министерство НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ   
Российской федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«ЮГОРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

институт ЦИФРОВОЙ ЭКОНОМИКИ

Курсовая работа по направлению

«Информатика и вычислительная техника»

на тему «Разработка игры с использованием ООП»

Выполнил: Юсуф М.А.

Группа 1191б

Научный руководитель:

доцент кафедры цифровых технологий

Сафонов Е. И.

Ханты-Мансийск 2022

Оглавление

[ГЛАВА 1. Разработка концепции веб-приложения 4](#_Toc126160253)

[ГЛАВА 2. База данных 6](#_Toc126160254)

[ГЛАВА 3 Ролевая модель. 9](#_Toc126160255)

[Заключение 11](#_Toc126160256)

[Список литературы 12](#_Toc126160257)

[ПРИЛОЖЕНИЕ 1 13](#_Toc126160258)

[ПРИЛОЖЕНИЕ 2 14](#_Toc126160259)

**ВВЕДЕНИЕ**

В качестве курсовой работы за 4 семестр по дисциплине «Разработка веб-приложения с использованием баз данных» было получено задание – написать сайт, использующий базу данных с не менее 4 таблицами. Дополнительное условие – на сайте должна быть реализована ролевая модель.

Мной был разработано веб-приложение с модулем авторизации, ролевой моделью. Он состоит из двух страниц: страницы авторизации/регистрации и основной. В базе данных хранится информация о пользователях, ролевой модели, объяктах, относящихся к пользователям.

Цель работы: разработать веб-приложение с использованием баз данных.

Для достижения цели необходимо выполнить следующие задачи:

1. Разработать концепцию приложения.
2. Создать базу данных и необходимые таблицы.
3. Разработать ролевую модель.
4. Написать серверную и клиентскую части приложения.
5. Разработать модель взаимодействия клиент-сервер.

**СОКРАЩЕНИЯ**

УЗ – Учетная запись

СУБД – Система управления базами данных

БД – база данных

Фронт – клиентская часть приложения

Бэк – серверная часть приложения

# ГЛАВА 1. Разработка концепции веб-приложения

Приложение будет состоять из двух страниц: страницы авторизации/регистрации и основной. Пользователю должна предоставляться возможность авторизоваться, регистрироваться. В качестве основной идеи мной было принято решение написать сайт для постановки свечек, поэтому у пользователя должна быть возможность, в зависимости от ролевой модели, поставить, убрать, поджечь, затушить свечку.

Страница авторизации: страница, на которой пользователь может авторизоваться под своей УЗ или же зарегистрировать новую УЗ. На ней должны присутствовать два поля для ввода логина и пароля, введённые символы в поле пароля символы должны заменяться на точки, чтобы обеспечить секретность введённого пароля. На форме должны быть кнопки авторизации и регистрации.

Основная страница должна отображать логин пользователя, отрисовывать подсвечник, поставленные пользователем свечки и их состояние (подожжённая/затушенная). На форме должно отображаться поле для ввода номера свечи, с которой пользователь будет взаимодействовать, кнопки взаимодействия: поставить, убрать, поджечь, затушить.

Ролевая модель состоит из трёх таблиц: таблицы с названием роли, таблицы с доступами компонент для ролей, таблицы с пользователями и их ролями.

Приложение состоит из двух частей: серверной и клиентской:

Клиентская часть выполняется на стороне пользователя. Реализована при использовании html, css, Javascript. На ней происходит описание всех кнопок, отрисовка свечей и подсвечника. При нажатии той или иной кнопки на серверную часть отправляется, соответствующий нажатой кнопки, запрос.

Серверная часть выполняет обработку запросов со стороны клиентской части. Написана на языке python при использовании библиотеки fastapi[1] для обработки запросов. На нём происходит работа с БД.

