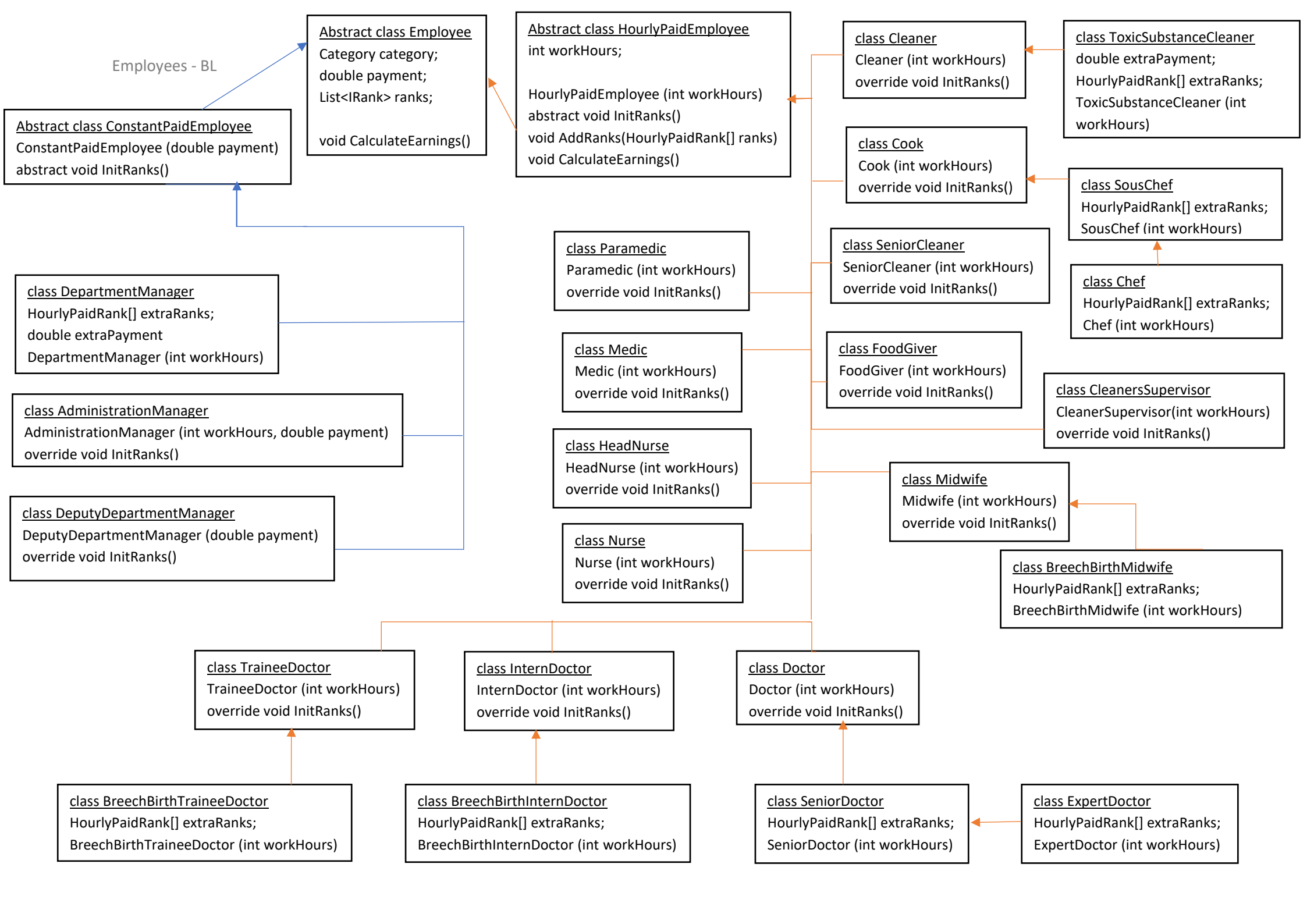
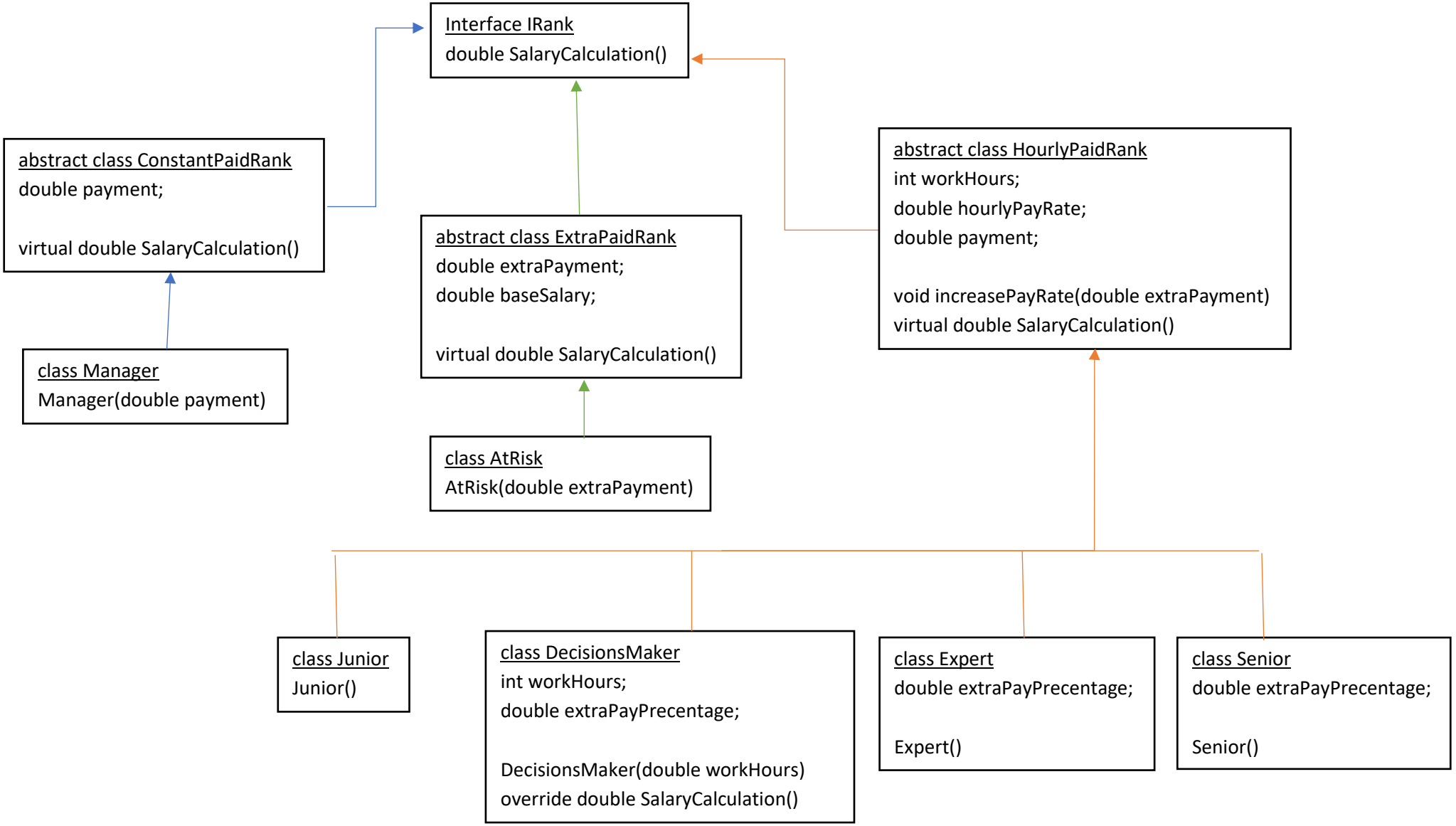

