**Минобрнауки России**

**Юго-Западный государственный университет**

Кафедра программной инженерии

**ОТЧЕТ**

о преддипломной практике

наименование вида и типа практики

на (в) ООО «МЦОБ. Онлайн-сервисы»

наименование предприятия, организации, учреждения

Студента 4 курса, группы ПО-01б

курса, группы

Шаталова Никиты Денисовича

фамилия, имя, отчество

Руководитель практики от предприятия, организации, учреждения

Оценка

должность, звание, степень

фамилия и. о. подпись, дата

Руководитель практики от университета

Оценка

должность, звание, степень

фамилия и. о. подпись, дата

Члены комиссии

подпись, дата фамилия и. о.

подпись, дата фамилия и. о.

Курск, 2024 г.

СОДЕРЖАНИЕ

[1 Анализ предметной области 4](#_Toc167879857)

[1.1 Общая информация 4](#_Toc167879858)

[1.2 Основные понятия и терминология проектного управления 8](#_Toc167879859)

[1. Фазы проекта 9](#_Toc167879860)

[1.4 Ключевые характеристики проекта 9](#_Toc167879861)

[1.5 Методологии и подходы к проектному управлению 10](#_Toc167879862)

[1.6 Участники проектом 12](#_Toc167879863)

[2 Техническое задание 14](#_Toc167879864)

[2.1 Основание для разработки 14](#_Toc167879865)

[2.2 Цель и назначение разработки 14](#_Toc167879866)

[2.3 Требования к программной системы 14](#_Toc167879867)

[2.3.1 Требования к данным программной системы 14](#_Toc167879868)

[2.3.2 Функциональные требования к программной системе 15](#_Toc167879869)

[2.3.3 Моделирование вариантов использования 16](#_Toc167879870)

[2.3.4 Требования пользователя к интерфейсу 18](#_Toc167879871)

[2.4 Нефункциональные требования к системе 19](#_Toc167879872)

[2.4.1 Требования к надёжности 19](#_Toc167879873)

[2.4.2 Требования к программному обеспечению 19](#_Toc167879874)

[2.4.3 Требования к аппаратному обеспечению 19](#_Toc167879875)

[2.4.4 Требования к оформлению документации 20](#_Toc167879876)

[3 Технический проект 21](#_Toc167879877)

[3.1 Общая характеристика организации решения задач 21](#_Toc167879878)

[3.2 Обоснование выбора технологии проектирования 21](#_Toc167879879)

[3.2.1 Описание используемых технологий и языков программирования 21](#_Toc167879880)

[3.2.1.1 Язык программирования PHP 21](#_Toc167879881)

[3.3 Проектирование архитектурной программной системы 23](#_Toc167879882)

[3.2.1 Описание сущностей программной системы 23](#_Toc167879883)

[3.3.2 Компоненты программной системы 25](#_Toc167879884)

[3.4 Проектирование пользовательского интерфейса 27](#_Toc167879885)

# Анализ предметной области

## Общая информация

В конце столетия Фредерик Тейлор (1856–1915 гг.) начал свои подробные исследования труда. Он применял при этом научные рассуждения, доказывая, что труд можно анализировать и улучшать, выделяя его элементарные составляющие. Он применял свои идеи к таким задачам на сталелитейных заводах, как засыпка песка и поднятие и перемещение деталей. До этого считалось, что единственный способ повысить производительность — это заставлять рабочих работать больше и дольше. В разрез с этим представлением Тейлор ввел понятие эффективной работы. Надпись на надгробии Тейлора в Филадельфии свидетельствует о важности его вклада в историю управления: «Отец научного управления».

Ученик Тейлора Генри Гант (1861–1919 гг.) очень подробно изучал последовательность операций при работе. Его исследование управления было сконцентрировано на кораблестроении во время Первой мировой войны. Диаграммы Ганта, включая отрезки задач и маркеры вех, показывают последовательность и продолжительность всех задач в процессе. Диаграммы Ганта оказались настолько полезным средством анализа для руководителей, что они практически не изменились за почти сто лет. Только в начале 1990-х гг. к отрезкам задач были впервые добавлены линии связей, которые отражают более точные зависимости между задачами.

Во время Второй мировой войны сложность правительственных и военных проектов, в также сокращение трудовых ресурсов привели к необходимости создания новых организационных структур. Были разработаны сложные сетевые диаграммы, которые назывались диаграммами PERT, и метод критического пути, которые предоставили руководителям возможность лучше контролировать очень сложные проекты с большим количеством инженерных работ (такие как создание боевых систем, требующее огромного числа задач и многочисленных операций в различные моменты времени).

Ученик Тейлора Генри Гант (1861–1919 гг.) очень подробно изучал последовательность операций при работе. Его исследование управления было сконцентрировано на кораблестроении во время Первой мировой войны. Диаграммы Ганта, включая отрезки задач и маркеры вех, показывают последовательность и продолжительность всех задач в процессе. Диаграммы Ганта оказались настолько полезным средством анализа для руководителей, что они практически не изменились за почти сто лет. Только в начале 1990-х гг. к отрезкам задач были впервые добавлены линии связей, которые отражают более точные зависимости между задачами.

