* + 1. УТВЕРЖДЁН

ХХХ.ХХХХХХХХ.ХХХХХ-01 90 01-ЛУ

**Распределённая учётная система «Бюджет и задачи»**

**Техническое задание**

**ХХХ.ХХХХХХХХ.ХХХХХ-01 90 01**

**Листов 25**

2023

СОДЕРЖАНИЕ

[1. Введение 3](#_Toc135043895)

[2. Глоссарий 4](#_Toc135043896)

[3. Основания для разработки 6](#_Toc135043897)

[4. Назначение разработки 7](#_Toc135043898)

[5. Описание системы 8](#_Toc135043899)

[6. Требования к программе или программному изделию 9](#_Toc135043900)

[6.1. Требования к функциональным характеристикам 9](#_Toc135043901)

[6.1.1. Функциональные возможности 9](#_Toc135043902)

[6.1.2. Входные данные системы 13](#_Toc135043903)

[6.1.3. Выходные параметры системы 14](#_Toc135043904)

[6.1.4. Ролевая система 14](#_Toc135043905)

[6.2. Требования к надёжности 14](#_Toc135043906)

[6.3. Условия эксплуатации 15](#_Toc135043907)

[6.4. Требования к составу и параметрам технических средств 15](#_Toc135043908)

[6.5. Требования к информационной и программной совместимости 15](#_Toc135043909)

[6.6. Требования к маркировке и упаковке 16](#_Toc135043910)

[6.7. Требования к транспортированию и хранению 17](#_Toc135043911)

[6.8. Специальные требования 17](#_Toc135043912)

[7. Создание общей картины приложения 18](#_Toc135043913)

[7.1. Структура 18](#_Toc135043914)

[7.2. Сценарии функционирования системы 19](#_Toc135043915)

[7.3. Диаграмма прецендентов 22](#_Toc135043916)

[7.4. Спецификация сценариев 22](#_Toc135043917)

[7.5. Логический дизайн 22](#_Toc135043918)

[7.6. Диаграммы конечных автоматов саг 26](#_Toc135043919)

[8. Требования к программной документации 27](#_Toc135043920)

[9. Технико-экономические показатели 28](#_Toc135043921)

[10. Стадии и этапы разработки 29](#_Toc135043922)

[11. Порядок контроля и приЁмки 30](#_Toc135043923)

[12. Профиль стандартов 31](#_Toc135043924)

[13. Приложения 32](#_Toc135043925)

1. Введение

Техническое задание выполнено на основе ГОСТ 19.201—78 «ЕСПД. Техническое задание. Требования к содержанию и оформлению» [1].

Распределённая учётная система «Бюджет и задачи» (далее просто Система) предназначена для автоматизации таких действий, как ведение заметок, бюджета (журналирование доходов и расходов) и текущих задач, а также обеспечение многопользовательского доступа к этим возможностям. Это позволяет применять Систему как для личного или семейного пользования, так и для небольших организаций, для которых программные системы, обладающие расширенным функционалом, избыточны или слишком дороги (не обеспечивают заметной выгоды по сравнению со стоимостью приобретения или подписки).

1. Глоссарий

|  |  |
| --- | --- |
| **Термин** | **Определение** |
| Ведение бюджета | Систематическое ведение записей доходов и расходов с последующим автоматизированным или ручным подсчётом сумм доходов и расходов, баланса, трат по категориям и других метрик, в зависимости от требуемой информации. |
| Валидация данных | Проверка на корректность, полноту и непротиворечивость входных, выходных и обрабатываемых данных |
| WEB-интерфейс | Интерфейс пользователя, предоставляемой системой через Web-браузер. В разрабатываемой системе только один веб-интерфейс. |
| Область видимости | В рамках Системы может быть личной или групповой. Используется как способ организации доступа к данным и его разграничения. Этот термин описывает только логику Системы и отличается от определения «области видимости» в программировании |
| СОА (SOA) | Сервис-ориентированная архитектура (Service Oriented Architecture), [модульный](https://ru.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D1%83%D0%BB%D1%8C%D0%BD%D0%BE%D1%81%D1%82%D1%8C_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)) подход к разработке [программного обеспечения](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5), основанный на использовании [распределённых](https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D1%81%D0%BF%D1%80%D0%B5%D0%B4%D0%B5%D0%BB%D1%91%D0%BD%D0%BD%D1%8B%D0%B5_%D0%B2%D1%8B%D1%87%D0%B8%D1%81%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F), [слабо связанных](https://ru.wikipedia.org/w/index.php?title=%D0%A1%D0%BB%D0%B0%D0%B1%D0%BE%D0%B5_%D1%81%D0%B2%D1%8F%D0%B7%D1%8B%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5&action=edit&redlink=1) заменяемых компонентов, оснащённых стандартизированными [интерфейсами](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F_%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B9) для взаимодействия по стандартизированным [протоколам](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%BE%D0%B9_%D0%BF%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB). |
| Профиль пользователя | Информация о пользователе портала, хранящаяся в портале |
| Сессия | Сессия на сайте – серия запросов к порталу, сделанных одним пользователем в заданный промежуток времени |
| Фронтенд | Серверное приложение, принимающее запросы от пользователя портала. На каждый из типов запросов от пользователя фронтенд определяет, как организовать выполнение запроса. Фронтенд принимает запросы от пользователя, анализирует их и в соответствии с заложенным алгоритмом выполняет запросы к бекендам |
| Бекенд | Серверное приложение, выполняющее определенную задачу, например, взаимодействие с СУБД. Бекенды принимают запросы от фронтенда |
| Сервер | Компьютер, выполняющий функции обслуживания пользователей при доступе к информационным ресурсам в вычислительных системах |
| Сборка | Процесс формирования из исходных программных кодов и скриптов, в соответствии с заданными настройками, готового к запуску программного продукта |
| Разворачивание | Развёртывание и запуск веб-приложения или сайта в его рабочей среде, то есть на сервере или хостинге, первичная инициализация веб-приложения (если требуется) |
| 3rd party application |  |
| API | Описание способов взаимодействия одной компьютерой программы с другими |
| Identity Provider |  |
| OpenID Connect | Открытый стандарт децентрализованной системы аутентификации, предоставляющей пользователю возможность создать единую учётную запись для аутентификации на множестве не связанных друг с другом интернет-ресурсов, используя услуги третьих лиц (приложение 4) |
| Authorization Flow |  |

1. Основания для разработки

Настоящее техническое задание на создание “Распределённой учётной системы «Бюджет и задачи»” утверждается МГТУ им. Н. Э. Баумана. Разработка ведётся в рамках обучения на курсе «Методология программной инженерии» и выполнения курсового проекта по предмету «Распределённые системы обработки информации» на кафедре «Программное обеспечение ЭВМ и информационные технологии» факультета «Информатика и системы управления».

1. Назначение разработки

Система должна сокращать затраты времени на создание заметок, задач и финансовых записей, а также хранить их централизованно в рамках одной системы, экономя, таким образом, время на переключение между различными программными продуктами, обладающими теми же функциями. Распределённый характер Системы призван решать задачи синхронизации данных между пользователями и не привязывать работу с Системой к конкретному устройству.

Предполагаемая целевая аудитория Системы – как отдельные пользователи, желающие пользоваться предоставляемым ею возможностями с различных устройств, так и небольшие команды (группы) пользователей, которых может привлечь возможность создания общего рабочего пространства, например, семьи или малый бизнес. Общими чертами пользователей являются потребность в интуитивно простом приложении, лёгком в освоении и использовании, поскольку, в противном случае, когнитивные затраты на работу с системой будут выше, чем ведение записей альтернативным способом (на бумаге, в отдельных приложениях).

1. Описание системы

Предлагаемый формат системы – веб-приложение, доступ к которому с помощью браузера может осуществляться как со стационарных, так и с мобильных устройств.

В рассматриваемой предметной области требуется, в первую очередь, возможность быстро и централизованно создавать записи о расходах или доходах, задачи и заметки, при этом заметка может являться прототипом задачи, а задача может сопровождаться определёнными изменениями бюджета. Добавление возможностей по анализу бюджета или истории задач может стать естественным путём дальнейшего совершенствования Системы.

В дальнейшем разрабатываемая Система может предоставляться пользователю как услуга (хранение данных и работа на облачных серверах, которыми пользователь не распоряжается), так и коробочное решение.

Аналогами Системы в области отдельных задач, выполняемых ею, являются все сервисы, позволяющие:

* делать заметки – бумажные блокноты, Google Keeps и аналоги, личный диалог (“Избранное”, где оставлять и читать какие-то сообщения может только пользователь) и каналы в Telegram и других мессенджерах;
* вести бюджет – таблицы Excel, приложение “Журнал расходов” на Android и аналоги;
* записывать задачи/планы – то же, что и для ведения заметок, канбан-доски в Github и Gitlab, отдельные реализации канбан-досок, календари и др.

Таким образом, у Системы достаточно много аналогов на разных платформах и в разных областях её функционала, однако её преимуществом станет объединение возможностей в рамках одного приложения.

1. Требования к программе или программному изделию
   1. Требования к функциональным характеристикам

В подразделе «Требования к функциональным характеристикам» должны быть указаны требования к составу выполняемых функций, организации входных и выходных данных, временным характеристикам и т. п.

* + 1. Функциональные возможности

Система должна предоставлять следующие возможности, представленные в виде таблиц 1-4. Содержащиеся в них функции приводятся с точки зрения Пользователя. Отличия версии для Администратора приводятся в отдельной строке внизу таблицы.

Одним из важных понятий является область видимости (далее ОВ), определение которой приводится в глоссарии. В Системе ОВ может быть личной (все создаваемые в этой ОВ записи видны только самому пользователю, который их создал) и групповой (записи в этой ОВ доступны только участникам выбранной группы).

Работа с пользовательскими учётными записями и группами

|  |  |
| --- | --- |
| Функции | 1) регистрация нового пользователя с валидацией вводимых данных;  2) аутентификация пользователей;  3) ролевое разграничение на пользователей и администраторов (требования к ролевой системе изложены в отдельном пункте);  4) работа с группами:  а) создание группы;  б) просмотр состава в виде списка логинов и имён участников группы;  в) просмотр собственного списка групп, в которых пользователь состоит;  г) приглашать другого пользователя по логину в группу;  д) выход из группы;  е) автоматическое удаление группы при выходе всех участников;  5) справка по предоставляемым функциям. |
| Основные объекты, их характеристика | **Профиль пользователя**: логин, пароль, имя Пользователя, электронная почта.  **Группа:** название, один или несколько Пользователей. |
| Связи объектов | *Один Пользователь может состоять во множестве Групп.*  *Одна Группа может включать множество Пользователей.* |
| Отличия у администратора | 1. Администратор не может зарегистрироваться, можно только создать нового или назначить существующего пользователя через базу данных; 2. Администратор может просматривать список и состав всех созданных пользователями групп. |

Работа с ведением бюджета

|  |  |
| --- | --- |
| Функции | 1. работа с категориями расходов и доходов:    1. создание категории с областью видимости:       * личная;       * для выбранной группы;    2. переименовать категорию;    3. удалить категорию;    4. при удалении категории все записи… (ещё не решил, что должно происходить); 2. работа с записями о доходах и расходах:    1. просмотр записи;    2. создание записи с выбором группы и категории;    3. изменение записи;    4. просмотр списка всех записей в сортировке по убыванию даты:       * по категориям;       * по группам;       * за всё время или за период: день, неделя, текущий месяц, год;    5. удаление записи; 3. вычисление баланса (по личным записям пользователя или по выбранной группе); 4. справка по предоставляемым функциям. |
| Основные объекты, их характеристика | **Категория:** название, принадлежность к ОВ (к личной ОВ текущего пользователя или к одной из его групп). Категории не должны повторяться (  **Запись о расходе или доходе:** комментарий произвольной длины, сумма в рублях, выбранная ОВ, категория в выбранной ОВ, автор, дата и время. |
| Связи объектов | *Одна Группа имеет одну ОВ.*  *Один Пользователь имеет одну ОВ.*  *Одна ОВ имеет множество Категорий.*  *Один Пользователь является автором множества Записей.*  *Одна Запись имеет одну ОВ, Категорию и одного автора.* |
| Отличия у администратора | 1. Администратор имеет возможность просматривать ОВ всех групп. |

Работа с заметками

|  |  |
| --- | --- |
| Функции | 1. просмотр списка всех заметок в сортировке по убыванию даты:    1. по выбранной группе;    2. личные; 2. просмотр заметки 3. создать заметку с выбором группы:    1. при создании заметки с меткой «Задача» должна автоматически создаваться задача, содержащая данную заметку в качестве вложения; 4. изменить заметку; 5. удалить заметку. |
| Основные объекты, их характеристика | **Заметка**: автор, дата и время, выбранная ОВ, заголовок, текст-содержание, опциональные теги (метки). |
| Связи объектов | *Одна Группа имеет одну ОВ.*  *Один Пользователь имеет одну ОВ.*  *Одна ОВ имеет множество Заметок.*  *Один Пользователь является автором множества Заметок.*  *Одна Заметок может иметь несколько тегов (меток).* |
| Отличия у администратора | 1. Администратор имеет возможность просматривать ОВ всех групп. |

Работа с задачами

|  |  |
| --- | --- |
| Функции | 1. просмотр списка всех задач в сортировке по убыванию даты:    1. по группе;    2. личные задачи; 2. просмотр задачи; 3. создание задачи (по умолчанию всем новым задачам задаётся статус «Создана» с выбором группы; 4. изменение задачи; 5. изменение статуса задачи; 6. просмотр комментариев к задаче в сортировке по убыванию даты; 7. добавление комментария к задаче; 8. удаление комментария к задаче; 9. удаление задачи. |
| Основные объекты, их характеристика | **Задача**: автор, дата и время, название, краткое описание, ссылка на Заметку (может быть пустой), статус задачи, который может принимать одно из заданных состояний (приложение 1).  **Комментарии**: имя автора, текст, время публикации. Комментарии видны только вместе с задачей.  **Статус:** строка из заданного разработчиками набора (приложение 1). |
| Связи объектов | *Одна Группа имеет одну ОВ.*  *Один Пользователь имеет одну ОВ.*  *Одна ОВ имеет множество Заметок.*  *Одна Задача может иметь один статус, множество Задач могут принимать один Статус.*  *Один Пользователь является автором множества Задач.*  *Одна Задача может ссылаться на одну Заметку.*  *Одна Задача может иметь множество Комментариев.* |
| Отличия у администратора | 1. Администратор имеет возможность просматривать ОВ всех групп. |

Также Система должна собирать статистику работы, а входящие в её состав сервисы – иметь служебные методы для проверки работоспособности. Эти возможности доступны только администраторам, поэтому в таблице 5 представлены функции полностью с точки зрения администратора.

Статистика

|  |  |
| --- | --- |
| Функции | 1. В разработке |
| Основные объекты, их характеристика |  |
| Связи объектов |  |

* + 1. Входные данные системы

При регистрации, работе со своей учётной записью и группами пользователь должен будет передавать следующую информацию:

* логин (строка длиной до 20 символов, без пробелов и специальных символов, кроме «\_» и «-», латиница, не может быть пустой);
* пароль (строка длиной до 20 символов, без пробелов, латиница и спец. символы, не может быть пустой);
* имя (строка длиной до 80 символов, латиница, кириллица, пробелы, апострофы, дефисы, без остальных спец. символов, не может быть пустым);
* электронная почта (строка длиной до 40 символов, должна соответствовать формату электронной почты: \*@\*.\*);
* название группы (строка длиной до 80 символов, не может быть пустой);
* логин пользователя, которого необходимо добавить в группу.

