Министерство образования и науки Пермского края

краевое государственное автономное профессиональное образовательное учреждение «Пермский авиационный техникум им. А.Д. Швецова»

**Дипломный Проект**

Пояснительная записка

Процесс стандартизации при дедуктивном выводе

АТДП.090207.19.217.ПЗ

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.О. Ощепков

(подпись, дата)

Рецензент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ И.О. Фамилия

(подпись, дата)

Консультант по

экономической части \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.В. Федорова

(подпись, дата)

Консультант по

технологической части \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ И.О. Фамилия

(подпись, дата)

Студент, гр. ИС-19-2 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Д.С. Токмаков

(подпись, дата)

2023

Содержание

[**Введение** 3](#_Toc135755025)

[1 Анализ предметной области 4](#_Toc135755026)

[2 Техническое задание 5](#_Toc135755027)

[3 Моделирование предметной области 6](#_Toc135755028)

[4 Реализация разработки 7](#_Toc135755029)

[4.1 Программные средства реализации проекта 7](#_Toc135755030)

[4.2 Структура информационной системы (сайта) 7](#_Toc135755031)

[4.3 Описание программы или программных модулей в соответствии с требованиями к проекту 7](#_Toc135755032)

[5 Тестирование разработанного продукта 8](#_Toc135755033)

[5.1 Анализ и выбор методов тестирования 8](#_Toc135755034)

[5.2 Результаты тестирования 8](#_Toc135755035)

[6 Экономическое обоснование разработки 9](#_Toc135755036)

[7 Внедрение и установка информационной системы (сайта) 10](#_Toc135755037)

[8 Сопровождение и продвижение информационной системы (сайта) 11](#_Toc135755038)

[9 Настройка безопасности информационной системы (сайта) 12](#_Toc135755039)

[**Заключение** 13](#_Toc135755040)

[**Список использованных источников (литературы)** 14](#_Toc135755041)

Введение

В современном мире быстро меняющихся и развивающихся направлений и технологий решающую роль играет управление проектами, которое может обеспечить бизнесу принятие эффективных решений инициатив широкого спектра. На пути менеджеров проектов и команд появляются препятствия в виде проблем с планированием, организации процессов и их контроле для достижения желаемых результатов в рамках установленных ограничений. Для решения этих проблем необходимо разработать систему управления проектами.

Целью дипломного проекта является разработка системы управления дорожными картами проектов, позволяющую спланировать и организовать деятельность, а также оптимизировать использование ресурсов. В рамках разработки можно выделить следующие задачи:

* анализ предметной области, выделение основных функции и требований системы;
* выбор среды и инструментов разработки;
* разработка пользовательского интерфейса в соответствии с современными отраслевыми практиками, а также обеспечение его интуитивности и визуальной привлекательности;
* разработка архитектуры базы данных (БД) для хранения и поиска данных и инфраструктуры для взаимодействия с ней;
* реализация таких основных функций системы, как создание проектов, управление задачами, визуализация данных на временной шкале и совместная работа;
* тестирование функциональности и удобства использования;
* обеспечение безопасности данных и реализация мер защиты конфиденциальной информации.

Итогом дипломного проекта является удобное и надёжное решение, которое удовлетворяет потребности в системе управления проектами.

1 Анализ предметной области

Для определения функций и требований системы следует рассмотреть различные принципы и практики управления проектами.

В разработке и управлении существует большое множество методологий и фреймворков. Несколько из них представлены ниже:

* «Каскадная модель» - подразумевает последовательное прохождение стадий, каждая из которых должна завершиться полностью до начала следующей. В модели легко управлять проектом. Благодаря её жесткости, разработка проходит быстро, стоимость и срок заранее определены;
* «Agile» - это итеративный подход к управлению проектами и разработке программного обеспечения, который помогает командам быстрее и с меньшими проблемами поставлять ценность клиентам;
* «Scrum» - это методика гибкого управления проектами, помогающая командам структурировать работу и управлять ею на основе набора ценностей, принципов и практик;
* «PRINCE2» - это подход к управлению проектами, основанный на организации и контроле над всем проектом от начала и до конца. Это означает тщательное планирование проекта перед стартом, четкую организацию всех стадий процесса и тщательное устранение всех недоработок до того, как проект будет завершен.

Перед выделением основных функций системы следует рассмотреть аналоги, присутствующие на рынке. Ниже представлена таблица с функциями и ценами различных платформ.

