**Содержание**

[1. Постановка задачи 3](#_Toc201919831)

[1.1 Функциональные требования 3](#_Toc201919832)

[1.2 Эксплуатационные требования 4](#_Toc201919833)

[2 Проектирование 5](#_Toc201919834)

[2.1 Выбор стратегии разработки и модели жизненного цикла 5](#_Toc201919835)

[2.2 Диаграмма вариантов использования 10](#_Toc201919836)

[2.3 Проектирование структуры сайта 11](#_Toc201919837)

[2.4 Моделирование бизнес-проектов 12](#_Toc201919838)

[2.5 Моделирование данных 14](#_Toc201919839)

[2.6 Диаграмма последовательности 15](#_Toc201919840)

[2.7 Диаграмма деятельности 16](#_Toc201919841)

[2.8 Диаграмма классов 16](#_Toc201919842)

[2.9 Диаграмма объектов 18](#_Toc201919843)

# **Постановка задачи**

Организационно-экономическая сущность задачи:

Наименование задачи: «PetHelper»

Цель разработки: создание сайта для облегчения жизни с домашними животными и помощи для тех, кто собирается завести домашнее животное.

Назначение: сайт «PetHelper» разрабатывается с целью облегчения процесса задачи поиска ветеринарных клиник, аптек, зоомагазинов, мест стрижки животных для хозяев, а также приютов, для тех, кто только хочет завести животное или занимается волонтерством.

Периодичность использования: по необходимости

Источники и способы получения данных: сеть интернет

* 1. **Функциональные требования**

Описание перечня функций и задач, которые должен выполнять будущий ПП:

Гость:

* просмотр главной страницы.
* регистрация на сайте.
* переход по ссылкам.
* просмотр отзывов.
* просмотр новостей.
* просмотр информации об уходе за животными.
* просмотр информации о местоположении мест стрижки животных, приютов, аптек, зоомагазинов, ветеринарных клиник.
* просмотр информации и местоположения волонтерских организаций в городе.

Пользователь:

* авторизация на сайте.
* доступ к просмотру услуг.
* просмотр новостей
* обратная связь.
* информирование об обновлениях.
* уведомления
* выход из личного кабинета.

Администратор:

* авторизация на сайте;
* ответы на обратную связь;
* добавление/удаление новостей, информации.
  1. **Эксплуатационные требования**

Требования к реализации: для реализации статических страниц и шаблонов

должны использоваться языки HTML и CSS. Для реализации интерактивных элементов клиентской части должны использоваться языки JavaScript.

Требования к надежности: Система может быть недоступна не более чем 24 часа в год. У администратора приложения должна быть возможность выгрузить и загрузить копию сайта.

Требования к интерфейсу: при разработке сайта должны быть использованы преимущественного бежево-коричневые оттенки. Основные разделы сайта должны быть доступны с первой страницы. Грамотный пользовательский интерфейс. Сайт должен адаптироваться под телефон и планшет.

**2 Проектирование**

## **2.1 Выбор стратегии разработки и модели жизненного цикла**

Таблица 1 **–** Выбор модели жизненного цикла на основе характеристик требований

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Критерии категории требований | Каскадная | V-образная | RAD | Инкрементная | Быстрого прототипирования | Эволюционная |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Являются ли требования к проекту легко определимыми и реализуемыми? | Да | Да | Да | Нет | Нет | Нет |
| Могут ли требования быть сформулированы в начале ЖЦ? | Да | Да | Да | Да | Нет | Нет |
| Часто ли будут изменяться требования на протяжении ЖЦ? | Нет | Нет | Нет | Нет | Да | Да |
| Нужно ли демонстрировать требования с целью их определения? | Нет | Нет | Да | Нет | Да | Да |
| Требуется ли проверка концепции программного средства или системы? | Нет | Нет | Да | Нет | Да | Да |
| Будут ли требования изменяться или уточняться с ростом сложности системы (программного средства) в ЖЦ? | Нет | Нет | Нет | Да | Да | Да |
| Нужно ли реализовать основные требования на ранних этапах разработки? | Нет | Нет | Да | Да | Да | Да |
|  | 3 | 3 | 4 | 4 | 4 | 4 |

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

Итог: На основе результатов заполнения табл. 3 подходящими являются RAD модель, инкрементная модель, быстрого прототипирования и эволюционная.

