Design Document

**Classes**

* **AlarmClock:**  Main class that listens to user requests and spread the task to the specific class below.
* **SetAlarm:** Listens to AlarmClock for user’s input and inserts input into database. Also remove and fetches alarm records. Superclass of AlarmEdit.
* **AlarmEdit:**  Modifies the settings of any existing alarm. Extends the SetAlarm class
* **Alarm:**  Each instance indicates an alarm schedule with no special mode attached to it. Also becomes the superclass for FailSafeAlarm and WakeUpAlarm.
* **FailSafeAlarm:**  An instance of the alarm with fail safe mode enabled. Extends the Alarm class and includes a

snooze counter. When fail safe is active, intialize ActiveAlarm with the ring duration user has

* **WakeUpAlarm:** Extends the Alarm class and calls WakeUpChallenge to generate the UI to the algorithm

problem. Simultaneously, ActiveAlarm is called to generate the alarm sound and will listen to the WakeUpAlarm if user answer’s question correctly.

* **ActiveAlarm:** Turns on the alarm ring or vibrate. Called by Alarm, FailSafeAlarm and WakeUpAlarm.
* **WakeUpChallenge:** Randomly generate a math problem with respect to the difficulty level passed through with parameter values.

The following diagram depicts the relationship between each class and some of the operations specific to the classes’ functionality.

