Ekupov, Velin

Abstract

This document contains the functional and non-functional requirements of the BHS Simulation Software. It is intended for the client and outlines the operation of the solution from a user’s standpoint.

User requirements specification

Version 1.0

# Introduction

## Version description:

Extended functional and non-functional requirements. Added Use Cases for better illustration of the simulation process.

## Overview

The goal of the project is to create a simulation software which will administer the complete process of luggage handling in any given airport. The product should be built in conformance to certain requirements and provide meaningful statistics which will help with optimizing resource allocation. Prodavam Metla should provide a single application, which should be able to:

* Calculate performance statistics based on a simulation model and settings
* Customize the settings in order to collect extensive data
* Load/Save statistics and model settings

Once the main requirements are met, Prodavam Metla will try to further extend the application toward better user experience and customizability.

# Project Scope

## In scope

* Simulate luggage processing.
* Configurable resources.
* Performance statistics.
* Visual overview of how the BHS processed baggage.

## Out of scope

* Handling passengers through airport.
* Scheduling airplane arrival and leave.

# Requirements

## Functional requirements

1. Configure simulation
   1. Enable users to configure a simulation. (e.g., configure the number of check-in spots, passengers, flights, luggage, etc.)
2. Statistics
   1. The simulation provides the means to determine the optimal resource allocation via statistics related to how the luggage has been processed.
3. Visual overview
   1. A visual representation of the BHS showing how the baggage travels from the check-in to its respective destination.
4. Save/Export data
   1. Enable users to export simulation results in an external file
5. Load/Import data
   1. Enable users to import a previously exported data from an external file

## Non-functional requirements

1. The application should run with reasonable performance
2. The UI should be intuitive and not confusing even for people less familiar with computers
3. Statistics should be relevant and detailed, however, not overwhelming.
4. The application is intended to run mainly on Windows platforms.
5. All possible exceptions should be prevented or, if not, nicely handled and inform the user

# Use Cases

## #1 Build Simulation

**Actor:** User

**Pre-condition(s):**

* Application has been started
* User is on “Simulation” screen

**Main Success Scenario:**

1. Application displays an empty window with elements on the side
2. User picks one or multiple elements of different types and places them
3. User builds an airport model

**Extensions:**

1. Wrong placement of an element

• User right - clicks on the wrongfully placed element to delete it

• Return to 2.

2. User wants to remove everything

• User clicks on the clear button for a fresh start

• Return to 2.

## #2 Run Simulation

**Actor:** User

**Pre-condition(s):**

* Application has been started
* User is on “Simulation” screen
* User has built a full model

**Main Success Scenario:**

1 User clicks on the Run button

**Extensions:**

1.1 User want to run a past model

* Go to Use Case #3

## #3Load Data

**Actor:** User

**Pre-condition(s):**

* Application has been started
* User is on “Administration” screen
* User has previously exported ( saved ) a model

**Main Success Scenario:**

1. Application displays the current model or empty window
2. User selects “Import Variables” button
3. Application displays a dialog, prompting the user to select a file
4. User selects and loads a file containing a past model
5. System reads the file and automatically builds the model on the screen

**Extensions:**

4.1 Unsupported file format

* A warning is displayed informing that the file format is not supported
* Return to 3.

4.2 File or data corrupted

* A warning is displayed informing that the file/data is corrupted
* Return to 3.

4.3 No file selected

* A warning is displayed, prompting the user to select a file
* Return to 3.

4.4 Operation is canceled

* Dialog is closed and user is returned to “Simulation” screen
* Return to 1.

## #4 Display Statistics

**Actors:** None

**Pre-condition(s):**

* Application has been started
* User is on “Statistics” screen
* At least one simulation has been started

**Main Success Scenario:**

1. Application displays calculated statistics to the user

## #5 Save Data

**Actors:** User

**Pre-condition(s):**

* Application has been started
* User is on “Simulation” screen
* At least one simulation has been built and run

**Main Success Scenario:**

1. User selects “Export Variables” button

2. Application displays a dialog, prompting the user to save a file

3. User saves file

4. Application returns to initial screen

**Extensions:**

3.1 User has no permissions to save at given location

* A warning is displayed informing the user that he has insufficient permissions
* Return to 3.

3.2 File already exists

* A warning is displayed informing the user that the file with that name already exists and asks if the user wants to overwrite it
* User selects to overwrite the file
* Go to 4.

3.2.1 User selects not to overwrite the file

* Return to 3.

# Use Case Diagram

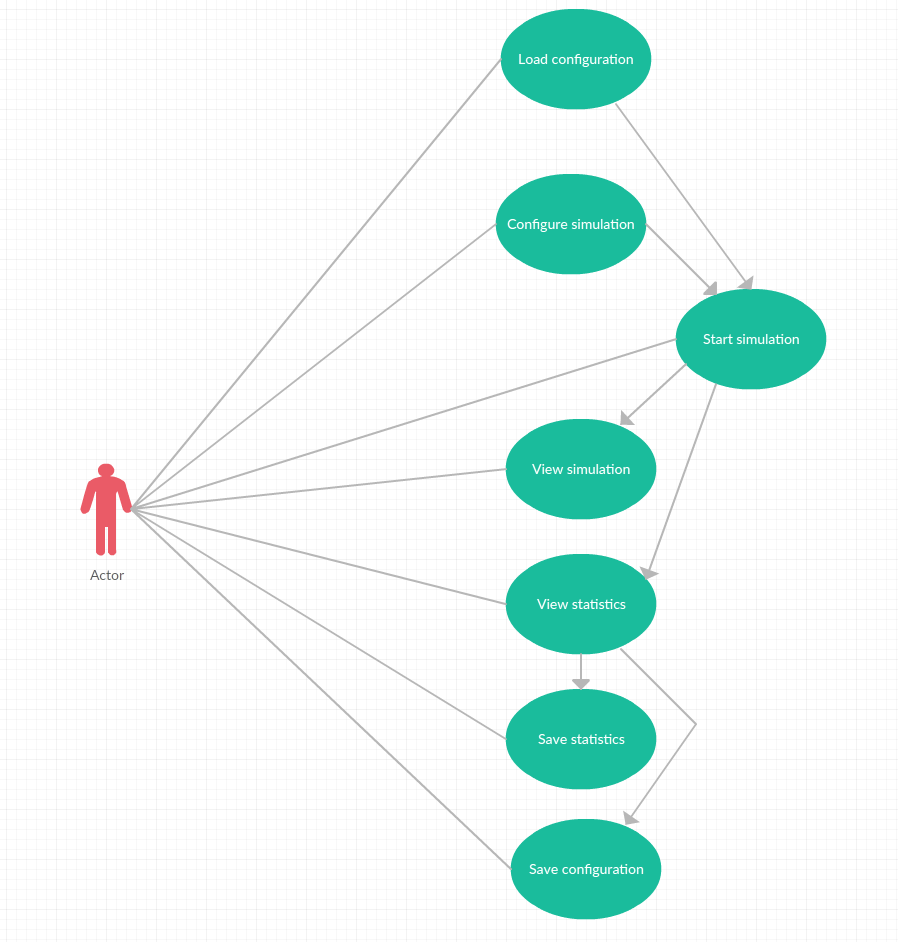


Figure Use case diagram. **Note: Arrows between use cases denote pre-conditions**