

University of Pretoria

Quality Requirements

COS 301 – Team Zeon

[Author name]

7-16-2014

Table of Contents

Introduction.....	2
Overview	2
Quality requirements.....	2
Performance	2
Reliability.....	3
Security	3
Scalability	3
Flexibility.....	3
Maintainability.....	4
Integrability.....	4
Usability.....	4
Cost	4
Quality Constraints	5
Transfer protocol.....	5
Cost of online hosting	5

Introduction

The following document serves as a list and explanation of the various quality requirements and standards specified for the Stream2Me project, as outlined by the clients and understood by the developers involved. This document also serves as a set of guidelines in determining whether or not the system performs according to the aforementioned standards and lists the various requirements that is expected of the system in order to provide quality service to the end-user.

Overview

The server must be able to accommodate for multiple clients accessing and/or using the connection at the same time. Integration of the client and server components must be performed in such a way that the performance of the system is not compromised.

The system must be secure and the transfer of media and other data must be kept as simple but effective as possible. No malevolent party or user should be able to gain access to information from another user that is not explicitly sent to him/her; nor should the information be corruptible or interceptable during transmission.

However, with many of the following quality requirements, there are associated constraints which could hamper or complicate the development process, and in some cases may not be possible to overcome with the current restrictions on the system.

Quality requirements

Performance

- The system must be able to send, receive and stream data in as close to real time as the transmission method allows.
- The system must be able to stream data via an Android mobile application.
- The system must be largely independent of additional software or operating system versions.
- The server must be able to accommodate for multiple clients accessing and/or using the connection at the same time. Integration of the client and server components must be performed in such a way that the performance of the system is not compromised.

- The system needs to be able to send, receive and stream data in as close to real time as the transmission medium (TCP) allows.
- The system must be able to stream data via an Android mobile application.
- The system must be largely independent of additional software or operating system versions.

Reliability

- If a device or computer goes offline or disconnects, the data streaming must be cancelled immediately but still be available for streaming once the device or computer re-establishes its connection.
- The system will not be functional unless there is a stable connection established with a server and, in turn, the Internet.

Security

- The system must be able to secure the transfer of data before and during transmission until it reaches its intended recipient.
- Data being streamed must not be corruptible or interceptable during its transmission.
- The system will make use of java's SSLFactorySockets (for both the desktop version as well as the android application) to ensure that the server is identified, validated authenticated aiding in the prevention of some malicious activities that may occur.
- The system must be secure and the transfer of media and other data must be kept as simple but effective as possible. No malevolent party or user should be able to gain access to information from another user that is not explicitly sent to him/her; nor should the information be corruptible or interceptable during transmission.

Scalability

- The system must be able to scale for multiple users both on the mobile and desktop application. It must be able to do so concurrently.
- Performance should, ideally, not be affected by the number of users on the system.
- Thought as the number of clients that are connected to the server increase, so will the performance of the network which with thus not be attributed to the application, but to physical architectural restriction.

Flexibility

- The system should be accessible from a mobile application interface as well as a desktop application interface.

- Both interfaces as well as the server that is necessary are written in java, and hence they can run on any machine provided that the machine in question java the necessary java development tools (JDK) in the case on the desktop version, and the necessary standard development tools (SDK) on a device running android for the mobile version.

Maintainability

- Both desktop and mobile versions of the application should be easy to maintain and not be co-dependent in any way.
- If the libraries and utilities were to be deprecated, the system is designed in such a way that allows the necessary changes to simply be made, recompiled and further distributed.

Integrability

- All layers of the application must integrate with the others, without the need for regular human attention/intervention to function as intended.
- The system architecture comprises of modulated facets that perform independent functionalities that do not depend on the other modules, and thus can be easily extended without the fear that a failure in one of the many modules might perpetuate into the other facets and even give rise to other unforeseen errors or interruptions of the systems functionality.

Usability

- The interface for both the Android and the Desktop application must be user-friendly and straightforward, preferably with help functionality and/or on-screen guidance.
- The interface is traditional of this type of product, being that it is a professional communication utility and so we have designed it in such a way that a user will be able to use it with little or no effort.

Cost

- The cost of the application largely depends on the users machine as well as the network traffic. Taking into account factors such as the processor that the user will be employing as well as the amount of traffic directed to the server.
- These will rise performance issues as the server has to stream images of various qualities and constant audio streams that are meant for specific if not all users from all users.
- Hence algorithms that will handle this overhead and optimise performance will be employed.

Quality Constraints

Transfer protocol

- The transfer protocol in use has diverted from UDP to TCP, which is more secure and is reliable, however it is much slower and thus packet transfer could be delayed depending on the size and type of media being streamed.
- The reason for this change is due to the poor quality of throughput provided when using UDP, and the difficulty in streaming audio and video data in sync with one another (i.e. video packets may be lost but audio may stream properly – the videos audio would appear to be poorly synchronised).

Cost of online hosting

- The cost of hosting a server on the Internet is often too high to accomplish without funding and the cost will be significantly larger if the application must be free-to-use.