

**Software Requirements**

**Specification**

**for**

**Distributed computing system for managing the activity for ‘Street Art Utopia’**

**Version 1.0**

**Prepared by Căpriceană Andrei Vlăduț**

**Spinu Cristian**

**Ruxanda Gabriel**

**Vodită Ionuț**

**Prepared for university project**

**12.03.2017**

# Table of Contents

[Table of Contents 1](#_Toc477090253)

[Revision History 1](#_Toc477090254)

[1. Introduction 2](#_Toc477090255)

[1.1 Purpose 2](#_Toc477090256)

[1.2 Product Scope 2](#_Toc477090257)

[1.3 Intended Audience and Reading Suggestions 2](#_Toc477090258)

[1.4 Definitions, Acronyms, and Abbreviations 2](#_Toc477090259)

[1.5 References 2](#_Toc477090260)

[2. Overall Description 3](#_Toc477090261)

[2.1 Product Perspective 3](#_Toc477090262)

[2.2 Product Functions 3](#_Toc477090263)

[2.3 User Classes and Characteristics 3](#_Toc477090264)

[2.4 Operating Environment 3](#_Toc477090265)

[2.5 Design and Implementation Constraints 3](#_Toc477090266)

[2.6 User Documentation 3](#_Toc477090267)

[2.7 Assumptions and Dependencies 4](#_Toc477090268)

[3. External Interface Requirements 4](#_Toc477090269)

[3.1 User Interfaces 4](#_Toc477090270)

[3.2 Hardware Interfaces 4](#_Toc477090271)

[3.3 Software Interfaces 4](#_Toc477090272)

[3.4 Communications Interfaces 4](#_Toc477090273)

[4. System Features 5](#_Toc477090274)

[4.1 System Feature 1 5](#_Toc477090275)

[4.2 System Feature 2 (and so on) 5](#_Toc477090276)

[5. Other Nonfunctional Requirements 5](#_Toc477090277)

[5.1 Performance Requirements 5](#_Toc477090278)

[5.2 Safety Requirements 6](#_Toc477090279)

[5.3 Security Requirements 6](#_Toc477090280)

[5.4 Software Quality Attributes 6](#_Toc477090281)

[5.5 Business Rules 6](#_Toc477090282)

[6. Other Requirements 6](#_Toc477090283)

[Appendix A: Glossary 6](#_Toc477090284)

[Appendix B: Analysis Models 6](#_Toc477090285)

[Appendix C: To Be Determined List 7](#_Toc477090286)

# 

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Comments** | **Version** |
|  |  |  |  |
|  |  |  |  |
| Căpriceană Andrei Vlăduț  Spinu Cristian  Ruxanda Gabriel  Vodită Ionuț | 12.03.2017 | First Revision | 1 |
|  |  |  |  |

# 1. Introduction

## 1.1 Purpose

The purpose of this document is to define the system requirements for a distributed system

that manages the activities of an art-enthusiasts club. It will explain the features of the system, the interfaces of the system and what the system will do.

## 1.2 Product Scope

This software system will be a web application for an art club. The computer system includes a user interface (Web browser) a database server that will store information about the association activity, application server. The user only has access to the service through the user interface

Management system will enable members of the association membership fees payment and management of resources (eg. conference rooms, exhibition halls)

Depending on the type of authentication (admin / user), the user is allowed managing members, adding payments for dues, adding resources.

## 1.3 Intended Audience and Reading Suggestions

This document is intended for both the users of the application and the developers of the system.

## 1.4 Definitions, Acronyms, and Abbreviations

## 1.5 References

*IEEE.IEEE Std. 830-IEEE Recommended Practice for Software Requirements*

*Web Engineering The Discipline of Systematic Development of Web Applications* **Edited by Gerti Kappel, Birgit Proll, Siegfried Reich, ¨ Werner Retschitzegger.**

