**Testing**

**Test Objectives – Adding a task.**

|  |  |
| --- | --- |
| Test Script: Adding a task |  |
| New task being added | Date:  11/09/2023 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Name** | | New user logging in | |  |  |
| **Use Case Tested:** | | User adding a task (VISION DOCUMENT 4.1) | |  |  |
| **Test Description:** | | The user will add a task on the main screen following the prompts and detailed options given. | |  |  |
| **Pre-conditions** | | * App is successfully installed. * User is signed in either through google or anonymously | |  |  |
| **Post-conditions** | | * User clicks the “+” button. * User inputs Enter title of task and Description details. * User clicks the “Add Task Button” | |  |  |
| **Notes:** | | At least 1 task must be created before the user can view tasks by preference, such as tasks completed or not completed.   * If the anonymous user tries to sign out, a warning is given regarding session information getting lost. | |  |  |
| **Result**  **(Pass/Fail/Warning/Incomplete)** | | Pass | |  |  |
|  | **TEST STEP** | | **EXPECTED TEST RESULTS** | **P** | **F** |
| 1. | User Clicks Task Button | | Sign in page gets displayed (Figure 4) |  |  |
| 2. | User Enters details and clicks “Add Task” | | App welcome page displayed (Figure 1) |  |  |
| 3. | Task appears on Home screen and databases | | Task appears on Home screen and databases |  |  |
|  |  |  |  |  |  |

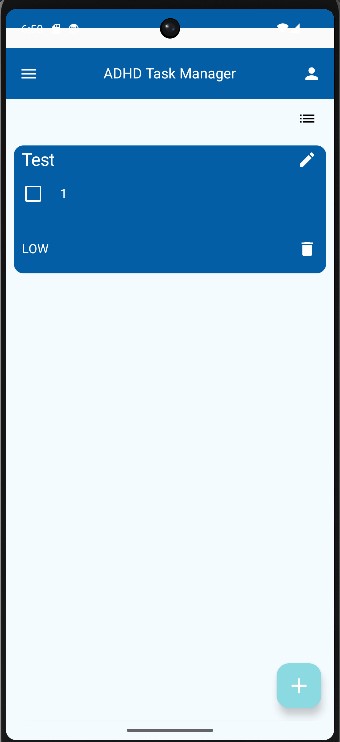
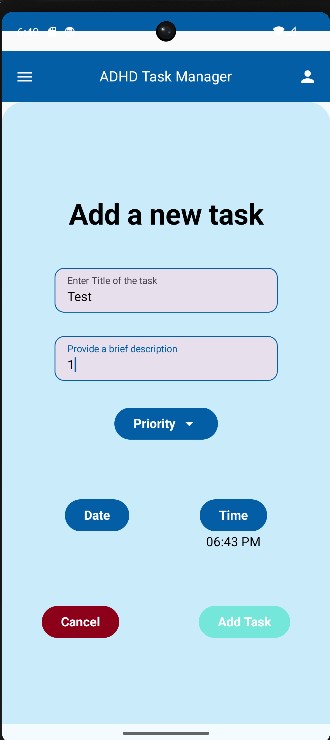


Figure .0 Task addition button clicked, user inputted details of task

Figure .0 Task addition button clicked, user inputted details of task

Figure 1.1 Task Added to home screen

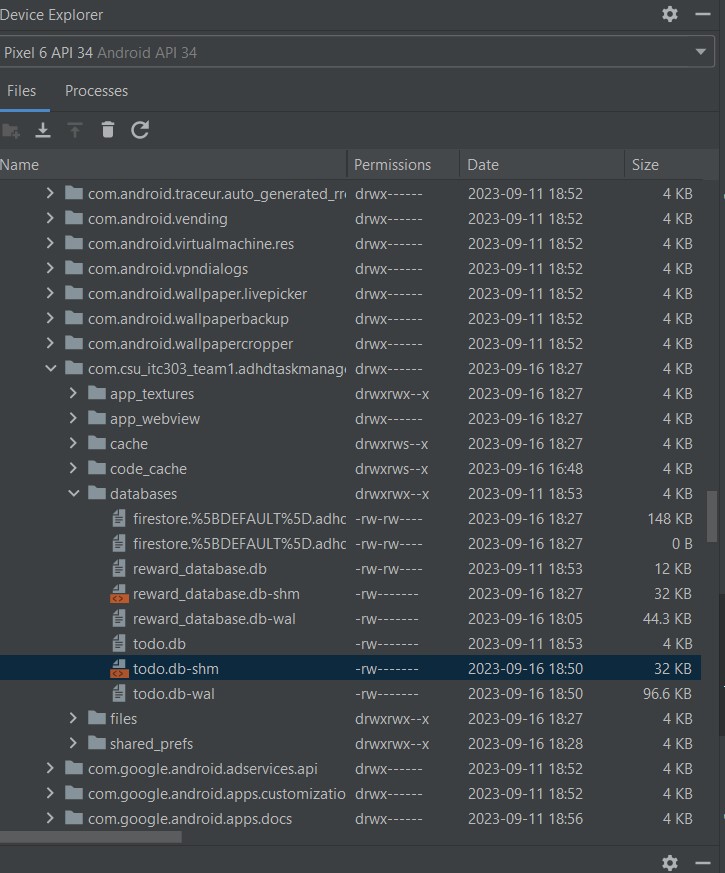


Figure .2 Task addition button clicked, todo.db database KB data increased and updated

|  |  |
| --- | --- |
| Test Script         Adding a task |  |
| Not filling in all inputs | Date:  11/09/2023 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Name** | | Not filling in all inputs | |  |  |
| **Use Case Tested:** | | User not completing the required inputs (VISION DOCUMENT 4.1) | |  |  |
| **Test Description:** | | The user will no fully complete the description work item, and then attempt to add the task. | |  |  |
| **Pre-conditions** | | * App is successfully installed * User is signed in either through google or anonymously | |  |  |
| **Post-conditions** | | * User clicks the “+” button * User inputs Enter title of task and leaves the description blank. * User clicks the “Add Task Button” | |  |  |
| **Notes:** | | At least 1 task must be created before the user can view tasks by preference, such as tasks completed or not completed.   * If the anonymous user tries to sign out, a warning is given regarding session information getting lost | |  |  |
| **Result**  **(Pass/Fail/Warning/Incomplete)** | |  | |  |  |
|  | **TEST STEP** | | **EXPECTED TEST RESULTS** | **P** | **F** |
| 1. | User Clicks Task Button | | The task add page appears |  |  |
| 2. | User Enters details and clicks “Add Task” | | The text inside the “Provide a brief description box” appears red and not allowing the user to click the add task button. |  |  |
|  |  |  |  |  |  |

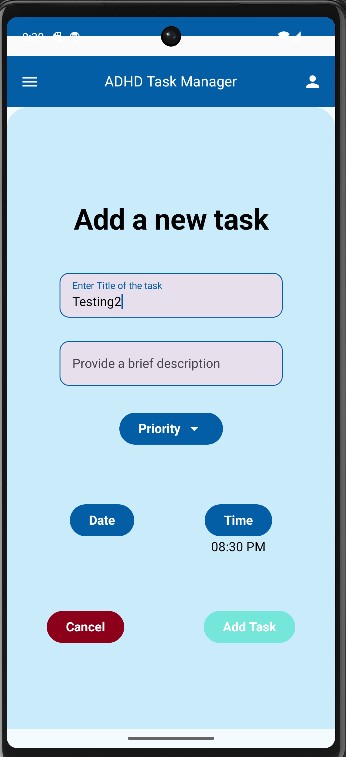
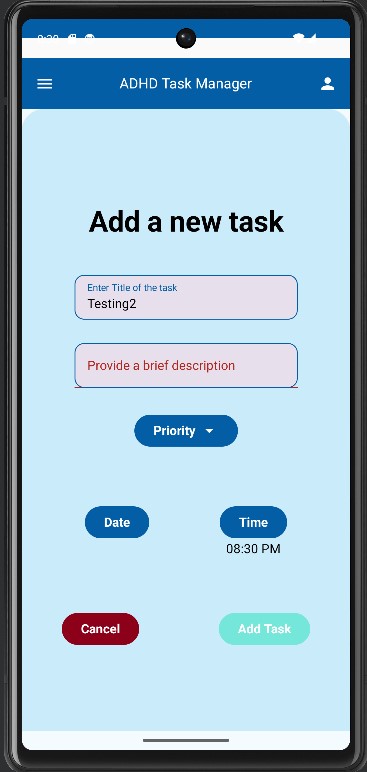


