Software Requirements Specification

for

DUTY NORMALIZER

Version 1.3

Prepared by

Team Number: 02

Aswin Sreekumar	**B200737CS**
Rithika Kathirvel	**B200055CS**
Navaneeth Shanavasan	**B200053CS**
Anand C L	**B200763CS**
Aditya Premjit	**B200703CS**

Project Manager: Sreekala S, Athira A. B.

Project Client: Sreekala S

Course: CS4096D Software Engineering Laboratory

Date: 23-03-2023

Contents

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
Draft 3	Aswin Sreekumar Rithika Kathirvel Navaneeth Shanavasan Anand C L Aditya Premjit	Revision made based on feedback received from reviewing team and client on 21-03-2022.	23-03-2022
Draft 2	Aswin Sreekumar Rithika Kathirvel Navaneeth Shanavasan Anand C L Aditya Premjit	Revision made based on project manager feedback received on 16-03-2022.	17-03-2022

1 Introduction

The team aims to build a duty normalizer mobile application that automates the task of allocating duties to students and faculties. The section of the document contains the document purpose, product scope, intended audiences and document overview. It also briefs the definitions and acronyms used, document conventions followed references and acknowledgment.

1.1 Document Purpose

The purpose of this document is to provide a detailed description of the requirements for the Duty Normalizer mobile application. The latest version of this document (Version 1.2) contains the application's description, functional requirements, and non-functional requirements. This document also provides a detailed use case analysis with a use case diagram.

1.2 Product Scope

The aim of this system is to streamline the process of allocation of duty to M.Tech/ Ph.D. Scholars and Adhocs. Additionally, this system also aims to facilitate the Administrator to generate reports on a particular task in order to view the history of task allocation. In the event of unavailability of a particular student or faculty, this application also allows reallocation of duty and updation of work hours accordingly. The duty is reassigned in a normalized sense, along with the updation of work hours accordingly. The administrator also has the ability to add people to a particular group individually or as a batch through CSV files and delete people from each group individually or through a common feature of their group. The administrator can delete a duty as they deem necessary. The administrator also has the provision to temporarily block a person from being allocated any duty as they deem appropriate.

Apart from this, the application also resets every student and faculty's work-hours to zero at the start of every academic year. We expect this application to be actively integrated into the day-to-day working of NITC exam duty and various other duty allocations and ensure a smooth and efficient operation.

1.3 Intended Audience and Document Overview

The document is intended for the user, the Administrator, who is the personnel that manages the assignment of duties and regulates work-hours, developers, the project manager, and the project client.

The next chapter, the overall description section of this document, gives an overview of the functionality of the application. It describes the informal requirements and is used to establish a context for the technical requirements specification in the next chapter.

The third chapter, the specific requirements section of this document, is written primarily for the developers and describes the details of the functionality of the product in technical terms. Both

sections of the document describe the same software product in its entirety but are intended for different audiences and thus use different languages.

1.4 Definitions, Acronyms and Abbreviations

Serial No.	Abbreviation/Acronym	Definition
1	NITC	National Institute of Technology Calicut
2	SDLC	Software development Life Cycle

1.5 Document Conventions

This document follows the IEEE formatting requirements. Use Arial font size 11 throughout the document for text and Times New Roman for headings.

1.6 References and Acknowledgments

IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.

2 Overall Description

2.1 Product Overview

The primary objective of the Duty normalizer is to seamlessly regulate the allocation of duty to various groups of students and faculties of NITC. As of now, the process of allocating Students and Faculties to duties is done manually, which includes a tedious log book and a lot of physical records to be written and kept for future reference. This application is a self-contained product. With the requirement of allocating duties in a uniformly distributed sense, it becomes crucial to maintain a record of each person's work-hours. The retrieval of the duty reports and reallocation of duty in case of unavailability are features of high importance.

The application enables the administrators to register students and faculties to a particular group individually or as a batch through spreadsheets along with the required details. Initially, all newly added people have their working hours set to zero. The administrator can assign duty by entering duty details along with the number of people required from each category. Based on work-hours,

duty is allocated in a normalized sense and work-hours are updated accordingly. The application allows the Administrator to reallocate a particular duty in case of the unavailability of a person. The administrator must enter the details of the duty to be reallocated and select the unavailable members. The Application will assign the duty to the new person and increase/decrease work hours accordingly.

Additionally, the application aims to implement report generation based on the duty, as it allows administrators to have a list of people assigned for a particular duty. The administrators may also delete individuals or members having particular attributes by searching and filtering using the attributes. The application also has a provision to reset the work-hours of all the students and facilities at the start of every academic year. The application also allows the administrator to delete a duty at their discretion. Lastly, The application will also allow the administrator to temporarily block a student or faculty from being allocated any duty. The administrator may unblock a person as and when they deem appropriate.

