|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Продольный бланк_ВятГУ_распорядительный акт+ | | | | |
|  | |  | | |
|  |  | |  |  |
|  |  | |  |  |
|  | | | | |
| **ТЕХНИЧЕСКИЙ ПРОЕКТ**  веб-приложения “NeuroKanban”  2023 | | | | |
|  | | | | |

**Содержание**

[**1 Описание проекта 3**](#_gjdgxs)

[**2. Основные функциональные возможности 4**](#_30j0zll)

[**3. Уточнение структуры данных 6**](#_1fob9te)

[**4. Формы представления данных 7**](#_3znysh7)

[**5. Разработка алгоритма решения задачи 8**](#_2et92p0)

[**6. Определение языка, структуры программы и требований к техническим средствам 13**](#_tyjcwt)

[**7. Требования к техническим средствам: 15**](#_1t3h5sf)

# **Описание проекта**

Название проекта: Интернет-сервис NeuroKanban для организации работы проектной группы над конкретными проектами посредством парадигмы канбан.

Цель проекта: разработка аналога популярных Интернет-сервисов, ушедших с Российского рынка, а также соответствие требованиям FOSS и работа внутри организации.

## **Основные функциональные возможности**

NeuroKanban должен обеспечивать возможности:

* авторизация в системе посредством логина и пароля;
* наличие ролей руководителя проекта и участника;
* навигация на сайте посредством пользовательского интерфейса;
* создание и редактирование проектов в системе;
* приглашение пользователей в проект от руководителей проектов;
* удаление пользователей из проекта;
* создание и редактирование «досок» внутри проектов;
* создание и редактирование групп задач (столбцов) на «досках»;
* создание, редактирование, перемещение задач между группами на «досках», а также установление различных характеристик задач, таких как трудоемкость (крайняя дата выполнения, уже затраченное время исполнения);
* обеспечение напоминаний о заканчивающемся сроке для выполнения задач;
* информирование руководителей проекта о статусе выполнения задач.

Бизнес-логика и ее декомпозиция представлены на рисунках 1 и 2.

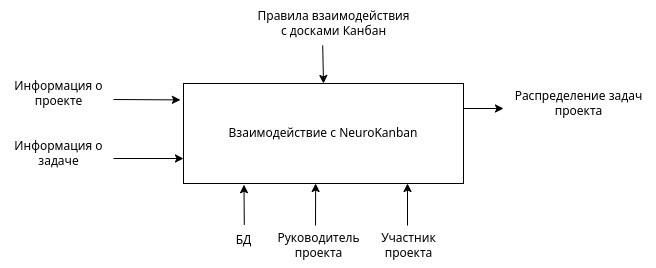


Рисунок 1 - функциональная диаграмма IDEF0

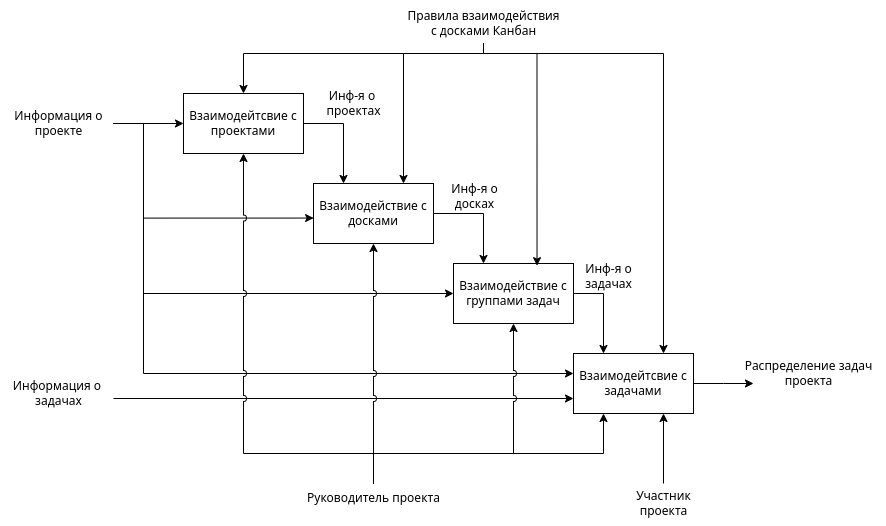


Рисунок 2 - декомпозиция IDEF0

## **Уточнение структуры данных**

Используемые структуры данных представлены посредством ER-диаграммы, изображенной на рисунке 3.

