

QUEUING ELIMINATION IN SMART CLINIC APPOINTMENT ASSISTANCE

Submitted in partial fulfillment of the required for the award of

Master of Computer Science

by

BHAVANI TEJASWI J

REGISTER NO (42734026)



DEPARTMENT OF COMPUTER SCIENCE

SCHOOL OF SCIENCE AND HUMANITIES

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade “A++” by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI 600 119

APRIL 2024



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited "A++" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **BHAVANI TEJASWI J (42734026)** who carried out the project entitled "QUEUEING ELIMINATION IN SMART CLINIC APPOINTMENT ASSISTANCE" under our supervision from January 2024 to April 2024.

INTERNAL GUIDE

Mrs. Jesintha J M.Sc., Assistant Professor

DEAN AND HEAD OF THE DEPARTMENT

Dr. Rekha Chakaravarthi M.E., Ph.D.,

Submitted for Viva voce examination held on

20/4/2024

Internal Examiner

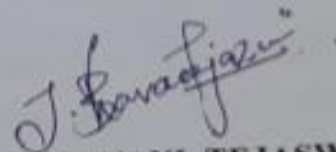
External Examiner

DECLARATION

I, BHAVANI TEJASWI J (42734026) hereby declare that the Project Report entitled "Queuing elimination in smart clinic appointment assistance" done by me under the guidance of Mrs. Jesintha J M.Sc., Assistant Professor, Department of Computer Science at SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY, Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai-600 119 is submitted in partial fulfilment of the requirements for the award of Master of Science degree in Computer Science.

DATE: 20/4/2024

PLACE: CHENNAI



BHAVANI TEJASWI J

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to Board of Management of **SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr.REKHA CHAKRAVARTHI M.E. Ph.D., Dean of Arts and Science** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my project guide **Mrs. Jesintha J M.Sc., Assistant Professor**, Department of Computer Science, for her valuable guidance, suggestions and constant encouragement which paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff member of the Department of Computer Science who were helpful in many ways for the completion of the project.

BHAVANI TEJASWI J

ABSTRACT

Many healthcare providers in primary health care use computerized information systems to a certain extent. However, the problem of heterogeneous information sources has impeded the integration of health care information within primary health facilities. How to facilitate information sharing among the different healthcare providers has been recognized as a growing area of significant importance due to its role in enhancing patient centered, continuity of care. The proposed project is online doctor appointment system that provides patients or any user an easy way of booking a doctor's appointment online. This is an application project in web based that overcomes the issue of managing and booking appointments on automated time allotment system. The task sometimes becomes very tedious for the compounder or doctor himself in manually allotting appointments for the users as per their availability. Hence this project offers an effective solution where users can view various booking slots available and select the preferred date and time. This system is very much useful to save the patients timing and also easy to analyze the doctor availability. The patient can easily interact with doctor for their problem and get solutions from the doctor.

Chapter No	TITLE	Page No
	ABSTRACT	i
1	INTRODUCTION	1
1.1	Introduction about the project	1
1.2	Existing system	1
1.3	Proposed system	2
2	LITERATURE SURVEY	4
3	SYSTEM DESIGN	8
3.1	Use case diagram for appointment scheduling	8
3.2	Sequence diagram for appointment scheduling	10
3.3	Activity diagram for appointment scheduling	12
3.4	Package diagram for appontment scheduling	14
3.5	Component diagram for appointment scheduling	16
3.6	Data flow diagram	17
4	SYSTEM ARCHITECTURE	20
5	SYSTEM IMPLEMENTATION	22
5.1	Priority queuing algorithm	22
5.2	Patient treatment time prediction algorithm	23
6	MODULE	25
6.1	Module description	25
7	CODING AND SCREENSHOT	27
7.1	Coding	27
7.2	Screenshot	34
8	CONCLUSION AND FUTURE WORK	36

8.1	Conclusion	36
8.2	Future work	36
	Reference	37

LIST OF FIGURES

FIGURE NO	NAME OF THE FIGURE	PAGE NO
3.1	Use case diagram for appointment scheduling	9
3.2	Sequence diagram for appointment scheduling	12
3.3	Activity diagram for appointment scheduling	13
3.4	Package detection for appointment scheduling	15
3.5	Component diagram for appointment scheduling	16
3.6	Level 0 DFD	18
3.7	Level 1 DFD	19
4.1	System Architecture	21
5.1	Priority queuing algorithm	24
7.1	Home page of appointment scheduling	34
7.2	Admin page	34
7.3	Patient register page	35
7.4	Doctor login page	35

LIST OF TABLE

TABLE NO	NAME OF THE TABLE	PAGE NO
2.1	Summary of literature survey	7

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION ABOUT THE PROJECT

If anybody is ill and wants to visit a doctor for checkup, he or she needs to visit the hospital and waits until the doctor is available. The patient also waits in a queue while getting appointment. If the doctor cancels the appointment for some emergency reasons then the patient is not able to know about the cancelation of the appointment unless or until he or she visits the hospital. As the mobile communication technology is developing rapidly, therefore, one can use the mobile's applications to overcome such problems and inconvenience for the patients. There is much work in the literature in this regard. An intelligent agent based appointment system has been proposed in which a scheduling system is provided for patients. The junior medical staff schedules appointment according to the priority level proposed an Android application that is used to remind the patients of their dosage timings through Alarm Ringing system so that they can stay fit and healthy. Searching doctors and hospitals along with navigation details are also available in the app so they can get proper treatment on time.

The norm dictates that everyone who is unwell and wants to see a doctor for a checkup must go to the hospital and wait until the doctor is available. In addition to waiting in line for an appointment, the patient. Patients can schedule appointments using a system that is based on intelligent agents and has been proposed.

1.2 EXISTING SYSTEM

In the existing system, mostly in hospitals waiting time is the high problem where the people need to wait a long queue at a maximum mean average of an hour. An solution need to be generated at a high level of artificial system embedded in it on reducing the queue system in hospitals.

- Manual Booking generation system can be done
- The implementation maintenance can be normal in booking
- Once an appointment is booked, patients are placed in a virtual queue.
- Online appointment scheduling systems and virtual queuing platforms may encounter technical glitches or downtime, leading to inconvenience for patients and staff.
- Clinic staff need to be trained to use and troubleshoot the various components of the system, which can be time-consuming and costly.

1.3 PROPOSED SYSTEM

The suggested system includes a forecast of patient waiting time. To look for local hospitals and schedule an appointment, an Android app will be created. The patient will be informed that they must see the doctor before one hour after the appointment is booked. Using the Patient Treatment Time Prediction (PTTP) algorithm, patients may also learn how long each patient's treatment will take. It is possible to analyze each event, and understanding the queuing system is simple. Use algorithms to dynamically adjust appointment schedules based on factors like doctor availability, patient arrival times, and estimated consultation durations. The patient receives the notification system as an email as a result of the recommendation system. The Hospital Queuing Prediction (HQR) system can be used to queue up patients. In this situation, when an emergency specialist system has been introduced, the key issue is finding a specialized doctor. Here, the system will be set up with several booking levels and a made-up prediction mechanism. To eliminate queuing in a smart clinic appointment assistance system, could implement features like online appointment scheduling, virtual queuing, notifications for appointment reminders or delays, and possibly integrating telemedicine options for certain consultations.

- The manual booking can be eliminated
- The system can be AI based booking with time prediction can be done
- Patients can book appointments online through a website. They can choose the date, time, and healthcare provider based on availability.
- At the clinic, self-service kiosks allow patients to check themselves in for their appointment without having to wait in line at the reception desk.

- The system automatically manages appointments, rescheduling or canceling them as necessary to optimize the clinic's schedule and reduce wait times.

By integrating these components, the queuing elimination system can streamline the appointment process, reduce wait times, and improve the overall patient experience in the clinic. Overall, the proposed system aims to streamline the appointment process, reduce waiting times, and improve the overall patient experience in the clinic.

CHAPTER 2

LITERATURE SURVEY

TITLE: Data Analytics and Predictive Modeling for Appointments No-show at a Tertiary Care Hospital

AUTHOR: Amani Moharram, Saud Altamini, Riya Alshammari;

JOURNAL NAME: IEEE

YEAR: 01 Jan - 31 Dec 2019.

This study [1] aims to develop an accurate machine learning model for predicting no-shows in pediatric outpatient clinics at King Faisal Specialist Hospital and Research Centre (KFSH&RC), and understand pediatric patients' characteristics who are most likely will not show to their scheduled appointments. Appointment no-show data collected from KFSH&RC data warehouse over the period (01 Jan - 31 Dec 2019). We analyzed a dataset that consists of 101,534 scheduled appointments for 35,290 pediatric patients. No-shows over the mentioned period was 11,573 for 8,105 patients. Three machine-learning algorithms, namely logistic regression, JRip, and Hoeffding tree, were compared to find the best one. The no-show rate in pediatric outpatient clinics was 11.39%. Accuracy, precision, recall, and F-score were selected to evaluate the built models performance. The precision and recall of the three models was around 90%. The F-score of the three models was similar and equal to 0.86. These models improved our capability to identify pediatric patients' characteristics at high risk of not attending their appointments.