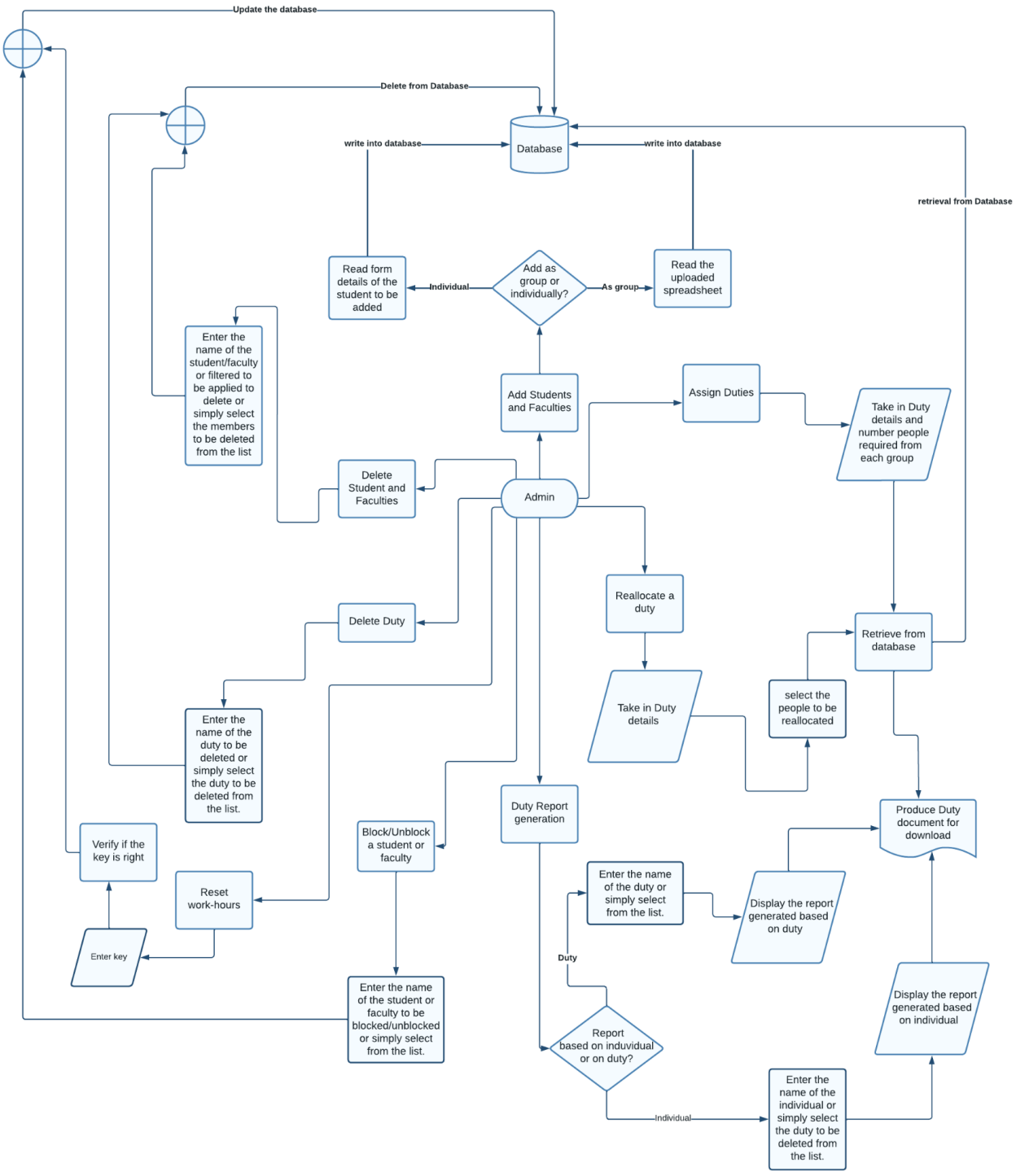


Fig. 1. High-level description of the product

2.2 Product Functionality

The Duty Normalizer Application allows the Administrators to:

- To assign duties to a specified number of people from each group
- To generate reports on a particular task in order to view the history of task allocation
- To reallocate a duty and update work hours in case of unavailability of a person.
- To add people to a particular group individually or as a batch through spreadsheets
- To delete people from each group individually or through a common feature.
- To delete a duty at the administrator's discretion.
- To temporarily block a student/faculty from being allocated any duty.
- To reset the work-hours after every academic year.

2.3 Design and Implementation Constraints

Database: Since the client requires no remote access, a local database will be used.

Hardware constraints: A smartphone with Android OS version 6 or above. Minimum 1 GB RAM to ensure seamless run of the application.

Internet: The smartphone need not be connected to the internet as the application uses a local database.

Design conventions and standards: This application development process will follow the principles specified by the SDLC.

Development Environment: The development environment used for the application will be Android Studio/VS Code/Pycharm.

Programming Standards: Effective Dart programming styles, as per official documentation, will be adhered to while developing the application. Consistent naming, ordering, and formatting will ensure that the application is maintainable

2.4 Assumptions and Dependencies

- The application will be provided directly to the administrator and will not be available for download on any public platform.
- There will only be one administrator.
- Remote Access is not required.
- The administrator always remembers the key given by the application developer.
- When deleting the duty at the administrator's discretion, the administrator is well aware of whether or not a particular duty has been completed by those assigned the duty, and the duty report for the same would not be available for future reference.

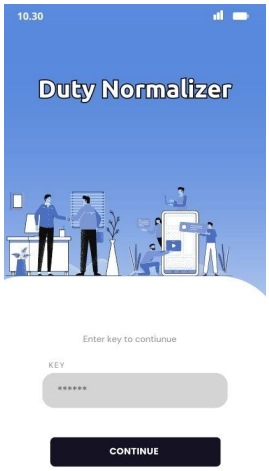
3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

User Interface for administrator is as follows:

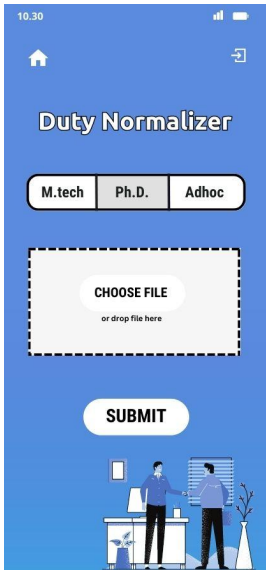
- Login Page:
  On this page, the user will enter a key given to them by the application developer and press the ‘Continue’ button to continue into the system.



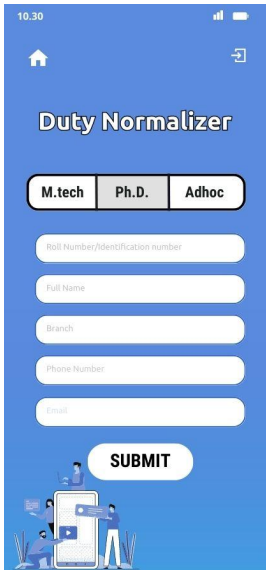
- Administrator’s Dashboard Page:
  On this page, there are buttons such as Assign Duties, Reallocate duty, Duty report generation, and delete duty. The taskbar at the bottom has icons for adding students/faculties to various categories, deleting students/faculties, Reset work hours, and block/unblock a student/faculty respectively.

- Add students/faculties Page:
  This page has two buttons namely add students/faculties in batches and Add students/faculties individually.

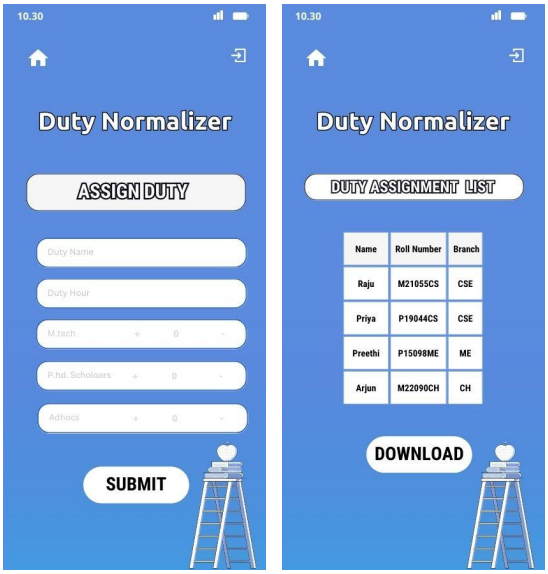
  By clicking add student/faculty in batch, the user will be reverted to a page with choose file button to upload the CSV file by selecting the category on top.



