

Sultan Qaboos University

College of Science

Department of Computer Science

COMP3401 – Introduction to Software Engineering

Block Duty Rota System

Complete project

**Submitted to:** Prof Zuhoor Al-Khanjari

**Group Name:** Software Geeks

**Coordinator:** Malak Al-Hinai 133706

**Member1:** Maysoon Ambusaidi 135352

**Member2:** Farah Al-Qaidhi 130194

**Member3:** Wafa Al-Maashani 130690

**Date:** 8 May 2023

**Contents**

[**Abstract** 4](#_Toc134476415)

[**Introduction** 4](#_Toc134476416)

[**Description** 5](#_Toc134476417)

[**Importance** 5](#_Toc134476418)

[**Objectives** 6](#_Toc134476419)

[**Input and output** 6](#_Toc134476420)

[**Functional requirements** 6](#_Toc134476421)

[**Non-functional requirements** 7](#_Toc134476422)

[**Objects** 7](#_Toc134476423)

[**Advantages and disadvantages** 7](#_Toc134476424)

[**Literature Review** 8](#_Toc134476425)

[**Project Management Definition** 11](#_Toc134476426)

[**Project Management Activities** 11](#_Toc134476427)

[**Tasks and Activity Network Diagram** 13](#_Toc134476428)

[**Requirements Analysis Phase Definition** 15](#_Toc134476429)

[**Functional Requirements** 16](#_Toc134476430)

[**Non-Functional Requirements** 17](#_Toc134476431)

[**Use Cases** 17](#_Toc134476432)

[**Process Specification Scenario** 22](#_Toc134476433)

[**Classes** 22](#_Toc134476434)

[**Objects** 23](#_Toc134476435)

[**Requirements Design Phase Definition** 25](#_Toc134476436)

[**Graphical Representations** 25](#_Toc134476437)

[**UML Use Cases Diagram** 26](#_Toc134476438)

[**UML Activity Diagram** 28](#_Toc134476439)

[**UML Sequence Diagram** 29](#_Toc134476440)

[**UML Class Diagram** 31](#_Toc134476441)

[**UML Object Diagram** 32](#_Toc134476442)

[**Algorithms** 33](#_Toc134476443)

[**Implementation** 33](#_Toc134476444)

[**Testing Phase** 37](#_Toc134476445)

[**Maintenance Phase** 38](#_Toc134476446)

[**Future Work** 39](#_Toc134476447)

[**Conclusion** 39](#_Toc134476448)

[**References** 40](#_Toc134476449)

# **Abstract**

The creation of a block duty rota schedule is essential to assemble the health care faculty activities within a hospital. However, it consumes a great deal of time and effort. This project aims to serve an automated system that will facilitate the schedule generation process. Part 1 starts with understanding the problem and it will analyzes the reasons behind the problem specifying the users, input, and output. In the second part, to organize the operations of the medical staff inside a hospital, we provided a literature review of research and an explanation of the notion of project management. Also, the Project Management Activities and Tasks activity diagram. In the third part, the report involves many functional and non-functional requirements. It also contained multiple use cases with certain scenarios. In the fourth part, we described the block duty rota system by using several UML diagrams such as a use case diagram, activity diagram, sequence diagram, class diagram, and object diagram. the last part concludes the implementation, Testing and maintenance phases and future work.

# **Introduction**

An organized health care faculty is fundamental to improve the workflow in hospitals. Block Duty Rota is a shift schedule that is made monthly either by a manager or by self-scheduling. Self-scheduling is defined as “methodology that shifts the responsibility of schedule creation on the faculty themselves” (D’souza et al., 2021). The schedule should have fair distribution of resources, equal night shifts, reduce the shortage of doctors and utilize extra staff efficiently and effectively, in order to ensure fairness in workload and prevent worker burnout in the future, which is quite common in healthcare settings.

The main difficulty is related to the number of health care faculty and their requests. Since the schedule needs to satisfy some constraints like regulations, policies, workload, seniority, aptitude, etc. (Al-Mudahka and Alhamad,2022) and it is done manually by an individual, therefore it takes considerable time and energy to ensure optimal scheduling and to try to fulfill the doctors’ requests at the same time.

Therefore, this project will deal with the issue by automating the scheduling process in The Royal Hospital. This system will help in creating a flexible and more suitable block duty rota schedule which will directly improve the productivity of the staff as well as the delivered quality patient care.

This report includes four parts. The first part implements the system interfaces of our project. The second part tests the system’ s input and output. The third part aims to ensure that the system continues to perform as it was designed to, and repairs or upgrades are done as needed. After the software is released into production, updates or upgrades will be needed. The fourth part explains the future work that can be made or added to the system(2022).

# **Description**

One of the Block Duty Rotas in the Emergency Department in The Royal Hospital schedules the working shifts of 2 levels of staff: specialists and medical officers. Each staff member works a total of 17 shifts monthly (a shift duration is 8 hours). Each worker should have a maximum of 5 night shifts per month. The morning and afternoon shifts are divided equally between the staff. The leaves and public holidays are also considered in the scheduling, each worker deserves a total of 48 days leave per year (6 consecutive weeks maximally) and 28 days of public holidays. Since the scheduling process is done totally by hand, it takes the coordinator a lot of time and effort. The Block Duty Rota System will decrease the coordinator’s role and reduce the scheduling costs by automating the schedule.

The System allows the staff to input their different requests related to holidays, preferable working times…etc at the beginning of each month. The coordinator would only be responsible for accepting or rejecting the requests. After replying to all the requests, the system generates a Block Duty Rota for the month automatically based on the worker’s standard number of shifts, holidays and the accepted requests.

# **Importance**

Department scheduling is an important activity within any hospital. It involves assignment of doctors to shifts under restrictions of time, efficiency, and cost (D’souza et al., 2021).

Health care systems attempt to provide best patient support as well as offering optimal support for health care as such healthcare applications must guarantee fair financial management, high quality medical services support and overall delivery of excellent services (Lalband and Kavitha,2019).

Automated scheduling programs are desirable to reduce work for administrators and improve the efficiency of medical facilities (Squires et al.,2022). Because of the lack of such systems in the ministries, providing an automated Block Duty Rota System will save more time for health care providers, it will manage the departments’ inner work along with arranging the duties, shifts and vacations. It will also work on each request from the user fairly while assigning shifts taking into consideration sick leaves, distances, … etc.

# **Objectives**

This projects’ aim is to save time and effort for employees and the department coordinator, because the scheduling process is a very long and exhausting process. This automated system will help to look forward to organizing the scheduling process and making it easier for all the users. In addition, it will help the workers to look after more important tasks and activities within the hospitals. Finally, the system will allow the users to implement their tasks quickly and accurately. As a result, work efficiency will be greatly increased and the errors caused by a human factor will decrease (Roznovsky,A.2023).

# **Input and output**

The users (health care faculty) of the systems will be asked to enter their requests regarding the shifts of that month such as specifying the night shifts days or weekend shifts with the reason behind that request. The system will generate the schedule depending on the users’ requests. In case there is a conflict between two or more requests the coordinator will accept one request and reject the others counting on the provided reasons along with a specified constraint. At the end, after replying for all requests the Block Duty Rota schedule of the month will be ready to view.

# **Functional requirements**

The system has functional requirements provided to two types of users. The functional requirements which are provided for the staff include logging in, sending requests and viewing the schedule. The requirements which are provided for the coordinator are logging in, accepting and rejecting the requests along with viewing the schedule.

# **Non-functional requirements**

The main non-functional requirements of the system are performance, safety, security and maintainability.

# **Objects**

The system will serve two types of users:

1)Health care staff : The people who input their different requests related to their shifts schedule

2)Coordinator : The person who manages the requests if there is a conflict.

# **Advantages and disadvantages**

This project will organize the scheduling system faster than it normally takes from the person who manages those schedules. It will provide a dynamic and neat workflow. In addition, providing more time to the faculty to focus on their other duties and activities as well as increasing the patient satisfaction due the provided services (Khristich, S.2023). Also, it will help the employees to know their remaining days of leave and to schedule based on their preferable working days.

On the other hand, the system needs effort in maintenance and development because the changes and updates in the Block Duty Rota are continuous. System sudden crashes can also be a disadvantage.

# **Literature Review**

Table 1: Literature review

|  |  |
| --- | --- |
| **Year/ Author** | **Purpose of Study/ Focus on** |
| 1992/ [1] | This paper proposed a management system that manages shifts and holidays of nurses using Formal Grammar based language. |
| 1993/ [2] | This paper discussed a decision support system (DSS). This system has a database and a number of algorithms. Therefore, it is flexible and could be implemented in different hospital environments. Also, the system uses heuristic modeling. |
| 1998/ [3] | The frequent change in staff rota schedules became a must when staff dissatisfaction, absence, and staffing costs increased. The researcher aims to describe a model for the process of generating the duty schedule (shifts schedule) of the nurses to elevate the efficiency of nurses and the services provided by them. |
| 2000/ [5] | This essay presents the findings of a thorough literature analysis on the effects of work schedules and shifts on healthcare workers' health. |
| 2003/ [6] | The paper tried to solve the scheduling system for hospital staff by using the simulation model which determines the staffing needs for each time and the ILP generates an ideal staffing schedule. |
| 2004/ [7] | The paper describes a Genetic Algorithms (GAs) approach to a manpower-scheduling problem arising at a major UK hospital. The approach taken is to use an indirect coding based on permutations of the nurses, and a heuristic decoder that builds schedules from these permutations. The results reveal that the proposed algorithm is able to find high quality solutions. |
| 2005/ [8] | In this paper, a computerized nursing schedule model is developed. The model is approached through a 0-1 linear program. It is implemented experimentally for a six-month period using LINGO and is considered to perform well. |
| 2006/ [11] | The researchers sought to evaluate the impact of the open scheduling system on the nurse's work-life blend and job satisfaction by using a structured questionnaire that is distributed to control the groups. After that, the nurses tried an open-rota system in which they created their shift schedule. |
| 2006/ [10] | The paper proposes an effective algorithm to solve a nurse scheduling problem by using genetic algorithm (GA). The authors proposed an effective algorithm to create and optimize the nurse schedule based on the cooperative GA. The effectiveness of the algorithm was shown by practical experiments. |
| 2006/ [9] | The software solution described in this paper illustrates how the master operation schedule affects the demand for various resources across the hospital. One may compare the master surgery schedule to the hospital's power plant. |
| 2007/ [12] | The paper describes the first ant colony optimization (ACO) approach applied to nurse scheduling.  Extensive computational experiments based on a four-week simulation period were used to evaluate three different scenarios varying the number of nurses and hospitals for six different hospitals’ demand intensities. The results reveal that the proposed ACO algorithm achieves highly significant improvements compared to a greedy assignment algorithm. |
| 2009/ [13] | This paper highlights the scheduling problem of medical residents in a local hospital in Turkey. The resident’s main role is to deliver patient care services by working in shifts through weekdays or weekends. Therefore, a monthly schedule for shifts is created to specify the duties of the residents by considering many constraints which may cause conflicts in the scheduling process. Due to that, a multi-objective programming model has been proposed to automate the process. |
| 2011/ [14] | The paper tried to balance the rosters along with staff satisfaction. They proposed a two-phase approach, initially by solving consecutive assignments. Secondly, improve and reassign the steps using the NSPLib dataset. |
| 2012/ [16] | The researchers tried to solve hospital staff scheduling system problems using Genetic algorithms. And they discussed using MATLAB to provide dynamic GUI. |
| 2017/ [17] | The project discusses the process of the self-scheduling rota system of nurses and its impact on their job satisfaction, autonomy, professional development, and the quality of the delivered services. |
| 2018/ [18] | The paper tried to solve the nurse scheduling problem by mathematical model for the nurse scheduling problem, which is based on the idea of a multi-commodity network ﬂow model. |
| 2018/ [19] | The study focused on the problem of scheduling the shifts of nurses in a hospital. It proposes a long-term nurse scheduling approach based on linear integer programming. A real case study is considered, namely the surgery department of the University Hospital in Cagliari, Italy. It is shown that in most of the cases, the proposed method finds solutions not considered by the hospital planner, greatly reducing redundancies or weakness in the staff, and satisfying all the required constraints. |
| 2020/ [20] | The COVID-19 pandemic brought new challenges in staff management due to the high number of patients in intensive care. As a solution, they created an emergency rota system that is customized and adapted to the crises more flexibly than the existing rota system by using an online free collaborative tool based on a system used in hospitals in London and Cambridge. |
| 2021/ [22] | The researchers designed a social practice-based, worker-centered, and well-being-oriented self-scheduling system which gives healthcare workers more control during shift planning.  In a nine-month appropriation study, they found that workers who were cautious about their social standing in the group or who had a more spontaneous personal lifestyle used the system less often than others. They concluded with guidelines to support individual planning practices, self-leadership, and for dealing with conflicts. |

