

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ
КАЗАХСТАН

Некоммерческое акционерное общество
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ
им. ГУМАРБЕКА ДАУКЕЕВА»
Кафедра IT - инжиниринг

«ДОПУЩЕНА К ЗАЩИТЕ»
Зав. Кафедрой, PhD, доцент Досжанова А.А.
_____ «__» _____ 2020

ДИПЛОМНЫЙ ПРОЕКТ

На тему: Разработка системы микрослужбы прогнозирования веб-API

Специальность 5В060200 Информатика

Выполнила Унгарова Алиман Болатовна

Группа: ИНФ-16-2

Научный руководитель: к.т.н., доцент, Балгабаева Л.Ш.

Консультанты:

по экономической части: к.э.н профессор Габелашвили Кахабер Ревазович
(ученая степень, звание, Ф.И.О.)

_____ «__» _____ 202__ г.
(подпись)

по безопасности жизнедеятельности: ассистент Тыщенко Елена Михайловна
(ученая степень, звание, Ф.И.О.)

_____ «__» _____ 202__ г.
(подпись)

по применению вычислительной техники:

ст. преп. Майкотов Мухит Нурдаулетович

(ученая степень, звание, Ф.И.О.)

_____ «__» _____ 202__ г.
(подпись)

преп. Пипия Маргарита Торникеевна

(ученая степень, звание, Ф.И.О.)

_____ «__» _____ 202__ г.
(подпись)

Нормоконтролер: ст. преп. Абсатарова Бибигуль Рыскуловна

(ученая степень, звание, Ф.И.О.)

_____ «__» _____ 202__ г.
(подпись)

Рецензент: _____

(ученая степень, звание, Ф.И.О.)

_____ «__» _____ 202__ г.
(подпись)

Алматы 2020

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ
КАЗАХСТАН

Некоммерческое акционерное общество
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ
им. ГУМАРБЕКА ДАУКЕЕВА»

Факультет «Институт Систем Управления и Информационных
Технологий»

Кафедра «IT - Инжиниринг»

Специальность 5В060200-Информатика

ЗАДАНИЕ

на выполнение дипломного проекта

Студенту Унгаровой Алиман Болатовне

Тема проекта: Разработка системы микрослужбы прогнозирования веб-API

Утверждена приказом по университету № ___ от «___» _____ 2020 г.

Срок сдачи законченного проекта «___» _____ 2020 г.

Исходные данные к проекту (требуемые параметры результатов исследования (проектирования) и исходные данные объекта): Spyder- среда разработки, VS Code – редактор кода.

Перечень вопросов, подлежащих разработке в дипломном проекте, или краткое содержание дипломного проекта:

- а) анализ микрослужб и возможностей их применения;
- б) разработка системы микрослужб пригнозирования;
- в) возможности эксплуатации;
- г) экономическая эффективность проекта;
- д) проектирование рабочего места оператора.

Перечень графического материала (с точным указанием обязательных чертежей): Визуальное представление принципа работы перцептрона, Визуальное представление взаимодействия слоёв в нейронной сети, Схема офисного помещения, Схема подключения оборудования, Диаграмма планируемого процесса работы системы.

Основная рекомендуемая литература:

1 Ричардсон К. Микросервисы. Паттерны разработки и рефакторинга. - М.: Питер, 2019. - 44 с.

2 Ньюмен С. Создание микросервисов - М.: Питер, 2016 – 78 с..

3 Рассел С. Искусственный интеллект: современный подход. - М.: Вильямс, 2016. - 578 с.

4 Лутц М. Программирование на Python, I том. - М.: Символ-плюс, 2015. - 992 с.

Консультация по проекту с указанием относящихся к ним разделов проекта

Раздел	Консультант	Сроки	Подпись
Экономическая часть	Габелашвили К.Р.	27.04.2020	
Безопасности жизнедеятельности	Тыщенко Е.М	30.04.2020	
Программная часть	Майкотов М.Н.	14.05.2020	
Нормоконтролер	Абсатарова Б.Р.	18.05.2020	

График
подготовки дипломного проекта

Наименование разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю	Примечания
Аналитическая часть	13.01.2020- 19.02.2020	
Проектная часть	20.01.2020- 22.02.2020	
Эксплуатационная часть	20.01.2020- 01.03.2020	
Экономическое обоснование	02.03.2020- 27.04.2020	
Безопасность жизнедеятельности	02.03.2020- 30.04.2020	

Дата выдачи задания «___» _____ 2020 г.

Заведующий кафедрой _____ Досжанова А.А.
(подпись) (Ф.И.О.)

Руководитель работы
(проекта) _____ Балгабаева Л.Ш.
(подпись) (Ф.И.О.)

Задание принял к исполнению
студент _____ Унгарова А.Б.
(подпись)

(Ф.И.О.)

Аңдатпа

Экономикалық цифровизация шартында және көтерме саудының электрондық коммерцияға аусуы барысында, уақытты үнемдеуге және де қызметкер еңбегінің тиімділігін күшейтуге әкелетін кез-келген шешім, осының бәрі бәсекеге қабілеттілігінің факторы болып табылады. Қазіргі заманғы электрондық сауда алаңдарында барлық процесстер мұқият салыстыра тексерілген және автоматтандырылған. Дегенімен, талдаудың көрсетуі бойынша,көптеген тендерлік сатып алулар біртепті сатып алуды құру сәтінде жоюға болатын себептер үшін аяқталмай қалады.

Бұл дипломдық жобада ЖШС «ЕТС-тендер» электрондық сауда алаңы үшін аяқталмай қалудың жоғары қатері бар тендерлік сатып алу сауда процедураларын анықатайтын жобалау микроқызмет жүйесін өңдеу жүргізілді. Ол үшін машиналық оқыту моделі оқытылып, жасалды. Front-end және back-end өңдеу жасалды.

Сондай-ақ осы компания үшін ДК операторының жұмыс орнын өңдеу және жобалаудың технико-экономикалық негіздеме жасалды.

