

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
“ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ”

Факультет компьютерных наук

Кафедра программирования и информационных систем

Awareness

Веб-приложение для повышения самоосознанности

Курсовая работа

09.03.02 Информационные системы и технологии

Обучающийся _____ Рыбас Д.В., б.о, 3курс
Обучающийся _____ Соболев И.И., б.о, 3курс
Руководитель _____ В.С. Тарасов, старший
преподаватель
Преподаватель _____ И.В. Клейменов
практики

Воронеж 2022

Содержание

Введение	3
1. Постановка задачи	4
2. Анализ предметной области	6
2.1. Глоссарий	6
2.2. Анализ аналогов	8
2.3. Анализ потребности	10
3. Продуктовые воронки	11
4. Диаграммы	13
4.1. Диаграмма прецедентов	13
4.2. Диаграмма последовательностей	14
4.3. Диаграмма взаимодействий	16
4.4. Диаграмма состояний	17
4.5. Модель IDEF0	18
4.6. Схема базы данных	21
5. Реализация	22
5.1. Средства реализации	22
5.2. Frontend-составляющая приложения	22
5.3. Backend-составляющая приложения	23
5.4. Система регистрации	24
5.5. Описание функциональной части приложения и графического пользовательского интерфейса	25
6. Тестирование	29
Список используемой литературы:	33

Введение

Становление человека как самоосознанной личности, пожалуй, один из основных трендов последних лет. Раньше люди не стремились к этому так, как сейчас по причине не закрытых основных материальных потребностей. Им, в первую очередь, нужно было думать о том, как прокормить себя и свою семью. Сейчас же у многих людей эти потребности закрыты, вследствие чего появились новые - духовные. Современный человек ищет ответы на вопросы: «Как мне быть самим собой и понимать, что я делаю в тот или иной момент?» «Какие чувства я испытываю ежедневно?». Можно обратиться к психологу за помощью в поиске ответов на данные вопросы, но не все люди имеют средства на это, а те, кто обратился к нему, не могут держаться с ним на связи на протяжении всего дня

Из сказанного выше следует, что людям необходимо место, в котором они будут структурировать свои мысли, отслеживать привычки, повышать самоосознанность. Можно заниматься этим, используя листок бумаги и ручку, что является неудобным в силу отсутствия автоматизации, но в эпоху цифрового мира, можно исправить данный недостаток.

Сказанное выше позволяет сделать вывод, что разработка веб-приложения на эту тему имеет высокую актуальность по причине большой востребованности людей в этом.

В данном курсовом проекте будет описан процесс разработки такого веб-приложения.

1. Постановка задачи

Цель данного курсового проекта – разработка веб-приложения, являющегося персональным помощником пользователя в повышении его осознанности путем ведения дневника по шаблону, дневника ситуаций, дневника эмоций, а также отслеживания привычек.

Задачи, которые необходимо решить в ходе развития проекта:

- Сбор информации об исследуемой предметной области;
- Проведение анализа полученных данных;
- Создание диаграмм, отражающих основные аспекты приложения;
- Разработка веб-приложения;
- Проведение тестирования веб-приложения;
- Размещение приложения на сервере.

Разработка веб-приложения разделена на разработку серверной и клиентской частей.

Разработка серверной части:

- Проработка и реализация базы данных;
- Реализация работы с используемыми API;

- Формирование запросов к базе данных.
- Разработка клиентской части:
- Проработка дизайна веб-приложения;
- Формирование веб-страниц приложения;
- Реализация взаимодействия с серверной частью.

К приложению предъявляются следующие требования:

- Предоставить пользователю возможность ведения дневника по шаблону. Пользователю будет предлагаться определённый шаблон, состоящих из тем для размышления, в который он будет вписывать свои развернутые ответы;
- Предоставить пользователю возможности ведения дневника эмоций. Пользователю будет предлагаться список эмоций, в котором он будет выбирать те, что испытал за день и список действий, в котором он будет выбирать те, что сделал в этот день, что позволит пользователю связывать эмоции и действия;
- Предоставить пользователю возможность ведения дневника ситуаций. Постановка приложением пользователя мысленно в определённые жизненные ситуации с предложением вариантов ответа. При выборе определенного варианта ответа, пользователь

будет получать потенциальный исход, а также совет, следует ли ему поступить иначе и поменять свой выбор или остановиться на данном;

- Предоставить клиенту возможность узнать, сколько дней подряд он имеет привычку;
- Предоставить клиенту возможность создать обращение в техническую поддержку и узнать ответ на это обращение внутри приложения;
- Предоставить неавторизованному пользователю возможность регистрации, авторизации.

2. Анализ предметной области

2.1. Глоссарий

В курсовом проекте будут использоваться следующие термины:

- Администратор - Специалист, уполномоченный управлять и администрировать определенный участок IT-среды;
- Веб-приложение (сайт) - Клиент-серверное приложение, в котором клиент взаимодействует с веб-сервером при помощи браузера;
- Дневник по шаблону - Предложенные приложениям шаблоны для заполнения дневника;

- **Дневник ситуаций** - Мысленная постановка приложением пользователя в определённые жизненные ситуации с предложением вариантов ответа;
- **Дневник эмоций** - Заполнение пользователем эмоций из списка, испытанных за день и действий, которые он делал за этот день, что позволит связывать эмоции с действиями;
- **Пользователь** - Лицо, которое использует приложение для выполнения конкретной функции;
- **Репозиторий** - Место, где хранятся и поддерживаются какие-либо данные;
- **СУБД** - Система управления базами данных. Комплекс программно-языковых средств, позволяющих создать базы данных и управлять данными;
- **Яндекс-метрика** - Бесплатный интернет-сервис компании Яндекс, предназначенный для оценки посещаемости веб-сайтов и анализа поведения пользователей;
- **Android** - Операционная система для мобильных устройств;

- API - Программный интерфейс приложения. Описание способов, которыми одна компьютерная программа может взаимодействовать с другой программой;
- Bootstrap - Свободный набор инструментов для создания сайтов и веб-приложений;
- CSS - Формальный язык, служащий для описания оформления внешнего вида документа, созданного с использованием языка разметки (HTML, XHTML, XML);
- FAQ - Собрание часто задаваемых вопросов по какой-либо теме и ответов на них;
- Flask - Фреймворк для создания веб-приложений на языке программирования Python, использующий набор инструментов Werkzeug, а также шаблонизатор Jinja2;
- GitHub - Сервис для совместной разработки и хостинга проектов.
- HTML - Стандартизированный язык разметки веб-страниц во Всемирной паутине;
- JavaScript - Мультипарадигменный язык программирования, используется как встраиваемый язык для программного доступа к объектам приложений;

- Linux - Семейство Unix-подобных операционных систем на базе ядра Linux;
- PostgreSQL - Свободная объектно-реляционная система управления базами данных;
- Python - Высокоуровневый язык программирования общего назначения;
- REST - Архитектурный стиль взаимодействия компонентов распределённого приложения в сети;
- SQLAlchemy - Программная библиотека на языке Python для работы с реляционными СУБД с применением технологии ORM;
- Windows 10 – Операционная система для персональных компьютеров и рабочих станций, разработанная корпорацией Microsoft в рамках семейства Windows NT.

2.2. Анализ аналогов

Для формулирования более конкретных требований к веб-приложению необходимо было провести анализ аналогов. К ним можно отнести такие приложения как, например, AVE, VOS, Daylio.

AVE - это приложение, позволяющее пользователю повышать уровень самоосознанности.

Сильные стороны сервиса:

- Наличие заданий, которые пользователь ставит себе на день;
- Наличие ленты новостей, где пользователи делятся своими мыслями;
- Наличие возможности тренировки различных навыков, указанных в приложении;
- Наличие системы достижений.

Слабые:

- Наличие платной подписки;
- Система уровней, по причине которой невозможно сразу после установки приложения раскрыть для себя всю его функциональность.

VOS – это приложение, позволяющее пользователю повышать уровень самоосознанности.

Сильные стороны сервиса:

- Возможность ведения дневника;
- Возможность отслеживания собственных эмоций пользователем.

Слабые:

- Наличие платной подписки;

- Меньшая функциональность в сравнении с AVE.

Daylio – это приложение, позволяющее пользователю отслеживать собственные эмоции.

Сильные стороны сервиса:

- Возможность отслеживания эмоций;
- Возможность увидеть график настроения.

