Meets Project

For the Meets Web Application

Version 1.1

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 24/10/2016 | 1.0 | Introduction and Overall Description added | Luca Carotenuto |
| 25/10/2016 | 1.1 | Specific Requirements (Functionality) added | Luca Carotenuto |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Introduction 4

1.1 Purpose 4

1.2 Scope 4

1.3 Definitions, Acronyms, and Abbreviations 4

1.4 References 4

1.5 Overview 4

2. Overall Description 4

3. Specific Requirements 5

3.1 Functionality 5

3.1.1 <Functional Requirement One> 5

3.2 Usability 5

3.2.1 <Usability Requirement One> 5

3.3 Reliability 5

3.3.1 <Reliability Requirement One> 6

3.4 Performance 6

3.4.1 <Performance Requirement One> 6

3.5 Supportability 6

3.5.1 <Supportability Requirement One> 6

3.6 Design Constraints 6

3.6.1 <Design Constraint One> 6

3.7 On-line User Documentation and Help System Requirements 6

3.8 Purchased Components 6

3.9 Interfaces 6

3.9.1 User Interfaces 7

3.9.2 Hardware Interfaces 7

3.9.3 Software Interfaces 7

3.9.4 Communications Interfaces 7

3.10 Licensing Requirements 7

3.11 Legal, Copyright, and Other Notices 7

3.12 Applicable Standards 7

4. Supporting Information 7

# Introduction

## Purpose

This document specifies the Software Requirements of the Meets project. It includes an overall description of the application as well as a section of specific requirements. The overall description indicates general requirements of the platform whereas the specific requirements go into further details of the provided functionality, design constraints and usability.

## Scope

The SRS applies to Meets, an event platform where the user can create, view and take part of events. It is a web based software application programmed in Java. Its scope is to be a mobile first app. Subsystems include a login system associated to a guest user, a meet-interacting system associated to a member of the platform and a meet-manipulating system solely associated to a meet owner.

## Definitions, Acronyms, and Abbreviations

* ***SRS*** - Software Requirements Specification

## References

tbd

## Overview

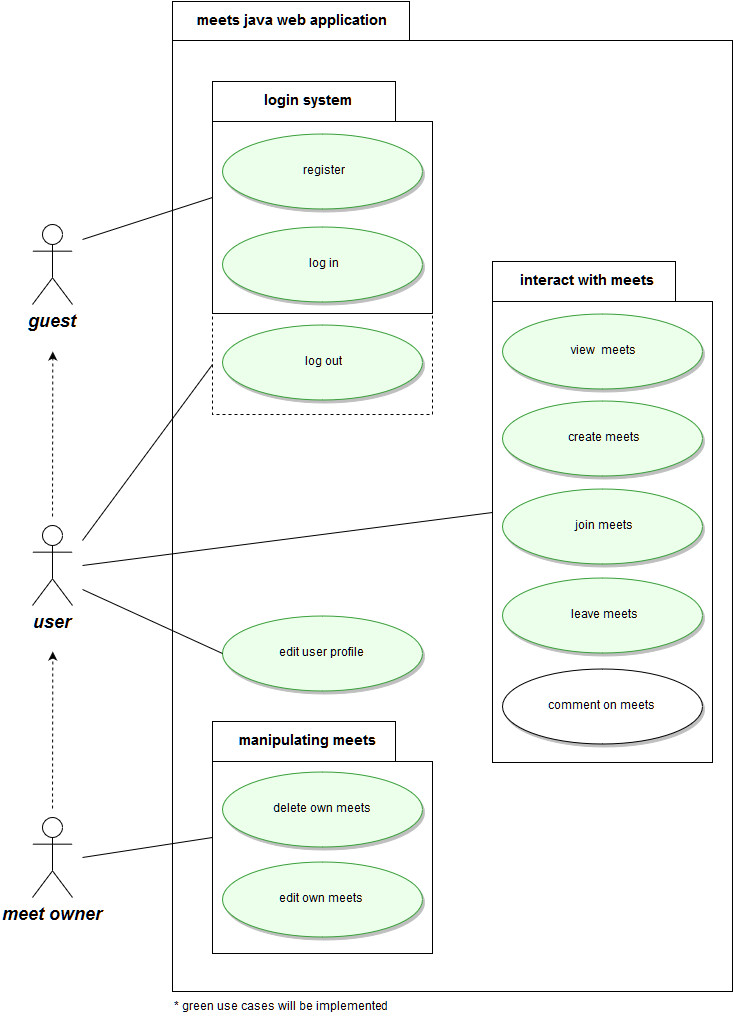
In the following an overall description of the requirements, functionality and constraints is given. Performance and reliability requirements are stated in the Specific Requirements section.

# Overall Description

The actions a user can take are highly dependent of its role in the system. The first time a user comes to the platform he as a guest is only able to register and login.

Users that have bypassed the login screen gain access to the system in the perspective of a member. Common use cases are ones that interact with meets like viewing, joining or leaving meets.

Certain functionality is only available for users that created meets, the so called meet owners. For them meet-manipulating are accessible by functions that allow deleting and editing existing meets.