При работе со списками расходов и доходов:

* для категорий:
  + название (строка длиной до 30 символов, не может быть пустой);
  + выбранная принадлежность (пользователю должен быть предоставлен список, включающий доступные группы и принадлежность «Личная»);
* для записей о расходах или доходах:
  + выбранная ОВ;
  + категория, выбранная из списка доступных категорий;
  + комментарий (до 400 символов);
  + сумма в рублях (вещественное число, не может быть отрицательным).

При работе с заметками:

* заголовок (строка длиной до 50 символов);
* содержание (текст длиной до 2000 символов);
* метка (строка длиной до 20 символов).

При работе с задачами:

* «встроенная» заметка;
* выбранный статус задачи (строка из заданного перечня в приложении 1);
* краткое описание (до 150 символов);
* текст комментария (до 300 символов);
  + 1. Выходные параметры системы

Выходными параметрами системы являются веб-страницы. Они должны содержать следующую информацию:

* введённые ранее пользователем данные (задачи, заметки, расходы и т.д.) в структурированном виде;
* баланс (вычисленный по расходам и доходам).
  + 1. Ролевая система

Ролевая система должна включать в себя две роли, обладающие следующими правами и возможностями:

1. Пользователь:
   1. имеет доступ к сервисам Системы в рамках зоны его личной и групповых зон видимости (видит и может изменять только то, к чему относится сам);
   2. может получать доступ к открытой информации всех других пользователей (в рамках разрабатываемой системы это только имя и логин);
2. Администратор:
   1. как и пользователь, имеет доступ к сервисам Системы в рамках зоны его личной и групповых зон видимости;
   2. может получать доступ к открытой информации всех других пользователей;
   3. имеет доступ ко всем групповым зонам видимости, то есть способен просматривать записи всех групп, созданных другими пользователями;
   4. не имеет доступа к личным зонам видимости;
   5. имеет доступ к странице администратора.
   6. Требования к надёжности

*В подразделе «Требования к надежности» должны быть указаны требования к обеспечению надёжного функционирования (обеспечения устойчивого функционирования, контроль входной и выходной информации, время восстановления после отказа и т.п.).*

1. Система должна обеспечивать сохранение успешно введённой в неё информации в случае сбоя, если сбой затрагивал только программную часть и не затронул сервер (сервера) с базами данных.
   1. Условия эксплуатации
2. К условиям эксплуатации Системы не предъявляется специальных требований сверх тех, которые обусловлены требованиями оборудования и вспомогательных программных систем (операционной системы и др.).
3. Для эксплуатации Системы необходимо наличие администратора для контроля работы программной части и системного администратора для обслуживания Системы на сервере (допускается совмещение обеих работ одним работником).
   1. Требования к составу и параметрам технических средств

*В подразделе «Требования к составу и параметрам технических средств» указывают необходимый состав технических средств с указанием их основных технических характеристик.*

Технические средства должны обеспечивать ряд условий для функционирования Системы.

1. Публичный IP-адрес для доступа к Системе.
2. Свободный доступ к Интернету для получения обновлений проекта.
3. Поддерживать Docker.
   1. Требования к информационной и программной совместимости

*В подразделе «Требования к информационной и программной совместимости» должны быть указаны требования к информационным структурам на входе и выходе и методам решения, исходным кодам, языкам программирования и программным средствам, используемым программой. При необходимости должна обеспечиваться защита информации и программ.*

1. Система должна использовать СУБД PostgreSQL.
2. API приложения должно быть построено в стиле REST API.
3. Отдельный сервис должен выполнять функцию Identity Provider, реализовать протокол OpenID Connect (приложение 4). Реализовать scope:
   1. openid – всегда передается, в ответ возвращается JWT;
   2. profile – базовая информация о пользователе;
   3. email – email пользователя.
4. Получение токена доступа должно быть реализовано по Authorization Flow.
5. Пользовательский интерфейс рассматривается как 3rd party application и авторизация пользователя должна выполняться через Authorization Flow.
6. Все методы /api/\*\* (кроме /api/v1/authorize и /api/v1/callback) должны быть защищены token-based авторизацией.
7. Реализовать ролевую модель:
   1. вручную создать пользователя с ролью Admin;
   2. для зарегистрированных пользователей по умолчанию устанавливать роль User.
8. Identity Provider должен обеспечивать возможность создания новых пользователей (запрос от пользователя с ролью Admin).
9. Каждый сервис при получении запроса выполняет валидацию JWT-токена с помощью JWKs, которые он получает из Identity Provider (запрос к Identity Provider делать не нужно).
10. JWT-токен пробрасывать между сервисами, при получении запроса валидацию токена так же реализовать через JWKs.
11. Информацию о пользователе получать из JWT-токена.
12. Если авторизация некорректная (отсутствие токена, ошибка валидации JWT токена, закончилось время жизни токена (поле exp в payload)), то отдавать ошибку 401.
13. Пользовательский интерфейс должен быть реализован как Одностраничное Приложение (Single Page Application) ([React](https://reactjs.org/), [Angular](https://angular.io/), [Vue](https://vuejs.org/)) или мобильный клиент.
    1. Использование CSS обязательно.
14. Серверная часть Системы должна быть реализована на языке Go без ограничений на применяемые фреймворки и библиотеки.
15. Выделить сервис статистики: в него отправлять через Kafka информацию о выполненных действиях. В зависимости от задания по пришедшим данным строить отчёт, доступ к которому должен быть только у пользователя с ролью Admin.
16. Необходимо развернуть Managed Kubernetes Cluster и настроить Ingress Controller (для публикации сервисов наружу можно использовать только Ingress).
17. Для разворачивания использовать [helm charts](https://helm.sh/docs/topics/charts/), они должны быть универсальным для всех сервисов и отличаться лишь набором параметров запуска.
    1. Требования к маркировке и упаковке

По соглашению с заказчиком данный пункт в настоящем ТЗ опускается.

* 1. Требования к транспортированию и хранению

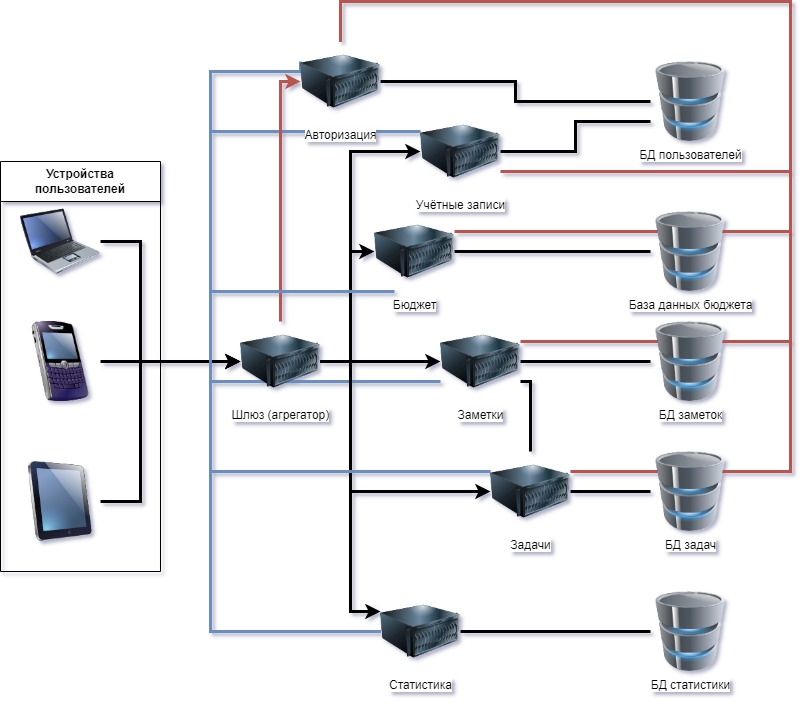
По соглашению с заказчиком данный пункт в настоящем ТЗ опускается.

