

1. Introduction

1.1 Purpose of Document

This is a Requirements Specification document for a Street designing application. This document describes the scope, objectives and goal of this application. In addition to describing non-functional requirements, this document models the functional requirements with use cases, and class models. This document is intended to direct the design and implementation of the target system in an object-oriented language.

1.2 Project Scope

The scope of this Project is a terminal-based application providing Street designing with several types of architectures on a 2D plane with variable lengths, positions, heights and other aspects unique to their types.

1.3 System Purpose

1.3.1 Users

The application provides full freedom for user and creates the Street and every architecture with the values user enters. Application checks every action if they are valid and warns the user whenever an exception occurs behalf of the user's inputs, thus design mistakes never happen. Numerical data and other unique to architecture type data are available for user to view, Street silhouette is also shown graphically for further visualization.

1.3.2 Responsibilities

Primary responsibilities of application:

- Creation of a street with the length user desires
- Fully editable Street
 - Add any type of architecture with any valid values that user desires
 - House type architecture creation
 - Market type architecture creation
 - Office type architecture creation
 - Playground type architecture creation
 - Remove any architecture that user selects from the list of existing architectures
- View the street's situation
 - Remaining and occupied length data viewing
 - Viewing list of architectures on the Street
 - Visual representation of the street

2. Capabilities and Restrictions

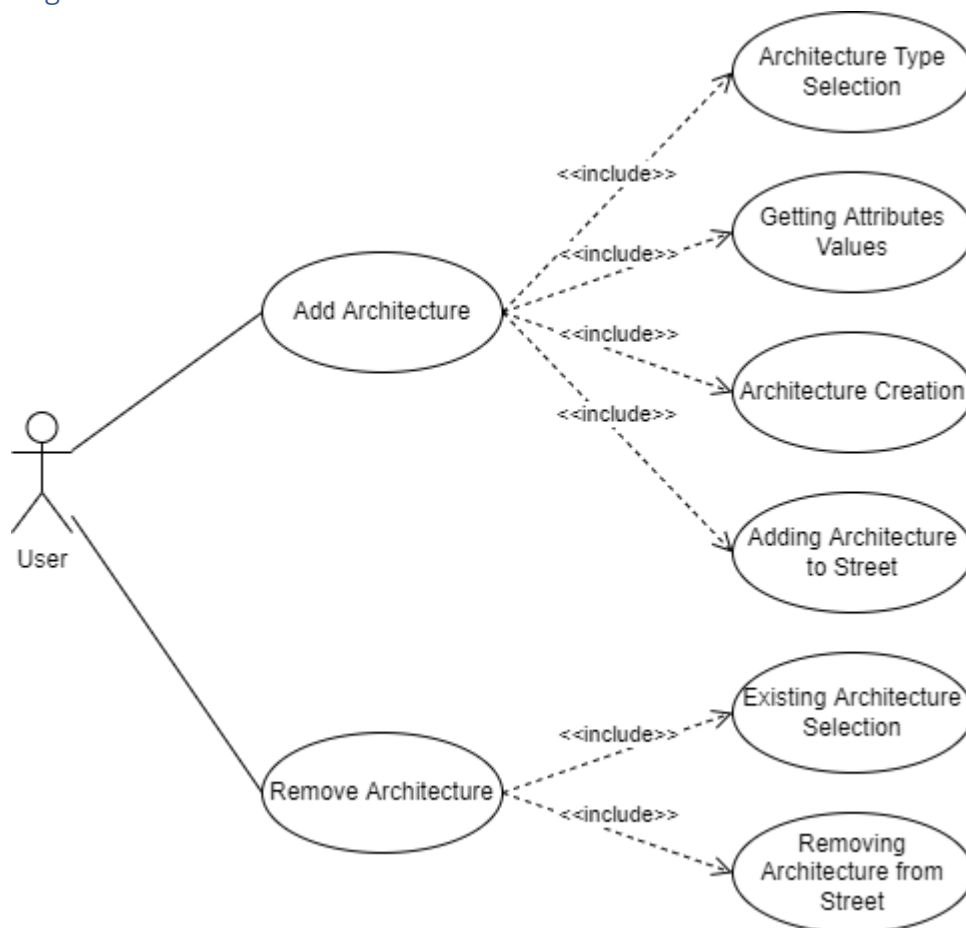
(Problem Solutions Approach)

Street is needed to keep all the architectures on it somewhere somehow. Architectures are also able to be built on left or right side of the street. For these reasons a 2D array is used for containing the architectures one 1D array for left and one for the right side of it. These 1D arrays are reallocated when their limit is reached or allocating memory more than they need, this way not much unnecessary memory is used. The allocation is incremented by the data size therefore making it efficient as speed wise too. Architectures are held consequential in the arrays so it's easier to shrink the array when needed.

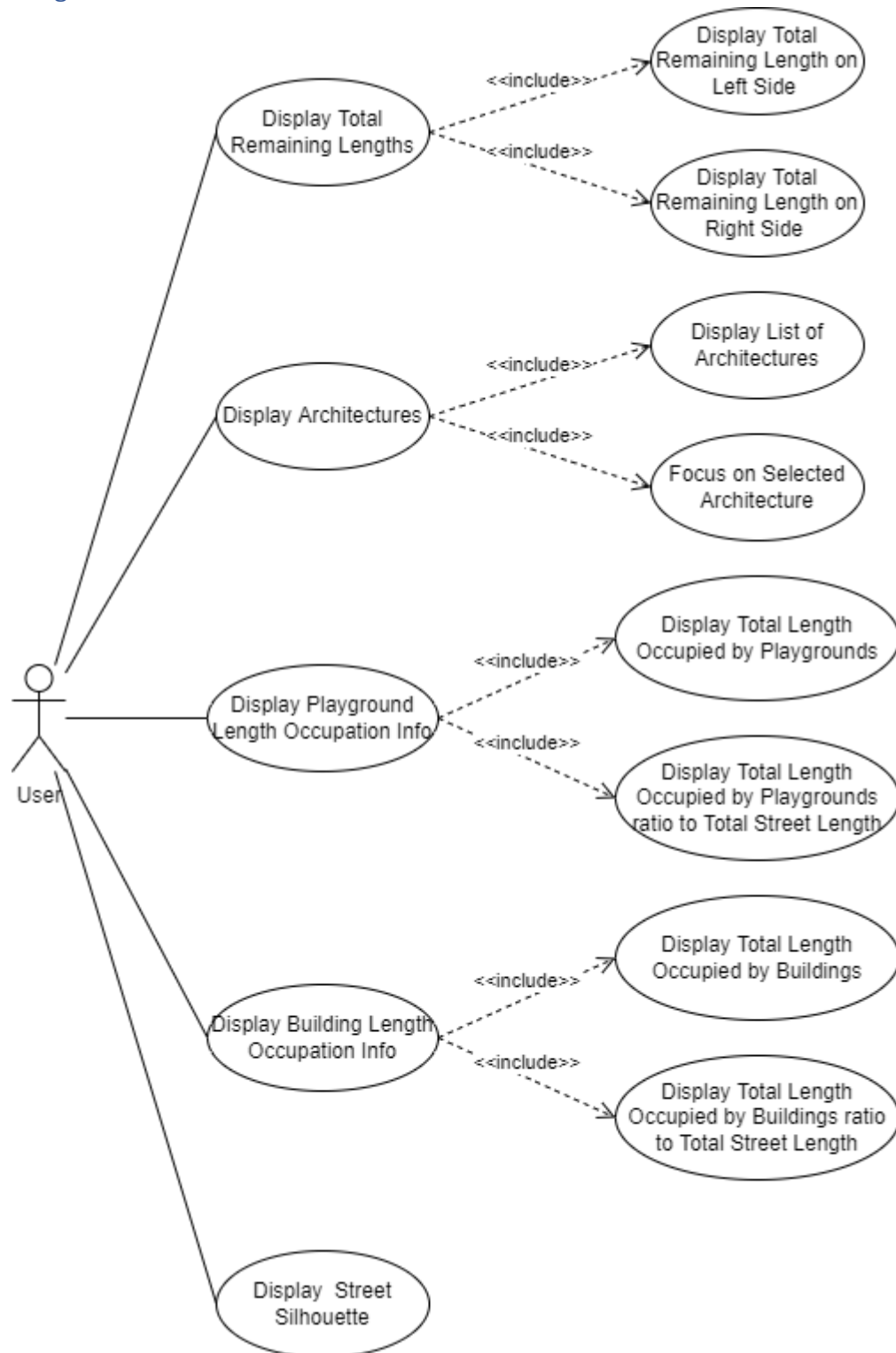
Application runs on a terminal, so the graphical visualization of the street is made with characters, therefore making it a maximum limit of length by the properties of user's monitor and the terminal's behavior. When the street's length exceeds the maximum number of characters that can be printed on one line on the terminal there will be overflows of the visualization.

3. Use Case Model

3.1 Editing Mode



3.2 Viewing Mode



4. Class Model