TITLE: Design of Intelligent and Secure Hospital Appointment Scheduling System

AUTHOR: Adamu Muhammad Noma; Kabiru Ibrahim Musa; Hussainin Mamman; Abdulkadir Danlami Mato;

JOURNAL NAME: IEEE

Year: 27 June 2022.

As a roadmap [2] to digitizing the healthcare delivery system in Nigeria to support effective decisions in healthcare management through harnessing and timely delivery of reliable data, we propose a design of an intelligent and secured appointment scheduling system. The main aim is to support healthcare administration in hospitals, while satisfying the global best

practice in healthcare delivery. The system negotiates appointment schedules with medical teams in such a way that it is completely automatic, and optimally satisfies the patients' demands for timely accessing medical service while efficiently utilizing healthcare personnel. With the system, patients have the option of seeking appointments in the comfort of their homes or through record officers in the hospitals. It can handle both daily basis walk-in patients and those seeking appointments with specialists. The system is also designed to be secured at the application layer through a fine-grained group-based access control on authenticated users. When implemented, the system would exist as a subcomponent of the Health Management System and be interoperable with the other components by communicating through standard data exchange formats such as JSON.

TITLE: Artificial Intelligence-Assisted Reduction in Patients' Waiting Time for Outpatient

AUTHOR: Xiaoqing Li; Dan Tian; weihua Li; Bin Dong;

JOURNAL NAME: Research Gate

YEAR: June 2020

Many studies indicate that patient satisfaction is significantly negatively correlated with waiting time. A well-designed healthcare system should not keep patients waiting too long for appointment and consultation. However, in China, patients spend considerable time waiting, and the actual time spent on diagnosis and treatment in the consulting room is comparatively less. **Methods:** We developed an artificial intelligence (AI)-assisted module that is embedded in hospital information systems. Through its use, outpatients were automatically recommended an imaging examination or a laboratory test based on their symptoms and chief complaint. Thus, they could get examined or tested before they went to see the doctor. People who saw a doctor in the traditional way were assigned to the conventional group, and those who used the AI-assisted system were assigned to the AI-assisted group. We conducted a 1:1 case-control study that applied propensity score matching to pair the data from patients in a pediatric tertiary hospital between August 1, 2019 and January 31, 2020. Waiting time was defined as the time from registration to preparation for a laboratory test or an imaging examination. The total cost included the registration fee, test fee, examination fee, and drug fee. The Wilcoxon rank-sum test

was used to compare the differences in time and cost between the AI-assisted group and the conventional group. The statistical significance level was set at 0.05 for two sides.

TITLE: Simulation-Based Analysis of Appointment Scheduling System in Healthcare Services:
A Critical Review

AUTHOR: Vladimir Simic., Muhammet Deveci., Dragan Pamucar

JOURNAL NAME: IEEE

YEAR: 2020

This review paper investigates the healthcare' simulation appointment scheduling system (SASS). It primarily aims to highlight critical appointment scheduling challenges and outline the most prominent research topics. This critical review also intends to provide a comprehensive summary of these works, revealing progress achieved on the topic while highlighting potential concerns in healthcare that can be modeled via simulation. In total, 350 articles published in the period 2000–2021 in the Web of Science database were collected; 250 articles were chosen for review. Several approaches are discovered using the keyword SASS analysis, with four simulation approaches central to the research subject and deserving additional consideration. Finally, this study indicates that future research should focus on improving the efficiency of simulation optimization methods by reducing their time cost and increasing satisfaction.

S.no	PAPER TITLE	JOURNAL NAME	AUTHOR NAME	YEAR	METHODOLOGY
1	Data Analytics and Predictive Modeling for Appointments No-show at a Tertiary Care Hospital	IEEE	Amani Moharram, Saud Altamini, Riya Alshammari;	01 Jan - 31 Dec 2019	MACHINE LEARNING
2	Design of Intelligent and Secure Hospital Appointment Scheduling System	IEEE	Adamu Muhammad Noma; Kabiru Ibrahim Musa; Hussainin Mamman; Abdulkadir Danlami Mato;	27 June 2022.	CNN
3	Artificial Intelligence-Assisted Reduction in Patients' Waiting Time for Outpatient	Research Gate	Xiaoqing Li; Dan Tian; weihua Li; Bin Dong;	June 2020	ARTIFICIAL INTELLIGENCE

Table 2.1 summary of literature survey

CHAPTER 3

SYSTEM DESIGN

A UML (Unified Modeling Language) diagram is a graphical representation of a system or process using standardized symbols and notation. UML diagrams are widely used in software engineering and other fields to visually depict various aspects of a system, such as its structure, behavior, interactions, and architecture.

There are several types of UML diagrams, each serving a different purpose and focusing on different aspects of the system being modeled. Some common types of UML diagrams include:

Unified Modeling Language (UML) diagram for the proposed automated college timetable scheduling system. Here's an example of a high-level UML class diagram for the system:

3.1 USE CASE DIAGRAM FOR APPOINTMENT SCHEDULING

A use case diagram is a graphical representation of the interactions between actors (users or external systems) and a system under consideration. It illustrates the various ways users interact with the system to accomplish specific tasks or achieve certain goals.

Key components of a use case diagram include:

Actors

Patient: The individual seeking medical assistance. The sequence starts with the patient initiating a request for an appointment via the patient interface. This reminder can include details such as date, time, and location of the appointment.

Admin: Receptionists or administrators managing appointments and patient flow.

Doctor: The medical professional providing the consultation.

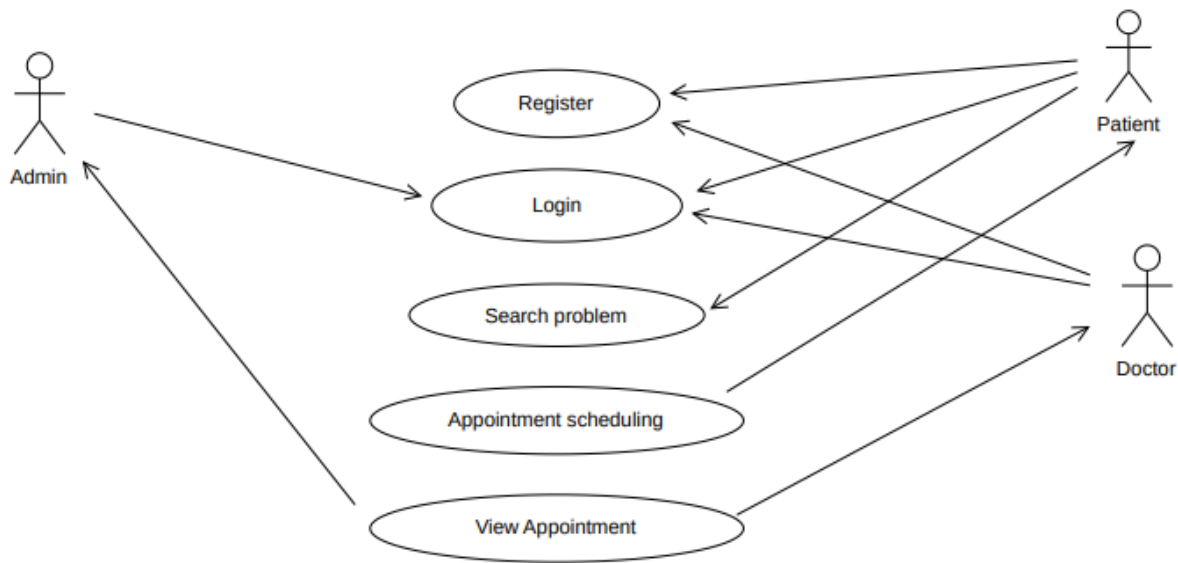


Fig 3.1 Use case diagram for appointment scheduling

Use Cases

Make Appointment: Patients can schedule appointments through the system, specifying their preferred date and time.

Join Virtual Queue: Patients can join a virtual queue remotely, indicating their intent to visit the clinic without physically waiting in line.

Monitor Queue: Clinic staff can monitor the virtual queue and patient flow in real-time.

Manage Appointments: Clinic staff can manage and adjust appointment schedules based on various factors such as doctor availability and patient load.

Notify Patient: The system can notify patients about their appointment status, estimated wait times, and any changes to their schedule.

Consultation: The doctor conducts the medical consultation with the patient at the appointed time.

Cancel Appointment: Patients or clinic staff can cancel appointments if necessary, triggering appropriate notifications and adjustments to the schedule.

