

«Домашняя бухгалтерия»

Итоговый проект

Итоговый проект основывается на выполненных домашних работах. Предполагается, что пользователь может вводить текущие и планируемые расходы и получать статистику по своим тратам за указанный интервал времени, в зависимости от назначения трат производится группировка данных. В информационной системе необходимо предусмотреть возможность ведения одного или более счетов и учитывать общий баланс пользователя.

На стороне сервера используется язык программирования Python:

- Django;
- Django REST framework (JSON).

Возможно использование вспомогательных библиотек Python.

На стороне клиента используется язык разметки HTML. Возможно использование сценариев JavaScript и вспомогательных библиотек языка. Возможно использование стилей CSS.

На стороне сервера необходимо использование одной из СУБД:

- SQLite;
- MySQL;
- PostgreSQL.

Состав

Компоненты аутентификации и авторизации

- 1) Подключение django.contrib.auth;
- 2) Реализация формы Register ("Регистрация в системе");
- 3) Реализация формы Login ("Вход в систему").

Компоненты и модели БД

- 1) Подключение СУБД;
- 2) Реализация модели Profile ("Профиль пользователя"):
 - 1) Реализация привязки (один-к-одному) к модели User ("Пользователь");
 - 2) Реализация полей Address ("Адрес"), PhoneNumber ("Номер телефона");
- 3) Реализация модели Account ("Счет"):
 - 1) Реализация привязки (многие-к-одному) к модели User ("Пользователь");
 - 2) Реализация уникального поля AccountNumber ("Номер счета");
- 4) Реализация модели Charge ("Зачисление/списание со счета"):
 - 1) Реализация привязки (многие-к-одному) к модели Account ("Счет");
 - 2) Реализация полей Amount ("Сумма зачисления/списания со счета"), TransactedAt ("Дата и время проведения");
- 5) Реализация формы создания/редактирования модели Profile ("Профиль пользователя"):
 - 1) Реализация проверок поля PhoneNumber ("Номер телефона"), должно быть в международном формате;
- 6) Реализация формы создания/редактирования модели Account ("Счет"):
 - 1) Реализация проверок поля AccountNumber ("Номер счета"), должно быть уникальным;
- 7) Реализация формы создания/редактирования модели Charge ("Зачисление/списание со счета"):
 - 1) Реализация проверок полей Amount ("Сумма зачисления/списания со счета") и TransactedAt ("Дата и время проведения"), списание (отрицательное значение Amount) не должно быть создано в будущем (TransactedAt меньше полуночи следующего дня).

Компоненты интерфейса

- 1) Реализация представления Main ("Главная страница");
- 2) Реализация представления UserRegister ("Регистрация в системе");
- 3) Реализация представления UserLogin ("Вход в систему");
- 4) Реализация представления UserLogout ("Выход из системы");
- 5) Реализация представления ProfileEdit ("Профиль пользователя"):
 - а) возможный параметр UserName ("Имя пользователя", по умолчанию - текущий пользователь);

- 6) Реализация представления AccountList ("Список счетов"):
 - а) возможный параметр UserName ("Имя пользователя", по умолчанию - текущий пользователь);
- 7) Реализация представления Account ("Счет"):
 - а) параметр AccountNumber ("Номер счета");
- 8) Реализация представления AccountStat ("Статистика по счету"):
 - а) параметр AccountNumber ("Номер счета");
 - б) вспомогательные параметры ("Даты");
- 9) Реализация представления AccountReport ("Отчет по счету"):
 - а) параметр AccountNumber ("Номер счета");
 - б) вспомогательные параметры ("Даты");
- 10) Реализация представления AccountCreate ("Создание счета"):
 - а) возможный параметр UserName ("Имя пользователя", по умолчанию - текущий пользователь);
- 11) Реализация представления AccountEdit ("Редактирование счета"):
 - а) параметр AccountNumber ("Номер счета");
- 12) Реализация представления AccountDelete ("Удаление счета"):
 - а) параметр AccountNumber ("Номер счета");
- 13) Реализация представления ChargeCreate ("Создание зачисления/списания со счета"):
 - а) параметр AccountNumber ("Номер счета");
- 14) Реализация представления ChargeEdit ("Редактирование зачисления/списания со счета"):
 - а) параметр ChargeID ("Идентификатор зачисления/списания со счета");
- 15) Реализация представления ChargeDelete ("Удаление зачисления/списания со счета"):
 - а) параметр ChargeID ("Идентификатор зачисления/списания со счета");
- 16) Реализация представления UserList ("Список пользователей");
- 17) Реализация представления UserEdit ("Пользователь"):
 - а) Параметр UserName ("Имя пользователя");
- 18) Реализация представления UserDelete ("Удаление пользователя"):
 - а) Параметр UserName ("Имя пользователя");
- 19) Реализация шаблонов.

