<Project Name>

System-Wide Requirements Specification

Usage note: There is procedural guidance within this template that appears in a style named InfoBlue. This style has a hidden font attribute allowing you to toggle whether it is visible or hidden in this template. Use the Word menu Tools🡪Options🡪View🡪Hidden Text checkbox to toggle this setting. A similar option exists for printing Tools🡪Options🡪Print.

# Introduction

# System-Wide Functional Requirements

[Statement of system-wide functional requirements, not expressed as use cases. Examples include auditing, authentication, printing, reporting.]

# System Qualities

[Qualities represent the URPS in FURPS+ classification of supporting requirements.]

For creating a good system, we have to provide some qualities like usability, realiability, performance and supportability.

## Usability

[Describe requirements for qualities such as easy of use, easy of learning, usability standards and localization.]

* The user shall be access easily his/her account.
* The user shall be easily login and logout from the system.
* The user shall be access what he/she wants at most 3 clicks.
* The user shall be freely navigate through the system.
* The system shall be at most 4 deep tabs.
* The user shall be do changes easily into his/her system.

## Reliability

[Reliability includes the product and/or system's ability to keep running under stress and adverse conditions. Specify requirements for reliability acceptance levels, and how they will be measured and evaluated. Suggested topics are availability, frequency of severity of failures and recoverability.]

* The system must secure the informations of the users on any condition.
* The system must do the transactions well with database.
* The system shall have a good connection with DB.
* The system shall stay under control on a bad situation.
* When a user change something his/her account, system provides these changes at any time (durability).

## Performance

[The performance characteristics of the system should be outlined in this section. Examples are response time, throughput, capacity and startup or shutdown times.]

* The system should be handle many users at the same time.
* Login time shall be less than 4 seconds.
* Logout time shall be less than 4 seconds.
* The system’s response time shall be less than 5 seconds.
* The database connection time shall be less than 2 seconds.

## Supportability

[This section indicates any requirements that will enhance the supportability or maintainability of the system being built, including adaptability and upgrading, compatibility, configurability, scalability and requirements regarding system installation, level of support and maintenance.]

* We will perform maintenance of the system monthly.
* Any bug will be resolve immediately.
* Users can be reach our app via any mobile device.

# System Interfaces

[Interface Requirements are part of the + in the FURPS+ classification of supporting requirements. Define the interfaces that must be supported by the application. It should contain adequate specificity, protocols, ports and logical addresses, and so forth, so that the software can be developed and verified against the interface requirements.]

## User Interfaces

[Describe the user interfaces that are to be implemented by the software. The intention of this section is to state requirements relating to the interface. Interface design may overlap the requirements gathering process.]

User Interface should be user-friendly and easy to navigate through the system’s functions. Detailed User Interface is added in Appendix B Document.

### Look & Feel

[Provide a description of the spirit of the interface. Your client may have given you particular demands such as style, colors to be used, and degree of interaction and so on. This section captures the requirements for the interface rather than the design for the interface.]

User interfaces is very important in a system. Because they are the only part that interact with the user. And user must have a good experience on our system. So user interface has a key role in the system. For to providing this, it should be simple and light. It should not tire the user when he/she navigate through the system. Color shall be designed for this navigating. UI should not be complex. It should not have more than 3-4 deep links. It should not have colors that can be tire user’s eyes.

### Layout and Navigation Requirements

Layout can not be complex. So when a user login to the system, he/she will encounter a simple layout. All the functionalities can be reach from the main window and can navigate between these functionalities. Each functionalities have its own window. User will be able to easily do his/her work via these tabs(windows). We have three user types. They have different layout, but have same simplicity for navigating via functionalities. And there should not be an ambiguity while navigating through windows. It can be confusing for the user. Navigation bars’ names and windows’ names should not be confusing or very similar. Because user can be confused while navigating

### Consistency

One of the requirement for a good user interface experience is consistency. The windows should match its content, navigation bar and the other functions should be consistent with its content. And it should be give the same experience to the user on every time or should provide a better experience while user using that window.

### User Personalization & Customization Requirements

[Requirements on content that should automatically displayed to users or available based on user attributes. Sometimes users allowed to customize the content displayed or to personalize displayed content.]

For creating many user types, we have to create different layouts, doing customization for that user type. In our system, admin, user and trainer have different layout. Because admin should be able to control over the system so we have providing this functionality with a different layout customization. Trainer should be able to control his class or a user should be able to organize his/her schedules. So basically we have to customize for every user and theirs functionalities.

