

# Test Plan

Brendan College, Alejandro Fernandez, Charles Karlson, Samantha Maddox (Group 6)

Project 1: Group Matching App

COP 4331, Fall 2021

---

## Contents of this Document

Overall Objective for Software Test Activity

Description of Test Environment

Overall Stopping Criteria

Description of Individual Test Cases

Appendices

---

### **Overall Objective for Software Test Activity**

- Test cases and testing should attempt to accomplish at least one of the following criteria:
    - Verification of general operation and behavior
    - Correcting behavior of the program under unexpected conditions or edge cases
    - Testing security measures through replication of malicious or careless behavior
- 

### **Description of Test Environment**

Tests should be performed in vitro isolated to different modules of the program. This shall be performed by individual developers on developers' systems before committing any modules that have been developed on or under maintenance. Multiple test cases shall be used.

Once in vitro tests have been passed, tests should be performed on the whole program in vivo. These tests may be performed by developers on application systems. Bug reports submitted by users may also be evaluated by developers and attempts may be made to reproduce the issue. Multiple test cases may be used.

---

### **Stopping Criteria**

Tests will conclude on a specific issue once the specific issue has been fixed and the related modules perform as intended. These conclusions should be verified by other developers and shall be

demonstrated to management upon request. “Fixed” shall mean that the concerned operations perform as intended under multiple sets of test cases. “As intended” should mean that the operation behaves normally, repeatably, and safely. If no errors are encountered after using all test cases, more test cases should be developed until all permutations have been exhausted.

Time spent on testing should proportionally match the severity of the issue. The issue should be relegated to further maintenance if higher priority issues arise or if a different module lacks functionality considered more important. In summary, work performed on modules (whether it be development, testing, or maintenance) should be prioritized by the developer according to their own intuition and expertise.

Modules should be in a complete-enough state before performing work on other modules. “Complete-enough” can be defined as retaining most of the module’s functionality (i.e. functions appropriately on most, but not necessarily all test cases) at that point. If the module relies on another undeveloped module, that module can be complete up to the point where it will need to rely on the undeveloped module(s).

The program will be considered for launch when almost all of the functionality and cosmetic appearance are demonstrated successfully.

---

## **Description of Individual Test Cases**

### *Test Cases*

1. Demonstrate that users can be correctly created
  - a. Create a user
  - b. The user will have:
    - i. ID = 1
    - ii. Email = [dev.mymeet@gmail.com](mailto:dev.mymeet@gmail.com)
    - iii. Username = test
    - iv. Password = test
    - v. Interests = {Sports, Movies & TV, School}
    - vi. FirstName = test
    - vii. LastName = test
  - c. Test will be run under two conditions:
    - i. This user does not exist
    - ii. This user does exist
  - d. Expected results:
    - i. Database will contain user
    - ii. Database will still contain user after logoff
    - iii. If user ID already exists, error message will be shown to client and database will not change
2. Demonstrate that users can be correctly deleted
  - a. Login as user
  - b. Navigate to settings, delete account

- c. Conditions:
      - i. User must exist
      - ii. User is not attached to any groups or events
    - d. Expected results:
      - i. Database will remove all entries related to the user
      - ii. Database will retain changes after logoff
      - iii. If user is attached to group or event, deletion is prevented
- 3. Demonstrate that users can create a group
  - a. Login as user
  - b. Create a group
  - c. Group will have:
    - i. ID = 1
    - ii. Interests = {Sports, Movies & TV, School}
  - d. Test will be run under two conditions:
    - i. This group does not exist
    - ii. This group does exist
  - e. Expected results:
    - i. Database will contain group and creator will be administrator
    - ii. Database will still contain group after user logoff
    - iii. If group ID already exists, error message will be shown to client and database will not change
- 4. Demonstrate that users can be correctly delete a group
  - a. Login as user
  - b. Navigate to settings, delete account
  - c. Conditions:
    - i. Group must exist
    - ii. User is a group administrator
  - d. Expected results:
    - i. Database will remove all entries related to the group, including all attached events
    - ii. Database will retain changes after logoff
    - iii. If user is not administrator, deletion will be prevented (deletion button will not be presented to non-administrators)
- 5. Demonstrate that users can correctly join a group
  - a. Login as user
  - b. Conditions
    - i. Search for group
    - ii. Groups is suggested based on interests
  - c. Expected results
    - i. Database will successfully create a user-group join relationship
    - ii. Program will successfully suggest a group of similar interests if that is requested.
- 6. Demonstrate that users can correctly leave a group
  - a. Login as user
  - b. Conditions
    - i. Is final administrator
    - ii. Is any other member
  - c. Expected results