* 1. Специальные требования

Специальные требования к Системе не предъявляются.

1. Создание общей картины приложения
   1. Структура

Общий вид, структура Системы представлены на рис. 5 в соответствии с СОА. Географически данная структура может быть разнесена по разным серверам (каждый отмеченный на схеме сервис, каждая БД), но может функционировать и на одном сервере. В таком случае допустимо все *логические* БД сервисов хранить под управлением одной СУБД, в одном сервисе.



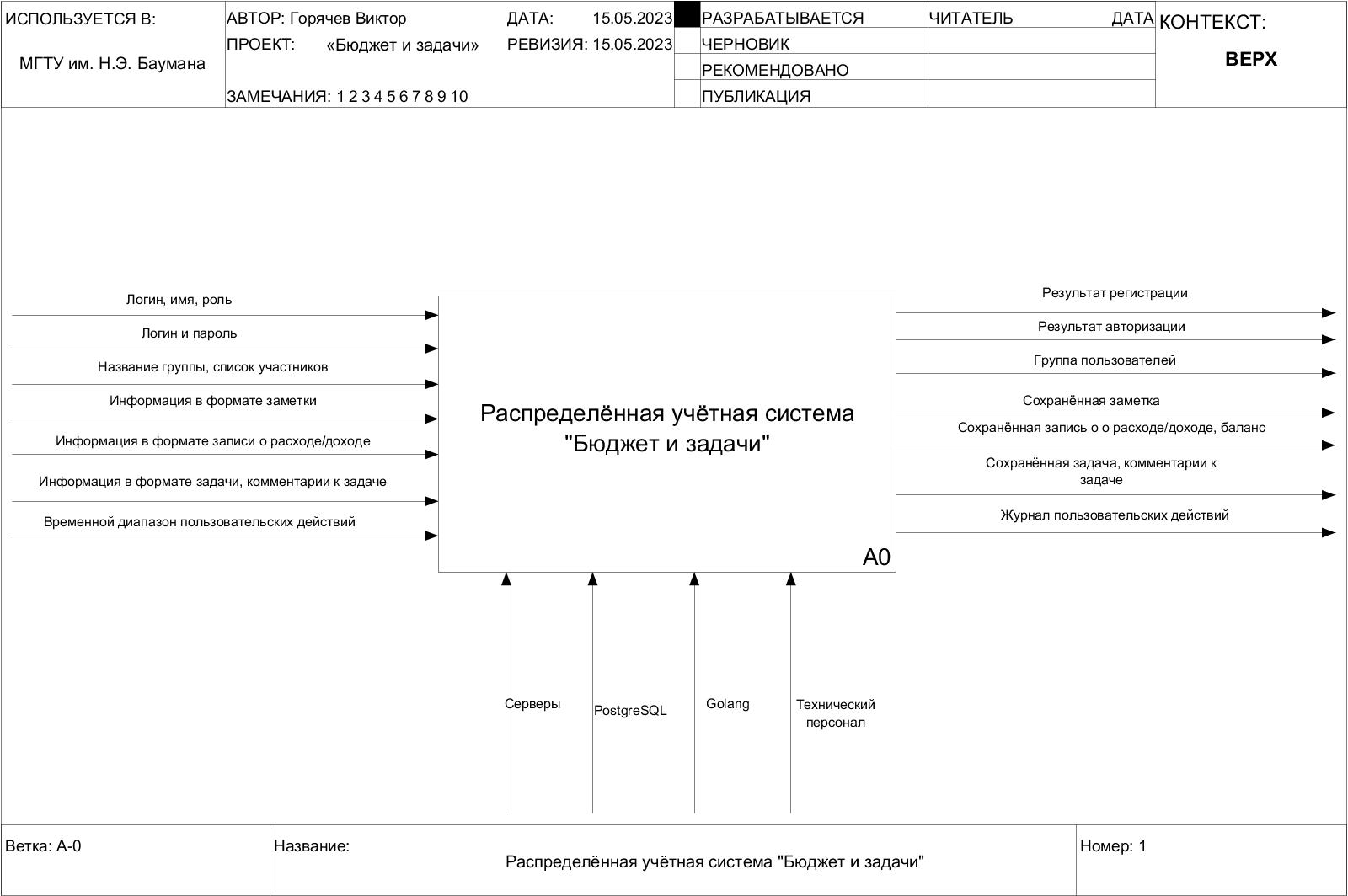
Структура системы. Красными линиями обозначены связи, используемые для проверки прав доступа, синими – взаимодействие в рамках сбора статистики, чёрными – выполнение сервисами основных функций. Стрелки обозначают только «заимствование» функционала сервисом-агрегатором.

Каждый сервис сформирован в соответствии с логическим делением функциональности, заданной требованиями. Исключение составляют сервисы авторизации и учётных записей, которые разделены с целью разделения логической части Системы (работа с учётными записями и группами) и инфраструктурной (система авторизации).

Пользователи взаимодействуют, используя браузер своего устройства, с сервисом-агрегатором, который позволяет им авторизоваться и использовать функционал Системы. В соответствии с требованиями, Шлюз отправляет пользователю при первом запросе фронтенд часть, после чего играет роль обработчика запросов от фронтенда, который пересылает запросы на другие сервера.

