Online Booking System Software Requirement Specification Document

# Business Requirement

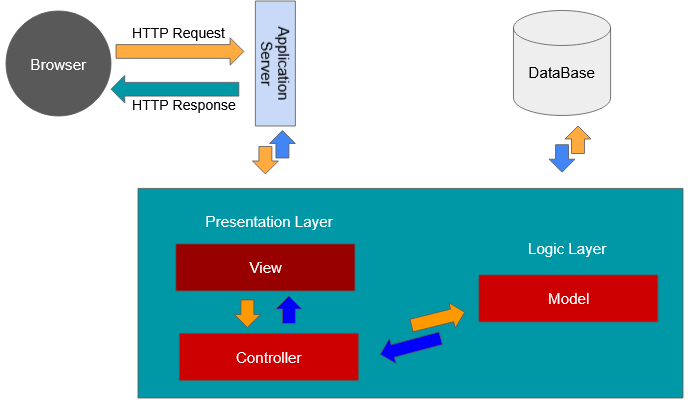
One hospital in Bhutan wants to keep their doctors and patients visiting and prescriptions online to track the medication and the treatment summary. The patient will be able to book online appointments to the hospital's doctors with their symptoms and running medications. Once the patient confirms the slot of appointments of the doctors, the doctor’s secretary will be able to check the patients details before they confirm the patient visit. The payments will be online after the secretary confirms the appointments. The hospital management will be able to see the reports for patient visit and the deposit of the patient. Once the doctor has seen the patient, he will write the prescription advice and this prescription advice can be visited later by the doctor and patient and management.

# Software requirement specification

We will develop an application that will provide a login page for patient, the system will register the patient from a registration page. The patient will provide an address, gender, and mobile number on the registration form. All the forms will have validations. There will be a safe routing system so no one can enter the dashboard without login. Only Admin can register new doctor, employee. On patient’s dashboard patient can take appointment and also can see previous treatment and medicines. Patient can pay fees after doctor’s secretary approves the appointment. On doctor’s dashboard doctor can see appointments and patient’s medical details. Admin can see total registered patients, employee and doctors count on dashboard. Admin can remove any doctor & any patient.

# Software Design Specification

### Overview of the proposed system:



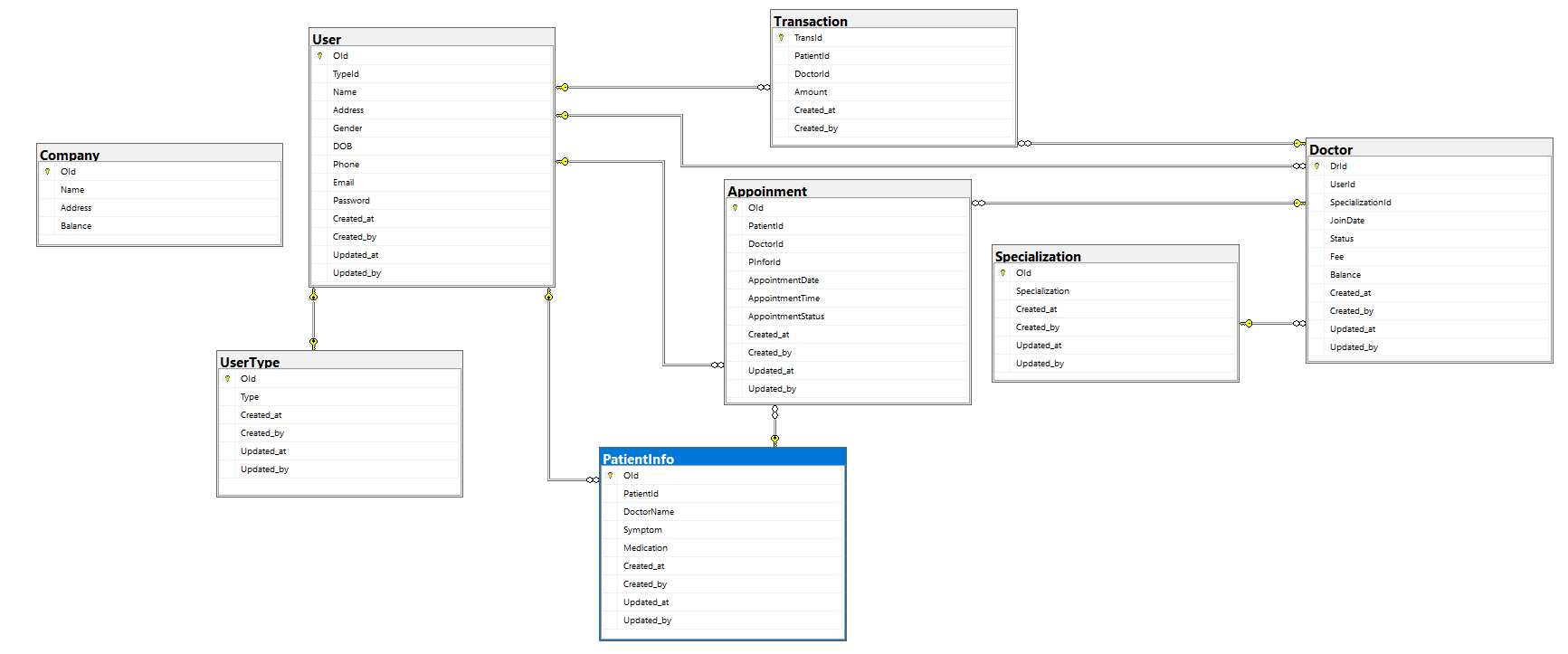
### UI Design:

### Database Design:

#### Table Name:

* User Table
  + OId (int)
  + TypeId (int, foreignKey = “UserType = OId”))
  + Name\* (varchar 64)
  + Address (varchar 128)
  + Gender\* (Varchar 64)
  + DOB\* (date)
  + Phone\* (varchar 64)
  + Email\* (Varchar 64)
  + Password\* (Varchar 64)
  + Created\_at (datetime)
  + Created\_by (Varchar 64)
  + Updated\_at (datetime)
  + Updated\_by (Varchar 64)
* UserType Table
  + OId (int)
  + Type (Varchar 64)
  + Created\_at (datetime)
  + Created\_by (Varchar 64)
  + Updated\_at (datetime)
  + Updated\_by (Varchar 64)
* PatientInfo Table
  + OId (int)
  + PatientId (int, foreignKey = “User = OId”)
  + DoctorName (Varchar 64)
  + Symptom (Varchar 128)
  + Medication (Varchar 128)
  + Created\_at (datetime)
  + Created\_by (Varchar 64)
  + Updated\_at (datetime)
  + Updated\_by (Varchar 64)
* Appoinment Table
  + OId (int)
  + PatientId (int, foreignKey = “User = OId”)
  + DoctorId (Varchar 64, foreignKey = “Doctor = DrId”)
  + PInfoId (int, foreignKey = “PatientInfo = OId”)
  + AppointDate (Date)
  + AppointTime (time 7)
  + AppointStatus (Varchar 64)
  + Created\_at (datetime)
  + Created\_by (Varchar 64)
  + Updated\_at (datetime)
  + Updated\_by (Varchar 64)
* Specialization Table
  + OId (int)
  + Specialization (Varchar 64)
  + Created\_at (datetime)
  + Created\_by (Varchar 64)
  + Updated\_at (datetime)
  + Updated\_by (Varchar 64)
* Doctor Table
  + DrId (Varchar 64)
  + UserId (int, foreignKey = “User = OId”)
  + SpecializationId (int, foreignKey = “Specialization = OId”)
  + JoinDate (datetime)
  + Status (Varchar 64)
  + Fee (Decimal 19,2)
  + Balance (Decimal 19,2)
  + Created\_at (datetime)
  + Created\_by (Varchar 64)
  + Updated\_at (datetime)
  + Updated\_by (Varchar 64)
* Transaction Table
  + TransId (Varchar 64)
  + PatientId (int, foreignKey = “User = OId”)
  + DoctorId (int, foreignKey = “Doctor = DrId”)
  + Amount (decimal 19,2)
  + Created\_at (datetime)
  + Created\_by (Varchar 64)
* Company Table
  + Name (Varchar 64)
  + Address (Varchar 128)
  + Balance (decimal 19,2)

#### Database Diagram:



# Software Development

### Tools:

* C#
* Asp.net Core
* JavaScript
* jQuery
* MSSQL Server
* Visual Studio 2019