

# СОДЕРЖАНИЕ

## ВВЕДЕНИЕ

### 1 Современные технологии и инструменты, для построения сервисов интернет коммерции

- 1.1 HTML
- 1.2 Web-серверы
- 1.3 Активные серверные страницы (ASP)
- 1.4 Взаимодействие между сервером и браузером
- 1.5 JSON
- 1.6 .NET Framework
- 1.7 ASP.NET
- 1.8 Web-сервисы
- 1.9 Базы данных

### 2 Общее описание С2С онлайн коммерции

- 2.1 Типы С2С коммерции
- 2.2 Подключение покупателей и продавцов
- 2.3 Репутация и доверие
- 2.4 Денежные операции в С2С коммерции

### 3 Реализация системы поиска исполнителей для решения бизнес-задач

- 3.1 Структура информации в базе данных
- 3.2 Серверная часть
  - 3.2.1 .....REST сервис
- 3.3 Клиентская часть

## ЗАКЛЮЧЕНИЕ

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

## ПРИЛОЖЕНИЕ А (обязательное)

## ПРИЛОЖЕНИЕ Б (обязательное)

## ВВЕДЕНИЕ

Сеть Интернет на сегодняшний день стала одним из основных инструментов деловой и общественной жизни. Новое поколение web-сервисов заменяет множество бизнес-процессов. Эти сервисы, включающие в себя различные системы на основе открытых стандартов, удобны для поддержания долговременных потоков бизнес-процессов при относительно небольших затратах [12].

Интернет стирает границы, ответ на любое событие поступает почти незамедлительно, а расстояния исчезают. По этой причине множество самых бытовых вещей переходят из материальной формы в виртуальную, что гораздо упрощает и увеличивает продуктивность их использования. Обмен данными, торговая деятельность, движения денежных средств, банковские и страховые сервисы, а также маркетинг и финансовый расчет, все чаще и чаще обретают свое отражение в масштабах Всемирной Сети. Все вышеперечисленные понятия слились в единое и создали сравнительно новую для России область экономики, как электронная коммерция.

Электронная коммерция – термин, использующийся для определения разных видов бизнеса или коммерческих сделок, содержащие в себе передачу информации через сеть Интернет. Она охватывает широкий диапазон видов бизнеса [14]. Электронная коммерция помогает покупателям вести обмен товарами, решая проблемы со временем и расстоянием. В ближайшем будущем границы между классической и электронной торговли будут более размытыми, так как все больше и больше компаний переносят разделы своих операций в сеть Интернет.

Быстрый рост рынка электронной коммерции наблюдается в течение последних 20 лет, это вызвано мгновенным ростом числа интернет-пользователей, повышением воздействия социальных сетей и других интерактивных онлайн-платформ, быстрым развитием систем электронных

платежей. Интернет технологии имеют возможность в несколько раз сократить как финансовые, так и временные расходы организаций.

Существующие подобные системы представляют прозрачный и безопасный веб-интерфейс к широкому диапазону корпоративной информации, услуг и приложений. Для того, чтобы выполнить поставленные задачи используют различное программное обеспечение, технологии и услуги от их поставщиков и компаний.

Как и в реальном мире, существует несколько общепризнанных категорий, на которые группируется онлайн коммерция. Обычно, такое разделение производится по целевой группе потребителей. Традиционно электронная торговля относится либо к «бизнес для бизнеса» (B2B), либо «бизнес для потребителя» (B2C). Говоря о потребительской коммерции, наверное, первое что приходит на ум большинства людей, так это гиганты электронной коммерции eBay и Amazon, являющиеся лидерами в B2C сегменте.

Однако, последние несколько лет в рынок вошло новое поколение электронной коммерции: «потребитель для потребителя» (C2C) и «человек для человека» (P2P). В последнее время C2C, укрепив свои позиции в сети Интернет, тем самым повлияла на финансовые условия, а также потребительское поведение.

Торговля C2C, как основной феномен, существовал длительное время до появления Интернета. Блошинные рынки, объявления в газетах – самая старая форма коммерции.

C2C рынок имеет очевидные преимущества, такие как удаление посредников из цепочки создания стоимости. Аналогично тому, как B2C коммерция C2C имеет ряд уникальных задач. Во-первых, существует первоначальная задача соединения покупателя и продавца, в то время как традиционные продавцы могут рекламировать свой бизнес, обычные же люди не могут себе позволить создать видимость тем же самым образом. После того, как покупатель и продавец связались, возникнет проблема с доверием. И

наконец, существует проблема с реальными денежными операциями. Компании имеют возможность получить деньги, используя множество различных способов и, как правило, способы принятия кредитных карт, прием финансовых операций, что является проблемой для физических лиц.

Быстрое развитие С2С коммерции породил целый ряд весьма успешных коммерческих компаний обеспечивающие вспомогательные услуги для С2С торговли. В то время как главные действующие лица в С2С коммерции являются физические лица (покупатель и продавец), наличие также коммерческой компании является необходимостью; существует определенная потребность в доверенных лицах, предоставляющие необходимую инфраструктуру для онлайн коммерции, чтобы решить все вышеупомянутые задачи.

В настоящее время, многие сервисы приходят к С2С модели. Производитель часто становится лишним в этой цепочки, и покупатель предлагает услуги для таких же покупателей. Не так давно было сложно представить, что народ стал сам создавать сервисы, которые имеют больший спрос, чем аналоги от крупных компаний. Подобным образом осуществляется развитие экономики и экономических отношений.

Для функционирования подобных систем необходимы следующие условия:

- Доступ к сети Интернет, при помощи которой выполняется взаимодействие между участниками онлайн коммерции;
- Базы данных, в которых хранится вся нужная информация;
- Веб-сервер, при помощи которого действует вся функциональность системы и осуществляется взаимодействие между участниками онлайн бизнеса;

Целью бакалаврской работы является разработка «С2С» web-сервиса поиска исполнителей бизнес-задач (представление WEB-сервиса со схемой торговли конечного потребителя с конечным потребителем).



# **1 Современные технологии и инструменты, для построения сервисов интернет-коммерции**

Очень важным этапом в создании любой системы является подбор инструментов и технологий, которые уже использовались для подобных задач. Еще нет универсального решения, которое бы подсказало, пути решения проблем онлайн коммерции. Но уже достаточно статей, в которых описаны известные инструменты, технологии, типичные проблемы и пути их решения.

Рассмотрим по порядку набор технологий, которые необходимы для создания современного бизнес-приложения и позволяющие производить информационный обмен.

## **1.1 HTML**

Изначально информация в сети интернет представлялась статически. Специально для этого был разработан язык разметки HTML. Но в такой реализации было невозможно динамическое изменение информации на стороне клиента, реагирующее на действия пользователя. Позже был создан новый стандарт DHTML, язык динамической разметки, который ввел возможности для создания web-узлов и web-приложений, взаимодействующих с пользователем. Это взаимодействие происходит за счет операций в объектной модели.

**Объектная модель (DOM)** на данный момент поддерживается всеми популярными браузерами. DOM – это модель, которая независима от платформы и языков программирования. При помощи нее можно манипулировать структурой документа, посредством использования языка JavaScript.

Для упрощения формирования страниц и определения вида страницы был разработан CSS – каскадные таблицы стилей. Использование CSS позволило упростить механизм формирования web-страниц.

Нельзя представить современную бизнес-логику клиентских приложений без общей модели событий, которые генерируются в результате действий пользователя или изменения состояния страницы и браузера. При помощи этих событий можно установить связь между действием и программной реакцией на это действие. Благодаря этим возможностям можно строить интерактивные web-приложения.

## **1.2 Web-серверы**

В основе web-сервисов лежит сервер. Который изначально хранил только статические страницы, на сегодняшнее время он способен выполнять сложные приложения, которые значительно расширяют web-узел.

