**Time Management App**

**Purpose:**

To help people manage their time more efficiently by displaying their daily activities graphically, and analyzing accurate records of time spent.

**First View – Daily View Controller**

Sub View – Foundation UI View for additional sub view overlays.

Pie Chart – Sub UI View

Changed default class from UI View to XY Pie Chart class.

Activity List – Sub Table View that dynamically fades in and fades out.

Tapping this table view’s cells will start and stop the event.

“>” Button slides to the Daily Activity Times view controller.

Segue is linked to the actual button, not the cells.

Add activity – Sub UI View that dynamically fades in and fades out.

Appends new activity to the list of activities array.

**Daily Actvity Table View Controller**

Lists the start and end times of each activity on a daily basis.

**Analyzer View Controller**

Bar Chart – Sub UI View

Changed default class from UI View to JB Chart class.

Two Datepickers – to pick a date range.

Calculate Button – calculates total hours per actvity within the inputed date range.

Determines the index of the starting day and the index of the end day.

Inside each day, a collection of activities is retrieved and the total amount of time for each activity is added up.

Basically, calculates the difference in times.

The calculation is displayed in the bar chart.

**About View Controller**

Freeform setting allows oversized view controller to accommodate a longer scrollview.

Instructions for the app are displayed step by step using screen shots and short answers.

Information on project teammembers is displayed at the bottom.

**Schema**

3D Array

Array 1 – Dates

Array 2 – Events

Array 3 – Start and End Times

Activity Type Array – array of strings for activity titles.

App stores an array of all the days.

In each day there is an array of events.

For each event there is an array of start and end times.

**Persistance Mechanism**

Data is transferred from one class to the next via segues.

JB Chart Library (Objective C) – implemented using cocoa touch framework targets.

Objective C functions were exported to Swift classes.

XY Pie Library (Objective C) – implemented using cocoa touch framework targets.

Objective C functions were exported to Swift classes.