**Application Overview:**

Our GYM management software offers a complete gym management system with inventory, customer data, that can be adapted to meet unique GYM in order to increase profit and better customer service while reducing labor costs, and decrease operation costs.

Our software is a comprehensive, next generation, ready-to-use GYM system for small, mid-size to large tier GYM. Our Software is built on latest technologies and standards, and based on industry "Best Practices" infrastructure with established and solid GYM business logic. It is user friendly and incredibly easy to use and as simple as you want, or as sophisticated as your business need. Our can cater the need of all kind of GYMs with user level customizations.

**Below are some of the modules our solution can assist you with:**

****

**Table Organization & Management**

1. Optimizing seat distribution by wait-listing functionalities
2. Instantly tracking orders for better efficacy in kitchen and order management

o Numerous reservation sheets for peak periods like Christmas and other events

Providing table mergers and partitions for big parties



**Reservation Service & Management**

1. On-the-fly facilities of modifying the reservations based on guest histories

Storing information about the guests including phone numbers

VIP pre-assessments and special seat allotment

Taking online reservations through website



**Back-office Management**

1. Order Processing
2. Advanced Cash Control

o Corporate reporting

o Labor Scheduling o Time & Attendance



**Time-Crunched Solutions**

1. Ordering quick meals on the go or from your table
2. Allowing quick orders by simplest GUI and minimum clicks

For this project, our group proposes to follow standard development model for software development. A brief overview of the waterfall model SDLC phases is as follows below:

1. **Requirement Analysis:** Our team, in consultation with the customer, studies the complete system in-depth as given in the contract and the related supporting documents, such that each process linked to the final preparation of the software is clearly understood. The SRS document will be prepared, containing the application overview, scope, objectives, need and purpose, software and hardware interfaces, any issues or concerns, and more.
2. **Software Design Phase:** Based on the software requirements and operation concepts documented in the base lined SRS, detailed design solutions are worked out depending upon performance, availability of reusable components, and integration with other applications. TSD and WFD documents are produced and base lined.
3. **Development and Integration:** Coding is done based on base lined TSD, WFD, and the SRS. Unit testing is done on completion of development of each unit.
4. **Testing:** Product testing will be done at this stage by the QC team in accordance with the test plan and test cases.
5. **Releases:** The product will be released to the client after the bug fixing and successful product verification by the QC team.
6. **Post Production Support and Maintenance:** Post production and support is provided on the project. In case of maintenance a change request log is maintained in order to keep track of changes and support requests.

No project specific risks have been identified at this time. However, a Risk Management Plan (RMP) is a standard part of every software project and is referenced in the overall Project Management Plan. We promise to do risk management throughout the life cycle of the project. The RMP is part of a project’s initiation and initial planning, but it is also maintained any time there are updates to project requirements, a customer initiates a change request, concerns are raised by team members, or concerns are raised by senior management stemming from their oversight of Project Status Reports. Risks are documented, analyzed, associated with risk mitigation plans, and tracked.

**Software Development Lifecycle**

**1.0** **Objectives:**

This section describes the Software Development Methodology being used by the Our team, while executing Software Projects / Applications. It further identifies the phases of the lifecycle, the entry and exit criteria of each phase and the quality control tasks related to each phase.

**2.0** **Scope:**

We follow a slightly modified version of the Waterfall model (to facilitate Rapid Application Development) in the Software Development projects. The model is described in detail below:

**3.0** **Phases:**

The execution phases involved are broadly divided in to three stages, which are listed below:

* Project Start-up:



Project Acquisition



Estimation



Requirement Analysis



Project Management Planning



Project QC Planning

* Project Execution:

Design

Development & Integration Testing

* Project Wind-up:



User Acceptance



Sign-Off and Project Closure

**4.0** **Phases - Entry, Task, Verification and Exit Criteria:**

The execution of the various phases is explained in detail below:

**4.1** **Requirement Analysis**

**4.1.1 Activity Definition**

In this activity, our team (in consultation with the customer) defines the procedure to study the complete system in-depth as given in the Contract and the related supporting documents, such that each process linked to the final preparation of the software is clearly understood.