Use cases represent specific functionalities or tasks that the system performs to fulfill user goals. Each use case describes a particular interaction between the user and the system. Use cases are depicted as ovals within the system boundary, often labeled with brief action-oriented descriptions.

Relationships

Patient-System Interaction: Patients interact with the system to make appointments, join the virtual queue, and receive notifications.

Clinic Staff-System Interaction: Clinic staff interact with the system to manage appointments, monitor the queue, and communicate with patients.

Doctor-System Interaction: Doctors may interact with the system to view their schedule and access patient information before consultations.

Relationships illustrate the connections between actors and use cases. The primary relationship in a use case diagram is the association between an actor and a use case, indicating which actors are involved in which use cases. Relationships can also show dependencies, generalizations, or extensions between use cases.

3.2 SEQUENCE DIAGRAM FOR APPOINTMENT SCHEDULING

A Sequence Diagram is a type of UML (Unified Modeling Language) diagram that illustrates the interactions between objects or components within a system in a sequential manner. It focuses on the flow of messages or method calls exchanged between these objects over time to accomplish a specific behavior or scenario.

Key components of a sequence diagram include:

Patient Makes Appointment: The sequence starts with the patient initiating the appointment process by interacting with the system, either through a mobile app or a website. The system receives the appointment request and checks for available time slots. Once a suitable time slot is found, the system confirms the appointment and sends a notification to the patient.

Admin Monitors Queue and Manage Appointments: Administrative staff monitor the virtual queue status and manage appointments. They may adjust schedules or allocate resources based on demand. If necessary, they inform the doctor about any changes to their schedule.

Doctor Reviews Schedule: The doctor accesses the system to review their schedule for the day. They see upcoming appointments and any changes made by the administrative staff.

Admin Handles Cancellations: If an appointment needs to be canceled or rescheduled, the administrative staff manage the process through the system. Patients affected by the changes are notified, and their appointments are updated accordingly. After the appointment is completed, the clinic interface updates the patient's status and records any necessary information.

The queue management system reflects the completed appointment, potentially freeing up slots for new appointments.

The system sends notifications to patients regarding their appointment details, including any changes. At the appointed time, the patient arrives for their consultation. The doctor accesses the patient's medical records and conducts the consultation. During the consultation, the doctor may update the patient's records in the system. If the doctor is running late, the system notifies affected patients accordingly. Incorporating the doctor and admin roles into the sequence diagram provides a comprehensive view of how the entire system operates to manage appointments, queues, and consultations in a smart clinic environment.

This sequence diagram illustrates how the various components of the system interact to eliminate queuing issues in a smart clinic appointment assistance environment, ensuring efficient appointment scheduling and patient flow.

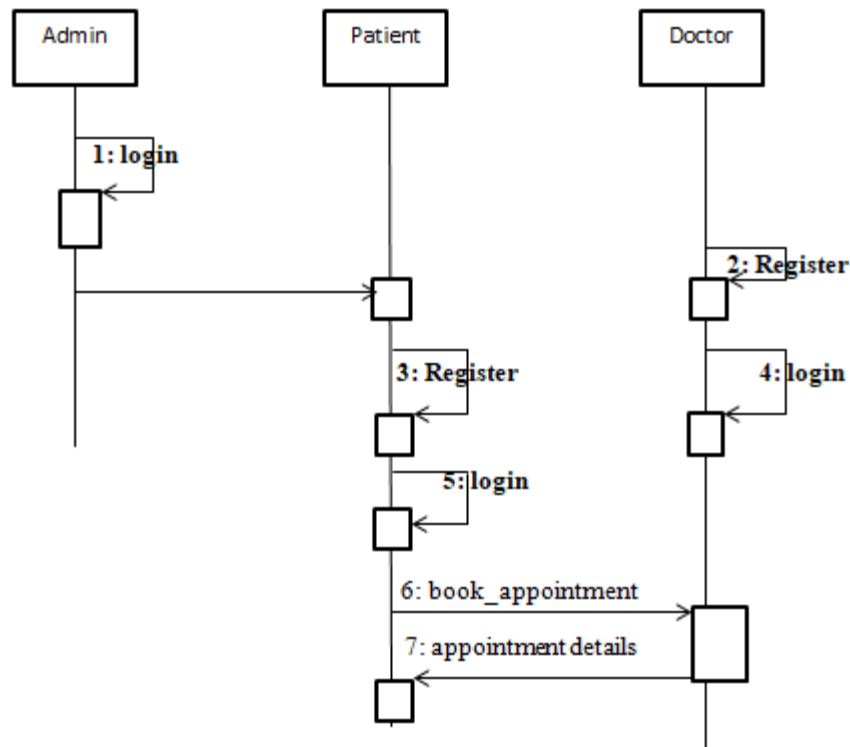


Fig 3.2 Sequence Diagram for appointment scheduling

3.3 ACTIVITY DIAGRAM FOR APPOINTMENT SCHEDULING

Start Appointment Scheduling Process: The process begins when a patient initiates an appointment scheduling request.

Patient Requests Appointment: The patient submits their appointment request through the patient app, providing details such as preferred date and time, reason for the appointment, and any specific doctor preferences.

System Validates Patient's Request: The system validates the patient's request, ensuring that all necessary information is provided and that it meets any predefined criteria or constraints.

System Checks Doctor's Availability: The system checks the availability of the requested doctor based on their schedule and any predefined constraints or preferences. The system presents the available appointment timeslots to the patient, considering both the patient's preferences and the doctor's availability.

Patient Selects Preferred Timeslot: The patient selects their preferred appointment timeslot from the options provided by the system.

System Confirms Appointment Reservation: Once the patient selects a timeslot, the system confirms the appointment reservation, updating both the patient's and the doctor's schedules accordingly.

End Appointment Scheduling Process: The process concludes after the appointment reservation is confirmed, and the appointment scheduling process is completed.

This activity diagram illustrates the sequential steps involved in the appointment scheduling process, from the initial request by the patient to the confirmation of the appointment reservation by the system. Each step ensures that the appointment is scheduled efficiently and in accordance with the patient's preferences and the doctor's availability.

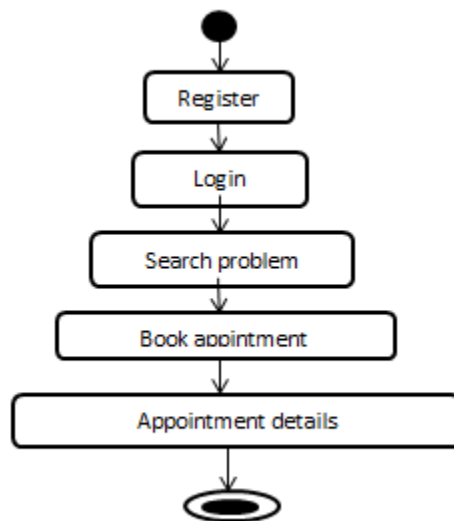


Fig 3.3 Activity Diagram for appointment scheduling

3.4 PACKAGE DIAGRAM FOR APPOINTMENT SCHEDULING

A Package Diagram is a type of UML (Unified Modeling Language) diagram that represents the organization and dependencies among packages in a system. In software engineering, a package is a collection of related elements, such as classes, interfaces, and other packages, grouped together to provide a cohesive unit of functionality. Package diagrams provide a high-level overview of the structure and organization of a system, helping to manage complexity and promote modularity.

Key components of a package diagram include:

Appointment Management: Modules in this handle appointment-related functionalities, including making appointments, canceling or rescheduling appointments, and managing appointment schedules. Administrative staff can access these modules to manage appointments on behalf of patients. This includes modules for managing the virtual queue, monitoring queue status, and prioritizing patients. Administrative staff utilize these modules to monitor and adjust the queue as needed, while doctors can view the queue status to prepare for upcoming consultations. If the slot is available, the scheduler proceeds to the next step. Otherwise, it suggests alternative times.

Doctor Management: This contains modules related to managing doctor schedules, accessing patient information, and conducting consultations. It includes functionalities for doctors to view their schedules, access patient records, and update medical information during consultations.

User Interface: This contains modules responsible for the user interface components, such as the patient-facing mobile app or website and the administrative staff's interface for managing appointments and queues. This includes modules for managing and storing data related to appointments, patient information, doctor schedules, and queue status. It may involve databases or other data storage systems to ensure data integrity and security.

