

Title:

Opening *Task Timer*

User Story:

As a *user* who wants to use the app. I want to be able to open it.

Conversation:

Obviously the app must first be installed from the Apple app store or Google play store. Then the app icon must be found by the user and pressed. The app opens to the homepage.

Acceptance Criteria:

1. When the app icon is pressed by the user, *Task Timer* opens to the homepage.

Title:

Creating a task in *Task Timer*

User Story:

As a *user*, I want to be able to create a new task by pressing an easily accessible button.

Conversation:

So once the user is on the homepage, there will be a large “plus” button in the bottom right corner of the homepage. Once the user presses this “plus” button, the create task window opens.

In this window, there is a text box which allows the user to enter their task. Then the user must press another button indicating the description of the task is complete, This button can be “next” which indicates the user is moving on to the next step.

After the “next” button is pressed by the user, it opens the “Date/Time” window which allows the user to enter a completion date and time to finish the task by. If no time is entered , by default it will select 23:59 of the date specified. The description and date is saved locally to the mobile device. Then the user can press the “finish” button to finish creating the task.

Acceptance Criteria:

2. When the create task button is pressed, the user can enter the name of the task and date to be completed.

Title:

Starting and pausing the timer in *Task Timer*

User Story:

As a user I want to be able to start and pause the current timing task.

Conversation:

Once the user has clicked on the task from the homescreen list, There is a “start” and “finish” button available for him to press. After clicking the start button, a timer will start to run and two new buttons will become available “Pause” and “Finish.” The timer will be stopped when either of these buttons are pressed. If the user decides to press the “Pause” button the timer stops and the “start” button reappears.

Acceptance Criteria:

1. When the user presses the ‘pause’ button, a ‘start’ button should appear for the user to start the task again.

Title:

Completing a task in *Task Timer*

User Story:

As a *user*, I want to be able to mark a finished task as complete.

Conversation:

On the homepage all the users tasks will appear in order from top to bottom, with more recent dates at the top. Simply swipe the task to the right off the screen to mark it as completed. The task will automatically be added to the completed task folder. In addition all tasks under the completed task move up in the list.

Acceptance Criteria:

2. When a task is swiped to the right on the homescreen, it disappears and all tasks under it move up by 1.

Title:

Checking stats for completed task

User Story:

As a *user*, I want to be able to see the time I spent completing certain tasks.

Conversation:

In the side task bar there will be a button which allows you to view completed tasks. Clicking on a completed task opens an interface which gives the user stats such as: total time spent working on the task(which is dependent on every time the user started and paused the timer), a pie chart which breaks the day up into intervals of 6 hour increments starting at midnight to 6am. This pie chart will show what percentage of the total time spent completing the task was spent in each interval, lastly it will show the date of completion.

Acceptance Criteria:

3. When a completed task is selected, stats for that completed task should be displayed to the user.

Title:

Creating an account for Task Timer

User Story:

As a *user*, I want to be able to create an account for Task Timer

Conversation

When opening the app for the first time, the login screen should appear. There should be a 'Create an Account' and a smaller "sign-in" button with the login screen(incase user already created account on different mobile device). If the Create Account button is pressed it will take the user to a page where you can create an account for Task Timer. The user can make an account using default information or Sign up using social media platforms like Facebook or Google.

Acceptance Criteria:

4. When the create account button is pressed, the user is asked to put in his information or Sign up with Facebook or Google account.

Title: assigning essential applications to be whitelisted in GET THERE

User Story:

As a user, I want the GET THERE program to allow me to set aside (whitelist) apps that do not affect my productivity, as they may obstruct me from accomplishing essential tasks if I am unable to use said apps.

Conversation:

The GET THERE application exists to boost productivity, and whitelisting only certain applications will force the user to stop themselves from being distracted by wasting time on non-essential apps or programs such as social media apps, games, etc.

As it stands, the user must finish this whitelisting the day before, to reinforce the notion that the applications which are not on this list will not be accessible the next day. By forcing the user to set this restriction themselves, the GET THERE program will also encourage the user to set their priorities and identify where their greatest enemies lie in being productive. In this way, the GET THERE program not only literally stops the user from being distracted, but also helps the user to compartmentalize essential apps from non-essential apps.

It would be a good idea to allow users to save this whitelist, as the essential apps that the user designate likely will not change from one work or school day to another. This will allow the user to carry over their whitelist from previous days with minimal hassle.

Users should have freedom in assigning their whitelists, as each person's idea of essential and non-essential applications will vary. However, basic services in the case of an emergency should be available always, even without needing the user to explicitly whitelist the app or service. These could include, but are not limited to, phone call, GPS function, flashlight, etc.

Acceptance Criteria:

1. User can assign apps to be whitelisted, and they will not be blocked from using these apps the next day.
2. *The user can save a whitelist configuration to be used in future days.*
3. The whitelist will always contain basic services in the case of an emergency, even without the user having to explicitly include said services.

Title: Emergency Mode

User Story:

As a user, I want the GET THERE program to limit my use of non-whitelisted apps. However, I also want to be able to use the phone without being restricted by GET THERE's program in the case of an emergency

Conversation:

Emergencies can range in urgency from life-threatening situation, to the user simply forgetting to whitelist a relatively important app the night before.

When the emergency mode is activated, the user will be able to use their phone normally, as if the GET THERE program was not installed in the first place.

The emergency mode will have a pre-set timer (5-10 minutes) to automatically deactivate after it has first been activated. This will limit users from abusing this feature to get around the restrictions that they set the night before, as it will be a pain to reactivate this mode every time they want to fool around.

Acceptance Criteria:

1. When the emergency mode is activated, the user will be able to use their phone normally without restriction.
2. The emergency mode will have a limit on the duration, and will switch off when this limit is reached as to prevent unnecessary user abuse