**School of Computing**

**CA326 Year 3 Project Functional Specification**

**Project Title**: Smart Rostering System

**Student 1 Name**: Xiaoyu Lyu  **ID Number**: 20100078

**Student 2 Name**: Yutong Wu **ID Number**: 20307383

**Staff Member Consulted**: Liting Zhou

**Document Finished On**: 29 November 2022

**Table of contents**

1. Introduction 1

1.1 Overview 1

1.2 Business Context 1

1.3 Glossary 1

2. General Description 1

2.1 Product / System Functions 1

2.2 User Characteristics and Objectives 2

2.3 Operational Scenarios 3

2.4 Constraints 6

3. Functional Requirements 8

4. System Architecture 12

5. High-Level Design 12

6. Preliminary Schedule 14

7. Appendices 15

**1. Introduction**

**1.1 Overview**  
Smart Rostering System will be a web application designed to solve the severe problem of the nursing schedule system. Due to the circumstances of the short hands of a nursing home, Smart Rostering System will help to provide a frame, or a fair prediction schedule solution based on the status of the medical staff who signed into the system.

**Functions**:

1. Smart Rostering System should generate a usable schedule draft every two weeks. It can highlight the day off, special events, rush hour and free slots, it also can be amended manually.
2. The night shift rotation reminder can be set up on this system.
3. The system should handle errors from user input or database.
4. All staff names, roles, and contracted hours should be displayed on the screen and allow allocations of specific areas or floors.
5. All staff can request a day off, annual leave, sick leave, or maternity leave on the system for the manager’s approval and all these data will be recorded, the system also should show the remaining holiday hours.
6. Who has approved day off, annual leave, sick leave or maternity leave should appear on the roster automatically.
7. The duty will be notified to staff by email.
8. The system should store the personal information of all staff as below:

* Contact number
* Home address
* Email address
* Employment status i.e. full-time, part-time, relief, weekends only, nights only.
* Contracted hours

**1.2 Business Context**

To be defined.

**1.3 Glossary**

To be defined.

**2. General Description**

**2.1 Product / System Functions**  
Smart Rostering System allows you to quickly build your duty/shift requirements and then presents staff members who meet those requirements for the day/time you require.

* Build your duty/shift requirements easily
* Specify how many staff of each area/floor require
* Quickly assign staff to areas/floors
* Work, Leave, Events and Overtime are already integrated
* Staff receive instant updates to their emails
* Days can quickly be duplicated up to 31 days, allowing you to save more time.

**Spot any shortages in staff**:

Smart Rostering System will show you any shortages you have throughout the day, allowing you to draft in extra support as and when you need it.

**The staff know, at any time:**

There is no need to post task sheets in the office, or memos to staff. Smart Rostering System shows staff members their assigned shifts on their dashboard when they log in to their accounts.

**Teams like working together**:

Teams can see plans one week in advance, so they are not constantly asking management what's happening.

**Map shifts and set work schedules**:

Get the right mix of staff day in, and day out with our agile shift planning technology.

* Quickly build your rota and fill in any gaps
* Set staffing levels for each shift, with the right seniority and qualifications
* See holiday bookings and TOIL to identify who’s available
* Flexibly change your schedule to accommodate last-minute demands

**Set shifts/duties to be fulfilled**:

Ensure your staff fulfil their duties.

* Build requirements quickly
* Specify how many staff of each area/floor you require
* Integrated Work, Leave, Events and Overtime for clearer planning

**Make public holidays a reason for celebration**:

Set separate rules for processing public holiday requirements and company events.

* Manage public holidays by country or region
* Add company-wide events and training days
* Set staffing requirements, including mandatory attendance
* Decide whether days are deducted from a staff holiday entitlement

**2.2 User Characteristics and Objectives**  
This section describes the features of the user community, including their expected expertise with software systems and the application domain. Explain the objectives and requirements for the system from the user's perspective. It includes a "wish list" of desirable characteristics, along with more feasible solutions that are in line with the business objectives.

1. Staff information is accessible on the side of rostering page for managers regarding contracted hours, and allocated areas.
2. Staffing ratio requirement on each floor.
3. Night rotation every 3 or 6 months.
4. Alter when staff are rostered consecutively for over 4 days or nights, between weekends to weekdays, alter errors such as roster day shift on Monday straight after Sunday night shift.
5. Display staff on leave on the roster automatically.
6. Space for training schedules.
7. Space for agency cover.
8. Space for extra shifts for staff to grab.
9. Auto-generate report for extra hours for staff.
10. Information rebalances of annual leave.
11. Able to add allocation beside the staff’s name and shift hours.