Во время Второй мировой войны сложность правительственных и военных проектов, в также сокращение трудовых ресурсов привели к необходимости создания новых организационных структур. Были разработаны сложные сетевые диаграммы, которые назывались диаграммами PERT, и метод критического пути, которые предоставили руководителям возможность лучше контролировать очень сложные проекты с большим количеством инженерных работ (такие как создание боевых систем, требующее огромного числа задач и многочисленных операций в различные моменты времени).

Планирование «снизу вверх». В этой тенденции основной упор делается на простую структуру проекта, сокращение цикла проекта, эффективную совместную работу в группе, более глубокое вовлечение членов рабочей группы и принятие решений. Эта тенденция широко известна как динамичное управление проектами, и она включает связанные методики, такие как Scrum, Crystal, Extreme Programming, Unified Process и т. д.

Планирование и анализ «сверху вниз». Эта тенденция характеризуется принятием решений в масштабе всего предприятия относительно портфеля проектов, который должна иметь организация, а также использованием технологий интеллектуального анализа данных для более понятного представления информации в портфеле.

Таким образом, основные вехи истории управления проектами.

30-50 годы – начало управления проектами:

1937г. – американским ученым Гуликом была осуществлена первая разработка по матричной организации для руководства и осуществления сложных проектов.

1956г. – компания «Дюпон де Немур» образовала группу для разработки методов и средств управления проектами.

1957г. – коллективом Remington Rand, возглавляемым Kelly и Walker, был разработан метод критического пути (CPM) с программной реализацией на ЭВМ UNIVAC.

1957-58 гг. – для программы Поларис (US Navy) была разработана и опробована система сетевого планирования PERT.

1959г. – комитетом Андерсона (NASA) был предложен системный подход к управлению проектом по стадиям его жизненного цикла, в котором особое внимание уделялось предпроектному анализу.

Разработанные в 1956-58гг. методы и техника сетевого планирования дали мощный толчок развитию УП.

Развитие УП в 50-е годы завершилось публикацией Gaddis в Harvard Business Review первой обобщающей статьи по управлению проектами.

60-е годы – развитие методов сетевого планирования:

Развитие УП концентрируется почти исключительно на методах и средства PERT и CPM.

Распространение сетевых методов УП в Европу и другие континенты.

Появление матричной формы организации.

Разрабатывается целостная система материально-технического обеспечения (1966)

Появляется система GERT (1966), использующая новую генерацию сетевых моделей

Создаются профессиональные организации управления проектами

В Европе – Международная Ассоциация управления проектами (IPMA) – 1965г.

В Северной Америке – Институт управления проектами (PMI) – 1969г.

70-е годы – развитие системного подхода к управлению проектами

в УП учитывается «внешнее» окружение проектов.

1971г. — решаются проблемы руководителя проекта и команды проекта.

1977г. – разрабатываются методы управления конфликтами.

1977-79 гг. – разрабатываются организационные структуры УП.

Создаются профессиональные организации УП:

В Австралии (AIPM)

В Азии (ENAA)

В 80-е годы – управление проектами сформировалось как сфера профессиональной деятельности.

В начале 80-х – высокий уровень неудач воплощения УП.

Развиваются методы УП в строительстве с ориентацией на заказчика.

В практику входят методы управления изменениями.

Развивается управление качеством в проекте.

Управление риском выделяется в самостоятельную дисциплину в сфере УП.

В США публикуется первая версия коллективной работы института IPM – Project Management Body of Knowledge (Свод знаний по УП), в которой определены место, роль и структура методов и средств УП и их вклад в общее управление.

В 90-е годы – новые направления и сферы приложения управления проектами:

Начало трансферта знаний и опыта УП в развивающиеся страны.

Осознание возможностей и полезности применения УП в нетрадиционных сферах.

Осознание необходимости и практическое начало процессов глобализации, унификации и стандартизации в области УП.

Начало разработки и использования в УП новых информационных технологий, в т.ч. сетевых.

Разработка и ввод в действие программ сертификации менеджеров проекта.

Разработка и ввод в действие международных (ISO 10006-10007) и национальных (APM, PMI, AI PM) стандартов по управлению проектами.

## Основные понятия и терминология проектного управления

Проект – это временное предприятие, направленное на создание уникального продукта, услуги или результата. В отличие от операционной деятельности, которая является постоянной и повторяющейся, проект имеет четко определенные начало и конец, а также уникальные цели и задачи, которые должны быть достигнуты.

Также стоит рассмотреть такие термины как:

Цели проекта - конкретные результаты или продукты, которые должны быть достигнуты в рамках проекта. Это могут быть как материальные, так и нематериальные результаты.

Сроки проекта - временные рамки, в которые проект должен быть завершён. Включает в себя дату начала и окончания проекта, а также ключевые вехи (milestones).