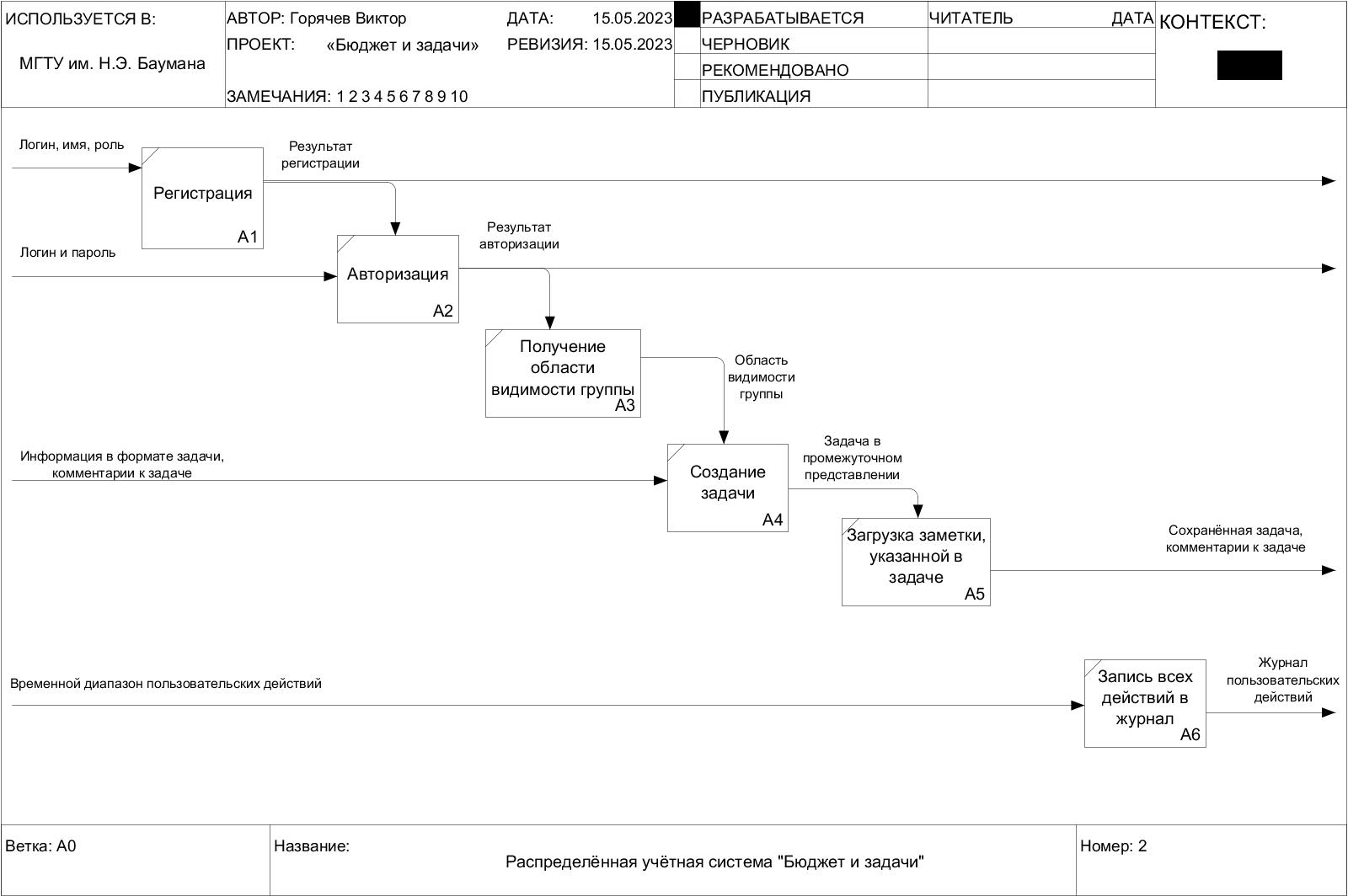
* 1. Концептуальный дизайн системы

Концептуальный дизайн системы содержит наиболее общие схемы описания функционала приложения с точки зрения пользователей. На рисунке 2 представлена диаграмма IDEF0 первого уровня. Данный вид диаграммы позволяет формализовать описание запросов пользователя и ответов системы на них, отобразив систему в виде «черного» ящика.



Концептуальная модель системы в нотации IDEF0

На рисунке 3 представлена детализированная диаграмма, которая определяет последовательность выполнения операций в системе при…



Детализированная концептуальная модель системы в нотации IDEF0 при создании задачи

* 1. Сценарии функционирования системы

Сценарии функционирования описывают конкретную последовательность действий, иллюстрирующую поведение пользователя при работе с приложением. Далее приведены подробные сценарии основных возможных действий пользователей.

**Авторизация пользователя**

1. Пользователь открывает сайт с экраном авторизации, который содержит поля для заполнения логина и пароля.

2. Пользователь вводит логин и пароль в форму и нажимает кнопку «Войти».

3. При обнаружении ошибок в введённых данных приложение уведомляет пользователя об этом с кратким описанием ошибки. При совпадении данных с записью в базе данных сервиса авторизации пользователь получает доступ к системе. При первичной авторизации пользователь перенаправляется на экран смены пароля. Иначе пользователь перенаправляется на основной экран.

**Регистрация пользователя**

1. Пользователь с ролью администратор нажимает на кнопку «Зарегистрировать нового пользователя».

2. Пользователь перенаправляется на экран регистрации нового пользователя, который содержит поля для заполнения имени, логина и роли.

3. Пользователь вводит данные в форму и для завершения регистрации нажимает на кнопку «Регистрация», тем самым подтверждая верность данных, а также согласие на их обработку и хранение.

4. Если пользователь с введенным для регистрации логином уже существует, то клиент перенаправляется на страницу ошибки. При успешной регистрации клиент перенаправляется на экран, содержащий профиль созданного пользователя в системе, а также одноразовый пароль для его авторизации.

**Создание группы и добавление новых участников**

1. Пользователь на главном экране нажимает кнопку «Группы».
2. Пользователь перенаправляется на экран, содержащий список групп, в которых данный пользователь состоит, а также кнопку «Создать».
3. Пользователь нажимает кнопку «Создать».
4. Пользователь перенаправляется на страницу, содержащую форму создания группы.
5. Пользователь заполняет поля и нажимает кнопку «Создать».
6. Пользователь перенаправляется на страницу, содержащую информацию о созданной группе – список участников, название и идентификатор – и кнопки «Пригласить» и «Покинуть группу».
7. Пользователь нажимает кнопку «Пригласить».
8. Пользователь предлагается форма для ввода имени (логина) другого пользователя, которого нужно добавить в группу.
9. Пользователь вводит логин и нажимает на кнопку «Пригласить».
10. Пользователь перенаправляется на страницу с информацией о группе (шаг 6).

**Покидание/удаление группы**

1. Пользователь на главном экране нажимает кнопку «Группы».
2. Пользователь перенаправляется на экран, содержащий список групп, в которых данный пользователь состоит.
3. Пользователь нажимает на группу в списке, которую хочет покинуть.
4. Пользователь перенаправляется на страницу, содержащую информацию о созданной группе – список участников, название и идентификатор – и кнопки «Пригласить» и «Покинуть группу».
5. Пользователь нажимает кнопку «Покинуть группу».
6. Пользователь перенаправляется на страницу из шага 2.
7. В случае, если пользователь являлся последним участником группы, она будет удалена.

**Получение списка заметок**

1. Авторизированный пользователь выбирает экран работы с заметками.
2. Пользователь перенаправляется на экран, содержащий список заметок в его личной области видимости, отсортированных по времени создания, и возможность (выпадающий список) для выбора области видимости одной из групп, в которой он состоит.
3. Пользователь выбирает одну из групп из списка групп.
4. Список на странице обновляется в соответствии с выбором.

**Создание заметки**

1. Авторизированный пользователь выбирает необходимую область видимости и нажимает кнопку создания заметки.
2. Пользователь перенаправляется на экран, содержащий форму создания заметки.
3. Пользователь заполняет поля новой заметки и нажимает на кнопку «Создать».

**Просмотр заметки**

1. Авторизированный пользователь выбирает необходимую область видимости.
2. Страница обновляется с учётом выбранной области видимости.
3. Пользователь нажимает на интересующую его заметку в списке.
4. Перенаправляется на экран, содержащий выбранную заметку.

**Редактирование заметки**

1. Авторизированный пользователь выполняет действия для **Просмотра заметки**.
2. Пользователь нажимает кнопку «Редактировать».
3. Пользователь перенаправляется на экран, содержащий форму редактирования заметки, аналогичную по содержанию форме создания, в которой поля уже предзаполнены существующими значениями.
4. Пользователь изменяет нужные поля и нажимает кнопку «Сохранить».

**Удаление заметки**

1. Авторизированный пользователь выполняет действия для **Просмотра заметки**.
2. Пользователь нажимает кнопку «Удалить».
3. Система запрашивает у пользователя подтверждение удаления.
4. Пользователь либо отказывается от удаления, либо подтверждает. В случае подтверждения он перенаправляется на экран со списком всех заметок и получает уведомление об удалении заметки.

**Просмотр списка задач**

1. Авторизированный пользователь выбирает страницу работы с задачами.
2. Пользователь перенаправляется на экран, содержащий список задач в его личной области видимости, отсортированных по времени создания, и возможность (выпадающий список) для выбора области видимости одной из групп, в которой он состоит.
3. Пользователь выбирает одну из групп из списка групп, в которых он состоит.
4. Список на странице обновляется в соответствии с выбором.

**Создание задачи**

1. Авторизированный пользователь выполняет действия для **Просмотра списка задач**.
2. Авторизированный пользователь выбирает необходимую область видимости и нажимает кнопку создания задачи.
3. Пользователь перенаправляется на экран, содержащий форму создания задачи.
4. Пользователь заполняет поля новой задачи и нажимает на кнопку «Создать».
5. Пользователь перенаправляется на страницу просмотра созданной задачи.

