NATURAL HEALING CENTER ONLINE SYSTEM

SOFTWARE DESIGN SPECIFICATION

SUBMITTED TO
Prof Kumar K

Submitted By
Utkarsh Sharma 12BCE0192
Poorvi Narang 12BCE0450

Table of Contents

1. INTRODUCTION

1.1 Purpose

The Software Design Document is made to outline the software design and architecture of a Natural Healing Center-Online System in detail. This document will highlight the understanding of the system design.

1.2 Scope

The Software design document will specify how to accomplish the functional and non-functional requirements captured in Software Requirement Specification (SRS). This document will assist programmers to understand the high level components, sub systems, interfaces and database design in detail.

1.3 Intended Audience

This document is intended for developers and administrative staff of Natural Healing Centre.

1.4 Glossary

Class Diagram: A class diagram is an illustration of the relationships and source code dependencies among classes in the Unified Modelling Language (UML). In this context, a class defines the methods and variables in an object, which is a specific entity in a program or the unit of code representing that entity.

Entity Relation Diagram: An ERD is a data modelling technique that can help define business processes and can be used as the foundation for a relational database.

Sequence Diagram: A Sequence diagram is an interaction diagram that shows how processes operate with one another and what is their order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence.

MVC: Model–view–controller (MVC) is a software architectural pattern for implementing user interfaces. It divides a given software application into three interconnected parts, so as to separate internal representations of information from the ways that information is presented to or accepted from the user.

1.5 References

1. http://www.cs.iit.edu/~oaldawud/CS487/project/software_design_specification.htm
2. http://www.slideshare.net/RaviYasas/example-for-sds-document-in-software-engineering
3. http://en.wikipedia.org/

1.6 Document Overview

Following chapters describe the system design and its architecture. It specifies the dependencies between modules and the architectural patterns of the whole system.
It also describes the user interface, activity diagrams, design patterns and database design for the required system in detail.

2. ARCHITECTURAL DESIGN

2.1 High Level components and Interfaces

2.1.1 Components

- Authentication Component
  This is a login system that generates a session for a particular user after password verification.
- Patient component
  This is one of the key components of the Natural Healing Center System. This is where the user logs in and can view his appointments. He/ she can also view her past medical records and prescriptions and make payments online.
- Doctor component
  In this component the doctor can view his appointments for the day and post a prescription for any patient.
- Manager component
  The manager can view all the appointments and cancel the appointments. He can also generate billing of the Healing Center.

2.1.2 Interfaces

- Authentication Component
  Login Screen: It consists of a username and password field for user to input details. It creates an active session for a registered user.
  Register Screen: It has the following input fields: - Name, Phone Number, Email, Age, Gender, Address, Username, and Password.

- Patient Component
  View and Book Appointment: It consists of Book appointment button and date preference of the user.
  Make Payment: It is a payment gateway module for a user to make the necessary payment online. It consists of a Card Type, Card number, Password and an amount field for the user to input data and make the payments
  View Feedback: All users can view the feedback in a list view in chronological order.

- Doctor Component
  View Schedule: A doctor can view his schedule for the day and look into the past records of the patient prior to the appointment.
  Write Prescription: It consists of a drop down option where a doctor can select the patient and a text field where he can enter the required prescription.
  View Medical History of Patient: The doctor can view a patient’s medical history based on his username.

- Manager Component
  Appointment Dashboard: The user can view all the appointments and give a conformation date or cancel the appointment accordingly. A button to confirm and a text field to send the time will be available along with a cancel button.
  Generate Billing: The manager can generate all the transactions of the system in the past.

2.2 Architectural Styles and Patterns

This model describes the methodology followed for development of the online system. The Natural Healing Centre Online System will follow MVC (Model-View-Controller) architecture, 3 tiers Client-Server Architecture and Object Oriented Design. Both are described in the section below.

2.2.1 MVC (Model-View-Controller)

Model–view–controller (MVC) is a software architectural pattern for implementing user interfaces. It divides a given software application into three interconnected parts, so as to separate internal representations of information from the ways that information is presented to or accepted from the user.

Model - The lowest level of the pattern which is responsible for maintaining data.

View - This is responsible for displaying all or a portion of the data to the user.

Controller - Software Code that controls the interactions between the Model and View.

We have used MVC architecture because there are multiple components of the system and all the all of them are independent from one another. Also there are many interactions on the page.

2.2.2 Three Tier Client-Server Architecture

Three-tier architecture (often referred to as n-tier architecture) is a client–server architecture in which presentation, application processing, and data management functions are physically separated.

We have separated the three tiers so as to make independent components to run efficiently with one another

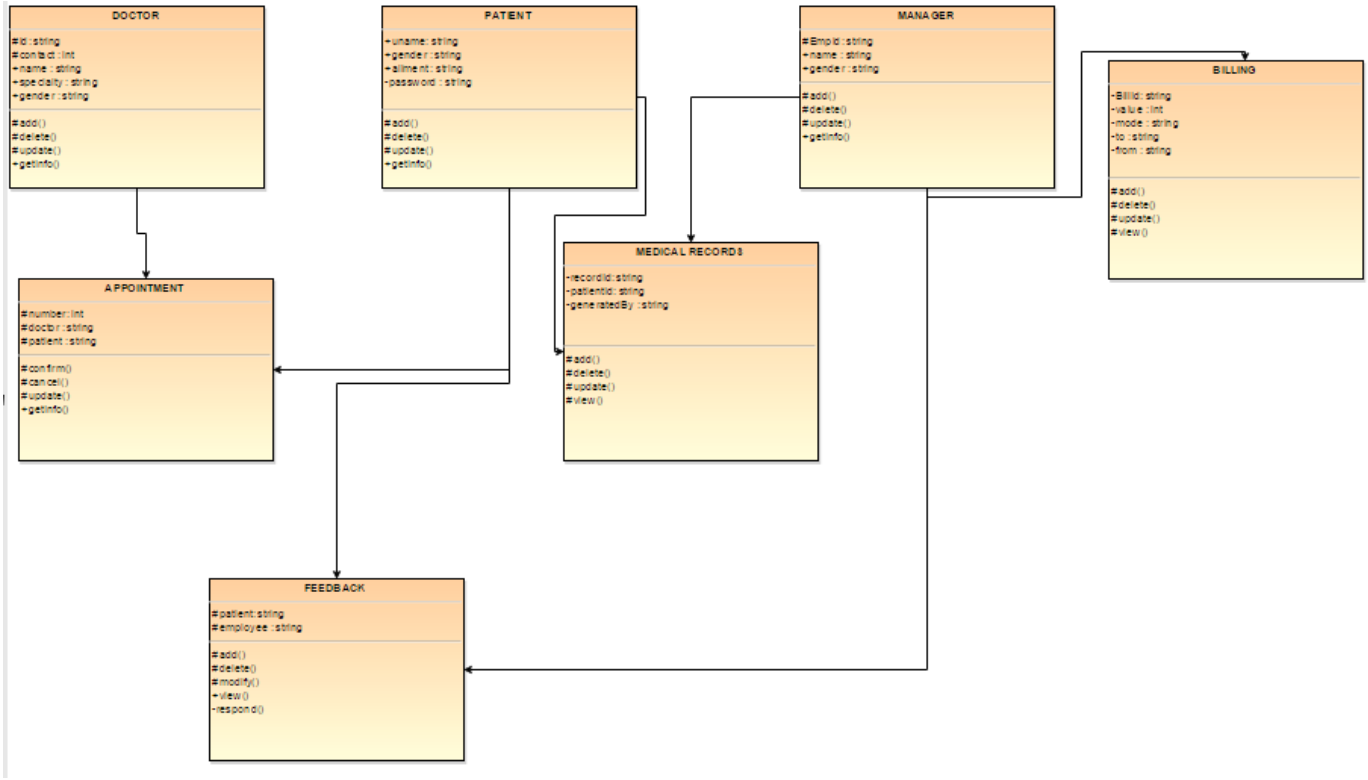
2.2.3 Object Oriented Design

An object contains encapsulated data and procedures grouped together to represent an entity.