Бюджет проекта - финансовые ресурсы, выделенные для выполнения проекта. Включает все затраты, связанные с проектом, такие как расходы на материалы, труд, оборудование и услуги.

Качество - степень соответствия продукта или услуги требованиям и ожиданиям заказчика. Включает в себя контроль за тем, чтобы результаты проекта соответствовали заданным стандартам и спецификациям.

Риск проекта - потенциальные события или условия, которые могут негативно повлиять на проект. Включает в себя как внутренние, так и внешние факторы, которые могут привести к отклонениям от плана.

Жизненный цикл проекта – это последовательность фаз, через которые проходит проект от начала до завершения. Основные фазы включают инициацию, планирование, исполнение, мониторинг и контроль, а также завершение.

Проектное управление – это применение знаний, навыков, инструментов и методов к проектным видам деятельности для удовлетворения требований проекта. Это включает планирование, организацию, мотивацию и контроль ресурсов с целью достижения конкретных целей и задач в рамках установленных временных, бюджетных и ресурсных ограничений.

## Фазы проекта

Инициация: определение и документирование целей и задач проекта, обоснование его необходимости, оценка начальных требований и ограничений. Подготовка устава проекта.

Планирование: разработка детального плана проекта, включающего объем работ, расписание, бюджет, ресурсы и план управления рисками. Создание структуры декомпозиции работ (WBS).

Исполнение: реализация плана проекта с координацией работы команды, распределением ресурсов и выполнением задач. Включает управление качеством.

Мониторинг и контроль: постоянное отслеживание прогресса проекта, контроль выполнения задач и внесение необходимых корректировок. Управление изменениями, контроль затрат и сроков, оценка и управление рисками.

Завершение: формальное закрытие проекта, включая сдачу конечного продукта заказчику, документирование уроков и архивацию проектной документации. Проведение итогового анализа и оценка достигнутых результатов.

## Ключевые характеристики проекта

Временность:

- проект всегда имеет определенные временные рамки: он начинается в конкретный момент времени и завершается после достижения своих целей или прекращения по другим причинам;

- конечная дата может быть обусловлена сроками договора, рыночными условиями или внутренними требованиями организации;

Уникальность:

- каждый проект уникален, поскольку направлен на создание продукта, услуги или результата, отличающегося от существующих аналогов;

- уникальность может проявляться в новых процессах, технологиях, дизайне или других аспектах;

Целевое направление:

- проект создается для достижения конкретных целей и задач, которые определяются на этапе инициации;

- цели могут включать разработку нового продукта, улучшение существующего процесса, внедрение новой системы или другие специфические результаты;

Ограниченность ресурсов:

- проект всегда ограничен в ресурсах: времени, бюджете, человеческих и материальных ресурсах;

- эффективное управление ресурсами является критическим фактором успешного выполнения проекта;

Многообразие заинтересованных сторон:

- проекты часто затрагивают множество заинтересованных сторон, таких как заказчики, пользователи, команда проекта, поставщики и регулирующие органы;

- управление ожиданиями и коммуникациями с заинтересованными сторонами является важной частью проектного управления.

## Методологии и подходы к проектному управлению

Проектное управление включает в себя различные методологии и подходы, каждая из которых предлагает свои инструменты, процессы и принципы для эффективного управления проектами. Выбор методологии зависит от типа проекта, его сложности, требований и среды, в которой он реализуется. Рассмотрим подробнее основные методологии и подходы к проектному управлению.

1. Классический (традиционный) метод

Основан на четкой структуре проекта, которая включает фазы: инициация, планирование, выполнение, контроль и завершение. Этот метод предполагает детальное планирование и строгий контроль сроков и ресурсов.

2. Гибкие методологии (Agile)

Подходы Agile фокусируются на гибкости и адаптивности к изменениям. Проекты разделяются на короткие итерации (спринты), после которых оцениваются результаты и вносятся коррективы.

3. Scrum

Один из наиболее популярных методов Agile, Scrum предполагает работу небольших кросс-функциональных команд, которые работают в спринтах (обычно 2-4 недели) и проводят регулярные встречи для обсуждения прогресса и проблем.

4. Kanban

Kanban фокусируется на визуализации работы и управлении потоком задач через доску Kanban. Основная цель – обеспечить равномерную загрузку команды и минимизировать время выполнения задач.

5. Lean

Методология Lean направлена на минимизацию потерь и повышение ценности для клиента. Основной принцип – улучшение процессов и снижение ненужных затрат.

6. PRINCE2 (Projects IN Controlled Environments)

Структурированный метод проектного управления, который предоставляет четкие принципы, процессы и шаблоны для управления проектами. PRINCE2 широко используется в Европе.

7. Critical Path Method (CPM)

Метод критического пути используется для определения самых длительных последовательностей задач в проекте, которые определяют минимально возможное время выполнения проекта.

8. Program Evaluation and Review Technique (PERT)

Метод PERT основан на использовании статистики для оценки времени выполнения каждой задачи и определения вероятности завершения проекта в срок.

