Роли:

(all) Кнопка создания заданий - 1

(all) Кнопка создания событий - 2

ВХОД

Авторизация||Регистрация

(all) Кнопка создания заявлений - 3

Главная страница

Личная информация: из представления EmployeesFullView по id сотрудника. Из-за того ,что таблица Employee и JobEmployee связь многие-многие, то инфа по сотруднику мб продублирована, потому что у сотрудника мб несколько должностей

Актуальные события: из представления EventsView. Будет брать только те события, даты которых еще не наступили, а также только те события, что относятся к сотруднику, за которого был выполнен вход.

Заявления: те заявления ,которые требуют подтверждения. В таблице LogStatement если поле approve 1 – заявление одобрено, 2 – отказано, 3 – нуждается в подтверждении. Поле approver – тот, сотрудник, который должен событие подтвердить. Т.е. на странице должны появляться те заявления на которых id сотрудника совпадает с полем approver, и поле approve – 3.

! здесь нужно, чтобы сотрудник мог посм инфу по заявлению и нажать

подтверждение при необходимости

Задания: из таблицы EmployeeTask, те, где id\_employee совпадает с id, выводится id задания. (хм, по идее тут скорее всего добавлю представление, а то как-то криво вышло)

(all) Кнопка просмотра всех сотрудников - 4

(admin) Кнопка редактирования всех данных по сотруднику- 5

all

all

all

1. Создание заданий :

Заполнение таблиц Task и

Employee\_Task. Поле name\_of\_task

это краткая суть задания,

comment\_fe это более подробное

описание

2. Создание событий :

Суть примерно та же, что и в заданиях, но в таблице NoticeEvent поле

id\_employee это тот, кто событие назначил, id\_recipient – тот, кто

событие получает.

3. Создание заявлений :

Поле days\_sum количество дней отсуствия, id\_approver – тот, кто будет

проверять заявление и подтверждать его. Поле type\_leave это тип заявления

(увольнение, отпуск и т.п.). Поле approve – состояние подтверждения

заявления (1- одобрено, 2- отказано, 3- ожидание ответа). Поле id\_employee

это сотрудник, что заявление составляет. Это в таблице LogStatement, она

связана с таблицей Document, там содержится путь к скану документа,

который является бумажной копией заявления (может отсуствовать).

Исходя из дат, когда сотрудника не будет, менять его роль на

заблокированного BLOCK в поле role в таблице Login, чтобы во время

его отсуствия на работе он не мог зайти в аккаунт.

4. Все сотрудники :

Через представление EmployeesView, в нем не содержатся все данные о сотруднике без личных данных. По-хорошему сюда же надо добавить фильтрации сотрудников по каким-либо пунктам, типа фамилии и т.п.

5. Редактирование данных :

Доступно только администратору. Здесь же и будет блокироваться пользователь при необходимости. Посмотреть все данные по представлению EmloyeeFullView, чтобы подтянулись все данные по сотруднику (и общие и личные) и дальше администратор мог смотреть какие поля ему необходимо изменять.

! также для администраторов можно сделать список всех сотрудников в полными данными.

! удалять аккаунт может тоже только админ ,соответственно, либо через PostgreSQL(саму бд).