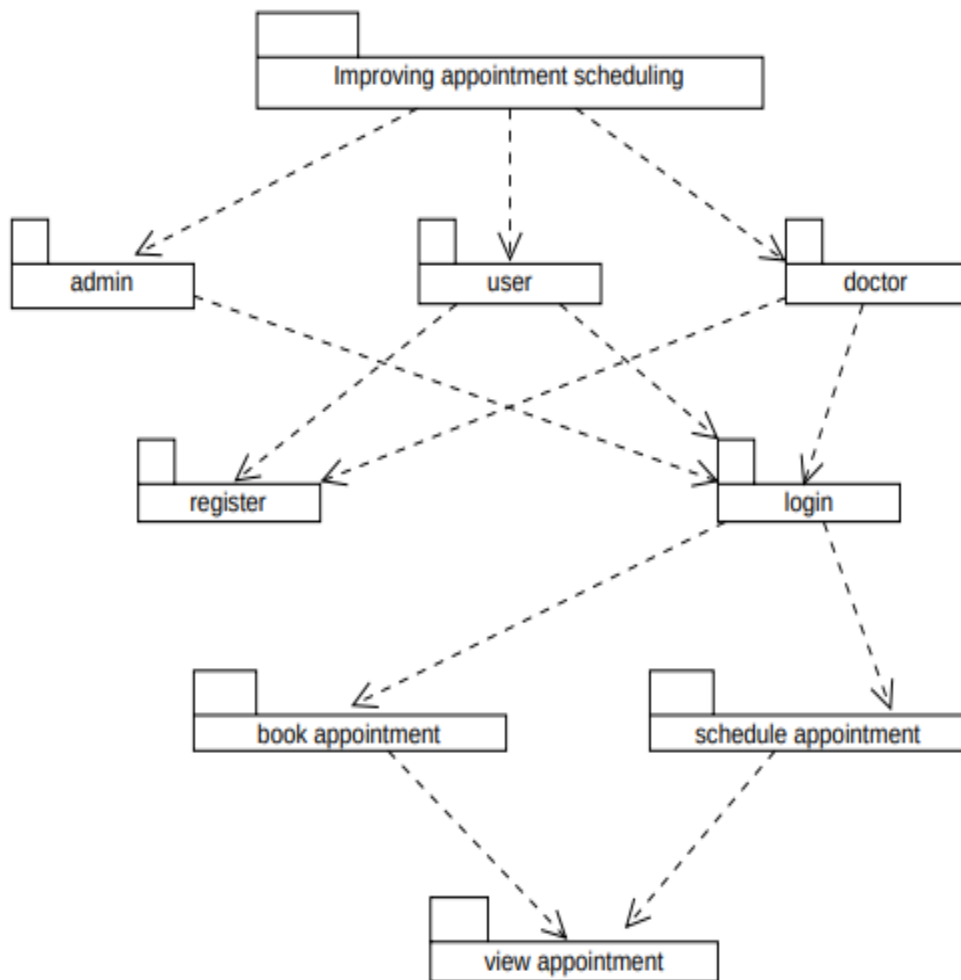


Fig 3.4 Package Detection for appointment scheduling

Package diagram provides a high-level overview of the package structure of the automated college timetable scheduling system, organizing classes and components into logical groupings based on their functionalities. By incorporating modules specific to the admin and doctor roles, the package diagram provides a comprehensive overview of the system's structure and functionalities in managing appointments and queues in a smart clinic environment.

3.5 COMPONENT DIAGRAM FOR APPOINTMENT SCHEDULING

Below is a simplified component diagram representing the components and their interactions in the automated appointment scheduling system:

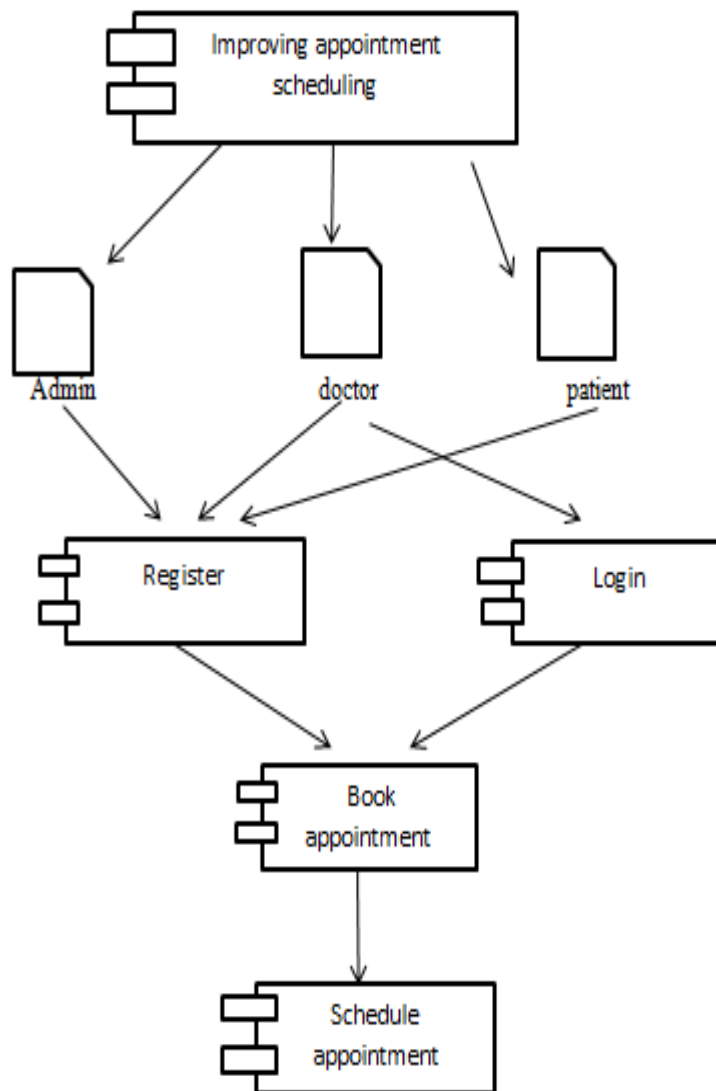


Fig 3.5 Component diagram for appointment scheduling

User Components: This component represents the user interface elements, such as the patient-facing mobile app or website and the administrative staff's interface. It includes components for appointment scheduling, queue management, and accessing patient information.

Admin Management Component: This component manages appointment-related functionalities, including making appointments, canceling or rescheduling appointments, and managing appointment schedules. It interacts with the user interface components to handle appointment requests and updates.

Doctor Management Component: This component manages doctor schedules, accesses patient information, and facilitates consultations. It includes functionalities for doctors to view their schedules, access patient records, and update medical information during consultations. It interacts with the user interface components to display relevant information to doctors and enable them to conduct consultations effectively.

3.6 DATA FLOW DIAGRAM

A two-dimensional diagram explains how data is processed and transferred in a system. The graphical depiction identifies each source of data and how it interacts with other data sources to reach a common output. Individuals seeking to draft a data flow diagram must identify external inputs and outputs, determine how the inputs and outputs relate to each other, and explain with graphics how these connections relate and what they result in. This type of diagram helps business development and design teams visualize how data is processed and identify or improve certain aspects.

LEVEL 0

DFD Level 0 is also called a Context Diagram. It's a basic overview of the whole system or process being analyzed or modeled. It's designed to be an at-a-glance view, showing the system as a single high-level process, with its relationship to external entities. It should be easily understood by a wide audience, including stakeholders, business analysts, data analysts and developers.

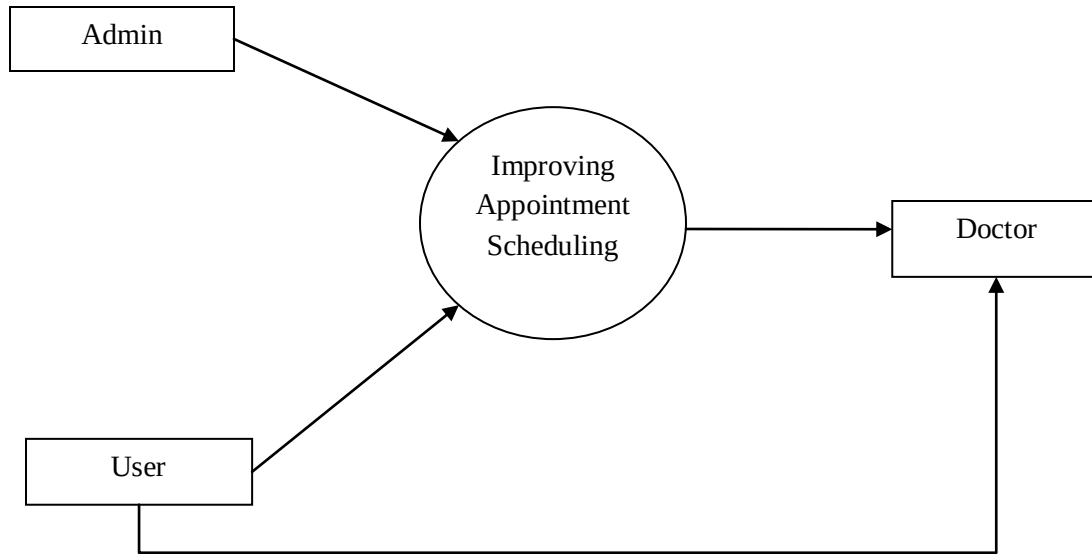


Fig 3.6 level 0 DFD of appointment scheduling

LEVEL 1

DFD Level 1 provides a more detailed breakout of pieces of the Context Level Diagram. You will highlight the main functions carried out by the system, as you break down the high-level process of the Context Diagram into its sub – processes. A level 1 data flow diagram (DFD) is more detailed than a level 0 DFD but not as detailed as a level 2 DFD. It breaks down the main processes into sub processes that can then be analyzed and improved on a more intimate level. Storage of patient information, appointment schedules, and queue data. Queue optimization suggestions, notifications to patients. Scheduled appointments, updated availability status.