9. Six Sigma

Методология управления качеством, которая использует статистические методы для улучшения процессов и минимизации дефектов. В проектном управлении используется подход DMAIC (Define, Measure, Analyze, Improve, Control).

10. Earned Value Management (EVM)

Метод управления стоимостью и графиком проекта, который позволяет измерять проектные достижения и прогнозировать будущие результаты на основе текущих данных.

Эти методы могут комбинироваться и адаптироваться в зависимости от специфики проекта, требований клиента и особенностей команды.

## Участники проектом

Заказчик (Client or Sponsor):

Определение: Лицо или организация, финансирующая проект и определяющая его требования. Заказчик является основным заинтересованным лицом и играет ключевую роль в постановке целей и ожиданий от проекта.

Обязанности:

- определение целей и требований проекта.

- обеспечение финансирования и ресурсов.

- утверждение основных решений и документов проекта.

- оценка и приемка конечных результатов проекта.

Руководитель проекта (Project Manager):

Определение: Лицо, ответственное за управление проектом и достижение его целей в установленные сроки и в рамках бюджета. Руководитель проекта координирует работу команды и взаимодействует с заказчиком и другими заинтересованными сторонами.

Обязанности:

- разработка и выполнение плана проекта.

- управление рисками и изменениями.

- контроль за сроками, бюджетом и качеством.

- координация работы команды проекта.

- отчетность перед заказчиком и другими заинтересованными сторонами.

Команда проекта (Project Team):

Определение: Группа специалистов, выполняющих различные задачи проекта. Команда проекта может включать сотрудников различных отделов и внешних подрядчиков.

Обязанности:

- выполнение задач, определенных в плане проекта.

- соблюдение стандартов качества и сроков выполнения работ.

- взаимодействие с другими членами команды и руководителем проекта.

- участие в регулярных совещаниях и предоставление отчетности по выполнению задач.

Пользователи (Users):

Определение: Лица или группы, которые будут использовать конечный продукт или услугу, созданную в рамках проекта. Пользователи могут быть как внутренними (сотрудники компании), так и внешними (клиенты или клиенты клиентов).

Обязанности:

Предоставление требований и пожеланий к конечному продукту или услуге.

Участие в тестировании и оценке продукта.

Предоставление обратной связи для улучшения качества и функциональности.

# Техническое задание

## Основание для разработки

Основанием для разработки является задание на выпускную квалификационную работу бакалавра «Разработка системы управления малыми проектами и контроля задач».

## Цель и назначение разработки

Разрабатываемая программно-информационная система предназначена для совместного создания и управления веб-сайтами, не требуя от пользователей навыков программирования и веб-разработки.

Задачами данной разработки являются:

- разработка пользовательского интерфейса (административной панели) системы;

- разработка структуры базы данных для хранения информации о сущностях системы;

- разработка серверной части системы;

- реализация управления пользователями и правами доступа;

- реализация возможности совместной работы нескольких пользователей над контентом;

- организация хранилища электронных файлов.

## Требования к программной системы

### Требования к данным программной системы

Входными данными для программной системы являются:

- содержимое страниц и записей (постов): текст, цифровых файлов;

- категории и теги для организации записей (постов);

- метаданные для страниц: заголовок, описание, ключевые слова;

- пользовательские данные: регистрационная информация (имя пользователя, пароль);

- настройки и параметры: название, адрес сайта.

Выходными данными для программной системы будут:

- отображение данных в виде веб-страниц;

- вывод содержимого цифровых файлов на веб-страницах;

### Функциональные требования к программной системе

На основании анализа предметной области в разрабатываемой программной системе должны быть реализованы следующие функции:

1. Аутентификация и авторизация пользователей.

2. Управление веб-страницами.

3. Управление записями (проектами).

4.Управление меню и навигацией на сайте.

5. Управление пользователями.

6. Управление командами.

На рисунке 2.1 представлены функциональные требования к системе в виде диаграммы прецедентов нотации UML.

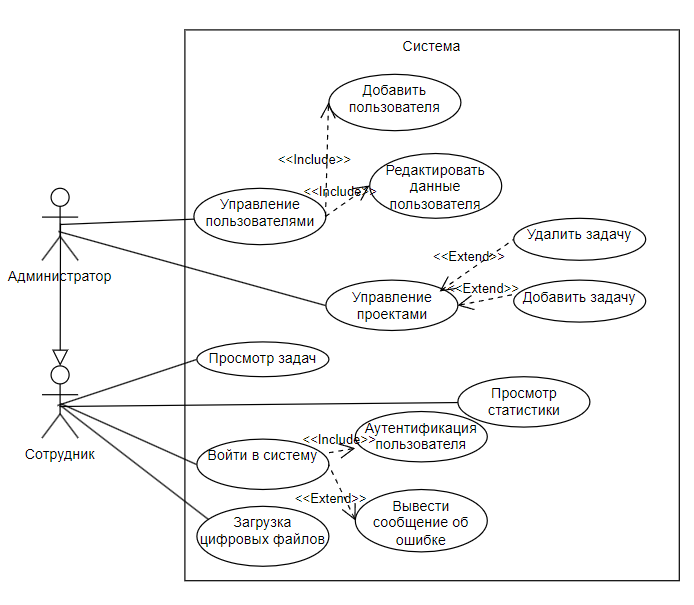


Рисунок 2.1 – Диаграмма прецедентов

### Моделирование вариантов использования

#### Варианты использования «Войти в систему»

Заинтересованные лица и их требования: пользователь хочет получить доступ к административной панели сайта.

