

Weekly Detailed Status Report

Period : 3 March 2025 – 17 April 2025

1. Phase 1: Initial CRUD & File-Load (3 Mar – 16 Mar)

Meetings :

- **3 Mar 2025** (Week 5 Kick-off)
- **7 Mar 2025** (Week 6 Review)

Key Objectives:

1. Finalize **database schema** and SQL setup script.
2. Implement **core CRUD** menu wiring to both SQL DB and in-memory BST.
3. Sketch out **file-load** bulk import functionality.
4. Scaffold **DatabaseIntegrationTests** and **FileIntegrationTests**.

What We Did :

- **DB + BST Sync:** On “Add Appointment,” write to DB first, then insert into BST. On “Remove,” delete from both.
- **SQL Script:** Reviewed and verified **db scripts/create database.sql—tables** for Appointments & Employees created correctly.
- **Menu Skeleton:** Built out the Admin/Employee menu stub (Add, Remove, Search, List, Load, Clear All).
- **AskOrBack Helper:** Added **AskStringOrBack/AskIntOrBack** loops to allow “back” at any prompt.
- **File Loader:** Read text file lines, parsed into **Appointment** objects, updated **nextAppointmentID**, and inserted into DB & BST.

- **Test Scaffolding :**
- **DatabaseIntegrationTests.cs** (insert/remove in try/finally against sandbox DB)
- **FileIntegrationTests.cs** (valid vs malformed lines)

Decisions & Changes :

- Malformed file lines should be logged (yellow warning) and **skipped**, not abort the entire load.
 - **nextAppointmentID** must jump past the highest ID in the imported file.
 - Integration tests must clean up test data in **finally** to avoid polluting the sandbox.
-

2. Phase 2: Input Validation & UI Polish (17 Mar – 9 Apr)

Meeting :

- **17 Mar 2025** (Week 7/8 Review)

Key Objectives :

1. Enforce **strict input validation** on patient/dentist names and dates.
2. Improve **user experience** by re-prompting on invalid input and adding visual spacers.
3. Finalize **FileIntegrationTests** and **PerformanceTests**.
4. Ensure **BST invariants** after bulk-load.

What We Did :

- **Regex Validation:** Patient and dentist names must match **`^[A-Za-z]+$`**.
- **Date Checks:** Reject dates earlier than today (year/month/day).
- **Loop-Until-Valid:** Wrapped each prompt in a **`while (!valid)`** loop so invalid entries display an error and let the user retry—no bouncing back to main menu.

- **UI Spacers:** Inserted blank `Console.WriteLine()` calls between prompts to reduce clutter.
- **FileIntegrationTests** extended for large-file performance and boundary conditions.
- **PerformanceTests:** Verified 10,000-appointment BST insertion completes in <1 s.

Decisions & Changes :

- Invalid input now shows a red error message, then re-asks the same question.
- Bulk-load now preserves BST order and DB state even if multiple malformed lines exist.
- Added unit tests for:
 - Empty-tree search
 - Remove from empty tree

3. Phase 3: Spectre.Console & Employee CRUD (10 Apr – 17 Apr)

Meetings:

- **10 Apr 2025** (Week 9 Kick-off)
- **17 Apr 2025** (Week 10 Final Review)

Key Objectives:

1. Replace basic console I/O with **Spectre.Console** menus, tables, panels, and spinners.
2. Implement full **Employee CRUD**: Add, Reset Password, **Remove**.
3. Finalize and categorize **all tests** (unit vs integration).
4. Polish **CI workflow** and documentation.

What We Did :

- **UI Overhaul:**
 - Upgraded to Spectre.Console v0.50 for rich rendering.
 - Replaced `Console.WriteLine` menus with `SelectionPrompt`, `Table`, `Panel`, and `Markup`.
 - Introduced a `Status` spinner for long-running tasks (bulk-load).
- **Employee Removal:**
 - Added `EmployeeDataAccess.RemoveEmployee(int id)` & confirmation prompt.
 - Wrote **EmployeeTests.cs** covering add, reset, and remove; flagged as `[TestCategory("Unit")]`.
- **Test Coverage:**
 - **Unit Tests:** AppointmentBST, Employee flows, input validation.
 - **File Integration Tests:** Bulk-load valid & malformed scenarios.
 - **Performance Tests:** BST insertion at scale.
 - **DB Integration Tests:** Appointment insert/remove against sandbox DB (run locally).
 - **CI Pipeline** configured to skip DB integration tests by default.
- **Documentation & CI:**
 - Updated **README** with SSMS instructions, "Back" workflow, and spinner usage.
 - GitHub Actions runs `make restore`, `make build`, and `make test` (skipping integration).
 - Final null-check guards added throughout code.

Decisions & Changes :

- CI pipeline will continue to skip DB integration tests; developers run them locally in SSMS.
- All tests are now green.
- Readme, SQL scripts, and makefile reviewed for completeness.

Recorded by : Yug Patel (Scrum Master)

Date: 17 April 2025