**Пояснительная записка к TestTask**

Разработана программа для подсчета зарплаты сотрудников для отдела HR компании.

**Описание БД.**

Для работы с приложением использована БД Sqlite.

База данных состоит из 3 таблиц: worker, head\_subordinate, user.

Таблица worker содержит в себе информацию о работнике (имя, дата поступления на работу).

Таблица head\_subordinate хранит в себе информацию о связях начальника с подчиненными.

Таблица user содержит информацию о пользователях с их правами.

Таблица head\_subordinate связана по ключу с таблицей worker.

На таблицу head\_subordinate “навешан” тригер -у сотрудника Employee не могут быть подчинённые.

На таблицу worker “навешан” тригер- при удалении работника, удаляются связи в таблице head\_subordinate.

**Описание классов Project 1.**

Созданы два статичных класса WorkerSalary и InsertDGV.

Класс InsertDGV служит для заполнения DataGriedView. В классе InsertDGV реализованы два метода заполнения DataGriedView. Один метод выводит всех сотрудников фирмы, а другой отображает всех подчинённых выбранного начальника. Класс создан, чтобы отдельно вынести общую функциональную часть.

Класс WorkerSalary служит для подсчета заработной платы сотрудника, в зависимости от занимаемой должности. В классе WorkerSalary реализованиы 3 метода:CalcEmployeeSalary-вычисление зарплаты Employee, CalcManagerSalary-вычисление зарплаты Manager,CalcSalesmanSalary -вычисление зарплаты Salesman.

**Описание Windows Form Project 1.**

**Form 1.** При запуске приложения открывается Form1-форма авторизации пользователя. Если в БД существует пользователь, и пароли совпадают, то в зависимости от прав пользователя открывается одна из форм. Если пользователь имеет права “WriteRead”, то открывается Form2-форма супер-пользователя. Если пользователь имеет право “Read”,то открывается Form3-форма для обычного пользователя.

**Form 2.** Form 2-форма супер-пользователя. На форме два DataGriedView для отображения всех сотрудников фирмы, а так же подчиненных, если они есть у данного сотрудника. Два DateTimePicker для задания начало и конца расчетного периода. Кнопка рассчитать - расчет заработной платы. Расчет заработной платы производиться для сотрудника, на которого наведён курсор в левой DataGriedView.Две кнопки добавить нового сотрудника и удалить сотрудника. При нажатии этой кнопки появляется Form4-форма добавления нового сотрудника. Кнопка удалить сотрудника(не работает/ не реализован функционал).

**Form 3.** Form 3-форма для обычного пользователя. При загрузки формы, в левой DataGriedView отображаются все сотрудники пользователя, а в правой все его подчиненные. При изменении в левой DataGriedView, правая DataGriedView не изменяется, запрещено просматривать подчиненных других сотрудников. На форме есть возможность выбрать начало и конец периода, за который нужно рассчитать заработную плату. При нажатии на кнопку рассчитать зарплату для сотрудника user производится вычисление его заработанной платы. При нажатии на кнопку рассчитать заработную плату подчиненного-должен происходить расчет заработной платы подчиненного(не долено/алгорит разработан и испотзуется в Form 2).

**Form 4.**Form 4-форма добавления нового сотрудника. В форме надо заполнить информацию о новом сотруднике. Чтобы сотрудник добавился в БД, необходимо нажать кнопку добавить.

**Тесты.**

1) Ввод пользователя с правами WriteRead





2)Ввод пользователя с правами Read





3) Ввод несуществующего пользователя





4) Еси не ввести одно из полей появится то же самое сообщение.

5) При вводе строки в неправильном формате в программе вылетает исключение(или сделать через try/catch или проверка через TryParse)

**Заключение.**

* Выполнено тестовое задание расчет заработной платы сотрудников отдела с использованием sqlite.
* Реализованы алгоритмы расчета заработной платы в зависимости от занимаемой должности(возможны некоторые неточности вычисления, так как реализовал алгоритмы, как смог понять условия)
* Написан краткий обзор решения тестовой задачи, описана архитектура. Возможно, к минусам данной архитектуры можно отнести то, что Form 2 и Form 3 имеют схожий интерфейс. Можно было сделать форму-родителя, а потом унаследовать элементы, некоторые из них сделав невидимыми. Но, чтобы разделить логику работы программы для обычного и супер-пользователя, были реализованы две похожие формы.Можно добавить кнопку -добавить связь начальник-подчиненный. Не реализован функционал кнопок: удаления сотрудника, в Form3 расчет заработной платы подчиненного(алгоритм разработан, применяется в Form 2). Алгоритмы вычисления вынесены в отдельный класс. Можно добавить еще меню, а так же кнопку редактирования сотрудника.
* Код покрыт примитивными тестами. Предполагается, что изначально в БД записи хранятся в правильном формате.
* Программа имеет простой графический интерфейс. Его можно еще дорабатывать.
* Есть возможность разграничения прав. Так как это тестовое задание, то таблица user сильно упрощена. Могут быть user с одинаковыми именами. Для того чтобы user можно было использовать в реальных проектах, таблицу надо по ключу соединить с worker.

Представлена первая версия проекта. Постарался сделать программу, с учетом требований тестового задания. Для использования в реальных целях, её нужно дорабатывать, дописывать проверки корректности данных, проводить дополнительные тесты. Правильно наименование форм сделать. Готов ее доработать, дописать недостающий функционал и выслушать замечания. А так же на собеседовании ответить на все интересующие вопросы, объяснить, как все реализовано и работает. Спасибо!!!