UNIVERSITY OF CALOOCAN CITY
COLLEGE OF ENGINEERING
Phase 8A Package 11, Block 199, German Village, barangay 176 Bagong Silang, Caloocan City, 1428

Progress Report No. 4	
Course Code: CPE201L	**Program: BSCPE**
Course Title: Data Structure and Algorithm	**Date Performed: October 21, 2025**
Section: 2A	**Date Submitted: October 21, 2025**
Name: **Laput, Mark Danielle E.** **Sumel, Hendrix Nathan L.**	**Instructor:** Engr. Maria Rizette H. Sayo

1.Objectives

Create a cross-platform task scheduler with:

- Intuitive task management (create, edit, complete, snooze)
- Customizable alarms with notifications, vibration, and sounds
- Local SQLite data storage
- Swipe gestures and undo functionality
- Recurring tasks and history tracking
- Android compatibility via plyer

2. Discussion

This Universal Task Scheduler combines **Kivy's flexible UI framework** with **SQLite's reliable local storage** to create a robust cross-platform application. The architecture leverages **plyer for Android-compatible notifications and vibration** while maintaining desktop functionality. Key implementations include:

- **Smart Alarm System**: Scheduled background checks trigger multi-sensory alerts without blocking the UI
- **Threaded Sound Management**: Custom audio playback runs separately to maintain interface responsiveness
- **Intuitive Gesture Controls**: Swipe recognition enables quick task completion and snoozing
- **Animated Settings Overlay**: Smooth sliding panels provide modern configuration experience
- **Dual File Chooser Strategy**: Fallback system ensures sound file selection works across all platforms

The application successfully balances sophisticated functionality with clean, maintainable code structure in a single-file implementation.

3. Materials and Equipment

1. **Hardware:**
   - Computer - windows
   - Android mobile device (for testing mobile features)
2. **Software & Platforms:**
   - Python - pycharm
   - Kivy framework (UI development)
   - SQLite (database)
   - Plyer library (cross-platform notifications/vibration)
   - Text editor/IDE (VS Code, PyCharm, etc.)
   - File system (for sound storage)
3. **Libraries & Components:**
   - Kivy (core, clock, properties, widgets)

- SQLite3 (database operations)
- Plyer (notification, vibrator, filechooser)
- Python standard library (os, threading, datetime)

4. **Storage:**
   - Local SQLite database (reminders.db)
   - Local sounds directory for custom audio files

4. Procedure

Setup and Execution

1. **Prerequisites:**
   - Ensure **Python 3** is installed on your system.
   - Install the required Python packages:

bash
pip install kivy[full] plyer

2. **Running the Application:**
   - Save the provided code into a single file, for example, task_scheduler.py.
   - Run the application from your terminal or command prompt:

bash
python task_scheduler.py

   - On Android, this would require packaging the app with **Buildozer**, which involves creating a buildozer.spec file and building the APK.

Usage Guide

1. **Adding a New Task:**
   - Click the "+ Add Task" button in the header.
   - Fill in the task title.
   - Use the year, month, day, hour, and minute spinners to set the due date and time.
   - Select priority, alarm type, and repeat mode.
   - (Optional) Upload a custom sound for this task using the "Upload Sound" button and preview it with "Preview".
   - Click "Save" to create the task.
2. **Managing Active Tasks:**
   - **View:** All active tasks are listed in the main "Active Tasks" panel.
   - **Swipe Gestures:**
     - **Swipe Right** on a task to mark it as "Done".
     - **Swipe Left** on a task to "Snooze" it for the default number of minutes.
   - **Button Actions:**
     - **Edit:** Opens the task in an edit popup.
     - **Done:** Manually complete the task.
     - **Snooze:** Manually snooze the task.
     - **Cancel:** Permanently cancel the task (with confirmation).
3. **Using the History Panel:**
   - Completed tasks appear in the "History" panel on the right.
   - Click "Do Again" on any history item to create a new, editable copy of that task.
4. **Configuring Settings:**
   - Click the " ⚙ " (gear) icon in the header to slide in the Settings Overlay.
   - Here you can:

- Toggle between Light and Dark themes.
- Set the default snooze duration.
- Set the default alarm type for new tasks.
- Upload, preview, or delete global default sounds.
- Reset all task sounds to the system default.

