

Design Document

First version



Bilger yahov

oLEKSANDR SUPRUNENKO

ILIA NIKUSHEV

GEORGI CHISHIRKOV

LYUBOMIR DIMOV

MENGCHUAN LIU

Traffic Lights System

Table of Contents

[Introduction 2](#_Toc446013005)

[Class Diagram 3](#_Toc446013006)

[Renderable inheritance overview 4](#_Toc446013007)

[Component classes 4](#_Toc446013008)

[Crossing classes 5](#_Toc446013009)

[Simulation classes 6](#_Toc446013010)

[Crossing container (Recycle bin and saved crossings) 7](#_Toc446013011)

[Undo classes 7](#_Toc446013012)

[Traffic manager Grid and System state classes 8](#_Toc446013013)

[Description of the classes and their members 9](#_Toc446013014)

[Sequence Diagrams 10](#_Toc446013015)

[Create a simulation 10](#_Toc446013016)

[Edit a road traffic flow 11](#_Toc446013017)

[Place a crossing 12](#_Toc446013018)

[Remove a crossing 13](#_Toc446013019)

[Select a crossing to place 14](#_Toc446013020)

[Start a simulation 15](#_Toc446013021)

[Stop a simulation 16](#_Toc446013022)

# Introduction

This document gives information about the class diagram for the *“Traffic Lights”* application. Description of the classes and the attributes and methods in each class is given. Furthermore, some sequence diagrams of the application are presented.

The class diagram is a static diagram. It represents the static view of the application. Our class diagram is not only used for visualizing, describing and documenting different aspects of a system but also for constructing executable code of the software application. The class diagram describes the attributes and operations of a class and also the constraints imposed on the system.

Our class diagram can be mapped directly with object oriented languages. It shows a collection of classes, interfaces, associations, collaborations and constraints.

The UML diagrams like activity diagram, sequence diagram can only give the sequence flow of the application but class diagram is a bit different. So it is the most popular UML diagram in the coder community.

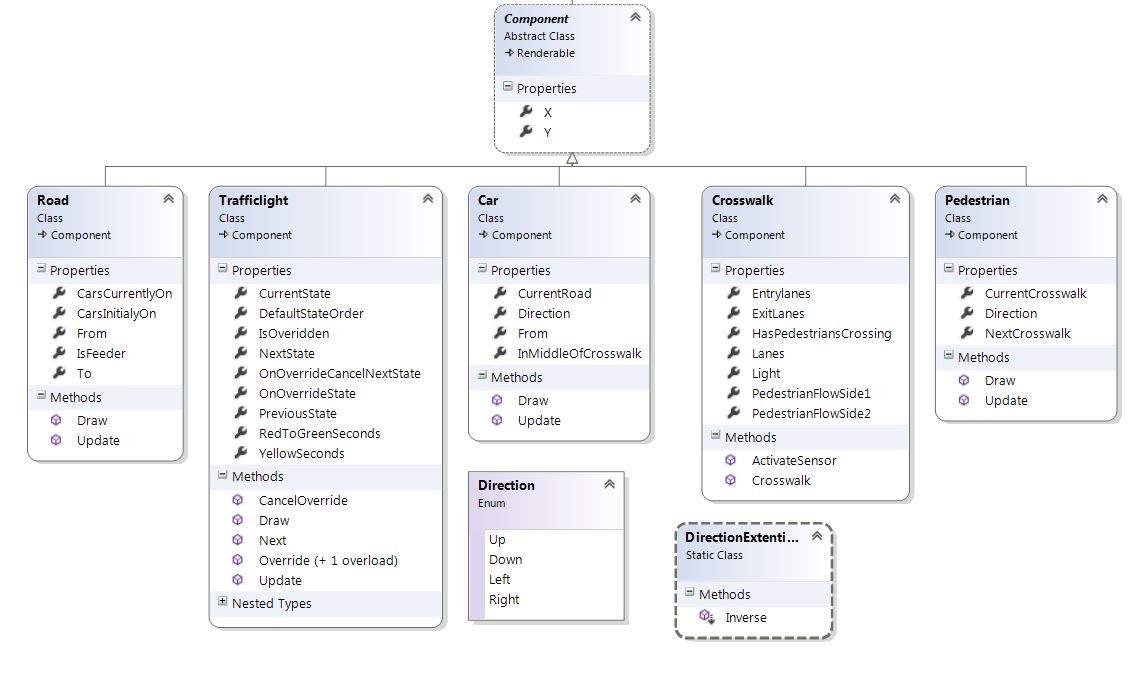
In the document can be found sequence diagrams which purpose is:

* Draw the activity flow of a system.
* Describe the sequence from one activity to another.
* Describe the parallel, branched and concurrent flow of the system.

