**Software Requirements Specification**

GOOD LIFE

**Version 1**

**Prepared by**

**Group Name: Team Good Life**

|  |  |  |
| --- | --- | --- |
| **Raloue R Kapoor** | **170911254** | **ralouekapoor@gmail.com** |
| **Ayush Srivastava** | **170911236** | **sri.ayush777@gmail.com** |
| **Sajal Gupta** | **170911200** | **sajal52@gmail.com** |
| **Sankalp Kashyap** | **170911184** | **sankalp112kashyap@gmail.com** |
| **Rishabh Raj Jaiswal** | **170911192** | **rraj411@gmail.com** |

|  |  |
| --- | --- |
| **Instructor:** | **Mrs. Aiswarya** |
| **Course:** | **Internet Tools and Technologies** |
| **Lab Section:** | ***A2*** |
| **Date:** | **18th September 2019** |
|  |  |

**Contents**

**REVISIONS III**

**1** **INTRODUCTION 1**

1.1 Document Purpose 1

1.2 Product Scope 1

1.3 Intended Audience and Document Overview 1

1.4 Definitions, Acronyms and Abbreviations 1

1.5 Document Conventions 1

1.6 References and Acknowledgments 2

**2** **OVERALL DESCRIPTION 3**

2.1 Product Perspective 3

2.2 Product Functionality 3

2.3 Users and Characteristics 3

2.4 Operating Environment 3

2.5 Design and Implementation Constraints 4

2.6 User Documentation 4

2.7 Assumptions and Dependencies 4

**3** **SPECIFIC REQUIREMENTS 5**

3.1 External Interface Requirements 5

3.2 Functional Requirements 6

3.3 Behaviour Requirements 6

**4** **OTHER NON-FUNCTIONAL REQUIREMENTS 7**

4.1 Performance Requirements 7

4.2 Safety and Security Requirements 7

4.3 Software Quality Attributes 7

**5** **OTHER REQUIREMENTS 8**

**APPENDIX A – DATA DICTIONARY 9**

**Revisions**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Primary Author(s)** | **Description of Version** | **Date Completed** |
| 1.0 | Raloue Ravi Kapoor  Ayush Srivastrava  Sankalp Kashyap  Sajal Gupta  Rishabh Raj Jaiswal | The current version is limited to people of manipal. We’ll surely expand the website so that it can be used efficiently by larger audience. | 18/09/2019 |

# 

# **Introduction**

Specification document for **Good Life**  - a cross platform product consisting of three major services for its users - Maintaining good Physical, Mental and Financial health. The document provides a complete example of a software design.

In the following sections, we specify the purpose of this document, its scope, the intended audience and the overview of the Good Life system and the sources used in the production of this document.

## **Document Purpose**

The purpose of this document is to provide a high-level overview of the architecture and a low level design for the **Good Life** Website. In addition, it outlines the different strategies and methods used to produce the most effective architectural structure for the website.

The purpose of this document is to convey information about the application's requirements, both functional and non-functional, to the reader. This document provides:

(a) A description of the environment in which the application is expected to operate.

(b) A definition of the application's capabilities.

(c) A specification of the application's functional and non-functional requirements.

The document is intended to serve several groups of audiences:

* First, it is anticipated that the SRS will be used by the **application designers.** Designers will use the information recorded here as the basis for creating the application's design.
* Second, the **client** for the project, the application user in our case, is expected to review this document. The SRS will serve to establish a basis for agreement between the client and development team about the functionality to be provided by the application.
* Third, the **application maintainers** will review the document to clarity their understanding of what the application does.

## **Product Scope**

The objective of **Good Life** is to provide a cross-platform product that allows its users to keep up with their fast paced lives by providing them with three major features - Maintaining Good Physical, Mental and Financial Health.

The user can register as a new user and can then login.The feature to regenerate the password if forgotten with the prerequisite that the user remembers answer to his security question. One of the objectives of the product is to provide user the option to actively promote sports culture in Manipal and stay fit.

