# **Научно исследовательская часть**

## **2.1 Постановка задачи проектирования**

Ежедневно у людей выходит из строя, ломается огромное количество вещей (телефон, стиральная машина, холодильник), из-за этого человек, у которого что-то сломалось, будет искать специалиста, способного починить необходимую вещь. Отсюда возникает необходимость: специалисту - найти заказчика и предоставить ему услугу по ремонту; потребителю – найти исполнителя и получить услугу по ремонту. Намного удобнее и эффективнее исполнителю найти заказчика, а заказчику исполнителя, позволит специальный сайт-агрегатор, на котором будут отображаться услуги, их исполнители и цена, а также который предоставит возможность связаться заказчику с исполнителем услуги напрямую, чтобы удобнее обеспечить процесс согласования цены, времени и других мелочей, относящихся к конкретному случаю. Поэтому необходимо реализовать такое веб-приложение, которое предоставляло бы вышеописанные возможности.

Для реализации веб-приложения необходимо использовать базу данных MongoDB, серверную платформу Node.js, язык программирования JavaScript, пакет Redux toolkit, фреймворки React.js, Express.js, Tailwind CSS.

**2.2 Описание предметной области.**

Процессы аккумулирования и отображения услуг по ремонту от различных исполнителей, а также коммуникации между заказчиком и исполнителем по поводу конкретной услуги относительно несложные, но комплексные. Чтобы успешно реализовать эти процессы необходимо проделать много работы и грамотно подойти к этому.

Сперва нужно предоставить возможность исполнителям добавлять свои услуги, а также изменять или удалять их при необходимости. Затем в удобной и понятной форме отображать их. Кроме того, для заказчика возможность сортировки по категориям сделает процесс поиска услуг гораздо удобнее. Следовательно, также нужно реализовать функционал сортировки. Далее необходимо упростить процесс выбора конкретного специалиста для заказчика. Сделать это можно с помощью комментариев (отзывов) к конкретной услуге исполнителя, а также непосредственной коммуникацией между исполнителем и заказчиком используя отдельный чат. Так как комментарии могут оставлять любые пользователи, то необходимо реализовать фильтрацию нецензурных и спам слов. Кроме того, нужно иметь уполномоченный персонал (модерация), который будет иметь возможность удаления недопустимых услуг и комментариев к ним. Следовательно, необходимо разграничить обычных пользователей с модерацией, а также их полномочия на сайте.

Самым логичным и универсальным вариантом для реализации вышеперечисленного функционала будет разработка «клиент-серверного» веб-приложения с использованием базы данных.

Веб-приложение — клиент-серверное приложение, в котором клиент взаимодействует с веб-сервером при помощи браузера. Логика веб-приложения распределена между сервером и клиентом, хранение данных осуществляется, преимущественно, на сервере, обмен информацией происходит по сети. Одним из преимуществ такого подхода является тот факт, что клиенты не зависят от конкретной операционной системы пользователя, поэтому веб-приложения являются межплатформенными службами.

Клиентская часть будет реализовывать пользовательский интерфейс, формировать запросы к серверу и обрабатывать ответы от него.

Серверная часть — получать запрос от клиента, выполнять вычисления, после этого формировать веб-страницу и отправлять её клиенту по сети с использованием протокола HTTP.

Для создания веб-приложений на стороне сервера используются разнообразные технологии и языки программирования, способные осуществлять вывод в стандартную консоль.

На стороне клиента будут использоваться:

* Для реализации GUI
* Tailwind CSS
* React.js
* Для формирования и отправки запросов на сервер:
* JavaScript
* Axios
* Redux toolkit

На стороне сервера:

* Для хранения данных
* MongoDB
* Для формирования запросов, обработки и отправки данных на клиент:
* JavaScript
* Node.js
* Express.js
* Mogoose

**2.3 Анализ аналогов и прототипов**

Для того чтобы сделать хороший проект, необходимо проанализировать и сравнить уже существующие, самые часто используемые веб-приложения агрегаторы, и выделить их основные преимущества и недостатки. Воспользовавшись недочётами других разработчиков, не повторим их в своем проекте и сделаем более удобное веб-приложение.