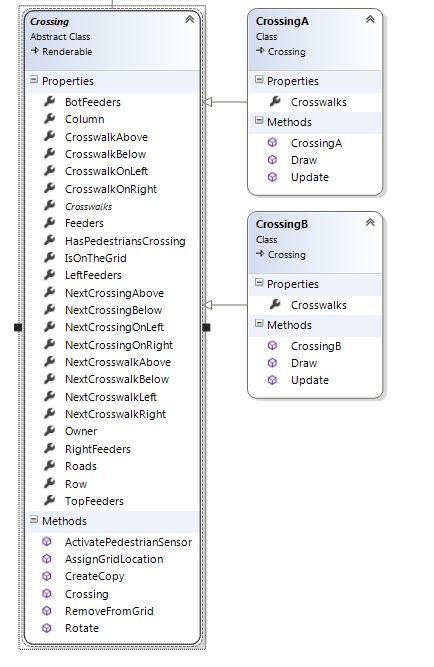
# Class Diagram

## C:\Users\user\Desktop\ProCP\Renderable.JPGRenderable inheritance overview

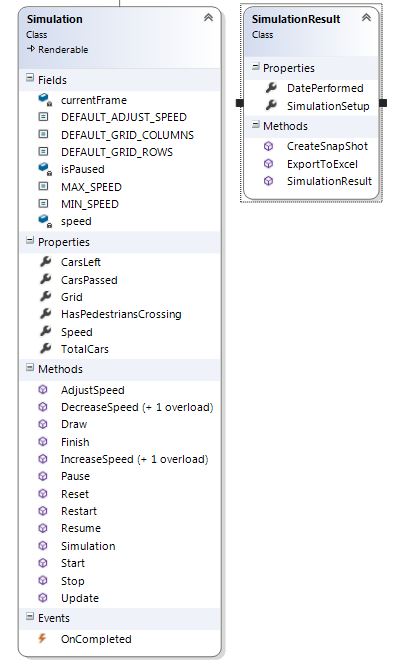
## Component classes



## Crossing classes



## Simulation classes

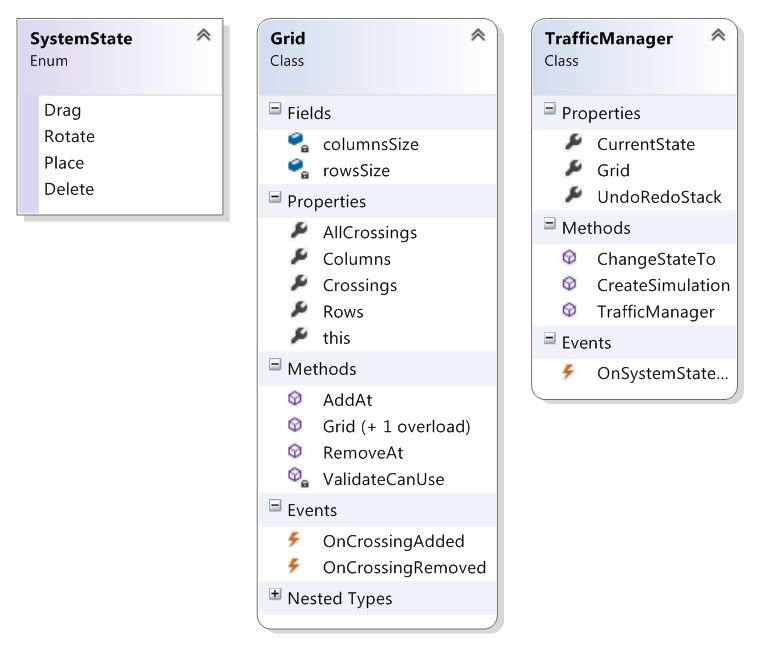


## C:\Users\user\Desktop\ProCP\Bin and Cpaste.JPGCrossing container (Recycle bin and saved crossings)

## Undo classes

## C:\Users\user\Desktop\ProCP\UndoRedo.JPG

## Traffic manager Grid and System state classes



# Description of the classes and their members [early draft]

ActionStack - Container that handles Undo and Redo of actions

AddCrosswalkAction, MoveCrosswalkAction, UpdatePropertiesAction, RotateAction – various actions that can be undone and redone in the system

