

# Clock Project

## 1 Objective

In this project, we want you to create a Java application with GUI interface. You will mainly use Swing components for creating a user interface (GUI) and then implementing the functions of the application using java programming. This project entitled as “**Clock**” and this project as a one of two project assigned for this course.

## 2 Descriptions

### 2.1 User Interface

Your application will has GUI like the suggested interface shown in Figure 1 with 2.

The main parts of the GUI are: **Timer** and **Alarms**. However, **you can create different GUI** using traditional menu, or any design you like but still you have to include the entire functions which are listed in the next section.

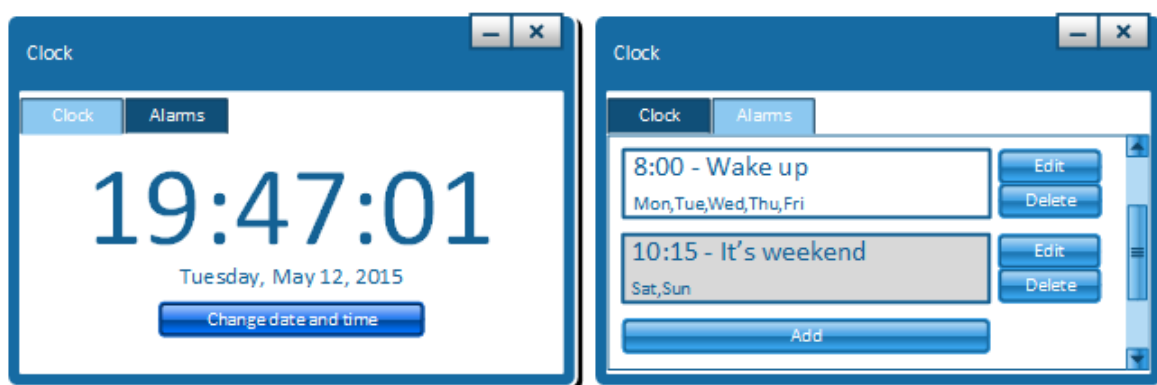


Figure 1: Clock sample user interface

### 2.2 Functions

The application will have 2 groups of functions:

#### ❖ Timer Function:

- The application must show the time. It is fine if the clock is not synchronize with your computer clock, but the clock must run.
- The application must **show the day of week, day, month and year, hour, minute and second**.
- User can **change the date and the time**. In this example, user must click [Change date and time] button to show the dialog below.

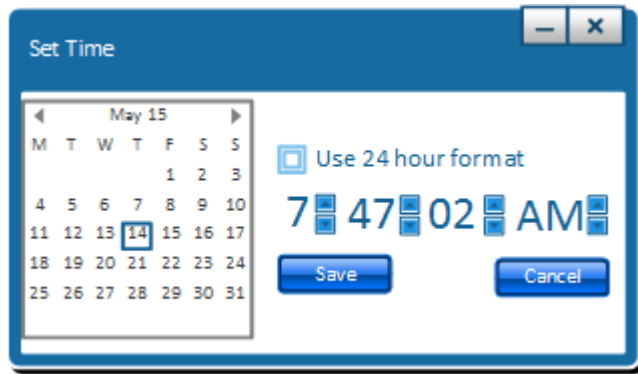


Figure 2: Change date and time dialog

- There is a setting option to **change the format of display time** like “7:47:02 PM” or “19:47:02”. This choice will apply for the main display (Figure 3) and edit the time in **Alarm Functions** too (Figure 4).

❖ **Alarm functions:**

This application **supports many alarms**.

- The sample dialog to set the alarm information is like below figure. User can **add new, edit, delete** the alarms.



Figure 5: Alarm function main GUI

- User can change the alarm information by clicking [Add] or [Edit] button, a dialog is shown like this.

