

# Software Design Document

Viper-Rocks 3338

May 10, 2025

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>System Architecture</b>	<b>2</b>
<b>3</b>	<b>Workflow Diagram</b>	<b>2</b>
<b>4</b>	<b>User Interface</b>	<b>2</b>
4.1	UI Frameworks . . . . .	2
4.2	Primary Screens . . . . .	3
4.3	Database Integration . . . . .	3
<b>5</b>	<b>Glossary</b>	<b>3</b>
<b>6</b>	<b>References</b>	<b>3</b>

## Version History

Version	Date	Description
1.0	May 16, 2025	Initial outline: core requirements and architecture
2.0	May 23, 2025	Added reporting and CSV export features
3.0	May 30, 2025	Snapshot 3: integrated TestRail, refined tech stack and UI frameworks
4.0	May 10, 2025	Snapshot 4: added workflow diagram and final polish

## 1 Introduction

- **Purpose:** Present the design blueprint for the Lunar Rocks! analysis tool, covering component interactions, data flows, and UI architecture
- **Intended Audience:** Development team, testers, and project stakeholders
- **Overview:** Snapshot 4 finalizes system design by adding a high-level workflow diagram and refining UI and architecture details

## 2 System Architecture

- **Architecture Pattern:** Separation of front-end (React + Next.js) and back-end (Node.js REST API)
- **Tech Stack:** React 18, Next.js 14, Konva.js for canvas tasks; Node.js + Express; PostgreSQL 16 with PostGIS; Sequelize ORM; QGIS for preprocessing
- **Data Flow Diagrams:** Level 0 outlines user interactions; Level 1 details core modules and data pipelines
- **Services:** CSV export endpoint; reporting engine aggregates metrics; TestRail connector displays test-run summaries

## 3 Workflow Diagram

## 4 User Interface

### 4.1 UI Frameworks

- Based on JPL's Explorer-1 Design System with React and TailwindCSS
- Accessibility focused keyboard navigation, screen-reader labels
- Simple clean interface with clearly labeled features

## 4.2 Primary Screens

- *Upload*: File picker and project selector dashboard
- *Scouting*: Auto-numbered segment gallery
- *Sizing Canvas*: Konva-based measurement tools with undo/redo
- *Classification*: Category icons with visual guidance
- *Reporting Dashboard*: Filterable tables and charts
- *Export Dialog*: CSV/PDF format chooser and download
- *TestRail Dashboard*: Shows linked test runs and pass/fail statuses

## 4.3 Database Integration

- Sequelize with connection pooling for time efficient queries
- Entities: users, images, segments, measurements, classifications, *test\_runs*

## 5 Glossary

**UI** User Interface

**API** Application Programming Interface

**DFD** Data Flow Diagram

**CSV** Comma-Separated Values

**PDF** Portable Document Format

**TR** TestRail

**ORM** Object-Relational Mapping

**WCAG** Web Content Accessibility Guidelines

## 6 References

- Project brief: <https://ascent.cysun.org/project/project/view/227>
- SRS source: <https://github.com/Chris11111111/Viper-Rocks/blob/main/docs/SRS.tex>

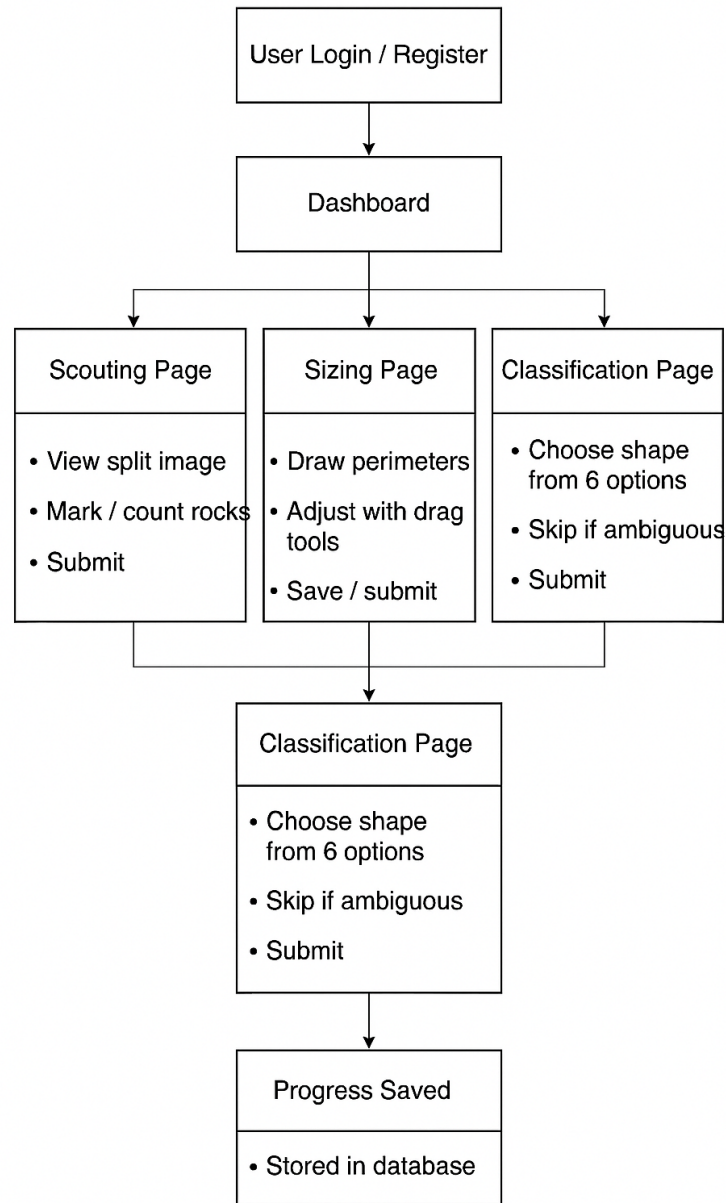


Figure 1: High-level user workflow from login through scouting, sizing, classification, and data storage