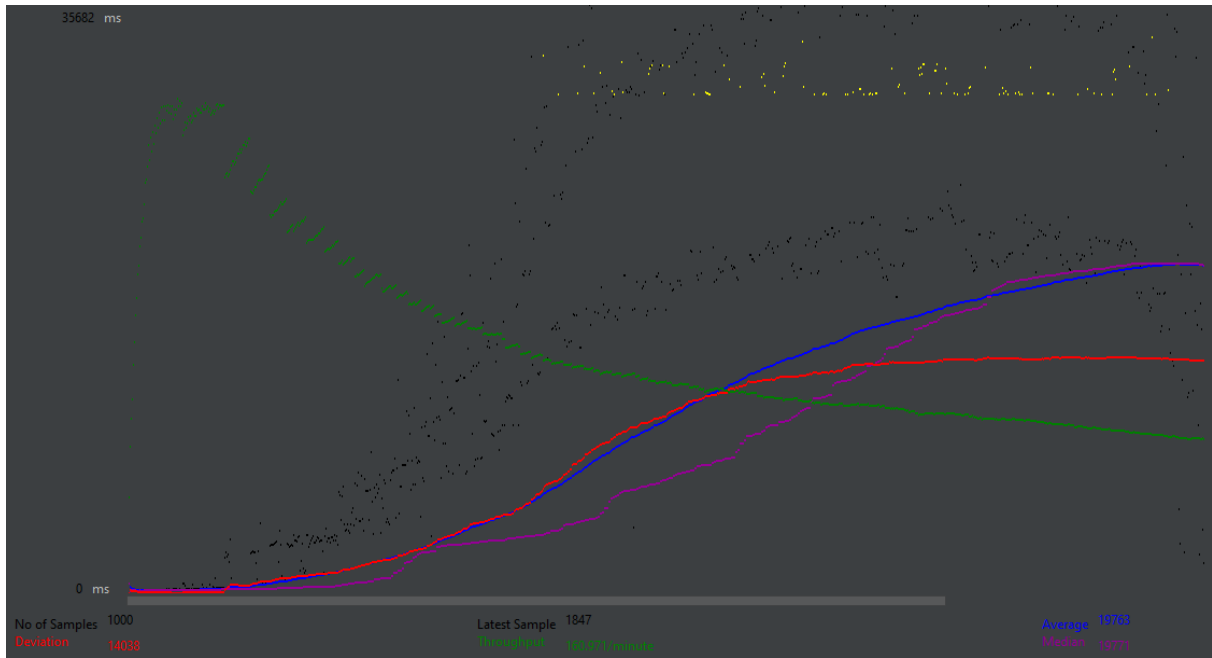
HTTP Request

POST Path: /api/event/create-event

☐ Redirect Automatically ☒ Follow Redirects ☒ Use KeepAlive ☐ Use multipart/form-data ☐ Browser-compatible headers

