

# Southern California Edison VR Training Simulation Software Design Document

Brailey Gonzalez, Kevin Guerrero, Derek Rosales

December 2025

# Contents

<b>Version History</b>	<b>2</b>
<b>1 Introduction</b>	<b>3</b>
1.1 Purpose . . . . .	3
1.2 Scope . . . . .	3
1.3 Intended Audience . . . . .	3
1.4 Overview . . . . .	3
1.5 References . . . . .	3
1.6 Definitions, Acronyms, and Abbreviations . . . . .	3
<b>2 System Architecture</b>	<b>4</b>
2.1 Workflow . . . . .	4
2.2 Site Breakdown . . . . .	4
2.3 Architecture Overview . . . . .	4
2.4 Data Flow . . . . .	4
<b>3 User Interface</b>	<b>5</b>
3.1 How to Use . . . . .	5
3.1.1 VR Client . . . . .	5
3.2 Database Explanation . . . . .	5
<b>Glossary</b>	<b>6</b>
<b>References</b>	<b>7</b>

## Version History

User	Date	Reason for Changes	Version
Brailey Gonzalez	12/5/25	Snapshot 1 Update	1.0
Kevin Guerrero	12/6/25	Snapshot 2 Update	2.0
Derek Rosales	12/7/25	Snapshot 3 Update	3.0
Team	12/9/25	Snapshot 4 Update	4.0

# **1 Introduction**

## **1.1 Purpose**

This Software Design Document (SDD) serves as a blueprint for the creation of the Worker Training Simulation (WTS) program. It provides a structured plan that outlines the system's architecture, core functionality, and design decisions. The document is intended to guide the development team by clearly defining how the VR training environment will be built, how users will interact with the system, and how data will be managed.

## **1.2 Scope**

The scope of this document includes the design and implementation of the WTS. It covers the VR framework setup, database integration, and task delegation for the development team.

## **1.3 Intended Audience**

The intended audience for this document includes the development team, and users who will specifically be trainees who perform field work such as electrical wiring, equipment handling, and safety tasks.

## **1.4 Overview**

The WTS application will be a VR-based simulation that allows workers to practice tasks safely in a virtual environment. The application will provide a user-friendly interface and will be designed to improve safety awareness, task proficiency, and experience.

## **1.5 References**

See the references section for a list of documents and resources referenced in this document.

## **1.6 Definitions, Acronyms, and Abbreviations**

See the glossary section for definitions of terms, acronyms, and abbreviations used in this document.

## 2 System Architecture

### 2.1 Workflow

- Trainee logs in through the VR client using their credentials.
- Trainee selects a training scenario.
- Trainee performs tasks such as wiring tasks and equipment management.
- The VR client logs trainee actions and sends them to the backend.
- The backend stores data in the PostgreSQL database.
- Scores are generated and stored for later review.

### 2.2 Site Breakdown

- **VR Client:** Developed in Unity with Oculus SDK and OpenXR support.
- **Backend Services:** Implemented in Node.js with Express for REST APIs.
- **Database:** PostgreSQL database for storing trainee accounts, scenarios, and training scores.
- **Version Control:** GitHub repository for collaboration and code management.

### 2.3 Architecture Overview

The VR client follows a Model-View-Controller (MVC) architectural pattern:

- **Model:** Represents training data and user progress.
- **View:** VR environment and user interface.
- **Controller:** Handles interactions and communication with backend APIs.

### 2.4 Data Flow

1. The trainee interacts with objects in the VR training room.
2. The VR client logs actions and sends them to the backend via REST API.
3. The backend stores data in PostgreSQL.
4. Scores are calculated and stored for each trainee session.

## 3 User Interface

### 3.1 How to Use

#### 3.1.1 VR Client

Launch the VR application on a supported headset. The main menu contains options for:

- **Begin Training:** Enter the simulation.
- **Select Training**
- **Perform Tasks**
- **Receive Score:** Score is displayed when training is complete.
- **Exit:** Close the application.

Inside the simulation:

- Trainees can pick up, move, and inspect objects.
- Feedback is provided through object highlights and end-of-session scores.

### 3.2 Database Explanation

The database for the WTS is implemented using PostgreSQL. It stores all data related to trainees and their training sessions, including:

- **Trainee Accounts:** Login credentials (username, password).
- **Profiles:** Basic trainee details such as name and department.
- **Training Sessions:** Records of each session including scenario ID, start/end time, and completion status.
- **Scores:** Performance scores generated from each training session.
- **Scenario Definitions:** Stored procedures, hazards, and scoring rules for training modules.

Database management tools such as pgAdmin and DBeaver will be used to administer PostgreSQL.

## Glossary

**WTS** - Worker Training Simulation

**VR** - Virtual Reality

**SDK** - Software Development Kit

**UI** - User Interface

**MVC** - Model-View-Controller

**PostgreSQL** - Relational database management system used for WTS

## References

Ascent - Project (2022). <https://ascent.cysun.org/project/project/view/187>