UNIVERSITY OF JAFFNA, SRI LANKA

Faculty of Engineering

ACADEMIC CALENDAR MANAGEMENT SYSTEM

GitHub: https://github.com/IsuruAkalanka/academic_calendar

by

CUMARANATHUNGA P.I.A. (2018/E/022)
ERANDA A.H. (2018/E/031)
PERERA U.L.K.K. (2018/E/091)
SACHINTHA D.C. (2018/E/104)
SATHURJAN J. (2018/E/108)

A project proposal submitted to the
Department of Computer Engineering
In partial fulfillment of the requirements for the
course module EC6060 software engineering for the
Degree of Bachelor of Science of Engineering

29th July 2022

Introduction and Cover Letter

Kavindya Perera
Leader – Team 01
Department of Computer Engineering
Faculty of Engineering
University of Jaffna

Dr. S. Thananjeyan
Senior Lecturer
Department of Electrical and Electronic Engineering
Faculty of Engineering
University of Jaffna

08th June 2022

Dear Dr. Thananjeyan,

Please find enclosed our detailed software proposal for your kind consideration.

At Department of Computer Engineering, Faculty of Engineering, University of Jaffna we are aware that creating **Academic Calendar Management System** software takes a mixture of technical excellence and clear communication and our team hires only the very best to ensure you receive both. We know that every client is unique and we strive to deliver an individual, innovative and affordable proposal and to follow it through with an outstanding delivery which is both on time and within budget.

We have more development in this area and our previous developments include theatre booking system in software construction, database management system projects

Please let us know if you would like to get in touch with our existing clients from whom you will receive nothing but positive endorsements. You may also wish to review our previous work and learn more about our development skills and knowledge.

We also pride ourselves on our after-sales client-care including our guarantees, staff-training and onsite and offsite support.
Finally, we realize that you are very busy and wanted to thank you in advance for your time spent reviewing our proposal.

Yours Truly,

Kavindya Perera

Computer Department

Faculty of Engineering, University of Jaffna.

1. CUSTOMER PROBLEM STATEMENT

a. Problem Statement

Executive Summary

The modern world has been irrevocably shaped by technology and for good reason. Many long-standing problems in industries can be solved with the right program. It's not enough to simply use pre-existing solutions, however.

Create Academic calendar management system

After a thorough but efficient preparation period, our team finally decided to create an academic calendar management system for AR.

We are expecting to give our main focus specially on the semester periods, exam periods, holiday periods, course registration periods, and exam registration periods. The resulting software will achieve:

- Clear visualization of when the semester begins/ ends
- Clear visualization of exam periods
- Clear visualization of holiday periods
- Clear visualization of course registration open/close date and the duration
- Clear visualization of exam registration open/close date and the duration
- Feature to add/update/delete all the above periods

Integrate MS Teams and Trello

With the coordination with Dr. Thananjeyan, we got the use of both MS Teams and Trello for this project.

We will achieve the goals listed above by utilizing agile methodology. This process allows us to clearly visualize the academic calendar and edit the particular durations.

Up to now, planning we decided to complete this project within ten-week time.

Technical obstacles

Technical skills

For this project we have to work with a particular tech stack which every member of the team might not be familiar with. Being familiar with the stack and being able to develop the requirements can be a bit of a challenge under the current deadlines.

Integration Issues

It might be a challenge to integrate what we are making with tools currently existing. We have to find ways to make our product compatible using APIs so the app can be used in other products as an external module.

Industry and market risks

Following are some of the applications already exist for calendar management purpose

- Infinity
- Google Calendar
- Apple Calendar
- Outlook Calendar
- Fantastical 2

Most of these providers provide following features to the user through their application

- Schedule creation and management
- Customizable views and layouts
- Necessary integrations with other apps
- Shareable appointments or entire calendars, event
- User-friendly interface

While these applications provide the above features, they are not meant for a task like academic semester scheduling. So, in this project we are focusing on that particular area. Possibility of user's current requirement change or another potential solution from the competitors to match the user requirements are potential risks.