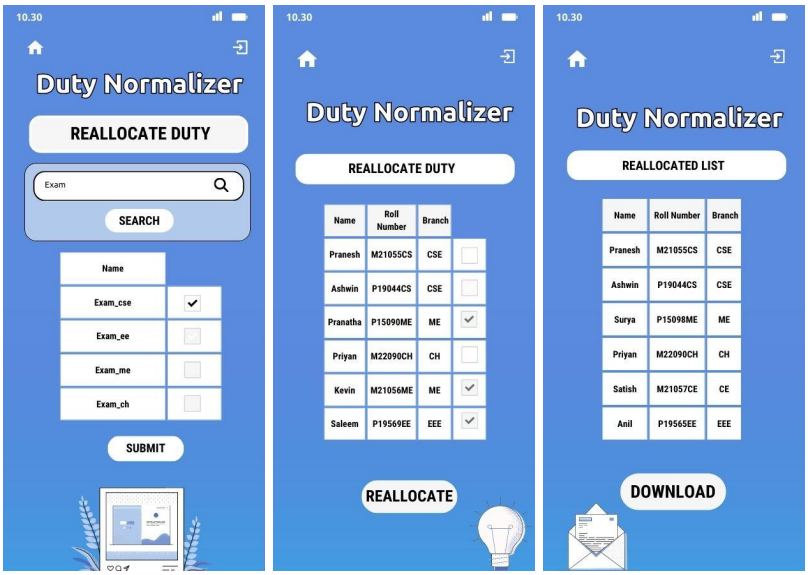
By clicking add student/faculty individually, the user will be reverted to a page that has a form to be filled out by selecting the category along with a submit button. Upon clicking the submit button, members are added.



- Assign Duty Page:
  This page has a form with duty details and the number of people required from each category along with a submit button. Upon filling out the form and clicking the submit button, duty will be allocated in a normalized sense and the assignment list will be displayed along with an option to download.

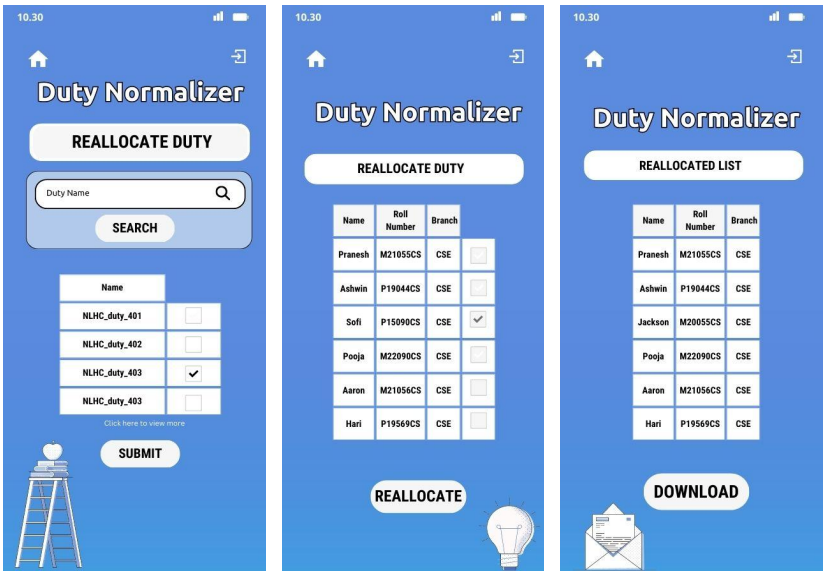


- Reallocate Duty Page:
  This page has a search bar where the name of the duty to be reallocated is to be entered along with a search button. Upon clicking search, the list of duties that match the name entered will be displayed. The administrator can select the duty he wishes to reallocate and click the ‘submit’ button. Upon clicking submit, the list of people assigned this duty will be displayed. Upon selecting members to be reallocated and upon clicking the reallocate button, the reallocated list will be displayed and the same will be available for download.



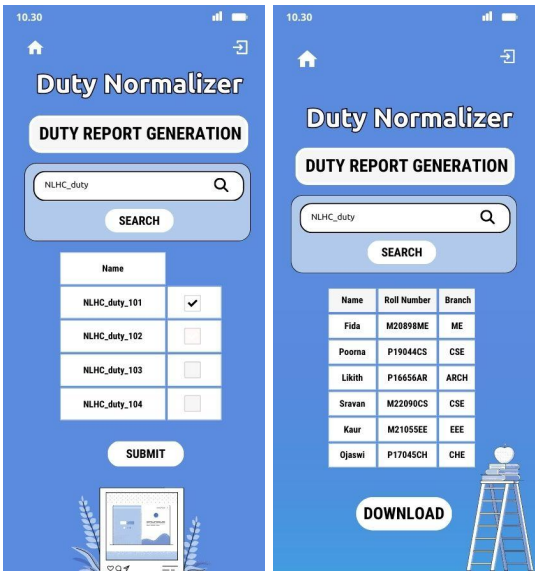
Additionally, the administrator can simply select the duty he wishes to reallocate from the list of duties instead of searching. Upon clicking submit, the list of people assigned this duty

will be displayed. Upon selecting members to be reallocated and upon clicking the reallocate button, the reallocated list will be displayed and the same will be available for download.

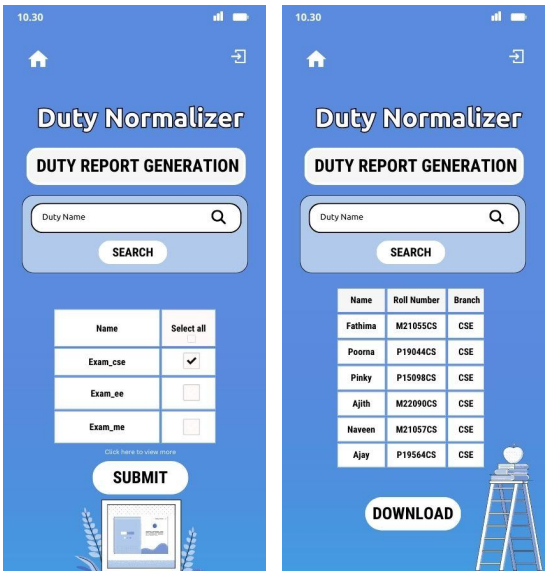


- Duty report Generation Page:
  The Duty report generation Page has two buttons namely generate report by student or generate report by duty.

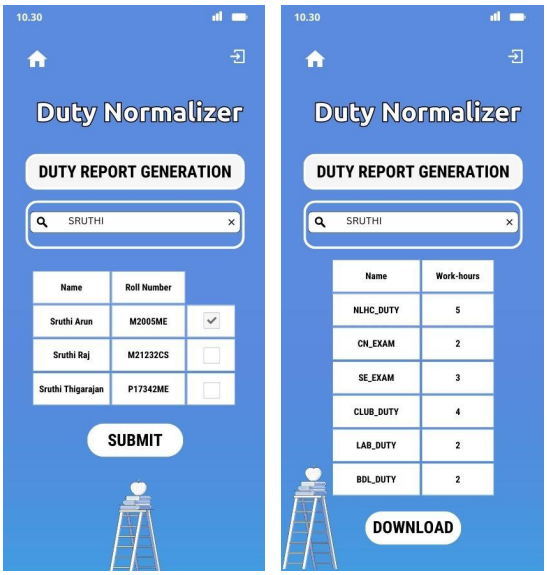
Upon clicking generate report on duty, the user is reverted to a page that has a search bar where the name of the duty can be entered along with a search button. Upon clicking the search button, a list of duty reports that match the name entered is displayed. The administrator can select the report he wishes to view/download and click submit. Upon clicking submit, the duty report is displayed along with an option to download in case the duty exists.



