In [ ]:

```python
import os
import pickle
from datetime import datetime

# Define the task class
class Task:
    def __init__(self, title, priority, due_date=None, completed=False):
        self.title = title
        self.priority = priority
        self.due_date = due_date
        self.completed = completed

# Define the TaskList class to manage tasks
class TaskList:
    def __init__(self, filename):
        self.filename = filename
        self.tasks = []
        self.load_tasks()

    def load_tasks(self):
        if os.path.exists(self.filename):
            with open(self.filename, 'rb') as file:
                self.tasks = pickle.load(file)

    def save_tasks(self):
        with open(self.filename, 'wb') as file:
            pickle.dump(self.tasks, file)

    def add_task(self, task):
        self.tasks.append(task)
        self.save_tasks()

    def remove_task(self, task_index):
        if 0 <= task_index < len(self.tasks):
            del self.tasks[task_index]
            self.save_tasks()
        else:
            print("Invalid task index.")

    def mark_task_completed(self, task_index):
        if 0 <= task_index < len(self.tasks):
            self.tasks[task_index].completed = True
            self.save_tasks()
        else:
            print("Invalid task index.")

    def list_tasks(self):
        if not self.tasks:
            print("No tasks.")
            return

        for i, task in enumerate(self.tasks):
            status = "Completed" if task.completed else "Incomplete"
            due_date = task.due_date.strftime("%Y-%m-%d") if task.due_date else "N/A"
            print(f"{i + 1}. Title: {task.title}")
            print(f"   Priority: {task.priority}")
            print(f"   Due Date: {due_date}")
            print(f"   Status: {status}")
            print()

def main():
    filename = "tasks.pkl"
    task_list = TaskList(filename)

    while True:
        print("\nOptions:")
        print("1. Add Task")
        print("2. Remove Task")
        print("3. Mark Task as Completed")
        print("4. List Tasks")
        print("5. Quit")

        choice = input("Enter your choice: ")

        if choice == "1":
            title = input("Enter task title: ")
            priority = input("Enter task priority (high/medium/low): ")
            due_date_str = input("Enter due date (YYYY-MM-DD, leave empty if none): ")

            try:
                due_date = datetime.strptime(due_date_str, "%Y-%m-%d") if due_date_str else None
            except ValueError:
                print("Invalid date format. Task not added.")
                continue

            task = Task(title, priority, due_date)
            task_list.add_task(task)
            print("Task added successfully.")

        elif choice == "2":
            task_list.list_tasks()
            task_index = int(input("Enter the index of the task to remove: ")) - 1
            task_list.remove_task(task_index)
            print("Task removed successfully.")

        elif choice == "3":
            task_list.list_tasks()
            task_index = int(input("Enter the index of the task to mark as completed: ")) - 1
            task_list.mark_task_completed(task_index)
            print("Task marked as completed.")

        elif choice == "4":
            task_list.list_tasks()

        elif choice == "5":
            break

        else:
            print("Invalid choice. Please try again.")

if __name__ == "__main__":
    main()
```

```
Options:
1. Add Task
2. Remove Task
3. Mark Task as Completed
4. List Tasks
5. Quit
Enter your choice: 4
1. Title: Call Mom
   Priority: low
   Due Date: N/A
   Status: Incomplete


Options:
1. Add Task
2. Remove Task
3. Mark Task as Completed
4. List Tasks
5. Quit
Enter your choice: high
Invalid choice. Please try again.

Options:
1. Add Task
2. Remove Task
3. Mark Task as Completed
4. List Tasks
5. Quit
Enter your choice: 1
Enter task title: Get Groceries
Enter task priority (high/medium/low): high
Enter due date (YYYY-MM-DD, leave empty if none): 2023-09-15
Task added successfully.

Options:
1. Add Task
2. Remove Task
3. Mark Task as Completed
4. List Tasks
5. Quit
Enter your choice: 3
1. Title: Call Mom
   Priority: low
   Due Date: N/A
   Status: Incomplete

2. Title: Get Groceries
   Priority: high
   Due Date: 2023-09-15
   Status: Incomplete

Enter the index of the task to mark as completed: 2
Task marked as completed.

Options:
1. Add Task
2. Remove Task
3. Mark Task as Completed
4. List Tasks
5. Quit
Enter your choice: 1
Enter task title: Get Dentist Appointment
Enter task priority (high/medium/low): medium
Enter due date (YYYY-MM-DD, leave empty if none): 2023-09-15
Task added successfully.

Options:
1. Add Task
2. Remove Task
3. Mark Task as Completed
4. List Tasks
5. Quit
Enter your choice: 3
1. Title: Call Mom
   Priority: low
   Due Date: N/A
   Status: Incomplete

2. Title: Get Groceries
   Priority: high
   Due Date: 2023-09-15
   Status: Completed

3. Title: Get Dentist Appointment
   Priority: medium
   Due Date: 2023-09-15
   Status: Incomplete

Enter the index of the task to mark as completed: 3
Task marked as completed.

Options:
1. Add Task
2. Remove Task
3. Mark Task as Completed
4. List Tasks
5. Quit
Enter your choice: 1
Enter task title: Pay Electricity Bill
Enter task priority (high/medium/low): high
Enter due date (YYYY-MM-DD, leave empty if none): 2023-09-22
Task added successfully.

Options:
1. Add Task
2. Remove Task
3. Mark Task as Completed
4. List Tasks
5. Quit
Enter your choice: 4
1. Title: Call Mom
   Priority: low
   Due Date: N/A
   Status: Incomplete

2. Title: Get Groceries
   Priority: high
   Due Date: 2023-09-15
   Status: Completed

3. Title: Get Dentist Appointment
   Priority: medium
   Due Date: 2023-09-15
   Status: Completed

4. Title: Pay Electricity Bill
   Priority: high
   Due Date: 2023-09-22
   Status: Incomplete


Options:
1. Add Task
2. Remove Task
3. Mark Task as Completed
4. List Tasks
5. Quit
```

In [ ]: