Test Plan and Report

Product Name: Smart Shift

Team Name: None / Same as Product Name

Product Owner: Clarence

Team Members: Liza Sokolova, Annie Liu, Clarence Matthew Ortega, Man Ying Ho, Skyler Cao

Release Name: Version 1.0

Release Date: 6/6/2023

Revision Number: 1.0

Release Plan: 1.3

Revision Date: 6/4/2023

System Test Scenarios

Sprint 1:

A. User Story 1: As a student who needs help managing time, I want a pomodoro timer to keep me on track and give me break times.
B. User Story 2: As a student who sometimes wants to keep working through a break time, I want the option to customize my pomodoro timer or skip past break times to keep working on a task.

Scenario 1: Using Timer (Pass)

1. Pressing Start should immediately begin the Work Timer.

2. After Work Time elapses, the user should be able to press a Move On button to start their Break Timer.
3. After the Break Timer elapses, the user should have an option to continue and begin the Work Timer again.

Scenario 2: Using Time:, Pausing, Resetting, and skipping. (Pass)

1. Pressing the Pause button should pause the timer where it currently is at.
2. The Resume button should be available after pausing the timer.
3. Pressing the Resume button should continue the timer.
4. Pressing the reset button should pause the timer, and reset the Timer to its original set time, (Set to the Work Time if in the work cycle, set to the Break Time if in the break cycle.)
5. During the Break cycle the user should see a “Skip Break” button that allows them to continue directly back to the Work cycle, even if the timer is not complete. This button resets the timer to the correct work duration, and pauses the timer.

Scenario 3: Customizing Timer (Pass)

1. Pressing the “25 Minute Work Time” button should open up a menu that gives the user an input method to set the amount of time they want to work for.
2. The user should only be able to enter in valid inputs (numbers).
3. The user enters in 1 minute.
4. Pressing the cancel button returns the user to the start screen, and no changes are made.
5. Pressing the confirm button returns the user to the start screen, and the button should reflect the changes to the Work Time Duration the user made. Users should now see “1 Minute Work Time”.

6. Above tests should also apply to the Break Time button, except the user should pick a time different from the work time (user enters in 2 minutes instead.)
7. Upon starting the timer, the timer counts down from the set 1 minute during the work cycle, and the set 2 minutes during the break cycle. Resetting the timer resets the time to 1 minute during the work cycle, and 2 minutes during the break cycle.

Scenario 4: Timer Edge Cases (Fail?)

1. Users are limited to how large they can set a time (Maximum time of 9999 minutes.)
2. Users cannot set a negative time.
3. The timer should operate as intended regardless of set time.

The timer does not count down, and will not transition to the next cycle if the user sets a duration of 0.

<u>Sprint 2:</u>

A. User Story 3: As a student, I want to be able to create an account, and access my account via different devices.
B. User Story 6: As a student with multiple tasks to do, I want a way to keep track of all these tasks and add them to a list.
C. User Story 8: As a student I want to keep track of things that I've completed already. (Mark as Complete Function).

Delete account from firebase beforehand needed.

Scenario 1: Creating Account/Creating existing Account

1. start app;

   select 'create account'; type

<Mobile Number> = 1234567890

<Name> = test

<Email> = test@test.com

<Password> = 123456789

User should be logged into account.

2. Selecting ‘log into existing account’ brings the user back to the login screen.
3. Logging out and repeating the process should result in an ‘account already exists” error.

Scenario 2: Logging out of Account:

1. Start app;

   User should be automatically logged into the account they created.

   select ‘logout’;

   User should be brought to the login screen, signed out of their account.

