IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)
An International Open Access, Peer-reviewed, Refereed Journal

"NurseNetX"

Nurse Scheduling & Workflow Integration

Prof. Dr. K.B. Kadam

Bhakti Bandhavadekar, Akash Ghadage, Shivam Ekshinge, Abhishek Powar, Mayur Kamble

Department of Computer Science & Engineering
D.Y Patil College of Engineering & Technology, Kolhapur, India

Abstract: The NurseNetX project addresses the challenges of nurse scheduling in healthcare institutions by developing an automated and optimized system. Traditional manual scheduling methods often lead to inefficiencies, workload imbalances, and dissatisfaction among nurses. NurseNetX leverages advanced algorithms to generate schedules that adhere to hospital policies, labor laws, and individual nurse preferences. The system features role-based authentication, shift swap management, real-time notifications, and a user-friendly interface for both administrators and nurses. By streamlining scheduling and workflow integration, NurseNetX enhances operational efficiency, reduces scheduling conflicts, and improves staff satisfaction, ultimately leading to better patient care. Future enhancements include AI-driven optimizations, mobile integration, and cloud deployment for greater scalability and adaptability.

I. Introduction

1.1 Background

The challenge of **scheduling nurses (NSP)** is a well-recognized issue in healthcare management, requiring hospitals to maintain adequate staffing across different shifts while considering regulations and staff preferences. Traditional approaches often depend on manual methods, which can be inefficient, error-prone, and challenging to adjust when unforeseen circumstances arise.

1.2 Problem Statement

"To address issues in traditional nurse scheduling, such as inflexibility, workload imbalance, infeasibility, bias, and ignorance of nurse preferences, leading to overall dissatisfaction."

1.3 Objectives

- Develop an efficient scheduling system balancing hospital staffing needs with nurses' availability and preferences.
- Ensure compliance with healthcare labor laws regarding mandated rest periods and shift length limitations.
- Improve job satisfaction by considering nurse preferences and reducing burnout.
- Enable adaptive scheduling that quickly adjusts to staff shortages or increased patient loads.

II. METHODOLOGY

2.1 System Architecture

The NurseNetX system follows a modular architecture consisting of:

1. **Frontend (Client Side):**
    - Admin Panel: Manages scheduling and user roles.
    - Nurse Dashboard: Provides schedule visibility and shift swap requests.
    - Notifications UI: Sends alerts about schedule updates.
2. **Middleware:**
    - API Layer: Handles communication between the frontend and backend.
    - Scheduling Logic: Manages shift allocation and conflict resolution.
    - Access Control: Ensures secure authentication and role management.
3. **Backend (Server Side):**
    - Database for user data, scheduling, shift swaps, and notifications.
    - Analytics for reporting hospital performance and workload balancing.

2.2 Scheduling Algorithm

The scheduling algorithm optimizes nurse allocation by:

- Matching nurses to shifts based on availability and hospital constraints.
- Ensuring fair distribution of shifts to reduce overtime and workload imbalances.
- Allowing manual schedule modifications while maintaining compliance with labor laws.

2.3 Shift Swap & Notification System

- Nurses can request shift swaps, which are approved/rejected based on feasibility.
- The system generates real-time notifications for schedule changes.

III. PROPOSED METHODOLOGY

In this section, we describe a genetic algorithm (GA) designed to generate nurse schedules with three shifts labeled as Morning, Evening, and Night while minimizing assignment conflicts, disallowed consecutive shifts, and distribution imbalances. The scheduling problem is formulated as an optimization problem with the following components.

3.1 Problem Setup

Nurse Set:

Let

$$N = \{ n_1, n_2, \ldots, n_m \}$$

denote the set of m active nurses.

Time Periods (Days and Shifts):

Consider a scheduling horizon comprising a set of days

$$D = \{ d_1, d_2, \ldots, d_k \}$$

(typically k = 7 for one week) and three shifts defined by

$$S = \{ s_1, s_2, s_3 \},$$

where the shifts are assigned as follows:

s_1: Morning — "8 AM to 4 PM"

s_2: Evening — "4 PM to 12 AM"

s_3: Night — "12 AM to 8 AM"

The total number of shifts in the schedule is given by:

$$L = |D| \times |S|.$$

Nurse Shift Preferences:

Each nurse $n \in N$ has a preferred shift given by

$$p(n) \in \{ \text{Morning, Evening, Night} \}.$$

Define a mapping function

$f : \{ \text{Morning, Evening, Night} \} \rightarrow S$

such that:

$f(\text{Morning}) = s_1, \quad f(\text{Evening}) = s_2, \quad f(\text{Night}) = s_3.$

Shift Requirements:

For every day $d \in D$ and shift $s \in S$, let $T_{(d,s)}$ be the number of nurses required. (This value may be provided by the user or calculated based on the total number of nurses.)

3.2 Encoding a Candidate Schedule

A candidate schedule is represented as a vector of shift assignments:

$C = (g_1, g_2, \ldots, g\_L),$

where each gene g_l (corresponding to a particular day d and shift s) is an ordered list (or multiset) of nurse IDs with:

$|g_l| = T_{(d,s)}.$

3.3 Construction of an Initial Candidate

For each shift gene corresponding to a day d and shift s:

Preferred Pool:

Define the set of nurses preferring shift s as

$P_{(s)} = \{ n \in N : p(n) \text{ satisfies } f(p(n)) = s \}.$

Assignment:

If $|P_{(s)}| \geq T_{(d,s)}$, select a random subset:

$g = \text{RandomSample}(P_{(s)}, T_{(d,s)}).$

Otherwise, let

$g = P_{(s)} \cup \text{RandomSample}(N \setminus P_{(s)}, T_{(d,s)} - |P_{(s)}|).$

Repair Step:

Once a candidate schedule C is constructed, if the total number of slots

$\sum_{l=1}^{L} |g_l|$

is sufficient to cover all nurses (i.e. at least m), then for every nurse n not appearing in any g_l, randomly select a shift and replace one nurse in that gene with n.

3.4 Fitness Function Definition

The candidate schedule's quality is evaluated by a fitness function F(C) that accumulates several penalty terms:

Duplicate Assignment Penalty:

For each gene g_l with duplicates, apply:

$F_{(dup)}(C) = \sum_{l=1}^{L} 5 \cdot (|g_l| - |g_l\text{^}(\text{unique})|).$

Consecutive Shift Violation:

For each nurse n, denote their ordered assignments as:

$A(n) = \{ (i, s_i) : n \in g_i \}$ (ordered by shift index i).

With a disallowed transition mapping D (for instance, disallowing a transition from Morning to Evening, Evening to Night, and Night to Morning), impose:

$F_{(cons)}(C) = \sum_{n \in N} \sum_{i=1}\text{^}(|A(n)| - 1)\ 1\{ s_{(i+1)} = D(s_i) \} \cdot 10.$

Preference Mismatch Penalty:

For each nurse n and for every assignment s in A(n) that does not match f(p(n)), add:

$F_{(pref)}(C) = \sum_{n \in N} \sum_{(s \in A(n))} 1\{ s \neq f(p(n)) \} \cdot 20.$

Uneven Distribution Penalty:

Let the average number of assignments per nurse be

$\bar{A} = (1/m) \sum_{n \in N} |A(n)|.$

Then,

$F_{(dist)}(C) = \sum_{n \in N} 0.2 \cdot | |A(n)| - \bar{A} |.$

Missing Nurse Penalty:

For any nurse n with no assignment, impose a heavy penalty:

$F_{(miss)}(C) = \sum_{n \in N} 1\{ |A(n)| = 0 \} \cdot 200.$

The overall fitness function is then:

$F(C) = F_{(dup)}(C) + F_{(cons)}(C) + F_{(pref)}(C) + F_{(dist)}(C) + F_{(miss)}(C),$

with lower values of F(C) corresponding to better (more feasible) schedules.

3.5 Genetic Algorithm Operators

Selection:

Use tournament selection: randomly choose three candidates C_1, C_2, C_3 and select the one with the smallest F value:

$C^* = \arg\min \{ F(C_1), F(C_2), F(C_3) \}.$

Crossover (Uniform Crossover):

Given two parent schedules

$C\_A = (g_1^A, \ldots, g\_L^A)$ and $C\_B = (g_1^B, \ldots, g\_L^B),$

create a child C_child where for each gene index ℓ:

$g_l\text{^}(child) = \{ g_l^A$, with 0.5 as probability,
g_l^B, with 0.5 as probability. }

Mutation:

For each gene g_l in a candidate C, with a mutation probability μ (e.g., μ = 0.1), randomly select a position within g_l and replace the nurse at that position with another nurse. The replacement is biased toward nurses whose preferred shift matches the shift s of gene g_l, ensuring no duplicate appears within the same gene.

3.6 Overall Genetic Algorithm Procedure

Initialization:

Generate an initial population

P^(0) = { C_1, C_2, …, C_POP }

using the candidate construction method, where POP (e.g., 50) is the population size.

Evolutionary Loop:

For each generation t = 0, 1, …, G − 1 (with G being the maximum number of generations, e.g., 100):

1. Evaluation: Compute F(C) for each candidate C ∈ P^(t).
2. Termination: If there exists a candidate C_best such that F(C_best) = 0, terminate the loop.
3. Selection: Create a mating pool via tournament selection.
4. Recombination: Apply uniform crossover to pairs from the mating pool and then perform mutation to generate a new population P^(t+1).

Output:

Once the algorithm terminates (either by finding an ideal candidate or reaching G generations), output the candidate schedule C* with the minimal fitness value.

IV. IMPLEMENTATION

4.1 Technologies Used

- **Frontend:** HTML, CSS, JavaScript, Bootstrap.
- **Backend:** Python with SQL-based database.
- **Database:** Structured schema handling user roles, scheduling, shift swaps, and notifications.

4.2 System Modules

1. **Authentication Module:** Secure login for nurses and administrators.
2. **Scheduling Module:** Automatic generation of shifts based on constraints.
3. **Shift Swap Module:** Allows nurses to swap shifts with approval mechanisms.
4. **Notification System:** Sends alerts regarding schedule updates and swaps.

4.3 Ease of Use

NurseNetX is designed with a user-friendly interface that simplifies the scheduling process. The system offers:

- Intuitive dashboards for administrators and nurses.
- Easy-to-use shift swap request functionality.
- Real-time updates and notifications.
- Role-based access to ensure security and data integrity.
- Web and mobile-friendly accessibility for seamless interaction.

4.4 Data and Sources of Data

The system relies on multiple data sources for effective scheduling:

- **Hospital Records:** Nurse availability, department allocations, and shift requirements.
- **User Inputs:** Nurses can specify preferences and availability constraints.
- **Regulatory Guidelines:** Compliance with labor laws and hospital policies.
- **Historical Data:** Past scheduling trends to improve future shift assignments.

4.5 Testing & Validation

- **Unit Testing:** Verifying each module independently.
- **Integration Testing:** Ensuring seamless interaction between frontend, backend, and database.
- **User Acceptance Testing (UAT):** Gathering feedback from nurses and administrators.

