* 1. **Постановка задачи**

**1 Организационно-экономическая сущность задачи:**

**Наименование задачи:**

Разработка Интернет-ресурса для студии йоги «Blissful Yoga»

**Цель разработки:**

Создание удобного, функционального и интерактивного веб-сайта, который позволит пользователям записываться на занятия, просматривать расписание, выбирать инструкторов, покупать абонементы и медитационные сессии, а также бронировать залы для самостоятельных тренировок.

**Назначение:**

Для студий йоги, тренеров по йоге и пользователей, интересующихся йогой и медитацией.

**Периодичность использования ПП:**

Постоянное использование в любое время.

**Источники и способы получения данных:**

* Ввод данных администраторами и тренерами студии (расписание, информация о тренерах, занятия и абонементы).
* Обратная связь от пользователей (оценки и отзывы).

**Информационная связь с другими задачами:**

* Система управления клиентами (CRM).
* Система онлайн-платежей.
* Электронная почта и SMS-уведомления для напоминаний о занятиях и подтверждений бронирования.

**Обзор существующих аналогичных ПП:**

* MINDBODY: Платформа для бронирования занятий и управления студиями йоги.
* FitMetrix: Решение для создания расписаний и управления занятиями.
* Zen Planner: Инструмент для управления фитнес-студиями и тренировочными залами.

**2 Функциональные требования:**

**Гость**

1. **Просмотр расписания занятий:**
   * Гость должен иметь возможность просматривать расписание занятий без необходимости регистрации.
2. **Просмотр профилей тренеров:**
   * Гость должен иметь доступ к профилям тренеров, включая их биографии, фотографии и отзывы.
3. **Регистрация/Авторизация:**
   * Гость должен иметь возможность зарегистрироваться или авторизоваться на сайте для получения дополнительных возможностей.

**Администратор**

1. **Управление расписанием занятий:**

* Администратор должен иметь возможность управлять общим расписанием занятий студии, включая добавление, изменение и удаление занятий.

1. **Управление профилями инструкторов:**

* Администратор должен иметь возможность добавлять, изменять и удалять профили инструкторов.

1. **Управление абонементами и ценами:**

* Администратор должен иметь возможность устанавливать и изменять цены на абонементы и отдельные занятия.

1. **Обработка обратной связи и отзывов:**

* Администратор должен иметь возможность просматривать и отвечать на отзывы и обратную связь от пользователей.

1. **Управление бронированием залов:**

* Администратор должен иметь возможность управлять бронированием залов для самостоятельных тренировок.

**3 Описание исходной (входной) информации:**

**Перечень исходной информации:**

* Данные о пользователях (имя, контакты).
* Информация о тренерах (биография, фото, квалификации).
* Расписание занятий и медитационных сессий.
* Информация об абонементах и стоимости занятий.
* Данные о залах для бронирования.

**Формы представления:**

* Электронные формы и таблицы для заполнения администраторами.
* Веб-формы для пользователей.

**Перечень пользователей исходной информации:**

* Администраторы студии.
* Тренеры.
* Пользователи сайта.

**4 Описание результатной (выходной) информации:**

**Перечень результатной информации:**

* Подтверждения записей на занятия.
* Напоминания о предстоящих занятиях.
* Информация об активных абонементах.
* Статистика посещений и занятий.

**Формы представления:**

* Печатная сводка.
* Электронные уведомления (email, SMS).
* Веб-интерфейс личного кабинета.

**Периодичность и сроки представления:**

* В режиме реального времени (записи, бронирования).
* Ежедневно (напоминания о занятиях).

**Перечень пользователей результатной информации:**

* Пользователи сайта.
* Администраторы студии.
* Тренеры.

**5 Описание используемой условно-постоянной информации:**

**Перечень условно-постоянной информации:**

* Справочник тренеров.
* Классификатор занятий и медитационных сессий.
* Таблицы с расписанием занятий.
* Справочник абонементов.

**Формы представления:**

* Электронные таблицы и базы данных.

**6 Нефункциональные (эксплуатационные) требования:**

**Требования к применению:**

* Удобный и интуитивно понятный интерфейс пользователя.
* Подробная документация для пользователей и администраторов.
* Учебные курсы по использованию функционала сайта.

**Требования к производительности:**

* Высокая скорость загрузки страниц и обработки запросов.
* Высокая пропускная способность для большого числа пользователей.

