```
import json
```

```
import os


TASKS_FILE = 'tasks.json'


def load_tasks():
    if os.path.exists(TASKS_FILE):
        with open(TASKS_FILE, 'r') as file:
            return json.load(file)
    return []


def save_tasks(tasks):
    with open(TASKS_FILE, 'w') as file:
        json.dump(tasks, file)


def display_menu():
    print("\nTo-Do List Application")
    print("1. View tasks")
    print("2. Add task")
    print("3. Mark task as done")
    print("4. Mark task as undone")
    print("5. Remove task")
    print("6. Save and Exit")


def view_tasks(tasks):
    if not tasks:
        print("\nNo tasks available.")
    else:
        print("\nTasks:")
        for i, task in enumerate(tasks):
            status = "Done" if task['completed'] else "Not Done"
            print(f"{i + 1}. {task['title']} - {status}")


def add_task(tasks):
    title = input("Enter task title: ").strip()
    if title:
        tasks.append({'title': title, 'completed': False})
        print("Task added.")
    else:
        print("Task title cannot be empty.")


def mark_task(tasks, completed=True):
    if not tasks:
        print("\nNo tasks available.")
        return
    try:
        task_num = int(input("Enter task number to mark: ")) - 1
        if 0 <= task_num < len(tasks):
            tasks[task_num]['completed'] = completed
            status = "done" if completed else "undone"
            print(f"Task marked as {status}.")
        else:
            print("Invalid task number.")
    except ValueError:
        print("Invalid input. Please enter a number.")


def remove_task(tasks):
    if not tasks:
        print("\nNo tasks available.")
        return
    try:
        task_num = int(input("Enter task number to remove: ")) - 1
        if 0 <= task_num < len(tasks):
            removed_task = tasks.pop(task_num)
            print(f"Task '{removed_task['title']}' removed.")
        else:
            print("Invalid task number.")
    except ValueError:
        print("Invalid input. Please enter a number.")
```

```
def remove_task(tasks):
    if not tasks:
        print("\nNo tasks available.")
        return
    try:
        task_num = int(input("Enter task number to remove: ")) - 1
        if 0 <= task_num < len(tasks):
            removed_task = tasks.pop(task_num)
            print(f"Task '{removed_task['title']}' removed.")
        else:
            print("Invalid task number.")
    except ValueError:
        print("Invalid input. Please enter a number.")


def main():
    tasks = load_tasks()

    while True:
        display_menu()
        choice = input("Choose an option: ").strip()

        if choice == '1':
            view_tasks(tasks)
        elif choice == '2':
            add_task(tasks)
        elif choice == '3':
            mark_task(tasks, completed=True)
        elif choice == '4':
            mark_task(tasks, completed=False)
        elif choice == '5':
            remove_task(tasks)
        elif choice == '6':
            save_tasks(tasks)
            print("Tasks saved. Exiting application.")
            break
        else:
            print("Invalid choice. Please choose a valid option.")


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

```
To-Do List Application
1. View tasks
2. Add task
3. Mark task as done
4. Mark task as undone
5. Remove task
6. Save and Exit
Choose an option: 1

No tasks available.

To-Do List Application
1. View tasks
2. Add task
3. Mark task as done
4. Mark task as undone
5. Remove task
6. Save and Exit
Choose an option: 2
Enter task title: typing
Task added.

To-Do List Application
1. View tasks
2. Add task
3. Mark task as done
4. Mark task as undone
5. Remove task
6. Save and Exit
Choose an option: 3
Enter task number to mark: 1
Task marked as done.

To-Do List Application
1. View tasks
2. Add task
3. Mark task as done
4. Mark task as undone
5. Remove task
6. Save and Exit
Choose an option: 5
Enter task number to remove: 2
Invalid task number.
```

```
To-Do List Application
1. View tasks
2. Add task
3. Mark task as done
4. Mark task as undone
5. Remove task
6. Save and Exit
Choose an option: 1

Tasks:
1. typing - Done

To-Do List Application
```