



**Southern Luzon State University
College of Engineering
S.Y. 2023 – 2024**



**CPE18 – SOFTWARE DESIGN & DEVELOPMENT
TIMECHECKR**

**SEAN CARLO G. FLORES
BSCPEIII – GF
JUNE 10, 2024**

INTRODUCTION:

With our cutting-edge student attendance management software, our software which was created with educators in mind, simplifies and improves the usability of student attendance tracking. With just a single click, teachers may mark each student as "Present" or "Absent" thanks to the software's user-friendly toggle controls. This guarantees easy and quick updating of attendance. Furthermore, the software counts the number of students who are designated as present automatically, giving a clear, instantaneous picture of attendance status and making it easier to identify absent pupils.

On the other hand, this program also features timer, where it can monitor the time being spent on each activities, an alarm feature that also help teachers or faculty to strictly observe time. On managing the schedules, this program also has time management where schedules are present for different rooms to never miss out a class. Lastly, there is also a calendar where users can navigate the day of the month.

Our software's main advantages include enhanced convenience, accuracy, and efficiency in managing attendance as well as meaningful insights on student attendance trends. For teachers who want to make attendance monitoring easier so they can concentrate more on teaching and less on administrative work, our Student Attendance Management Software is the ideal choice.

TABLE OF CONTENTS

SLDC(Activity 1)	4
Homepage Design / Menu & Logo (Images)	4
Software Data Modeling (Use Case & Flow Chart).....	5
Software Layout Design (F1 – F5 Based on Flowchart).....	7
Summary of Codes (Per Class).....	9
Screenshot of Actual Usage.....	44

Concept Creation

Abstract

The attendance rate is important because students are more likely to succeed in academics when they attend school consistently. It's difficult for the teacher and the class to build their skills and progress if a large number of students are frequently absent.
-by The Greatschools Editorial Team – 2023

With the statement above, the software to be developed will be a live attendance checker that can be flashed on screen. And upon the end of the class, it can be reset for the other subject to be used. This programmed software can be utilized to easily track attendance of the students and even faculty members.

Title

The name '**TimeCheckr**' is made to display the attendance of the students/faculty members in school which can easily identify the attendance upon looking at the display.

Intended Audience

Timecheckr can be useful for students or faculty members to help them easily identify the ones that are present that time or that day.

Features

- **Attendance Checkr** – Allows admin/teacher to tick their name if they are present
- **Calendar** – It enables the user to check the dates upon the given year.
- **Alarm** – Helps school to notify the students about the time.
- **Timer** – Counts the minutes of attendance that the user is in.
- **Time Management** – Helps students to show their assigned schedule from Monday to Friday.

Sources

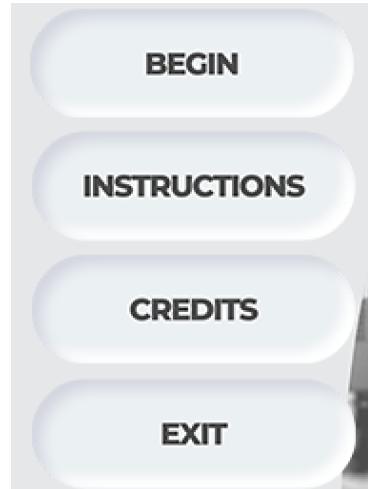
The Greatschools Editorial Team – 2023

Homepage/MENU & LOGO

Homepage:



Menu:

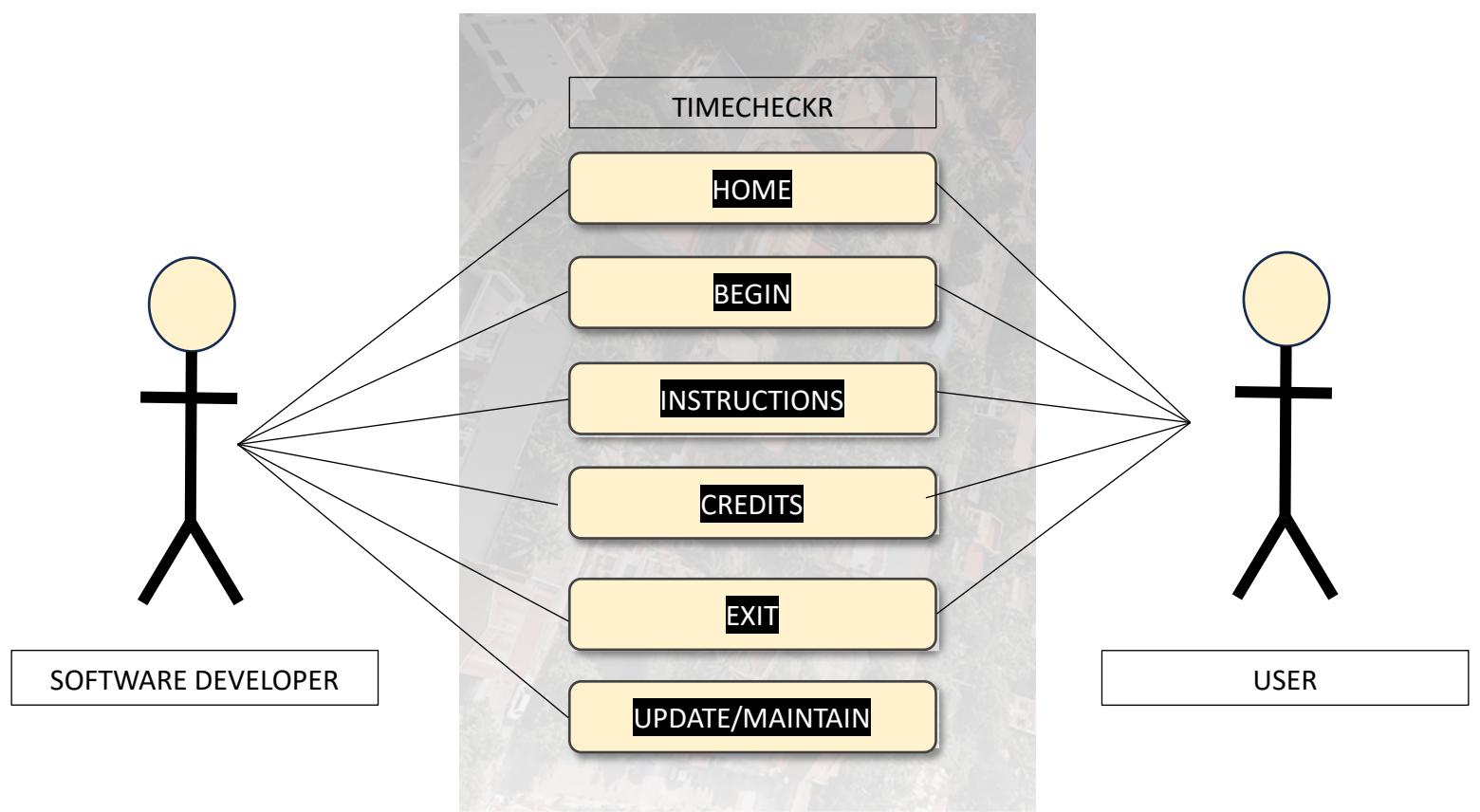


Logo:

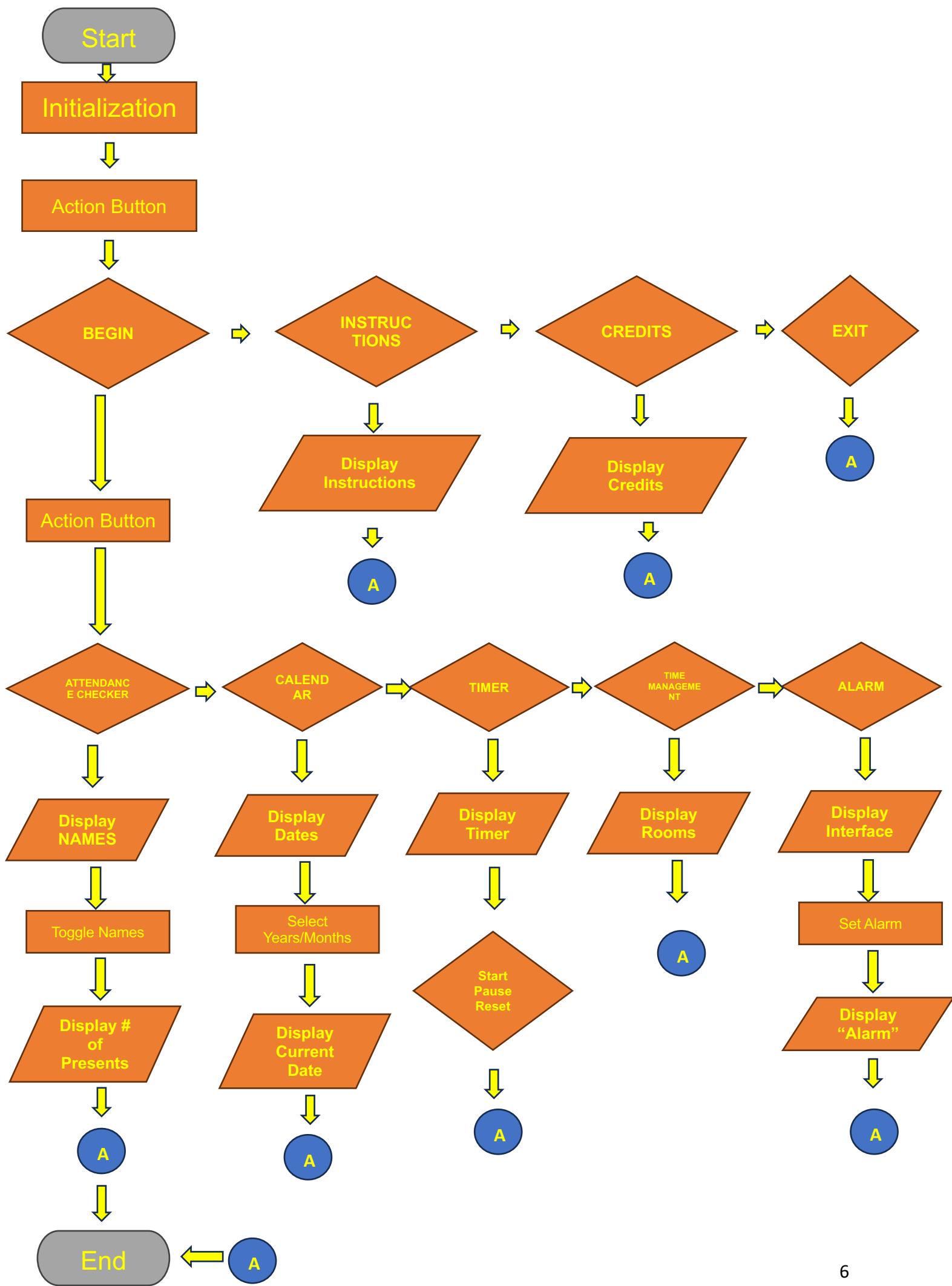


SOFTWARE DATA MODELLING

UseCase Diagram:



Flowchart
FLOWCHART DIAGRAM:

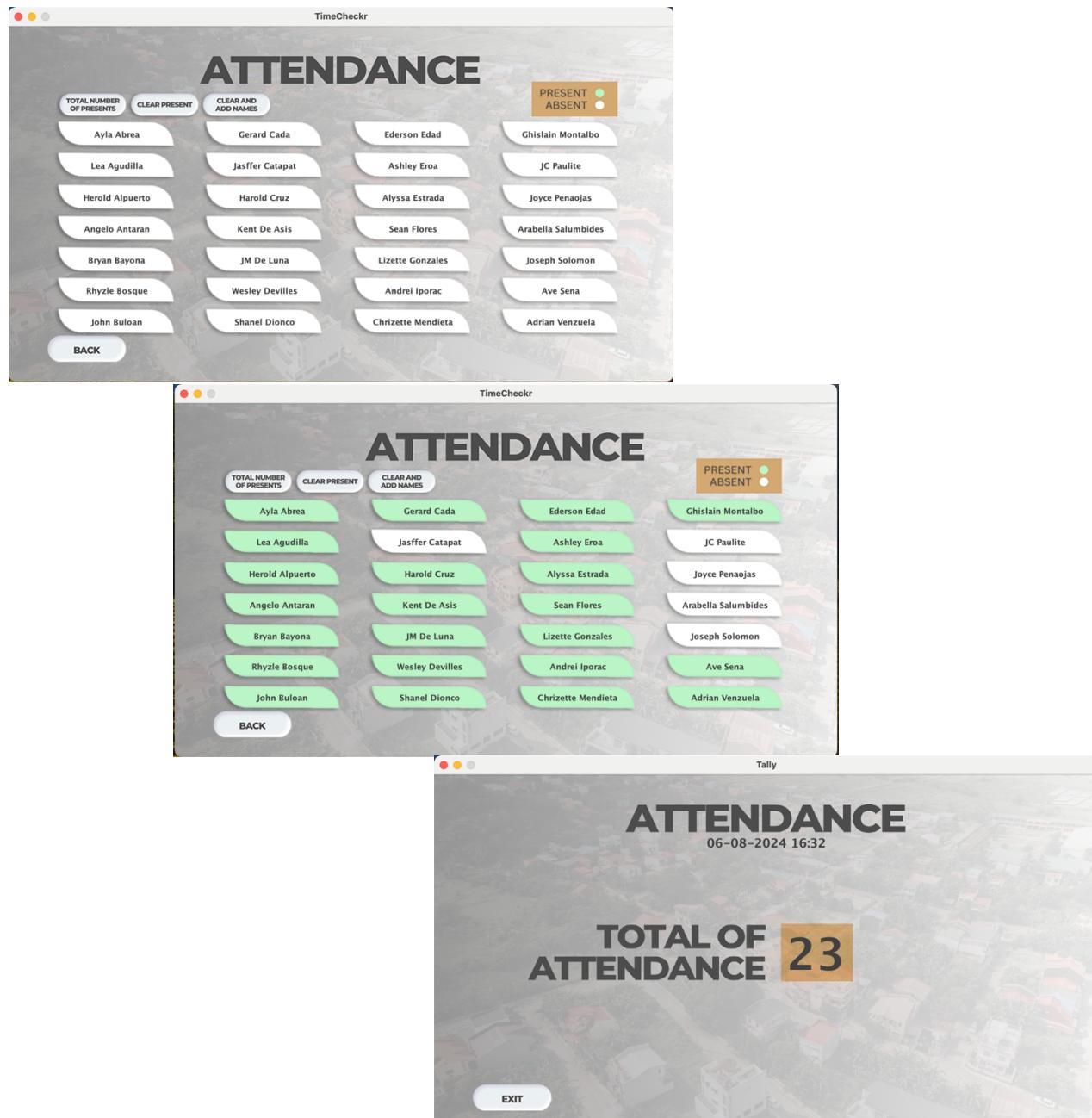


DISCUSSION

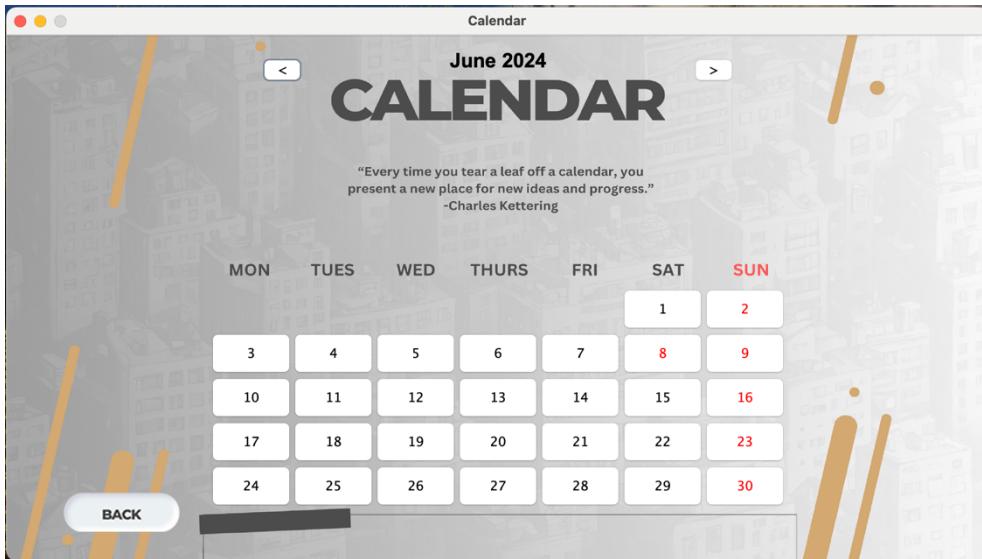
The use case diagram show the different features of TimeCheckr that can be accessed by the developer and its user/s. The first button shows the Begin button where if it is clicked, the five features are able to be accessed, the first feature there is the Attendance Checker where the user can identify the number of present students, the next is the Timer to set a time alongside with Alarm to strictly measure time and the Calendar and Time Management feature to track days. The next button present to the homepage is the Instruction button where users can see the procedures to see how the features will work and also next to it is the Credits button to showcase the developer of the program and lastly, the button of exit to terminate the program.

SOFTWARE LAYOUT DESIGN

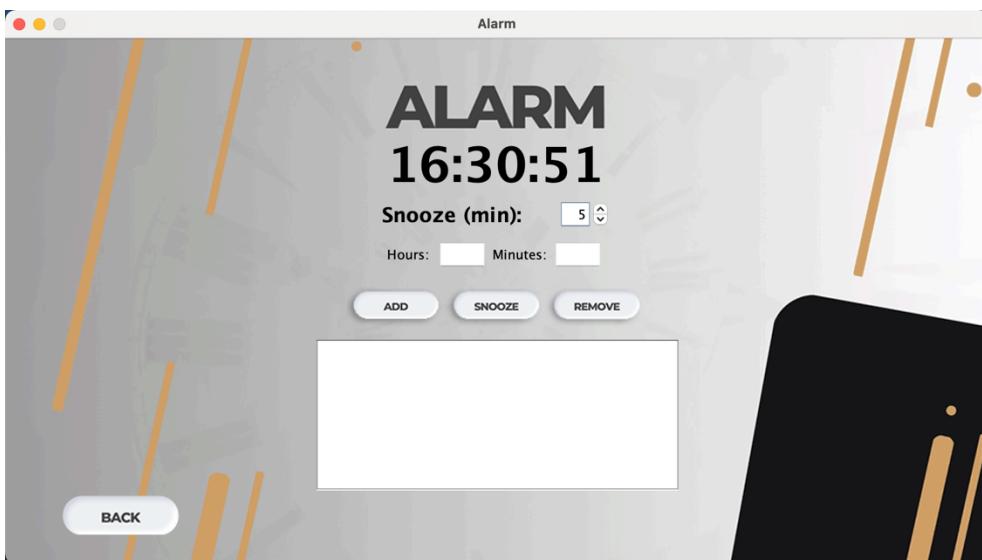
Feature 1



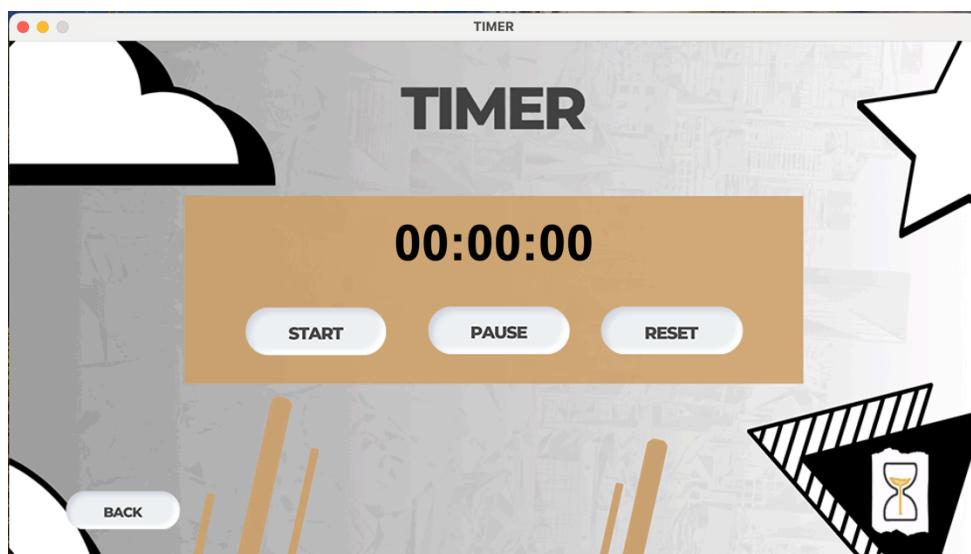
Feature 2



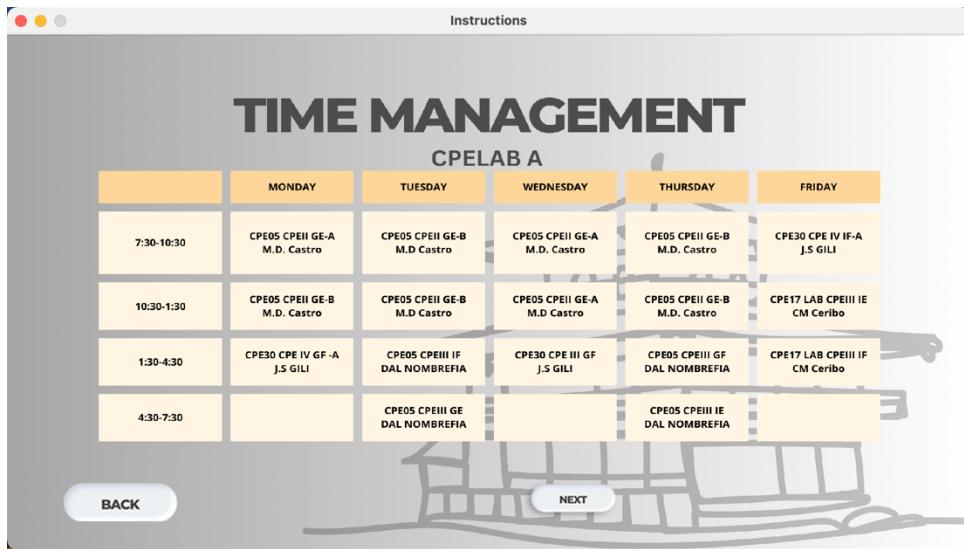
Feature 3



Feature 4



Feature 5



SUMMARY OF CODES (PER CLASS)

//Main Class

```
package Package_TimeCheckr;

import javax.sound.sampled.*;
import javax.swing.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.IOException;
import java.net.URL;

public class TimeCheckr_Main {

    // main page of the application
    public static void main(String[] args) {
        new Next();
    }

    class Next extends JFrame {

        public Next() {
            JFrame frame = new JFrame("TimeCheckr");

            // custom panel for displaying the background image
            JLabel background = new JLabel(new ImageIcon(getClass().getResource("/BG.png")));
            frame.add(background);
            frame.setContentPane(background);

            JButton Begin = new JButton();//creating instance of JButton
            Begin.setIcon(new ImageIcon(getClass().getResource("/Begin.png")));
        }
    }
}
```

Button Begin = new JButton(); //creating instance of JButton
 Begin.setIcon(new ImageIcon(getClass().getResource("/Begin.png")));

```

Begin.setBounds(390, 240, 210, 70); //x axis, y axis, width, height
Begin.setContentAreaFilled(false); //this is the one to remove background of button
Begin.setBorder(null);
Begin.setOpaque(false);
Begin.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        playSound("/UI.wav");
        frame.dispose();
        new Function_Select();
    }
});
});

JButton Instruction = new JButton(); //creating instance of JButton
Instruction.setIcon(new ImageIcon(getClass().getResource("/Instructions.png")));
Instruction.setBounds(390, 305, 210, 70); //x axis, y axis, width, height
Instruction.setBorder(null);
Instruction.setContentAreaFilled(false);
Instruction.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        playSound("/UI.wav");
        frame.dispose();
        Instructions I = new Instructions();
        I.setVisible(true);
        I.AboutDisplay();
    }
});
);

JButton Credits = new JButton(); //creating instance of JButton
Credits.setIcon(new ImageIcon(getClass().getResource("/Credits.png")));
Credits.setBounds(390, 370, 210, 70); //x axis, y axis, width, height
Credits.setBorder(null);
Credits.setContentAreaFilled(false);
Credits.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        playSound("/UI.wav");
        frame.dispose();
        new Credit();
    }
});
);

JButton exit = new JButton(); //creating instance of JButton
exit.setIcon(new ImageIcon(getClass().getResource("/Exit.png")));
exit.setBounds(390, 435, 210, 70); //x axis, y axis, width, height
exit.setContentAreaFilled(false);
exit.setBorder(null);
exit.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        System.exit(0);
    }
});
);

frame.add(Begin); //adding button in JFrame
frame.add(Instruction);
frame.add(exit);

