

Software Design Description  
Heavid  
1.1  
April 21, 2021  
MCS Department, Biola University

## Revisions

Target Audience: Those interested in finding and creating workouts tailored to each muscle group, especially those with hindered gym access during the pandemic

Project Team Members: Luke Ebbinghaus, Misheel Bayarsaikhan, Keziah Bulseco, Vanessa Garcia

### Version Control History:

<b>Version</b>	<b>Primary Author(s)</b>	<b>Description of Version</b>	<b>Date Completed</b>
0.1	Luke Ebbinghaus, Misheel Bayarsaikhan, Keziah Bulseco, Vanessa Garcia	Initial team additions	2/16/21
0.2	Luke Ebbinghaus, Misheel Bayarsaikhan, Keziah Bulseco, Vanessa Garcia	Added to System Architecture Design and created ER Diagram	2/19/21
0.3	Luke Ebbinghaus, Misheel Bayarsaikhan, Keziah Bulseco, Vanessa Garcia	Created Requirements Traceability Matrix, user interface screens, and component descriptions	2/22/21
1.0	Luke Ebbinghaus	Final Revisions, added User Interface Description, Design Description	2/22/21
1.1	Luke Ebbinghaus	Updated Traceability matrix to better match requirement changes. Updated descriptions. Updated ER Diagram	4/21/21

## Table of Contents

1 INTRODUCTION	1
1.1 Project Overview	1
1.2 Design Overview	1
1.3 Requirements Traceability Matrix	1
2 SYSTEM ARCHITECTURAL DESIGN	3
2.1 Chosen System Architecture	3
2.1.1 2-Tier	3
2.1.2 Layered Architectural Design	3
2.2 Interface and Design Description	4
3 DETAILED DESCRIPTION OF COMPONENTS	4
3.1 Database	4
3.1.1 Entities	4
3.1.2 Business Rules	4
3.1.3 ER Diagram	4
3.2 Network Layer	4
3.3 Database Interface	5
3.4 Development Environment	5
3.5 Component Diagram	5
4 USER INTERFACE DESIGN	5
4.1 Screen Images	5
4.2 User Interface Description	6

# 1 INTRODUCTION

## 1.1 Project Overview

A health app with built-in workout management features. The primary function will be to create an interface where the user can find workouts relevant to specific body parts through an intuitive diagram. Additionally, the user will be able to create their own workout plans with an input of difficulty and targeted muscle groups. The user can then modify this plan as desired using the same diagram interface.

## 1.2 Design Overview

This application will be designed based on a two-tier system architecture which bridges a database and client applications (2.1.1). The database will be implemented as shown in the ER Diagram with four entities (3.1). In more detail, there will be a data layer, data access layer, business layer, and presentation layer (2.1.2). To accomplish component integration, the development environment will interface with a database interface which communicates over TCP/IP to the database server (3). The user interface will consist of a single screen with variations and a pop-up description window (4).

## 1.3 Requirements Traceability Matrix

Req. ID	Rel. ID	Functional Requirements				Status	Design Components	Non-functional requirements
		Main requirements	Sub requirements	Category	Description			
1	2.1.1.1	Overall Project	Scalable Window	Required	There will be a fixed minimum window size to preserve visibility.	Work in Progress	SRS Document SDD - User Interface Design	Performance, usability, availability
2	2.1.1.2		Main Window with Popup Functionality	Required	One main window with pop-up functionality and dynamic sizing	Work in Progress		Performance, usability, availability
3	2.1.1.3	Diagram	Clickable Diagram	Required	Clickable diagram of a person will show exercises depending on the muscle group chosen	Work in Progress		Performance, usability, availability
4	2.1.1.4		Scalable, Omnipresent Diagram	Required	Person diagram on all screens	Work in Progress		Performance, usability, availability
5	2.1.1.5	Exercises	Clickable exercises and shows description	Required	Listed exercises will be clickable with extra information available	Work in Progress		Performance, usability, availability
6	2.1.1.6		Displays current exercise-list on home screen	Required	Clickable working-list of exercises in the screen which shows the current exercises added by the user into their workout	Work in Progress		Performance, usability, availability
7	2.1.1.7	Workout	Add to / start workout button	Required	Add to / start workout functionality which transitions to a workout plan section	Work in Progress		Performance, usability, availability
8	2.1.1.8		Difficulty Filters	Required	Difficulty filters for the exercises will be selectable when a muscle group is selected	Work in Progress		Performance, usability, availability
1	2.1.2.1		Device Types	Expected	The software would be supported only on desktop devices. Mac and Linux, but not Windows	Work in Progress		Performance, usability, portability
2	2.1.2.2		Nature of Data	Expected	The data we would be working with would be the string data type names of classification of the exercises by the difficulty and muscle group(s) that the exercise falls under in MySQL	Work in Progress		Portability, Maintainability

