|  |
| --- |
| FONTYS UNIVERSITY OF APPLIED SCIENCE |
| Design Document |
| Parcel Handling Simulation |
|  |
| **GDS - Group 3** |
| **9/13/2010** |

|  |  |  |
| --- | --- | --- |
| Document name |  | Design Document |
| Project name |  | Parcel Handling Simulation |
| Version |  | 1.0 |
| Department |  | ICT |
| Client |  | Casper Schellekens |
| School tutor |  | Peter Boots |

|  |  |  |
| --- | --- | --- |
| Group leader |  | Qian Li |
| Group members |  | Antoine Girard |
|  |  | John Ibeagha |
|  |  | Kritian Kolev |
|  |  | Sebastien Lepage |

**Document Change Control**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Related pages | Brief Description of Change |
| 1.0 | 2010-09-22 | All |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Review and Approval**

|  |  |  |  |
| --- | --- | --- | --- |
| **Review** | | | |
| Version | Date | Reviewed by | Position/Role |
| 1.0 |  | Casper Schellekens | School tutor |
| 2.0 |  | Casper Schellekens | School tutor |

|  |  |  |  |
| --- | --- | --- | --- |
| **Approval** | | | |
| Version | Date | Approved by | Position/Role |
| Final |  | Casper Schellekens | School tutor |

**Table of Contents**

[1. Introduction 1](#_Toc272962109)

[1.1 Purpose 1](#_Toc272962110)

[1.2 Users 1](#_Toc272962111)

[2. Project Design 2](#_Toc272962112)

[2.1 Class Diagram 2](#_Toc272962113)

# Introduction

## Purpose

This document represents the design and the architecture of the Parcel Handling Simulation project. The design is introduced in detail for the whole group to understand the whole architecture of the application.

## Scope

This document covers the class diagram for the Parcel Handling Simulation application and a clear description of all the methods used in our design. Specifically, this design covers the following aspects:

* UML modelling;
* Custom User Interface components;
* C# programming

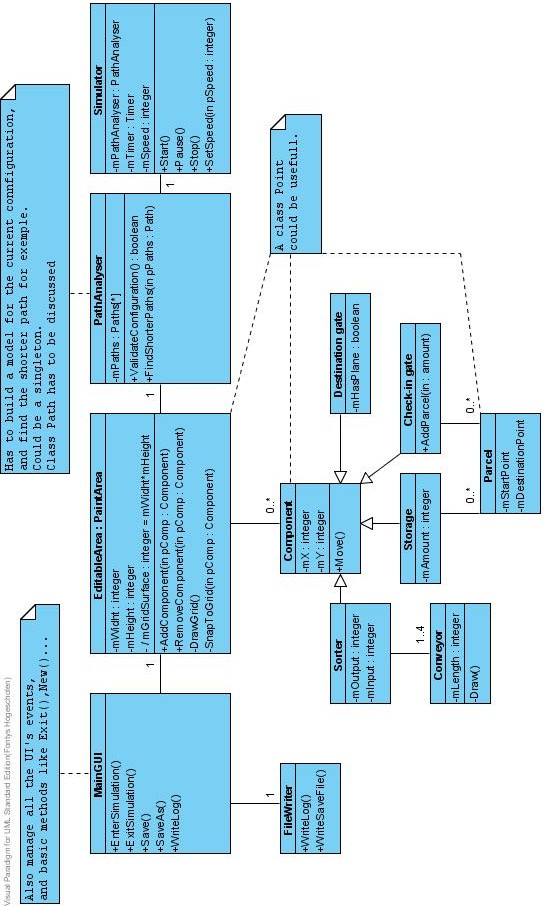
## Users

This document is intended to act as a technical tool for software engineers involved in the development of this project. For better understanding of our application, the knowledge required is:

* UML – Unified Modeling Language
* OOD – Object-oriented Design
* C# programming language
* .NET Remoting

# Project Design

## Class Diagram



**Class MainGUI:**

* EnterSimulation()
  + [Editor mode] Switches the client application to simulation mode via Simulator.Start()
* ExitSimulation()
  + [Simulation mode] Switches the client application back to editor mode via Simulator.Stop()
* Save()
  + [Editor mode] Saves the current simulation layout.
* SaveAs()
  + [Editor mode] Saves the current simulation layout, always providing the user a dialog box to choose file name and location
* WriteLog()
  + Handles the logging system, records a log of all user actions (check facts)

**Class FileWriter**

* WriteLog()
  + Writes to the log file
* WriteSaveFile()
  + Writes the simulation layout data to the selected file

**Class EditableArea**

* AddComponent(Component pComp)
  + Links the provided component to the selected simulation grid cell
* RemoveComponent(Component pComp)
  + Removes the provided component from the simulation grid
* DrawGrid()
  + Draws the simulation grid lines
* SnapToGrid(Component pComp)
  + Snaps the provided component to its closes simulation grid cell

**Class PathAnalyser**

* ValidateConfiguration()
  + Basic sanity check to ensure there's a path between each check-in desk and its destination gates
* FindShorterPaths(Path pPaths)
  + Finds the shortest paths from each check-in desk to its destination gates and storage facilities

**Class Simulator**

* Start()
  + Starts the simulation
* Pause()
  + Pauses the simulation
* Stop()
  + Stops the simulation and returns it to its starting state
* SetSpeed(int pSpeed)
  + Adjusts the speed of the simulation. Immediate effects.

**Component**

* Move()
* Calculates and modifies the component's next location based on the simulation layout and settings

**Conveyor**

* Draw()
* Redraws the conveyor belt section

**CheckInGate**

* AddParcel(int amount)
* Generates a new parcel according to the check-in gate preferences and puts it in the simulation system