|  |  |
| --- | --- |
|  | **Venue Booking System**  **Technical Design Document** |
| |  |  |  |  | | --- | --- | --- | --- | |  | **Prepared By / Last Updated By** | **Reviewed By** | **Approved By** | | **Name** | S Shamini  Dhanashree Patil  Bhamini Poorna  Bartick Chatterjee  Domakonda Raviteja | Sainath |  | | **Role** | POD Members |  |  | | **Signature** |  |  |  | | **Date** | 19-05-2022 |  |  | |
|  |

Table of Contents

[1.0 Introduction 4](#_Toc14171023)

[1.1 Purpose of this document 4](#_Toc14171024)

[1.2 Project overview 4](#_Toc14171025)

[2.0 Solution Summary 5](#_Toc14171026)

[2.1 Scope 5](#_Toc14171027)

[2.2 Assumptions 5](#_Toc14171028)

[2.3 Dependencies 5](#_Toc14171029)

[2.4 Risks 5](#_Toc14171030)

[3.0 Schematic Diagram 6](#_Toc14171031)

[4.0 System Design 7](#_Toc14171032)

[4.1 Proposed design 7](#_Toc14171033)

[4.2 Component inventory 7](#_Toc14171034)

[5.0 Database Design 8](#_Toc14171035)

[5.1 Data Model 8](#_Toc14171036)

[5.2 Tables Structure 8](#_Toc14171037)

[6.0 Appendices 9](#_Toc14171038)

[6.1 Glossary 9](#_Toc14171039)

[6.2 Other 9](#_Toc14171040)

[7.0 Terms & Conditions 10](#_Toc14171041)

[8.0 Change Log 10](#_Toc14171042)

# Introduction

## Purpose of this document

The purpose of this document is the technical design, component details and Database design. This will also capture the scope, assumptions, risk, dependencies of this project.

## Project Overview

Venue booking system is a web application in which service providers or a venue owner can register their venues and customers can book venues for events as per their location, availability and interest. This makes the workload simple for customers to book a venue for an event they need to search for without visiting the actual place. In this application, the user needs to be registered and log in for further advantages.

The customer can search for venues by specifying names or by giving the location if venues are present in that particular location. It’s an application in which dealers and customers can select events based on their location, availability, and area. This makes the workload simple for users to book an event. In this application, the user needs to be registered and log in, the user will see the full information about the venue and if the person is interested, he will book the request, which will be sent to the organizer who added the particular venue and if it’s available according to the date he will accept and the user will be able to see the status that the booking is confirmed.

Software Requirements:

* The system will require a Windows Operating System.
* Eclipse, MySQL Workbench, Tomcat Server

Technologies:

* Frontend: React
* Backend: MySQL, Spring Boot with REST

Hardware Requirements:

* The system should be able to run and access through PC/Laptop. The system software requirements are as follows:
* RAM minimum 4GB
* Processor: Intel Pentium or higher

# Solution Summary

## Scope

* This document applies specifically to the Venue Booking System.
* The System will be implemented using technologies: Spring Boot with REST, MYSQL.
* The document will provide details of design elements for all development done to meet the specified system.
* The document also includes any configuration details that are applicable for the design components.

## Assumptions

* Tomcat will be used to host application.
* Development team will use developers' machine for initial development and UI testing.
* Once development is complete, the platform code will be merged and deployed to POD leaders’ Cognizant machine for development integration validation.