Figure 2.1 When the user clicks the add task button the Description box turns red

Figure 2.0 User adding a task and leaving the description blank

|  |  |
| --- | --- |
| Test Script: Deleting a task. |  |
| Deleting a task | Date:  11/09/2023 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Name** | | Deleting a task | |  |  |
| **Use Case Tested:** | | User deleting a task that has previously been added (VISION DOCUMENT 4.1) | |  |  |
| **Test Description:** | | The user will attempt to delete a task once it has been created. | |  |  |
| **Pre-conditions** | | * App is successfully installed. * User is signed in either through google or anonymously. * A task has been created | |  |  |
| **Post-conditions** | | * User clicks the “+” button. * User input the details of the task and adds the task, it will appear on the home screen. * The user will then click the trash icon on the task card created on the home screen. | |  |  |
| **Notes:** | | At least 1 task must be created before the user can attempt to delete it. | |  |  |
| **Result**  **(Pass/Fail/Warning/Incomplete)** | |  | |  |  |
|  | **TEST STEP** | | **EXPECTED TEST RESULTS** | **P** | **F** |
| 1. | User adds a task | | The task card is visible on the home screen. |  |  |
| 2. | User Enters presses the trash icon on said task. | | The task card is deleted and removed from the home screen. |  |  |
|  |  |  |  |  |  |

A screenshot of a phone

Description automatically generatedA white screen with black border

Description automatically generated

Figure 3.0 – Task card added to home screen, User presses trash can icon in bottom right of task card

Figure 3.1 Once pressed the Users created task card is erased from the users history and home screen

|  |  |
| --- | --- |
| Test Script: Home screen task card order. |  |
| Home screen task card order | Date:  11/09/2023 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Name** | | Home screen task card order | |  |  |
| **Use Case Tested:** | | The order of the task card arrangement (VISION DOCUMENT 4.1) | |  |  |
| **Test Description:** | | The user will attempt to change the order of the task cards according to either priority, date, completed and non completed | |  |  |
| **Pre-conditions** | | * App is successfully installed. * User is signed in either through google or anonymously. * More than one task has been created | |  |  |
| **Post-conditions** | | * User clicks the “+” button. * User input the details of the task and adds the task, it will appear on the home screen. * The user will then repeat this process until at least one more task card is created. | |  |  |
| **Notes:** | | At least 2 tasks must be created before the user can attempt see visual changes. | |  |  |
| **Result**  **(Pass/Fail/Warning/Incomplete)** | | Pass | |  |  |
|  | **TEST STEP** | | **EXPECTED TEST RESULTS** | **P** | **F** |
| 1. | User adds a task | | The task card is visible on the home screen. |  |  |
| 2. | User repeats process for multiple tasks to appear. | | The task card is deleted and removed from the home screen. |  |  |
| 3. | User changes the order of the tasks by either clicking priority, date, completed or non-completed. | | The order of the task’s cards will change in correspondence with what menu order they have chosen. |  |  |
|  |  |  |  |  |  |

A screen shot of a cell phone

Description automatically generated

Figure 4.1 Similarly to Fig.4.0 these cards have been chosen to be in order by priority, as seen by low, medium, high.

A screenshot of a cell phone

Description automatically generatedA cell phone with a blue box

Description automatically generatedA screen shot of a cell phone

Description automatically generated

Figure 4.3 By selecting the non-completed card title organiser the two uncompleted tasks are listed.

Figure 4.2 By selecting the completed organising card title the only completed task card is shown.

Figure 4.0- By selecting the by date card organiser the cards have arranged themselves in order of the date to be started.

|  |  |
| --- | --- |
| Test Script: Task card consignments |  |
| Task card consignments | Date:  23/09/2023 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Name** | | Task card consignments | |  |  |
| **Use Case Tested:** | | To evaluate task card operation of timer and date (VISION DOCUMENT 4.1) | |  |  |
| **Test Description:** | | The user will add a date and time to the task and ensure description is given within task card on home screen | |  |  |
| **Pre-conditions** | | * App is successfully installed. * User is signed in either through google or anonymously | |  |  |
| **Post-conditions** | | * User clicks the “+” button. * User input the details of the task and adds the task, including adding a date and time it will appear on the home screen. | |  |  |
| **Notes:** | |  | |  |  |
| **Result**  **(Pass/Fail/Warning/Incomplete)** | | Pass | |  |  |
|  | **TEST STEP** | | **EXPECTED TEST RESULTS** | **P** | **F** |
| 1. | User clicks the “+” button on the home screen | | The add task consignment details pop up |  |  |
| 2. | User adds conditions of the task for consignment details | | The user is able to set a date and time with their task |  |  |
| 3. | User Adds task | | The task card appears on the home screen with the consignment details displayed |  |  |
|  |  |  |  |  |  |

A screenshot of a phone

Description automatically generatedA screenshot of a phone

Description automatically generated

Figure 5.1 User has clicked the time selection for the task.

Figure 5.0 – User has clicked the plus button and is given consignment details.

A screenshot of a calendar

Description automatically generatedA screenshot of a phone

Description automatically generated

Figure 5.3 User has clicked the add task button and can see the date and time stamp for the task on the task card.

Figure 5.2 User selects the date selection for the task.

**Test Objectives – Rewards: Completing Tasks**

|  |  |
| --- | --- |
| Test Script: Rewards: Completing Task |  |
| Task being submitted as completed | Date:  11/09/2023 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Name** | | Task being submitted as completed | |  |  |
| **Use Case Tested:** | | User task being submitted with following subsequence’s of updating data. (VISION DOCUMENT 4.1) | |  |  |
| **Test Description:** | | The user once completing a task will have updated local rewards\_database, firebase database, completed task and leaderboard section of application. | |  |  |
| **Pre-conditions** | | * App is successfully installed. * User is signed in either through google or anonymously | |  |  |
| **Post-conditions** | | * User clicks the “+” button. * User add a task using the option provided. * User then clicks the checked box on the task card to signify task completion. | |  |  |
| **Notes:** | | At least 1 task must be created before the user can view tasks by preference, such as tasks completed or not completed. | |  |  |
| **Result**  **(Pass/Fail/Warning/Incomplete)** | | Pass | |  |  |
|  | **TEST STEP** | | **EXPECTED TEST RESULTS** | **P** | **F** |
| 1. | User Clicks Task Button and creates task | | Task is created and appears on the home screen |  |  |
| 2. | User clicks checkbox to signify task completion | | The checkbox becomes ticked |  |  |
| 3. | Local rewards\_database is updated | | Rewards\_database shows updated task completion/increases in size |  |  |
| 4. | Firebase Database ‘points’ is updated by 3 points | | In firebase, users’ points should increase by 3 |  |  |
| 5. | Completed task is updated in completed app page. | | Completed task is shown in completed tasks page on app |  |  |
| 6. | Leaderboard updated with user task completion | | The leaderboard increases by three points for user |  |  |
|  |  |  |  |  |  |

A black cell phone with a white screen

Description automatically generatedA screen shot of a cell phone

Description automatically generated

Figure 6.1 Completed tasks page showing no completed task yet.

