**Software Requirements Specification**

for

**Booking Management System**

**Version 1.0**

Prepared by Tripathi and Wang

SimpleIT Solutions

10/24/2021

# **1.** **Introduction**

## **1.1** **Purpose**

This SRS contains software requirements: both functional and non-functional for the complete project. It also contains additional requirements which can be completed given the developers complete the core functionality within a reasonable time frame.

## **1.2** **Document Conventions**

Every requirement statement has its own priority. The requirements are written in simple language without explanation of how the requirement is to be implemented. The high priority functional requirements are essential before beginning the first testing. Software can be considered a prototype after all medium priority requirements are also implemented so that the prototype can be shown to the client to get feedback and adjust accordingly.

## **1.3** **Intended Audience and Reading Suggestions**

This SRS is for fellow developers, testers and documentation writers. It is written in the simplest format so that there is no or very little technical ambiguity.

## **1.4** **Product Scope**

The software is a means to replace the traditional booking system where users had to make a phone call every time a new booking had to be made. It is also to utilize internet technologies to attract more climbers/ climbing enthusiasts. The system is also a significant update in terms of customer’s convenience as they are able to register from anywhere easily as long as they have internet connectivity.

## **1.5** **References**

[1] SRS template by Karl E. Wiegers

# **2.** **Overall Description**

## **2.1** **Product Perspective**

The product is a new system to be implemented with the intention of replacing the traditional methods of booking solely for the convenience of users.

## **2.2** **Product Functions**

* The major function of the system is to allow users to create accounts and login to the system, and to make a booking for the court for any particular day.
* It is also a great way to manage coaches and view business statistics for admin.
* The system sends notifications and reminders to both users and coaches.

## **2.3** **User Classes and Characteristics**

* Professional climbers: visit often/ have a membership, use nfc devices to track routes conquered
* Beginner climbers: book a coach, may sign up for a short term membership, rent equipments
* Climbing enthusiasts: book a coach and court for one day, rent equipments

## **2.4** **Operating Environment**

Since it is a web based registration system, it requires a browser like google chrome to operate and can be operated from any device: laptop, phones or tablets. Internet connection is necessary.

## **2.5** **Design and Implementation Constraints**

System should be highly responsive (i.e. no lagging/ loading) and should take as minimal RAM as possible. Cloud based databases are to be used. The system is developed using HTML, CSS and Javascript. Any framework/library can be used as required.

## **2.6** **User Documentation**

Manuals, customer support and video tutorials

## **2.7** **Dependencies**

Payment gateway like Paypal and Credit Card

Cloud based database: (Google cloud services)

# **3.** **System Features**

## **3.1** **Create a user account (User Registration)**

3.1.1 Description and Priority

A user should be able to register to the system by using email. Email and phone number verification should be made by the system.

It is of high priority.

3.1.2 Stimulus/Response Sequences

A user is provided with a form asking information like name, address, email and phone number. After the user provides all the information and clicks submit, the system should store all the data in a database while also sending an email for email verification. User provides the verification code and if the code is correct, he or she will be registered to the system. The system performs real-time error checking on user input as user types in the form.

3.1.3 Functional Requirements

REQ-1: User should be presented with a form

REQ-2: Error checking as the user inputs data in the form and goes to the next field. If there is an error, the field is highlighted. Users are not allowed to submit the form as long as there are errors in the form.

REQ-3: After a user submits, email verification is done by the system. User will enter the verification code.

REQ-4: System will store all provided data along with login details in a database.

## **3.2** **Make a booking**

3.1.1 Description and Priority

A user should be able to book a court for any particular day

High priority (core feature)

3.1.2 Stimulus/Response Sequences

A user selects on which day he wants to book the court for. System will then show available courts with available timings. The system will also show if coaches are available on the selected timings. After the user makes a selection, the data will be stored in the database and he will be forwarded to make a payment. If he wants to book a court in a time that is not available he will be asked to make a new selection while also keeping him in the waitlist so that he will have an option should the space become available.

If a user has a membership, and is currently active, his booking will be confirmed immediately and will not be forwarded to make a payment. However, he will not be allowed to make unlimited bookings and if bookings exceed his allowed limit, he will be asked to make payment for the extra bookings.

3.1.3 Functional Requirements

REQ-1: User should be able to login using his/her credentials

REQ-2: User should be able to pick a date and time along with the court he likes to make a booking.

REQ-3: Users should be able to book a coach at the selected time if a coach is available at the time. Moreover, users are able to see the timings of coaches.

REQ-3: User should have the option to stay in the waitlist if he/she wants to and a notification should be sent to the user if the space becomes available.