**Entry Criteria**

****

Project Synopsis / Contract has been received from the Sales Team



The Sales team has done the Project Kick-Off with all the necessary documents (like Approved Contract/Synopsis, SRTM) attached with the e-mail.

**4.1.2 Inputs**

****

Kick-Off email



Project Synopsis/Contract document



Any other relevant reference material like (Communication E-mails etc.)

**4.1.3 Tasks and Procedures**

****

SRS is prepared by the Team Lead (authorized by the Project Manager). SRS contains Application Overview, Scope, Objectives, Need and Purpose, Software and Hardware Interfaces and any doubts/queries etc.



Acceptance Criteria of the Project with reference to scope, functionality, performance, and security level is identified, as appropriate.



Implementation Plan is drafted. In case of a module-wise implementation, prioritization is agreed with the Customer.



Hardware, Software and Infrastructure Requirements are identified.



A Draft SRS is prepared (according to the SRS Guidelines defined in the Our Quality Management System). The draft SRS defines the functional and sub- functional, Performance and Interface requirements. The SRS also defines the operational concepts in the form of screen designs, use cases, etc. as appropriate. The documented SRS is further analyzed to ensure that:

* Requirements are necessary and sufficient
* Constraints in terms of costs, schedule, performance, reusable components, and risks are adequately addressed.



The determined requirements and operation concepts are presented to the customer in the form of prototypes, screen designs, etc. to ensure fulfillment of the requirements in the live/production environment. The results of the presentations are documented and the customer comments are analyzed and suitably incorporated in the SRS.



SRS is then reviewed and approved by the Project Manager (or any other person authorized by him/her) and subsequently e-mailed to the Customer for review. The Customer Review comments are then added to the SRS. The discrepancies, if any are discussed with the customer. The SRS is finally labeled in VSS (Microsoft Visual SourceSafe) as per the Configuration Management guidelines.



The Requirements Traceability Matrix (RTM) is updated as per the Base lined SRS and discussed with the team.

**4.1.4 Control Mechanism**

****

Review of SRS by the Project Manager (or any other person authorized by him/her)



Acceptance / Approval of SRS by Customer



SRS, SRTM, Communication E-mails are stored in VSS (under the control of Configuration Manager)

**4.1.5** **Approvals**

The SRS is reviewed by the Project Manager and approved by the Customer.

**4.1.6** **Outputs**

****

Base lined Software Requirements Specification (SRS)



Results of Prototype Demos



Customer Sign-off on SRS



Updated SRTM

**4.1.7** **Exit Criteria**

****

SRS Approval e-mail from the Customer with no further modifications



Base lined SRS

The SRS is reviewed and approved by the Customer. Unless approval from Customer is obtained, the next phase is not started. However, some base activities, which are not dependent on the customer’s approval, may be continued under the sole discretion of the Project Manager (to ensure quicker turn-around time for the project).

**4.1.8** **Quality Records**

****

SRS Review Report



Results of the presentation made to the customer



Sign off e-mail from Customer

**4.2** **Software Design**

**4.2.1 Activity Definition**

In this activity, the Our team converts the requirements as expressed in the analysis phase to the level of programmable processes.

**4.2.2 Entry Criteria**

****

Base lined SRS



An e-mail from the Project Manager confirming the start of the Design Phase.

**4.2.3 Inputs**

****

Base lined SRS



Updated SRTM

**4.2.4 Tasks and Procedures**

****

Based on the software requirements and operation concepts documented in the base lined SRS, detailed design solution(s) are worked out depending upon performance, availability of reusable components, and integration with other applications (s). Detailed design is selected on the basis of different criteria such as:



Complexity of the application



Technology considerations



Possibility of changes in requirements

Limitations in terms of end user’s capability



Ease of operations

Project Lead is responsible for preparing a TSD (Technical Specification Document). Preparation of the TSD is made with reference to the Our guidelines for high-level and low-level design.



After the first draft of TSD is prepared, it is reviewed by the Project Manager (or any other person authorized by him/her). After fixing/verifying the issues (if any), the TSD is base lined in the VSS.