В основе сервера, построенного на технологиях Microsoft, лежит IIS (Internet Information Server). IIS – это компонент Microsoft Windows, популярный и распространенный в силу простоты конфигурации веб-сервер. [13]

Изначально IIS представлял собой только базовый сервер. Через годы корпорация Microsoft развила в нем широкие возможности. Одной из важных является возможность хостинга ASP.NET приложений, а также интеграция другими технологиями корпорации Microsoft. Благодаря этому фактору на данный момент появилось целое множество хостинг-провайдеров, предлагающие свои услуги на базе IIS[1].

## **1.3 Активные серверные страницы (ASP)**

Чуть позже возникла новая технология, которую создала корпорация Microsoft – ASP (Active Server Pages – активные серверные страницы) [ CITATION Хил \1 1033 ]. Рассмотрим основные преимущества ASP.

Главной является то, что на клиентской части не видим исполняемый код, а только готовый собранный HTML код. На серверной части выполняется код, наполненный переменными, которые обозначены специальными символами. Это позволяет выполнять необходимые методы, для формирования страниц, не задумываясь о совместимости программ просмотра, так как он выполняется на серверной стороне.

Для сборки более сложных страниц, обычно не хватает средств скриптовых языков, так как у них нет функциональных возможностей доступа к данным и файловой системе. По этой причине в них добавили дополнительные возможности и объекты ASP. Принцип их работы изображен на рисунке 1.

ASP-компоненты – это объекты ActiveX, которые содержатся в документе и явным образом определяют его для браузера. У ActiveX есть альтернатива – апплет, которые также загружаются и выполняются в браузере.



Рисунок 1 – Обработка активной страницы сервером



## 1.4 Взаимодействие между сервером и браузером

Связь между браузером и сервером происходит по принципу осуществления транзакции без состояния, во время которой web-сервер отправляет данные на клиентскую часть и забывает о ней, до тех пор, пока не будет снова осуществлен запрос.

Огромное влияние на инициализацию и последовательность выполнения web-приложения оказывает моменты его запуска и завершения, для этого важно определение и учет пользовательских сессий.

Веб-приложение может иметь отношение к множеству браузеров, которые обращаются к различным веб-страницам. Во время первого обращения заводится сессия, которая имеет отношения к браузеру. Длительность сессии определяется временем с первого обращения до последнего запроса к страницам. Приложение завершается, если в нем не осталось не одной активной сессии.

## 1.5 JSON

Уже на данном этапе можно спроектировать систему при помощи взаимодействия приложений. В силу специфики описанной выше появляется ряд преимуществ. Клиентская часть отделена от серверной, и уже не нужно исполнять ее код на том же физическом уровне, что и бизнес-логика. При помощи вышеописанных технологий уже возможно построить сервис-ориентированную архитектуру системы.

Благодаря внедрению принципа «написал один раз, выполнил где угодно» возникла перспектива гораздо ускорить внедрение сервис-ориентированной архитектуры. Становится явным, для нейтральности и технической независимости приложения, должен быть и формат обмена данных также нейтральным. JSON полностью подходит под эти нужды.

JSON – это простой и очень эффективный формат для передачи данных через веб. Структура JSON выглядит как пара значений «ключ/значение» и массивы значений [ CITATION Дже15 \l 1033 ]. С подробной спецификацией можно ознакомиться в открытых источниках [ CITATION JSO16 \l 1049 ].

## 1.6 .NET Framework

Как мы разобрались выше, что для проектирования надежной системы необходимы не только технологии, но и хорошие инструменты. Зачастую под инструментами подразумеваются платформенные решения и среды разработки.

Очередной разработкой Microsoft является **.NET Framework**. .NET Framework включает в себя языки программирования и платформу выполнения, а также библиотеки классов, которые представляют различные встроенные функциональные возможности. [CITATION Вве12 \p 17 \l 1049 ]

**.NET Framework** – платформа, разработана для простой разработки приложений в распределенной сети Интернет. Эта платформа является оболочкой для операционной системы, что избавляет разработчика от учета специфических особенностей систем. Это дает возможность переноса приложений на различные платформы и аппаратные средства. Эта платформа состоит из двух главных частей CLR (Common Language Runtime) и библиотеки классов (рисунок 2).

- **CLR** – это среда выполнения (runtime environment), представляющая службы необходимые всем приложениям .NET. CLR также представляет общую систему типов данных. Благодаря этому CLR имеет возможность работать со всеми приложениями, написанными на .NET. [6, стр. 35]

- Библиотеки классов (Class Libraries). Платформа .NET Framework имеет огромный набор сборок в библиотеках классов. Сборки имеют многочисленные типы данных, которые доступны разработчикам для создания всяческих приложений. [6, стр. 35]

CLR имеет множество проверок безопасности, причиной тому служит то, что основная часть .NET была создана для web-приложений. Эта особенность понижает производительность. Но для распределенных узлов производительность чаще зависит не от процессора, а связана с

производительностью внешнего ресурса, к примеру, файловая система или база данных сервера.

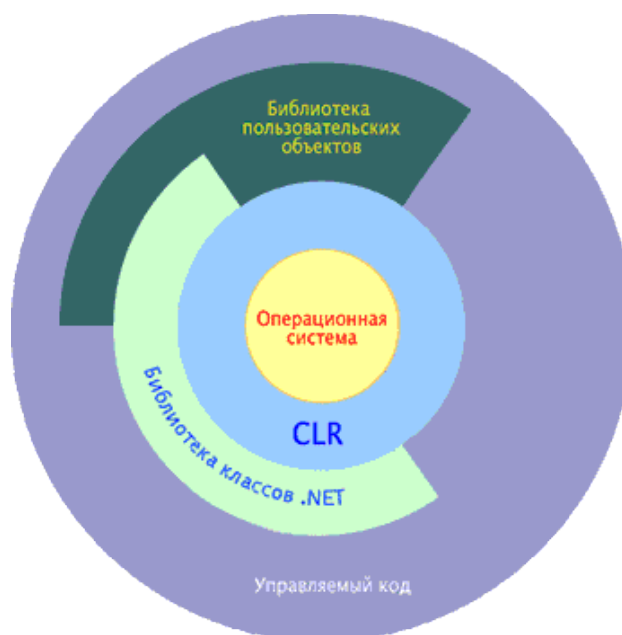
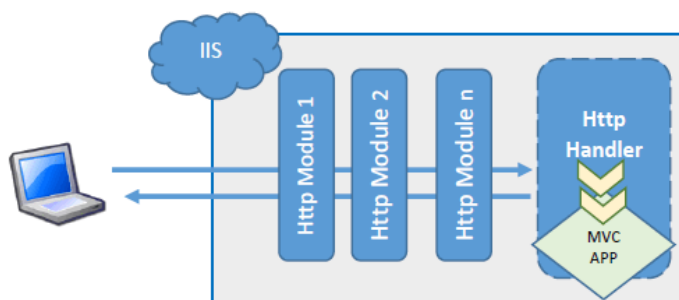


Рисунок 2 – Схематическое представление .NET Framework

## 1.7 ASP.NET

В главе 1.3 уже были описаны основные преимущества ASP. Корпорация Microsoft усовершенствовала ее и представила, как часть архитектуры .NET [ CITATION Мак03 \l 1033 ]. Разработчики взаимодействуют с ней, обращаясь к типам из библиотеки для создания программ и web-форм. Когда приходит запрос с клиентской стороны, служба ASP.NET запускается, выполняет код и отправляет ответ клиенту (рисунок 3).



### Рисунок 3 – Схема обработки HTTP-запросов ASP.NET приложения

ASP.NET решил основную проблему ASP, в которой происходил повторный анализ страниц при каждом запросе, что сильно замедляло процесс обработки запросов и выдачи результатов. В ASP.NET все происходит иначе, при первом выполнении приложения оно собирается в бинарную .dll (Dynamic Link Library) библиотеку. Такой подход в программировании с использованием кода поддержки позволяет скомпилировать классы страниц в динамическую библиотеку до того, как эти страницы будут использоваться. И в последующее время все запросы будут обрабатываться этой библиотекой, а в случае необходимости она будет перекомпилирована заново.