Посмотрим на преимущества и недостатки Яндекс Услуг https://uslugi.yandex.ru/

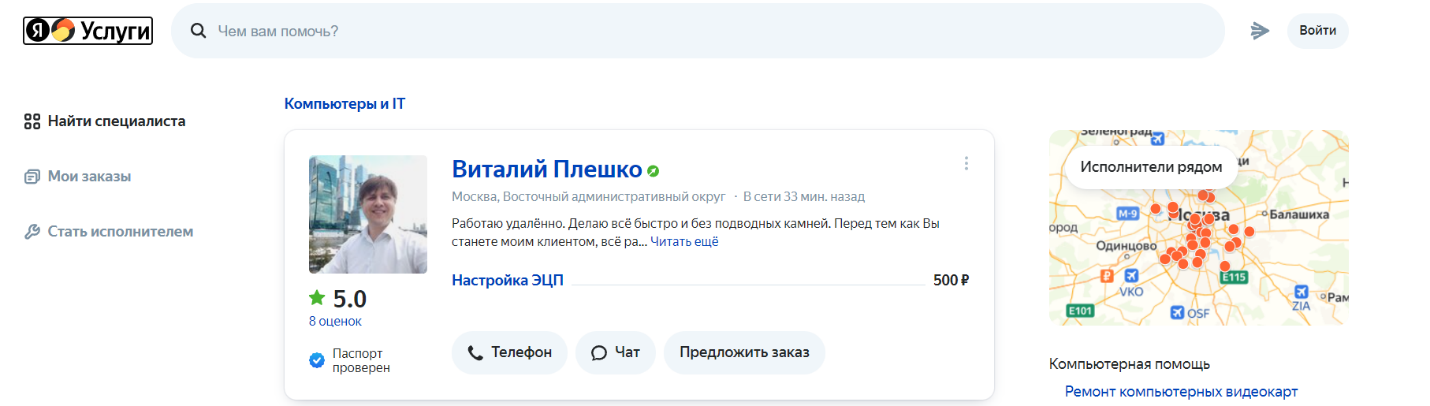


Рисунок 2.1 - Интерфейс сайта Яндекс Услуги

Преимущества:

* возможность стать исполнителем, а также разместить заказ;
* возможность сортировки услуг по различным критериям;
* информативная карточка исполнителя со всей необходимой информацией;

Недостатки:

* загруженный интерфейс, изобилие функционала => сложно ориентироваться на сайте;
* отсутствие возможности связаться с исполнителем напрямую через сайт, а только через дополнительные приложения (WhatsApp, Яндекс Мессенджер и т.д.).

