**Statement Of Work**

Activity Tracker Application is an application that provides option to track what you are doing in everyday life. Most of the cell phones or mobile applications itselves are tracking the time spend on them but to track realtime activities like time spent on a project, time spent while cooking, time spent while driving or showering, there are not much option to keep track on these activities. This Activity Tracker Application allows the client to track such activities that are being done on our everyday lives. A client can also track multiple activities at the same time like listening music while playing games, talking with a friend on the phone while cooking or trading stocks while watching a tv show.

The Application has 4 buttons that allows to add which activity to track on the main screen, a delete activity button that deletes the desired activity that is being tracked from the screen, a Start/Stop button that starts or stops the stopwatch for the corresponding activity, and a Background color changer button to change the background color of the activity that is being tracked to make tracking simpler.

**Patterns Used**

Design patterns that are being used by the application are Command Pattern, Factory Method Pattern, Decorator Pattern and Singleton Pattern.

**Factory Method Pattern:**

Firstly, the 4 buttons are created and added to the StopWatchApp Frame by the Button factory.

The StopWatches for the activities are also being created by the StopWatch Factory when the AddActivityButton is pressed.

**Singleton Pattern:**

To access these buttons we need to instantiate the app frame, but since we only need one application, the frame is created within inside with eager instantiation and the created frame is returned by the getInstance method.

**Command Pattern:**

After the creation of these buttons and being able to access them, the corresponding commands are assigned to them such as AddActivityCommand is assigned to AddActivityButton or DeleteActivityCommand assigned to DeleteActivityButton to act accordingly when they are pressed.

**Decorator Pattern:**

The created StopWatches are coming with a default background color, to change the background of them, there is a decorator called StopWatchDecorator that changes the background color to the desired color such as red, green or blue.