SCAN & GO

Introduction

Architectural Diagram

Research

Project Plan

Summary

INTRODUCTION

Ollscoil Teicneolaíochta an Atlantaigh

Atlantic Technological University

Cian O'Donnell
G00358872
BEng Software & Electronic Engineering

Description

Scan & Go is a radio-frequency identification (RFID) door access project that can be implemented in manufacturing, healthcare, and other sectors of the economy.

Door access is granted or denied based on an employee's job role and the job requirements of the room being accessed.

All employee scans are logged to a database, these logs can be viewed on a web application.

New employees are added to the system through the web app. Employee information is also edited or deleted through the web application.

Technologies

The **MKR1010 Wi-Fi board** handles Wi-Fi connections with the aid of its built-in ESP32 chip. The on-board LED signals access granted (green) or access denied (red) to the user.

The **MFRC-522** RFID card reader is connected to the Wi-Fi board, which will receive card IDs when a card is scanned.

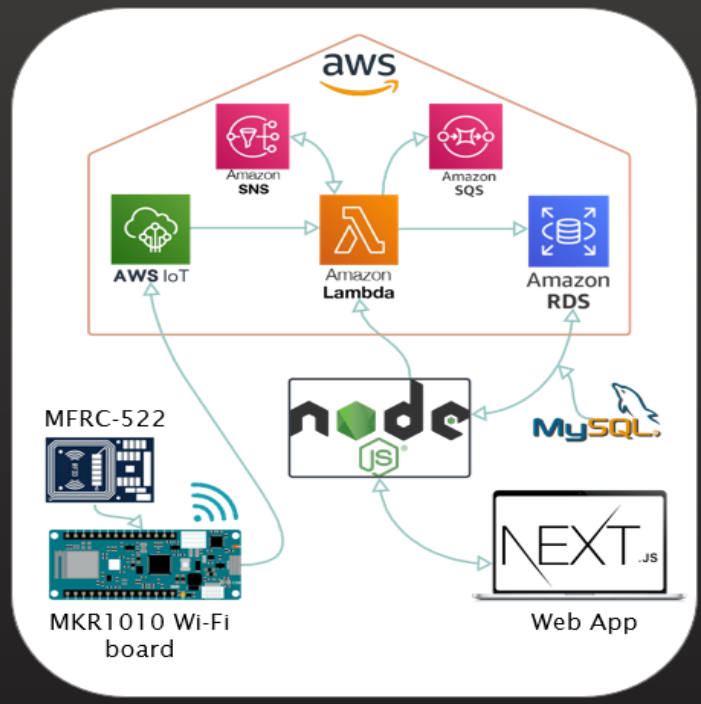
Amazon Web Services (AWS):

- **IoT Core** is used to receive published card data from the Wi-Fi board.
- **Lambda** functions are used to receive card data from IoT Core and send the RFID card data to a relational database (**RDS**).
- **Simple Notification Service (SNS)** and **Simple Queue Service (SQS)** are used as a way of queuing and pushing all data received in the lambda function, ensuring all data gets sent to the database.
- **RDS** is used as the cloud hosted server for the database.

MySQL is the relational database that holds all employee information and door access tables.

NodeJS is used for the Lambda function, connecting to the RDS database, and writing the backend code for the web app.

Next.js is a React framework that's used to create a full stack web application.



Approach

The approach was to have RFID card IDs read from the card reader and have these card IDs added to a database. The card IDs will then be assigned to employees so they have their own employee badge.

Assigning cards and adding, editing or deleting employee information will be done through the web application.

Employees can then scan their badge and they will be granted or denied access to a room depending on their job role. All card scans will be logged to a database and these logs can be accessed on the web app.

Results

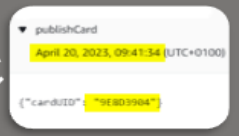
Finding a means to publish data from the Wi-Fi board into cloud services proved difficult throughout this project. Google Cloud was not compatible with the Wi-Fi board. For this reason, AWS was the cloud service chosen.

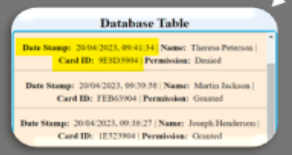
Their SNS and SQS services were also essential in ensuring that all data received from the IoT core was seamlessly delivered into the database.