Далее проанализируем сайт https://youdo.com/

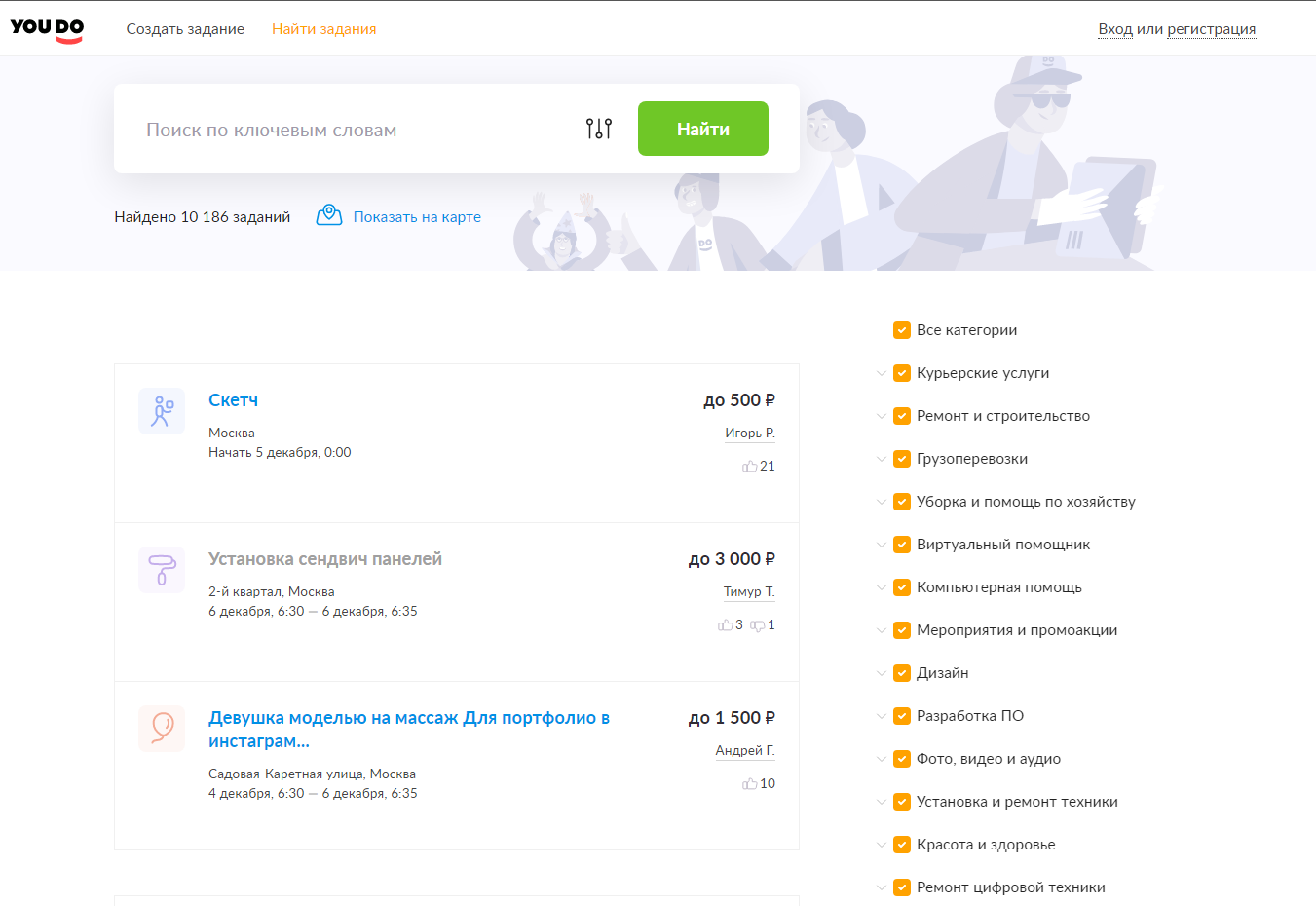


Рисунок 2.2 – Интерфейс сайта https://youdo.com/

Преимущества:

* приятный и удобный интерфейс;
* лаконичная карточка услуги со всей необходимой информацией, но без излишеств;

Недостатки:

* отсутствие фотографии исполнителя на карточке услуги;
* отсутствие возможности связаться с заказчиком/исполнителем напрямую на сайте:
* отсутствие быстрой возможности искать исполнителей собственноручно (для этого нужно создавать конкретный заказ на услугу, который не факт что увидят и захотят делать).

Проанализировав аналоги и прототипы, можно сделать вывод, что существующие веб-приложения агрегаторы не позволяют быстро решать все поставленные задачи, поэтому необходимо разработать собственное веб-приложение, которое бы предоставляло удобные возможности размещения и отображения услуг по ремонту, а также быстрой связи с конкретным исполнителем.

## **2.4 Перечень задач, подлежащих решению в процессе разработки.**

* разработка пользовательского интерфейса с использованием React.js и Tailwind CSS;
* разработка базового функционала веб-приложения (пользовательский интерфейс, клиентская часть с использованием React.js, Tailwind CSS и JavaScript, серверная часть с использованием Node.js, Express.js, MongoDB);
* разработка модели базы данных;
* разработка запросов с использованием Axios, Redux, Mongoose;
* привязка запросов к пользовательскому интерфейсу;
* привязка базы данных к серверной части веб-приложения для хранения данных;
* тестирование разработанного приложения;
* исправление выявленных ошибок.

**2.5 Обоснование выбора инструментов и платформы для разработки.**

Для успешной разработки проекта, используется операционная система Windows 10. Windows 10 — [операционная система](https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0) для [персональных компьютеров](https://ru.wikipedia.org/wiki/%D0%9F%D0%B5%D1%80%D1%81%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9_%D0%BA%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80) и [рабочих станций](https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D0%B1%D0%BE%D1%87%D0%B0%D1%8F_%D1%81%D1%82%D0%B0%D0%BD%D1%86%D0%B8%D1%8F), разработанная корпорацией [Microsoft](https://ru.wikipedia.org/wiki/Microsoft" \o "Microsoft) в рамках семейства [Windows NT](https://ru.wikipedia.org/wiki/Windows_NT).

