

# CS 459: Software Engineering Senior Project

Spring 2025

## Test Plan

Last Modified: 2025-04-29

Project Title	HCAR Client Database
Sponsoring Company	Humboldt Community Access and Resource Center
Sponsor(s)	Pennie Lee, Kim Nash, Wes Patterson
Students	<ol style="list-style-type: none"><li>1. Orlando Trujillo-Ortiz</li><li>2. Carson Gustafson</li><li>3. Justin Crittenden</li><li>4. Michael Goodwyn</li></ol>

# ABSTRACT

## TABLE OF CONTENTS

ABSTRACT.....	2
TABLE OF CONTENTS.....	2
1. INTRODUCTION.....	2
2. TEST CASES.....	2
3. TECHNIQUES FOR TEST GENERATION.....	2
4. TRACEABILITY.....	3
5. CONFIGURATION MANAGEMENT.....	4
6. ENGINEERING STANDARDS AND MULTIPLE CONSTRAINTS.....	4
7. ADDITIONAL REFERENCES.....	4

### 1. INTRODUCTION

- introduction to the entire document
- purpose and scope of the document
- description of the structure of the document

### 2. TEST CASES

- High-Level Test Cases
  - o Summarize key system-level or acceptance tests.
  - o Reference the user stories/use cases from Requirements Documentation.

### 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.

<b>Table. 1: Basic Generated Code Coverage Report For Tested Files (End of Sprint #6)</b>					
<b>File</b>	<b>% Stmts</b>	<b>% Branch</b>	<b>% Funcs</b>	<b>% Lines</b>	<b>Uncovered Line #s</b>
<b>All files</b>	81.03	76.47	70.08	80.84	
Address.js	66.66	100	20	66.66	22-34
CaseNote.js	66.66	100	9.09	66.66	56-92
Client.js	98.8	100	96.55	98.8	175
ClientBuilder.js	97.64	100	96.55	97.64	232-233
ContactInfo.js	66.66	100	25	66.66	23-31
Insurance.js	66.66	100	33.33	66.66	24-28
Medication.js	66.66	100	12.5	66.66	37-67
Programs.js	66.66	100	50	66.66	19
QueryParser.js	67.2	72.88	93.33	66.11	...62, 167-173, 191-195, 205-206, 237-278, 298-299, 319-320
QueryParserBuilder.js	93.75	100	100	93.75	63
SupportStaff.js	66.66	100	16.66	66.66	39-55

## 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.