

# REQUIREMENTS ANALYSIS

BANNARI AMMAN INSTITUTE OF  
TECHNOLOGY

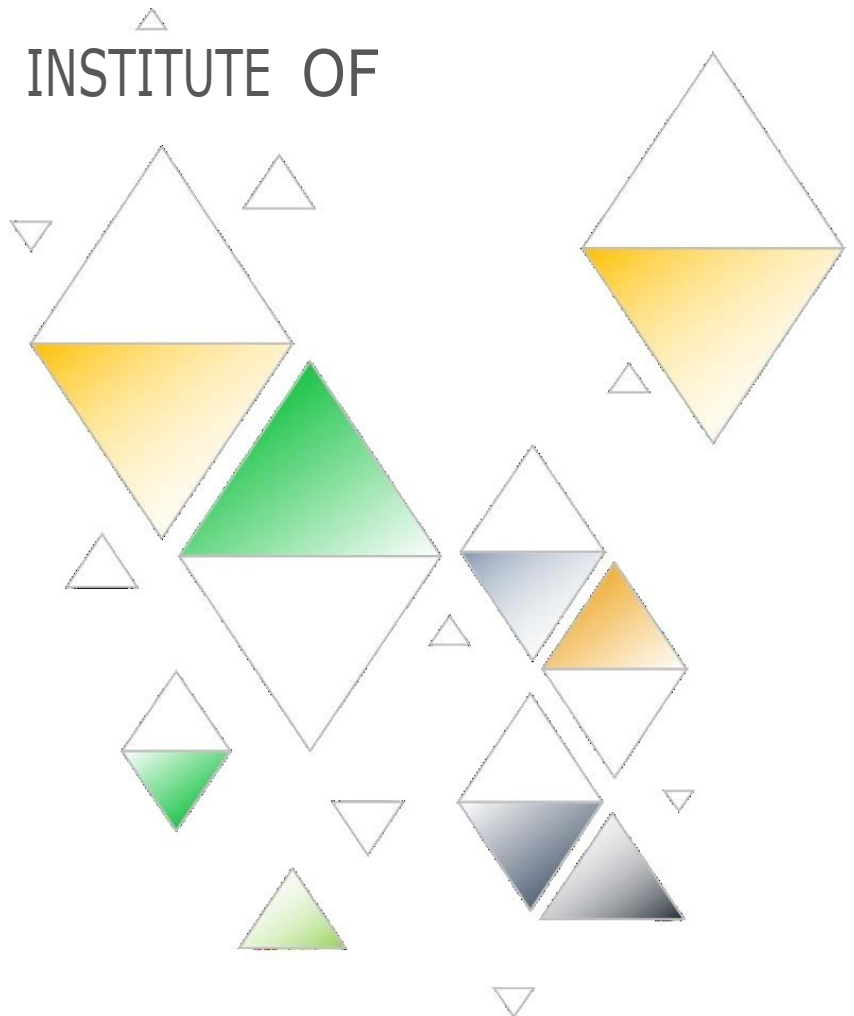
AUTHOR  
KIRUTHIKA DEVI A

DATE PREPARED  
12/07/2024 -01/08/2024

PROJECT TITLE

BIT MAILER

SEAT NUMBER 83



DOCUMENT AUTHORIZATION MEMORANDUM

BIT MAILER

---

Project Manager

NAME	SIGNATURE	DATE
KIRUTHIKA DEVI A		

Director

NAME	SIGNATURE	DATE

# TABLE OF CONTENTS

INTRODUCTION .....	4
Purpose.....	4
Operating Environment.....	5
Process Flow .....	5
DESCRIPTION .....	6
Product Context .....	6
User Classes and Characteristics .....	8
REQUIREMENTS .....	9
Performance Requirements.....	9
Security Requirements.....	10
Usability Requirements.....	10
Other Requirements .....	10
Entity A: Administrator.....	11
Entity B: Faculty .....	13
Entity C: Students.....	14
ER diagram .....	15

# INTRODUCTION

## Purpose

The BIT Mailer project aims to create a streamlined and efficient mailing system within educational institutions offering distinct functionalities for different user roles:

- ✓ Faculty can send mail requests to students through a centralized platform, ensuring all communication is organized and easily accessible.
- ✓ An admin verifies each mail request to prevent scheduling conflicts and ensure that communications are relevant and timely.
- ✓ By having an admin verify the mail requests, the system ensures that only appropriate and necessary communications are sent to students, maintaining a high standard of information flow.
- ✓ The platform fosters better coordination between faculty, students, and administrators, promoting a more collaborative and supportive educational environment.
- ✓ Administrators have access to overall mailing activity, allowing them to monitor and analyze communication patterns and effectiveness, contributing to continuous improvement of institutional processes.
- ✓ Utilizing modern web technologies, the BIT Mailer ensures that all communications and data are secure and private, adhering to institutional policies and regulations.

