Android Patient Tracket

Supplementary Specification

Version <1.0>

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <20/03/2018> | <1.0> | First vision of supplementary specification | Albert Erika-Timea |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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 4

3. Design Constraints 4

Supplementary Specification

# Introduction

The Supplementary Specification captures the system requirements that are not readily captured in the use cases of the use-case model. Such requirements include:

* Legal and regulatory requirements, including application standards.
* Quality attributes of the system to be built, including usability, reliability, performance and supportability requirements.
* Other requirements such as operating systems and environments, compatibility, requirements and design constraints.

This document covers all the non-functional requirements.

[The introduction of the **Supplementary Specification** provides an overview of the entire document.

The **Supplementary Specification** captures the system requirements that are not readily captured in the use cases of the use-case model. Such requirements include:

Legal and regulatory requirements, including application standards.

Quality attributes of the system to be built, including usability, reliability, performance, and supportability requirements.

Other requirements such as operating systems and environments, compatibility requirements, and design constraints.]

# Non-functional Requirements

*[Define system quality attributes in terms of scenarios according to the following template:*

* *Quality attribute definition*
* *Source of stimulus: the entity (human or another system) that generated the stimulus or event*
* *Stimulus: a condition that determines a reaction of the system*
* *Environment: the current condition of the system when the stimulus arrives*
* *Artifact: is a component that reacts to the stimulus. It may be the whole system or some pieces of it*
* *Response: the activity determined by the arrival of the stimulus*
* *Response measure: the quantifiable indication of the response*
* *Tactics*

*]*

## Availability

* Quality attribute definition: Availability refers to the time period when the system is usable and working. It can be represented as a given percentage of the predefined running period when the system is down. It is affected by system errors, apparent attacks etc. (by anything that can change the normal functioning of a system).
* Source of stimulus: the sources that can produce actions that reduce availability can be system errors, malicious attacks, design problems.
* Stimulus: timeout error, incorrect response, component failure

Referring to a source\* these can be categorized into 4 groups:

1. Omission (component fails to respond)
2. Crash (components keeps failing over time)
3. Timing (component doesn’t respond at the expected time)
4. Response (response sent is incorrect)

* Environment: In case continuous failures appear, he user can send a notification to the system administrator; in case responses seems to be influenced by an attack, the system shuts down.
* Artifact: in our case the resource, that needs to be highly available, is the dataset holding patient information, but also the entire system
* Response: notification, switch to degraded mode
* Response measure: how fast is the system repaired after notifying

## Performance

* Quality attribute definition: it defines the capability of how fast a system ca respond to a request in a given time interval. It can be measured by the time needed for response (latency) or by the number of requests completed in a given time period (throughput).
* Source of stimulus: it can be both internal (component failure, poor infrastructure) or external (hardware performance) - these are the ones that may lead to low performance, but every time the user interacts with the system it determines a reaction
* Stimulus: periodic events, sporadic events, stochastic events
* Environment: the system can operate in a normal mode (handling events with an expected performance) but also in an overload mode (too many requests for the software and hardware components to handle in time)
* Artifact: the system components should be available (hardware and software) as well as the entire system
* Response: in case of normal mode the response to the request should be the event processing, otherwise the level of service changes
* Response measure: it can be indicated by the latency, throughput, miss rate, data loss

## Security

* Quality attribute definition: security of a software is defined by the capability of it to protect itself from malicious actions and data loss or data deprecation caused by this.
* Source of stimulus: in normal case, the stimulus comes from an authorized user but it can also come from an unknown source that could perform a malicious attack on the system
* Stimulus: both an authorized user and an attacker are performing the same action: requesting changing data, displaying data, adding data, deleting data. The attacker may also want to change system behavior.
* Environment: the system can work in normal mode (authorized and identified users) or in emergency mode (unknown source)
* Artifact: system, dataset
* Response: in case of authorized user process events; in case of malicious attack restrict availability, block and store access
* Response Measure: probability of identifying and blocking attacker, percentage of legitimate access denied

## Testability

* Quality attribute definition: testability indicates how easy it is to create tests and execute them on the system to identify if it operates in the expected way.
* Source of stimulus: we can design unit tests, system testers and client acceptance testers, the user itself can test if everything is working as needed
* Stimulus: the execution of designed tests
* Environment: we have to test our system when designing it, in the meantime while developing (development time - the different components), and the whole system at the end of development
* Artifact: system or system component being tested
* Response: execute the planned test and store the result, evaluating if the result is not the one expected
* Response measure: how difficult it is to identify the fault, time to perform tests, time to prepare test environment.

## Usability

# Design Constraints

[This section needs to indicate 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.]