```

```

frame.add(Credits);
frame.setBounds(360, 230, 980, 551);

frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.setLayout(null); //using no layout managers
frame.setVisible(true);//making the frame visible
frame.setResizable(false);
}

// Method to play sound
public static void playSound(String resourcePath) {
    try {
        URL soundURL = Next.class.getResource(resourcePath);
        if (soundURL != null) {
            AudioInputStream audioStream =
AudioSystem.getAudioInputStream(soundURL);
            Clip clip = AudioSystem.getClip();
            clip.open(audioStream);
            clip.start();
        } else {
            System.err.println("Sound file not found: " + resourcePath);
        }
    } catch (UnsupportedAudioFileException | IOException | LineUnavailableException
e) {
        e.printStackTrace();
    }
}
}

```

//Instructions Class

```

package Package_TimeCheckr;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.sound.sampled.*;
import java.io.IOException;

class Instructions extends JFrame implements ActionListener {

    // Background label for the About screen
    JLabel AboutBG;

    // Method to set up the frame properties
    void frame() {
        // Back button
        JButton Back = new JButton();
        Back.setIcon(new ImageIcon(getClass().getResource("/Home.png")));
        Back.setBounds(10, 440, 210, 70); // x axis, y axis, width, height
        Back.setBorder(null);
        Back.setContentAreaFilled(false);
        Back.setOpaque(false);
    }
}

```

```

Back.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        playSound("/Back.wav");
        dispose();
        new Next();
    }
});
add(Back);

// Set default close operation, size, title, and layout
setBounds(360, 230, 980, 551);
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setLayout(null); // using no layout managers
setVisible(true); // making the frame visible
setResizable(false);
}

// Method to display the About screen
void AboutDisplay() {
    // Set background image for the About screen
    AboutBG = new JLabel(new ImageIcon(getClass().getResource("/Back.png")));
    setContentPane(AboutBG);

    // Display the first About section
    Ab1();
}

// Method to display the first About section
void Ab1() {
    AboutBG = new JLabel(new ImageIcon(getClass().getResource("/Ins1.png")));
    setContentPane(AboutBG);

    // Configure and add the Next button
    ImageIcon Next = new ImageIcon(getClass().getResource("/nxt.png"));
    JButton nextButton = new JButton(Next);
    nextButton.setCursor(new Cursor(Cursor.HAND_CURSOR));
    nextButton.setOpaque(false);
    nextButton.setBorder(null);
    nextButton.setBorderPainted(false);
    nextButton.setContentAreaFilled(false);
    nextButton.setFocusPainted(false);

    nextButton.setBounds(550, 400, 100, 100);

    add(nextButton);

    // Action listener to move to the next section
    nextButton.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            setVisible(false);
            Ab2();
        }
    });
}

frame();
}

```

```

// Method to display the second About section
void Ab2() {
    AboutBG = new JLabel(new ImageIcon(getClass().getResource("/Ins2.png")));
    setContentPane(AboutBG);

    // Configure and add the Previous and Next buttons
    ImageIcon Prev = new ImageIcon(getClass().getResource("/prev.png"));
    JButton prevButton = new JButton(Prev);
    prevButton.setCursor(new Cursor(Cursor.HAND_CURSOR));
    prevButton.setOpaque(false);
    prevButton.setBorder(null);
    prevButton.setBorderPainted(false);
    prevButton.setContentAreaFilled(false);
    prevButton.setFocusPainted(false);

    prevButton.setBounds(340, 401, 100, 100);

    ImageIcon Next = new ImageIcon(getClass().getResource("/nxt.png"));
    JButton nextButton = new JButton(Next);
    nextButton.setCursor(new Cursor(Cursor.HAND_CURSOR));
    nextButton.setOpaque(false);
    nextButton.setBorder(null);
    nextButton.setBorderPainted(false);
    nextButton.setContentAreaFilled(false);
    nextButton.setFocusPainted(false);

    nextButton.setBounds(550, 400, 100, 100);

    add(prevButton);
    add(nextButton);

    // Action listener to move to the previous section
    prevButton.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            setVisible(false);
            AboutDisplay();
        }
    });
}

// Action listener to move to the next section
nextButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        setVisible(false);
        Ab3();
    }
});
frame();
}

// Method to display the third About section
void Ab3() {
    AboutBG = new JLabel(new ImageIcon(getClass().getResource("/Ins3.png")));
    setContentPane(AboutBG);

    // Configure and add the Previous and Next buttons
}

```

```

ImageIcon Prev = new ImageIcon(getClass().getResource("/prev.png"));
JButton prevButton = new JButton(Prev);
prevButton.setCursor(new Cursor(Cursor.HAND_CURSOR));
prevButton.setOpaque(false);
prevButton.setBorder(null);
prevButton.setBorderPainted(false);
prevButton.setContentAreaFilled(false);
prevButton.setFocusPainted(false);

prevButton.setBounds(340, 401, 100, 100);

ImageIcon Next = new ImageIcon(getClass().getResource("/nxt.png"));
JButton nextButton = new JButton(Next);
nextButton.setCursor(new Cursor(Cursor.HAND_CURSOR));
nextButton.setOpaque(false);
nextButton.setBorder(null);
nextButton.setBorderPainted(false);
nextButton.setContentAreaFilled(false);
nextButton.setFocusPainted(false);

nextButton.setBounds(550, 400, 100, 100);

add(prevButton);
add(nextButton);

// Action listener to move to the previous section
prevButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        setVisible(false);
        Ab2();
    }
});

// Action listener to move to the next section
nextButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        setVisible(false);
        Ab4();
    }
});

frame();
}

void Ab4() {
    AboutBG = new JLabel(new ImageIcon(getClass().getResource("/Ins4.png")));
    setContentPane(AboutBG);

    // Configure and add the Previous and Next buttons
    ImageIcon Prev = new ImageIcon(getClass().getResource("/prev.png"));
    JButton prevButton = new JButton(Prev);
    prevButton.setCursor(new Cursor(Cursor.HAND_CURSOR));
    prevButton.setOpaque(false);
    prevButton.setBorder(null);
    prevButton.setBorderPainted(false);
    prevButton.setContentAreaFilled(false);
    prevButton.setFocusPainted(false);
}

```

```

prevButton.setBounds(340, 401, 100, 100);

ImageIcon Next = new ImageIcon(getClass().getResource("/nxt.png"));
JButton nextButton = new JButton(Next);
nextButton.setCursor(new Cursor(Cursor.HAND_CURSOR));
nextButton.setOpaque(false);
nextButton.setBorder(null);
nextButton.setBorderPainted(false);
nextButton.setContentAreaFilled(false);
nextButton.setFocusPainted(false);

nextButton.setBounds(550, 400, 100, 100);

add(prevButton);
add(nextButton);

// Action listener to move to the previous section
prevButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        setVisible(false);
        Ab3();
    }
});

// Action listener to move to the next section
nextButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        setVisible(false);
        Ab5();
    }
});

frame();
}

// Method to display the fourth About section
void Ab5() {
    AboutBG = new JLabel(new ImageIcon(getClass().getResource("/Ins5.png")));
    setContentPane(AboutBG);

    // Configure and add the Previous button
    ImageIcon Prev = new ImageIcon(getClass().getResource("/prev.png"));
    JButton prevButton = new JButton(Prev);
    prevButton.setCursor(new Cursor(Cursor.HAND_CURSOR));
    prevButton.setOpaque(false);
    prevButton.setBorder(null);
    prevButton.setBorderPainted(false);
    prevButton.setContentAreaFilled(false);
    prevButton.setFocusPainted(false);

    prevButton.setBounds(340, 401, 100, 100);

    add(prevButton);

    // Action listener to move to the previous section
    prevButton.addActionListener(new ActionListener() {

```

```

        public void actionPerformed(ActionEvent e) {
            setVisible(false);
            Ab4();
        }
    });

    frame();
}

// Method to play sound
public static void playSound(String soundFile) {
    try {
        AudioInputStream audioStream = AudioSystem.getAudioInputStream(
            Instructions.class.getResource(soundFile)
        );
        Clip clip = AudioSystem.getClip();
        clip.open(audioStream);
        clip.start();
    } catch (UnsupportedAudioFileException | IOException | LineUnavailableException
ex) {
        ex.printStackTrace();
    }
}

@Override
public void actionPerformed(ActionEvent arg0) {
    // TODO Auto-generated method stub
}
}

```

//Credits

```

package Package_TimeCheckr;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.sound.sampled.*;
import java.io.IOException;

public class Credit {

    public Credit(){
        JFrame frame = new JFrame("Credits");

        frame.setContentPane(new
ImageIcon(getClass().getResource("/CreditsBG.png"))));
        // Add the Back button
        JButton Back = new JButton(); // creating instance of JButton
        Back.setIcon(new ImageIcon(getClass().getResource("/Home.png")));
        Back.setBounds(10, 440, 210, 70); // x axis, y axis, width, height
        Back.setBorder(null);
        Back.setContentAreaFilled(false);
        Back.setOpaque(false);

```

```

Back.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        playSound("/Back.wav");
        frame.dispose();
        new Next();
    }
});
frame.add(Back);

frame.setBounds(360, 230, 980, 551);
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.setLayout(null);
frame.setVisible(true);
frame.setResizable(false);
}

// Custom JPanel for displaying a background image
static class BackgroundPanel extends JPanel {
    private Image backgroundImage;

    public BackgroundPanel(Image backgroundImage) {
        this.backgroundImage = backgroundImage;
    }

    @Override
    protected void paintComponent(Graphics g) {
        super.paintComponent(g);
        if (backgroundImage != null) {
            g.drawImage(backgroundImage, 0, 0, getWidth(), getHeight(), this);
        }
    }
}

// Method to play sound
public static void playSound(String soundFile) {
    try {
        AudioInputStream audioStream = AudioSystem.getAudioInputStream(
            Credit.class.getResource(soundFile)
        );
        Clip clip = AudioSystem.getClip();
        clip.open(audioStream);
        clip.start();
    } catch (UnsupportedAudioFileException | IOException | LineUnavailableException
ex) {
        ex.printStackTrace();
    }
}

//Function Select Class
package Package_TimeCheckr;

import javax.sound.sampled.*;
import javax.swing.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

```

```

import java.io.IOException;
import java.net.URL;

public class Function_Select {

    // main page of the application
    public Function_Select() {

        JFrame frame = new JFrame("Function Select");
        frame.setContentPane(new
        ImageIcon(getClass().getResource("/Function_SelectBG.png")));

        JButton Back = new JButton();//creating instance of JButton
        Back.setIcon(new ImageIcon(getClass().getResource("/Home.png")));
        Back.setBounds(10, 440, 210, 70);//x axis, y axis, width, height
        Back.setBorder(null);
        Back.setContentAreaFilled(false);
        Back.setOpaque(false);
        Back.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                playSound("/Back.wav");
                frame.dispose();
                new Next();
            }
        });
    });

    JButton AT = new JButton();//creating instance of JButton
    AT.setIcon(new ImageIcon(getClass().getResource("/AC.png")));
    AT.setBounds(40, 280, 210, 70);//x axis, y axis, width, height
    AT.setContentAreaFilled(false);
    AT.setBorder(null);
    AT.setOpaque(false);
    AT.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            playSound("/UI.wav");
            frame.dispose();
            Attendance attendance = new Attendance();
            attendance.Attendance();
        }
    });
}

JButton TM = new JButton();//creating instance of JButton
TM.setIcon(new ImageIcon(getClass().getResource("/TM.png")));
TM.setBounds(560, 350, 210, 70);//x axis, y axis, width, height
TM.setBorder(null);
TM.setContentAreaFilled(false);
TM.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        playSound("/UI.wav");
        frame.dispose();
        Time_Management time_management = new Time_Management();
        time_management.setVisible(true);
        time_management.TMDisplay();
    }
});

