Verification and Validation Report: Software Engineering

Team #2, Campus Connections
Waseef Nayeem
Zihao Du
Matthew Miller
Firas Elayan
Abhiram Neelamraju
Michael Kim

April 4, 2024

1 Revision History

Date	Version	Notes
Mar 4	1.0	Add functional requirements evaluation
Mar 6	1.0	Add non-functional requirements evaluation
Mar 6	1.0	Load test, Usability test, unit test results
Mar 6	1.0	Add Automated test, non-dynamic test result and changes due to testing
Apr 2	1.1	Resolve team 11 feedback
Apr 3	1.1	Resolve TA feedback about functional and non functional tests

2 Symbols, Abbreviations and Acronyms

Symbol	Description
APK	Android Application Package
AR	Augmented reality
Azure	A cloud computing platform run by Microsoft
CI	Continuous integration
JHE	John Hodgins Engineering Building
JMeter	Load testing tool for analyzing and measuring the performance
NUnit	An open-source unit testing framework for the .NET Framework and Mono
OpenCover	An open source code coverage tool
SRS	Software Requirements Specification
TA	Teaching Assistant
UI	User Interface
UTF	Unity Test Framework
VnV	Verification and Validation
XML	Extensible Markup Language

Contents

List of Tables

List of Figures

This document describes the test results of the verification and validation (VnV) plan for CampusConnections. The VnV plan was continuously updated as the project evolved. The following document records the results of the current version of the VnV plan. It provides results of functional and non-functional requirements tests, unit tests, changes that will be implemented in the system as a result of the tests, and various traceability tables.

3 Functional Requirements Evaluation

The following section outlines the results of functional requirements testing. The process and test performed follow the VnV Plan. To summarize, all the tests are tested manually and passed, indicating that all the functional requirements in the Software Requirements Specification (SRS) document are covered.

3.1 Pre-Registration Settings

This section covers all tests related to functional requirements about pre-registration settings.

1. **FRT-PR1**

   Name: Agree To Consent Form

   Initial State: The user does not have an account, and they starts to register an account. A consent form appears asking for access to the device and permission to collect user data

   Input: The user agrees to all the terms and conditions and clicks 'Agree' and continues to complete the registration process

   Expected Output: A notification shows the registration succeeds and the user is redirected to the login screen

   Actual Output: A notification shows the registration succeeds and the user is redirected to the login screen

   Results: Pass

2. **FRT-PR2**

   Name: Disagree To Consent Form

Initial State: The user does not have an account, and they starts to register an account. A consent form appears asking for access to the device and permission to collect user data

Input: The user rejects the terms and conditions and clicks 'Disagree' and continues to complete the registration process

Expected Output: The registration fails and a warning will show up notifying the user that they cannot create an account unless they agree to the consent form

Actual Output: The registration fails and a warning will show up notifying the user that they cannot create an account unless they agree to the consent form

Results: Pass

3.2 User Account

This section covers all tests related to functional requirements about the account and user profile.

1. **FRT-UA1**

   Name: Successful Account Creation

   Initial State: The user does not have an account and is not logged in to the application

   Input: All information needed to create an account:

   - Email: testUA1@gmail.com
   - password: FRT-UA1
   - nickname: UA1

   Expected Output: An Account with corresponding information is created in the database with the account initialized to INITIAL_USER_STATE

   Actual Output: An Account is created in the database with the following attributes and other attributes are initialized to INITIAL_USER_STATE:

   - Email: testUA1@gmail.com
   - password: FRT-UA1

- nickname: UA1

Results: Pass

2. **FRT-UA2**

   Name: Unsuccessful Account Creation With Existing Email

   Initial State: The user does not have an account and is not logged in to the application

   Input: All information needed to create an account:

   - Email: qtest@gmail.com (this is an existing test account)
   - password: FRT-UA1
   - nickname: UA1

   Expected Output: Account creation fails with a warning telling the user the email has already been used

   Actual Output: Account creation fails with a warning telling the user the email has already been used

   Results: Pass

3. **FRT-UA3**

   Name: Successful Account Login

   Initial State: The user has an account and is not logged in to the application

   Input: All information needed to login:

   - Email: FRT-UA3@test.com (this account exists in the system already)
   - password: FRT-UA3

   Expected Output: User successfully logs into the application and goes to the menu page

   Actual Output: User successfully logs into the application and goes to the menu page

   Results: Pass

4. **FRT-UA4**

   Name: Unsuccessful Account Login With Wrong Password

   Initial State: The user has an account and is not logged in to the application

   Input: All information needed to login:

   - Email: FRT-UA3@test.com (this account exists in the system already)
   - password: FRT321 (wrong password)

   Expected Output: Login fails with a warning telling the user the password is wrong

   Actual Output: Login fails with a warning telling the user the password is wrong

   Results: Pass

5. **FRT-UA5**

   Name: Account Deletion

   Initial State: The user has an account and is logged into the application

   - Email: FRT-UA5@gmail.com (this is an existing test account)
   - password: FRT-UA5
   - nickname: UA5

   Input: User clicks on the delete account button on the profile page and confirms the deletion

   Expected Output: The user is redirected to the login page and the account cannot be logged in any more

   Actual Output: The user is redirected to the login page and the account FRT-UA5@gmail.com cannot be logged in any more

   Results: Pass

6. **FRT-UA6**

   Name: Reset Password

   Initial State: The user has an account:

- Email: campusconnections@gmail.com (this is an existing test account)
- password: qtesting

Input: Email address and new password

- new password: QTesting

Expected Output: Password is successfully reset

Actual Output: Password is successfully reset to be QTesting

Results: Pass

7. **FRT-UA7**

   Name: Avatar Creation and Modification

   Initial State: The user has an account with DEFAULT_AVATAR

   Input: URI represents the new avatar:

   - URI: https://upload.wikimedia.org/wikipedia/commons/2/2f/Google_2015_logo.svg

   Expected Output: The user changes the avatar to a Google logo

   Actual Output: The user changes the avatar to a Google logo

   Results: Pass

8. **FRT-UA8**

   Name: Email Verification

   Initial State: The user has an account whose email has not been verified yet

   - Email: fuz15@mcmaster.ca (this is an existing test account)
   - password: password

   Input: User clicks on 'Verify Email' button on user profile page and follows instructions on the email sent from the system

   Expected Output: That email above is verified as a valid email address

   Actual Output: fuz15@mcmaster.ca is verified as a valid email address in the system

Results: Pass

9. **FRT-UA9**

   Name: Edit Profile

   Initial State: The user has an account

   - Email: qtest@gmail.com (this is an existing test account)
   - password: qtesting
   - program: Software Engineering

   Input: New Profile:

   - newProgram: Computer Science
   - newLevel: 4

   Expected Output: The program and level are updated

   Actual Output: The program and level are updated to be Computer Science and 4

   Results: Pass

3.3 Social Networking System

This section covers all tests related to functional requirements about interactions between friends.