**2.3 Operational Scenarios**

This section describes a set of scenarios that illustrate, from the user's perspective, what will be experienced when utilizing the system under various situations.

|  |  |  |
| --- | --- | --- |
| **USE CASE 1** | Create New Roster | |
| **Goal in Context** | A new roster is created | |
| **Scope & Level** | System, Sub-function | |
| **Preconditions** | Internet is connected, manager logged in the account, manager selects correct dates | |
| **Success End Condition** | Client displays the new roster. | |
| **Failed End Condition** | Client doesn’t display the new roster. | |
| **Primary,**  **Secondary Actors** | Manager  Client, System | |
| **Trigger** | Manager clicks on the “Make New Roster” button | |
| **DESCRIPTION** | **Step** | **Action** |
|  | 1 | Manager opens a web browser. |
|  | 2 | Manager enters the Smart Rostering System. |
|  | 3 | Client displays the login page. |
|  | 4 | Manager logs in to the account. |
|  | 5 | Manager clicks on the “Make New Roster” button |
|  | 6 | Manager selects the date to start with and the date to end with. |
|  | 7 | System generates a new roster. |
|  | 8 | Client displays the new roster. |
| **EXTENSIONS** | **Step** | **Branching Action** |
|  | 4a | Manager enters wrong username or password: an error message “Wrong username or password, please check your username or password and try again.” is shown, action failed. |
|  | 6a | Selected period is too short: an error message “Period has to be two weeks, please try again.” is shown, action failed. |
| **VARIATIONS** |  | **Branching Action** |
|  | 1 | Manager clicks on the “Cancel” button when the roster is generating, action cancelled. |

|  |  |  |
| --- | --- | --- |
| **USE CASE 2** | Approve the leave request from staff | |
| **Goal in Context** | Leave request is approved | |
| **Scope & Level** | System, Sub-function | |
| **Preconditions** | Internet is connected, staff requested leave on the system, manager logged in the account | |
| **Success End Condition** | Leave request is approved | |
| **Failed End Condition** | Leave request is not approved | |
| **Primary,**  **Secondary Actors** | Manager  Client | |
| **Trigger** | Manager clicks on the “Holidays Approval” section | |
| **DESCRIPTION** | **Step** | **Action** |
|  | 1 | Manager opens a web browser. |
|  | 2 | Manager enters the Smart Rostering System. |
|  | 3 | Client displays the login page. |
|  | 4 | Manager logs in to the account. |
|  | 5 | Manager clicks on the “Holidays Approval” section. |
|  | 6 | Client displays a list of staff who request leave and the status of those requests. |
|  | 7 | Manager clicks on one of the staff listed. |
|  | 8 | Client opens a new page to display a calendar with the highlighted days that this staff is requesting and the leaves already approved in this role. |
|  | 9 | Manager clicks on the “Approve” button. |
|  | 10 | Status of this request is updated to “Approved”. |
|  | 11 | Leave request is approved. |
| **EXTENSIONS** | **Step** | **Branching Action** |
|  | 4a | Manager enters wrong username or password: an error message “Wrong username or password, please check your username or password and try again.” is shown, action failed. |
| **VARIATIONS** |  | **Branching Action** |
|  | 1 | Manager closes the calendar page, and the status of this request is updated to “Viewed”. |

|  |  |  |
| --- | --- | --- |
| **USE CASE 3** | View Roster | |
| **Goal in Context** | Roster is shown | |
| **Scope & Level** | System, Sub-function | |
| **Preconditions** | Internet is connected, staff logs in to the account, roster is published by manager | |
| **Success End Condition** | Roster is shown | |
| **Failed End Condition** | Roster is not shown | |
| **Primary,**  **Secondary Actors** | Staff  Client | |
| **Trigger** | Staff logs in to the account | |
| **DESCRIPTION** | **Step** | **Action** |
|  | 1 | Staff opens a web browser. |
|  | 2 | Staff enters the system. |
|  | 3 | Client displays the login page. |
|  | 4 | Staff logs in to the account. |
|  | 5 | Client redirects to the dashboard automatically. |
|  | 6 | Roster is shown. |
| **EXTENSIONS** | **Step** | **Branching Action** |
|  | 4a | Staff enters wrong username or password: an error message “Wrong username or password, please check your username or password and try again.” is shown, action failed. |
| **VARIATIONS** |  | **Branching Action** |
|  | 1 | Staff is new, there’s no existing shift on the roster, dashboard is blank. |