Аннотация

В условиях цифровизации экономики и перехода оптовой торговли в электронную коммерцию, любые решения, приводящее к экономии времени и повышению эффективности процесса труда сотрудника, являются фактором повышения конкурентоспособности. На современных электронных торговых площадках все процессы тщательно выверены и автоматизированы. Тем не менее, анализ показал, что многие тендерные закупки не завершаются из-за ряда однотипных причин, которые можно было бы исключить на моменте создания закупки.

В данном дипломном проекте была проведена разработка системы микрослужб прогнозирования, которая выявляет торговые процедуры с высоким риском незавершенности процедуры тендерной закупки для электронной торговой площадки ТОО «ЕТС-Тендер». Для этого была создана и обучена модель машинного обучения, была проведена front-end и back-end разработка.

Также сделано технико-экономическое обоснование разработки системы и проектирование рабочего места оператора ПК для этой компании.

Annotation

In terms of digitization of the economy and the transition to wholesale trade in e-commerce, any decision that leads to time savings and increase the efficiency of the process of employee labor, there is a factor in increasing competitiveness. On modern electronic trading platforms, all processes are thoroughly verified and automated. Never the less, the analysis showed that many tender purchases are not completed due to a number of similar reasons that could have been excluded at the time of purchase.

This diploma project included the development of a system of prediction microservices, which identifies trade procedures with high risk of non-compliance with the tender procurement procedure for the electronic trading platform of ETS-Tender LLP. For this purpose, was created and trained a model of machine learning, was conducted front-end and back-end development.

Also, the feasibility study of the system development and design of the working place of the PC operator for this company was made.

Содержание

Введение	9
1 Аналитическая часть	12
1.1 Понятие микрослужб и их особенностей	12
1.1.1 Разбиение через сервисы	14
1.1.2 Организация вокруг потребностей бизнеса	15
1.2 Примеры зарубежных сервисов микрослужбы	16
1.3 Характеристика деятельности компании	17
1.4 Постановка задачи	20
2 Проектная часть	22
2.1 Выбор микрослужб и проектирование системы	22
2.2 Выбор инструментов для создания системы	24
2.2.1 Анализ современных языков программирования для работы с машинным обучением	24
2.2.2 Выбор языков программирования Python и JavaScript	25
2.2.3 Выбор среды разработки и редактора кода	26
2.2.3.1 Возможности Visual Studio Code	28
2.3 Разработка микрослужб прогнозирования системы	30
2.3.1 Создание микрослужбы «Модель»	30
2.3.2 Описание работы главных элементов микрослужбы «API»	32
2.3.3 Описание микрослужбы «Frontend»	34
3 Эксплуатационная часть	36
3.1 Интерфейс	36
3.2 Алгоритм действий для работы с системой	37
3.3 Системные требования	38
4 Экономическая часть	40
4.1 Техничко-экономическое обоснование проекта	40
4.2 Расчет трудоёмкости разработки системы	40
4.3.2 Затраты на электроэнергию	42
4.3.3 Затраты на оплату труда	43
4.3.4 Социальные налоги	44
4.4 Расчет результатов от создания и использования Системы	45
5. Безопасность жизнедеятельности	49
5.1 Охрана труда в ТОО «ЕТС-Тендер»	49
5.1.1 Риски для безопасной жизнедеятельности человека в помещении (среде)	49
5.1.2 Соответствие Требованиям микроклимата помещения	509
5.1.3 Соответствие Требованиям к освещению помещения	509
5.1.4 Опасность поражения электрическим током	521
5.1.5 Опасность возникновения пожара	531
5.1.6 Риски для безопасной жизнедеятельности человека, связанные с компьютером (машиной)	532

5.1.7 Соответствие Требованиям по нормам излучения электромагнитных волн, акустического шума и концентрации вредных веществ в воздухе	543
5.1.8 Меры, принимаемые сотрудниками для безопасной жизнедеятельности человека и охраны его труда (человек).....	54
5.2 Проектирование рабочего места оператора ТОО «ЕТС-Тендер»	554
5.2.1 Соответствие Требованиям размещения и эксплуатации ПК	554
5.2.2 Выбор офисной мебели.....	55
5.2.3 Выбор оборудования.....	56
Заключение.....	58
Список литературы.....	609
Приложение А.....	61
Приложение Б.....	665
Приложение В.....	709

Введение

Сегодня во всем мире рынок розничной электронной коммерции растет опережающими темпами, вытесняя традиционную торговлю. Чтобы добиться конкурентоспособности в глобальной экономике три года назад Казахстан стал более активно развивать цифровую инфраструктуру на государственном уровне. Это стало еще одним стимулом для расширения деятельности коммерческих электронных торговых площадок в Казахстане. Спрос на электронные услуги растет очень быстро. Не только крупные компании стремятся снизить свои издержки за счет использования современных цифровых технологий, но и малый и средний бизнес находят заказы с помощью электронных торговых площадок.

По итогам 2019 года мировой объем розничной электронной торговли превысил \$3,5 трлн без учета продаж билетов на авиаперевозки и билетов на мероприятия. Последние несколько лет темпы прироста этого показателя превышают 20%. Лидерами по объему интернет торговли являются Китай и США, с общей долей 71,3% от мирового рынка [15].

Объем рынка электронной торговли Казахстана только за 2019 год увеличился в 1,8 раз и превысил 700 млрд тенге. Ее доля в общей розничной торговле добавила почти 1%-й пункт и составила 3,7%. Это говорит о существенном потенциале роста. Электронные торговые площадки наряду с интернет-магазинами, занимают эту нишу.

Таким образом, очевидно, что электронная торговля выделяется на фоне других сфер человеческой деятельности. Торговля стимулирует создание отечественных производств, которые легко находят сбыт внутри Казахстана, а также обеспечивают сами себя необходимыми для них услугами и материалами по самым выгодным ценам, с необходимыми сроками поставки, в течение заданного периода времени.

В республике работает более двух десятков электронных торговых платформ. Кроме того, существует примерно столько же агрегаторов электронных торговых компаний, цель которых собрать со всех площадок информацию и в удобном виде предоставить ее потребителям – заказчикам и поставщикам.