Компоненты сервисов в стиле REST (допустимый формат - JSON)

- 1) Сервисы модели Profile ("Профиль пользователя"):
 - 1) Сервис просмотра сущности Profile ("Профиль пользователя");
 - а) возможный параметр UserName ("Имя пользователя", по умолчанию - текущий пользователь);
 - 2) Сервис изменения сущности Profile ("Профиль пользователя");
 - а) возможный параметр UserName ("Имя пользователя", по умолчанию - текущий пользователь);
- 2) Сервисы модели Account ("Счет"):
 - 1) Сервис просмотра списка сущностей Account ("Счет");
 - а) возможный параметр UserName ("Имя пользователя", по умолчанию - текущий пользователь);

- 2) Сервис создания сущности Account ("Счет");
 - a) возможный параметр UserName ("Имя пользователя", по умолчанию - текущий пользователь);
- 3) Сервис просмотра сущности Account ("Счет");
 - a) параметр AccountNumber ("Номер счета");
- 4) Сервис изменения сущности Account ("Счет");
 - a) параметр AccountNumber ("Номер счета");
- 5) Сервис удаления сущности Account ("Счет");
 - a) параметр AccountNumber ("Номер счета");
- 6) Сервис просмотра статистики по балансу сущности Account ("Счет");
 - a) параметр AccountNumber ("Номер счета");
 - b) вспомогательные параметры ("Даты");
- 3) Сервисы модели Charge ("Зачисление/списание со счета"):
 - 1) Сервис просмотра списка сущностей Charge ("Зачисление/списание со счета");
 - a) параметр AccountNumber ("Номер счета");
 - 2) Сервис просмотра сущности Charge ("Зачисление/списание со счета");
 - a) параметр ChargeID ("Идентификатор зачисления/списания со счета");
 - 3) Сервис изменения сущности Charge ("Зачисление/списание со счета");
 - a) параметр ChargeID ("Идентификатор зачисления/списания со счета");
 - 4) Сервис удаления сущности Charge ("Зачисление/списание со счета");
 - a) параметр ChargeID ("Идентификатор зачисления/списания со счета").

Права доступа

В системе определены роли:

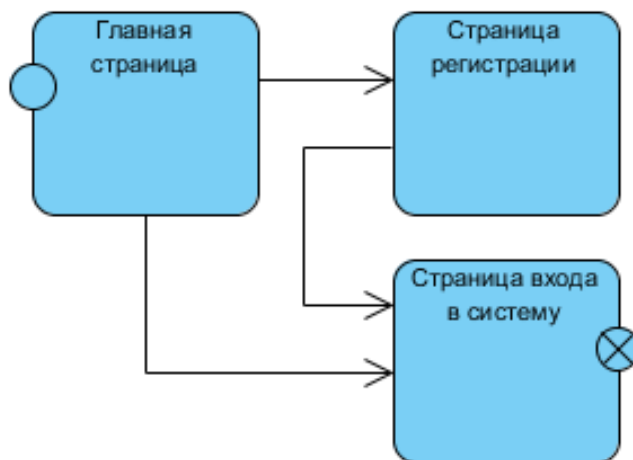
- Guest ("Гость"), пользователь не прошедший аутентификацию;
- User ("Пользователь"), пользователь прошедший аутентификацию;
- Administrator ("Администратор"), пользователь с особыми правами.