Figure 6.0 – Taks uncompleted by user

A screenshot of a computer

Description automatically generated



A screenshot of a phone

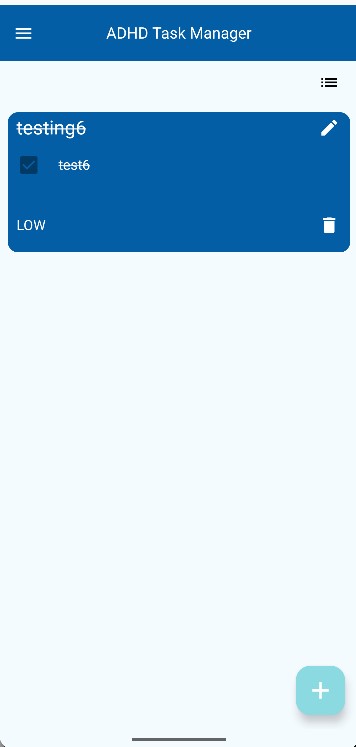
Description automatically generated

Figure 6.4 – After checkbox complete task shows in completed section of app

Figure 6.3 – User has successfully completed task and clicked checkbox

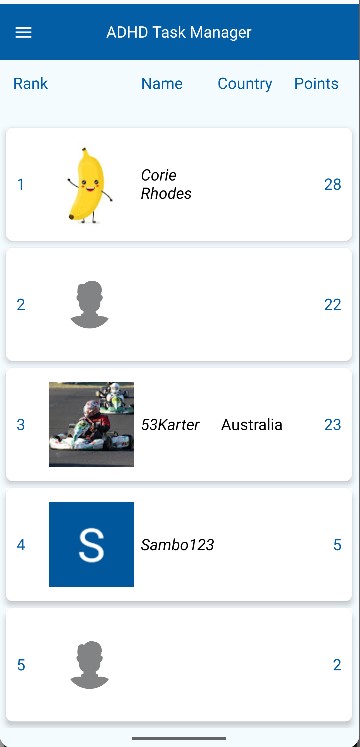


Figure 6.5 – Leaderboard for user increases by three points (Ssambo123

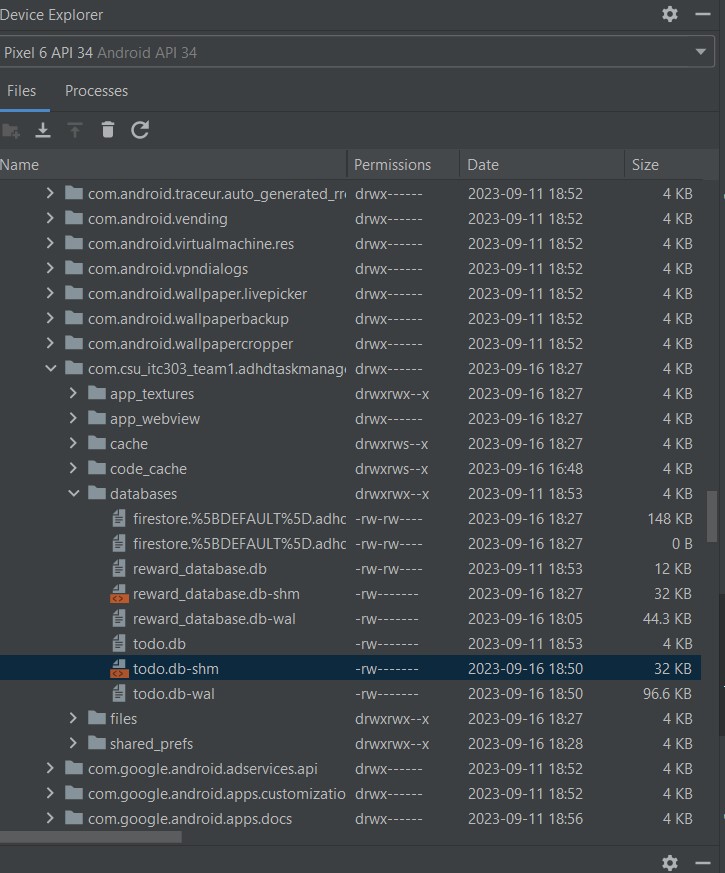


Figure 6.6 – User rewards\_database file increases in size and establishes task completed reward.

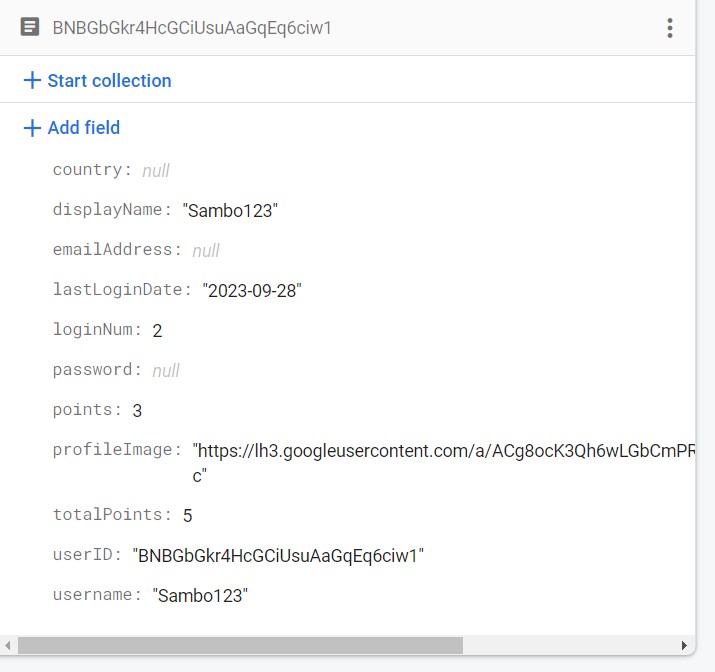


Figure 6.7 – Firebase storage points section increases by 3 points.

|  |  |
| --- | --- |
| Test Script: Task is unable to be completed multiple times. |  |
| Task is unable to be completed multiple times | Date:  25/09/2023 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Name** | | Task is unable to be completed multiple times | |  |  |
| **Use Case Tested:** | | To ensure user cannot spam tasks, they must only be able to be checked once. (VISION DOCUMENT 4.1) | |  |  |
| **Test Description:** | | The user once completing a task will be unable to unselect and select a task again. | |  |  |
| **Pre-conditions** | | * App is successfully installed. * User is signed in either through google or anonymously | |  |  |
| **Post-conditions** | | * User clicks the “+” button. * User add a task using the option provided. * User then clicks the checked box on the task card to signify task completion. * User then attempts to click the checkbox to change its state. | |  |  |
| **Notes:** | | At least 1 task must be created before the user can view tasks by preference, such as tasks completed or not completed. | |  |  |
| **Result**  **(Pass/Fail/Warning/Incomplete)** | | Pass | |  |  |
|  | **TEST STEP** | | **EXPECTED TEST RESULTS** | **P** | **F** |
| 1. | User Clicks Task Button and creates task | | Task is created and appears on the home screen |  |  |
| 2. | User clicks checkbox to signify task completion | | The checkbox becomes ticked |  |  |
| 3. | User attempts to uncheck task | | Checkbox stays checked and is unable to be edited. |  |  |
|  |  |  |  |  |  |

A screenshot of a phone

Description automatically generatedA screen shot of a cell phone

Description automatically generated

Figure 7.0 – Taks uncompleted by user

Figure 7.1 – User has successfully completed task and unable to deselect or complete same task again.

|  |  |
| --- | --- |
| Test Script: User database information returns upon logging out. |  |
| User database information returns upon logging out | Date:  25/09/2023 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Name** | | User database information returns upon logging out | |  |  |
| **Use Case Tested:** | | Upon a user logging out, when logging back in their user data should still be saved and visible. (VISION DOCUMENT 4.1) | |  |  |
| **Test Description:** | | The user once completing a task will be unable to unselect and select a task again. | |  |  |
| **Pre-conditions** | | * App is successfully installed. * User is signed in either through google or anonymously | |  |  |
| **Post-conditions** | | * User clicks the sign out button * User logs back in * User then clicks the checked box on the task card to signify task completion. | |  |  |
| **Notes:** | | At least 1 task must be created before the user can view tasks by preference, such as tasks completed or not completed. | |  |  |
| **Result**  **(Pass/Fail/Warning/Incomplete)** | | Pass | |  |  |
|  | **TEST STEP** | | **EXPECTED TEST RESULTS** | **P** | **F** |
| 1. | User navigates to sign once signed in navigates to the sign out page of the app. | | Once clicked the app will display the Sign in screen. |  |  |
| 2. | User clicks google sign in button | | The user should be prompted to sign in with google and taken to home screen. |  |  |
| 3. | User checks awards and tasks | | User is able to see completed task to ensure saved data. |  |  |
|  |  |  |  |  |  |

A screenshot of a phone

Description automatically generatedA screenshot of a phone

Description automatically generated

Figure 8.1 Once signed back in from the sign in page the user can see previously completed tasks saved by the databases