The SRTM is updated based on the TSD.



The unit test cases / module / integration / system test cases as appropriate for the project are prepared based on the SRS and the TSD. The unit / module / integration / system test cases are reviewed and base lined before the commencement of testing.

**4.2.5 Control Mechanism**

****

The TSD is reviewed by the Project Manager (or any other person authorized by him/her).



TSD is stored in VSS (under the control of Configuration Manager)

**4.2.6** **Approvals**

****

The TSD is Reviewed by the Project Manager and base lined in the VSS by the Configuration Manager.

**4.2.7** **Outputs**

****

Selected Design



Base lined Technical Specification Document (TSD)



Design Alternatives (If Applicable)



Decision to develop, use reusable components



Updated SRTM

**4.2.8** **Exit Criteria**

****

TSD Approval e-mail from the Project Manager with no further modifications suggested.



Base lined TSD

**4.3** **Development and Integration:**

**4.3.1 Activity Definition**

In this activity, the Our team converts the selected and base lined design solution into software units, test the software units, integrate the software units and integrate the software application with other application(s) according to the requirements of the project.

**4.3.2 Entry Criteria**

****

Base lined TSD

An e-mail from the Project Manager confirming the start of the Coding Phase.

**4.3.3 Inputs**

****

Base lined TSD



Updated SRTM

**4.3.4 Tasks and Procedures**

****

Coding is done based on the base lined TSD.



User interface is created.



Database is created and integrated with user interface.



Software Unit Testing is carried out by the programmer to ensure error free programming. The Unit Testing is conducted based on the base lined unit test cases that are prepared by the programmer prior to start of coding or else by an independent team which is not involved in coding of the software units for which test cases have to be prepared.



Coding is carried out as per the Our Coding Guidelines (platform-specific).



Code Review is conducted as per the requirements of the project.



After Software Unit Testing and Code Review, the software units are base lined.



After coding, the RTM is updated (if required).



The software units are integrated together in a given sequence (any specific hardware / software environments required for integration of software units are planned in the Project Plan and the integration environment is provided for integrating the various software units). One of the following options are opted here (depending on the type and complexity of the application):

**Bottom-up approach**

The software units of the system are integrated starting from the software units that do not call any other software units of the system. The components that directly call these tested software units are

tested next. The process is repeated until all the software units of the system have been integrated and tested. This approach is used when the critical software units of the system are at lower levels in the software unit hierarchy.

Dummy software units are written for higher-level software units that are not ready. Once the higher-level software units are ready, they replace the corresponding dummy software units and then the combination is tested again.

**Top-down approach**

The software units of the system are integrated starting from software units that are at the topmost level in the system. The software units directly called by these software units are added one by one and the combination is then tested. This process is repeated until all the software units of the system have been integrated and tested. This approach is used when the critical software units of the system are at higher levels in the software unit’s hierarchy.

If software units called by higher-level components are not ready, dummy software units are written. The dummy software units are replaced when the required software units are ready and the combination is tested again.

**Combination approach**

This approach is a combination of bottom-up and top down approaches. In this approach the software units of the system are integrated starting from the bottom as well as from the top simultaneously. This approach is used if the number of software units in the system is large.

**Big-Bang approach**

In this approach all the software units of the system are integrated and tested together. This approach is used only when the number of software units in the system is small.

**4.3.5 Control Mechanism**

****

Our Coding team ensures error-free running of software units through effecting unit-testing and code reviews (with the help of Unit-testing Checklist and Code Review Checklist respectively).



List of base lined software units that are integrated together is prepared and base lined.

**4.3.6** **Approvals**

****

Software Units are tested by the Unit-testing team and base lined by the Configuration Manager.

**4.3.7** **Outputs**

****

Software Units



Integrated Software Units (Software Application ready for Testing)



Updated SRTM

**4.3.8** **Exit Criteria**

****

Base lined Software Units



Integrated Software Units

**4.3.9** **Quality Records**

****

Unit Testing results/ defect logs



Code Review Report



Review Record for list of software units

**4.4** **Testing**

**4.4.1 Activity Definition**

In this activity, the Our testing team validates the software developed with respect to its functional and environmental requirements.