REQ-4: System will store all provided data in a database.

## **3.3** **Make a payment**

3.3.1 Description and Priority

After a booking has been done, a user should be able to proceed to payment. A booking will be confirmed only after payment is made.

It is also of high priority.

3.3.2 Stimulus/Response Sequences

A user is presented with the hourly rate of the court. Users are able to select additional items like water, climbing equipment and shower if they want to. Total cost is calculated and presented to the user. Users will be able to pay using paypal or credit card. After payment is successful, a confirmation message is displayed and an email with a calendar invite is sent to the user.

3.3.3 Functional Requirements

REQ-1: User should be able to select additional items like water and climbing equipments

REQ-2: Total cost should be calculated

REQ-3: Field to enter promo codes for discounts

REQ-4: Option to select a payment method and after selection users should be forwarded to payment gateway

REQ-5: System should verify if the payment was successful. If successful, a confirmation message is displayed and an email notification is sent. If unsuccessful, the user is asked to make the payment again.

REQ-6: System will store all relevant data in a database.

## **3.4 Sign up for membership**

3.4.1 Description and Priority

Users should be able to join as a member for a specific amount of time. They have certain benefits according to the membership plans (1-month, 3-months or 1-year)

It is of medium priority.

3.4.2 Stimulus/Response Sequences

A user has an option to join the membership club. There are three membership types: (one month, three months and one year). And also under membership types there are subscription types. On the basic subscription, users are allowed to book a pitch for 3 sessions per week, on intermediate subscription, users are allowed to book a pitch for 5 sessions per week and on advanced subscription users are allowed to book a pitch for 10 sessions per week. The membership also includes options to whether or not to include a coach. After the user selects required preferences, he will be forwarded to make a payment. After payment is successful, a confirmation message is displayed and an email is sent to the user.

3.4.3 Functional Requirements

REQ-1: User should be able to select membership type, include or exclude coach and subscription type

REQ-2: Field to enter promo codes for discounts

REQ-3: Option to select a payment method and after selection users should be forwarded to payment gateway

REQ-4: System should verify if the payment was successful. If successful, a confirmation message is displayed and an email notification is sent. If unsuccessful, the user is asked to make the payment again.

REQ-5: System will store all relevant data in a database.

## **3.5 Coach login**

3.4.1 Description and Priority

Coaches should be able to see their session times for any particular day and mark their availability for future sessions.

It is of medium priority.

3.4.2 Stimulus/Response Sequences

Coaches are provided with login details and can only login to the system. Coach accounts are managed by admins. After coaches login they can mark the times for which they are available so that users can book them. They can view the sessions for which they are booked. Also coaches are able to view the number of sessions coached.

3.4.3 Functional Requirements

REQ-1: Coaches should be able to login to the system

REQ-2: After login success, they can view the timings for which they are booked

REQ-3: They are also able to mark the available timings for which they can coach for the next week

REQ-4: They are also able to view the hours of past sessions coached

REQ-5: System will store all relevant data in a database.

## **3.5 Admin login**

3.4.1 Description and Priority

Admins should be able to log in the system to view/remove bookings by users, view revenue by month/week, and manage coaches.

It is of Medium priority.

3.4.2 Stimulus/Response Sequences

There is only one login detail for admin and every time admin logs in two factor authentication is a must. Admins can create or delete coach accounts, view all bookings made including history, view revenue by month or week. They are also able to see revenue in charts and line graphs by day/week/month so that they can track if their business is getting better.

3.4.3 Functional Requirements

REQ-1: Admins should be able to login to the system, but two factor authentication using email or mobile apps like Duo is required

REQ-2: After login success, they can view all bookings made including history

REQ-3: Admins manage coach accounts as well. They can either create or delete coach accounts.

REQ-4: Admins are also able to view revenue by month/week/day in raw form or in a bar graph or line graph or a chart.

REQ-5: Admins are also able to close the registration so that users won’t be able to book a pitch for the day. This feature is useful when there is maintenance of the court/ climbing wall.

# **4.** **Other Nonfunctional Requirements**

## **4.1** **Performance Requirements**

The user interface should be super smooth, with no lagging and loading screens after the initial load.

## **4.2** **Security Requirements**

* All user data should be stored in a confidential manner. No one else except the user is allowed to access the data.
* Membership data and payment data should be accessible only to admins and not by anyone else.
* All passwords should be stored as hash values
* For admin login, two factor authorization using a phone is a must. System will auto logout after 20 minutes of inactivity.

## **4.3** **Software Quality Attributes**

As it is a web-based system, it should be responsive (i.e. mobile friendly). It should be optimized in such a way so that when viewed from phones, it is easy to use and clear with all the functionality available.

