

Software Requirements Specification

Project: <Development Of A Biofeedback Application>

Phase: Requirements Specification

Authors:
Lukas Heeg
Jan Schmitt
Tim Schmitt
Kerim Turacan

Documentname: AB-BiofeedbackApplication-SRS-1

Version: 1.2

Creation date: 18.04.2020

File: BioFeedbackApplication SRS.pdf

Modifications – Document Status

Version	Status	Creation Date	Editor	Modifications
1.0	Planned	16.04.2020	Lukas Heeg	Initial Document
1.1	Under Construction	18.04.2020	Lukas Heeg	Major Additions
1.2	Presented	18.04.2020	Lukas Heeg	Minor Modifications

(Status: = planned, under construction, presented, accepted)

TABLE OF CONTENTS

Software Requirements Specification	1
1 Introduction	4
1.1 Purpose.....	4
1.2 Scope.....	4
1.3 Definition, Acronyms and Abbreviations	4
1.4 Overview.....	4
2 General Description	5
2.1 Product Perspective.....	5
2.2 Product Functions	5
2.3 User Characteristics	5
2.4 General Constraints.....	5
2.5 Assumptions and Dependencies	5
3 Specific Requirements	6
3.1 Functional Requirements	6
3.1.1 Storing the player's information	6
3.1.2 Choosing a game mode	6
3.1.3 Pulse measuring.....	6
3.1.4 Moving an object in different directions	6
3.1.5 Interaction with other game objects	6
3.1.6 Creating a comprehensible score.....	6
3.2 External Interface Requirements.....	7
3.2.1 User Interfaces.....	7
3.2.2 Hardware Interfaces	7
3.2.3 Software Interfaces.....	7
3.3 Performance Requirements	7
3.4 Design Constraints	7
3.4.1 Standards Compliance.....	7
3.4.2 Hardware Limitations.....	7
3.5 Software Systems Attributes	7
3.5.1 Simplicity	7
3.5.2 Comparable scores	7
3.5.3 Challenging game concept	7

1 Introduction

1.1 Purpose

The purpose of this document is to explicitly describe the functionality of the product. It precisely describes what the product is supposed to do and lists any constraints that the product must fulfill. It lists the inputs/outputs required by the product.

1.2 Scope

The biofeedback application will test the effects of auditive and visual reactions to the user's heartrate by letting him play stressful games while monitoring his pulse. As a result of multiple runs with different game settings a statement about auditive and visual influence on a person's activity may be made.

1.3 Definition, Acronyms and Abbreviations

GUI Graphical User Interface

1.4 Overview

The SRS will specify in detail the software requirements for the biofeedback application. Section 2 presents a general description of the biofeedback application and its relationship within the operating environment (computer and external peripherals and systems). A complete list and description of the product functions and features will be provided. The type of user and user characteristics will be discussed. This Section also provides a discussion of any general constraints imposed on the product and any assumptions that are made regarding the operating environment of the product.

Section 3 will detail the software requirements of the biofeedback application. The behavioral requirements of the biofeedback application and operating environment will be discussed. The external interface, the hardware interfaces, the software interfaces, and the communication interfaces of the product will be outlined. Performance requirements will be discussed in section 3. Included in this discussion are operational requirements, exception handling, and testing requirements. Design constraints concern for the design phase of the product development will be addressed. Section 3 also describes the following product attributes: availability, security, maintainability, transferability and portability.

2 General Description

2.1 Product Perspective

The biofeedback application will be based on Java. A pulse sensor will be needed as an additional input device.

2.2 Product Functions

The product will have the following functions due to the biofeedback experiment:

- Reading and recording data from the pulse sensor
- Storing information about the players

The game that will keep the user stressed during each run has the following functions:

- Converting the keyboard input into a visual move of an object on screen
- Creating a comprehensible score based on the user's actions in the game
- Choosing between three different modes how to handle with the user' current heartrate:
 - Counteracting to the user's stress
 - Making the user aware of his stress level by adjusting the game design to it
 - No reaction at all
- Being able to change several design elements as a reaction

2.3 User Characteristics

Due to the experimental purpose of the application there is no exclusive group of people who qualify as player. Therefore, the controls will be self-explanatory. There only must be one instructor who knows how to handle the pulse sensor so that correct input data is guaranteed.

2.4 General Constraints

The language used in the code, the GUI and all documents is English. A usage of Open Source packages is possible.

2.5 Assumptions and Dependencies

The software will be developed with eclipse and tested with Microsoft Windows 10. It should be able to run on any other device that supports the latest version of Java.

3 Specific Requirements

3.1 Functional Requirements

3.1.1 Storing the player's information

Purpose: As there will be some people taking part in this experiment, the storage of age, weight and name is necessary for the result.
Inputs: The user can type in the information via keyboard.
Operations: The information is written into a file by the software.

3.1.2 Choosing a game mode

Purpose: An instructor chooses a specific game mode for the player.
Inputs: Selection with a mouse.
Operations: The modes that were explained earlier in 2.2 are executed by changing colors of background and game objects, playing different music samples or showing different insertions.
Outputs: On screen and through speakers.

3.1.3 Pulse measuring

Purpose: The software can react to the user's pulse.
Inputs: The pulse sensor measures the user's pulse.
Operations: With the information gained the game reacts by changing its design depending on the mode it is currently running. Also, the pulse variation is recorded in a file.
Outputs: In one game mode there will be an indicator for the player to see his level of stress. At the end of the game there will be a graphical evaluation of the player's heartrate.

3.1.4 Moving an object in different directions

Purpose: An in-game action to avoid or aim at other objects.
Inputs: The player presses the arrow keys of the keyboard.
Outputs: A movement of the game object is visualized on screen.

3.1.5 Interaction with other game objects

Purpose: To create an appealing game the player has the possibility of different actions, for example to shoot several objects.
Inputs: The player presses the space bar of the keyboard.
Outputs: Depending on the type of interaction it can be visualized on screen.

3.1.6 Creating a comprehensible score

Purpose: Validating the effects of the different game modes.
Operations: During the game the software counts several moves and successful actions.
Outputs: The score can be shown as a number on screen.

3.2 External Interface Requirements

3.2.1 User Interfaces

The GUI will display the game and all the functions from 3.1 with an output on screen. Since there are only few self-explanatory controls, it will be easy to use.

3.2.2 Hardware Interfaces

The biofeedback application will be controlled by a typical mouse and keyboard and a pulse sensor.

3.2.3 Software Interfaces

Depending on the type of sensor, an application for evaluating the data will be required. At the time of version 1.2 it is not certain, which sensor will be used for that purpose.

3.3 Performance Requirements

There will be no noticeable delay while interacting with the GUI and recording data. Data from the pulse sensor will be called up every two seconds. Then the game will react immediately.

3.4 Design Constraints

3.4.1 Standards Compliance

The written code follows the principles of object oriented programming.

3.4.2 Hardware Limitations

The software is designed for every modern intel, AMD or equivalent processor architecture.

3.5 Software Systems Attributes

3.5.1 Simplicity

The product should be self-explanatory so that the players can focus on the game which is the intention of this experiment. This can be tested by checking if every possible game interaction is being used from the beginning.

3.5.2 Comparable scores

Since the possible statement about the biofeedback that is studied is based on them, it is very important that they reflect the players performance in the game. This can be tested by letting the same person play the game multiple times. If the scores are not similar at all, the score system might not be fair.

3.5.3 Challenging game concept

The game should not be too relaxing nor too challenging so that design changes in both directions can be used. If both cases appear in one run, it can be assumed that the game difficulty is ideal.