Слабые:

- Наличие платной подписки;
- Меньшая функциональность в сравнении с вышеперечисленными аналогами.

2.3. Анализ потребности

После изучения аналогов разрабатываемого проекта можно сделать вывод, что ни один из них не является идеальным решением поставленной задачи. Но существование большого количества сервисов для повышения самоосознанности говорит о том, что пользователь нуждается в этом. Именно поэтому разработка проекта Awareness является актуальной. Команде разработчиков необходимо учесть достоинства аналогов, а также их недостатки, которые нужно попытаться исправить.

3. Продуктовые воронки

В данное веб приложение при помощи сервиса Яндекс.Метрика были добавлены следующие продуктовые воронки. Они позволят определить заинтересованность пользователей в различных составляющих веб-приложения.

Создание дневника по шаблону

Всего визитов:
1 ➔ 100 %

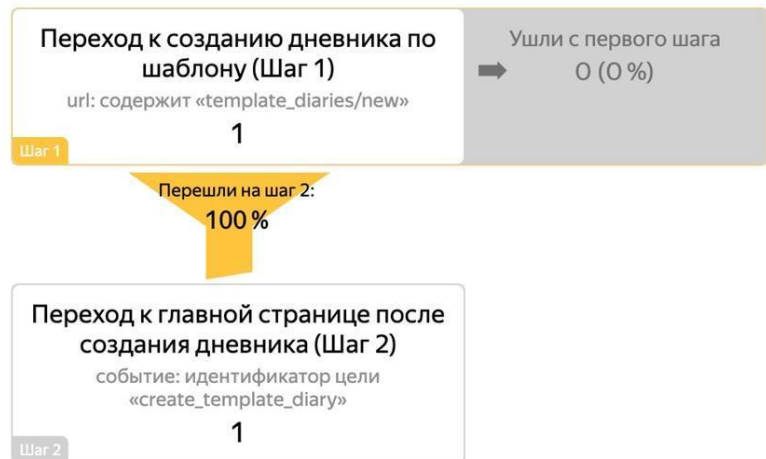


Рисунок 1 - Продуктовая воронка для попытки к созданию дневника по шаблону

Регистрация

Всего визитов:
Нет данных по цели.

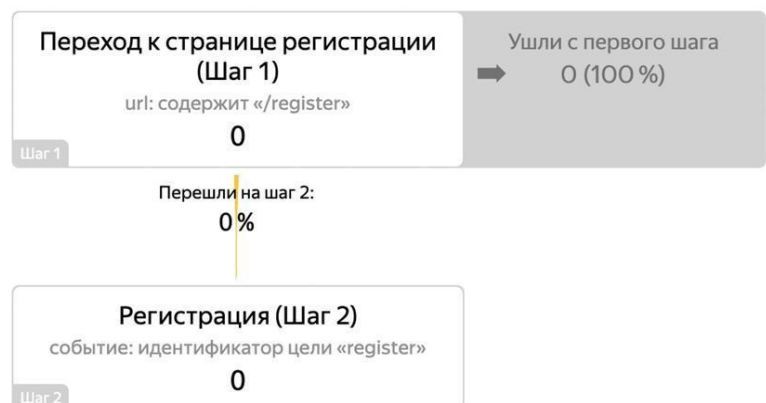


Рисунок 2 - Продуктовая воронка для перехода на страницу регистрации

Авторизация

Всего визитов:

1 ➔ 100 %

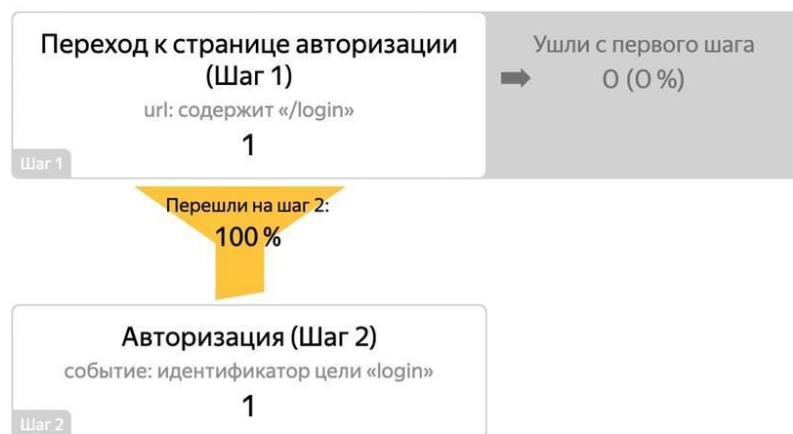


Рисунок 3 - Продуктовая воронка для перехода на страницу авторизации

4. Диаграммы

4.1. Диаграмма прецедентов

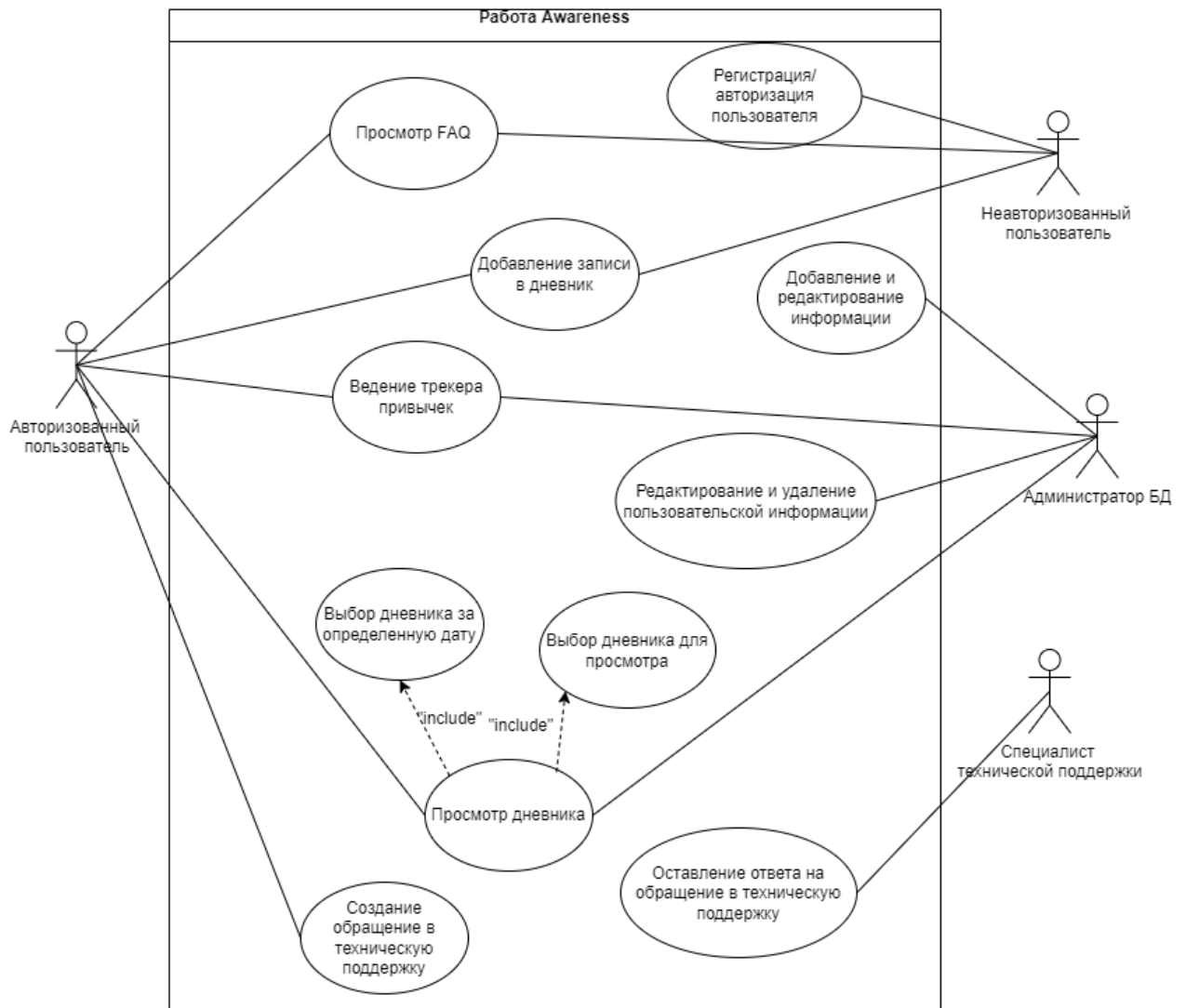


Рисунок 4 - Диаграмма прецедентов (use-case)

На рисунке 4 представлены функции пользователей в веб приложении Awareness.