System should be adaptable as well. It should be fully functional when viewed from devices of all dimensions. The code should also be well documented (i.e. clear comments).

# **5.** **Other Requirements**

* Users are able to reset the password in case they forget. However, verification via email or phone is a must.
* Coaches are also able to reset the password but verification is a must.
* A mobile application for users so that when they reach the top of the wall, they hover it over an NFC device installed so that they can view it on their phones which path they have taken to reach the top of the wall.
* Admins can view the most favoured path and make future plans for maintenance accordingly.
* System will generate a unique QR code for customers that customers can present to the front table to verify themselves and also provide the customers with equipment they rented during booking.

**Feasibility Analysis**

**Technical Feasibility**

1. Developers’ and analysts’ familiarity with the business area:

The business area is a web based registration system. Team members have decent experience working with web development but are not expert in it. However, since the platform has to be responsive as well (mobile friendly) team members have very little knowledge about responsive web design, and may take some effort to learn this design paradigm.

1. Familiarity with technology:

The technical tools we are going to use:

* Programming language: Javascript for implementing all the functional requirements
* Database management system: Mongoose and google cloud services
* HTML and CSS for layout design and styling

Team members have decent experience with HTML, JS and CSS but very little knowledge on working with databases.

1. Project Size:

It is about one and a half months for 2 people

1. Risks

The project has moderate risks as developers are experienced and fast learners. Even though they are not experts in the field, they are able to learn new technology in a reasonable amount of time with decent knowledge. The project would be of high risks if all additional requirements were to be implemented.

1. Conclusion

Even though risks are moderate, the project is a challenging one, but as the developers are motivated and skilful, it is possible to complete the project within a reasonable amount of time.

**Economic Feasibility**

| **Costs** | **Period1** | **Period2** | **Period3** | **Period4** | **Period5** | **Period6** | **Total** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Salaries | 20 | 20 | 20 | 20 | 0 | 0 | 80 |
| H/W & S/W | 20 | 0 | 0 | 0 | 0 | 0 | 20 |
| Training | 30 | 0 | 0 | 0 | 0 | 0 | 30 |
| Support and Maintenance | 0 | 0 | 0 | 0 | 10 | 10 | 20 |
| **Total costs** | 70 | 20 | 20 | 20 | 10 | 10 | 150 |
| **Benefits** |  |  |  |  |  |  |  |
| Increase # of customers | 0 | 0 | 0 | 0 | 300 | 300 | 600 |
| Decrease costs | 0 | 0 | 0 | 0 | 30 | 30 | 60 |
| **Total Benefits** | 0 | 0 | 0 | 0 | 330 | 330 | 660 |
| NCF | (70) | (20) | (20) | (20) | 320 | 320 | 510 |
| CNCF | (70) | (90) | (110) | (130) | 190 | 510 | 1020 |

Numbers are in thousands of DHS

NCF: Net Cash Flow

CNCF: Cumulative Net Cash Flow

One period corresponds to two weeks

H/w and S/w correspond to Hardware and Software respectively

1. The return on investment (ROI):

ROI = (Total Benefits - Total Costs) / Total Costs

= (660-150)/150

=340 %

2. The break-even point (BEP):

BEP = (Period.net cash flow - Cumulative.net cash flow) / Period.net cash flow

= (320-190)/190 = 68%

= 0.68\*1\*14 =9.50 ≈ 10 days

So the Project will take 8 weeks and 10 days which is 56 days.

Conclusion:

The ROI is good for this booking system, and the BEP is reasonable, so the risk is low.

**Project Size**

We estimate the **project size** using the function points analysis (FPA).

**Function Point Estimation**

| **Functionality** | **Input** | **Output** | **Queries** | **File** | **Interface** |
| --- | --- | --- | --- | --- | --- |
| Registration | 3 | 2 | 1 | 1 | 0 |
| Login | 2 | 1 | 1 | 1 | 0 |
| Booking | 1 | 1 | 1 | 1 | 0 |
| Payment | 1 | 1 | 1 | 1 | 1 |
| Membership | 1 | 1 | 1 | 1 | 0 |
| Notification | 0 | 1 | 3 | 1 | 0 |

| **Information**  **domains** | **Count** | **Weighting Factor** | | | **Total** |
| --- | --- | --- | --- | --- | --- |
| Simple | Average | Complex |
| EIs | 8 | 4\*3 | 3\*4 | 1\*6 | 30 |
| EOs | 7 | 4\*4 | 3\*5 | 0\*7 | 31 |
| EQs | 8 | 2\*3 | 3\*4 | 3\*6 | 36 |
| ILFs | 6 | 2\*7 | 4\*10 | 0\*15 | 54 |
| EIFs | 1 | 0\*5 | 1\*7 | 0\*10 | 7 |
| **Total Unadjusted Function Points (TUFP) =** | | | | | 158 |

