**ITECH 7201**

**Assignment 2**

**Part B - Group Task**

**Human Resource Management System**

**Employee Management System**

**Leave Management System**

**Attendance Management System**

# Details of Team members

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Student ID** | **Name** | **Sub system responsible for** | **Contribution for Part B** | **Color code** |
| 30432670 | Syeda Tamanna Sheme | Employee  Management System | 33.34% |  |
| 30437681 | Tanvir Iqbal | Leave management System | 33.34% |  |
| 30423990 | Hansi Nipunika  Panwillaarachchi  Kotudura Bandanage | Attendance  Management System | 33.34% |  |

Table of Contents

[Details of Team members 2](#_Toc22237)

[1. Introduction 4](#_Toc22238)

[1.1. System Scenario 4](#_Toc22239)

[1.2. Overview of Purpose 5](#_Toc22240)

[1.3. Overview of Scope 6](#_Toc22241)

[2. System Analysis and Design Background 7](#_Toc22242)

[3. Consolidated class diagram 7](#_Toc22243)

[4. Justification of Classes 10](#_Toc22244)

[4.1. Class Justification 10](#_Toc22245)

[4.2. Identify the subsystem for the attributes and classes 14](#_Toc22246)

[5. Codes for Class definition and method definition 16](#_Toc22247)

# Introduction

The assignment involves developing a Human Resource Management System (HRMS) to centralize and automate HR functions such as employee management, leave management, and attendance management. In Part A, we focused on the Employee Management System, defining the system scenario and purpose, detailing its scope, and developing the domain model. This included creating classes like Employee, Supervisor, and HRManager, along with their relationships and functionalities. We also implemented code for key operations such as creating, updating, and viewing employee details. This work sets the foundation for integrating other subsystems into a comprehensive HRMS.

In this assignment, we will implement the system design created in the first assignment, focusing on our designated subsystems. Each team member will translate use case diagrams, class diagrams, and sequence diagrams into actual Java code for their subsystem. We will incorporate interface and inheritance implementation, as well as design patterns like singleton and state, to enhance the system's structure and efficiency. Additionally, we will develop comprehensive unit tests to validate the functionality of our subsystems and participate in a peer code review process to ensure code quality and adherence to standards. The implementation will be managed using GitHub, ensuring proper version control and collaboration within the team.

## Functionality

## Overview of Purpose

## Overview of Scope

System Analysis and Design Background

# Consolidated class diagram

# Justification of Classes

## Class Justification

### 