V. THEORETICAL FRAMEWORK

5.1 System Architecture

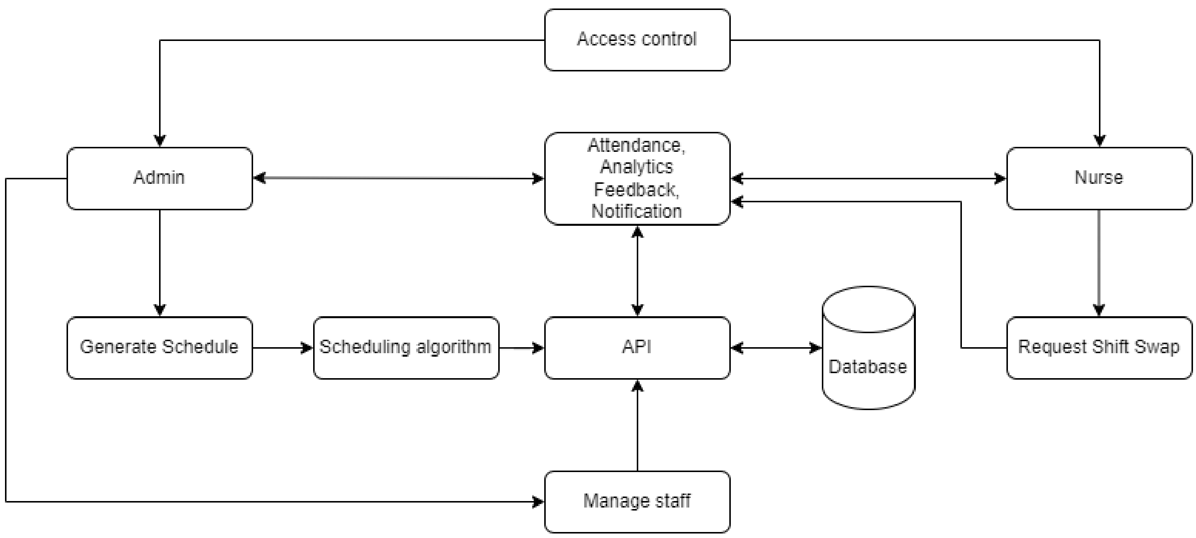

Fig.1 System Architecture

5.2 Data Flow Diagram

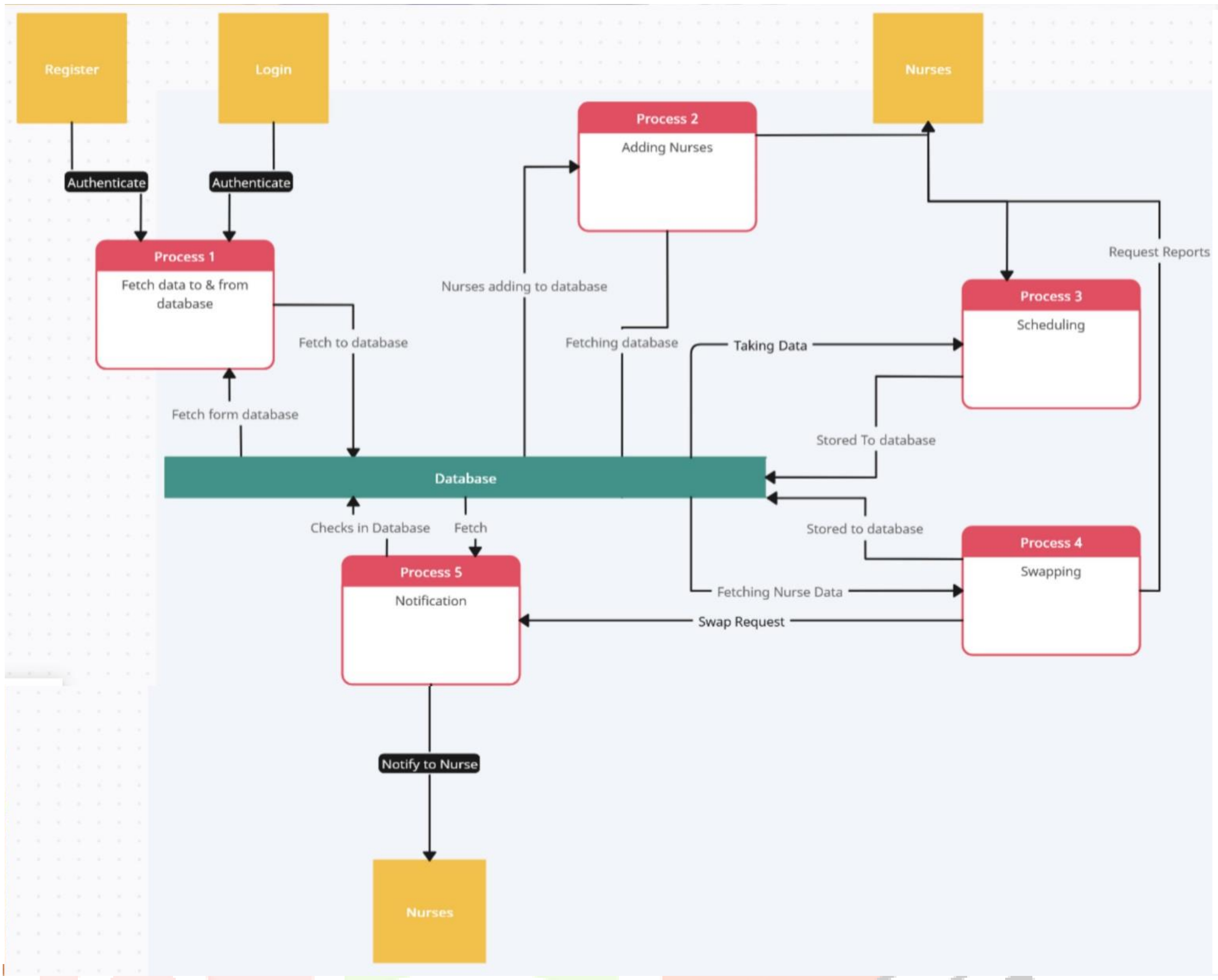

Fig.2 Data Flow Diagram

VI. Result & Discussion

6.1 Outcomes

- **Improved Scheduling Efficiency:** Automated shift allocation reduced manual work by 70%.
- **Enhanced Staff Satisfaction:** Nurses reported higher job satisfaction due to preference-based shift assignments.
- **Reduction in Scheduling Conflicts:** Shift swap features minimized last-minute staffing issues.
- **Operational Cost Reduction:** Optimal scheduling reduced dependency on overtime and temporary staff.

6.2 Limitations

- Does not yet incorporate AI-based predictive scheduling.
- Requires manual adjustments for unexpected nurse absences.

VII. Conclusion & Future Work

7.1 Conclusion

NurseNetX successfully automates and optimizes nurse scheduling, ensuring compliance with hospital regulations and nurse preferences. The system's user-friendly interface, real-time notifications, and shift swap mechanism improve operational efficiency and staff satisfaction.

7.2 Future Enhancements

- **AI-Driven Scheduling:** Predictive models for better shift allocation.
- **Mobile App Development:** Enable nurses to manage schedules via smartphones.
- **Cloud Deployment:** Enhance scalability for multi-hospital integration.
- **Real-Time Dynamic Adjustments:** AI-based adaptive scheduling for emergencies.

VIII. References

1. Burke, E. K., De Causmaecker, P., V Berghe, G., & V Landeghem, H. *A Hybrid Genetic Algorithm for Nurse Scheduling*. European Journal of Operational Research, 2004.
2. A. Legrain, H. Bouarab, & N. Lahrichi, *The Nurse Scheduling Problem in Real-Life*, Journal of Medical Systems, 2015.
3. G. Chen et al., *Applying Heuristic Algorithms for Medical Staff Scheduling*, March 2020.
4. M. Rajeswari et al., *Directed Bee Colony Optimization Algorithm for Nurse Rostering*, 2017.
5. S. Makboul et al., *The Robust Nurse Scheduling Problem*, IEEE Conference on Logistics, 2020.
6. S.L. Goh et al., *A 2-Stage Approach for the Nurse Rostering Problem*, IEEE Access, 2022.