**Требования к реализации:**

* Использование современных веб-технологий (HTML5, CSS3, JavaScript).
* Поддержка работы на различных устройствах (мобильные, планшеты, ПК).
* Интеграция с популярными системами онлайн-платежей.

**Требования к надежности:**

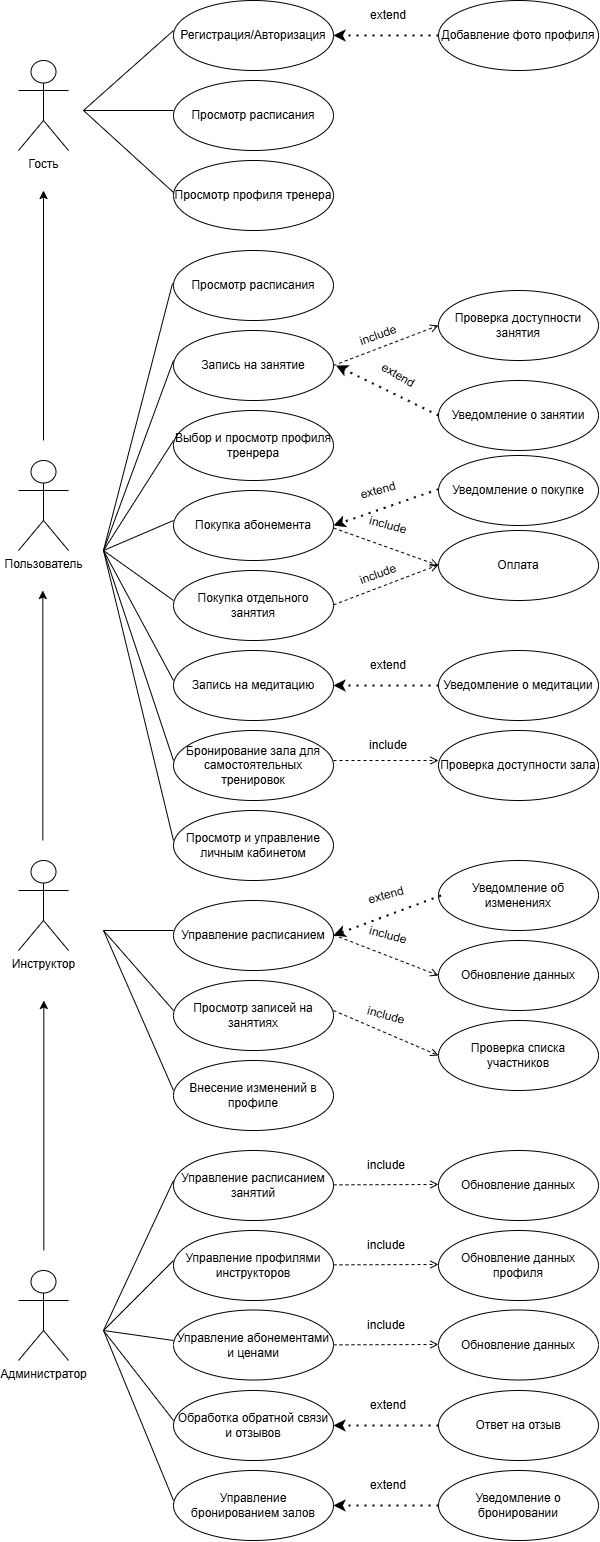
* Высокая устойчивость к сбоям.
* Возможности восстановления данных после сбоев.
* Регулярное резервное копирование данных.