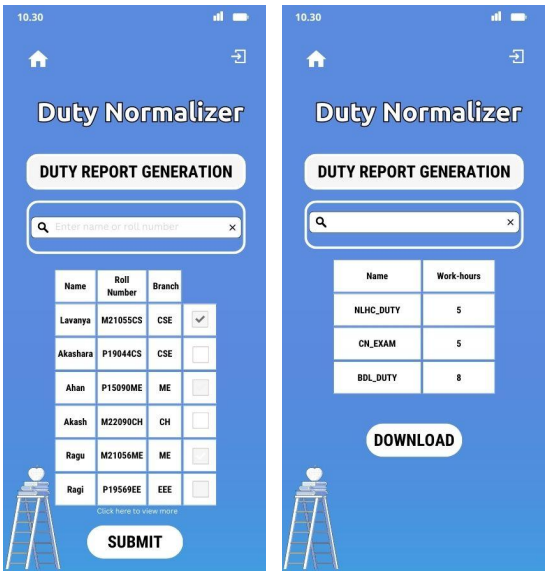
The user also has the option to simply select the duty from the list displayed and click the submit button. Upon doing so, the duty report is displayed along with an option to download.



Upon clicking generate report on individual, the user is reverted to a page that has a search bar where the name of the student can be entered. Upon searching, the matched names are displayed. The administrator can select the student from the list and upon clicking submit, the duties allocated to the student along with work-hours are displayed and the same is available for download.



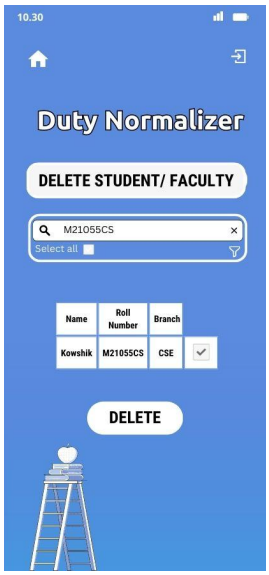
The user also has the option to simply select a student from the list displayed and click the submit button. Upon doing so, the duty report of the student is displayed, along with an option to download.



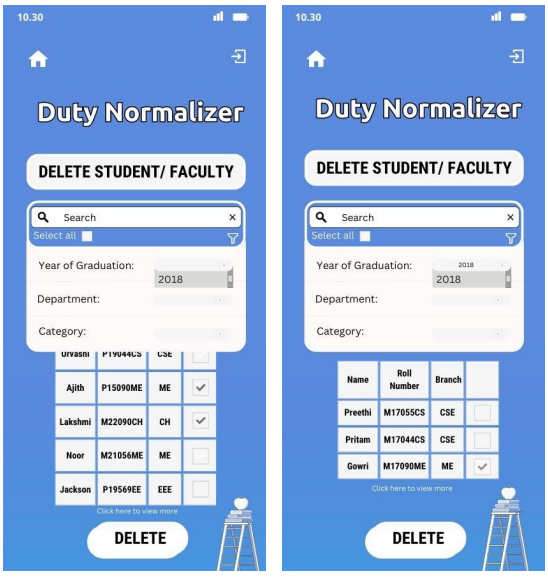
- Delete students/Faculties Page:
  This page has a search bar where the name of the student to be deleted can be entered or the common feature. If no name is entered in the search bar, the entire list is displayed and the user can select the members to be deleted and click delete.

  If the name is entered in the search bar, the names which match with the name entered are displayed and the administrator can select and delete the members they wish to delete.

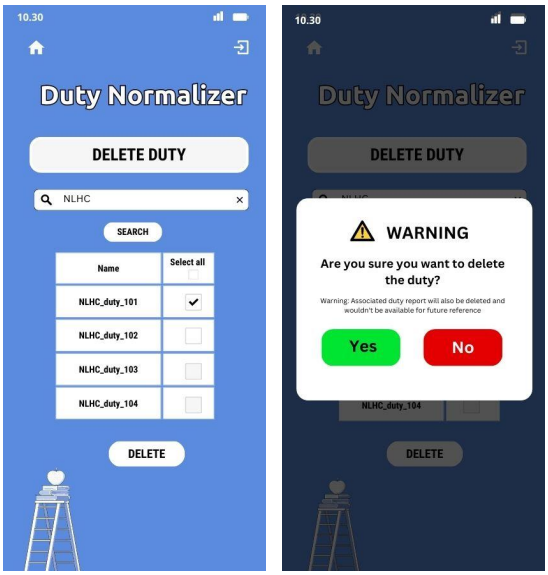
Additionally, the search bar also allows the user to enter the roll number and the roll number is matched and displayed. The administrator can select and delete the member if they wish.



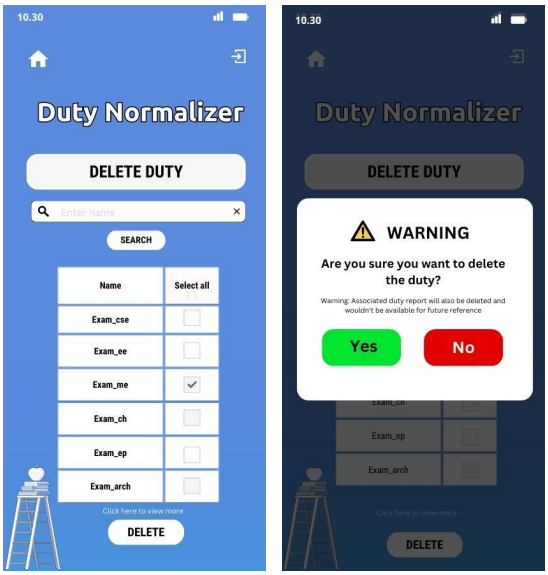
Additionally, the administrator may also click on the filter icon and apply filters. The students are filtered based on the criteria and displayed. The administrator can select and delete the members they wish to delete.



- Delete Duty Page:
  This page has a search bar where the name of the duty to be deleted can be entered along with a delete button. The user can enter the name of the duty they wish to delete and a list of all the matched duties are displayed.

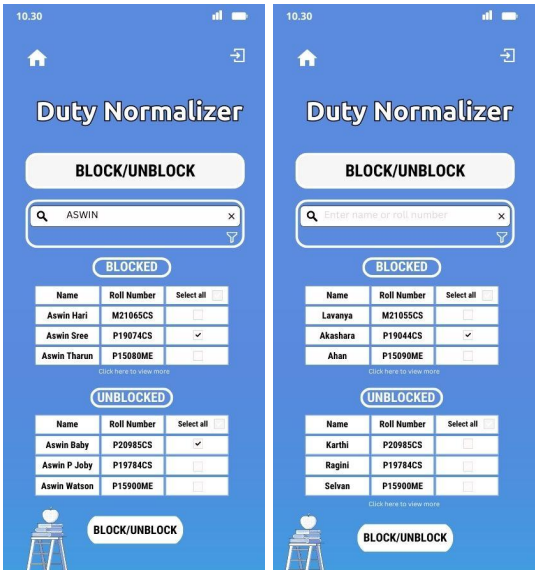


Additionally, since all the duties are displayed. Users can simply select the duty they wish to delete instead of searching and upon clicking delete, a warning pops up. Upon clicking yes, the duty is deleted.