2. SYSTEM OF REQUIREMENTS

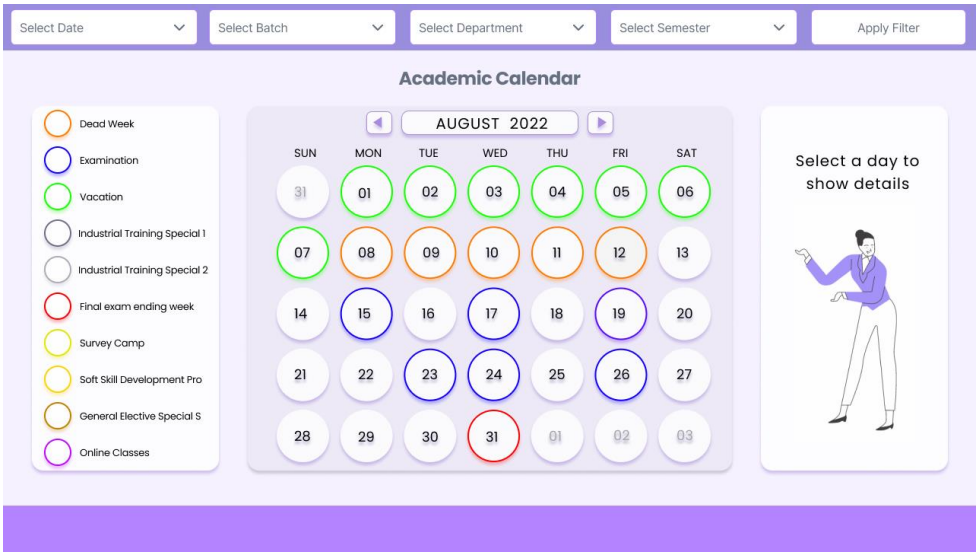
a. Enumerated Functional Requirements

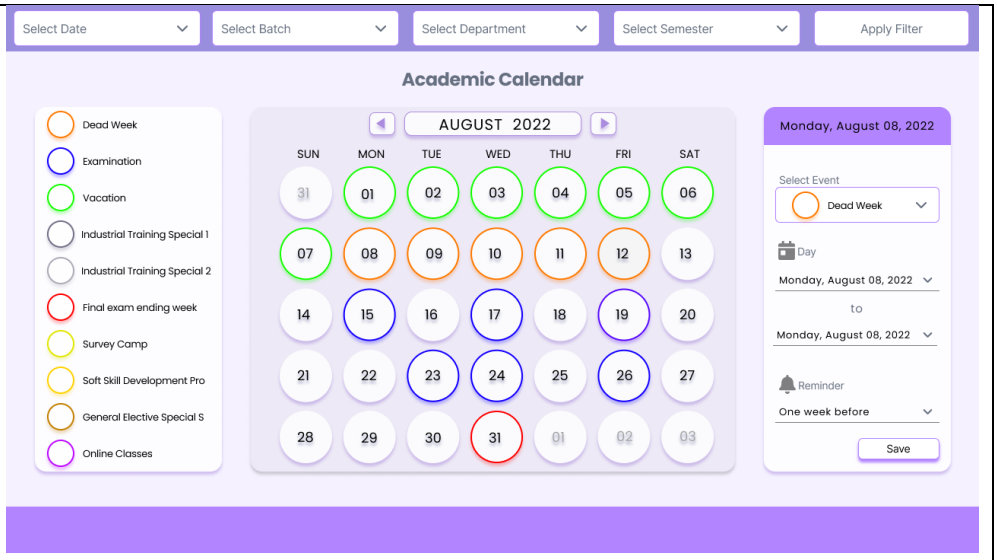
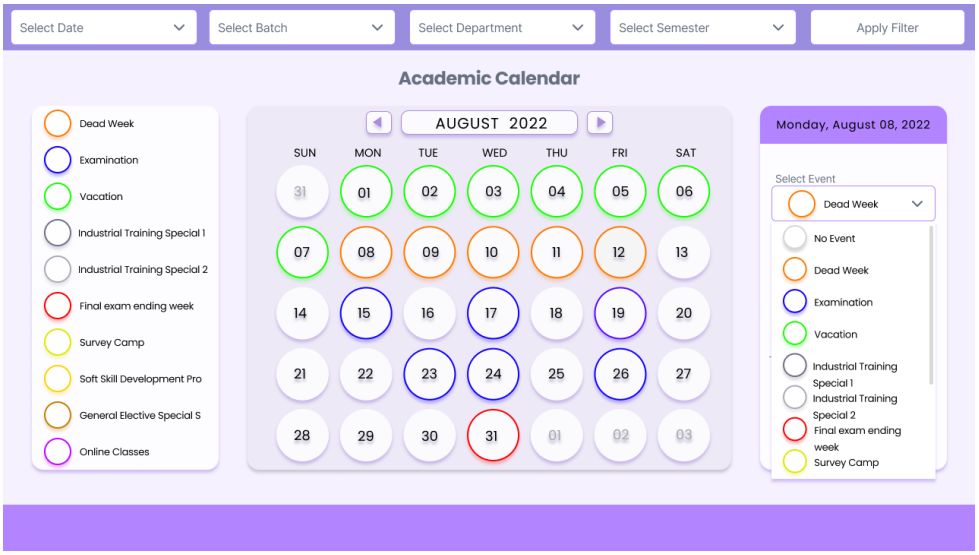
IDENTIFIER	PRIORITY WEIGHT	REQUIREMENTS
REQ-1	10	Basic CRUD operations for event managing
REQ-2	10	System should allow Administrative/ Staff/ All Undergraduates to view the summary
REQ-3	8	Filter the calendar based on the user preferences
REQ-4	10	System should allow API for other systems to use notifications system
REQ-5	7	Should send notifications automatically
REQ-6	9	Should expose APIs for other users to use
REQ-7	10	Login functionality for authorized users and API users