Данная платформа предоставляет простой и удобный доступ к использованию широкого набора инструментов для разработки:

* редакторы кода;
* менеджеры пакетов;
* базы данных;
* интегрированные среды для разработки.

Согласно Statcounter.com, общий рейтинг операционных систем, показывает, что лидером является Windows, который установлен на 76,58% всех компьютеров, OS X — 18,93% и Linux — 1,62%. Платформа Windows является самой популярной и доступной, также для этой платформы создано много удобных средств, для облегчения выполнения проекта.

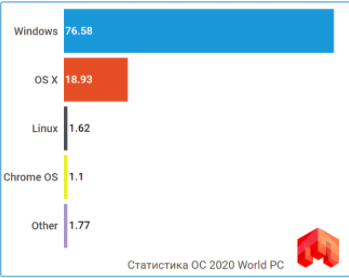


Рисунок 2.3 - Статистика использования операционных систем в мире.

**VS Code**

Для успешной разработки проекта, используется редактор кода Visual Studio Code.

Visual Studio Code (VS Code) от Microsoft – [текстовый редактор](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%B4%D0%B0%D0%BA%D1%82%D0%BE%D1%80_%D0%B8%D1%81%D1%85%D0%BE%D0%B4%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BA%D0%BE%D0%B4%D0%B0) для [кроссплатформенной](https://ru.wikipedia.org/wiki/%D0%9A%D1%80%D0%BE%D1%81%D1%81%D0%BF%D0%BB%D0%B0%D1%82%D1%84%D0%BE%D1%80%D0%BC%D0%B5%D0%BD%D0%BD%D0%BE%D1%81%D1%82%D1%8C) разработки. Распространяется бесплатно, разрабатывается как программное обеспечение с [открытым исходным кодом](https://ru.wikipedia.org/wiki/%D0%9E%D1%82%D0%BA%D1%80%D1%8B%D1%82%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5), но готовые сборки распространяются под [проприетарной лицензией](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%BF%D1%80%D0%B8%D0%B5%D1%82%D0%B0%D1%80%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5" \o "Проприетарное программное обеспечение). Функциональные возможности позволяют создавать, тестировать и разворачивать полноценные проекты.

Ключевые особенности среды разработки VS Code:

* подсветка и дополнение кода при указании типа данных, создании языковых конструкций (классов, методов) – технология IntelliSense;
* функционал отладки кода;
* встроенные команды Git;
* расширение функционала добавлением новых языковых модулей и тем;
* простота развёртывания проектов.

Ниже перечислены некоторые популярные возможности Visual Studio Code, которые помогут вам повысить продуктивность разработки программного обеспечения.

* волнистые линии и [быстрые действия](https://docs.microsoft.com/ru-ru/visualstudio/ide/quick-actions?view=vs-2019).

Волнистые линии обозначают ошибки или потенциальные проблемы кода прямо во время ввода. Эти визуальные подсказки позволяют устранять проблемы немедленно и не ждать, пока ошибка будет обнаружена во время сборки или запуска программы. Если навести указатель мыши на волнистую линию, на экран будут выведены дополнительные сведения об ошибке. Кроме того, в поле слева может появляться значок лампочки с быстрыми действиями по устранению ошибки.

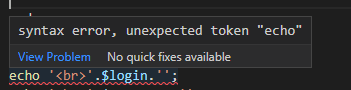


Рисунок 2.4 - Волнистые линии и быстрые действия

* поиск в Visual Studio Code

Среда Visual Studio Code может показаться сложной, ведь там столько разных меню, параметров и свойств. Чтобы быстро находить функции интегрированной среды разработки и элементы кода, в Visual Studio Code представлен единый компонент поиска (**CTRL**+**Q**).