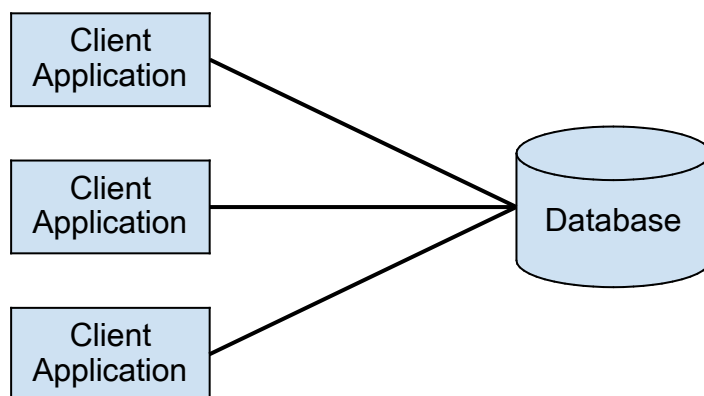
2	2.3.1.2	System Attributes	Security: User does not modify the database	Expected	Database cannot be changed by the user	Work in Progress	SRS Document	Security
1	2.3.2.1		Interoperability: Cross-Platform	Expected	The application will be tested to function in 3 different operating systems: Linux, Windows, and MacOS	Work in Progress		Interoperability, Portability
1	2.3.3.1		Performance: Changes done locally		Workout changes will be done locally	Work in Progress		Performance, security, maintainability
2	2.3.3.2		RequiredPerformance: Consistent database size and structure	Expected	Database will not change size unless explicitly updated	Work in Progress		Reliability
3	2.1.2.3	Hardware Functionality	Operating System Compatibility	Expected	The user would be using a desktop device to access the application which is created using C++ through QT	Work in Progress	SRS Document	Interoperability, Portability
4	2.1.2.4		Communication Protocol	Expected	The communication protocol that we will incorporate in our project is TCP to connect to the database	Work in Progress		Localization, security, portability, reliability
1	2.1.3.1	Software Interfaces	Application Development	Required	The main application will be implemented with C++14 standards	Work in Progress	SRS Document SDD - User Interface Design	Performance, Compatibility
2	2.1.3.2		UI Development	Required	The application will interface with Qt 6	Work in Progress		Performance, usability, compatibility
3	2.1.3.3		Database Development	Required	The application will interface with MariaDB and MySQL	Work in Progress		Performance, reliability, usability, security, compatibility
1	2.1.4.1	Communications Functions	GUI integration	Expected	Software application will be bundled with a graphical user interface by default	Work in Progress	SDD - System Architectural Design	Performance, usability
2	2.1.4.2		GUI and Application Interface	Required	Interface between application and GUI will be handled by Qt	Work in Progress		Performance
3	2.1.4.3		Network Server Database Driver	Required	Database will communicate with the software through MySQL Connector/C++ and TCP/IP	Work in Progress		Performance, usability, reliability, maintainability
4	2.1.4.4		Data Storage	Required	Data on exercises and their attributes will be stored in the database to be queried by the application	Work in Progress		Performance, reliability,
1	2.2.1	Software Product Features	Data Requirements	Expected	Database of variety of exercises based on different muscle groups and the difficulty of each exercise	Work in Progress	SRS Document	Performance, usability
2	2.2.2		Image integration	Required	Illustrations of each exercise for a better understanding of the motions (picture or animation)	Work in Progress		Performance, usability
3	2.2.3		Save functionality	Required	Local save functionality for workout plans	Work in Progress		Performance, reliability, maintainability, usability
1	2.3.1.1		Security: Local Saving	Expected	The saved workouts will be safely stored in the user's machine	Work in Progress		Performance, availability, security