**The total processing complexity (PC)**

| **Tasks** | **Complexity** |
| --- | --- |
| Familiarity with technology | 2 |
| Online data entry | 2 |
| ​​Data communication | 3 |
| Transaction rate | 2 |
| Multiple sites | 2 |
| Performance | 3 |
| Operating ease | 2 |
| **Total Processing Complexity (TPC) =** | 16 |

\*Complexity is from 0 to 3: (0 = no effect on project complexity; 3 = great effect on project complexity)

**The adjusted processing complexity (APC)**

APC = 0.65 + (0.01 \* TPC)

APC = 0.65 + (0.01 \* 16) = 0.81

**The total adjusted function points (TAFP)**

TAFP = TUFP \* APC = 158 \* 0.81 = 127.98

**Converting Function Points to Line Of Code (LOC)**

• 70% will be done in Javascript

• 30% will be done in HTML and CSS (for layouts only)

**Number of lines of code (LOC) = TAFP \* # of (LOC\FP) \* %**

For Javascript = (127.98) \*(71.11)\*(70/100) = 6370.46 LOC

For HTML/CSS = (127.98) \*(15)\*(30/100) = 575.91 LOC

So the total LOC = 6946.37 LOC = 6.95 KLOC

**Estimating the effort**

Effort

= 2.4 \* KLOC

= 2.4 \*

= 18.38

**Estimating the schedule time**

Time

= 2.5 \* (effort)

= 2.5 \* (18.38)= 7.55

**Estimating the number of persons**

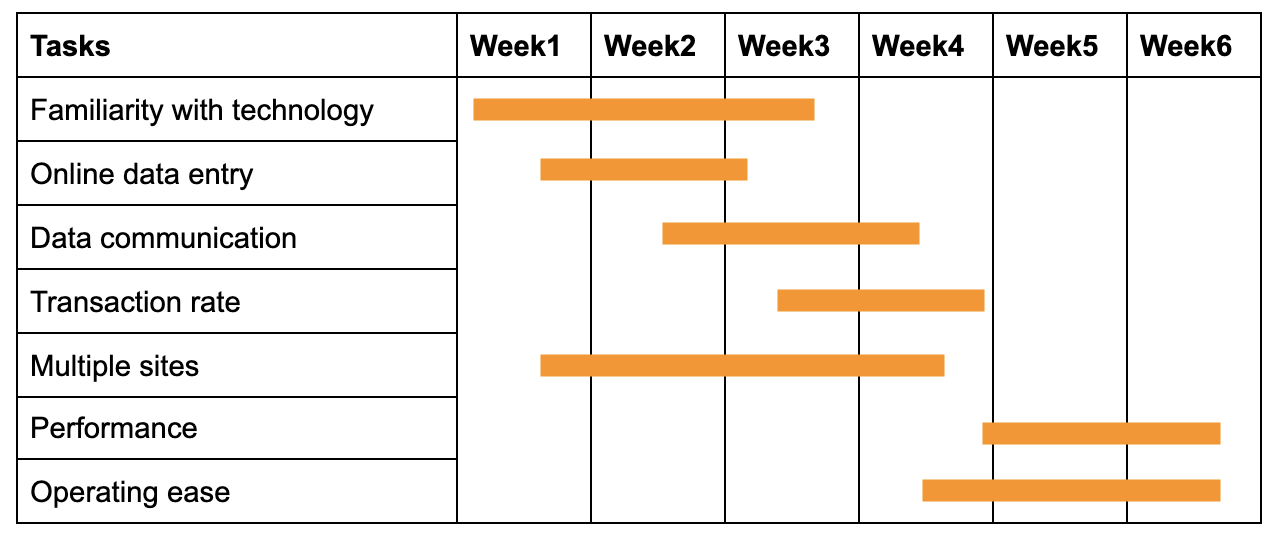
average of # of persons

= effort/time

= 18.38 / 7.55 = 2.43 persons

**Development Workplan**

We draw the project development workplan via Gantt Chart.



**Process Model: Scrum**

We decide to use Scrum, one of the agile models for the following reasons:

* Frequent changes are required in the future, since many functions are included.
* The team members have strong cohesion and are qualified to fulfill the requirements.
* We have customers who are ready to have a meeting with our team throughout the development process .
* The project size is not very big and the time is limited. Agile models can help reduce the total development time.