Предусловие: пользователь зарегистрирован в системе и знает свои

данные для входа (логин и пароль).

Постусловие: система авторизует пользователя в соответствии с его ролью.

Основной успешный сценарий:

1. Пользователь вводит адрес административной панели сайта в браузере для перехода на страницу входа в систему.

2. Пользователь вводит логин и пароль.

3. Система проверяет корректность введенных данных и аутентифицирует пользователя.

4. Пользователь перенаправляется на главную страницу административной панели сайта.

#### Варианты использования «Управление пользователями»

Заинтересованные лица и их требования: администратор хочет управлять учетными записями пользователей и их ролями в системе.

Предусловие: пользователь авторизован и имеет права администратора.

Постусловие: изменения в учетных записях пользователей сохранены в системе.

Основной успешный сценарий:

1. Администратор открывает раздел управления пользователями.

2. Администратор выбирает действие (создание, редактирование, удаление пользователя).

3. Администратор вводит необходимую информацию о пользователе.

4. Система сохраняет изменения и обновляет список пользователей.

#### Варианты использования «Управления проектами»

Заинтересованные лица и их требования: пользователь хочет упралять постами.

Предусловие: пользователь авторизован в системе и имеет соответствующие права доступа.

Постусловие: изменения в проектах сохранены и отображаются на соответствующей странице сайта.

Основной успешный сценарий:

1. Пользователь открывает раздел управления проектами.

2. Пользователь создает новый проект или выбирает существующий для редактирования.

3. Пользователь вносит необходимые изменения в текст, заголовок, метаданные и т. д..

4. Пользователь сохраняет или публикует новый проект.

#### Варианты использования «Загрузка файлов»

Заинтересованные лица и их требования: пользователь хочет иметь возможность загрузки и удаления файлов.

Предусловие: пользователь авторизован в системе и имеет соответствующие права доступа.

Постусловие: файлов загружены и доступны для использования на сайте.

Основной успешный сценарий:

1. Пользователь открывает раздел управления файлами.

2. Пользователь нажимает соответствующую кнопку для загрузки файла и выбирает файл.

3. Система сохраняет файл в соответствующую папку на сервере.

### Требования пользователя к интерфейсу

Интерфейс пользователя административной панели разрабатываемой системы управления содержимым должен включать следующие компоненты:

1. Навигационное меню.

2. Интерфейс управления проектом (задачами).

3. Интерфейс управления меню и навигацией на сайте.

4. Интерфейс управления пользователями.

5. Интерфейс управления файлами.

Навигационное меню предназначено для перемещения по разделам и функциям.

Интерфейс управления проектом (задачами) отображает список записей, включает кнопки для создания, редактирования, удаления записей, поля для ввода данных при добавлении или редактировании записи.

Интерфейс управления медиафайлами отображает список загруженных файлов и включает кнопки для загрузки, просмотра, редактирования и удаления файлов.

Интерфейс управления пользователями отображает список пользователей системы и включает кнопки для создания, редактирования и удаления учетных записей пользователей.

Интерфейс управления меню и навигацией включает возможность управления навигационными меню и ссылками на страницы.

При реализации пользовательского интерфейса должны быть использованы следующие технологии:

- язык разметки веб-страниц HTML – для описания структуры страниц веб-интерфейса;

- каскадные таблицы стилей СSS – для стилизации элементов интерфейса;

## Нефункциональные требования к системе

### Требования к надёжности

В приложении не должно возникать критических ошибок, приводящих к экстренному завершению работы.

### Требования к программному обеспечению

Для реализации программной системы должены быть использованы следующие технологии:

- язык программирования PHP – для разработки серверной части;

- СУБД MySQL – для хранения данных и организации данных;

- веб-сервер Apache HTTP Server – для обработки клиентских запросов.

### Требования к аппаратному обеспечению

Для работы приложения, установленного на компьютер, необходимо дисковое пространство не менее 100 Мб, свободная оперативная память в размере не менее 1024 Мб, видеокарта с не менее 1024 Мб видеопамяти, клавиатура, мышь, установленная операционная система Windows, macOS или Linux, архитектура процессора х86 (Windows) или x64 (Windows, macOS, Linux). Для доступа к административной панели системы, потребуется браузер Google Chrome, Mozilla Firefox или Microsoft Edge.

### Требования к оформлению документации

Разработка программной документации и программного изделия должна производиться согласно ГОСТ 19.102-77 и ГОСТ 34.601-90. Единая система программной документации.