**Требования к интерфейсу:**

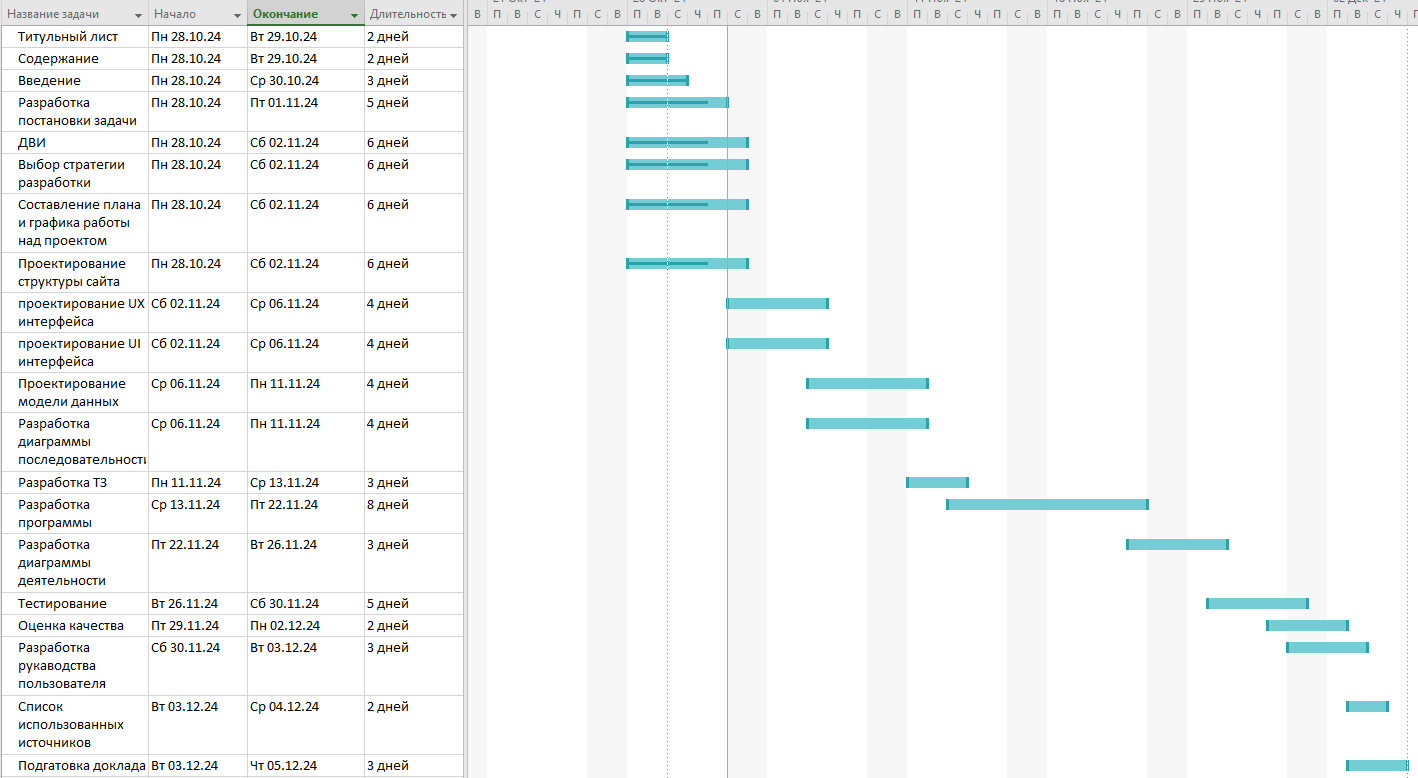
* Взаимодействие с пользователями через веб-интерфейс.
* Поддержка различных внешних устройств (компьютеры, мобильные телефоны).
* Возможность интеграции с внешними системами уведомлений и платежей.

Таким образом,

* 1. **Диаграмма вариантов использования**



* 1. **Разработка плана работы над проектом**

****

* 1. **Выбор стратегии разработки и модели жизненного цикла**

Для разработки интернет-ресурса для студии йоги «Blissful Yoga» следует выбрать стратегию разработки и модель жизненного цикла. Осуществляем выбор посредством составления таблиц:

Таблица 3 **–** Выбор модели жизненного цикла на основе характеристик требований

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № критерия | Критерии категории требований | Каскадная | V-образная | RAD | Инкрементная | Быстрого прототипирования | Эволюционная |
| 1. | Являются ли требования к проекту легко определимыми и реализуемыми? | Да | Да | Да | Нет | Нет | Нет |
| 2. | Могут ли требования быть сформулированы в начале ЖЦ? | Да | Да | Да | Да | Нет | Нет |
| 3. | Часто ли будут изменяться требования на протяжении ЖЦ? | Нет | Нет | Нет | Нет | Да | Да |
| 4. | Нужно ли демонстрировать требования с целью их определения? | Нет | Нет | Да | Нет | Да | Да |
| 5. | Требуется ли проверка концепции программного средства или системы? | Нет | Нет | Да | Нет | Да | Да |
| 6. | Будут ли требования изменяться или уточняться с ростом сложности системы (программного средства) в ЖЦ? | Нет | Нет | Нет | Да | Да | Да |
| 7. | Нужно ли реализовать основные требования на ранних этапах разработки? | Нет | Нет | Да | Да | Да | Да |

Вычисления: 4 за каскадную, 4 за V- образную, 5 за RAD, 5 за инкрементную, 3 за быстрого прототипирования и 3 за эволюционную.

