The International Journal of Engineering and Science (IJES)
|| Volume || 13 || Issue || 2 || Pages || PP 26-53 || 2024 ||
ISSN (e): 2319 – 1813 ISSN (p): 2319 – 1805

Clinical Surgery Scheduling System: A Novel Approach to Enhancing Hospital Efficiency Using Priority Algorithms

Solomon Eje[1], Laud Ochei[2], Chigoziri B. Marcus[3]
Department of Computer Science
University of Port Harcourt, Rivers State, Nigeria
Email: solomon0852@gmail.com[1], laud.ochei@uniport.edu.ng[2], chigoziri.marcus@uniport.edu.ng[3]

----------ABSTRACT----------
One way of enhancing hospital efficiency is through the scheduling of operations, such as clinical surgery. Hospital operations are faced with several critical challenges, including meeting growing demand, adhering to stringent deadlines, mitigating conflicts, and optimizing resource utilization. Although several scheduling strategies have been used in the past to enhance hospital operations, difficulties persist because of the limitations imposed by the hospital's environment in terms of resources, time, space, and expertise. This study aims to develop a clinical surgery scheduling system based on a priority scheduling approach to enhance hospital operation efficiency. The system leverages the LAMP framework to implement a clinical surgery scheduling system by applying a priority scheduling algorithm and rigorously evaluating the system with synthetic datasets. The system prioritizes patients based on urgency and seamlessly integrates them into the surgical schedule, resulting in streamlining operations and improving patient outcomes. The findings reveal substantial enhancements in average wait time and turnaround time for scheduled patients. The study provides valuable insights into the efficacy of priority scheduling and paves the way for future research studies aimed at optimizing hospital operations through innovative algorithmic approaches.

Keywords: Scheduling, Clinical Surgery Scheduling System, Hospital Efficiency, Priority Algorithms

---

Date of Submission: 14-02-2024 Date of acceptance: 28-02-2024

I. Introduction

In contemporary healthcare systems, the effective utilization of time through scheduling systems is crucial for ensuring efficient patient care delivery. While scheduling systems encompass various methods for organizing time, the advent of software applications equipped with sophisticated algorithms has revolutionized scheduling practices. These computerized systems employ algorithms and rules to streamline appointment management and operational activities within healthcare facilities (Seida, 2019).

Over the past few decades, healthcare expenditures have surged, underscoring the importance of prudent resource management. Among the critical areas requiring meticulous attention is the operating room, which holds significant potential for cost savings. Despite its pivotal role, operating rooms often operate below their targeted utilization rates, with research indicating that inefficiencies in scheduling contribute significantly to this shortfall (Min & Yin, 2017).

Patient prioritization is fundamental in clinical surgery scheduling, as it reflects the urgency of medical and social needs. Typically, patients are categorized into urgency groups, each assigned a recommended timing for treatment. However, the criteria for determining patient priority lack standardization, leading to inconsistencies in scheduling practices. While medical conditions and disability are often considered, a standardized approach is needed to ensure equitable access to surgical services (Min & Yin, 2017).

This study focuses on addressing the challenge of building an effective scheduling system for elective surgery patients from a waiting list, with a particular emphasis on integrating patient priority considerations into the scheduling model. Following outpatient consultations, patients are assigned priorities categorized as emergent, urgent, or elective, based on their medical needs. Subsequently, patients are scheduled according to their priority and the capacity of the operating room, typically within a week or a few days in advance. The construction of a patient queue based on priority aims to facilitate timely access to surgery and optimize resource allocation (Min & Yin, 2017).

This study aims to present the design and implementation of a Clinical Surgery Scheduling System based on a priority scheduling algorithm. The main contributions of the paper are -

1. Present a review of scheduling algorithms and supporting tools for the development of a scheduling system using priority algorithms.
2. Designing and implementing a Clinical Surgery Scheduling System in hospitals based on a priority algorithm

3. Demonstrating the application of the Priority Scheduling Algorithm to the clinical surgery scheduling system.
4. Evaluating the performance of the clinical surgery system based on a synthetic dataset

The findings of this study demonstrate the potential of priority scheduling in improving average wait times and turnaround times for patients with scheduled appointments.

The rest of the paper is organised as follows: Section 2 is the background of the problem. Section 3 is the review of related concepts and literature. Section 4 is the analysis and design, and Section 5 is the implementation of the system. Section 6 is the results and discussion of the results. Section 7 concludes the research with future work.

II. Background of the Problem

Efficient scheduling of operations within hospital settings is paramount for ensuring timely delivery of healthcare services amidst the ever-increasing demand for medical attention. Hospitals grapple with the challenge of orchestrating operations effectively to optimize resource utilization and minimize delays. This challenge is exacerbated by the intricate structure of hospital systems, where various units must operate in sync to achieve maximal performance. This research highlights the complexity of hospital workflows, with studies revealing that a significant proportion of hospital visits involve multi-disciplinary cases, requiring coordination among multiple specialities to provide comprehensive care (Smith et al., 2018).

The collision of duties among medical practitioners and the allocation of operating theatre resources often results in delays and setbacks in essential clinical operations. Such delays can have severe consequences, potentially leading to adverse patient outcomes and, in extreme cases, loss of life. Consequently, the imperative for efficient scheduling practices within hospital environments cannot be overstated. Streamlining scheduling processes is essential not only for enhancing operational efficiency but also for improving patient outcomes and satisfaction.

To address the challenges inherent in hospital scheduling, various strategies and algorithms have been proposed. One such approach is the utilization of heuristic scheduling algorithms tailored specifically for surgery arrangements. These algorithms aim to optimize the allocation of surgical resources by integrating different strategies for generating initial patient populations and selecting patients for surgery based on predefined priorities (Tonkins, 2015). By leveraging heuristic techniques, these algorithms can navigate the complexities of scheduling surgical procedures, considering factors such as surgical urgency, resource availability, and patient preferences.

The relevance of scheduling algorithms extends beyond hospital administrators and staff to encompass a broad spectrum of stakeholders, including patients, medical practitioners, and the community at large. For hospitals, the adoption of priority scheduling algorithms offers the potential to improve the efficiency of surgical operations, reduce wait times, and enhance resource utilization. Patients stand to benefit from expedited access to surgical care, leading to better health outcomes and reduced morbidity. Medical practitioners can leverage these algorithms to optimize their workflow, ensuring that surgeries are performed at the earliest appropriate time to achieve optimal results and mitigate complications. Moreover, the broader community benefits from improved healthcare delivery, leading to enhanced well-being and quality of life for individuals within the community (Jones et al., 2020).

In summary, the application of priority scheduling algorithms in the design and implementation of clinical surgery scheduling systems holds immense promise for addressing the challenges associated with hospital operations. By employing heuristic approaches to optimize surgery arrangements, these algorithms have the potential to revolutionize scheduling practices, leading to more efficient resource allocation, reduced delays, and improved patient outcomes.

The problem scenario is described as follows:

Patients seeking surgery with varying degrees of surgical needs and types who arrive at different times are collected in each clinic. How can these patients be appropriately assigned a bed for the surgery while taking into consideration the earliest deadline (wait time) using the priority scheduling algorithm?

III. Review of Literature

This section presents an overview of related concepts and discusses related work.

3.1 Overview of Related Concepts:

This section discusses the different types of scheduling including priority scheduling algorithms and their application in healthcare to schedule clinical surgery.

3.1.1 Types of Scheduling System

There are six main types of process scheduling algorithms; First Come First Served, Shortest-Job-First (SJF), Shortest Remaining Time, Priority Scheduling, Round Robin, and Multilevel Queue. Each of these encompasses many technologies and application areas. Different algorithm techniques are being employed in different areas

based on the objective the algorithm seems to achieve and the nature of work in the area of its application. Before choosing a scheduling system, it is important to consider what features and needs the proposed scheduler seeks to address and what it may need in the future Weglarz (2001)
The following section reviews the various scheduling types.
1. First Come First Served (FCFS) Scheduling Algorithm
This is the easiest and simplest type of scheduling approach also known as First in First out (FIFO). Incoming jobs are inserted into the tail (back) of the ready queue and the next process to be executed is taken out from the head of the queue. The CPU is always assigned to the process at the beginning of the queue Peter Alfke (19 Jun 1998). Jobs are considered only based on arrival time, this makes the general wait time to be quite high as the method becomes poor in performance. Other jobs with smaller burst times may wait for a very long time if a lengthy CPU-bound job dominates the CPU. This in turn may lead to a lengthy queue of ready jobs. This effect, according to Jha et al, 2017is called the "convoy effect"
2. Shortest Remaining Time First(SRTF) Scheduling Algorithm
This is also known as Shortest Job First (SJF). It is a preemptive scheduling system where the process closest to its completion is allocated to the CPU. It is mostly applied in batch environments where short jobs are required to be given preference (Mor, 2003). A process in this scheduling method is associated with the length of its next CPU burst; the OS uses these lengths to schedule the process in the shortest possible time. This algorithm type is referred to as a 'selfish' algorithm since longer jobs wait indefinitely for shorter jobs to use up their burst times (Cheng et al., 2006). The SRT supports preemptive and non-preemptive scheduling. The non-preemptive assumes all process arrives at the same time and once a CPU resource is allocated to a process, the process keeps the resource until completion Harchol-Balter (2003).