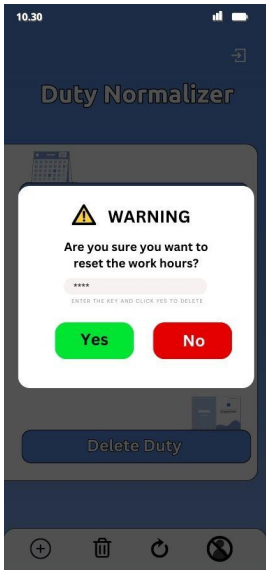


- Block/Unblock student or faculty Page:
  When the Block/Unblock icon on the front page is clicked, the administrator is directed to the block/unblock page, where they may temporarily block a person or unblock a blocked person. The page has a search bar where the name of the person can be entered. All the matched names in both the lists are displayed. The administrator may block or unblock as they deem appropriate. By selecting members from the blocked list and clicking the 'block/unblock' button, the administrator essentially unblocks the person. Similarly, by

  selecting members from the unblocked list and clicking the 'block/unblock' button, the administrator essentially blocks the person. Additionally, the administrator can also simply select from the list instead of searching and block/unblock as they prefer.



- Reset work-hours:
  When the reset icon on the dashboard is clicked, a danger pop-up pops up. Upon entering the valid key and clicking yes, the work hours of all the members are reset.



.

3.1.2 Hardware Interfaces

There must be a smartphone with a minimum of 1 GB RAM to ensure a seamless run of the application.

3.1.3 Software Interfaces

The smartphone's Operating system must be Android OS version 6 or above.

3.2 Functional Requirements

F1: The system shall allow the administrator to authenticate into the system after verifying key credentials.
F2: The system shall allow the administrators to add the students/facilities to the database in batches or individually
F3: The system shall allow the administrator to assign duties to the students/faculties by entering required duty details and the number of people required from each category.
F4: The system shall allow the administrator to reallocate a particular duty in case of unavailability of a student/faculty with new student/faculty from the same category.
F5: The system shall allow the administrator to view duty report(s) by searching and selecting the name(s) of the student/faculty or duty.
F6: The system shall allow the administrator to download duty reports.
F7: The system shall allow the administrator to delete duty report(s) by searching and selecting the name(s) of the duty.

F8: The system shall allow the administrator to delete students/facilities from the database.
F9: The system shall allow the administrator to temporarily block a student/faculty from being allocated for a duty.
F10: The system shall allow the administrator to unblock a student/faculty for being allocated to a duty.
F11: The system shall allow the administrator to reset every student's/faculty's work-hours to zero after entering a valid key.
F12: The system shall allow the administrator to log out.

3.3 Use Case Model

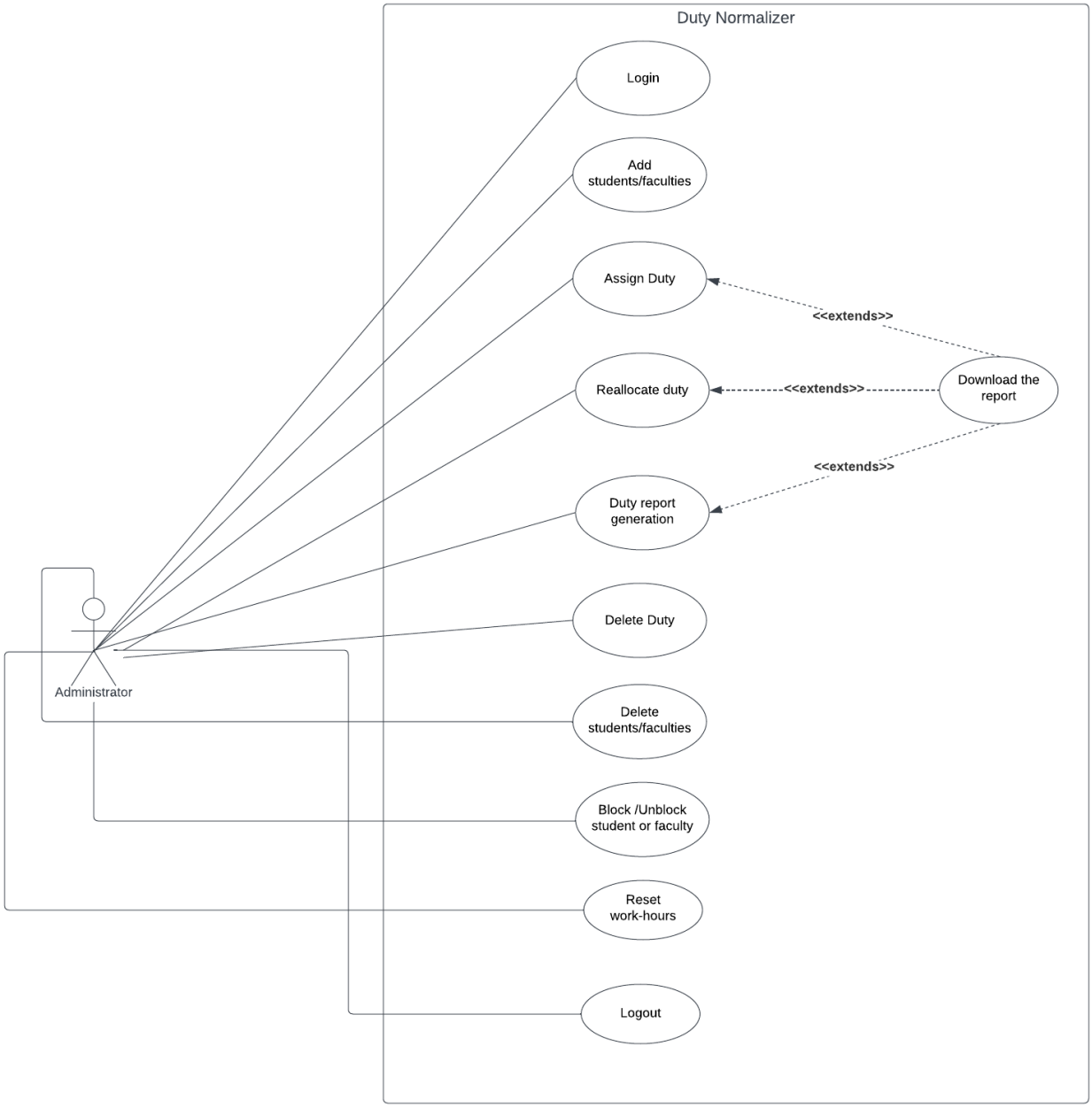


Fig. 2. Use Case Model

3.3.1 Use Case #1 (Login – U1)

Author – Aswin Sreekumar

Purpose – Allow the administrator to authenticate into the system after verifying key credentials

Requirements Traceability – F1

Priority – High.

Preconditions – None.