Car – a renderable component within the simulation that moves on a road

Component – abstract class containing coordinates for various objects within the simulation

Crossing – abstract class defining the basic properties and methods for the crossings

CrossingA – implements Crossing and is the basic type, with no crosswalks

CrossingB – implements Crossing, similar to type A but has 2 parralel crosswalks on each side

SimulationResults – extracts the results from a simulation into an Excel

Trafficlight – a component that controls the movement of cars and pedestrians

UndoableAction – abstract class containing basic logic for undoing and redoing of actions

SystemState – current state of the system

Pedestrian – a component that moves on the sidewalks and crosswalks of the roads

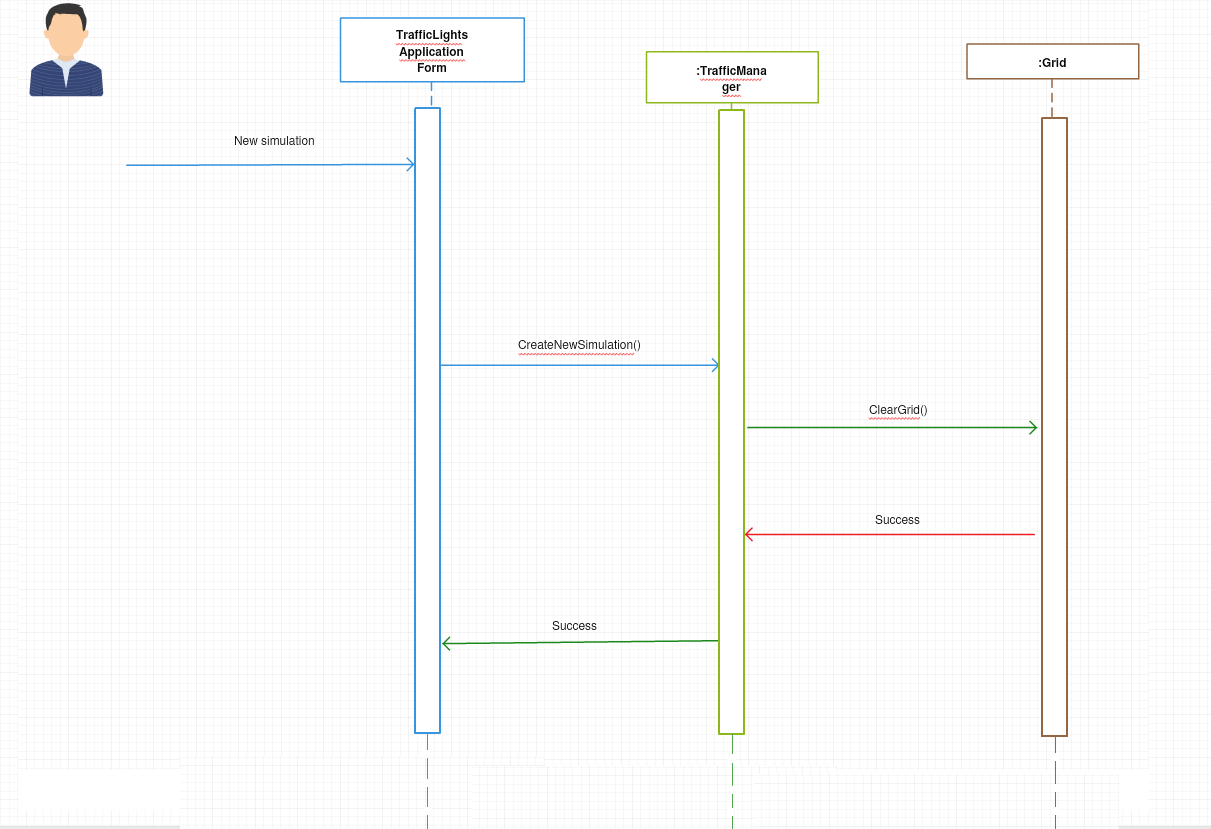
Grid – represents 1 square of the 3x3 grid within the system

