

Phase 2 Source – Group 3
University of Maryland Global Campus
Oct 02, 2024
CMSC 495 Capstone

Team Members:
Cook, Sean Coutros, Peter
Daley, Tasciana
Geisler, Nicholas
Holman, Trevor
Jefferson, Michael
Malixi, Matthew Arvhie
Mussa, Nezifa
Suarez, Karolyn

Introduction

The Calendar/Task Manager project is a Java-based application that enables users to manage tasks and events within an interactive calendar. This Phase II documentation outlines the key developments and refinements made to enhance the project. This phase focuses on improving the design, adding core functionalities, refactoring the code, and finalizing the deployment setup.

Objective

The goal of Phase II is to refine the software through:

1. **Refactoring and Finalizing the Project Setup:** Improving code structure for readability and maintainability.
2. **Enhancing Core Functionality:** Implemented all functional requirements that were omitted from Phase I.
3. **Design Improvements:** Enhancing the user interface and user experience, making the application visually appealing and easy to navigate.
4. **Test Cases:** Verifying the functionality of the application to ensure the system performs as expected based on the project requirements.
5. **Final Deployment:** Preparing the application for release, ensuring it runs smoothly in the production environment.


1. Refactor the Software

During Phase II, we focused on improving the internal structure of the code in addition to implementing the remainder of the core functionalities. This involved:

- **HTML Improvements:** Added CSS styling as well as JavaScript to appropriate HTML files to improve user experience and promote dynamic interaction with the webpages.
- **Improved Code Readability:** Code was refactored for readability by standardizing function names, adding comments, and organizing files within the project to follow best practices.
- **CSS and JavaScript Dependencies:** Addressed issues encountered in Phase I where CSS and JavaScript libraries were not being recognized by the Maven setup. We however were not able to have them as separate files stored within our static folder. That is tech debt that we owe.

Tool Configuration

- **Java Spring Boot:** Finalized the setup of Spring Boot and resolved port conflicts, which resulted in the project running on localhost:8081 due to other programs using port 8080.



Spring Tools 4 (aka Spring Tool Suite 4) 4.25.0.RELEASE

Spring Tools 4 is the next generation of Spring Boot tooling for your favorite coding environment. Largely rebuilt from scratch, it provides world-class support... [more info](#)

by [Broadcom](#), EPL

[spring](#) [Spring IDE](#) [Cloud](#) [Spring Tool Suite](#) [STS](#)

★ 4275 📦 Installs: **2.96M** (23,503 last month) Installed

- **H2 Database:** Maintained H2 as the backend database for task and event persistence. The configuration was finalized to ensure all CRUD operations work smoothly.
- **IDE:** The project is developed using Eclipse IDE version 4.29.0 with Spring Tools 4 (v4.24.0). All team members ensured their environments were properly set up for collaborative work.



- **Version Control System:** The team utilized the GitHub repository ([GitHub Repo](#)) for version control. Using Git made collaboration easier and improved the efficiency of code review.

2. Enhancing Core Functionality

Calendar and Event Management

- **Event/Task Population:** Implemented functionality to display events and tasks on the calendar in real-time. This was a key feature missing from Phase I. The JavaScript dependency used allows dynamic data population within the calendar interface, showing tasks on the appropriate dates. Additionally, you can select events and tasks that are populated on the calendar for more information.
- **CRUD Operations:** All CRUD functionalities for tasks and events were improved to provide a more seamless user experience. Users can now create, update, delete, and view tasks/events directly from the home page (will link to other HTML pages).

Recurring Events and Filtering

- **Recurring Events:** Added functionality to make events recurring. A checkbox was integrated into the form, allowing users to set recurring events. We resolved an issue where the checkbox was not initially translating into the correct Boolean value in the backend.
- **Filtering by Category:** The ability to filter tasks/events by category was also implemented, allowing users to easily find and manage specific events (e.g., work-related tasks vs. personal tasks).

Reminder Notifications

- **Reminder Feature:** A reminder notification system was implemented for both tasks and events. Users can opt to receive reminders, ensuring they stay on top of upcoming deadlines or events. This was achieved using JavaScript to trigger reminders based on event deadlines. The reminders were formatted so that the date and times are easily readable.

Form Validation

- **Input Fields:** Form validation was applied to all fields for events and tasks to ensure that the user has supplied a valid input. Additionally certain fields were marked as required or optional. This was implemented in JavaScript to provide a friendly user experience.

Error Handling

- **Input Fields:** Error handling was put in place to deny the saving of events or tasks with invalid inputs. Instead a simple pop up window is displayed to the user informing them of what is wrong/missing. This was also implemented with JavaScript.

3. Design Improvements

User Interface Enhancements

- **Multi Page Design:** The design of the project revolves around several HTML pages to display various information surrounding the events and tasks to the users. The home page features an interactive calendar that displays the events and tasks stored in the H2 database. Other pages are dedicated to event/task CRUD features as well as displaying the events/tasks in greater detail than the home page.
- **Interactive JavaScript Elements:** JavaScript was heavily used to create a more dynamic and engaging interface. Features like the interactive calendar and task filtering offer a better user experience.