Таблица 1 – Сравнение систем управления проектами

|  |  |  |
| --- | --- | --- |
| **Название** | **Функции** | **Цена/месяц на человека** |
| Asana | Хронология (диаграмма Ганта), канбан-доски, календарь, отчёты, цели, формы, автоматизация, рабочая нагрузка, интеграция с другими приложениями | Basic - 0€  Premium – 13,49€  Business – 30,49€ |
| Trello | Представления, автоматизация, шаблоны, интеграции, улучшения (плагины) | Free - 0$  Standard – 5$  Premium – 10$  Enterprise – 17,50$ |
| Microsoft Project | Диаграммы Ганта, управление ресурсами, планирование, составление бюджета, отчёты, сетка задач, доски, дорожная карта. | Project 1 – 9,40€  Project 3 – 28,10€  Project 5 – 10€ |
| Jira | Дорожные карты, Scrum-доски, отчёты, аналитика, автоматизация, просмотр репозитория, DevOps, | Free - 0$  Standard – 7,75$  Premium – 15,25$  Enterprise – 14$ (только годовая подписка от 800 человек) |

Система управления проектами должна реализовывать интуитивно понятный и удобный интерфейс, позволяющий пользователям создавать, редактировать и визуализировать дорожные карты проектов. Также система позволит эффективно управлять задачами, способствуя эффективному планированию и мониторингу на протяжении всего жизненного цикла проекта.

Одной из ключевых особенностей системы является визуализация данных в виде диаграммы Ганта. Такой подход позволяет получить целостное представление о ходе реализации проекта.

Система должна включать функцию совместной работы, позволяющие пользователям работать над одним проектом. Руководитель проекта должен иметь возможность управления над участниками и иметь самые полные права над проектом в то время, как участники могут только управлять задачами.

От системы также требуется уделение приоритетного внимания безопасности и конфиденциальности данных.

Разработка и внедрение системы позволит менеджерам и командам эффективно планировать, выполнять и сдавать проекты в срок, в рамках бюджета и в соответствии с установленными стандартами качества. Централизованная платформа для деятельности по управлению проектами повысит производительность, сотрудничество и возможности принятия решений, что приведет к улучшению результатов проекта.

2 Техническое задание

Назначение и область применения продукта

Тип сайта – система управления дорожными картами проектов.

1. Основные требования к интерфейсу системы:

* функциональность (соответствие задачам пользователя);
* понятность и логичность;
* обеспечение высокой скорости работы пользователя;
* наличие призыва к действию там, где это необходимо.

Требования к тексту элементов управления:

* названия элементов отражают их функции;
* названия элементов краткие, но понятные пользователю.

Требования к элементам управления:

* элементы меню сгруппированы;
* элементы меню имеют четкое разделение;
* ссылки визуально выделены.

Содержимое системы предоставляется заказчиком и включается разработчиком в ресурс.

1. Структура сайта

Карта сайта представлена на рисунке 1.

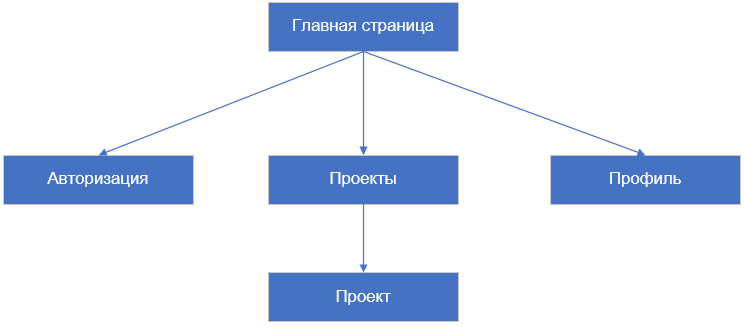


Рисунок 1 - Карта сайта

Первое, что видит пользователь при входе – главная страница.

Пользователь может авторизироваться в системе, перейдя на соответствующую страницу.

На странице профиля, пользователь может увидеть и изменить информацию о себе.

На странице с проектами, пользователь видит все доступные ему проекты.

На странице с проектом, пользователь может просмотреть диаграмму Ганта, управлять задачами и участниками, изменять информацию о проекте.

1. Функциональные требования

Пользователь начинает работу в систему с главной страницы.

Требования к главной странице:

* страница предоставляет возможность перехода на страницы авторизации и проектов;
* на странице расположен слоган.

Требования к странице входа в систему:

* страница предоставляет возможность выбора авторизации из нескольких провайдеров (GitHub и Google обязательны);
* при успешной авторизации на всех страницах изменяется шапка, в которой появляется аватар пользователя и выпадающий список с возможностью выйти из аккаунта и перейти на страницу с личным кабинетом.

Требования к личному кабинету пользователя:

* страница отображает аватар пользователя, его почту и имя профиля;
* страница отображает количество проектов, владельцем которых является пользователь;
* страница отображает количество проектов, участником которых является пользователь;
* страница отображает кнопку удаления профиля;
* пользователь имеет возможность редактирования имени своего профиля;
* пользователь имеет возможность удалить свой профиль;
* для удаления профиля нужно нажать на кнопку «Delete Account», которая откроет модальное окно;
* в модальном окне удаления профиля, требуется ввести адрес электронной почты и нажать на кнопку «Delete Account»;
* при удалении профиля удаляются все проекты, владельцем которых он был;
* при удалении профиля пользователь удаляется из всех проектов, где он был участником.