Scenario 3: Logging into Account:

1. Start app;

   select ‘login’;

   User should see an error message informing them that their email is invalid;

   type <Email> = email that doesn’t exist;

   type <password> = password

   select ‘login’;

   User should see an error message informing them of an account not found

   type <Email> = test@test.com;

   type <password> = ;

   select ‘login’;

User should see an error message informing them of missing password;

type <password> = 321;

User should see an error message informing them of an incorrect password;

type <password> = 123456789;

User should be able to login to their account successfully;

Scenario 4: Creating Task Bite and testing the Mark as Urgent feature.

1. Navigate to ‘Task Bites’
2. Select + button;

   User should see options to add a task;

   type

   <Description> = test;

   <Date> = 2023-06-05;

   Mark as Urgent;

   Mark as Important;

   select Add;
3. User should see their created task appear, with description, date, and marked tags

Scenario 5: Editing Task Bite and testing the Mark as Urgent.

1. Using the created task in Scenario 4.
2. Select the edit function.
3. <Description> = edit_test;

   <Date> = 2023-06-06;

   Unmark as urgent;

   Unmark as important;

Confirm the edit;

Users should see the task be edited with the new parameters.

Scenario 6: Test illegal/invalid inputs (fail)

1. See Scenario 4, Scenario 5.
2. Attempt to create a task with empty description and empty date.
3. Attempt to create a task with a non empty description and empty date.
4. Attempt to create a task with an empty description and valid date.
5. Attempt to create a task with a negative date, and valid description.
6. Attempt to create a task with a very large date, and valid description.
7. Attempt to create a task with a valid description, and an incorrectly formatted date.
8. Run the same tests as above, but edit a pre-existing task rather than adding a new one.

Scenario 7: Test display for multiple tasks.

1. See Scenario 4.
2. Create multiple tasks.

   Users should be able to see the tasks displayed in a list.

Sprint 3:

A. User Story 5.0: As someone with specific needs or preferences, I want to be able to customize my personal settings for notifications

B. User Story 10.0: As a student who wants to finish all my tasks before or by a due date I want to be able to prioritize my tasks by due date or some other variable

C. User Story 14.0: As a student who has a workload of varying difficulties and importance, I want a way to assign priority categories to each task in order to come out with an efficient way to tackle my work schedule.

Scenario 1: Customize Pomodoro timer settings (Pass)

1. Navigate to ‘Settings’
2. Under “Set Default Work Time”, click the display that says “_ minutes”
3. User should see a modal that allows the user to change the work time
4. Change the setting to “5” and click “Change”
5. The display under “Set Default Work Time” should say “5 Minutes”
6. If the user navigates back to the “Flow Timer,” they should see “5 Minute Work Time”
7. Navigate back to ‘Settings’ and repeat steps 2-6 but with Break Time

Scenario 2: Customize Account Settings (Pass)

1. Navigate to ‘Settings’
2. Click ‘Account Settings’ on the bottom
3. User should see a blank circle that says “Edit Image,” their profile name, and the email they used to register
4. Click “Edit Image” and select an image from your gallery to use as a profile picture (may not store correctly if image size is too big)
5. After clicking “Choose” on the desired picture, the picture that was chosen should appear inside the circle in ‘Account Settings’
6. If the user clicks the pencil icon next to their name, it should bring up a modal that says “Change Username”

7. If the user types in “Testing,” and clicks ‘Change,” they should see their name change to “Testing” in the Account Settings page.

Scenario 3: Filtering tasks by a specific variable (Pass)

1. Navigate to “Task Bites”
2. Click the “+” button on the top right, which should bring up a screen to add a task
3. Type Description = <HW 1>
    a. Date = <2023-06-07>
    b. Urgent = <Unchecked>
    c. Important = <Unchecked>
4. The task should now pop up in the task list
5. Click on the rectangle that says “Select Option” above the task list, and there should be a dropdown of possible options to filter by like “Urgent” or “Complete”
6. Click on the box that says “Urgent,” and the task list should adjust to only include Urgent tasks. The task we added earlier, HW1, should not appear here.
7. Click on the box that says “Not Urgent” while “Urgent” is still selected, and no tasks should appear.
8. Uncheck “Urgent,” and the task HW1 should appear on the list.
9. Uncheck the “Not Urgent” filter, and all boxes should be unchecked. Click on the checkbox next to “In Progress,” HW1 should still be on the list.
10. If the user clicks on the HW1, they should be able to edit the task. The user should click the “Complete” checkbox and click “Update,” and it should disappear from the Task View until “In Progress” is no longer selected as a filter.

