

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»**

Кафедра Математического и компьютерного
моделирования

Проектирование и разработка информационной системы:

«Дашборд» для руководителя

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 441 группы

направление 09.03.03 — Прикладная информатика

механико-математического факультета

Попова Даниила Константиновича

Научный руководитель
доцент, к.ф.-м.н., доцент

Е.Ю. Крылова

Зав. кафедрой
зав. каф., д.ф.-м.н., доцент

Ю.А. Блинков

Саратов 2023

Введение. В условиях быстро развивающихся технологий и нарастающей конкуренции увеличивается требования к руководителям современных организаций. Важной задачей руководителя является эффективный контроль и анализ операционных данных, что позволяет более точно и осознанно принимать стратегические решения в своей деятельности. Однако, в большинстве случаев, сбор и анализ данных оказывается трудоемким и малоэффективным процессом.

Дашборд (англ. - Dashboard) — это информационная система (ИС), которая используется для отображения данных в удобном и понятном виде с целью мониторинга и анализа бизнес-показателей, трендов и событий. Дашборд позволяет визуализировать различные типы информации в реальном времени. Он обычно разрабатывается на основе данных, которые активно используются в компании, например, продажи, финансы, производство, маркетинг и т.д. Дашборды предоставляют пользователю интерактивный доступ к данным и могут быть настроены таким образом, чтобы адаптироваться к потребностям конкретного пользователя. Они могут использоваться для принятия стратегических решений, обнаружения возможностей для увеличения прибыли, улучшения производительности и оптимизации бизнес-процессов.

Основными функциями дашборда являются:

- анализ бизнес-показателей и метрик;
- мониторинг процессов и операций;
- обнаружение трендов и прогнозирование будущих результатов;
- планирование, контроль и управление бизнес-процессами;
- взаимодействие с другими системами и инструментами.

Основными этапами разработки дашборда как программного обеспечения (ПО) являются: анализ требований бизнеса, проектирование дашборда, реализация дашборда, тестирование и внедрение.

Таким образом, проектирование и разработка информационной системы дашборда для руководителя является актуальной задачей, так как данный инструмент обеспечивает быстрый доступ к ключевым индикаторам производительности.

Цель работы. Целью данной бакалаврской работы является построение информационной системы «дашборд для руководителя».

Задачи работы. В процессе выполнения работы были поставлены следующие задачи:

1. формирование технического задания;
2. проектирование ИС средствами UML нотаций;
3. проектирование базы данных;
4. разработка серверной части ПО;
5. проектирование пользовательского интерфейса;
6. разработка клиентской части ПО;
7. настройка функционирования ПО в сети «интернет».

Краткая характеристика материалов исследования. Работа основана на источниках, указанных в списке литературы. В работе произведены проектирование и реализация информационной системы при помощи фреймворков FastAPI, Vue.js, базы данных PostgreSQL и UML нотаций.

Описание структуры. Работа состоит из введения, четырех разделов, заключения, списка использованных источников содержащего 20 наименований и приложения.

Основное содержание работы. Во введении кратко описывается содержание представленной работы, а так же раскрываются особенности проектирования и разработки информационных систем.

В первом разделе рассматриваются основные теоретические аспекты проектирования информационной системы и технологии для разработки ПО.

UML (Unified Modeling Language) является наиболее универсальным и распространенным инструментом для проектирования информационных систем. UML позволяет проектировать как структурную, так и поведенческую составляющие системы. С помощью UML можно создавать различные диаграммы, такие как диаграммы классов, диаграммы последовательностей, диаграммы прецедентов и т.д. UML является стандартом в разработке ПО и широко используется в индустрии.

ER-диаграмма (Entity-Relationship diagram) — это графическая модель данных, используемая для описания сущностей (entities), их атрибутов и взаимосвязей в базе данных. Она представляет собой удобный инструмент для проектирования и разработки баз данных.

Клиент-серверная архитектура — это подход к построению информационных систем, при котором все основные вычислительные ресурсы распределены между двумя типами программных модулей: клиентом и сервером, которые обмениваются данными между собой.