Изначально торговые площадки создавались как тендерные платформы для государственных и частных холдингов. Но сейчас сюда вовлечены буквально все компании, которые нацелены на быстрый рост и стабильную долгосрочную работу. Благодаря электронной торговле за счет снижения издержек и конкурентного выбора, компании могут сэкономить до 20% своих средств, направленных на закупки. Именно в этом и заключается на сегодня интерес к данному рынку. По сути, с рынка уходят лишняя цепочка посредников, которая как раз и жила за счет маржи.

Для государства большой плюс от электронной торговли в том, что уходят все теневые обороты, растут налоговые поступления.

На казахстанском рынке в тройку самых крупных компаний в секторе электронной коммерции входит ТОО «ЕТС-Тендер». Компания начала свою деятельность в июле 2017 года как проект АО «Товарная биржа «ЕТС». Уже через год начинанию решили придать форму собственности в виде ТОО. За два года новая компания смогла войти в тройку с самым высоким оборотом, благодаря современным технологиям и методикам, которые были предоставлены одним из акционеров: АО «Центр развития экономики» (РФ) – с торговой маркой B2B-Center.

Со своей стороны, компания ТОО «ЕТС-Тендер» проанализировав два с половиной года своей успешной деятельности, пришла к выводу, что часть создаваемых заявок Заказчика не доходит до своего логического завершения – сделка так и не происходит, несмотря на то, что на нее ушли усилия трех компаний – Заказчика, Поставщика и ТОО «ЕТС-Тендер», на площадке которой клиенты нашли друг друга. Такая ситуация характерна для работы всех электронных площадок. В открытом доступе нет информации, осознают ли конкуренты и смогли ли ее решить с помощью современных цифровых технологий. Поэтому компания решила сама найти выход.

Анализ проблемы показал, что ее можно решить с помощью создания микрослужбы. Суть микрослужбы – на этапе создания заявок, зная ее основные данные, понять, будет ли процесс закупки завершен или нет. Исключая на этапе создания некачественные заявки, компания ТОО «ЕТС-Тендер» экономит время и труд сотрудников, зарабатывает, используя сохраненные часы на создание качественных заявок.

Предметом анализа стали причины, по которой значительное число заявок, по которым начинается процедура совершения сделки, так и не завершается, а затем устранение этой причины. В результате внедрения микрослужбы повысится эффективность деятельности электронной торговли. Практическая ценность заключается в существенном повышении объема обрабатываемых заявок, по которым процесс заключения сделки завершается. При этом оценка качества создаваемой заявки проводится за считанные минуты. Создание микрослужбы ведет к существенному повышению конкурентоспособности компании и лояльности клиентов к ней. Ведь тендерная процедура занимает в среднем семь-десять дней.

В процессе написания дипломной работы применены методы: системного подхода к объекту и предмету дипломной работы, анализа и синтеза, сравнения, обобщения, абстракции, конкретизации темы дипломной работы с помощью практического примера, анализа теоретических и практических источников, анализа нормативно-правовой основы темы дипломной работы, что характеризует дипломную работу с точки зрения целостности, достоверности, всеобщности и полноты. Работа графически оформлена с использованием иллюстраций. Анализ языков программирования и технических возможностей создания системы микрослужбы позволили добиться обоснованных и достоверных практических результатов.

В роли статистической информационной базы были использованы, аналитические статьи на специализированных и информативных англоязычных сайтах, поскольку именно там собрана самая современная и полная информация.

Структура дипломной работы состоит из введения, пяти глав, заключения, списка литературы и приложений. Введение включает актуальность, предмет, объект, цель, задачи, методику, и практическую значимость дипломной работы. Первая глава посвящена анализу предметной области с использованием определений, в ней раскрыты различные понятия, проведен сопоставительный анализ существующих систем. Во второй главе осуществляется обоснованность выбора инструмента разработки. Третья глава содержит описание программной реализации. В заключении отмечаются основные результаты дипломной работы, излагаются определенные выводы по теме.

1 Аналитическая часть

1.1 Понятие микрослужб и их особенностей

Для решения проблемы ТОО «ЕТС-Тендер» необходимо создать элемент к общему сложному массиву уже существующей системы электронной торговой площадки. Лучшим способом для этого может стать создание микрослужбы через архитектурный стиль микрослужб (микросервисов).

Архитектурный стиль микрослужб - это метод, при котором единое приложение строится как набор небольших сервисов, каждый из которых работает в собственном процессе и коммуницирует с остальными, используя стандартные легковесные протоколы, как правило HTTP [1].

Микрослужба - это независимая программа, которая выполняет определенные функции и взаимодействует с другими приложениями через определенный интерфейс и с помощью общепринятых протоколов.

Современной технологией разработки сложных программных систем является микросервисный подход через создание микрослужб. В ее основе лежит сервис-ориентированный подход — это одна из основных концепций построения распределенных информационных систем различных областей применения. Эти сервисы построены вокруг бизнес-потребностей и развертываются независимо, с использованием полностью автоматизированной среды. Существует абсолютный минимум централизованного управления этими сервисами. Сами по себе эти сервисы могут быть написаны на разных языках и использовать разные технологии хранения данных.

Суть в том, что микрослужба принимается как самостоятельная программная единица с определенным интерфейсом, практически не связанная с внешней средой. Микрослужба может повторно использоваться, при этом она возвращается к исходному состоянию, что позволяет применять в различных контекстах много раз. В этом основное отличие концепции микросервиса от веб-сервиса. Веб-сервис сохраняет все изменения и его проблематично использовать повторно для совершенно иного контекста.

Ключевая идея – независимость от окружающей среды, слабая с ней связанность и высокий уровень повторного использования. Микрослужба может использоваться повторно для решения разных задач и в разных контекстах. Поэтому система, построенная на микрослужбах, удобна для постепенного последовательного внедрения, масштабирования и высоких нагрузок. Это позволяет создавать комплекс микрослужб параллельно несколькими командами разработчиков. При этом программирование не начинается «с нуля», а используются уже имеющиеся части [14].

Модификация алгоритма работы программы также осуществляется изменением схемы передачи и преобразования данных между

существующими микросервисами. Поэтому прогноз функциональных и нефункциональных аспектов работы при такой сборке системы становится важным, так как часто микрослужбы используются именно для использования в веб-программировании на сайтах, через интернет-среду, где нагрузка с каждым годом растет. Поэтому создание комплексов из микрослужб, использующих доступные ресурсы и готовых к масштабированию – это актуальная задача.