## 1.8 Web-сервисы

В прошлом интернет выглядел как множество серверов, хранящих в себе статические страницы, которые имели ссылки друг на друга. В настоящее время нельзя представить всемирную сеть без интерактивных web-приложений, которые реагируют на действия пользователей, хранят множество информации в базах данных, а также могут выдавать динамические web-страницы. Но, развитие Интернет не обошлось только web-приложениями. Из-за потребности взаимодействия и интеграции иных приложений появились новые технологии. Появились сервис-ориентированные архитектуры бизнес приложений, благодаря которым появился протокол REST. REST позволяет обмениваться информацией по известным форматам как JSON, XML, которые с легкостью могут распознать и сервер и клиент. Основой сервис-ориентированного Web являются Web-сервисы [8, стр. 10] – программные средства, взаимодействующие между

собой благодаря стандартным технологиям. Web-сервисы можно выделить в отдельную технологию, по следующим критериям:

- Web-сервисы не имеют языковых, инструментальных и платформенных зависимостей.
- Они придерживаются открытых стандартов: JSON – формат обмена данными, HTTP – протокол передачи данных.
- Применяют одинаковые пути поиска и публикаций.

Рассмотрим основные стандарты на которых основаны REST Web-сервисы.

Web-API это способ выполнения удаленного вызова методов при помощи HTTP запросов благодаря **REST** (Representational State Transfer - передача состояния представления). В прошлом вызов метода на сервере и получения ответа были весьма проблематичными, а настроить эту передачу было еще более проблематично. При помощи REST протокола эта задача гораздо упростилась. Эта технология, в основе которой может использоваться такой формат как JSON, делает вызов серверных методов более простым. REST сервер принимает HTTP запросы, после которых создает нужные объекты, проводит заданные операции и отвечает клиенту, вернув либо код успешного завершения операции, либо результаты выполнения. При использовании ASP.NET мы можем использовать все возможности .NET и C# на серверной стороне службы. Эти web-приложения можно использовать и на других платформах, имеющих доступ к HTTP (рисунок 4).

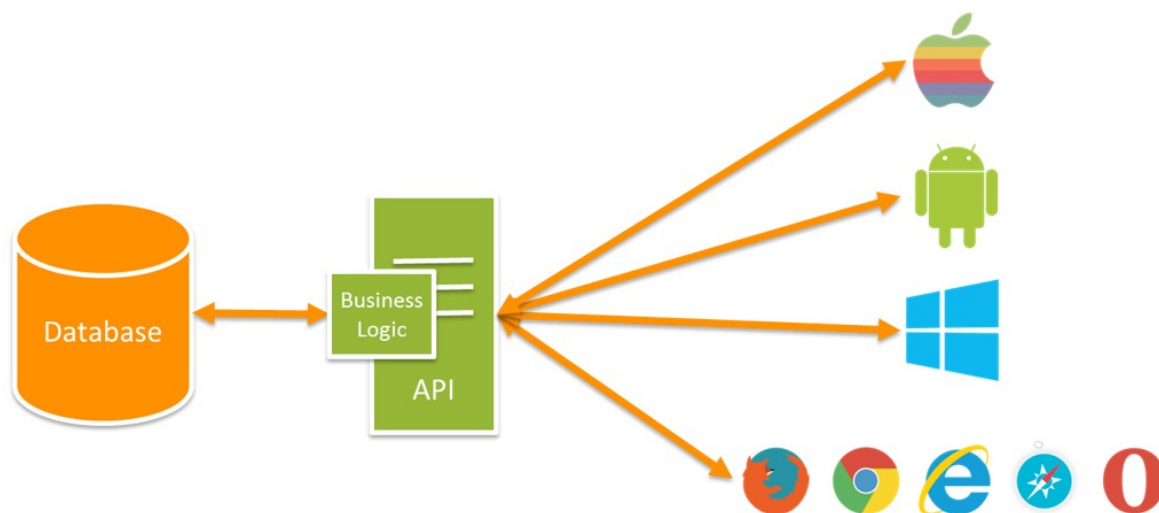


Рисунок 4 – Использование RESTful API на различных платформах

**Web-сервис** – это сервис-ориентированная архитектура (Service-Oriented Architecture, SOA) [ CITATION Ген04 \l 1049 ]. Принцип проектирования ПО как службы, которая отделялась от клиентской части, был одним из идей разработки ПО в течение нескольких лет (рисунок 5). Изначально пытались сделать вычисления распределенными, и такие технологии как CORBA и DCE привели к мысли, что web-приложения должны быть отдельны от пользовательского интерфейса. Такие сервисы обычно имеют свой API (application programming interface – интерфейс программирования приложений). API определяет функционал и конечные точки которые представляет web-сервис.

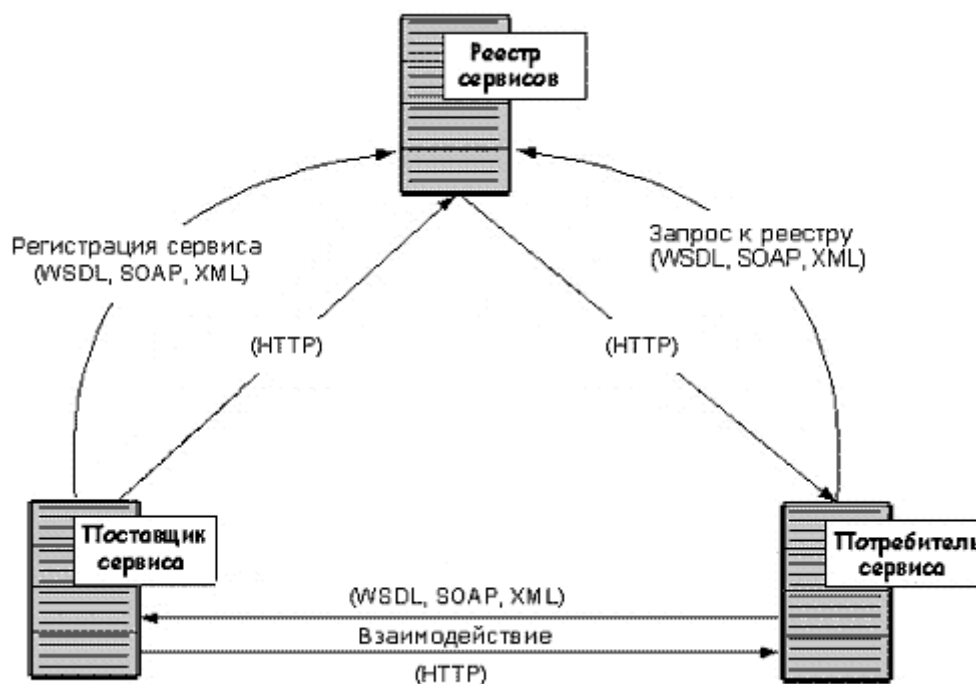


Рисунок 5 – Общая схема SOA

Данный метод разделения логики дал множество преимуществ. Теперь серверная и клиентская части находятся на разных физических уровнях и могут масштабироваться не влияя друг на друга.

Обратим внимание на самые распространенные проблемы, возникающие при создании сервиса для электронной коммерции:

1. Объединение множества приложений и получение актуальной информации, для обработки этим приложением;
2. Web-сервисы могут использовать HTTPS протокол, который предоставляет зашифрованный канал между сервером и клиентом. Протоколы HTTP и HTTPS являются синхронными. Это означает, что каждый вызов службы, будет ожидать ответа. Если служба не работает или тратит на выполнение большое количество времени, то вызывающий клиент, чаще всего, воспринимает как неудачу. На этот случай нужны асинхронные вызовы. Асинхронность – возможность вызывающего клиента работать вне механизма запрос-ответ. Она



очень важна в бизнес-приложениях, потому, что многие бизнес-операции требуют несколько диалогов для записи операций. И не все эти диалоги возможны при синхронном взаимодействии. С# представляет несколько механизмов для реализации асинхронного взаимодействия с сервисом;

3. Как правило, существует смесь программных средств различной природы. Web-сервисы представляют независимость от платформы. Основные средства сервиса являются универсальными: JSON – нейтрален относительно языков программирования, а HTTP есть практически на любой платформе;
4. Использование несколько бизнес-систем и безопасный доступ к хранилищу данных;

## **1.9 Базы данных**

Одним из важных этапов создания web-приложения является проектирование базы данных и ее процедур. Самые распространенные системы управления базами данных, которые используются в настоящее время, это – IBM DB2, Informix, Microsoft SQL Server, Oracle, Sybase и MySQL и тд. С прогрессом в серверных технологиях, на базы данных возлагается ответственность не только за хранение информации, но и за хранение процедур, различных объектов, триггеров, которые помогают сохранить целостность базы данных и дает возможности обработки транзакций.

