الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

ECOLE NATIONALE SUPÉRIEURE D'INFORMATIQUE

المدرسة الوطنية العليا للإعلام الآلي
(المعهد الوطني للتكوين في الإعلام الآلي سابقا)
Ecole nationale Supérieure d'Informatique
ex. INI (Institut National de formation en Informatique)

SCI Project Report
2nd Year Higher Cycle (2CS)
Option : Computer Systems (SQ)

NotifyTrack - Smart IoT Notification Gateway for ESI

Directed by :

- BOUKHETALA Zaineb
- KHELLAS Yacine
- REMIL MahaFatimaZahra
- KHADIR Amina
- HENNANE DouaaElIkhlas
- BENYAHIA Yahia

Supervised by :

- Mr. SEHAD Abdennour

Academic year : 2024 - 2025

Table des matières

Table des matières **I**

Table des figures **II**

General Introduction III
Problem Statement III
Proposed Solution III
System Architecture IV
Implementation Detailed V
General Conclusion VII
Références VII

Bibliographie **VIII**

Table des figures

1 NotifyTrack System Architecture IV
2 Successful SMS Alert Notification V
3 Node-RED Dashboard for Real-Time Sensor Monitoring VI
4 Web Interface for Notification Management VI

1. Introduction

In modern educational institutions, effective communication is essential to ensure smooth academic and administrative operations. Students and staff rely on the timely delivery of key information such as exam schedules, assignment deadlines, meetings, and emergency alerts. The fast pace of academic life requires communication systems that are both reliable and immediate.

2. Problem Statement

In our environment (oued semar, university residences and in a lot of other places), unstable or limited internet connectivity often hinders the effectiveness of traditional communication methods. This can lead to delays or failures in delivering important messages, resulting in missed deadlines, disorganization, and slow responses to urgent situations. The lack of a dependable and continuous communication system at ESI for example creates significant challenges that affect student engagement, staff coordination, and the overall efficiency of campus operations.

3. Proposed Solution

To address these challenges, we propose a Smart Notification Gateway System that delivers real-time SMS alerts for both personal and institutional events. This system integrates multiple data sources to ensure timely and actionable notifications for students, faculty, and staff.

Key features include :

— **Google Calendar integration :** SMS reminders for scheduled events.

— **Automated facility monitoring :** IoT sensors track environmental conditions, triggering alerts for anomalies.

— **Emergency alert system :** Instant notifications for security threats, power failures, or medical emergencies.

— **Transportation updates :** Real-time bus (kous) notifications for schedule changes and delays.

This system enhances safety, efficiency, and communication, ensuring reliable notifications even in areas with poor internet connectivity.

4. System Architecture

NotifyTrack is an IoT-driven smart notification system that leverages **edge computing** and **cloud services** to automate SMS alerts for both **event scheduling** and **environmental monitoring**. At its core, a Raspberry Pi 4 acts as the **central controller**, orchestrating data processing, SMS transmission, and IoT communications.