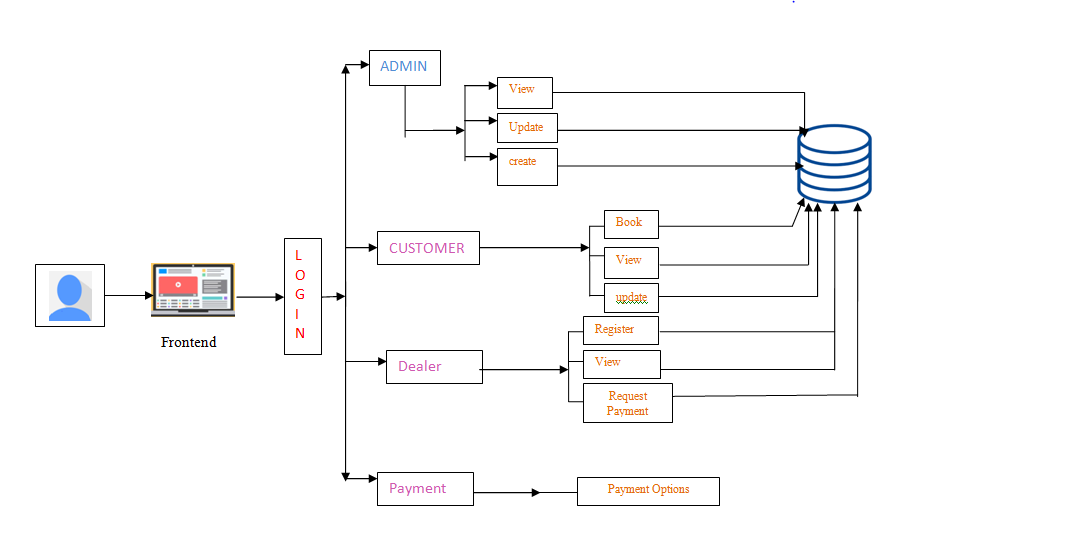
## Dependencies

Availability of software's on the cognizant machine-like Eclipse, MySQL

## Risks

It might be possible that the data stored might get lost due to the damage of storage

# 3.0 Schematic Diagram



# 4.0 System Design

## 4.1. Proposed design

* Dealer:

The dealer needs to register the application before getting logged in. After getting logged in, the application welcomes Dealer with Dealer home screen, showing the functionalities of Dealer. He can add venues and view the Customer requests and accepts.

* Customer:

Customer needs to be registered into the application before getting logged in, after getting the login access Customer will search for the venues based on the location. Then Customer will see the list of venues available in that location and book any one of them and the request will be forwarded to the Dealer. Dealers can take a view of customer requests and can accept or decline the request. When the request is accepted by the dealer, the logged in customer can see that their booking has been confirmed.

* Admin:

Admin will be able to view the list of registered customers ,dealers and the venues registered by the dealers, The admin will also be able to modify the venue list.

* Payment:

The Payment form has to filled by the customers. It contains various inputs such as , Customer Username, Dealer Username, Amount and Payment options. The dealer firsts requests for the payment and then the customer accepts the request and pays the given amount after which the venue request will be accepted by the dealer

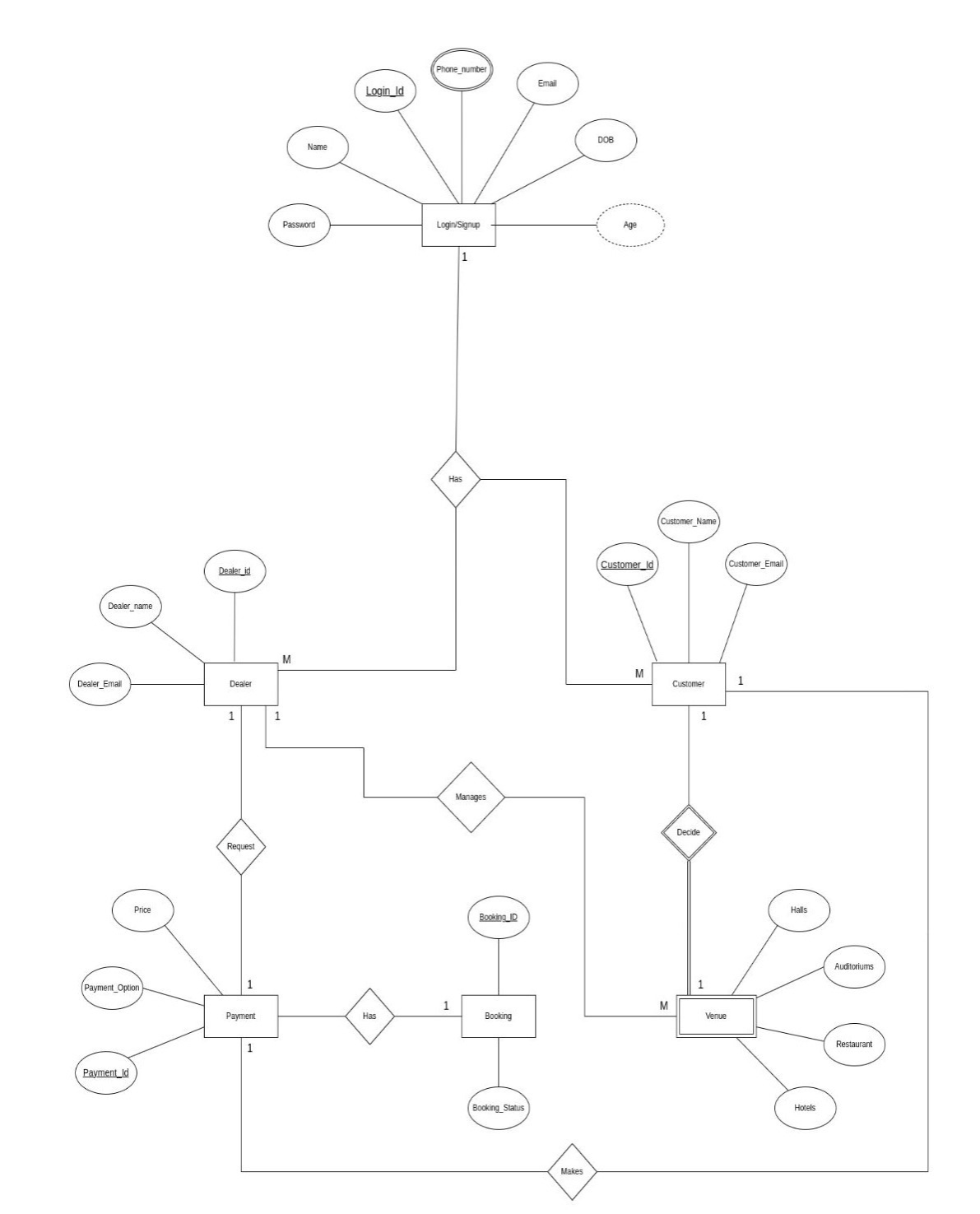
After the payment is successful , the customer and dealer will be provided with the transaction id.

