**Requirements Analysis Document**

*Application: <Application name>*

Date: February 16, 2021

**Index**

[**1. Introduction**](#_if4iqrppu03m) **3**

[1.1 Purpose of the system](#_3znysh7) 3

[1.2 Scope of the system](#_1qokf8y0r8dm) 3

[1.3 Objectives and success criteria of the project](#_fd7ehw4jr01g) 3

[1.4 Definitions, Acronyms, and abbreviations](#_a7dsx62tdzq1) 3

[**2. System Description**](#_1t3h5sf) **3**

[2.1 Functional Requirements](#_j3dv6r4fqt6e) 4

[2.1.1 Client](#_1oq7ej1s99f6) 4

[2.1.1.1 Purchase](#_zg2zzseb4prj) 4

[2.1.1.2 Reservation](#_t2wi09dvvmr7) 4

[2.1.1.3 Waiting list](#_26o6nig9aqr7) 4

[2.1.1.4 Passes](#_xkyeh1pl9m5c) 4

[2.1.2 Manager/Administrator](#_7n3d9j5wnui5) 4

[2.1.2.1 Configuration of areas (seat disabling)](#_x5ccc4x4d3d) 4

[2.1.2.2 Creation of events](#_w1a0hoomosrx) 4

[2.1.2.3 Scheduling performances](#_jnski538gf4h) 4

[2.2 Non-functional Requirements](#_tm1d0l4o0o0j) 5

[**3. Use Cases**](#_2s8eyo1) **5**

[3.1 Use Case diagram](#_17dp8vu) 6

[3.2 Use case descriptions](#_lxyonyennafz) 6

[3.2.1 Use Case <name>](#_8jqfqxdlsbix) 6

[**4. Mockups**](#_lnxbz9) **6**

# **1. Introduction**

## 1.1 Purpose of the system

The proposed system consists of a machine that can be accessed by administrators and clients of a theater hall.

The application, when entered by an administrator, can be used to register events, schedule performances and modify different parameters and configurations of the theater. It enables clients to purchase and reserve tickets for the already scheduled performances.

## 1.2 Scope of the system

Summary of what the system has to do and what not (required functionality).

## 1.3 Objectives and success criteria of the project

The system needs to meet the following requirements:

For the clients:

* A purchase and reservation mechanism.
* An algorithm to select a given number of available seats within an area.
* A waiting list for each performance.

For managers:

* The possibility of configuring several areas within the theater. Also, making capacity restrictions in events and the possibility to disable seats (unsold)\*\*\*
* A mechanism to configure new events, and schedule performances for those events.
* Managers can also configure passes: annual passes and cycle passes; users can make use of them to pay for their tickets.

## 1.4 Definitions, Acronyms, and abbreviations

**Event**: a show that can be performed multiple times in the theater. There are three types of events in the theater: Musical Concert, Dance and Theater Play.

**Performance**: it is the representation of an event in a single date and time.

**Cycle**:

**Area**: a part of the theater. It can be simple (either sitting or standing area, with a given capacity), or composite (made up of other areas). For every area of the theater, all performances of the same event have the same prices.

# **2. System Description**

## 2.1 Functional Requirements

Types of users and description of the functions each one of them can perform. This is the main part of the document, so please write a detailed description of each required functionality, enough for someone reading this part to completely understand what the system has to accomplish.

### 2.1.1 Client

#### 2.1.1.1 Purchase

The client can purchase for a performance, and area, a certain amount of tickets (block of tickets - maximum number of tickets set by the manager). The process is the following: the client has to log in, search for an event, select the performance they are interested in, select an area and number of tickets to be bought. If the selected area is a sitting area, then choose between manual and automatic selection of the seats.

An algorithm has to be implemented for this automatic selection, for which the client can choose one of the following heuristics: centered in rows and columns, centered in lower rows, centered in upper rows, farthest from all sold seats.

The payment can be done with a credit card (extern service), or with a [pass](#_xkyeh1pl9m5c).

Finally, a pdf file has to be generated, with an authentication code that is unique for each purchase. The ticket must contain: name of the show, time of performance, price, area, and seat if it is a sitting area.

#### 2.1.1.2 Reservation

…

#### 2.1.1.3 Waiting list

…

#### 2.1.1.4 Passes

…

### 2.1.2 Manager/Administrator

#### 2.1.2.1 Configuration of areas (seat disabling)

…

#### 2.1.2.2 Creation of events

…

#### 2.1.2.3 Scheduling performances

…

## 2.2 Non-functional Requirements

Description of non-functional requirements, categorized by type (performance, reliability, etc. ) .

# **3. Use Cases**

## 3.1 Use Case diagram

The use case diagram of the system

## 3.2 Use case descriptions

Write a **minimum of 3 detailed use cases**, using the following template:

## 3.2.1 Use Case <name>

Primary Actor : <List of primary actors>

Stakeholders and Goals:

<List of Stakeholders and objectives>

Preconditions:

<Explains the state that the system must be in for the use case to be able to start>

Success guarantee (Post-conditions):

<List of conditions that will be true when the use cases ends successfully>

Main Success Scenario:

<List of steps that describe how the use case goal can be achieved and all related stakeholder interests can be satisfied>

Extensions (Alternative paths):

< Alternate and Exception Scenarios >

Special Requirements:

<List of non-functional requirements>

Technology and Data Variations List:

<List of different technological options for functionality in use>

Frequency:

<Estimated frequency of occurrence>

Open Issues:

<Open issues or issues not solved yet, to consider in a future application version >

# **4. Mockups**

Include mockup visualizations of the application user interface. The mockups should cover all the application functionality, and should include indications (e.g., clicking a button, selecting a menu item) on how to navigate between the different windows.