Accordingly, the user can play sports regularly and connect to fellow sports enthusiasts by reserving slots online through the web services provided. He/she can choose venue and timings. Others can join the request posted by the user. Users can follow their favourite sports and players through embedded videos and post their sports related images and videos, which is an asset in promoting sports culture in the campus. Users will be granted points and a leaderboard will be maintained to make things more interactive and exciting. The leaders of the points table can organize sporting events and will be provided incentives like coupons/vouchers as well.

The second major feature of the product in order to fulfill its objective is to help the user stay mentally fit in order to fight the odds of life in a calm manner. So, it provides users the feature to follow guided meditation through the web services offered. The user can follow and post meditation and spiritual content through the product online. The user can organize meditation sessions as well.

The third major feature of the product is to help users maintain their finances so that they are not short of money when needed. The product allows the users to add their daily transaction details, be it - credit or debit. They get an expected spending summary through the thoughtful analysis of the product in an interactive manner. This feature allows the user to set their monthly monetary budget.

The product also features one motivational quote of the day to its users.

Another feature of the product is to provide a “Problem of the Day” tab for the budding coders of Manipal, which will show a new coding problem daily from famous competitive coding websites like <https://www.geeksforgeeks.org/>, <https://www.codechef.com/>.

* 1. **Intended Audience and Document Overview**

The intended audience of this document is the faculty of ICT Department and the developers. It will also serve as a reference for the team members. The document contains the software specification that is used to create the product, as well as hardware interfaces required for proper functioning. It serves as a reference for its users, that is, the students of MIT, Manipal.

## **Definitions, Acronyms and Abbreviations**

**CSS:** Cascading Style Sheets

**HTML:** HyperText Markup Language

**JS**: JavaScript

**IEEE:** Institute of Electrical and Electronics Engineers

**Manipalites:** Anyone who is a resident of Manipal.

**SRS:** Software Requirements Specification

**SQL:** Structured Query Language

## **Document Conventions**

The document follows the IEEE formatting requirements.

* Font family Arial
* Text size 12
* Heading Size 14
* Single spaced
* Margin 1”
* Justified alignment
* Italics for comments

## **References and Acknowledgments**

* NodeJS - <https://nodejs.org/en/docs/>
* MySQL - ([https://www.mysql.com](https://www.mysql.com/))
* Dropbox - (<https://www.dropbox.com/?landing=dbv2>)
* HTML - <https://devdocs.io/html/>
* CSS - <https://devdocs.io/css/>
* SkySports - [https://www.skysports.com](https://www.skysports.com/)

# **Overall Description**

## **Product Perspective**

This product is a free website with different features available for all users based on their needs. This website provides the user to play and organize sporting events. It allows the user to post their sports related images and videos and follow their favourite sports. The users can throw sporting challenges online which would promote sports culture in the campus. Our website also provides guided meditation content, daily motivational quotes and videos. People can organize meditational sessions. It helps the users to contact the counsellor, if they feel. Our website also provides an interactive budget management system through which the user can keep a track of his/her expenditures and incomes. A thoughtful analysis is provided to the users so that they manage their money well.

The website provides an interactive interface to the users by giving them points for each activity performed on the site and maintains a leaderboard for the same. Perks like organizing tournaments, coupons etc. are given to the members of the leaderboard on attaining certain points.

* 1. **Product Functionality**
* The user can create an account if he/she doesn’t have one.
* The user can change password.
* The user can regenerate his password if forgotten.
* The user can book a slot for sport events in various stadiums/fields.
* The user can organise meditation sessions and can contact counsellor if needed.
* The user can keep a track of his/her expenditure and income through our website’s budget management feature.
* The user can view the above stats which our website provides after thoughtful analysis.
* The user can upload and view all the public content.
* The user can ask for help from the chatbot if any doubts are there.

## **Users and Characteristics**

The website has 3 main users:

1. **The people of Manipal-** They are the main front-end users of the website and the website has been specifically designed to tackle the problems that an average Manipalite faces. The user has access to various features of the website like booking slots for playing, viewing embedded videos, uploading fitness videos, listening to podcasts, attempting problem of the day. They can view the quote of the day and listen to motivational tunes. They can compete against each other with respect to a leaderboard to gain points and later get some perks like organising tournaments, getting discount coupons,etc.