## 2 Общее описание С2С онлайн коммерции

### 2.1 Типы С2С коммерции

С2С системы и модели могут быть классифицированы по ряду направлений (таблица 1). С одной стороны, можно классифицировать по их «чистоте», анализируя сколько внешних игроков, кроме покупателя и продавца, должны быть вовлечены в сделку. Другим критерием классификации может быть место нахождения. Уровень близости также имеет некоторые важные последствия при доверии и денежно-кредитных сделках.

Другая категоризация может быть сделана по цене. Фиксирована ли она заранее или есть активная ценовая конкуренция. Есть целый ряд моделей с фиксированной ценой, где цену товара заранее фиксирует продавец. С другой стороны, есть также ряд вариаций на моделях аукциона, где покупатели открыто конкурируют за товары и услуги для продажи. Конечно, даже с моделями фиксированной цены никто не может предотвратить любые ценовые повторные переговоры, которые происходят между покупателем и продавцом. В этом смысле все модели С2С электронной коммерции основаны на гибкую ценовую политику.

Таблица 1 – Различные пути классификации С2С коммерции

Классификация	Категории и примеры	
	1	3
Расстояние	Продавец и покупатель находятся далеко друг от друга	Покупатель и продавец поблизости
	Покупатель из России заказывает на eBay аукционе с Канады	Покупатель из России заказывает товар на avito

### Окончание таблицы 1

1	2	3
Цена	Фиксированная	Гибкая
Количество внешних участников	Нет	Много
	Купить что-то у друга	Использовать поиск Google, найти товар на Aliexpress, использовать кредитную карту для покупки

Некоторые из классификаций, перечисленные здесь, также связаны между собой; например, для крупных систем, как правило, невозможность для покупателя и продавца быть далеко друг от друга, без участия третьих лиц. Создание связи между покупателем и продавцом на расстоянии – один из сложных аспектов.

На данный момент доминируют два типа С2С коммерции; аукционы с гибкими ценами, где покупатели и продавцы, как правило, географически удалены друг от друга и онлайн объявления с более фиксированными ценами – переговоры в этой модели, если таковые имеются, имеют место между потенциальным покупателем(ями) или продавцом, в отличие от открытого аукциона, где все предложения видны всем пользователям.

## 2.2 Подключение покупателей и продавцов

Как уже ранее упоминалось, первая проблема С2С коммерции является объединение покупателей и продавцов. В то время как традиционные торговцы имеют видимость через физическое присутствие, как магазины. Также могут позволить себе рекламировать свои услуги при помощи средств массовой информации. Эти действия, как правило, слишком дороги для потребителя. Какие действия совершает физическое лицо, чтобы найти

нужного человека, чтобы купить то, что продается по максимально доступной цене?

Конечно же начинается с четкой постановки задачи и необходимости ее решения. Не удивительно, что многие компании решают проблему объединения. Эти две наиболее успешные модели были перенесены из реального мира в виртуальный. Одним из ключевых факторов успеха, особенно для сайтов аукционов, является достаточно большая пользовательская база. Поэтому существует ограниченное число таких услуг. В области интернет-аукционов C2C, существует мировой лидер: eBay.

Как заработать деньги на интернет-аукционе? C2C интернет-аукцион является одним из примеров так называемых многогранных рынков; поставщику услуг, сайта аукциона, C2C нужно получать доход более чем с одной стороны, участвующей в сделках. Интернет-аукционы практически никогда не взимают плату от покупателей, поскольку такая практика может быстро отбить желание пользоваться сервисом. Вместо того, поставщики услуг могут взимать ряд сборов от продавцов. Во-первых, может быть основной взнос на основе стоимости пункта от продажи, либо фиксированная сумма, процент от цены или же все вместе. Во-вторых, функция видимости, такие как изображения, заметные шрифты, размещение в результатах поиска, платные услуги и т.д. И наконец, такие компании, как eBay, которые владеют платформой платежей могут получать доход от обработки денежных транзакций.

Со стороны электронных объявлений, с точки зрения поставщиков услуг, ситуация на рынке более рассеяна. Тем не менее, и здесь первопроходцем и в какой-то степени имеет доминирующую роль сервис Craigslist. Этот сервис родом из Сан-Франциско. Первоначально был службой в качестве локальных объявлений. В настоящее время этот сервис охватывает более 200 городов в 35 странах по всему миру. Craigslist получает прибыль, в основном, от взимания платы за корпоративные объявления; для физических

лиц размещение объявлений полностью бесплатно. В настоящее время Craigslist получает более 8 миллионов рекламных объявлений ежемесячно.

Помимо аукционов и объявлений о продаже, существуют различные форумы и группы новостей, как еще один способ объединения покупателей и продавцов. Существуют буквально тысячи таких форумов и новостных групп по всему миру, и они, как правило, сосредоточены на торговле товаров и услуг, которые представляют интерес для любой географически ограниченной зоной, специальные группы интересов.

В целом можно сказать, что проблема объединения покупателей и продавцов была решена таким образом, в то время как, существует значительное пространство для улучшения. На сегодняшний день эти сервисы способны обеспечить достойный уровень услуг по соединению продавцов с потенциальными покупателями. Но до сих пор остается сложной задачей это поиск соответствующих каналов продаж, для специфических групп по интересам.

### **2.3 Репутация и доверие**

Вторая проблема в С2С коммерции – это надежность; по сравнению с В2С торговлей, завоевать доверие является более сложным, когда покупатель и продавец являются физическими лицами, и с Интернетом появляются недостатки анонимности – потенциал для мошенничества или введения в заблуждение гораздо больше, чем в реальном мире. Тем не менее, есть способы смягчения проблемы доверия и успеха аукциона С2С системы многим обязаны развитию систем репутации.

Одним из ключевых вопросов в любой коммерции – это доверительные отношения, необходимые для покупки. В В2С коммерции покупатель должен доверять продавцу для доставки заказанных товаров. Если продавец ведет бизнес в авторитетной манере, он в конечном итоге получит репутацию заслуживающую доверие, которая поможет клиентам выбрать надежного продавца.

Кроме того, онлайн рейтинговые услуги, как BizRate.com и Epinions.com, где клиенты могут дать обратную связь и оценить торговцев, помогают уменьшить риск клиентов и больше доверять продавцу. Если коммерсант имеет в целом хорошую репутацию, это явный признак того, что можно считать его более надежным, чем торговец с большим количеством отрицательных отзывов.

Потребность в доверии, в первую очередь, связано с тем, что обычно доставка проданных товаров и передача денег не может быть завершена в одно и то же время. На практике обычно, либо покупатель платит заранее, и, таким образом, должен доверять продавцу, чтобы доставили товар или продавец должен доверять покупателю, чтобы он своевременно перевел деньги. Есть место для мошенничества, если другая сторона не заслуживает доверия. Несложно заметить, что для любой системы торговли C2C проблема доверия должна быть решена приемлемым способом.

Когда сделка C2C (то есть обмен денег и услуг или товарами) происходит лицом к лицу, вопрос доверия решить намного легче. В этом случае обмен денег может быть одновременным с обменом товаров или услуг. Однако, часто нет возможности для покупателя и продавца встретиться – что и есть одно из главных преимуществ электронной коммерции, что они не вынуждены встречаться.