## Operating Environment

COMPONENTS	STACK	DESCRIPTION
Backend	Node.js and Express.js	Building the server-side application and handling API requests
Frontend	React	Building the user interface and interacting with the backend API.
Database	MongoDB	A NoSQL document-based database, for storing and retrieving data.
API	OpenAPI	OpenAPI is a language-agnostic specification for describing RESTful APIs

## Process flow

TASK NAME	START DATE	END DATE	DURATION (days)	STATUS
Requirement gathering	12-07-2024	12-07-2024	1	COMPLETED
Planning and documentation	13-07-2024	17-07-2024	5	COMPLETED
Entity flow	18-07-2024	19-07-2024	2	COMPLETED
Wireframes	20-07-2024	24-07-2024	4	COMPLETED
Design and prototyping				
Database design				
Backend development				
Testing				
Deployment				

# DESCRIPTION

Here are some additional things that can be included in the performance tracking application:

## 1. **Notification System:**

- **Push Notifications:** Send push notifications to students and faculty for important updates and reminders.
- **SMS Alerts:** Enable SMS alerts for critical messages or changes in the schedule.

## 2. **Advanced Conflict Resolution:**

- **Conflict Suggestions:** Provide alternative times or dates when a scheduling conflict is detected.
- **Real-time Conflict Updates:** Notify users immediately if a new conflict arises after an email has been scheduled.

## 3. **Integration with Other Platforms:**

- **Calendar Integration:** Sync with Google Calendar, Outlook, or other calendar systems to automatically update schedules.

## 4. **Feedback and Help Center:**

- **User Feedback System:** Implement a system for users to provide feedback and suggestions.
- **Help Center and FAQs:** Provide a help center with frequently asked questions and troubleshooting guides.

## User Classes and Characteristics

USER CLASS	CHARACTERISTICS
Student	<div><div><div></div><div><b>VIEW SCHEDULE</b></div></div><div><div>✓</div><div>Ability to view an up-to-date schedule of classes and events.</div></div><div><div>✓</div><div>Access to a real-time dashboard displaying their schedules and any upcoming events or activities.</div></div><div><div><div></div><div><b>RECEIVE EMAILS</b></div></div><div><div>✓</div><div>Receive conflict-free emails approved by the admin.</div></div><div><div>✓</div><div>View email content directly from the portal.</div></div></div></div>
Faculty	<div><div><div></div><div><b>SUBMIT MAILER REQUESTS:</b></div></div><div><div>✓</div><div>Access to a mailer request form.</div></div><div><div>✓</div><div>Ability to input mail content, scheduling time, category, and recipient details.</div></div><div><div>✓</div><div>Submit the request for admin review.</div></div><div><div><div></div><div><b>RECEIVE FEEDBACK:</b></div></div><div><div>✓</div><div>Receive notifications if a scheduling conflict is detected.</div></div><div><div>✓</div><div>Access to feedback from the admin about the conflict.</div></div><div><div>✓</div><div>Ability to adjust and resubmit the mailer request based on admin feedback.</div></div></div></div>

Administrators	<p><b>REVIEW MAILER REQUESTS:</b></p> <ul style="list-style-type: none"> <li>✓ Access to all submitted mailer requests.</li> <li>✓ Review requests for scheduling conflicts against the existing student schedules.</li> </ul> <p><b>APPROVE OR REJECT REQUESTS:</b></p> <ul style="list-style-type: none"> <li>✓ Approve mailer requests that have no scheduling conflicts, triggering the email to be sent to students.</li> <li>✓ Reject mailer requests that have conflicts, providing feedback to the faculty about the nature of the conflict.</li> </ul> <p><b>MANAGE SYSTEM OPERATION</b></p> <ul style="list-style-type: none"> <li>✓ Oversee the overall operations of the mailing system.</li> <li>✓ Manage user roles and permissions within the system.</li> <li>✓ Ensure smooth functioning and resolving any issues that arise in the process.</li> </ul>
----------------	--

### User Stories:

- As a student, I want to view a unified schedule of my classes and events to organize my day effectively.
- As a faculty member, I need to ensure my communications reach students without conflicting with their other scheduled activities.