Технология создания комплексов из микрослужб является современной реализацией принципов сервис-ориентированного программирования (СОП) и сервис-ориентированной архитектуры (СОА), которые развиваются последние двадцать лет. Впервые принципы СОП были воплощены в концепции, называемой «веб-сервисы». Они были удачно совмещены с методологией управления бизнеспроцессами (BPM, business process management), которая пришла на смену реинжинирингу бизнес-процессов. В СОА сервисы стали технологической основой процессов, а процессы – организационным ядром сервисов. Аналогичным образом сервисная, технологическая модель предприятия отображалась на процессную. Комплекс из микрослужб, с организационной точки зрения, выполняет бизнес-процессы предприятия. А это значит, что может быть разделена на отдельные бизнес-процессы, как, впрочем, и процессная модель. Каждая микрослужба для выполнения своей работы может «вызывать» другие микрослужбы.

Для того, чтобы объяснить стиль микросервисов, лучше всего сравнить его с монолитом (monolithic style): приложением, построенном как единое целое. Enterprise приложения часто включают три основные части: сервер, пользовательский интерфейс (состоящий как правило из HTML страниц и javascript) и базу данных (как правило реляционной, со множеством таблиц). Серверная часть обрабатывает HTTP запросы, выполняет доменную логику, запрашивает и обновляет данные в БД, заполняет HTML страницы, которые затем отправляются браузеру клиента. Любое изменение в системе приводит к пересборке и развертыванию новой версии серверной части приложения [2].

В монолитном сервере вся логика по обработке запросов выполняется в единственном процессе. При этом используются возможности изначально выбранного языка программирования для разделения приложения на классы, функции и namespace. Это позволяет запускать и тестировать приложение на машине разработчика и использовать стандартный процесс развертывания для проверки изменений перед выкладыванием их в продакшн. Также такой сервер позволяет масштабировать монолитное приложения горизонтально, путем запуска нескольких физических серверов за балансировщиком нагрузки.

Монолитные приложения до последнего времени вполне устраивали разработчиков и пользователей. Но сейчас все больше приложений развертываются в облаке. Любые изменения, даже самые небольшие,

требуют пересборки и развертывания всего монолита. С течением времени, становится труднее сохранять хорошую модульную структуру, изменения логики одного модуля имеют тенденцию влиять на код других модулей. Масштабировать придется все приложение целиком, даже если это требуется только для одного модуля этого приложения. Весь этот трудоемкий процесс и разочаровывает его создателей.

Эти неудобства привели к созданию и более широкому применению архитектурного стиля микрослужбам. Кроме того, построение приложений в виде набора сервисов и возможности независимого развертывания и масштабирования, позволяет видеть четкую физическую границу каждого сервиса. Это позволяет разным микрослужбам быть написанными на разных языках программирования. Они также могут разрабатываться разными командами. Микрослужбы это не инновация. Его корни уходят далеко в прошлое, к принципам проектирования, использованным в Unix. Но преимущества его очевидны.

Какими основными свойствами обладают микрослужбы: разбиение через сервисы, организация вокруг потребностей бизнеса, малый объем и возможность создания малым числом разработчиков.

1.1.1 Разбиение через сервисы

Архитектура микросервисов использует библиотеки, при этом библиотека представляет собой набор компонент, которые подключаются к программе и вызываются ею в том же процессе, в то время как сервисы - это компоненты, выполняемые в отдельном процессе и коммуницирующие между собой через веб-запросы или *remote procedure call (RPC)*. При этом под компонентами здесь понимается единица программного обеспечения, которая может быть независимо заменена или обновлена.

Независимое развертывание – главное достоинство библиотеки, состоящей из компонент. В приложении, состоящем из нескольких библиотек, работающих в одном процессе, любое изменение приводит к переразвертыванию всего приложения. Но если приложение разбито на несколько сервисов, то изменения, затрагивающие какой-либо из них, потребуют переразвертывания только изменившегося сервиса. Конечно, какие-то изменения будут затрагивать также и интерфейсы, что, в свою очередь, потребует некоторой координации между разными сервисами. Но при созданной хорошей архитектуре микросервисов минимизирует такую координацию путем установки правильных границ между микросервисами, а также механизма эволюции сервисов.

Более явный интерфейс между компонентами помогает предотвратить нарушение их инкапсуляции из-за того, что большинство языков программирования не имеют хорошего механизма для объявления Published Interface, а значит и создания четкой документации и дисциплины.

Микрослужбы же позволяют избежать этого через использование явного механизма удаленных вызовов.

Использование сервисов подобным образом, конечно, имеет свои недостатки: удаленные вызовы работают медленнее, чем вызовы в рамках процесса, и поэтому API должен быть менее детализированным (coarse-grained), что часто приводит к неудобству в использовании. Если нужно изменить набор ответственностей между компонентами, сделать это сложнее из-за того, что нужно пересекать границы процессов.

В первом приближении можно наблюдать, что сервисы соотносятся с процессами как один к одному. На самом деле сервис может содержать множество процессов, которые всегда будут разрабатываться и развертываться совместно. Например, процесс приложения и процесс базы данных, которую использует только это приложение.

1.1.2 Организация вокруг потребностей бизнеса

При создании приложения его разбивают на части в соответствии с применяемыми технологиями. Это приводит к образованию UI команды, серверной команды и БД команды. Даже небольшие изменения в одном из направлений, отнимают много времени из-за необходимости кросс-командного взаимодействия. Это приводит к тому, что команды размещают любую логику на тех слоях, к которым имеют доступ.

Микросервисный подход к разбиению подразумевает разбиение на службы в соответствии с потребностями бизнеса. Такие службы включают в себя полный набор технологий, необходимых для этой бизнес-потребности, в том числе пользовательский интерфейс, хранилище данных и любые внешние взаимодействия. Это приводит к формированию кросс-функциональных команд, имеющих полный набор необходимых навыков: user-experience, базы данных и project management.

Кросс-функциональные команды отвечают за построение и функционирование каждого продукта и каждый продукт разбит на несколько отдельных сервисов, общающихся между собой через шину сообщений.