1. **FRT-SN1**

   Name: Successful Friend Request

   Initial State: The user is logged in with the following account:

   - Femail: FRT-SN1@test.com
   - password: testing

   Input: A valid email to send the request:

   - Temail: FRT-SN1-F@test.com

Expected Output: A Friend request is sent to the target user and available to be accepted or rejected by the target user in the request list

Actual Output: A Friend request is sent to the target user and available to be accepted or rejected by the target user in the request list

Results: Pass

2. **FRT-SN2**

   Name: Friend Request Acceptance

   Initial State: A friend request was sent from an account (Femail) to the target account (Temail):

   - Femail: FRT-SN1@test.com
   - Temail: FRT-SN1-F@test.com

   Input: The request is accepted

   Expected Output: Two users are added to each other's friend lists

   Actual Output: Two users are added to each other's friend lists

   Results: Pass

3. **FRT-SN3**

   Name: Successful Friend Rejection

   Initial State: A friend request was sent from an account (Femail) to the target account (Temail):

   - Femail: FRT-SN1@test.com
   - Temail: FRT-SN1-F@test.com

   Input: The request is rejected

   Expected Output: The request is declined and no friend is added for both accounts

   Actual Output: The request is declined and no friend is added for both accounts

   Results: Pass

4. **FRT-SN4**

   Name: Friend Deletion

   Initial State: A friend (Femail) exist in the friend list of the test account (Temail):

   - Temail: FRT-SN4@test.com
   - Femail: FRT-SN4-F@test.com

   Input: User deletes the chosen friend

   Expected Output: The corresponding friend is deleted from the list

   Actual Output: The corresponding friend FRT-SN4-F@test.com is deleted from the friend list

   Results: Pass

5. **FRT-SN5**

   Name: Friend Messaging

   Initial State: A friend (Femail) exist in the friend list of the test account (Temail):

   - Temail: FRT-SN5@test.com
   - Femail: FRT-SN5-F@test.com

   Input: Message: ‘Hello World’

   Expected Output: The corresponding message is sent to the friend

   Actual Output: The corresponding message is sent to the friend

   Results: Pass

6. **FRT-SN6**

   Name: Friend Sharing Event

   Initial State: A friend (Femail) exist in the friend list of the test account (Temail):

   - Temail: FRT-SN6@test.com
   - Femail: FRT-SN6-F@test.com

Input: Message that contains event name and follows some specific pattern: Hey, check this event: _E_[EXPO]

Expected Output: User is redirected to the event page with that event once they click on the message

Actual Output: User is redirected to the event page with a filter on event name: EXPO

Results: Pass

7. **FRT-SN7**

   Name: Friend Sharing Lecture

   Initial State: A friend (Femail) exist in the friend list of the test account (Temail):

   - Temail: FRT-SN7@test.com
   - Femail: FRT-SN7-F@test.com

   Input: Message that contains lecture code and follows some specific pattern: ‘Hey, are you in this lecture: _L_[SFRWENG 4G06]’

   Expected Output: User is redirected to the lecture page with that lecture once they click on the message

   Actual Output: User is redirected to the lecture page with a filter on lecture code: SFRWENG 4G06

   Results: Pass

3.4 Lectures and Events

This section covers all tests related to functional requirements about lectures and events and how users can interact with them.

1. **FRT-LE1**

   Name: Bookmark Event

   Initial State: A sample event:

   - Name: EXPO

Input: On the event page, user clicks on the bookmark button on the pop-up window with details of the sample event

Expected Output: The event is bookmarked to the user's event list

Actual Output: The event EXPO is bookmarked to the user's event list

Results: Pass

2. **FRT-LE2**

   Name: Unbookmark Event

   Initial State: A sample event that is already been bookmarked:

   - Name: EXPO

   Input: On the event page, user clicks on the unbookmark button on the pop-up window with details of the sample event

   Expected Output: The event is removed from the user's event list

   Actual Output: The event EXPO is removed from the user's event list

   Results: Pass

3. **FRT-LE3**

   Name: Bookmark Lecture

   Initial State: A sample lecture:

   - Code: SFWRENG 4G06

   Input: On the lecture page, user clicks on the bookmark button on the pop-up window with details of the sample lecture

   Expected Output: The lecture is bookmarked to the user's lecture list

   Actual Output: The lecture SFWRENG 4G06 is bookmarked to the user's lecture list

   Results: Pass

4. **FRT-LE4**

   Name: Unbookmark Lecture

   Initial State: A sample lecture that is already been bookmarked:

   - Code: SFWRENG 4G06

   Input: On the lecture page, user clicks on the unbookmark button on the pop-up window with details of the sample lecture

   Expected Output: The lecture is removed from the user's lecture list

   Actual Output: The lecture SFWRENG 4G06 is removed from the user's lecture list

   Results: Pass

5. **FRT-LE5**

   Name: Administrator Add Event

   Initial State: User is logged in as an administrator

   - email: campusconnections@gmail.com
   - password: testing

   Input: Sample event:

   - name: Test event
   - description: Sample event for system test
   - time: 0
   - duration: 0
   - location: Online
   - isPublic: true
   - organizer: Team 2

   Expcted Output: The event is added to the event list

   Actual Output: The event Test event is added to the event list

   Results: Pass

6. **FRT-LE6**

   Name: Administrator Add Event With Existing Name

   Initial State: User is logged in as an administrator

   - email: campusconnections@gmail.com
   - password: testing

   Sample event exists:

   - name: Test event
   - description: Sample event for system test
   - time: 0
   - duration: 0
   - location: Online
   - isPublic: true
   - organizer: Team 2

   Input: New sample event:

   - name: Test event
   - description: New sample event
   - time: 100
   - duration: 30
   - location: TBD
   - isPublic: false
   - organizer: TBD

   Expcted Output: The sample event is update to be the same as the input event

   Actual Output: The sample event is update to be the same as the input event

   Results: Pass

7. **FRT-LE7**

   Name: Administrator Edit Event

   Initial State: User is logged in as an administrator

   - email: campusconnections@gmail.com
   - password: testing

   Input: Sample event name and new location:

   - name: Test event
   - location: ITB AB102

   Expected Output: The test event location is updated to the new one

   Actual Output: The test event location is updated to ITB AB102

   Results: Pass

8. **FRT-LE8**

   Name: Administrator Delete Event

   Initial State: User is logged in as an administrator

   - email: campusconnections@gmail.com
   - password: testing

   Input: Sample event (already in the system) name:

   - name: Test event

   Expected Output: The event is deleted and disappears from the list

   Actual Output: The Test event is deleted and disappears from the list

   Results: Pass

9. **FRT-LE9**

   Name: Administrator Add Lecture

   Initial State: User is logged in as an administrator

   - email: campusconnections@gmail.com

- password: testing

Input: Sample lecture:

- code: TEST 1T03
- name: Test lecture
- time: 12:00 - 13:00, Mon
- location: Online
- instructor: NA

Expected Output: The lecture is added to the lecture list

Actual Output: The lecture TEST 1T03 is added to the lecture list

Results: Pass

10. **FRT-LE10**

    Name: Administrator Add Lecture With Existing Code

    Initial State: User is logged in as an administrator

    - email: campusconnections@gmail.com
    - password: testing

    A sample lecture exists:

    - code: TEST 1T03
    - name: Test lecture
    - time: 12:00 - 13:00, Mon
    - location: Online
    - instructor: NA

    Input: New sample lecture:

    - code: TEST 1T03
    - name: New test lecture
    - time: 12:00 - 13:00, Wed
    - location: TBD

- instructor: TBD

Expected Output: The sample lecture is update to be the same as the input lecture

Actual Output: The sample lecture is update to be the same as the input lecture

Results: Pass

11. **FRT-LE9**

    Name: Administrator Edit Lecture

    Initial State: User is logged in as an administrator

    - email: campusconnections@gmail.com
    - password: testing

    Input: Sample lecture name and new location:

    - code: TEST 1T03
    - location: ITB AB102

    Expected Output: The test lecture location is updated to the new one

    Actual Output: The test lecture location is updated to ITB AB102

    Results: Pass

12. **FRT-LE10**

    Name: Administrator Delete Lecture

    Initial State: User is logged in as an administrator

    - email: campusconnections@gmail.com
    - password: testing

    Input: Sample lecture (already in the system) name:

    - code: TEST 1T03

Expected Output: The lecture is deleted and disappears from the list

Actual Output: The test lecture is deleted and disappears from the list

Results: Pass

13. **FRT-LE11**

    Name: Event Information

    Initial State: A sample event exists:

    - name: EXPO

    Input: User clicks on the sample event

    Expected Output: All event information is shown in a pop-up window

    Actual Output: All event information is shown in a pop-up window:

    - name
    - description
    - location
    - time
    - duration
    - organizer
    - isPublic

    Results: Pass

14. **FRT-LE12**

    Name: Lecture Information

    Initial State: A sample lecture exists:

    - code: SFWRENG 4G06

Input: User clicks on the sample lecture

Expected Output: All lecture information is shown in a pop-up window

Actual Output: All lecture information is shown in a pop-up window:

- code
- name
- instructor
- time
- location

Results: Pass

15. **FRT-LE13**

    Name: Lecture Filter by Code

    Initial State: Some software engineering lectures exist in the system with the following course codes (other attributes of lecture do not affect the test and can be empty):

    - SFWRENG 4G06
    - SFWRENG 4E03
    - COMPSCI 4TB3

    Input: Filter:

    - FilterString: SFWRENG

    Expected Output: All lectures which do not contain the FilterString in the code are removed from the list (COMPSCI 4TB3 in this case)

    Actual Output: All lectures which do not contain the SFWRENG in the code are removed from the list (COMPSCI 4TB3 in this case)

    Results: Pass

16. **FRT-LE14**

    Name: Event Filter by Name

    Initial State: Some events exist in the system with following names (other attributes of event do not affect the test and can be empty):

- Job Fair: March 4
- Job Fair: March 10
- Rev 0 Demo

Input: Filter:

- FilterString: Job Fair

Expected Output: All events which do not contain the FilterString in the name are removed from the list (Rev 0 Demo in this case)

Actual Output: All events which do not contain the Job Fair in the name are removed from the list (Rev 0 Demo in this case)

Results: Pass

3.5 AR Camera

This section covers all tests related to functional requirements about AR camera.