- i. If is not final admin, it will delete the user-group join relationship from the database
    - ii. If it is the final admin it will prevent leave group and will require group delete or another admin
- 7. Demonstrate that users can correctly create an event in a group
  - a. User belongs to a group
  - b. Conditions
    - i. Only one condition
  - c. Event will have
    - i. Name = Operation Christmas
    - ii. Location = North Pole
    - iii. Date = December 25, 2021
    - iv. Time = 12:00 PM
    - v. Duration = 24 hours
  - d. Expected Results
    - i. Event will be created with the above information that auto enerates an ID based of of the Group ID and an unique Event ID
    - ii. Will be persistent after log off
- 8. Demonstrate that users can correctly join an event in a group
  - a. User navigates to group page or user searches for event ID
  - b. Test conditions (against both above options)
    - i. Event does not exists
    - ii. Event exists, user is already a member
    - iii. Event exists, user is not a member
  - c. Expected behavior
    - i. If event does not exist, returns an error
    - ii. If user is already a member, returns different error
    - iii. If user is not a member, creates a persistent user event join item in the database, otherwise no change in database
- 9. Demonstrate that users can correctly add images to an event in a group
  - a. User navigates to event page
  - b. Test conditions
    - i. Event must exist to be in event page
  - c. Expected behavior
    - i. User is able to add image
    - ii. Database retains changes after logoff
- 10. Demonstrate that users can correctly leave an event in a group
  - a. User navigates to event page
  - b. Test conditions
    - i. User is in an existing event
  - c. Expected behavior
    - i. User no longer has access to event if event has concluded
    - ii. User will be able to rejoin event otherwise
- 11. Demonstrate that users can correctly delete an event in a group
  - a. User navigates to event page
  - b. Test conditions
    - i. User is an event creator and event has not already started
    - ii. User is an event creator and event has started

- iii. User is not event creator and event has not started
    - iv. User is not event created and event has started
  - c. Expected behavior
    - i. Event no longer exists in the database only if user is creator and event has not started
- 12. Demonstrate that users can correctly send messages in a group
  - a. User navigates to group page
  - b. Test conditions
    - i. User is a member of the group
    - ii. User is an administrator of the group
    - iii. User is not a member of the group
  - c. Expected behavior
    - i. Members and administrators are able to send and receive messages
    - ii. Nonmembers are not able to send and receive messages
- 13. Demonstrate that users can correctly remove a different user in a group
  - a. User navigates to group page
  - b. Test conditions
    - i. User is an administrator of the group
    - ii. User is an administrator of another group (but not this group)
    - iii. User is a member of the group
    - iv. User is not a member of the group
  - c. Expected behavior
    - i. User is able to remove other members of the group only if they are an administrator of that group
    - ii. Creates kicked from group relationship and removes group members relationship

#### **Trace of Individual Test Cases to Requirements**

Requirement ID	Requirement Description	Test Case Reference	Status
I	Users shall be able to manage users	1, 2, 5, 6, 13	In progress
II	Users shall be able to manage groups	3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13	In progress
III	Users shall be able to manage events	7, 8, 9, 10, 11	In progress

#### **Appendices:**

None.