Creating a data flow diagram for queuing elimination in a smart clinic appointment assistance system involves identifying the various components and how data flows between them. This diagram showcases how data flows between different components of the system, facilitating queuing elimination in a smart clinic appointment assistance environment.

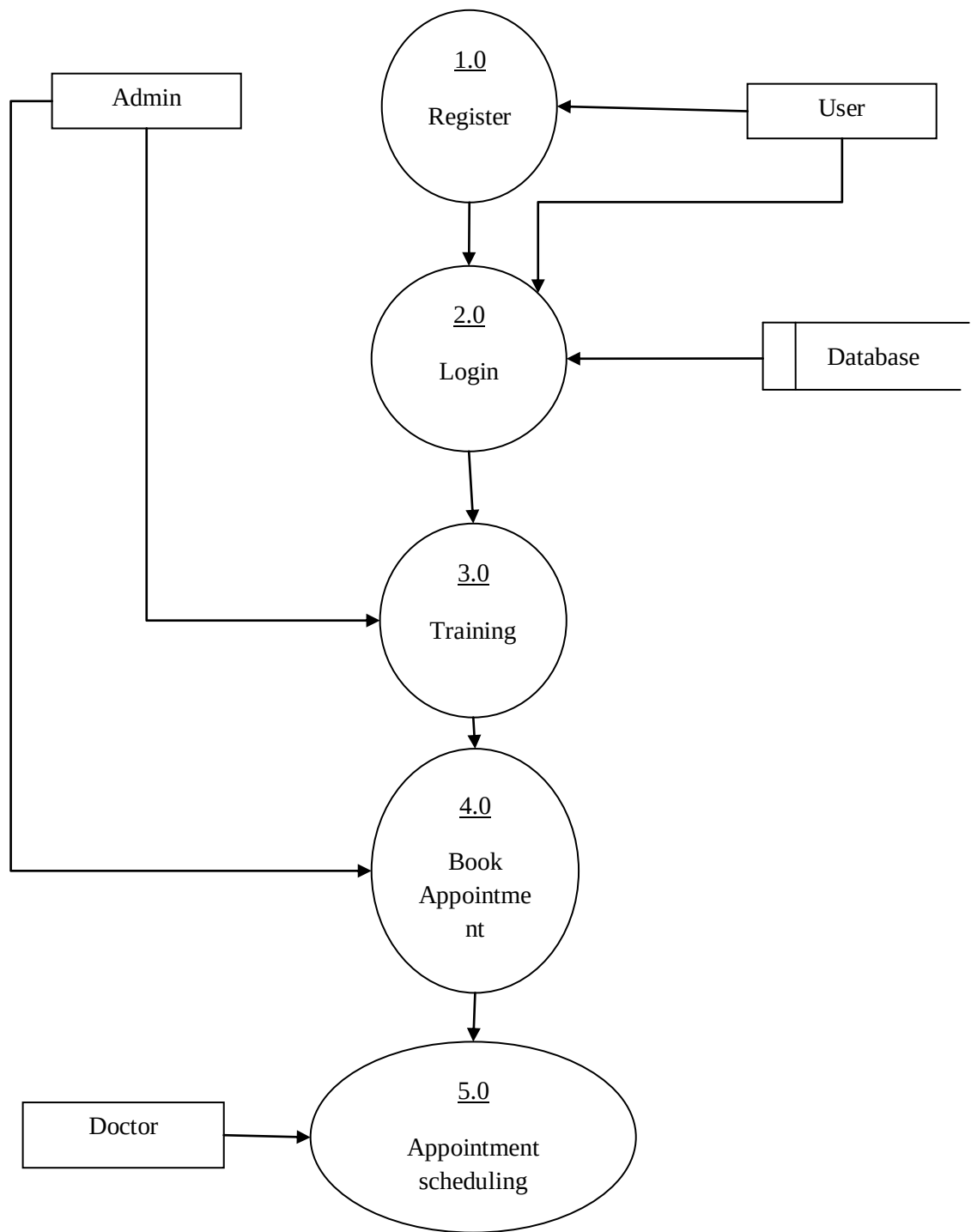


Fig 3.7 level 1 DFD of appointment scheduling

CHAPTER 4

SYSTEM ARCHITECTURE

The system architecture of an appointment scheduling system typically involves several components working together to facilitate the scheduling process efficiently. Here's a basic outline of such an architecture:

Admin Panel: This interface allows clinic administrators to manage the system, including adding and updating doctor profiles, adjusting scheduling parameters, generating reports, and overseeing overall system performance.

Doctor Panel: Doctors access this interface to view their schedules, manage appointments, update patient records, and communicate with patients or other staff members. They can also receive notifications and alerts about schedule changes.

Patient App: Patients use this mobile or web application to register, schedule appointments, receive reminders, view their medical history, and communicate with doctors or clinic staff. The app provides a user-friendly interface for managing appointments and accessing healthcare services.

Appointment System: This core component of the system assists patients in scheduling appointments, optimizes appointment slots based on machine learning predictions, and sends notifications and reminders to patients and doctors. It interfaces with the admin panel, doctor panel, and patient app to coordinate scheduling and communication.

Admin Login: The admin logs in using a username and password through the admin panel interface.

Doctor Login: Doctors log in using their credentials (e.g., username and password) through the doctor panel interface.

Patient Login: Patients register or log in using their credentials (e.g., email or phone number and password) through the patient app interface.

Each login process provides access to the respective functionalities and interfaces tailored to the user's role (admin, doctor, or patient) within the smart clinic appointment assistance system.