**2.4 Constraints**  
This section lists general constraints placed upon the design team.

1. The system should use a user-friendly UI design and follow the UI design guidelines. The UI should be convenient, easy to understand and memorised.
2. The system should use a solid database. Use data analysis techniques such as collecting, cleaning, storing, and so on.
3. Smart Rostering System should work on both Mac OS and Windows.
4. The layout of the roster should be as below:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Nursing | | | | | | | | |
| Role | Name | Mon DD/MM | Tue DD/MM | Wed DD/MM | Thu DD/MM | Fri DD/MM | Sat DD/MM | Sun DD/MM |
| DON |  | Shift | Shift | Shift | Shift | Shift | Shift | Shift |
| ADON |  | Shift | Shift | Shift | Shift | Shift | Shift | Shift |
| CNM3 |  | Shift  Area | Shift  Area | Shift  Area | Shift  Area | Shift  Area | Shift  Area | Shift  Area |
| CNM2 |  | Shift  Area | Shift  Area | Shift  Area | Shift  Area | Shift  Area | Shift  Area | Shift  Area |
| CNM1 |  | Shift  Area | Shift  Area | Shift  Area | Shift  Area | Shift  Area | Shift  Area | Shift  Area |
| Nurses |  | Shift  Area | Shift  Area | Shift  Area | Shift  Area | Shift  Area | Shift  Area | Shift  Area |
| Night Nurses |  | Shift  Area | Shift  Area | Shift  Area | Shift  Area | Shift  Area | Shift  Area | Shift  Area |
| HCA | | | | | | | | |
| Role | Name | Mon DD/MM | Tue DD/MM | Wed DD/MM | Thu DD/MM | Fri DD/MM | Sat DD/MM | Sun DD/MM |
| Team leaders |  | Shift  Area | Shift  Area | Shift  Area | Shift  Area | Shift  Area | Shift  Area | Shift  Area |
| HCAs |  | Shift  Area | Shift  Area | Shift  Area | Shift  Area | Shift  Area | Shift  Area | Shift  Area |
| Night HCAs |  | Shift  Area | Shift  Area | Shift  Area | Shift  Area | Shift  Area | Shift  Area | Shift  Area |
| Activities |  | Shift | Shift | Shift | Shift | Shift | Shift | Shift |
| Physio |  | Shift | Shift | Shift | Shift | Shift | Shift | Shift |
| Ancillary | | | | | | | | |
| Role | Name | Mon DD/MM | Tue DD/MM | Wed DD/MM | Thu DD/MM | Fri DD/MM | Sat DD/MM | Sun DD/MM |
| Admin |  | Shift | Shift | Shift | Shift | Shift | Shift | Shift |
| Finance |  | Shift | Shift | Shift | Shift | Shift | Shift | Shift |
| HR |  | Shift | Shift | Shift | Shift | Shift | Shift | Shift |
| Reception |  | Shift | Shift | Shift | Shift | Shift | Shift | Shift |
| Maintenance |  | Shift | Shift | Shift | Shift | Shift | Shift | Shift |
| Catering | | | | | | | | |
| Role | Name | Mon DD/MM | Tue DD/MM | Wed DD/MM | Thu DD/MM | Fri DD/MM | Sat DD/MM | Sun DD/MM |
| Head chef |  | Shift | Shift | Shift | Shift | Shift | Shift | Shift |
| Chefs |  | Shift | Shift | Shift | Shift | Shift | Shift | Shift |
| Kitchen Assistants |  | Shift  Area | Shift  Area | Shift  Area | Shift  Area | Shift  Area | Shift  Area | Shift  Area |
| Housekeeping | | | | | | | | |
| Role | Name | Mon DD/MM | Tue DD/MM | Wed DD/MM | Thu DD/MM | Fri DD/MM | Sat DD/MM | Sun DD/MM |
| Supervisor |  | Shift  Area | Shift  Area | Shift  Area | Shift  Area | Shift  Area | Shift  Area | Shift  Area |
| Cleaners |  | Shift  Area | Shift  Area | Shift  Area | Shift  Area | Shift  Area | Shift  Area | Shift  Area |
| Laundry Assistant |  | Shift | Shift | Shift | Shift | Shift | Shift | Shift |

**3. Functional Requirements**

This section describes the possible effects of the system, in other words, what the system must accomplish.