Для роли Guest ("Гость") должны быть доступны представления:

- 1) Main ("Главная страница"), шаблон должен содержать описание проекта;
- 2) UserRegister ("Регистрация в системе"), шаблон должен содержать:
 - a) обязательные поля UserName ("Имя пользователя"), Email ("E-mail") и PhoneNumber ("Номер телефона");
 - b) необязательное поле Address ("Адрес");
 - c) обязательное Password ("Пароль") и, возможно, дополнительное поле для проверки введенного пароля;
 - d) кнопку Submit ("Отправить"), при нажатии на которую создается новый пользователь;
- 3) UserLogin ("Вход в систему"), шаблон должен содержать:
 - a) обязательные поля UserName ("Имя пользователя") и Password ("Пароль");
 - b) кнопку Submit ("Отправить"), при нажатии на которую осуществляется аутентификация.

Все остальные представления (а также все REST API) должны быть недоступны роли Guest ("Гость").

Интерфейсы роли "Гость"

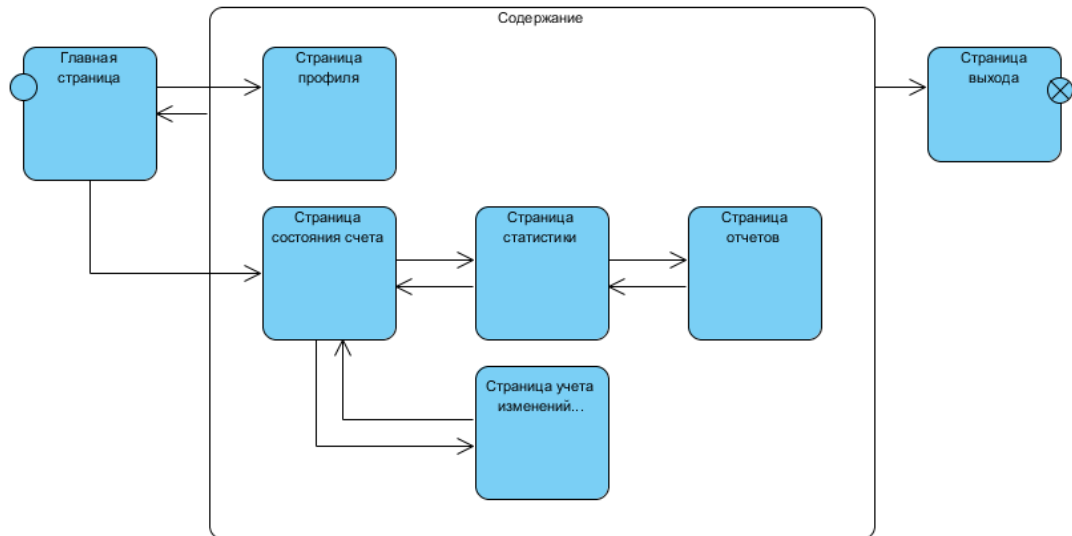


Для роли User ("Пользователь") должны быть доступны представления:

- 1) Main ("Главная страница"), где должно быть осуществлено перенаправление на представление AccountList ("Список счетов");
- 2) UserLogout ("Выход из системы"), где должно быть осуществлено завершение сеанса пользователя;
- 3) ProfileEdit ("Профиль пользователя") без параметров, шаблон должен содержать:
 - a) форму редактирования сущности Profile ("Профиль пользователя");
- 4) AccountList ("Список счетов") без параметров, шаблон должен содержать:
 - a) список сущностей Account ("Счет") с текущим балансом и ссылками на представления Account ("Счет"), AccountEdit ("Редактирование счета") и AccountDelete ("Удаление счета") по каждому счету;
 - b) ссылку на представление AccountCreate ("Создание счета");
- 5) Account ("Счет"), шаблон должен содержать:
 - a) список сущностей Charge ("Зачисление/списание со счета") по текущему счету с суммой зачисления/списания, датой проведения и ссылкой на представление ChargeDelete ("Удаление зачисления/списания со счета") по каждому зачислению/списанию;
 - b) ссылку на представление ChargeCreate ("Создание зачисления/списания со счета") для текущего счета;
 - c) ссылку на представление AccountStat ("Статистика по счету") для текущего счета;
- 6) AccountStat ("Статистика по счету"), шаблон должен содержать:
 - a) вспомогательные параметры ("Даты") для фильтрации зачислений/списаний;
 - b) список сгруппированных по месяцам сумм зачислений/списаний по текущему счету;
 - c) ссылку на представление AccountReport ("Отчет по счету");
- 7) AccountReport ("Отчет по счету"), где должно быть осуществлено формирование CSV-файла сгруппированных по месяцам сумм зачислений/списаний по текущему счету;
- 8) AccountCreate ("Создание счета") без параметров, шаблон должен содержать:
 - a) форму создания сущности Account ("Счет");
- 9) AccountEdit ("Редактирование счета"):
 - a) форму редактирования сущности Account ("Счет");
- 10) AccountDelete ("Удаление счета"), где должно быть осуществлено удаление счета;
- 11) ChargeCreate ("Создание зачисления/списания со счета"), шаблон должен содержать:
 - a) форму создания сущности Charge ("Зачисление/списание со счета");
- 12) ChargeDelete ("Удаление зачисления/списания со счета"), где должно быть осуществлено удаление зачисления/списания.

При каждом действии роли User ("Пользователь") над каким-либо объектом необходимо осуществлять проверку принадлежности этого объекта пользователю, в случае обращения к "чужим" данным доступ должен быть запрещен. В случае REST API необходимо пользоваться тем же списком разрешенных действий и так же проверять принадлежность.

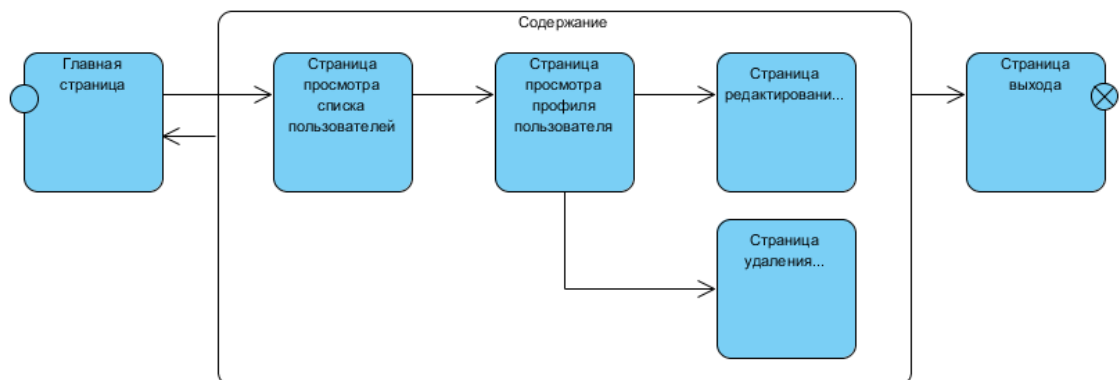
Интерфейсы роли "Пользователь"



Для роли Administrator ("Администратор") должны быть доступны все представления (в том числе и по "чужим" данным), включая представления:

- 1) Main ("Главная страница"), где должно быть осуществлено перенаправление на представление userList ("Список пользователей");
- 2) userList ("Список пользователей"), шаблон должен содержать:
 - а) список сущностей User ("Пользователей") с именами пользователей, ссылками на представления UserEdit ("Пользователь"), UserDelete ("Удаление пользователя") и AccountList ("Список счетов") по каждому пользователю;
- 3) UserEdit ("Пользователь"), шаблон должен содержать:
 - а) форму редактирования User ("Пользователей");
- 4) UserDelete ("Удаление пользователя"), где должно быть осуществлено удаление пользователя.

Интерфейсы роли "Администратор"



Примечания

Описанные требования обязательны для исполнения в части содержания (перечисленные компоненты должны быть реализованы), визуальная составляющая (графический интерфейс пользователя) остается на усмотрении разработчика. Дополнительная функциональность может быть реализована и будет оцениваться отдельно от основной работы.