Требования к странице проектов:

* страница содержит список всех проектов в виде плитки, в которых пользователь является участником или владельцем;
* пользователь имеет возможность открыть модальное окно создания проекта, нажав на первый элемент в списке проектов;
* пользователь имеет возможность создания нового проекта, указав название и необязательное описание и нажав на кнопку «Create» в модальном окне;
* пользователь имеет возможность открыть проект, нажав на нужный элемент в списке.

Требования к странице проекта:

* страница показывает название проекта в верхней части;
* страница показывает такие вкладки, как «View», «Tasks», «Contributors», «Settings»;
* страница не позволяет пользователю перейти на вкладку «Contributors» и «Settings», если он не является владельцем этого проекта, и отображает эти вкладки в полупрозрачном виде;
* страница встречает пользователя вкладкой «View»;
* вкладка «View» отображает временную диаграмму Ганта, где на горизонтальной оси расположены даты, а на вертикальной оси расположены статусы задач;
* задача на диаграмме расположена в строке в соответствии со своим статусом и заполняет столбцы от указанного начала до указанного конца;
* задачи с одинаковым статусом отображаются на разных строках;
* на элементе задачи отображаются период задачи и её название;
* на вкладке «Tasks» отображается форма добавления новой задачи и список имеющихся задач;
* в форме добавления задачи можно ввести название, комментарий (необязательное поле), выбрать дату и статус задачи;
* для добавления задачи требуется ввести данные в форму и нажать на кнопку добавления;
* для удаления задачи требуется в списке задач нажать на кнопку удаления у соответствующей соответствующего элемента;
* для изменения задачи требуется перейти в режим изменения, нажав на кнопку изменения;
* перейдя в режим изменения задачи, пользователь может изменять те же поля формы, что были в форме добавления задачи;
* для сохранения задачи требуется нажать на кнопку сохранения изменений;
* пользователь имеет возможность создавать новые статусы задач в рамках проекта;
* для создания нового статуса задачи требуется нажать на выпадающий список статусов и нажать на кнопку «Add new», которая откроет модальное окно создания нового статуса;
* в модальном окне добавления статуса требуется ввести название и нажать кнопку «Create»;
* для удаления статуса требуется нажать на кнопку удаления рядом с соответствующим названием в списке статусов;
* кнопка удаления статуса не отображается, когда статус привязан хотя бы к одной задаче или является одним стандартных статусов;
* на вкладке «Contributors» отображается строка поиска участников и список участников текущего проекта;
* для добавления нового участника требуется указать в поисковой строке имя или адрес электронной почты пользователя, выбрать из списка представленных и нажать на кнопку добавления;
* элемент списка участников отображает аватар пользователя, имя профиля и адрес электронной почты;
* для удаления пользователя из участников нужно нажать на кнопку удаления рядом с соответствующим участником;
* на вкладке «Settings» отображается информация о проекте, кнопка сохранения изменений о проекте и его удаления;
* для изменения информации о проекте нужно ввести новую информацию в поля название и описание проекта (необязательное поле) и нажать на кнопку «Update»;
* для удаления проекта нужно нажать на кнопку «Delete», которое откроет модальное окно;
* в модальном окне удаления требуется ввести название своего проекта и нажать на кнопку «Delete Project»;
* при удалении проекта удаляются все статусы и задачи, связанные с этим проектом;
* при удалении проекта он открепляется от всех участников и владельца.

Система управления реализована для одного типа пользователей.

Работа в системе возможна только для авторизованных пользователей. Исключением является страница главная страница. При попытке перехода на страницы сайта кроме главной пользователя перенаправляет на страницу авторизации.

1. Требования к хранению данных

Все данные системы хранятся в структурированном виде под управлением реляционной системы управления базами данных (СУБД).

3 Моделирование предметной области

Модель предметной области - это визуальное представление концептуальных классов или объектов реального мира в терминах предметной области.

Целью моделирования предметной области является:

* + идентификация классов понятий или концептуальных классов;
  + детализация объектов предметной области;
  + концептуальное представление понятий предметной области.

IDEF0 - нотация графического моделирования, используемая для создания функциональной модели, отображающей структуру и функции системы, а также потоки информации и материальных объектов, связывающих эти функции. К ее особенностям можно отнести:

* использование контекстной диаграммы;
* поддержка декомпозиции;
* доминирование;
* выделение 4 типов стрелок.

Функциональная модель IDEF0 представляет собой набор блоков, каждый из которых представляет собой «черный ящик» со входами и выходами, управлением и механизмами, которые детализируются (декомпозируются) до необходимого уровня. Наиболее важная функция расположена в верхнем левом углу. А соединяются функции между собой при помощи стрелок и описаний функциональных блоков. При этом каждый вид стрелки или активности имеет собственное значение. Данная модель позволяет описать все основные виды процессов, как административные, так и организационные. Стрелки могут быть:

* Входящие - вводящие, которые ставят определенную задачу;
* Исходящие - выводящие результат деятельности;
* Контроль (сверху вниз) - механизмы управления (положения, инструкции и пр);
* Механизмы (снизу вверх) - что используется для того, чтобы произвести необходимую работу.

На рисунке А.1 (приложение А) представлен основной блок IDEF0 A-0.

Вводящими стрелками являются:

* «Опыт»;
* «Информация из сторонних источников».

Стрелки контроля:

* «Описание предметной области»;
* «Техническое задание»;
* «Спецификация языка программирования»;
* «План разработки».

Механизмы:

* «Аналитик-архитектор»;
* «Программист»;
* «ПО».

Исходящие стрелки:

* Система управления

На рисунке А.2 (приложение А) представлена декомпозиция IDEF0 A-1. Она показывает более детальное описание модели.

Unified Modeling Language (UML) - унифицированный язык моделирования, подразумевает создание модели, описывающей объект, подходит для широкого класса проектируемых программных систем, различных областей приложений, типов организаций, уровней компетентности, размеров проектов.

Варианты использования диаграмм UML:

* Проектирование. UML-диаграммы помогут при моделировании архитектуры больших проектов, в которой можно собрать как крупные, так и более мелкие детали и создать каркас (схему) приложения.
* Реверс-инжиниринг - создание UML-модели из существующего кода приложения, обратное построение. Может применяться в проектах поддержки, где существует написанный код, но документация неполная или отсутствует.
* Из моделей можно извлекать текстовую информацию и генерировать относительно удобочитаемые тексты - документировать. Текст и графика будут дополнять друг друга.

С помощью графической нотации UML можно визуализировать систему, объединить все компоненты в единую структуру, уточнять и улучшать модель в процессе работы. На общем уровне графическая нотация UML содержит 4 основных типа элементов:

* фигуры;
* линии;
* значки;
* надписи.

Диаграмма вариантов использования - это графическое изображение возможных взаимодействий пользователя с системой. Диаграмма вариантов использования показывает различные варианты использования и различные типы пользователей системы и часто сопровождается другими типами диаграмм. Случаи использования изображаются либо кругами, либо эллипсами. Действующие лица (акторы) часто изображаются в виде человечков. Акторы и прецеденты взаимодействуют друг с другом при помощи следующих отношений:

* Ассоциация;
* Обобщение;
* Включение;
* Расширение.

На рисунке А.3 (приложение А) представлена диаграмма прецедентов (Use case). В качестве актора выступает «Пользователь». Система имеет следующие прецеденты:

* Авторизироваться (ассоциируется с пользователем);
* Выйти из аккаунта (ассоциируется с пользователем);
* Создать проект (ассоциируется с пользователем);
* Редактировать проект (расширяет «Открыть проект»);
* Открыть проект (ассоциируется с пользователем);
* Редактировать профиль (ассоциируется с пользователем);
* Управлять участниками (расширяет «Открыть проект»);
* Редактировать задачу (расширяет «Открыть проект»);
* Создать задачу (расширяет «Открыть проект»).

Диаграмма последовательности – UML-диаграмма, на которой для некоторого набора объектов на единой временной оси показан жизненный цикл объекта (создание-деятельность-уничтожение некой сущности) и взаимодействие акторов (действующих лиц) информационной системы в рамках прецедента.

Взаимодействие между акторами отображается при помощи специальных стрелок, передающих управление от отправителя к получателю. Стрелки демонстрируют ход сценария и те события, которые происходят во время анализируемого прецедента. Всего существует 5 видов стрелок:

* Синхронное сообщение - актор-отправитель передаёт ход управления актору-получателю, которому необходимо провести в прецеденте некоторое действие. Пока проводимое актором-получателем действие не будет завершено, актор-отправитель теряет возможность производить какие-либо действия. Графически изображается как сплошная линия со стрелкой в виде закрашенного треугольника.
* Ответное сообщение - ответ на синхронное сообщение. Обычно, содержит какое-либо возвращаемое изначальному актору-отправителю значение, также возвращающее ему управление. Графически изображается пунктирной линией с открытой стрелкой.
* Асинхронное сообщение - актор-отправитель передаёт ход управления актору-получателю, которому необходимо провести в прецеденте некоторое действие, при этом не теряя возможность совершать другие действия. Графически изображается сплошной линией с открытой стрелкой.
* Потерянное сообщение - сообщение без адресата (есть отправитель, нет получателя).
* Найденное сообщение - сообщение без отправителя.