**Получение журнала пользовательский действий**

1. Пользователь с ролью администратор нажимает на кнопку «Журнал пользовательских действий».

2. Пользователь переход на экран журнала пользовательский действий, отображающий действия пользователей в виде списка записей вида:

"{Время}: {Логин} {Действие} {Дополнительные параметры}".

3. Пользователь задаёт фильтры поиска записей. Экран журнала пользовательский действий обновляет список действий.

* 1. Диаграмма прецендентов

В системе выделены 2 основных роли: пользователь и администратор,

диаграммы прецедентов для этих ролей изображены на рисунках 5 и 6. На рисунке 4 представлена диаграмма прецедентов с точки зрения неавторизированного пользователя.

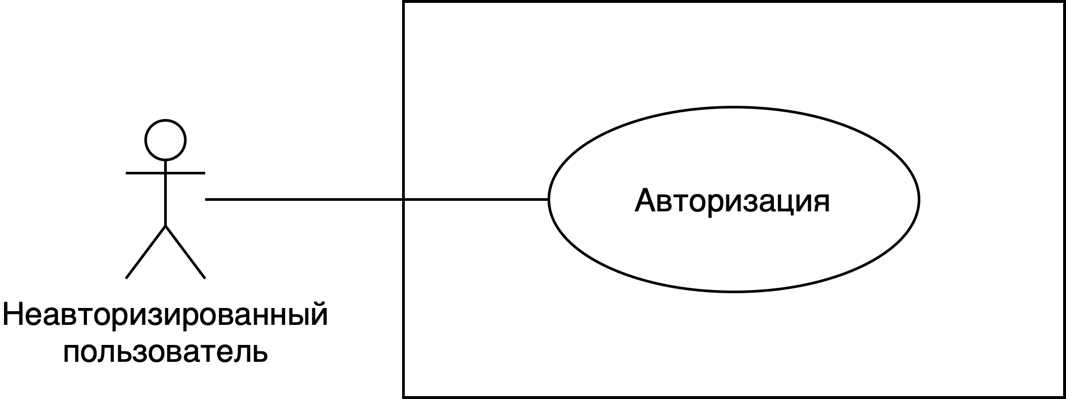


Диаграмма прецедентов с точки зрения неавторизованного пользователя

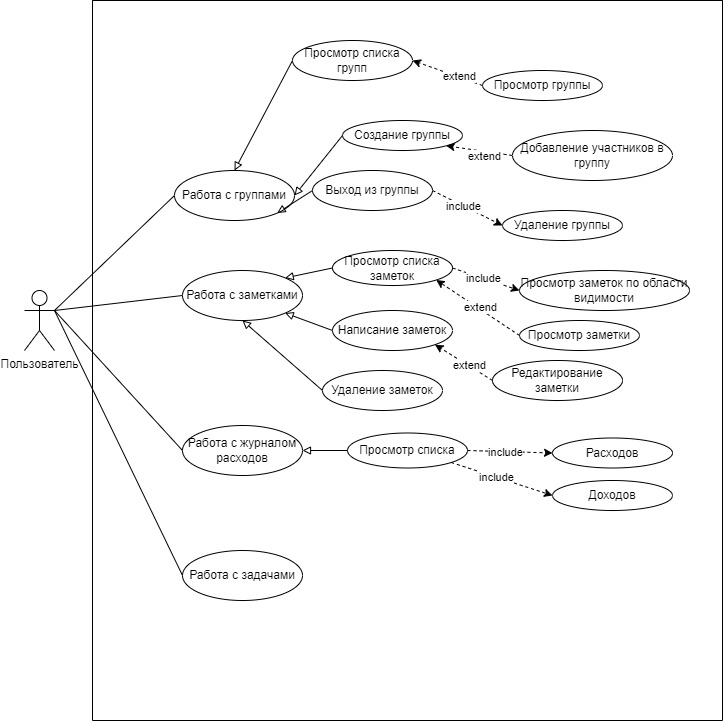


Диаграмма прецедентов с точки зрения пользователя.



Диаграмма прецедентов с точки зрения администратора

* 1. Спецификация сценариев

Приведенные сценарии могут иметь как основной поток выполнения, который выполняется чаще всего, так и альтернативные потоки, описывающие выполнение запроса при отклонении от основного хода сценария. В данном разделе приведены спецификации для описанных выше сценариев.

**Спецификация сценария «Авторизация»**

|  |  |
| --- | --- |
| **Нормальный ход сценария** | |
| **Действия пользователя** | **Отклик системы** |
| Пользователь заходит на сайт | Открывается страница авторизации для ввода данных |
| Пользователь вводит данные в поля и нажимает кнопку «Войти» | Открывается главная страница |

|  |  |
| --- | --- |
| **Альтернативный ход сценария** | |
| **Действия пользователя** | **Отклик системы** |
| Пользователь впервые заходит на сайт | Открывается страница авторизации для ввода данных |
| Пользователь вводит данные в поля и нажимает кнопку «Войти» | Открывается страница смены пароля |
| Пользователь вводит новый пароль в поля и нажимает кнопку «Сменить пароль» | Открывается главная страница |

|  |  |
| --- | --- |
| **Альтернативный ход сценария** | |
| **Действия пользователя** | **Отклик системы** |
| Пользователь заходит на сайт | Открывается страница авторизации для ввода данных |
| Пользователь вводит данные в поля и нажимает кнопку «Войти» | Выводится ошибка с сообщением о неверном логине или пароле |

**Спецификация сценария «Регистрация»**

|  |  |
| --- | --- |
| **Нормальный ход сценария** | |
| **Действия пользователя** | **Отклик системы** |
| Пользователь с ролью администратор нажимает на кнопку «Зарегистрировать нового пользователя» | Открывается страница регистрации нового пользователя для ввода данных |
| Пользователь вводит данные в поля и нажимает кнопку «Регистрация» | Открывается страница, содержащий профиль созданного пользователя в системе, а также единоразовый пароль для его авторизации |

|  |  |
| --- | --- |
| **Альтернативный ход сценария** | |
| **Действия пользователя** | **Отклик системы** |
| Пользователь с ролью администратор нажимает на кнопку «Зарегистрировать нового пользователя» | Открывается экран регистрации нового пользователя для ввода данных |
| Пользователь вводит данные в поля и нажимает кнопку «Регистрация» | Открывается экран ошибки с сообщением о существовании пользователя с тем же логином |

* 1. Логический дизайн

На основе функциональных требований к выделенным подсистемам, а также объектов, о которых необходимо хранить данные в системе, была разработана схема данных приложения. Результат ее проектирования отображен на условной схеме базы данных, представленной на рисунке 2.