**Requirement: User Registration**

|  |  |
| --- | --- |
| ID: | User Registration |
| Title: | User Registration |
| Scope: | To be able to create a registration function for new users to log into. |
| Description: | This is an easy-to-use system at the start of the website which makes it easy for new users to join. The system needs to take in the input and store it. |
| Pre-conditions: | * System must be live * System must be able to take in information. * System must be connected to a database to store it. |
| Activation: | User registration starts when a new employee asks to join the application |
| Post-conditions: | User registration only functions when a new member wants to join the application. |
| Termination: | Once the new user adds all of their info, the system will no longer see this user as a new user, and they are free to use all the functions within the system. |
| Success: | If successful, the new user will be able to register |
| Priority: | High |

**Requirement: Management Controls**

|  |  |
| --- | --- |
| ID: | Management Controls |
| Title: | Management Controls |
| Scope: | To be able to create pages to which only the employers have access. |
| Description: | There will be pages to which only the employer can have access. These pages include the new roster page and the requests for time off and shift change accept or deny page. These need to be only accessible by certain users otherwise everyone could choose their own rosters. |
| Pre-conditions: | * System must be live * System must be able to take in information. * System must be connected to a database to store it. * The user is logged on. * System must be able to differentiate between the type of users accessing the rosters |
| Activation: | Activation should be automatic when the user creates a registration account. At registration, the system should be giving the users in question certain capabilities. |
| Post-conditions: | This function is in use when the employer is logged on. |
| Termination: | This function will terminate when the employer logs out. |
| Success: | If successful, the employer should be the only person who has access to the restricted pages. |
| Priority: | High |

**Requirement: Registered Staff systems**

|  |  |
| --- | --- |
| ID: | Registered Staff systems |
| Title: | Registered Staff |
| Scope: | To be able to create a system that is only available to people that are part of that organisation or business. |
| Description: | The system should take new users looking to register and during this process be able to assign them to the place of employment in which they fit. The system should also block them from accessing any other streams in the progress. |
| Pre-conditions: | * System must be live * System must be able to take in information. * System must be connected to a database to store it. * The stream in which the user needs to go into must be a business that uses the application. |
| Activation: | This will come into activation when the users are registering with the system |
| Post-conditions: | This system will be in constant use once the streams begin to make sure nobody else has access to another stream. |
| Termination: | This does not terminate while the application is still live as it is a security function and needs to make sure that it cannot be breached easily. |
| Success: | If successful only the people in that business will have access to the stream that they are in. |
| Priority: | High |

**Requirement: Data requirements**

|  |  |
| --- | --- |
| Back-Up & Recover | The data should have a backup and recovery plan for if ever anything goes wrong. |
| Security Requirement | The data should be safely secured to keep hackers away from it. |
| Data Reusability | The data in the system that is saved in the database should be reusable. |
| Data Integrity | The integrity of the data should stay intact. |
| Data Reliability | The data should be reliable and safe without the fear of it being lost or tampered with. |

**Requirement: User requirements**

|  |  |
| --- | --- |
| Log In/Register | Once a company is set up in the system a new user should be able to just sign up and use the system straight away. It should also be clear and easy for returning users to be able to log in and see the roster or chat with other users from your job. This is a major factor as if it is not easy to log in the system will not be used. Clients need something to be easy to use. |
| View | Clients should be able to clearly read what is on the UI and be able to decipher what is happening on the page. |
| Accessibility | As it is a web application the system should be able to hold a lot of foot traffic at the one time because if it is not able to handle lots of people being on the page at the onetime then it won’t be a very good website. |
| Security | With login details and information being stored about them in a database they need to be confident in knowing that their information is safely stored. |

**Requirement: Environmental requirements**

|  |  |
| --- | --- |
| Portability | From an environmental perspective the application should be portable to limit any unneeded machines. This is also in use in this application as it is all cloud-based. |
| Reusability | The application and all the information inside of the application should be reusable to not make internet clutter or dirt. |

**Requirement: Usability requirements**

|  |  |
| --- | --- |
| Performance/Response Time | The application should have good performance and response time as a slow response time could lead to less foot traffic. |
| Easy to Navigate | The application should be easy to navigate |
| Availability requirement | The application should be available quickly and should be ready to react to commands. |