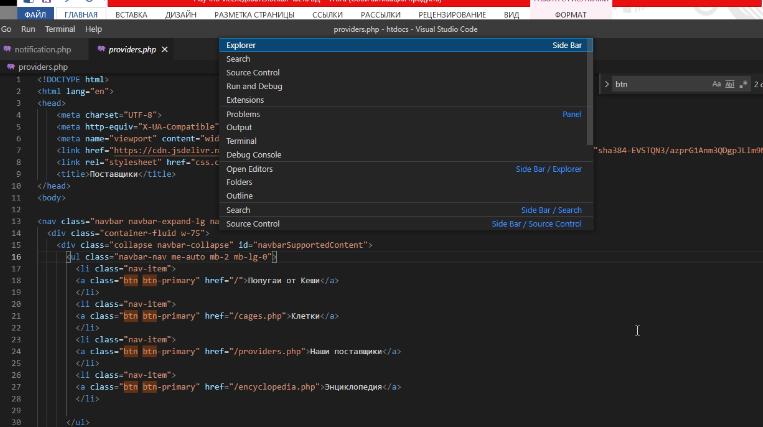


Рисунок 2.5 - Поиск в Visual Studio Code

Согласно опросу, проведенного среди 3240 разработчиков в феврале 2018 организацией Standard, разработчики выделили такие средства разработки, как наиболее часто используемые:

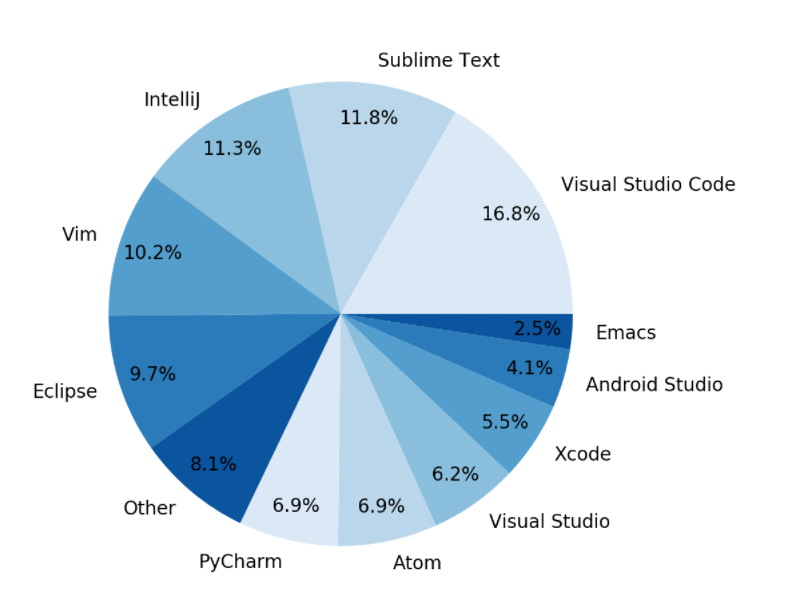


Рисунок 2.6 - Статистика использования платформ для разработки систем в мире.

Первое место среди IDE у Visual Studio Code с 16.8% использующих его разработчиков.

Также, Visual Studio Code обладает хорошим отладчиком, с помощью которого можно легко отследить ошибки в коде и без проблем их исправить, позволяет удобно и быстро создавать проекты и работать с ними.

По сравнению со средой разработки Eclipse, Visual Studio Code имеет более простой интерфейс в использовании.

**NPM**

Кроме того, следует выбрать *менеджер пакетов NPM*. Менеджеры пакетов представляют собой наборы инструментов для автоматизации различных процессов: установка, настройка, обновление и удаление приложений. NPM предлагает расширенный функционал, включая следующее:

* менеджер пакетов Node.js, позволяющий разработчикам делиться пакетными модулями;
* реестр NPM, представляющий собой открытую библиотеку пакетов с открытым исходным кодом для Node.js, веб-приложений для фронтенда и прочего;
* NPM также является менеджером пакетов в интерфейсе командной строки для их установки и публикации.

**React.js**

React — это JavaScript фреймворк и библиотека для разработки пользовательского интерфейса. Он универсален и гибок; JSX делает написание пользовательских компонентов проще; поддерживает мобильную разработку; более простое и дружелюбное написание SEO разметки; используется минимум в 1.6 миллионах сайтах; предлагает лучшую экосистему, больше шаблонов и дополнительных инструментов. Поэтому для разработки клиентской части выбор был сделан в пользу данного фреймворка.

**Node.js**

Одним из наиболее известных средств для back-end разработки программных решений на JavaScript считается платформа Node.js. Она очень проста и понятна интуитивно, что является огромным плюсом для начинающих разработчиков. При этом Node.js позволяет работать с серверными технологиями.