Post conditions – The administrator will be successfully authenticated into the system.

Actors – Administrator

Extends – None.

Flow of Events

1. **Basic Flow:**

Actor's Action	System's Response
1. The user enters the login key (The login key is provided by the application developer) and clicks the 'Continue'' button.	2. The system verifies the entered login key with the predefined key (set by the application developer) and logs the user into the system.

2. **Alternate Flow:** None

3. **Exceptions:**

Actor's Action	System's Response
1. The user enters the login key (The login key is provided by the application developer) and clicks the 'Continue'' button.	2. The system displays an error message stating that the entered login key is incorrect.

Includes – None

Notes/Issues – None

3.3.2 Use Case #2 (Add students/Faculties – U2)

Author – Rithika Kathirvel

Purpose – To add new students/Faculties into the database.

Requirements Traceability – F2

Priority – High

Preconditions – Administrator must be authenticated.

Post conditions – New students/faculties will be added to the database successfully.

Actors – Administrator

Extends – None

Flow of Events

1. **Basic Flow:**

Actor's Action	System's Response
1. Actor clicks the 'Add students/faculties' button from the Administrator's Dashboard Page.	2. System redirects to Add Students/Faculty page.
3. Actor chooses 'Add student/faculty individually'	4. The system displays options to choose Mtech, Ph.D. or Adhoc form with fields to enter details and a submit button.
5. Actor enters the details of the student/faculty to be added in the provided fields.	6. System reads the details entered and writes them into the database and displays added successfully

2. **Alternate Flow:**

Actor's Action	System's Response
1. Actor clicks the 'Add students/faculties' button from the Administrator's Dashboard Page.	2. System redirects to Add Students/Faculty page.
3. Actor chooses 'Add student/faculty in batch'.	4. System displays options to choose Mtech, Ph.D. or Adhoc, an option to upload a file and a submit button.
5. Actor uploads a CSV file and clicks the 'Submit' button.	6. System reads the uploaded CSV file, writes in the database, and displays added successfully.

3. **Exception:**

Actor's Action	System's Response
1. Actor clicks the 'Add students/faculties' button from the Administrator's Dashboard Page.	2. System redirects to Add Students/Faculty page.
3. Actor chooses 'Add student/faculty in batch'.	4. System displays options to choose Mtech, Ph.D. or Adhoc, an option to upload a file and a submit button.
5. Actor uploads a CSV file and clicks the 'Submit' button.	6. System reads the CSV file uploaded and if a record already exists, skip and read the rest and proceed accordingly.

Actor's Action	System's Response
1. Actor clicks the 'Add students/faculties' button from the Administrator's Dashboard Page.	2. System redirects to Add Students/Faculty page.
3. Actor chooses 'Add student/faculties individually'	4. The system displays options to choose Mtech, Ph.D. or Adhoc, a form with fields to enter details and a submit button.

Actor's Action	System's Response
5. The actor enters the details of the student/faculty to be added in the provided fields.	6. The system reads the form and displays that the record already exists.

Includes – None

Notes/Issues – None

3.3.3 Use Case #3 (Assign Duty – U3)

Author – Anand C L

Purpose – Allow the administrator to assign duties to students/faculties.

Requirements Traceability – F3

Priority – High

Preconditions – The list of students/faculties should not be empty and the administrator should be authenticated.

Post conditions – The administrator gets a list of required students/faculty for the corresponding duty.

Actors – Administrator

Extends – None

Flow of Events

1. **Basic Flow:**

Actor's Action	System's Response
1. Actor chooses "Assign Duty"	2. The system displays fields to provide a unique name to the duty, work-hours required for the duty and the number of

	people required from each category.
3. Actor enters a unique name for the duty, the number of students/faculty required from each category and the work hour assigned for that particular duty and clicks the “Submit” button.	4. The system generates a list of required students/faculty as submitted by the Actor in a normalized manner (The student/faculty with lower work-hours are given more priority).

2. **Alternative Flow:** None

3. **Exceptions:**

Actor’s Action	System’s Response
1. Actor chooses “Assign Duty”	2. The system displays fields to provide a unique name to the duty, work-hours required for the duty and the number of people required from each category.
3. Actor enters a unique name for the duty, the number of students required from each category and the work hour needed for that particular duty and clicks the “Submit” button.	4. The system shows an error message stating there is no sufficient number of students/faculty present in a category.

Actor’s Action	System’s Response
1. Actor chooses “Assign Duty”	2. The system displays fields to provide a unique name to the duty, work-hours required for the duty and the number of people required from each category.
3. Actor enters a name for the duty, the number of students required from each category and the work hour needed for that particular duty and clicks the “Submit” button.	4. The system shows an error message stating that the task name is already taken and provides suggestions for a new name.

Includes – None

Notes/Issues – None

3.3.4 Use Case #4 (Reallocate Duty – U4)

Author – Anand C L

Purpose – Allows the user to reallocate the students/faculties assigned to a duty.

Requirements Traceability – F4

Priority – Medium.

Preconditions – The administrator must be authenticated.

Post conditions – The new list of students/faculty for the particular duty will be created and their work-hours will be updated.

Actors – Administrator

Extends – None

Flow of Events

1. **Basic Flow:**

Actor's Action	System's Response
1. Actor chooses 'Reallocate Duty'	2. The system displays a list of existing duties and a field to search for duty by name.
3. Actor searches for the duty by entering the name and clicks the 'Search' button.	4. The system filters and displays the list of duties with names similar to the one entered by the Actor.
5. Actor selects the required duty from the list and clicks the 'Submit' Button.	6. The system displays the list of people allocated for that duty.
7. Actor selects the people to be reallocated and clicks on 'Reallocate'.	8. The system generates and allocates new students/faculty for the duty.

2. **Alternate Flow:** None

3. **Exception:**

Actor's Action	System's Response
1. Actor chooses 'Reallocate Duty'	2. The system displays a list of existing duties and a field to search for duty by name.
3. Actor searches for the duty by entering the name and clicks the 'Search' button.	4. System displays no duties that match the name entered

Actor's Action	System's Response
1. Actor chooses 'Reallocate Duty'	2. System displays a list of existing duties and a field to search for duty by name.
3. Actor searches for the duty by entering the name and clicks the 'Search' button.	4. The system filters and displays the list of duties with names similar to the one entered by the Actor.
5. Actor selects the required duties from the list and clicks the 'Submit' Button.	6. The system displays the list of people allocated for the duty.
5. Actor selects the people to be reallocated and clicks on 'Reallocate'.	6. The system displays that there is no other person available in a particular selected category for reallocation.

Includes – None

Notes/Issues – None

3.3.5 Use Case #5 (Duty Report Generation – U5)

Author – Aditya Premjit

Purpose – Allow the administrator to view duty reports by searching and selecting the name(s) of the student/faculty or duty.

Requirements Traceability – F5

Priority – High

Preconditions – The administrator must be authenticated.

Post conditions – A Report is displayed on the Administrator's device.

