Abstract

The purpose of this document is to outline the design patterns used in the creation of the Pacemaker Android Application

Pacemaker android app – pattern solution

Design Patterns

Colm Carew

Email: colmcarew2@gmail.com

Student ID: 20053766

Module: Design Patterns

Course: MSc. Communications Software

Date: 17/04/2016

# ­­­­Patterns Used

* Creational Patterns
  + Builder
  + Singleton
* Structural Patterns
  + Adapter
  + Façade
* Behavioural Patterns
  + Memento
  + Model View Controller
  + Strategy
* Architectural Pattern
  + Half Sync / Half Async

# Builder Pattern

The builder pattern is a creational pattern used for creating complex objects.

The builder pattern solves the problem of creating an object with many parameters. Some of these parameters may be mandatory and others will be optional. This eliminates the need for multiple overridden constructors.

The builder pattern is robust and more maintainable however it is quite verbose and requires code duplication.

The builder pattern can be seen implemented in the ActivitiesList activity in the Android project.



**Figure 1.0 : Builder Example**

# Singleton

The singleton patterns is a creational pattern and restricts the instantiation of a class to one object.

This is useful for when one object is needed throughout the application.

In Android the Singleton can be accomplished via having a custom class extending the Application class. When the object of the extended application class is required in an activity they can be obtained by using the getApplication() method.

In the case of this Android App it is used for sharing the logged in user, their details and associated methods amongst the necessary Android Activities.



**Figure 2.0 Singleton Used in the Android App**