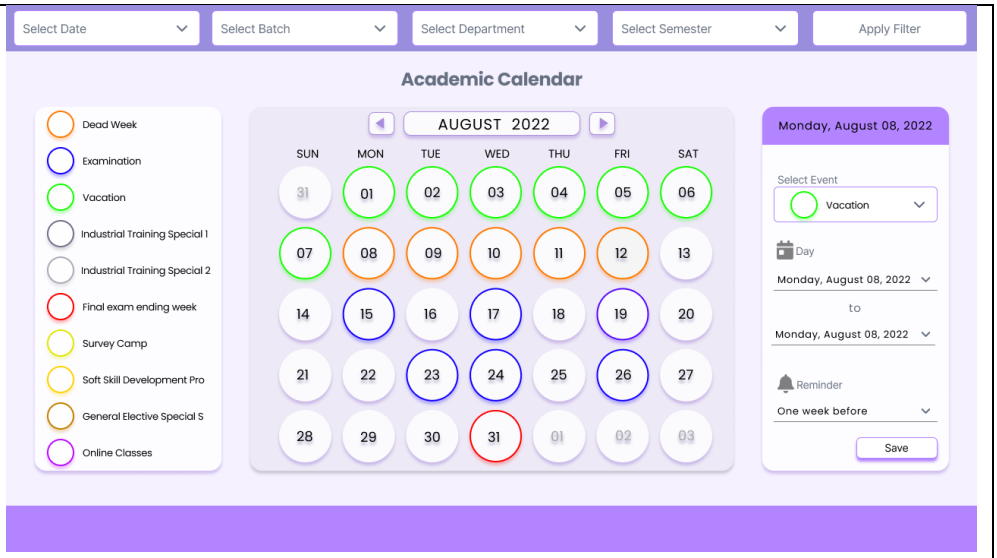
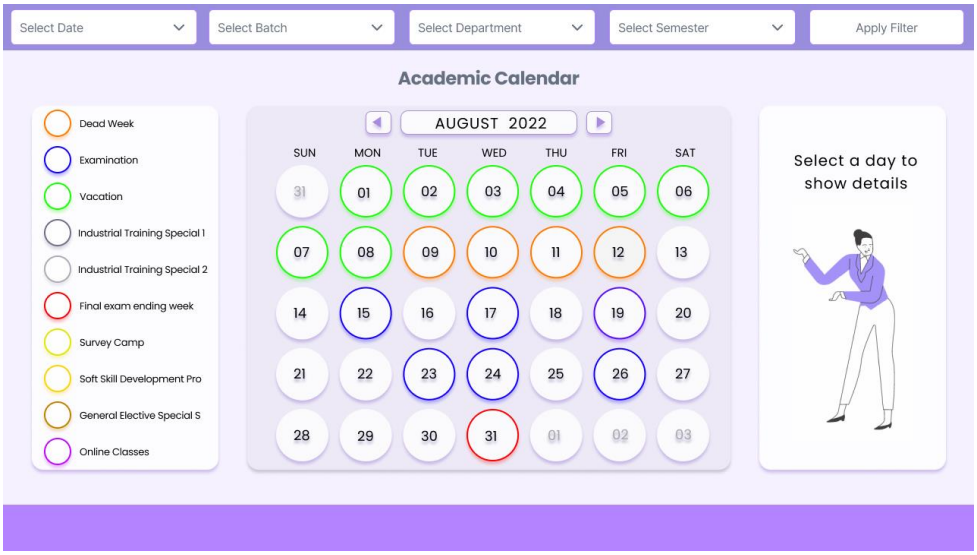
b. Enumerated Non-Functional Requirements