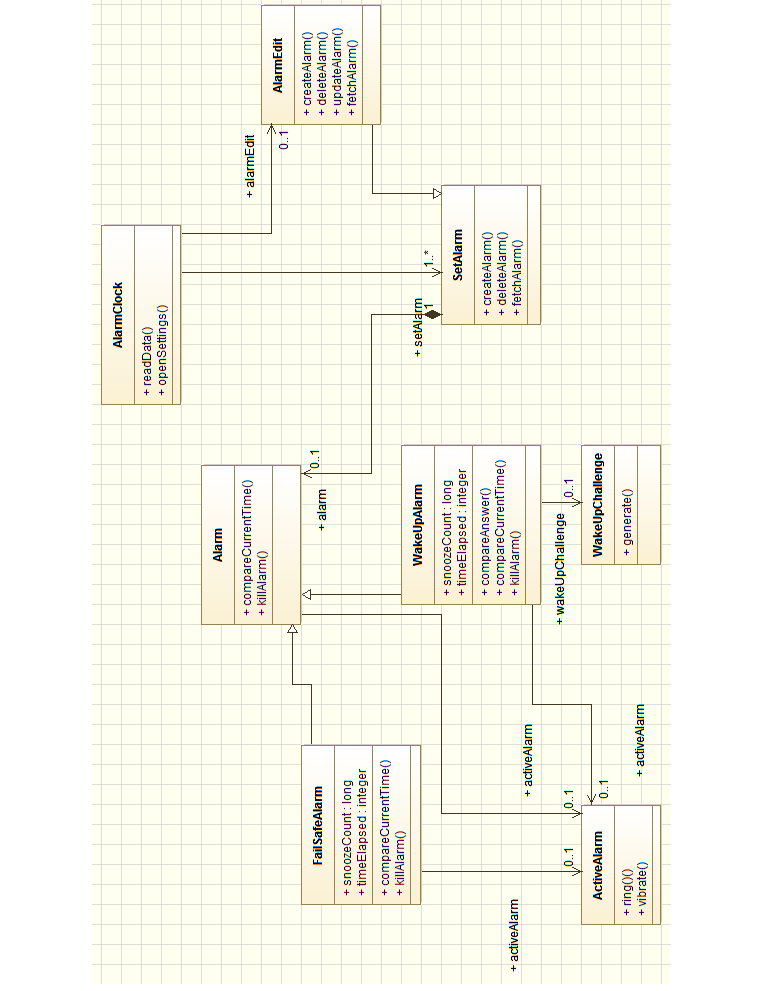
****

Diagram 1: Class diagram of the alarm clock app

**User Interface**

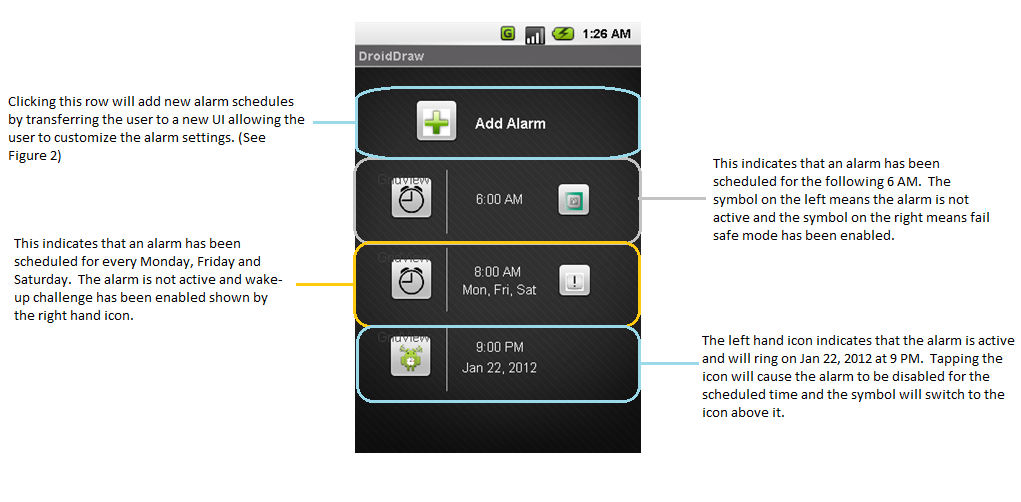
****

Figure 1: Main Menu of alarm app

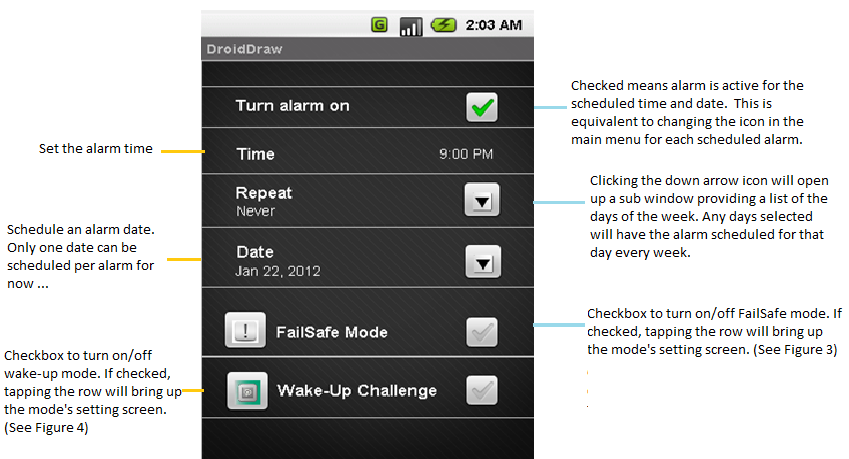


Figure 2: Alarm settings. User can access this screen either by adding an alarm

or modifying an existing alarm.



Figure 3: FailSafe mode settings.

