1. **Introduction**

This section gives a scope description and overview of everything included in this SRS document. Also, the purpose for this document is described and a list of abbreviations and definitions is provided.

**Purpose**

The purpose of this document is to give a detailed description of the requirements for the “Club of art enthusiasts” software. It will illustrate the purpose and complete declaration for the development of system. It will also explain system constraints, interface and interactions with other external applications. This document is primarily intended to be proposed to a customer for its approval and a reference for developing the first version of the system for the development team.

**Scope**

The application will allow you to manage club activities for art enthusiasts. The information system includes a user interface (web browser), a database server that stores information about the association's activity, application server. The user has access to the service only through the user interface

The system will provide the possibility of managing the members of the association, paying the membership fees,

member and resource management (ex: Conference rooms, Exhibition halls).

Depending on the type of authentication (admin / user), the user will have permissions for managing members, adding payments for contributions, adding resources.

**Definitions, acronyms, and abbreviations**

|  |  |
| --- | --- |
| Term | Definition |
| User | Someone who interacts with the mobile phone application |
| Admin/Administrator | System administrator who is given specific permission for managing and controlling the system |
| Web-Portal | A web application |
| DESC | Description |
| DEP | Dependency |
| TAG | A unique, persistent identifier |
| DEFINED | The official definition of a term |

**Overview**

The remainder of this document includes three chapters and appendixes. The second one provides an overview of the system functionality and system interaction with other systems. This chapter also introduces different types of stakeholders and their interaction with the system. Further, the chapter also mentions the system constraints and assumptions about the product.

The third chapter provides the requirements specification in detailed terms and a description of the different system interfaces. Different specification techniques are used in order to specify the requirements more precisely for different audiences.

The fourth chapter deals with the prioritization of the requirements. It includes a motivation for the chosen prioritization methods and discusses why other alternatives were not chosen.

**Overall description**

This section will give an overview of the whole system. The system will be explained in its context to show how the system interacts with other systems and introduce the basic functionality of it.

**2.1 User characteristics**

There are two types of users that interact with the system: users and administrators. Each of these two types of users has different use of the system so each of them has their own requirements.

The users are able to create an event and use a resource for it. This means that the user has to be able to search for a resource, and choose a resource. In order for the users to get a relevant search result there are multiple criteria the users can specify and all results match all of those.

The administrators are managing the overall system so there is no incorrect information within it. The administrator can manage the information for each resource and event.

**Specific requirements**

This section contains all of the functional and quality requirements of the system. It gives a detailed description of the system and all its features.

**3.1 External interface Requirements**

This section provides a detailed description of all inputs into and outputs from the system. It also gives a description of the hardware, software and communication interfaces and provides basic prototypes of the user interface.

**3.1.1 User interfaces**

A first-time user of the web application should see the log-in page when he/she opens the application. If the user has not registered, he/she should be able to do that on the log-in page or register page.

If the user is not a first-time user, he/she should be able to see the search page directly when the application is opened. Here the user chooses the type of search he/she wants to conduct.

Every user should have a profile page where they can edit their e-mail address, phone number and password.

**Functional requirements**

This section includes the requirements that specify all the fundamental actions of the software system.



Register

Login

Create/edit events



Add/edit members

Booking resource

user

admin

Invite members

View events

Mark fee as paid

Change available resources

Define fee

Pay fee - subscription

**User Class 1 - The User**

***Functional requirement 1.3***

**ID: FR3**

TITLE: User registration – web application

DESC: The user must provide user-name, password and e-mail address. The user can choose to provide a regularly used phone number.

RAT: In order for a user to register on the application.

***Functional requirement 1.4***

**ID: FR4**

TITLE: User log-in – web application

DESC: Given that a user has registered, then the user should be able to log in to the application. The log-in information will be stored on the pc and in the future the user should be logged in automatically.

RAT: In order for a user to register on the application.

***Functional requirement 1.5***

**ID: FR6**

TITLE: Search by price - web application

DESC: A user should be able to input a maximum and a minimum price range. The result is displayed in a grid by default.

RAT: In order for a user to search by price.

**User Class 3 - Administrator**

***3.2.3.1 Functional requirement 3.1***

**ID: FR26**

**Feature: Administrator log in**

In order to administer the system an administrator should be logged in.