## 4.2. Component inventory

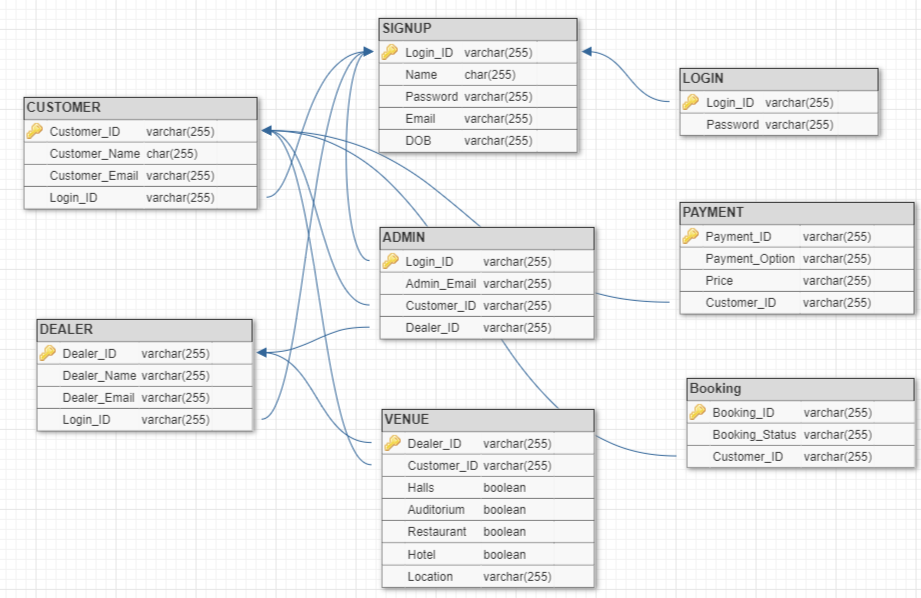
* Main screen
* Signup
* Login
* Admin
* List of venues
* Manage venues
* Logout
* Dealer
* Home screen
* Add venues
* My Requests
* Logout
* Customer
* Home screen
* List of venues
* Customer information
* My Bookings
* Logout
* Payment
* Payment options
* Price

# 5.Database Design

## 5.1. Data Model



## 5.2. Tables Structure



# 6.Appendices

## 6.1. Glossary

|  |  |
| --- | --- |
| **Acronyms** | **Definition** |
| REST | Representational State Transfer |
| UI | User Interface |

## 

## 6.2.Other

## 7.Terms & Conditions

***Disclaimer: Please do not circulate or distribute this document outside of Cognizant Network, We have a Zero Tolerance Policy. Kindly adhere to 100% Compliance at all times.***

# 8.Change Log

*Please note that this table needs to be maintained even if a Configuration Management tool is used.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version Number | Changes made | | | |
| V<n.n> | *<If the change details are not explicitly documented in the table below, reference should be provided here>* | | | |
| Page no | Changed by | Effective date | Changes effected |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |