|  |
| --- |
|  |
| Beautywares CRM |
| Design Specification |
|  |
| **Arunkumar Krishnamoorthy**  **Ram Software Engineering Labs Pvt Ltd** |
| **4/28/2015** |

|  |
| --- |
| This document details system design for BeautyWares CRM. |

**Document Revision History**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Version No.** | **Date** | **Author** | **Reviewed By** | **Status** | **Comment** |
| 0.1 | 05-May-2015 | Arunkumar K | -NA- | Draft | Initial Creation |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**Acronyms and Abbreviations**

|  |  |
| --- | --- |
| **Abbreviation** | **Description** |
| MEAN | Though the original extension is MongoDB ExpressJS AngularJS NodeJS stack, in the context of the project it refers to MySQL ExpressJS AngularJS NodeJS stack |
|  |  |
|  |  |

Table of Contents

[1 Introduction 2](#_Toc388037948)

[1.1 Document Outline 3](#_Toc388037949)

[1.2 Document Description 5](#_Toc388037950)

[1.2.1 Introduction 5](#_Toc388037951)

[1.2.2 System Overview 6](#_Toc388037952)

[2 Design Considerations 6](#_Toc388037953)

[2.1 Assumptions and Dependencies 6](#_Toc388037954)

[2.2 General Constraints 6](#_Toc388037955)

[2.3 Goals and Guidelines 7](#_Toc388037956)

[2.4 Development Methods 7](#_Toc388037957)

[3 Architectural Strategies 7](#_Toc388037958)

[4 System Architecture 8](#_Toc388037959)

[4.1 Subsystem Architecture 9](#_Toc388037960)

[5 Policies and Tactics 10](#_Toc388037961)

[6 Detailed System Design 10](#_Toc388037962)

[6.1 Classification 11](#_Toc388037963)

[6.2 Definition 11](#_Toc388037964)

[6.3 Responsibilities 11](#_Toc388037965)

[6.4 Constraints 11](#_Toc388037966)

[6.5 Composition 11](#_Toc388037967)

[6.6 Uses/Interactions 11](#_Toc388037968)

[6.7 Resources 12](#_Toc388037969)

[6.8 Processing 12](#_Toc388037970)

[6.9 Interface/Exports 12](#_Toc388037971)

[6.10 Detailed Subsystem Design 12](#_Toc388037972)

[7 Glossary 13](#_Toc388037973)

[8 Bibliography 13](#_Toc388037974)

# Introduction

This document describes high level design of BeautyWares CRM solution.

## Document Outline

This design document, used for high-level design, is a "living document" in that it gradually evolves to include low-level design details (although perhaps the "Detailed Design" section may not yet be appropriate at the high-level design phase).

# Design Considerations

This section describes many of the issues which need to be addressed or resolved before attempting to devise a complete design solution.

## Assumptions and Dependencies

Following are the assumptions or dependencies regarding the software and its use.

* First iteration doesn’t include any integration with TallyERP9
* All end users access this system via web browser
* Though end users will access the system via hand-held smart phones and tablets, no separate app will be developed as part of this solution
* The system would be shutdown end of every day

## General Constraints

Following are the constraints that have a significant impact on the design of the system's software.

* End users of this system are not tech-savvy. Hence the UI has to be highly intuitive and user-friendly.
* This solution would be deployed in Beautyware’s intranet in a desktop that already runs Tally ERP9 server. As a result, design should take into consideration the limited resource available for this system in terms of hard disk space, RAM and processing speed.
* The UI design has to be responsive to work across different screen sizes.
* As the operation and maintenance cost needs to be kept to very minimum, open source software, servers and frameworks have to be used.
* As the system needs to be integrated with Tally ERP9 in later iterations, design should consider enterprise data to come from master database in Tally.
* As the solution needs to be taken to other customers too, design should not be tightly-coupled to Beautyware’s specific requirements.
* Only authenticated and authorized users should access any part of this system.
* Data entry should be faster. Accessibility should be assisted with keyboard shortcuts. In addition, layout of fields should assist faster data entry by end users.

## Goals and Guidelines

* As users are not so Tech Savvy people, User Interface should be very intuitive
* As users are not naturally English speaking, all labels and messages should be simple and clear. If possible, labels need to be internationalized to specify in local languages
* System has to be accessed within BeautyWares intranet through web browser concurrently by multiple users
* System should be accessible through desktops, tablets and smartphones. Hence the UI should be responsive across multiple screen sizes.
* System should be low in maintenance effort and cost. Hence the complete technology stack should be on an open source stack with user-friendly administration and maintenance interface.

## Development Methods

System will be built using prototyping development methodology. Initially a prototype will be developed to demonstrate

1. Complete user navigation and user experience
2. Interaction and integration between different layers
3. Validation of technology choices in each layer
4. Database choice – either conventional RDBMS on MySQL or Mongo DB

# Architectural Strategies

1. UI / Presentation layer should leverage Javascript and CSS framework that ensures
   1. Processing power of client machines is leveraged to generate and manipulate HTML DOM. This also implies that load on the server to generate HTML is minimized. Server should be utilized for generating and processing data and business logic
   2. Compatibility across different web browsers
   3. All data entry validations for required fields, data format are done in client side and appropriately prompted to the user
   4. Responsive across different screen sizes
2. Middleware should be based on layered approach where there is clear demarcation of Model, View, Controller
3. Strategically – based on market trends and Ram SE Labs’ technology roadmap – it is decided to use Javascript stack on Node JS. As part of prototyping it should be decided which Javascript stack should be chosen on Node JS to build all layers – either MEAN stack or based on Sails JS.
4. JSON should be chosen as the de-facto standard to exchange data between client and server
5. As this CRM solution should be taken to other customers in different industries, care should be taken to decouple customer specific details from the core business logic.
6. The complete system should be platform independent. In case of Beautywares, the system should be tested and deployed on Windows Operating System.
7. System should provide different interface points / APIs so that it can be integrated with other legacy systems to pull / push customer data, user data, order / sales data. With respect to Beautywares, system should be able to interface with Tally ERP9 to exchange enterprise data.

# System Architecture

## Data Model