Employees - BL
Abstract class Employee
Category category;
double payment;
List<IRank> ranks;
void CalculateEarnings()
Abstract class HourlyPaidEmployee
int workHours;
HourlyPaidEmployee (int workHours)
abstract void InitRanks()
void AddRanks(HourlyPaidRank[] ranks)
void CalculateEarnings()
Abstract class ConstantPaidEmployee
ConstantPaidEmployee (double payment)
abstract void InitRanks()
class Cleaner
Cleaner (int workHours)
override void InitRanks()
class ToxicSubstanceCleaner
double extraPayment;
HourlyPaidRank[] extraRanks;
ToxicSubstanceCleaner (int workHours)
class Cook
Cook (int workHours)
override void InitRanks()
class SousChef
HourlyPaidRank[] extraRanks;
SousChef (int workHours)
class Chef
HourlyPaidRank[] extraRanks;
Chef (int workHours)
class DepartmentManager
HourlyPaidRank[] extraRanks;
double extraPayment
DepartmentManager (int workHours)
class Paramedic
Paramedic (int workHours)
override void InitRanks()
class SeniorCleaner
SeniorCleaner (int workHours)
override void InitRanks()
class AdministrationManager
AdministrationManager (int workHours, double payment)
override void InitRanks()
class Medic
Medic (int workHours)
override void InitRanks()
class FoodGiver
FoodGiver (int workHours)
override void InitRanks()
class CleanersSupervisor
CleanerSupervisor(int workHours)
override void InitRanks()
class DeputyDepartmentManager
DeputyDepartmentManager (double payment)
override void InitRanks()
class HeadNurse
HeadNurse (int workHours)
override void InitRanks()
class Midwife
Midwife (int workHours)
override void InitRanks()
class Nurse
Nurse (int workHours)
override void InitRanks()
class BreechBirthMidwife
HourlyPaidRank[] extraRanks;
BreechBirthMidwife (int workHours)
class TraineeDoctor
TraineeDoctor (int workHours)
override void InitRanks()
class InternDoctor
InternDoctor (int workHours)
override void InitRanks()
class Doctor
Doctor (int workHours)
override void InitRanks()
class BreechBirthTraineeDoctor
HourlyPaidRank[] extraRanks;
BreechBirthTraineeDoctor (int workHours)
class BreechBirthInternDoctor
HourlyPaidRank[] extraRanks;
BreechBirthInternDoctor (int workHours)
class SeniorDoctor
HourlyPaidRank[] extraRanks;
SeniorDoctor (int workHours)
class ExpertDoctor
HourlyPaidRank[] extraRanks;
ExpertDoctor (int workHours)