```

```

});

JButton Timer = new JButton();//creating instance of JButton
Timer.setIcon(new ImageIcon(getClass().getResource("/Timer.png")));
Timer.setBounds(210, 350, 210, 70);//x axis, y axis, width, height
Timer.setBorder(null);
Timer.setContentAreaFilled(false);
Timer.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        playSound("/UI.wav");
        frame.dispose();
        Timer timer = new Timer();
        timer.Timer();
    }
});

JButton Calendar = new JButton();//creating instance of JButton
Calendar.setIcon(new ImageIcon(getClass().getResource("/Calendar.png")));
Calendar.setBounds(385, 280, 210, 70);//x axis, y axis, width, height
Calendar.setContentAreaFilled(false);
Calendar.setBorder(null);
Calendar.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        playSound("/UI.wav");
        frame.dispose();
        Calendar calendar = new Calendar();
        calendar.setVisible(true);
    }
});

JButton Alarm = new JButton();//creating instance of JButton
Alarm.setIcon(new ImageIcon(getClass().getResource("/Alarm.png")));
Alarm.setBounds(730, 280, 210, 70);//x axis, y axis, width, height
Alarm.setContentAreaFilled(false);
Alarm.setBorder(null);
Alarm.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        playSound("/UI.wav");
        frame.dispose();
        Alarm alarm = new Alarm();
        alarm.Alarm();
    }
});

//adding the frame with setbounds
frame.add(Back);
frame.add(AT);//adding button in JFrame
frame.add(TM);
frame.add(Timer);
frame.add(Calendar);
frame.add(Alarm);
frame.setBounds(360, 230, 900, 600);

frame.setSize(980, 551);

```

```

frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.setLayout(null); //using no layout managers
frame.setVisible(true); //making the frame visible
frame.setResizable(false);

}

// Method to play sound
public static void playSound(String resourcePath) {
    try {
        URL soundURL = Function_Select.class.getResource(resourcePath);
        if (soundURL != null) {
            AudioInputStream audioStream = 
AudioSystem.getAudioInputStream(soundURL);
            Clip clip = AudioSystem.getClip();
            clip.open(audioStream);
            clip.start();
        } else {
            System.err.println("Sound file not found: " + resourcePath);
        }
    } catch (UnsupportedAudioFileException | IOException | LineUnavailableException
e) {
        e.printStackTrace();
    }
}
}

```

// Attendance Checkr Class

```

package Package_TimeCheckr;

import javax.sound.sampled.*;
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.IOException;
import java.net.URL;

public class Attendance {

    private static JLabel presentCountLabel; // Declare JLabel for tally count
    private static JLabel[] nameLabels; // Declare JLabel array for names
    private static JButton[] buttons; // Declare JButton array
    private static String[] defaultNames = {
        "Ayla Abrea", "Lea Agudilla", "Herold Alpuerto", "Angelo Antaran",
        "Bryan Bayona", "Rhyzle Bosque", "John Buloan", "Gerard Cada",
        "Jasffer Catapat", "Harold Cruz", "Kent De Asis", "JM De Luna",
        "Wesley Devilles", "Shanel Dionco", "Ederson Edad", "Ashley Eroa",
        "Alyssa Estrada", "Sean Flores", "Lizette Gonzales", "Andrei Iporac",
        "Chrizette Mendieta", "Ghislain Montalbo", "JC Paulite", "Joyce Penaojas",
        "Arabella Salumbides", "Joseph Solomon", "Ave Sena", "Adrian Venezuela"
    };

    private static int presentCount = 0; // Track present count
}

```

```

public void Attendance() {
    JFrame frame = new JFrame("TimeCheckr");
    JLabel background = new JLabel(new
        ImageIcon(getClass().getResource("/AttendanceBG.png")));
    frame.add(background);
    frame.setContentPane(background);

    JLAYERED_PANE layeredPane = new JLAYERED_PANE();
    layeredPane.setBounds(0, 0, 980, 551);
    frame.add(layeredPane);

    buttons = new JButton[28]; // Array to hold buttons
    nameLabels = new JLabel[28]; // Array to hold labels for names

    // Create buttons and labels for names
    for (int i = 0; i < 28; i++) {
        buttons[i] = new JButton();
        buttons[i].setIcon(new ImageIcon(getClass().getResource("/Absent.png"))); // Default image
        buttons[i].setContentAreaFilled(false);
        buttons[i].setBorder(null);
        buttons[i].setOpaque(false);
        buttons[i].addActionListener(new ButtonClickListener()); // Add action listener to each button
        layeredPane.add(buttons[i], JLAYERED_PANE.DEFAULT_LAYER);

        nameLabels[i] = new JLabel(defaultNames[i]);
        nameLabels[i].setFont(new Font("ZSansSerif", Font.BOLD, 12));
        nameLabels[i].setForeground(Color.darkGray); // Set text color
        nameLabels[i].setHorizontalAlignment(JLabel.CENTER);
        layeredPane.add(nameLabels[i], JLAYERED_PANE.PALETTE_LAYER); // Add the labels to a higher layer
    }

    // Position buttons and labels for names
    positionButtonComponents(buttons);
    positionLabelComponents(nameLabels);

    // Add tally count label
    presentCountLabel = new JLabel("");
    presentCountLabel.setForeground(Color.black); // Set text color
    presentCountLabel.setFont(new Font("Arial", Font.BOLD, 16)); // Set font
    presentCountLabel.setBounds(750, 450, 150, 30); // Set bounds
    layeredPane.add(presentCountLabel, JLAYERED_PANE.PALETTE_LAYER);

    // Add the Back button
    JButton Back = new JButton(); // creating instance of JButton
    Back.setIcon(new ImageIcon(getClass().getResource("/Back.png")));
    Back.setBounds(10, 440, 210, 70); // x axis, y axis, width, height
    Back.setBorder(null);
    Back.setContentAreaFilled(false);
    Back.setOpaque(false);
    Back.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            playSound("/Back.wav");
        }
    });
}

```

```

        frame.dispose();
        new Function_Select();
    }
});

layeredPane.add(Back, JLayeredPane.DEFAULT_LAYER);

// Add the "Clear and Add Names" button
JButton clearAndAddButton = new JButton("Clear and Add Names");
clearAndAddButton.setIcon(new
ImageIcon(getClass().getResource("/Clearandadd.png"))); // Default image
clearAndAddButton.setBounds(282, 90, 120, 50); // x axis, y axis, width, height
clearAndAddButton.setContentAreaFilled(false);
clearAndAddButton.setBorder(null);
clearAndAddButton.setOpaque(false);
clearAndAddButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        clearAndAddNames();
    }
});

layeredPane.add(clearAndAddButton, JLayeredPane.DEFAULT_LAYER);

// Add the "Clear All" button
JButton clearAllButton = new JButton("");
clearAllButton.setIcon(new ImageIcon(getClass().getResource("/ClearAll.png"))); // Default image
clearAllButton.setBounds(170, 90, 120, 50); // x axis, y axis, width, height
clearAllButton.setContentAreaFilled(false);
clearAllButton.setBorder(null);
clearAllButton.setOpaque(false);
clearAllButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        clearAll();
    }
});

layeredPane.add(clearAllButton, JLayeredPane.DEFAULT_LAYER);

// Add the "View Tally" button
JButton viewTallyButton = new JButton("VIEW TALLY");
viewTallyButton.setIcon(new
ImageIcon(getClass().getResource("/Totalnumber.png"))); // Default image
viewTallyButton.setBounds(70, 90, 120, 50); // x axis, y axis, width, height
viewTallyButton.setContentAreaFilled(false);
viewTallyButton.setBorder(null);
viewTallyButton.setOpaque(false);
viewTallyButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        JFrame tallyFrame = new JFrame("Tally");
        Tally tally = new Tally();
        tallyFrame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        tallyFrame.setSize(800, 600);
        tallyFrame.setLayout(null);
    }
});

```

```
tally.initialize(tallyFrame, presentCount); // Pass current present count to Tally
tallyFrame.setVisible(true);
}
});

layeredPane.add(viewTallyButton, JLayeredPane.DEFAULT_LAYER);

frame.setBounds(360, 230, 980, 551);
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.setLayout(null);
frame.setVisible(true);
frame.setResizable(false);
}

// Custom JPanel for displaying a background image
static class BackgroundPanel extends JPanel {
    private Image backgroundImage;

    public BackgroundPanel(Image backgroundImage) {
        this.backgroundImage = backgroundImage;
    }

    @Override
    protected void paintComponent(Graphics g) {
        super.paintComponent(g);
        if (backgroundImage != null) {
            g.drawImage(backgroundImage, 0, 0, getWidth(), getHeight(), this);
        }
    }
}

// ActionListener for buttons
static class ButtonClickListener implements ActionListener {
    @Override
    public void actionPerformed(ActionEvent e) {
        JButton button = (JButton) e.getSource();
        Icon currentIcon = button.getIcon();
        if (currentIcon != null && currentIcon.toString().contains("Absent.png")) {
            button.setIcon(new ImageIcon(getClass().getResource("/Present.png")));
            button.setContentAreaFilled(false); // Set content area filled
            button.setBorder(null); // Set border to null
            button.setOpaque(false); // Set opaque to false
            updatePresentCount(1); // Increment present count
        } else {
            button.setIcon(new ImageIcon(getClass().getResource("/Absent.png")));
            button.setContentAreaFilled(false); // Set content area filled
            button.setBorder(null); // Set border to null
            button.setOpaque(false); // Set opaque to false
            updatePresentCount(-1); // Decrement present count
        }
    }
}

// Position buttons and labels for names
static void positionButtonComponents(JButton[] buttons) {
    int[] xPositions = {66, 66, 66, 66, 66, 66, 284, 284, 284, 284, 284, 284, 284, 284, 721,
    721, 721, 721, 721, 503, 503, 503, 503, 503, 503};
}
```

```

        int[] yPositions = {135, 181, 228, 274, 320, 365, 411, 411, 365, 320, 274, 228, 181,
135, 411, 365, 320, 274, 228, 181, 135, 135, 181, 228, 274, 320, 365, 411};

        for (int i = 0; i < buttons.length; i++) {
            buttons[i].setBounds(xPositions[i], yPositions[i], 187, 46);
        }
    }

static void positionLabelComponents(JLabel[] nameLabels) {
    int[] xPositions = {66, 66, 66, 66, 66, 66, 66, 284, 284, 284, 284, 284, 284, 284, 284, 503,
503, 503, 503, 503, 503, 721, 721, 721, 721, 721, 721, 721, 721};
    int[] yPositions = {105, 151, 198, 244, 290, 335, 381, 105, 151, 198, 244, 290, 335,
381, 105, 151, 198, 244, 290, 335, 381, 105, 151, 198, 244, 290, 335, 381};

    for (int i = 0; i < nameLabels.length; i++) {
        nameLabels[i].setBounds(xPositions[i], yPositions[i], 187, 108); // Position labels
for names above buttons
    }
}

// Update present count
static void updatePresentCount(int increment) {
    presentCount += increment;
    if (presentCount < 0) {
        presentCount = 0;
    }
    presentCountLabel.setText("");
    Tally.updatePresentCount(increment); // Update Tally count if Tally is open
}

// Clear all names and prompt user to enter new names
static void clearAndAddNames() {
    for (int i = 0; i < nameLabels.length; i++) {
        String newName = JOptionPane.showInputDialog("Enter new name for button " +
(i + 1) + ":");

        if (newName != null && !newName.trim().isEmpty()) {
            nameLabels[i].setText(newName);
        } else {
            nameLabels[i].setText(defaultNames[i]); // Reset to default name if no input
        }
    }
}

// Clear all present counts and reset all buttons to Absent
void clearAll() {
    presentCount = 0;
    presentCountLabel.setText("");
    for (JButton button : buttons) {
        button.setIcon(new ImageIcon(getClass().getResource("/Absent.png")));
        button.setContentAreaFilled(false);
        button.setBorder(null);
        button.setOpaque(false);
    }
    Tally.updatePresentCount(-presentCount); // Reset count
}

// Method to play sound