# Технический проект

# Общая характеристика организации решения задач

Необходимо спроектировать и разработать серверную и клиентскую части программно-информационной системы.

Система управления содержимым веб-сайта состоит из различных компонентов, предназначенных для управления, создания, редактирования и публикации контента на веб-сайте. Основные компоненты разрабатываемой:

1. Интерфейс управления (административная панель).

2. База данных.

3. Редактор контента.

При разработке программы должно быть уделено внимание следующим ключевым аспектам:

1. Простота использования.

2. Масштабируемость.

3. Безопасность.

4. Производительность.

# Обоснование выбора технологии проектирования

Выбранные для разработки программно-информационной системы языки и технологии предоставляют функции для создания эффективных и функциональных кроссбраузерных веб-приложений, позволяя создавать легко поддерживаемые и масштабируемые программные продукты.

# Описание используемых технологий и языков программирования

# Язык программирования PHP

PHP (Hypertext Preprocessor) – распространённый интерпретируемый язык общего назначения с открытым исходным кодом, который создавался специально для ведения веб-разработок, и код на нём встраивается непосредственно в HTML-код. Синтаксис языка берёт начало из языков C, Java и Perl и лёгок для изучения. Основная цель языка – помочь веб-разработчикам создавать динамически генерируемые веб-страницы.

Главная область применения PHP – написание скриптов, работающих на стороне сервера; таким образом, PHP способен выполнять все то, что выполняет любая другая программа CGI, например, обрабатывать данные форм, генерировать динамические страницы или отсылать и принимать cookies.

PHP отличается от JavaScript тем, что PHP-скрипты выполняются на сервере и генерируют HTML, который посылается клиенту.

Существуют три основных области применения PHP:

- cоздание скриптов для выполнения на стороне сервера;

- cоздание скриптов для выполнения в командной строке;

- cоздание оконных приложений, выполняющихся на стороне клиента. PHP доступен для большинства операционных систем, включая Linux, многие модификации Unix (такие как HP-UX, Solaris и OpenBSD), Microsoft Windows, macOS, RISC OS и многие другие. Также в PHP включена поддержка большинства современных веб-серверов, таких как Apache, IIS и многих других.

Таким образом, PHP предоставляет свободу выбора операционной системы и веб-сервера. Более того, появляется выбор между использованием процедурного или объектно-ориентированного программирования (ООП) или же их сочетания.

Использование PHP не ограничивается выводом HTML. Возможности PHP включают вывод файлов различных типов, таких как изображения или 20 PDF-файлы, шифрование данных и отправку электронной почты. Можно выводить любой текст, например JSON или XML. PHP может автоматически генерировать эти файлы и сохранять их в файловой системе вместо вывода на печать, формируя серверный кеш для динамического содержимого.

Одним из значительных преимуществ PHP является поддержка широкого круга баз данных. Можно воспользоваться модулем, специфичным для отдельной базы данных (таким как mysql) или использовать уровень абстракции от базы данных, такой как PDO, или подсоединиться к любой базе данных, поддерживающей Открытый Стандарт Соединения Баз Данных (ODBC), с помощью одноимённого модуля ODBC.

PHP также поддерживает взаимодействие с другими сервисами через такие протоколы, как LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (на платформах Windows) и многих других.

# Проектирование архитектурной программной системы

# Описание сущностей программной системы

Исходя из требований изложенных в техническом задании, можно выделить следующие основные сущности проектируемой системы:

- «Пользователь»;

- «Команды»;

- «Отчёты»;

- «Проекты»;

- «Календари»;

- «Задачи».

В состав сущности «Пользователь» можно включить атрибуты, представленные в таблице 3.1.

Таблица 3.1 – Атрибуты сущности «Пользователь»

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Обязательное | Описание |
| 1 | 2 | 3 | 4 |
| user\_id | Integer | True | Уникальный идентификатор |
| Name | String | True | Имя |
| Email | String | True | Адрес ел. почты |
| Role | String | True | Роль |
| password | String | true | Хэш пароля |

В состав сущности «Команды» можно включить атрибуты, представленные в таблице 3.2.

Таблица 3.2 – Атрибуты сущности «Команды»

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Обязательное | Описание |
| 1 | 2 | 3 | 4 |
| user\_id | Integer | True | Уникальный идентификатор |
| Project\_id | Integer | True | Уникальный идентификатор проекта |
| Id | Integer | True | Уникальный идентификатор соответствия |

В состав сущности «Задачи» можно включить атрибуты, представленные в таблице 3.3.

Таблица 3.3 – Атрибуты сущности «Задачи»

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Обязательное | Описание |
| 1 | 2 | 3 | 4 |
| task\_id | Integer | True | Уникальный идентификатор задачи |
| Project\_id | Integer | True | Уникальный идентификатор проекта |
| description | String | True | Описание задачи |
| priority | String | True | Приоритет |
| Due\_date | Datetime | true | Дата |

В состав сущности «Проекты» можно включить атрибуты, представленные в таблице 3.4.