# **Project Management Definition**

Each task of a project has to go through multiple phases in order to ensure the highest chances of success. Project management is the process of directing the activities of a team in a project to make sure that the software is achieved and delivered within a given requirement and constraints [15]. There are five phases of project management. First, project initiation which describes the purpose of the project and defines the goals, size, constraints, risks and stockholders. Second, project planning is the key to a successful project which starts by developing a project plan and identifying the cost, time, resources, responsibilities, and millstones. Third, project execution which is the phase of putting the activities and tasks of the plan into action to produce the project deliverables. Fourth, project control, this phase usually overlaps with the execution phase to measure the progress and performance of the project and also deals with any issue that occurs. Fifth, project complete, it is the last point of a project. The project will be closed either it was a successful project or a failed one [4].

# **Project Management Activities**

**People Management:**

we need to manage people in the project and assign them to the project earlier. We have distributed enough members for each task in our project. We tried our best to ensure that each member benefits from the work. Also, we collectively worked on some tasks to help the flow of the project.

**Risk Management:**

Risk management is allocating all the possible risks and threats on any project and trying to prepare to solve them [25]. We have managed to collect all our work 1 day before the deadline, so that we can review it to check errors and solve them. We have chosen good team members where we can handle sudden risks.

**Project Planning:**

Project managers are responsible for planning, estimating and scheduling project development and assigning people to tasks. We first agreed on the idea of the project. After that, we started the proposal and the group coordinator distributed the tasks between our group members. We normally take about 7 days to get the work done in the early phases.

**Project Reporting:**

Project coordinator is involved in informing the customers with the project progress and weekly reports must be done. We have provided multiple reports to the project coordinator which shows our project progress. Such as, project proposal, project management and the literature review report.

**Proposal Writing:**

A project proposal is a document that includes an idea of the project and the objectives of it. Also, it gives a brief idea of the project so the reader can approve it [24]. We have written all the objectives and the definition of our problem in the proposal phase. We tried to briefly ensure that the idea and the objectives of the project are clear.

# **Tasks and Activity Network Diagram**



Table 2: Tasks management

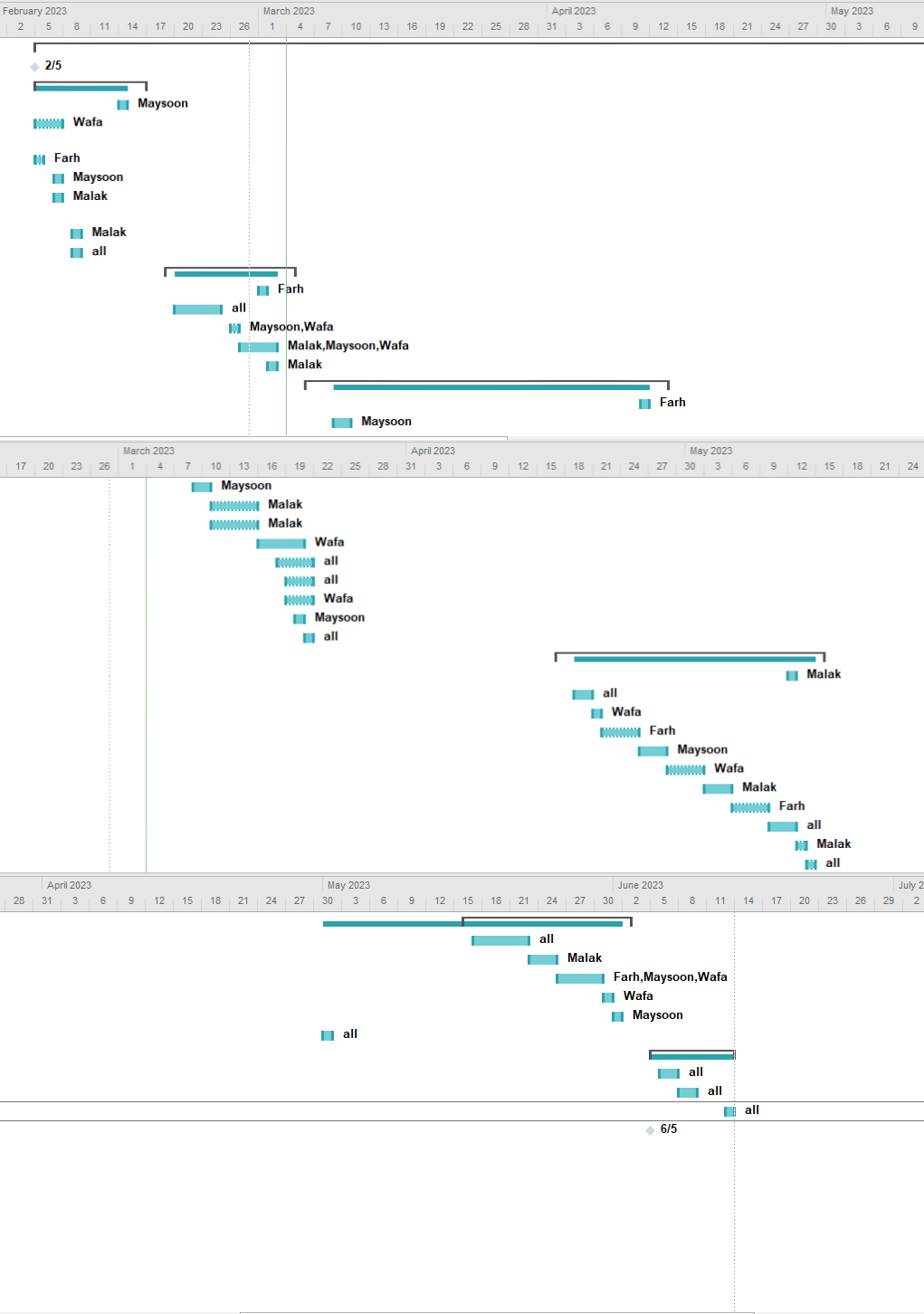
**

Figure 1:Tasks diagram

Table 1 and Figure 1 represent the tasks management of our project. Our work will be done in six phases (project proposal, project management and literature review, requirements analysis phase, project design, complete project and presentation). Each phase is restricted with a time limit and is divided into subtasks to simplify the work. Some tasks are done consequently while others are done in conjunction with each other. We also decided the team members who are responsible of doing each task. We made some tasks individual while we will work on others as a group. The purpose of this is to strengthen the work and to make it simple and organized. We also kept an enough slot between the planned finishing deadline of each phase and the actual one in case of any unexpected delay. We attached the activity network diagram of our task management plan in the appendix.

# **Requirements Analysis Phase Definition**

When the previous phase's objectives have been met, the Requirements Analysis Phase begins. Documentation pertaining to user requirements from the Concept Development Phase and the Planning Phase shall be used as the foundation for additional user needs analysis and the development of detailed requirements.

Successful completion of the Requirements Analysis Phase should include the following:

* 1. Approved requirements definition
  2. Development of Functional Requirements Documentation and specifications
  3. Non-Functional requirement
  4. Planning of planned test activities
  5. Execution of required procurement activities
  6. Approval to move forward to the Design Phase
  7. Use cases
  8. Process specification scenario
  9. Classes
  10. Objects

The Requirements Analysis Phase's goal is to convert the needs and high-level requirements specified in previous phases into unambiguous (measurable and testable), traceable, complete, consistent, and stakeholder-approved requirements.

# **Functional Requirements**

The functional requirements for a system describe what the system should do. These requirements depend on the type of software being developed, the expected users of the software and the general approach taken by the organization when writing requirements (Sommerville, 2011). We considered a number of functional requirements in our system.

**Functional requirements for staff:**

* log in:

Each staff member should be able to log in to the system using his hospital account.

* send request:

The staff member could send a request for holidays, working times, etc. to the coordinator.

* view schedule:

Any staff member is able to view the schedule after approval.

**Functional requirements for coordinator:**

* log in:

The coordinator should also be able to log in as coordinator using his hospital account.

* manage request:

The coordinator can accept or reject the staff requests.

* view schedule:

The coordinator is also able to view the schedule after approval.

## **Non-Functional Requirements**

Requirements that are not explicitly concerned with the functionality of a system but with its properties (Glinz, 2007). They places constrains and restrictions on the design and implementation of the product. This will help in testing and measuring the success of a software system. The main non-functional requirements of this system are:

**Performance:**

A system performance is essential as it indicates how effective and efficient is the system to be able to achieve its desired outcomes. For example, in our system the performance is measured by how fast and accurate it manages the requests and generates the schedule.