```

```

static void playSound(String soundFile) {
    try {
        URL soundURL = Attendance.class.getResource(soundFile);
        AudioInputStream audioStream = null;
        AudioSystem.getAudioInputStream(soundURL);
        Clip clip = AudioSystem.getClip();
        clip.open(audioStream);
        clip.start();
    } catch (UnsupportedAudioFileException | IOException | LineUnavailableException ex) {
        ex.printStackTrace();
    }
}

```

//Trally Class

```

package Package_TimeCheckr;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.text.SimpleDateFormat;
import java.util.Date;

public class Tally extends JFrame{

    private static JLabel presentCountLabel; // Declare JLabel for tally count
    private static JLabel dateLabel; // Declare JLabel for date

    public void initialize(JFrame frame, int initialCount) {

        JLabel pane = new JLabel(new ImageIcon(getClass().getResource("/TallyBG.png")));
        frame.add(pane);
        // Set the background image
        frame.setContentPane(pane);

        // Set frame properties
        frame.setBounds(360, 230, 980, 551);
        frame.setDefaultCloseOperation(JFrame.HIDE_ON_CLOSE);
        frame.setLayout(null);
        frame.setVisible(true);
        frame.setResizable(false);

        // Initialize the present count label
        presentCountLabel = new JLabel("" + initialCount);
        presentCountLabel.setForeground(Color.darkGray); // Set text color
        presentCountLabel.setFont(new Font("ZSansSerif", Font.BOLD, 65)); // Set font
        presentCountLabel.setHorizontalAlignment(SwingConstants.CENTER); // Center align text
        presentCountLabel.setBounds(488, 212, 150, 100); // Set bounds
        frame.getContentPane().add(presentCountLabel);
    }
}

```

```

// Initialize the date label
dateLabel = new JLabel();
dateLabel.setForeground(Color.darkGray); // Set text color
dateLabel.setFont(new Font("ZSansSerif", Font.BOLD, 18)); // Set font
dateLabel.setHorizontalAlignment(SwingConstants.CENTER); // Center align text
dateLabel.setBounds(290, 75, 400, 50); // Set bounds
frame.getContentPane().add(dateLabel);

// Timer to update the date label every second
javax.swing.Timer timer = new javax.swing.Timer(1000, new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        updateDateLabel();
    }
});
timer.start();

// Update the date label initially
updateDateLabel();

JButton exit = new JButton(); // creating instance of JButton
exit.setIcon(new ImageIcon(getClass().getResource("/Exit_1.png")));
exit.setBounds(10, 440, 210, 70); // x axis, y axis, width, height
exit.setContentAreaFilled(false);
exit.setBorder(null);
exit.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        frame.dispose();
    }
});

frame.add(exit);
}

// Update present count
public static void updatePresentCount(int increment) {
    if (presentCountLabel != null) {
        String text = presentCountLabel.getText();
        int count = Integer.parseInt(text) + increment;
        if (count < 0) {
            count = 0;
        }
        presentCountLabel.setText("" + count);
    }
}

// Update the date label with the current date and time
private static void updateDateLabel() {
    SimpleDateFormat formatter = new SimpleDateFormat("MM-dd-yyyy HH:mm");
    Date now = new Date();
    dateLabel.setText(formatter.format(now));
}

```

```
}
```

//Calendar Class

```
package Package_TimeCheckr;

import javax.swing.*;
import javax.swing.border.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.time.*;
import java.time.format.*;
import java.util.Locale;
import java.io.IOException;
import java.net.URL;
import javax.sound.sampled.*;

public class Calendar extends JFrame {
    private JLabel monthLabel;
    private JPanel calendarPanel;

    private int currentYear;
    private int currentMonth;

    public Calendar() {
        setTitle("Calendar");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(980, 551);
        setLocationRelativeTo(null);
        setResizable(false);

        JPanel contentPane = new JPanel(new BorderLayout());
        contentPane.setOpaque(false);

        // Background image panel
        JPanel backgroundPanel = new JPanel() {
            @Override
            protected void paintComponent(Graphics g) {
                super.paintComponent(g);
                ImageIcon imagelcon = new ImageIcon(getClass().getResource("/CalendarBG2.png"));
                Image image = imagelcon.getImage();
                g.drawImage(image, 0, 0, getWidth(), getHeight(), this);
            }
        };
        backgroundPanel.setLayout(new BorderLayout());

        monthLabel = new JLabel("", JLabel.CENTER);
        monthLabel.setBorder(new EmptyBorder(10, 20, 10, 20));
        monthLabel.setFont(new Font("Arial", Font.BOLD, 20));
        monthLabel.setForeground(Color.BLACK);
```

```

monthLabel.setBounds(300, 10, 380, 30);

// Action listeners for previous and next buttons
JButton prevButton = new JButton("<");
JButton nextButton = new JButton(">");
prevButton.addActionListener(e -> previousMonth());
nextButton.addActionListener(e -> nextMonth());

prevButton.setBounds(250, 20, 50, 30);
nextButton.setBounds(680, 20, 50, 30);

JPanel controlPanel = new JPanel(null);
controlPanel.setOpaque(false);
controlPanel.add(prevButton);
controlPanel.add(monthLabel);
controlPanel.add(nextButton);

controlPanel.setPreferredSize(new Dimension(200, 50));

calendarPanel = new JPanel(new GridLayout(0, 7));
calendarPanel.setBorder(new EmptyBorder(200, 200, 50, 200));
calendarPanel.setOpaque(false);
calendarPanel.setPreferredSize(new Dimension(600, 400));

JButton Back = new JButton();
Back.setIcon(new ImageIcon(getClass().getResource("/Back.png")));
Back.setBounds(10, 440, 210, 70);
Back.setBorder(null);
Back.setContentAreaFilled(false);
Back.setOpaque(false);
contentPane.add(Back);
Back.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        playSound("/Back.wav");
        dispose();
        new Function_Select();
    }
});

LocalDate today = LocalDate.now();
currentYear = today.getYear();
currentMonth = today.getMonthValue();
refreshCalendar(currentYear, currentMonth);

JPanel centerPanel = new JPanel(new BorderLayout());
centerPanel.setOpaque(false);
centerPanel.add(calendarPanel, BorderLayout.CENTER);

backgroundPanel.add(controlPanel, BorderLayout.NORTH);
backgroundPanel.add(centerPanel, BorderLayout.CENTER);

contentPane.add(backgroundPanel, BorderLayout.CENTER);
setContentPane(contentPane);
}

private void refreshCalendar(int year, int month) {
    currentYear = year;
}

```

```

currentMonth = month;

LocalDate date = LocalDate.of(year, month, 1);
Month currentMonth = date.getMonth();
monthLabel.setText(currentMonth.getDisplayName(TextStyle.FULL,
Locale.getDefault()) + " " + year);
calendarPanel.removeAll();

LocalDate firstDayOfMonth = date.withDayOfMonth(1);
int startingDayOfWeek = firstDayOfMonth.getDayOfWeek().getValue();

for (int i = 1; i < startingDayOfWeek; i++) {
    calendarPanel.add(new JLabel(""));
}

int daysInMonth = currentMonth.length(Year.isLeap(year));
for (int day = 1; day <= daysInMonth; day++) {
    JButton dayButton = new JButton(String.valueOf(day));
    dayButton.setFocusPainted(false);

    if (LocalDate.of(year, month, day).equals(LocalDate.now())) {
        dayButton.setForeground(Color.RED);
    } else if ((day + startingDayOfWeek - 1) % 7 == 0) {
        dayButton.setForeground(Color.RED);
    }

    calendarPanel.add(dayButton);
}

revalidate();
repaint();
}

private void previousMonth() {
    if (currentMonth == 1) {
        refreshCalendar(currentYear - 1, 12);
    } else {
        refreshCalendar(currentYear, currentMonth - 1);
    }
}

private void nextMonth() {
    if (currentMonth == 12) {
        refreshCalendar(currentYear + 1, 1);
    } else {
        refreshCalendar(currentYear, currentMonth + 1);
    }
}

static void playSound(String soundFile) {
    try {
        URL soundURL = Calendar.class.getResource(soundFile);
        AudioInputStream audioStream
        = AudioSystem.getAudioInputStream(soundURL);
        Clip clip = AudioSystem.getClip();
        clip.open(audioStream);
        clip.start();
    }
}

```

```

        } catch (UnsupportedAudioFileException | IOException | LineUnavailableException
ex) {
    ex.printStackTrace();
}
}
}

```

//Alarm Class

```

package Package_TimeCheckr;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.text.SimpleDateFormat;
import java.util.*;
import java.util.Timer;
import java.util.TimerTask;
import java.awt.BorderLayout;
import java.awt.Toolkit;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.Date;
import java.util.Iterator;
import java.util.List;
import java.awt.Font;
import javax.swing.border.MatteBorder;
import java.awt.Color;
import javax.sound.sampled.*;
import java.io.File;
import java.io.IOException;
import java.net.URL;

public class Alarm {

    private static final int DEFAULT_SNOOZE_DURATION = 5; // Default snooze duration
    in minutes
    private static final String ALARM_SOUND = "/Alarmsound.wav"; // Path to your alarm
    sound file in resources
    private static int snoozeDuration = DEFAULT_SNOOZE_DURATION;
    private static JLabel timeLabel;
    private static DefaultListModel<String> alarmListModel;
    private static JList<String> alarmList;
    private static Map<String, Timer> alarms = new HashMap<>();
    private static Clip alarmClip;

    public void Alarm() {
        JFrame frame = new JFrame("Alarm");
        frame.setContentPane(new
        ImageIcon(getClass().getResource("/AlarmBG.gif"))));
        frame.setLayout(null); // Set layout manager to null
        // Back button
        JButton back = new JButton();
        back.setIcon(new ImageIcon(getClass().getResource("/Back.png")));

```

```

back.setBounds(10, 440, 210, 70); // Set bounds for the back button
back.setBorder(null);
back.setContentAreaFilled(false);
back.setOpaque(false);
back.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        playSound("/Back.wav");
        frame.dispose();
        new Function_Select();
    }
});
frame.add(back);

int frameWidth = 980;
int frameHeight = 551;
int centerX = frameWidth / 2;
int currentY = 60; // Lower the starting Y position

// Time label
timeLabel = new JLabel("Set Time:");
timeLabel.setFont(new Font("ZSansSerif", Font.BOLD, 48));
timeLabel.setBounds(centerX - 110, currentY, 400, 130); // Centered horizontally,
adjusted vertically
frame.add(timeLabel);
currentY += 100; // Move down for the next component

// Snooze duration label
JLabel snoozeLabel = new JLabel("Snooze (min):");
snoozeLabel.setFont(new Font("ZSansSerif", Font.BOLD, 20));
snoozeLabel.setBounds(centerX - 115, currentY, 150, 30); // Centered horizontally
frame.add(snoozeLabel);

// Snooze duration spinner
JSpinner snoozeSpinner = new JSpinner(new
SpinnerNumberModel(DEFAULT_SNOOZE_DURATION, 1, 60, 1));
snoozeSpinner.setBounds(centerX + 60, currentY, 50, 30); // Positioned next to the
snooze label
snoozeSpinner.addChangeListener(e -> snoozeDuration = (Integer)
snoozeSpinner.getValue());
frame.add(snoozeSpinner);
currentY += 40; // Move down for the next component

// Hours and Minutes inputs
JLabel hoursLabel = new JLabel("Hours:");
hoursLabel.setBounds(centerX - 110, currentY, 100, 30); // Positioned to the left
frame.add(hoursLabel);

JTextField hoursField = new JTextField();
hoursField.setBounds(centerX - 60, currentY, 50, 30); // Positioned next to the hours
label
frame.add(hoursField);

JLabel minutesLabel = new JLabel("Minutes:");
minutesLabel.setBounds(centerX - 5, currentY, 100, 30); // Positioned to the right
frame.add(minutesLabel);

JTextField minutesField = new JTextField();

