Choreography Analysis and Timing

Software Design Document

Team:

Noffar Levi

Yana Shayev

Raam Banin

Prof. Boaz Ben Moshe

Date: 12/20/2020

**TABLE OF CONTENTS**

**1. INTRODUCTION 3**

1.1 Purpose 3

1.2 Scope 3

1.3 Overview 3

**2. SYSTEM OVERVIEW 3**

**3. SYSTEM ARCHITECTURE 3**

3.1 Architectural Design 3

3.2 Decomposition Description 4

3.3 Design Rationale 4

**4. DATA DESIGN 4**

4.1 Data Description 4

4.2 Data Dictionary 4

**5. COMPONENT DESIGN 5**

**6. HUMAN INTERFACE DESIGN 5**

6.1 Overview of User Interface 5

6.2 Screen Images 6

6.3 Screen Objects and Actions 7

**7. REQUIREMENTS MATRIX 7**

1. **INTRODUCTION**
   1. **Purpose**

This software design document describes the architecture and system design of a choreography analysis and timing system and the precise implementation and details required to satisfy the requirements as specified in the Software Requirements Specifications (SRS). It is assumed that the reader read the SRS, since this document also defines the implementation details of the design behavior given the requirements within it.

* 1. **Scope**
  2. **Overview**

The design description defined in this document serves multiple purposes:

* To describe the functional structure, data and algorithms to be implemented
* To identify required system resources
* To be used to assess the impact of requirement changes
* To be used to verify compliance with requirements
* To aid in maintenance activity

1. **System Overview**

Today dancers of all ages and levels of expertise regularly use mental imagery to better the outcome of their performance. However, the trainees do not receive feedback on their results. This system will answer this need. The system grants a simple way to manage the practices and more. Functionality options will be choice of user, number of users, body position, number of points, exercise mode, folder for each user and show a log of user history.

1. **System Architecture** 
   1. **Architectural Design**

**Diagram

Description automatically generated**

**3.2**

**Decomposition Description**

**Diagram

Description automatically generated**

* 1. **Design Rationale**

The rationale for selection the architecture described in 3.1 section was to create an independent component for each subsystem this way the components will have a separate functionally and will reduce the resources cost. Using this architecture will improve the system modularity and makes it more flexible for changes as opposed to using a single component.

1. **Data Design**
   1. **Data Description**

The information will be transformed into class object entities and will be stored in a database. For the analysis and efficient search the database that will store the training details and results as collections and documents.

* 1. **Data Dictionary**

Entities list:

1. **Company** – Entity that represents the company of the main entities like : **Coach/researcher, Trainee**.
2. **Coach/Researcher** – Entity responsible of adding new trainee and creating the practice that the trainee will preform
3. **Trainee** – Entity that will contain the details of the trainee (fullname and id)
4. **Trainee file** – Entity that contains the history of the trainee’s training results
5. **Practice** – Entity that contains : practice details , break points and the trainee.
6. **Automatic Training** – Entity that contains the **Practice** details and audio file
7. **Manual Training** – Entity that contains the **Practice** details, connection to clicker and calculations
8. **Examination** – Entity that contains the type and the results of the examination
9. **Training Results**– Entity that contains the values of the training results and which trainee it is related to.
10. **Component Design**
    1. View component of trainee file : (**Access: [Coach, Researcher]**)  
       - Showing trainee details  
       - Show history of practice details + results for each practice
    2. Create component of trainee: (**Access: [Coach, Researcher]**)  
       - Add new trainee
    3. Create component of practice: (**Access: [Coach, Researcher]**)  
       - Create new practice details

-Edit existing practice

- Add/Remove trainee

* 1. Integration component for clicker: (**Access: [Trainee]**)  
     - Allow pulling results from clicker
  2. Querying component: (**Access: [Coach, Researcher]**)  
     - Allow efficient & fast search on data and trainees’ practice results  
     - Allow analyzing information trainees’ practice.

1. **Human Interface Design**
   1. **Overview of User Interface**

Trainee: Once logged in, she/he will have several options to choose

1. Practice mode (Automatic/Manual)
2. Number of trainees
3. Body position
4. Eye position
5. Break point times
6. Results of the current practice

Trainer: Once logged in, she/he will have several options to choose

1. Register a trainee
2. Remove a trainee
3. Log history of specific trainee

(include details of each specific practice)

1. Log history of all trainees

**6.2 Screen Images**

Diagram

Description automatically generated**Diagram

Description automatically generated**

* 1. **Screen Objects and Actions**

Trainer:

* Login Page: Will enter a general id that will be the same for all the trainers.
* Next Page: Click the option you wants.
* Register: Register a new trainee (only a trainer can do this a trainee that is not registered can not execute a practice).
* Get Log: The trainer will specify which trainees results to display.

And then the results will be exported to an excel sheet.

Trainee:

* Login Page: The trainee will enter their personal ID.
* Practice Details: The trainee will have to specify the details of the current exercise.
* Practice: When the trainee is ready she will click the start button and hear a beep that will indicate the beginning of the practice. If she needs to stop before the end she will click the stop button.
* Results: When the practice finishes a pop up with the results will automatically appear.

1. **Requirements Matrix**