Таблица 3.4 – Атрибуты сущности «Проекты»

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Обязательное | Описание |
| 1 | 2 | 3 | 4 |
| name | String | True | Имя проекта |
| Project\_id | Integer | True | Уникальный идентификатор проекта |
| description | String | True | Описание проекта |
| Start\_date | Datetime | True | Дата начала |
| End\_date | Datetime | true | Дата окончания |

В состав сущности «Календари» можно включить атрибуты, представленные в таблице 3.5.

Таблица 3.5 – Атрибуты сущности «Календари»

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Обязательное | Описание |
| 1 | 2 | 3 | 4 |
| Event\_id | String | True | Уникальный идентификатор проекта события |
| Project\_id | Integer | True | Уникальный идентификатор проекта |
| title | String | True | Заметка |
| event\_date | Datetime | True | Дата начала |
| description | String | true | Описание |

В состав сущности «Доклады» можно включить атрибуты, представленные в таблице 3.6.

Таблица 3.6 – Атрибуты сущности «Доклады»

|  |  |  |  |
| --- | --- | --- | --- |
| Поле | Тип | Обязательное | Описание |
| 1 | 2 | 3 | 4 |
| report\_id | String | True | Уникальный идентификатор доклада |
| Project\_id | Integer | True | Уникальный идентификатор проекта |
| title | String | True | Заметка |
| Created\_at | Datetime | True | Дата загрузки |
| content | String | true | Ссылка |

# Компоненты программной системы

Диаграмма компонентов используется для визуализаци программной системы, ее структурных компонентов и связей (зависимостей) между компонентами. Компоненты могут быть программными модулями, библиотеками, пакетами и другими элементами, которые реализуют определенные функции системы. На рисунке 3.1 изображена диаграмма компонентов проектируемой системы.

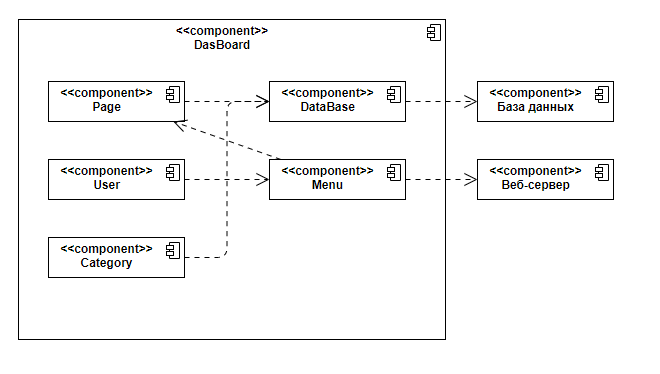


Рисунок 3.1 – Диаграмма компонентов

Разрабатываемая программно-информационная система состоит из следующих основных компонентов:

1. Frontend (Пользовательский интерфейс) – включает файлы, которые описывают пользовательский интерфейс административной панели, а также папку с файлами которые определяют внешний вид и структуру веб-страниц сайта, включая HTML, CSS и JavaScript.

2. Backend (Серверная часть) включет следующие компоненты:

- Controllers (Контроллеры) – обрабатывает входящие HTTP-запросы клиента, вызывают методы моделей и определяют соответствующие представления для отображения данных;

- Models (Модели) – управляют данными и бизнес логикой, взаимодействуют с базой данных для получения, обновления и удаления данных;

- Views (Отображения) – отвечают за формирование HTML-страниц на основе данных полученных из моделей.

3. База данных – хранит структурированные данные в виде записей в таблицах, каждая таблица представляет определенную сущность системы.

4. Веб-сервер – программное обеспечение, которое принимает HTTPзапросы от клиентов и отвечает на них, предоставляя нужные ресурсы, такие как HTML-страницы, изображения, видео и другие данные.

Описание компонентов программной системы:

1. Page – отвечает за создание и управление статическими страницами и организацию разделов сайта.

2. Category – используется для организации контента на сайте, содержит функции для структурирования постов по темам или разделам.

3. User – этот компонент управляет информацией о пользователях сайта. Включает создание, редактирование и удаление учетных записей, управление ролями и правами доступа.

4. Database – обеспечивает создание и управление данными которые храняться в таблицах БД, отвечает за выполнение запросов к базе данных для получения, добавления, обновления и удаления информации.

5. Menu – обеспечивает навигацию по сайту. Этот компонент позволяет создавать и управлять навигационными меню, которые могут содержать ссылки на страницы, проекты, задачи и другие элементы сайта.

# Проектирование пользовательского интерфейса

На основании требований к пользовательскому интерфейсу представленных в пункте 2.3.4 технического задания, был разработан интерфейс административной панели системы и интерфейс пользователя. Для создания пользовательского интерфейса используется язык разметки HTML и веб-фреймворк Bootstrap

На рисунке 3.3 представлен макет главного окна административной панели. Макет содержит следующие элементы:

1. Меню для отслеживания задач подчинённых.
2. Форма для редактирования задач.
3. Кнопка «Добавить задачу»



Рисунок 3.3 – Макет главного окна административной панели.