```

```

minutesField.setBounds(centerX + 55, currentY, 50, 30); // Positioned next to the
minutes label
frame.add(minutesField);
currentY += 40; // Move down for the next component

// Add button
JButton addButton = new JButton("Add");
addButton.setBounds(centerX - 155, currentY, 120, 50); // Positioned to the left
addButton.setIcon(new ImageIcon(getClass().getResource("/Add.png")));
addButton.setContentAreaFilled(false);
addButton.setBorder(null);
addButton.setOpaque(false);
frame.add(addButton);
addButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        try {
            int hours = Integer.parseInt(hoursField.getText());
            int minutes = Integer.parseInt(minutesField.getText());
            addAlarm(hours, minutes);
        } catch (NumberFormatException ex) {
            JOptionPane.showMessageDialog(frame, "Please enter valid numbers for
hours and minutes.");
        }
    }
});

// Remove button
JButton removeButton = new JButton("Remove");
removeButton.setBounds(centerX + 45, currentY, 120, 50); // Positioned to the right
removeButton.setIcon(new ImageIcon(getClass().getResource("/Remove.png")));
removeButton.setContentAreaFilled(false);
removeButton.setBorder(null);
removeButton.setOpaque(false);
frame.add(removeButton);
removeButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        String selectedAlarm = alarmList.getSelectedValue();
        if (selectedAlarm != null) {
            removeAlarm(selectedAlarm);
        }
    }
});
currentY += 0; // Move down for the next component

// Snooze button
JButton snoozeButton = new JButton("Snooze");
snoozeButton.setBounds(centerX - 55, currentY, 120, 50); // Centered horizontally
snoozeButton.setIcon(new ImageIcon(getClass().getResource("/Snooze.png")));
snoozeButton.setContentAreaFilled(false);
snoozeButton.setBorder(null);
snoozeButton.setOpaque(false);
frame.add(snoozeButton);
snoozeButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        snoozeAlarm();
    }
});

```

```

currentY += 60; // Move down for the next component

// Alarm list model and list
alarmListModel = new DefaultListModel<>();
alarmList = new JList<>(alarmListModel);
JScrollPane scrollPane = new JScrollPane(alarmList);
scrollPane.setBounds(centerX - 180, currentY, 360, 150); // Centered horizontally
frame.add(scrollPane);

// Setting frame properties
frame.setBounds(360, 230, frameWidth, frameHeight);
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.setVisible(true);
frame.setResizable(false);

// Start clock timer to update time label
Timer clockTimer = new Timer();
clockTimer.scheduleAtFixedRate(new TimerTask() {
    @Override
    public void run() {
        SwingUtilities.invokeLater(() -> updateClock());
    }
}, 0, 1000);
}

private static void updateClock() {
    SimpleDateFormat dateFormat = new SimpleDateFormat("HH:mm:ss");
    String time = dateFormat.format(new Date());
    timeLabel.setText(time);
}

private static void addAlarm(int hours, int minutes) {
    Calendar calendar = Calendar.getInstance();
    calendar.set(Calendar.HOUR_OF_DAY, hours);
    calendar.set(Calendar.MINUTE, minutes);
    calendar.set(Calendar.SECOND, 0);

    SimpleDateFormat dateFormat = new SimpleDateFormat("HH:mm");
    String alarmTime = dateFormat.format(calendar.getTime());

    if (!alarms.containsKey(alarmTime)) {
        Timer alarmTimer = new Timer();
        alarmTimer.schedule(new TimerTask() {
            @Override
            public void run() {
                Calendar now = Calendar.getInstance();
                if (now.get(Calendar.HOUR_OF_DAY) == calendar.get(Calendar.HOUR_OF_DAY) &&
                    now.get(Calendar.MINUTE) == calendar.get(Calendar.MINUTE)) {
                    SwingUtilities.invokeLater(() -> {
                        playAlarmSound();
                        JOptionPane.showMessageDialog(null, "Alarm!");
                        stopSpecificAlarm(alarmTime);
                    });
                }
            }
        });
    }
}

```

```

        }, calendar.getTime(), 60000); // Schedule the task to run at the specified time and
repeat every minute
        alarms.put(alarmTime, alarmTimer);
        alarmListModel.addElement(alarmTime);
    }
}

private static void removeAlarm(String alarmTime) {
    if (alarms.containsKey(alarmTime)) {
        alarms.get(alarmTime).cancel();
        alarms.remove(alarmTime);
        alarmListModel.removeElement(alarmTime);
    }
}

private static void stopSpecificAlarm(String alarmTime) {
    if (alarms.containsKey(alarmTime)) {
        alarms.get(alarmTime).cancel();
        alarms.remove(alarmTime);
        alarmListModel.removeElement(alarmTime);
    }
    if (alarmClip != null && alarmClip.isRunning()) {
        alarmClip.stop();
    }
}

private static void snoozeAlarm() {
    Calendar calendar = Calendar.getInstance();
    calendar.add(Calendar.MINUTE, snoozeDuration);

    Timer snoozeTimer = new Timer();
    snoozeTimer.schedule(new TimerTask() {
        @Override
        public void run() {
            Calendar now = Calendar.getInstance();
            if (now.get(Calendar.HOUR_OF_DAY) == calendar.get(Calendar.HOUR_OF_DAY) &&
                now.get(Calendar.MINUTE) == calendar.get(Calendar.MINUTE)) {
                SwingUtilities.invokeLater(() -> {
                    playAlarmSound();
                    int result = JOptionPane.showConfirmDialog(null, "Snooze alarm!", "Alarm",
                    JOptionPane.OK_CANCEL_OPTION);
                    if (result == JOptionPane.OK_OPTION) {
                        stopAlarmSound();
                        snoozeTimer.cancel();
                    }
                });
            }
        }
    }, calendar.getTime(), 60000); // Check every minute
}

private static void stopAlarmSound() {
    if (alarmClip != null && alarmClip.isRunning()) {
        alarmClip.stop();
    }
}

```

```

private static void playAlarmSound() {
    playSound(ALARM_SOUND);
}

static void playSound(String soundFile) {
    try {
        URL soundURL = Alarm.class.getResource(soundFile);
        AudioInputStream audioStream
        = AudioSystem.getAudioInputStream(soundURL);
        alarmClip = AudioSystem.getClip();
        alarmClip.open(audioStream);
        alarmClip.start();
    } catch (UnsupportedAudioFileException | IOException | LineUnavailableException
ex) {
    ex.printStackTrace();
}
}
}

```

//Timer Class

```

package Package_TimeCheckr;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.sound.sampled.*;
import java.io.File;
import java.io.IOException;
import java.net.URL;

public class Timer {

    private static int seconds = 0;
    private static int minutes = 0;
    private static int hours = 0;
    Timer timer;

    public void Timer() {
        JFrame frame = new JFrame("TIMER");
        frame.setContentPane(new
        JLabel(new
ImageIcon(getClass().getResource("/TimerBG.gif"))));

        JLabel timerLabel = new JLabel("00:00:00");
        timerLabel.setBounds(385, 170, 210, 70);
        timerLabel.setHorizontalAlignment(SwingConstants.CENTER);
        timerLabel.setFont(new Font("Arial", Font.BOLD, 50));

        javax.swing.Timer timer = new javax.swing.Timer(1000, new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                seconds++;
                if (seconds == 60) {
                    minutes++;
                    seconds = 0;

```

```

        }
        if (minutes == 60) {
            hours++;
            minutes = 0;
        }
        timerLabel.setText(String.format("%02d:%02d:%02d", hours, minutes,
seconds));
    }
});

JButton startButton = new JButton("");
startButton.setIcon(new ImageIcon(getClass().getResource("/Start.png")));
startButton.setBounds(210, 250, 200, 85);
startButton.setContentAreaFilled(false);
startButton.setBorder(null);
startButton.setOpaque(false);
startButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        if (!timer.isRunning()) {
            timer.start();
        }
    }
});

JButton stopButton = new JButton("");
stopButton.setIcon(new ImageIcon(getClass().getResource("/Pause.png")));
stopButton.setBounds(395, 250, 200, 85);
stopButton.setContentAreaFilled(false);
stopButton.setBorder(null);
stopButton.setOpaque(false);
stopButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        timer.stop();
    }
});

JButton resetButton = new JButton("");
resetButton.setIcon(new ImageIcon(getClass().getResource("/Reset.png")));
resetButton.setBounds(570, 250, 200, 85);
resetButton.setContentAreaFilled(false);
resetButton.setBorder(null);
resetButton.setOpaque(false);
resetButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        timer.stop();
        seconds = 0;
        minutes = 0;
        hours = 0;
        timerLabel.setText("00:00:00");
    }
});

JButton back = new JButton();
back.setIcon(new ImageIcon(getClass().getResource("/Back.png")));

```

```

back.setBounds(10, 440, 210, 70);
back.setBorder(null);
back.setContentAreaFilled(false);
back.setOpaque(false);
back.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        playSound("/Back.wav");
        frame.dispose();
        seconds = 0;
        minutes = 0;
        hours = 0;
        timer.stop();
        new Function_Select();
    }
});

frame.add(back);
frame.add(timerLabel);
frame.add(startButton);
frame.add(stopButton);
frame.add(resetButton);

frame.setBounds(360, 230, 900, 600);
frame.setSize(980, 551);
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.setLayout(null);
frame.setVisible(true);
frame.setResizable(false);
}

static void playSound(String soundFile) {
try {
    URL fileURL = Timer.class.getResource(soundFile);
    if (fileURL != null) {
        AudioInputStream audioStream = 
        AudioSystem.getAudioInputStream(fileURL);
        Clip clip = AudioSystem.getClip();
        clip.open(audioStream);
        clip.start();
    } else {
        System.err.println("Sound file not found: " + soundFile);
    }
} catch (UnsupportedAudioFileException | IOException | LineUnavailableException
ex) {
    ex.printStackTrace();
}
}
}

```

//Timer Management Class

```

package Package_TimeCheckr;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;

```

```

import java.awt.event.ActionListener;
import java.io.IOException;
import javax.sound.sampled.*;

public class Time_Management extends JFrame implements ActionListener {
    // Background label for the About screen
    JLabel AboutBG;

    // Method to set up the frame properties
    void frame() {
        JButton Back = new JButton(); // creating instance of JButton
        Back.setIcon(new ImageIcon(getClass().getResource("/Back.png")));
        Back.setBounds(10, 440, 210, 70); // x axis, y axis, width, height
        Back.setBorder(null);
        Back.setContentAreaFilled(false);
        Back.setOpaque(false);
        Back.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                playSound("/Back.wav");
                dispose();
                new Function_Select();
            }
        });
        add(Back);

        // Set default close operation, size, title, and layout
        setBounds(360, 230, 980, 551);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(null); // using no layout managers
        setVisible(true); // making the frame visible
        setResizable(false);
    }

    // Method to display the TM screen
    void TMDisplay() {
        // Set background image for the About screen
        AboutBG = new JLabel(new ImageIcon(getClass().getResource("/Back.png")));
        setContentPane(AboutBG);

        // Display the first About section
        Ab1();
    }

    // Method to display the first About section
    void Ab1() {
        AboutBG = new JLabel(new ImageIcon(getClass().getResource("/TM1.png")));
        setContentPane(AboutBG);

        // Configure and add the Next button
        ImageIcon Next = new ImageIcon(getClass().getResource("/Next.png"));
        JButton nextButton = new JButton(Next);
        nextButton.setCursor(new Cursor(Cursor.HAND_CURSOR));
        nextButton.setOpaque(false);
        nextButton.setBorder(null);
        nextButton.setBorderPainted(false);
        nextButton.setContentAreaFilled(false);
    }
}

