Software requirement specification (SRS) document template

Project name:	MuscleFlow
Date:	19/09/2022
Version:	1.0
By:	Henri Eloy, Marius Ortega, Samuel Pariente, Aurélien Pouxviel

Revision history

Version	Author	Version description	Date completed
0.1	Marius Ortega	First commit to doc	27/09/2022

Review history

Reviewer	Version reviewed	Signature	Date
Samuel Pariente	0.1		30/09/2022
Henri Eloy	0.1		30/09/2022
Aurélien Pouxviel	0.1		30/09/2022

Approval history

Approver	Version approved	Signature	Date

Table of contents

1 Introduction

Describe the purpose of the document.

This document aims at explaining technical, design, hardware, and software wise requirement for our application to work. Reading this document will allow you to understand more holistically the processes, ideas, and standards that MuscleFlow will be built on.

1.1 Product scope

List the benefits, objectives, and goals of the product.

MuscleFlow aims at helping gym clients to perform better and being proud of their gym sessions. Our main objective with muscle flow is to let people create their training, make them keep their efforts up and go to the gym without any frustration or anxiety.

1.2 Product value

Describe how the audience will find value in the product.

Creating a training isn't always easy when you first start gym. In addition, keeping track of the exercises you should do/you have done and remembering the weights you should use for each of them can quickly get complicated if gym is only a side hobby. Our product can help users in each of these somewhat tedious tasks.

1.3 Intended audience

Write who the product is intended to serve.

As introduced earlier, this application can provide a structured environment for newcomers. However, our product can also be helpful for more advanced people. Indeed, the great variety of exercises of the application and visualization of one’s progress can be appreciated by anyone.

1.4 Intended use

Describe how will the intended audience use this product.

This application ought to be used 10 min before your gym session (to check daily goals, add or suppress exercises of the incoming session), during the said session (to validate the exercises that have been done) and a few minutes after (to see the performance during the session and how it compares to previous session).

1.5 General description

Give a summary of the functions the software would perform and the features to be included.

Our application is a gym session assistant. It allows you to keep track of your progress through summary graphics and achievements. It also contains a training editor to help you build your most fitted sport program. In addition, there is a calendar feature to schedule training and get reminded of them. Finally, MuscleFlow also increases or decreases your training difficulty through time and depending on your daily energy.

2 Functional requirements

List the design requirements, graphics requirements, operating system requirements, and constraints of the product.

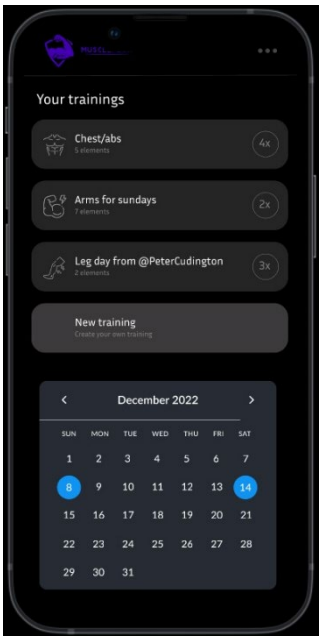
- Operating system requirements :

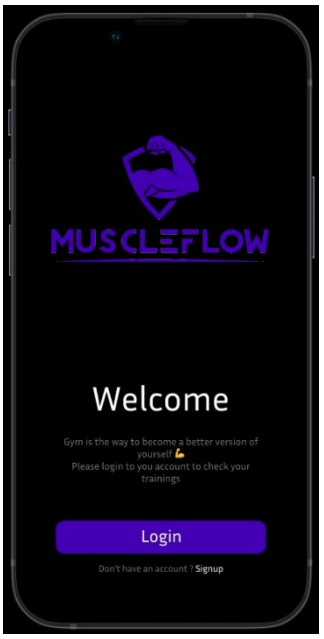
  Our application will require at least IOS 10 for apple users and Android O for other users.

- Design and Graphics requirements :

  MuscleFlow should be a straightforward and simple application to use.
  Thus, all pages must be minimalistic and practical.
  Buttons should be rounded and elegant.
  Every interface element must big easy to click on because the gym environment can make it difficult to click accurately.