На рисунке А.4 (приложение А) представлена диаграмма последовательности (Sequence diagram) на прецедент авторизации. Действующими лицами на диаграмме являются:

* Пользователь;
* Браузер;
* Приложение;
* Сервер авторизации.

От пользователя требуется нажимать на кнопки и вводить данные для входа у сторонних сервисов. Приложение и сервер авторизации обмениваются токенами и сообщениями, чтобы сгенерировать токены доступа и сессии, которые нужны для авторизации.

Схема «сущность-связь» (также ERD или ER-диаграмма) — это разновидность блок-схемы, где показано, как разные «сущности» (люди, объекты, концепции) связаны между собой внутри системы. ER-диаграммы применяются для проектирования и отладки реляционных баз. ER-диаграммы состоят из прямоугольников, ромбов, овалов и соединительных линий для отображения сущностей, их атрибутов и связей.

В ER-моделях выделяют три уровня детализации:

* Концептуальная модель данных — схема наивысшего уровня с минимальным количеством подробностей. Достоинство этого подхода заключается в возможности отобразить общую структуру модели и всю архитектуру системы.
* Логическая модель данных содержит более подробную информацию. На этом уровне определяются более подробные операционные и транзакционные сущности. Логическая модель не зависит от технологии, в которой она будет применяться.
* Физическая модель данных: на основе каждой логической модели данных можно составить физические модели. В них должно присутствовать достаточно технических подробностей для составления и внедрения самой базы данных.

На рисунке А.5 (приложение А) представлена схема базы данных.

Сущностями схемы являются:

* Account – содержит информацию о профиле пользователя;
* ContributedProject – промежуточная таблица для связи «многие-ко-многим» между сущностями Account и Project. Отслеживает все проекты, в которых пользователь является участником;
* Project – содержит информацию о проекте;
* Task – содержит информацию о задаче. Каждая задача может принадлежать только одному проекту и содержать один из статусов проекта;
* Status – содержит информацию о статусах. Каждый статус может принадлежать только одному проекту.

Моделирование предметной области позволяет идентифицировать, классифицировать и детализировать основные сущности и связи между различными частями системы, что может привести к сокращению времени разработки и более эффективному распределению ресурсов.

4 Реализация разработки

4.1 Программные средства реализации проекта

Перед разработкой приложения требуется определить ряд технических и программных средств, которые будут использоваться. Создание веб-сайта включает в себя кодирование, дизайн и другие направления.

Редактор кода - это инструмент разработки, предоставляющий интерфейс для написания и редактирования кода. Он может содержать функции статического анализа, организации кода, отладки и обнаружения ошибок.

Ниже расположена таблица сравнения различных редакторов кода.

|  |  |  |  |
| --- | --- | --- | --- |
| Особенности | VS Code | Neovim | WebStorm |
| Платформа | Windows, macOS, Linux | Windows, macOS, Linux | Windows, macOS, Linux |
| Поддержка языков | Широкий спектр языков программирования | Широкий спектр языков программирования | Основной упор на JavaScript и TypeScript |
| Пользовательский интерфейс | Современный, интуитивный и настраиваемый | Минималистичный и основанный на терминале | Утонченная и богатая IDE |
| Экосистема расширений | Обширный и динамичный маркетплейс | Обширная экосистема плагинов | Ограниченная, но включает важные плагины для веб-разработки |
| Отладка | Встроенный отладчик с точками останова | Поддержка отладки с помощью плагинов | Полнофункциональный отладчик с расширенными возможностями |
| Управление версиями | Интеграция с Git и инструментами контроля версий | Интеграция с Git и плагины | Интеграция с Git и функции, специфичные для систем контроля версий |
| Инструменты продуктивности | Богатый набор расширений и инструментов продуктивности | Высокая настраиваемость и расширяемость | Умное автозаполнение кода, рефакторинг и инструменты продуктивности |
| Совместная работа | Live Share для совместной работы в реальном времени | Возможности совместной работы через плагины | Ограниченные возможности совместной работы |
|  |  |  |  |
| Производительность | Быстрый и отзывчивый | Легковесный и очень эффективный | В целом хорошая производительность, но может требовать ресурсов |
| Сложность изучения | Легко осваивается и подходит для начинающих | Более крутой и крутая кривая обучения | Пользовательский интерфейс дружественный с подробной документацией |
| Стоимость | Бесплатный и с открытым исходным кодом | Бесплатный и с открытым исходным кодом | Коммерческая лицензия с бесплатной пробной версией |

Редакторы: кастомизация, удобность и привлекательность интерфейса, сравнение фич

* Microsoft Visual Studio;
* VS Code;
* WebStorm;
* Neovim.

Фреймворки: сравнение по производительности, удобности, фанбаза

React;

Vue;

Svelte;

Solid;

Astro.

