SUBJECT-

OBJECT ORIENTED PROGRAMMING AND METHODOLOGY

TEAM MEMBERS-

MUKUND KUKREJA
AMAN SHRIVASTAVA
AYUSHMAN RAWAT
KHUSHI VISHWAKARMA
MAYANK VISHWAKARMA

BRANCH-

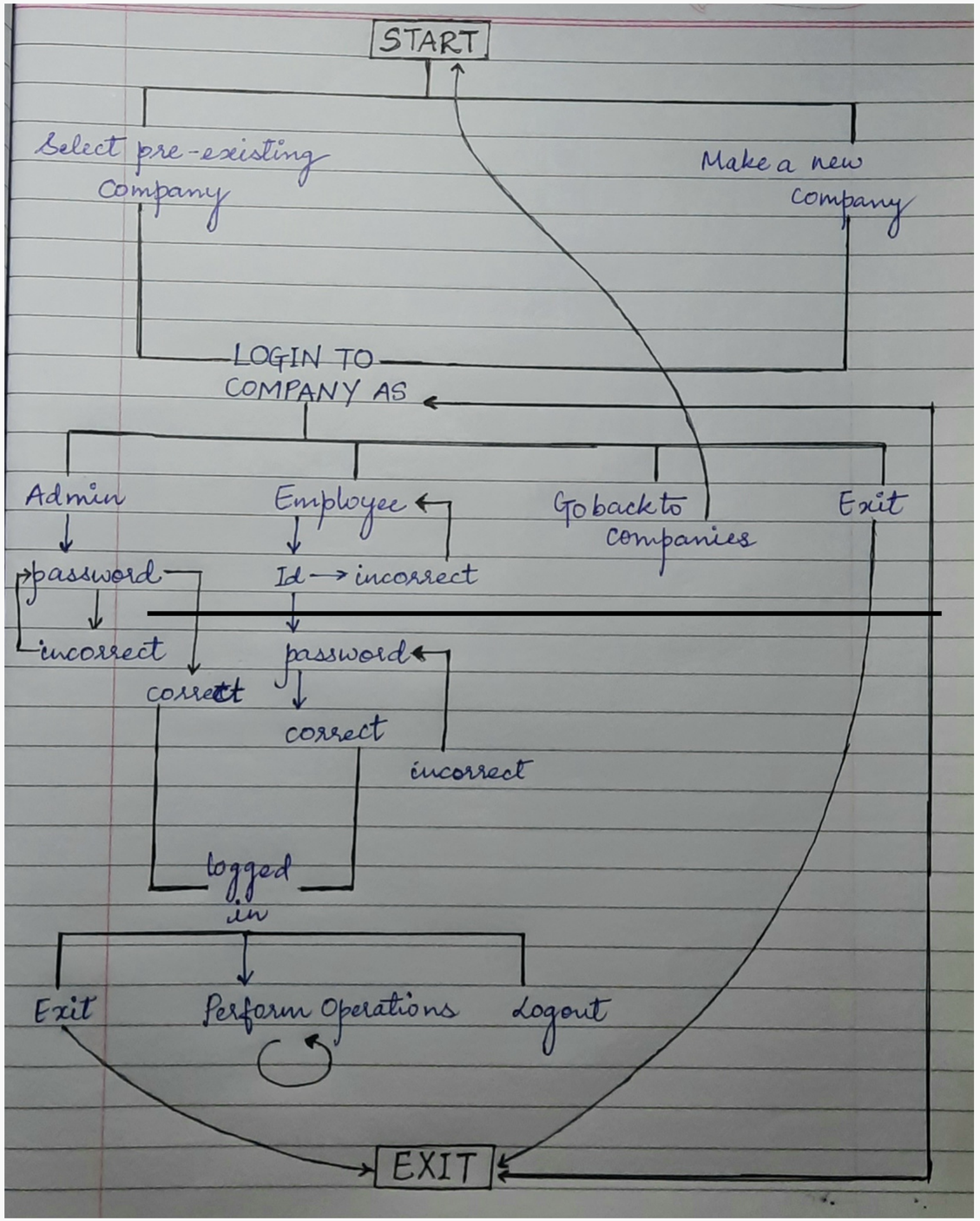
AIML-1 (3RD SEMESTER)