1. **FRT-AR1**

   Name: Successful Building Recognition

   Initial State: User is at the front door of JHE

   Input: Clear camera view

   Expected Output: The building (JHE) is recognized with its name and description shown as an AR object

   Actual Output: The building (JHE) is recognized with its name and description shown as an AR object

   Results: Pass

2. **FRT-AR2**

   Name: Unsuccessful Building Recognition

   Initial State: User is out of campus

   Input: Clear camera view

   Expected Output: No AR objects are shown

Actual Output: No AR objects are shown

Results: Pass

3. **FRT-AR3**

   Name: Building Lectures/Events

   Initial State: User is in JHE lobby

   Input: Clear camera view

   Expected Output: Event and lecture information separated by room number at the corresponding locations of the building

   Actual Output: Event and lecture information separated by room number at the corresponding locations of the building

   Results: Pass

3.6 Map and Location

This section covers all tests related to functional requirements about the map and location tracking in the system.

1. **FRT-MAP1**

   Name: User Location

   Initial State: User allows the user to use their real-time location

   Input: User enters the map page

   Expected Output: A model representing the user shows up on the map and moves correspondingly when the user moves

   Actual Output: A model representing the user shows up on the map and moves correspondingly when the user moves

   Results: Pass

2. **FRT-MAP2**

   Name: Friend Locations

   Initial State: User has some friends who are willing to share locations:

   - email1: MAP2-1@test.com
   - email2: MAP2-2@test.com

Input: User enters the map page

Expected Output: Additional models representing friends show up on the map and move correspondingly when friends move

Actual Output: Additional models representing friends show up on the map and move correspondingly when friends move

Results: Pass

4 Nonfunctional Requirements Evaluation

The following section outlines the results of non-functional requirements testing. The process and test performed follow the VnV Plan. Most of the tests are tested manually while some of them are tested in another way, for instance, load testing is tested with JMeter while some UI requirements are tested by conducting a survey, etc. Some of the tests fail because their related requirements are removed due to the change of the project's scope, these tests will be marked in red. In general, most of the tests in the plan succeed, indicating that non-functional requirements in the Software Requirements Specification (SRS) document are covered.

4.1 Look and feel

This section corresponds to the Look and feel tests in VnV Plan and Look and Feel requirements in SRS.

1. **NFRT-LF1**

   Name: Survey for feedback on application layout

   Initial State: Survey taker is given an account:

   - email: mtest@gmail.com
   - password: mtesting

   Input: Usability Survey in section 5

   Expected Output: Tasks are completed successfully and "Immediate Visual Response when Clicking" and "Appealing Colour Scheme" questions get average scores that are great than MIN_SCORE

Actual Output: Tasks are completed successfully and “Immediate Visual Response when Clicking” and “Appealing Colour Scheme” questions get average scores of 3.8 and 4.8 respectively

Result: Pass

2. **NFRT-LF2**

   Name: Visual inspection with different screen sizes

   Input/Condition: User opens the application on the phone with all different screen sizes in the SCREEN_VIEWPORTS list

   Expected Output: For all different pages all visual elements on the screen are within the borders of the screen for all screens in the SCREEN_VIEWPORTS list

   Actual Output: All elements are within the borders of the screen without overlapping for all screen sizes in the SCREEN_VIEWPORTS list

   Result: Pass

3. **NFRT-LF3**

   Name: Visual inspection of color scheme

   Initial State: NA

   Input: User opens the application on the phone

   Expected Output: For all different pages the colour scheme is the same

   Actual Output: For all different pages the colour of elements and texts is always maroon, gold, black and white

   Result: Pass

4.2 Usability and Humanity

This section corresponds to the Usability and Humanity tests in VnV Plan and Usability and Humanity requirements in SRS.

1. **NFRT-UH1**

   Name: Survey for feedback on understandability and easy of use

   Initial State: Survey taker is given an account:

- email: mtest@gmail.com
- password: mtesting

Input: Usability Survey in section 5

Expected Output: Tasks are completed successfully and "No Technical or Software-Specific Language" question gets an average score that is great than MIN_SCORE

Actual Output: Tasks are completed successfully and "No Technical or Software-Specific Language" question gets an average score of 4

Result: Pass

2. **NFRT-UH2**

   Name: Walkthrough of user guide

   Input: GitHub web page (see details in section 7)

   Expected Output: The participants can see the main features are explained in the GitHub repo

   Actual Output: The participants can see the main features are explained in the GitHub repo

   Result: Pass

3. **NFRT-UH3**

   Name: Visual inspection of color contrast

   Initial State: NA

   Input: User opens the application checks the color contrast statically

   Expected Output: The color contrast is greater than 4.5:1, the Web Content Accessibility Guidelines' AA standards for accessibility

   Actual Output: The color contrasts are listed following:

   - Maroon - White: 10.94:1
   - White - Black: 21:1
   - Gold - Black: 14.97:1
   - Maroon - Gold: 7.8:1

   Result: Pass

4.3 Performance

This section corresponds to the Performance tests in VnV Plan and Performance requirements in SRS.

1. **NFRT-P1**

   Name: AR camera recognition

   Initial State: The user is near or in a target building (JHE)

   Input: User turns on AR camera

   Expected Output: Corresponding AR objects appears within RECOGNITION_TIME

   Actual Output: Corresponding AR objects appears within 1 second

   Result: Pass

2. **NFRT-P2**

   Name: Real-time location update

   Initial State: User allows the application to use device location

   Input: User turns on the map and walks around on campus

   Expected Output: The user model on the map is updated within LOCATION_UPDATE_TIME when the user is moving

   Actual Output: The user model is updated within 0.5 second if the user is outdoor, and indoor location update time is around 5 second (and sometimes not very accurate)

   Result: Pass

3. **NFRT-P3**

   Name: AR camera accuracy

   Initial State: User is near or in a target building (JHE)

   Input: User turns on the AR Camera and repeat multiple times

   Expected Output: AR objects shows up for at least AR_ACCURACY * number of tests times

   Actual Output: AR objects always shows up when walking around JHE lobby

   Result: Pass

4. **NFRT-P4**

   Name: Warning when internet connection is lost

   Initial State: User has no internet connection

   Input: User opens the application

   Expected Output: There is a pop-up window telling the user the internet is lost

   Actual Output: Nothing

   Result: Fail

   Reason: Due to a change in the scope of the project, the corresponding requirement is moved out of the scope, therefore this test fails because the feature is not implemented

5. **NFRT-P5**

   Name: Rudimentary functions when the server connection is lost

   Initial State: Server is turned down

   Input: User opens the application

   Expected Output: The application still works with limited functionalities

   Actual Output: The lecture and event pages still work, the single user map still works, the friend system still works

   Result: Pass

6. **NFRT-P6**

   Name: Code inspection for server restart

   Initial State: NA

   Input: Server settings (see details in section 7)

   Expected Output: Successfully convinced the participants the following:

   - Server attempts to restart when it crashes

   Actual Output: Successfully convinced the participants that the server is set to restart once it goes down