Next как SSR платформа для React.

Git как система контроля версий.

Typescript как ЯП.

4.2 Структура информационной системы (сайта)

фываыа

4.3 Описание программы или программных модулей в соответствии с требованиями к проекту

Ыафыва

5 Тестирование разработанного продукта

5.1 Анализ и выбор методов тестирования

фываыаыав

5.2 Результаты тестирования

Фываыаыав

6 Экономическое обоснование разработки

Фываыаыа

7 Внедрение и установка информационной системы (сайта)

Фывафыаыав

8 Сопровождение и продвижение информационной системы (сайта)

Фываыфаываыа

9 Настройка безопасности информационной системы (сайта)

Фывафываыа

Заключение

Фываыфава

Список использованных источников (литературы)

1. Ещё раз про семь основных методологий разработки // [Хабр]. URL: <https://habr.com/ru/companies/edison/articles/269789/> (дата обращения: 5.23.2023);
2. Модели и методологии разработки ПО // [GeekBrains]. URL: <https://gb.ru/posts/methodologies> (дата обращения: 5.23.2023);
3. Что такое agile? // [Atlassian]. URL: <https://www.atlassian.com/ru/agile> (дата обращения: 5.23.2023);
4. Что такое Scrum? // [Atlassian]. URL: <https://www.atlassian.com/ru/agile/scrum> (дата обращения: 5.23.2023);
5. PRINCE2 — ведущая методология управления проектами // [wrike]. URL: <https://www.wrike.com/ru/blog/prince2-vedushhaya-metodologiya-upravleniya-proektami/> (дата обращения: 5.23.2023);
6. Моделирование предметной области // [Studme]. URL: <https://studme.org/174167/informatika/modelirovanie_predmetnoy_oblasti> (дата обращения: 5.24.2023);
7. UML для бизнес-моделирования: зачем нужны диаграммы процессов // [Evergreen]. URL: <https://evergreens.com.ua/ru/articles/uml-diagrams.html> (дата обращения: 5.24.2023);
8. IDEF0. Знакомство с нотацией и пример использования // [TRINION]. URL: <https://trinion.org/blog/idef0-znakomstvo-s-notaciey-i-primer-ispolzovaniya> (дата обращения: 5.24.2023);
9. Диаграмма прецедентов // [Википедия]. URL: <https://en.wikipedia.org/wiki/Use_case_diagram> (дата обращения: 5.25.2023)
10. Что такое ER-диаграмма и как ее создать? // [Lucidchard]. URL: <https://www.lucidchart.com/pages/ru/erd-диаграмма> (дата обращения: 5.25.2023);