Parameters Body Data Files Upload

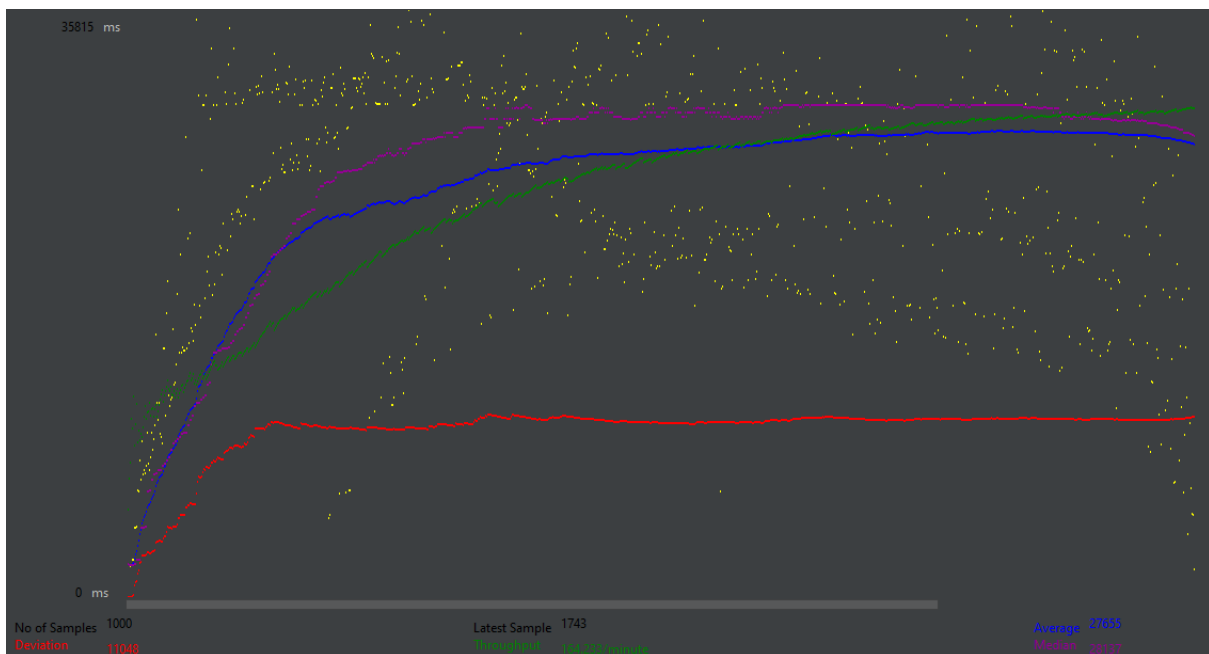
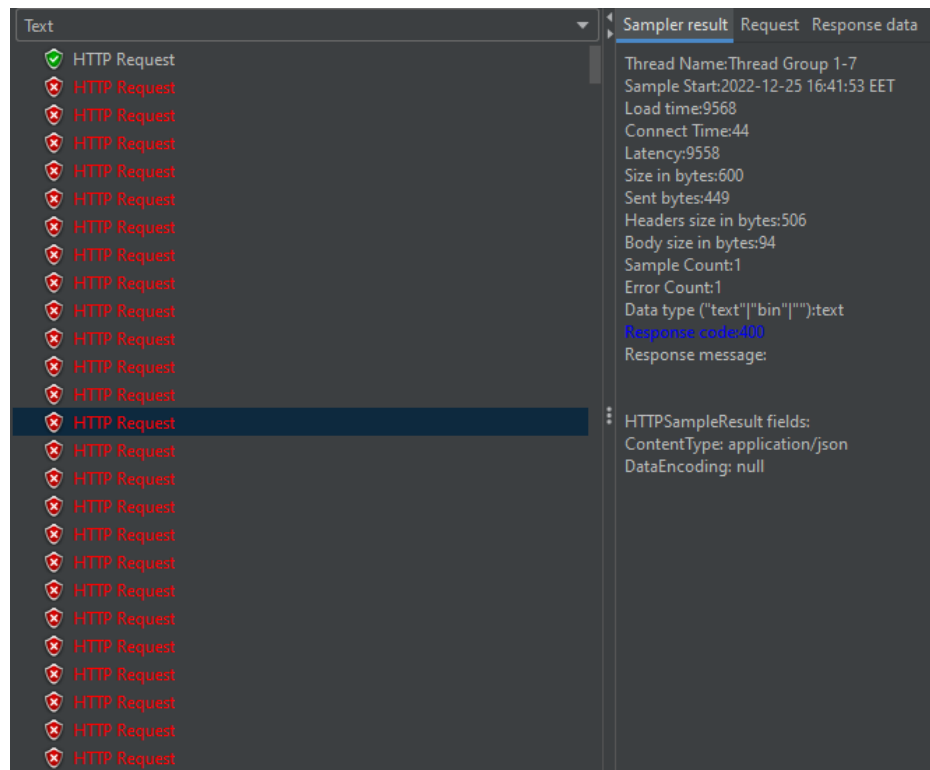
```
1 {  
2   "dateTime": "2022-12-25T12:04:31.881Z",  
3   "description": "for performance test",  
4   "eventId": 0,  
5   "eventName": "testEvents",  
6   "groupId": 43,  
7   "location": "kutuphane",  
8   "type": "Sport"  
9 }
```



- Accept event: PUT

Web Server	
Protocol [http]:	Server Name or IP: eventwise-env.eba-ycrtzz8.eu-central-1.elasticbeanstalk.com
HTTP Request	
PUT	Path: /api/event/accept-event?eventId=47

Only the first requested returned success because an event can be accepted if and only if it is not accepted yet. Therefore, the remaining requests returned 400 response code.



4.3 User Acceptance Tests Results

User Acceptance Test results can be shown below. Each test case's purpose, input, and output values are explained in the Black Box Testing section.