```

```

nextButton.setFocusPainted(false);

nextButton.setBounds(525, 420, 100, 100);

add(nextButton);

// Action listener to move to the next section
nextButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        setVisible(false);
        Ab2();
    }
});

frame();
}

// Method to display the second About section
void Ab2() {
    AboutBG = new JLabel(new ImageIcon(getClass().getResource("/TM2.png")));
    setContentPane(AboutBG);

    // Configure and add the Previous and Next buttons
    ImageIcon Prev = new ImageIcon(getClass().getResource("/Previous.png"));
    JButton prevButton = new JButton(Prev);
    prevButton.setCursor(new Cursor(Cursor.HAND_CURSOR));
    prevButton.setOpaque(false);
    prevButton.setBorder(null);
    prevButton.setBorderPainted(false);
    prevButton.setContentAreaFilled(false);
    prevButton.setFocusPainted(false);

    prevButton.setBounds(345, 420, 150, 100);

    ImageIcon Next = new ImageIcon(getClass().getResource("/Next.png"));
    JButton nextButton = new JButton(Next);
    nextButton.setCursor(new Cursor(Cursor.HAND_CURSOR));
    nextButton.setOpaque(false);
    nextButton.setBorder(null);
    nextButton.setBorderPainted(false);
    nextButton.setContentAreaFilled(false);
    nextButton.setFocusPainted(false);

    nextButton.setBounds(525, 420, 100, 100);

    add(prevButton);
    add(nextButton);

    // Action listener to move to the previous section
    prevButton.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            setVisible(false);
            TMDisplay();
        }
    });
}

// Action listener to move to the next section

```

```

nextButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        setVisible(false);
        Ab3();
    }
});

frame();
}

// Method to display the third About section
void Ab3() {
    AboutBG = new JLabel(new ImageIcon(getClass().getResource("/TM3.png")));
    setContentPane(AboutBG);

    // Configure and add the Previous and Next buttons
    ImageIcon Prev = new ImageIcon(getClass().getResource("/Previous.png"));
    JButton prevButton = new JButton(Prev);
    prevButton.setCursor(new Cursor(Cursor.HAND_CURSOR));
    prevButton.setOpaque(false);
    prevButton.setBorder(null);
    prevButton.setBorderPainted(false);
    prevButton.setContentAreaFilled(false);
    prevButton.setFocusPainted(false);

    prevButton.setBounds(345, 420, 150, 100);

    ImageIcon Next = new ImageIcon(getClass().getResource("/Next.png"));
    JButton nextButton = new JButton(Next);
    nextButton.setCursor(new Cursor(Cursor.HAND_CURSOR));
    nextButton.setOpaque(false);
    nextButton.setBorder(null);
    nextButton.setBorderPainted(false);
    nextButton.setContentAreaFilled(false);
    nextButton.setFocusPainted(false);

    nextButton.setBounds(525, 420, 100, 100);

    add(prevButton);
    add(nextButton);

    // Action listener to move to the previous section
    prevButton.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            setVisible(false);
            Ab2();
        }
    });
}

// Action listener to move to the next section
nextButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        setVisible(false);
        Ab4();
    }
});
}

```

```

        frame();
    }

void Ab4() {
    AboutBG = new JLabel(new ImageIcon(getClass().getResource("/TM4.png")));
    setContentPane(AboutBG);

    // Configure and add the Previous and Next buttons
    ImageIcon Prev = new ImageIcon(getClass().getResource("/Previous.png"));
    JButton prevButton = new JButton(Prev);
    prevButton.setCursor(new Cursor(Cursor.HAND_CURSOR));
    prevButton.setOpaque(false);
    prevButton.setBorder(null);
    prevButton.setBorderPainted(false);
    prevButton.setContentAreaFilled(false);
    prevButton.setFocusPainted(false);

    prevButton.setBounds(345, 420, 150, 100);

    ImageIcon Next = new ImageIcon(getClass().getResource("/Next.png"));
    JButton nextButton = new JButton(Next);
    nextButton.setCursor(new Cursor(Cursor.HAND_CURSOR));
    nextButton.setOpaque(false);
    nextButton.setBorder(null);
    nextButton.setBorderPainted(false);
    nextButton.setContentAreaFilled(false);
    nextButton.setFocusPainted(false);

    nextButton.setBounds(525, 420, 100, 100);

    add(prevButton);
    add(nextButton);

    // Action listener to move to the previous section
    prevButton.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            setVisible(false);
            Ab3();
        }
    });
}

// Action listener to move to the next section
nextButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        setVisible(false);
        Ab5();
    }
});

frame();
}

void Ab5() {
    AboutBG = new JLabel(new ImageIcon(getClass().getResource("/TM5.png")));
    setContentPane(AboutBG);

    // Configure and add the Previous and Next buttons
}

```

```

ImageIcon Prev = new ImageIcon(getClass().getResource("/Previous.png"));
JButton prevButton = new JButton(Prev);
prevButton.setCursor(new Cursor(Cursor.HAND_CURSOR));
prevButton.setOpaque(false);
prevButton.setBorder(null);
prevButton.setBorderPainted(false);
prevButton.setContentAreaFilled(false);
prevButton.setFocusPainted(false);

prevButton.setBounds(345, 420, 150, 100);

ImageIcon Next = new ImageIcon(getClass().getResource("/Next.png"));
JButton nextButton = new JButton(Next);
nextButton.setCursor(new Cursor(Cursor.HAND_CURSOR));
nextButton.setOpaque(false);
nextButton.setBorder(null);
nextButton.setBorderPainted(false);
nextButton.setContentAreaFilled(false);
nextButton.setFocusPainted(false);

nextButton.setBounds(525, 420, 100, 100);

add(prevButton);
add(nextButton);

// Action listener to move to the previous section
prevButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        setVisible(false);
        Ab4();
    }
});

// Action listener to move to the next section
nextButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        setVisible(false);
        Ab6();
    }
});

frame();
}

void Ab6() {
    AboutBG = new JLabel(new ImageIcon(getClass().getResource("/TM6.png")));
    setContentPane(AboutBG);

    // Configure and add the Previous and Next buttons
    ImageIcon Prev = new ImageIcon(getClass().getResource("/Previous.png"));
    JButton prevButton = new JButton(Prev);
    prevButton.setCursor(new Cursor(Cursor.HAND_CURSOR));
    prevButton.setOpaque(false);
    prevButton.setBorder(null);
    prevButton.setBorderPainted(false);
    prevButton.setContentAreaFilled(false);
    prevButton.setFocusPainted(false);
}

```

```

prevButton.setBounds(345, 420, 150, 100);

ImageIcon Next = new ImageIcon(getClass().getResource("/Next.png"));
JButton nextButton = new JButton(Next);
nextButton.setCursor(new Cursor(Cursor.HAND_CURSOR));
nextButton.setOpaque(false);
nextButton.setBorder(null);
nextButton.setBorderPainted(false);
nextButton.setContentAreaFilled(false);
nextButton.setFocusPainted(false);

nextButton.setBounds(525, 420, 100, 100);

add(prevButton);
add(nextButton);

// Action listener to move to the previous section
prevButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        setVisible(false);
        Ab5();
    }
});

// Action listener to move to the next section
nextButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        setVisible(false);
        Ab7();
    }
});

frame();
}

// Method to display the fourth About section
void Ab7() {
    AboutBG = new JLabel(new ImageIcon(getClass().getResource("/TM7.png")));
    setContentPane(AboutBG);

    // Configure and add the Previous button
    ImageIcon Prev = new ImageIcon(getClass().getResource("/Previous.png"));
    JButton prevButton = new JButton(Prev);
    prevButton.setCursor(new Cursor(Cursor.HAND_CURSOR));
    prevButton.setOpaque(false);
    prevButton.setBorder(null);
    prevButton.setBorderPainted(false);
    prevButton.setContentAreaFilled(false);
    prevButton.setFocusPainted(false);

    prevButton.setBounds(345, 420, 150, 100);

    add(prevButton);

    // Action listener to move to the previous section
    prevButton.addActionListener(new ActionListener() {

```

```

        public void actionPerformed(ActionEvent e) {
            setVisible(false);
            Ab6();
        }
    });

    frame();
}

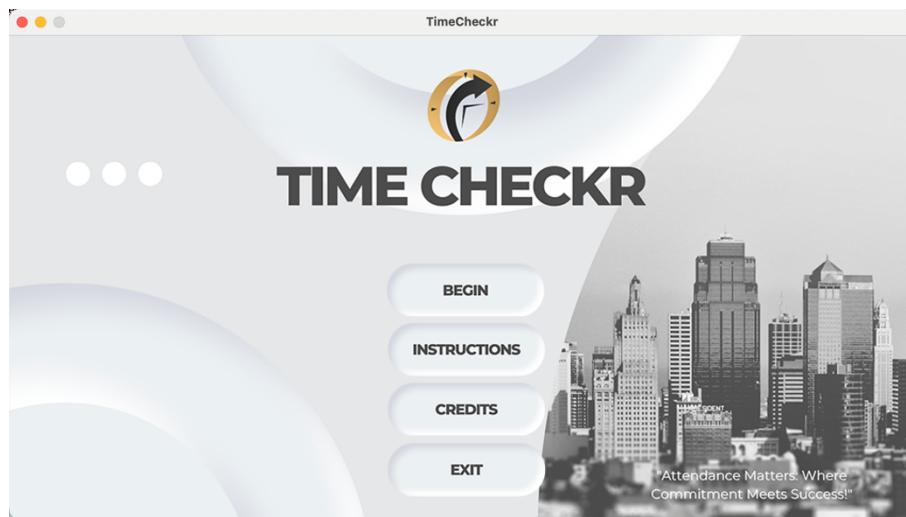
// Method to play sound
public static void playSound(String soundFile) {
    try {
        AudioInputStream audioStream = AudioSystem.getAudioInputStream(
            Time_Management.class.getResource(soundFile)
        );
        Clip clip = AudioSystem.getClip();
        clip.open(audioStream);
        clip.start();
    } catch (UnsupportedAudioFileException | IOException | LineUnavailableException
ex) {
        ex.printStackTrace();
    }
}

@Override
public void actionPerformed(ActionEvent e) {
    // TODO Auto-generated method stub
}
}

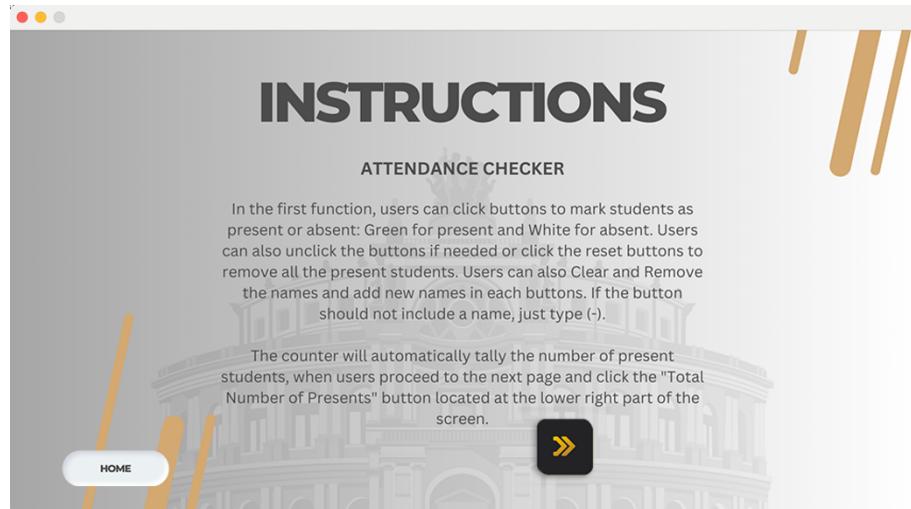
```

SCREENSHOT OF ACTUAL USAGE

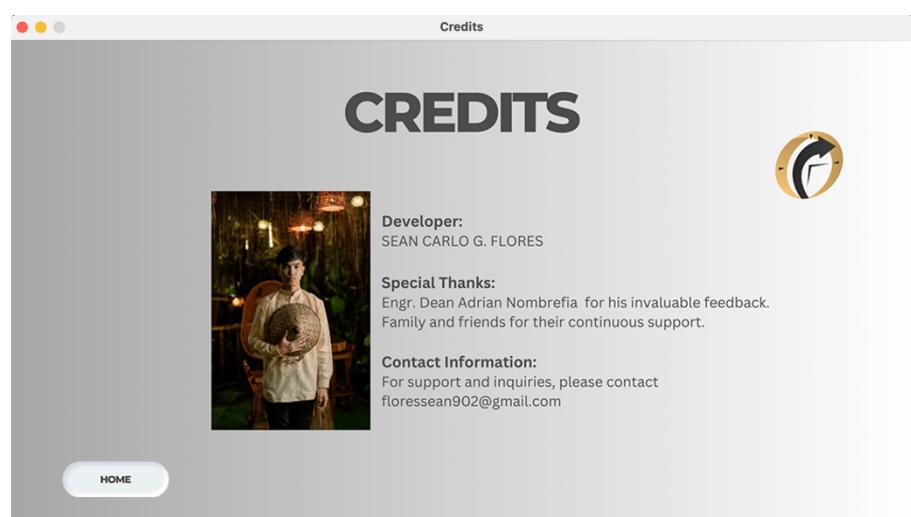
Homepage:



