# 1. Project Overview (1 Paragraph)

## Team:

Group Members: Calvin White, Pratham Snehi

Group Number: 12

## Systems purpose:

The main goal of this project is to create a employee system where it keeps a list of employees full-time, part-time, and interns. As well as allowing you to create, delete, and get details of the employees.

## Key functionalities:

Creating a company that has employees that can be full-time, part-time, or interns. Using a switch case it will walk you through the employee creating process. You can change the salary amounts and if it’s negative it will prompt you again for a positive number.

# 2. Class Descriptions

## Interface: Employee

Attributes:

Methods:

public double calculateSalary();

public void displayDetails();

## Enum: Jobtypes

Attributes:

FullTime

PartTime

Intern  
Methods:

## Abstract Class: AbstractEmployee

### Attributes:

protected String employeeName

protected JobTypes jobType

protected double basePay

static int totalEmployees

protected int employeeNumber  
Methods:

public void displayDetails()

## Class: FullTimeEmployee

### Attributes:

 double bonusAmount

employeeName

jobType

bonusAmount

basePay

employeeNumber

### Methods:

public FullTimeEmployee(String name, JobTypes jobType, double basePay, double bonusAmount)

 public double calculateSalary()

## Class: PartTimeEmployee

### Attributes:

 private double hoursWorked

employeeName

jobType

hoursWorked

basePay

employeeNumber  
Methods:

 public PartTimeEmployee(String name, JobTypes jobType, double basePay, double hoursWorked)

public double calculateSalary()

## Class: InternEmployee

### Attributes:

employeeName

jobType

basePay

employeeNumber  
Methods:

public InternEmployee(String name, JobTypes jobType, double basePay)

public double calculateSalary()

## Class: Company

### Attributes:

private ArrayList<Employee> employeeArray;  
Methods:

public Company ()

public void showEmployees ()

public void addEmployee (String employeeName, int jobSelection, double basePay)

public void deleteEmployee (int employeeNumber)

## Class: Main

### Attributes: Methods:

main()

# 3. Implementation Details (Short Summary)

When you run the main you enter a loop where you can select choices using a switch case. You can create employees, change salary’s, view employee details, or exit the program. Within the choices, for example creating employee, it will prompt you for your name, type of employee, and pay. Then you can choose to do one of the other options like view employees, change salary, or exit

# 4. Test Cases (Table Format)

|  |  |
| --- | --- |
| Test Scenario | Expected Outcome |
| Create a full-time employee with a bonus | Salary = base + bonus |
| Create a part-time employee with hourly pay | Salary = hours × rate |
| Create an intern with a fixed stipend | Salary = stipend amount |
| Store multiple employees in a list | List holds all employees |
| Retrieve and display employee details | Name, Job Type, Salary appear correctly |
| Attempt to set a negative salary | Throws an error or sets to minimum valid value |
| Remove an employee from the list | Employee is deleted successfully |
| Handle incorrect input for job type | Displays an error message |
| \*Add a new employee type (e.g., Freelancer) | \*System supports new employee category |
| \*Increase an employee's salary | \*Salary updates correctly |
| \*=bonus | \*=bonus |

# 5. Challenges & Solutions (Brief Explanation)

Working with Enum was new to us so it was a learning experience because Enum’s are declared in a different way than other variables.

# 6. Conclusion & Future Enhancements (Optional)

We wanted to add a GUI this time around but found it difficult with the amount of time we had left on the project, so instead of we went with the switch case but next time we would like to do a GUI.