Развертывание веб-приложений в сети «интернет» будет производиться с помощью программы `docker`, установленной на виртуальном сервере и веб-сервера `Traefik` — это современный и эффективный подход, который позволяет распределять и управлять ресурсами, а также обеспечивает новый уровень безопасности и надежности.

В результате проведенного анализа различных инструментов для разработки был выбран Python для серверной части, базы данных PostgreSQL и фреймворка FastApi, а также Vue.js для клиентской стороны.

Python — это высокоуровневый, интерпретируемый язык программирования, который широко используется для создания серверной части веб-приложений. Python имеет множество библиотек и фреймворков для веб-разработки, что делает его очень гибким и мощным инструментом. Большое количество проектов с открытым исходным кодом также доступно на Python. Это позволяет разработчикам быстро получить поддержку сообщества и решить возникающие проблемы.

FastAPI — это новый, но очень быстрый и быстроразвивающийся фреймворк для REST API на Python. Он оптимизирован для высокой производительности, имеет поддержку асинхронных операций и генерирует автоматическую документацию API в Swagger. Он также имеет удобную и гибкую систему зависимостей, которая упрощает создание и использование классов, реализующих модульную архитектуру.

PostgreSQL — это мощная реляционная база данных с открытым исходным кодом, которая обеспечивает высокую производительность, удобную систему управления, расширяемость и надежность. PostgreSQL обладает широкими возможностями при разработке сложных приложений, так как имеет возможность использовать многие виды данных и различные типы индексов.

Vue.js — это мощный, простой в изучении и использовании JavaScript-фреймворк для создания современных веб-приложений. Он имеет обширную документацию и богатый набор инструментов, которые позволяют разработ-

чикам быстро создавать интерактивные веб-приложения. Vue.js также имеет большое сообщество разработчиков и множество плагинов, что делает его очень гибким и расширяемым.

Таким образом, выбор Python для серверной части, фреймворка FastAPI, базы данных PostgreSQL и Vue.js для клиентской части является оптимальным и обоснованным для создания современного веб-приложения, которое будет обладать высокой производительностью, надежностью и легкостью разработки.

Второй раздел посвящен проектированию и разработки серверной части ПО.

Серверная часть приложения отвечает за обработку, хранение и передачу данных, а также за выполнение бизнес-логики, такой как проверка подлинности пользователей, обновление базы данных и вычисление результатов действий. Именно на серверной части происходит обработка запросов, поступающих от клиента, и формирование ответов, включая пересылку необходимых данных.

В данном разделе посвященном разработке серверной части приложения, в рамках которой будет проектироваться и описываться весь необходимый функционал необходимого для работы серверной части приложения.

В техническом задании было описано что руководитель использует приложение «дашборд руководителя» для просмотра списка проектов и общей информации по бюджету проекта. Менеджер проектов заполняет справочники компаний, сотрудников, проектов в рамках «битрикс24» и выставляет права доступа сотрудникам в приложении «бюджет проектов» для разграничения доступа к информации, а так же прикрепляет компанию заказчика к проекту. Сотрудник использует только приложение «бюджет проектов» для просмотра доступных проектов и заполняет статьи расходов, доходов бюджета проекта.

Согласно техническому заданию была разработана UML-диаграмм вариантов использования ИС «дашборд для руководителя», представленная в соответствии с рисунком 1.