## Interfaces to External Systems or Devices

### Software Interfaces

[This section describes software interfaces to other components of the software system. These may be purchased components, components reused from another application or components being developed for subsystems outside of the scope of this SRS, but with which this software application must interact.]

We are doing a Gym Automation. So we have to store the data of the users, trainers etc. For this purpose we have to use a Database. For this we will use MySQL database and our developing language is Java. So we have to connect with this Database and Java Application. We can accomplish this with JDBC (Java Database Connectivity). It’s a tool that provides a connection between Database and our application.

### Hardware Interfaces

[This section defines any hardware interfaces that are to be supported by the software, including logical structure, physical addresses, expected behavior, and so on.]

We don’t have any hardware interfaces.

### Communications Interfaces

We don’t have any communication interfaces.

# Business Rules

[Business rules are statements that define or constrain some aspect of the business. Business rules are often represented as production rules when they are meant to be directly executed by an IT System: a production rule is an independent statement of programming logic that specifies the execution of one or more actions in the case that its conditions are satisfied. Production Rules define the operation semantic for the system in a technologic independent way. They constrain the behavior expressed in system use cases.

Organize this document on rule classes, a high level grouping of candidate or actual rules about one **business concept** with a specific kind of **logic processing**, example: Driver Risk Assessment Rules or Customer Validation Rules.]

## Course Related Rules

### Number of Courses That User Can Take Daily Rule

If user tries to take one more course in that day, system won’t allow to do that.

### Course Quota Rule

If user tries to take a course that its quota full, system won’t allow to do that.

### Course Time Rule

If user tried to change his/her course time at the same day, if course’s quota is full at that time,system won’t allow to change the his/her course time.

### Swapping Course Rule

If two users wants two swap their courses, system won’t allow to do that. They have to contact with the trainer.

### Penalty Rule

If user did not attend the courses three in a row, system will give him/her a penalty that he/she won’t take course for a week

## User Related Rules

### User Creation Rule

Only admin can create an user.

### User Information Update Rule

Only trainer or admin can update user’s information, his/her body specs that changing in a time.

# System Constraints

[Constraints are part of the + in the FURPS+ classification of supporting requirements. Describe any design; implementation or deployment constraints on the system being built that have been mandated and must be adhered to. Examples include software implementation languages, prescribed use of developmental tools, third-party components or class libraries, platform support, resource limits and requirements on the shape, size or weight of the resulting hardware housing the system.]

Our system will be written in Java. We will use MySQL as a Database and connect with them JDBC. On the server side, we will use Tomcat server and for the storing these we will use Google Clouds. In the server side, we will user JSF for connecting to UI and Server.

Our system can be used as a Browser Web Application. You can access anywhere if you have a browser. And we will use Eclipse for develop this java project.

# System Compliance

## Licensing Requirements

[Define any licensing enforcement requirements or other usage restriction requirements that are to be exhibited by the software.]

We are developing this system for the gyms. So if a gym wants to use this application, it has to contact us and buy a license from us according to use this application. Other than that we don’t want any licensing.

## Legal, Copyright, and Other Notices

[This section describes any necessary legal disclaimers, warranties, copyright notices, patent notice, wordmark, trademark, or logo compliance issues for the software.]

## Applicable Standards

[This section describes by reference any applicable standards and the specific sections of any such standards that apply to the system being described. For example, this could include legal, quality and regulatory standards, industry standards for usability, interoperability, internationalization, operating system compliance, and so forth.]

We will use Java 8.0 during the project. Oracle provides Java 8.0 coding standards. We will use Tomcat 8 server. Apache provides server standards for Tomcat 8.

PepeFit runs only via Browser with HTML5 standard.

Our application’s language is English. We will use Europe’s Metric System for providing the body specs.

# System Documentation

[Describes the requirements, for on-line user documentation, help systems, help about notices, and so on. Set expectations for the documentation and to identify who will be responsible for creating it.]

When the product comes out, we will provide detailed user guidance manuals. For admins, for trainers and for the users. These manuals will be different with each other corresponding their roles in the system. They will include “ how the system works ?“, “How can you access the PepeFit ?” etc.

The document will be prepared by the whole SadPepe members.

# Appendices

* SRS - Appendix A - Tabular Descriptions of Use Cases
* SRS - Appendix B - GUI Prototypes
* SRS - Appendix C - E/R Diagram