SUBMITTED TO-

PROF. JUHI JAIN MAM

EMPLOYEE MANAGEMENT SYSTEM USING THE CONCEPTS OF OOPs

FLOWCHART



USE CASE DIAGRAM

CLASSES

1. COMPANY

2. DEPARTMENT

3. EMPLOYEE

4. DETAILS

1 CLASS COMPANY -

Data Members:

- **public:**

- string companyName
- Employee* CEO (Pointer to Employee obj that is the CEO of the company)
- Company *prev, *next (Company Doubly Linked List pointers)

- **private:**

- Department *firstDepartment, *lastDepartment (Department Doubly Linked List pointers)
- Employee *firstEmployeeOfCompany, *lastEmployeeOfCompany (Employee Doubly Linked List pointers)
- string adminPassword
- int noOfDepartments, noOfEmployees
- int noOfLeavesAllowed, workingDays
- float salaryReductionPercentage

Member Functions:

- **public:**

- string getSalaryPolicy()
- float getEmployeeSalary(Employee *emp)

- void displayDepartments()
- void displayAllEmployees()
- void displayAllEmployeesWithDepartments()

- Department* makeAndAddDepartment()
- Department* addDepartment(Department *department)
- Department* getDepartment(int index)
- Department* selectADepartment()

- Employee* setCEO(Employee* emp)
- Employee* makeAndAddCEO()
- Employee* getEmployeeFromIndex(int index)
- Employee* selectAnEmployee()
- Employee* getEmployeeFromId(string id)

Friends:

- **classes:**
  friend class Department;

- **functions:**
  friend void startProgram()

  friend bool login(Company *company)
  friend bool startProgramForEmployee(Company *company,Employee *emp)
  friend bool startProgramForAdmin(Company *company)

  friend Company* makeANewCompany(Company *lastCompany)

Constructors:

- Company(string companyName) - Parametrized Constructor

2 CLASS DEPARTMENT –

Constructors:

- Department(string name)

Data Members:

- **public:**

- string departmentName
- Company *company
- Employee* HOD
- int noOfEmployees
- Department *next, *prev

- **private:**

- Employee *firstEmployee, *lastEmployee

Member Functions:

- **public:**

- void setHOD(Employee* emp)
- void displayEmployees()

- string getEmployeeId(Employee* emp)

- Employee* addEmployee(Employee *emp,string post="---",bool makeHOD = false)
- Employee* makeAndAddEmployee(bool makeHOD = false)
- Employee* getEmployeeByIndex(int index)
- Employee* selectAnEmployee()

3 CLASS EMPLOYEE -

Constructors:

- Employee() - Default Constructor
- Employee(string name, int age,float salary,int leaves=0, string DOB = "---", string address = "---", string phoneNumber = "---",string post="---",string departmentName="---") - Parametrized Constructor

Data Members:

- **public:**
  - Details *details
  - string companyName, departmentName, post
  - string id, password
  - int leaveCount
  - float annualSalary
  - Employee *prev, *next

Member Functions:

- **public:**

- static Employee* makeEmployee()
- void displayDetails(int curSalary)

4 CLASS DETAILS -

Constructors:

- Details(string name, int age, string DOB = "---", string address = "---", string phoneNumber = "---")
-Parametrized Constructor

Data Members:

- **public:**
  - string name
  - int age
  - string DOB
  - string address
  - string phoneNumber

Member Functions:

- **public:**
  - static Details* getDetailsFromUser()

GLOBAL FUNCTIONS USED:

- Company* getExampleCompany()
- Company* makeANewCompany(Company *lastCompany)
- Company* getCompanyByIndex(Company* firstCompany,int index)

- void printAdminOperations(Company *company)
- void printEmployeeOperations(Company* company)
- int printCompanies(Company *firstCompany)

- bool startProgramForAdmin(Company *company)
- bool startProgramForEmployee(Company *company,Employee *emp)
- bool login(Company *company)

- void startProgram()
- int main()

Concepts used:

- Classes & Objects
- Constructors (Default, Parametrized, Copy)
- Data abstraction - access specifiers (private, public)
- friend function & friend class
- Forward declaration of classes
- Dynamic Memory allocation using 'new' keyword
- Doubly Linked List

Concepts not used:

- Polymorphism
- Inheritance

Header Files used:

- iostream - cin, cout, endl
- string - getline

Thank you!