1. **System-** The system will be doing the background calculations of leaderboard, checking if the slot is empty or not before assigning, auto-replying to the messages/queries of different users, checking if the video to be uploaded has been marked inappropriate by YouTube or not.

1. **Database Manager-** The database manager will reply to queries in case the chatbot does not understand the query, will suspend a user’s account if multiple complaints are registered against him, will change the availability of different venues in case any construction work is going on, approve and reject requests for uploading content and change the algorithms for different calculations if required.

## **Operating Environment**

* Processor:  Intel dual core or above
* Processor Speed: 1.0GHZ or above
* RAM:  1 GB RAM or above
* Hard Disk:  20 GB hard disk or above
* Front End:  HTML, CSS, Javascript, Bootstrap
* Back End: Nodejs, SQL
* Database:  MySQL
* Operating System- Windows 10

## **Design and Implementation Constraints**

* **Memory-** The system should have a minimum of 1GB RAM and 20GB Hard disk.
* **Language-** The user should know English to use the website.
* **Security Consideration-** The user should remember the answer to his security question.
* **Connectivity-** A reliable internet connection is needed to access the website.

## **User Documentation**

A user-manual explaining the different tabs of the website will be made as a video and will be uploaded on the website. It will be available on YouTube as well and users can view it again and again. The website will also show how the points are being assigned for different tasks for leaderboard calculation.

## **Assumptions and Dependencies**

It is assumed that the users will be filling slots only if they are going to attend the event and the users need to have a knowledge of English and basic knowledge of booking systems. If the user attempts to regenerate his password, it is assumed that he remembers the answer to the security question asked at the time of account creation.

# **Specific Requirements**

## **External Interface Requirements**

### **User Interfaces**

The designed software is going to be used by varied types of users like sports enthusiasts, fitness enthusiasts and students who would like to handle their finances well.

Different interfaces will be designed for different types of user. The interface will be unique to every

user and every user will have its functionalities.

### **Hardware Interfaces**

The System itself does not require any hardware to run. A computer with any operating system can be used to use this website. Minimum system requirements are covered under section 2.4.

### **Software Interfaces**

The website makes use of HTML, CSS and Javascript for frontend. It makes use of Node.js for the backend. It makes use of SQL for accessing the database. The database which we are going to use is MySQL. We shall make use of PHPMyAdmin which is a web application for controlling the database.

**3.1.4 Communications Interfaces**

Proper security will be there between all the messages passed between the users so that the

privacy of the users is maintained. Proper error handling will be there in case communication is not

properly established. Precautions will be taken if the user tries to perform SQL injection while

communicating with the other user.

## **Functional Requirements**

**Sports Enthusiasts**

**Login**

Each user will have to login into the system so that he/she can gain access to the website.

**Logout**

Each user can log out of the system whenever he wants.

**Register**

Each user must be registered before he/she can use the website. The data of the user is stored in MySQL database.

**Book Ground for playing as a host**

If a player wants to play with his/her friends on the ground then he/she can book it using our website.

**Join the game for playing created by someone else**

If a player wants to play with his friend who has hosted the game then he/she can join it.

**View sports news**

If the user is interested in viewing the news regarding sports he can view it. We shall be feeding in fresh news everyday.

**Chat**

The players can chat with each other regarding any topic or they can challenge their friends for the same.

**View reserved slots**

Users can view which all grounds are books on which dates and time. This is done so that there is transparency in the system

**View leaderboard**

Users using our system regularly will be given more points and the leaderboard will be visible to everyone.

**Generate booking records**

Users will be allowed to download a csv file of all the events which they have organized for their own record

**Finance management for student**

**Login**

The user must login to the system to use it.

**Logout**

The user can logout from the system anytime he/she wants.

**Register**

Each user must be registered before he/she can use the website. The data of the user is stored in MySQL database.

**Add daily transaction details**

People can add their daily transaction details ,that is, debited and credited details.