**Scenario: Successful log-in**

Given the administrator wants to log in when the administrator logs in with an administrator account then the administrator should be logged in as an administrator.

***3.2.3.2 Functional requirement 3.2***

**ID: FR27**

**Feature: Manage club resources**

**Scenario: Add a new resource**

Given the administrator is logged in

When the administrator creates a new resource

Then the new resource should be added to the list of resources

**Scenario: Editing an existing resource**

Given the administrator is logged in

When the administrator edits an existing resource

Then the resource should be updated in the list of resources

**Scenario: Delete a resource**

Given the administrator is logged in

When the administrator deletes a resource

Then the deleted resource should be removed from the list of resources

***3.2.3.3 Functional requirement 3.3***

**ID: FR28**

**Feature: Manage club information**

In order to manage club information

An administrator

Should be logged in to the web-portal

**Scenario: Add** club **information**

Given the administrator is logged in

When the administrator adds club information

Then the information should be added to the club

**Scenario: Delete** club **information**

Given the administrator is logged in

And information about a club exists

When the administrator deletes the information

Then the information about the club should be deleted

**Scenario: Edit restaurant information**

Given the administrator is logged in

And information about a club exists

When the administrator edits the information

Then the information about the club should be edited

***3.2.3.4 Functional requirement 3.4***

**ID: FR29**

**Feature: Manage users**

In order to keep track of the users

An administrator

Should be able to manage the users

**Scenario: Edit an existing user’s information**

Given the administrator is logged in

When the administrator edits an existing user

Then the user information should be updated

**Scenario: Delete/Inactivate an existing user**

Given the administrator is logged in

When the administrator deletes an existing user

Then the user should be deleted

## Performance requirements

The requirements in this section provide a detailed specification of the user interaction with the software and measurements placed on the system performance.

#### Prominent search feature

**ID: QR1 – quality requirement**

TITLE: Prominent search feature

DESC: The search feature should be prominent and easy to find for the user.

RAT: In order to for a user to find the search feature easily.

DEP: none

#### Usage of the search feature

**ID: QR2**

TITLE: Usage of the search feature

DESC: The different search options should be evident, simple and easy to understand.

RAT: In order to for a user to perform a search easily.

DEP: none

#### Usage of the result in the grid view

**ID: QR3**

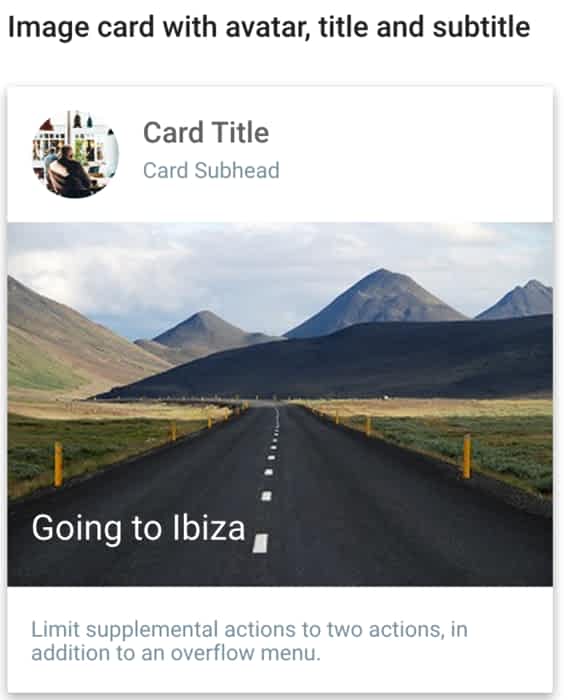
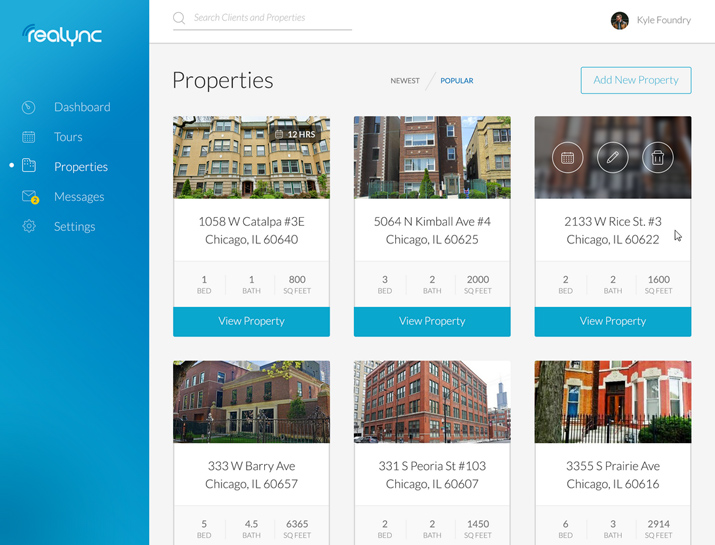
TITLE: Usage of the result in the grid view