На рисунке Рисунок 6 показаны необходимые доверительные отношения, которые должны быть для того, чтобы C2C сделка состоялась. Вдобавок общему доверию покупателя и продавца, должны доверять поставщикам услуг, оказывающим содействие. К ним относятся посредник для назначения «места встречи», обычно хост-аукцион или аналогичный поставщик, а также (если не тот же самый сервис) поставщик финансовых услуг, для завершения финансовой сделки. В итоге, если товар был продан, обе стороны должны доверять службе доставки, используемой для пересылки товаров. Были предложены полностью распределенные системы для

цифровой коммерции и электронных денег, но маловероятно, что такие системы получают значительную долю рынка в ближайшей перспективе.

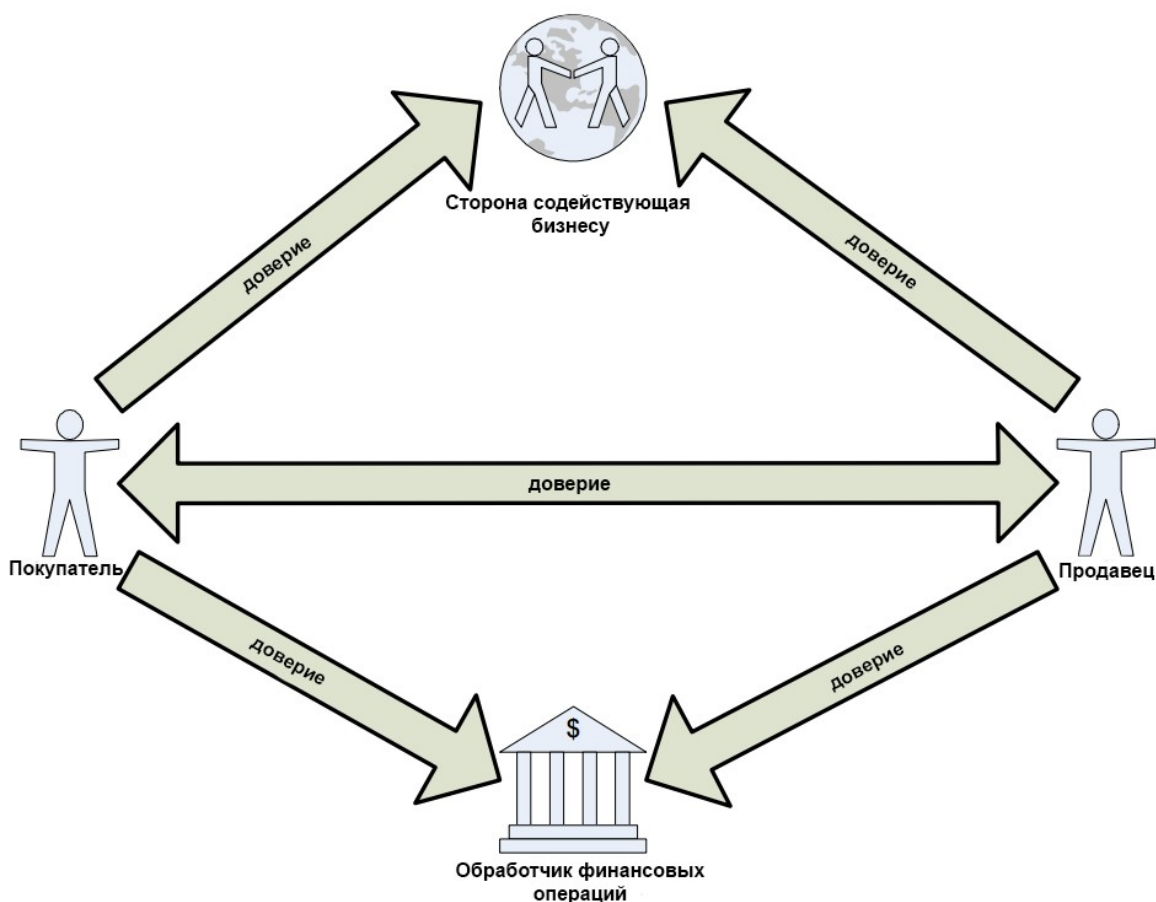


Рисунок 6 – Необходимые доверительные отношения в сделках С2С

## 2.4 Денежные операции в С2С коммерции

В дополнение к проблемам, их мы уже рассмотрели (связи и доверие), есть еще одна задача для успешной С2С электронной коммерции – надежное и простое осуществление денежных операций между покупателем и продавцом. Как правило, это требует другого доверенного поставщика услуг, для участия в сделке.

### 3 Реализация системы поиска исполнителей для решения бизнес-задач

#### 3.1 Структура информации в базе данных

Прежде всего выясним, какую информацию мы будем хранить в базе данных. В базе данных сервиса будет храниться информация об объявлениях, когда-либо созданных, в том числе и завершенных. Объявления, в свою очередь, подразделяются по категориям. У каждого объявления есть владелец. Не стоит забывать о важной составляющей электронного объявления – предложенной ценой или оферты, которая содержит текст предложения и цену, за которую исполнитель готов выполнить задание. На основании вышесказанного создадим систему таблиц, в которых будет содержаться следующая необходимая информация:

1. dbo.Bids – основная таблица с объявлениями (таблица 2);
2. dbo.Offers – таблица оферт (таблица 3);
3. dbo.Category – таблица категорий (таблица 4);
4. dbo.ApplicationUsers – таблица пользователей сервиса;

Таблица 2 – Основная таблица с объявлениями (dbo.Bids)

Наименование атрибута	Тип данных	Длина поля	Описание
1	2	3	4
Id	int	4	Идентификатор объявления
CategoryId	int	4	Идентификатор категории
UserId	nvarchar	128	Идентификатор пользователя
Title	nvarchar	150	Название

Окончание таблицы 2

1	2	3	4
---	---	---	---



Payment	decimal	18	Цена за выполнение
Description	nvarchar	200	Краткое описание
DatePublication	datetime	–	Дата публикации
DateExpiration	datetime	–	Дата завершения
TelNumber	nvarchar	11	Номер телефона
FullDescription	nvarchar	max	Полное описание
Closed	Boolean	–	Объявление закрыто/открыто
Address_Id	int	4	Идентификатор адреса
Address_Latitude	decimal	18	Широта
Address_Longitude	decimal	18	Долгота
Address_Locality	nvarchar	100	Местонахождение
Address_Street	nvarchar	100	Улица
Address_House	nvarchar	10	Дом

Таблица 3 – таблица оферт (dbo.Offers)

Наименование атрибута	Тип данных	Длина поля	Описание
1	2	3	4
Id	int	4	Идентификатор оферты
BidId	int	4	Идентификатор объявления
UserId	nvarchar	128	Идентификатор
Body	nvarchar	max	Текст предложения
Payment	decimal	18	Предложенная стоимость
TelNumber	nvarchar	max	Телефонный номер

Таблица 4 – таблица категорий (dbo.Category)

Наименование атрибута	Тип данных	Длина поля	Описание
CategoryId	Int	4	Идентификатор

Title	nvarchar	30	Название
-------	----------	----	----------

На рисунке Рисунок 7 показаны связи между этими таблицами.