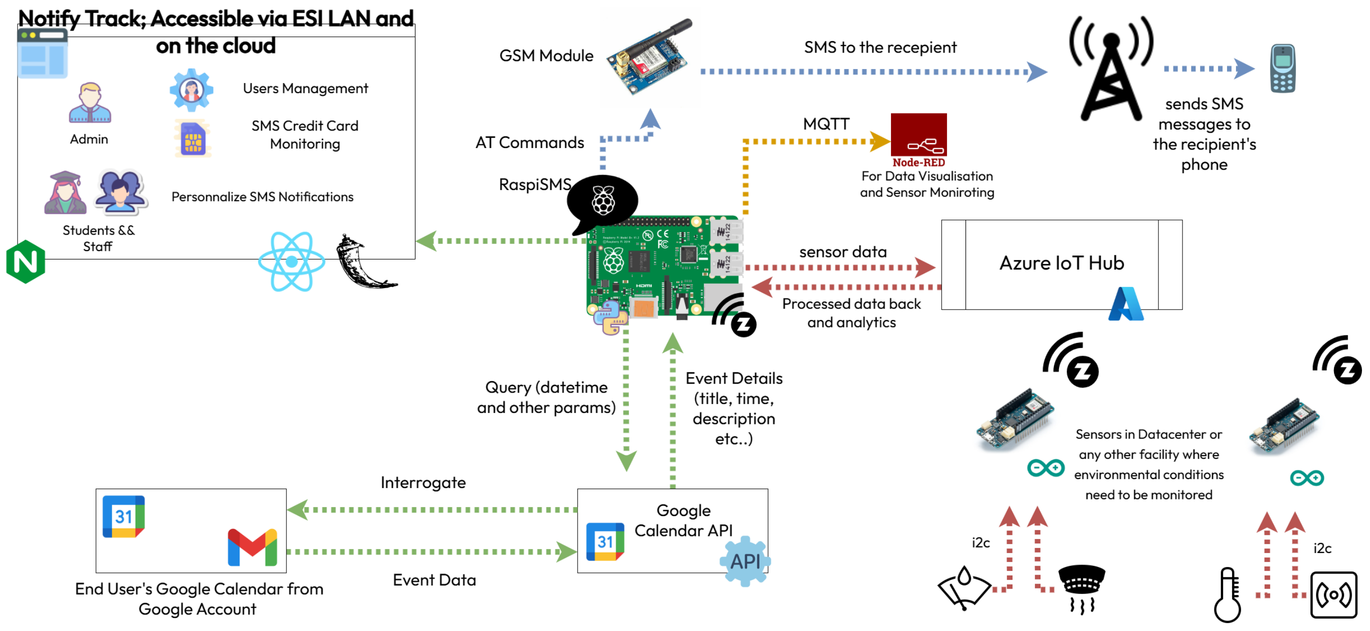

Notify Track; Accessible via ESI LAN and on the cloud
Users Management
Admin
SMS Credit Card Monitoring
Personnalize SMS Notifications
Students && Staff
GSM Module
SMS to the recepient
sends SMS messages to the recipient's phone
AT Commands
MQTT
Node-RED
For Data Visualisation and Sensor Moniroting
RaspiSMS
sensor data
Azure IoT Hub
Processed data back and analytics
Query (datetime and other params)
Event Details (title, time, description etc..)
Sensors in Datacenter or any other facility where environmental conditions need to be monitored
Interrogate
Google Calendar API
API
End User's Google Calendar from Google Account
Event Data
i2c
i2c

FIGURE 1 – NotifyTrack System Architecture

The system is modular and consists of 2 primary functions :

4.1. User Management and Event Scheduling

The event scheduling module ensures that students and faculty are **always informed about important events**.

Each **user authorizes** NotifyTrack via OAuth to grant limited access to their Google Calendar. The system then **retrieves event details** (lectures, exams, meetings) and matches them with stored user contact information.

Scheduled events are processed and queued in the **RaspiSMS database**, where RaspiSMS, built on Gammu, executes **AT commands** to control the SIM800L GSM module. The message is then dispatched at the scheduled time.

Through a user-friendly web interface, individuals can **customize their notification settings**, such as early reminders, daily summaries, or urgent alerts. An **admin panel** is also available for managing notifications and **monitoring the GSM module's status**.

4.2. IoT-Based Environmental Monitoring

This module ensures **safety conditions** in key areas (e.g., data centers, labs, and classrooms) are **actively monitored**.

A network of **Arduino-based sensors** measures temperature, humidity, and air quality, transmitting data to a Raspberry Pi.

Node-RED processes sensor readings, and a periodic check triggers **RaspiSMS alerts** if thresholds are exceeded (e.g., overheating).

Sensor data is stored in **Azure IoT Hub** for **historical analysis, advanced analytics, and scalability**, ensuring efficient monitoring for large-scale deployments.