We have used OOPs design concept to Improve software maintainability, faster development, higher understandbitilty of the system.

3. COMPONENT DESIGN

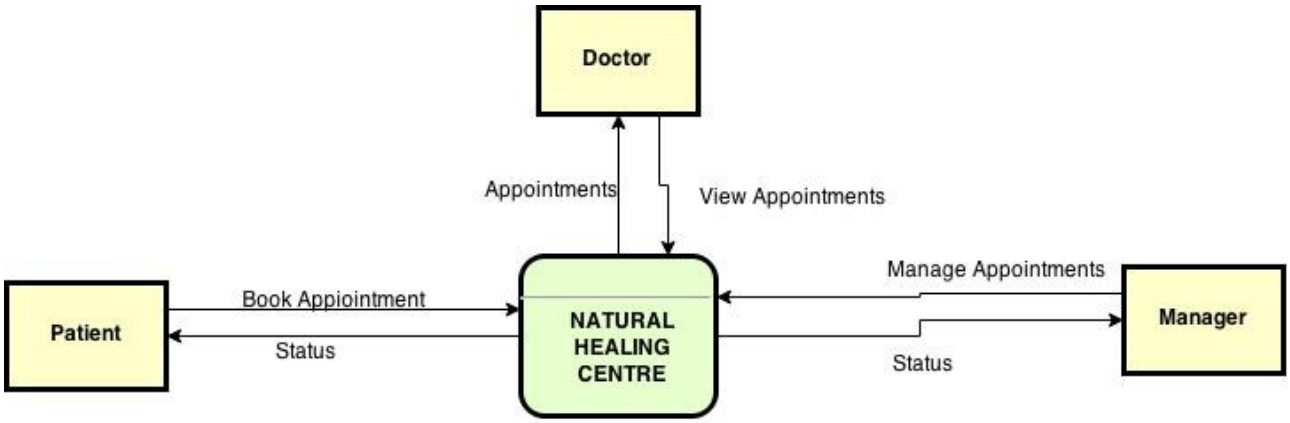
3.1 Class Diagram



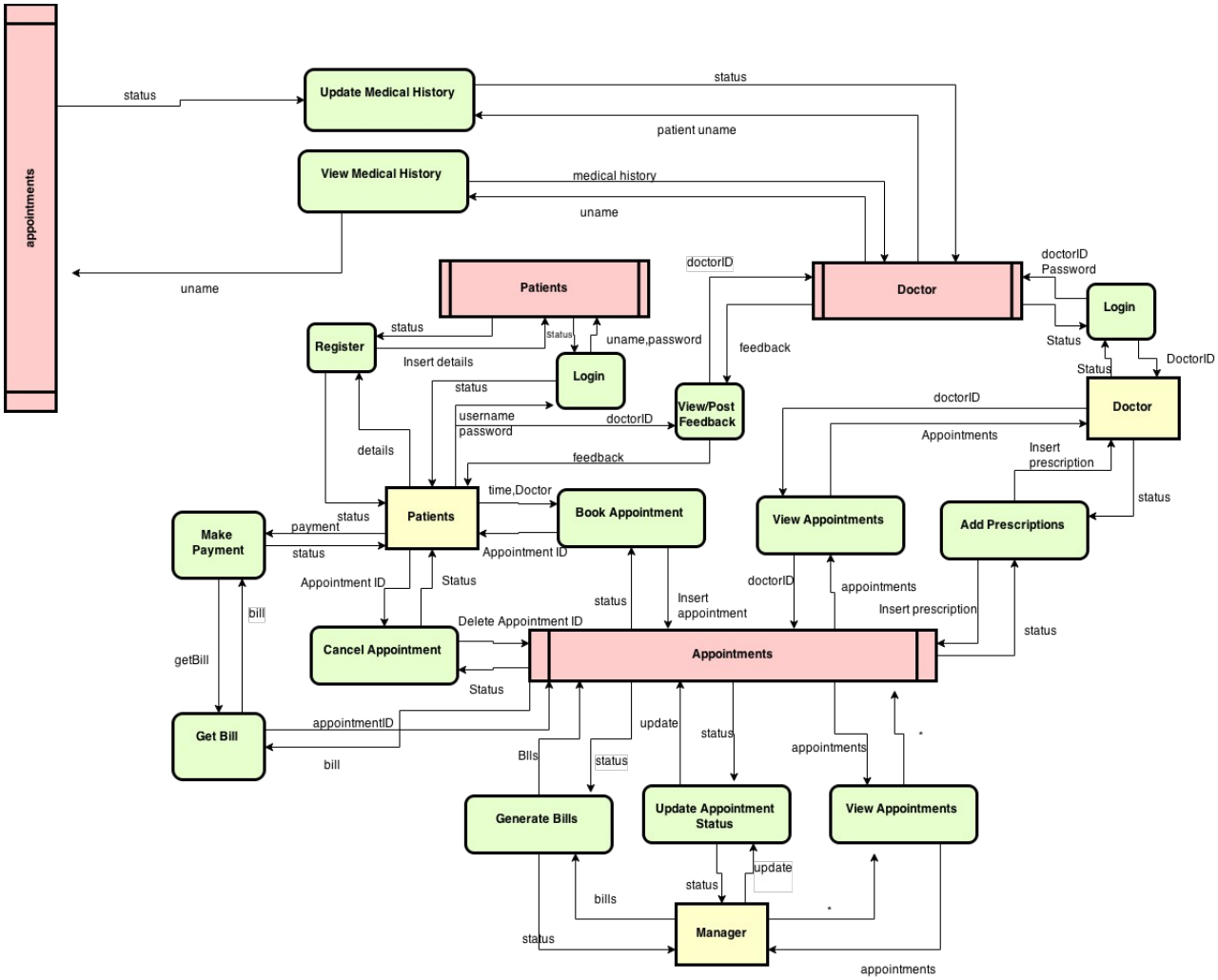
The main classes are Doctor Class, Patient class, Manager Class, Appointment Class and Pharmacy class. Class diagram is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations and relationships among objects.