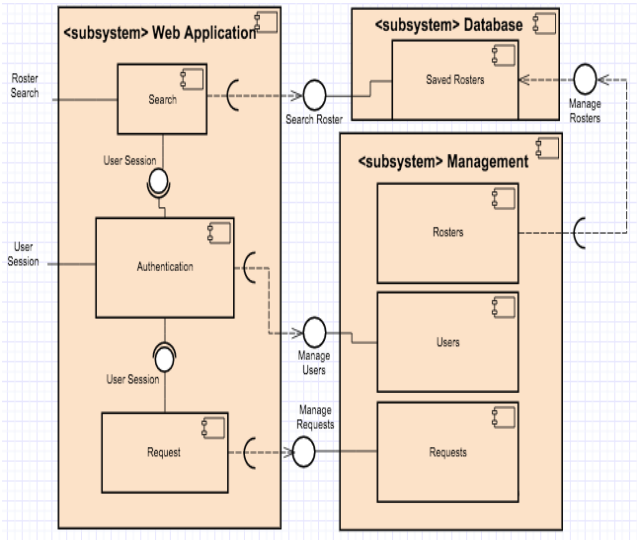
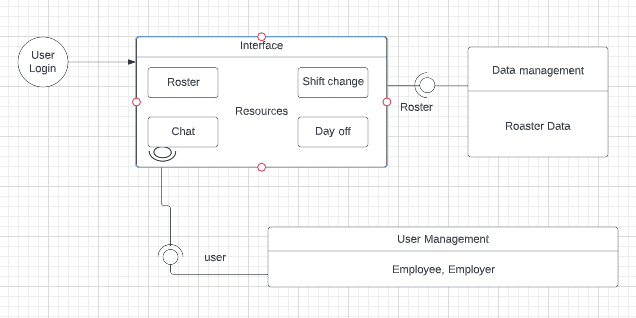


Fig. 1. Component Diagram

**4. System Architecture**

This section describes a high-level overview of the anticipated system architecture showing the distribution functions across (potential) system modules.

Fig. 2. Structure Diagram

**5. High-Level Design**

This section sets out the high-level design of the system. It includes the system model showing the relationship between system components and the system and its environment.

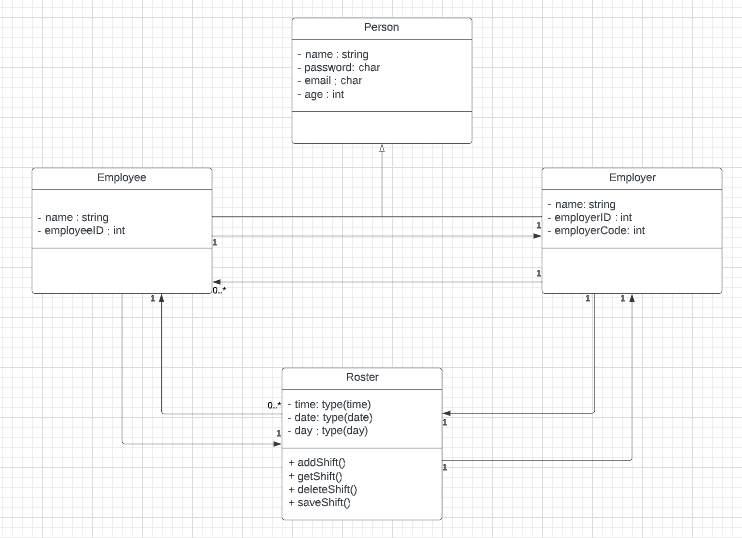


Fig. 3. Class Diagram

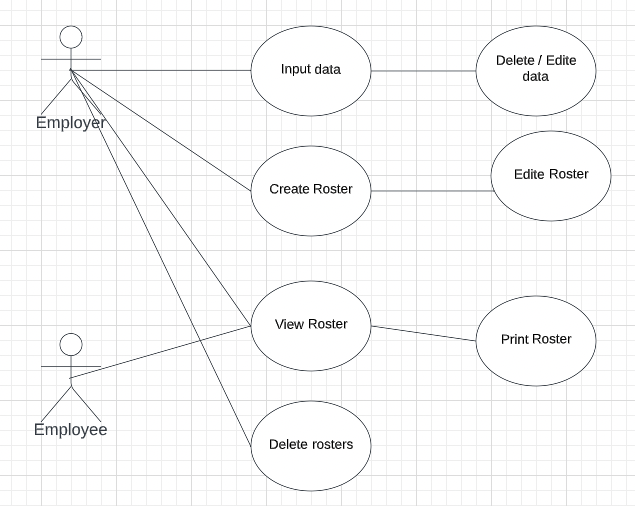
****

Fig. 4. User Case Diagram

**6. Preliminary Schedule**

This section provides an initial version of the project plan, including the major tasks to be accomplished, their interdependencies, and their tentative start/stop dates. The plan also includes information on hardware and software resource requirements.

**Hardware Requirements**:

PC/Mac

**Software Requirements**:

1. Mac OS/Windows 10
2. Google Chrome/Firefox/Microsoft Edge
3. VSCode
4. MySQL
5. Postman
6. Figma

Diagram

Description automatically generated

Fig. 5. PERT Chart of Project Plan

Timeline

Description automatically generated

Fig. 6. GANTT Chart of Project Plan

**7. Appendices**

To be defined.