

# TalentElite-Advanced Employee Management System

Individual Project

Author : Vincent 39226719

Language:  
Python C++ Javascript Go

23/24: LZPMPC005M: Programming Clinic

May 26, 2024

# Overview

- 1 Introduction
- 2 System Strcuture
- 3 Methodology
- 4 UML diagram
  - Class Diagram
  - Usecase Diagram
  - Sequence Diagram
    - Go Auth Sequence Diaram
    - C++ database backup Sequence Diaram
    - Django content Sequence Diaram
    - Express.js Email Sequence Diaram
    - Go API logging Sequence Diaram
- 5 Database Security and API operation logging

# Introduction

This is a corporate employee management system that includes login authentication, employee information display, logging, email mass mailing, and database backup functionalities. The project employs various front-end and back-end technologies to achieve a comprehensive and user-friendly enterprise management solution.

For further informations please visit [TalentElite:Github](#), and if you have suggestions and any comments please do not hesitate to contact at [y.cheng22@lancaster.ac.uk](mailto:y.cheng22@lancaster.ac.uk).

# System Strcuture

See below

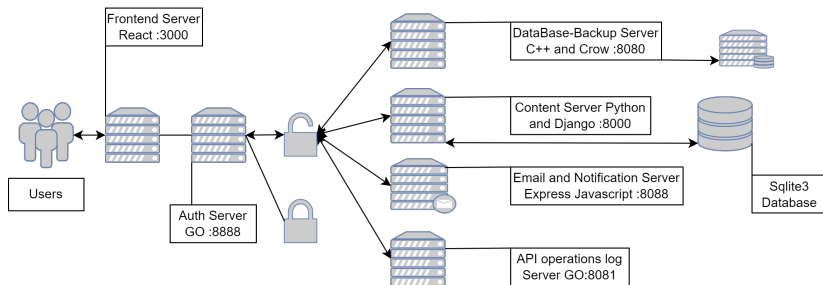


Figure: System structure.

# Methodology

- Go (Golang) Back-end:  
Handles user login authentication, generates and returns JWT tokens. Provides logging API, recording all accesses to the Django API. Main technologies: gorilla/mux for routing, golang-jwt/jwt for JWT token generation and verification.
- Django Back-end:  
Manages corporate employee information (CRUD operations). Provides API endpoints for front-end access, including employee information and department information. Used for logging and backup functionalities. Main technologies: Django framework, Django REST framework, JWT authentication.
- C++ Back-end (Crow):  
Handles the database backup functionality. Periodically backs up the Django project's database and stores the backup files in a specific directory. Main technologies: C++ language, Crow library (for simple HTTP server), file operations.
- Express.js Back-end:  
Handles the email mass mailing functionality. Sends emails to all employees via an SMTP server. Main technologies: Node.js, Express framework, nodemailer library.
- Front-end Technologies React Front-end:  
Used to build the user interface, displaying login page, employee information pages, etc. Communicates with the back-end via Axios, sending and receiving data. Main technologies: React framework, react-router-dom for routing management, Axios for HTTP requests.
- Semi Design UI Library:  
Used to beautify the front-end interface, providing a set of modern UI components. Main technologies: @douyinfe/semi-ui library, @douyinfe/semi-icons library.

# UML diagram

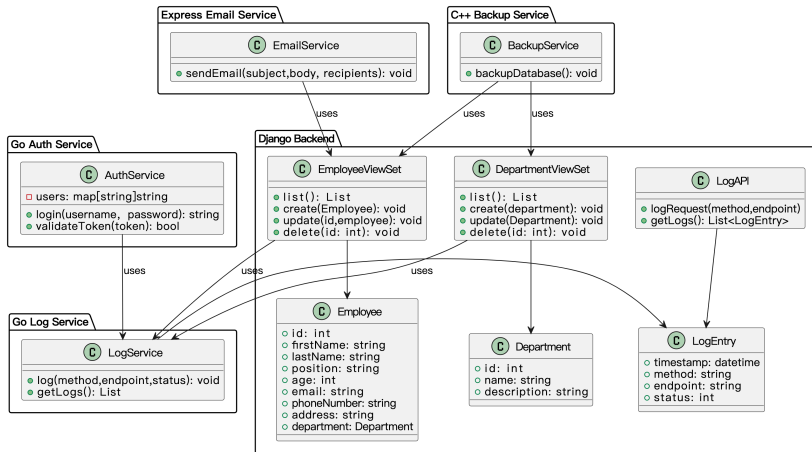


Figure: Class Diagram.