Приложения А

**(обязательное)**

**Диаграммы проектирования**

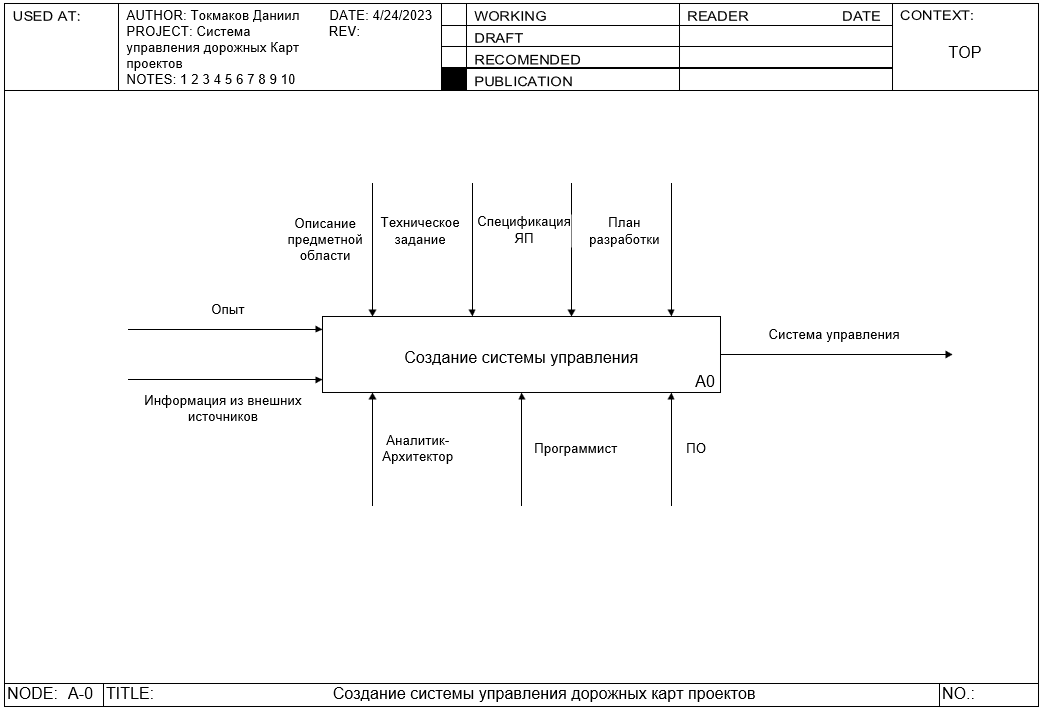


Рисунок A.1 - Основной блок IDEF0

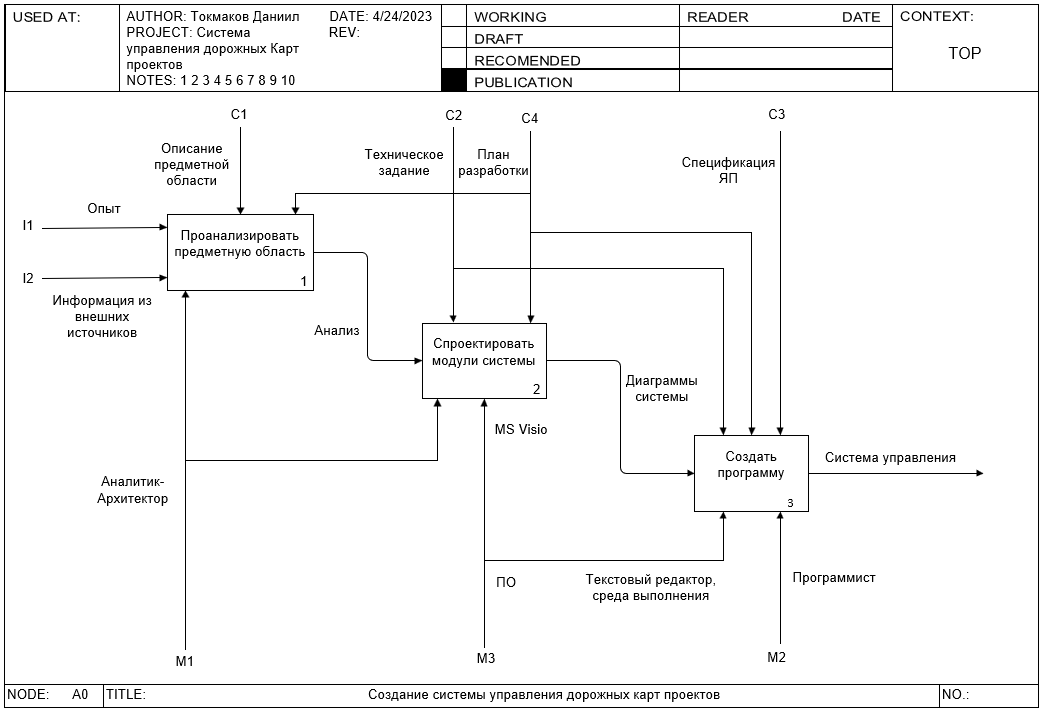


Рисунок A.2 - Декомпозиция IDEF0