3.2 Data Flow Diagram

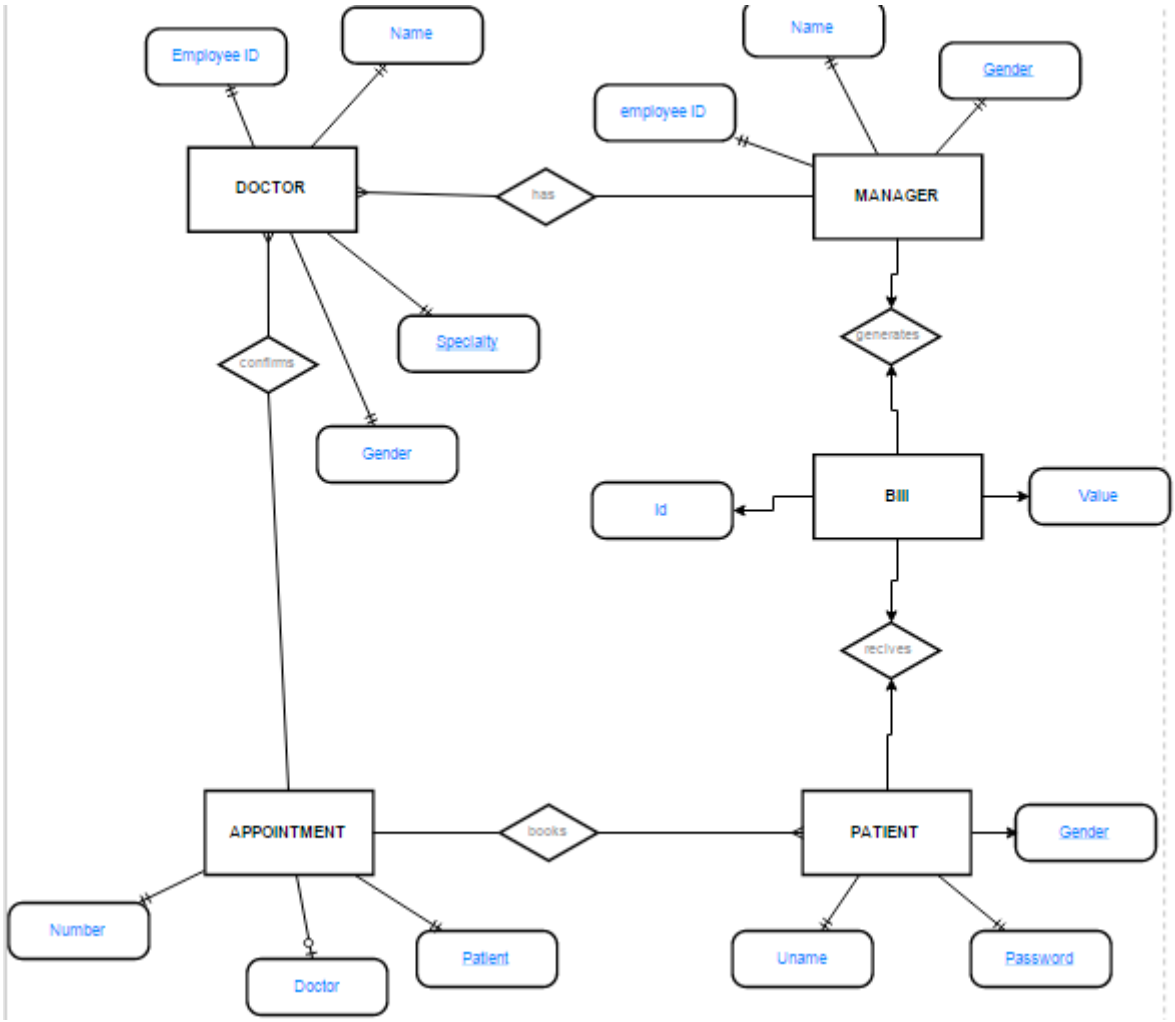
3.2.1 Level 0 Data Flow Diagram



3.2.2 Level 1 Data Flow Diagram

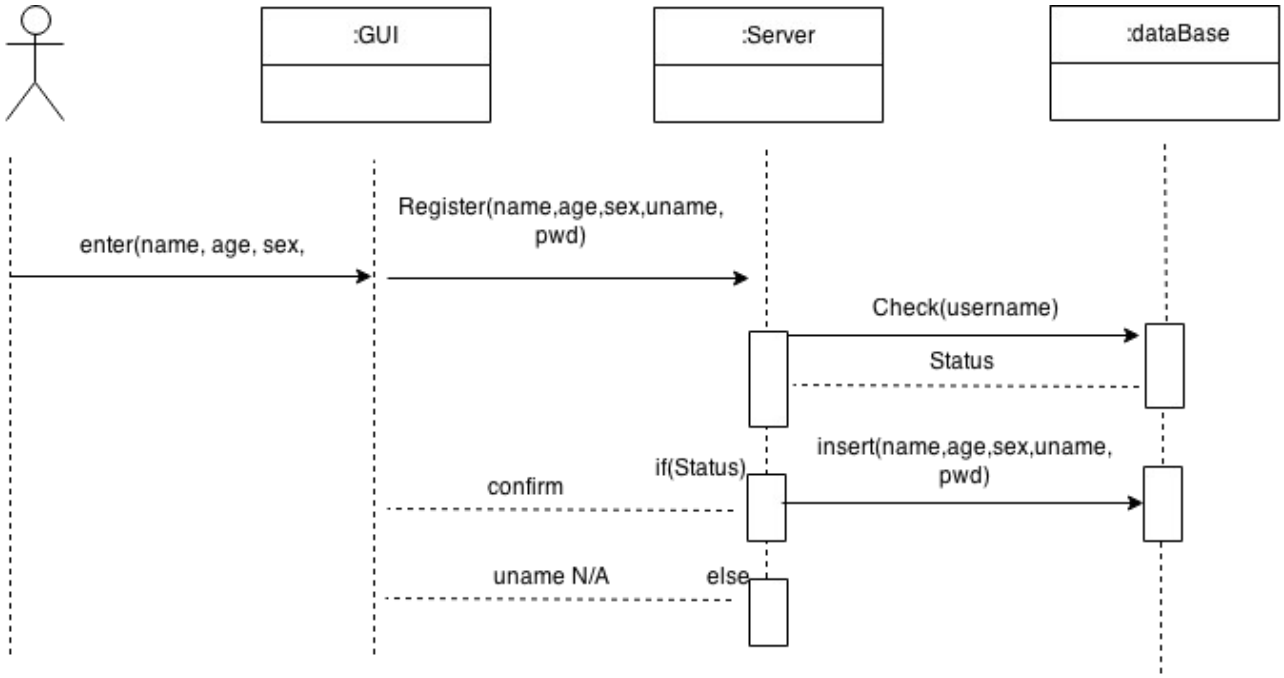


3.3 Entity Relationship Diagram



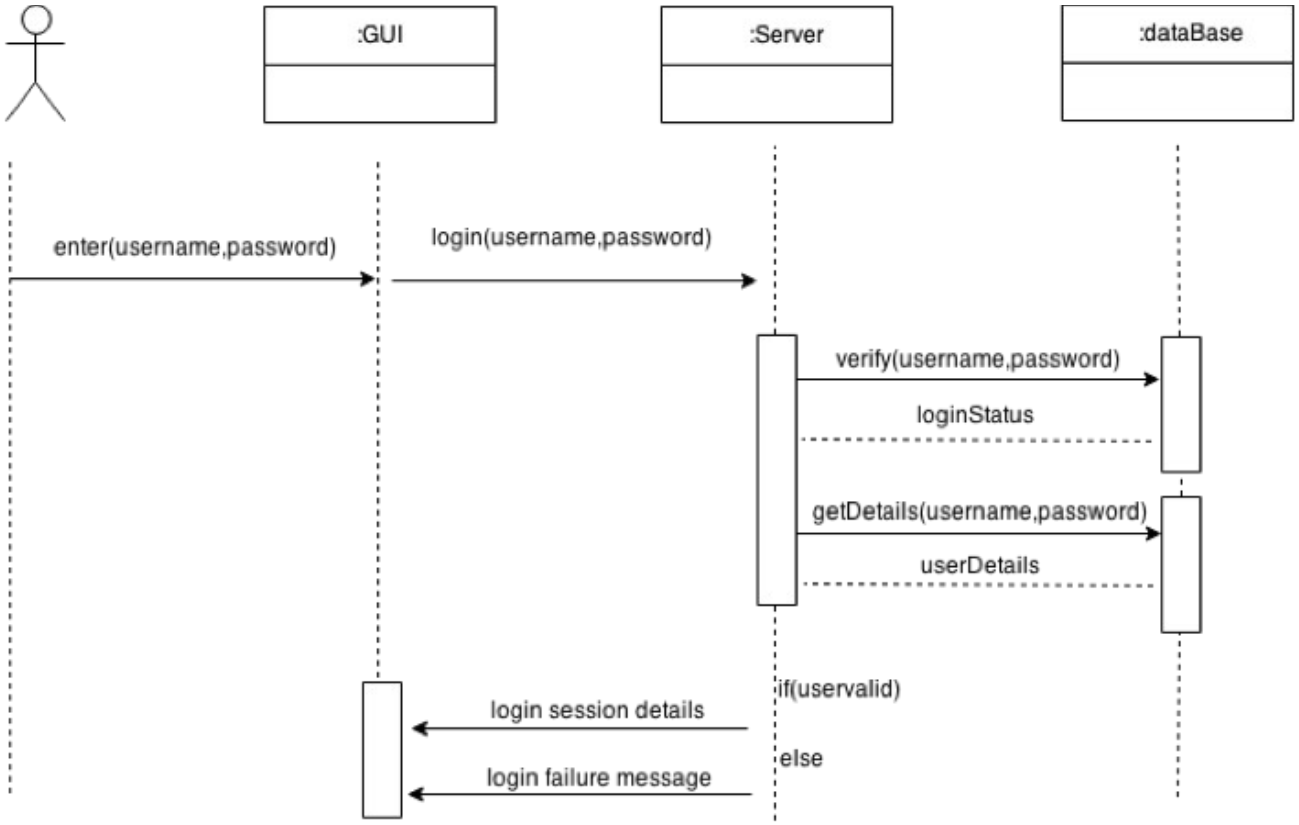
3.4 Sequence Diagram

3.4.1 Register Module [U]



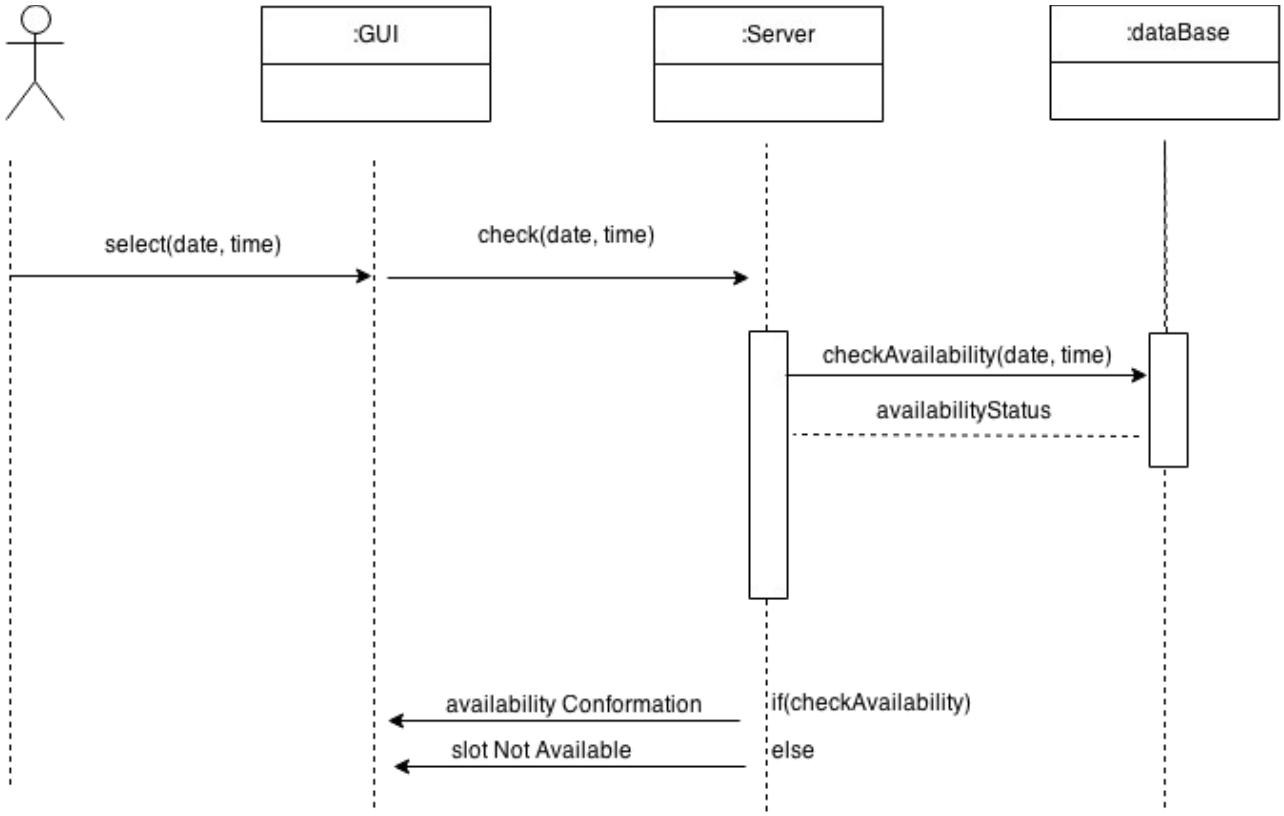
Use Case	Register
Precondition	none
DESC	For a user to register in order to book an appointment
Basic Path	1. patient enter the given details in the form 2. Server generates a new user id for the user. 3. Server updates the database with the details of the new user.