1	2.4.1.1	Database Requirements	Type of Data	Required	String texts of exercise names, muscle groups	Work in Progress	SDD - Components - ER Diagram	Usability, availability
2	2.4.1.2			Expected	BLOB images of exercises	Work in Progress		Usability, availability
1	2.4.2.1		High-Level Entities	Expected	Exercise: A specific exercise	Work in Progress		Availability
2	2.4.2.2			Expected	Muscle Group: A targeted muscle group	Work in Progress		Availability
1	2.4.3.1		Business Rules	Required	Each exercise must work one or more muscle group(s)	Work in Progress		Performance, usability, availability
2	2.4.3.2			Required	A muscle group must target one or many exercises	Work in Progress		Performance, usability, availability

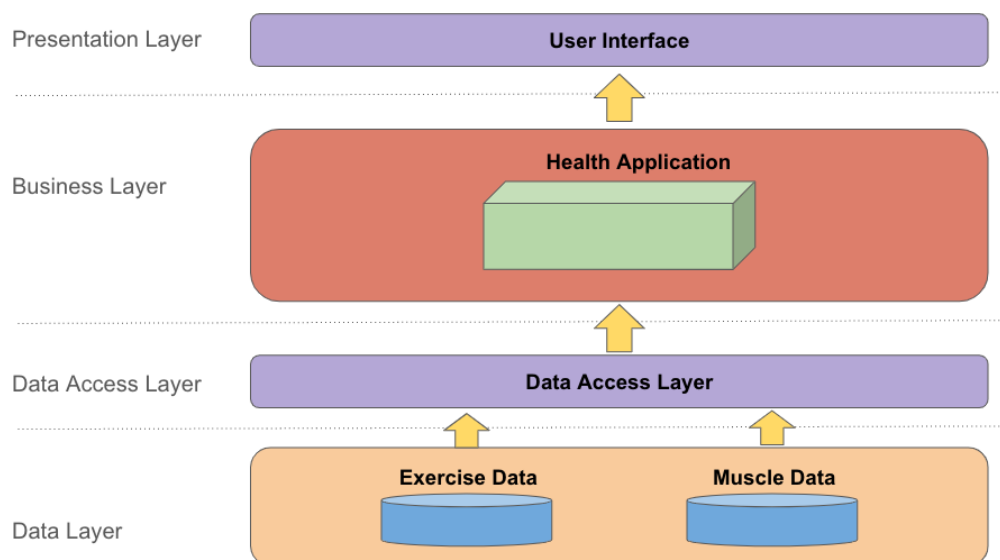
## 2 SYSTEM ARCHITECTURAL DESIGN

### 2.1 Chosen System Architecture

#### 2.1.1 2-Tier: One database connects to one or many client applications



#### 2.1.2 Layered Architectural Design:



## 2.2 Interface and Design Description

A two-tier system architecture will bridge one or many client applications to the MySQL database server. The data layer implemented in MySQL will interface with a data access layer through MySQL Connector/C++. The application, implemented in C++, will be stored on the client machine with a bundled user interface. This user interface will be designed in Qt.