Theme and Styling

- **CSS Challenges:** Initially, there were challenges getting the CSS to load within the Maven project. Ultimately we had to resort to adding CSS styling directly within the HTML pages. We finalized two themes: a dark theme for the home page and event/task CRUD features and then a light theme to display all events/tasks or filter them by category.
- **Responsive Design:** The front-end is designed to be responsive across various devices, ensuring users can access the calendar/task manager on various operating systems.

4. Testing

Testing

- **Test Cases:** We extended the test cases implemented in Phase I to cover new functionalities like recurring events, filtering, and reminder notifications. These tests ensure that tasks and events are correctly created, updated, and deleted from the database.
- **Integration Testing:** Integration testing focused on verifying that the JavaScript functionality, such as calendar population and reminder notifications, interacted correctly with the Spring Boot backend.

Issue Resolution

- **Empty Event and Task Fields:** Resolved an issue where events and tasks could be submitted and saved to the database with empty fields that should have been required. Additionally we solved issues where the checkbox for both recurring and reminders would not return the correct boolean value.
- **Calendar Population:** We encountered a bug where tasks were not being displayed in the calendar. After several iterations, we successfully linked the tasks/events to display on the correct dates using JavaScript.

Quality Assurance

Throughout Phase II, the team conducted regular code reviews to ensure best practices were followed. Each team member worked on separate functionalities, ensuring minimal overlap and efficient collaboration.

1. Test Cases

Table 1. Calendar App Traceability Matrix - Create Event

Test Case	Input/Output	Expected Result	Actual Result	Outcome (Pass/Fail)
1.1	Navigate to app URL (http://localhost:8081/)	The app homepage loads in the browser.	The app home page loads in selected browser	Pass
1.2	Click "Create Event" hyperlink on the homepage	User Interface (UI) transitions to event creation form.	UI transitions to the event creation form.	Pass
1.3	Enter event information: Title: "Event 1" Date: "10/04/2024" Time: "1:00 PM" Recurring: Yes Category: "Cat1" Location: "Adelphi, MD" Reminder: "10/03/2024"	Event created with the input information	Event created with the input information	Pass
1.4	Click "Add Event"	UI indicates event is created.	Event is created successfully.	Pass
1.5	View all events	All entered events appear in the UI.	Event appears in the calendar view.	Pass
1.6	Select "Home"	UI returns to the homepage.	UI returns to the homepage.	Pass

Table 2 – Calendar App Traceability Matrix Delete Event

Test Case	Input/Output	Expected Result	Actual Result	Outcome (Pass/Fail)
2.1	Navigate to: (http://localhost:8081/events/table)	All calendar events appear as a list/table in the UI.	All calendar events appear as a list/table in the UI.	Pass
2.2	Click "Delete" hyperlink	Alert message asking for confirmation appears.	Alert: "Are you sure you want to delete this event?"	Pass
2.3	Click "OK"	Event is removed from the list.	Event is removed from the list.	Pass
2.4.1	Select "Create New Event"	UI transitions to event creation form (See Table 1).	UI transitions to event creation form.	Pass
2.4.2	Select "Home"	UI returns to the homepage (See Table 1).	UI returns to the homepage.	Pass

Table 3 – Calendar App Traceability Matrix Edit Event

Test Case	Input/Output	Expected Result	Actual Result	Outcome (Pass/Fail)
3.1	Navigate to http://localhost:8081/events/table	All calendar events appear as a list/table in the UI.	All calendar events appear as a list/table in the UI	Pass
3.2	Click "Edit" on the first event	UI transitions to event update form, showing event information.	UI transitions to event update form, showing event information.	Pass
3.3	Edit event information: Title: "UpdatedEvent1" Date: "10/06/2024" Time: "2:00 PM" Recurring: No Category: "Cat2" Location: "Silver Spring, MD" Reminder: + 48hrs	Updated event is saved with the new information.	Updated event is saved with the new information.	Pass
3.4	Click "Update Event"	UI returns to event list, showing the updated information.	UI returns to event list, showing the updated information.	Pass
3.4.1	Select "Create New Event"	UI transitions to event creation form (See Table 1).	UI transitions to event creation form.	Pass
3.4.2	Select "Home"	UI returns to the homepage (See Table 1).	UI returns to the homepage.	Pass

Table 4 – Calendar App Traceability Matrix Create Task

Test Case	Input/Output	Expected Result	Actual Result	Outcome (Pass/Fail)
4.1	Navigate to: (http://localhost:8081/)	The app homepage loads in the browser.	The app homepage loads in the browser	Pass
4.2	Click "Create Task" on the homepage	UI transitions to task creation form.	UI transitions to task creation form.	Pass
4.3	Enter task information: Title: "Task1" Priority: "1" Deadline: "10/08/2024" Category: "Cat2" Reminder: Current date/time + 24hrs	Task created with the input information.	Task created with the input information.	Pass
4.4	Click "Add Task"	UI indicates task is created.	UI indicates task is created.	Pass
4.5	View all tasks	All entered tasks appear in the UI.	All entered tasks appear in the UI.	Pass
4.6.1	Select "Create Another Task"	UI transitions to task creation form (See 4.3).	UI transitions to task creation form	Pass