Alternative Path	If server is busy in step 2, display “try later” message
Post condition	The database is updated with the patient details
DEP	none

3.4.2 Login Module [U]



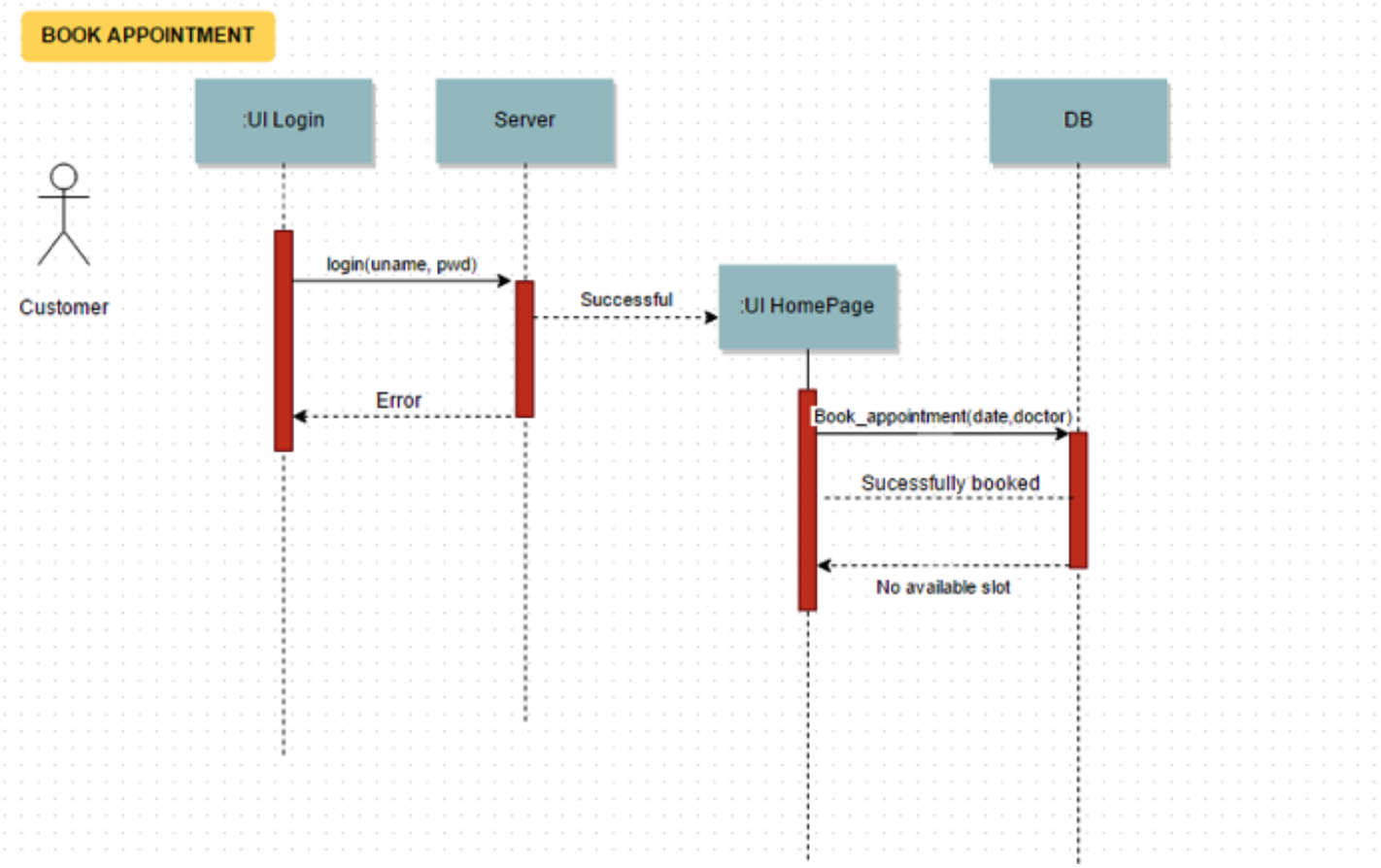
Use Case	Login
Precondition	User must have registered.
DESC	For a user to register in order to book an appointment
Basic Path	1. user enters his username and password 2. Server queries the username and password from database. 3. If they match. User is granted an active session.
Alternative Path	If username and password do not match in step 3, “User not found” message is displayed
Post condition	The user is granted an active session
DEP	3.2.11