3. Round Robin (RR) Scheduling Algorithm
This is comparable to FCFS (First Come First Served) scheduling, but pre-emption is added to switch between processes. The round-robin scheduling algorithm is designed specifically for time-sharing systems. A clock interrupt is generated at intervals. When the interrupt occurs, the currently running process is placed in the ready queue and the next ready job is selected on a FCFS basis. Guowang Miao, 2016. This practice is well known as time-slicing since each process is given a slice of time before being preempted. Time-slicing practice reduces the penalty that short jobs suffer with the FCFS scheduling approach (Po, 2019). The process may have a CPU burst less than the time quantum. If however, the CPU burst of the currently executing process is longer than the time quantum, a context switch occurs and the process is put at the tail of the ready queue. Since the round-robin scheduling algorithm approach is based on the length of time quantum or time-slice, if the quantum is very short, the short process moves and completes execution rapidly. This algorithm is easy to implement and starvation-free since all processes are executed without priority (also known as cyclic execution). Silberschatz, (2010)

4. Priority Scheduling Algorithm
Priority scheduling is a programming process which is based on primacy. In this scheduling approach, each process is allotted a priority and the process with the highest priority takes precedence over other processes and is allocated the CPU. Jobs with equal priorities are carried out on an FCFS basis. Shortest Job First (SJF) is an instance of priority scheduling. Based on pre-emption, priority can either be pre-emptive or non-pre-emptive.
Priority scheduling allocates CPU resources based on task priorities, with higher-priority processes receiving precedence. Processes with equal priorities adhere to a First Come First Served (FCFS) approach. The determination of process priorities can be internal or external, depending on factors such as memory requirements, I/O burst ratios, and time limits.

5. Pre-emptive scheduling: Pre-emption is the act of temporarily interrupting a task being carried out by a computer system. In pre-emptive mode, jobs are assigned with priorities. Currently running processes are interrupted and moved from the running state to the ready state or from the waiting state to the ready state by the operating system when a new process with higher priority enters the system. The resources (mainly CPU circles) are allowed to process for a limited amount of time and then are taken away, and the process is again placed back in the ready queue if that process still has CPU burst time remaining. That process stays in the ready queue till it gets the next chance to execute Khanna, (1992).
6. Non-preemptive Scheduling: Also known as cooperative multitasking. In non-preemptive mode, once a process enters into a running state, it continues to execute until it terminates or blocks itself to wait for Input/output or by requesting some operating system service. Here, scheduling does not interrupt a process running. Instead, it waits till the process completes its CPU burst time and then it can allocate the CPU to another process Joe Bartel (November 5, 2011).

3.1.2 How Priority Scheduling Algorithm Works

In the Shortest Job First scheduling algorithm, the priority of a process is usually the inverse of the CPU burst time, i.e. the larger the burst time the lower the priority of that process. In the case of priority scheduling the priority is not always set as the inverse of the CPU burst time, rather it can be internally or externally set. But the scheduling is done based on the priority of the process where the most urgent process is processed first, followed by the ones with lesser priority in order. Processes with the same priority are executed in an FCFS manner (Krishna, 1997).

The priority of a process, when internally defined, can be decided based on **memory requirements**, the **ratio of I/O burst to CPU burst, time limits**, the **number of open files**, etc. On the contrary, external priorities are set based on criteria outside the operating system, like the importance of the process.

Illustration of Priority Scheduling

Given five processes, P1, P2 to P5 with different arrival times, burst times and unique priorities. Below is a clear explanation of how priority scheduling works.

Priority Criteria:

1. The lower the value of priority, the higher the priority of a process.
2. Based on FCFS

PROCESS	PRIORITY	BURST TIME	ARRIVAL TIME
P1	1	4	0
P2	2	3	0
P3	1	7	6
P4	3	4	11
P5	2	2	12

Execution steps:

(i) At time = 0, Process P1 and P2 arrive concurrently but P1 has a higher priority of **1** than P2 with priority **2**. Therefore based on the above conditions, execution begins with process P1, which has burst time 4. While P2 is in the queue

Timer		
0	P1	Process 1 (P1) starts execution

(ii) At time = 1 to time = 4, no arrival of a new process, P1 continues and complete execution at time = 4.

(i) At time 4, P1 has finished its execution and P2 starts execution.

Timer		
4	P2	Process 2 (P2) starts execution till Time = 5

(ii) At time=6, P3 arrives with a higher priority of **1**, and preempts P2 that is currently running with priority **2**. Therefore, P2 whose burst time is remaining 1 is sent to the tail of the queue. P3 begins execution.

Timer		
6	P3	P2 is preempted by P3 and P3 begins execution till time= 13

(v) At time = 7 to time = 10, no new process arrives, as a result, P3 continue execution while P2 is still in the waiting queue.

(vi) At time = 11, P4 arrives with a lesser priority of **4** compared to P3 which is already running with a higher priority of **1**. So, P3 continues execution.

(viii) At time = 12, P5 arrives but P3 continue execution since its priority is higher than P5.

(ix) At time=13, P3 completes execution. And P2, P4 and P5 are in the ready queue. P2 and P5 have equal priority but the arrival time of P2 is before P5. Therefore P2 begins execution.

(x) At time= 14, P2 completes its burst time and finishes execution. P4 and P5 are in the waiting state. P5 has the highest priority, so it begins execution.

Timer	P5	P2 completes execution and P5 begins execution till time = 16
14		

(xi) At time= 16, P5 completes it execution. P4 is the only process left. It begins execution.

Timer	P4	P5 Completes it execution and P4 begins execution till time = 20
16		

(xii) At time= 20, P4 complete its execution and no process is left.

Calculating the average waiting time for illustration above
Waiting time = start time – arrival time + wait time for next burst
Where:

P1 = 0 – 0 = 0 P2 = 4 – 0 + 7 = 11 P3 = 6 – 6 = 0
P4 = 16 – 11 = 5 P5 = 14– 12 = 2
AWT = (0 + 11 +0 +5 + 2) / 5 = 18/5 = 3.6

The advantages of priority scheduling include the ability to execute processes based on priority therefore, high-priority processes do not need to wait for long which saves time. This technique provides a good mechanism where the relative importance of each process may be precisely defined.
Priority scheduling also has challenges including if higher priority processes take lots of resources, lower priority processes may starve and will be postponed for an indefinite time. Also, if processes with higher priority keep entering the system, lower-priority processes may never get resources which will lead to a convoy effect.

3.1.3 Clinical Surgery Scheduling System: A Case Study
Clinical surgery scheduling is a critical aspect of healthcare operations, facilitating the efficient allocation of resources and the timely delivery of surgical care to patients in need. This section discusses what clinical surgery scheduling entails, why it is necessary, areas it can be used and how it works.

What is a Clinical Surgery Scheduling System?
Clinical surgery refers to the specialized branch of medicine focused on diagnosing, treating, and managing various medical conditions, injuries, and diseases through surgical interventions aimed at restoring or improving bodily function. Surgeons in clinical surgery collaborate with multidisciplinary teams of healthcare professionals to provide comprehensive patient care throughout the surgical process, from preoperative evaluation and surgical intervention to postoperative management and follow-up. (Brunicardi et al., 2020; Townsend Jr. et al., 2019; Williams et al., 2018).

A clinical surgery scheduling system is an automated platform designed to facilitate the scheduling of surgical procedures, encompassing various aspects such as patient appointments, surgeon availability, operating room allocation, and resource management. In essence, it serves as a centralized tool for managing surgical workflows and ensuring the smooth execution of surgical procedures within healthcare facilities (Shiel, 2018).

Why Clinical Surgery Scheduling?
The importance of clinical surgery scheduling cannot be overstated, as it directly impacts patient outcomes, resource utilization, and overall hospital performance. Effective scheduling systems are essential for optimizing surgical workflows, minimizing conflicts, and ensuring that surgeries are conducted on time. By incorporating scheduling algorithms and prioritization techniques, these systems enhance accuracy, efficiency, and operational throughput, ultimately leading to improved patient care and satisfaction.

Areas of Application of Clinical Surgery Scheduling
Clinical surgery scheduling systems find application across various healthcare settings, including primary care, speciality clinics, and elective surgery departments (Gupta & Denton, 2019). These systems play a crucial role in ensuring timely access to healthcare services for patients while maximizing the utilization of available resources. By streamlining appointment scheduling and surgical planning, they contribute to the overall efficiency and effectiveness of healthcare delivery.

How Clinical Surgery Scheduling Systems Work

At the core of a clinical surgery scheduling system lies the prioritization of surgical cases based on predefined criteria such as medical urgency, surgical complexity, and resource availability. The system collates patient data, applies scheduling algorithms to assign priority levels to surgical cases, and generates scheduling outputs accordingly (Kazemian et al., 2017). Surgical cases with higher priority levels are scheduled first, ensuring that critical procedures receive prompt attention and allocation of resources.

This study proposes the implementation of a clinical surgery scheduling system based on priority scheduling algorithms. The system will prioritize surgical cases based on factors such as medical urgency, surgical complexity, and available resources. By integrating priority-based scheduling into the operational workflow, we aim to optimize resource allocation, minimize wait times, and enhance patient outcomes.

3.2 Review of Related Work:

The evolution of scheduling systems in healthcare has been driven by the need to enhance patient access, improve resource utilization, and optimize clinical workflows (Seida, 2019). Scheduling in healthcare settings, particularly in the context of clinical surgery, has garnered significant attention due to its crucial role in optimizing resource utilization and enhancing patient outcomes. As Seida (2019) emphasizes, scheduling systems have evolved from manual processes to sophisticated software applications, employing various algorithms to manage appointments and activities efficiently. However, despite advancements, challenges persist in achieving optimal scheduling practices.

Research indicates a pressing need for effective scheduling strategies in healthcare facilities, given the escalating clinical care expenditures and the imperative to maximize operating room utilization (Min & Yin, 2017). Operating rooms are pivotal components of hospital operations, yet studies highlight their underutilization, often falling short of the targeted 80% utilization rate (Min & Yin, 2017). Ineffectual scheduling contributes significantly to this inefficiency, necessitating innovative approaches to address the issue.

