

Implementation Document

Focus!

Prepared by ProFlo

Mohammad Usman Sohail

Alexina Boudreaux

Cameron Chilson

Forrest Dunlap

Patrick Truong

1. Preface	3
2. Introduction	3
3. Changes To The Model Class Diagram	4
4 Changes to the Model Class Diagram	7

1. Preface

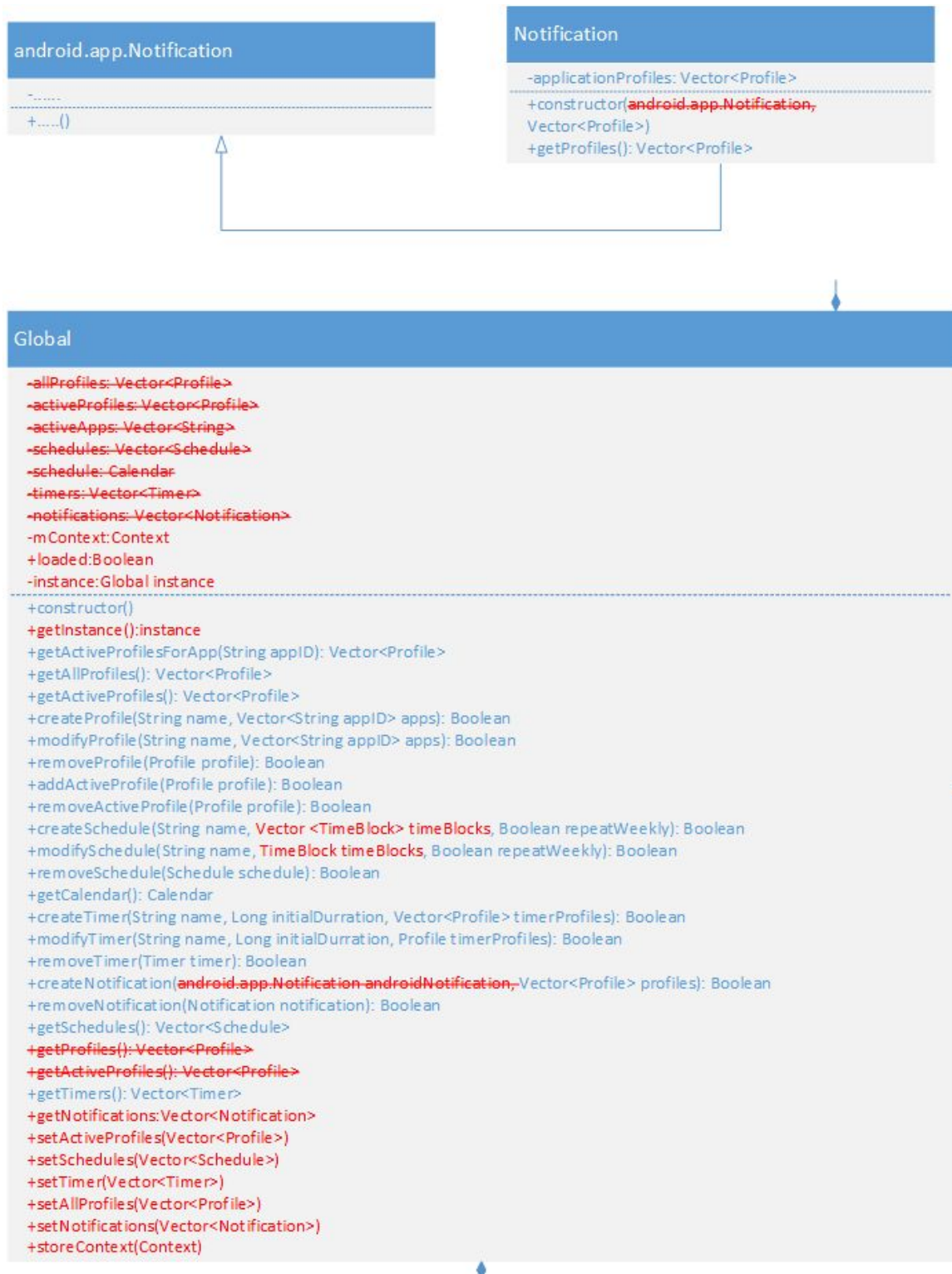
The purpose of this Implementation Document is to provide an overview of the changes that were made from the original Software Design Documentation as well as our rationalization as to why these changes were needed and beneficial. The diagrams below will note the changes which are seen in the red font. Anything that is red with a strikethrough was removed, anything that is just red was added, and any blue remained the same.

2. Introduction

Focus! Is implemented on Android using an implicit invocation architectural style, with a Model-View-Controller pattern. The model contains any and all data, the Android GUI is our view, and the controller is the backend, and all data manipulation. Due to our use of an implicit invocation style, we can use listeners to control much of the application. MVC was chosen largely due to the dynamic nature of the application, and the requirements of working within Android.

No changes were made to the original architectural style or design as the high level parts seemed to work exceptionally well together. There were however, small tweaks made to the class diagrams of both the model and the controller. Section 3 will go into detail about the changes to the class diagram for the model as well as the reasoning and section 4 will provide the same for the class diagram of the controller.

3. Changes To The Model Class Diagram





3.1 Changes to Notification

The only change that was made to the Notification class was in the constructor, originally it took in an android.app.Notification object, however, since it already inherits from that class there was no need for that to be passed in.

3.2 Changes to Global

There were multiple made in the Global class. The first was utilizing the TimeBlock class we created for creating and modifying the schedule. Previously a vector of Triplets was passed through, by utilizing the TimeBlock class it will be much easier and cleaner when utilizing these methods in other classes.