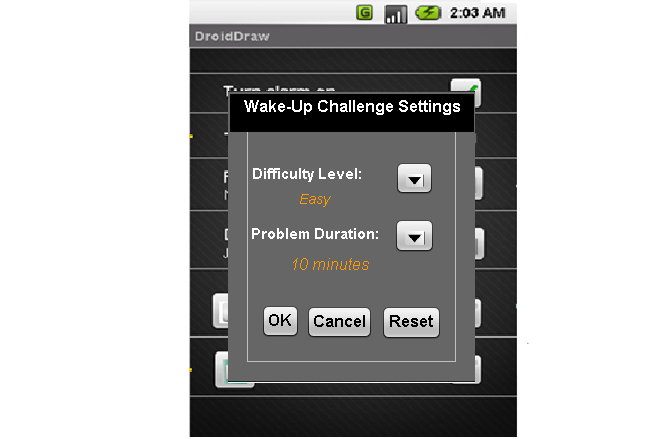


Figure 4: Wake-Up Challenge Settings

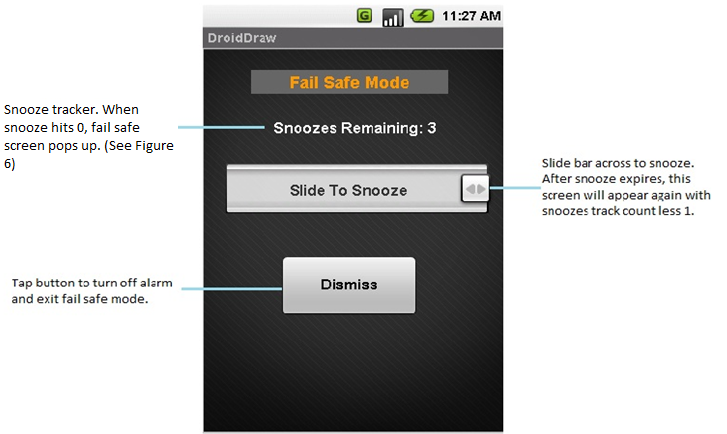


Figure 5: Alarm with Fail Safe Mode on.

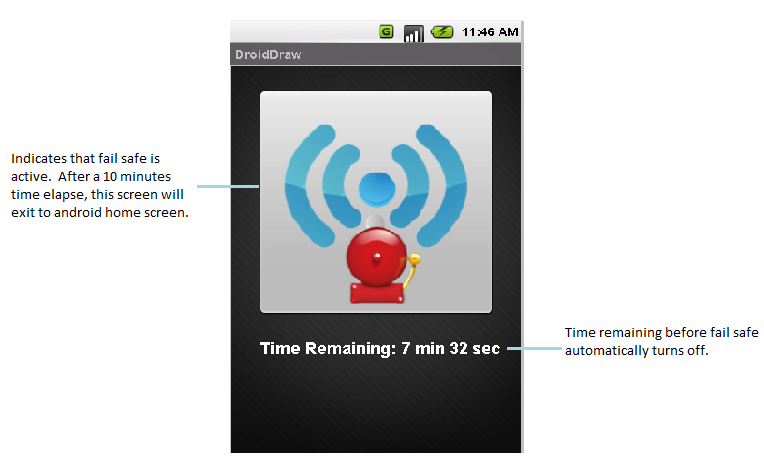


Figure 6: Fail Safe Active.

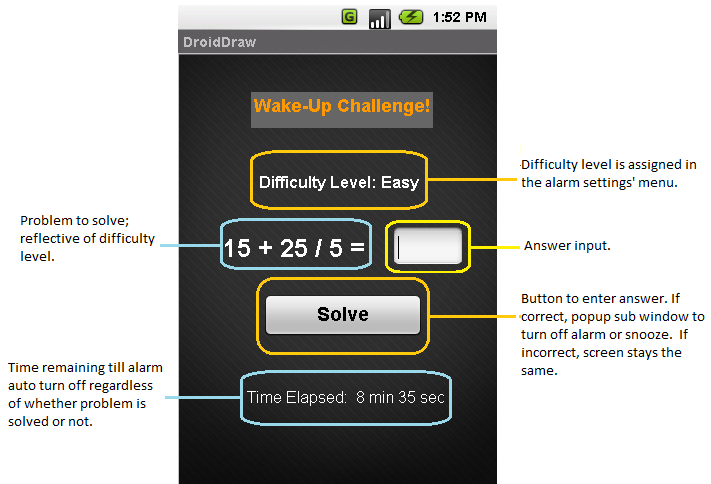


Figure 7: Wake-Up Challenge Active

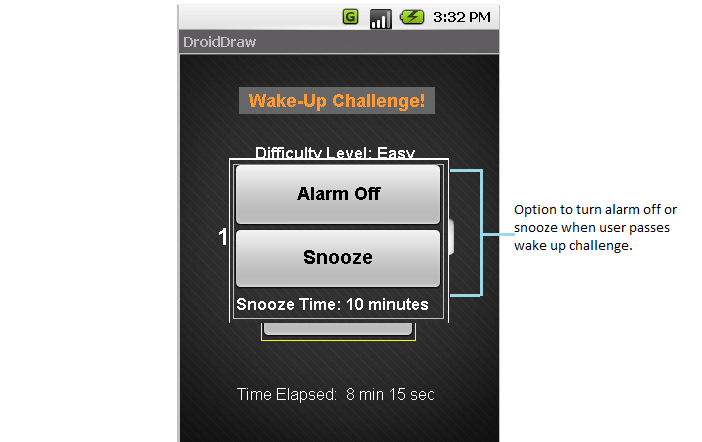


Figure 8: Wake-Up Challenge solved