**View statistics about the summary of previous transactions done**

People can view their previous transaction summary with the help of pie charts and bar graphs.We shall also be using more complex graphs like box and other plots

**View expected spending details**

We shall also be doing some analysis to show what the expected spending of the user will be over a period of time. This will help the user in understanding how much the user is spending.

**Set monthly limit**

The user can put a specific monthly limit so that he doesn’t spend more than that. Proper notification will be given to the user if he/she exceeds this limit.

**Students interested in Mental fitness**

**Login**

The user must login to the system to use it.

**Logout**

The user can logout from the system anytime he/she wants.

**Register**

Each user must be registered before he/she can use the website. The data of the user is stored in MySQL database.

**Post**

The user can post some content related to meditation which the community can see.The post can be an image,video or plain text.Various Ted talks and other motivational videos can be shown as a post to the user.

**View post**

The user can view the post which other people posts.

**Comment**

The user can comment on the post if he/she wishes to.

**Like**

The user can like the post which other users have posted.

**Notifications**

The users can get notifications if someone likes or comments on the users post

**View quote of the day**

A motivational quote of the day will be shown to the user whenever he logs in. It is shown by default to the user

**Listen to podcast**

The user will be able to listen to podcast related to meditation if he/she wishes to.

## **Behaviour Requirements**

### **Use Case View**

*<A use case defines a goal-oriented set of interactions between external actors and the system under consideration. Since sometimes we will not be able to specify completely the behaviour of the system by just State Diagrams, we use use-cases to complete what we have already started in section 3.3.1.*

*TO DO: Provide a use case diagram which will encapsulate the entire system and all possible actors. Do not include detailed use case descriptions (these will be needed when you will be working on the Test Plan), but make sure to include a short description of what every use-case is, who are the actors in your diagram. For more information please refer to your UML guide and the MiniThermostat SRS example file.>*

# **Other Non-functional Requirements**

## **Performance Requirements**

1. The website won’t take more than 12 seconds to load.
2. Any new registration won’t take more than 10 seconds to be updated in the database.
3. The embedded videos won’t take more than 5 seconds to start playing provided the internet speed is good.
4. The quote of the day won’t take more than 2 seconds to load once the website is launched..
5. The leaderboard update frequency has been set to one hour, and the new leaderboard will be displayed after the calculations within 20 seconds.
6. The chatbot will reply within 5 seconds and if the chatbot isn’t able to understand the question, then the time taken to reply depends on the admin.
7. The page indicating the booked and empty slots will load within 5 seconds.

## **Safety and Security Requirements**

1. The embedded videos are taken directly from YouTube and the care is taken that no age restricted videos are uploaded on the website. Similarly the care is taken that meditation tones or podcasts are age restricted and relevant.

2. Only the Database Manager has the authority to make any changes to the database. (Like changing algorithm for leaderboard calculation, adding new tasks, approving videos, etc)

3. Each user has his/her own login credentials which will be used to keep track of their activities to assign the leaderboard positions, book their slots, etc.

4. The queries received by the chatbot are recorded to improve performance of the chatbot and to keep it secure.

5. The problem of the day section will have coding problems from known sources like [www.hackerrank.com](http://www.hackerrank.com), [www.hackerearth.com](http://www.hackerearth.com) and [www.codechef.com](http://www.codechef.com) .

## **Software Quality Attributes**

**4.3.1. Reliability:** The reliability of the slot information varies because it is entirely user dependent as some users may book a slot for playing some sport but may not show up for it. Apart from it, all the contents of the website are 100% reliable since everything goes through the admin.

**4.3.2. Reusability:** The website can be used daily. The users can create events easily and the website keeps on updating the leaderboard every hour. The user can experience new quotes as well as a new problem of the day everyday, thereby making the website reusable.

**4.3.3. Robustness:** The website is robust as if the data has been entered correctly, the calculation of the leaderboard will be perfect and the fair chance of everyone getting to play will be taken care of very well. Also if any user books a slot and doesn’t show up then others can report this to us under our customer feedback Mail-ID.

**4.3.4. Correctness:** All the website content apart from the slots booked will be correct as they pass through the database manager.

**4.3.5. Maintainability:** The website can be maintained and be reused for years as the only data which will need updation is the change in venues since there might be new grounds and stadiums constructed in the future.

