|  |  |
| --- | --- |
|  | **ДЕПАРТАМЕНТ ОБРАЗОВАНИЯ И НАУКИ ГОРОДА МОСКВЫ**  **Государственное бюджетное профессиональное  образовательное учреждение города Москвы**  **«Московский колледж архитектуры и градостроительства»  (ГБПОУ «МКАГ»)** |

**Отчёт**

**на тему: «Фрилансерский поиск»**

Выполнили:

Тарасюк Максим

Стариков Даниил

Москва 2025 г.

**Оглавление**

Оглавление

[Формулировка идеи приложения 3](#_Toc192835645)

[Идея веб-приложения 3](#_Toc192835646)

[Использование гибких методов разработки 3](#_Toc192835647)

[Итерации (спринты) 3](#_Toc192835648)

[Проектирование приложения с помощью UML 4](#_Toc192835649)

[Диаграмма классов 4](#_Toc192835650)

[Диаграмма последовательности (добавление новой компании) 6](#_Toc192835651)

[Процесс добавления новой компании 6](#_Toc192835652)

# Формулировка идеи приложения

# Идея веб-приложения

Название: "**Фрилансерский поиск**"

Описание: Веб-приложение, которое позволяет фрилансерам находить компании, предоставляющие информацию о себе через Google Форму. Пользователи могут просматривать профили компаний, их услуги и контактные данные. Компании могут самостоятельно добавлять и обновлять свою информацию через форму.

# Использование гибких методов разработки

# Итерации (спринты)

**Спринт 1: Основные функции**

- **Определение требований:**

- Создание базы данных для хранения информации о компаниях.

- Реализация интерфейса для создания и редактирования профилей компаний.

- **Функционал**:

- Интерфейс для добавления новых компаний.

- Форма для редактирования информации о компании (название, описание, услуги, контакты).

**- Тесты:**

- Проверка корректности сохранения данных.

- Тестирование интерфейса на удобство использования.

**Спринт 2: Поиск и фильтрация**

- Определение требований:

- Реализация функции поиска по названию компании и фильтрации по категориям услуг.

- **Функционал**:

- Интерфейс для поиска и фильтрации компаний.

- Отображение результатов поиска.

**- Тесты:**

- Проверка корректности работы поиска.

- Тестирование фильтрации по различным критериям.

**Спринт 3: Отзывы и рейтинги**

- Определение требований:

- Реализация системы отзывов и рейтингов для компаний.

**- Функционал:**

- Интерфейс для добавления отзывов.

- Отображение среднего рейтинга компании.

**- Тесты:**

- Проверка корректности добавления и отображения отзывов.

- Тестирование расчета среднего рейтинга.

# Проектирование приложения с помощью UML

# Диаграмма классов

**Класс Company**

**Описание:** Этот класс представляет компанию, которая предоставляет информацию о своих услугах и контактных данных.

**Атрибуты:**

String name - название компании.

String description - описание компании и ее услуг.

String services - перечень услуг, предоставляемых компанией.

String contactInfo - контактная информация (телефон, email и т.д.).

List<Review> reviews - список отзывов о компании.

**Методы:**

addCompany() - метод для добавления новой компании в базу данных.

updateCompany() - метод для обновления информации о компании.

getCompanyInfo() - метод для получения полной информации о компании.

getAverageRating() - метод для расчета среднего рейтинга на основе отзывов.

**Класс User**

**Описание:** Этот класс представляет пользователя (фрилансера), который ищет компании и оставляет отзывы.

**Атрибуты:**

String username - имя пользователя.

String password - пароль для доступа к системе.

List<Review> myReviews - список отзывов, оставленных пользователем.

**Методы:**

searchCompany(String query) - метод для поиска компаний по ключевым словам.

leaveReview(Company company, String content, int rating) - метод для добавления отзыва к выбранной компании.

getMyReviews() - метод для получения списка всех отзывов, оставленных пользователем.

**Класс Review**

**Описание:** Этот класс представляет отзыв о компании, который оставляет пользователь.

**Атрибуты:**

String content - текст отзыва.

int rating - рейтинг, выставленный компании (например, от 1 до 5).

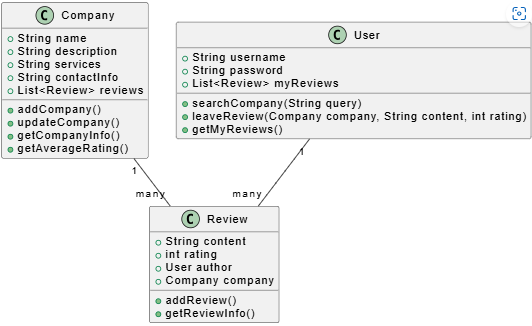
User author - пользователь, оставивший отзыв.

Company company - компания, к которой относится отзыв.

**Методы:**

addReview() - метод для сохранения отзыва в базе данных.

getReviewInfo() - метод для получения информации о отзыве (автор, текст, рейтинг).



# Диаграмма последовательности (добавление новой компании)

**Участники:**

**User:** Пользователь, который хочет добавить новую компанию.

**WebApp:** Веб-приложение, обрабатывающее запросы пользователя.

**Database:** База данных, где хранится информация о компаниях.

# Процесс добавления новой компании

**Пользователь заполняет форму**

**User** инициирует процесс, открывая веб-страницу с формой добавления компании.

Пользователь вводит название компании, описание, услуги и контактные данные.

**Отправка данных на сервер**

**User** нажимает кнопку "Добавить компанию".

**WebApp** получает данные из формы.

**Валидация данных**

**WebApp** проверяет, корректны ли введенные данные (например, не пустые поля, правильный формат контактов).

Если данные некорректны, **WebApp** возвращает сообщение об ошибке пользователю.

**Сохранение данных в базе**

Если данные валидны, **WebApp** отправляет запрос на сохранение данных в **Database**.

**Database** сохраняет информацию о новой компании.

**Подтверждение сохранения**

**Database** отправляет обратно подтверждение о том, что данные успешно сохранены.

**WebApp** получает это подтверждение.

**Уведомление пользователя**

**WebApp** информирует **User** о том, что компания успешно добавлена.

**User** видит сообщение на экране и может перейти к просмотру списка компаний.