Неавторизованный пользователь: регистрация/авторизация в системе, просмотр FAQ, заполнение дневника без возможности его сохранения;

Авторизованный пользователь: просмотр дневников через веб-приложение за определенную дату, просмотр FAQ, создание обращение в техническую поддержку, добавление записи в дневник, ведение трекера привычек;

Специалист технической поддержки: оставление ответа на обращение пользователя;

Администратор БД: добавление и редактирование информации, редактирование и удаление пользовательской информации, просмотр информации о дневниках и трекера привычек пользователя.

4.2. Диаграмма последовательностей

Диаграмма последовательностей служит для описания взаимодействия объектов системы по мере их проявления в сценарии. В данном случае имеем 3 основных сценария.



Рисунок 5 - Диаграмма последовательностей для авторизации

Неавторизованный пользователь вводит данные для входа (логин и пароль) в форму приложения, которое отправляет запрос в БД и в случае положительного ответа позволяет пользователю войти в приложение (с возвратом на главную страницу), в противном случае будет предложено ввести данные заново.

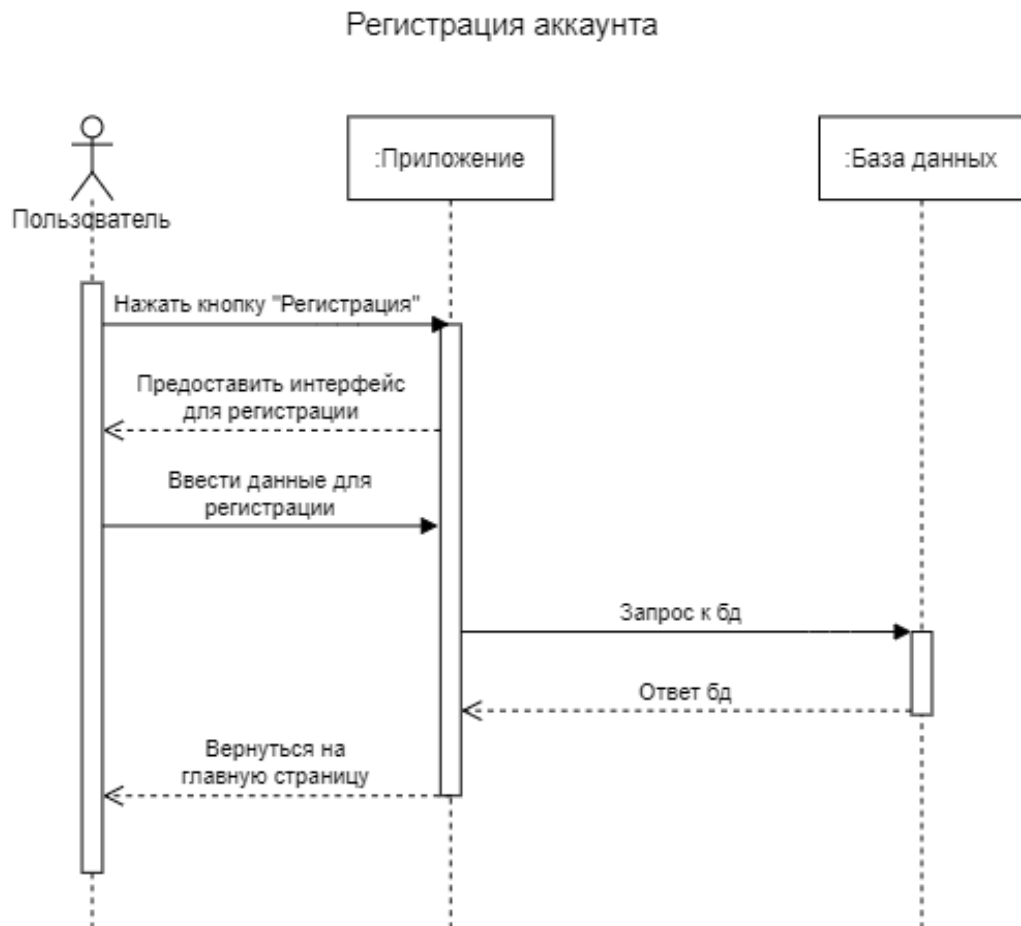


Рисунок 6 - Диаграмма последовательности для регистрации

Незарегистрированный пользователь после нажатия на кнопку регистрации получает от приложения форму для ввода данных регистрации. После отправки этих данных приложение делает запрос к БД и в случае успешного ответа (т.е. данные введены корректно и не совпадают с существующим) пользователь получает зарегистрированный аккаунт.



Рисунок 7 - Диаграмма последовательностей для заполнения дневника

Пользователь выбирает необходимый ему дневник, после чего приложение предлагает пользователю заполнить его. Пользователь вводит данные, после чего приложение посылает запрос в базу данных. В случае положительного ответа в БД записывается информация о данных дневника.

4.3. Диаграмма взаимодействий

Данная диаграмма определяет возможные способы взаимодействия пользователей с системой. Позволяет видеть все взаимодействия запросов в системе, служит дополнением к диаграмме последовательностей.

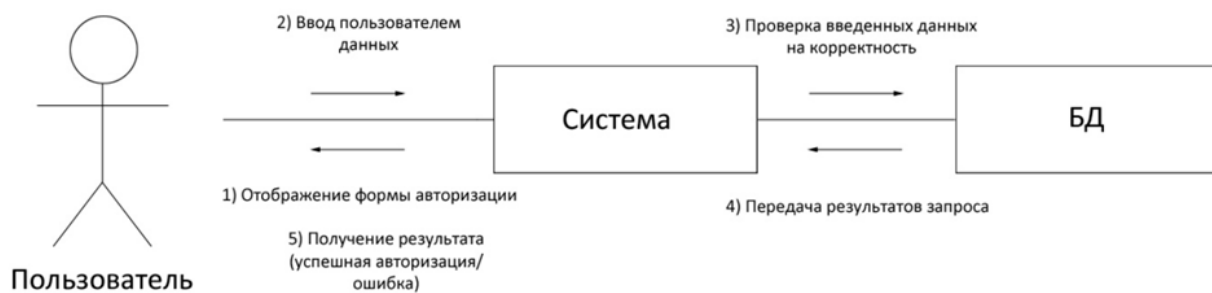


Рисунок 8 - Диаграмма взаимодействия для авторизации

4.4 Диаграмма состояний

Данные диаграммы описывают переход объектов из одних состояний в другие.