Next we adapted the class to use instances instead of storing all the date in a single object. This enables us to integrate the SharedPreferences that Android uses much more smoothly providing more adaptability and increasing the app's ability to scale. There were also getters and setters added which utilize these SharedPreferences and help to set some of the global variables within each instance.

Finally, there were a few removals made from the original diagram. The first was in the createNotification method and that was removed so it could match the newly changed constructor in the Notification class(see above for further explanation). The removal of the getProfiles and getActiveProfiles methods are simply because they are duplicates and exist further up in the UML.

3.3 Changes to Profile

The only change made to the active class was the added parameter of a boolean representing if a profile is active or not in the constructor. This allows for the active status of a profile to be set on initialization and prevent undefined behavior if isActive() is called before that variable is set.

3.4 Changes to Timer

There were 2 major changes made to timer. The first was implementing a timer member which utilizes android's CountDownTimer object. The main decision for this was we decided it would be much simpler to use the pre existing object instead of trying to rewrite a timer when one already existed.

The second change made was to add a Boolean that tracks whether a timer is currently paused or not. This will be a much easier method than attempting to use the timer object itself and see if it is currently counting down or if it is inactive.

The final change made was to simply get rid of the duplicate getProfiles method.

4 Changes to the Model Class Diagram

Android.Activity.MainActivity

```
-globalData: Global
-profiles: Android.view
-schedules: Android.view
-timers: Android.view
-notifications: Android.view
-profileFrame: FrameLayout
-schedulesFrame: FrameLayout
-timerFrame: FrameLayout
-notificationsFrame: FrameLayout
+PROFILE_STATUS: String
+SCHEDULE_STATUS: String
+TIMER_STATUS: String
+ACTION_NOTIFICATION_LISTENER_SETTINGS: String
+ENABLED_NOTIFICATION_LISTENERS: String
+ANDROID_MESSAGING: String
+ANDROID_EMAIL: String
+ANDROID_GESTURE_BUILDER: String
+ANDROID_API_DEMOS: String
-profileActive: Boolean
-scheduleActive: Boolean
-timerActive: Boolean
-isOn: Boolean
-hasStarted: Boolean
-activeFrame: Int
-schedulesChanged: Boolean
-timersChanged: Boolean
-profileChanged: Boolean
-notificationsChanged: Boolean
-layouts: Vector<FrameLayout>
-toolbar: Vector<Integer>
-mTextMessage: TextView
```

Android.Activity.ModifyProfileActivity

```
modifyProfile: Android.view
-globalData: Global
-TAG: String
-nameModified: Boolean
-someAppChecked: Boolean
-errorLayout: RelativeLayout
-doneConfirm: RelativeLayout
-blocked: Vector<ApplicationInfo>
-onCreate(Bundle): Void
+onCreateOptionsMenu(Menu): Boolean
+onOptionsItemSelected(MenuItem): Boolean
+validate(): Boolean
+displayDoneConfirmation()
+displayErrorMessage(String)
+fillLayout()
+createAppRow(ApplicationInfo, int)
-createProfile(String-profileName, Vector<String-AppID>): Boolean
-editProfile(String-profileName, Vector<String-AppID>): Boolean
-notifyError(): Void
-onPostExecute(): Void
-onQuit(): Void
```

Android.Activity.ModifyTimerActivity

```
modifyTimer: Android.view
-globalData: Global
-somethingChanged: Boolean
-errorLayout: RelativeLayout
-doneConfirm: RelativeLayout
-profileNames: Vector<String>
```



```
-numHours: int
-numMinutes: int
-onCreate(Bundle): Void
+onCreateOptionsMenu(Menu): Boolean
+onOptionsItemSelected(): Boolean
+displayDoneConfirmation()
+displayErrorMessage(String)
+createTimer(String timerName, Long initialDuration,
Vector<Profiles> timerProfiles): Boolean
+editTimer(String timerName, Long initialDuration, Vector<Profiles>
timerProfiles): Boolean
+notifyError(): Void
+onPostExecute(): Void
+onQuit(): Void
```

```

modifySchedule():Android.view
-name Modified:Boolean
-someBlockChanged:Boolean
-startPickerActive:Boolean
-newStartHour:int
-newStartMin:int
-newStopHour:int
-newStopMin:int
-errorLayout:RelativeLayout
-doneConfirm:RelativeLayout
-profileNames:Vector<String>

-globalData:Global
-onCreate(Bundle):Void
+onOptionsItemSelected:Boolean
+createSchedule
+fillLayout
+displayBackConfirmation
+displayErrorMessage(String)
+validate():Boolean
+onCreateOptionsMenu(Menu):Boolean
+selectProfilesBlock
+createNewTimeBlock
+setTime(TextView,TimePicker,TimePicker)
+createTimeBlock(int,int,String,String,int,Vector<Integer>,int)
+createSchedule(String-scheduleName,Boolean-repeatWeekly,
Vector<Triplet<Long-starttime,Long-endtime,Vector<String>>
times>):Boolean
+editSchedule(String-scheduleName,Vector<Triplet<Long-starttime,
Long-endtime,Vector<String>>times>):Boolean
+notifyError():Void
-onAddTimeBlock():Void
-onPostExecute():Void
-onQuit():Void

```

We also included many global variables that would be used throughout the rest of the app, as well as things such as layout and frame transitions. In short the majority of the changes within the activity class stem from us changing how we decided to structure the class interactions, as well as the scope of what we use the activity class for.