This procedure is displayed below:

An RFID card is scanned. This employee will be denied access.

Card ID is published to the IoT core with a timestamp.





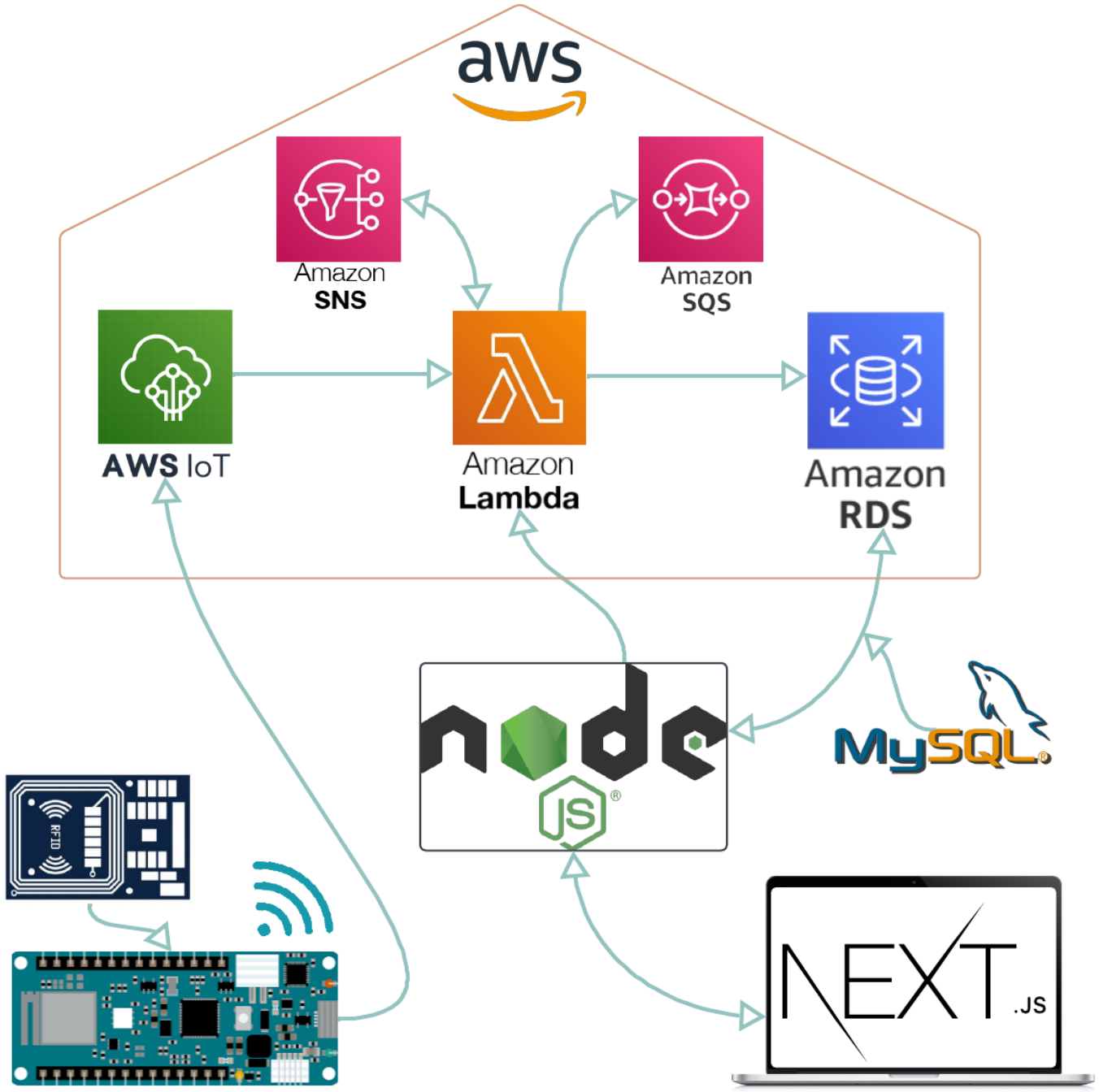
Scan has been logged in the database with employee information and door permission for their employee badge.