Диаграмма состояния для пользователя

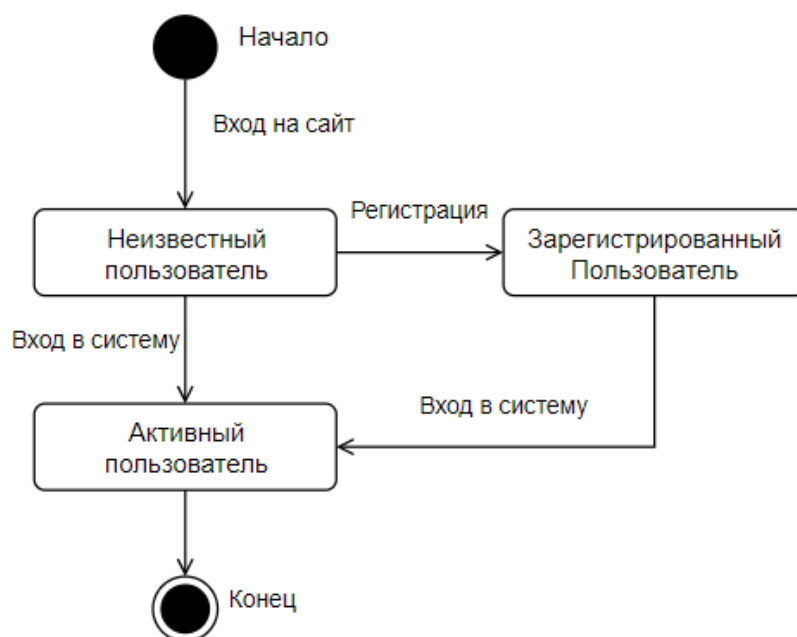


Рисунок 9 - Диаграмма состояний для авторизации

Диаграмма состояния для дневника

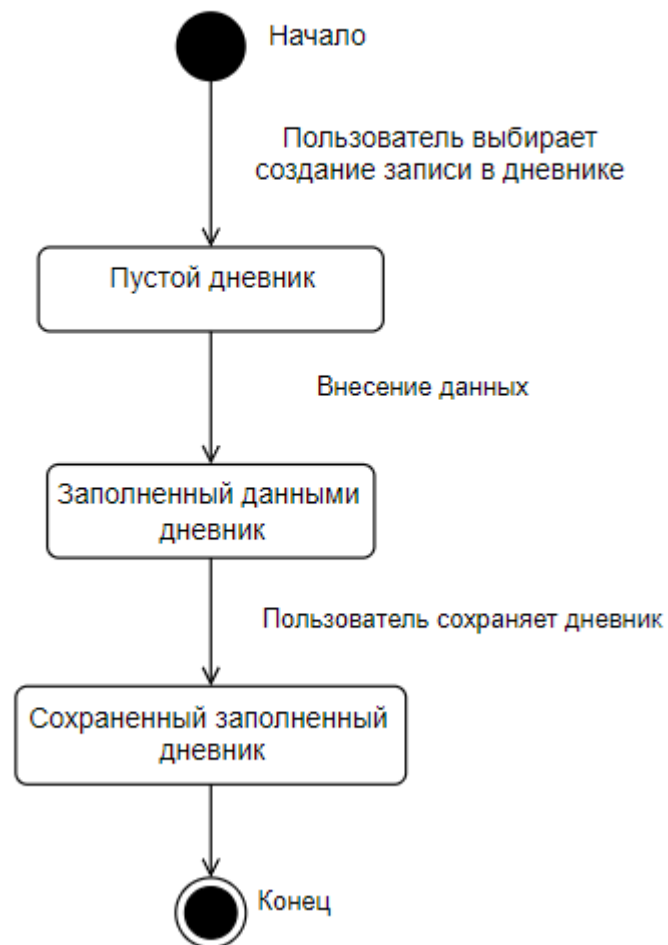


Рисунок 10 - Диаграмма состояний для дневника

4.5. Модель IDEF0

IDEF0 модель отображает структуру и функции приложения, а также потоки информации и материальных объектов, преобразуемые этими функциями.