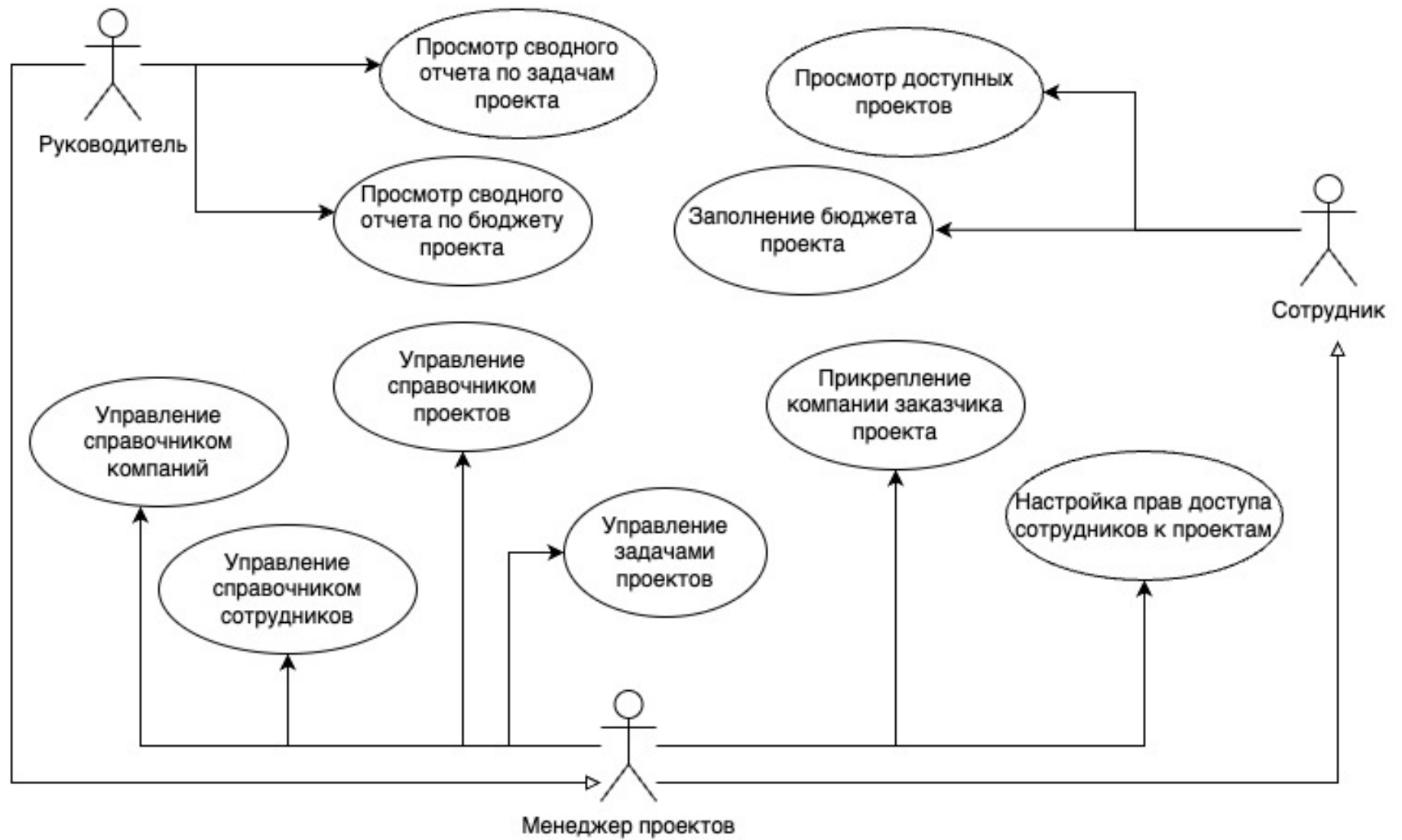


Рисунок 1 — Диаграмма вариантов использования ИС «дашборд для руководителя»

Также была разработана ER диаграмма базы данных, которая представлена в соответствии с рисунком 2.