IDENTIFIER	PRIORITY WEIGHT	REQUIREMNTS
REQ-8	10	As a system, only authorized users have the eligibility for the system
REQ-9	6	As s system should work properly under any traffic condition
REQ-10	6	As a system, it system should be responsive and user friendly
REQ-11	7	As a system, it should be optimized and should worth for the hosting costs.

c. User Interface Requirements

IDENTIFIER	PRIORITY WEIGHT	REQUIREMENTS
REQ- 12	10	Landing page 
REQ -13	10	Selecting a date (8th of August, 2022)

		
REQ -14	10	Selecting an event (Click Vacation) 
REQ -15	10	Saving updated details (Click Save)

		
REQ -16	10	Updated Calendar 

3. FUNCTIONAL REQUIREMENT SPECIFICATIONS

a. Actors and Goals

ACTOR	ACTOR'S GOAL	USE CASE NAME
AR System Admin	To do all the operations CRUD for the events	Event handling (UC - 1)
Staff/undergraduates	To view the summary	View the Summary (UC - 2)
IT Management	To look in to System Faults	System management (UC - 3)
API users	To send notifications to selected group of students of staff	Notification sending (UC - 4)

Use cases

i. Casual Description

UC#1 Event Handling

Here AR System Admin is having all the responsibility to handle all the events. He is responsible for creating, viewing, updating and deleting all the events

1. Semester start/ end dates
2. Semester start/ close Registration dates
3. Holidays
4. Dead Weeks
5. Mid/ End Semester Examinations
6. Exam Registrations

UC#2 View the Summary

In the system all the Undergraduates as well as all the Staff are having the access to view the Summary**.**