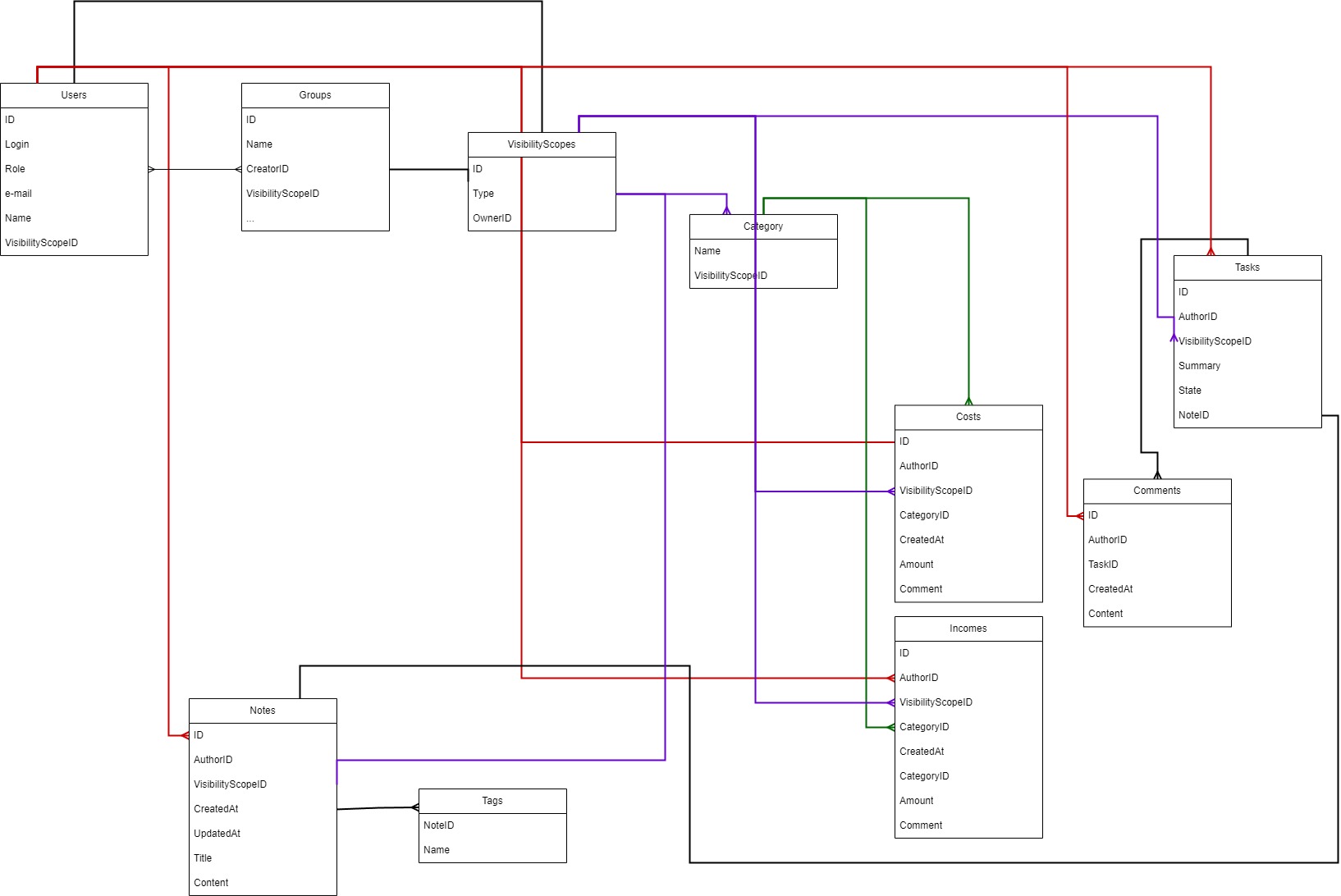


Схема базы данных системы

**Спецификация таблицы Users**

Данные о профиле пользователя

|  |  |  |
| --- | --- | --- |
| **Имя атрибута** | **Тип атрибута** | **Описание атрибута** |
| ID | guid | Идентификатор пользователя |
| Login | string |  |
| Name | string | Имя пользователя |
| Role | int | Права пользователя |
| Email | string | Электронная почта |
| VisibilityScopeID | int | Идентификатор личной ОВ пользователя |

**Спецификация таблицы GroupMembers**

Данные о связи групп и пользователей

|  |  |  |
| --- | --- | --- |
| **Имя атрибута** | **Тип атрибута** | **Описание атрибута** |
| UserID | guid | Идентификатор пользователя |
| GroupID | guid | Идентификатор группы |

**Спецификация таблицы Groups**

Данные о группе

|  |  |  |
| --- | --- | --- |
| **Имя атрибута** | **Тип атрибута** | **Описание атрибута** |
| ID | guid | Идентификатор группы |
| Name | string | Название группы |
| CreatorID | guid | Идентификатор создателя группы |
| VisibilityScopeID | int | Идентификатор ОВ группы |

**Спецификация таблицы VisibilityScopes**

Данные об областях видимости

|  |  |  |
| --- | --- | --- |
| **Имя атрибута** | **Тип атрибута** | **Описание атрибута** |
| ID | int | Идентификатор ОВ |
| Type | int | Тип ОВ, групповая или личная |
| OwnerID | guid | Идентификатор владельца ОВ |

**Спецификация таблицы Category**

Данные о категориях доходов/расходов

|  |  |  |
| --- | --- | --- |
| **Имя атрибута** | **Тип атрибута** | **Описание атрибута** |
| ID | int | Идентификатор |
| Name | string | Название категории |
| VisibilityScopeID | int | Идентификатор ОВ категории |

**Спецификация таблицы Costs**

Данные о расходах

|  |  |  |
| --- | --- | --- |
| **Имя атрибута** | **Тип атрибута** | **Описание атрибута** |
| ID | int | Идентификатор записи |
| AuthorID | guid | Идентификатор автора |
| VisibilityScopeID | int | Идентификатор ОВ |
| CategoryID | int | Идентификатор категории |
| CreatedAt | timestamp | Время создания |
| Amount | int | Сумма расхода |
| Comment | string | Комментарий |

**Спецификация таблицы Incomes**

Данные о доходах

|  |  |  |
| --- | --- | --- |
| **Имя атрибута** | **Тип атрибута** | **Описание атрибута** |
| ID | int | Идентификатор записи |
| AuthorID | guid | Идентификатор автора |
| VisibilityScopeID | int | Идентификатор ОВ |
| CategoryID | int | Идентификатор категории |
| CreatedAt | timestamp | Время создания |
| Amount | int | Сумма дохода |
| Comment | string | Комментарий |

**Спецификация таблицы Notes**

Данные о заметках

|  |  |  |
| --- | --- | --- |
| **Имя атрибута** | **Тип атрибута** | **Описание атрибута** |
| ID | int | Идентификатор записи |
| AuthorID | guid | Идентификатор автора |
| VisibilityScopeID | int | Идентификатор ОВ |
| UpdatedAt | timestamp | Время обновления |
| CreatedAt | timestamp | Время создания |
| Title | string | Заголовок |
| Content | string | Содержание |

**Спецификация таблицы Tags**

Данные о метках (тегах)

|  |  |  |
| --- | --- | --- |
| **Имя атрибута** | **Тип атрибута** | **Описание атрибута** |
| NoteID | int | Идентификатор родительской заметки |
| Name | string | Название-содержание |

**Спецификация таблицы Tasks**

Данные о профиле пользователя

|  |  |  |
| --- | --- | --- |
| **Имя атрибута** | **Тип атрибута** | **Описание атрибута** |
| ID | int | Идентификатор записи |
| AuthorID | guid | Идентификатор автора |
| VisibilityScopeID | int | Идентификатор ОВ |
| UpdatedAt | timestamp | Время обновления |
| CreatedAt | timestamp | Время создания |
| Summary | string | Краткое содержание |
| State | string | Состояние |
| NoteID | int | Идентификатор поясняющей заметки |

**Спецификация таблицы Comments**

Данные о профиле пользователя

|  |  |  |
| --- | --- | --- |
| **Имя атрибута** | **Тип атрибута** | **Описание атрибута** |
| ID | guid | Идентификатор пользователя |
| AuthorID | guid | Идентификатор автора |
| TaskID | int | Идентификатор задачи |
| CreatedAt | timestamp | Время создания |
| Content | string | Текст |