Таблица 2 **–** Выбор модели жизненного цикла на основе характеристик команды  
разработчиков

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Критерии категории команды разработчиков проекта | Каскадная | V-образная | RAD | Инкрементная | Быстрого прототипирования | Эволюционная |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Являются ли проблемы предметной области проекта новыми для большинства разработчиков? | Нет | Нет | Нет | Нет | Да | Да |
| Являются ли инструментальные средства, используемые в проекте, новыми для большинства разработчиков? | Да | Да | Нет | Нет | Нет | Да |
| Является ли структура процесса разработки более значимой для разработчиков, чем гибкость? | Да | Да | Нет | Да | Нет | Нет |
| Изменяются ли роли участников проекта на протяжении ЖЦ? | Нет | Нет | Нет | Да | Да | Да |
| Важна ли легкость распределения человеческих ресурсов проекта? | Да | Да | Да | Да | Нет | Нет |
| Приемлет ли команда разработчиков оценки, проверки, стадии разработки? | Да | Да | Нет | Да | Да | Да |
|  | 2 | 2 | 1 | 2 | 5 | 6 |

Вычисления: 2 за каскадную, 2 за V-образную, 1 за RAD, 2 за инкрементную, 5 за быстрого прототипирования и 6 за эволюционную.

Итог: На основе результатов заполнения табл. 4 подходящей является эволюционная модель.

Таблица 3 **–** Выбор модели жизненного цикла на основе характеристик коллектива пользователей

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Критерии категории коллектива пользователей | Каскадная | V-образная | RAD | Инкрементная | Быстрого прототипирования | Эволюционная |
| Будет ли присутствие пользователей ограничено в ЖЦ разработки? | Да | Да | Нет | Да | Нет | Да |
| Будут ли пользователи оценивать текущее состояние программного продукта (системы) в процессе разработки? | Нет | Нет | Нет | Да | Да | Да |
| Будут ли пользователи вовлечены во все фазы ЖЦ разработки? | Нет | Нет | Да | Нет | Да | Нет |
| Будет ли заказчик отслеживать ход выполнения проекта? | Нет | Нет | Нет | Нет | Да | Да |
|  | 2 | 2 | 2 | 1 | 2 | 2 |

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

Итог: На основе результатов заполнения табл. 5 подходящими являются каскадная, V-образная, RAD, быстрого прототипирования и эволюционная модели.

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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Критерии категории типов проекта и рисков | Каскадная | V-образная | RAD | Инкрементная | Быстрого прототипирования | Эволюционная |
| Разрабатывается ли в проекте продукт нового для организации направления? | Нет | Нет | Нет | Да | Да | Да |
| Будет ли проект являться расширением существующей системы? | Да | Да | Да | Да | Нет | Нет |
| Будет ли проект крупно- или среднемасштабным? | Нет | Нет | Нет | Да | Да | Да |
| Ожидается ли длительная эксплуатация продукта? | Да | Да | Нет | Да | Нет | Да |
| Необходим ли высокий уровень надежности продукта проекта? | Нет | Да | Нет | Да | Нет | Да |
| Предполагается ли эволюция продукта проекта в течение ЖЦ? | Нет | Нет | Нет | Да | Да | Да |
| Велика ли вероятность изменения системы (продукта) на этапе сопровождения? | Нет | Нет | Нет | Да | Да | Да |
| Является ли график сжатым? | Нет | Нет | Да | Да | Да | Да |
| Предполагается ли повторное использование компонентов? | Нет | Нет | Да | Да | Да | Да |
| Являются ли достаточными ресурсы (время, деньги, инструменты, персонал)? | Нет | Нет | Нет | Нет | Да | Да |
|  | 3 | 2 | 3 | 6 | 8 | 6 |

Вычисления: 3 за каскадную, 2 за V-образную, 3 за RAD, 6 за инкрементную, 8 за быстрого прототипирования и 6 за эволюционную.

Итог: На основе результатов заполнения табл. 6 подходящей является каскадная модель.

Общий итог: Каскадная: 10; V-образная: 9; RAD: 10; Инкрементная: 13; Быстрого прототипирования: 19; Эволюционная: 18.

в итоге заполнения табл. 3 – 6 наиболее подходящей является модель быстрого прототипирования.