   Result: Pass

7. **NFRT-P7**

   Name: AR camera help button

   Initial State: User turns on the AR camera

   Input: User clicks the help button

   Expected Output: A message telling the user weather, light and decorations may affect AR camera appears

   Actual Output: A message telling the user weather, light and decorations may affect AR camera appears

   Result: Pass

8. **NFRT-P8**

   Name: Load testing for server

   Initial State: The server is online and open to connections

   Input: Load testing with JMeter (see details in section 6)

   Expected Output: The server is able to handle up to MAX_CAPACITY users connecting to the server simultaneously

   Actual Output: The server is able to handle 1000 connections at the same time

   Result: Pass

9. **NFRT-P9**

   Name: Code inspection for database capacity

   Initial State: NA

   Input: Database documentation (see details in section 7)

   Expected Output: Successfully convinced the participants the following:

   - The database has enough space to store all the user, lecture and event information

   Actual Output: Successfully convinced the participants that the current plan has enough space for expected number of users

   Result: Pass

10. **NFRT-P10**

    Name: Code walkthrough for adding new building

    Initial State: NA

    Input: Source code (see details in section 7)

    Expected Output: Successfully convinced the participants the following:

    - A new target building can be added without causing the application running any slower

    Actual Output: Successfully convinced the participants that adding a new building is just like adding a new scene and will not affect the speed of the application

    Result: Pass

11. **NFRT-P11**

    Name: Code Peer Evaluation For Longevity

    Initial State: NA

    Input: Source code (see details in section 7)

    Expected Output: Successfully convinced the participants the following:

    - The product is built in a robust manner, increasing the likelihood that it can operate for an extended period without major malfunctions.
    - The application intentionally avoids features that may lead to compatibility issues and strives to develop a project that functions effectively across various versions of operating systems and devices.

    Actual Output: Successfully convinced the participants that the application is robust enough to work without major malfunctions for a long time and the finalized product will remain compatible with recent promised operating systems and devices.

    Result: Pass

4.4 Operational and Environmental

This section corresponds to the Operational and Environmental tests in VnV Plan and Operational and Environmental requirements in SRS.

1. **NFRT-OE1**

   Name: Visual inspection for application download

   Initial State: User has a phone that uses Android 11 or above

   Input: User wants to download the application

   Expected Output: The product can be downloaded from an APK file directly

   Actual Output: The application can be downloaded from an APK file the team releases

   Result: Pass

4.5 Maintainability and Support

1. **NFRT-MS1**

   Name: Survey for maintenance time

   Initial State: Survey taker is given an account:

   - email: mtest@gmail.com
   - password: mtesting

   Input/Condition: Usability Survey in section 5

   Expected Output: Tasks are completed successfully and "Common periods of usage" question gets an average score that is great than MIN_SCORE

   Actual Output: Tasks are completed successfully and "Common periods of usage" questions get average scores of 3.2

   Result: Fail

   Reason: It seems that students may use the application after school

2. **NFRT-MS2**

   Name: Check for feature request

   Initial State: A public GitHub repo exists for this application

   Input: User goes to the GitHub repo

   Expected Output: User can read issues created by the team and also create new issues

   Actual Output: User can read issues created by the team and also create new issues

   Result: Pass

3. **NFRT-MS3**

   Name: Android version test

   Initial State: NA

   Input: The application is installed on devices with Android 11 and above version

   Expected Output: The application works without any error about compatibility

   Actual Output: The application works without any error about compatibility

   Result: Pass

4.6 Security

1. **NFRT-S1**

   Name: Access test

   Initial State: Three accounts with different accesses is ready:

   - Admin: campusconnections@gmail.com
   - User: mtest@gmail.com
   - Guest gtest@gmail.com

Input: The user starts the application with the three accounts

Expected Output: At each level of access, the application constrains the possible actions to what is specified in requirement S-A1, S-A2, S-A3.

Actual Output: Possible Actions:

- Administrator: Everything, include Add/Edit/Delete actitives
- User: Friend system, Profile system, lecture and event viewing, map system
- Guest: Public event viewing, single-user map, Profile system

Result: Pass

2. **NFRT-S2**

   Name: Special character warning

   Initial State: User starts to register

   Input: User enters nickname with some special character: ‘; DELETE *’

   Expected Output: A warning message is displayed telling the user that the special characters are not allowed and stops the user from registering

   Actual Output: A warning message is displayed telling the user that the special characters are not allowed and stops the user from registering

   Result: Pass

3. **NFRT-S3**

   Name: Email format warning

   Initial State: User starts to register

   Input: User enters a not email string in the email field: ‘SELECT * FROM TABLE’

   Expected Output: A warning message is displayed telling the user that the input is not an email address and stops the user from registering

Actual Output: A warning message is displayed telling the user that the input is not an email address and stops the user from registering

Result: Pass

4.7 Privacy

1. **NFRT-PRV1**

   Name: Code inspection for legitimate use of personal data

   Initial State: NA

   Input: Source code (see details in section 7)

   Expected Output: Successfully convinced the participants the following:

   - The usage of a user's personal information by the product abides by the Privacy Act, The Personal Information Protection and Electronic Documents Act, and Canada and Ontario's data protection laws

   Actual Output: Convinced the participants that the personal data we collect is handled legitimately

   Result: Pass

2. **NFRT-PRV2**

   Name: Code inspection for removing unused accounts

   Initial State: NA

   Input: Source code (see details in section 7)

   Expected Output: Successfully convinced the participants the following:

   - Any accounts that are not active for a long time (a semester) will be removed from the system

   Actual Output: Convinced the participants that the authentication system and database will clean inactive data every semester

   Result: Pass

3. **NFRT-PRV3**

   Name: Code Walkthrough For User Personal Data

   Input: Source code (see details in section 7)

   Expected Output: Successfully convinced the participants the following:

   - User's personal information does not appear in the database if the user did not grant permission

   Actual Output: Successfully convinced the participants that the user personal information is collected under permission only

   Result: Pass

4. **NFRT-PRV4**

   Name: Code Walkthrough For Data Transmission Encryption

   Input: Source code (see details in section 7)

   Expected Output: Successfully convinced the participants the following:

   - The product only transmits encrypted data from server to user

   Actual Output: Successfully convinced the participants that the data send from/to the server are encrypted

   Result: Pass

5. **NFRT-PRV5**

   Name: Leaving campus warning

   Initial State: User opens the map

   Input: User moves out of the campus

   Expected Output: A warning message is displayed telling the user that the map is not available out of campus

   Actual Output: A warning message is displayed telling the user that the map is not available out of campus and the user is redirected to the menu page once the warning message is closed

   Result: Pass

4.8 User Safety

1. **NFRT-US1**

   Name: Warning when starting AR camera

   Initial State: User allows the application to use camera

   Input: User turns on the AR camera

   Expected Output: A warning telling the user to be aware of their surroundings is displayed upon start-up of the camera

   Actual Output: A warning telling the user to be aware of their surroundings is displayed upon start-up of the camera

   Result: Pass

4.9 Culture

1. **NFRT-CUL1**

   Name: Survey for feedback on cultural requirements

   Initial State: Survey taker is given an account:

   - email: mtest@gmail.com
   - password: mtesting

   Input/Condition: Usability Survey in section 5

   Expected Output: Tasks are completed successfully and "Cultural Friendliness" question gets an average score that is great than MIN_SCORE

   Actual Output: Tasks are completed successfully and "Cultural Friendliness" questions get average scores of 4.8

   Result: Pass

4.10 Compliance

1. **NFRT-COM1**

   Name: Code walkthrough on compliance requirements

   Initial State: NA

   Input: Source code (see details in section 7)

Expected Output: Successfully convinced the participants the following:

- The data collected will be handled as per the same legal requirements for the university
- The application can abide by the guidelines set by university staff

Actual Output: Successfully convinced the participants the following:

- The data collected will be handled as per the same legal requirements for the university
- The application can abide by the guidelines set by university staff

