# CS 459: Software Engineering Senior Project

## Spring 2025

# Test Plan

Last Modified: 2025-05-16

| | |
|---|---|
| Project Title | HCAR Client Database |
| Sponsoring Company | Humboldt Community Access and Resource Center |
| Sponsor(s) | Pennie Lee, Kim Nash, Wes Patterson |
| Students | 1. Orlando Trujillo-Ortiz<br>2. Carson Gustafson<br>3. Justin Crittenden<br>4. Michael Goodwyn |

# ABSTRACT

# TABLE OF CONTENTS



# 1. INTRODUCTION

This document outlines the details behind testing criteria/constraints, as well as the techniques we used to generate our test suites. Tests in this project were run using Jest, a testing framework for JavaScript code. Test criteria and coverage will be listed in a tabular form below, for each of the files that the test suite covered.

# 2. TEST CASES

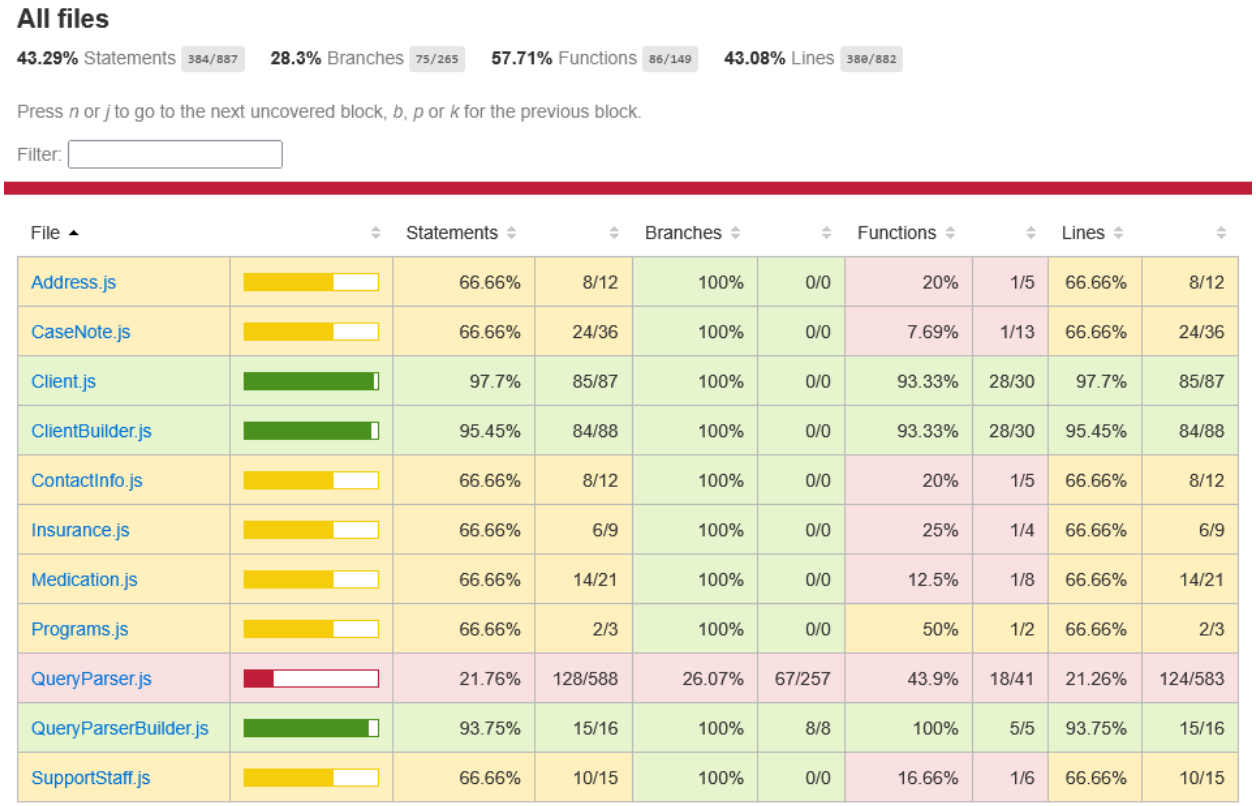
All test case results can be viewed from Fig. 1 below. Test case result information shows the coverage on each file for branch, statement, and line coverage respectively for the current files in the project. It is worth noting that test cases for the queryParser.js file were not covered as the project came to its deadline, and will be fixed in a future release.