5. **Handling Alarms:**
   - When a task is due, a popup will appear with "Snooze" and "Done" buttons.
   - The system will also trigger a notification, vibration (if enabled), and play the assigned sound.

5. Output

Screenshot of your outputs based on the procedures.

6. Conclusion

The Universal Task Scheduler project successfully demonstrates the development of a full-featured, practical desktop and mobile application using Python and the Kivy framework. The application meets its core objectives by providing a powerful yet intuitive system for task management, reminder scheduling, and history tracking.

Key successes include the effective integration of a responsive GUI with a persistent SQLite database, the implementation of a non-blocking alarm system with custom sounds, and the creation of a polished user experience with gestures and an undo feature. The use of plyer abstracted platform-specific APIs, making the application a strong candidate for deployment on Android.

This project serves as an excellent foundation. Future work could expand its capabilities by adding features like task categories/filtering, more sophisticated time pickers, web or cloud synchronization for cross-device access, and integration with calendar APIs. Overall, the Universal Task Scheduler is a robust, functional, and well-architected application that effectively solves the problem of personal task and reminder management.

UNIVERSITY OF CALOOCAN CITY
COLLEGE OF ENGINEERING
Phase 8A Package 11, Block 199, German Village, barangay 176 Bagong Silang, Caloocan City, 1428

Lab Activity Rubric

Criteria	Ratings						Pts
SO 7 PI 1 **Student Outcome 7.1** Acquire and apply new knowledge from outside sources. threshold: 4.8 pts	6 pts Excellent | Educational interests and pursuits exist and flourish outside classroom requirements,knowledge and/or experiences are pursued independently and applies knowledge learned into practice	5 pts Good | Educational interests and pursuits exist and flourish outside classroom requirements,knowledge and/or experiences are pursued independently	4 pts Satisfactory | Look beyond classroom requirements, showing interest in pursuing knowledge independently	3 pts Unsatisfactory | Begins to look beyond classroom requirements, showing interest in pursuing knowledge independently	2 pts Poor | Relies on classroom instruction only	1 pts Very Poor | No initiative or interest in acquiring new knowledge	6 pts
SO 7 PI 2 **Student Outcome 7.2** Learn independently threshold: 4.8 pts	6 pts Excellent | Completes an assigned task independently and practices continuous improvement	5 pts Good | Completes an assigned task without supervision or guidance	4 pts Satisfactory | Requires minimal guidance to complete an assigned task	3 pts Unsatisfactory | Requires detailed or step-by-step instructions to complete a task	2 pts Poor | Shows little interest to complete a task independently	1 pts Very Poor | No interest to complete a task independently	6 pts
SO 7 PI 3 **Student Outcome 7.3** Critical thinking in the broadest context of technological change threshold: 4.8 pts	6 pts Excellent | Synthesizes and integrates information from a variety of sources; formulates a clear and precise perspective; draws appropriate conclusions	5 pts Good | Evaluate information from a variety of sources; formulates a clear and precise perspective.	4 pts Satisfactory | Analyze information from a variety of sources; formulates a clear and precise perspective.	3 pts Unsatisfactory | Apply the gathered information to formulate the problem	2 pts Poor | Gather and summarized the information from a variety of sources but failed to formulate the problem	1 pts Very Poor | Gather information from a variety of sources	6 pts
SO 7 PI 4 **Student Outcome 7.4** Creativity and adaptability to new and emerging technologies threshold: 4.8 pts	6 pts Excellent | Ideas are combined in original and creative ways in line with the new and emerging technology trends to solve a problem or address an issue.	5 pts Good | Ideas are creative and adapt the new knowledge to solve a problem or address an issue	4 pts Satisfactory | Ideas are creative in solving a problem, or address an issue	3 pts Unsatisfactory | Shows some creative ways to solve the problem	2 pts Poor | Shows initiative and attempt to develop creative ideas to solve the problem	1 pts Very Poor | Ideas are copied or restated from the sources consulted	6 pts
							Total Points: 24