

# Verification and Validation Report: Software Engineering

Team #2, Campus Connections

Waseef Nayeem

Zihao Du

Matthew Miller

Firas Elayan

Abhiram Neelamraju

Michael Kim

March 5, 2024

# 1 Revision History

Date	Version	Notes
Mar 4	1.0	Add functional requirements evaluation

## 2 Symbols, Abbreviations and Acronyms

symbol	description
JMeter	Load testing tool for analyzing and measuring the performance
SRS	Software Requirements Specification
UI	User Interface
VnV	Verification and Validation

# Contents

<b>1</b>	<b>Revision History</b>	<b>i</b>
<b>2</b>	<b>Symbols, Abbreviations and Acronyms</b>	<b>ii</b>
<b>3</b>	<b>Functional Requirements Evaluation</b>	<b>1</b>
3.1	Pre-Registration Settings . . . . .	1
3.2	User Account . . . . .	2
3.3	Social Networking System . . . . .	6
3.4	Lectures and Events . . . . .	9
3.5	AR Camera . . . . .	15
3.6	Map and Location . . . . .	16
<b>4</b>	<b>Nonfunctional Requirements Evaluation</b>	<b>17</b>
4.1	Usability . . . . .	17
4.2	Performance . . . . .	17
4.3	etc. . . . .	17
<b>5</b>	<b>Comparison to Existing Implementation</b>	<b>17</b>
<b>6</b>	<b>Unit Testing</b>	<b>17</b>
<b>7</b>	<b>Changes Due to Testing</b>	<b>17</b>
<b>8</b>	<b>Automated Testing</b>	<b>17</b>
<b>9</b>	<b>Trace to Requirements</b>	<b>18</b>
<b>10</b>	<b>Trace to Modules</b>	<b>20</b>
<b>11</b>	<b>Code Coverage Metrics</b>	<b>21</b>

## List of Tables

1	Traceability Between Functional Test Cases and Functional Requirements, FR-1 to FR-3-5 . . . . .	18
2	Traceability Between Functional Test Cases and Functional Requirements, FR-3-6 to FR-6 . . . . .	19

## 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 with corresponding information is created in the database with the account initialized to INITIAL\_USER\_STATE

**Results:** Pass

## 2. FRT-UA2

**Name:** Unsuccessful 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: 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



**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 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 nickname: QTesting

**Expected Output:** Password is successfully reset

**Actual Output:** Password is successfully reset

**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](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:** That email above is verified as a valid email address

**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
- newProgram: Computer Science

**Input:** New Profile:

- newProgram: Computer Science
- newLevel: 4

**Expected Output:** The program and level are updated

**Actual Output:** The program and level are updated

**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:

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

**Input:** A valid email to send the request:

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

**Expected Output:** A Request is sent to the target user

**Actual Output:** A Request is sent to the target user

**Results:** Pass

## 2. FRT-SN2

**Name:** Friend Request Acceptance

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

- Semail: 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 (Semail) to the target account (Temail):

- Semail: 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 is deleted from the 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 that event once they click on the message

**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 that lecture once they click on the message

**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:** Save Event

**Initial State:** A sample event:

- Name: EXPO

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

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

**Actual Output:** The event is saved to the user's event list

**Results:** Pass

### 2. FRT-LE2

**Name:** Unsave Event

**Initial State:** A sample event that is already been saved:

- Name: EXPO

**Input:** On the event page, user clicks on the unsave 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 is removed from the user's event list

**Results:** Pass

### 3. FRT-LE3

**Name:** Save Lecture

**Initial State:** A sample lecture:

- Code: SFWRENG 4G06

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

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

**Actual Output:** The lecture is saved to the user's lecture list

**Results:** Pass

### 4. FRT-LE4

**Name:** Unsave Lecture

**Initial State:** A sample lecture that is already been saved:

- Code: SFWRENG 4G06

**Input:** On the lecture page, user clicks on the unsave 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 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

**Expected Output:** The event is added to the event list

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

**Results:** Pass

#### 6. FRT-LE6

**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 the new one

**Results:** Pass

#### 7. FRT-LE7

**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 event is deleted and disappears from the list

**Results:** Pass

#### 8. FRT-LE8

**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 is added to the lecture list

**Results:** Pass

#### 9. 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 the new one

**Results:** Pass

#### 10. 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 lecture is deleted and disappears from the list

**Results:** Pass

#### 11. 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

**Results:** Pass

12. **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

**Results:** Pass

13. **FRT-LE13**

**Name:** Lecture Filter by Code

**Initial State:** Some software engineering lecture exists:

- SFWRENG 4G06
- SFWRENG 4E03

**Input:** Filter:

- FilterString: SFWRENG

**Expected Output:** All lectures which do not contain the FilterString in the code are removed from the list

**Actual Output:** All lectures which do not contain the FilterString in the code are removed from the list

**Results:** Pass

14. **FRT-LE14**

**Name:** Event Filter by Name

**Initial State:** Some job fair event exists:

- Job Fair: March 4
- Job Fair: March 10

**Input:** Filter:

- FilterString: Job Fair

**Expected Output:** All events which do not contain the FilterString in the name are removed from the list

**Actual Output:** All events which do not contain the FilterString in the name are removed from the list

**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 is recognized with name and description shown as an AR object

**Actual Output:** The building is recognized with 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 Usability

### 4.2 Performance

### 4.3 etc.

## 5 Comparison to Existing Implementation

This section will not be appropriate for every project.

## 6 Unit Testing

## 7 Changes Due to Testing

[This section should highlight how feedback from the users and from the supervisor (when one exists) shaped the final product. In particular the feedback from the Rev 0 demo to the supervisor (or to potential users) should be highlighted. —SS]

## 8 Automated Testing

## 9 Trace to Requirements

Test IDs	Functional Requirement IDs												
	FR1-1	FR2-1	FR2-2	FR2-3	FR2-4	FR2-5	FR2-6	FR2-7	FR3-1	FR3-2	FR3-3	FR3-4	FR3-5
FRT-PR1	X												
FRT-PR2	X												
FRT-UA1		X											
FRT-UA2		X											
FRT-UA3				X									
FRT-UA4				X									
FRT-UA5			X										
FRT-UA6					X								
FRT-UA7						X							
FRT-UA8							X						
FRT-UA9								X					
FRT-SN1									X				
FRT-SN2									X				
FRT-SN3									X				
FRT-SN4										X			
FRT-SN5											X		
FRT-SN6												X	
FRT-SN7													X

Table 1: Traceability Between Functional Test Cases and Functional Requirements, FR-1 to FR-3-5

Test IDs	Functional Requirement IDs											
	FR3-6	FR4-1	FR4-2	FR4-3	FR4-4	FR4-5	FR4-6	FR4-7	FR4-8	FR5-1	FR5-2	FR6-1
FRT-SN2	X											
FRT-SN3	X											
FRT-LE1		X										
FRT-LE2		X										
FRT-LE3			X									
FRT-LE4			X									
FRT-LE5				X								
FRT-LE6				X								
FRT-LE7				X								
FRT-LE8					X							
FRT-LE9					X							
FRT-LE10					X							
FRT-LE11						X						
FRT-LE12							X					
FRT-LE13								X				
FRT-LE14									X			
FRT-AR1										X		
FRT-AR2										X		
FRT-AR3											X	
FRT-MAP1												X
FRT-MAP2												X

Table 2: Traceability Between Functional Test Cases and Functional Requirements, FR-3-6 to FR-6



## 10 Trace to Modules

## 11 Code Coverage Metrics

### References

## Appendix — 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.)