Сегодня Node считается одной из лидирующих платформ для веб-разработки, и большая часть веб-инструментов, серверных и клиентских, работают с этим инструментом.

Одна из причин, по которой рекомендуется Node.js по сравнению с другими фреймворками: поскольку JavaScript также используется на стороне клиента, то получается единый язык для всего приложения, вследствие чего нужно учить на один язык меньше.

Один из важнейших плюсов платформы – это асинхронность в сочетании с событийным подходом. В отличие от потокового, событийно-ориентированное программирование основано на каких-то внешних действиях. Т.е. выполнение программы зависит от действий пользователя или поступившего в программу сетевого пакета и т.д.

Такое решение упрощает программирование интерактивных приложений, т.е. основанных на работе с использованием ввода-вывода. Это могут быть онлайн-чаты или игровые приложения, красочные интерактивные веб-сервисы или всевозможные калькуляторы, рейтинги, голосования и т.д.

Кроме того, к преимуществам платформы относят:

* простой и широко известный JavaScript;
* богатая стандартная библиотека;
* огромное внешних библиотек и готовых модулей.

Над постоянным улучшением платформы работают тысячи заинтересованных программистов из разных стран. Проект распространяется с открытым кодом, а потому развивается огромными темпами.

**MySQL/NoSQL MongoDB**

*Базы данных SQL*

Также известные как реляционные базы данных, определяют и управляют данными на основе языка структурированных запросов (SQL). Они чаще всего используются для обработки структурированных данных, которые организуют элементы данных и стандартизируют то, как они будут связаны друг с другом и с их различными свойствами.

*Базы данных NoSQL*

Также известные как нереляционные базы данных, позволяют хранить и извлекать неструктурированные данные с использованием динамической схемы. NoSQL широко распространены из-за своей гибкой способности создавать уникальную структуру, которая может быть в форме документа, графика, столбца, организованные как структура данных.

SQL на протяжении десятилетий значительно превосходил нереляционные альтернативы, но NoSQL быстро сокращает разрыв с популярными базами данных. Хотя многие организации предпочитают переходить с устаревших баз данных, таких как Oracle, но не все переходят на NoSQL.

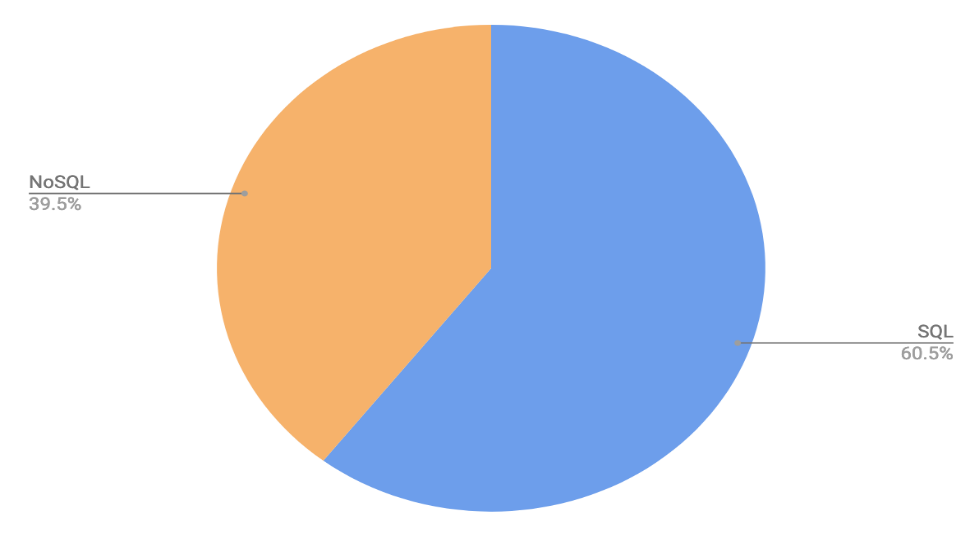


Рисунок 2.7 — Использование SQL и NoSQL

Используя данные исследования ScaleGrid — служба размещения баз данных в общедоступных и частных облаках, можно увидеть, что MySQL доминирует с 38,9% использования, за ней следуют MongoDB с 24,6%, PostgreSQL с 17,4%, Redis с 8,4% и Cassandra с 3,0%. Oracle отстал всего на 1,8%, а пользователи CouchDB, Berkeley DB, Microsoft SQL Server, Redshift, Firebase, Elasticsearch и InfluxDB объединили категорию «Другие» с 2,4%. Нельзя не отметить рост популярности MySQL, MongoDB и PostgreSQL.