**4.3.6. Usability:**  The website is very easy to use as the user just has to register and his ID will be created. Following this, he can easily book grounds to play, for a particular time. The user can easily access the quote of the day and the problem of the day sections. The videos are easy to view and upload.

# **Other Requirements**

*<This section is* ***Optional.*** *Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>*

**Appendix A – Data Dictionary**

**Manipalites:** The people of Manipal.

**Robust:** The ability of the website to handle failures.

**Expenditure:** The amount spent by a person.

**Appendix B - Group Log**

As soon as the group was made, the members decided to keep a meeting every week, to decide the flow of the website and discuss the status of the work.

The first meeting involved the selection of ideas for the website in which we decided that we should make something that will be useful for most of the manipalites.

In the second meeting we discussed the different functionalities that the website can offer. We also discussed about the flow of how all the functionalities will perform under different sections,i.e., mental, physical and financial health.

In the third meeting we divided the work of writing the **Synopsis** for the project and then met again to compile the final document.

In the fourth meeting we divided the work of writing the **SRS** for the project and then met again to compile the final document.

## **Introduction**

Software Requirement Specification (SRS) document usually contains a software vendor’s understanding of a customer’s software requirements. This document ensures that the software vendor and the customer are in agreement as to the features required in the software system being built. SRS is created after the initial requirement elicitation phase in which Software vendor interacts with the customer to understand the software needs. Usually SRS documentation is prepared by a business analyst who has some technical background.

An SRS is written in precise, clear and plain language so that it can be reviewed by a business analyst or customer representative with minimal technical expertise. However it also contains analytical models (use case diagrams, entity relationship diagrams, data dictionary etc.) which can be used for the detailed design and the development of the software system. SRS is one of the most critical pieces of software development since it acts as the bridge betweens the software developers and business analysts. An incomplete or incorrect SRS can have disastrous effects on a software project.

In this article I explain the major sections of a typical Software Requirement Specification document. I also provide a generic SRS template which can be customized for your project needs.

## **What is the need for an SRS document?**

Software Requirements Specification is usually the first deliverable for any software project. As they say, first impression is the best impression!, and you should ensure that even the first draft of an SRS is of high quality.

The benefits of a good SRS are,

* A contract between the customer and the software vendor – A good SRS document specifies all the features required in the final system including technical requirements and interface requirements. SRS document is used by the customer to determine whether the software vendor has provided all the features in the delivered software system. To the Software vendor it provides a solid foundation to fix the scope of the software system.
* Enables costing and pricing of the project – A well defined SRS enables software developers to accurately estimate the amount of effort required to build the software product. Function point analysis and SMC are some the techniques adopted for estimating effort.
* Input for detailed design – A good SRS enables experienced developers to convert the requirements directly to a technical design. For example, a well defined data dictionary can be easily converted to a database specification.
* Management of customer expectations – Since SRS precisely defines project scope, it ensures that customer expectations don’t change during software development. If they do, SRS can be modified and costing/pricing can be done again on the changes required.

## **What are the contents of an effective SRS document?**

There is no single precise template for writing good Software Requirement Specifications. The contents of an SRS document depends on the software product being developed and also on the expertise of the people doing the requirement elicitation. Different business/technology domains in a company usually have their own customized version of SRS template. Still a good Software Requirement Specification (SRS) usually contains project scope section, functional requirements, requirement analysis models, external interface requirements and non functional requirements. Each of these are explained below.

## **Scope of the project/ Product vision**

One of the most important items in the requirements specification is the precise scope definition of the project. Accuracy of this is important since SRS is also used for estimation and costing. This section should include a brief overview of the project and should also indicate the goals of the project including its benefits. Sometimes it is better to separate the project scope into a separate document.

If the project is for the development of a product, product vision defines the scope and the target user base of the product.

## **Functional Requirements**

Functional requirements specify the business requirements of the project in detail. Usually business requirements are specified in terms of the actions that user performs on the software system. This is known as the use case model. But not all requirements need to be specified as use cases. Functional requirements should contain a combination of use cases and plain textual description of system features. System features are specified at a higher level and use cases attempt to translate into user actions.