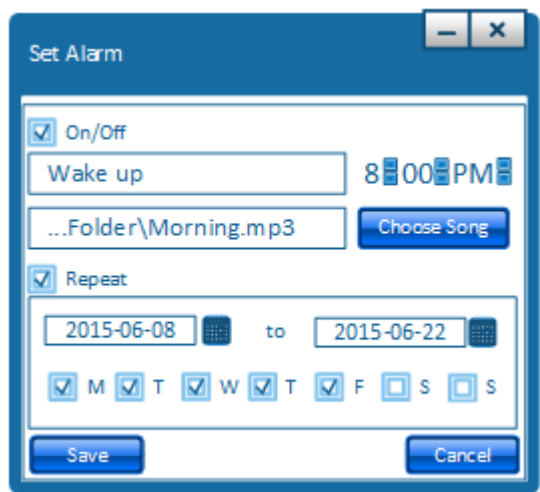


Figure 6: Set Alarm dialog

- Each alarm has these information:
  - Is this alarm is on or off (in reference GUI, the alarm with grey background is disable).
  - Alarm label.
  - Alarm time.
  - Path to .mp3 file, which can be select using [Open Dialog] when, clicking the [Choose Song] button. This song will play when the alarm is active.
  - Can be repeat on every selected day of week and in the selected date range. If the [Repeat] choice is not selected, it means it is only set one times then automatically turn off.
- The information of all alarms must be stored in a file or many files, so the application will read from them when application start, and write to them when user change the alarm information. Student can freely choose how to save the data as a text file or as a binary file.
- When the time come and match the alarm, then the application must **show the alarm dialog** like below and play the selected song in [Set Alarm] dialog

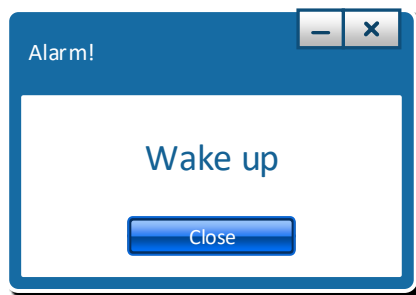


Figure 7: Dialog when alarm is active.

**Hint:**

- Student may have to search for external library to use in your program to play mp3 file.
- Student may have to create threads to update timer and check the alarm.
- The GUI shown in this document are for reference, student can freely design different GUI, but make sure the application have all the functions above.

### 3 Extra Credit

In this project, you can earn extra credit by

- Designing friendlier user interface.
  - More points for **material design**. Sample library: <http://www.jfoenix.com/>
- Add Stop watch function
- Add Countdown watch function
- Change the skin (Look-and –Feel) of GUI.

### 4 Project Submission

Here are the details for how to submit your project:

1. Create an executable JAR file named **clock.jar** that contains your runnable application.
2. Create a new directory using your student IDs. Example: **20150001\_20150002**
3. Put the **jar file** and your **Eclipse project folder** into the directory.

4. Put also **README.txt** file to the directory. This file should contain student name (in English), the version of Java you used as well as any special info we might need to know about your program (example: let us know if you did some extra things).
5. **ZIP** this directory and submit it via e-class.

**Notes:**

All the comments on the code and the names of variables, classes must be in English.

Please take care to remove any platform dependencies, such as hardcoded path names or dependence on a particular look-and-feel that may not exist on all platforms.

Also, if you use any images in your application, please make sure that you included these images in your JAR file and that your code will refer to them and load them properly when they're in this JAR file (see [this page](#) for some details on how to include and load images from within a JAR file).

## 5 Grading

Grading for this assignment will follow this breakdown:

- ✓ 50% functionality
- ✓ 30% architectural design, coding style (code convention)
- ✓ 20% commenting
- ✓ 20% extra credit

Please let the LA or the professor know if you have any questions or something you don't understand by visiting Lab Room #1207 or Prof Room #317 at hi-tech Center or by asking question on e-class.

If you have questions about the project requirement, I suggest you post it in public so other student can see them too but if you have a question about your project implementation, then post the question in private.

Good luck!