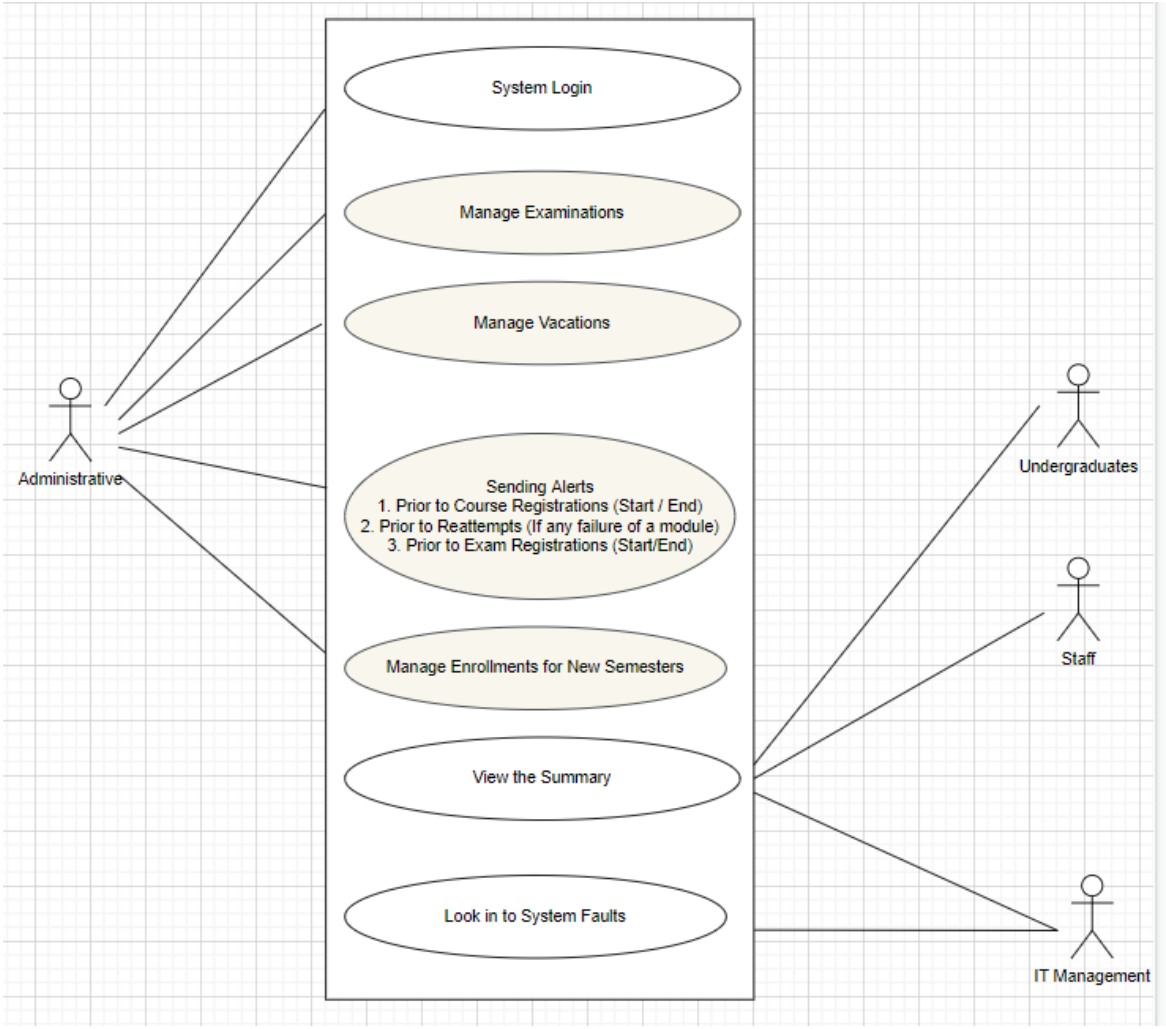
UC#3 System Management

If any System fault occurs, IT Management should look in to the System faults and do the needful.

UC#4 Send Alerts/ Notifications

AR System Admin should allow API for other systems to use notifications system.

ii. Use Case Diagram



iii. Traceability Matrix

The below table depicts the mapping of the various requirements of our system with the use cases defined previously. The requirements are given based on a scale from 1 to 10, 1 being the lowest priority and 10 being the highest priority.

REQ'T	PW	UC1	UC2	UC3	UC4
REQ-1	10	x			
REQ-2	10		x		
REQ-3	8		x		
REQ-4	10	x			
REQ-5	7				x
REQ-6	9				
REQ-7	10	x		x	x
REQ-8	10			x	x
REQ-9	6				
REQ-10	6				
REQ-11	7				
Max Weight		10	10	10	10
Total Weights		30	18	20	27

iv. Fully Dressed Description

Use Case UC#1Event Handling

Related Requirements: REQ -1, REQ -4, REQ -7

Initiating Actor: AR System Admin

Actor's Goal: To handle events in a proper manner

Participating Actors: - AR System Admin

Precondition: System should be active and running, AR System Admin should have established connection with the System

Postcondition: AR System Admin can do any required operations for events

Failed and condition: AR System Admin should have to wait for some more time and try again, for the requirements

Use Case UC#2View the Summary

Related Requirements REQ -2, REQ -3

Initiating Actor: AR System Admin, Staff, Undergraduates

Actor's Goal: To view the summary to schedule their curriculum

Participating Actors: AR System Admin, Staff, Undergraduates

Precondition: System should be active and running, Users should have established connection with the System

Postcondition: Users can view the Summary

Failed and condition: Try again later again, for the requirements

Use Case UC#3: System Management

Related Requirements: REQ -7, REQ -8

Initiating Actor: AR System Admin

Actor's Goal: To handle events in a proper manner

Participating Actors: - IT management

Precondition: System should be active and running

IT management should have established connection with the System

Postcondition: IT management can do any required operations for events

Failed and condition: IT management should take necessary actions to make the system up and running again

Use Case UC#4: Send Alerts/ Notifications

Related Requirements: REQ -4, REQ -7, REQ -8

Initiating Actor: AR System Admin, API users

Actor's Goal: To handle events in a proper manner

Participating Actors: - API users, Students and staff

Precondition: System should be active and running

AR System Admin or API users should have established connection with the System and authenticated

Postcondition: AR System Admin can do any required operations for events

Failed and condition: Wait for some time and try again.