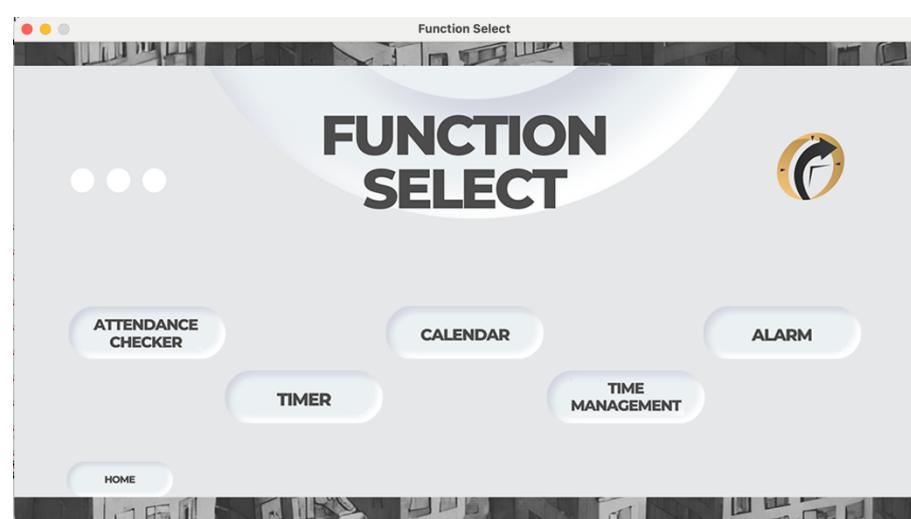
Instructions:



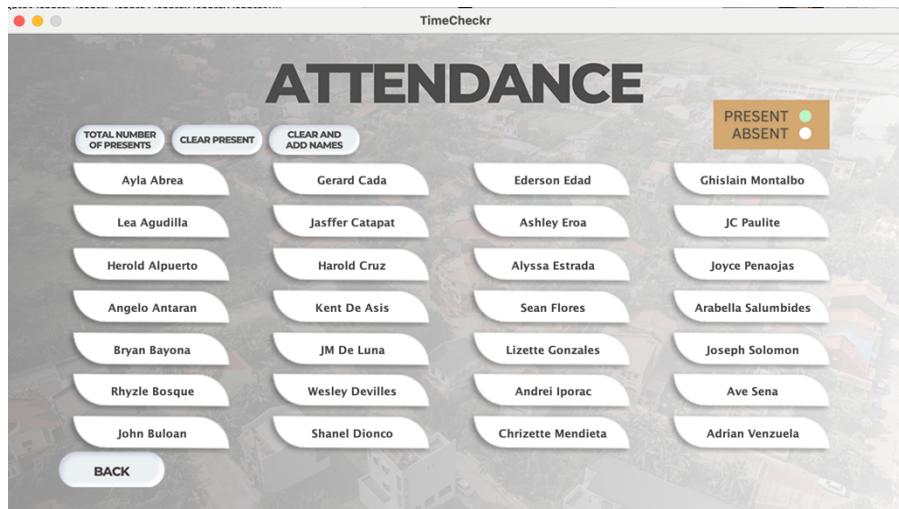
Credits:



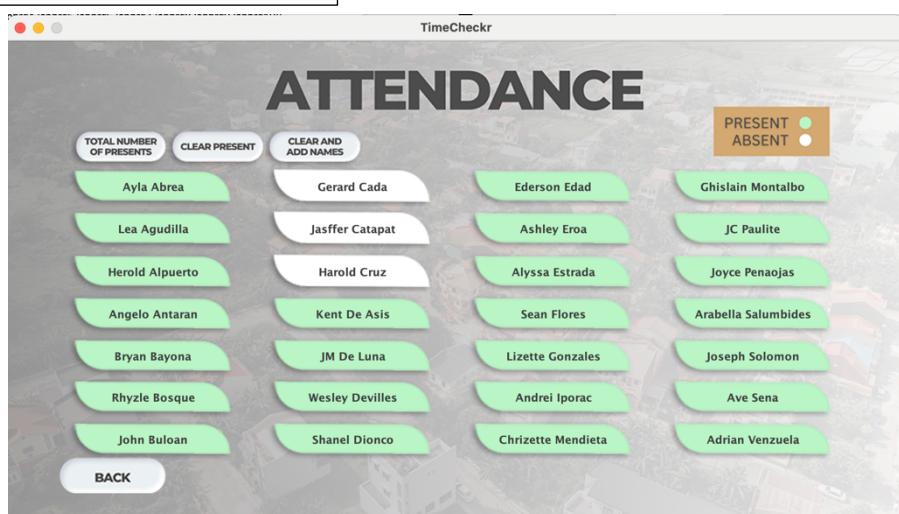
Begin/Function Select:



Feature 1:



Here is an example usage for the first feature.

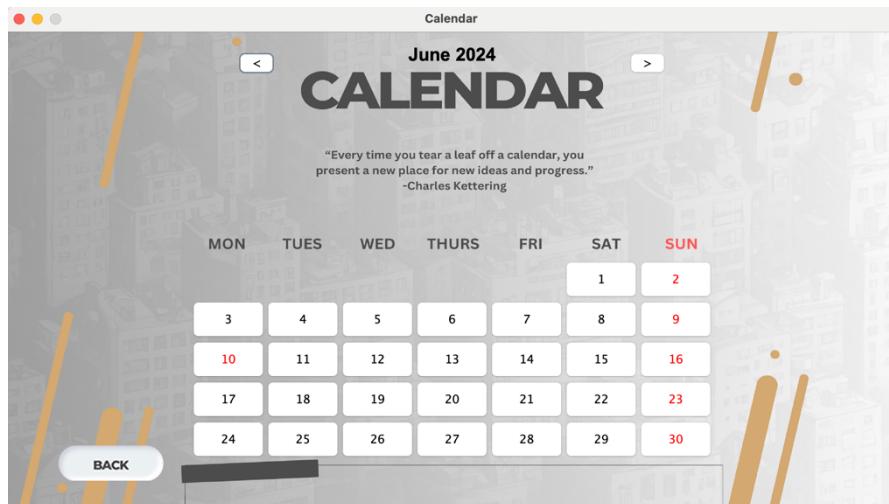


The user can select students and can rename the names and also clear all the selected names of students



This will be the new interface after clicking the "Total Number of Presents"

Feature 2:



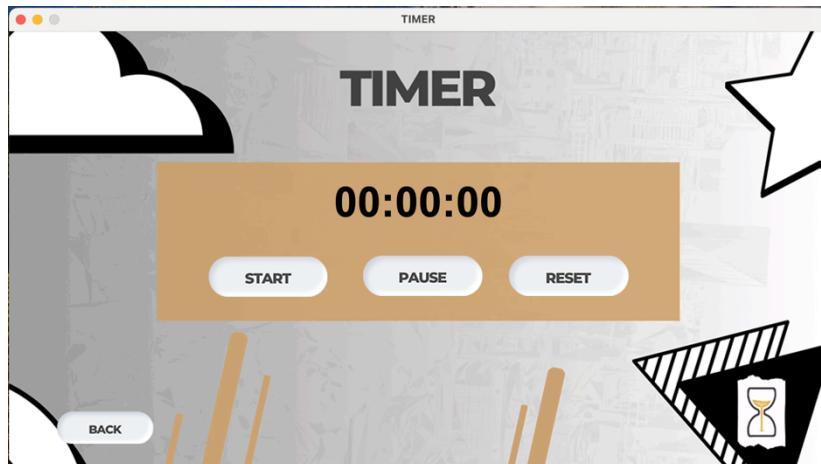
In Calendar, the user can scrub through months and years.

Feature 3:

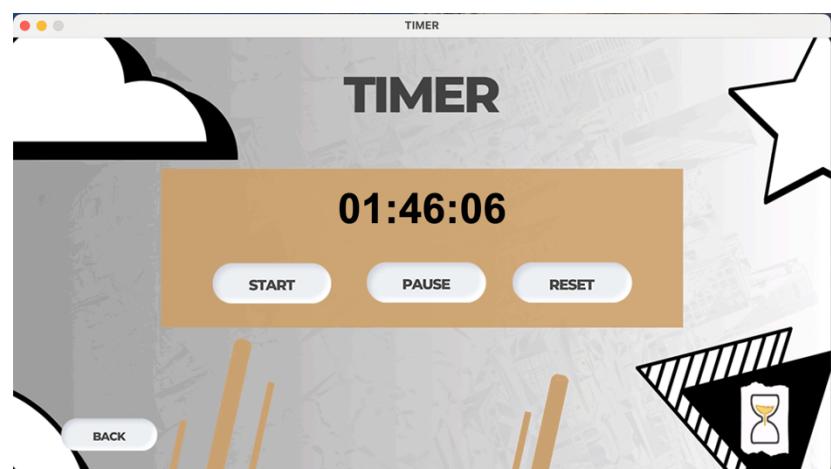


In the Alarm Feature, the user can place multiple alarms and snooze it if desired.

Feature 4:



In Timer, the user can start, pause and reset the timer.



Feature 5:

A screenshot of the Time Management feature for CPELAB A. The title "TIME MANAGEMENT" and "CPELAB A" are at the top. Below is a weekly schedule grid from Monday to Friday, 7:30-10:30, 10:30-1:30, 1:30-4:30, and 4:30-7:30. The grid shows subject names and instructors. A "NEXT" button is at the bottom right. A "BACK" button is in the bottom left. A callout box on the right says: "Here, the user can check the time of their subjects."

	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY
7:30-10:30	CPE05 CPEII GE-A M.D. Castro	CPE05 CPEII GE-B M.D Castro	CPE05 CPEII GE-A M.D. Castro	CPE05 CPEII GE-B M.D. Castro	CPE30 CPE IV IF-A J.S GILI
10:30-1:30	CPE05 CPEII GE-B M.D. Castro	CPE05 CPEII GE-B M.D Castro	CPE05 CPEII GE-A M.D Castro	CPE05 CPEII GE-B M.D. Castro	CPE17 LAB CPEIII IE CM Ceribo
1:30-4:30	CPE30 CPE IV GF-A J.S GILI	CPE05 CPEIII IF DAL NOMBREFIA	CPE30 CPE III GF J.S GILI	CPE05 CPEIII GF DAL NOMBREFIA	CPE17 LAB CPEIII IF CM Ceribo
4:30-7:30		CPE05 CPEIII GE DAL NOMBREFIA		CPE05 CPEIII IE DAL NOMBREFIA	

Here, the user can check the time of their subjects.

A screenshot of the Time Management feature for CPELAB B. The title "TIME MANAGEMENT" and "CPELAB B" are at the top. Below is a weekly schedule grid from Monday to Friday, 7:30-10:30, 10:30-1:30, 1:30-4:30, and 4:30-7:30. The grid shows subject names and instructors. A "NEXT" button is at the bottom right. A "BACK" button is in the bottom left. A "PREVIOUS" button is also present.

	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY
7:30-10:30	CPE30 CPE IV GE -A J.S Gili	CPE23 CPEIV GF DAL Nombrefia			ECE01L ENGR SEVILLA
10:30-1:30	CPE30 CPE IV GE -A J.S Gili	CPE23 CPEIV GE DAL Nombrefia	CPE17L CPEIII GE CM Ceribo	ECE01L ENGR SEVILLA	CPE16L CPEIII GF JV Obmerga
1:30-4:30	CPE19L CPEIII IF JV Manzанero	CPE17L CPEIII GF CM Ceribo	CPE19L CPEIII GF JV Manzанero	CPE16L CPEIII IF JV Obmerga	CPE16L CPEIII IE JV Obmerga
4:30-7:30			CPE16L CPEIII IE JV Obmerga		CPE16L CPEIII GE JV Obmerga