Крупные монолитные приложения тоже могут быть разбиты на модули вокруг бизнес-потребностей, хотя обычно этого не происходит. Основная проблема здесь в том, что такие приложения имеют тенденцию к организации вокруг слишком большого количества контекстов. Если монолит охватывает множество контекстов, отдельным членам команд становится слишком сложно работать с ними из-за их большого размера. Кроме того, соблюдение модульных границ в монолитном приложении требует существенной дисциплины. Явно очерченные границы компонент микрослужб упрощает поддержку этих границ.

1.2 Примеры зарубежных сервисов микрослужбы

Мониторинг открытых источников информации показал, что в Казахстане нет аналогов сервисов микрослужбы, используемых для автоматизации некоторых аспектов работы цифровых систем. Поэтому в работе приведены примеры только зарубежных сервисов микрослужб.

Так, грузоперевозчик ПЭК и компания "Неофлекс" запустили Центр управления перевозками (ЦУП) с помощью сервисов микрослужб. Он позволил компании ПЭК существенно повысить точность планирования грузоперевозок и установить новые стандарты качества работы в индустрии сборных грузов.

Центр управления перевозками, предназначенное для повышения эффективности планирования и прогнозирования перевозок, а также мониторинга остатков на складах. ЦУП компании ПЭК позволяет в режиме реального времени прогнозировать загрузку 189 складов по всей России на месяц вперед с высокой степенью точности. Департаменты транспортной и складской логистики взаимодействуют в едином информационном поле в рамках сквозного бизнес-процесса.

Технологической основой ЦУП является стек opensource технологий Big Data и Fast Data: микросервисы написаны с использованием Akka Framework, а задачи потоковой обработки данных реализованы с помощью Spark Streaming. Обмен сообщениями между компонентами реализован через топики Apache Kafka. Использование микросервисной архитектуры значительно упростило масштабирование и внедрение доработок. В архитектуре использовано три типа баз данных: оперативные данные хранятся в IMDB, для срочной отчетности используется PostgreSQL, а для хранения исторических данных кластер Hadoop [16].

Компания Genesys, мировой лидер в области решений для контакт-центров и омниканального обслуживания клиентов, объявила об увеличении продаж облачных решений компании на 32% в годовом выражении в 2019 году.

Компании, мигрировавшие на Genesys, добиваются улучшения ключевых бизнес-показателей: Coca-Cola Business Service (Северная Америка) снизила целевые и косвенные затраты от идеи до воплощения и получения выгоды от компьютерных разработок на 50%. Heineken (Мексика) повысила эффективность звонков более чем на 40%, показатель успешных продаж - на 6%, а также снизила трудозатраты в среднем на 1,5 человеко-часа в день на каждого оператора.

Одним из шагов на пути к успеху Genesys стало ускорение внедрения и миграции масштабных контакт-центров со сложной архитектурой и высокими требованиями за счет использования собственных более 100 стандартных сценариев использования, переход к стратегии DevOps и использование архитектуры на базе микросервисов позволяют клиентам Genesys быстрее внедрять новые функции и получать соответствующие

бизнес-результаты. По классификации Genesys, к нижнему сегменту среднего бизнеса относятся компании с количеством сотрудников до 3 тыс. человек; к верхнему сегменту среднего бизнеса — компании с 3–10 тыс. сотрудников; крупными считаются компании, в которых более 10 тыс. сотрудников [17].

Банковская группа Lloyds в марте 2020 года объявила о совершенствовании своей аналитической системы и средств обработки данных с последующим переводом их функционирования в режим реального времени. Это позволит эффективнее выявлять случаи мошенничества, а клиенты смогут увидеть проводки сразу после выполнения своей транзакции, благодаря усовершенствованиям в том числе и с использованием микросервисов.

Решение поставленной задачи потребует от Lloyds серьезных архитектурных изменений. Для достижения требуемой скорости выполнения операций команде Маккола придется интегрировать унаследованные системы с современными технологиями, в том числе со средствами отслеживания измененных данных (Change Data Capture, CDC) и коннекторами Kafka. Идеальным вариантом для компании стало использование лямбда-архитектуры с микросервисами, записывающими и считывающими данные перед их выгрузкой в аналитическую систему.

К инструментам, которые банк использует в своем конвейере реального времени, относятся HBase (хранение данных и их обогащение в процессе прохождения), Hive (загрузка данных для анализа), Storm (обработка в реальном времени), Spark (потребление и загрузка данных в различные системы) и Kafka (брокер сообщений). Все они развернуты непосредственно на территории банка [18].

Резюмируя все три примера, очевидно, что особенно крупным компаниям трудно уйти от устаревших решений сиюминутно, и использование архитектуры микрослужб помогают решать их проблемы, поэтапно создавать улучшения, тем самым предлагая своим клиентам максимально удобные и простые услуги.

1.3 Характеристика деятельности компании

Выбор использования архитектуры микрослужбы для ТОО «ЕТС-Тендер» связано как с особенностями самой компании, так и самого рынка электронной коммерции. В Казахстане существует более двух десятков электронных торговых площадок. Они подразделяются на несколько типов. Во-первых, это порталы, на которых крупные компании совершают закупки для осуществления своей хозяйственной деятельности. Например, АО ФНБ «Самрук-Казына» или АО «Халык Банк». Во-вторых, это ЭТП, деятельность которых заключается в том, чтобы предоставлять свою площадку для клиентов – Закупщиков и Поставщиков. Например, mp.kz, ТОО «ЕТС-

Тендер», ТОО «KazKontract». В-третьих, агрегаторы тендеров. Это компании, которые не работают напрямую с Закупщиками и Поставщиками. Они агрегируют тендеры на своей площадке уже созданных на ЭТП, описанных во втором примере. В-четвертых, это компании, которые обучают Поставщиков и Заказчиков работать на порталах всех типов. В любом случае – это технологически продвинутые компании, обладающие современными сложными с точки зрения архитектуры, системами. Дорабатывать их, вносить какие-то изменения – довольно сложная задача, поскольку любое вмешательство может повлечь волну изменений в налаженном механизме.