# ГЛАВА 2. База данных

В качестве СУБД принято решение использовать sqlite . по причине лёгкого подключения. БД хранится в основном проекте под названием sqlite\_base.db . Для взаимодействия бэка с БД используется библиотека SQLAlchemy[2]. Для управления миграциями используется Alembic[3]. Для него дополнительно выделяется одна таблица под названием – alembic\_version.

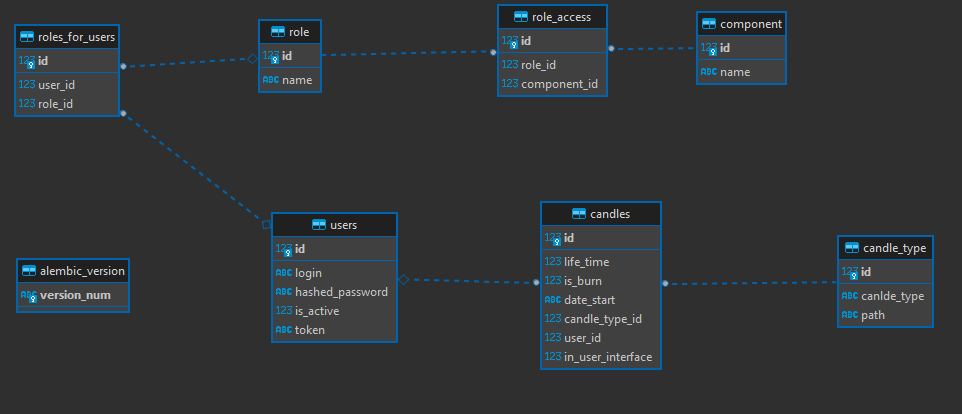
БД состоит из 8 таблиц: users, candles, candle\_type, roles\_for\_users, role, role\_acess, component. Схема БД приведена на рисунке 1.

Рисунок 1 – Схема БД

Таблица users содержит 5 столбцов:

* Id – идентификатор записи, является уникальным числовым полем, не может быть пустым.
* Login – значение логина пользователя, является уникальным строковым полем, не может быть пустым.
* Hashed\_password – захэшированное значение пароля пользователя, не пустое строковое поле.
* Is\_active – флаг состояния пользователя, показывающий активен ли в данный момент пользователь или нет, не может быть пустым, может принимать одно из двух значений – 1/0.
* Token – токен сессии пользователя, уникальное строкове поле, заполняющиеся при каждом входе пользователя в систему.

Таблица candles содержит 7 столбцов:

* Id – уникальный идентификатор записи, числовое непустое поле.
* Life\_time – время жизни свечи в секундах. Непустое числовое поле.
* Is\_burn – непустое состояние свечи, показывающее зажжена ли свеча, или нет. Может принимать одно из двух значений – 1/0.
* Date\_start – дата и время, когда свеча была зажжена, поле типа datetime.
* Candle\_type\_id – вид свечи, является ключом к таблице candle\_type, числовое, не может быть пустым.
* User\_id – пользователь, владеющий этой свечёй, является ключом к таблице users, числовое, не может быть пустым.
* In\_user\_interface – на какой позиции у пользователя находится свеча, числовое, непустое.

Таблица candle\_type состоит из 3 столбцов:

* Id – уникальный идентификатор записи, непустое числовое поле.
* Candle\_type – тип свечи, строковое непустое поле.
* Path – путь к изображению данной свечи в проекте, строковое поле.

Таблица role состоит из двух столбцов:

* ID – уникальный идентификатор записи, непустое числовое поле.
* Name – наименование роли, строковое, непустое, уникальное.

Таблица roles\_for\_users состоит из 3 столбцов:

* Id – уникальный идентификатор записи, непустое числовое поле.
* User\_id – ключ для таблицы users.
* Role\_id – ключ для таблицы role

Таблица component состоит из двух столбцов:

* Id – уникальный идентификатор записи, непустое числовое поле.
* Name – наименование компоненты, строковое, непустое, уникальное.