На рисунке 3.4 представлен макет окна административной панели. Макет содержит следующие элементы:

1. Меню для отслеживания задач сотрудником.
2. Панель управления для навигации между элементами отображения.

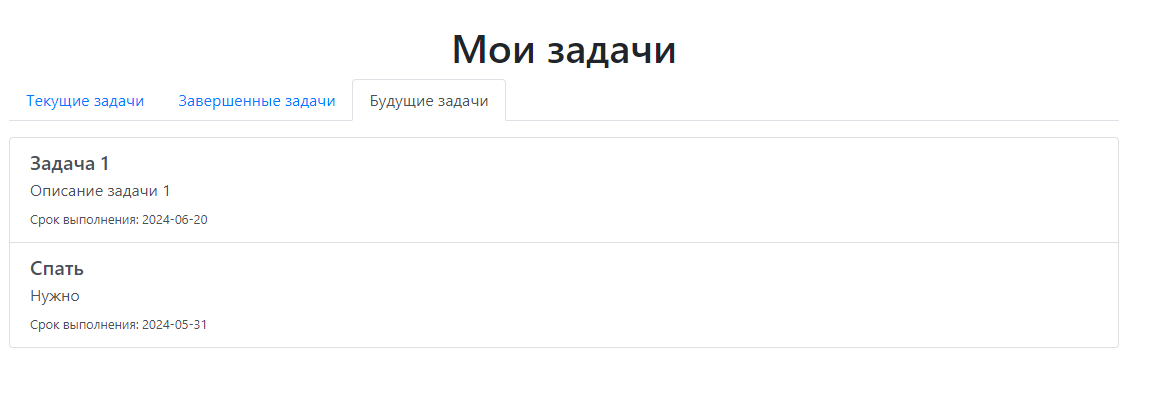


Рисунок 3.4 – Макет окна пользовательской панели.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

* + 1. Бретт, М. PHP и MySQL. Исчерпывающее руководство / М. Бретт. – Санкт-Петербург : Питер, 2016. – 544 с. – ISBN 978-5-496-01049-8. – Текст : непосредственный. Безкоровайный М.М., Костогрызов А.И., Львов В.М. Инструментально–моделирующий комплекс для оценки качества функционирования информационных систем «КОК». Руководство системного аналитика. М.: Синтег, 2021.
    2. Веру, Л. Секреты CSS. Идеальные решения ежедневных задач / Л. Веру. – Санкт-Петербург : Питер, 2016. – 336 с. – ISBN 978-5-496-02082-4. – Текст : непосредственный.
    3. Фримен, А. Практикум по программированию на JavaScript / А. Фримен. – Москва : Вильямс, 2013. – 960 с. – ISBN 978-5-8459-1799-7. – Текст : непосредственный.
    4. Гизберт, Д. PHP и MySQL / Д. Гизберт. – Москва : НТ Пресс, 2013. – 320 с. – ISBN 978-5-477-01174-2. – Текст : непосредственный.Гуленко В.В. Формы мышления. // Соционика, ментология и

психология личности, N 4, 2022.

* + 1. Лоусон, Б. Изучаем HTML5. Библиотека специалиста / Б. Лоусон, Р. Шарп. – Санкт-Петербург : Питер, 2013 – 286 с. – ISBN 978-5-459-01156-2. – Текст : непосредственный. Дал У., Мюрхауг Б., Нюгород К. СИМУЛА–67. Универсальный язык программирования. М.: Мир, 2020.
    2. Макфарланд, Д. Большая книга CSS / Д. Макфарланд. – СанктПетербург : Питер, 2012. – 560 с. – ISBN 978-5-496-02080-0. – Текст : непосредственный. Дьяконов В. Mathematica 4: учебный курс. СПб: Питер, 2022.
    3. Емельянов С.В, Коровин С.К. Новые типы обратной связи. М.: Наука, 2021.
    4. Дэкетт, Д. HTML и CSS. Разработка и создание веб-сайтов / Д. Дэкетт. – Москва : Эксмо, 2014. – 480 с. – ISBN 978-5-699-64193-2. – Текст : непосредственный. Калашникова Т.Г. Исследование и разработка методов и моделей правдоподобных рассуждений в интеллектуальных системах поддержки

принятия решений. Диссертация на соискание ученой степени кандидата технических наук. Таганрог, 2021.

* + 1. Титтел, Э. HTML5 и CSS3 для чайников / Э. Титтел, К. Минник. – Москва : Вильямс, 2016 – 400 с. – ISBN 978-1-118-65720-1. – Текст : непосредственный. Карташев М.И. Адаптивный тестовый контроль в системах дистанционного образования / Никитин М.М, Строганов В.Ю., Карташев М.И.

// В мире научных открытий № 9 (21). Красноярск: НИЦ, 2021.

* + 1. Карташев М.И. Использование нечетких множеств при определении количественных оценок связности учебного материала / Николаев А.Б.