Таблица 1 – Перечисление электронных торговых площадок в Казахстане

Электронные торговые площадки Казахстана						
	регион	площадка	название компании	Адрес	Контактные данные	электронный адрес
1	Алматы	mp.kz	нет данных	ул. Байзакова, 280, северная башня БЦ «Almaty Towers», 22 этаж	7 (727) 330-99-77	mp@mp.kz
2		kazcontract.kz	ТОО "KazKontract"	ул.Ходжанова 58/5	7 (727) 249 7777	info@kazcontract.kz
3		etbemp.kz	ТОО «Электронная торговая площадка ЕТБ»	ул. Егизбаева 13, блок 5, офис 67	7 (727) 317-53-05	info@etbemp.kz
4		bas.kz	TOP Tender	г. Алматы, пр. Аль-Фараби, 19/1, БЦ «Нурлы Тау», блок 3Б, 7 этаж.	7 (727) 357 21 22	info@bas.kz
5	Нур-Султан	tenderbot.kz	нет данных	ул. Сарайшык 5/4	7 (717) 291 98 98	lid@tenderbot.kz
6		tenderplus.kz	нет данных	Абая 29/306	7 (727) 318 76 09	support@tenderplus.kz
7		zakupki.kz	Business Consulting Group	к. 41, БЦ «Silk Way Cent	7 (717) 257 90 00	client@zakupki.kz
8		eoz.kz	ТОО «Центр мониторинга и экспертизы рынка»	БЦ "Изумрудный квар	7 (702) 111 14 33	support@eoz.kz
9		zakup.sk.kz	АО ФНБ "Самрук-Казына"	нет данных	7 (717) 255 22 66	support2@skc.kz

Особенно это касается ТОО «ЕТС-Тендер», которое входит в тройку крупнейших компаний второго описанного типа. Генеральным директором фирмы является Ахтырская Наталья Александровна, которая смогла под своим началом сгруппировать сильную разноплановую команду в Департаменте информационных технологий. Компания имеет государственную лицензию на занятие деятельностью по электронной коммерции выданная Комитетом регистрационной службы и оказания правовой помощи Министерства юстиции.

Кроме того, следует отметить, что ТОО «ЕТС-Тендер» – совместный продукт двух лидеров России и Казахстана в области биржевой торговли и электронных закупок.

Товарная биржа ЕТС - ведущая биржевая площадка Казахстана:
- опыт работы на рынке корпоративных закупок с 2010 года;

- наибольшая капитализация среди товарных бирж.

b2b-center - крупнейшая система электронных закупок коммерческого сегмента России:

- 18 лет работы с крупнейшими отраслевыми закупщиками;

- №1 в номинации «наивысший уровень качества услуг в области проведения корпоративных закупок»;

- №1 в номинации «гибкость и широта торгово-закупочного функционала».

Таким образом, казахстанская компания интегрировала российскую систему на местный рынок. В рамках одной процедуры система объединяет проверенных заказчиков и квалифицированных поставщиков, предлагающих лучшие условия.

Взаимодействие заказчика и поставщика в системе ТОО «ЕТС-Тендер» происходит в рамках торговой процедуры. Заказчик объявляет торги на закупку, после чего участники системы автоматически получают уведомление с приглашением к участию в торгах. Поставщик может сам найти интересные ему торги в разделе «Торговая площадка», используя поиск по наименованию компании-заказчика, номеру торгов или рубрике закупаемой продукции, и подать заявку на участие в подходящих процедурах.

ТОО «ЕТС-Тендер» позволяет полностью автоматизировать процесс закупки и продажи. Обеспечивает участие в торгах крупнейших заказчиков. Позволяет наблюдать за ходом торгов и следить за результатами. Торги проходят в режиме реального времени. В результате клиенты сокращают время подачи предложений за счет автоматизации всех этапов участия в торгах. Клиенты получают бесплатную рассылку о новых торгах по интересующим тематикам. Система помогает анализировать рыночный спрос на любую категорию продукции, а также действия конкурентов.

Клиентская база поставщиков довольно обширна – 450 тысяч компаний в Казахстане и России. Свои закупки на площадке ТОО «ЕТС-Тендер» проводят больше 100 крупнейших компаний Казахстана: банки, телекоммуникационные, строительные, производственные компании и субъекты естественных монополий, а также быстроразвивающиеся компании из других секторов экономики. Перечислим некоторые из них: Группа компаний KazMinerals, АО «Казатомпром», Группа компаний «Казахстанские коммунальные системы», Компания по управлению активами «Алмалы», ТОО «КарТел» (Beeline), Кондитерская фабрика АО «Рахат», АО АК «Алтыналмас», АО «КТЖ» и многие другие. Самыми популярными категориями торгов являются закупки сырья и материалов, IT-оборудования, мебели, строительных и рекламных услуг, а также полиграфической продукции.

В распоряжении каждого сотрудника имеется персональный компьютер, с установленной операционной системой Windows 10 и вычислительной мощностью, удовлетворяющей системным требованиям

пакета приложений Microsoft Word 2019. Для ведения базы данных, задания формул и выполнения расчетов используется табличный процессор Microsoft Excel 2019. Для формирования отчетов – текстовый процессор Microsoft Word 2019. Сбор и анализ информации для клиентов происходит автоматически. Каждый клиент имеет доступ к личному кабинету, в которую он вносит необходимые ему для проведения торгов данные. В представленном интуитивно понятном интерфейсе на сайте ets-tender.kz есть возможность заполнять все необходимые и дополнительные данные.

Составление отчета происходит автоматически. Необходимо отметить, что процесс одной закупки в зависимости от специфики может занимать от одной до двух недель. Связано это с самим процессом торгов, в проведении которых необходимо подгрузить большое число бухгалтерских и юридических документов для чистоты совершения сделки.

В век цифровизации возможность своевременно анализировать и реагировать на результат анализа – важное конкурентное преимущество любой компании. Благодаря этому было достаточно просто понять проблему компании и найти для нее решение с помощью создания микрослужбы, которая легко вписывается в сложную автоматизированную систему ТОО «ЕТС-Тендер», не нарушая ее целостности. Исходя из этого, можно сделать вывод, что компания ТОО «ЕТС-Тендер» нуждается в разработке соответствующей микрослужбы.