Таблица role\_access состоит из 3 столбцов:

* Id – уникальный идентификатор записи, непустое числовое поле.
* Role\_id – ключ для таблицы role
* Component\_id – ключ для таблицы component

Таблица alembic\_version требуется для отслеживания версионности миграций БД, в ней только один столбец с текущей версией миграции.

Используя SQLAlchemy в таблицу вставляются / редактируются / удаляются / читаются записи. Всё взаимодействие описано в src/crud.py . Редактирование самой БД производится путём наката миграций, при помощи Alembic. Все модели хранятся в src/models.py .

# ГЛАВА 3. Ролевая модель.

Доступ к функциям производится путём обращения к компоненте, соответствующей данной функции. Пользователь может иметь несколько ролей, если у него есть роль, для которой компонента доступна, то и вся функция будет доступна. Роли пользователей хранятся в отдельной таблице roles\_for\_users, такое решение было принято из-за возможности наличия у пользователя нескольких ролей.

У одной роли может быть доступ к нескольким компонентам. Соотношение какой роли какой компонент доступен описывается в таблице role\_access.

Все проверки ролевой модели производятся на бэке при обработке запроса. При вызове функции сразу после получения id пользователя вызывается метод получения всех компонент, доступных пользователю по всем его ролям. В полученном списке ищется необходимая компонента. Если она не найдена, выполнение функции прекращается с ошибкой 403 и описанием «deny: role access», в обратном случае выполнение функции продолжается.

В текущий момент существует 3 роли: User, Vip, Admin.

Описанные компоненты:

* Add\_candle – добавление свечи
* Rem\_candle – удаление свечи
* Burn – поджечь свечу
* Unburn – потушить свечу
* One\_candle – возможность поставить свечу на первую позицию
* Two\_candle - возможность поставить свечу на вторую позицию
* three\_candle - возможность поставить свечу на вторую позицию

Для роли User доступны компоненты: add\_candle, rem\_candle, burn, one\_candle.

Для роли Vip доступны компоненты: two\_candle, unburn

Для Admin доступна компонента three\_candle

# ГЛАВА 4. Реализация серверной и клиентской частей приложения.

Приложения разделено на две части, фронт и бэк. На фронте вызываются основные функции и происходит отрисовка изображений. На бэке эти вызовы обрабатываются.

Взаимодействие между фронтом и бэком происходит по средствам FastAPI.

Файлы и директории, относящиеся к бэку:

* sqlite\_base.db – БД
* src/crud.py – файл с описанием всех обращений к БД
* src/database.py – описание подключения к БД
* src/schemas.py – описание схем объектов БД для работы с pydentic [4].
* src/models.py – описание моделей SQLAlchemy.
* src/main.py – файл с описанием всех функций, в том числе запросов FastAPI.

# Заключение

В ходе курсовой работы была реализована двумерная игра, использующая ООП. Были выполнены все поставленные задачи:

1. Разработана идею игры.
2. Написан движок для её реализации.
3. Реализована игру.

Приложение имеет потенциал в дальнейшем развитии, благодаря упрощенной генерации карт при помощи конструктора и гибкости движка в плане написания новых событий.

# Список литературы

1. FASTAPI
2. Алхимия
3. Алембик
4. pydentic

# ПРИЛОЖЕНИЕ 1

Формат заполнения документа, описывающий карту:

Первая строка- строка комментария

Строки со 2 по 12 – строки массива карты

Строка 13 – первые два числа – координаты старта, третье число – скорость игрока

Пример:

строка комментария

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

0 0 30

# ПРИЛОЖЕНИЕ 2

Формат документа, описывающий события карты

Пример:

Первая строка пропускается

Класс ивента 1;

Имя Ивента 1;

координаты ивента 1 (x y) ;

шаг ивента 1 ;

шаг ивента 2 ;

........

EVENT\_END

Класс ивента 2;

Имя Ивента 2;

координаты ивента 2 (x y) ;

шаг ивента 1 ;

шаг ивента 2 ;

........

EVENT\_END

...........

EVENT\_END

EVENTS\_END