Figure 8.0 – User clicks sign out button once already being signed in

**Test Objectives – Usability**

|  |  |
| --- | --- |
| Test Script: Usability |  |
| Pomodoro Timer | Date:  25/09/2023 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Name** | | Pomodoro Timer | |  |  |
| **Use Case Tested:** | | The pomodoro timer functionality. VISION DOCUMENT 4.1) | |  |  |
| **Test Description:** | | The user will test to ensure the timer counts down from the amount chosen by the user which alternates between break and work timer. | |  |  |
| **Pre-conditions** | | * App is successfully installed. * User is signed in either through google or anonymously | |  |  |
| **Post-conditions** | | * User inputs details in settings for timer * User navigates and plays timer. * User tests stop and pause functionality | |  |  |
| **Notes:** | | Timer does not operate under a broadcast functionality system. | |  |  |
| **Result**  **(Pass/Fail/Warning/Incomplete)** | | Pass | |  |  |
|  | **TEST STEP** | | **EXPECTED TEST RESULTS** | **P** | **F** |
| 1. | User navigates to settings and inputs 1 minutes into work timer and 2 minutes into break timer. Then clicks the blue check mark to save data. | | The data is saved, and user is free to move to the pomodoro timer page. |  |  |
| 2. | User clicks start button on timer | | The timer should start counting down from 1 minute |  |  |
| 3. | User awaits completion of work timer to observe initiation of break timer. | | Once the work timer ends the break timer should start counting down from 2 minutes with a green colour. |  |  |
| 4. | User clicks pause button mid timer | | The timer will pause until the user clicks the play button again. |  |  |
| 5. | User clicks stop button | | The timer will reset back to the beginning of the work timer countdown of 1 minute and awaits user to clicks start button again. |  |  |
| 6. | Do not disturb notification pop-up | | Upon clicking start button, the do not disturb pop up setting should be displayed to the user for the timer. |  |  |
|  | A screenshot of a phone  Description automatically generated  Figure 9.0 – User input 1 minute and 2 minutes into Pomodoro Timer adjustment section of settings screen, followed by the click of the tick button. |  | A screen shot of a phone  Description automatically generated  Figure 9.1 – User navigates to the timer screen and clicks the play button. |  |  |

A screenshot of a phone

Description automatically generatedA screenshot of a phone

Description automatically generated

Figure 9.3 – User clicks the pause button on the timer.

Figure 9.2 – After the work timer ends the break timer begins with green colour.

A screenshot of a phone

Description automatically generatedA screen shot of a clock

Description automatically generated

Figure 9.5 Upon clicking the start button the Do not disturb feature occurs.

Figure 9.4 User clicks the stop button on the timer and the timer resets.

|  |  |
| --- | --- |
| Test Script: Usability |  |
| Side Menu functionality | Date:  26/09/2023 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Name** | | Side Menu functionality | |  |  |
| **Use Case Tested:** | | Usability of side menu VISION DOCUMENT 4.1) | |  |  |
| **Test Description:** | | The user will navigate to each page to ensure correct usability of main navigation architecture. | |  |  |
| **Pre-conditions** | | * App is successfully installed. * User is signed in either through google or anonymously | |  |  |
| **Post-conditions** | | * User opens app on home page. * User one by one navigates to each screen on the side menu to ensure proper usability | |  |  |
| **Notes:** | | At least 1 task must be created before the user can view tasks by preference, such as tasks completed or not completed. | |  |  |
| **Result**  **(Pass/Fail/Warning/Incomplete)** | | Pass | |  |  |
|  | **TEST STEP** | | **EXPECTED TEST RESULTS** | **P** | **F** |
| 1. | User Opens app on home screen | | Upon opening of app, user should be able to see the home screen of completed tasks. |  |  |
| 2. | Upon clicking the top left menu icon or swiping right the user selects the Complete tasks screen | | The use is taken to the completed tasks screen |  |  |
| 3. | Upon clicking the top left menu icon or swiping right the user selects the rewards screen | | The use is taken to the rewards screen |  |  |
| 4. | Upon clicking the top left menu icon or swiping right the user selects the leaderboard screen | | The use is taken to the leaderboard screen |  |  |
| 5. | Upon clicking the top left menu icon or swiping right the user selects the pomodorotimer screen | | The use is taken to the pomodorotimer screen |  |  |
| 6. | Upon clicking the top left menu icon or swiping right the user selects the settings screen | | The use is taken to the settings screen |  |  |
| 7. | Upon clicking the top left menu icon or swiping right the user selects the help page screen | | The use is taken to the help page screen |  |  |
| 8. | Upon clicking the top left menu icon or swiping right the user selects the sign out screen. | | The use is taken to the sign out screen |  |  |

A screenshot of a phone

Description automatically generatedA screenshot of a phone

Description automatically generated

Figure 10.1 Completed task screen.

Figure 10.0 – App side menu for user navigation

A screenshot of a cell phone

Description automatically generatedA screenshot of a phone

Description automatically generated

Figure 10.2 User navigation to Completed Tasks screen.

Figure 10.3 – User navigation to Rewards screen.

A screenshot of a phone

Description automatically generatedA screenshot of a phone

Description automatically generated

Figure 10.4 User navigation to Leaderboard screen

Figure 10.5 User navigation to pomodoro timer screen

A screen shot of a white sheet

Description automatically generatedA screenshot of a computer

Description automatically generated

Figure 10.7 User navigation to help screen.

Figure 10.6 User navigation to Settings screen

|  |  |
| --- | --- |
| Test Script         User login |  |
| <Test Name>New user logging in | Date:  07/09/2023 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Name** | | New user logging in | |  |  |
| **Use Case Tested:** | | New user logging anonymously (VISION DOCUMENT 4.1) | |  |  |
| **Test Description:** | | This test is to see if a new user is granted access to the app. | |  |  |
| **Pre-conditions** | | * App is successfully installed | |  |  |
| **Post-conditions** | | * Welcome message (figure 1) * User can tap “+” symbol to add/create a task.  (figure 2) | |  |  |
| **Notes:** | | At least 1 task must be created before the user can view tasks by preference, such as tasks completed or not completed.   * Top right corner of the screen in figure 2 * If the anonymous user tries to sign out, a warning is given regarding session information getting lost (Figure 3) | |  |  |
| **Result**  **(Pass/Fail/Warning/Incomplete)** | | Pass | |  |  |
|  | **TEST STEP** | | **EXPECTED TEST RESULTS** | **P** | **F** |
| 1. | User runs app | | Sign in page gets displayed (Figure 4) |  |  |
| 2. | Sign in anonymously | | App welcome page displayed (Figure 1) |  |  |
|  |  |  |  |  |  |

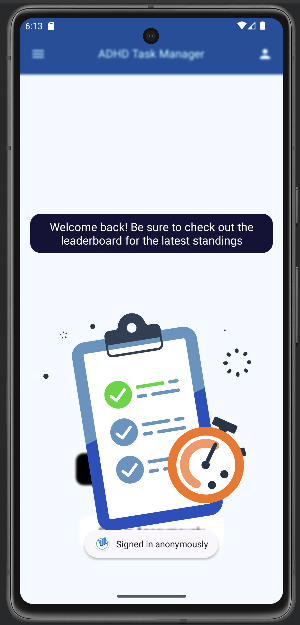
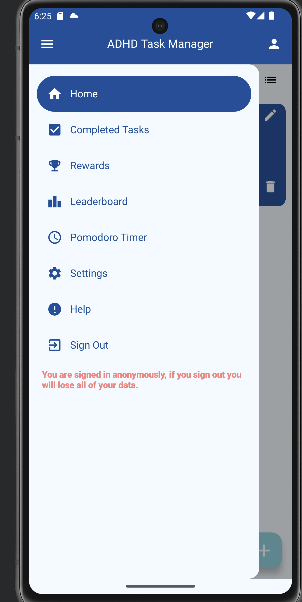


Figure 11.1

Figure 11.0

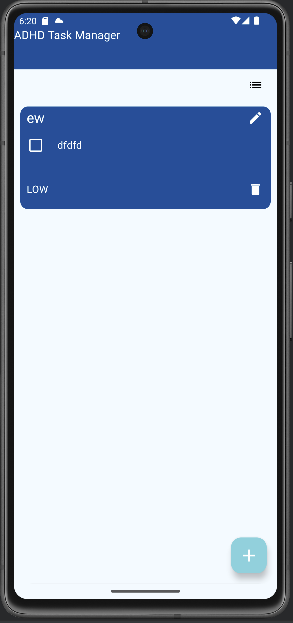


Figure 11.2



Figure 11.3

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Script         User login** | |  | | | |
| **<Test Name>New user logging in** | | Date:  07/09/2023 | | | |
| **Test Name** | | User login with Google account | | | |
| **Use Case Tested:** | | Signing in with incorrect Google account | | | |
| **Test Description:** | | User denied access if incorrect email provided. | | | |
| **Pre-conditions** | | * App has to be successfully installed. * User needs a valid Google account. * The Google account needs to be added in the settings of the device. (Figure 1) | | | |
| **Post-conditions** | | If the email is incorrect, an error message gets displayed. (figure 1) | | | |
| **Notes:** | |  | | | |
| **Result (Pass/Fail/Warning/Incomplete)** | | Pass | | | |
|  | TESTSTEP | | **EXPECTED TEST RESULTS** | **P** | **F** |
| 1. | Run the app | | Sign in page appears. |  |  |
| 2. | User enters incorrect email | | Error message gets displayed |  |  |
|  |  | |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Data Table | | | | | |
|  | **1** | **2** | **3** | **4** | **5** |
| Email | itc@gmail.com |  |  |  |  |
| [Data field 2] | [data set 1 input value for field 2] |  |  |  |  |
| [Data field 3] | [data set 1 input value for field 3] |  |  |  |  |

