**Real Time Monitoring of Real Life Management Systems**

Software Requirements Specification

**SRS\_7C\_T8\_1.0**

**10/09/2017**

|  |  |  |
| --- | --- | --- |
| Sl.No | Name | USN# |
| 1 | NL Ramya | 01FB14ECS124 |
| 2 | Nachiketa Ramesh | 01FB14ECS126 |
| 3 | Nirmit Shetty | 01FB14ECS134 |
| 4 | Prerana Mehta | 01FB14ECS164 |
| 5 | Priyanka Narendra | 01FB14ECS166 |
| 6 | R Hariharan | 01FB14ECS172 |
| 7 | Rahul M | 01FB14ECS178 |
| 8 | Madhusudan H | 01FB15ECS429 |
| 9 | Nagaveni NK | 01FB15ECS433 |

Guide : **Prof. Phalachandra**

# **Revision History**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date** | **Description** | **Author** | **Approver** | **Comments** |
| 10/09/17 | SRS\_7C\_T8\_1.0  (Section 2.3) | R Hariharan |  |  |
| 10/09/17 | SRS\_7C\_T8\_1.0  (Section 2.1) | N L Ramya |  |  |
| 10/09/17 | SRS\_7C\_T8\_1.0  (Section 2.2) | N L Ramya |  |  |
| 10/09/17 | SRS\_7C\_T8\_1.0 (Section 1.2) | Nagaveni N K |  |  |
| 10/09/17 | SRS\_7C\_T8\_1.0 (Section 1.3) | Nagaveni N K |  |  |
| 13/09/17 | SRS\_7C\_T8\_1.0 (Section 1.2 scope) | N L Ramya |  | Change based on Review by team C2 - describing analytics in more detail. |
| 13/09/17 | SRS\_7C\_T8\_1.0(section 2.2 and 2.3) | N L Ramya |  | Change based on review by team C2 - Aligning block diagram |

# **Document Approval**

The following Software Requirements Specification has been accepted and approved by the following:

|  |  |  |  |
| --- | --- | --- | --- |
| **Signature** | **Printed Name** | **Title** | **Date** |
|  | <Your Name> |  |  |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

[**Revision History**](#_dr5jg52pgwqt) **2**

[**Document Approval**](#_pqykypyg8jkf) **2**

[**1. Introduction**](#_f3ma9jo6vizh) **4**

[1.1 Purpose](#_d6my04ca1rnm) 4

[1.2 Scope](#_xblnd0gkmkl6) 4

[1.3 Definitions, Acronyms, and Abbreviations](#_2gvk3kof148x) 4

[**2. General Description**](#_qaqjrfbwkw0s) **4**

[2.1 Product Perspective](#_goldg02oa6i8) 4

[2.2 Product Features](#_97w69k8jpvzg) 6

[2.3 User Characteristics](#_tuiqgem1zcfx) 7

[2.4 Operating Environment](#_u3jos3pz52t0) 8

[2.5 General Constraints](#_opbodph852bw) 8

[2.6 Assumptions and Dependencies](#_mckqjdtq7qqo) 8

[2.7 Documentation](#_xo949gcryiur) 8

[**3. Specific Requirements**](#_mqtwkrtjp4au) **8**

[3.1 Functional Requirements](#_p1nm9qgsvab8) 9

[3.2 Non-Functional Requirements](#_b1d6wg8v3ju7) 9

[3.3 External Interface Requirements](#_z337ya) 9

[3.3.1 User Interfaces](#_3j2qqm3) 9

[3.3.2 Hardware and Software in terms of how they would interact or how they would be executed](#_4i7ojhp) 10

[3.3.3 Software Interfaces](#_qyqu1maq1rm0) 10

[**4 Appendix A: Glossary**](#_ghxc3i4l3jlf) **10**

# **1. Introduction**

## **1.1 Purpose**

The purpose of this document is to provide a detailed description of the **Real Time Monitoring of Real Life Management** software. The document explains the features of the system, the interface provided by the system, the working of the system and the conditions for its operation.This document is intended for the developers of the software.

## **1.2 Scope**

1. The software product is **“Real Time Monitoring of Hotel Management System”** being used for the Nisarga chain of hotels in Bangalore.
2. The software’s reporting feature allows to view all the restaurant’s bills in real-time, reducing the dependency on staff for the sales reports. Live bills can be tracked for each restaurant outlet. Custom reports can be created as per needs and help make smart and data-driven decisions for the business with ease. The software ensures that the restaurant kitchen never runs out of stock by tracking the inventory in real-time by giving automatic reminders and alerts.