  Here is a first draft to explain what is expected :





- Constraints of the product :

  Requires cellular data connection (3G, 4G, 5G) or Wi-Fi connection.
  Requires to be able to use hands and sufficient sight of the user.
  Requires at least an iPhone 5s or equivalent for other brands.
  Requires at least IOS 10 or Android O to function.

3 External interface requirements

3.1	**User interface requirements**	**Describe the logic behind the interactions between the users and the software (screen layouts, style guides, etc).**

As explained previously, our application is meant to be used in situations that don't allow the user to use accurately his smartphone accurately. Thus, the layouts are mainly going to incorporate relatively big buttons. The user should only perform touch, slide, or tap actions during his gym session. For the visual to be consistent, fonts will be slightly larger than usual (especially in the training features part of the application. In addition, the dark mode will be the one activates by default because it will emphasize the minimalist yet practical aspect of the application.

3.2	**Hardware interface requirements**	**List the supported devices the software is intended to run on, the network requirements, and the communication protocols to be used.**

The software is intended to function on iPhone 5s (and equivalent for other brands) up until the last released phone. Our application is Non-Real Time & Asymmetric and will use a best effort process to transmit information. Thus, we can't have any guarantee of the quality on the user-side delay because this protocol works on a "first come, first served" basis. Though, an ITU (International telecommunication Unit) suggests that the acceptable delay for an action is 0-150ms. We can use this information to determine what server capacity we would need.

3.3	**Software interface requirements**	**Include the connections between your product and other software components, including frontend/backend framework, libraries, etc.**

Firstly, there will be basic design libraries to get adequate color palettes, icons, or transitions. The second main interaction will be with the local host server during the development phase, and then with the server.

3.4	**Communication interface requirements**	**List any requirements for the communication programs your product will use, like emails or embedded forms.**

Our product will feature a chat so users can report any issue they encounter. They will also have the possibility to link their email address if they desire to receive email notification. Finally, there will be classical application’s notifications to remind the users of its future training. In later version of the application, we would like to include a chatbot and a hotline to improve the user experience.

4 Non-functional requirements

4.1 Security	**Include any privacy and data protection regulations that should be adhered to.**

To process users' data ethically, we will adhere to the GDPR standard that is based on 4 main principles. Finality principle, proportionality and relevance principle, limited time conservation principle and finally security and confidentiality principle.

4.2 Capacity	**Describe the current and future storage needs of your software.**

Currently (during its development phase), MuscleFlow only requires a database of 10 to 20Go maximum. This database allows us to store our training exercises database and user's information. In a possible future, if the application gets commercialized, the database's size should be increased.

4.3 Compatibility	**List the minimum hardware requirements for your software.**

Our application minimum hardware requirements are : 1.3 Ghz dual core processor, PowerVR G6430 GPU (or equivalent), 1Go random access memory, 16 Go storage disk.

4.4 Reliability	**Calculate what the critical failure time of your product would be under normal usage.**

There isn't a proper critical failure time strictly speaking. The only condition for our app to work is to have access to cellular data to consult the exercise's database and similar processes.

4.5 Scalability	**Calculate the highest workloads under which your software will still perform as expected.**

Given that our application is hosted on Hostinger with a 10,99€ per month subscription, our software can perform normally up until 25 000 monthly users.

4.6 Maintainability	**Describe how continuous integration should be used to deploy features and bug fixes quickly.**

Our application will integrate a chat enabling user to report bugs to our support. If the issue can be fixed the support will proceed. Otherwise, the issue is reported to the developer team to add a corrective in the next patch.

4.7 Usability	**Describe how easy it should be for end-users to use your software.**

The software should be extremely easy and quick to use. Indeed, it has to be accessible from the gym in a instance to validate exercises and to modify a training program a few minutes before a session.

4.8 Other	**List any additional non-functional requirements.**

No additional information to add.

5 Definitions and acronyms

GPU	Video random access memory, processor specialized for matrices.