Actors – Administrator

Extends – None

Flow of Events

1. **Basic Flow:**

Actor's Action	System's Response
1. Actor chooses 'Duty Report Generation'.	2. System redirects to the 'Duty Report Generation' page.
3. Actor chooses 'Generate report on individual'.	4. System displays the list of existing students/faculties and a search bar to search the student/faculty by name.
5. Actor searches for the student/faculty by entering the name and clicks the 'Search' button.	6. The system filters and displays the list of students/faculties with names similar to the one entered by the Actor.
7. Actor selects the required students/faculties from the list and clicks the 'Submit' button.	8. System generates a report that consists of a list of duties along the number of work-hours for each duty for the selected students/faculties.

2. **Alternate Flow**:

Actor's Action	System's Response
1. Actor chooses 'Duty Report Generation'.	2. System redirects to the 'Duty Report Generation' page.
3. Actor chooses 'Generate report on duty'.	4. System displays the list of existing duties and a search bar to search the duty by name.
5. Actor searches for the duty by entering the name and clicks the 'Search' button.	6. The system filters and displays the list of duties with names similar to the one entered by the Actor.
7. Actor selects the required duties from the list and clicks the 'Submit' button.	8. System generates a report that consists of work-hours for the duty, a list of people allocated for the duty along with their category(Mtech, Ph.D., Adhoc) for the selected duties.

Actor's Action	System's Response
1. Actor chooses 'Duty Report Generation'.	2. System redirects to the 'Duty Report Generation' page.
3. Actor chooses 'Generate report on duty'.	4. System displays the list of existing duties and a search bar to search the duty by name.
5. Actor clicks the 'Select all' button and then clicks the 'Submit' button.	6. System generates reports for all the existing duties.

Actor's Action	System's Response
1. Actor chooses 'Duty Report Generation'.	2. System redirects to the 'Duty Report Generation' page.
3. Actor chooses 'Generate report on individual'.	4. System displays the list of existing students/faculties and a search bar to search the student/faculty by name.
5. Actor clicks the 'Select all' button and then clicks the 'Submit' button.	6. System generates reports for all the existing students/faculties.

3. **Exception:**

Actor's Action	System's Response
1. Actor chooses 'Duty Report Generation'.	2. System redirects to the 'Duty Report Generation' page.
3. Actor chooses 'Generate report on duty'.	4. System displays the list of existing duties and a search bar to search the duty by name.
5. Actor searches for the duty by entering the name and clicks the 'Search' button.	6. System displays no duties that match the name entered

Actor's Action	System's Response
1. Actor chooses 'Duty Report Generation'.	2. System redirects to the 'Duty Report Generation' page.
3. Actor chooses 'Generate report on individual'.	4. System displays the list of existing students/faculties and a search bar to search the student/faculty by name.
5. Actor searches for the student/faculty by entering the name and clicks the 'Search' button.	6. System displays no students/faculties that match the name entered.

Includes – None

Notes/Issues – None

3.3.6 Use Case #6 (Download the Report – U6)

Author – Navaneeth Shanavasan

Purpose – Allow the administrator to download duty reports.

Requirements Traceability – F6

Priority – High

Preconditions – The administrator must be authenticated and there exists a report as per the required details

Post conditions – A Report in pdf format is saved in the Administrator's device.

Actors – Administrator

Extends – U3, U4, U5

Flow of Events

1. **Basic Flow:**

Actor's Action	System's Response
1. Actor clicks the 'Download' button.	2 . System Generates a report in PDF format and saves it on the Administrator's device

2. **Alternate Flow:** None

3. **Exception:** None

Includes – None

Notes/Issues – None

3.3.7 Use Case #7 (Delete Duty – U7)

Author – Navaneeth Shanavasan

Purpose – Allow the administrator to delete duty report(s).

Requirements Traceability – F7

Priority – High

Preconditions – The administrator must be authenticated.

Post conditions – Removes the specified duty from the database

Actors – Administrator

Extends – None

Flow of Events

1. **Basic Flow:**

Actor's Action	System's Response
1. Actor chooses 'Delete Duty'.	2. System displays the list of existing duties and a search bar to search the duty by name
3. Actor searches for the duty by entering the name and clicks the 'Search' button.	4. The system filters and displays the list of duties with names similar to the one entered by the Actor.
5. Actor selects the duties to be deleted from the list and clicks the 'Delete' button.	6. System displays a warning to the Actor to confirm the deletion.
7. Actor clicks 'Yes'.	8. System removes the duty from the database and displays 'The Duty has been deleted Successfully'.

2. **Alternate Flow:** None

3. **Exception:**

Actor's Action	System's Response
1. Actor chooses 'Delete Duty'.	2. System displays the list of existing duties and a search bar to search the duty by name.
3. Actor searches for the duty by entering the name and clicks the 'Search' button.	4. System displays no duties that match the name entered

Includes – None

Notes/Issues – None

3.3.8 Use Case #8 (Delete Students/Faculties – U8)

Author – Navaneeth Shanavasan

Purpose – Allow the administrator to delete students/facilities from the database.

Requirements Traceability – F8

Priority – High

Preconditions – The administrator should be authenticated.

Post conditions – Removes the specified students/faculties from the database

Actors – Administrator

Extends – None

Flow of Events

1. **Basic Flow:**

Actor's Action	System's Response
1. The actor chooses 'Delete students/faculties'.	2. System displays a list of enrolled candidates along with an option to search for a specific person or filter based on properties.
3. Actor searches for the person by entering the name.	4. The system filters and displays the list of people with names similar to the one entered by the Actor.
5. Actor selects the people to be deleted from the list and clicks the 'Delete' button.	6. System displays a warning to the Actor to confirm the deletion.
7. Actor clicks 'Yes'.	8. System removes the people from the database and displays 'The selected

	candidates have been removed Successfully'.

2. Alternate Flow:

Actor's Action	System's Response
1. The actor chooses 'Delete students/faculties'.	2. System displays a list of enrolled candidates along with an option to search for a specific person or filter based on properties.
3. Actor fills in the required properties.	4. The system filters and displays the list of people with the specified properties.
5. Actor selects the people to be deleted from the list and clicks the 'Delete' button.	6. System displays a warning to the Actor to confirm the deletion.
7. Actor clicks 'Yes'.	8. System removes the people from the database and displays 'The selected candidates have been removed Successfully'.

3. Exception:

Actor's Action	System's Response
1. The actor chooses 'Delete students/faculties'.	2. System displays a list of enrolled candidates along with an option to search for a specific person or filter based on properties.
3. Actor searches for the person by entering the name.	4. System displays that no such candidate exists.

Actor's Action	System's Response
1. The actor chooses 'Delete students/faculties'.	2. System displays a list of enrolled candidates along with an option to search for a specific person or filter based on properties.

3. Actor fills in the required properties.	4. System displays that no such candidate exists.

Includes – None

Notes/Issues – None

3.3.9 Use Case #9 (Block/Unblock Student or Faculty – U9)

Author – Aditya Premjit

Purpose – Allow the administrator to block/unblock a student/faculty from being allocated for a duty.