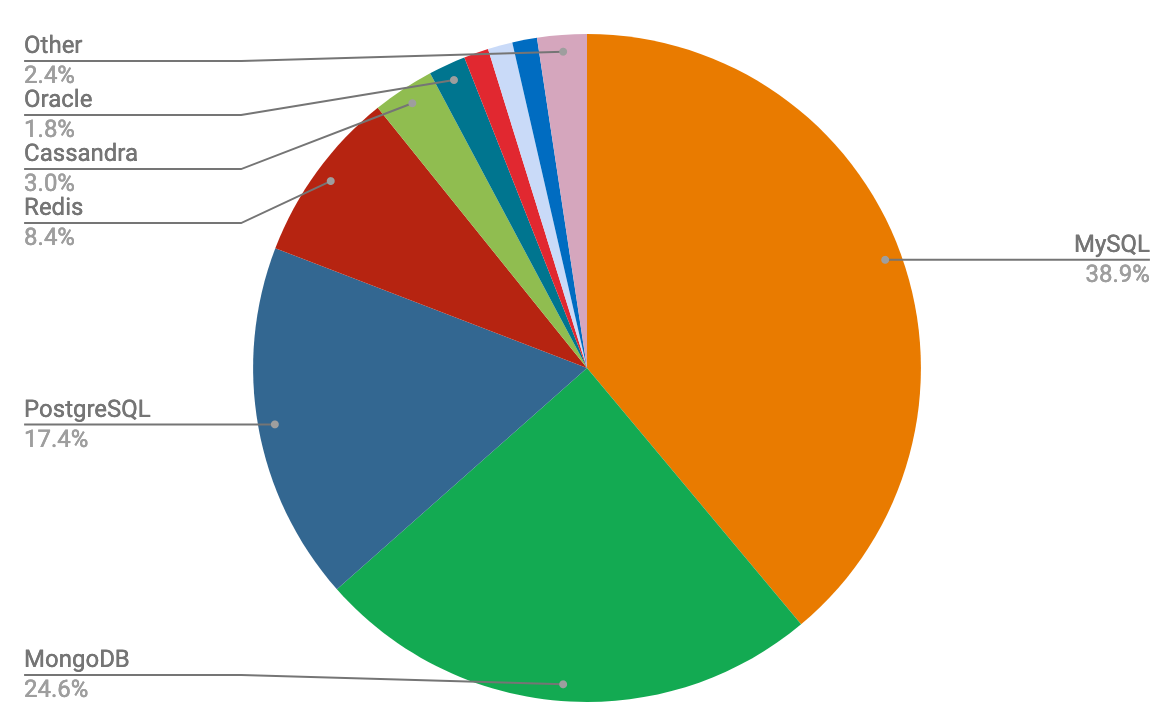


Рисунок 2.8 — Самые популярные базы данных

MongoDB построена на масштабируемой архитектуре, которая стала популярной среди разработчиков всех видов для разработки масштабируемых приложений с развивающимися схемами данных. В качестве базы данных документов, MongoDB позволяет разработчикам легко хранить структурированные или неструктурированные данные. Она использует формат, подобный JSON, для хранения документов. Этот формат напрямую сопоставляется с нативными объектами в большинстве современных языков программирования, что делает его естественным выбором для разработчиков, поскольку им не нужно думать о нормализации данных. MongoDB был создан для людей, создающих интернет-приложения и бизнес-приложения.

В MongoDB записи хранятся в виде документов в сжатых файлах BSON. Документы можно получить непосредственно в формате JSON, который имеет много преимуществ:

* это естественная форма для хранения данных;
* он удобочитаем;
* структурированная и неструктурированная информация может храниться в одном и том же документе;
* можно вложить JSON для хранения сложных объектов данных;
* JSON имеет гибкую и динамическую схему, поэтому добавление полей или удаление поля не является проблемой.
* документы сопоставляются с объектами в большинстве популярных языков программирования.

Исходя из вышеописанных преимуществ, было принято решение использовать базу данных MongoDB для разработки данного веб-приложения агрегатора.