b. Sequence Diagrams

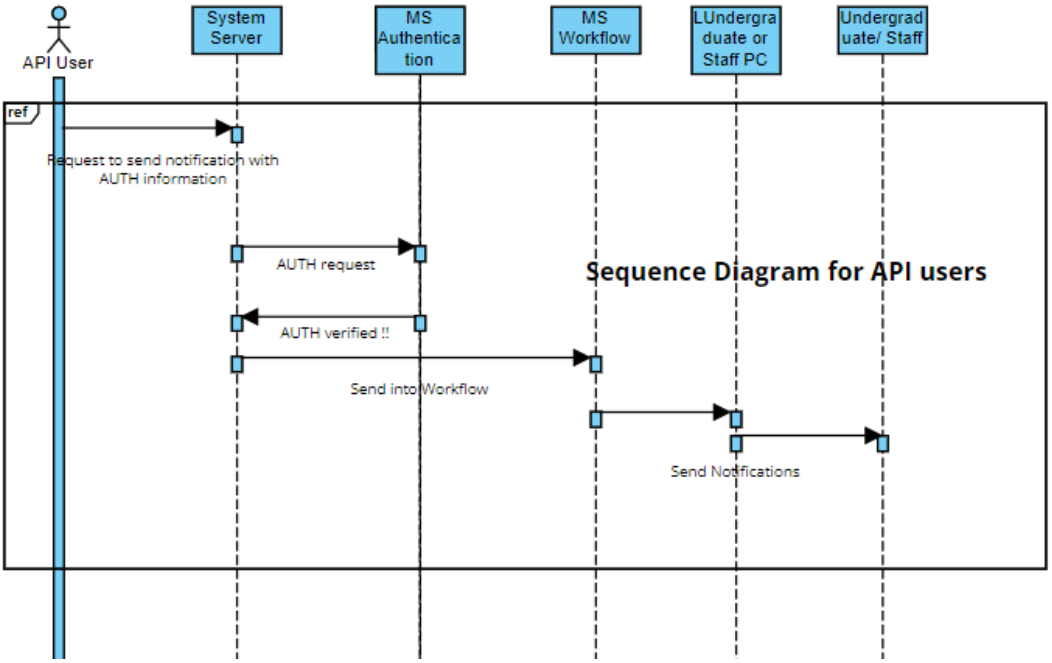


FIGURE 1 : SEQUENCE DIAGRAM FOR API USERS

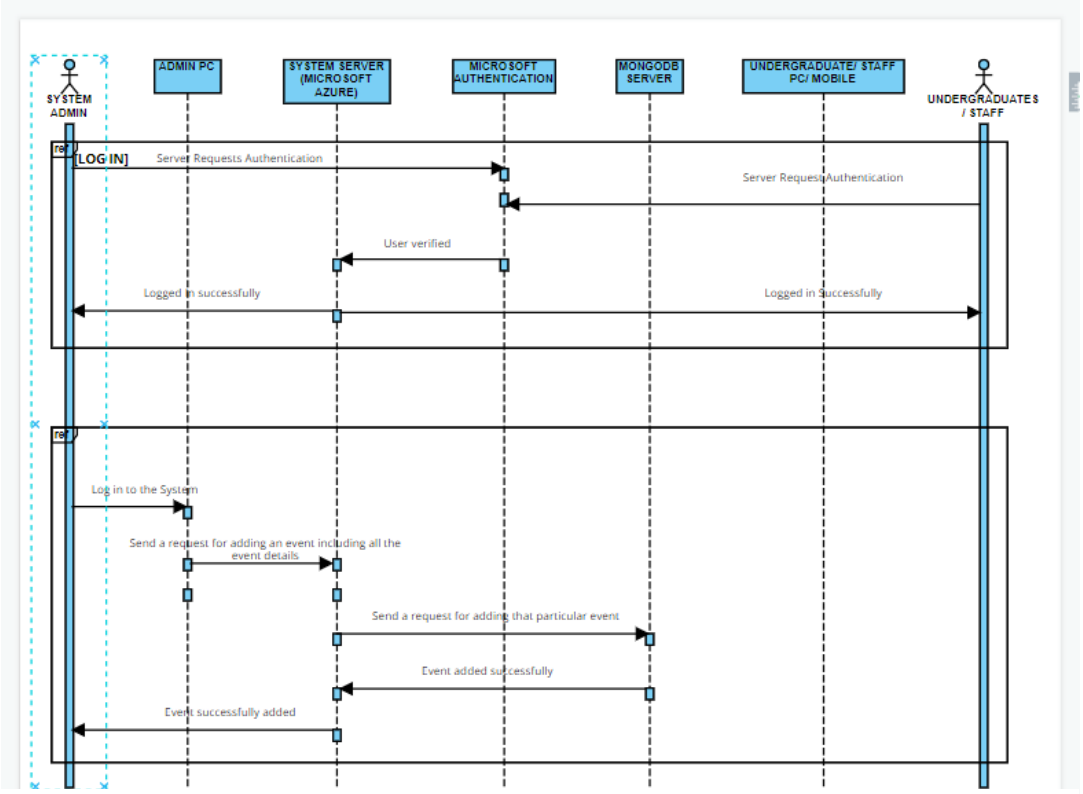