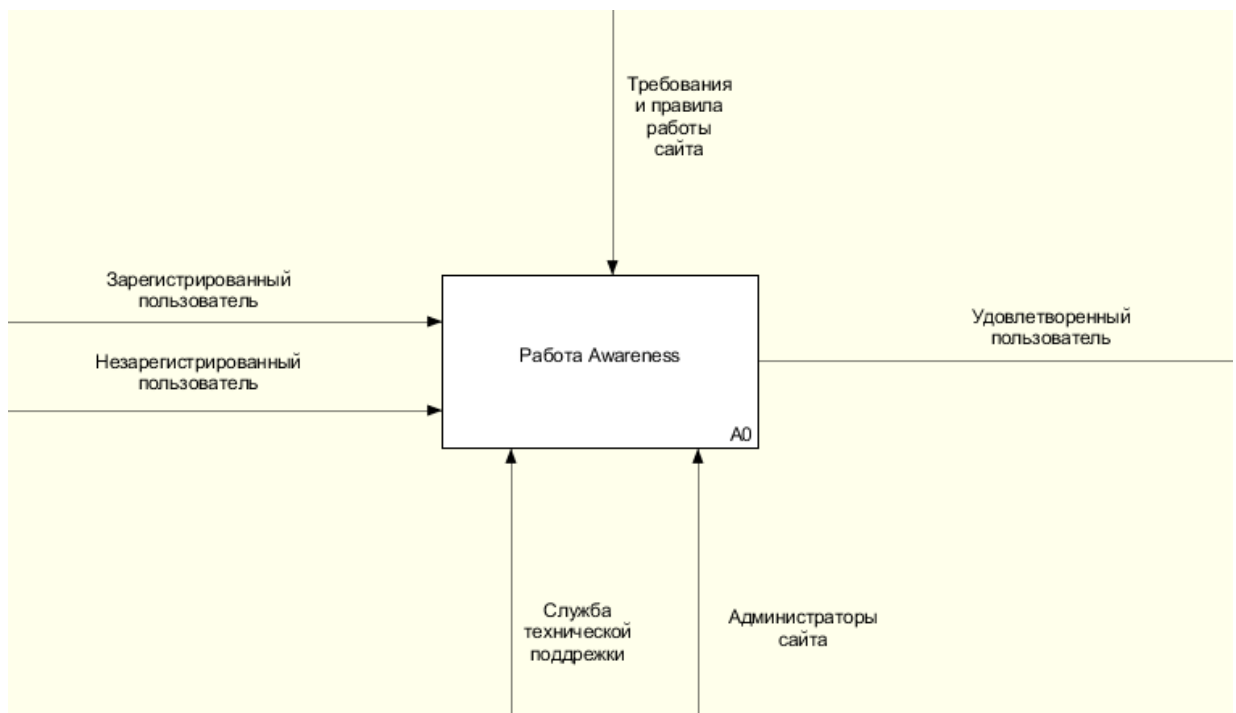


Рисунок 11 - Модель IDEF0 – уровень A-0

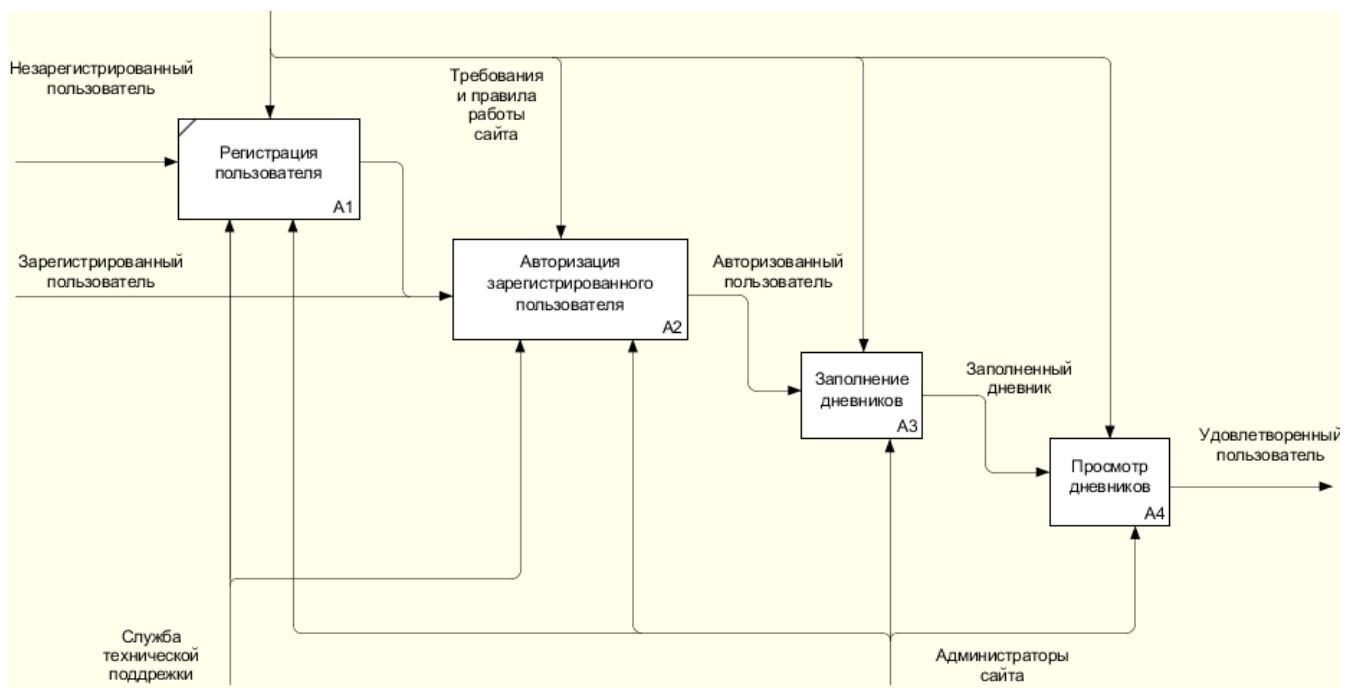


Рисунок 12 - Модель IDEF0 – уровень A0

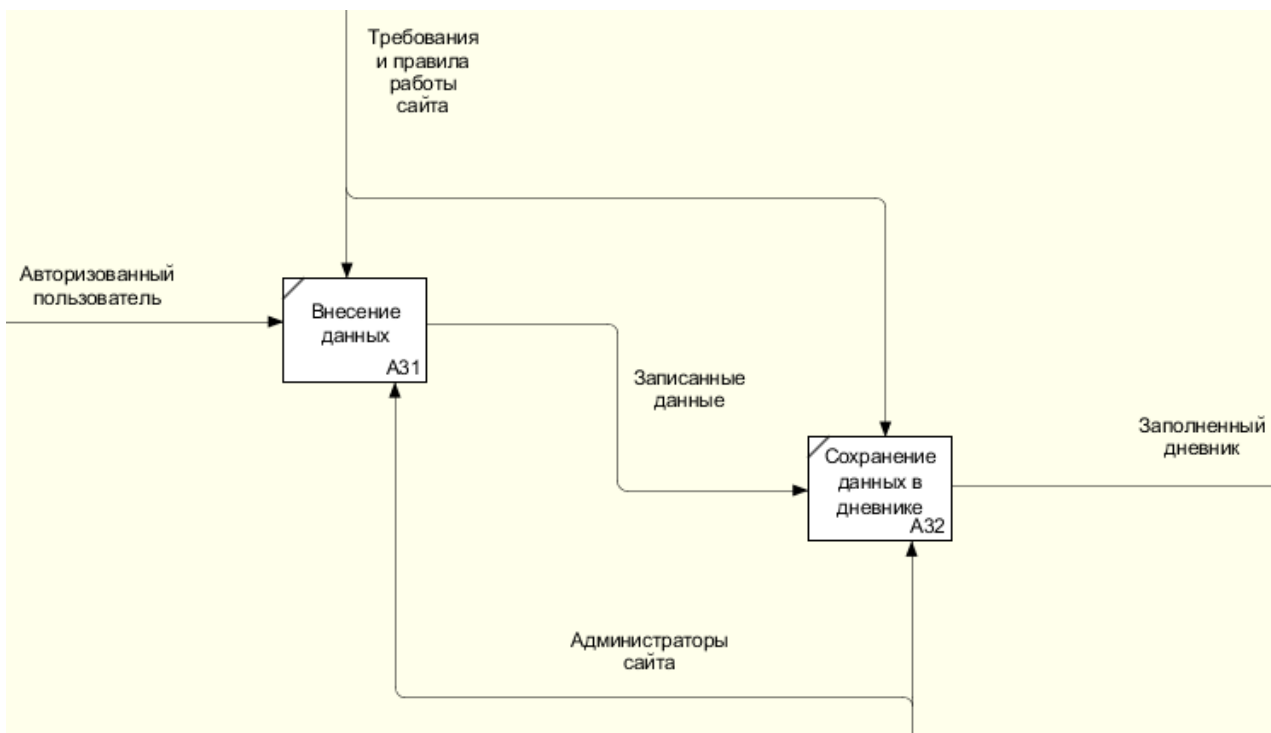


Рисунок 13 - Модель IDEF0 – уровень А3 (заполнение дневников)

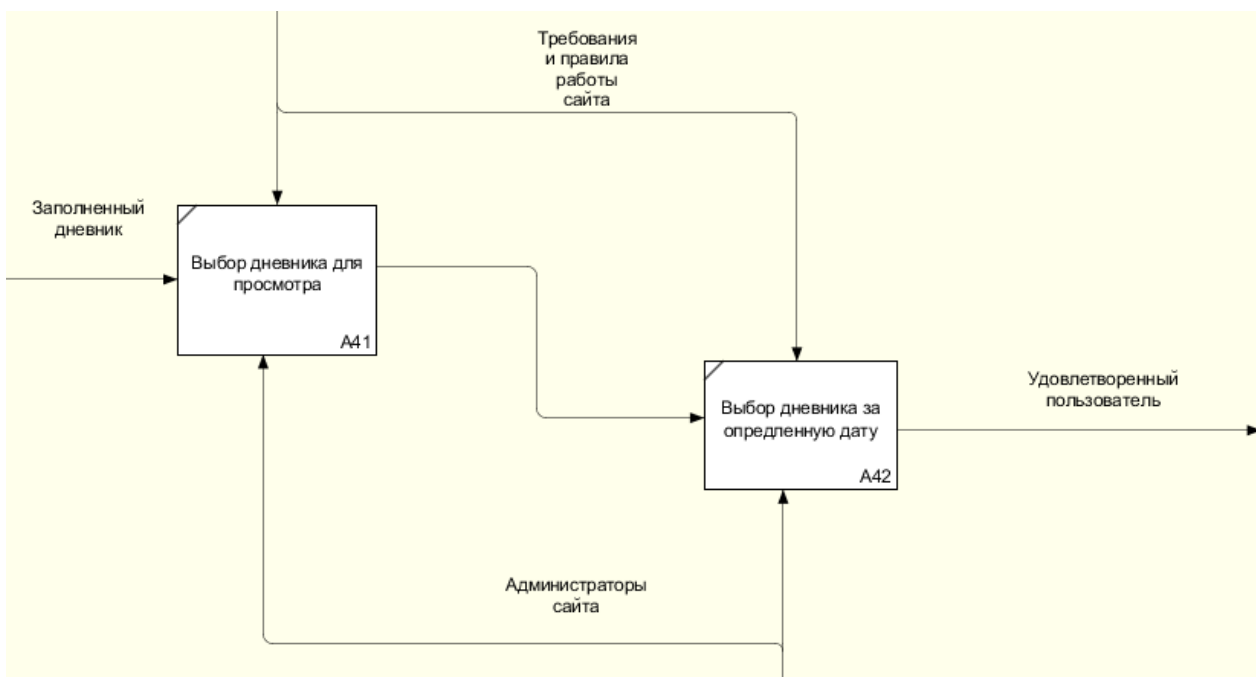


Рисунок 14 - Модель IDEF0 – уровень А4 (просмотр дневников)

4.6. Схема базы данных

Данная диаграмма является представлением базы данных в виде наглядных графических диаграмм.