* 1. Диаграммы последовательности действий

Текст

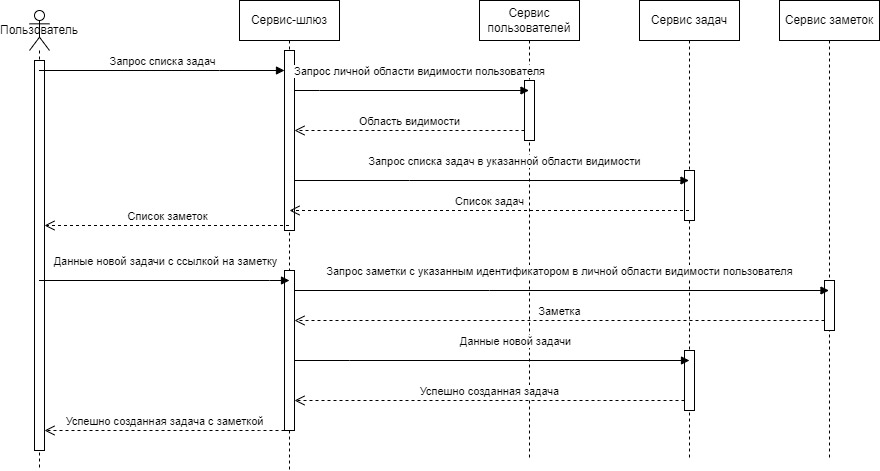


Диаграмма последовательности создания задачи с ссылкой (идентификатором) существующей заметки в личной области видимости пользователя. Стандартный ход выполнения без ошибок

1. Требования к программной документации

В разделе «Требования к программной документации» должен быть указан предварительный состав программной документации и, при необходимости, специальные требования к ней.

Исполнитель должен подготовить и передать Заказчику следующие документы:

* расчётно-пояснительную записку в соответствии с приведёнными ниже требованиями.

Расчётно-пояснительная записка должна состоять из трёх частей:

1. **Аналитический раздел.** В данном разделе приводится обзор существующих систем, описываются основные требования к системе. Здесь же формулируется бизнес-логика будущей системы.
2. **Конструкторский раздел.** Описание архитектуры и алгоритмов, используемых в системе, ссылок на стандартные алгоритмы. В этом разделе приводится общая схема системы и описание каждого компонента в отдельности. Обязательно описать выделенные сущности, чем они характеризуются, дать описание ролевой модели, описать схему взаимодействия систем с помощью Диаграммы последовательности действий.
3. **Технологический раздел.** В данном разделе приводится описание типов и структур данных (структура БД), а также описываются нюансы реализации. Здесь же требуется описать как выполняется сборка и разворачивание системы. Описывается тестирование системы и поведение в случае отказа составляющих её компонентов.

Записка должна быть оформлена по ГОСТ 7.32-2017. В конце обязательно привести список литературы.

1. Технико-экономические показатели

По соглашению с заказчиком данный пункт в настоящем ТЗ опускается.

1. Стадии и этапы разработки

По соглашению с заказчиком данный пункт в настоящем ТЗ опускается.

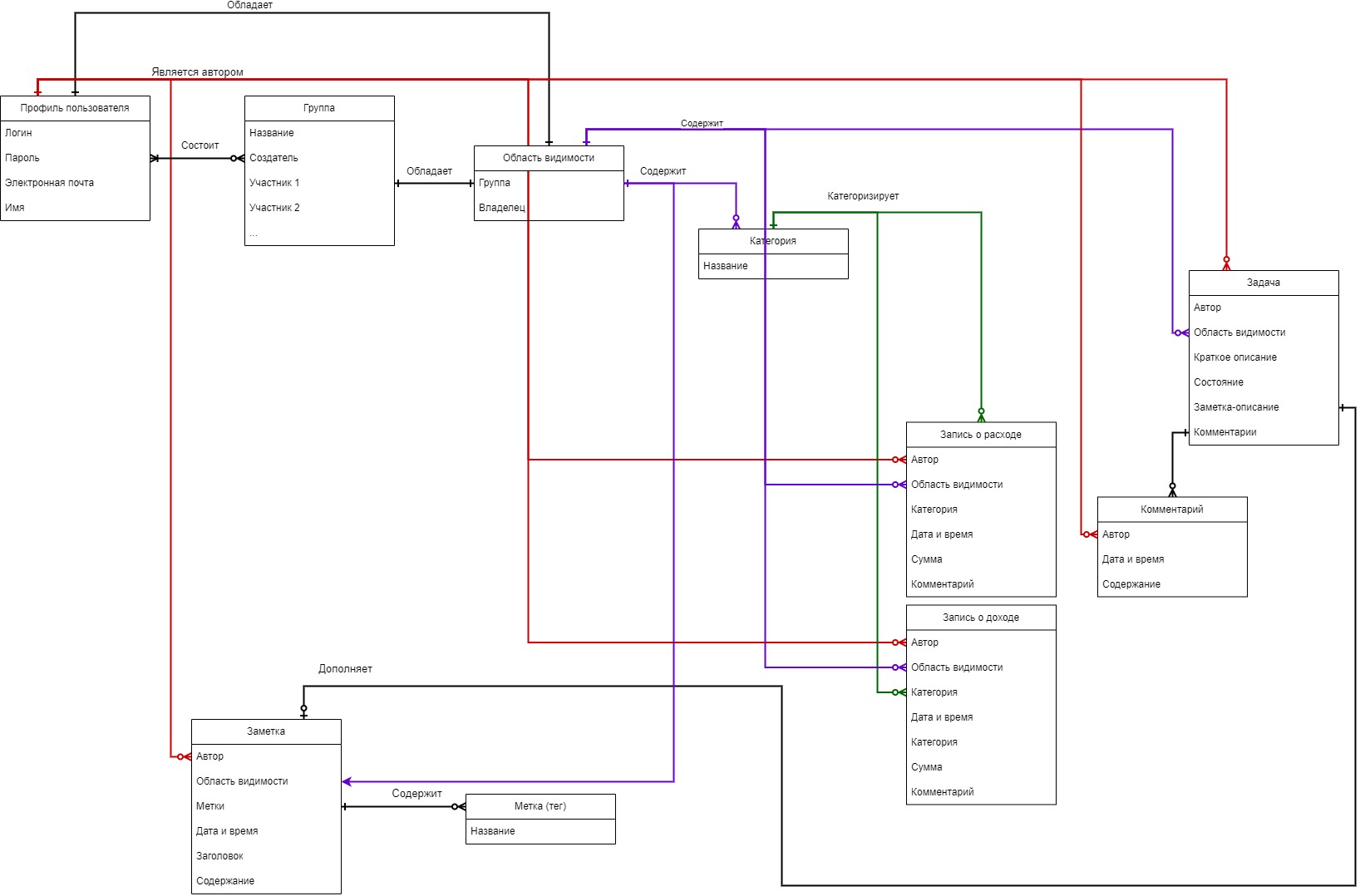
1. Порядок контроля и приЁмки

По соглашению с заказчиком данный пункт в настоящем ТЗ опускается.

1. Профиль стандартов

Профиль стандартов.

1. Техническое задание. Требования к содержанию и оформлению (ГОСТ 19.201-78). [Электронный ресурс] Режим доступа: URL: <http://fmi.asf.ru/library/book/Gost/19201-78.html>.
2. Приложения
3. Состояния, которые может принимать задача: «В очереди», «В работе» (допустим вариант «Выполняется»), «Выполнена», «Отменена», «Ожидание» (выполнение приостановлено по независящим от исполнителя причин, например, в ожидании проверки).
4. Пособие по использованию jwt-токенов: <https://jwt.io/introduction>.
5. Пособие по безопасности при использовании jwt-токенов: <https://auth0.com/docs/secure/tokens/json-web-tokens/json-web-key-sets>.
6. Протокол OpenID Connect. Режим доступа: URL: <https://auth0.com/docs/authenticate/protocols/openid-connect-protocol>.
7. ER-диаграмма предметной области в нотации Crow foot.



1. Текст