Result: Pass

5 Usability Survey Result

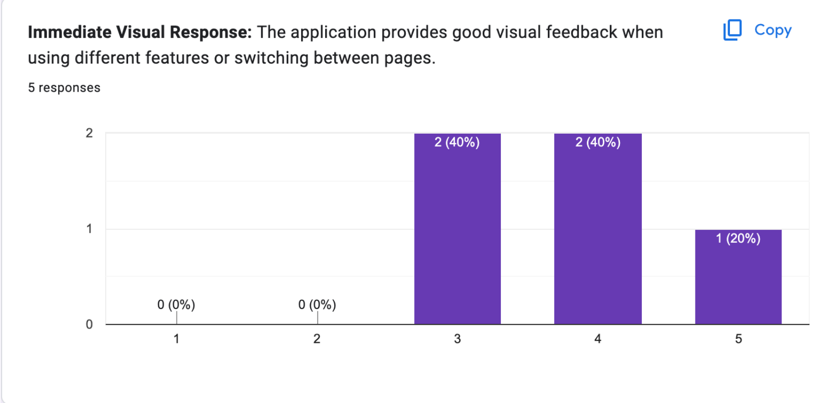
We first conduct the survey on 5 students, and it seems that we are getting repetitive feedback, so we assume this is already a good sample size. The result of the survey is the following:

5.1 Tasks:

- Administrator: 1 tester - all tasks are completed successfully
- User: 3 testers - all tasks are completed successfully
- Guest: 1 tester - all tasks are completed successfully

5.2 Rating

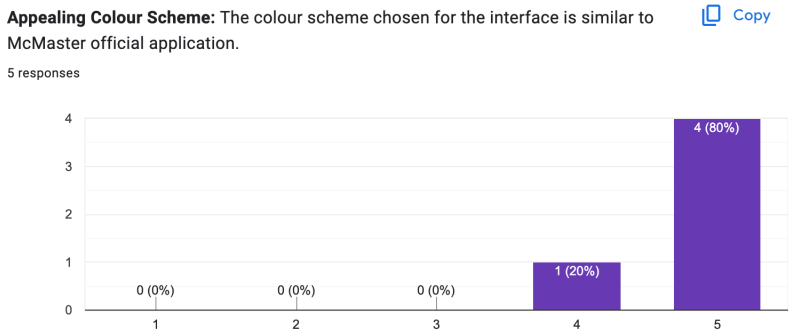
- Immediate Visual Response when Clicking: **3.8**



Immediate Visual Response: The application provides good visual feedback when using different features or switching between pages.

5 responses

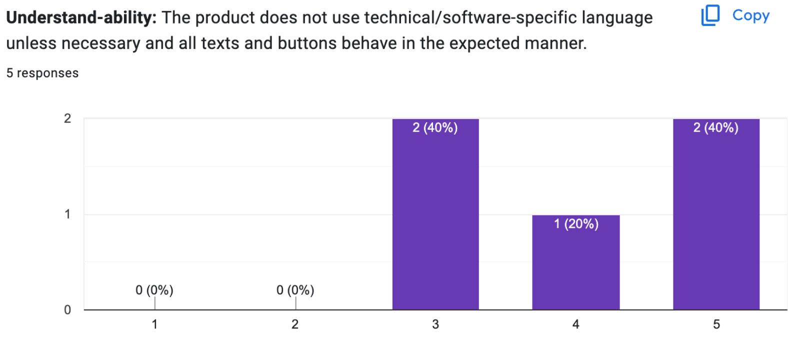
- Appealing Colour Scheme: **4.8**



Appealing Colour Scheme: The colour scheme chosen for the interface is similar to McMaster official application.

5 responses

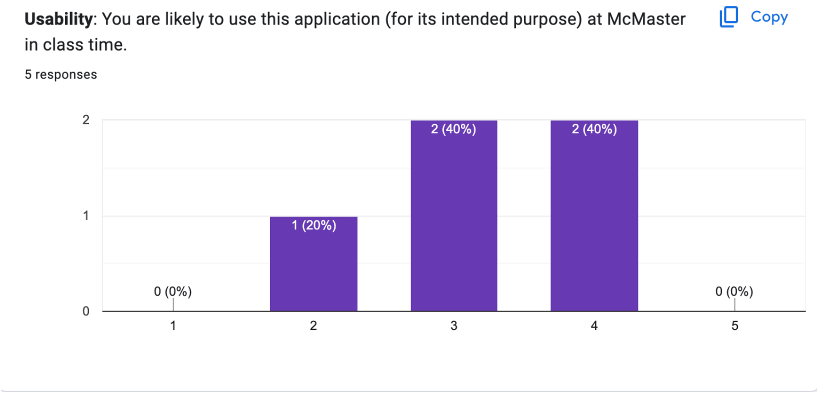
- No Technical or Software-specific Language: **4**



Understand-ability: The product does not use technical/software-specific language unless necessary and all texts and buttons behave in the expected manner.

5 responses

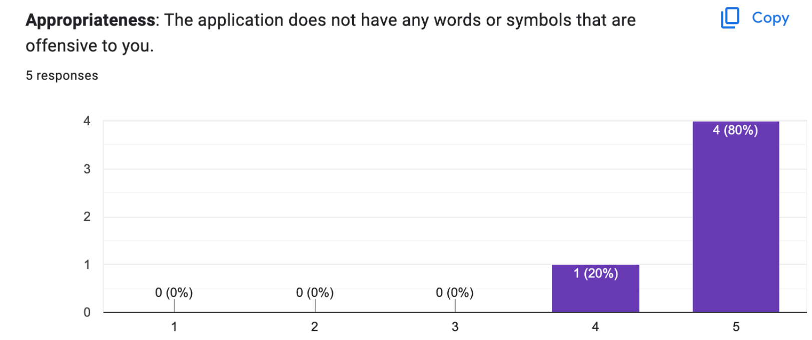
- Common periods of usage: **3.2, Fail**



Usability: You are likely to use this application (for its intended purpose) at McMaster in class time.

5 responses

- Cultural Friendliness: **4.8**



Appropriateness: The application does not have any words or symbols that are offensive to you.

5 responses

5.3 Open-ended Question

- Most difficult to use feature: **AR camera, pin, chatting**

  1. The most difficult to use feature is probably the AR camera feature as it requires the user to use look through their phone camera while in buildings on campus. Additionally, it is also probably the most cool aspect of the app.
  2. The lecture list isn't intuitive, there should be some calendar feature to show what days each lecture is running instead of just clicking on them, and maybe some more filters for building/location/course content.

3. The ”pin” feature being hidden in the settings and the ”AR Camera” not providing any feedback as to why we weren’t seeing any were both confusing
4. The pinned/bookmarked events and lectures show up in a location (in settings) that is somewhat difficult to find. You also can’t tap them to see more information about them on the pinned screen which is inconvenient.
5. Friend and chatting is unintuitive, don’t know who I’m chatting with.

- Most likely to use feature: **Lecture/Event, AR camera**
   1. Probably also the feature for looking up lecture events and times. This is useful as I can just open the app and view where specific lectures are located and choose to attend them or not.
   2. I would use the AR camera feature to see interesting building history in between classes.
   3. The feature that provided a schedule of all of the events was very nice
   4. I could see the list of events at different locations being pretty useful, would be a good way to keep track of what’s going on at Mac all in one place.
   5. Events can be useful to see what is going on on campus and save events we want to go to

6 Load Test

This section covers results load testing of the backend server in details. Testing was performed using Apache JMeter to simulate multiple concurrent users accessing the backend chat server. The objective of the test was to find the upper bound of concurrent users that can access the server without it crashing. We discovered that it is rather difficult to crash a server being hosted on a cloud platform as there are several safeguards that prevent that. Instead, we noticed that if too many connections are being made too quickly, then the server will just reject the most recently requested connections.

We tested with various amounts of simulated users and observed how the server behaved. The results are plotted in the chart below.

No. of Users	Server Rejected Connections?	Server Crashed?
1	No	No
10	No	No
100	No	No
500	No	No
1000	No	No
5000	Yes	No
10000	Yes	No

Table 1: Load Testing Results

We also observed analytics data from Microsoft Azure concerning the average response time of the server in relation to the number of requests.