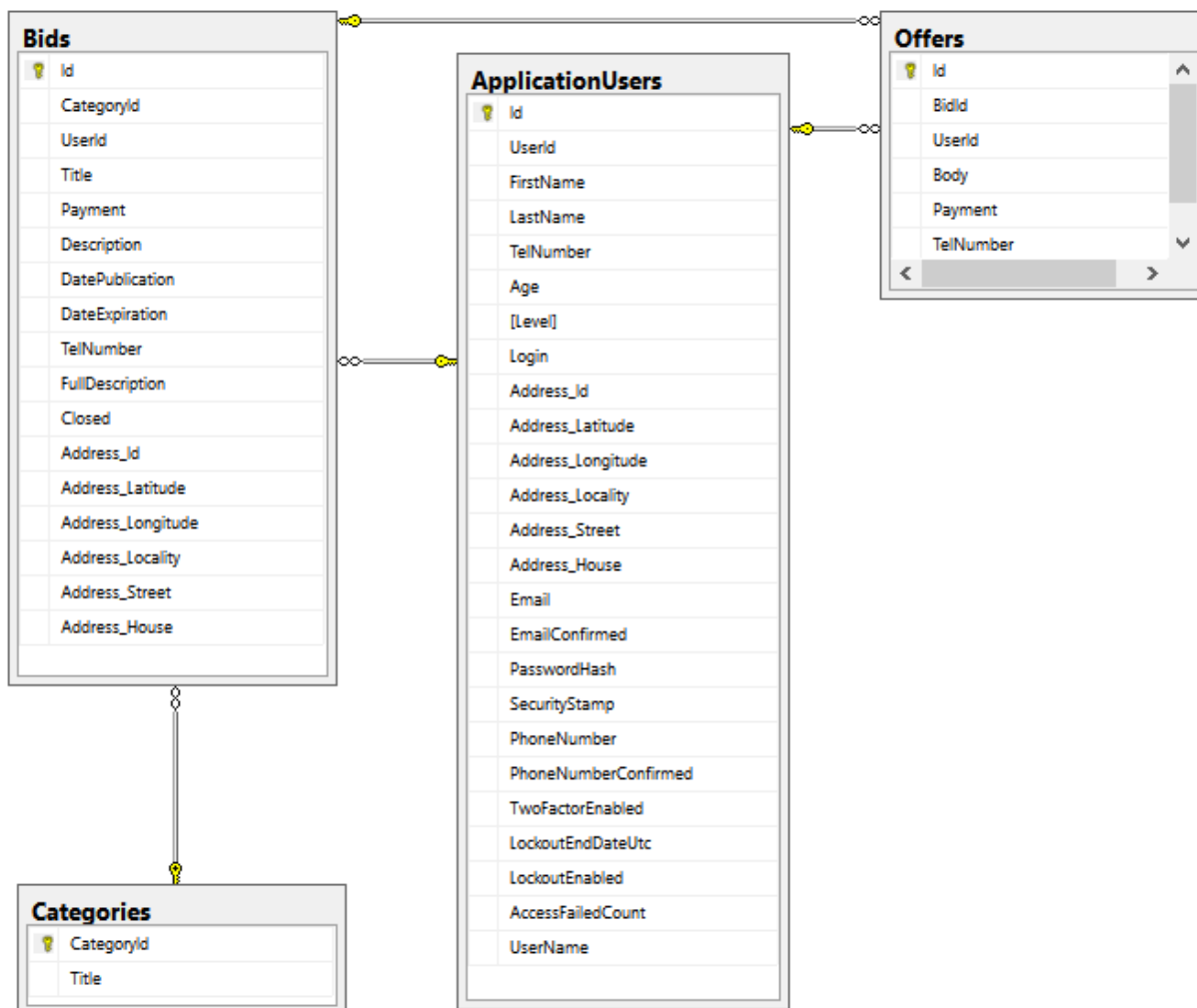


Рисунок 7 – Диаграмма базы данных

### 3.2 Серверная часть

В качестве СУБД был выбран Microsoft SQL Server.

Инфраструктура ASP.NET не имеет встроенных средств для доступа к данным, существует множество библиотек, которые могут упростить работу с базой данных [ CITATION Дже15 \l 1049 ]. Одной из них является Entity Framework – объектно-ориентированная технология доступа к данным,

созданная корпорацией Microsoft, позволяющая разрабатывать приложения избегая прямой работы с базой данных в коде. Entity Framework позволяет приложениям запрашивать и обновлять данные.

Платформа Entity Framework имеет возможность сначала создавать модель классов, и позже по ним формировать структуру базы данных. Благодаря этому мы имеем строгую типизацию и решаем проблему ручного ORM-преобразования. Этот подход имеет название **Code First**.

### 3.2.1 REST сервис

Для реализации шаблона MVC воспользуемся инструментом для создания web-сервиса – Web API. В основе Web API лежит архитектура REST (Representation State Transfer). Существенной особенностью REST является то, что эта технология задействует HTTP-запросы для создания, обновления, считывания и удаления данных. REST представляет собой облегченный подход к веб-службам [10, стр. 270].

В Web API нет прямых маршрутов к ресурсам. Все правила маршрутизации находятся в специальном файле WebApiConfig.cs:

```
public static class WebApiConfig
{
    public static void Register(HttpConfiguration config)
    {
        config.Routes.MapHttpRoute(
            name: "LazyToDoApi",
            routeTemplate: "api/{controller}/{id}",
            defaults: new { id = RouteParameter.Optional }
        );
    }
}
```

На примере показан маршрут, где во втором параметре задан шаблон маршрутизации. В котором обязательный параметр контроллер и не обязательный – некоторый id. Например, api/bid/id. Сначала будет маршрутизация по контроллеру, то есть, обратится к контроллеру BidController, затем будет передан в него заданный идентификатор id. Причем фреймворк передаст его в соответствии с методом HTTP.

Также можно вызвать метод `MapHttpAttributeRoutes()`, который просмотрит все действия контроллеров.

ASP.NET Web API 2 поддерживает определения маршрутов введением `RoutePrefix` и `Route` атрибутов, в параметрах которых указывается URL.

Например:

```
// GET: api/bid/all
[HttpGet]
[Route("all")]
[ResponseType(typeof(BidsDto))]
public HttpResponseMessage GetBids([FromUri]GetBidsUri uriData)
{
    ...
    return this.Request.CreateResponse(HttpStatusCode.BadRequest);
}
```

Для упрощения маршрутизации есть возможность использовать префиксы `RoutePrefix`:

```
[RoutePrefix("api/bid")]
public class BidApiController : ApiController
{
    // GET: api/bid/all
    [HttpGet]
    [Route("all")]
    public HttpResponseMessage GetBids([FromUri]GetBidsUri uriData)
    {
    }
}
```

Определим интерфейс запросов для REST API (таблица 5).

Таблица 5 – Интерфейс запросов в соответствии с REST

Ресурс\Метод	GET	POST
1	2	3
api/bid/all	Получить весь список объявлений	–

Окончание таблицы 5

1	2	3
/api/bid/search	Поиск по объявлениям	–
/api/bid/add	–	Добавление объявления
/api/bid/user/{id}	Получение объявлений от определенного пользователя	–
/api/bid/detail/{id}	Получение полной информации	–

	по объявлению	
/api/bid/similar	Похожие объявления	—

Подобным образом можно построить семантику запросов и для других сущностей.

Объединим классы в пакеты по принципу замыкания, с учетом шаблона MVC, рисунок Рисунок 8.

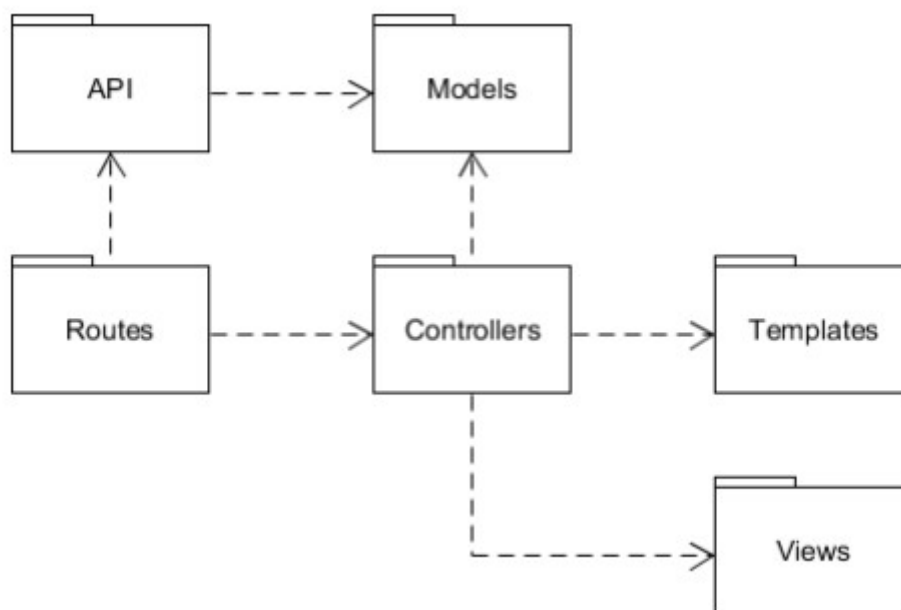


Рисунок 8 – Диаграмма пакетов Web API

Ниже, в таблицах 6, 7, 8, 9, 10, 11 описаны компоненты и входящие в них классы.