## **2.2 Диаграмма вариантов использования**

На рисунке 1 представлена диаграмма вариантов использования

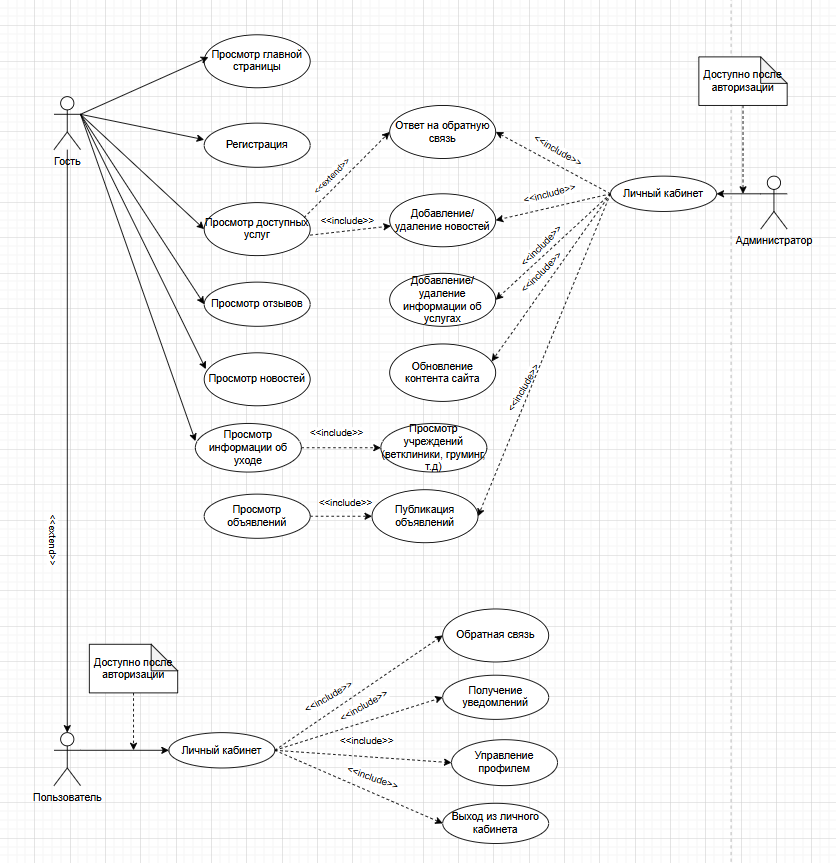
****

Рисунок 1 – Диаграмма вариантов использования

## **2.3 Проектирование структуры сайта**

Диаграмма отображает иерархическую структуру веб-сайта, включая основные разделы, подразделы и связи между ними. Позволяет визуализировать навигацию и логику взаимодействия пользователя с контентом. На рисунке 2 представлена структура сайта.

**Изображение выглядит как диаграмма, линия, снимок экрана, План

Содержимое, созданное искусственным интеллектом, может быть неверным.**

Рисунок 2 – Структура сайта

## **2.4 Моделирование бизнес-проектов**

Диаграмма демонстрирует ключевые бизнес-процессы системы, их этапы, участников и потоки данных. На рисунках 3 и 4 представлено моделирование бизнес-проектов.

**Изображение выглядит как текст, диаграмма, План, Технический чертеж

Содержимое, созданное искусственным интеллектом, может быть неверным.**

Рисунок 3 – Функциональный блок и интерфейсные дуги

Изображение выглядит как текст, диаграмма, План, Технический чертеж

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 4 – Структура SADT-модели

**2.5 Моделирование данных**

**Изображение выглядит как текст, диаграмма, линия, снимок экрана

Содержимое, созданное искусственным интеллектом, может быть неверным.**ЕR-диаграмма (или аналогичная) показывает сущности базы данных, их атрибуты и связи. Используется для проектирования эффективной и нормализованной структуры хранения информации. На рисунке 5 представлен процесс моделирования данных.

Рисунок 5 – Моделирование данных

**2.6 Диаграмма последовательности**

UML-диаграмма, иллюстрирующая взаимодействие объектов системы в рамках конкретного сценария с учетом временной последовательности. Полезна для анализа логики работы функций. На рисунке 5 представлена диаграмма последовательности.

**Изображение выглядит как текст, диаграмма, Параллельный, линия

Содержимое, созданное искусственным интеллектом, может быть неверным.**

Рисунок 5 – Диаграмма последовательности

## **2.7 Диаграмма деятельности**

## 

UML-диаграмма, описывающая алгоритм выполнения процесса (например, пользовательского сценария). Включает ветвления, параллельные действия и точки принятия решений. На рисунке 7 представлена диаграмма деятельности.

Изображение выглядит как текст, снимок экрана

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 7 – Диаграмма деятельности

## **2.8 Диаграмма классов**

UML-диаграмма, отображающая статическую структуру системы в виде классов, их атрибутов, методов и взаимосвязей (ассоциации, наследование, агрегация и др.). Используется для проектирования объектно-ориентированной архитектуры и документирования ключевых сущностей системы. На рисунке 8 представлена диаграмма классов.

**Изображение выглядит как текст, диаграмма, План, Параллельный

Содержимое, созданное искусственным интеллектом, может быть неверным.**

Рисунок 8 – Диаграмма классов

## **2.9 Диаграмма объектов**

UML-диаграмма, показывающая конкретные экземпляры классов (объекты) и их связи в определённый момент времени. Используется для иллюстрации состояния системы на примере реальных данных, часто — для тестирования или уточнения диаграммы классов. На рисунке 9 представлена диаграмма объектов.

**Изображение выглядит как диаграмма, линия, текст, Параллельный

Содержимое, созданное искусственным интеллектом, может быть неверным.**

Рисунок 9 – Диаграмма объектов