FIGURE 2 : SEQUENCE DIAGRAM FOR SIGN IN AND EVENT SCHEDULING(ADD)

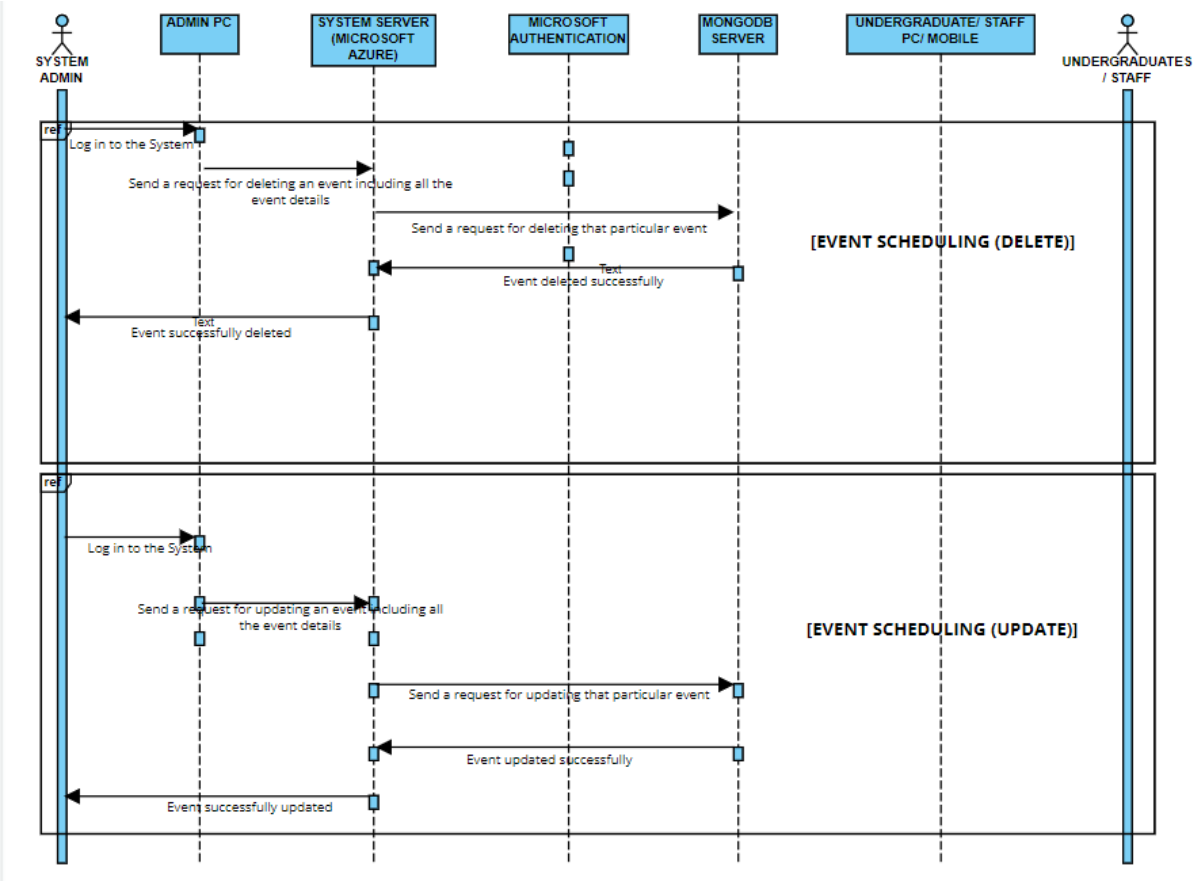


FIGURE 1 : SEQUENCE DIAGRAM FOR EVENT SCHEDULING(DELETE AND UPDATE)