**4.4.2 Entry Criteria**

****

Software is ready for System Testing.



Software modules /sub-modules have passed the integration testing phase.



System Test Plan / Test Cases are base lined



An e-mail from the Project Manager notifying the start of the testing phase

**4.4.3 Inputs**

****

Base lined and Reviewed System Test Plan



Base lined and Reviewed System Test Cases



Software Build for testing



Updated SRTM

**4.4.4 Tasks and Procedures**

****

As scheduled in the Project Plan, the Our testing team carries out the system testing with the help of the reviewed test cases and test data.



The type of testing to be performed is detailed in the Test Plan and the execution scenarios for each type of testing are listed in the System test cases.



The automation tools are also used (if needed) for functional/performance/stress/load testing.



All the test results are noted and properly recorded by the Test Engineer.



Any result deemed to be a defect, will be recorded by the Test Engineer in our Defect Tracking tools (Bug Report or Black-flag). The analysis of the results is carried out by Test Manager.

Areas impacted by any modifications in code (as a result of the defects reported by the test team) are identified, tested again and properly documented.



The System Testing procedure is carried out till the approval is obtained on the Test Results from the Project Manager.



Test Results and Test Data are stored in the VSS under the control of the Configuration Manager.

**4.4.5 Control Mechanisms**

****

Testing is done based on the approved Test plan/Test Cases.



Testing procedure is controlled by means of the Test Plan.



On approval, the Software/System is base lined in the VSS.



All the documents/records are also stored in the VSS (base lined wherever applicable).

**4.4.6 Approvals**

****

Test Manager approves the Test Plan and Test Cases



Test Manager approves the Test Results for each build.



Project Manager signals / approves the end of the testing cycles

**4.4.7 Outputs**

****

Test Results



System-Tested Software



Test Plan / Test Cases Review Notes

**4.4.8 Exit Criteria**

System Test Results are approved by the Project Manager.

**4.4.9 Quality Records**

****

Test Data and Test Results



Defect Tracking Report



Review Report

**4.5** **Release**

**4.5.1 Activity Definition**

In this activity, the Our team identifies the need for control of performing replications of software deliverables, maintaining proper backup of the deliverables, accurate identification, final inspection of the deliverables before shipment, shipment of deliverables to the Customer, and installation of software at the Customer's site.

**4.5.2 Entry Criteria**

****

Developed and tested software



An e-mail from the Project Manager to release the Project / Product to the customer

**4.5.3 Inputs**

****

Developed and tested software

**4.5.4 Tasks and Procedures**

****

Release of any deliverable to the client is made from the Base lined folder of the respective project in the VSS.



The mode of delivery can be electronic media (through direct uploading) or physical media like CD, Diskettes etc.



In case of Physical Delivery, identification/verification of the recipients and the number of copies of the deliverables is carried out.

In case of delivery through Electronic Media (like uploading to the web server), the Release Plan of the application is created.



The Web Server configuration is re-verified to ensure that every back-up and monitoring system is in place.



Release Note is also released along with the deliverable which includes the following:



User Documentation



Hardware / Software dependencies



After delivering the project/product, acceptance from client is received through an e-mail or a customer-specific acceptance note.

**4.6** **Post Production Support and Maintenance:**

**4.6.1 Activity Definition**

In this activity, the Our team provides support for the smooth running of software as agreed with the Customer, so as to assist user in phasing out the old system, fixing bugs if any, data posting and user training.

**4.6.2 Entry Criteria**

Support request from customer (any media, even a phone call/email) within the Warranty period

**4.6.3 Inputs**

****

Support request from the customer within warranty period

**4.6.4 Tasks and Procedures**

****

Upon receiving the support request from the Customer, the Change Request Log is updated.



Impact Analysis for the change requested is done.



If required, a change request is raised and handled according to the Configuration Management Process.



If the Customer reports a defect, then the Project Bug Report is updated and the defect is handled according to the Configuration Management Process.



If the Customer reports an issue which is neither a defect nor a CR, then the Project Manager resolves the same by meeting / discussing the issue with the Customer as per the agreement.