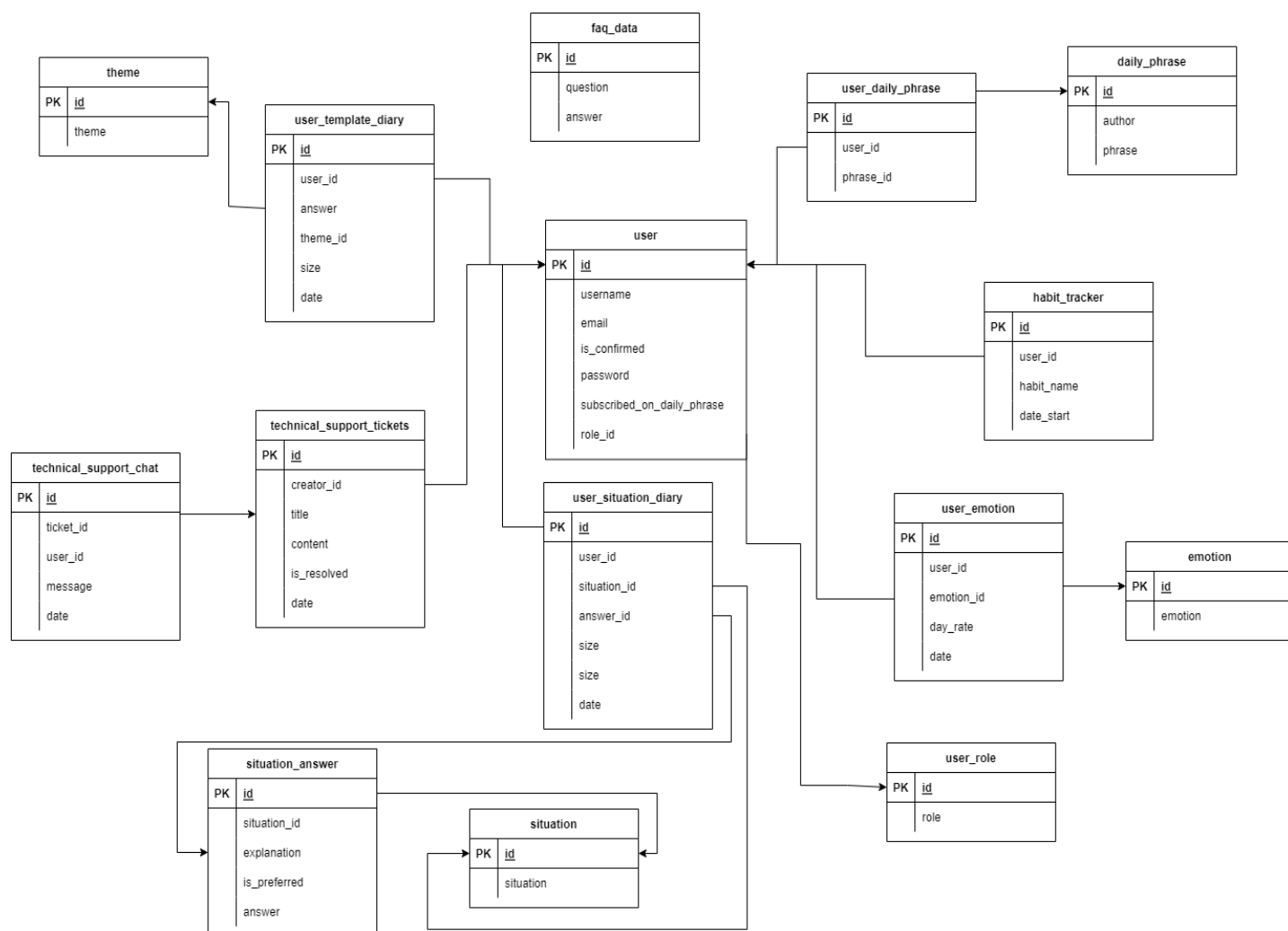


Рисунок 15 – Схема базы данных

5. Реализация

5.1. Средства реализации

В качестве средств реализации были использованы:

- Язык программирования Python 3, так как к нему можно подключить множество дополнительных модулей и библиотек. Python 3 является проверенным решением на рынке для подобных приложений;
- Фреймворк Flask для языка Python, позволяющий создавать легковесные веб-приложения. Выбран был по причине возможности быстрого разворачивания приложения с его помощью, а также большого количества пакетов-расширений;
- База данных PostgreSQL была выбрана по причине своей объектно-реляционной модели, хорошей масштабируемости, а также большому количеству графических интерфейсов взаимодействия с ней, к примеру, IntelliJ Idea Datagrip;
- Фреймворк Vue языка Javascript, позволяющий создавать пользовательский интерфейс. Был выбран по причине возможности быстрого разворачивания UI с его помощью;
- SQLAlchemy, так как эта ORM позволяет взаимодействовать с базой данных с помощью Python-кода.

Для развертывания приложения использована платформа Heroku.

5.2. Frontend-составляющая приложения

Frontend-составляющая приложения реализована на языке HTML с использованием CSS, а также языка Javascript, в частности фреймворк Vue.js. Для оформления страниц использовался фреймворк Bootstrap.

5.3. Backend-составляющая приложения

В качестве фреймворка для создания приложения используется Flask. На рисунке 16 показано, как части Flask взаимодействуют друг с другом.

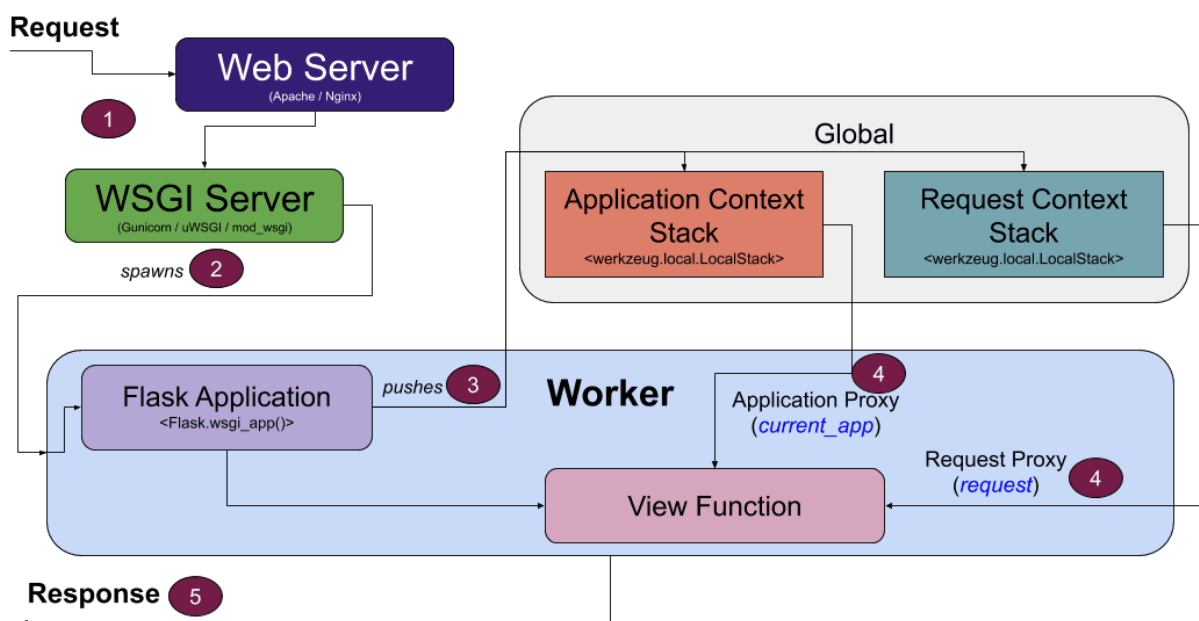


Рисунок 16 – Схема функционирования фреймворка Flask

1. HTTP - запросы приходят на Веб-сервер, который маршрутизирует их на WSGI - сервер. Apache и Nginx - популярные веб - серверы, а Gunicorn, uWSGI, mod_wsgi - популярные WSGI серверы;
2. WSGI - сервер порождает поток для обработки запроса. Каждый такой поток обрабатывает только один запрос за раз;

3. Приложение Flask создаёт контексты приложения и запроса и кладёт их в соответствующие стеки. В контексте приложения хранятся данные уровня приложения, то есть различные переменные конфигурации, подключение к базе данных. Контекст запроса хранит данные, необходимые для обработки запроса;
4. View Function получает контекст запроса и контекст приложения с помощью прокси - объектов, после чего выполняет необходимые операции с ними и генерирует ответ;
5. После генерации ответа контексты приложения и запроса убираются из соответствующих стеков.

В качестве базы данных используется PostgreSQL.

5.4. Система регистрации

Ниже перечислены ключевые концепты системы регистрации:

- С помощью модуля itsdangerous и секретного ключа из конфигурации приложения генерируются уникальные для каждого пользователя токены, с помощью которых происходит валидация пользователей при регистрации. Токены активны в течение 30 минут после отправки запроса о регистрации.
- При регистрации пароль пользователя хешируется алгоритмом SHA256, после чего записывается в базу в виде `sha256$salt$hash`, где `salt` - случайно сгенерированная строка для обеспечения большей устойчивости пароля от подбора, а `hash` - хешированный пароль.

5.5. Описание функциональной части приложения и графического пользовательского интерфейса

На рисунке 17 изображена главная страница веб-приложения.