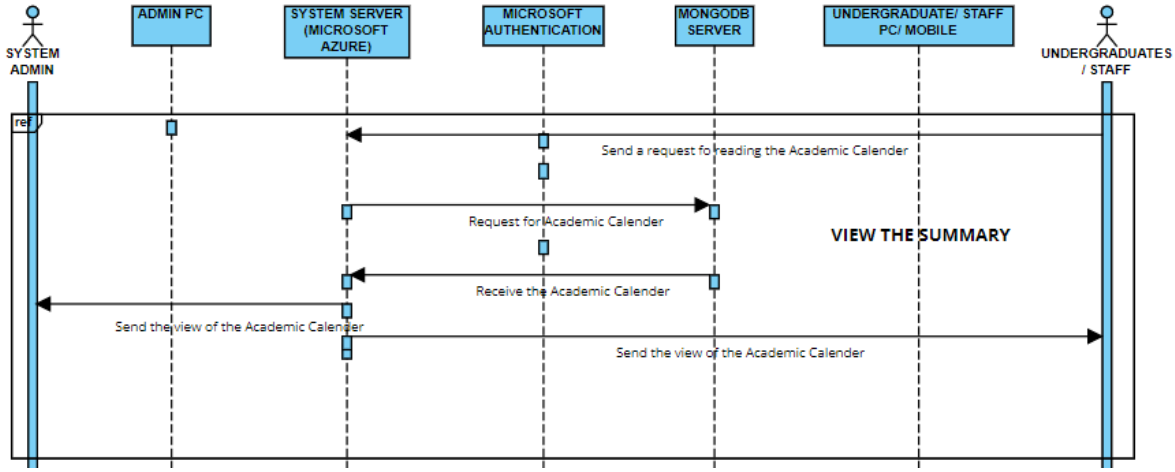
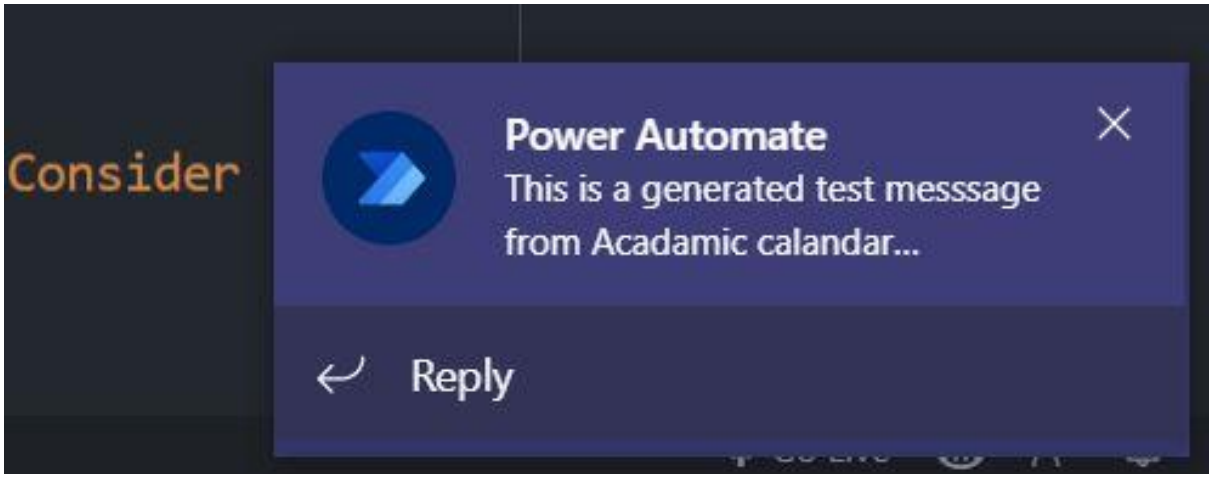


FIGURE 2 : SEQUENCE DIAGRAM FOR VIEW THE SUMMARY

TEST CASE RESULTS




```
PASS ./app.test.js (11.167 s)
  DB CRUD Operations
    √ GET -> /events --> array of events (4376 ms)
    √ POST -> /events/create -->  (204 ms)
    √ PUT -> /api/events/update/:id -->  (324 ms)
    √ DELETE -> /api/events/delete/:id -->  (312 ms)
  Notifications
    √ POST -> /api/notifications/sendnotifications -->  (4534 ms)

Test Suites: 1 passed, 1 total
Tests:       5 passed, 5 total
Snapshots:   0 total
Time:        11.282 s, estimated 12 s
Ran all test suites.
```