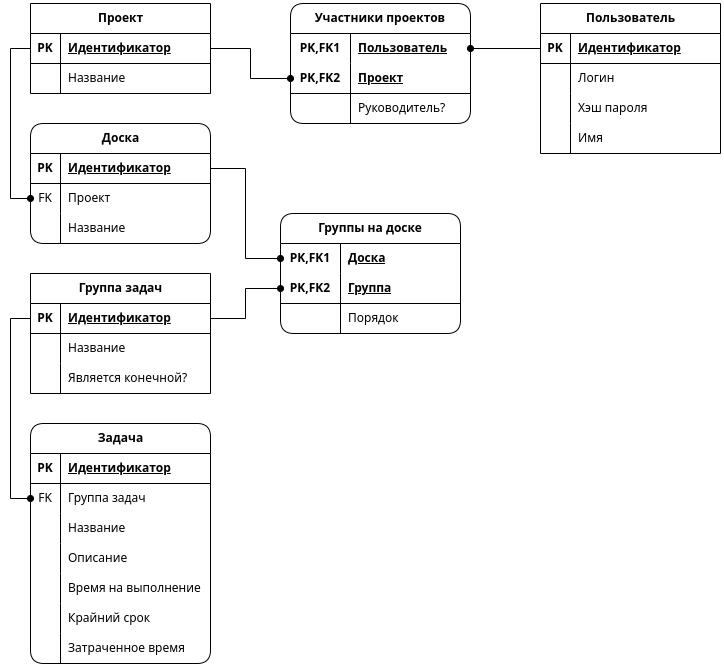


Рисунок 3 - концептуальная ER-диаграмма

## **Формы представления данных**

При попадании в систему, пользователь в первую очередь выполняет вход в систему посредством ввода логина и пароля в соответствующих полях ввода.

После входа пользователь попадает на экран с проектами, где может быть выбран существующий проект, либо создан новый.

После выбора проекта, пользователь попадет на экран «досок» проекта, где можно выбрать существующую «доску», либо создать ее, а также пригласить участника в проект или удалить участника. После выбора «доски» проекта, происходит переход на экран выбранной «доски», где можно создать новую группу задач, создать новую задачу в любой из групп, взаимодействовать с созданными задачами.

## **Разработка алгоритма решения задачи**

В системе возможны три роли: администратор, руководитель проекта, участник проекта.

Добавление пользователей:

* Первичные пользователи добавляются администраторами (сразу после развертывания системы), посредством прямого доступа к СУБД к таблице “Пользователи”;
* Другие же пользователи получают доступ к системе посредством приглашения другими пользователями в проекты.

Авторизация пользователей:

* для авторизации в системе пользователь должен предоставить на экране входа полученные данные для входа от администратора, если является одним из первых пользователей, либо от другого пользователя, что приглашает его в проект в системе;
* после авторизации пользователь попадает на экран проектов;

Дальнейшие действия доступны только авторизованным пользователям.

Добавление проекта:

* пользователи могут создать новый проект, нажав соответствующую кнопку на экране проектов;
* при создании проекта указывается его название, а затем нажимается кнопка добавления проекта;
* после создания проекта, пользователь становится его руководителем;

Редактирование проекта:

* редактировать проект может только руководитель данного проекта, нажав на соответствующую кнопку на экране проектов на необходимой проекте;
* при редактировании доступно изменение названия проекта;

Удаление проекта:

* удалять проект может только руководитель данного проекта, нажав на кнопку редактирования на экране проектов на необходимом проекте;
* на форме редактирования проекта должна присутствовать кнопка удаления данного проекта, нажатием на которую будет происходить удаление проекта из системы;

Добавление доски:

* пользователь должен перейти в проект, нажав на него на экране проектов, а затем по нажатию соответствующей кнопки может создать доску, если он является руководителем этого проекта;
* при создании доски указывается ее название, а затем нажимается кнопка создания доски;

Редактирование доски:

* редактировать доску может только руководитель данного проекта, нажав на соответствующую кнопку на экране выбранного проекта на необходимой доске;
* при редактировании доступно изменение названия доски;

Удаление доски:

* удалять доску может только руководитель данного проекта, нажав на кнопку редактирования на экране проекта на необходимой доске;
* на форме редактирования доски должна присутствовать кнопка удаления данной доски, нажатием на которую будет происходить удаление доски из проекта;

Добавление группы задач:

* создать группу задач можно после перехода на экран проекта, а затем на экран доски проекта;
* создать группу задач может только руководитель данного проекта, нажав на соответствующую кнопку на экране доски проекта;
* при создании указывается название группы и является ли эта группа конечной в доске;

Редактирование группы задач:

* редактировать группу задач может только руководитель данного проекта, нажав на соответствующую кнопку на экране доски проекта на необходимой группе;
* при редактировании доступно изменение названия группы задач и то, является ли группа конечной на доске задач;

