Bogdan Rogoz

Watch2Gether
Supplementary Specification

Version 1.0

Revision History

Date	Version	Description	Author
17/03/2018	1.0	Initial documentation	Rogoz Bogdan

Table of Contents

1. Introduction

This document contains the non-functional requirements for the Watch2Gether project. That is, information regarding the application’s response time, availability and other important aspects are covered. This information is presented in order to ensure a proper functioning and a steady development cycle.

2. Non-functional Requirements

2.1 Availability

The application should have at least a 90% availability rate. It should not be severely impacted by external factors, such as power outages, unless they have a duration longer than 5-10 minutes. Since media uploading tasks are done concurrently with the streaming, they should not impact downtime. In case of an important update, it should be applied as soon as possible, at any moment of the day.

2.2 Performance

Website elements should load as fast as possible. A good threshold point would be 5 seconds. Although the connection speed plays an important role, functional modules should be as light as possible, in order to minimize loading times. Streaming efficiency is outside the scope of this project.

2.3 Security

The hosting website should present a certificate to the user, in order to make use of the HTTPS technology, so the communication channels can be automatically encrypted by the browser. Data sent through the log-in system must also be encrypted in order to prevent attackers from “sniffing” the data and use it for their own purposes. Each room should also be protected by a temporary key, which is valid until the room is no longer in use.

2.4 Testability

The application should consist of several small modules, each one having its own test suite. This way, during development, it is much easier to reduce the number of possible culprits in case of an error.

2.5 Usability

The application’s user interface should consist of a limited number of graphic elements, the more important / accessed ones being larger, so the user can easily spot them. This way, the interface is kept as minimalistic and intuitive as possible.
The interface should be in English, with multi-language support coming in future updates.

3. Design Constraints

The project will be a web application having a client – server architecture. The server side will be implemented in Java, while the client will be implemented in HTML5, CSS3 and Javascript.