Patient prioritization plays a crucial role in scheduling, with various methods proposed for classifying patients into urgency groups based on medical and social needs (Min & Yin, 2017). However, determining the criteria for patient priority remains a challenge, encompassing medical condition, disability, and social factors, alongside disease-specific outcomes (Min & Yin, 2017). Consequently, there is a need for scheduling systems that can incorporate patient priority effectively.

Numerous studies have focused on addressing the challenges associated with scheduling elective surgeries, emphasizing the importance of considering patient priority in the scheduling process (Cihoric et al., 2020; Sahni et al., 2019). Strategies such as heuristic scheduling algorithms have been proposed to optimize surgery arrangements and improve patient outcomes. Various approaches have been proposed to prioritize surgical cases based on clinical urgency, patient acuity, and surgical complexity. For example, studies have explored the use of urgency scoring systems, such as the Surgical Apgar Score, to prioritize surgical cases based on preoperative variables. Other research has focused on developing decision support tools and predictive models to aid in the prioritization of surgical cases, taking into account factors such as patient demographics, comorbidities, and surgical risk (Sahni et al., 2019).

Integrating priority scheduling algorithms into clinical surgery scheduling systems offers several benefits, including improved resource allocation, reduced wait times, and enhanced patient outcomes (Luo et al., 2018). By assigning priority levels to surgical cases based on clinical need and available resources, these systems enable healthcare providers to optimize surgical schedules, allocate operating room time more effectively, and ensure that urgent procedures are prioritized appropriately. Moreover, the integration of priority scheduling algorithms facilitates real-time decision-making and adaptive scheduling, allowing healthcare providers to respond dynamically to changes in patient acuity and resource availability.

Several review studies have explored the landscape of scheduling algorithms and systems in healthcare. Kazemian et al. (2017) conducted a comprehensive review of machine learning algorithms' application in clinical scheduling, highlighting their potential to enhance scheduling efficiency. Similarly, Bartolini et al. (2016) provided an extensive review of scheduling algorithms for elective surgery, emphasizing the need for robust scheduling approaches to address uncertainty in operating room durations.

The literature also encompasses surveys of scheduling systems in healthcare, examining the challenges and opportunities in this domain. Belien et al. (2015) surveyed various scheduling problems in healthcare, while Guinet and Chaabane (2017) reviewed trends and challenges in scheduling systems. These studies underscore the complexity of scheduling in healthcare settings and the importance of adopting innovative approaches to improve efficiency.

Other relevant studies include those by Shtub and Aharonson (2019), Bruni et al. (2019), and Chou and Yang (2018), which delve into the models, algorithms, and challenges of operating room scheduling. Kim (2015) provides insights into the intricacies of operating room scheduling, while Afonso et al. (2017) discuss surgery scheduling under uncertainty.

In summary, the existing body of research underscores the critical role of scheduling algorithms and systems in optimizing resource utilization and enhancing patient care in clinical surgery settings. By integrating patient priority into scheduling processes and leveraging advanced algorithms, such as heuristic approaches and machine learning, healthcare facilities can achieve more efficient and effective scheduling practices.

IV. System analysis and design

4.1 System Analysis

4.1.1 Analysis of the Proposed System

Scheduling patients for surgeries has long been a challenging task, and despite various efforts by scholars, existing automated schedulers still have limitations. These limitations stem from factors such as the choice of algorithm and the tendency of research to address specific problems without considering other crucial factors, leading to suboptimal system performance post-implementation.

For instance, Gupta and Denton's work on 'Appointment scheduling in health care: Challenges and opportunities' primarily focuses on appointment scheduling, overlooking critical questions regarding facility size, equipment availability, staff allocation, and resource optimization (Gupta & Denton, 2019). Issues similar to those encountered in their system can be found in the study by Chao et al. (2003). However, the proposed system aims to address these limitations by incorporating considerations of available resources into its framework.

The proposed system will utilize priority-based scheduling, whereby patient prioritization will be determined based on factors such as the patient's specific medical condition, severity (categorized as emergent, urgent, or elective), disease stage, and other relevant parameters. These inputs will be used to assign priority values to patients, with those having higher priority values being scheduled first, followed by those with lower priority values. Unlike a traditional first-come-first-serve approach, the proposed system operates on a set of conditions for priority assignment. Nevertheless, patients with equal priorities will be treated on a first-come-first-serve basis.

Crucially, the proposed system will consider questions related to the availability of resources, ensuring that scheduling decisions are made in consideration of factors such as operating room capacity, surgeon availability, and equipment utilization. By integrating resource considerations into the scheduling process, the proposed system aims to optimize resource allocation, minimize wait times, and improve overall efficiency.

4.1.2 System Models

In this section, a visual representation of the proposed system will be provided. The models we will be considering in this section are the process models (use case diagram, flowchart) and data models (entity relationships diagram).

Process Model

In this section, we will provide two process models, namely flowchart and Use Case diagram

Flowchart

Flowcharts are graphical representations used in designing and documenting complex processes. Like other types of diagrams, they help visualize what is going on and thereby help the viewer to understand a process and perhaps also find flaws, bottlenecks, and other less obvious features within it. Flow chart indicates a step-wise transition of the actions and decisions taken.

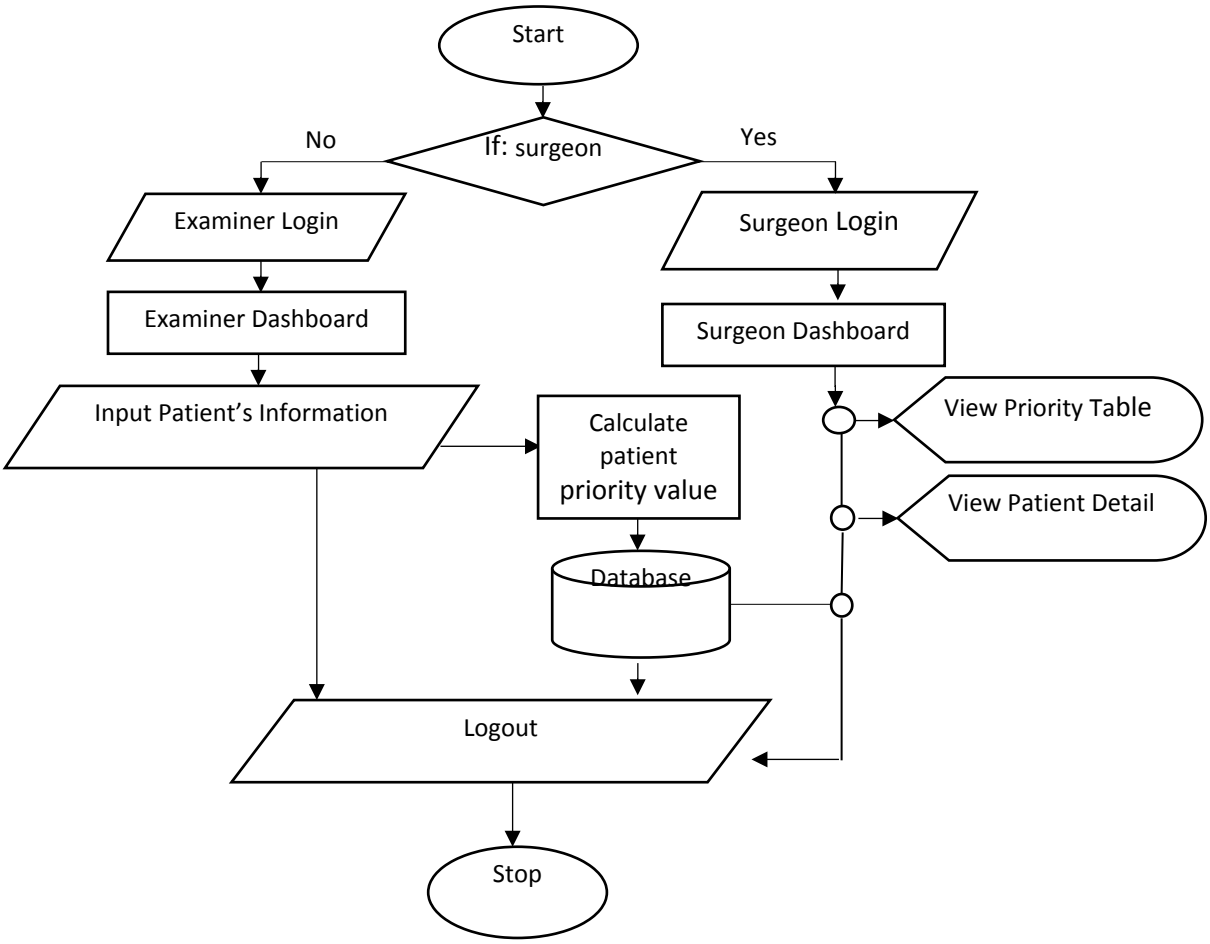


Figure 4.1. Proposed System Flowchart

The flowchart above shows the system flow of the proposed system as it concerns the three major parties (the examiner who examines the patients and enters their details into the system, the surgeon who gets to view the output after the priority algorithm has worked on the data inputted by the medical examiner).

System Use Case Diagram

The use case diagram below defines the relationship between the two principal users of the system, the medical examiner and the surgeon. The examiner examines a patient and uploads the data of those who will require surgery into the system, he/she submits the data, and the priority algorithm within the system works on the data and adds it to a schedule table, which the surgeon can view and know the patient they are assigned to and when the surgery will be conducted, both the examiner and the surgeon will need to login to the system to perform any activity and will both logout when done.

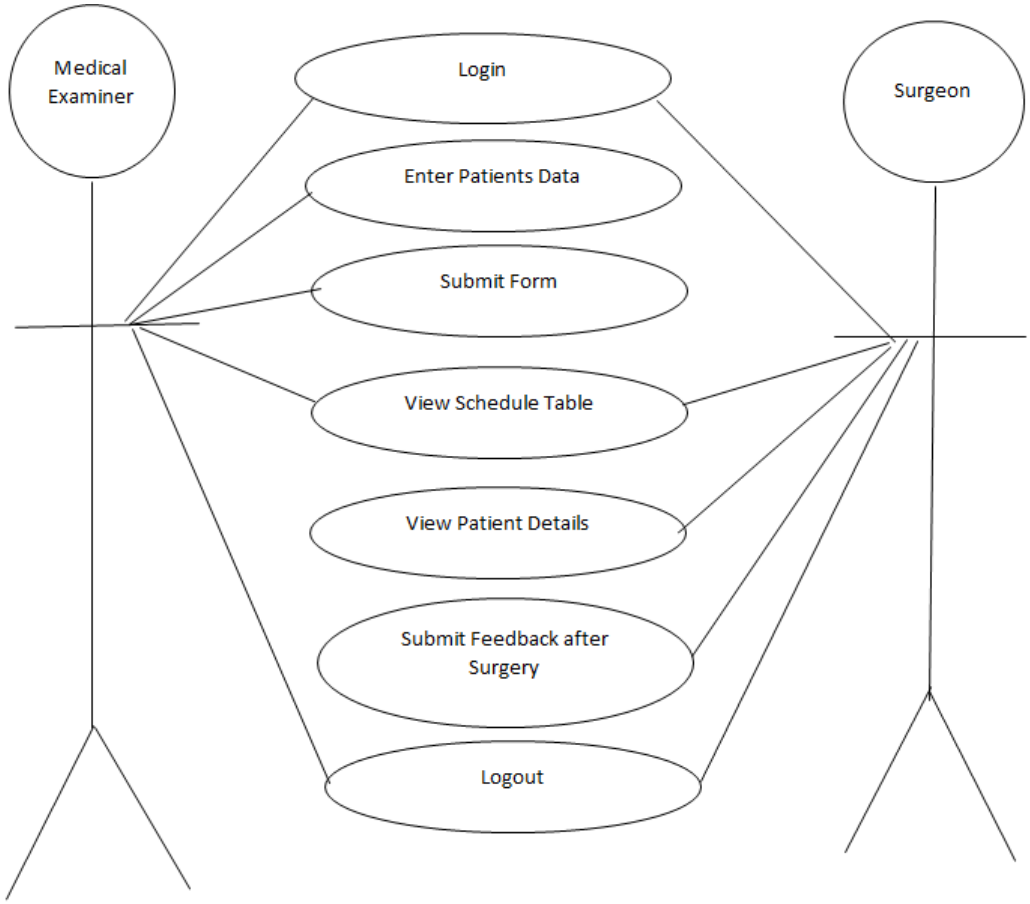


Figure 4.3. System Use Case Diagram

Data Model

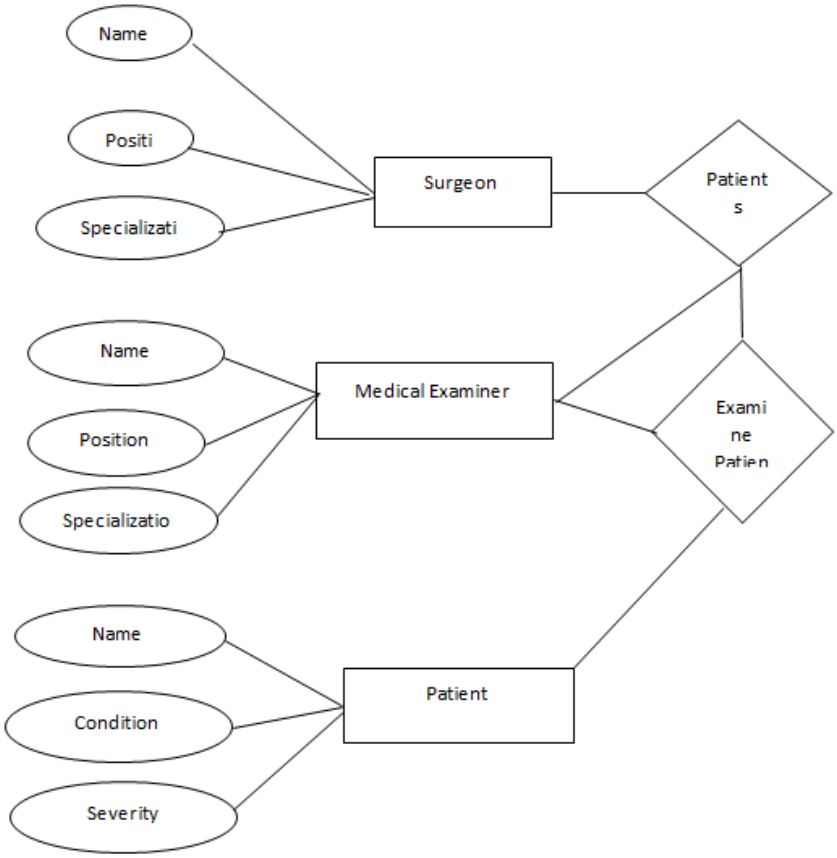


Figure 4.2. Proposed Systems Entity Relationships

4.2 System Design

The Proposed System will have three basic components.

a. The data collection component where the patient's data is collected; will be an input form

b. The data processing component where the inputted data is used to generate an average (this average is generated using a priority algorithm on the inputted data)

c. The output component, where the processed patient's data is displayed to the assigned surgeon in the order of their priority.

4.2.1 Proposed System Architecture Design

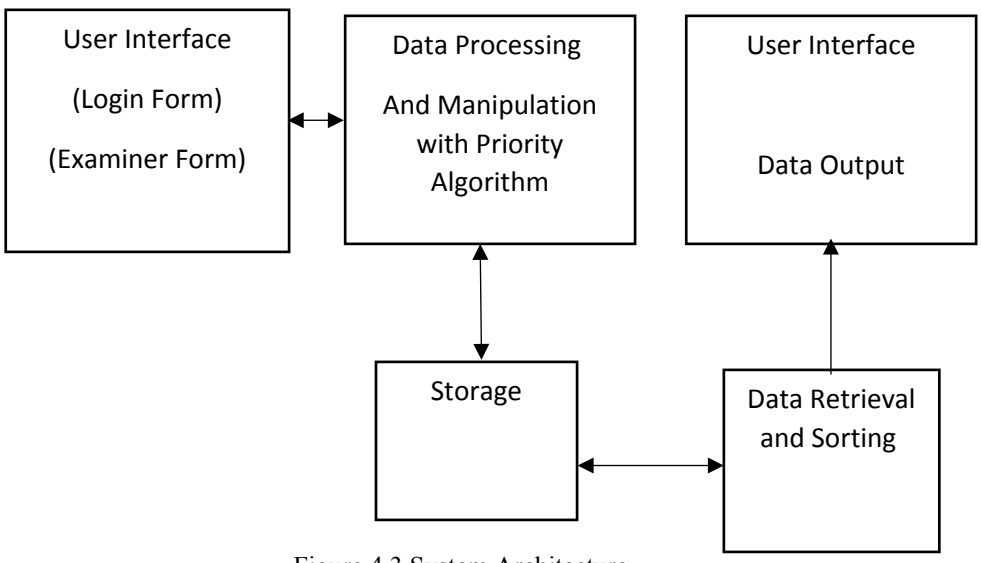


Figure 4.3 System Architecture

The proposed system architecture is a simple one, the components are the

i. User interface (Login Form, Examiner Form, Data Output)
ii. Data Processing: where input data is transformed by the priority algorithm
iii. Storage: This will be an SQL database For storing, user data and patient priority data
iv. Data Retrieval and Sorting: This is where the priority data stored in the database are sorted before being outputted.

4.2.2 User Interface

The user interface design is in two categories: the input design and the output design.

Input Design

The main input for the system is the Patient Examination Form as shown below.

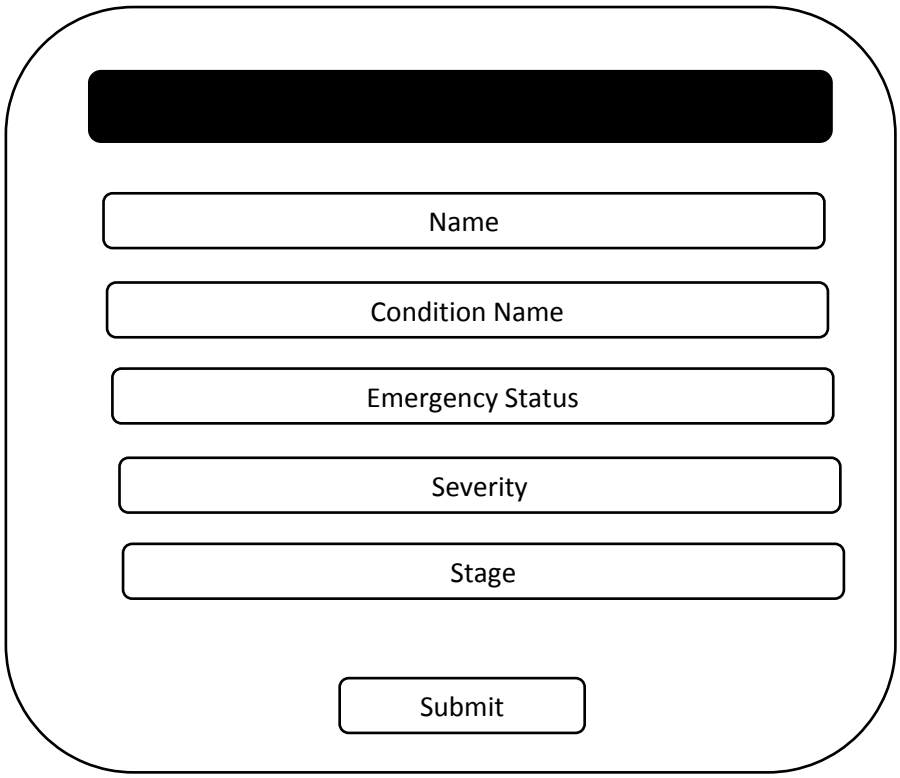


Figure 4.4. Patient Form

4.2.3 Output Design

(a) Schedule / Priority table

S/N	Patient Name	Condition	Schedule Date	Surgeon	View Patient Detail	Status
01	Jane Doe	Breast Cancer	10/12/2020	John Doe	[Details]	Pending

Table 2: Schedule/Priority Table

4.2.4 Program Design

The object-oriented design methodology will be used for the proposed system; all entities in the system will be defined within a class from where several instances will be created.

Object Design

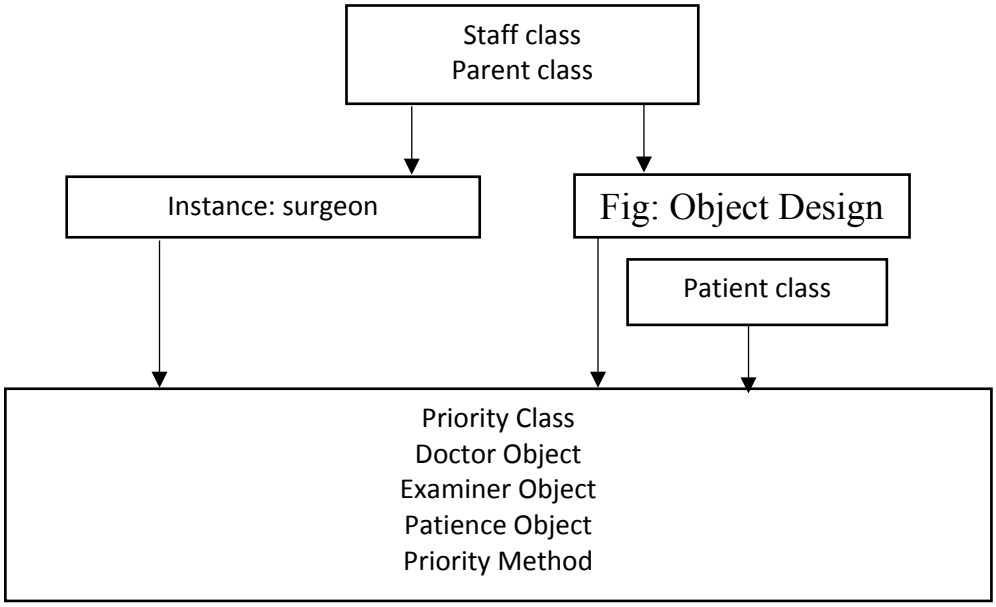


Figure 4.6. Object Design

3.4.4. Database Design

a. Patient Table

Id	Name	Condition	severity

b. Staff table

Id	Name	position	role	Specialization

b. Priority Table

Id	Patient_id	Staff_id	date	status

V. System Implementation and Testing

This section presents system implementation and testing

5.1 System Implementation

The system was implemented with the PHP and LAMP web application frameworks. Overall, the system implements a priority-based scheduling algorithm for clinical surgery, where patients are prioritized based on their condition severity, urgency, and emergency status, and surgeons are assigned based on their specialization and availability. A breakdown of the program logic and functionalities of the system is presented below.

1. postEntry(Request $request):

- This function handles the creation of a new patient entry in the system.
- It extracts information about the patient's condition, severity, urgency, and emergency status from the request.
- Creates a new Patient object and saves it to the database.
- Calculates the priority of the patient based on their emergency status, severity, and urgency.
- Creates a new Entry object associated with the patient, calculates and adjusts the due date based on priority, and assigns a surgeon to the entry.
- If a surgeon is not available, it returns an error message.
- Finally, it saves the entry and redirects to the schedules page.

2. adjustDueDate($score):

- This function adjusts the due date for surgery entries based on the priority score.
- It retrieves all entries with a lower priority score that are scheduled for a date after today.

- For each entry found, it adds 24 hours to the due date and updates the entry.

3. assignSurgeon($surgeon_type, $due_date, $score):

- This function assigns an available surgeon to a patient entry based on the surgeon's specialization, the due date of the surgery, and the priority score of the entry.
- It first checks for an available surgeon of the specified type (surgeon_type).
- If no surgeon of the specified type is available, it checks for a surgeon who is already assigned to another entry scheduled for the same date ($due_date) but with a higher priority score ($score).
- If a suitable surgeon is found, it returns the surgeon object. Otherwise, it returns false.

4. addSurgeon(Request $request):

- This function handles the registration of a new surgeon.
- It creates a new user with the provided name, email, and password.
- Creates a new Surgeon object associated with the user and saves it to the database.

5. validator(array $data) and create(array $data):

- These functions are related to user registration.
- validator validates the input data for user registration.
- create creates a new user with the provided data.

5.2 System Testing

This section will show the test results of the proposed system in the form of screenshots, the key components of the system like user authentication; patient examination form, examination result, surgeon assignment and more will be shown.

Index Page

The index page is the first page a user (Medical examiner or surgeon) is directed to when they enter the system URL in the browser, the user can then navigate to the login page to access their respective dashboards.

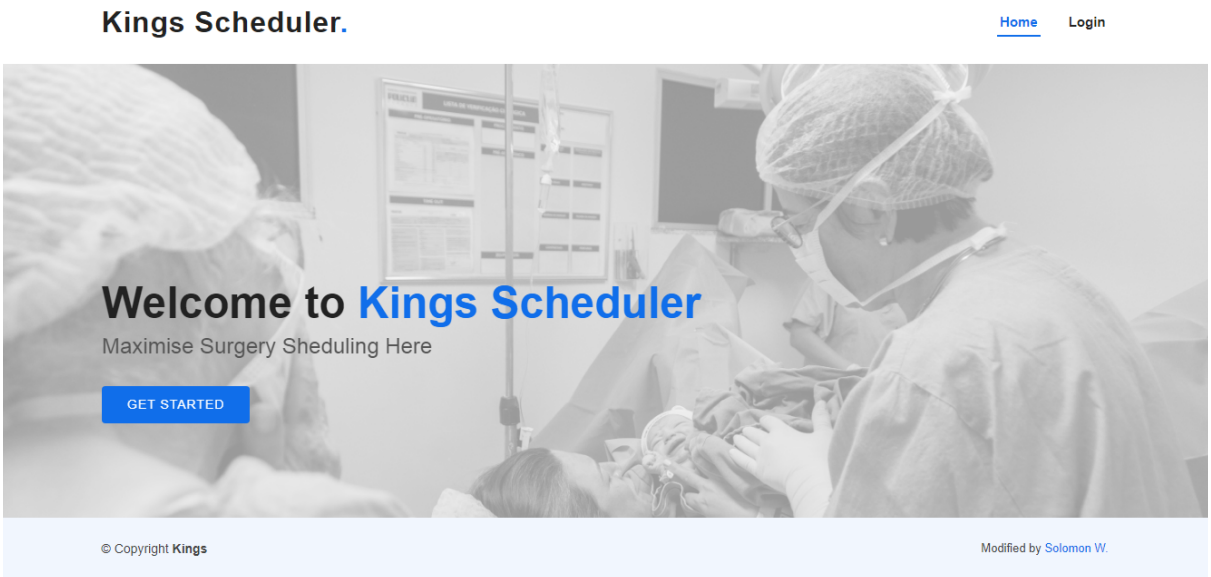


Figure 5.1. Index Page

The system has a single login page for both the medical examiner and the surgeon. The two types of users are redirected to their respective dashboards after Authentication based on their roles. The medical examiner or admin has a dedicated registration page that is separate from that of the surgeon and the patient. The system also provides a functionality for surgeon registration. The surgeon is added to the platform by an admin; afterwards, they are issued their login credential to access relevant patient information.

Patient Entry Form

The patient's information is entered directly into the system by the medical examiner (admin), the form data is used to create a schedule entry for the patient and the patient details are saved for reference.

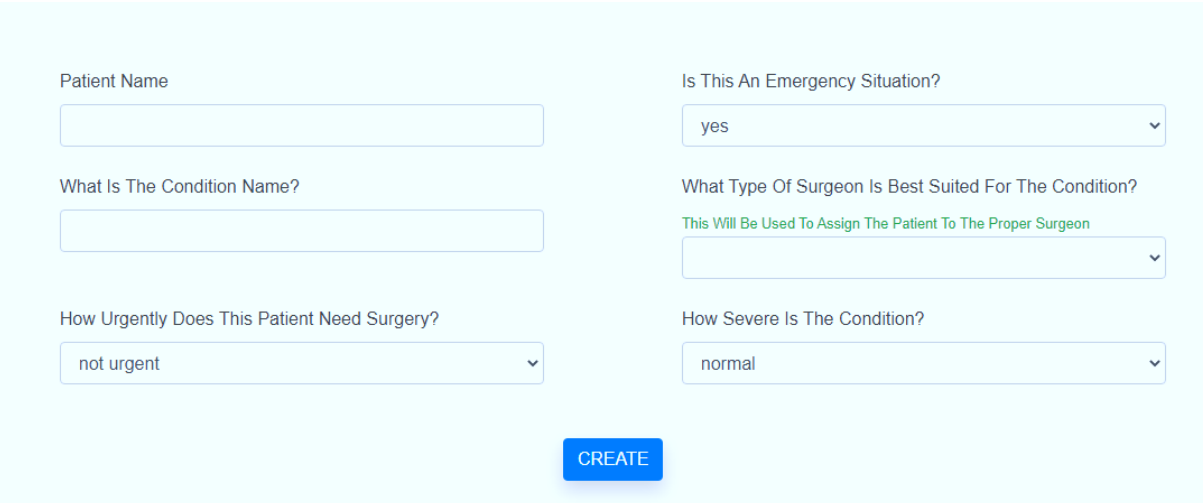
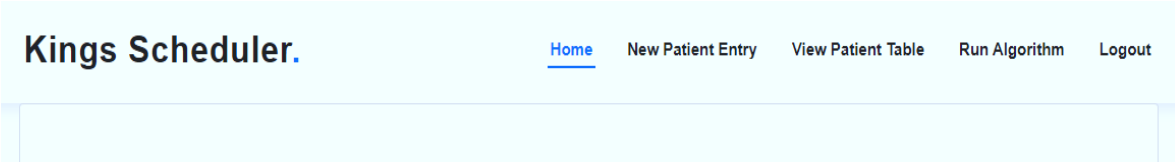


Fig 4.5 Patient Entry Form

User Dashboard

Both the examiner and the surgeon share similar dashboard features with very few exceptions, like the ability to add new surgeons and enter patients' data (unique to the examiner) and the ability to change patients' surgery status (unique to the surgeon).

Fig 4.6 General Dashboard Menu



VI. Results and Discussion

This section shows a sample result and discusses the results. It also shows the screenshot of the priority schedule table containing information that supports the analysis of the results.

6.1 Dataset

The structure of the dataset is presented below in a tabular form. It shows different appointments with arrival and burst times. The dataset for the study is provided in Appendix A for 200 hospital appointments.

The table below shows the arrival and burst time of fifty (50) appointments (A1, A2,...AP50), Calculate the Wait time, Average wait time and Average turnaround time

Appointment	Arrival Time (min)	Burst Time (min)
A1	0	2
A2	1	5
A3	2	4
A4	3	1
A5	4	1
A6	5	2
A7	6	3

Figure 6.1. Dataset for scheduling clinical surgery based on priority scheduling.

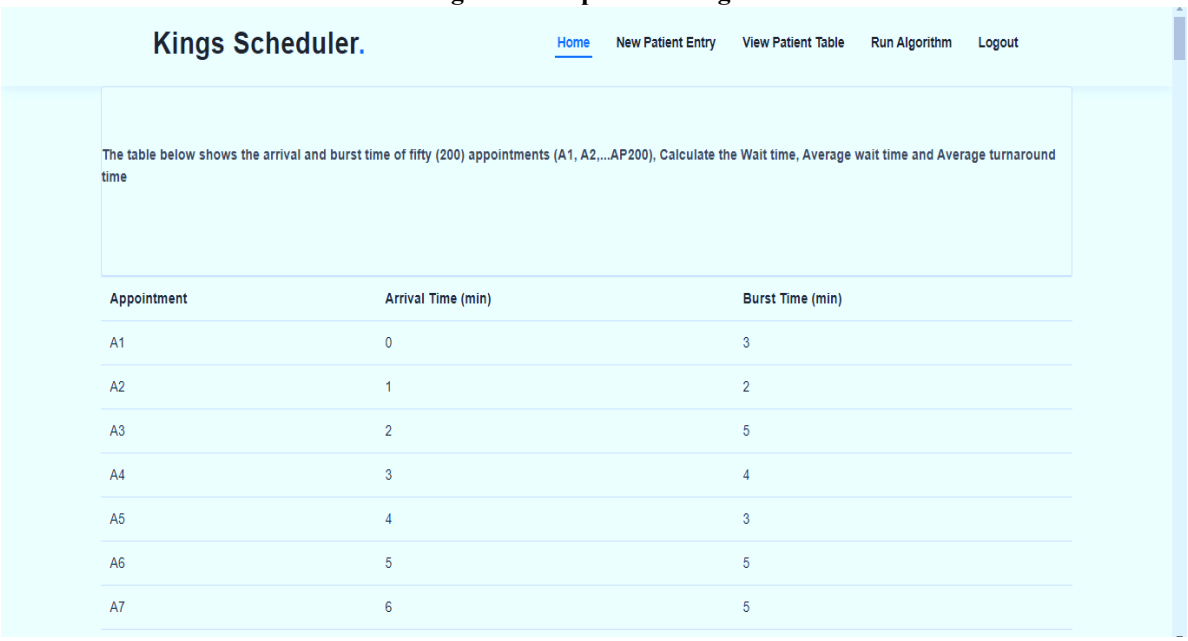
6.2. Results

The result table shows each appointment with priority assigned. The priority algorithm computes and generates a result table from the data set generated above.

Input Page

The input page of the system is presented below.

Figure 6.2. Input Data Page



Result Page

The results page is presented below.

RESULT / OUTPUT:

Appointment	Arrival Time (min)	Burst Time (min)	Completion Time (min)	Turn Around Time (min)	Waiting Time (min)	Priority
A1	0	2	2	2	0	1
A2	1	5	7	6	1	2
A3	2	4	11	9	5	3
A4	3	1	12	9	8	4
A5	4	1	13	9	8	5
A6	5	2	15	10	8	6
A7	6	3	18	12	9	7
A8	7	4	22	15	11	8
A9	8	3	25	17	14	9
A10	9	4	29	20	16	10
A11	10	1	30	20	19	11

Figure 6.3. Result Table [1]

A40	39	4	120	81	77	40
A41	40	5	125	85	80	41
A42	41	3	128	87	84	42
A43	42	3	131	89	86	43
A44	43	2	133	90	88	44
A45	44	3	136	92	89	45
A46	45	5	141	96	91	46
A47	46	2	143	97	95	47
A48	47	2	145	98	96	48
A49	48	2	147	99	97	49
A50	49	2	149	100	98	50

Average Waiting Time: 49.8minutes **Avarege Turn Around Time: 52.78minutes**

Figure 6.3. Result Table [2]

The results shown in a tabular form contain the following information –
(i) Appointment - The appointment shows the patients who are queued for surgeries in their order of priority.
(ii) Arrival Time - The arrival time is the time the patient data was entered into the system.
(iii) Burst Time - This is the amount of time an appointment needs to run from start to completion.
(iv) Turn Around time - To the total amount of time it takes an appointment to complete execution. From the time of submission to waiting, down to the completion of the appointment.
(v) Waiting Time - The interval (mins) between the submission of an appointment and the commencement of execution
(vi) Schedule Date - The date scheduled for the surgery is determined by the priority algorithm.
The results show that priority scheduling can be used to improve the average wait time and turnaround time for patients whose appointments have been scheduled.

The following results show the different average turnaround and wait times at different numbers of appointments.

S/N	NO. OF APPOINTMENTS	AVERAGE TURNAROUND TIME (ATT)	AVERAGE WAIT TIME (AWT)
1	**200**	**218.9**	**215.72**
2	**175**	**191.01**	**187.78**
3	**150**	**162.24**	**159.01**
4	**125**	**134.14**	**130.99**
5	**100**	**105.48**	**102.26**
6	**75**	**77.04**	**73.97**
7	**50**	**50.42**	**47.44**
8	**25**	**25.48**	**22.53**

Table 1. Results showing the different average turnaround and wait times at different numbers of appointments.

VII. Conclusion and Future Work

The implementation of a priority-based scheduling system presents a promising solution to the challenges encountered in hospital operations, particularly in clinical surgery scheduling. This study has demonstrated the feasibility and effectiveness of prioritizing patients based on urgency, leading to improvements in scheduling efficiency and patient outcomes.

Recommendations stemming from our findings urge hospitals to adopt priority-based scheduling systems to optimize resource utilization and enhance patient care. By integrating priority algorithms into their operational workflows, healthcare facilities can streamline surgical scheduling processes, minimize delays, and

ensure timely access to surgical care for patients in need. Further research should focus on real-world implementation and the continued refinement of the proposed scheduling algorithms. Addressing the remaining challenges, such as robust data integration, standardized prioritization criteria, and decision support tool implementation, is crucial for the widespread adoption and success of priority-based scheduling systems in a clinical environment.

While this research predominantly explores scheduling, it delves into the application of priority scheduling in clinical surgery, which forms the core of our study. The development of an offline computerized scheduling system, which generates optimal schedules for surgical cases, underscores our commitment to addressing real-world challenges in healthcare operations. By leveraging a relational database (i.e., MySQL) for data storage and incorporating criteria such as earliest deadline, emergency level, and estimated surgery time, we have laid the foundation for efficient and effective surgical scheduling.

Moving forward, the refinement of priority algorithms, integration of machine learning techniques, and development of personalized scheduling approaches tailored to individual patient needs represent exciting avenues for future research. By embracing innovation and leveraging advanced technologies, we can further enhance the effectiveness and impact of clinical surgery scheduling systems, ultimately improving patient outcomes and healthcare delivery.

References

[1]. Afonso, A. (2017). Surgery scheduling under uncertainty: A review. International Journal of Production Economics, 193, 274-286.
[2]. Bailey, H. R., & Love, M. B. (Eds.). (2018). Bailey & Love's Short Practice of Surgery. CRC Press.
[3]. Bartolini, M., Dellino, G., & Iori, M. (2016). OR and electives management in hospitals: Empirical evaluation of surgery durations, yields and residual capacity. Health Care Management Science, 19(2), 124-136.
[4]. Belien, J., Demeulemeester, E., & Cardoen, B. (2015). Building cyclic master surgery schedules by integrating surgery clustering and surgical team assignments. Omega, 51, 66-81.
[5]. Bruni, M. E., Caraffa, V., Rizzi, A., & Dellino, G. (2019). An exact algorithm for scheduling cyclic operations on a single machine to minimize total weighted tardiness. Computers & Operations Research, 107, 161-171.
[6]. Brunicardi, F. C., Andersen, D. K., Billiar, T. R., Dunn, D. L., Hunter, J. G., Matthews, J. B., & Pollock, R. E. (Eds.). (2020). Schwartz's Principles of Surgery. McGraw-Hill Education.
[7]. Chao, I., Chang, K., & Lin, S. (2003). Outpatient appointment scheduling using the theory of constraints. Computers & Industrial Engineering, 45(4), 637-652.
[8]. Cheng, C., Lin, C., Hsu, P., & Wu, K. (2006). A Genetic Algorithm for the Minimization of the Total Weighted Tardiness in the Job-shop Scheduling Problem. WSEAS Transactions on Computers, 5(10), 2263-2270.
[9]. Cihoric, M., Pauli, M., Zucca, E., & Mamisch-Saupe, N. (2020). Strategy and efficiency of patient prioritization for elective surgery—a Swiss experience. World Journal of Surgery, 44(7), 2240-2248.
[10]. Guinet, A., & Chaabane, S. (2017). Healthcare scheduling systems: Trends and challenges. Computers & Industrial Engineering, 111, 245-255.
[11]. Gupta, D., & Denton, B. T. (2019). Appointment scheduling in health care: Challenges and opportunities. IIE Transactions on Healthcare Systems Engineering, 9(1), 21-36.
[12]. Harchol-Balter, M. (2003). Task Assignment with Unknown Duration. Queueing Systems, 43(1-2), 5-25.
[13]. Jha, K., Patidar, V., & Patidar, S. (2017). Scheduling algorithm effect on overall performance of real-time systems. International Journal of Computer Applications, 158(5), 27-33.
[14]. Jones, S., O'Brien-Pallas, L., Murphy, G. T., Birch, S., & Naylor, D. (2020). Strengthening healthcare services by investing in scheduling excellence. Health Policy, 124(2), 185-191.
[15]. Kazemian, P., Lunardini, D. J., Sano, H., & Moslehi, K. (2017). Machine learning algorithms and their application to classify operating room times using electronic anesthesia records. Journal of Biomedical Informatics, 66, 243-251.
[16]. Kim, S. (2015). Operating room scheduling: Literature review. Journal of the Korean Operations Research and Management Science Society, 40(4), 87-107.
[17]. Krishna, K. (1997). Operating Systems. Tata McGraw-Hill Education.
[18]. Luo, Y., Gao, Z., & Tu, Y. (2018). A two-phase heuristic algorithm for nurse scheduling problem. Journal of Intelligent Manufacturing, 29(1), 49-63.
[19]. Min, W., & Yin, H. (2017). A model for optimizing operating room scheduling. Journal of Computational and Applied Mathematics, 311, 364-379.
[20]. Mor, M. (2003). Optimal control of two parallel machines with sequencedependent setup times. Operations Research Letters, 31(5), 382-390.
[21]. Peter Alfke (19 Jun 1998). FIFO. Retrieved from https://pdfserv.maximintegrated.com/en/an/FIFO.pdf
[22]. Po, L. (2019). Enhanced Periodic Round Robin CPU Scheduling Algorithm. IEEE Access, 7, 132517-132532.
[23]. Sahni, N., Dalal, A., Arora, V., & Duraiswami, R. (2019). A review on appointment scheduling systems in healthcare. Journal of Medical Systems, 43(2), 1-20.
[24]. Seida, A. (2019). Healthcare scheduling systems: From manual processes to sophisticated algorithms. Journal of Healthcare Management, 64(6), 376-389.
[25]. Shiel, W. C. (2018). Surgery definition and types. Retrieved from https://www.medicinenet.com/surgery/article.htm
[26]. Silberschatz, A., Galvin, P. B., & Gagne, G. (2010). Operating System Concepts. Wiley.
[27]. Smith, M., Chon, S., & Yi, H. (2018). Analysis of patient flow in hospitals using a queueing network model. Health Care Management Science, 21(4), 539-554.
[28]. Sulaiman, O. (n.d.). Surgery definition. Retrieved from https://neurosurgery.ufl.edu/about-neurosurgery/what-is-neurosurgery/
[29]. Tonkins, A. (2015). Heuristic scheduling algorithms for operating room planning. Journal of Operations Management, 36, 25-40.
[30]. Townsend Jr, C. M., Beauchamp, R. D., Evers, B. M., & Mattox, K. L. (Eds.). (2019). Sabiston Textbook of Surgery: The Biological Basis of Modern Surgical Practice. Elsevier.
[31]. Weglarz, J. (2001). Scheduling algorithms in multitasking operating system: a comparative analysis. European Journal of Operational Research, 134(2), 245-267.

Appendix

Appendix A - Sample of dataset used to run the program (200 appointments)

Appointment	Arrival Time (min)	Burst Time (min)
A1	0	1
A2	1	2
A3	2	1
A4	3	4
A5	4	3
A6	5	3
A7	6	5
A8	7	3
A9	8	5
A10	9	2
A11	10	3
A12	11	4
A13	12	3
A14	13	3
A15	14	4
A16	15	1
A17	16	2
A18	17	3
A19	18	5
A20	19	2
A21	20	3
A22	21	3
A23	22	3
A24	23	3
A25	24	3
A26	25	4
A27	26	2
A28	27	1
A29	28	2
A30	29	5
A31	30	5
A32	31	5
A33	32	4
A34	33	1
A35	34	3
A36	35	5
A37	36	2
A38	37	1
A39	38	1

A40	39	2
A41	40	3
A42	41	2
A43	42	3
A44	43	1
A45	44	3
A46	45	5
A47	46	4
A48	47	2
A49	48	4
A50	49	5
A51	50	2
A52	51	3
A53	52	4
A54	53	3
A55	54	3
A56	55	5
A57	56	2
A58	57	3
A59	58	2
A60	59	5
A61	60	5
A62	61	5
A63	62	5
A64	63	5
A65	64	2
A66	65	1
A67	66	2
A68	67	2
A69	68	3
A70	69	4
A71	70	5
A72	71	4
A73	72	2
A74	73	1
A75	74	3
A76	75	4
A77	76	3
A78	77	1
A79	78	4
A80	79	5
A81	80	4
A82	81	5

A83	82	3
A84	83	5
A85	84	5
A86	85	4
A87	86	2
A88	87	4
A89	88	3
A90	89	5
A91	90	1
A92	91	5
A93	92	4
A94	93	3
A95	94	1
A96	95	5
A97	96	4
A98	97	5
A99	98	5
A100	99	2
A101	100	5
A102	101	2
A103	102	4
A104	103	3
A105	104	3
A106	105	3
A107	106	2
A108	107	5
A109	108	1
A110	109	2
A111	110	2
A112	111	1
A113	112	4
A114	113	2
A115	114	4
A116	115	5
A117	116	4
A118	117	1
A119	118	4
A120	119	2
A121	120	3
A122	121	1
A123	122	5
A124	123	3
A125	124	1

A126	125	3
A127	126	2
A128	127	5
A129	128	4
A130	129	5
A131	130	3
A132	131	5
A133	132	1
A134	133	4
A135	134	4
A136	135	5
A137	136	1
A138	137	3
A139	138	3
A140	139	4
A141	140	3
A142	141	5
A143	142	3
A144	143	1
A145	144	5
A146	145	3
A147	146	5
A148	147	5
A149	148	5
A150	149	3
A151	150	3
A152	151	5
A153	152	3
A154	153	5
A155	154	4
A156	155	1
A157	156	5
A158	157	2
A159	158	2
A160	159	1
A161	160	4
A162	161	4
A163	162	1
A164	163	1
A165	164	1
A166	165	5
A167	166	4
A168	167	4

A169	168	4
A170	169	5
A171	170	4
A172	171	4
A173	172	2
A174	173	4
A175	174	4
A176	175	2
A177	176	3
A178	177	1
A179	178	3
A180	179	4
A181	180	3
A182	181	4
A183	182	1
A184	183	1
A185	184	5
A186	185	4
A187	186	1
A188	187	4
A189	188	2
A190	189	3
A191	190	1
A192	191	3
A193	192	3
A194	193	2
A195	194	3
A196	195	5
A197	196	4
A198	197	3
A199	198	1
A200	199	4

Appendix E – RESULTS

(i) Result1 - Arrival time, burst time, completion time, turnaround time, waiting time and priority

Appointment	Arrival Time (min)	Burst Time (min)	Completion Time (min)	Turn Around Time (min)	Waiting Time (min)	Priority
A1	0	1	1	**1**	**0**	1
A2	1	2	3	**2**	**0**	2
A3	2	1	4	**2**	**1**	3
A4	3	4	8	**5**	**1**	4
A5	4	3	11	**7**	**4**	5
A6	5	3	14	**9**	**6**	6
A7	6	5	19	**13**	**8**	7
A8	7	3	22	**15**	**12**	8
A9	8	5	27	**19**	**14**	9
A10	9	2	29	**20**	**18**	10
A11	10	3	32	**22**	**19**	11
A12	11	4	36	**25**	**21**	12
A13	12	3	39	**27**	**24**	13
A14	13	3	42	**29**	**26**	14
A15	14	4	46	**32**	**28**	15
A16	15	1	47	**32**	**31**	16
A17	16	2	49	**33**	**31**	17
A18	17	3	52	**35**	**32**	18
A19	18	5	57	**39**	**34**	19
A20	19	2	59	**40**	**38**	20
A21	20	3	62	**42**	**39**	21
A22	21	3	65	**44**	**41**	22
A23	22	3	68	**46**	**43**	23
A24	23	3	71	**48**	**45**	24
A25	24	3	74	**50**	**47**	25
A26	25	4	78	**53**	**49**	26
A27	26	2	80	**54**	**52**	27
A28	27	1	81	**54**	**53**	28
A29	28	2	83	**55**	**53**	29
A30	29	5	88	**59**	**54**	30
A31	30	5	93	**63**	**58**	31
A32	31	5	98	**67**	**62**	32
A33	32	4	102	**70**	**66**	33
A34	33	1	103	**70**	**69**	34
A35	34	3	106	**72**	**69**	35
A36	35	5	111	**76**	**71**	36
A37	36	2	113	**77**	**75**	37
A38	37	1	114	**77**	**76**	38
A39	38	1	115	**77**	**76**	39

A40	39	2	117	**78**	**76**	40
A41	40	3	120	**80**	**77**	41
A42	41	2	122	**81**	**79**	42
A43	42	3	125	**83**	**80**	43
A44	43	1	126	**83**	**82**	44
A45	44	3	129	**85**	**82**	45
A46	45	5	134	**89**	**84**	46
A47	46	4	138	**92**	**88**	47
A48	47	2	140	**93**	**91**	48
A49	48	4	144	**96**	**92**	49
A50	49	5	149	**100**	**95**	50
A51	50	2	151	**101**	**99**	51
A52	51	3	154	**103**	**100**	52
A53	52	4	158	**106**	**102**	53
A54	53	3	161	**108**	**105**	54
A55	54	3	164	**110**	**107**	55
A56	55	5	169	**114**	**109**	56
A57	56	2	171	**115**	**113**	57
A58	57	3	174	**117**	**114**	58
A59	58	2	176	**118**	**116**	59
A60	59	5	181	**122**	**117**	60
A61	60	5	186	**126**	**121**	61
A62	61	5	191	**130**	**125**	62
A63	62	5	196	**134**	**129**	63
A64	63	5	201	**138**	**133**	64
A65	64	2	203	**139**	**137**	65
A66	65	1	204	**139**	**138**	66
A67	66	2	206	**140**	**138**	67
A68	67	2	208	**141**	**139**	68
A69	68	3	211	**143**	**140**	69
A70	69	4	215	**146**	**142**	70
A71	70	5	220	**150**	**145**	71
A72	71	4	224	**153**	**149**	72
A73	72	2	226	**154**	**152**	73
A74	73	1	227	**154**	**153**	74
A75	74	3	230	**156**	**153**	75
A76	75	4	234	**159**	**155**	76
A77	76	3	237	**161**	**158**	77
A78	77	1	238	**161**	**160**	78
A79	78	4	242	**164**	**160**	79
A80	79	5	247	**168**	**163**	80
A81	80	4	251	**171**	**167**	81
A82	81	5	256	**175**	**170**	82

A83	82	3	259	**177**	**174**	83
A84	83	5	264	**181**	**176**	84
A85	84	5	269	**185**	**180**	85
A86	85	4	273	**188**	**184**	86
A87	86	2	275	**189**	**187**	87
A88	87	4	279	**192**	**188**	88
A89	88	3	282	**194**	**191**	89
A90	89	5	287	**198**	**193**	90
A91	90	1	288	**198**	**197**	91
A92	91	5	293	**202**	**197**	92
A93	92	4	297	**205**	**201**	93
A94	93	3	300	**207**	**204**	94
A95	94	1	301	**207**	**206**	95
A96	95	5	306	**211**	**206**	96
A97	96	4	310	**214**	**210**	97
A98	97	5	315	**218**	**213**	98
A99	98	5	320	**222**	**217**	99
A100	99	2	322	**223**	**221**	100
A101	100	5	327	**227**	**222**	101
A102	101	2	329	**228**	**226**	102
A103	102	4	333	**231**	**227**	103
A104	103	3	336	**233**	**230**	104
A105	104	3	339	**235**	**232**	105
A106	105	3	342	**237**	**234**	106
A107	106	2	344	**238**	**236**	107
A108	107	5	349	**242**	**237**	108
A109	108	1	350	**242**	**241**	109
A110	109	2	352	**243**	**241**	110
A111	110	2	354	**244**	**242**	111
A112	111	1	355	**244**	**243**	112
A113	112	4	359	**247**	**243**	113
A114	113	2	361	**248**	**246**	114
A115	114	4	365	**251**	**247**	115
A116	115	5	370	**255**	**250**	116
A117	116	4	374	**258**	**254**	117
A118	117	1	375	**258**	**257**	118
A119	118	4	379	**261**	**257**	119
A120	119	2	381	**262**	**260**	120
A121	120	3	384	**264**	**261**	121
A122	121	1	385	**264**	**263**	122
A123	122	5	390	**268**	**263**	123
A124	123	3	393	**270**	**267**	124
A125	124	1	394	**270**	**269**	125

A126	125	3	397	**272**	**269**	126
A127	126	2	399	**273**	**271**	127
A128	127	5	404	**277**	**272**	128
A129	128	4	408	**280**	**276**	129
A130	129	5	413	**284**	**279**	130
A131	130	3	416	**286**	**283**	131
A132	131	5	421	**290**	**285**	132
A133	132	1	422	**290**	**289**	133
A134	133	4	426	**293**	**289**	134
A135	134	4	430	**296**	**292**	135
A136	135	5	435	**300**	**295**	136
A137	136	1	436	**300**	**299**	137
A138	137	3	439	**302**	**299**	138
A139	138	3	442	**304**	**301**	139
A140	139	4	446	**307**	**303**	140
A141	140	3	449	**309**	**306**	141
A142	141	5	454	**313**	**308**	142
A143	142	3	457	**315**	**312**	143
A144	143	1	458	**315**	**314**	144
A145	144	5	463	**319**	**314**	145
A146	145	3	466	**321**	**318**	146
A147	146	5	471	**325**	**320**	147
A148	147	5	476	**329**	**324**	148
A149	148	5	481	**333**	**328**	149
A150	149	3	484	**335**	**332**	150
A151	150	3	487	**337**	**334**	151
A152	151	5	492	**341**	**336**	152
A153	152	3	495	**343**	**340**	153
A154	153	5	500	**347**	**342**	154
A155	154	4	504	**350**	**346**	155
A156	155	1	505	**350**	**349**	156
A157	156	5	510	**354**	**349**	157
A158	157	2	512	**355**	**353**	158
A159	158	2	514	**356**	**354**	159
A160	159	1	515	**356**	**355**	160
A161	160	4	519	**359**	**355**	161
A162	161	4	523	**362**	**358**	162
A163	162	1	524	**362**	**361**	163
A164	163	1	525	**362**	**361**	164
A165	164	1	526	**362**	**361**	165
A166	165	5	531	**366**	**361**	166
A167	166	4	535	**369**	**365**	167
A168	167	4	539	**372**	**368**	168

A169	168	4	543	**375**	**371**	169
A170	169	5	548	**379**	**374**	170
A171	170	4	552	**382**	**378**	171
A172	171	4	556	**385**	**381**	172
A173	172	2	558	**386**	**384**	173
A174	173	4	562	**389**	**385**	174
A175	174	4	566	**392**	**388**	175
A176	175	2	568	**393**	**391**	176
A177	176	3	571	**395**	**392**	177
A178	177	1	572	**395**	**394**	178
A179	178	3	575	**397**	**394**	179
A180	179	4	579	**400**	**396**	180
A181	180	3	582	**402**	**399**	181
A182	181	4	586	**405**	**401**	182
A183	182	1	587	**405**	**404**	183
A184	183	1	588	**405**	**404**	184
A185	184	5	593	**409**	**404**	185
A186	185	4	597	**412**	**408**	186
A187	186	1	598	**412**	**411**	187
A188	187	4	602	**415**	**411**	188
A189	188	2	604	**416**	**414**	189
A190	189	3	607	**418**	**415**	190
A191	190	1	608	**418**	**417**	191
A192	191	3	611	**420**	**417**	192
A193	192	3	614	**422**	**419**	193
A194	193	2	616	**423**	**421**	194
A195	194	3	619	**425**	**422**	195
A196	195	5	624	**429**	**424**	196
A197	196	4	628	**432**	**428**	197
A198	197	3	631	**434**	**431**	198
A199	198	1	632	**434**	**433**	199
A200	199	4	636	**437**	**433**	200
			AVERAGE	**218.9**	**215.72**	

(ii) RESULT TABLE SHOWING DIFFERENT AVERAGE TURNAROUND AND WAIT TIME AT DIFFERENT NUMBER OF APPOINTMENTS

S/N	**NO. OF APPOINTMENTS**	**AVERAGE TURNAROUND TIME (ATT)**	**AVERAGE WAIT TIME (AWT)**
1	**200**	**218.9**	**215.72**

2	**175**	**191.01**	**187.78**
3	**150**	**162.24**	**159.01**
4	**125**	**134.14**	**130.99**
5	**100**	**105.48**	**102.26**
6	**75**	**77.04**	**73.97**
7	**50**	**50.42**	**47.44**
8	**25**	**25.48**	**22.53**