UML-diagram

Groupmembers: Lucas Jacobs, Bart Steer, Tom Strijbos, Metodi Tarnev, Alex Vartic
Pcn: 490692, 491604, 479570, 486517, 4030435
Class: CB-07
Group: 2
Tutor: Aleyna Kartal
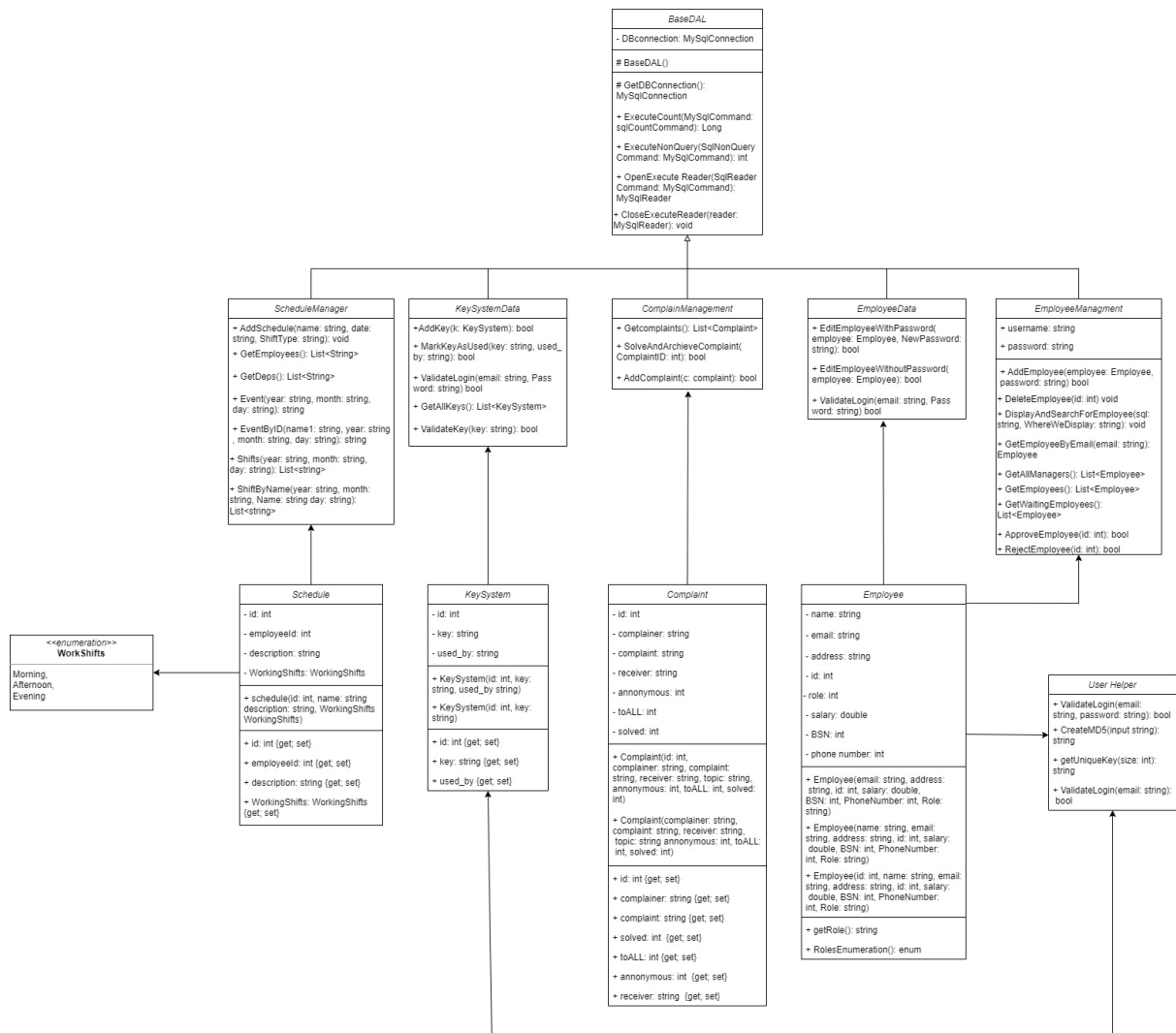Date: 20-02-2022
Wordcount: 1182