**Reliability:**

Which indicates how to perform the systems functions for long periods without facing breakdowns or failures. Our system has to be able to work under different conditions and satisfy the performance standards.

**Maintainability:**

Our system is expected to be operating for long periods of time, so, the system has to be designed and implemented to be easily maintained and repaired in case a failure does occur.

**Security:**

One of the most important features in our system is security, to protect against unauthorized access, modification, or use of the system resources such as personal data.

## **Use Cases**

A use case is a description of how the system and its users interact with each other and it is used to simplify complicated systems’ concepts. Using use cases will show the system demands earlier which will benefit the system development process.

Table 3: Use case 1

|  |  |
| --- | --- |
| **Use case** | Log in |
| **Actors** | Coordinator, staff level1, staff level2 |
| **Purpose** | Secure the system where only the users can access the system |
| **Description** | The user will be having a username and password. He/she must enter the correct password associated with his/her username. |
| **Type** | Primary use case |
| **Pre-condition** | - |
| **Post condition** | A welcome page will be generated. |

Table 4: Use case 2

|  |  |
| --- | --- |
| **Use case** | Manage Requests |
| **Actors** | Coordinator, staff level1, staff level2 |
| **Purpose** | Requesting, accepting and rejecting staff’s calls for shifts per month. |
| **Description** | After logging in the staff level1 & staff level2 will have to apply their preferences for shifts. Then, the coordinator will receive those requests and he/she will make the choice of rejecting or accepting them. |
| **Type** | Primary use case |
| **Pre-condition** | Must log in |
| **Post condition** | The staff will receive a message that his/her request has been accepted or rejected. |

Table 5: Use case 3

|  |  |
| --- | --- |
| **Use case** | Manage Schedule |
| **Actors** | Coordinator, staff level1, staff level2 |
| **Purpose** | Generating a monthly staff schedule and edit it |
| **Description** | After all the requests have been entered the system will generate a monthly schedule where all the staff can see and only the coordinator will be able to edit it in case something happens. |
| **Type** | Primary use case |
| **Pre-condition** | All the requested must be entered for a month |
| **Post condition** | The staff can review their schedules. |

Table 6: Use case 4

|  |  |
| --- | --- |
| **Use case** | Request |
| **Actors** | staff level1, staff level2 |
| **Purpose** | Requesting a shit |
| **Description** | After logging in the staff level1 & staff level2 will have to apply their preferences for shifts. |
| **Type** | Secondary use case |
| **Pre-condition** | Must log in |
| **Post condition** | The staff will receive a message that his/her request has been accepted or rejected. |

Table 7: Use case 5

|  |  |
| --- | --- |
| **Use case** | reject |
| **Actors** | Coordinator |
| **Purpose** | rejecting staff’s calls for shifts per month. |
| **Description** | After the staff have applied his/her request, The coordinator will reject the request if it does not satisfy certain criteria. |
| **Type** | Secondary use case |
| **Pre-condition** | Must log in |
| **Post condition** | The staff will receive a message that his/her request has been rejected. |

Table 8: Use case 6

|  |  |
| --- | --- |
| **Use case** | accept |
| **Actors** | Coordinator |
| **Purpose** | accepting staff’s calls for shifts per month. |
| **Description** | After the staff have applied his/her request, The coordinator will accept the request if it does satisfy certain criteria. |
| **Type** | Secondary use case |
| **Pre-condition** | Must log in |
| **Post condition** | The staff will receive a message that his/her request has been accepted. |

Table 9: Use case 7

|  |  |
| --- | --- |
| **Use case** | View schedule |
| **Actors** | Coordinator, staff level1, staff level2 |
| **Purpose** | Show staff’s own schedule along with the all the staff schedule for the coordinator. |
| **Description** | After the staff has entered his/her request and all the requests have been processed. Each staff member will be able to view his/her schedule. Also, the coordinator will be able to see the staff's schedule after the requesting period is done. |
| **Type** | Secondary use case |
| **Pre-condition** | Must log in |
| **Post condition** | Schedules. |

Table 10: Use case 8

|  |  |
| --- | --- |
| **Use case** | Edit Schedule |
| **Actors** | Coordinator |
| **Purpose** | The coordinator can modify the schedule and edit |
| **Description** | The coordinator will be able to enter shifts for special circumstances. Also edit and manipulate the schedule taking into consideration many factors. |
| **Type** | Secondary use case |
| **Pre-condition** | Must log in |
| **Post condition** | A message that a modification has been made. |

## **Process Specification Scenario**

The first step is that the staff will log in to the system by their hospital account. Next, the staff will send their different requests related to holidays, preferable and working time…. etc. At the beginning of each month. then, the coordinator will responsible for accepting or rejecting the requests according to constraints. after replying to all requests, the system generates a Block Duty Rota for the month automatically based on the worker’s standard number of shifts, holidays, and the accepted requests.

Steps:

1. Staff will log in to the system by their hospital account.
2. Staff will send their different requests related to holidays, preferable and working time…. etc.
3. Coordinator will responsible for accepting or rejecting the requests according to constraints.
4. The coordinator replying to all requests.
5. The system generates a Block Duty Rota for the month automatically based on the worker’s standard number of shifts, holidays, and the accepted requests.

## **Classes**

In UML, a class represents an object or a group of objects that share a joint structure and behavior. Classes, or instances of classes, are common model elements in UML diagrams. A class sets the operations, attributes, relationships, and semantics that instances, or objects, of the class has (IBM,2021).

Table 9: Classes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Classes** | **Staff** | **Coordinator** | **Schedule** | **Request** |
| **Attributes** | StaffName  staffUserName  staffPassword  staffAge  monthlyDuty  holidays | cooredinatorName  cooredinatorUserName  cooredinatorPassword | shiftDuty  duration | type  comment |
| **Operations** | log\_in()  sendRequest()  viewSchedule() | log\_in()  manageRequest()  viewSchedule()  viewRequest() | generateSchedule() |  |

## **Objects**

Objects are instances of a class, which encapsulates some specified data and a set of associated operations (Chin and Chanson,1991). Some objects may correspond to a real-word entities.

Table 10: Objects

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Classes** | **Staff** | **Coordinator** | **Schedule** | **Request** |
| **Objects** | staffLevel1  staffLevel2 | coordinator | schedule | request |
| **Attributes** | staffName  staffUserName  staffPassword  staffAge  monthlyDuty  holidays | cooredinatorName  cooredinatorUserName  cooredinatorPassword | shiftDuty  duration | type  comment |

**staffLevel1 object example:**

* StaffName: Dr. Maryam Khalid
* staffUserName: dr68653
* staffPassword: 12345678@
* staffAge: 46
* monthlyDuty: by default, 17 shifts
* holidays: 76 days

**staffLevel2 object example:**

* StaffName: Dr. Mahmood Qais
* staffUserName: dr55564
* staffPassword: 87654321@
* staffAge: 30
* monthlyDuty: by default, 17 shifts
* holidays: 76 days

**coordinator object example:**

* cooredinatorName: Dr. Maha Fahad
* cooredinatorUserName: dr23781
* cooredinatorPassword: 379173maha

**schedule object example:**

* duration: by default, 8 hours
* shiftDuty: {

6 am - 1 pm: Ahmed, Salim, Amira

1 pm - 7 pm: Sumaya, Nadia, Majed

}

**request object example:**

* type: night shifts
* comment: I would like to have no nights shift during the weekends of this month…

# **Requirements Design Phase Definition**

The Design Phase is the fundamental phase of the System development Life Cycle. The list of requirements that we have developed in the definition phase is used to make design choices. In the design phase, one or more designs are created to obtain the project result. Depending on the project subject, the design phase includes flow-charts, UML diagrams, HTML screen designs, photo impressions, prototypes (Halwai,S.,(2021).

# **Graphical Representations**

Graphical representations in software engineering are visual models that depict the structure, behavior, and relationships between different elements in software systems. They provide a means of abstracting complex information and presenting it in a simplified and understandable form(Kang et al., 2013, p. 234). Some examples of graphical representations commonly used in software engineering include:

* Flowcharts: Used to represent the flow of control and data in a system or process.

* UML diagrams: A standardized set of graphical notations for describing object-oriented software systems, including use case diagrams, class diagrams, and sequence diagrams.

* Entity-Relationship diagrams: Used to represent the relationships between data entities in a database.

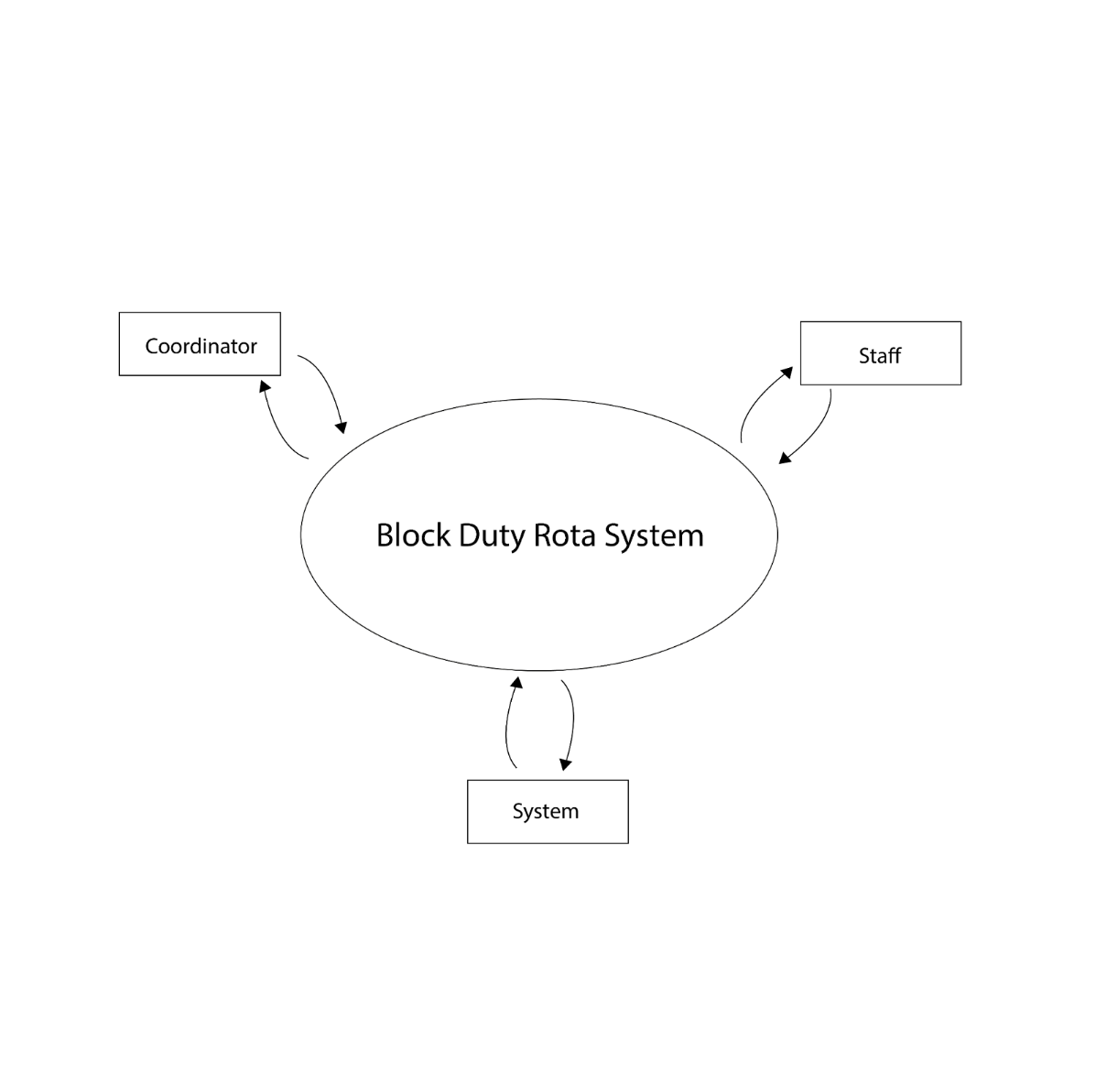
* State transition diagrams: Used to describe the behavior of a system in response to different events and inputs.

