

Test Case Descriptions & Final Results

CF1.1: Core Functionality - Event Creation Test 1

CF1.2: This test will test that the functionality of task creation works properly.

CF1.3: The user will be able to navigate the “Event creation” page and shall be able to create an event with a given name, starting time and ending time or duration.

CF1.4: Input will be task name, and either a duration of task or a combination of starting time and ending time.

CF1.5: Output will be Incomplete task data stored in memory with a name and task type.

CF1.6: Normal

CF1.7: Whitebox

CF1.8: Functional

CF1.9: Unit

Result: Task successfully created with required data and stored in database.
Pass

CF2.1: Core Functionality - Event Creation Test 2

CF2.2: This test will test that the functionality of task type creation works properly.

CF2.3: The user will be able to select the “duration” drop down in the task creation page and the app will pop up a “maximum duration” and “minimum duration” option for the user to select. This selection will provide the user with an input box that takes integers only.

CF2.4: Input will be task name, duration radio button.

CF2.5: Output will be incomplete task data stored in memory with a name and task type and duration. App should not allow user to enter a non-integer in the duration input box.

CF2.6: Abnormal

CF2.7: Whitebox

CF2.8: Functional

CF2.9: Unit

Result: After multiple iterations of the task form, the feature to be tested here is no longer included. N/A

CF3.1: Core Functionality - Event Completion Test

CF3.2: This test will test that the functionality of completing tasks works properly.

CF3.3: The user will be able to swipe right on a task on the home page to mark it as complete.
CF3.4: Input will be a "right swipe" on an incomplete task.
CF3.5: Output will be visual and database reflection of the task being completed.
CF3.6: Abnormal
CF3.7: Whitebox
CF3.8: Functional
CF3.9: Unit
Result: Tasks are now able to be marked "completed" by the user by swiping right on them from the home page. Pass

CF4.1: Core Functionality - Event Creation Test 3

CF4.2: This test will test that the functionality of task creation works properly.
CF4.3: The user will be able to place an event on a specific day of the week or day of month. Event creation shall allow event recurrence: Including daily recurrence, weekly recurrence, and monthly recurrence.
CF4.4: Input will be task name, task type, and one of many recurrence options (as described above).
CF4.5: Output will be incomplete task data stored in memory with a name and task type which includes recurrence. Repeat this test for every recurrence option.
CF4.6: Normal
CF4.7: Whitebox
CF4.8: Functional
CF4.9: Unit
Result: Scope was changed to no longer include recurrence option. N/A

DIS1.1: Display Test 1

DIS1.2: Ensure that all core functionality components exist cohesively in a way which is visually uncluttered.
DIS1.3: Run application, act as a user, pop up different screens and possible combinations of screens and make sure that everything looks good.
DIS1.4: Input will be taps on the screen.
DIS1.5: Output will be visually uncluttered UI.
DIS1.6: Abnormal
DIS1.7: Whitebox
DIS1.8: Performance
DIS1.9: Integration

Result: Working as intended. Pass

SOC1.1: Friend Code Test 1

SOC1.2: This test will ensure that users are able to add friends via a friend code.

SOC1.3: This test will consist of entering a friend code into the app, which will link two users as 'friends' on the app.

SOC1.4: Input will be user 2 entering the friend code of user 1 into the app.

SOC1.5: Output will be some sort of positive indicator that users 1 and 2 are linked as friends in the app.

SOC1.6: Normal

SOC1.7: Blackbox

SOC1.8: Functional

SOC1.9: Integration

Result: Our application was unable to make this functional. Fail

SOC2.1: Task Sharing Test 1

SOC2.2: This test will ensure that a user's tasks are visible to the social groups they chose to share them with.

SOC2.3: This test will consist of user 1 creating a task and sharing it with select social groups.

SOC2.4: Input will be a user 1's task information including which social groups they want to share it with.

SOC2.5: Output will be the task information on the calendar of user 2, given that user 2 is in the social group that user 1 shared the task with.

SOC2.6: Normal

SOC2.7: Blackbox

SOC2.8: Functional

SOC2.9: Integration

Result: Our application was unable to make this functional. Fail

SOC3.1: Task Sharing Test 2

SOC3.2: This test will ensure that a user's tasks are not visible to friends that aren't in the social groups that it was shared with.

SOC3.3: This test will consist of a user creating a task and sharing it with select social groups.

SOC3.4: Input will be a user 1's task information including which social groups they want to share it with.

SOC3.5: Output will be the task information NOT appearing to user 2. User 2 should be friends with user 1, but not part of the social group that user 1 shared the task with.

SOC3.6: Normal

SOC3.7: Blackbox

SOC3.8: Functional

SOC3.9: Integration

Result: Our application was unable to make this functional. Fail

SOC4.1: Task Sharing Test 3

SOC4.2: This test will ensure that only tasks a user has chosen to share are visible to their friends/social group members.

SOC4.3: This test will consist of user 1 creating a task without sharing it, and user 2 not being able to see the task.

SOC4.4: Input will be a user 1 creating a task and not sharing it with anybody.

SOC4.5: Output will be the task information NOT appearing to user 2, even if user 2 is friends with user 1 or in a social group with user 1.

SOC4.6: Normal

SOC4.7: Blackbox

SOC4.8: Functional

SOC4.9: Integration

Result: Our application was unable to make this functional. Fail

RL1.1: Release Test 1

RL1.2: This test will ensure that our app is able to be loaded onto a mobile device.

RL1.3: This test will consist of loading the app onto a mobile device and making sure it launches properly.

RL1.4: Input will be all of the necessary app files going onto the device.

RL1.5: Output will be the app successfully launching on the device.

RL1.6: Normal

RL1.7: Whitebox

RL1.8: Functional

RL1.9: Integration

Result: Though emulated successfully, we never attempted to get this app on a real mobile device. Fail

P1.1: Pet Functionality

- P1.2: The purpose of this test is to make sure that the pet appears on the screen and that their actions mimic what the user schedules themselves to be doing.
- P1.3: Loading the initial home screen and making sure that the pet appears and is animated in a visually appealing way, making different tasks and making sure the animal mirrors appropriately.
- P1.4: Input will be creating tasks.
- P1.5: Output will be visually appealing animations and appropriate context reponses.
- P1.6: Normal
- P1.7: Whitebox
- P1.8: Functional
- P1.9: Unit
- Result: This functionality was far beyond the scope of our project. The virtual pet is visible, but there are no animations currently implemented. Fail**

Test Case Matrix

Test Identifier	Normal/ Abnormal	Blackbox/ Whitebox	Function al/ Performa nce	Unit/ Integrat ion	Result
CF1	Normal	Whitebox	Function al	Unit	Pass
CF2	Abnormal	Whitebox	Function al	Unit	N/A
CF3	Abnormal	Whitebox	Function al	Unit	Pass
CF4	Normal	Whitebox	Function al	Unit	N/A
DIS1	Abnormal	Whitebox	Performa nce	Integrat ion	Pass
SOC1	Normal	Blackbox	Function al	Integrat ion	Fail
SOC2	Normal	Blackbox	Function al	Integrat ion	Fail
SOC3	Normal	Blackbox	Function al	Integrat ion	Fail
SOC4	Normal	Blackbox	Function al	Integrat ion	Fail
RL1	Normal	Whitebox	Function al	Integrat ion	Fail
P1	Normal	Whitebox	Function al	Unit	Fail