3.4.3 Check Availability[U]



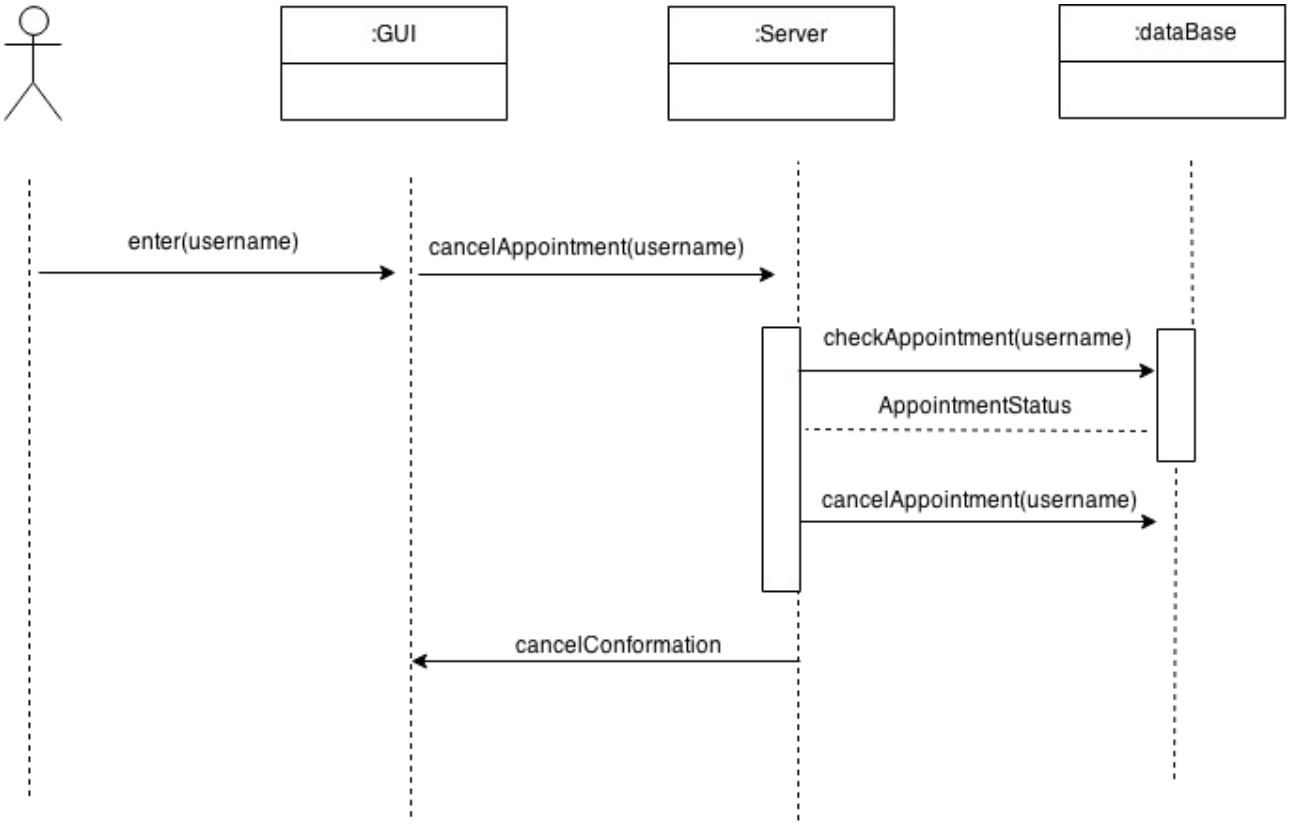
Use Case Name	Check availability
Precondition	The user has a login session
DESC	A user should be able to see if the appointment is available for a particular doctor.
Basic Path	1. User selects a date 2. Selected date is sent to server 3. Server Queries the database with the selected date 4. If Available, the server returns 'Available' to the user
Alternative Path	In step 4, If not Available, Server returns 'Not Available' to user.x`
Post condition	The availability status is confirmed
DEP	none

3.4.4 Book Appointment[U]



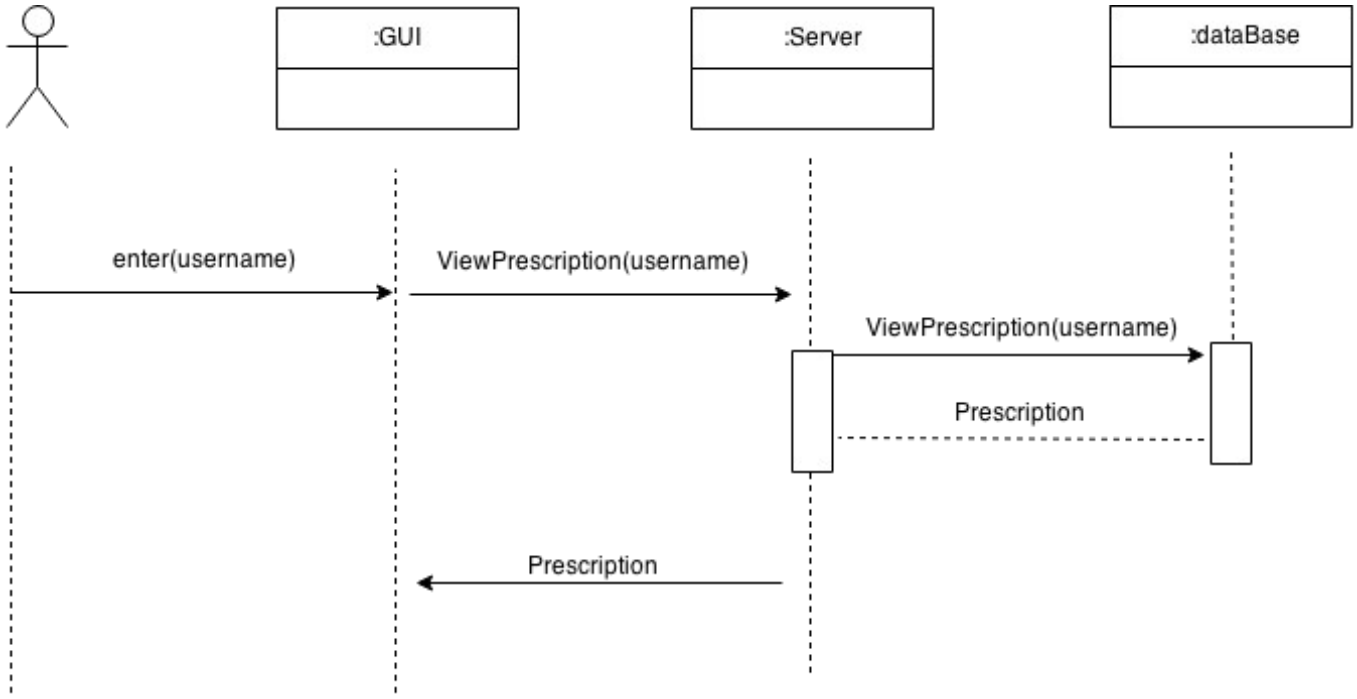
Use Case Name	Book Appointment
Precondition	The user has confirmed the availability for appointment
DESC	If there is availability for the doctor, the user should be able to book an appointment
Basic Path	1. User enters all details in a form. 2. The form is submitted to the server. 3. Server adds a new appointment to the queue in database.
Alternative Path	If server is busy, 'try again in some time' message is displayed to the user.
Post condition	Appointment Queue is updated in the database.
DEP	3.2.1