* Gantt charts: Used to represent project schedules, showing the timeline of tasks and their dependencies.

By using graphical representations in software engineering, developers can better communicate and collaborate with stakeholders, clarify requirements, and identify potential issues early in the development process.

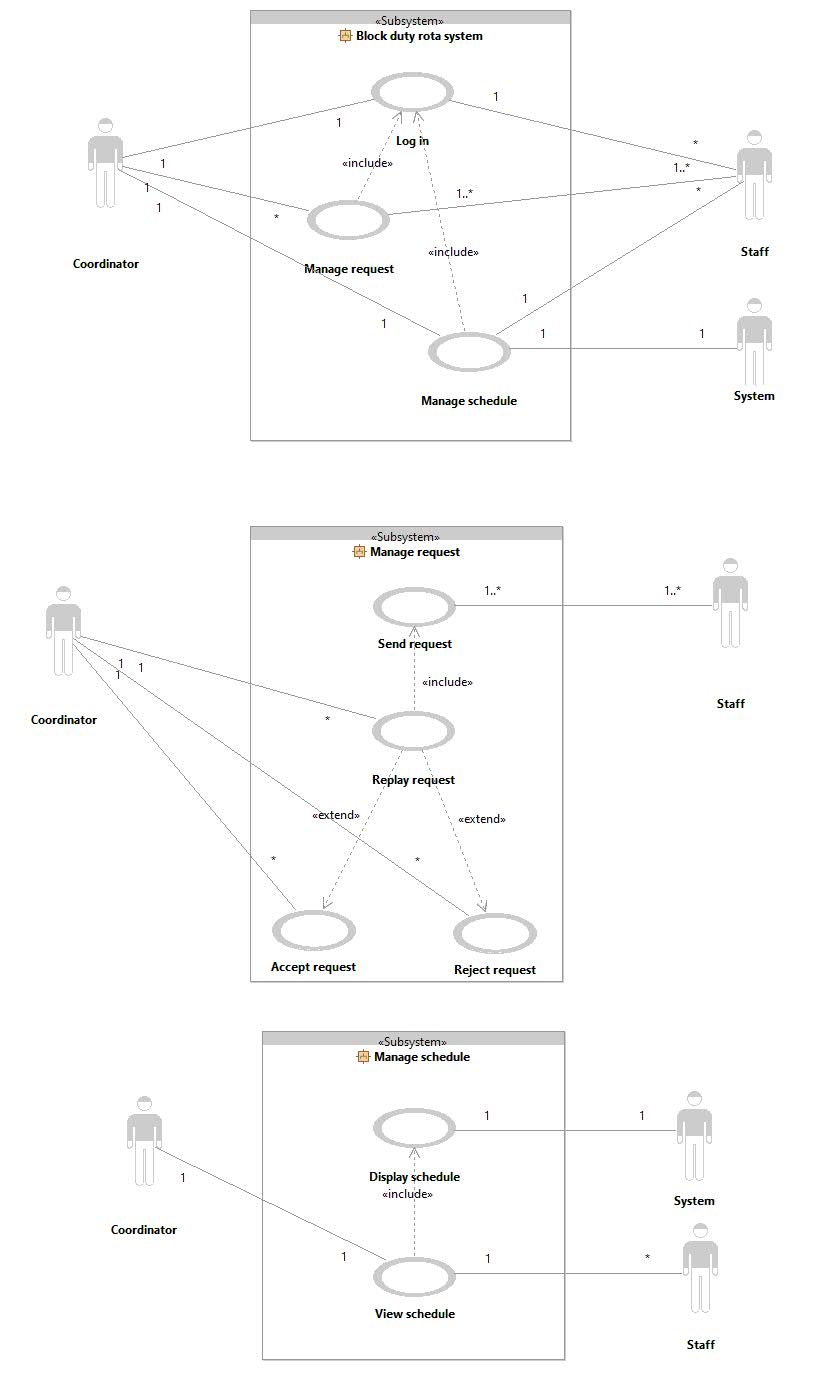
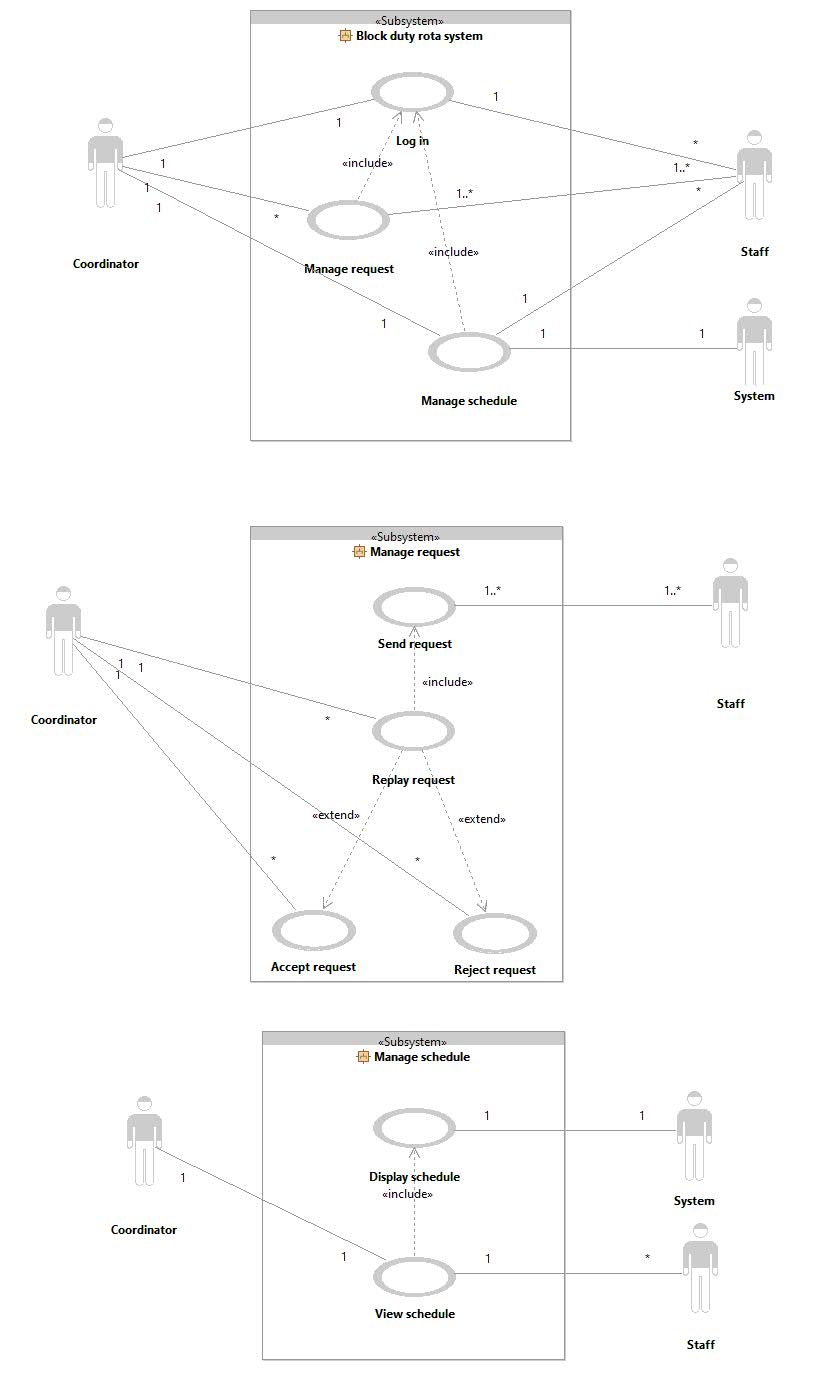
In our project we used a number of UML diagrams to describe our system including class diagram, activity diagram, sequence diagram, use case diagram and object diagram.

# **UML Use Cases Diagram**

Use case diagram Represent the high-level functions of a system from the user’s point of view (Sommerville, 2011). It illustrates the interactions between the users and the system in order to capture the system's requirements.

*Figure 2: use case diagram level 0*

Level 0 of the use case diagram illustrates the users who interact with the system as shown in figure2.

