

Team members:

- Cheng, Aaron
- Howell, Alexander
- Little, Andrew
- Nguyen, Trung

Test case 1: Account creation

Input:

Name: name of the user
DOB: birthdate of the user
Email: email of the user
Password: password of the user

Expected Output:

Parent account object with correct name, DOB, email, password (ExpectedParent)

Actual output:

ActualParent

Pass/Fail Criterion:

ExpectedParent == ActualParent

Test case 2: Parent account creates child account

Input:

PIN: PIN login for child
Name: Name of child
DOB: birthdate of child
Sex: Male, female, or other, as it relates to the child

Expected Output:

ExpectedChild with correct Name, Age, Sex, PIN

Actual Output:

ActualChild

Pass/Fail Criterion:

ExpectedChild == ActualChild

Test case 3: Parent-Child relation is created

Input:

Parent: aParent
Child: aChild

Expected Output:

Parent object with correct associated array of child objects.
Child object with correct associated array of parent objects.

Actual Output:

ActualParent

ActualChild

Pass/Fail Criterion:

ActualParent.child[].length >= 0

ActualChild.parent[].length >= 0 and <=2

ActualParent.isChild(ActualChild) == True

ActualChild.isParent(ActualParent) == True

Test case 4: Deletion of child account

Input:

Parent account associated with the child account

Child account need to be deleted

Expected Output:

Parent no more associates to the DeletedChild

Actual Output:

The parent account after calling delChild() method

Pass/Fail Criterion:

Parent.isChild(DeletedChild) == False

Test Case 5: Creating Goal Object

Input:

GoalName, Description

Expected Output:

ExpectedGoal with correct GoalName and Description

Actual Output:

ActualGoal

Pass/Fail Criteria:

ExpectedGoal == ActualGoal

Test Case 6: Creating Reward Object

Input:

AdditionalScreenTime

Expected Output:

ExpectedReward with correct AdditionalScreenTime

Actual output:

ActualReward

Pass/Fail Criterion:

ExpectedReward == ActualReward

Test case 7: Creating a task

Input:

NameOfTask: New assigned task to be completed by Child

StartTime: When new task should begin progress

EndTime: When new task is due

Goal: Objective of new task

Reward: ScreenTime or other, granted to child after completion of new task

Status: Not completed, or completed

Proofs: Form of proof of completion, if applicable

AutoApprove: On or Off, if task requires manual approval or not

Expected Output:

ExpectedTask with correct details

Actual Output:

ActualTask

Pass/Fail Criterion:

ExpectedTask == ActualTask

Test case 8: Parent account assigns a task to child account

The assignment of tasks is represented by TaskManager Class

Input:

Parent: who assign the task

Child: to whom the task is assigned

Task: task

Expected Output:

ExpectedTaskManager with correct associated parent account, child account, and task

Actual Output:

ActualTaskManager

Pass/Fail Criterion:

ExpectedTaskManager == ActualTaskManager

Test case 9: Setting screen time for child account

Input:

Time will be set for the screen time of a child account.

Expected Output:

Child account's screen time has been setup correctly (ExpectedScreenTime)

Actual Output:

Child account's screen time after calling setScreenTime() method

Pass/Fail Criterion:

Child.getScreenTime() == ExpectedScreenTime

Test case 10: Reducing/Adding the screen time of child account

Input:

Time will be reduced/added from the screen time of a child account.

Expected Output:

Child account's screen time has been reduced/added correctly (adjustedScreenTime)

Actual Output:

Child account's screen time after calling incrementTime()/decrementTime() method

Pass/Fail Criterion:

Child.getScreenTime() == adjustedScreenTime

Test case 11: A recurring task is set by Parent

Input:

Time: time interval for recurrence

Child: aChild whose task is being assigned

Parent: aParent who assigns the task

Task: the task will be set to be recurring

Expected Output:

A set of TaskManager objects represent the set of recurring tasks associated with aParent, aChild.

ExpectedTaskManager1, ExpectedTaskManager2,

Actual Output:

A set of recently created TaskManager objects.

ActualTaskManager1, ActualTaskManager2,...

Pass/Fail Criteria:

ExpectedTaskManager1, ExpectedTaskManager2, == ActualTaskManager1, ActualTaskManager2,...

Test Case 12: Parent approves the child's task completion

Input:

Parent

TaskManager

Expected output:

TaskManager.Task.Status: "Completed"

Actual output:

TaskManager

Pass/Fail criterion:

TaskManager.Task.Status == "Completed"

Test Case 13: Parent denies the child's task completion

Input:

Parent

TaskManager

Expected output:

TaskManager.Task.status: "Failed"

Actual output:

TaskManager

Pass/Fail criterion:

TaskManager.Task.Status == "Failed"

Test Case 14: Task not Started

Input: Task

Start Time > CurrentTime

Expected Output: status == "not started"

Actual Output: Task

Pass/Fail output : Task.getStatus == "not started"

Test Case 15: Task in process

Input:

Task
Start Time < CurrentTime

Expected Output:

status == "in process"

Actual Output:

Task

Pass/Fail output :

Task.getStatus == "in process"

Test Case 16: Parent grants reward of screen time to child for completing a task

Input:

Parent
Child
Task

Expected Output:

Child with adjusted screen time

Actual output:

Child

Pass/Fail Criterion:

Child.getScreenTime() == OldScreenTime + RewardedScreenTime

Test Case 17: the screen time will decrease when child uses phone

Input:

Child
Amount of time used

Expected Output:

Child with adjusted screen time

Actual output:

Child

Pass/Fail criteria:

child.getScreenTime == OldScreenTime - amountOfTimeUsed

Test Case 18: If the screen time equals 0, the child account cannot use the phone.

Input:

ScreenTime <= 0

Expected Output:

Child with status "Blocked"

Actual Output:

Child

Pass/Fail criteria:

child.IsBlocked() == true