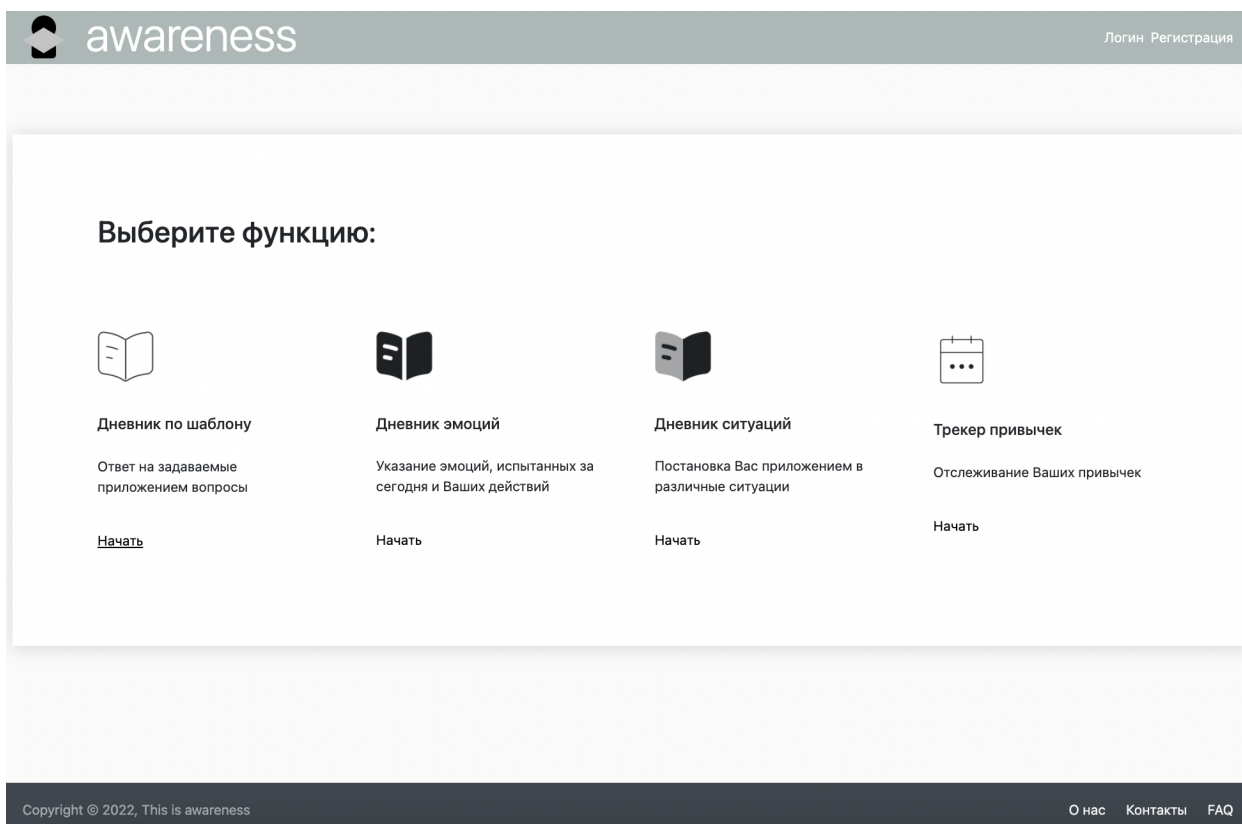


Рисунок 17 - Главная страница

При помощи навигационной панели, содержащейся на всех страницах, неавторизованный пользователь может перейти на страницы авторизации и регистрации, а на нижней навигационной панели также на вкладку FAQ, где он может найти ответ на интересующий его вопрос.

Авторизованный пользователь может просмотреть свои дневники, перейти на страницу выбора просмотра дневников и настроек профиля.

На странице, изображенной на рисунке 21, пользователь сможет просмотреть нужный ему дневник, выбрав его из списка.

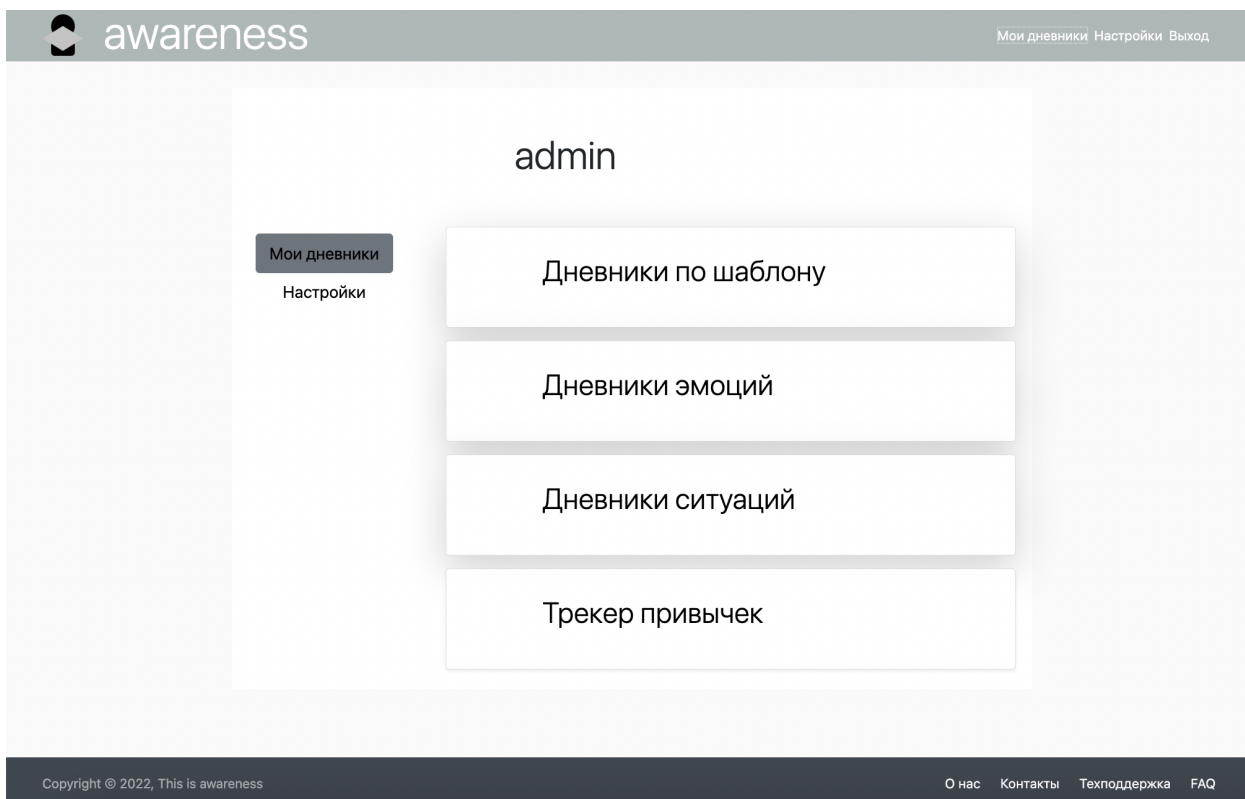


Рисунок 21 - Страница выбора дневника для просмотра

На странице настроек пользователь может изменить свой пароль от аккаунта. На рисунке 22 изображена страница заполнения дневника по шаблону. При переходе на данную страницу пользователя предлагается описать одну тему, выбранную случайным образом из БД.

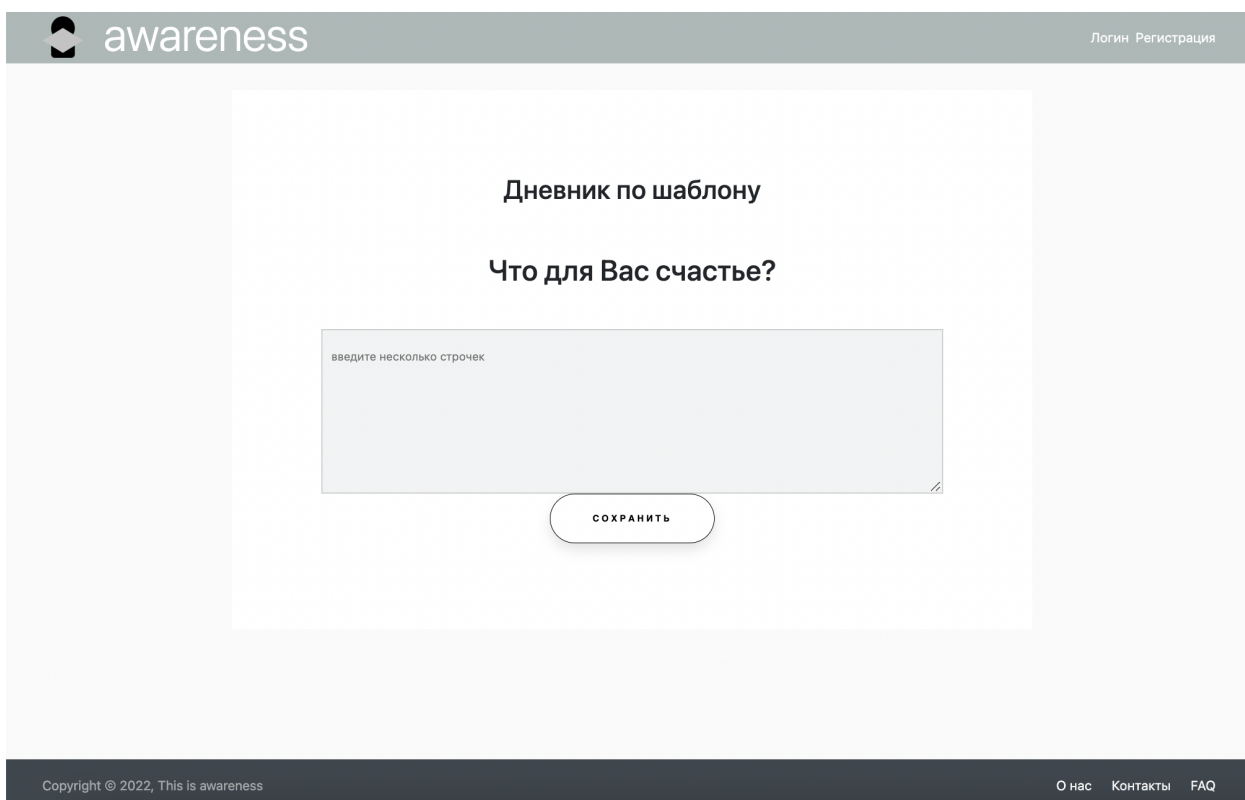



Рисунок 22 - Страница заполнения дневника по шаблону

Заполнение дневника эмоций пользователем происходит в 2 этапа. На рисунке 23 изображена страница, на которой пользователь выбирает те эмоции, которые он испытывал за этот день, а также выбор тех действий и дел, которые он сегодня сделал на рисунке 24.