Implementation of a recipe management system which would automatically deduct the amount of stock consumed based on the recipe of each dish isn’t currently supported.

1. The system contains three main subsystems, the management end (hotel staff), the cloud computing system (analytics end) and the monitoring end (real-time updates/reports). The current management / monitoring system at the Nisarga Hotel is mostly manual accounting and checking. The technical backbone provide by us will ease the job of the Nisarga hotel manager and make the hotels functioning more efficient.
2. Functioning:

* **The Management End :** Here the data is being accumulated and transferred within and outside the system. It consists of local database **(Mysql)** where the data gets stored before being pushed on to the cloud.
* **The Cloud Storage :** The centralized system where all the hotel details gets stored (**Azure**). Here analytics on the data is performed to generate reports. The new updates are sent to the electron application. The analytics we plan to include are such as the most active waiter and the most trending dish. We also plan to have bar graphs showing the number of orders per day and per month and also the top most ordered dishes. We are also including the number of tables filled currently and average income to the outlet of the restaurant.
* **Monitoring End :** Two applications would be provided to monitor their hotels.

1. **Progressive Electron Application :** which provides all the information regarding the data accumulated. It should have reports showing the statistics on the working of the system and show all the information logged. The analytics we plan to include are such as the most active waiter and the most trending dish. We also plan to have bar graphs showing the number of orders per day and per month and also the top most ordered dishes. We are also including the number of tables filled currently and average income to the outlet of the restaurant. This could also be imported as a mobile application

## **1.3 Definitions, Acronyms, and Abbreviations**

1. RTMS : Real Time Monitoring System
2. USP : Unique Selling Proposition
3. MTBF : Mean Time Between Failure

# **2. General Description**

## **2.1 Product Perspective**

The real-time monitoring software replaces the traditional manual supervision system. The entire system consists of 4 major clients. In addition to these, the system also has additional staff to maintain and keep the system up to date. There are multiple roles that are responsible for this including the project lead, web application lead, electron application lead and database manager.

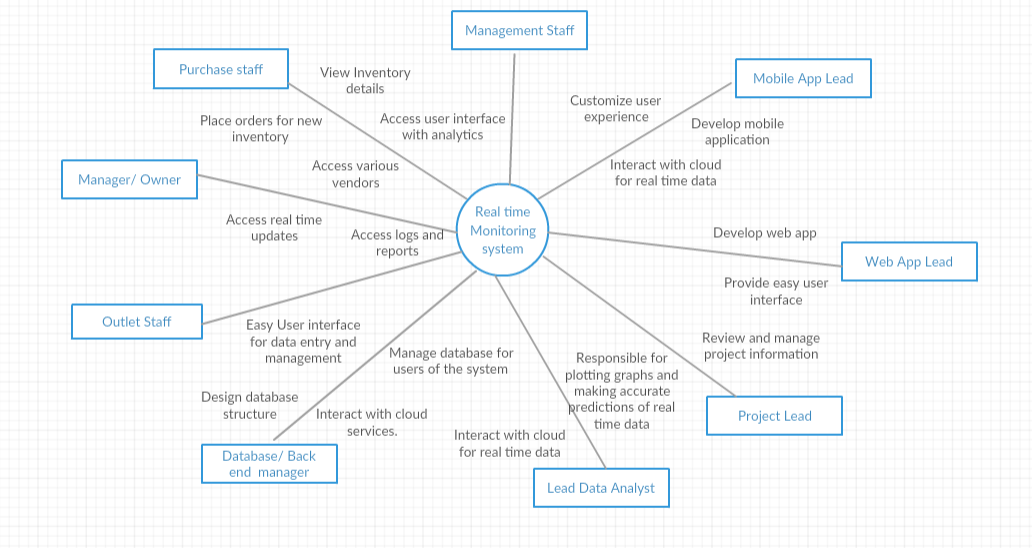


Figure 1. Overall picture of the system

## **2.2 Product Features**

We divide the project into 3 major parts with 4 types of users of the system - The owner, outlet staff, purchase staff and management staff:

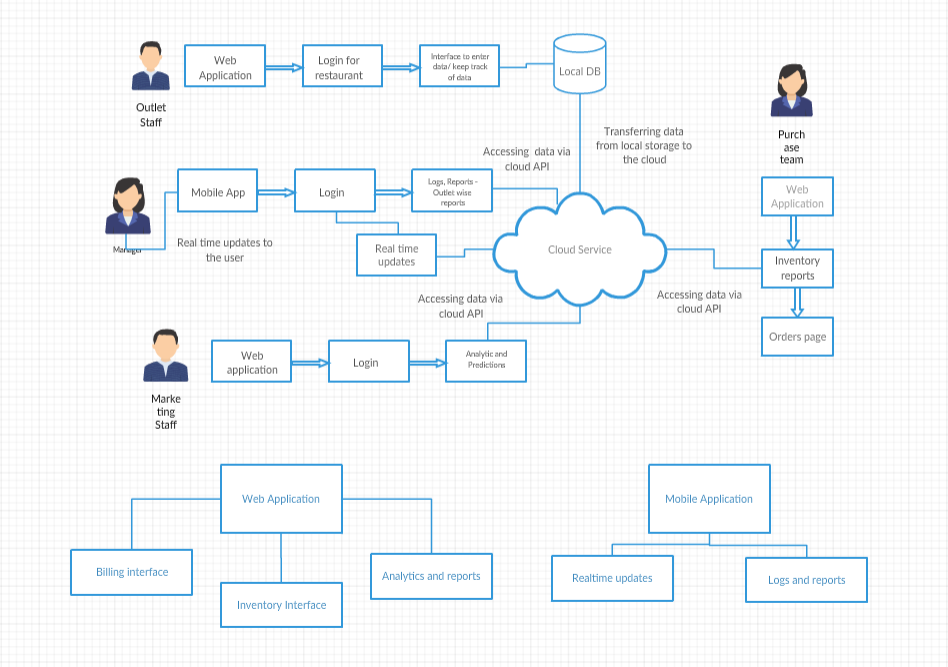
1. Web application for data entry and management: This is mainly used by outlet staff, purchase staff and management staff. We propose to make a web application for this purpose.

* The outlet staff will be provided with functionalities such as: Interface to easily view billing data, update new orders and transactions to the system.
* The purchase staff will have access to: Inventory manager which will keep track of all the goods and their respective quantities. Furthermore, they will be able to places orders to different vendors
* The management staff can do the following: View analytics of previous orders - make predictions as to what the customer currently requires.

1. All of the data that is entered by the outlet staff is flushed to cloud. This data could be from multiple outlets. There is one centralised cloud where all the data gets stored.
2. Application for the owner of the restaurant where he gets real-time updates of actions taking place in the restaurant and also has access to logs and reports.

**The main project features are:**

* View all restaurant bill-wise reports in real time.
* Inventory Management module to keep tight control on stock and prevent wastage and theft.
* Live-data notified on electron application to eliminate sabotage and data tampering.
* Web application for logs and reports on the functioning of the system.
* View and analyze the sales report of each restaurant outlet in detail.
* Transferring of data to a centralised cloud system periodically.

** Figure 2**. *Major features and how they relate.*

## **2.3 User Characteristic**

*The ‘****User-Space’*** *of our product can be divided into two main categories:*

* *The Hotel Management* ***Staff*** *(Billing & Accounting Staff, Inventory Management Staff & Waiters)*
* *The Hotel* ***Manager/Owner***

|  |  |
| --- | --- |
| ROLE | CHARACTERISTICS |
| Billing & Accounting Staff | The mentioned staff generates bills for the orders placed and then manages the total money/revenue collected for each day. The staff categorizes the payment methods and tallies the total cash received with the total amount cash payments received as on that day. His/Her task is also to report the account details on a daily basis. |
| Inventory Management Staff | Inventory stats is managed by the mentioned staff. He/She requires to update the usage of the items in the stock on a periodic basis, say every three hours. He/She can then set the menu based on the available items in the inventory. This role can be given to the head chef of the kitchen. He/She can also place request for order of items with appropriate information about the item and the level of urgency of its requirement. |
| Waiters | The role of the waiter is to take in orders from the table and generate **Kitchen Order Ticket (KOT)** for the same. He/She coordinates with the Billing & Account staff during the payment of the order. |
| Hotel Manager/Owner | He/She monitors the whole functioning of the system. He/She checks the account and revenue information provided by the Billing & Account staff. He/She checks the request made by the Inventory Management staff and places the order based on its urgency. The Manager/Owner basically controls the whole process. |
| Product Customer Support | The customer support will be the product company’s employees who try to help out clients with the problems they are facing. They also communicate with the clients about latest software updates. If any major problem is faced then they communicate to the product development team.. |