Again there is no fixed format for use case description, but it usually contains the following information,

* Use case diagram – For a small systems, a single diagram can be used to depict all the use cases in the system.
* List of actors and their details – This identifies the various types of users interacting with the software system.
* Use case description – Purpose of the use case and how and when it is invoked by the user. This should also include an identifier for easy reference.
* Preconditions – List of system states/conditions that must be true for the successful execution of the use case. This section is optional and could be easily incorporated into the basic steps section.
* Basic steps – These indicates the various fine grained steps required for the execution of the use case.
* Alternate steps – These indicate alternate events of the use case being described.
* Business validations/rules – These indicates various types of input validations or business rules required in the use case being described.
* Post conditions – Indicates the results of the use case. Please note that this section is optional and could be incorporated into the basic steps section.

To ensure that all the business requirements are addressed in the final software product, a traceability matrix document is used. Traceability matrix tracks each requirement through various phases of software development (detailed design, unit test plans, system testing plans, user acceptance test plans and code components). This requires that every requirement in the SRS should be identifiable by a unique number or tag.

For software projects where majority of features are available as user interfaces, it is better to complement this section with screen prototypes. These user interfaces can change during detailed design, but having a draft version of user interface in the requirements document helps a lot in communicating business requirements. However some customers insist on having finalized user interfaces in the requirements specification document.

## **Requirement Analysis Models**

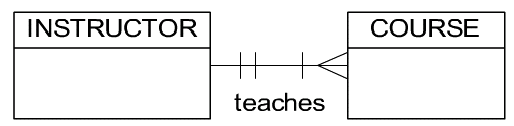
Once the overall use cases in the system are identified in requirements elicitation, requirement analysis models can be developed to drill down to specifics of each requirement. For example, a use case such as “Add customer” may not specify all the customer details that needs to be captured by the system. This is usually specified in the data dictionary model and also in the screen prototype.

Requirement Analysis models act as the bridge between functional requirements and the detailed design of the software system. For example, Use cases lead to user interface design, data dictionary and entity relationship diagrams are used for designing database schema and class diagrams.

Following are some of the widely used requirement analysis models,

## **Entity Relationship Diagrams**

Entity relationship model diagram (ERD) is a conceptual representation of the data in a software system. During detail design this model is mapped in to the physical database model. There are different diagramming conventions available for creating ER diagrams. Following is a sample ERD in Crow’s foot notation (this is taken from the ERD of a course registration Web application requirements),



This diagram indicates that there is one and only one instructor for a course and an instructor can have one or more courses. The relationship is captured as instructor “teaches” course.

## **Data Dictionary**

Data dictionary in a requirements document is an extension of the entity relationship diagrams. Which ER diagrams specify system entities and their relationships, a data dictionary lists all the attributes pertaining to each of those entities.

In a data dictionary, each attribute of the entity data in system is analyzed in detail including type of attribute, whether it is optional and a brief description of the attribute. Please see the sample SRS template section for more details.

In addition to the above models, sometimes it is useful to develop state transition diagrams and data flow diagrams. To describe a complex process flow or a workflow in the application, process flow diagrams or flowcharts can be used.

## **External Interface Requirements**

It is very rare that we have a standalone software system. Usually a software system interacts with a number of external applications for data input and output. For example, an e-business application usually needs to be integrated to an external payment gateway. All the external interface requirements are detailed in this section. The important thing to document here are the entities that are passed across the external interfaces.

## **Non Functional Requirements**

Non functional or technical requirements specify how the software system should operate. In contrast functional requirements specify what a software system should do. Some of the non functional requirements are derived from the functional requirements. Non functional requirements captured include performance requirements, application scalability, application security, maintainability, usability, availability, logging and auditing, data migration requirements, multi lingual support etc. Please note that only a subset of the list are applicable for a specific project.

## **Importance of a good SRS template**

A good SRS template ensures that all important information required in a Software Requirement Specification is captured during requirement elicitation. Following is the table of contents taken from the SRS template linked below.

## **Contents of Software Requirements Specification (SRS) Template**