 awareness

Мои дневникиНастройкиВыход

Выберите из списка эмоции, которые вы испытали сегодня:

Эмоции

☒ Радость

☐ Счастье

☐ Грусть

☒ Печаль

Рисунок 23 - Страница заполнения дневника эмоций (выбор эмоции)

☐ Тоска

Действия

☐ Спорт

☐ Общение с близкими

☐ Прогулка

☐ Учёба

Оценка дня:
★★★★★

СОХРАНИТЬ

Рисунок 24 - Страница заполнения дневника эмоций (выбор действия)

Пользователь имеет возможность ведения дневника ситуаций, в котором предлагается выбор поступка в определенной ситуации, а также выводится верный вариант поведения в ситуации. Страница заполнения дневника ситуаций представлена на рисунке 25.

Дневник ситуаций

Варианты ответа:

- ☒ Оставить всё, как есть. Пусть извиняется сам
- ☐ Сделать первый шаг
- ☐ Навсегда забыть об этом человеке

Выбран: Оставить всё, как есть. Пусть извиняется сам
Правильный ответ: Сделать первый шаг

СОХРАНИТЬ

ПРАВИЛЬНЫЙ ОТВЕТ

Рисунок 25 - Страница заполнения дневника ситуаций

Пользователь имеет возможность узнать, сколько дней подряд он имеет привычку с помощью трекера привычек. Страница добавления новой привычки представлена на рисунке 26.

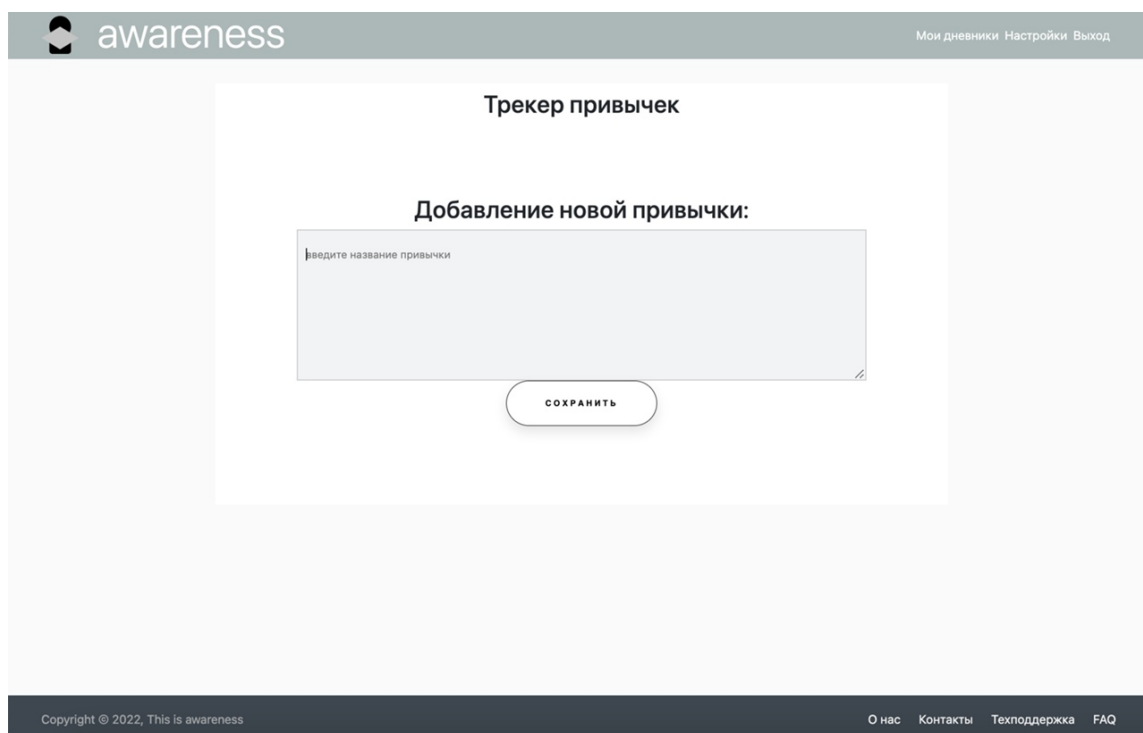


Рисунок 26 - Страница добавления новой привычки

6. Тестирование

После работ по созданию приложения было произведено тестирование его backend сервисов. Результаты тестирования представлены в таблице 1.

Тип запроса	Функция	Результат тестирования
POST	Регистрация пользователя	Успешно
POST	Авторизация пользователя	Успешно
POST	Изменение пароля для страницы пользователя	Успешно
GET	Выбор просмотра дневника по шаблону из списка дневников	Успешно
POST	Заполнение данных дневника эмоций	Успешно
GET	Выбор просмотра определенного дневника ситуаций по id	Успешно
GET	Выбор просмотра определенного дневника по шаблону по id	Успешно

POST	Обработка поступивших данных по созданию новой привычки в трекере привычек	Успешно
GET	Вывод данных об определенной привычке в трекере привычек	Успешно
GET	Выбор просмотра определенного дневника эмоций по id	Успешно
GET	Выбор просмотра дневника эмоций из списка дневников	Успешно
GET	Выбор просмотра дневника ситуаций из списка дневников	Успешно

Заключение

В ходе выполнения данной курсовой работы были выполнены следующие задачи:

- Собрана необходимая информация об исследуемой предметной области;
- Проведен анализ полученных данных;
- Созданы диаграммы, отражающие основные аспекты приложения;
- Разработано веб-приложение;
- Проведено тестирование веб-приложения;
- Приложение было размещено на сервере.

Цель курсовой работы считается достигнутой, разработанное веб-приложение удовлетворяет поставленным перед ним требованиям:

- Пользователь имеет возможность ведения дневника по шаблону, состоящий из тем для размышления, в который он вписывает свои развернутые ответы;
- Пользователь имеет возможность ведения дневника эмоций, в котором предлагается выбор списка эмоций, которые он испытал за день и список действий, которые он сделал в этот день.
- Пользователь имеет возможность ведения дневника ситуаций, в котором предлагается выбор поступка в определенной ситуации, а также выводится верный вариант поведения в ситуации.
- Пользователь имеет возможность узнать, сколько дней подряд он имеет привычку;

- Пользователь имеет возможность создать обращение в техническую поддержку и узнать ответ на это обращение внутри приложения;
- Неавторизованный пользователь имеет возможность регистрации, авторизации.

Список используемой литературы:

1. Bootstrap 5.2 Documentation [сайт] – URL: <https://getbootstrap.com/docs/5.2/> (дата обращения: 25.03.2022 – 7.06.2022).
2. Flask Documentation [сайт] – URL: <https://flask-doc.readthedocs.io/en/latest/> (дата обращения: 25.03.2022 – 7.06.2022).
3. Heroku Documentation [сайт] - URL: <https://devcenter.heroku.com/categories/reference/> (дата обращения: 25.03.2022 – 7.06.2022).
4. Vue.js Documentation [сайт] - URL: <https://vuejs.org/guide/introduction/> (дата обращения: 25.03.2022 – 7.06.2022).
5. Postgre SQL Documentation [сайт] - URL: <https://www.postgresql.org/docs/> (дата обращения: 25.03.2022 – 7.06.2022).