Table 5 – Calendar App Traceability Matrix Delete Task

Test Case	Input/Output	Expected Result	Actual Result	Outcome (Pass/Fail)
5.1	Navigate to http://localhost:8081/tasks/table	All tasks appear as a list/table in the UI.	All tasks appear as a list/table in the UI.	Pass
5.2	Click "Delete" hyperlink	Alert message asking for confirmation appears.	Alert: "Are you sure you want to delete this task?"	Pass
5.3	Click "OK"	Task is removed from the list.	Task is removed from the list.	Pass
5.4	Select "Create New Task"	UI transitions to task creation form (See Table 4).	UI transitions to task creation form.	Pass
5.4.1	Select "Home"	UI returns to the homepage (See Table 1).	UI returns to the homepage.	Pass

Table 6 – Calendar App Traceability Matrix Edit Task

Test Case	Input/Output	Expected Result	Actual Result	Outcome (Pass/Fail)
6.1	Navigate to http://localhost:8081/tasks/table	All tasks appear as a list/table in the UI.	All tasks appear as a list/table in the UI.	Pass
6.2	Click "Edit" hyperlink	UI transitions to task update form, displaying task information.	UI transitions to task update form, displaying task information.	Pass
6.3	Edit task information: Title: "UpdatedTask" Priority: "2" Deadline: "10/09/2024" Category: "Cat2" Reminder: Current date/time + 48hrs	Task updated with new information.	Task updated with new information.	Pass
6.4	Click "Update Task"	UI returns to task list, showing updated information.	UI returns to task list, showing updated information.	Pass
6.4.1	Select "Create New Task"	UI transitions to task creation form (See Table 4)	UI transitions to task creation form.	Pass
6.4.2	Select "Home"	UI returns to the homepage (See Table 1).	UI returns to the homepage.	Pass

Table 7 – Calendar App Traceability Matrix Track Tasks

Test Case	Input/Output	Expected Result	Actual Result	Outcome (Pass/Fail)
7.1	Navigates to (http://localhost:8081/tasks/table)	All tasks appear as a list/table in the UI.	All tasks appear as a list/table in the UI.	Pass
7.2.1	Click hyperlink to initiate "tracking" by priority.	Tasks are sorted by priority.	All tasks appear in the list/table, sorted by priority.	Pass

Table 8 – Calendar App Traceability Matrix Reminder Notifications

Test Case	Input/Output	Expected Result	Actual Result	Outcome (Pass/Fail)
8.1	Navigate to (http://localhost:8081/tasks/table)	All tasks appear as a list/table in the UI.	All tasks appear as a list/table in the UI.	Pass
8.2.1	Click "Create Event"	User Interface (UI) transitions to event creation form.	User Interface (UI) transitions to event creation form.	Pass
8.2.2	Enter event information: Title: "Notify Event" Date: "10/10/2024" Time: "3:00 PM" Category: "Cat3" Location: "Bethesda, MD" Reminder: Testing + time	Event is created with reminder set.	Event is created with reminder set.	Pass
8.3.1	Click "Create Task"	Event is created with reminder set.	Event is created with reminder set.	Pass
8.3.2	Enter task information: Title: "Notify Task" Priority: "3" Deadline: "10/20/2024" Category: "Cat4" Reminder: Testing day + time	Task is created with reminder set.	Task is created with reminder set.	Pass
8.4.1	At reminder time, check for event notification	App sends notification reminding user of the event.	App sends notification reminding user of the event.	Pass
8.4.2	At reminder time, check for task notification	App sends notification reminding user of the task.	App sends notification reminding user of the task.	Pass

Table 9 – Calendar App Traceability Matrix Filter by Category

Test Case	Input/Output	Expected Result	Actual Result	Outcome (Pass/Fail)
9.1	Navigate to (http://localhost:8081/tasks/table)	All tasks appear as a list/table in the UI.	All tasks appear as a list/table in the UI.	Pass
9.2	Use filtering option to filter tasks by category	Only tasks with the selected categories are displayed.	Only tasks with the selected categories are displayed.	Pass
9.3	Navigate to (http://localhost:8081/events/table)	All events appear as a list/table in the UI.	All events appear as a list/table in the UI.	Pass
9.4	Use filtering option to filter events by category	Only events with the selected categories are displayed.	Only events with the selected categories are displayed.	Pass

5. Final Deployment

Deployment Overview

- The project is now fully deployable however without a permanent server and domain name, the application will still need to be ran as a Maven project and accessed via localhost:8081.
- All functionalities—CRUD operations, calendar interactions, recurring events, filtering by category, and reminders—are fully operational.
- The final deployment includes a packaged Maven project that is easy to run and install.

Production Readiness:

We ensured that all key features are robust and production ready. The system was tested on multiple devices and browsers, ensuring that it functions smoothly across different platforms.

6. Conclusion

Phase II of the Calendar/Task Manager project addressed all issues from Phase I and added critical new functionalities, including calendar population, recurring events, and reminder notifications. The team worked collaboratively to improve the user interface, optimize performance, and ensure the project is ready for final deployment.