# REQUIREMENTS

Include all of the requirements you collected from stakeholders.

Here are the key requirements for the performance tracking application in a concise form:

1. User Management (registration, authentication, roles)
2. Student portal (View Schedule, Receive Emails, Notification System)
3. Faculty Portal (Mailer Request Form, View Submission Status, Feedback Handling:)
4. Administrator Portal (Review Mailer Requests, Approve/Reject Requests, Manage System Operations, Analytics and Reporting)
5. Conflict Resolution
6. Notification and Alerts
7. Data Security and Privacy
8. Integration and Interoperability
9. Scalability and Performance
10. User experience and Accessibility
11. Maintenance and Support

## Performance Requirements

- Handle lots of users without slowing down
- Fast response times for dashboards and reports
- Real-time data updates
- Ability to grow and scale as needed

## Security Requirements

- Access control based on user roles
- Encryption of data in transit and at rest
- Secure authentication like multi-factor or single sign-on
- Regular security checks and testing
- Follow data privacy regulations

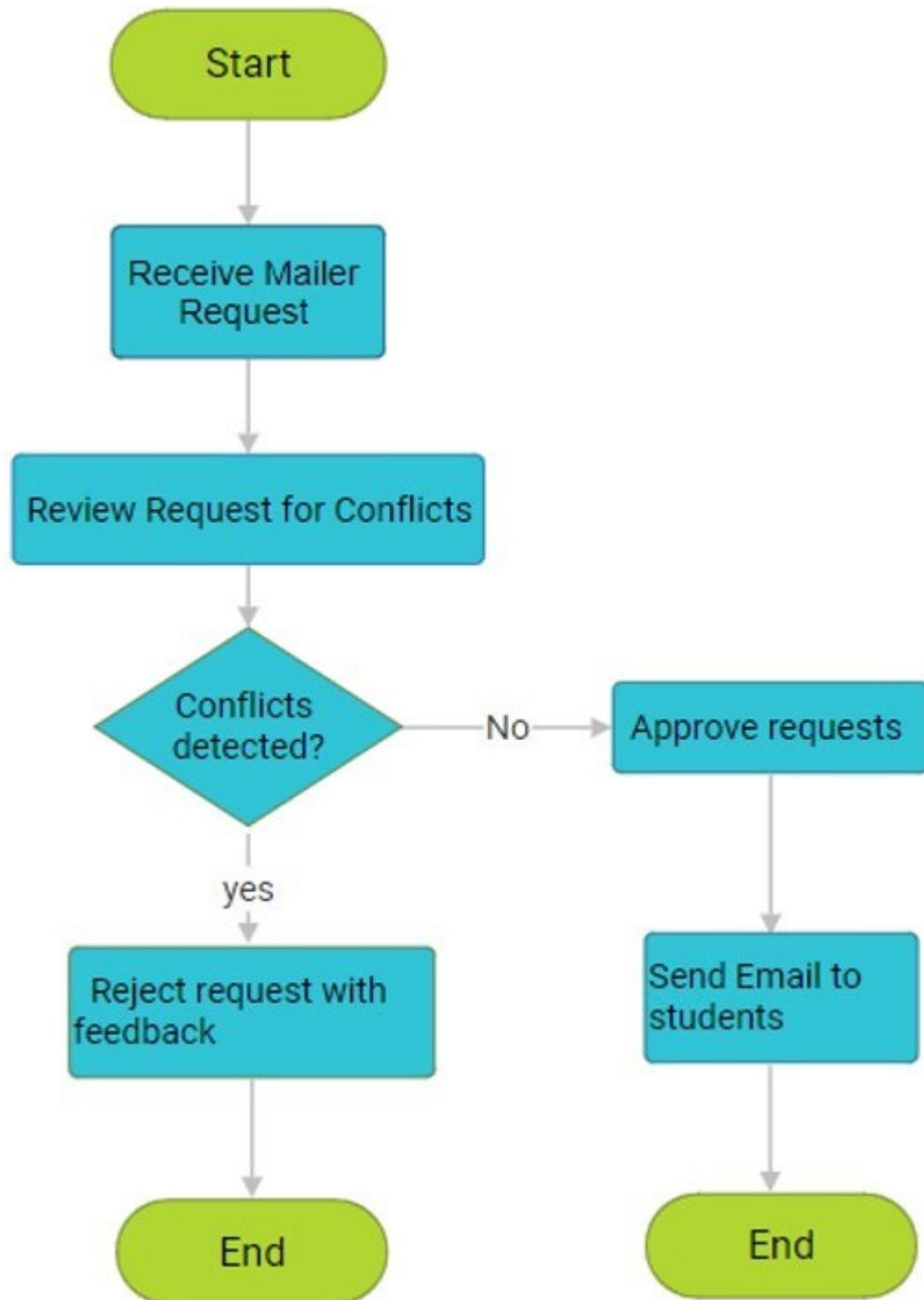
## Usability Requirements

- Easy to use interfaces for all users
- Support multiple languages
- User guides and training
- Consistent and appealing design
- Customizable dashboards and reports

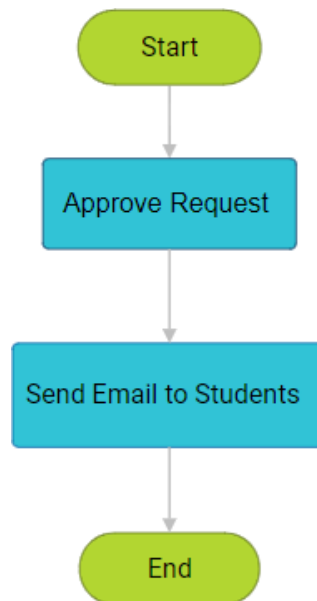
## Other Requirements

- Integrate with existing Email system
- Data backup and recovery
- Logging and error handling
- Flexible architecture for future changes
- Automated testing and deployment
- Follow best coding practices

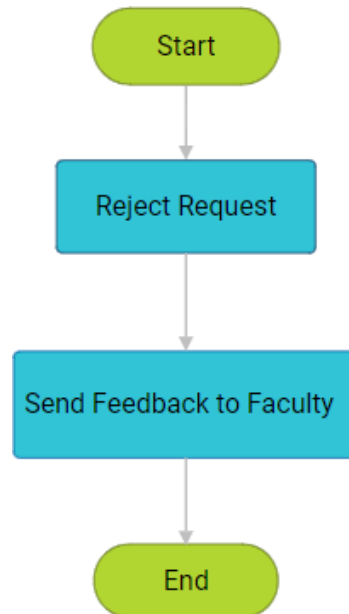
## ENTITY A: ADMINISTRATOR



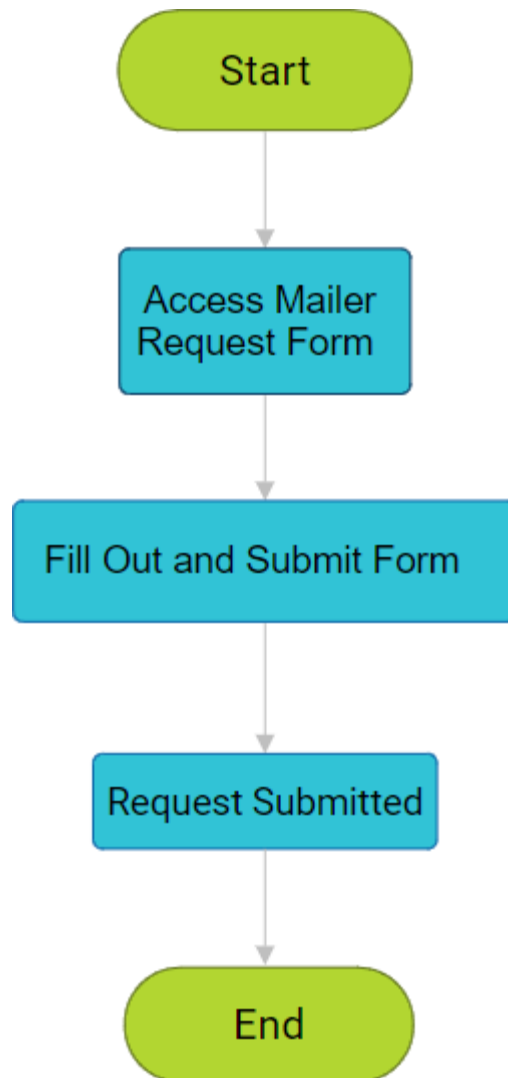
## ENTITY A: APPROVING REQUESTS



## ENTITY A: REJECTING REQUESTS



## ENTITY B: FACULTY



## ENTITY D: STUDENT

