

# Robot Operator Application

## Overview

Create an Android application that allows visualization and annotation of construction site rooms with robot placement. This tests your ability to handle 3D rendering, spatial data, and interactive annotations in a construction automation context.

## Core Requirements

1. 3D Room Visualization
  - Implement a [3D room viewer](https://skfb.ly/6YYLL) (<https://skfb.ly/6YYLL>) that can:
    - Render a basic rectangular room
    - Show proper perspective and lighting
    - Allow camera movement (pan, zoom, rotate)
2. Room Annotation System
  - Create an annotation system that allows (for now assume only rectangular selections):
    - Marking areas on walls that need finishing
    - Saving annotations with position data
    - Different annotation types (spray area, sand area, obstacles)
3. Robot Placement **[Bonus]**
  - Implement functionality to:
    - Place robot model in the room (ur10e urdf - [https://github.com/Daniella1/urdf\\_files\\_dataset/blob/main/urdf\\_files/ros-industrial/xacro\\_generated/universal\\_robots/ur\\_description/urdf/ur10e.urdf](https://github.com/Daniella1/urdf_files_dataset/blob/main/urdf_files/ros-industrial/xacro_generated/universal_robots/ur_description/urdf/ur10e.urdf) )

## Technical Requirements

1. Architecture
  - MVVM architecture
  - Dependency Injection using Hilt
  - Single activity, multiple fragments design
2. Required Technologies
  - OpenGL ES for 3D rendering
  - Kotlin Coroutines
  - Room database for storing annotations
  - Jetpack Compose for 2D UI elements

## Must-Have Features

1. Room Creation
  - Basic room visualization
  - Camera controls
2. Annotation Tools
  - Wall selection
  - Area marking
  - Annotation type selection
  - Save/load annotations
3. Robot Placement
  - Drag and drop robot placement

## **Submission Requirements**

1. README with:
  - Setup instructions
  - Used libraries and tools
  - Technical decisions
  - Known limitations
2. Short demo video showing key features

## **Technical**

- Kotlin
- Python for API's