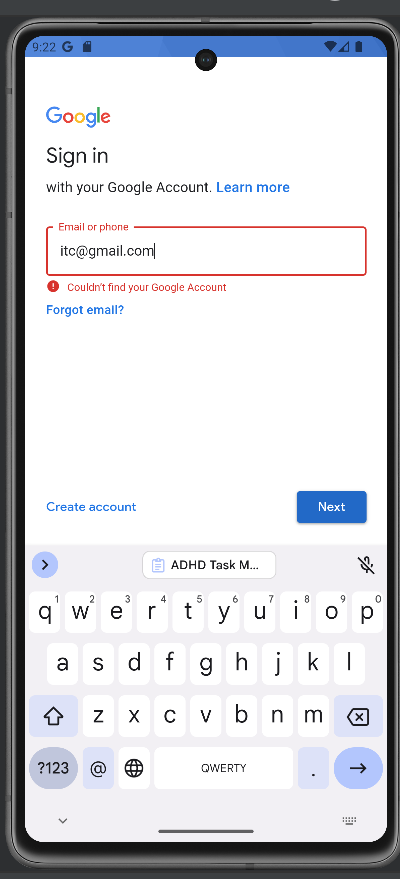


Figure 12.0

|  |  |
| --- | --- |
| Test Script: User login |  |
| New user logging in | Date:  07/09/2023 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Name** | | New user logging in | |  |  |
| **Use Case Tested:** | | New user logging anonymously (VISION DOCUMENT 4.1) | |  |  |
| **Test Description:** | | This test is to see if a new user is granted access to the app. | |  |  |
| **Pre-conditions** | | * App is successfully installed | |  |  |
| **Post-conditions** | | * Welcome message (figure 1) * User can tap “+” symbol to add/create a task.  (figure 2) | |  |  |
| **Notes:** | | At least 1 task must be created before the user can view tasks by preference, such as tasks completed or not completed.   * Top right corner of the screen in figure 2 * If the anonymous user tries to sign out, a warning is given regarding session information getting lost (Figure 3) | |  |  |
| **Result**  **(Pass/Fail/Warning/Incomplete)** | | Pass | |  |  |
|  | **TEST STEP** | | **EXPECTED TEST RESULTS** | **P** | **F** |
| 1. | User runs app | | Sign in page gets displayed (Figure 4) |  |  |
| 2. | Sign in anonymously | | App welcome page displayed (Figure 1) |  |  |
|  |  |  |  |  |  |

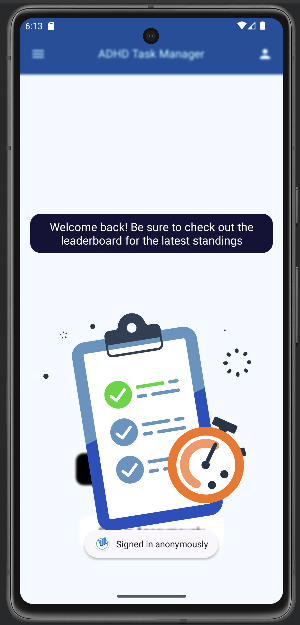
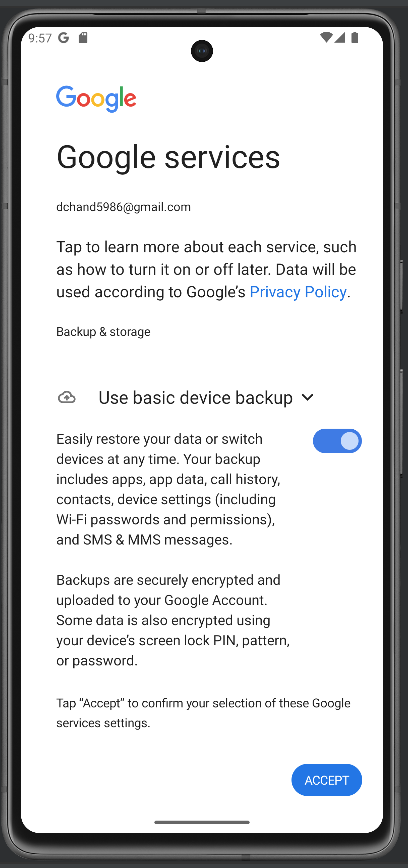


Figure 13.0

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Script: User login** | |  | | | |
| **New user logging in** | | Date:  07/09/2023 | | | |
| **Test Name** | | User login with Google account | | | |
| **Use Case Tested:** | | Signing in with correct Google account (VISION DOCUMENT 4.1) | | | |
| **Test Description:** | | User given access if correct email provided. | | | |
| **Pre-conditions** | | * App has to be successfully installed. * User needs a valid Google account. * The Google account needs to be added in the settings of the device. (Figure 1) | | | |
| **Post-conditions** | | * If the email is correct, T&C’s page appear which the user has to accept (figure 1). | | | |
| **Notes:** | | * Accepting the terms and conditions results in figure 2. * User login credentials gets stored and thus need not be entered for subsequent sign in. * Login authentication is done by Google. | | | |
| **Result (Pass/Fail/Warning/Incomplete)** | | Pass | | | |
|  | TESTSTEP | | **EXPECTED TEST RESULTS** | **P** | **F** |
| 1. | Run the app | | Sign in page appears. |  |  |
| 2. | User enters correct email | | T&C’s page gets displayed |  |  |
|  |  | |  |  |  |

A screenshot of a phone

Description automatically generated

Figure 14.0

Figure 14.1

|  |  |
| --- | --- |
| Test Script         Usability |  |
| Settings Page Dark Mode | Date:  27/09/2023 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Name** | | Dark Mode operation on the settings page | |  |  |
| **Use Case Tested:** | | Upon a user being logged in as anonymous or signed in with a google account, the dark mode operates on all screens and app remembers chosen state by user upon closing off app.(VISION DOCUMENT 4.1) | |  |  |
| **Test Description:** | | The user will stress test all sections of the app to ensure correct dark mode implementation | |  |  |
| **Pre-conditions** | | * App is successfully installed. * User is signed in either through google or anonymously | |  |  |
| **Post-conditions** | | * User signs in * User navigates to settings and checks dark mode. * User navigates too all screens. * User logs out and logs back in to ensure dark mode is still enabled | |  |  |
| **Notes:** | | - | |  |  |
| **Result**  **(Pass/Fail/Warning/Incomplete)** | | Pass | |  |  |
|  | **TEST STEP** | | **EXPECTED TEST RESULTS** | **P** | **F** |
| 1. | User Signs in | | Once signed in the user will be taken to the home screen. |  |  |
| 2. | User navigates to settings page and changes dark mode value | | Once the dark mode is activated the user will see the app change to a dark theme. |  |  |
| 3. | All pages of app, have changed theme | | From the user’s perspective all the apps colour themes should be changed to dark mode. |  |  |
| 4. | User Signs out and then signs back in to ensure the has remembered users dark theme state | | Once signed back in the user should still see the app in dark mode. |  |  |
| 5. | User turns of dark mode | | Once the user turns of dark mode the app should resume to its standard white and blue theme. |  |  |
|  |  |  |  |  |  |

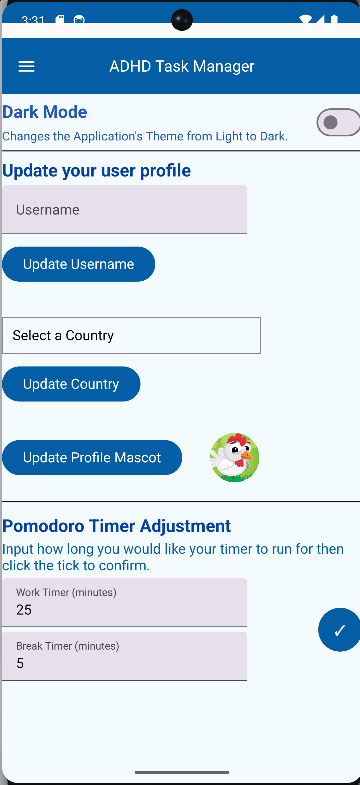
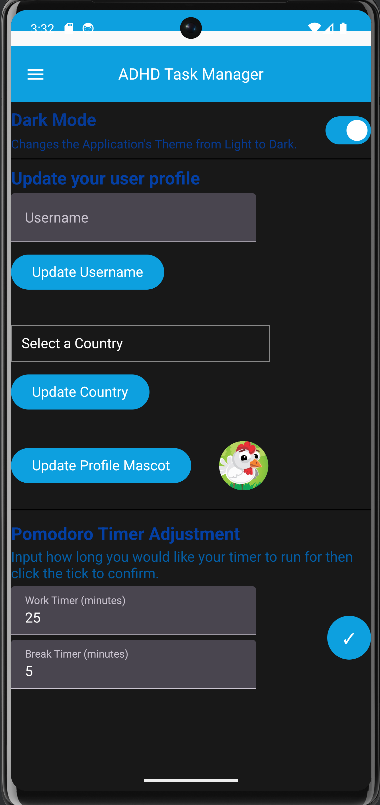


Figure 15.1 – User activates dark mode.