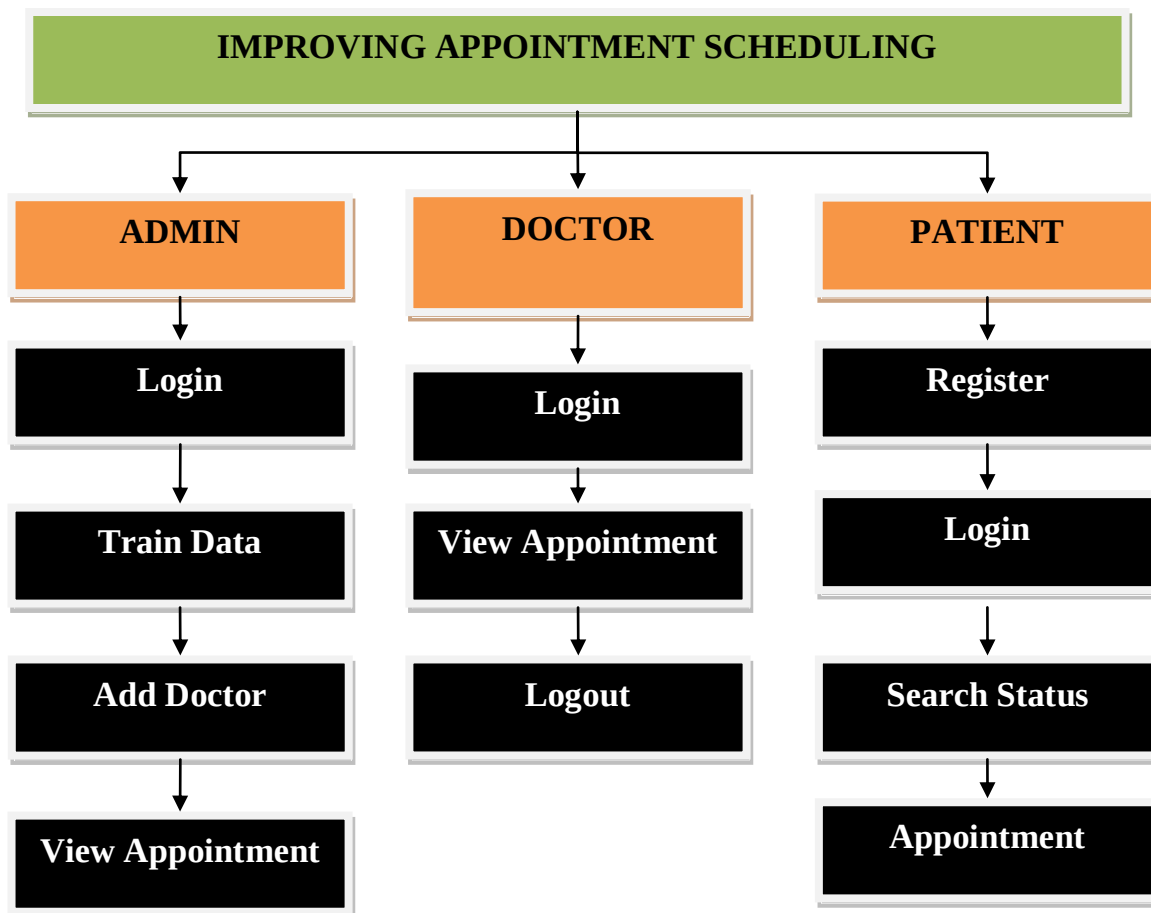


Fig 4.1 System Architecture for appointment scheduling

CHAPTER 5

SYSTEM IMPLEMENTATION

5.1 PRIORITY QUEUING ALGORITHM

Priority queuing algorithm is a method used in smart clinic appointment assistance to prioritize appointments based on certain criteria, such as the urgency of the patient's condition or the importance of the appointment. Priority Queues are abstract data structures where each data/value in the queue has a certain priority. For example, In airlines, baggage with the title “Business” or “First-class” arrives earlier than the rest.

Assigning Priority Levels: Each appointment is assigned a priority level based on predetermined criteria. For example, urgent cases might be assigned a higher priority level than routine check-ups.

Queue Formation: Appointments are then placed into separate queues based on their priority levels. Higher priority appointments are placed in queues with higher priority, ensuring they are attended to first.

Scheduling Appointments: When a time slot becomes available, appointments are scheduled from the appropriate queue based on their priority level. Higher priority appointments are scheduled first, followed by lower priority appointments.

Dynamic Adjustment: The priority queues may be dynamically adjusted based on changes in the clinic's workload, patient needs, or other factors. For example, if a patient's condition worsens while waiting, their appointment priority may be increased.

Optimization: The goal of priority queuing is to optimize the scheduling process by ensuring that urgent cases are seen promptly while still accommodating less urgent appointments efficiently.

By using priority queuing, smart clinic appointment assistance can help reduce waiting times for patients with urgent medical needs while maintaining overall efficiency in appointment scheduling.

5.2 PATIENT TREATMENT TIME PREDICTION ALGORITHM

A patient treatment time prediction algorithm typically involves analyzing various factors such as the patient's medical history, current condition, type of treatment, available resources, and historical data on similar cases. Machine learning techniques are often used to train models that can predict treatment times based on these factors. These models learn patterns from past data to make accurate predictions for future cases. The algorithm might incorporate regression analysis, decision trees, or neural networks, depending on the complexity of the prediction task and the available data.

Data Collection:

Patient demographics

Medical history

Current condition

Treatment details

Resource availability

Preprocessing:

Data cleaning

Feature engineering

Normalization/standardization

Machine Learning Model:

Regression model (e.g., linear regression, random forest regression)

Neural network model (e.g., feedforward neural network, recurrent neural network)

Decision tree model

Training:

Split data into training and validation sets

Train the model using historical data

Tune hyperparameters

Prediction:

Input new patient data

Use the trained model to predict treatment time.

Evaluation:

Measure prediction accuracy using metrics like Mean Absolute Error (MAE) or Root Mean Squared Error (RMSE)

Deployment:

Integrate the model into the healthcare system

Monitor performance and update model as needed

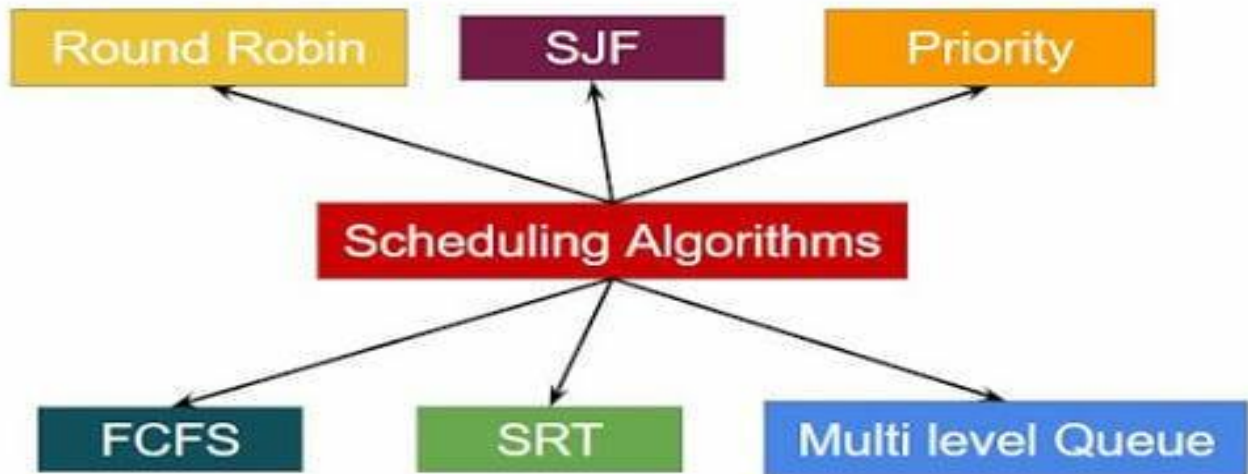


Fig 5.1 priority queuing algorithm

CHAPTER 6

MODULE

6.1 MODULE DESCRIPTION

Admin

Login: Admin is the one who will manage the user and doctor details. Admin is the one who verify the doctor details and predict the disease retrieving from the database. Admin will add the training datasets for the user verification.

Admin Training: The admin will make a training session where the disease with the symptoms and the scheduling timing will be added with the generated system. The admin will make a module access system with the identified pattern. The treatment for the disease will be suggested by the admin.

Patient Appointment: The appointments and the tablet booking made by the user will be known here by the admin. The admin will verify the doctor appointments booking and the medicine booking so that they can give the user with the medicine.

Add Doctor: Doctor will be added with their details in-order to treat the patient using this web application. Doctor should enroll their specialist details while enrolling. Such, that this helps the admin to suggest them to the patients.

Patient

Patient Register and Login: User should register and create an account by entering their details. They will be provided with a separate login session with a username and password. They have to login for predicting their disease. The sequence starts with the patient initiating a request for an appointment via the patient interface. The patient's request is received by the system.

Make Appointment: In make appointment module is used to predict the disease and make appointment by using the symptoms given by the user. The admin will train the datasets which will be very useful to predict the user disease. Thus the disease prediction module also helps the patient, to diagnose the disease and treat them. Admin also suggest them the tablets for the disease

Schedule Appointment: The admin will suggest the doctor for the treatment to the patients. Thus the patient can also interact with the doctor. So, that the user needs not to search the doctors anytime whether they are available or not they can predict their disease and interact with that specified doctor. The admin will suggest the doctor for the treatment to the patients. The appointment will be automatically schedule with the generated system.

Doctor

Doctor Login: The doctor will make a login session where the doctor will be done with the generated pattern where the doctor will be given with a username and password.

View Appointments: The appointments made by the user will be known here by the doctor. The doctor will verify the appointments booking and accept the user side efficiency process with timing.

CHAPTER 7

CODING AND SCREENSHOT

7.1 CODING

```
import os
import random
from datetime import date, timedelta
import re
import pandas as pd
from datetime import datetime
from urllib import request
import pymysql
import ar_master
import smtplib, ssl
from flask import Flask, render_template, flash, request, session, current_app,
send_from_directory
from werkzeug.utils import redirect, secure_filename
from time import strftime

port = 587
smtp_server = "smtp.gmail.com"
sender_email = "serverkey2018@gmail.com"
password ="extazee2021"

# ps = PorterStemmer()
app = Flask(__name__, static_folder="static")
app.config.from_object(__name__)
app.config['SECRET_KEY'] = '7d441f27d441f27567d441f2b6176a'
user='root'
password="
```

```

host='localhost'
database='python_improving_appointment_scheduling'
mm = ar_master.master_flask_code()
HOME
@app.route("/")
def homepage():
    return render_template('index.html')
@app.route("/admin")
def admin():
    return render_template('admin.html')
@app.route("/admin_login", methods = ['GET', 'POST'])
def admin_login():
    error = None
    if request.method == 'POST':
        if request.form['uname'] == 'admin' and request.form['pass'] == 'admin':
            return render_template('admin_home.html',error=error)
        else:
            return render_template('admin.html', error=error)

@app.route("/admin_home")
def admin_home():
    return render_template('admin_home.html')

@app.route("/admin_doctor",methods = ['GET', 'POST'])
def admin_doctor():
    data=mm.select_direct_query("select
doctor_id,doctor_name,gender,email,address,experience,specialist from doctor_details")
    return render_template('admin_doctor.html',items=data)

```

```

@app.route("/admin_patient",methods = ['GET', 'POST'])
def admin_patient():
    data=mm.select_direct_query("select id,name,contact,email,address      from user_details")
    return render_template('admin_patient.html',items=data)

```

```

@app.route("/admin_appointment",methods = ['GET', 'POST'])
def admin_appointment():

    data=mm.select_direct_query("select      id,patient,symptoms,problem,doctor,date,time      from
booking_details")
    return render_template('admin_appointment.html',items=data)

```

```

@app.route("/admin_add_hospital", methods = ['GET', 'POST'])
def admin_add_hospital():
    if request.method == 'POST':
        hospital_name = request.form['hname']
        area = request.form['area']
        dean = request.form['dean']
        hospital_code=request.form['hcode']
        address=request.form['address']
        username=request.form['username']
        password=request.form['password']

        maxin=mm.find_max_id("hospital_details")
        qry=("insert into hospital_details values('" + str(maxin) + "','" + str(hospital_name) + "','" +
str(
        area) + "','" + str(dean) + "','" + str(hospital_code) + "','" + str(address) + "','" + str(
        username) + "','" + str(password) + "','0','0')")

```

```

address=request.form['address']
username=request.form['username']
password=request.form['password']
experience=request.form['experience']
specialist=request.form['specialist']

maxin=mm.find_max_id("doctor_details")
qry=("insert into doctor_details values('" + str(maxin) + "','" + str(doctor_id) + "','" + str(
    doctor_name) + "','" + str(gender) + "','" + str(email) + "','" + str(address) + "','" + str(
    username) + "','" + str(password) + "','" + str(experience) + "','" + str(specialist) + "','0','0')")
result=mm.insert_query(qry)
print(result)
return render_template('admin_add_doctor.html',flash_message=True,data="Success")
return render_template('admin_add_doctor.html'
@app.route("/doctor")
def doctor():
    return render_template('doctor.html')
@app.route("/patient")
def patient():
    return render_template('patient.html')

a1 = str(final_time.hour)
a2 = str(final_time.minute)
a3=str(a1)+":"+str(a2)
maxin=mm.find_max_id("booking_details")
qry = ("insert into booking_details values('" + str(maxin) + "','" + str(patient) +
"', '"+str(patient_name)+'','" + str(symptoms) + "','" + str(problem) + "','" + str(doctor) + "','" +
str(date) + "','" + str(a3) + "','" + str(contact) + "','" + str(email) + "','" + str(address) + "','0','0')")
result = mm.insert_query(qry)
if result==1:

```

```

        return render_template('patient_home.html', flash_message=True, data="Booking
Successfully")
    else:
        return render_template('patient_search2.html', flash_message=True, data="Failed")

return render_template('patient_search2.html')

```

```

@app.route('/patient_appointment', methods = ['GET', 'POST'])
def patient_appointment():
    patient=session['patient']
    data=mm.select_direct_query("select      id,symptoms,problem,doctor,date,time      from
booking_details where patient_name='"+patient+"'")
    return render_template('patient_appointment.html',items=data)
if __name__ == '__main__':
    app.run(debug=True, use_reloader=True)

```

PATIENT

```

/*# sourceMappingURL=bootstrap.css.map */

</style>
<!-- //for-mobile-apps -->

<link href="css/font-awesome.css" rel="stylesheet">
<link href="//fonts.googleapis.com/css?family=Oswald:400,500,600,700"
rel="stylesheet">
<link
href="//fonts.googleapis.com/css?family=Open+Sans:300,300i,400,400i,600,600i,700,700i,800,
800i&subset=cyrillic,cyrillic-ext,greek,greek-ext,latin-ext,vietnamese" rel="stylesheet">

<style type="text/css">
<!--

```

```

.style1 {color: #FF0000}
-->
</style>
</head>

<body>
  <!-- header -->
  <section class="w3layouts-header py-2">
    <div class="container">
      <!-- header -->
    </div>
  </section>
  <header>
    <nav class="navbar navbar-expand-lg navbar-light bg-gradient-secondary">
      <h1>
        <a class="navbar-brand" href="#">
          Appointment
        </a>
      </h1>
      <button class="navbar-toggler ml-md-auto" type="button" data-toggle="collapse"
data-target="#navbarSupportedContent" aria-controls="navbarSupportedContent"
aria-expanded="false" aria-label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse" id="navbarSupportedContent">
        <ul class="navbar-nav ml-lg-auto text-center">
          <li class="nav-item active mr-3">
            <a class="nav-link" href="/patient_home">Home
              <span class="sr-only">(current)</span>
            </a>
          </li>
          <li class="nav-item mr-3">
            <a class="nav-link" href="/patient_search">Search</a>
          </li>
        </ul>
      </div>
    </nav>
  </header>

```

```

    </li>

    <li class="nav-item mr-3">
      <a class="nav-link" href="/patient_appointment">appointment</a>
    </li>

    <li class="nav-item mr-3">
      <a class="nav-link" href="/">Logout</a>
    </li>
  </ul>
</div>
</nav>
</header>
<!-- //header -->
</div>