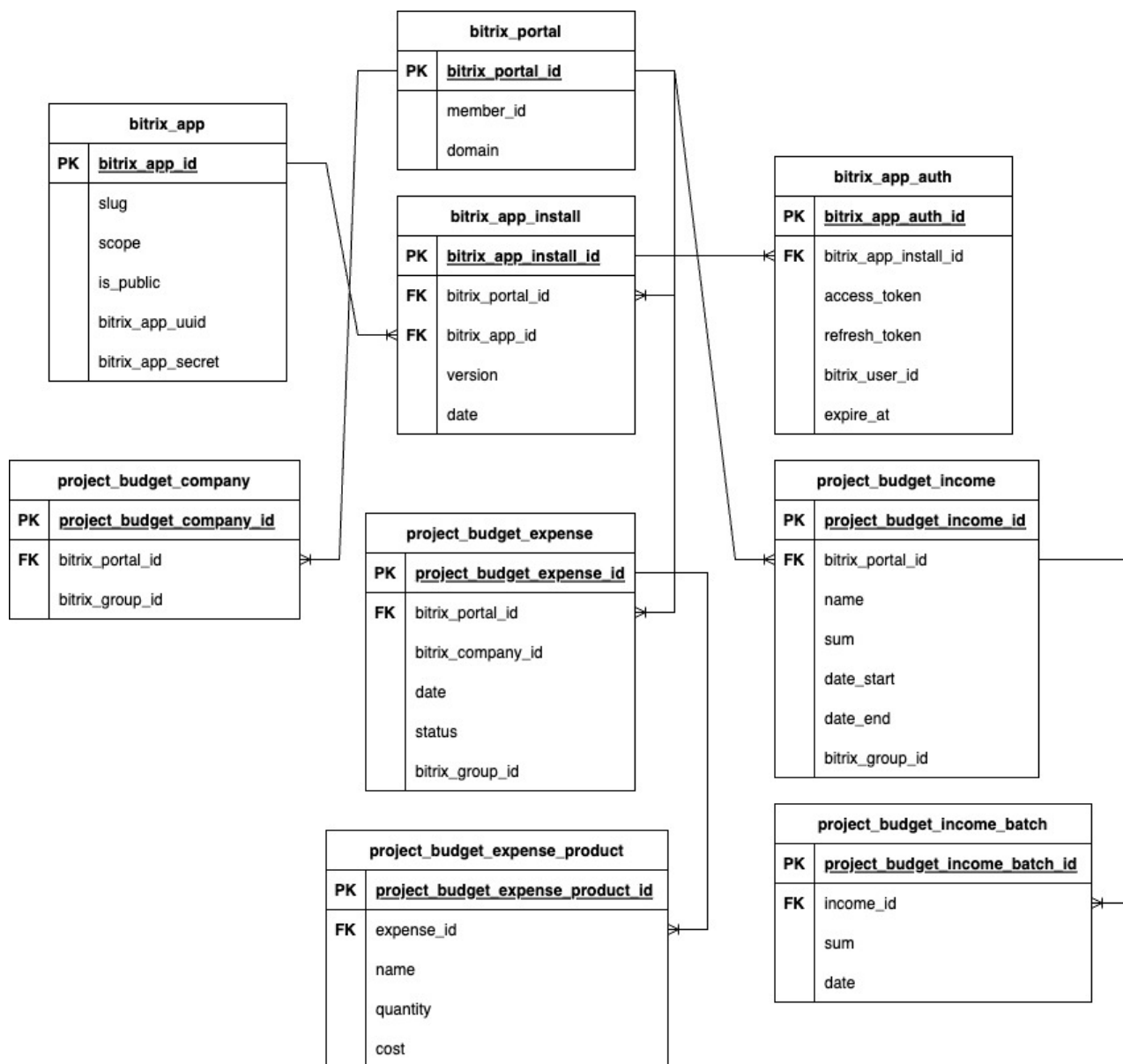


Рисунок 2 — ER диаграмма базы данных

При разработке базовой структуры проекта на FastAPI была использована среда для запуска приложения docker, что позволило создать это приложение с высокой воспроизводимостью, портируемостью и изолированностью. Кроме того, для создания миграций базы данных была использована библиотека alembic, которая предоставляет мощный и гибкий инструмент для управления структурами базы данных.

После была проведена разработка интерфейса REST API для взаимодействия клиента с серверной частью ПО. В рамках данной разработки были определены спецификации API, которые описывают возможности и параметры запросов и ответов. Ключевой особенностью была высокая степень гибкости и масштабируемости API, что позволило разработчикам менять и добавлять функциональность API без необходимости в изменении клиентского приложения.

Третий раздел посвящен проектированию и разработки клиентской части ПО.