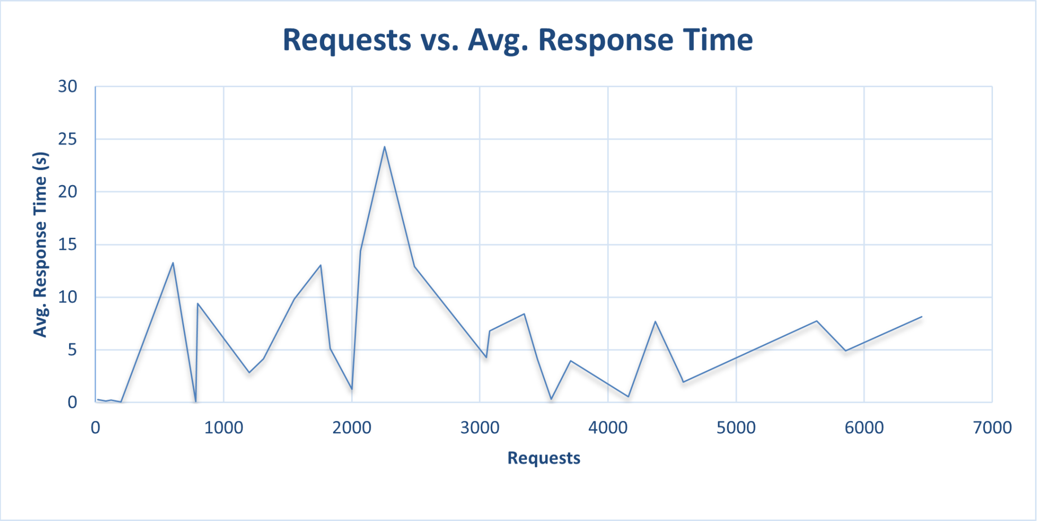


Figure 1: Requests vs. Response Time

There does not appear to be an observable trend. We suspected this was due to limitations in our testing setup. Upon further investigation we discovered that our Azure app server has a limit of around 2000 connections at the basic tier which is the tier we used. We also discovered that JMeter could not create more than 5000 connections per IP. We concluded that a workaround for these limitations would be out of scope. It would both require paying for a higher tier of Azure service and the creation of a custom distributed JMeter testing setup which would be infeasible in our timeframe.

7 Non dynamic Tests Result

The weekly meetings mentioned in the VnV Plan, such as the weekly code review and TA feedback sessions, will not be detailed in this section due to their frequency and the fact that many of the conclusions drawn in these meetings have already been integrated and addressed in other sections, particularly non-requirements. However, their meeting minutes are accessible on our GitHub repository for those interested in further details.

Table 2: Nondynamic Test Result

Test Method	**Result**	**Related Test(s)**
Code Inspection for user data	• User permission: Pass, the application forces to user to sign a consent form when they register • Legitimate use of data: Pass, the supervisor points out that data are not used out of the application for any business purpose • Compliance: Pass, nothing violates McMaster privacy polices	NFRT-P3, NFRT-PRV1, NFRT-COM1
Database Walkthrough	• Capacity: Pass, the free plan has a storage of 1GB data, which is enough for our users at this stage • Clean: Pass, Firebase Authentication support some cloud functions to clean inactive users	NFRT-P15, NFRT-PRV2
Server & Data Transmission Walkthrough	• Restart: Pass, the server is set to restart when it crashes • Encryption: pass, the data is encrypted in https	NFRT-P4, NFRT-P12
Longevity Peer Evaluation	• Longevity: Pass, nothing in the code is dependent on the version of the device	NFRT-P17

Table 3: Nondynamic Test Result Cont

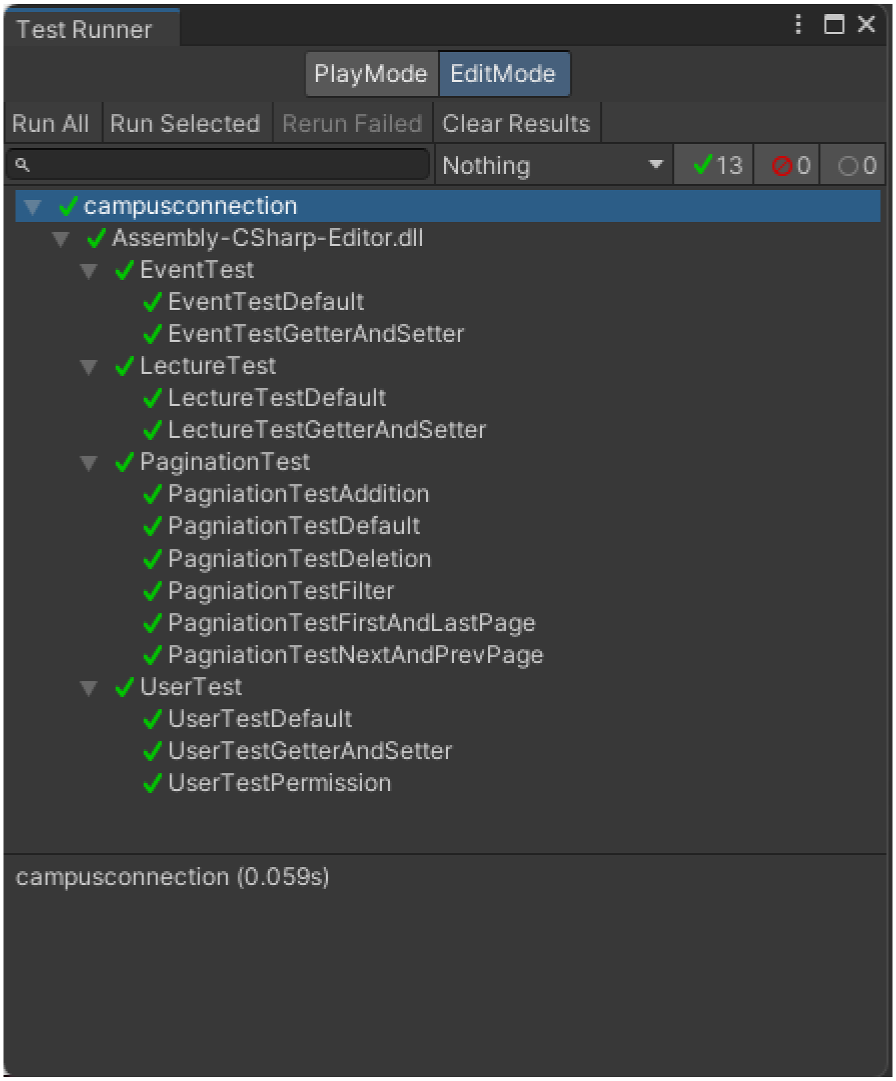
Test Method	Result	Related Test(s)
Code Walkthrough with Primary Reviewers	• Issues: Pass, ia new building can be added like a game object in the project	NFTR-P16
GitHub Walkthrough with Reviewers	• Issues: Pass, issues are public to read and write in the repo • Instructions: Pass, there are instructions for developers and users in the repo	NFRT-UH2, NFRT-MS2

8 Comparison to Existing Implementation

Not applicable to this project since there is no existing implementation.

9 Unit Testing

The following section outlines the results of unit testing. The process and test performed follow the VnV Plan. Due to the time constraint, we skipped the unit tests for Database and Authentication modules, but we will add unit tests for them in Revision 1. Other than the two modules, all the tests pass, which indicates the low level modules of the system is implemented correctly and robustly.



```
▼ ✓ campusconnection
    ▼ ✓ Assembly-CSharp-Editor.dll
        ▼ ✓ EventTest
              ✓ EventTestDefault
              ✓ EventTestGetterAndSetter
        ▼ ✓ LectureTest
              ✓ LectureTestDefault
              ✓ LectureTestGetterAndSetter
        ▼ ✓ PaginationTest
              ✓ PagniationTestAddition
              ✓ PagniationTestDefault
              ✓ PagniationTestDeletion
              ✓ PagniationTestFilter
              ✓ PagniationTestFirstAndLastPage
              ✓ PagniationTestNextAndPrevPage
        ▼ ✓ UserTest
              ✓ UserTestDefault
              ✓ UserTestGetterAndSetter
              ✓ UserTestPermission

campusconnection (0.059s)
```

9.1 User

This section contains tests related to the module: User in module hierarchy.