```

7.2 SCREENSHOTS

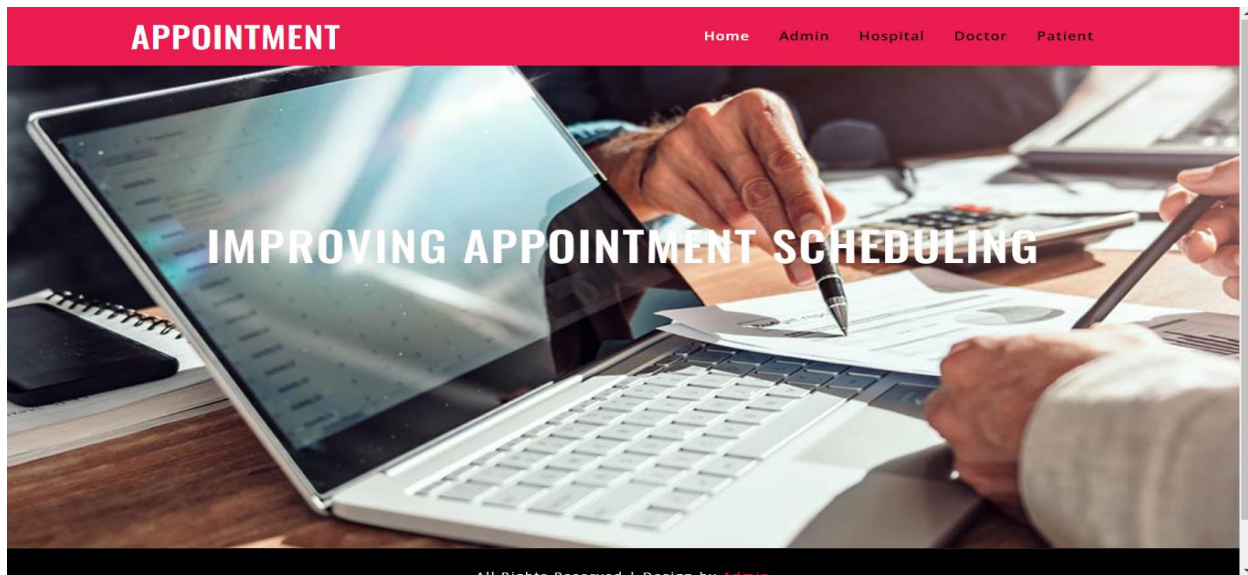


Fig 7.1 home page of appointment scheduling

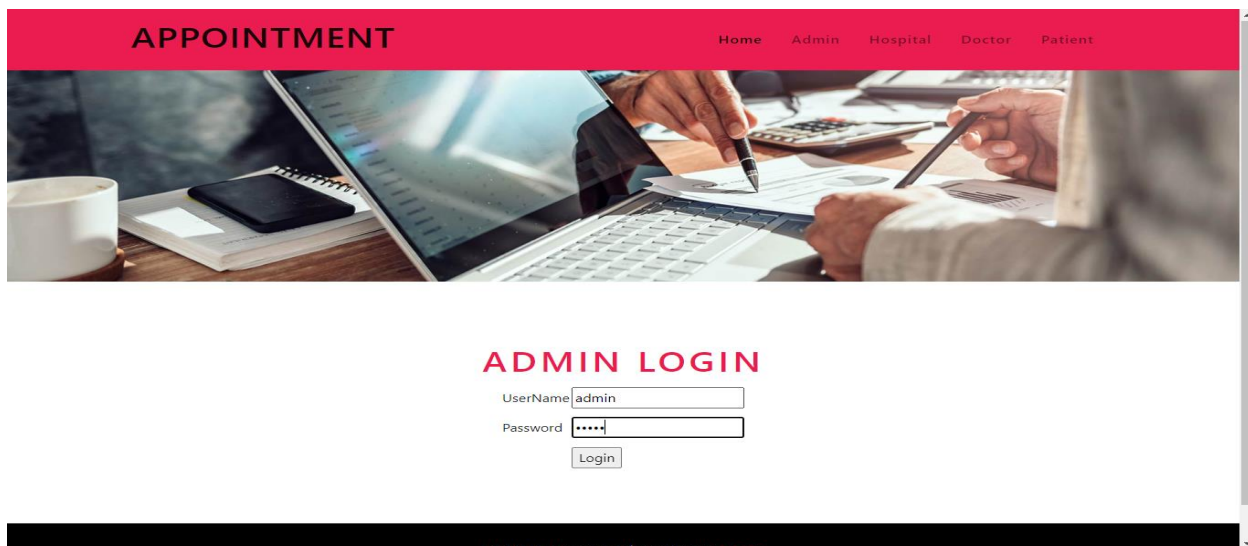


Fig 7.2 Admin page



PATIENT REGISTER

Name	kalai
Contact	9876453210
Email	kalai@gmail.com
Address	trichy
Username	kalai
Password	...
Submit	

Fig 7.3 patient login page



DOCTOR LOGIN

UserName	raja
Password	...
Login	
New User	

All Rights Reserved | Designed by [www.dreamtost.com](#)

Fig 7.4 doctor login page

CHAPTER 8

CONCLUSION AND FUTURE WORK

8.1 CONCLUSION

In this project, the suggested creating an app web application that would utilize artificial intelligence to diagnose health issues and schedule appointments automatically. Appointment scheduling is helpful website for determining what steps should be taken to enhance one's health by simply removing negative outcomes without a long waiting time. The website app will be helpful in to reduce the waiting time for the concern patients for appointment booking. The system automatically creates a scheduling so that timing conflicts are avoided and waiting times are minimized.

8.2 FUTURE ENHANCEMENT

In future systems the implementation can be known with the further identification of the disease with hyper parameters. The search of the disease with the identification can be made more relevance system. Google maps can be also added to get nearby hospitals in emergency cases. Integrate telemedicine capabilities into the appointment assistance system to offer remote consultations for non-urgent cases. This can help reduce the number of patients physically present in the clinic at any given time, easing congestion and minimizing wait times. Implement AI algorithms to assist with triage by analyzing patient symptoms and urgency levels. This helps prioritize appointments and allocate resources more effectively, ensuring that patients with urgent medical needs receive prompt attention. Introduce virtual health assistants powered by natural language processing (NLP) to answer patient queries, provide medical advice, and assist with appointment scheduling. These assistants can improve accessibility to healthcare services and alleviate the burden on clinic staff. Incorporate IoT devices and remote monitoring technologies to track patient vital signs, medication adherence, and disease progression outside of clinic visits. This data can be integrated into the appointment assistance system to support proactive care management and early intervention.

REFERENCES

1. Tiago Salgado de Magalhães Taveira-Gomes, 2017, Reinforcement Learning for primary care appointment scheduling, Faculdade de Engenharia da Universidade do Porto Mestrado de Engenharia da Informação
2. T. Cayirli and E. Veral, 2003 “Outpatient scheduling in health care: a review of literature,” *Prod. Oper. Manag.*, vol. 12, no. 4, pp. 519–549,
3. D. Gupta and B. Denton, 2008 “Appointment scheduling in health care: Challenges and opportunities,” *IIE Trans.*, vol. 40, no. 9, pp. 800–819
4. Xiaojun Zhang, 2012, Developing an online patient Appointment Scheduling system based on web services architecture, Chinese Academy of Sciences EET ALAPAMI 2012 Conference Proceedings
5. S. Sri Gowthem, & K.P. Kaliyamurthi, 2015, Smart Appointment Reservation System, *International Journal of Innovative Research in Science, Engineering and Technology*, Vol. 4, Issue .
6. Xiaojun Zhang, Dr. Ping Yu, Dr. Jun Yan, Hongxiang Hu, and Dr. Niraj Gourea, 2012,
7. Peng Zhao, Illhoi Yoo, Jaie Lavoie, Beau James Lavoie, and Eduardo Simoes, 2017, Web-Based Medical Appointment Systems: A Systematic Review Vol 19, No 4 (2017): April
8. D van Brenk, 2016, Reducing Waiting Times In The Pre-Anaesthetic Clinic Of Vu University Medical Center, Master Thesis university of Twente
9. Al-Harajin RS, Al-Subaie SA, Elzubair AG. The association between waiting time and patient satisfaction in outpatient clinics: findings from a tertiary care hospital in Saudi Arabia.
10. Xie W, Yang X, Cao X, Liu P. Effects of a comprehensive reservation service for non-emergency registration on appointment registration rate, patient waiting time, patient satisfaction

and outpatient volume in a tertiary hospital in China. *BMC Health Serv Res.* (2019) 19:782. 10.1186/s12913-019-4652-6

11. Harding KE, Bottrell J. Specific timely appointments for triage reduced waiting lists in an outpatient physiotherapy service. *Physiotherapy.* (2016) 102:345–50. 10.1016/j.physio.2015.10.011

12. Sharma KA, Yadav A, Sridhar C, Malhotra N, Biji S, Kumari K. Reducing the waiting time to initiation of infertility treatment at a tertiary care centre in India. *BMJ Open Qual.* (2020) 9:e000975. 10.1136/bmjoq-2020-000975

13. Krier M, Kaltenbach T, McQuaid K, Soetikno R. Potential use of telemedicine to provide outpatient care for inflammatory bowel disease. *Am J Gastroenterol.* (2011) 106:2063–7. 10.1038/ajg.2011.329

14. Parikh A, Gupta K, Wilson AC, Fields K, Cosgrove NM, Kostis JB. Short message service reminders reduce outpatient colonoscopy nonattendance rate: a randomized controlled study. *J Gastroenterol Hepatol.* (2021) 36:1044–50. 10.1016/j.amjmed.2009.11.022

15. Zhang M, Zhang C, Sun Q, Cai Q, Yang H, Zhang Y. Questionnaire survey about use of an online appointment booking system in one large tertiary public hospital outpatient service center in China. *BMC Med Inform Decis Mak.* (2014) 14:49. 10.1186/1472-6947-14-49

16. Harding KE, Snowdon DA, Prendergast L, Lewis AK, Kent B, Leggat SF. Sustainable waiting time reductions after introducing the STAT model for access and triage: 12-month follow up of a stepped wedge cluster randomised controlled trial. *BMC Health Serv Res.* (2020) 20:968. 10.1186/s12913-020-05824-z

17. Almomani I, AlSarheed A. Enhancing outpatient clinics management software by reducing patients' waiting time. *J Infect Public Health.* (2016) 9:734–43. 10.1016/j.jiph.2016.09.005

18. Pabinger C, Lothaller H, Lorenz A, Dammerer D. Telemedicine versus on-site treatment at a surgical university clinic: study of 225 consecutive patients. *Int J Med Inform.*

19. Li X, Tian D, Li W, Dong B, Wang H, Yuan J. Artificial intelligence-assisted reduction in patients' waiting time for outpatient process: a retrospective cohort study. *BMC Health Serv Res.* (2021)
20. Liang H, Tsui BY, Ni H, Valentim CCS, Baxter SL, Liu G. Evaluation and accurate diagnoses of pediatric diseases using artificial intelligence. *10.1038/s41591-018-0335-9*
21. Merkow RP, Kaji AH, Itani KMF. The CONSORT framework. *JAMA Surg.* (2021) 156:877–8. *10.1001/jamasurg.2021.0549*
22. Challen R, Denny J, Pitt M, Gompels L, Edwards T, Tsaneva-Atanasova K. Artificial intelligence, bias and clinical safety. *BMJ Qual Saf.* (2019) 28:231–7. *10.1108/IJHCQA-03-2020-0052*
23. Yan J, Yao J, Zhao D. Patient satisfaction with outpatient care in China: a comparison of public secondary and tertiary hospitals. *Int J Qual Health Care.* (2021)
24. Dinh MM, Enright N, Walker A, Parameswaran A, Chu M. Determinants of patient satisfaction in an Australian emergency department fast-track setting. *Emerg Med J.* (2013)
25. Monaghan TF, Agudelo CW, Rahman SN, Wein AJ, Lazar JM, Everaert K. Blinding in clinical trials: seeing the big picture. *Medicina (Kaunas).* (2021) 57:647. *10.3390/medicina57070647*