1.4 Постановка задачи

Одной из особенностей специальности “Информатика” является овладение навыками разработки любой системы, способной помочь коммерческой организации улучшить качество обслуживания клиентов, делать даже больше, чем ожидает Заказчик или Поставщик, если говорить об электронной торговле в формате b2b. В результате компания может значительно улучшить свои экономические показатели, получить прибыль.

Важной составляющей такой работы является анализ первичных данных, на основе которого и создается модель будущей системы.

Необходимо понять почему число незавершенных процедур закупок в настоящий момент имеет нынешний показатель, является ли он объективным, и можно ли на него повлиять так, чтобы снизить их число. Такой анализ возможен благодаря наличию информации, собранной за два с половиной года работы компании, для которой и была сделана разработка системы.

Если анализ покажет, что есть возможность автоматически, при введении определенного набора данных, исключить заявки, процедура завершения которой имеет малые шансы, то стоит разработать сервис, выполняющей работу по определению вероятности незавершения процедуры

закупки. Для этого нужно выяснить, какие параметры являются ключевыми, и влияют на вероятность завершения тендерной процедуры закупки.

Ценность создания такого сервиса очевидна. Во-первых, экономится время сотрудников. По предварительным данным один сотрудник ведет одновременно несколько процедур закупок, которые по времени могут длиться от одной до двух недель. Если исключить из них те, что изначально не несут в себе целесообразность создания, то сотрудник может сэкономить рабочее время. Во-вторых, экономятся деньги, за счет того, что высвобожденные человеко-часы будут направлены на создание заявок, которые будут оплачены.

Но необходимо создать сервис таким образом, чтобы он легко встраивался в уже имеющуюся информационно-технологическую систему компании, и не привела к ее изменениям.

Сервис должен иметь возможность либо масштабироваться в связи с возможным увеличением потока заявок, либо изначально быть основанным на технологиях, которые могут справиться с любым наплывом.

Современные цифровые технологии предполагают максимальный комфорт для пользователя. Поэтому электронный сервис может быть сама по себе сложной и состоять из нескольких подсистем, но юзер должен в результате видеть перед собой максимально простой интерфейс, которым легко пользоваться, и обучение работе на нем должно быть интуитивно понятным.

Сервис должен представлять собой форму, в которую можно внести ряд основных данных, которые микрослужба анализировала и давала тут же отчет. Суть отчета – это ответ, в котором указывается с какой величиной вероятности будет завершена тендерная процедура закупки.

2 Проектная часть

2.1 Выбор микрослужб и проектирование системы

Для системы были выбраны 3 микрослужбы:

- микрослужба «модель»;
- микрослужба «front»;
- микрослужба «api».

Микрослужба «Модель» является моделью машинного обучения, которая обучена распознавать определенные типы закономерностей для выполнения прогнозов. Нейронные сети — это структура машинного обучения, которая пытается имитировать модель обучения естественных биологических нейронных сетей. Биологические нейронные сети имеют взаимосвязанные нейроны с дендритами, которые получают входы, затем на основе этих входов они генерируют выходной сигнал через аксон к другому нейрону. Модель имитирует этот процесс с помощью искусственных нейронных сетей. Аналогом нейрона в машинном обучении является перцептрон.

Принцип работы перцептрона. На рисунке 2.1 входные данные $x_1, x_2 \dots x_n$ имеют вес $w_1, w_2 \dots w_n$. Перцептрон получает входные данные, умножает их на вес, а затем передает их в функцию активации для получения исхода. Смещение всегда добавляется для избежания проблемы, когда все входные данные могут быть равны нулю (потому что тогда весь исход будет равен нулю, т.е. будет бесполезен) [3].

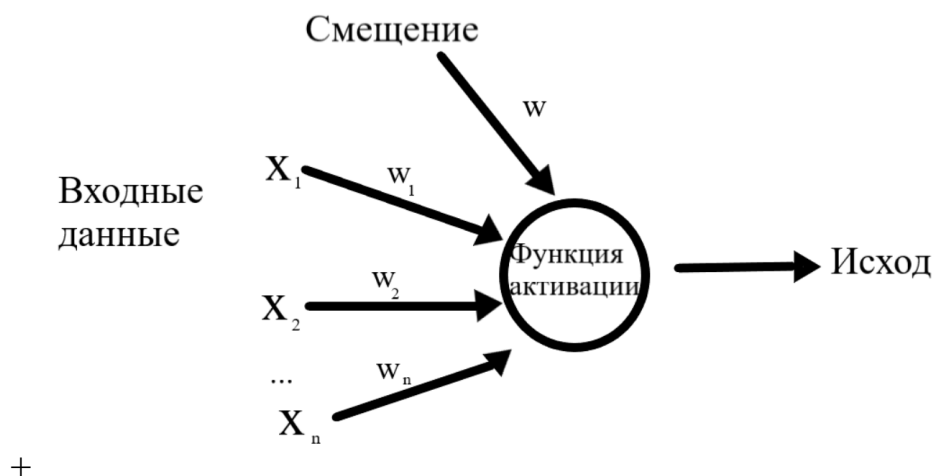


Рисунок 2.1. - Визуальное представление принципа работы перцептрона

Чтобы создать нейронную сеть, добавляются слои перцептронов, превращаясь многослойную модель перцептрона. Она состоит из входного слоя(input), который непосредственно принимает входные данные, скрытого

слоя(hidden) и выходного слоя(output), который будет создавать конечный результат. Взаимодействие этих слоев представлено на рисунке 2.2.