Итог: На основе результатов заполнения табл. 3 подходящей является RAD модель и инкрементная модель.

Таблица 4 **–** Выбор модели жизненного цикла на основе характеристик команды разработчиков

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № критерия | Критерии категории команды разработчиков  проекта | Каскадная | V-образная | RAD | Инкрементная | Быстрого прототипирования | Эволюционная |
| 1. | Являются ли проблемы предметной области проекта новыми для большинства разработчиков? | Нет | Нет | Нет | Нет | Да | Да |
| 2. | Являются ли инструментальные средства, используемые в проекте, новыми для большинства разработчиков? | Да | Да | Нет | Нет | Нет | Да |
| 3. | Изменяются ли роли участников проекта на протяжении ЖЦ? | Нет | Нет | Нет | Да | Да | Да |
| 4. | Является ли структура процесса разработки более значимой для разработчиков, чем гибкость? | Да | Да | Нет | Да | Нет | Нет |
| 5. | Важна ли легкость распределения человеческих ресурсов проекта? | Да | Да | Да | Да | Нет | Нет |
| 6. | Приемлет ли команда разработчиков оценки, проверки, стадии разработки? | Да | Да | Нет | Да | Да | Да |

Вычисления: 5 за каскадную, 5 за V-образную, 4 за RAD, 5 за инкрементную, 2 за быстрого прототипирования и 1 за эволюционную.

Итог: На основе результатов заполнения табл. 4 подходящими являются каскадная, V-образная и инкрементная модели.

Таблица 5 **–** Выбор модели жизненного цикла на основе характеристик коллектива пользователей

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № критерия | Критерии категории коллектива пользователей | Каскадная | V-образная | RAD | Инкрементная | Быстрого прототипирования | Эволюционная |
| 1. | Будет ли присутствие пользователей ограничено в ЖЦ разработки? | Да | Да | Нет | Да | Нет | Да |
| 2. | Будут ли пользователи оценивать текущее состояние программного продукта (системы) в процессе разработки? | Нет | Нет | Нет | Да | Да | Да |
| 3. | Будут ли пользователи вовлечены во все фазы ЖЦ разработки? | Нет | Нет | Да | Нет | Да | Нет |
| 4. | Будет ли заказчик отслеживать ход выполнения проекта? | Нет | Нет | Нет | Нет | Да | Да |

Вычисления: 1 за каскадную, 1 за V-образную, 1 за RAD, 2 за инкрементную, 3 за быстрого прототипирования и 3 за эволюционную.

Итог: На основе результатов заполнения табл. 5 подходящей является модель быстрого проектирования и Эволюционная.

Таблица 6 **–** Выбор модели жизненного цикла на основе характеристик типа проектов и рисков

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № критерия | Критерии категории типов проекта и рисков | Каскадная | V-образная | RAD | Инкрементная | Быстрого прототипирования | Эволюционная |
| 1. | Разрабатывается ли в проекте продукт нового для организации направления? | Нет | Нет | Нет | Да | Да | Да |
| 2. | Будет ли проект являться расширением существующей системы? | Да | Да | Да | Да | Нет | Нет |
| 3. | Будет ли проект крупно- или среднемасштабным? | Нет | Нет | Нет | Да | Да | Да |
| 4. | Ожидается ли длительная эксплуатация продукта? | Да | Да | Нет | Да | Нет | Да |
| 5. | Необходим ли высокий уровень надежности продукта проекта? | Нет | Да | Нет | Да | Нет | Да |
| 6. | Предполагается ли эволюция продукта проекта в течение ЖЦ? | Нет | Нет | Нет | Да | Да | Да |
| 7. | Велика ли вероятность изменения системы (продукта) на этапе сопровождения? | Нет | Нет | Нет | Да | Да | Да |
| 8. | Является ли график сжатым? | Нет | Нет | Да | Да | Да | Да |
| 9. | Предполагается ли повторное использование компонентов? | Нет | Нет | Да | Да | Да | Да |
| 10. | Являются ли достаточными ресурсы (время, деньги, инструменты, персонал)? | Нет | Нет | Нет | Нет | Да | Да |

Вычисления: 2 за каскадную, 3 за V-образную, 3 за RAD, 7 за инкрементную, 7 за быстрого прототипирования и 9 за эволюционную.

