MedPat: a Doctor-Patient platform

Version <1.0>

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 20/03/2018 | 1.0 | First rendition. | Turcu Lucian Andrei |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Introduction 4

2. Non-functional Requirements 4

2.1 Availability 4

2.2 Performance 4

2.3 Security 4

2.4 Testability 4

2.5 Usability 5

3. Design Constraints 5

# Introduction

This document presents the quality attributes of the system, along with other requirements such as operating systems and environments, compatibility requirements and design constraints.

# Non-functional Requirements

## Availability

The system will be available 24/7 to both the patients and the users. However, it is up to the doctor to manage his own schedule in which he or she is available to the patients.

## Performance

1. Simultaneous users

The number of users that the system supports at any given time highly depends on the hardware that the server is installed on and the provided internet connection. It should support around 500 users without any considerable hardware requirements.

1. Database access response time

The system shall provide access to the database with no more than 10 seconds latency.

1. Transaction response time

The system must be able to complete all transactions in 2 minutes.

## Security

All user passwords will be encrypted using the Spring Security 5 modules, which are primarily focused on OAuth2.

## Testability

The system will be tested using Junit tests and manual tests on a local database with dummy accounts.

## Usability

1. Android compliance

The application will comply to the Android design standards, including the Material UI design, to be in line with the current trends.

1. Ease-of-Use

The application will be designed to be easily used by a basic smartphone user, by having clearly named items and descriptions where necessary.

# Design Constraints

The main programming language used is Java and the application will be based on Spring Boot to provide a platform for designing a reliable client-server architecture. The database engine will be SQL. Dependencies will be managed using Maven.