 *Figure 3: use case diagram level 1*

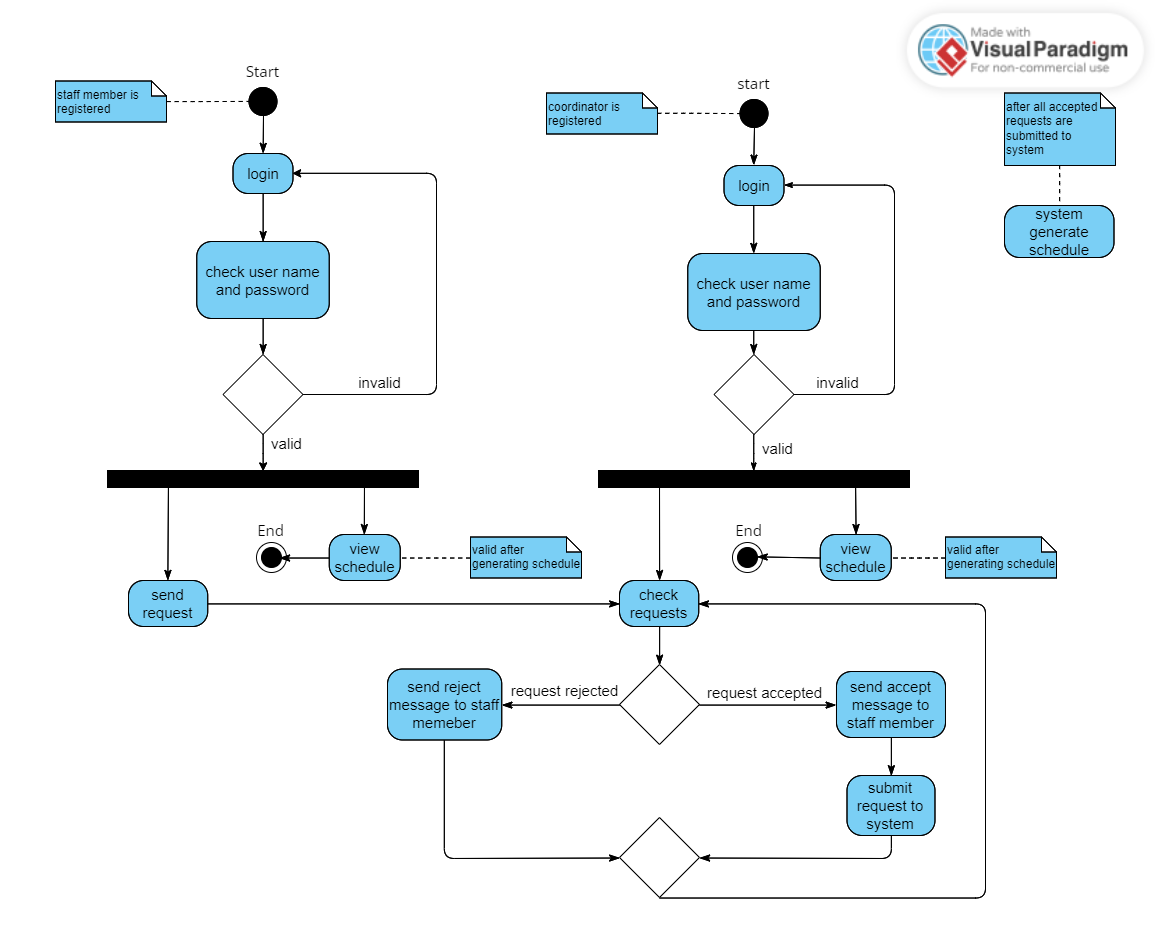
*Figure 4: use case diagram level 2*

Level 1 of the use case diagram describes the use case processes  that are supported by our system and how the users interact with them. As shown in figure3 There are three use cases (log in, manage request and manage schedule).

In Level 2 of the diagram each use case process is deconstructed and redefined into a use case diagram. Also, it shows the interaction between the users and use cases. In figure4 Manage request includes four use cases (send request, replay request, accept request and reject request). Manage schedule has two  use cases (display schedule and view schedule).

# **UML Activity Diagram**

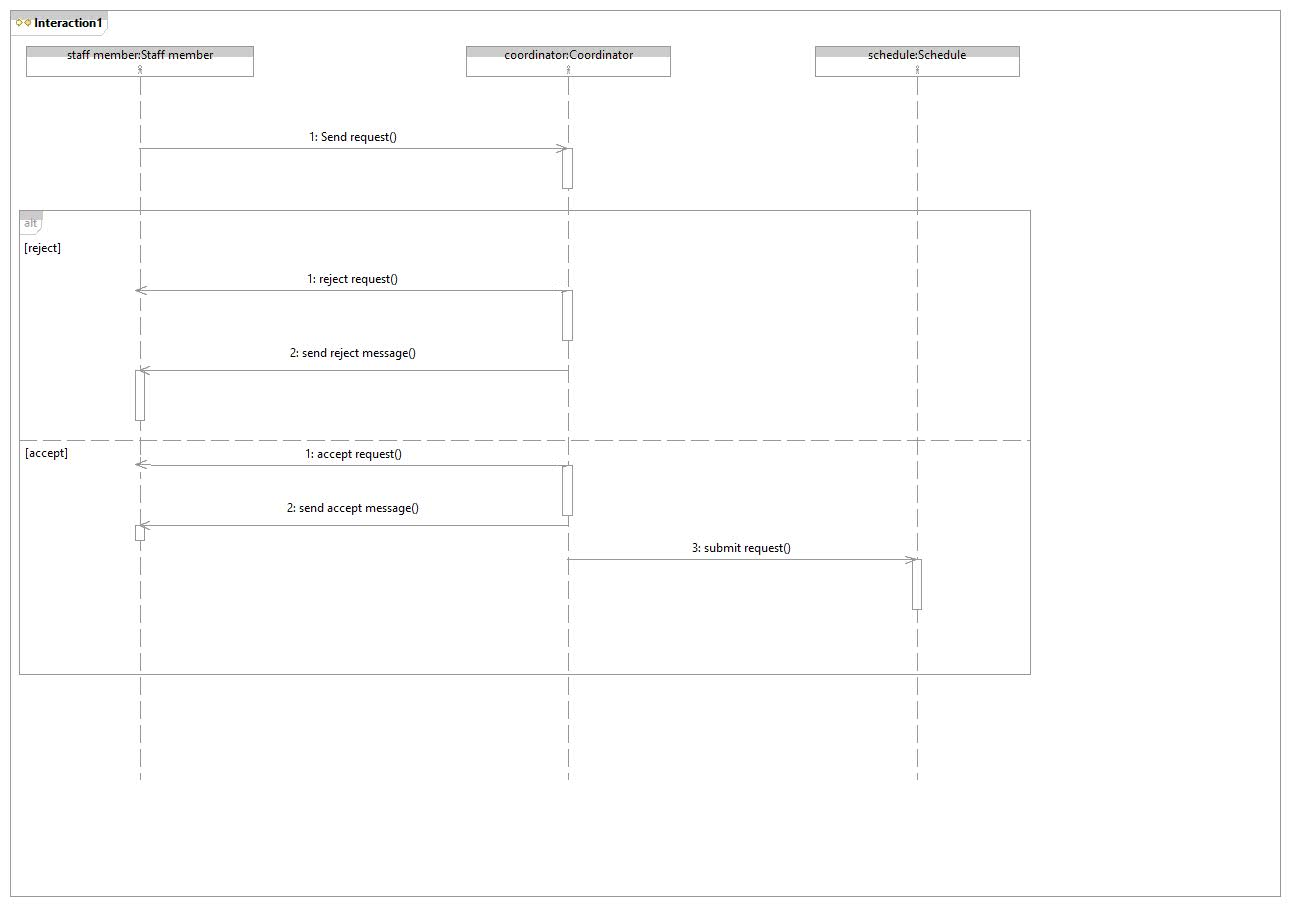
An activity diagram is a diagram which represents the parallel behavior of an operation as a set of actions (Sommerville, 2011). Figure5 represents the activity diagram for our system.



*Figure 5: activity diagram*

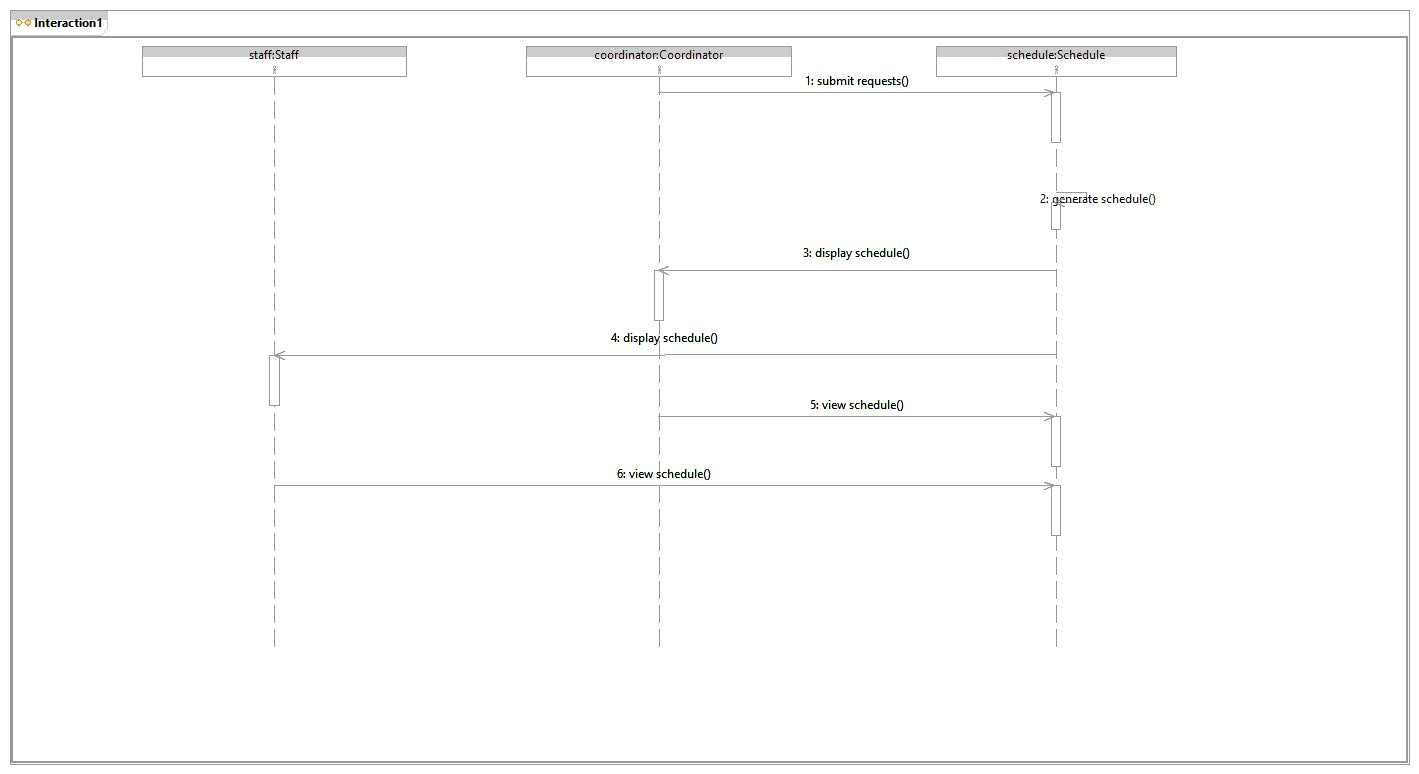
This activity diagram shows the flow of actions in the block duty rota system.  There are two possible starting nodes: one for staff users and the other is for the coordinator. If the user is a staff member, he will get two options after logging in and verifying the username and password entered. The first one is a send request and the second is a view schedule. When choosing a send request the user sends his request to the coordinator who has already logged in to the system. The coordinator also has two options, either check request or view schedule. When choosing a check request could either reject or accept the staff requests. One after another if the request is rejected a rejection message will be sent to the staff member. Similarly, if the request is accepted an accept message will be sent to the staff member and the request is submitted into the system. After all requests are checked, the system will generate the schedule and the staff and coordinator will be able to view the schedule from the view schedule option.