Итог: На основе результатов заполнения табл. 6 подходящей является эволюционная модели.

Общий итог: в итоге заполнения табл. 3 – 6 , 12 – за каскадную, 13 – за V-образную, 13 – за RAD, 19 – за инкрементную, 15 – за быстрого прототипирования, 16 – за эволюционную. Наиболее подходящей является инкрементная модель.

* 1. **Инструменты разработки**

**Конструктор WordPress**

WordPress — это одна из самых популярных платформ для создания веб-сайтов, которая предоставляет гибкие возможности для создания и управления контентом.

**Основные преимущества WordPress:**

1. **Простота использования:** Интуитивно понятная панель администрирования, не требующая глубоких знаний программирования.
2. **Шаблоны и плагины:** Большое количество готовых шаблонов и плагинов для расширения функциональности сайта.
3. **SEO дружелюбность:** Встроенные возможности для оптимизации сайта под поисковые системы.
4. **Мобильная адаптация:** Большинство шаблонов адаптированы под мобильные устройства.
5. **Интеграции:** Поддержка интеграций с различными сервисами (например, онлайн-платежи, социальные сети, аналитика).

**Основные функции для сайта йоги и медитации:**

* **Создание расписания занятий:** Плагины для создания и управления расписанием.
* **Запись на занятия:** Формы записи и бронирования через плагины.
* **Страницы с профилями тренеров:** Шаблоны и плагины для создания профилей тренеров с биографиями и отзывами.
* **Интеграция с платежными системами:** Плагины для онлайн-оплат абонементов и отдельных занятий.
* **Блоги и новости:** Встроенные возможности для ведения блогов и публикации новостей.
* **Раздел с отзывами:** Плагины для добавления и управления отзывами клиентов.

**Разработка на C#**

Для более индивидуальной разработки сайта с использованием C# потребуется веб-фреймворк, такой как [**ASP.NET**](https://asp.net/) Core. Этот подход позволяет создавать сложные и кастомизированные веб-приложения с высокой производительностью.

**Основные преимущества использования C# и** [**ASP.NET**](https://asp.net/) **Core:**

1. **Высокая производительность:** [**ASP.NET**](https://asp.net/) Core является одним из самых быстрых фреймворков для веб-разработки.
2. **Гибкость:** Возможность создания сложных функциональных решений, полностью адаптированных под нужды клиента.
3. **Безопасность:** Встроенные средства защиты, поддержка аутентификации и авторизации.
4. **Поддержка различных СУБД:** Легкая интеграция с различными системами управления базами данных (SQL Server, MySQL, PostgreSQL и т.д.).
5. **Интеграция с облачными сервисами:** Поддержка интеграции с Azure и другими облачными провайдерами.

**Основные функции для сайта йоги и медитации:**

* **Пользовательская аутентификация и авторизация:** Создание и управление учетными записями пользователей.
* **Просмотр расписания и запись на занятия:** Интерактивный календарь с возможностью записи на занятия.
* **Профили тренеров:** Страницы с информацией о тренерах, их квалификацией и отзывами.
* **Онлайн-платежи:** Интеграция с платежными системами для покупки абонементов и отдельных занятий.
* **Бронирование залов:** Возможность бронирования залов для самостоятельных тренировок.
* **Личный кабинет:** Управление личными данными, записями и абонементами.
* **SEO-оптимизация:** Инструменты для оптимизации сайта под поисковые системы.

**Заключение**

Выбор между WordPress и разработкой на C# зависит от ваших конкретных потребностей и возможностей:

* **WordPress** подходит для быстрого запуска сайта с минимальными затратами и без необходимости программирования.
* **C# и** [**ASP.NET**](https://asp.net/) **Core** подходят для создания индивидуальных и сложных веб-приложений с высокими требованиями к производительности и безопасности.