```java
import javax.swing.*;
import javax.sound.sampled.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.Calendar;
import java.util.Timer;
import java.util.TimerTask;

public class ReminderApp {
    private JFrame frame;
    private JComboBox<String> dayComboBox;
    private JComboBox<String> timeComboBox;
    private JComboBox<String> activityComboBox;
    private JButton setReminderButton;

    public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> new ReminderApp().createAndShowGUI());
    }

    private void createAndShowGUI() {
        frame = new JFrame("Reminder App");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(300, 200);
        frame.setLayout(new GridLayout(4, 2));

        String[] days = {"Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"};
        dayComboBox = new JComboBox<>(days);
        frame.add(new JLabel("Select Day:"));
        frame.add(dayComboBox);

        String[] times = {
            "06:00", "07:00", "08:00", "09:00", "10:00", "11:00", "12:00",
            "13:00", "14:00", "15:00", "16:00", "17:00", "18:00", "19:00",
            "20:00", "21:00", "22:00", "23:00"
        };
        timeComboBox = new JComboBox<>(times);
        frame.add(new JLabel("Select Time:"));
        frame.add(timeComboBox);

        String[] activities = {"Wake up", "Go to gym", "Breakfast", "Meetings", "Lunch", "Quick nap", "Go to library", "Dinner", "Go to sleep"};
        activityComboBox = new JComboBox<>(activities);
        frame.add(new JLabel("Select Activity:"));
        frame.add(activityComboBox);

        setReminderButton = new JButton("Set Reminder");
        setReminderButton.addActionListener(new ReminderButtonListener());
        frame.add(setReminderButton);

        frame.setVisible(true);
    }

    private class ReminderButtonListener implements ActionListener {
        @Override
        public void actionPerformed(ActionEvent e) {
            String selectedDay = (String) dayComboBox.getSelectedItem();
            String selectedTime = (String) timeComboBox.getSelectedItem();
            String selectedActivity = (String) activityComboBox.getSelectedItem();

            if (selectedDay != null && selectedTime != null && selectedActivity != null) {
                setReminder(selectedDay, selectedTime, selectedActivity);
            }
        }

        private void setReminder(String day, String time, String activity) {
            String[] timeParts = time.split(":");
            int hour = Integer.parseInt(timeParts[0]);
            int minute = Integer.parseInt(timeParts[1]);

            Calendar now = Calendar.getInstance();
            Calendar reminderTime = Calendar.getInstance();

            reminderTime.set(Calendar.HOUR_OF_DAY, hour);
            reminderTime.set(Calendar.MINUTE, minute);
            reminderTime.set(Calendar.SECOND, 0);

            // Set the day of the week for the reminder
            int dayOfWeek = getDayOfWeek(day);
            reminderTime.set(Calendar.DAY_OF_WEEK, dayOfWeek);

            if (reminderTime.before(now)) {
                reminderTime.add(Calendar.WEEK_OF_YEAR, 1);
            }

            Timer timer = new Timer();
            TimerTask task = new TimerTask() {
                @Override
                public void run() {
                    playSound();
                    JOptionPane.showMessageDialog(frame, "Reminder: " + activity);
                }
            };

            long delay = reminderTime.getTimeInMillis() - now.getTimeInMillis();
            timer.schedule(task, delay);
```

```java
        }

        private int getDayOfWeek(String day) {
            switch (day) {
                case "Monday": return Calendar.MONDAY;
                case "Tuesday": return Calendar.TUESDAY;
                case "Wednesday": return Calendar.WEDNESDAY;
                case "Thursday": return Calendar.THURSDAY;
                case "Friday": return Calendar.FRIDAY;
                case "Saturday": return Calendar.SATURDAY;
                case "Sunday": return Calendar.SUNDAY;
                default: return Calendar.MONDAY;
            }
        }

        private void playSound() {
            try {
                AudioInputStream audioInputStream = AudioSystem.getAudioInputStream(getClass().getResource("chime.wav"));
                Clip clip = AudioSystem.getClip();
                clip.open(audioInputStream);
                clip.start();
            } catch (Exception ex) {
                ex.printStackTrace();
            }
        }
    }
}

```