Requirements Traceability – F9, F10

Priority – High

Preconditions – The administrator should be authenticated.

Post conditions – The selected students/faculties are temporarily blocked/unblocked from being allocated for duties.

Actors – Administrator

Extends – None

Flow of Events

1. **Basic Flow:**

Actor's Action	System's Response
1. The actor clicks the 'Block/Unblock students or faculties' button.	2. System displays a list of both blocked and unblocked candidates along with an option to search for a specific person.
3. Actor searches for the person by entering the name.	4. The system filters and displays the list of people with names similar to the one entered by the Actor.
5. Actor selects unblocked people from the	6. The system blocks the selected people

list and clicks the ‘Block/Unblock’ button.	from being allocated for duties.

2. Alternate Flow:

Actor’s Action	System’s Response
1. The actor clicks the ‘Block/Unblock students or faculties’ button.	2. System displays a list of both blocked and unblocked candidates along with an option to search for a specific person.
3. Actor searches for the person by entering the name.	4. The system filters and displays the list of people with names similar to the one entered by the Actor.
5. Actor selects both unblocked and blocked people from the list and clicks the ‘Block/Unblock’ button.	6. System blocks the selected unblocked people from being allocated for duties and unblocks the selected blocked people from being allocated for duties.

Actor’s Action	System’s Response
1. The actor clicks the ‘Block/Unblock students or faculties’ button.	2. System displays a list of both blocked and unblocked candidates along with an option to search for a specific person.
3. Actor searches for the person by entering the name.	4. The system filters and displays the list of people with names similar to the one entered by the Actor.
5. Actor selects blocked people from the list and clicks the ‘Block/Unblock’ button.	6. System unblocks the selected people for being allocated duties.

3. Exception:

Actor’s Action	System’s Response
1. The actor clicks the ‘Block/Unblock students or faculties’ button.	2. System displays a list of both blocked and unblocked candidates along with an option to search for a specific person.
3. Actor searches for the person by	4. System displays that no such candidate

entering the name.	exists.

Includes – None

Notes/Issues – None

3.3.10 Use Case #10 (Reset work-hours – U10)

Author – Rithika Kathirvel

Purpose – To reset all work-hours to zero at the start of every academic year.

Requirements Traceability – F11

Priority – High

Preconditions – Administrator should be authenticated.

Post conditions – All the work hours will be set to zero.

Actors – Administrator

Extends – None

Flow of Events

1. **Basic Flow:**

Actor's Action	System's Response
1. The actor clicks the 'Reset work-hours' button	2. The system displays a warning to the Actor to confirm the reset and asks for a key.
3. The Actor enters the key and clicks 'Yes'.	4. The work-hours of all the students/faculties are set to zero.

2. **Alternate Flow:** None

3. **Exception:**

Actor's Action	System's Response
1. The actor clicks the Reset work-hours button	2. The system displays a warning to the Actor to confirm the reset and asks for a key.
3. The Actor enters the key and clicks 'Yes'.	4. The System displays that the entered key is incorrect.

Includes – None

Notes/Issues – None

3.3.11 Use Case #11 (Log out – U11)

Author – Aswin Sreekumar

Purpose – The administrator is logged out of the system.

Requirements Traceability – F12

Priority – Low

Preconditions – The User must be authenticated into the system.

Post conditions – The User is successfully logged out.

Actors – Administrator

Extends – None

Flow of Events

1. **Basic Flow:**

Actor's Response	System's Response
1. The user clicks the 'Logout' button.	2. The system logs the user out and returns to the login screen.

2. **Alternate Flow:** None

3. **Exception:** None

Includes – None

Notes/Issues – None

4 Other Non-functional Requirements

4.1 Performance Requirements

- **System:** The application will run on all devices running Android 6 (Marshmallow) or later. The application will be responsive.
- **Response Time:** The application should load and be usable within 3 seconds. It should update the interface on interaction within 2 seconds.

4.2 Safety and Security Requirements

The application will be provided directly to the administrator and will not be available for download on any public platform. Hence, there will be no vulnerabilities. The application also authenticates the user hence ensuring authorized access.

4.3 Software Quality Attributes

4.3.1 Reliability

The application does not require Internet connectivity. For fetching any data, connectivity is dispensable to the functioning of this application. Hence, this application is reliable.

4.3.2 Correctness

The application will be correct in terms of adhering to the functional requirements and will exhibit all the necessary functionalities.

4.3.3 Availability

The application will run seamlessly. Since the data is stored in a local database and not on any cloud, the application can be used by the user all the time without connectivity.

4.3.4 Scalability

The development team will write clean codes and use scalable relational databases. Investing in an SSD card for their mobile phone can scale the application to handle more records.

4.3.5 Maintainability

The development team will follow the best practices and adhere to the officially documented effective Dart techniques. In addition, software modularity will be followed throughout, making the application maintainable. An extensive maintenance document will be provided for future reference.

4.3.6 Portability

The application will be made using Flutter and will run on devices running Android 6.0 or higher.

4.3.7 Extensibility

The code for the application will be available, and extensions can easily be created as and when required.

4.3.8 Reusability

The components of this application are very simple and will likely not be reused.

4.3.9 Usability

The application will have a UI that will be simple to use and explain. New users will be able to use it with ease.

4.3.10 Robustness

The application will have robust features correctly implementing the above use cases and will handle exceptions when required.

5 Other Requirements

There is no other requirement needed for this project.

Appendix A - Activity Log

Serial Number	Meeting Date	Meeting time	Description
1	17-02-2023	6 pm - 7 pm	Discussed the requirements with the owner over a google meet through an hour-long session. The major functionalities required and the constraints were made clear.
2	10-03-2023	8 pm - 10 pm	The group met offline, and the details of the functionalities provided to the users were finalized.
3	13-03-2023	5 pm - 7 pm	The UI design was finalized through an offline meeting.
4	16-03-2023	6 pm - 11 pm	The use case description was finalized through a group discussion.
5	21-03-2023	6 pm - 9 pm	The revisions were discussed and work was divided.
6	22-03-2023	7 pm - 10 pm	The revised UI designs were finalized through an offline meeting.
7	23-03-2023	11 am - 12 pm	Discussion based on the mail reviewed from the client.
8	23-03-2023	6 pm - 9 pm	The document was reviewed and finalized before submission.

Serial Number	Team Member Name	Contribution
1	Aswin Sreekumar	Hardware Interface, Functional Requirements F1, F2, F3, F4, F6, F9, Use case model, Use case #1, #11
2	Rithika Kathirvel	Introduction, User Interfaces, Software Interface, Functional Requirements F5, F7, F8, F10, F11, Software Quality Attribute 4.3.1- 4.3.2, Use case #2, #10
3	Navaneeth Shanavasan	Permonfance Requirements, Safety and Security Requirements and Software Quality Attribute 4.3.3 - 4.3.7, Use case #6, #7, #8
4	Anand C L	Design and Implementation, Assumptions, and Dependencies, Other Requirements, Use case #3, #4
5	Aditya Premjit	Product overview, High-level diagram of the product, Production functionality, Use case #5, #9