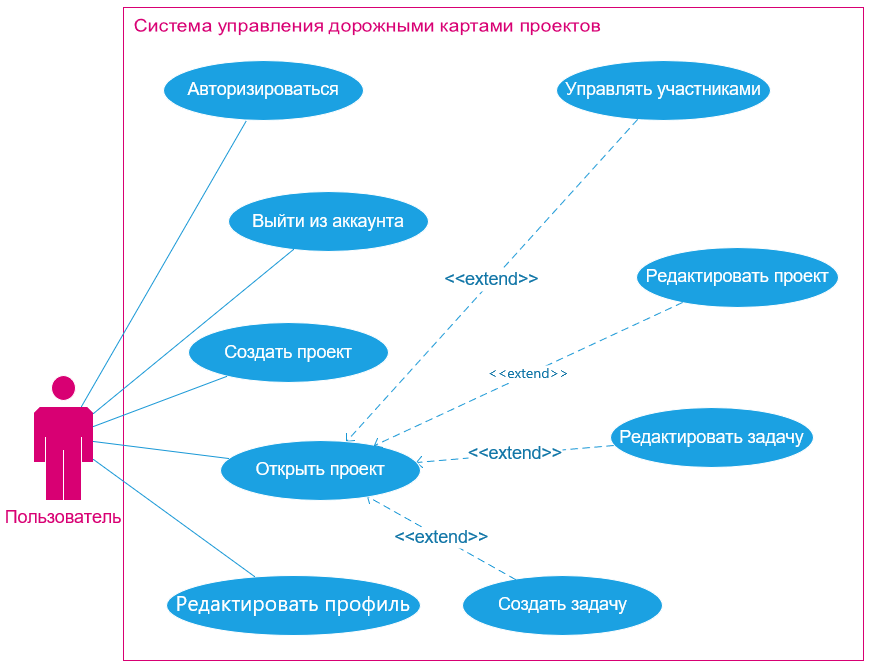


Рисунок А.3 – Диаграмма прецедентов

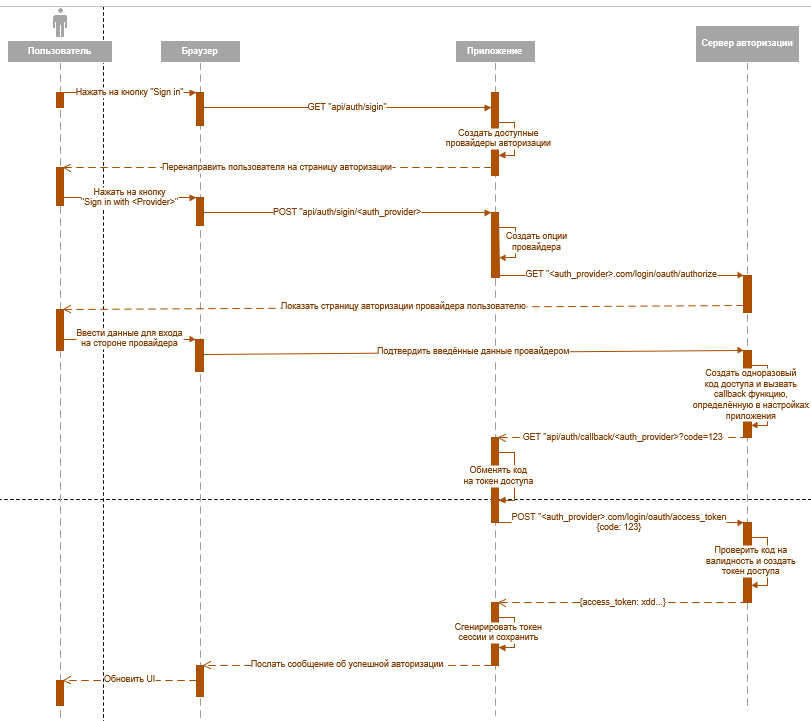


Рисунок А.4 – Диаграмма последовательности

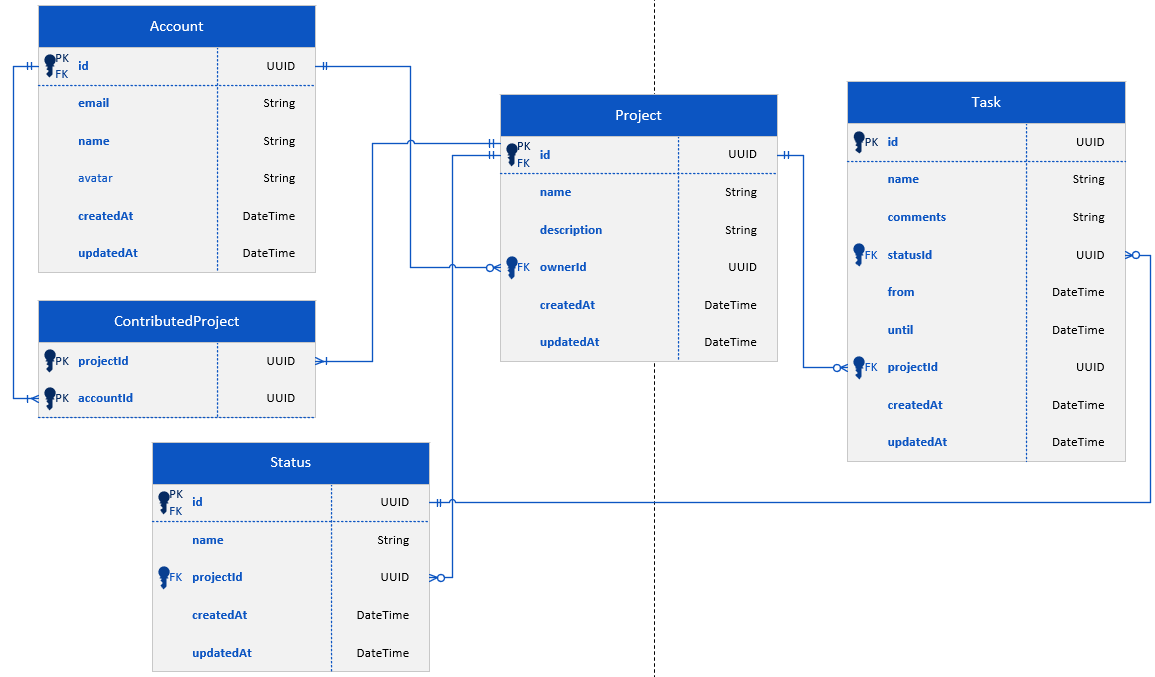


Рисунок А.5 – Схема базы данных