1. **FRT-M1-1**

   Name: Successful default user creation

   Initial State: NA

   Input: User identifier, email: unitTest@test.com

Expected Output: A user is successfully created with following attributes:

- email: unitTest@test.com
- nickname: ‘’
- photoURI: null
- program: ‘’
- level: 0
- lectures: empty list of string
- event: empty list of string
- friends: empty list of string
- friendInvitation: empty list of string

Actual Output: Same as expected output

Result: Pass

2. **FRT-M1-2**

   Name: Successful new user creation

   Initial State: NA

   Input: User:

   - email: unitTest@test.com
   - nickname: ‘Mr. Unit Test’
   - photoURI: http://example/com/photo.jpg
   - program: ‘software engineering’
   - level: 1
   - lectures: {‘Lecture 1’, ‘Lecture 2’}
   - event: {‘Event 1’, ‘Event 2’}
   - friends: {‘unitTest1@test.com’, ‘unitTest2@test.com’}
   - friendInvitation: {‘unitTest3@test.com’}

   Expected Output: A user is successfully created with attributes above

   Actual Output: Same as expected output

   Result: Pass

9.2 Event

This section contains tests related to the module: Event in module hierarchy.

1. **FRT-M2-1**

   Name: Successful default event creation

   Initial State: NA

   Input: Event identifier, name: unit test event

   Expected Output: An event is successfully created with following attributes:

   - name: unit test event
   - description: PLACEHOLDER
   - organizer: PLACEHOLDER
   - duration: DEFAULT_DURATION
   - time: DEFAULT_TIME
   - isPublic: false
   - location: PLACEHOLDER

   Actual Output: Same as expected output

   Result: Pass

2. **FRT-M2-2**

   Name: Successful new event creation

   Initial State: NA

   Input: Event:

   - name: unit test event2
   - description: A unit test sample event
   - organizer: Team 2
   - duration: 30
   - time: 1708922625
   - isPublic: true

- location: Online

Expected Output: An event is successfully created with attributes above

Actual Output: Same as expected output

Result: Pass

9.3 Lecture

This section contains tests related to the module: Lecture in module hierarchy.

1. **FRT-M3-1**

   Name: Successful default lecture creation

   Initial State: NA

   Input: Lecture identifier, code: UNIT TEST3

   Expected Output: A lecture is successfully created with following attributes:

   - code: UNIT TEST3
   - name: PLACEHOLDER
   - instructor: PLACEHOLDER
   - time: PLACEHOLDER
   - location: PLACEHOLDER

   Actual Output: Same as expected output

   Result: Pass

2. **FRT-M3-2**

   Name: Successful new lecture creation

   Initial State: NA

   Input: Lecture:

   - code: UNIT TEST3
   - name: Unit test Lecture

- instructor: Team 2
- time: 12:30 - 13:30, Mon
- location: Online

Expected Output: A lecture is successfully created with attributes above

Actual Output: Same as expected output

Result: Pass

9.4 Pagination

This section contains tests related to the module: Pagination and Filter in module hierarchy.

1. **FRT-M4-1**

   Name: Successful pagination creation

   Initial State: NA

   Input: List of Pizza:

   - medium, cheese
   - large, veggie

   pageCount = 2, filterBy and filterString are null

   Expected Output: A pagination class is created with following attributes:

   - entryList: List of pizza from input
   - filteredList: same as entryList
   - maxPage: 1
   - currentPage: 1
   - filterBy: null
   - filterString: null

   Actual Output: Same as expected output

   Result: Pass

2. **FRT-M4-2**

   Name: Successful pagination creation with filter

   Initial State: NA

   Input: List of Pizza:

   - medium, cheese
   - large, veggie
   - medium, pepperoni

   pageCount = 2, filterBy = ‘size’ and filterString = ‘medium’

   Expected Output: A pagination class is created with following attributes:

   - entryList: List of pizza from input
   - filteredList: List of medium pizza only
   - maxPage: 1
   - currentPage: 1
   - filterBy: ‘size’
   - filterString: ‘medium’

   Actual Output: Same as expected output

   Result: Pass

3. **FRT-M4-3**

   Name: Successful pagination addition

   Initial State: List of Pizza:

   - medium, cheese
   - large, veggie

   pageCount = 2, filterBy and filterString = null

   Input: New pizza added to the list:

   - medium, pepperoni

Expected Output: A pagination class is created with following attributes:

- entryList: List of all three pizza
- filteredList: same as entryList
- maxPage: 2
- currentPage: 1
- filterBy: null
- filterString: null

Actual Output: Same as expected output

Result: Pass

4. **FRT-M4-4**

   Name: Successful pagination deletion

   Initial State: List of Pizza:

   - medium, cheese
   - large, veggie
   - medium, pepperoni

   pageCount = 2, filterBy and filterString = null

   Input: New pizza to be deleted from the list:

   - medium, pepperoni

   Expected Output: A pagination class is created with following attributes:

   - entryList: List of two pizza (cheese and veggie)
   - filteredList: same as entryList
   - maxPage: 1
   - currentPage: 1
   - filterBy: null
   - filterString: null

Actual Output: Same as expected output

Result: Pass

5. **FRT-M4-5**

   Name: Next Page

   Initial State: List of Pizza:

   - medium, cheese
   - large, veggie
   - medium, pepperoni

   pageCount = 2, filterBy and filterString = null

   Input: Call next page method twice

   Expected Output: Current page is updated only once:

   - currentPage: 2

   Actual Output: Same as expected output

   Result: Pass

6. **FRT-M4-6**

   Name: Previous Page

   Initial State: List of Pizza:

   - medium, cheese
   - large, veggie
   - medium, pepperoni

   pageCount = 2, filterBy and filterString = null
   Call next page method first

   Input: Call previous page method twice

   Expected Output: Current page is updated only once:

   - currentPage: 1

Actual Output: Same as expected output

Result: Pass

7. **FRT-M4-7**

   Name: Last Page

   Initial State: List of Pizza:

   - medium, cheese
   - large, veggie
   - medium, pepperoni

   pageCount = 2, filterBy and filterString = null

   Input: Call last page method

   Expected Output: Current page is updated:

   - currentPage: 2

   Actual Output: Same as expected output

   Result: Pass

8. **FRT-M4-8**

   Name: First Page

   Initial State: List of Pizza:

   - medium, cheese
   - large, veggie
   - medium, pepperoni

   pageCount = 2, filterBy and filterString = null
   Call next page method first

   Input: Call first page method

   Expected Output: Current page is updated:

   - currentPage: 1

   Actual Output: Same as expected output

   Result: Pass

9.5 Authentication

This section contains tests related to the module: Authentication in module hierarchy.

1. **FRT-M5-1**

   Name: Get current user email

   Initial State: Current logged in user email: 'unitTest@test.com'

   Input: The user navigates to the User Profile Page.

   Expected Output: The current user email (unitTest@test.com) is returned in the email section of the profile.

   Actual Output: The current user email (unitTest@test.com) is returned correctly in the profile.

   Result: Pass

2. **FRT-M5-2**

   Name: Is email verified

   Initial State: Current logged in user is email verified: true

   Input: The user navigates to the User Profile Page.

   Expected Output: The verify email button is no longer visible.

   Actual Output: The verify email button is no longer visible.

   Result: Pass

3. **FRT-M5-3**

   Name: Verify email

   Initial State: Current logged in user email: 'unitTest@test.com'

   Input: Verify email button is pressed.

   Expected Output: An email is sent to the email address to verify the email

   Actual Output: unitTest@test.com recieves the email verification email, which allows them to click on a link that will verify their email.

   Result: Pass

9.6 Database

This section contains tests related to the module: Database in module hierarchy.

1. **FRT-M6-1**

   Name: Retrieve Values From Database

   Initial State: Logged in with student account.

   Input: Retrieve the data under '/events/public/EXPO/name'

   Expected Output: The name of the event 'EXPO' is returned.

   Actual Output: The name of the event 'EXPO' is returned.

