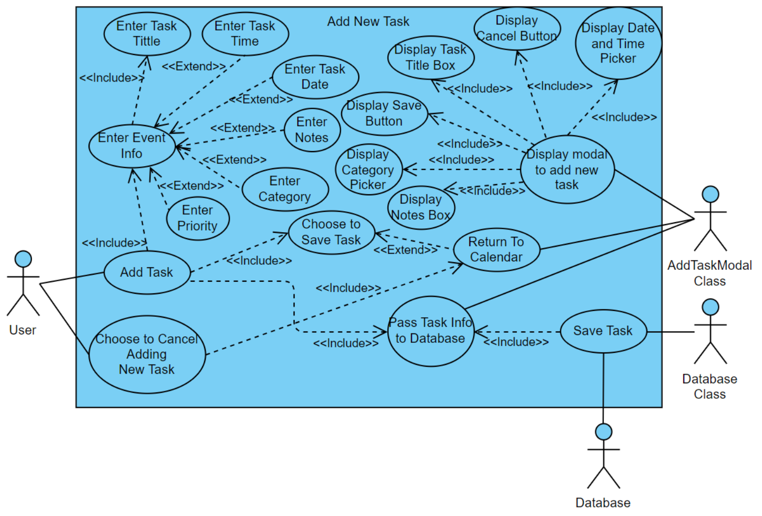
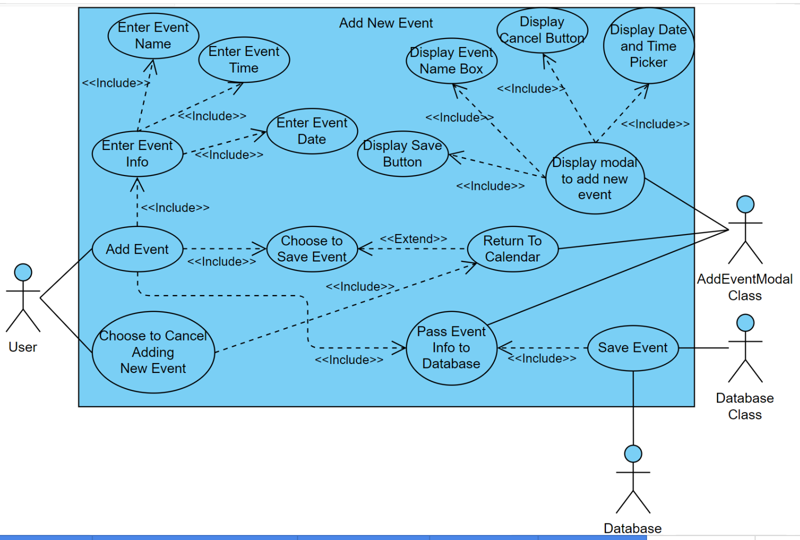
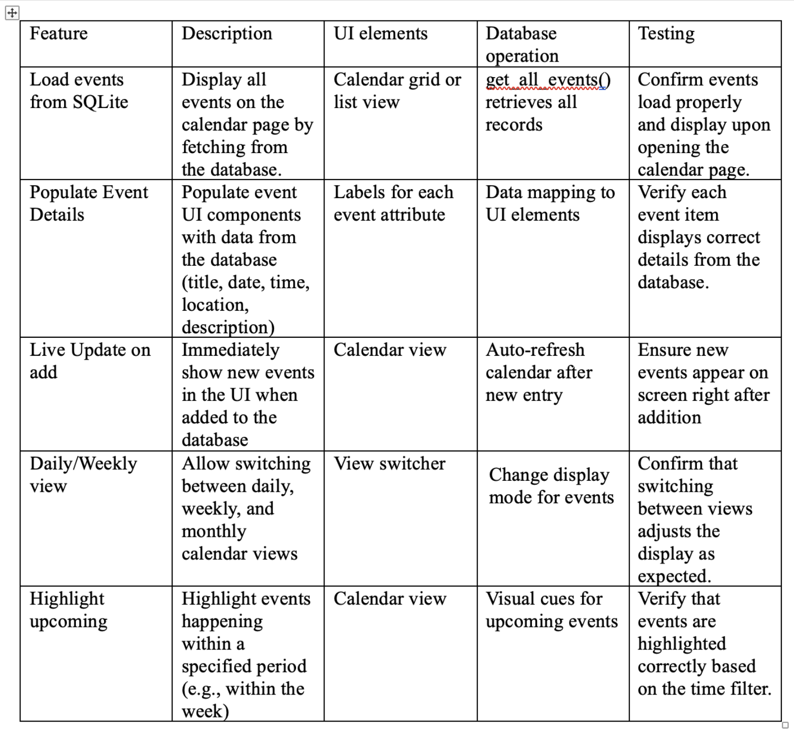
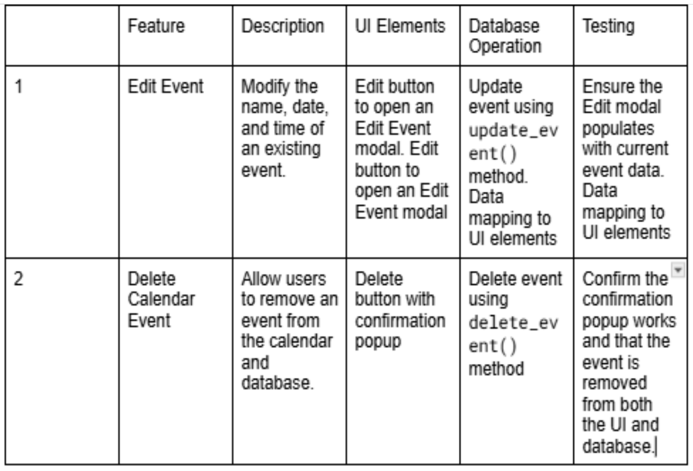
	Requirement:	Requirement Document:	
1	Integrate Add Task with the SQLite for storage of New Tasks		
2	Integrate Add Events with the SQLite for storage of New Events		
3	Integrate so that the To-do list page pulls data using SQLite and populates the page	(see table below)	

Feature	Description	UI Elements	Database Operation	Testing
Load Tasks from SQLite	Display all tasks on the To-do list page by fetching from the database.	Task list area	get_all_tasks() retrieves all records	Confirm tasks load properly and display upon opening the To-do list page.
Populate Task Details	Populate task UI components with data from the database (title, due date, priority, category).	Labels for each task attribute	Data mapping to UI elements	Verify each task item displays correct details from the database.
Live Update on Add	Immediately show new tasks in the UI when added to the database.	Task list view	Auto-refresh task list after new entry	Ensure new tasks appear on screen right after addition.

	Requirement:	Requirement Document:
4	Integrate so that the Calendar view pulls data using SQLite and populates the page	
5	Implement the ability to edit and delete calendar events	
6	Implement the ability to edit, delete, and mark as completed to do list items	

Feature	Description	UI elements	Database operation	Testing
Load events from SQLite	Display all events on the calendar page by fetching from the database.	Calendar grid or list view	get_all_events() retrieves all records	Confirm events load properly and display upon opening the calendar page.
Populate Event Details	Populate event UI components with data from the database (title, date, time, location, description)	Labels for each event attribute	Data mapping to UI elements	Verify each event item displays correct details from the database.
Live Update on add	Immediately show new events in the UI when added to the database	Calendar view	Auto-refresh calendar after new entry	Ensure new events appear on screen right after addition
Daily/Weekly view	Allow switching between daily, weekly, and monthly calendar views	View switcher	Change display mode for events	Confirm that switching between views adjusts the display as expected.
Highlight upcoming	Highlight events happening within a specified period (e.g., within the week)	Calendar view	Visual cues for upcoming events	Verify that events are highlighted correctly based on the time filter.

	Feature	Description	UI Elements	Database Operation	Testing
1	Edit Event	Modify the name, date, and time of an existing event.	Edit button to open an Edit Event modal. Edit button to open an Edit Event modal	Update event using `update_event()` method. Data mapping to UI elements	Ensure the Edit modal populates with current event data. Data mapping to UI elements
2	Delete Calendar Event	Allow users to remove an event from the calendar and database.	Delete button with confirmation popup	Delete event using `delete_event()` method	Confirm the confirmation popup works and that the event is removed from both the UI and database.

Feature	Description	UI Elements	Database Operation	Testing
Edit Task	Modify the title, due date, category, and priority of an existing task.	Edit button to open an Edit Task Modal.	Update task using update_task() method.	Ensure modal populates with current task data.
Delete Task	Remove a task from the list and the database.	Delete button with confirmation popup.	Delete task using delete_task() method.	Verify confirmation popup works and task is removed.
Mark Task as Completed	Toggle the completion status of a task, updating the UI and database.	Checkbox or toggle to mark task as completed.	Toggle completion status using update_task() method.	Check that the completion status is updated correctly in both UI and database.