Figure 15.0 – User navigated to settings page with dark Mode unchecked.

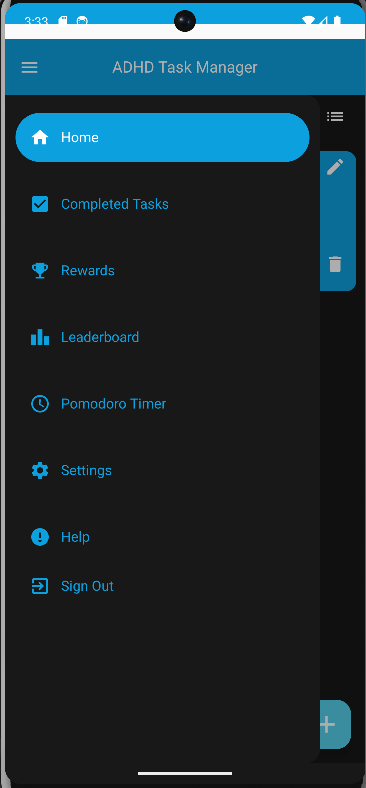


Figure 15.2 – All pages correctly displaying dark mode with above example of home screen with side menu.

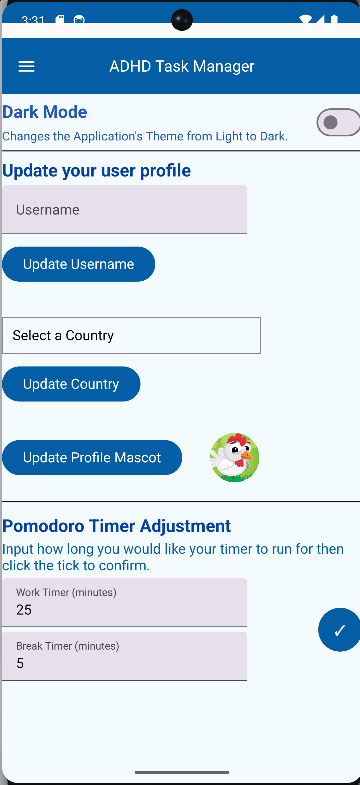


Figure 15.3 – User navigates back to settings screen and disables dark mode successfully.

|  |  |
| --- | --- |
| Test Script         Usability |  |
| Update Username | Date:  026/09/2023 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Name** | | Update Username | |  |  |
| **Use Case Tested:** | | User updates their username (VISION DOCUMENT 4.1) | |  |  |
| **Test Description:** | | This test is to see whether the user can successfully change their username which is displayed on the rewards screen and leaderboard | |  |  |
| **Pre-conditions** | | App is successfully installed | |  |  |
| **Post-conditions** | | The user will navigate to settings screen and type in a username and check whether the username was updated in the Firestore and therefore leaderboard and rewards screen. | |  |  |
| **Notes:** | | At least 1 task or sign in must be created before the user can view themselves on the leaderboard. | |  |  |
| **Result**  **(Pass/Fail/Warning/Incomplete)** | | Pass | |  |  |
|  | **TEST STEP** | | **EXPECTED TEST RESULTS** | **P** | **F** |
| 1. | User will sign in to the app | | Once signed in the user will be taken to the home screen |  |  |
| 2. | User will navigate to settings page and update username (ie. Sam123) | | Once on the settings screen the user will type in their username and click “update username” which will prompt a toast message saying that the username was updated successfully. |  |  |
| 3. | User will then check Firestore for successful update | | Firestore should update the users profile section “username” to what they have typed in (ie Sam123) |  |  |
| 4. | User will then check rewards screen | | The last update username should be displayed at the top. (ie Sam123) |  |  |
| 5. | User will then check leaderboard screen | | The last update username should be displayed on the user’s leaderboard card. (ie Sam123) |  |  |
|  |  |  |  |  |  |

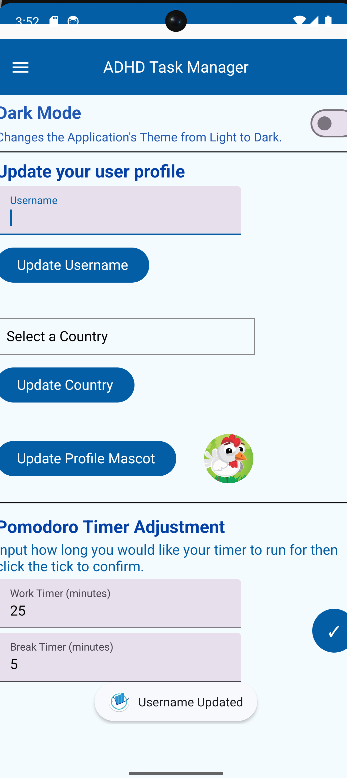


Figure 16.1 – After clicking “update username” button a toast message at the bottom is displayed to say it has been successful.

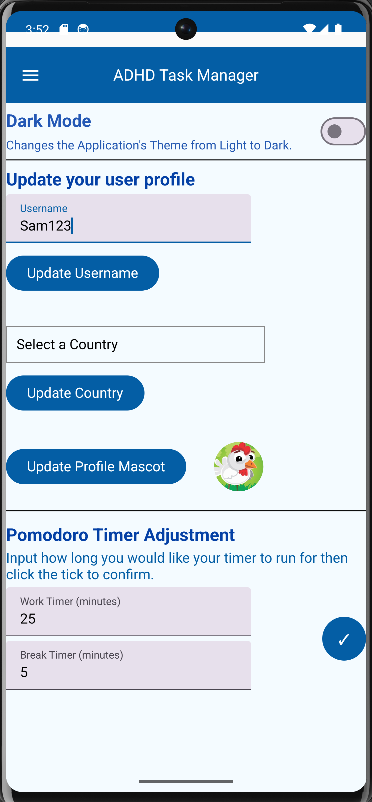


Figure 16.0 User has navigated to settings screen and typed in new username.

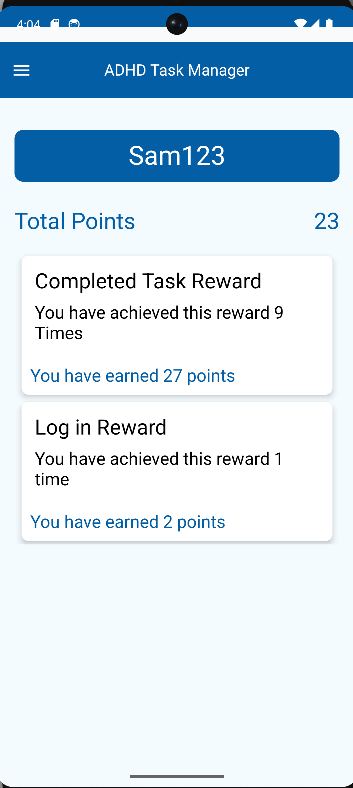
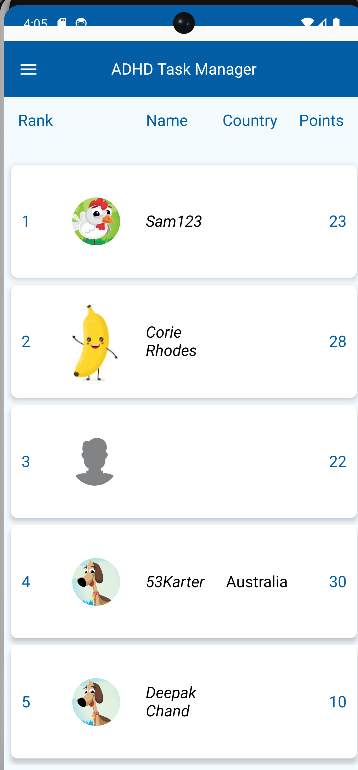


Figure 16.3 Leaderboard Screen correctly showing new username at the top.