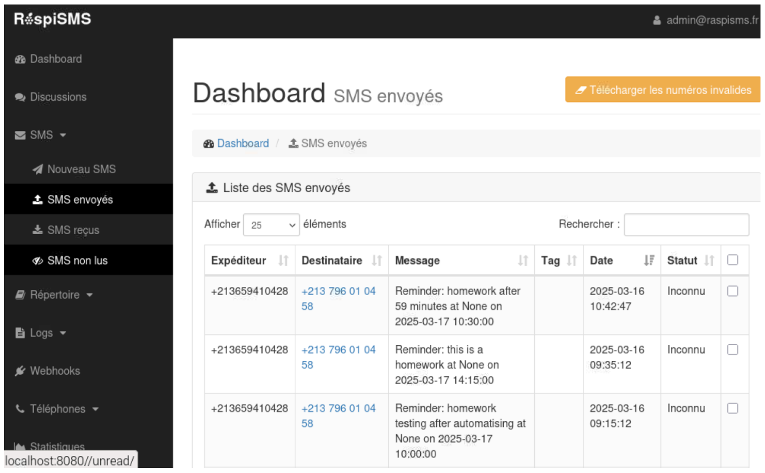
5. Implementation Details

5.1. System Setup

The NotifyTrack system is built around a **Raspberry Pi 4**, which acts as the central controller for SMS alerts and environmental monitoring. The system runs :
— **Flask API** for event scheduling and system control.
— **RaspiSMS** for SMS automation via the **SIM800L GSM module**.
— **Node-RED** for real-time IoT data processing and visualization.

5.2. Event Scheduling and SMS Notifications

The system integrates with **Google Calendar API** via OAuth to retrieve scheduled events. Events are stored in a local database and queued in **RaspiSMS**, which triggers SMS alerts using **Gammu** commands. Users can manage their notifications via a web dashboard, while administrators monitor SMS transmission status.



Dashboard SMS envoyés

Liste des SMS envoyés

Afficher 25 éléments

Rechercher :

Expéditeur	Destinataire	Message	Tag	Date	Statut	☐
+213659410428	+213 796 01 04 58	Reminder: homework after 59 minutes at None on 2025-03-17 10:30:00		2025-03-16 10:42:47	Inconnu	☐
+213659410428	+213 796 01 04 58	Reminder: this is a homework at None on 2025-03-17 14:15:00		2025-03-16 09:35:12	Inconnu	☐
+213659410428	+213 796 01 04 58	Reminder: homework testing after automatising at None on 2025-03-17 10:00:00		2025-03-16 09:15:12	Inconnu	☐

FIGURE 2 – Successful SMS Alert Notification

5.3. Environmental Monitoring System

A network of **Arduino-based sensors** (temperature, humidity, air quality) sends real-time data to the Raspberry Pi . **Node-RED** processes these readings and triggers SMS alerts if thresholds are exceeded.