   Result: Pass

2. **FRT-M6-2**

   Name: Set Value to Database

   Initial State: Logged in with admin account.

   Input: Set the data under the following entry:

   - path: '/events/public/EXPO/description'
   - value: 'EXPO, yes!'

   Expected Output: The corresponding value changes to the new value

   Actual Output: The database is updated to the new value.

   Result: Pass

3. **FRT-M6-3**

   Name: Delete Data From Database

   Initial State: Logged in with Admin account.

   Input: Delete the entry at: '/events/public/EXPO'

   Expected Output: The corresponding event is removed from database.

   Actual Output: The event is removed from database.

   Result: Pass

10 Changes Due to Testing

10.1 Changes due to Rev 0 demo feedback

In Revision 0 of the demonstration, our instructor, Dr. Smith, pointed out that certain features, such as access control and the event list page, had not yet been implemented. These aspects were accorded higher priority in the development plan and have been successfully completed.

10.2 Changes due to supervisor's feedback

During the weekly meeting, our supervisor, Dr. Yuan, emphasized the necessity of reducing the sample size for the usability survey at this stage, which had previously been set at 50. Additionally, Dr. Yuan suggested incorporating tasks into the survey to facilitate thorough testing of the application by users. Consequently, the team collectively decided to integrate tasks and open-ended questions into the usability survey. This adjustment aims to gather more insightful feedback, discerning which aspects users like most and pinpointing features requiring improvement from a usability perspective.

10.3 Changes due to usability test result

The only test fails is the "Common periods of usage" rating question. It has come to light that survey participants express a preference for utilizing the application after school so that they won't miss any events happen in the evening. This observation is directly linked to maintenance requirement MS-M1. The team will change that major update time in that requirement to be summer break or weekend midnight to affect fewer users.

The absence of instructions for the AR camera has been repeatedly highlighted in the feedback. Consequently, the team will resolve this issue by adding instructions when user clicks on the help button.

Participants have noted that bookmarked lectures and events are not readily accessible, as they are currently hidden within the settings. In response to this feedback, the team will create a link to the bookmarked lectures and events page directly from the list pages.

Each of the aforementioned changes will be documented as a separate issue in the GitHub repo for future discussion.

10.4 Changes due to load test result

The load test in section 6 shows that NFRT-P16 passes without any issues – with fewer than 1000 users connecting to the application, the server handles the load effectively, as indicated by the nearly negligible response time observed in the figure. However, it is crucial to note that upon scaling up to accommodate approximately 40,000 concurrent requests after Revision 1, the server begins to reject connections and getting a significant increase in response time.Therefore, this is a good warning for the team regarding the need to scale the server infrastructure adequately to support a larger user base.

10.5 Changes due to failing tests

NFRT-P10 fails due to a change of the scope. The team will update SRS and VnV Plan so that internet connection lost warning is no longer a requirement.

10.6 Changes due to user enumeration vulnerability

User enumeration occurs when a malicious actor utilizes brute-force techniques to guess or confirm valid users within the system. In our application, this vulnerability can occur during the login process, where attackers can determine the validity of an email by analyzing error messages. Therefore, the team proposes implementing a more generic error message during login attempts (FRT-UA4). Additionally, a new test should be added to verify unsuccessful login attempts using an email not registered in the system.

On the other hand, the password recovery page effectively prevents this type of attack by only checking for email format without revealing whether the email exists in the system.

Moreover, there are concerns regarding attackers exploiting server response times to detect user existence but that does not seem to be a problem with our application as Firebase's real-time database responds to valid and invalid login attempts with indistinguishable time differences.

Considering the project's scope and potential visibility (most of our customers will be students and department staff), enumeration attacks can be identified quickly. With majority of users having McMaster email and non-McMaster emails having guest privilages, this mitigates possible user enumeration attacks significantly. In addition, the email verification process will allow Campus Connections team to easily identify suspicious activity. With the aforementioned measures, the application sufficiently safeguards against such attacks.

11 Automated Testing

The majority of tests are run from local machine due to Unity and unity package size and behavior. When attempting to setup the automated tests, there were issues due to how Vuforia was not a package that could be imported automatically due to how they host their library. Communication with the database occurs asynchonously, which cause problems when testing automatically using CI/CD. This caused problems as the program failed to build automatically. The package was too big to be included in the repository, and Large File Storage had limits. This also resulted in many tests being done locally.

11.1 Testing Instruments

The tests were performed using Unity Testing Framework, NUnit made for Unity. The test files are located in multiple files with the respective names of the modules they are testing under Assets/Editor. OpenCover was planned for coverage testing and will produce a report into XML file. These results are visible after the automated tests are completed. The CI with GitHub Actions will be completed in phase 2.

11.2 Linters

Linters were added into CI using Super-linter. This is an open source linter maintained and used by many people, which shows its reliability. The linter only checks the new contents in the repository, making it efficient and concise. The code linter has been active and we have fixed issues brought up by the test results such as duplicate code and formatting issues. The results of linters are visible under the GitHub Actions tab.

12 Trace to Requirements

See Traceability tables between Test Cases and Requirements in VnV Plan

13 Trace to Modules

See Traceability tables between Test Cases and Modules in VnV Plan, notice that M5 and M6 are not completed yet in this iteration.

14 Code Coverage Metrics

The code coverage report can be seen in the zip folder. The report contains all files in the project, which some are libraries or Unity files. The relevant code coverage data is shown in the table below.

Due to the time constraints, authentication and database tests were pushed to phase 2. The testing files are located in folder.

Module	Code Coverage	Test Module
User	100%	UserTest
Pagination	95.8%	PaginationTest
Event	100%	EventTest
Lecture	100%	LectureTest

Appendix

14.1 Symbolic Parameters

Check the Symbolic Parameter Table in VnV Plan for more details.

The following table only contains symbols that appear only in the report

Table 4: **Symbolic Parameter Table**

Symbolic Parameter	**Description**	**Value**
Femail	The friend email, an email that exists in the application database, usually used with target email to test user communication	Value depends on the test it belongs to
Temail	The target email, an email that exists in the application database, usually used with friend email to test user communication	Value depends on the test it belongs to

14.2 Reflection

The information in this section will be used to evaluate the team members on the graduate attribute of Reflection. Please answer the following question:

1. In what ways was the Verification and Validation (VnV) Plan different from the activities that were actually conducted for VnV? If there were differences, what changes required the modification in the plan? Why did these changes occur? Would you be able to anticipate these changes in future projects? If there weren't any differences, how was your team able to clearly predict a feasible amount of effort and the right tasks needed to build the evidence that demonstrates the required quality? (It is expected that most teams will have had to deviate from their original VnV Plan.)

The execution of our testing plan was very different from how we initially envisioned in our VnV Plan. The VnV plan was written before any implementation had been done. This resulted in us making several assumptions about how the final product would work. Of course, as our project evolved, many of our assumptions would need to be changed.

One of the main ways our actual VnV testing effort varied was the scope of what was tested. Due to time constraints, some of our planned features needed to be cut. This resulted in some proposed test cases no longer being applicable. While it is natural for requirements and scope to change throughout a project's life, the invalidation of these tests could potentially have been mitigated by creating the VnV Plan when the structure of the project is more concrete. In future VnV planning, we would either do our VnV planning after completing some of the implementation, or by iterating on the previous VnV by making amendments and revisions.

Another thing that impacted our VnV testing effort was our level of familiarity with the testing tools available and how they interacted with our technology stack. For example, we initially wanted to test our database-facing front-end code using the Unity Test Framework (UTF). Due to our lack of experience with UTF and Firebase, we assumed that it would be possible to test Firebase code using UTF. This ended up not being the case and the tests that were planned were scrapped or pushed to our feature waiting list. In future VnV planning, we would put more effort into learning the tools and technology needed to test our project before coming up with test plans.

Comparing how our testing efforts went with how we planned them taught us that it is best to be conservative with the scope of features as some development time must be allocated to testing those features. We also learned to value iterative improvement by keeping the documentation "alive" with revisions along with the project. Lastly, we gained experience with a wide variety of testing tools and techniques that will be valuable in future VnV planning and execution.