# Specific Requirements

## Functionality

The use case diagram provides an overall overview of the provided functionality. There are 3 different subsystems that group the use cases in logical groups:

* + Login system
  + Interacting with meets
  + Manipulating meets

Each one of the actors guest, user and meet owner is capable of executing specific use cases of the subsystem. In this section each use case is described in detail.

* **Login System**

***Register***

Unregistered uest users first need to register to the system in order to be able to log in to the system.

***Log in***

Registered users are able to log in to the system by providing their user name and password.

***Log out***

Logged in users can log out of the system.

* **Interacting with meets**

***View meets***

Members of the platform can view meets that have been created by themselves or by other members of the community.  
  
***Create meets***

Members of the platform can create their own meets.  
  
***Join meets***

Members of the platform can join existing meets.  
  
***Leave meets***

Users that have joined a meet can decide to leave the meet if they are unable to attend the meet or they happened to have changed their mind.  
  
***Comment on meets***

Members of the platform can comment on meets to ask questions eg.

* **Manipulating meets**

***Delete own meets***

Users that created a meet are provided with functionality to delete the meet.   
  
***Edit own meets***

Users that created a meet are provided with functionality to edit the meet.

## Usability

### Mobile first

### Intuitive usability

## Reliability

[Requirements for reliability of the system should be specified here. Some suggestions follow:

* Availability—specify the percentage of time available ( xx.xx%), hours of use, maintenance access, degraded mode operations, and so on.
* Mean Time Between Failures (MTBF) — this is usually specified in hours, but it could also be specified in terms of days, months or years.
* Mean Time To Repair (MTTR)—how long is the system allowed to be out of operation after it has failed?
* Accuracy—specifies precision (resolution) and accuracy (by some known standard) that is required in the system’s output.
* Maximum Bugs or Defect Rate—usually expressed in terms of bugs per thousand lines of code (bugs/KLOC) or bugs per function-point( bugs/function-point).
* Bugs or Defect Rate—categorized in terms of minor, significant, and critical bugs: the requirement(s) must define what is meant by a “critical” bug; for example, complete loss of data or a complete inability to use certain parts of the system’s functionality.]

### <Reliability Requirement One>

[The requirement description.]

## Performance

[The system’s performance characteristics are outlined in this section. Include specific response times. Where applicable, reference related Use Cases by name.

* Response time for a transaction (average, maximum)
* Throughput, for example, transactions per second
* Capacity, for example, the number of customers or transactions the system can accommodate
* Degradation modes (what is the acceptable mode of operation when the system has been degraded in some manner)
* Resource utilization, such as memory, disk, communications, and so forth.

### <Performance Requirement One>

[The requirement description goes here.]

## Supportability

[This section indicates any requirements that will enhance the supportability or maintainability of the system being built, including coding standards, naming conventions, class libraries, maintenance access, and maintenance utilities.]

### <Supportability Requirement One>

[The requirement description goes here.]

## Design Constraints

[This section indicates any design constraints on the system being built. Design constraints represent design decisions that have been mandated and must be adhered to. Examples include software languages, software process requirements, prescribed use of developmental tools, architectural and design constraints, purchased components, class libraries, and so on.]

### <Design Constraint One>

[The requirement description goes here.]

## On-line User Documentation and Help System Requirements

[Describes the requirements, if any, for o-line user documentation, help systems, help about notices, and so forth.]

## Purchased Components

[This section describes any purchased components to be used with the system, any applicable licensing or usage restrictions, and any associated compatibility and interoperability or interface standards.]

## Interfaces

[This section defines the interfaces that must be supported by the application. It should contain adequate specificity, protocols, ports and logical addresses, and the like, so that the software can be developed and verified against the interface requirements.]

### User Interfaces

[Describe the user interfaces that are to be implemented by the software.]

### Hardware Interfaces

[This section defines any hardware interfaces that are to be supported by the software, including logical structure, physical addresses, expected behavior, and so on.]

### Software Interfaces

[This section describes software interfaces to other components of the software system. These may be purchased components, components reused from another application or components being developed for subsystems outside of the scope of this **SRS** but with which this software application must interact.]

### Communications Interfaces

[Describe any communications interfaces to other systems or devices such as local area networks, remote serial devices, and so forth.]

## Licensing Requirements

[Defines any licensing enforcement requirements or other usage restriction requirements that are to be exhibited by the software.]

## Legal, Copyright, and Other Notices

[This section describes any necessary legal disclaimers, warranties, copyright notices, patent notices, wordmark, trademark, or logo compliance issues for the software.]

## Applicable Standards

[This section describes by reference any applicable standard and the specific sections of any such standards which apply to the system being described. For example, this could include legal, quality and regulatory standards, industry standards for usability, interoperability, internationalization, operating system compliance, and so forth.]

# Supporting Information

[The supporting information makes the **SRS** easier to use. It includes:

* Table of contents
* Index
* Appendices

These may include use-case storyboards or user-interface prototypes. When appendices are included, the **SRS** should explicitly state whether or not the appendices are to be considered part of the requirements.]