Таблица 6 – Классы компонента Routes

Класс	Описание
Server	Делегирование обработки запроса к контроллерам.
Api	Делегирование запросов к API

Таблица 7 – Классы компонента Controllers

Класс	Описание
BidApiController	API для объявлений
CategoryApiController	API для категорий
ProfileApiController	API для профиля пользователя
AccountApiController	API для действий с аккаунтом пользователя

AccountController	Контроллер для действий с аккаунтом
HomeController	Основной контроллер для действий на домашней странице

Таблица 8 – Классы компонента API

Класс	Описание
API	Обработка запросов. Возвращает ответ в формате JSON

Таблица 9 – Классы компонента Models

Класс	Описание
Bid	Объявление
Address	Адрес объявления
Offer	Предложение на выполнение объявления
User	Пользователь

Таблица 10 – Классы компонента Views

Класс	Описание
Index	Начальная страница с объявлениями
Profile	Профиль пользователя
Search	Результаты поиска
AddBid	Добавление объявления
BidDetail	Детали объявления

Таблица 11 – Классы компонента Templates

Класс	Описание
ConfirmEmail	Страница подтверждения Email
Registration	Страница регистрации
Login	Вход на сайт
Error	Форма ошибки
Help	Страница помощи
Landing	Лендинг страница

На рисунке Рисунок 9 показано поведение, когда Router делегирует обработку запроса контроллеру или API, по заданному маршруту.

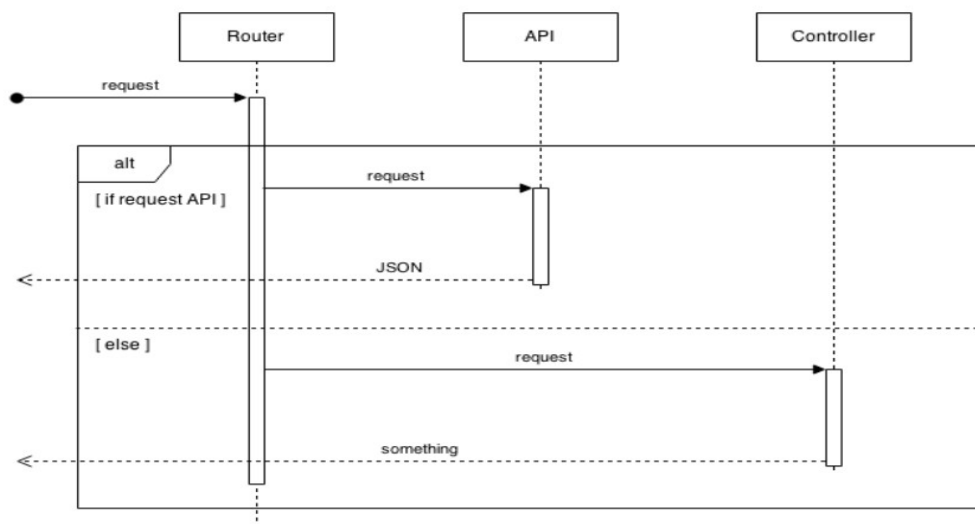


Рисунок 9 – Выбор ответственного за обработку запроса

### 3.3 Клиентская часть

Клиентская часть в основном состоит из HTML, CSS и JavaScript. В нашем случае, она содержит часть бизнес-логики и функциональный пользовательский интерфейс, которые позволяют получать, отправлять и обрабатывать данные с сервера.

В нашем архитектурном решении есть особенность клиентской части, то что ее разработка ведется вне зависимости от серверной и лишь связана с ней маршрутами запросов данных. Изначально HTML загружается без данных, и лишь после запрашиваются данные с сервера.

В качестве основы клиентской части был взят JavaScript фреймворк Knockout.js, который делает разработку клиентской логики более легкой. Knockout.js создает нужный JavaScript код за разработчика. Также он избавляет нас от рутинной части, связанной с обращением клиента к серверным методам и передачей модели для обработки [ CITATION Kno12 \l 1049 ].

## ЗАКЛЮЧЕНИЕ

Целью написания данной работы являлась разработка веб-приложения для поиска исполнителей бизнес-задач. Использование этого приложения гораздо облегчает поиск исполнителей, также имея возможность заработка для другой стороны.

Социально-практическая важность этой работы заключается в том, что аналоги существуют только в двух городах России и не содержат все преимущества каждой из них. Периферийные регионы больше нуждаются в данных услугах, так как меньше развита сфера услуг и люди больше имеют частные владения, где возникает большое количество задач, для которых нужны исполнители.

В ходе разработки были получены навыки:

- анализа и проектирования;
- создания одностраничных веб-приложений;
- проектирования баз данных;

В дальнейшем планируется сопровождение этого веб-приложения. Будут привязаны платежные и идентификационные службы, для безопасности пользования данным веб-сервисом.

Поставленные задачи работы успешно выполнены, цель достигнута.



## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Основы безопасности [Электронный ресурс] // Yandex.ru : yandex.ru – [Б. м. : Б. н.], [20–]. – Режим доступа: <https://yandex.ru/support/webmaster/protecting-sites/basics.xml?lang=ru>
2. Хилайер С. Программирование Active Server Pages / С. Хилайер, Д. Мизик. – М. : Русская редакция, 1999. – 280 с.
3. Чедвик Д. Т. ASP.NET MVC 4. Разработка реальных веб-приложений с помощью ASP.NET MVC. / Д.Т. Чедвик, Х. Панда. - М. : Вильямс, 2013. – 432 с.
4. JSON [Электронный ресурс] // Introducing Json : json.org – [Б. м. : Б. н.], [20–]. – Режим доступа: <http://www.json.org/json-ru.html>
5. Введение в Microsoft.NET для начинающих [Электронный ресурс] // Realcoding : www.realcoding.net. – [Б. м. : Б. н.], [2003-2012]. Режим доступа: <http://www.realcoding.net/article/view/2873>
6. Пачеко К. Delphi for.NET. Руководство разработчика. / К. Пачеко. – М. – Вильямс, 2005. – 960 с.
7. Макдональд М. ASP.NET. / М. Макдональд - М. СПб. : БХВ-Петербург, 2003 – 885 с.
8. С# для профессионалов. / С. Робинсон, О. Корнес, Д. Глинн, Б. Харвей, К. Макквин, Д. Моемека. – М. : Лори – 2003. – 478 с.
9. Java основы Web-служб. / Г. Бекет, М. Куннумпурат, Ш. Роди, А. Тост. – М. : Кудиц-образ – 2004. – 464 с.
10. Вайсфельд М. Объектно-ориентированное мышление. / М. Вайсфельд. – СПб. : Питер – 2004. – 304 с.
11. Knockout MVC – Сила Knockout.js для ASP.NET MVC [Электронный ресурс] // Habrahabr : Habrahabr.ru. – [Б. м. : Б. н.], [2012]. Режим доступа: <https://habrahabr.ru/company/enterra/blog/145903>
12. Разработка web-сервиса «Выбор музыкального инструмента» [Электронный ресурс] // Allbest : allbest.ru. – [Б. м. : Б. н.], [2012]. Режим

доступа:

<http://knowledge.allbest.ru/programming/3c0a65625a3ac68a4c43b89421206c26.html>

13. Мобильное приложение HTML5: ошибка или успех [Электронный ресурс] // Habrahabr : habrahabr.ru. – [Б. м. : Б. н.], [2012]. Режим доступа: <https://habrahabr.ru/post/185316/>