3.4.5 Cancel Appointment [U]



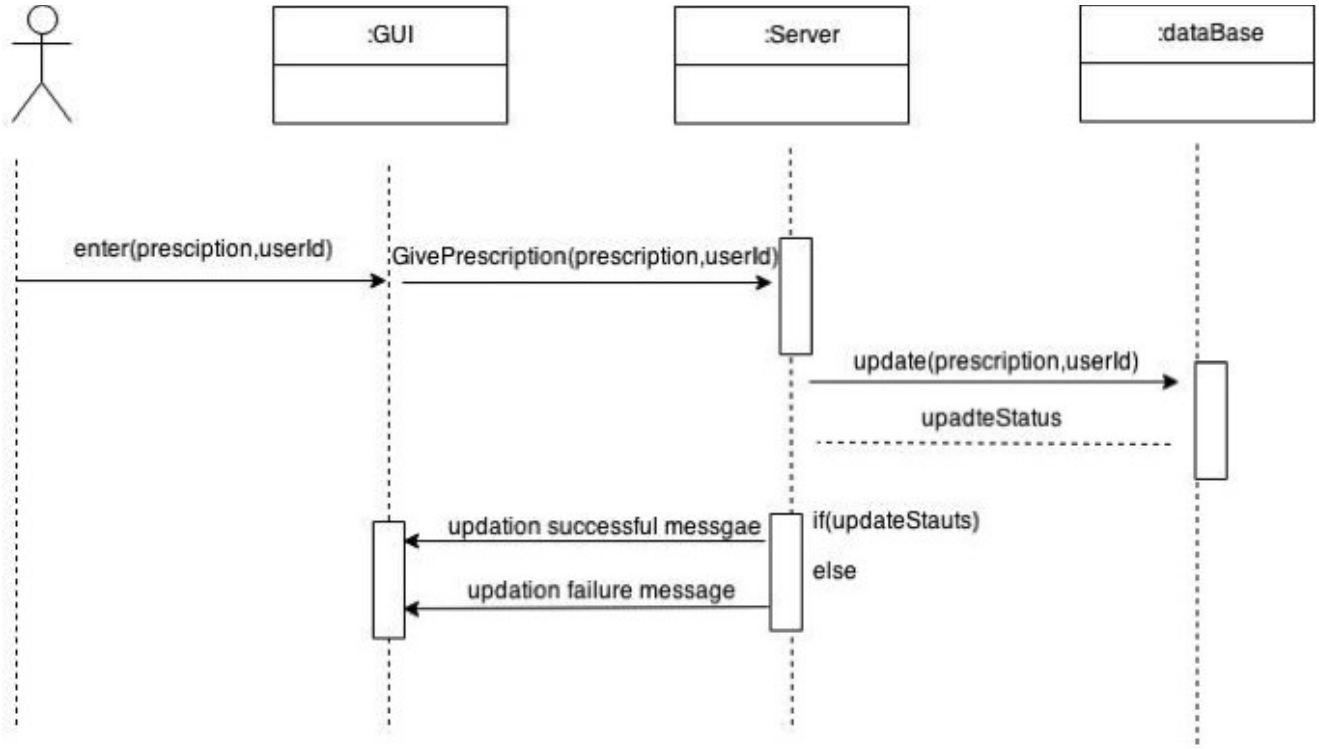
Use Case Name	Cancel appointment
Precondition	A patient must have booked an appointment.
DESC	The system manager should be given the option to cancel an appointment which has been booked by the patient.
Basic Path	1. Admin selects the appointment to be cancelled 2. Server updates database with the removed appointment list
Alternative Path	If server is busy in step 2, appropriate message is displayed
Post condition	The selected appointments are removed from queue
DEP	3.2.3

3.4.6 View Prescription [P]



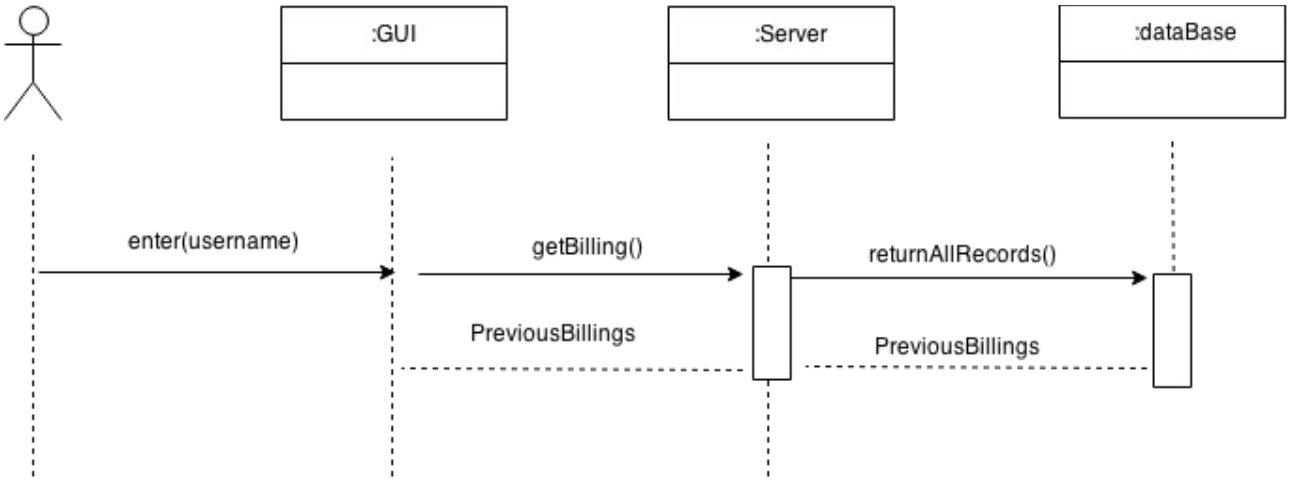
Use Case Name	View Prescription
Precondition	The user should have booked an appointment with the doctor
DESC	The user should be able to view what medicines the doctor has prescribed after the treatment.
Basic Path	1. Patient selects the prescription option. 2. Server queries the database for the prescription information along with the user id. 3. Server returns the appropriate prescription and it is displayed to the user accordingly.
Alternative Path	If server returns an empty string in step 3, then 'No Prescription Updated' message is displayed.
Post condition	The appropriate prescription is viewed.
DEP	3.2.3

3.4.7 Give Prescription [P]



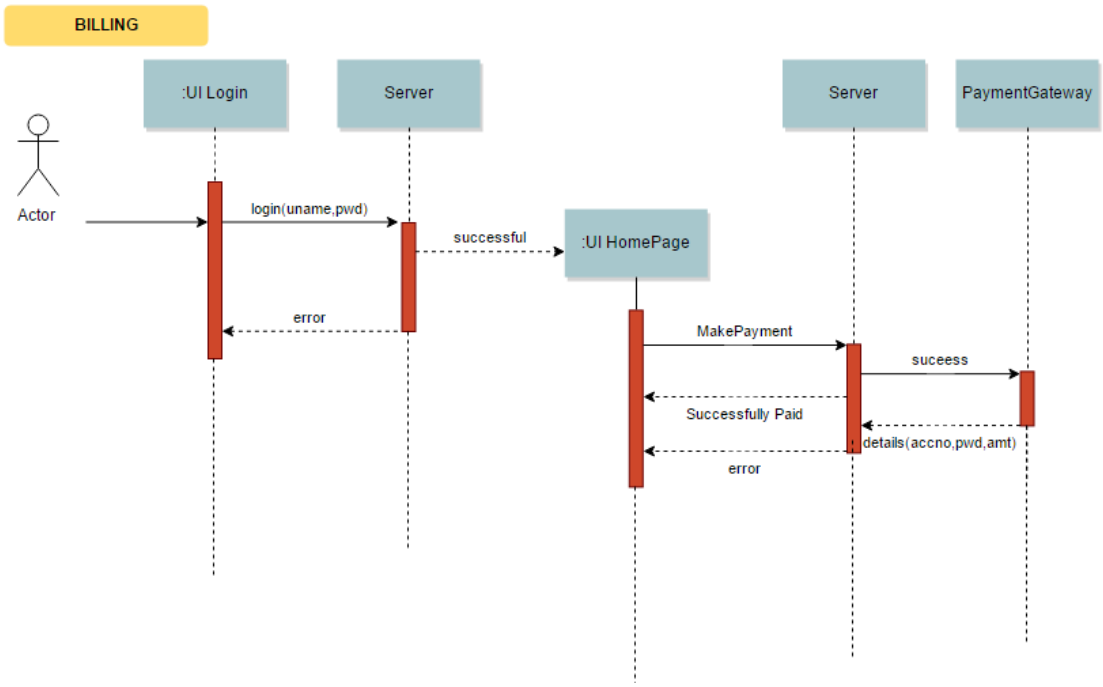
Use Case Name	Give Prescription
Precondition	Patient must have booked an appointment for a doctor to give prescription and doctor must have logged in.
DESC	The doctor should be able to give a prescription for a particular patient and the information should be private to only the user.
Basic Path	1. Doctor selects a patient from a patient list 2. Doctor posts a prescription for the user 3. Server updates the prescription for the selected user.
Alternative Path	If server is busy is step 2, 'try again in some time' message is displayed to the doctor
Post condition	The appropriate prescription is updated accordingly.
DEP	3.2.2