RecycleManager – the recycle bin where removed crossings are stored

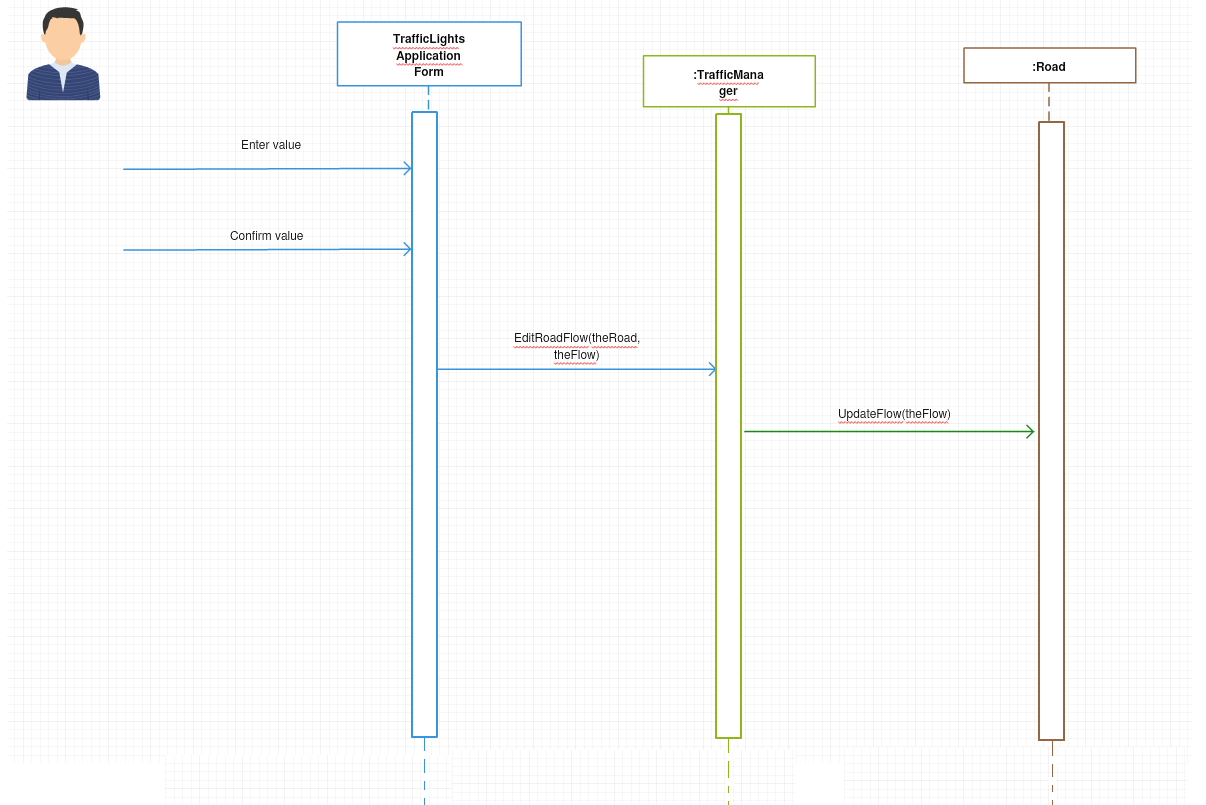
SavedManager – the place where user saved crossings are stored with predefined properties