# **UML Sequence Diagram**

A sequence diagram is an interaction diagram  that describes the interactions between groups of objects in a system (Sommerville, 2011).

*Figure 6: sequence diagram 1*

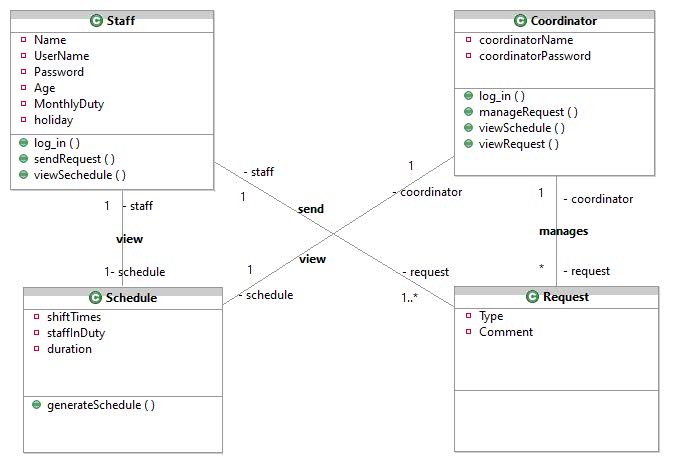
Here in figure6 we have staff members, coordinator and system objects. The staff member will send a request to the coordinator(arrow from The staff to the coordinator) who will either accept or reject the request(two arrows from the coordinator to The staff). In case the coordinator accepts the request an accept message will be sent to the staff member(arrow from the coordinator to The staff). After that, the accepted request will be submitted to the schedule (arrow from the coordinator to The schedule ). On the other hand, if the coordinator rejects the request an rejection message will be sent to the staff member(arrow from the coordinator to The staff).



*Figure 7: sequence diagram 2*

Here in figure7 the coordinator will submit the accepted request to the  schedule(arrow from the coordinator to The schedule ). Then the schedule will be generated (arrow from the schedule to The schedule ). After that ,the schedule will appear for the staff member and to the coordinator (two arrows from the schedule to The staff member  and to the coordinator). Both the coordinator and the staff member will be able to view the schedule whenever they want (arrow from the staff member to the schedule and another arrow from the coordinator  to the schedule).

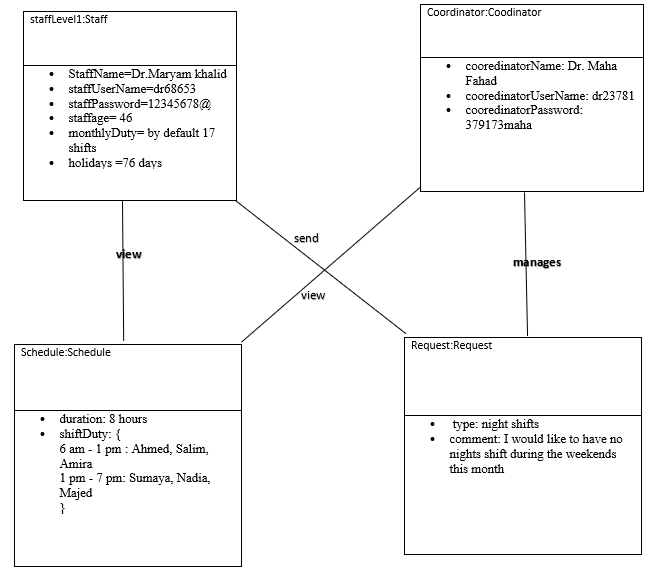
# **UML Class Diagram**

A class represents a group of objects that share a joint structure and behavior.The  class diagram describes a static structure in terms of a system class and its relationships(Sommerville, 2011).

*Figure 8: class diagram*

This Class diagram shows our system classes and the relationships between them. Each class has its own attributes and operations. All the relationships in the diagram are of type simple association. Moreover, each relation has a name and a multiplicity. For instance, as shown in figure (8) there is an association relationship between coordinator class and schedule class called view with a one-to-one multiplicity.

# **UML Object Diagram**

Objects are instances of a class, which encapsulates some specified data and a set of associated operations(Chin and Chanson,1991). The object diagram demonstrates a specific instance of a class diagram at some point of time.

*Figure 9: object diagram*

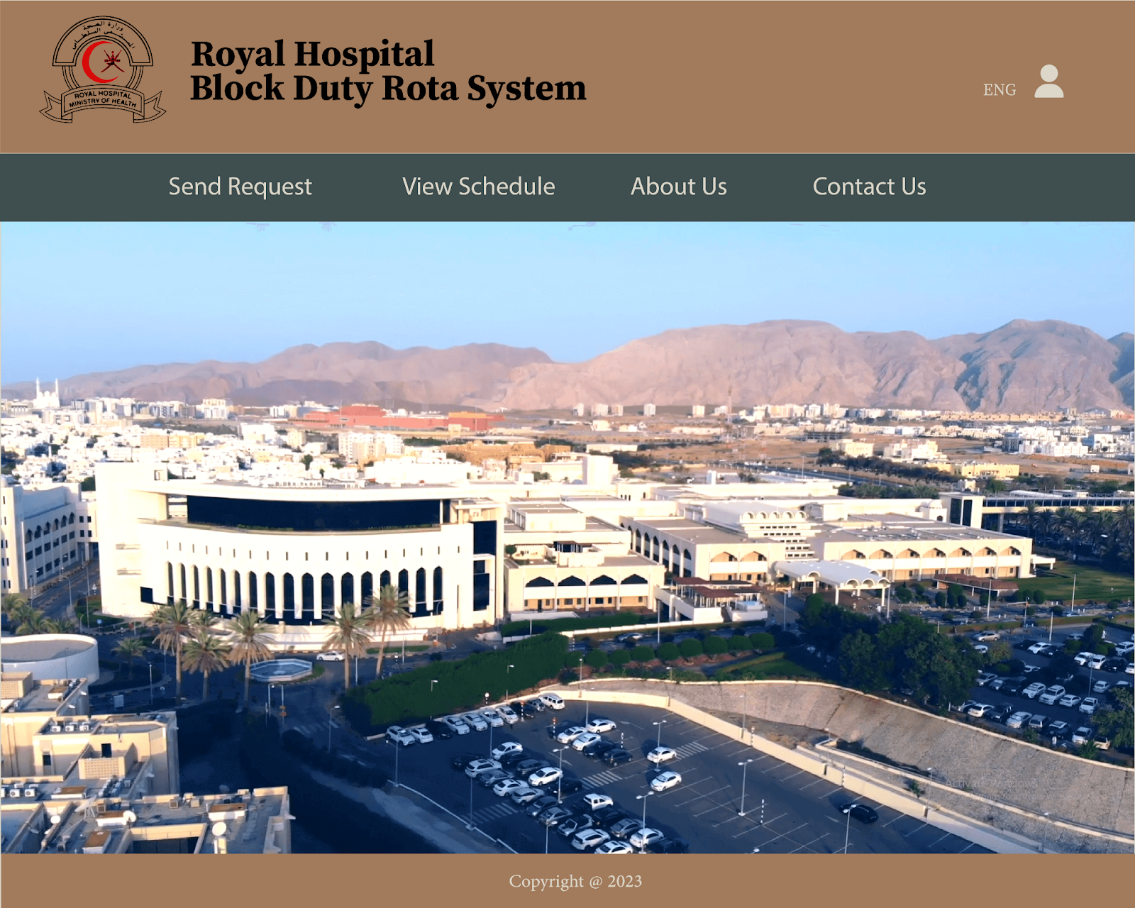
The object diagram represents the objects of each class and the relationship between them. also shows  the value of an attribute. All the relationships in the diagram are types of simple association. In addition,each relationship has a function. for example,there is an association relationship between a staff-level 1 object and a request object called send.

# **Algorithms**

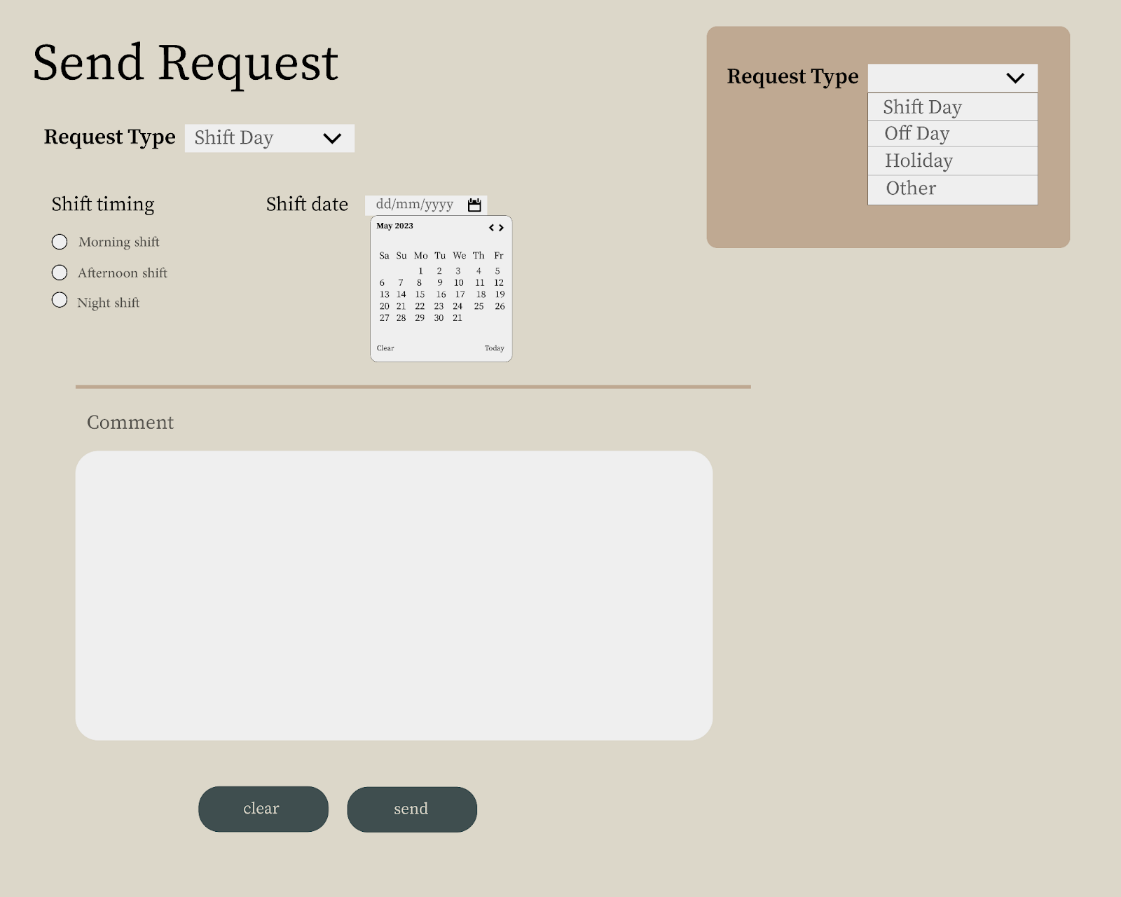
1. The users (staff and coordinator) log in to the system using Username and password, the system checks for the Username and password validity, if one of them wasn’t valid the system will ask them to re-login.
2. Staff will send requests to the coordinator about their shifts timing, night shifts days etc, along with writing the reasons behind that request. Each staff member can either send one or more requests or don’t send.
3. The coordinator checks and manages all the requests by accepting or rejecting them. The system sends back messages to the staff to inform them that the coordinator has either accepted their requests or rejected them.
4. After managing all the requests, the system generates a schedule according to the accepted requests and displays it to the users .
5. The generated schedule can be viewed by the users at any time during that month.