Figure 16.2 – Rewards Screen correctly showing new username at the top.

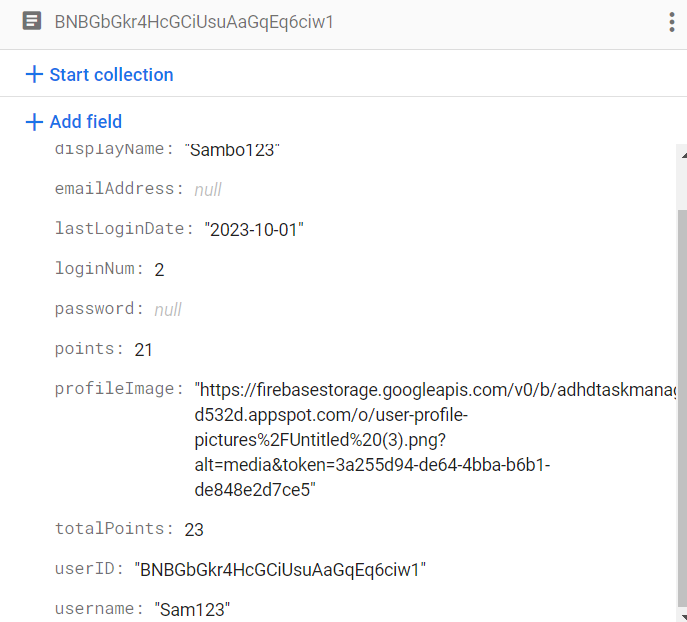


Figure 16.4 Firebase store correctly showing new username

|  |  |
| --- | --- |
| Test Script         Usability |  |
| Update Country | Date:  026/09/2023 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Name** | | Update Country | |  |  |
| **Use Case Tested:** | | User updates their country  (VISION DOCUMENT 4.1) | |  |  |
| **Test Description:** | | This test is to see whether the user can successfully change their country which is displayed on the and leaderboard | |  |  |
| **Pre-conditions** | | App is successfully installed | |  |  |
| **Post-conditions** | | The user will navigate to settings screen and type in a username and check whether the username was updated in the firestore and therefore leaderboard screen. | |  |  |
| **Notes:** | | At least 1 task or sign in must be created before the user can view themselves on the leaderboard. | |  |  |
| **Result**  **(Pass/Fail/Warning/Incomplete)** | | Pass | |  |  |
|  | **TEST STEP** | | **EXPECTED TEST RESULTS** | **P** | **F** |
| 1. | User will sign in to the app | | Once signed in the user will be taken to the home screen |  |  |
| 2. | User will navigate to settings page and update country (ie.France) | | Once on the settings screen the user will click update country and will be displayed a list of countries to select from, once chosen a toast message will display to say it was successful. |  |  |
| 3. | User will then check firestore for successful update | | Firestore should update the users profile section “country” to what they have typed in (ie France) |  |  |
| 4. | User will then check leaderboard screen | | The last update country should be displayed on the users card. (ie Sam123) |  |  |

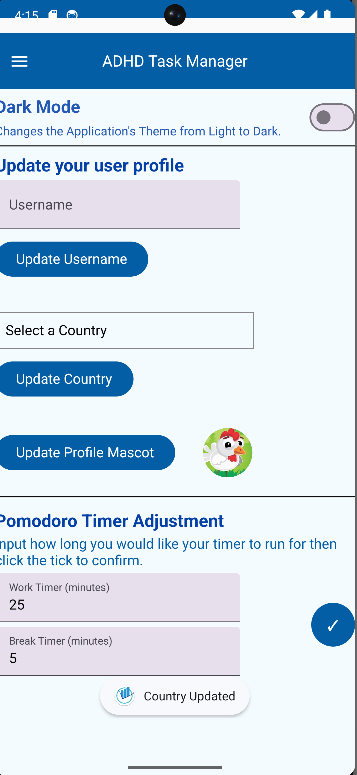


Figure 17.1 User has clicked a country and then clicked ‘update country’ to display country selected toast message.

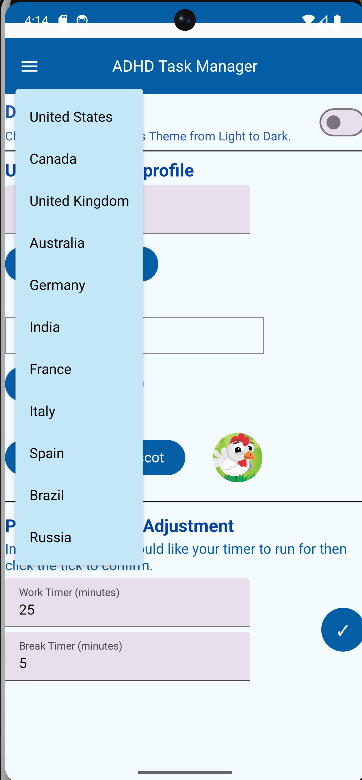


Figure 17.0 – User has navigated to settings screen and has clicked the ‘select a country box to display countries.



Figure 17.2 – Firestore has correctly been updated in the users country section

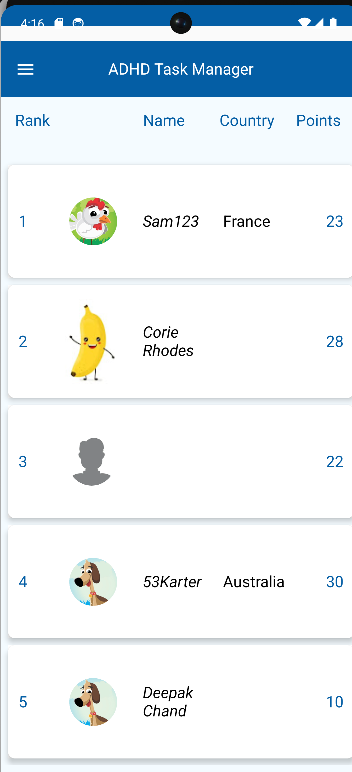


Figure 17.4 – User’s leaderboard information has been updated.

|  |  |
| --- | --- |
| Test Script         Usability |  |
| Update Mascot | Date:  026/09/2023 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Name** | | Update Mascot | |  |  |
| **Use Case Tested:** | | User updates their Mascot (VISION DOCUMENT 4.1) | |  |  |
| **Test Description:** | | This test is to see whether the user can successfully change their mascot which is displayed on the and leaderboard and settings screen. | |  |  |
| **Pre-conditions** | | App is successfully installed | |  |  |
| **Post-conditions** | | The user will navigate to settings screen and select a mascot and check whether the mascot was updated in the firestore and therefore leaderboard screen.and settings screen. | |  |  |
| **Notes:** | | At least 1 task or sign in must be created before the user can view themselves on the leaderboard. | |  |  |
| **Result**  **(Pass/Fail/Warning/Incomplete)** | | Pass | |  |  |
|  | **TEST STEP** | | **EXPECTED TEST RESULTS** | **P** | **F** |
| 1. | User will sign into the app | | Once signed in the user will be taken to the home screen |  |  |
| 2. | User will navigate to settings page and update mascot (ie.Fox) | | Once on the settings screen the user will click the ‘update mascot’ button which will display a list of images from the firestore database. |  |  |
| 3. | User will then check firestore for successful update, and leaderboard. | | The user once selected will be able to see the custom toast message of a successfully chosen mascot and updated on the leaderboard and firestore. |  |  |

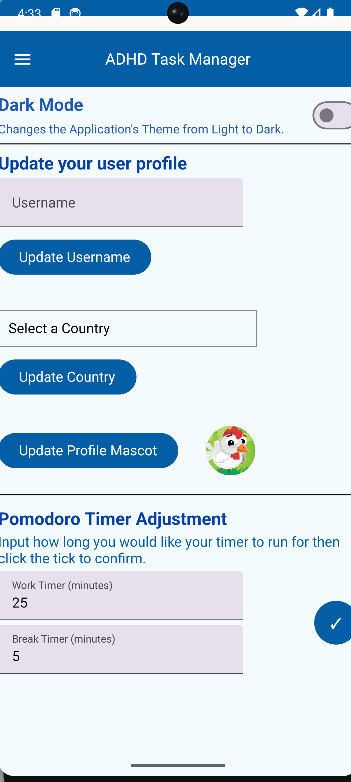


Figure 18.0 – User navigates to settings page.

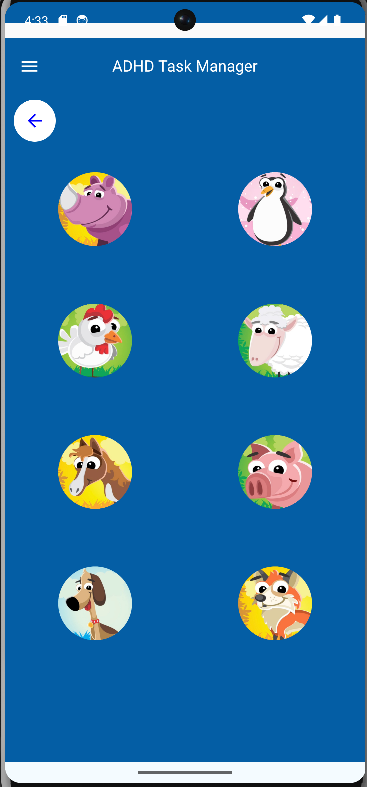


Figure – 18.1 – User is given options to choose a mascot after clicking the update Mascot button.

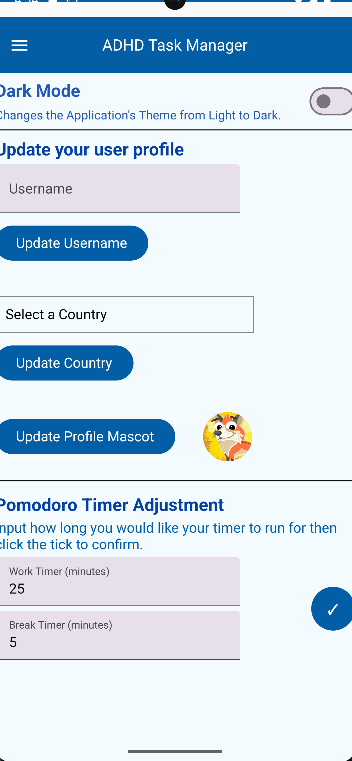
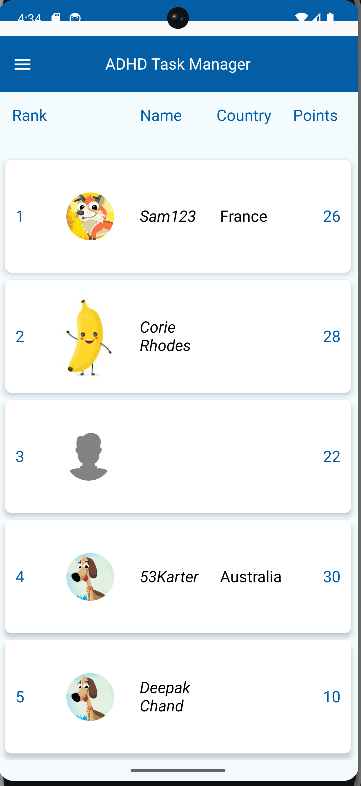


Figure 18.3 – User can see update on leaderboard screen of mascot

Figure 18.2 – User can see update of mascot in settings page

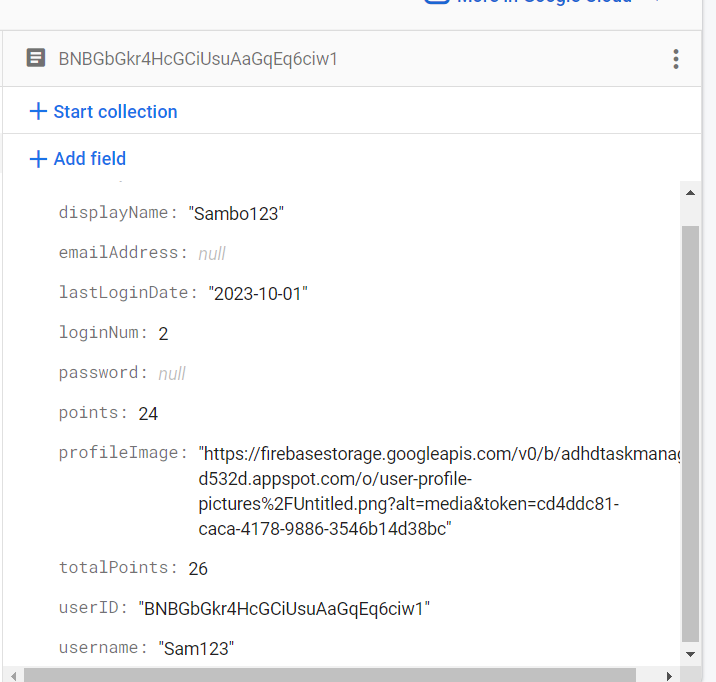


Figure 18.4 – Update of profileImage occurs after update of mascot image

|  |  |
| --- | --- |
| Test Script         Usability |  |
| Feel Good statement messages | Date:  026/09/2023 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Name** | | Feel Good statement messages | |  |  |
| **Use Case Tested:** | | User can visually see positive reinforcement messages  (VISION DOCUMENT 4.1) | |  |  |
| **Test Description:** | | This test is to see whether the user can successfully see the visual positive affirmations. | |  |  |
| **Pre-conditions** | | App is successfully installed | |  |  |
| **Post-conditions** | | The user for the first affirmation will open the app and once logged in should see the welcome back message. The user should then create a task and then complete it to see the congratulations message affirmation | |  |  |
| **Notes:** | | At least 1 task or sign in must be created before the user can undertake this test. | |  |  |
| **Result**  **(Pass/Fail/Warning/Incomplete)** | | Pass | |  |  |
|  | **TEST STEP** | | **EXPECTED TEST RESULTS** | **P** | **F** |
| 1. | User will sign in to the app | | Once signed in the user will see the welcome back toast message. |  |  |
| 2. | User will then create a task and click the checkbox to complete the task | | Once the checkbox is ticked the user will see the “congrats on completing a task” message and a confetti animation. |  |  |

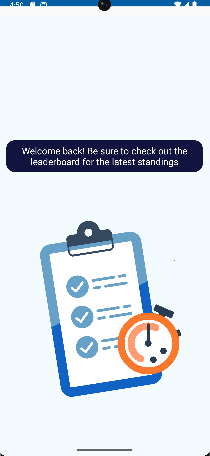
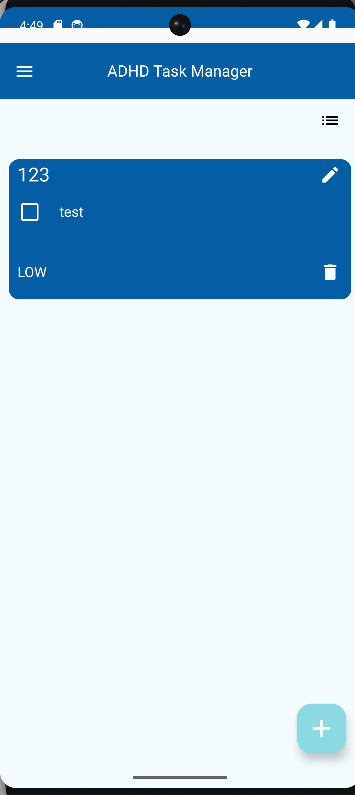


Figure 19.1 User creates a task and then clicks the checkbox

Figure 19.0 – When the user logs in they are welcomes with the welcome back message

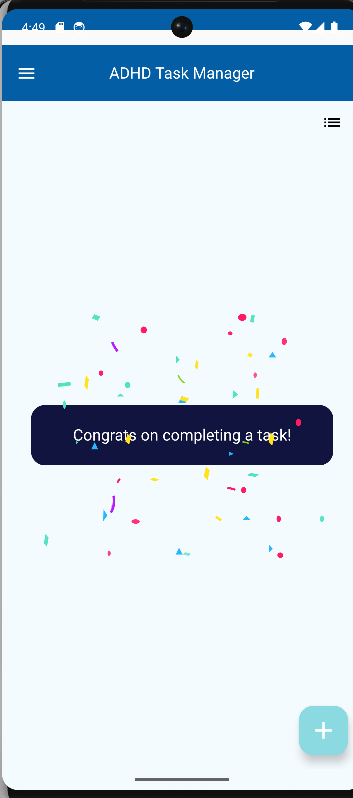


Figure 19.2 – Once the checkbox is completed the user is greeted with a congrats on completing a task toast message.