3.4.8 Generate Billing [P]



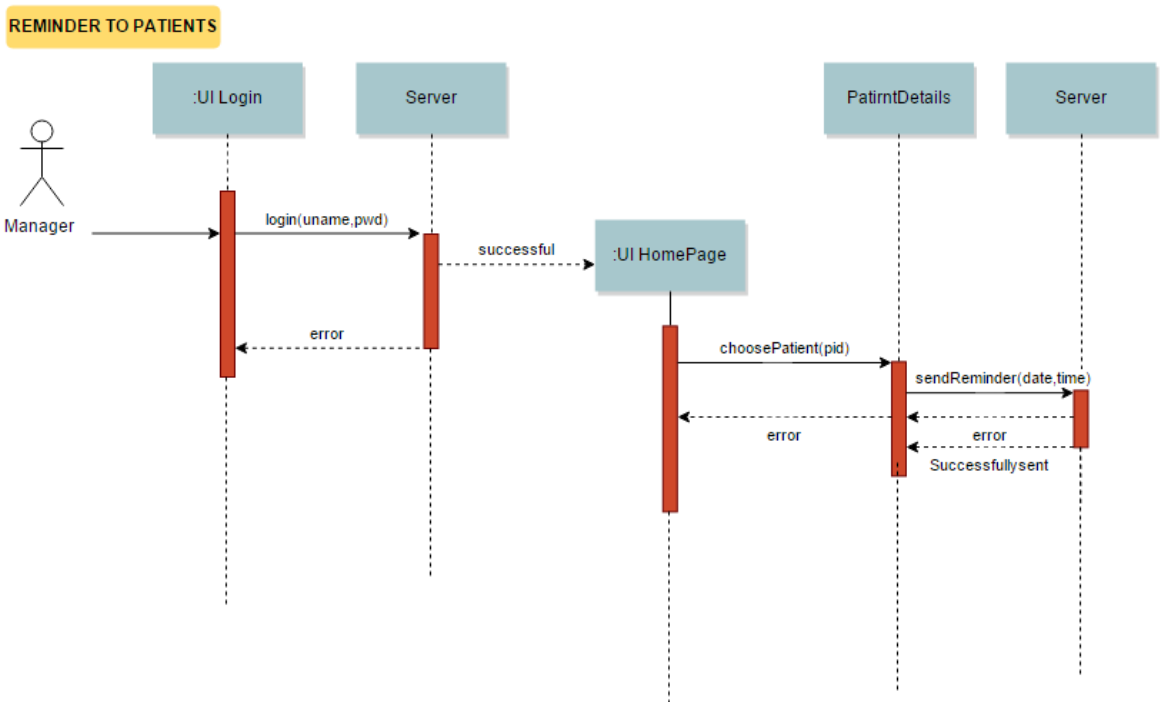
Use Case Name	Generate Billings
Precondition	The session must be of a system manager.
DESC	The system manager should be able to view all the old bills so as to keep a record of income.
Basic Path	1. Admin selects the show old bills option. 2. Server queries the database for all the bills. 3. The server responds by returning all the bills.
Alternative Path	If the server returns an empty string, 'No bills found' message is displayed
Post condition	All the bills are viewed by the system manager
DEP	3.2.3

3.4.9 Billing [P]



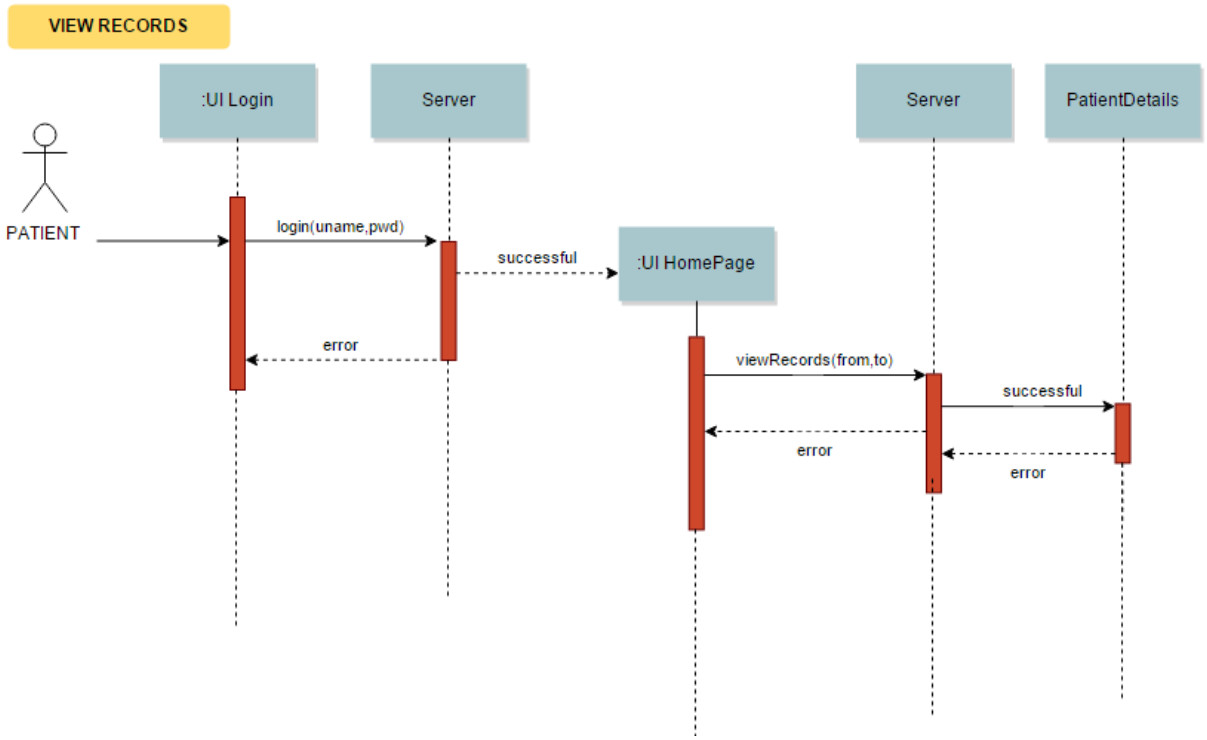
Use Case	Billing
Precondition	A patient must have booked an appointment.
DESC	The user should be able to make the payments online via debit/credit card after an appointment is booked.
Basic Path	1. User selects make payment option. 2. User fills in appropriate bank details. 3.The server process the input and generates the e-bill if transaction is successful
Alternative Path	If the transaction is not successful, display 'Low Balance' to the user.
Post condition	Bill is generated
DEP	3.2.2