# Sequence Diagrams

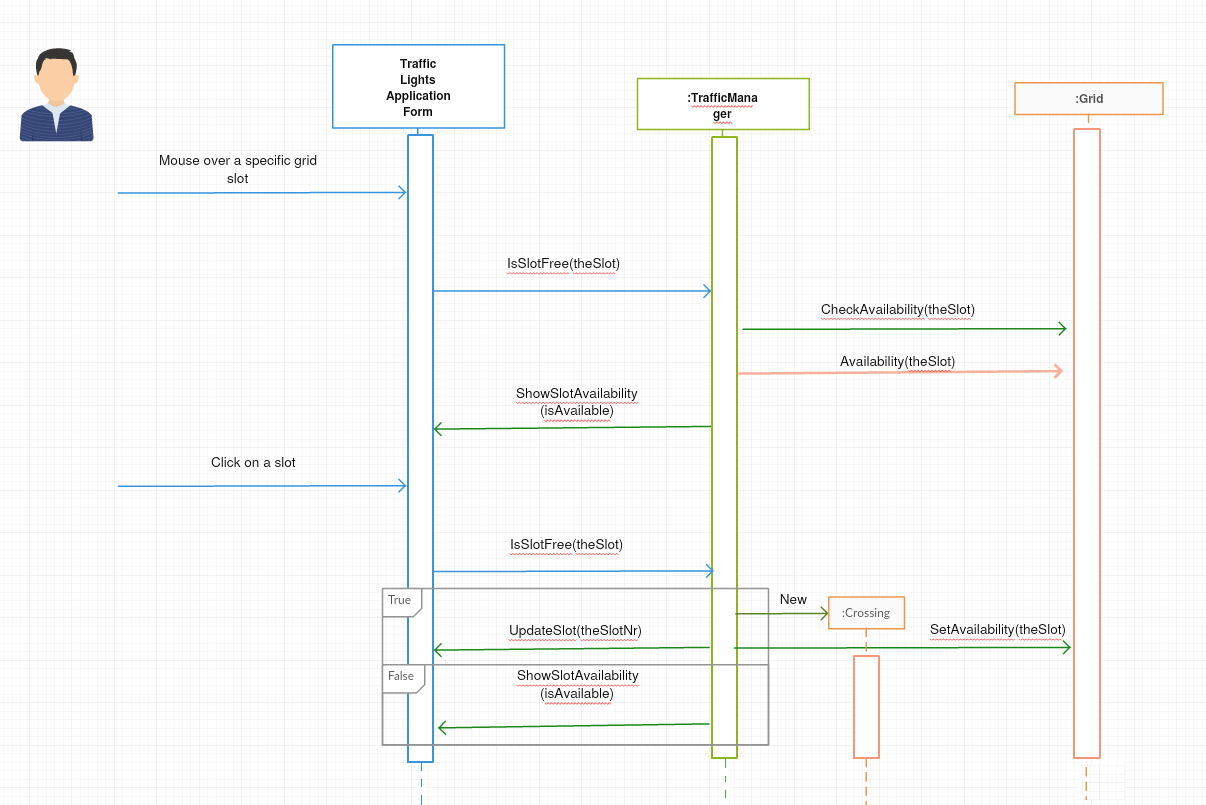
## Create a simulation



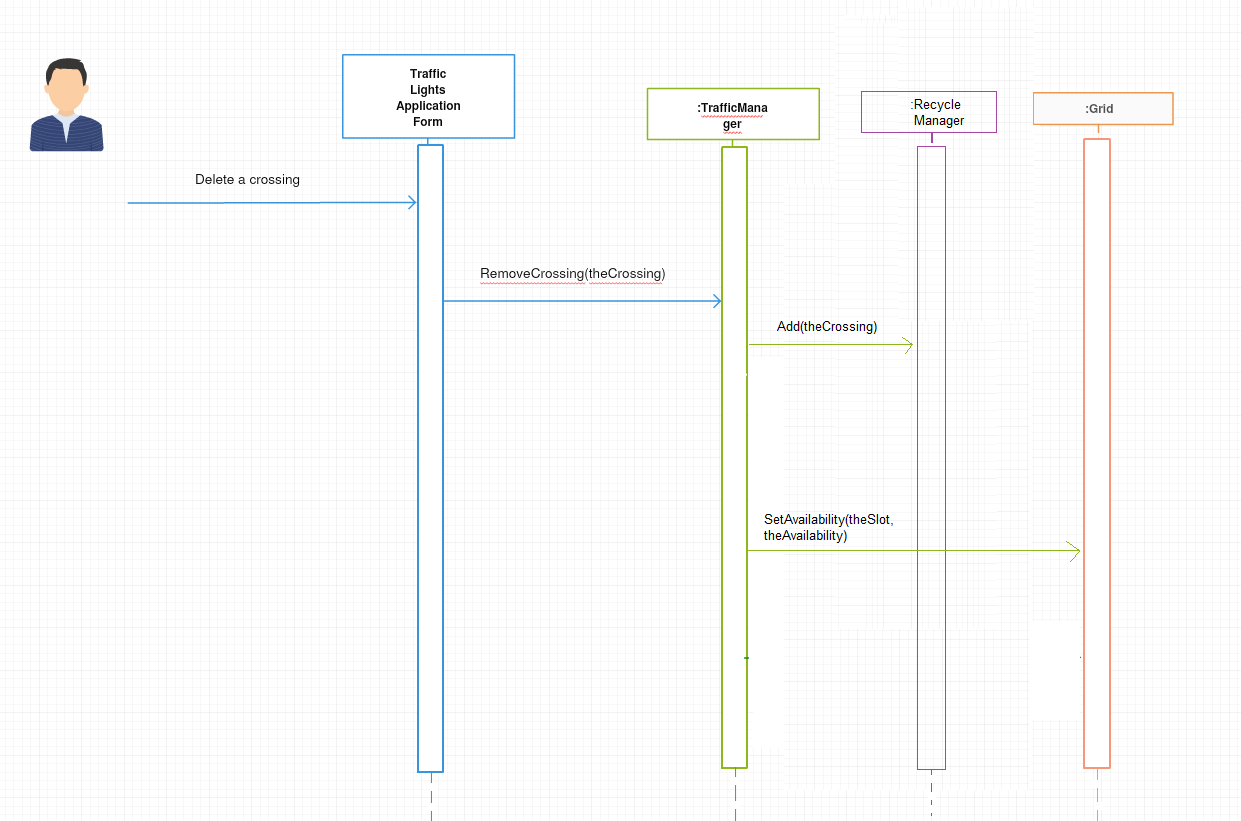
## Edit a road traffic flow



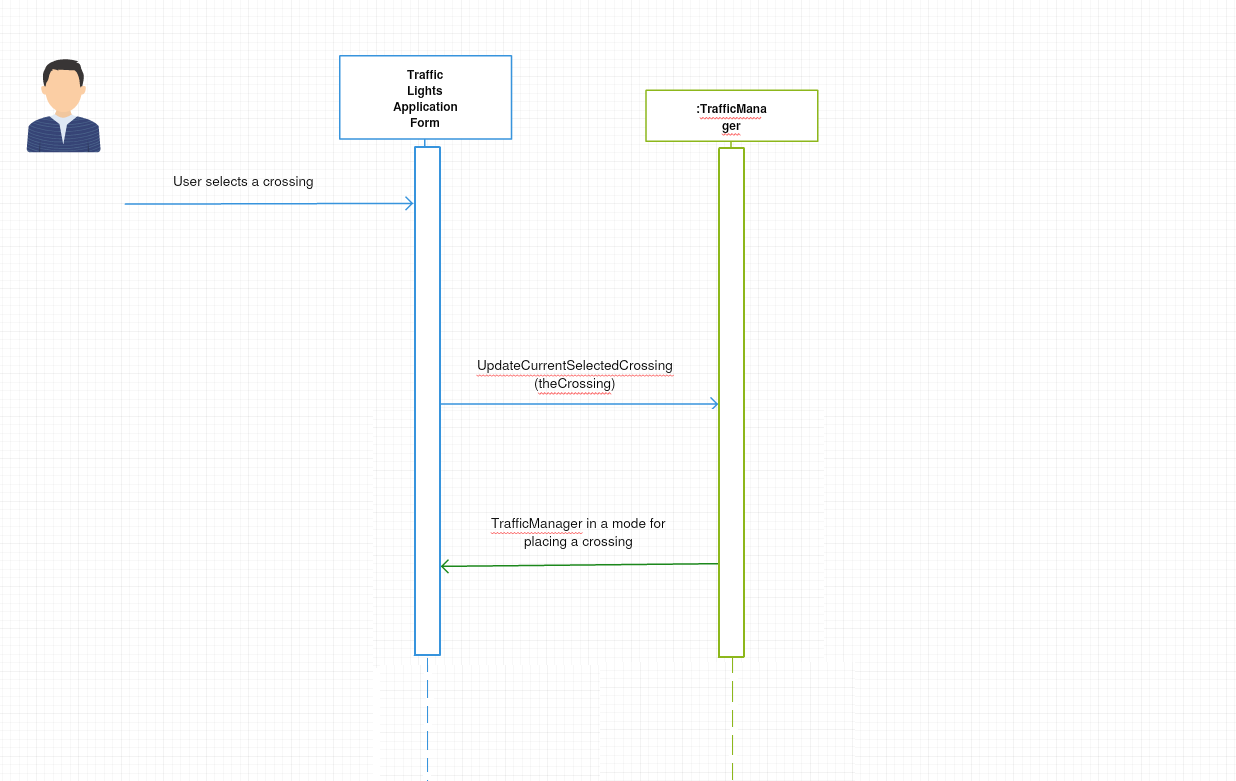
## Place a crossing



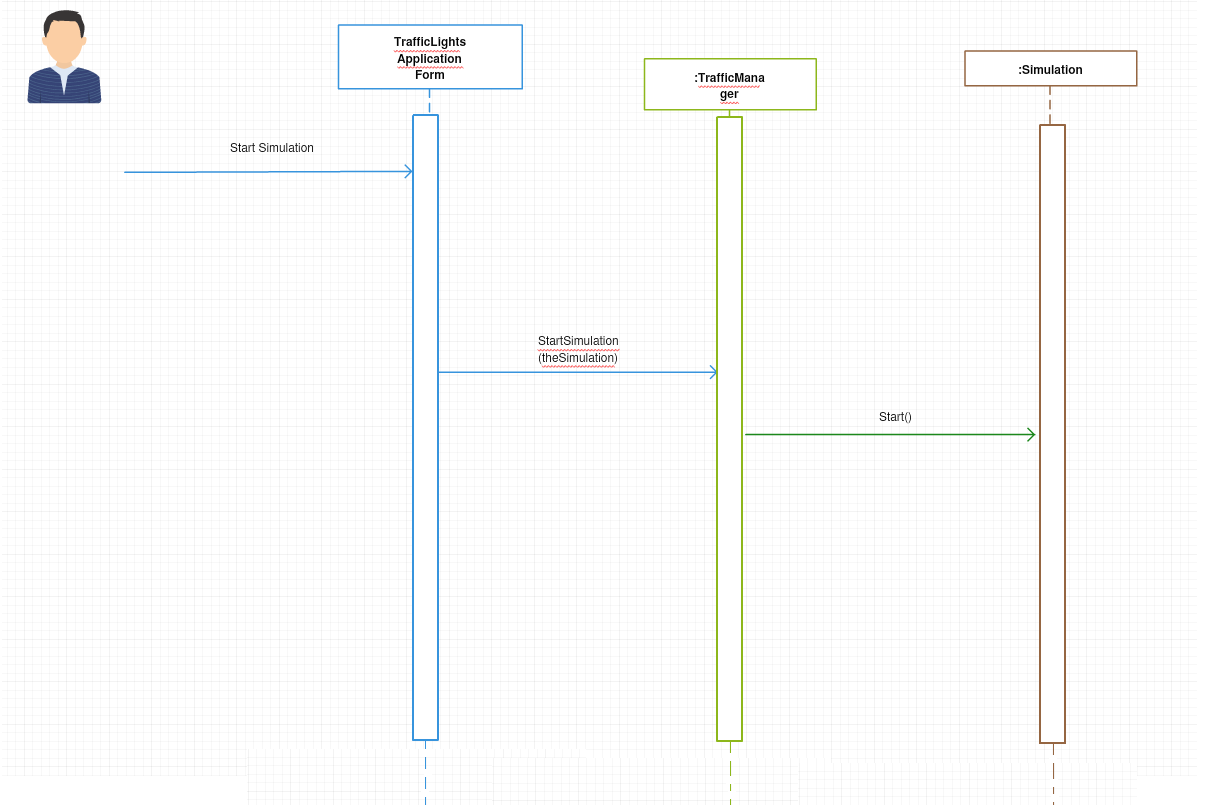
## Remove a crossing



## Select a crossing to place



## Start a simulation



## Stop a simulation