BaseDAL

- DBconnection: MySqlConnection

BaseDAL()

GetDBConnection(): MySqlConnection

+ ExecuteCount(MySqlCommand: sqlCountCommand): Long

+ ExecuteNonQuery(SqlNonQuery Command: MySqlCommand): int

+ OpenExecute Reader(SqlReader Command: MySqlCommand): MySqlReader

+ CloseExecuteReader(reader: MySqlReader): void

ScheduleManager

+ AddSchedule(name: string, date: string, ShiftType: string): void

+ GetEmployees(): List<String>

+ GetDeps(): List<String>

+ Event(year: string, month: string, day: string): string

+ EventByID(name1: string, year: string, month: string, day: string): string

+ Shifts(year: string, month: string, day: string): List<string>

+ ShiftByName(year: string, month: string, Name: string day: string): List<string>

KeySystemData

+AddKey(k: KeySystem): bool

+ MarkKeyAsUsed(key: string, used_by: string): bool

+ ValidateLogin(email: string, Pass word: string) bool

+ GetAllKeys(): List<KeySystem>

+ ValidateKey(key: string): bool

ComplainManagement

+ Getcomplaints(): List<Complaint>

+ SolveAndArchieveComplaint(ComplaintID: int): bool

+ AddComplaint(c: complaint): bool

EmployeeData

+ EditEmployeeWithPassword(employee: Employee, NewPassword: string): bool

+ EditEmployeeWithoutPassword(employee: Employee): bool

+ ValidateLogin(email: string, Pass word: string) bool

EmployeeManagment

+ username: string

+ password: string

+ AddEmployee(employee: Employee, password: string) bool

+ DeleteEmployee(id: int) void

+ DisplayAndSearchForEmployee(sql: string, WhereWeDisplay: string): void

+ GetEmployeeByEmail(email: string): Employee

+ GetAllManagers(): List<Employee>

+ GetEmployees(): List<Employee>

+ GetWaitingEmployees(): List<Employee>

+ ApproveEmployee(id: int): bool

+ RejectEmployee(id: int): bool

Schedule

- id: int

- employeeId: int

- description: string

- WorkingShifts: WorkingShifts

+ schedule(id: int, name: string description: string, WorkingShifts WorkingShifts)

+ id: int {get; set}

+ employeeId: int {get; set}

+ description: string {get; set}

+ WorkingShifts: WorkingShifts {get; set}

<<enumeration>> WorkShifts

Morning,
Afternoon,
Evening

KeySystem

- id: int

- key: string

- used_by: string

+ KeySystem(id: int, key: string, used_by string)

+ KeySystem(id: int, key: string)

+ id: int {get; set}

+ key: string {get; set}

+ used_by {get; set}

Complaint

- id: int

- complainer: string

- complaint: string

- receiver: string

- annonymous: int

- toALL: int

- solved: int

+ Complaint(id: int, complainer: string, complaint: string, receiver: string, topic: string, annonymous: int, toALL: int, solved: int)

+ Complaint(complainer: string, complaint: string, receiver: string, topic: string annonymous: int, toALL: int, solved: int)

+ id: int {get; set}

+ complainer: string {get; set}

+ complaint: string {get; set}

+ solved: int {get; set}

+ toALL: int {get; set}

+ annonymous: int {get; set}

+ receiver: string {get; set}

Employee

- name: string

- email: string

- address: string

- id: int

- role: int

- salary: double

- BSN: int

- phone number: int

+ Employee(email: string, address: string, id: int, salary: double, BSN: int, PhoneNumber: int, Role: string)

+ Employee(name: string, email: string, address: string, id: int, salary: double, BSN: int, PhoneNumber: int, Role: string)

+ Employee(id: int, name: string, email: string, address: string, id: int, salary: double, BSN: int, PhoneNumber: int, Role: string)

+ getRole(): string

+ RolesEnumeration(): enum

User Helper

+ ValidateLogin(email: string, password: string): bool

+ CreateMD5(input string): string

+ getUniqueKey(size: int): string

+ ValidateLogin(email: string): bool

Figure 1: UML-diagram