Conclusion

Scan & Go has delivered on its requirements. A system has been created to read employee card scans when their RFID cards are scanned. Logs of these card scans are saved to a database and can be viewed on the web application.

Employee information can be successfully added, edited or deleted through the Scan & Go web application.

Employee information is able to be edited or deleted through the web app.



ARCHITECTURAL DIAGRAM

Technologies Used

Hardware:

- MKR WiFi 1010
- MFRC/RC-522 Module

Software:

- Amazon Web Services (AWS)
- Node.js
- Next.js
- Arduino IDE

Language:

- JavaScript
- C++
- CSS

RESEARCH

Hardware			Database			Cloud Service		
ESP 32	VS	MKR WiFi 1010	Mongo	VS	MySQL	GOOGLE	VS	AWS

ESP32

- Compatible with RFID module
- Compatible with Cloud IoT Services

MKR WiFi 1010

- Encryption available
- Compatible with RFID module
- Compatible with Cloud IoT services
- On-board LED

Choice:
MKR WiFi 1010

MongoDB

- Shows data in JSON documents
- Collections
- Not Supporting joining tables
- No need to define schema
- Cost effective

MySQL

- Used in manufacturing Industries
- Schemas
- Supports joining tables
- Shows data in rows and columns
- Structured data

Choice:
MySQL

Google Cloud

- Previous experience with google services
- Supports IoT, RDS services, cloud functions
- Authentication setup

Amazon Web Services

- Compatible with the MKR WiFi 1010 board
- IoT Core, RDS, Hosting web app, Lambda functions, SNS and SQS services

Choice:
AWS

PROJECT PLAN

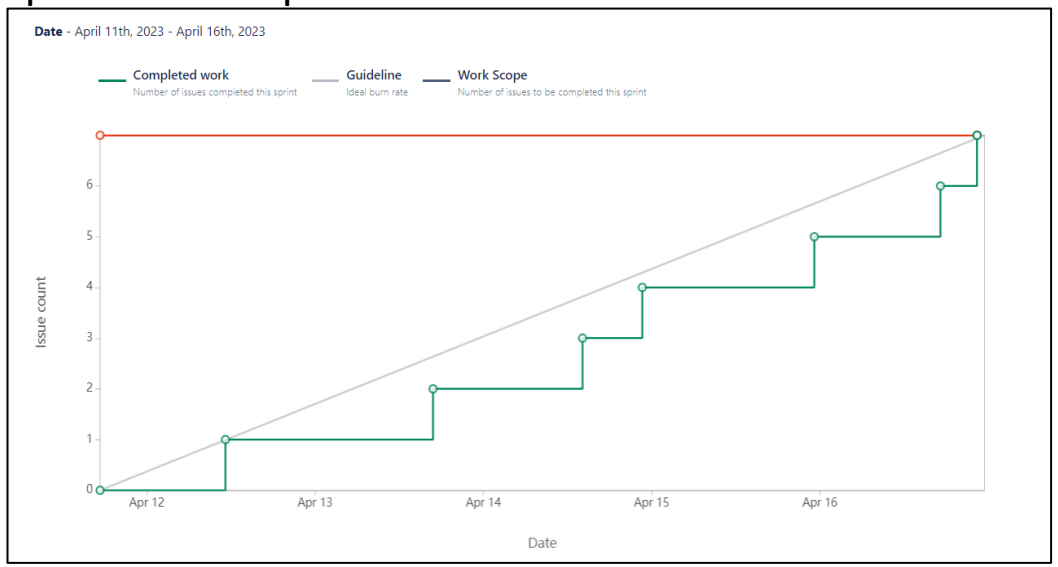
JIRA

Roadmap:

Cumulative Report:

Sprint: Poor Sprint

Sprint: Good Sprint



SUMMARY

Challenges

- Updating RFID card states between unregistered, denied and accepted
- Cloud services
- Project management

Project achievements

- Encrypted WiFi board sending RFID card data securely
- Authenticated login and protected routes on the web app
- Web app deployed securely on AWS
- Web app with functioning Creating, Read, Update, Delete (CRUD) operations

THANK YOU

https://github.com/CianODonnellGIT/FYP