# C:\Users\DELL\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\50943285.tmp**Implementation**

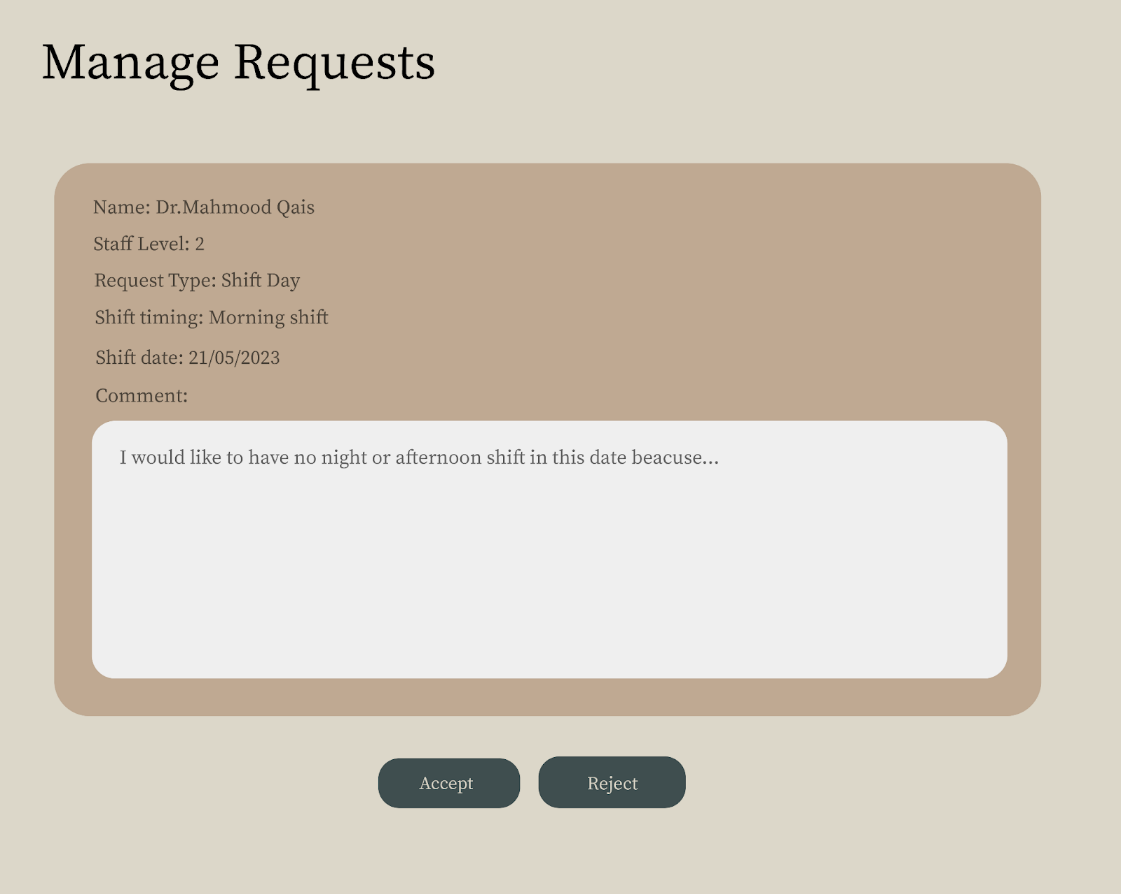
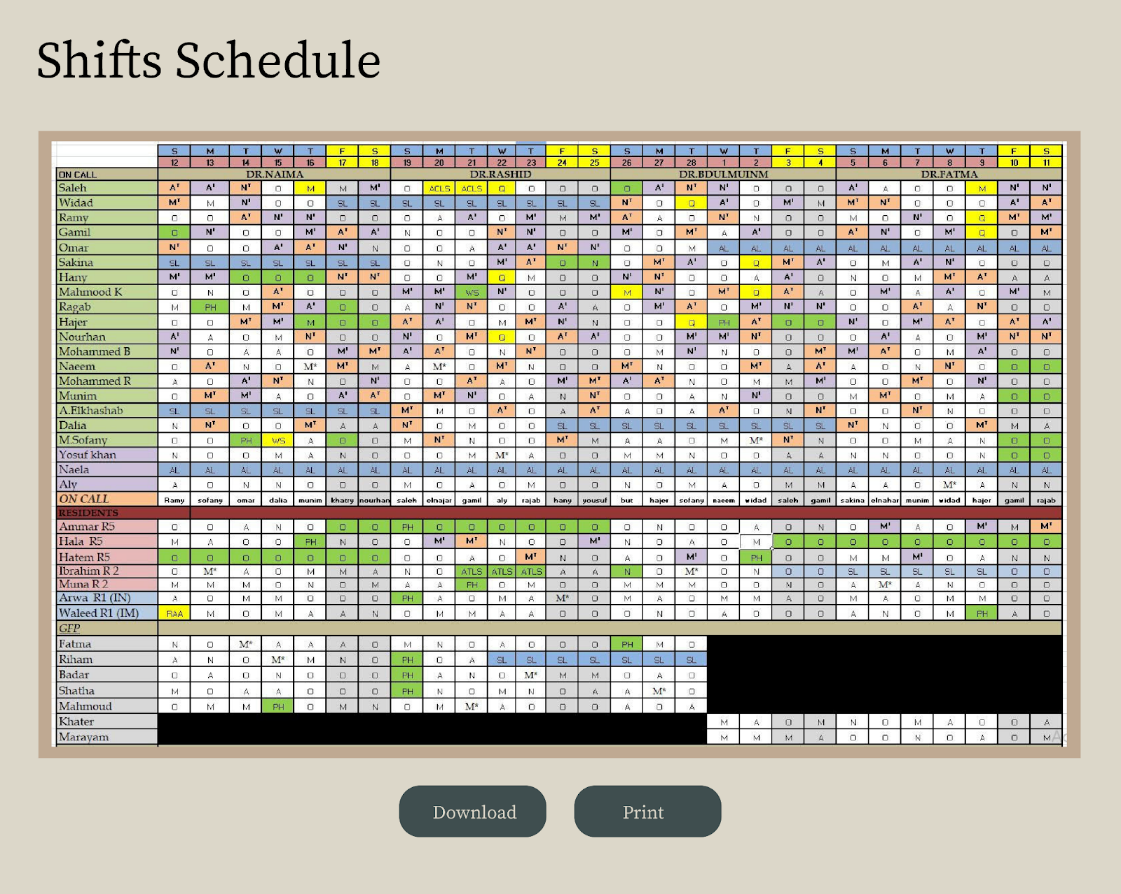
*Figure 10: Home page*

 *Figure 11: Log In*

*Figure 12: Staff interface*

 *Figure 13: Coordinator interface*

*Figure 14: Send Request page*

 *Figure 15: Manage Request page*

*Figure 16: Shifts Schedule* Figure 10 shows the index page of our system, while Figure 11 illustrates the login form, where the users will enter their username and password. Since there are different users of the system each user type has an interface with a specific service. Figures 12 and 13 show the Staff page and Coordinator page respectively.

Figure 14 shows the page where the staff can send a request to the coordinator. First, they choose the request type. Second, the page will display a form with all the needed information for the selected request type. Third, fill out the form and press the send button. Figure 15 shows the page where the coordinator manages all the received requests from the staff, counting on the provided information along with a specified constraint he will either accept or reject the request. After that, they can view, print and download the schedule as shown in Figure 17.

# **Testing Phase**

We used black box testing by partitioning our test cases into categories.

**Category 1: Number of shifts**

1. Is the number of shifts for each staff member < 17?
2. Is the number of shifts for each staff member >17?
3. Is the number of night shifts for each staff member > 5?
4. Is the number of night shifts for each staff member < 5?
5. Are the number of morninf and afternoon shifts devided equally between staff members?
6. Are all shift slots covered?

**Categeroy 2: Shifts duration**

1. Is the duration of each shift > 8 hours?
2. Is the duration of each shift < 8 hours?

**Categeroy 3: Number of holidays**

1. Is the total comulative number of leave days for each staff member this year < 48 (excluding public holidays)?
2. Is the total comulative number of leave days for each staff member this year > 48 (excluding public holidays)?
3. Are there holidays exceeding 6 consecutive weeks?
4. Is the number of public holidays for each staff member > 28 days?
5. Is the number of public holidays for each staff member < 28 days?

**Categeroy 4: Requests**

1. Are all requests recieved by the coordinator?
2. Did all the requests recieve a reply?
3. Are all accepted requests considered in the schedule?