Удаление группы задач:

* удалять группу задач может только руководитель данного проекта, нажав на кнопку редактирования на экране доски проекта на необходимой группе;
* на форме редактирования группы должна присутствовать кнопка удаления группы задач, нажатием на которую будет происходить удаление группы с доски;

Добавление задачи:

* добавить задачу можно после перехода на экран проекта, а затем на экран доски проекта;
* добавить задачу может только руководитель данного проекта, нажав на соответствующую кнопку на элементе доски проекта - группе задач;
* при добавлении задачи необходимо указать ее название и крайний срок выполнения;
* необязательными для указания являются описание задачи, время на ее исполнение;

Редактирование задачи:

* редактирование задачи может выполнять только руководитель проекта;
* редактировать задачу можно по нажатию на кнопку редактирования на самой задаче, размещенной на доске задач;
* при редактировании могут изменяться все данные задачи, что были внесены при прошлом редактировании или добавлении задачи;

Удаление задачи:

* удаление задачи может выполнять только руководитель проекта;
* удалять задачу можно по нажатию на кнопку редактирования на самой задаче, размещенной на доске задач;
* на форме редактирования задачи должна присутствовать кнопка удаления задачи, нажатием на которую будет происходить удаление задачи с доски;

Перемещение задачи между группами задач:

* перемещать задачу между группами на доске задач может каждый участник проекта;
* при перемещении необходимо указать затраченное время на выполнение данной задачи;

Приглашение участников в проект:

* может выполнить только руководитель проекта, нажав на соответствующую кнопку на экране проекта;
* при добавлении новых участников в проект, необходимо указать электронный адрес нового участника, куда в автоматизированная система отправит данные для входа в систему;
* если же этот человек уже есть в системе, то при указании почты участника, он будет автоматически добавлен в проект;

Удаление участников из проекта:

* может выполнить только руководитель проекта, нажав на кнопку участников на экране проекта;
* на экране участников будет возможность как просмотреть всех участников проекта, так и удалить любого из них при нажатии на соответствующую кнопку на карточке участника.

## **Определение языка, структуры программы и требований к техническим средствам**

Выбранным архитектурным решением является использование архитектуры “Клиент-Сервер” с использованием СУБД. Серверная часть обрабатывает запросы от клиентов, работает с базой данных и возвращает результаты запросов клиентам. Клиентская часть делает запросы к серверной части, получает ответ от сервера и выводит его в нужном виде на интерфейсе сайта.

Данное решение подразумевает, что клиентов, что одновременно взаимодействуют с сервером будет множество, в то время как взаимодействовать с СУБД будет лишь один бэкенд сервер (серверная часть веб-сайта).

* Клиентская часть (Фронтенд):
  + язык программирования TypeScript. Выбран в связи с наличием статической типизации и большим набором библиотек для реализации клиентской части веб-приложений;
  + библиотека Socket.io. Необходима для загрузки обновлений с сервера, не используя перезагрузку страницы;
  + фреймворк React. Выбор пал на него, так как он имеет виртуализацию DOM, что позволяет делать веб-страницы без загрузок новых страниц, а также компонентный подход к реализации;
  + веб-сервер nginx. Выбран так как является простым в развертывании, а также благодаря простой поддержке как обычных, так и WebSocket соединений.
* Серверная часть (Бэкенд):
  + язык программирования TypeScript;
  + фреймворк express.js. Был выбран, так как является гибким, быстрым в маршрутизации, имеющим хорошую поддержку REST API;
  + библиотека Socket.io. Необходима для поддержки WebSocket соединения с клиентом;
  + СУБД SQLite. Выбрана так как приложение не имеет необходимости в хранении и обработке больших массивов данных, а также нет необходимости доступа к базе данных с разных серверов компаний серверной части приложения, так как для каждой компании будет реализована отдельная серверная часть.

Как серверная, так и клиентская часть развертываются за счет Docker-контейнеров и инструмента Docker Compose, позволяющего разворачивать приложения с несколькими контейнерами. Данная система развертывания выбрана так как благодаря ней появляется возможность развертывания на многих конфигурациях серверов, использующих различные операционные системы, так как работа происходит при помощи контейнеризации.

## **Требования к техническим средствам:**

Серверная часть:

* Операционная система: GNU/Linux Debian 12,
* ОЗУ: 1 ГБ либо более;
* Жесткий диск: 5 ГБ либо более;
* Быстрое и надежное сетевое подключение с высокой скоростью передачи данных.

Клиентская часть:

* Устройство, поддерживающее браузер (смартфон, ПК);
* Доступ к интернету для взаимодействия с веб-сайтом.