Scenario 4: Sorting task by ascending and descending based on due date (Pass)

1. Navigate to “Task Bites”
2. Click the “+” button on the top right, which should bring up a screen to add a task
3. Type Description = <Task1>
    a. Date = <2023-06-07>
    b. Urgent = <Unchecked>
    c. Important = <Unchecked>
4. Click “Add,” “Task1” should appear on the list
5. Click the “+” button on the top right, which should bring up a screen to add a task
6. Type Description = <Task2>
    a. Date = <2023-06-12>
    b. Urgent = <Unchecked>
    c. Important = <Unchecked>
7. Click “Add,” “Task 2” task should appear on the list
8. By default, the list should be in descending order, meaning that Task1 should be above Task2.
9. Click the calendar icon on the top right, which should change the list into ascending order, and now Task2 should be above Task1

Sprint 4:

A. User Story 4.0: As a forgetful student, I want to receive notifications before an upcoming deadline to remind me to finish an assignment and when a timer goes off.
B. User Story 9.0: As a goal oriented student, I want an easy way to set goals for myself and check progress over time, so that I do not lose focus on achieving the goals.

C. User Story 11.0: As a student who wants to keep track of what I learned, I want to be able to frequently record what I've learned (reflection - when break starts, write what you accomplished during the pomodoro period) so that I can review it later for studying

D. User Story 17.0: As a studious student, I want to work multiple periods of the pomodoro timer, so that I can do multiple slices of pomodoro periods

Scenario 1: Notification Valid

1. Check your phone settings to ensure that notifications are allowed for Expo Go
2. Navigate to 'Settings'
3. Enable timer notifications and confirm your choice
4. Navigate to 'Flow TImer'
5. Set an alarm and wait for the work timer to count down
6. User should see a notification when the work timer ends
7. Click 'Move On' and wait for the break timer to count down
8. User should see a notification when the break timer ends

Scenario 2: Tracking task goal progress

1. Create a task (specific inputs and names do not matter)
2. Mark the task as complete
3. Navigate to "Productivity Scope" and click "View Goals"
4. The first tab should say "1 tasks completed today" (assuming no other tasks were marked complete that day" and the pie chart should accurately reflect that
5. The daily goal should be displayed, and the user can click "Set Goal" to change their goal

6. If the user changes their goal, the pie chart should adjust accordingly. If the user changes their goal to 1 task completed and confirms the changes, the pie chart should fill up and say “Goal Completed”

Scenario 3: Creating a Reflection

1. Navigate to ‘Flow Timer’
2. Click ‘Start’
3. At end of Work Cycle, click ‘End Work’
4. User should see options to create a reflection;

   type

   <Title> = test;

   <Description> = Test description;

   select ‘Add Reflection’;

Scenario 4: Test Viewing Reflections display (agenda and individual reflections)

1. Click ‘Productivity Scope’
2. Click ‘View Reflections’
3. User should see agenda of created reflections appear, listed by date created in the calendar and named according to the ‘title’
4. Click a single reflection
5. User should see a reflection, with all of its details viewable, no editing allowed.

Scenario 5: Test invalid/illegal inputs (Pass - conditionally (known issue can cause Failures))

1. See Scenario 4, 5;
2. Attempt to create a reflection with empty title and description.
3. Attempt to create a reflection with an empty description.

4. Attempt to create a reflection with an empty title

Unit Tests

N/A - Could not create any automated Unit Tests, all testing was done manually with scenarios