## **2.4 Operating Environment**

|  |  |
| --- | --- |
| OE-1: | System is not dependent on geographical areas. |
| OE-2: | System shall operate in newest versions of all web browsers. |
| OE-3: | Users need to login to access the system, with a hierarchy of privileges based on the account. |
| OE-4: | Data is filled at the interface provided to the outlet staff. |
| OE-5: | Service is real time critical, but some latency can be tolerated, and the  frontend must inform the user of this latency (last updated time). |
| OE-6: | Company owned data will be stored in the database, so the App Engine database must be secure. |
| OE-7: | jQuery javascript library will be needed to create the user interface for the system. |

## **2.**5 **Assumptions and** Constraints

**This section presents the assumptions that will affect the proposed system, such as:**

1. The first assumption made is that restaurants have made all their information computerized.
2. We require an active internet connection to send new data to the cloud.
3. Availability of resources and information for different implementations to make the system more generic.
4. The end user must be having a device with a browser.

# **3. Specific Requirements**

## **3.1 Functional Requirements**

|  |  |  |
| --- | --- | --- |
| FR1 | Admin Accounts | The system should provide a user system with administrative powers. We implement this by only giving the owner the power to add new accounts |
| FR2 | Data Entry | The system shall data entry for bills, orders and inventory. We do this in our management end. We have html, javascript, php and a sql database to store all this data locally. |
| FR3 | Real Time Data Synchronisation | The system shall synchronise this data with the cloud service. We are using Azure cloud. We have API’s that are called to update and retrieve from the cloud. These are REST API’s |
| FR4 | Aggregator | The system should provide an administrator accessible cloud service which holds the aggregated data. We have Azure cloud with all the data |
| FR5 | Analyser | The system should provide a service at the cloud end that performs analytics on the aggregated cloud data. We have REST API’s running that do analytics on the AZURE cloud |
| FR6 | Visualiser | The system should provide a frontend that provides visualisation of the data and its analytics. We use an electron app to do this. |

## **3.2 Non-Functional Requirements**

* Reliability: Annual Failure Rate of 0.1-0.5 % can be expected (Due to cloud service provider). Our system is reliable as all updates to the cloud are stored in local db even without internet connection and is flushed when we get internet.
* Security: Every update to the system will be logged.

We make sure that the security requirements are met in the following ways:

1. User can’t login without a valid username and password.

2. Only an admin of the webpage can create accounts for the manager. The manager is in turn responsible for creating accounts for waiters and other staff. Hene no external user can create accounts.

3. Password is encrypted when it is sent via POST.

* Survivability: The system will survive any natural disaster as it will maintained on the cloud, and local catastrophes will not affect it.

## **3.3 External Interface Requirements**

### **3.3.1 User Interfaces**

UI-1: For the hotel staff: Web application shall permit entry of orders as and when they come in. It will also provide an interface for the user to check if the transaction is complete and flushed to the cloud.

UI-2: For the hotel owner: An electron app will display real time updates and show logs and reports of various outlets in one place

UI-3: For the management staff: A web application will show real time analytics based on the data coming in from the cloud.

### **3.3.2 Hardware and Software in terms of how they would interact or how they would be executed.**

Systems required:

1. A system for data entry - Personal computer
2. Server to store the data on - Azure cloud
3. System to view analytics on - Electron app

Cloud API’s to be used for interaction with the server.

### **3.3.3 Software Interfaces**

SI-1: Management end: to enter the data of the restaurant.

1. Page to enter orders by waiter
2. Page to enter inventory orders by inventory staff
3. Page to enter bills
4. Page to set up accounts

SI-2: Database - The system shall communicate with a database through a programmatic interface for the following operations:

SI-2.1: To update the order, inventory and bill details as they come in.

SI-2.2: To allow owner to view details such as logs and reports.

SI-2.3: To allow management staff to access data, and flush intermediate results to cloud.

SI-3: Cloud - Azure cloud has a database with franchise ID attached. We have REST API’s to update to the cloud and access from the cloud.

SI-4: Electron app - This displays analytics that was talked about in previous sections. It is a desktop app but can also be used on a mobile.

# 