# **Maintenance Phase**

[Software maintenance](https://en.wikipedia.org/wiki/Software_maintenance) is the process of changing, modifying, and updating software to keep up with customer needs. There are 4 types of Maintenance which are: Corrective Software Maintenance, Preventative Software Maintenance, Perfective Software Maintenance and Adaptive Software Maintenance. In the future, we will do changing and updating to our system and that changing includes:

* Enhancements: adding new features or improving existing features to the system.
* Performance optimization: improving the speed, efficiency, and reliability of the System.
* Adapting the software to run on new hardware or software platforms.
* Improving the design and architecture of the software to make it more maintainable and scalable
* Updating documentation, reviewing and testing the system, and implementing to prevent future problems
* Finding and fixing errors and bugs in the software by using the feedback coming from users

# **Future Work**

We must ensure maintaining the system effectively in the future and provide needed upgrades. Also, we can develop the system and make it as a commercial tool so that the users can use it in their institutional systems.

# **Conclusion**

In the end, Scheduling the Block Duty Rotas in hospitals is considered a big complex task laying on top of the departments’ coordinators' arms. Automating the process leads to great impacts on time management, productivity, efficiency, and satisfaction of the employees in the working space. The Block Duty Rota system project has gone through multiple phases to deliver a high-quality product that satisfies the stakeholders. To sum up, First, starting with a project proposal where we described the problem and set the objectives, advantages, and disadvantages. Second, in The Project Management & Literature Review phase, we overviewed the previously published similar systems and illustrated the Tasks and Activity Network Diagram. Third, the Requirements Analysis Phase, where we provided the system’s use cases, scenarios, and functional and non-functional requirements. This phase act as a strong foundation for the next steps, to ensure that the system is delivered on time and within the budget. Fourth, The Project Design Phase provided a clear vision of the project through UML, Sequence, Object, Activity, Class, and Use case diagrams to show how the system will work and the interaction between the different actors. Fifth, In the implementation phase we converted the design specifications into interfaces. Also, we tested the system to ensure it meets the requirements and is ready to be delivered.

# **References**

* A. A. Constantino, E. L. de Melo, D. Landa-Silva and W. Romão, "A heuristic algorithm for nurse scheduling with balanced preference satisfaction," 2011 IEEE Symposium on Computational Intelligence in Scheduling (SCIS), Paris, France, 2011, pp. 39-45, doi: 10.1109/SCIS.2011.5976549.
* Aickelin, U., & Dowsland, K. A. (2004). An indirect genetic algorithm for a nurse-scheduling problem. *Computers & Operations Research, 31*(5), 761–778.
* Al-Mudahka,I. Alhamad,R.(2022) “ON A TIMETABLING PROBLEM IN THE HEALTH CARE SYSTEM” Published by EDP Sciences, ROADEF, SMAI 2022
* Azaiez, M. N., & Al Sharif, S. S. (2005). A 0-1 goal programming model for nurse scheduling. Computers & Operations Research, 32(3), 491–507. doi:10.1016/s0305-0548(03)00249-1
* Beliën, J., Demeulemeester, E., & Cardoen, B. (2006). *Visualizing the Demand for Various Resources as a Function of the Master Surgery Schedule: A Case Study. Journal of Medical Systems,*
* Centeno, M. A., Giachetti, R., Linn, R., & Ismail, A. M. (n.d.). A simulation-ILP based tool for scheduling ER staff. Proceedings of the 2003 International Conference on Machine Learning and Cybernetics (IEEE Cat. No.03EX693).
* Chilman, C. (2022) Take online courses. earn college credit. Research Schools, Degrees & Careers, Study.com | Take Online Courses. Earn College Credit. Research Schools, Degrees & Careers. Study. Available at: https://study.com/learn/lesson/maintenance-phase-sdlc-overview-outcomes.html (Accessed: May 4, 2023).
* Chin, R. S., & Chanson, S. T. (1991). Distributed, object-based programming systems. ACM Computing Surveys, 23(1), 91–124.
* Class diagrams in UML modeling. (n.d.). Class Diagrams in UML Modeling. <https://www.ibm.com/docs/en/rsm/7.5.0?topic=structure-class-diagrams>.
* D’souza, B., Rao, S,S., et al., (2021) "ROTA: A system for automated scheduling of nursing duties in a tertiary teaching hospital in south India", JHealth Informatics Journal, 8(10), 1-12, 2021.
* *El Adoly, A. A., Gheith, M., & Nashat Fors, M. (2018). A new formulation and solution for the nurse scheduling problem: A case study in Egypt. Alexandria Engineering Journal.*
* Glinz, M. (2007). On Non-Functional Requirements. 15th IEEE International Requirements Engineering Conference (RE 2007).
* Gondane,M and Zanwar,D.(2012). Staff Scheduling in Health Care Systems. *IOSR Journal of Mechanical and Civil Engineering (IOSRJMCE) ISSN: 2278-1684 Volume 1, Issue 6 (July-Aug 2012).*
* Gutjahr, W. J., Rauner, M. S., & Part Special Issue: Logistics of Health Care Management. (2007). An aco algorithm for a dynamic regional nurse-scheduling problem in austria. *Computers and Operations Research, 34*(3), 642–666. <https://doi.org/10.1016/j.cor.2005.03.018>
* Hardy, B., & Marshall, C. (2020). Emergency schedule management for COVID-19. British Journal of Anaesthesia.
* Halwai, S. (2021). SDLC Design Phase – Everything You Need to Know.OpenXcell.<https://www.openxcell.com/blog/design-phase-in-sdlc/#:~:text=What%20is%20the%20Design%20Phase,to%20achieve%20the%20project%20result>
* Ibbs, C. W., & Kwak, Y. H. (2000). Assessing Project Management Maturity. *Project Management Journal*, *31*(1), 32–43.
* Indeed Editorial Team. (2023). Project Design in Project Management: Why It’s Important. Indeed Career Guide. <https://www.indeed.com/career-advice/career-development/project-design>
* Halwai, S. (2021). SDLC Design Phase – Everything You Need to Know.OpenXcell.<https://www.openxcell.com/blog/design-phase-in-sdlc/#:~:text=What%20is%20the%20Design%20Phase,to%20achieve%20the%20project%20result>
* Sommerville, L. (2011). SOFTWARE ENGINEERING (9th ed., p. 84). Pearson Education.
* UML Sequence Diagram Tutorial. (n.d.). Lucidchart. <https://www.lucidchart.com/pages/uml-sequence-diagram#:~:text=A%20sequence%20diagram%20is%20a,to%20document%20an%20existing%20process>.
* Indeed Editorial Team. (2023). Project Design in Project Management: Why It’s Important. Indeed Career Guide. <https://www.indeed.com/career-advice/career-development/project-design>
* Kang, K. C., Kim, S., & Kim, S. (2013). Software engineering: Architecture-driven software development. John Wiley & Sons.
* Khristich,S.(2023) “Medical Staff Management and Scheduling Software Development“, published by TATEEDA GLOBAL [https://tateeda.com/blog/medical-staff-management-and-scheduling-software-development 25-1-2023](https://tateeda.com/blog/medical-staff-management-and-scheduling-software-development%2025-1-2023).
* Lad, A. (2022) What is a use case? definition, template, and how to write one, LogRocket Blog. Available at: https://blog.logrocket.com/product-management/what-is-a-use-case-template-how-to-write/#what-is-a-use-case (Accessed: March 20, 2023).
* Lalband,N., D,Kavitha.(2019) “Software Engineering for Smart Healthcare Applications”,International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-8, Issue-6S4, April 2019.
* Ohki, M., Morimoto, A., Miyake, K., & 2006 3rd International IEEE Conference on Intelligent Systems London, UK 2006 Sept. 4 - 2006 Sept. 6. (2006). 2006 3rd international ieee conference intelligent systems. *In Nurse scheduling by using cooperative ga with efficient mutation and mountain-climbing operators* (pp. 164–169). essay, IEEE. <https://doi.org/10.1109/IS.2006.348411>
* Okada, M. (1992). An approach to the generalized nurse scheduling problem—Generation of a declarative program to represent institution-specific knowledge. Computers and Biomedical Research, 25(5), 417–434. doi:10.1016/0010-4809(92)90002-r
* Poissonnet, C. M., & Veron, M. (2000). *Health effects of work schedules in healthcare professions. Journal of Clinical Nursing, 9(1), 13–23.*
* PRYCE, J., ALBERTSEN, K., & NIELSEN, K. (2006). Evaluation of an open-rota system in a Danish psychiatric hospital: a mechanism for improving job satisfaction and work-life balance. Journal of Nursing Management, 14(4), 282–288.
* Randhawa, S. U., & Sitompul, D. (1993). A heuristic-based computerized nurse scheduling system. Computers & Operations, Research,20(8),837–844.doi:10.1016/0305-0548(93)90105-r
* Roznovsky,A.(2023) “Hospital Management Systems: Features,
* Requirements and Benefits” Published by  Light IT Global <https://light-it.net/blog/how-to-develop-a-quality-management-system-in-a-hospital/> .
* Sommerville, I. (2011) Software Engineering. 9th Edition, Pearson.
* Squires,M.,et al.,(2021) “A novel genetic algorithm based system for the scheduling of medical treatments”, published by Elsevier 12 December 2021
* Teahan. (1998). *Implementation of a self-scheduling system: a solution to more than just schedules! Journal of Nursing Management, 6(6), 361–368.*
* Topaloglu, S. (2009). A shift scheduling model for employees with different seniority levels and an application in healthcare. European Journal of Operational Research, 198(3), 943–957.
* Tucci, L. (2023, January 17). *What is risk management and why is it important?* Security. Retrieved February 28, 2023, from <https://www.techtarget.com/searchsecurity/definition/What-is-risk-management-and-why-is-it-important>
* Uhde, A., Laschke, M., & Hassenzahl, M. (2021). Design and appropriation of computer-supported self-scheduling practices in healthcare shift work. *Proceedings of the Acm on Human-Computer Interaction, 5*(Cscw1), 1–26. <https://doi.org/10.1145/3449219>
* UML Sequence Diagram Tutorial. (n.d.). Lucidchart. <https://www.lucidchart.com/pages/uml-sequence-diagram#:~:text=A%20sequence%20diagram%20is%20a,to%20document%20an%20existing%20process>.
* Wright, C., McCartt, P., Raines, D., & Oermann, M. H. (2017). Implementation and Evaluation of Self-Scheduling in a Hospital System. Journal for Nurses in Professional Development, 31(1), 19–24.
* W.T(2022) *What is Proposal Writing?* (n.d.). Technical Writer HQ. <https://technicalwriterhq.com/writing/proposal-writing/>
* Zanda, S., Zuddas, P., & Seatzu, C. (2018). Long term nurse scheduling via a decision support system based on linear integer programming: a case study at the university hospital in cagliari. *Computers & Industrial Engineering, 126*, 337–347. <https://doi.org/10.1016/j.cie.2018.09.027>

# 