# 3. TECHNIQUES FOR TEST GENERATION

Tests for this project are created using the Jest JavaScript library. This project utilizes both Unit and Integration test suites to ensure code quality. Some Unit tests make use of “mocks” to simulate usage of modules pertaining to database connectivity that would hinder the performance of the test suites. Integration tests in this project may make realized connections to external services, such as the project’s cloud MySQL database.

Design methods for test cases vary depending on the complexity and importance of the

relevant code units or modules being tested. More primitive class structures intended to hold data only receive simple happy path testing due to the nature of Test Driven Development encouraging tests to be written before functionality. 2-Point Boundary Value Analysis is applied when applicable during the test design process. Code coverage is measured by default using Statement, Branch, and Function coverage by the Jest test runner.



**All files**

**43.29%** Statements 384/887 **28.3%** Branches 75/265 **57.71%** Functions 86/149 **43.08%** Lines 380/882

Press *n* or *j* to go to the next uncovered block, *b*, *p* or *k* for the previous block.

Filter:

| File | Statements | | Branches | | Functions | | Lines | |
|---|---|---|---|---|---|---|---|---|
| Address.js | 66.66% | 8/12 | 100% | 0/0 | 20% | 1/5 | 66.66% | 8/12 |
| CaseNote.js | 66.66% | 24/36 | 100% | 0/0 | 7.69% | 1/13 | 66.66% | 24/36 |
| Client.js | 97.7% | 85/87 | 100% | 0/0 | 93.33% | 28/30 | 97.7% | 85/87 |
| ClientBuilder.js | 95.45% | 84/88 | 100% | 0/0 | 93.33% | 28/30 | 95.45% | 84/88 |
| ContactInfo.js | 66.66% | 8/12 | 100% | 0/0 | 20% | 1/5 | 66.66% | 8/12 |
| Insurance.js | 66.66% | 6/9 | 100% | 0/0 | 25% | 1/4 | 66.66% | 6/9 |
| Medication.js | 66.66% | 14/21 | 100% | 0/0 | 12.5% | 1/8 | 66.66% | 14/21 |
| Programs.js | 66.66% | 2/3 | 100% | 0/0 | 50% | 1/2 | 66.66% | 2/3 |
| QueryParser.js | 21.76% | 128/588 | 26.07% | 67/257 | 43.9% | 18/41 | 21.26% | 124/583 |
| QueryParserBuilder.js | 93.75% | 15/16 | 100% | 8/8 | 100% | 5/5 | 93.75% | 15/16 |
| SupportStaff.js | 66.66% | 10/15 | 100% | 0/0 | 16.66% | 1/6 | 66.66% | 10/15 |

*Fig. 1 2025-05-16 Code Coverage Report from All Test Suites*

# 4. TRACEABILITY

| User Story ID | User Story Summary | Test Case ID |
|---|---|---|
| #1 | View client demographics and non-medical information | |
| #2 | Add new clients | |
| #6 | Case Note Creation | |
| #7 | Case Note Viewing | |

| | | |
|---|---|---|
| #8 | Secure User Login | |
| #10 | Client Filtering | |
| #13 | Elevated Managerial Access | |

# 5. CONFIGURATION MANAGEMENT

This Test Plan is version controlled via Google Docs document history. Static PDF copies of this plan can be found on the project's GitHub repository in the Formal Documentation directory.