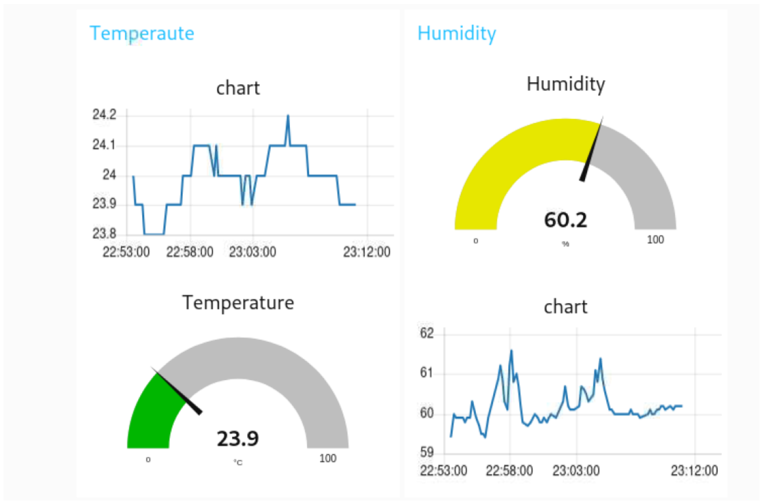
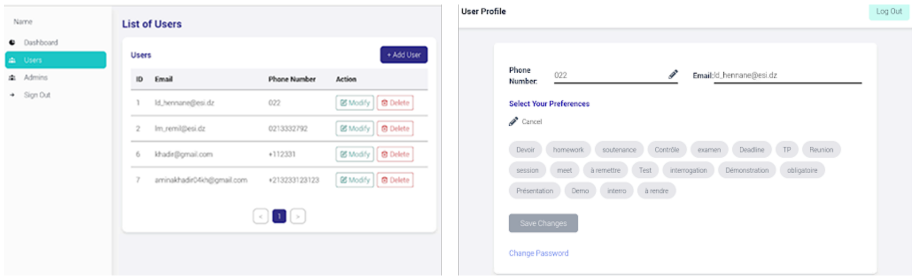


FIGURE 3 – Node-RED Dashboard for Real-Time Sensor Monitoring

5.4. Web Interface and System Automation

The **React.js frontend** enables users to customize notifications and view system status. It interacts with the **Flask backend**, which communicates with **RaspiSMS** for SMS scheduling and **Azure IoT Hub** for storing environmental data.



List of Users

Users

ID	Email	Phone Number	Action
1	ld_hennane@esi.dz	022	Modify Delete
2	lm_remil@esi.dz	0213332792	Modify Delete
6	khadir@gmail.com	+112331	Modify Delete
7	aminakhadir04kh@gmail.com	+213233123123	Modify Delete

User Profile

Phone Number: 022

Email: ld_hennane@esi.dz

Select Your Preferences

Devoir, homework, soutenance, Contrôle, examen, Deadline, TP, Reunion, session, meet, à remettre, Test, interrogation, Démonstration, obligatoire, Présentation, Demo, interro, à rendre

FIGURE 4 – Web Interface for Notification Management

5.6 Final System Testing

To validate system performance, the following tests were conducted :

- **Event Scheduling Test** : Verified correct SMS triggering for scheduled events.
- **Sensor Monitoring Test** : Ensured alerts were sent when environmental conditions exceeded thresholds.
- **User Interface Test** : Checked responsiveness and usability of the web dashboard.

6. Conclusion

In today's educational environment, reliable communication is essential. This project tackled unreliable internet connectivity by developing a **Smart Notification Gateway System**, leveraging SMS for real-time notifications. By integrating IoT devices, a web dashboard, and cloud-based analytics, the system ensures timely delivery of critical information, even in areas with poor connectivity.

The system combines a **GSM module** for SMS alerts, a **RaspiSMS platform** for event management, a **Node-RED dashboard** for real-time monitoring, and **Azure IoT Hub** for data storage and analytics. These components provide a scalable solution for environmental monitoring, emergency alerts, and exam reminders, enhancing communication efficiency.

Currently supporting a single sensor, the system is designed to scale by integrating **Arduinos with Zigbee** for multi-sensor data processing. The **Azure IoT Hub** allows future implementation of **machine learning** for predictive analytics, further expanding its capabilities.

This project showcases the power of IoT and cloud computing in improving communication reliability and lays the foundation for future advancements in academic settings.

Bibliographie

[1] https://nodered.org/docs/

[2] https://tailscale.com/kb

[3] http://documentation.raspisms.fr/

[4] https://www.instructables.com/Raspberry-Pi-Tutorial-How-to-Use-the-DHT-22/

[5] https://help.sia-connect.com/en_US/azure-iot-hub-device-provisioning-service/monitoring-iot-hub-events-messages-&-metrics

[6] https://learn.microsoft.com/en-us/azure/iot-hub/

[7] https://developers.google.com/calendar/api/guides/overview