DESC: The results displayed in the grid view should be user friendly and easy to understand. Selecting an element in the result grid should only take one click.

RAT: In order to for a user to use the list view easily.

DEP: none

Example:



#### Usage of the result in the map view

**ID: QR4**

TITLE: Usage of the result in the map view

DESC: A resource will be displayed as a pin on the map and the user will have the possibility to zoom in and out.

RAT: In order to for a user to use the map view easily.

DEP: none

#### Response time

**ID: QR5**

TAG: ResponseTime

GIST: The fastness of the search

SCALE: The response time of a search

METER: Measurements obtained from 1000 searches during testing. MUST: No more than 2 seconds 100% of the time.

WISH: No more than 1 second 100% of the time.

## Software system attributes

The requirements in this section specify the required reliability, availability, security and maintainability of the software system.

#### Reliability

**ID: QR6**

TAG: SystemReliability

GIST: The reliability of the system.

SCALE: The reliability that the system gives the right result on a search.

METER: Measurements obtained from 1000 searches during testing.

MUST: More than 98% of the searches. PLAN: More than 99% of the searches. WISH: 100% of the searches.

#### Availability

**ID: QR7**

TAG: SystemAvailability

GIST: The availability of the system when it is used.

SCALE: The average system availability (not considering network failing). METER: Measurements obtained from 1000 hours of usage during testing. MUST: More than 98% of the time.

PLAN: More than 99% of the time. WISH: 100% of the time.

**ID: QR8**

TITLE: Internet Connection

DESC: The application should be connected to the Internet.

RAT: In order for the application to communicate with the database.

DEP: none

#### Security

**ID: QR9**

TAG: CommunicationSecurity

GIST: Security of the communication between the system and server.

SCALE: The messages should be encrypted for log-in communications, so others cannot get user-name and password from those messages.

METER: Attempts to get user-name and password through obtained messages on 1000 log-in session during testing.

MUST: 100% of the Communication Messages in the communication of a log-in session should be encrypted.

Communication Messages: Defined: Every exchanged of information between client and server.

**ID: QR10**

TAG: UserLoginAccountSecurity

GIST: Security of accounts.

SCALE: If a user tries to log in to the web portal with a non-existing account then the user should not be logged in. The user should be notified about log-in failure. METER: 1000 attempts to log-in with a non-existing user account during testing.

MUST: 100% of the time.

**ID: QR11**

TAG: AdminLoginAccountSecurity GIST: Security of accounts.

SCALE: If an admin tries to log in to the web portal with a non-existing account then the admin should not be logged in. The admin should be notified about log-in failure.

METER: 1000 attempts to log-in with a non-existing user account during testing. MUST: 100% of the time.

**ID: QR12**

TAG: UserCreateAccountSecurity

GIST: The security of creating account for users of the system.

SCALE: If a user wants to create an account and the desired user name is occupied, the user should be asked to choose a different user name.

METER: Measurements obtained on 1000 hours of usage during testing. MUST: 100% of the time.

#### Maintainability

**ID: QR13**

TITLE: Application extendibility

DESC: The application should be easy to extend. The code should be written in a way that it favors implementation of new functions.

RAT: In order for future functions to be implemented easily to the application.

DEP: none

**ID:QR14**

TITLE: Application testability

DESC: Test environments should be built for the application to allow testing of the applications different functions.

RAT: In order to test the application.

DEP: none

#### Portability

**ID: QR15**

TITLE: Application portability

DESC: The application should run on Google Chrome and other web browsers.

RAT: The adaptable platform for the application to run on.