## 3 DETAILED DESCRIPTION OF COMPONENTS

### 3.1 Database

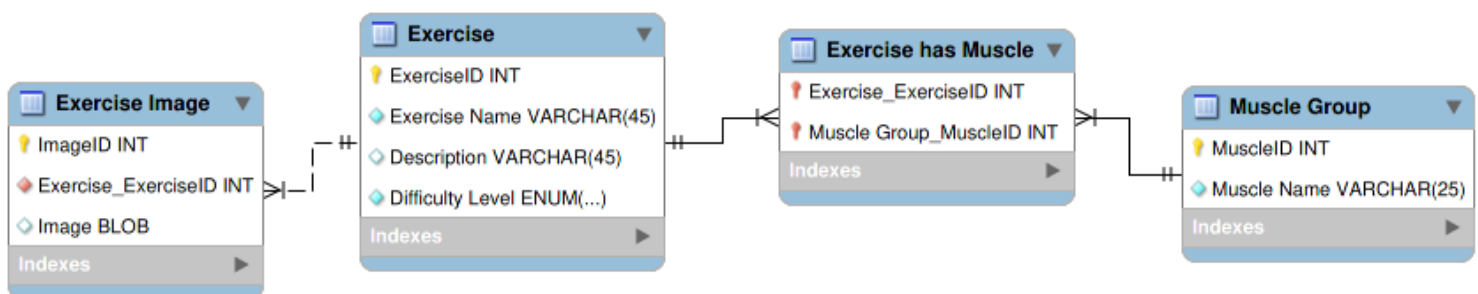
#### 3.1.1 Entities

- Exercise: Stores ID, Name of Exercise, Description of Exercise
- Muscle: Stores ID and Name of Muscle
- Exercise Image: Stores Image of Exercise

#### 3.1.2 Business Rules

- Each exercise must work for one or more muscle groups.
- A muscle group must target one or many exercises.
- Each exercise can have one or many images.
- Each image can correspond to only one exercise.

#### 3.1.3 ER Diagram



### 3.2 Network Layer

The Network layer consists of the TCP/IP connections required for the Database Interface tools to integrate with the database server.

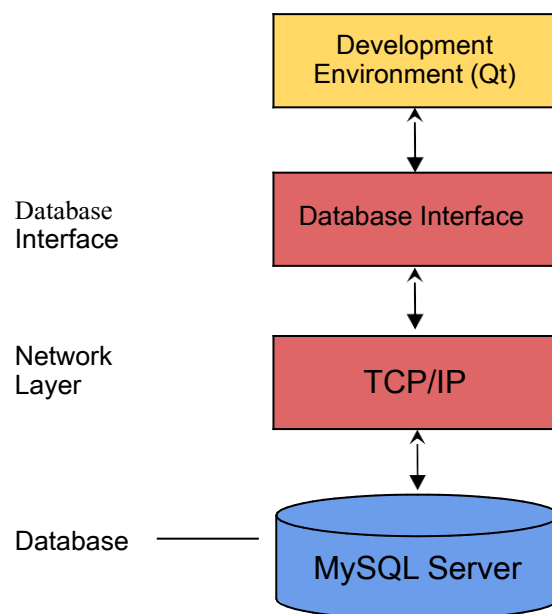
### 3.3 Database Interface

The database interface tools allow a connection to the database server through the use of a driver; we will use MySQL Connector/C++ to establish this connection.

### 3.4 Development Environment

Our development environment will include Qt Creator which integrates C++ programming with the UI design tools and graphics libraries implemented in Qt. This environment will include calls to MySQL Connector/C++ to establish a connection.

### 3.5 Component Diagram



## 4 USER INTERFACE DESIGN

### 4.1 User Interface Description

The user can navigate one screen with several variations and a pop-up for descriptions. Starting with a diagram of a person, the user can either navigate to a saved workout shown in the corner or hover the mouse over a muscle group of the diagram. When the latter is performed, the spot where the mouse is will highlight; when clicked, an exercise window will come up with relevant exercises for the muscle group chosen. Filters can be set for the difficulty, and a workout plan can be added to.



## 4.2 Screen Images