3.4.10 Reminder to Patients [P]



Use Case	Reminder to patients
Precondition	The user must have booked an appointment
DESC	The system should send the appointment details to the patient via email. The details will consist of the date of appointment, it's timings, etc.
Basic Path	1. Server queries database for appointment details and email id. 2. Server sends an email to user regarding the date of appointment, timings etc.
Alternative Path	If email is not sent, the server tries again after some time.
Post condition	Reminder is sent to patients
DEP	3.2.2

3.4.11 View Records [P]



Use Case	View Records
Precondition	Doctor should have an active session.
DESC	The doctor can view a patient's medical history and edit it.
Basic Path	1. Patient enters his username. 2. Patient authenticates himself using biometrics/OTP. 3. Doctor accesses patient's medical history
Alternative Path	If authentication fails in step 2, the process terminates.
Post condition	The doctor can edit the medical history of the patient.
DEP	None

3.5 Database Tables

3.5.1 user

Attribute	uname	lname	fname	gender	email	phone	pass	id
DataType	varchar	varchar	varchar	varchar	varchar	varchar	varchar	int
length	20	20	20	20	20	20	20	5

3.5.2 appointment

Attribute	pateintUname	day	month	time	id
DataType	varchar	varchar	varchar	varchar	int
length	20	20	20	20	5

3.5.3 manager

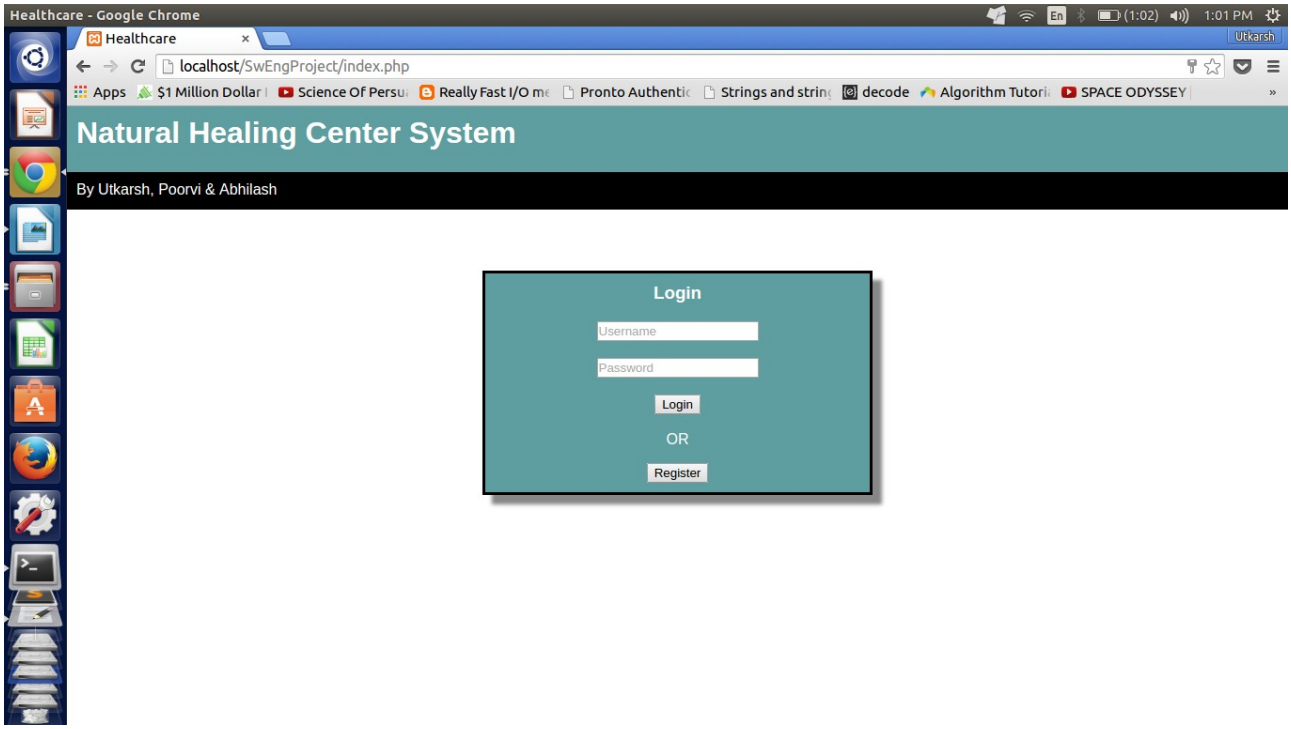
Attribute	uname	lname	fname	gender	email	phone	pass	id
DataType	varchar	varchar	varchar	varchar	varchar	varchar	varchar	int
length	20	20	20	20	20	20	20	5

3.5.3 doctor

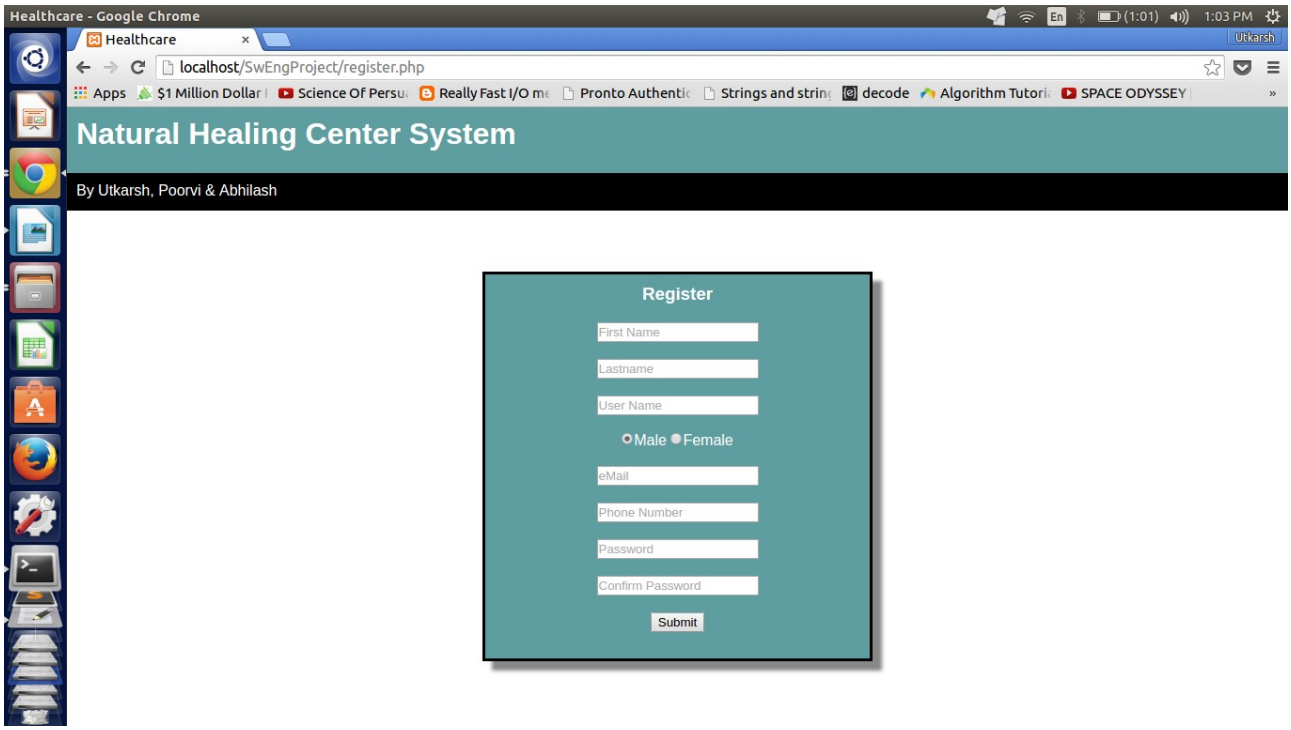
Attribute	uname	lname	fname	gender	email	phone	pass	id
DataType	varchar	varchar	varchar	varchar	varchar	varchar	varchar	int
length	20	20	20	20	20	20	20	5

3.6 UserInterface Design

3.5.1 Login Screen



3.5.2 Register Screen



3.5.3 Home Screen

3.5.3 Booking Status