Согласно техническому заданию, в процессе разработки клиентской части ПО был разработан графический интерфейс, который обладал необходимым функционалом для руководителя. Для реализации этого требования был разработан прототип графического интерфейса в среде Figma.

В процессе разработки прототипа графического интерфейса в Figma были проработаны основные элементы интерфейса, стилистика, расположение элементов и их функционал. Прототип позволил получить полное представление о будущем интерфейсе и его функциональных возможностях.

После создания прототипа графического интерфейса был описан процесс разработки клиентской части ПО, описана базовая структура проекта, взаимодействие с серверной частью.

Базовая структура проекта включала в себя файлы HTML, CSS и JavaScript, организованные по отдельным папкам и подпапкам в соответствии с их функциональным назначением. Также были созданы отдельные файлы для обработки AJAX-запросов к серверу.

Для взаимодействия с серверной частью были использованы REST API и AJAX-запросы. При этом все запросы были асинхронными, что позволило повысить скорость работы приложения и улучшить взаимодействие с сервером.

Таким образом, разработка клиентской части ПО включала в себя создание базовой структуры проекта, использование необходимых технологий и библиотек, а также организацию взаимодействия с серверной частью с помощью REST API и AJAX-запросов. В результате было разработано рабо-

тающее приложение с удобным графическим интерфейсом и возможностью получения и обработки данных с сервера в режиме реального времени.

Четвертый раздел отображает процесс настройки функционирования ПО в сети «интернет».

Была реализована настройка автоматического развертывания с использованием GitLab CI/CD. Эта конфигурация позволяет автоматически выполнять задачи тестирования, настройки и запуска новой версии дашборда для множества целевых систем на основе GitLab CI/CD.

GitLab Runner — это агент, который запускает задачи, созданные в проекте на GitLab. Он позволяет запускать задачи на разных операционных системах и архитектурах. GitLab Runner может быть установлен на отдельной машине или использоваться в качестве сервиса в Docker-контейнере.

Был настроен GitLab Runner, который выполняет задачи по сборке и тестированию новых версий дашборда. После успешной сборки и тестирования, контейнер с новой версией автоматически разворачивается на тестовых и эксплуатационных площадках с помощью GitLab CI/CD.

Цикл автоматической развертки с нуля до активной работы на целевой системе занимает всего несколько минут, что значительно сокращает время развертывания новых версий дашборда.

Для автоматической развертки был использован инструмент Continuous Integration/Continuous Delivery (CI/CD), который позволяет автоматизировать процессы сборки, тестирования и развертывания программного обеспечения.

CI/CD позволил значительно сократить время развертывания новых версий дашборда, т.к. весь процесс проходит автоматически и не требует ручной работы. Благодаря этому удалось увеличить скорость разработки и улучшить качество программного обеспечения благодаря более быстрой обратной связи и быстрому выявлению ошибок.

При использовании CI/CD, после каждого коммита в репозиторий происходит автоматическая сборка, запуск тестов и развертывание на целевой системе. Это позволяет быстро обнаруживать ошибки и устранять их еще до того, как они станут критичными.

Таким образом, благодаря использованию CI/CD удалось существенно ускорить цикл развертывания новых версий дашборда и повысить качество программного обеспечения.

Заключение. В результате использования средств UML для проектирования информационной системы дашборда для руководителя была создана модель вариантов использования, позволяющая быстро получать и управлять необходимыми данными для пользователей разного уровня.

Для хранения данных была спроектирована база данных на основе ER диаграммы, определяющей наличие связей между сущностями и их атрибутами.

Так же была спроектирована и разработана серверная и клиентская часть программного обеспечения с использованием современных технологий и языков программирования для создания высокопроизводительного, надежного и удобного в использовании программного обеспечения позволяющая руководству компании оперативно получать информацию, анализировать ее и принимать правильные решения для сохранения и расширения бизнеса.

В ходе работы были реализованы и подробно описаны все стадии разработки информационных систем.

Задачи, поставленные в начале работы полностью выполнены.

Код, приведенный в приложении, полностью готов к использованию.

В приложениях также представлен акт о внедрении разработанной информационной системы в компанию BPM-Control.