# UML diagram

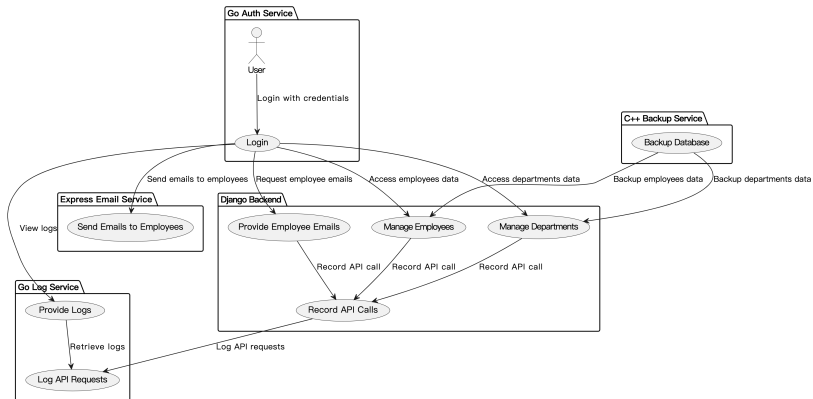
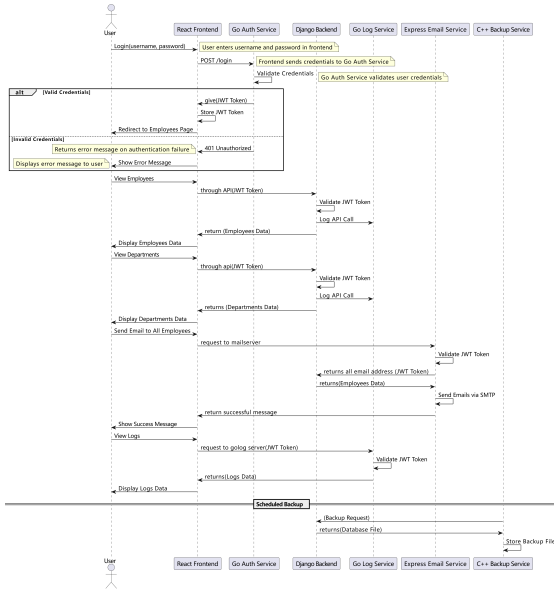


Figure: Usecase Diagram.

# UML diagram





# UML diagram

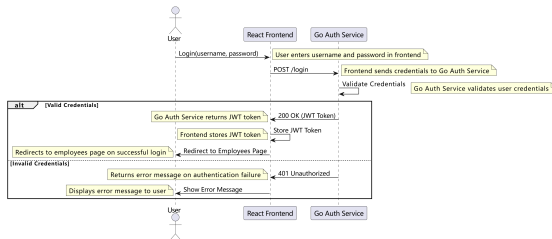


Figure: Go Auth Sequence Diagram.

# UML diagram

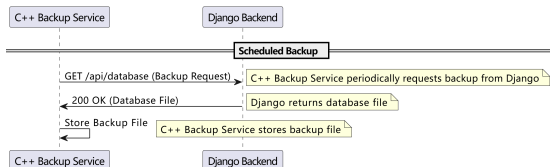


Figure: C++ database backup Sequence Diagram.

# UML diagram

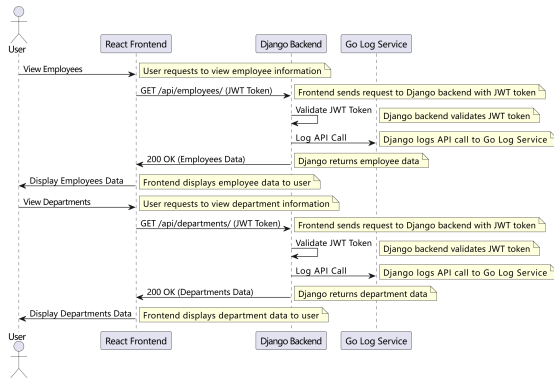


Figure: Django content Sequence Diagram.

# UML diagram

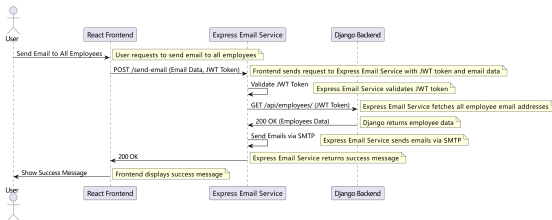


Figure: Express.js Email Sequence Diagram.

# UML diagram

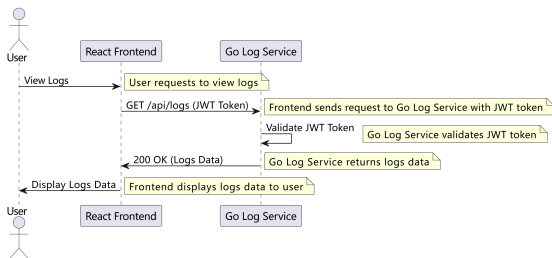


Figure: Go API logging Sequence Diagram.

# Database Security and API operation logging

## Heading

- 1 Database Backup
- 2 API Operation Logging

- C++ Crow web server for excute Database backup.(no worry database been hack)
- GO server recording API operations and time ,locate whos edit data.

# The End