14. Виртуальные социальные сети как феномен информационного общества [Электронный ресурс] // Cyberleninka : cyberleninka.ru. – [Б. м. : Б. н.], [2014]. Режим доступа: <http://cyberleninka.ru/article/n/virtualnye-sotsialnye-seti-kak-fenomen-informatsionnogo-obschestva>

# ПРИЛОЖЕНИЕ А

(обязательное)

## Скриншоты пользовательского интерфейса

The screenshot displays a web application interface for a service marketplace. At the top, there are navigation links: "ГЛАВНАЯ" (Home), "ПОМОЩЬ" (Help), "РЕГИСТРАЦИЯ" (Registration), and "ВХОД" (Login). A search bar with the placeholder "Поиск..." and a button "Искать" (Search) is located at the top center. To the right of the search bar is a button "ДОБАВИТЬ ОБЪЯВЛЕНИЕ" (Add Advertisement). Below the search bar is a map of Saransk with several location pins. To the left of the map is a vertical menu with categories: "ВСЕ" (All), "КУРЬЕРСКАЯ РАБОТА" (Courier Work), "УБОРКА" (Cleaning), "РЕМОНТ" (Repair), "ДРУГОЕ" (Other), "СБОРКА МЕБЕЛИ" (Furniture Assembly), "СТРОИТЕЛЬСТВО" (Construction), "ВЫВОЗ МУСОРА" (Waste Removal), and "ПРОЧЕЕ" (Miscellaneous). The main content area displays a list of six job listings, each with a title, description, location, and price:

Task	Price (руб.)
Выкопать картошку Выкопать картошку за моим домом	1500
Разгрузить Разгрузить мешки из КАМАЗ	500
Покрасить Покрасить забор перед домом	2300
Отвезти Отвезти на автобус упаковку коробок для пиццы	500
Привезти товар Забрать и привезти товар	2000
Забрать груз СРОЧНО! Забрать пакет 7кг замороженного корма для птиц	3000

At the bottom of the listings, there are navigation buttons: "1", "2", and "Следующая" (Next). The footer contains links: "Домой" (Home), "О нас" (About Us), "Помощь" (Help), "Подать объявление" (Post Advertisement), "Реклама на сайте" (Advertising on Site), "API", "Пользовательское соглашение" (Terms of Service), and "Контакты" (Contacts). Copyright information: "© Zverit 2015. Все права защищены." (All rights reserved.)

Рисунок А1 – Список всех доступных объявлений

# Продолжение ПРИЛОЖЕНИЯ А

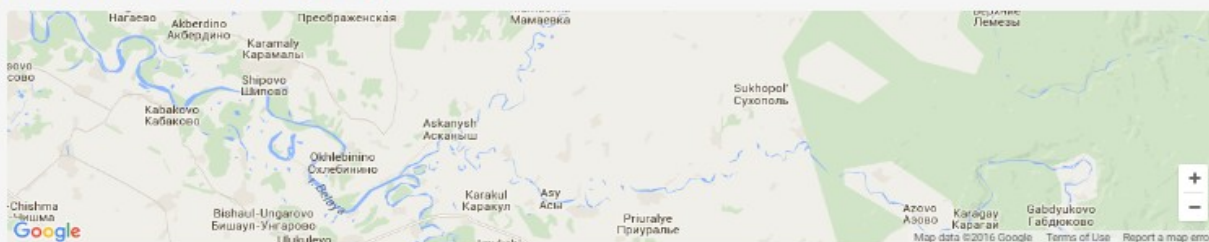


## Разгрузить

Категория: УБОРКА

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Sed ut perspiciatis, unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam eaque ipsa, quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt, explicabo. Nemo enim ipsam voluptatem, quia voluptas sit, aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos, qui ratione voluptatem sequi nesciunt, neque porro quisquam est, qui dolorem ipsum, quia dolor sit, amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incidunt, ut labore et dolore magnam aliquam quaerat voluptatem.

Плата за выполненную работу: **500 руб.**



О ПРЕДЛОЖЕНИЕ ДЕЙСТВУЕТ ДО: 2018.08.12

Автор: Odinn

Дата публикации  
22.02.2016 0:00:00

### Откликнулись

- Zverit**  
Lor34em ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua  
89091234456 **456 руб.**
- Odinn**  
Lo44rem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua  
89091234456 **456 руб.**

### Похожие объявления

- Доставить футболки**  
Доставить футболки от м.Динамо до м. Алесеевская **2200 руб.**
- Привезти товар**  
Забрать и привезти товар **2000 руб.**
- Подработка** **2300 руб.**

Рисунок А2 – Подробная информация по выбранному объявлению

## Продолжение ПРИЛОЖЕНИЯ А

### Регистрация

Логин	<input type="text" value="Логин"/>
Email	<input type="text" value="Email"/>
Пароль	<input type="password" value="Пароль"/>
Повторите пароль	<input type="password" value="Повторите пароль"/>

[ЗАРЕГИСТРИРОВАТЬСЯ](#)

Рисунок А3 – Страница регистрации

### Вход

<input type="text" value="Email"/>
<input type="password" value="Пароль"/>

Запомнить?

[Войти](#)

[Ещё не зарегистрированы? Регистрация](#)

Рисунок А4 – Страница входа

# Окончание ПРИЛОЖЕНИЯ А



[ГЛАВНАЯ](#)

[ПОМОЩЬ](#)

[РЕГИСТРАЦИЯ](#)

[ВХОД](#)

Результаты по запросу: в

Выкопать картошку Выкопать картошку за моим домом Саранск	1500 руб.
Отвезти Отвезти на автобус упаковку коробок для пиццы Саранск	500 руб.
Привезти товар Забрать и привезти товар Саранск	2000 руб.
Заменить электропроводку Заменить электропроводку, розетки, выключатели на площади 50 м2 Саранск	5000 руб.
Доставить футболки Доставить футболки от м.Динамо до м. Алесеевская Саранск	2200 руб.
Нужен водитель Перевезти мебель Саранск	5600 руб.

[1](#) [2](#) [3](#) [Следующая](#)

Рисунок А5 – Страница поиска объявлений

# ПРИЛОЖЕНИЕ Б

## (обязательное)

### Подробное описание API

#### API LazyToDo

##### Introduction

Provide a general description of your APIs here.

##### CategoryApi

API	Description
<a href="#">GET api/category/all</a>	No documentation available.

##### BidApi

API	Description
<a href="#">GET api/bid/all?Page={Page}&amp;PageSize={PageSize}&amp;CategoryId={CategoryId}</a>	No documentation available.
<a href="#">GET api/bid/search?Page={Page}&amp;PageSize={PageSize}&amp;SearchString={SearchString}</a>	No documentation available.
<a href="#">PUT api/bid/add</a>	No documentation available.
<a href="#">GET api/bid/user/{id}</a>	No documentation available.
<a href="#">GET api/bid/detail/{id}</a>	No documentation available.
<a href="#">GET api/bid/similar?Size={Size}</a>	No documentation available.

##### AccountApi

API	Description
<a href="#">POST api/account/registration</a>	No documentation available.
<a href="#">GET api/account/user/current</a>	No documentation available.
<a href="#">POST api/account/login</a>	No documentation available.

Рисунок Б1 – Список всех конечных точек API

## Окончание ПРИЛОЖЕНИЯ Б

### GET api/category/all

#### Request Information

##### URI Parameters

None.

##### Body Parameters

None.

#### Response Information

##### Resource Description

Collection of Category

Name	Description	Type	Additional information
CategoryId		integer	None.
Title		string	None.

#### Response Formats

##### application/json, text/json

Sample:

```
[
  {
    "CategoryId": 1,
    "Title": "sample string 2"
  },
  {
    "CategoryId": 1,
    "Title": "sample string 2"
  }
]
```

##### text/html

Sample:

```
[{"CategoryId":1,"title":"sample string 2"},{"CategoryId":1,"title":"sample string 2"}]
```

Рисунок Б2 – Детальное описание конечной точки API