# 2. Overall Description

## 2.1 Product Perspective

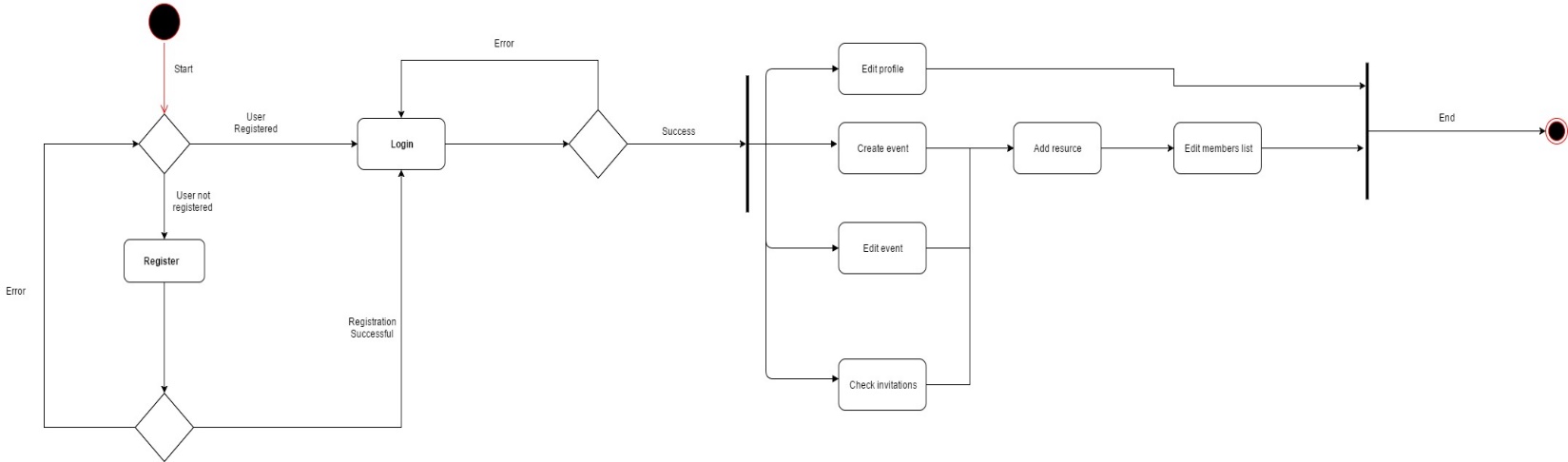
2.1.1

Activity diagrams

Figure 1 represents the actions that an admin can do in the application.

## C:\Users\andre\Downloads\Admin.jpg

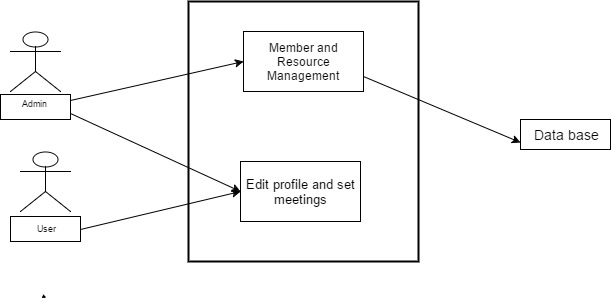
Figure 2 represents the actions that a user can do in the application.



2.1.2

Use-Cases

Figure 1 shows the use case diagram of the interaction between the four distinct users (user and admin) and the application.



## 2.2 Product Functions

The product will have the following main functions:

* New user registration
* New admin registration
* Set a profile and /or edit it
* Setting up conferences
* Setting up art exhibitions
* Managing payment
* Managing membership status
* Generating reports

## 2.3 User Classes and Characteristics

The application users will be art lovers which will have regular accounts and some of them will have administrator privileges chosen by the majority of them.

## 2.4 Operating Environment

The application supports most of the web browsers. (Google Chrome, Mozilla Firefox, Opera, Safari etc.)

## 2.5 Design and Implementation Constraints

Backend language: ASP.NET

Frontend: HTML, CSS, JavaScript

Framework: Bootstrap

## 2.6 User Documentation

The application will be released together with the Software Requirements Specifications documents and the testing documents.

## 2.7 Assumptions and Dependencies

*The application will be fully available only to those who have an account, and will be password protected. If the data is not introduced correctly, an error message will pop up.*

# 3. External Interface Requirements

## 3.1 User Interfaces

*<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface*

## 3.2 Hardware Interfaces

*Not Applicable.*

## 3.3 Software Interfaces

*This product’s development requires support of the following products: mySql, JDK and HTML editors, Windows Operating System.*

## 3.4 Communications Interfaces

*Client (system user) on Internet will be using HTTP/HTTPS protocol.*

# 4. System Features

## 4.1 System Feature 1

|  |  |
| --- | --- |
| 1. | Access to the application must be protected with user and password. |
| 2. | In making the communication protocol will take into consideration the possibility to connect more users simultaneously. |
| 3. | Administrators will be able to add / edit members pay dues adding, modifying the available resources, defining payment amounts for subscriptions. |
| 4. | Users will be able to create / edit events and invited select members, reserving resources for the event, where they were invited to view events |
| 5. | For each resource registered will be retained when available and when it is already reserved. |
| 6. | A resource can be booked for an event only if the period of the event is not reserved for another event |
| 7. | For each new booking, the resource becomes unavailable for other reservations the day before the event and after the event. |
| 8. | Any event that requires a resource created will generate spending 200 Lei / day / resource for the club. |
| 9. | It will generate a report of amounts receivable from members. |
| 10. | It will generate an income and expenditure report for a given month. |
| 11. | It will generate a report with the resources available in a given timeframe. |
| 12. | It will generate a report for the calendar booking of certain resources. |
| 13. | A resource can be booked anytime by a user if 15. and 16. conditions are met |
| 14. | A resource can be reserved by a person who is not a member by paying a fee of 400 Lei / day |
| 15. | A resource can be booked anytime by an administrator, if not already occupied |
| 16. | We can no longer make any reservation of resources in a given month by members if the amount of expenditure exceeds revenues. |

## 4.2 System Feature 2 (and so on)

# 5. Other Nonfunctional Requirements

## 5.1 Performance Requirements

*<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>*

## 5.2 Safety Requirements

*Not Applicable.*

## 5.3 Security Requirements

*Users Profile are public.*

*Only the payment status is private knew only by the user and the admins.*

## 5.4 Software Quality Attributes

*Rule 1 (required): Insert one space between the comment delimiter (//) and the comment text*

*Rule 2 (required): No identifier in one name space should have the same spelling as an identifier in another name space, with the exception of structure member and union member names.*

*Rule 3 (advisory): Write only one declaration per line.*

*Rule 4 (advisory): Add at least one blank line between method definitions and property definitions.*

*Rule 5 (advisory): Place the comment on a separate line, not at the end of a line of code.*

## 5.5 Business Rules

*Not Applicable.*

# 6. Other Requirements

*<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>*

# Appendix A: Glossary

# Appendix B: Analysis Model

# Appendix C: To Be Determined List