ExampleTest

Status

17 passed

17 tests, 3 m 11 s 836 ms

Filter tests:

Tests	Duration	Pixel_5_API_31
Test Results	2 m 55 s	17/17
ExampleTest	2 m 55 s	17/17
test0_checkSignup	12 s	
test1_checkLogin	2 s	
test20_checkCreateGroup	12 s	
test21_checkUpdateGroup	10 s	
test30_checkCreateEvent	11 s	
test31_checkUpdateEvent	6 s	
test32_checkRejectEvent	5 s	
test33_checkAcceptEvent	8 s	
test40_checkAddPeopleToGroup	9 s	
test41_checkRemovePeopleToGroup	9 s	
test50_checkDeleteEvent	6 s	
test51_checkDeleteGroup	9 s	
test52_createGroup	12 s	
test53_checkExitFromGroup	7 s	
test60_checkUpdateProfile	17 s	
test61_checkChangePasswordAndLogOut	19 s	
test70_checkDeleteAccount	12 s	

4.4 Security Tests Results

- Successful login with correct information

Name	Description
<div><div>loginRequest * required</div><div>(body)</div></div>	<div>loginRequest</div> <div><div>Example Value Model</div><div><pre>{ "password": "security", "username": "securityTester" }</pre></div></div>

Code	Details
200	<p>Response body</p> <pre>{ "id": 24, "token": "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXZWw6MmYpIiwiaWF0IjzZWN1cm18eVRlc3Rlc1IsImh0cCI6MTY3MTk3jg1Miw1ZXhwIjoxNjcxOTgwMDUyfq-p0srY3tHHHAf1pKDfPjsxhLT9YXmL9XET4sG2FwdQYzoWc00NIWmG3Qhorfdr9tMb8PtFVqn-SjqXwcoNMypp", "username": "securityTester", "email": "security@evetwise.com", "roles": ["ROLE_USER"], "authType": "Bearer" }</pre> <p>Download</p>

- Wrong password: There is an account but the wrong password is given. Then, it should return unauthorized.

Name	Description
loginRequest * required <i>(body)</i>	loginRequest Example Value Model <pre>{ "password": "wrongPass", "username": "securityTester" }</pre>

Code	Details
401	<p>Error:</p> <p>Response body</p> <pre>{ "timestamp": "2022-12-25T14:02:06.966+00:00", "status": 401, "error": "Unauthorized", "path": "/api/account/login" }</pre>

- Wrong username: Wrong username (does not belong to any account) is given. Then, it should return bad request.

Name	Description
loginRequest * required <i>(body)</i>	loginRequest Example Value Model <pre>{ "password": "security", "username": "wrongUser" }</pre>

Code	Details
400 <i>Undocumented</i>	Error: Response body <pre>{ "path": "/api/account/login", "message": "There is no such an account" }</pre>

- Trying to reach a page without being logged in
 - Get profile

Request URL	
<code>http://eventwise-env.eba-ycrptzz8.eu-central-1.elasticbeanstalk.com/api/account/profile</code>	
Server response	
Code	Details
401	Error: Response body <pre>{ "timestamp": "2022-12-25T14:05:21.185+00:00", "status": 401, "error": "Unauthorized", "path": "/api/account/profile" }</pre>

- List user's events

Request URL	
<code>http://eventwise-env.eba-ycrptzz8.eu-central-1.elasticbeanstalk.com/api/event/list-user-events</code>	
Server response	
Code	Details
401	Error: Response body <pre>{ "timestamp": "2022-12-25T14:06:19.506+00:00", "status": 401, "error": "Unauthorized", "path": "/api/event/list-user-events" }</pre>

- SQL injection
 - Try to inject the login page ("="" is Always True)

Name	Description
loginRequest * required <i>(body)</i>	loginRequest <div> Example Value Model </div> <pre> { "password": "\" or \"\"=\"", "username": "\" or \"\"=\"\" }</pre>

Code	Details
400 <i>Undocumented</i>	Error: Response body <pre> { "path": "/api/account/login", "message": "There is no such an account" }</pre>

- Try to inject the login page (Batched SQL Statements)

Name	Description
loginRequest * required <i>(body)</i>	loginRequest <div> Example Value Model </div> <pre> { "password": "security", "username": "securityTester; DROP TABLE Event" }</pre>

Code	Details
400 <i>Undocumented</i>	Error: Response body <pre> { "path": "/api/account/login", "message": "There is no such an account" }</pre>

As can be seen, SQL injection cannot be applied to the system.