Interface IRank
double SalaryCalculation()
abstract class ConstantPaidRank
double payment;
virtual double SalaryCalculation()
abstract class ExtraPaidRank
double extraPayment;
double baseSalary;
virtual double SalaryCalculation()
abstract class HourlyPaidRank
int workHours;
double hourlyPayRate;
double payment;
void increasePayRate(double extraPayment)
virtual double SalaryCalculation()
class Manager
Manager(double payment)
class AtRisk
AtRisk(double extraPayment)
class Junior
Junior()
class DecisionsMaker
int workHours;
double extraPayPrecentage;
DecisionsMaker(double workHours)
override double SalaryCalculation()
class Expert
double extraPayPrecentage;
Expert()
class Senior
double extraPayPrecentage;
Senior()

DAL & WardDataManagement - BL

class ShiftDal
DataTable GetAllShifts()
DataTable GetShiftsByEmployeeID(int employeeID)
Int AddShift(int shiftID, int employeeID, string hourIn, int hourOut)
bool isExist(int shiftID)
DataTable GetShift(int shiftID)

class DBHelper
string connectionString;
bool connectionOpen;
SqlConnection dbConnection;

bool openConnection()
void closeConnection()
SqlDataReader readData(string sql)
int writeData(string sql)
DataTable GetDataTable(string sql)

class EmployeeDAL
DataTable GetAllEmployees()
DataTable GetEmployeeByType(string employeeType)
Int AddEmployee(int id, string type, int category, int
workHours, bool isHourlyPaid, double constantPayment)
bool isExist(int employeeID)
int updateEmployeePayment(int employeeID, double payment)

class Shift
DateTime hourIn;
DateTime hourOut;

Shift(hourIn, hourOut)

class ManageShifts
string dateFormat

bool AddShift(Shift shift)
List<Shift> GetShiftsByEmployeeID(int employeeID)

class ManageEmployees
Employee AddEmployee(Employee employee)
Employee GetEmployeeByID(int employeeID)
List<Employee> GetEmployeesByType(EmployeeTypeEnum employeeType)