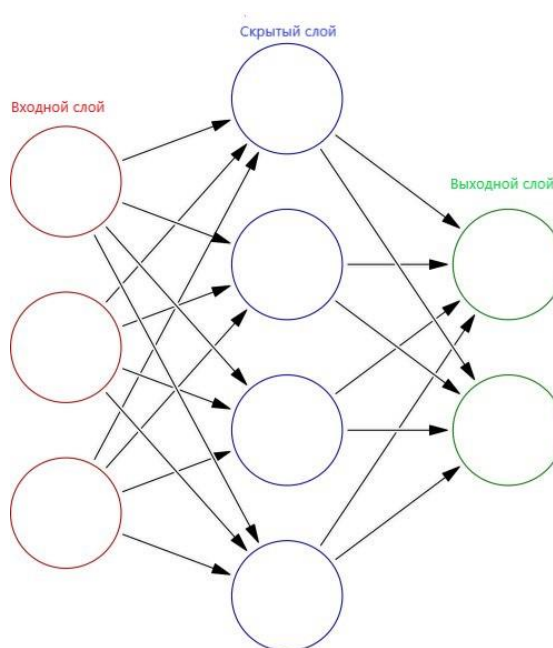


Рисунок 2.2 – Визуальное представление взаимодействия слоев в нейронной сети

Микрослужба «Модель» разрабатывается в первую очередь, так как для ее создания требуется выбрать набор данных, определить их значимости создания и обучения модели.

Front – это микрослужба, являющаяся интерфейс-системой для пользователя. Она будет считаться front-end частью и представлять собой приложение с простым для восприятия интерфейсом.

API – это микрослужба, которая представляет собой интерфейс прикладного программирования, то есть некоторый структурированный интерфейс запроса/ответа.

Диаграмма последовательности отображает (рис. 2.3) планируемый процесс работы системы для вывода вероятности успешности тендера. Для этого Менеджер или любой другой пользователь системы вводит данные в приложении «Front». Передача данных происходит через API и отправляется до модели. Модель производит вычисления и возвращает в ответ вероятность в процентах.

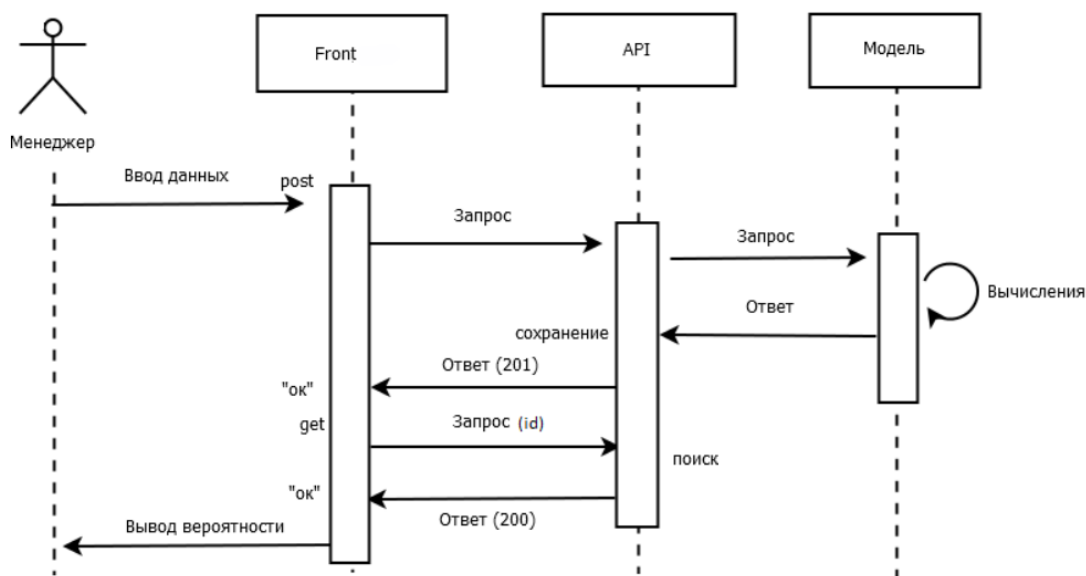


Рисунок 2.3 – Диаграмма последовательности

2.2 Выбор инструментов для создания системы

2.2.1 Анализ современных языков программирования для работы с машинным обучением

Язык программирования – это разработанный для написания программ язык, в котором собраны правила, позволяющие компьютеру выполнить какой-либо вычислительный процесс. Для разработчиков на сегодняшний день представлен большой спектр языков программирования, которые могут быть использованы в зависимости от того, что нужно создать, выполнить или организовать. Самыми распространёнными языками для работы с данными являются R, Python, JavaScript, C++ и Java. Анализ выбора необходимо произвести с точки зрения машинного обучения.

R – язык программирования для статистической обработки данных и работы с графикой, а также свободная программная среда вычислений с открытым исходным кодом в рамках проекта GNU. Этот язык программирования, появившийся в начале 90-ых годов, чаще всего используется для статистической обработки данных. В машинном обучении его применяют для решения задач, приближенных к классификации, также для формирования дерева решений. К его обычной установке прилагается множество дополнительных встроенных функций. Возможность визуально представить данные. Но этот язык не отличается высокой производительностью и имеет нестандартное индексирование [6].

Python – высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности

разработчика и читаемости кода, имеющий большое количество библиотеки программных пакетов, в частности для решения задач, касающихся машинного обучения. Из-за простоты синтаксиса является очень популярным среди начинающих программистов [4].

JavaScript – мультипарадигменный язык программирования. Является реализацией стандарта ECMAScript. JavaScript обычно используется как встраиваемый язык для программного доступа к объектам приложений. В области машинного обучения не сильно развит, в нем не хватает инструментов для работы с данными, и он имеет ограниченное применения в данном направлении [5].

C++ является одним из самых старых языков программирования, получивших широкое распространение, появившийся в 1983 году. Это компилируемый, статически типизированный язык программирования общего назначения. Поддерживает такие парадигмы программирования, как процедурное программирование, объектно-ориентированное программирование, обобщённое программирование. В нем есть достаточное количество программ машинного обучения и C++ обеспечивает более высокий уровень контроля, но разработка на этом языке обычно подразумевает написание объемного и сложного кода, больше соответствующего монолитным программам, а не микросервисной архитектуры [7].

Java – типизированный объектно-ориентированный язык программирования, разработанный компанией Sun Microsystems. Язык общего назначения, часто используемый в разработке приложений. Отличается высокой типобезопасностью и имеет широкий спектр функций и алгоритмов для создания моделей машинного обучения. Минусом является то, что у него мало библиотек для статистической обработки данных [8].

2.2.2 Выбор языков программирования Python и JavaScript

Для разработки любой, системы безопасность и возможность написания кода без затрат всегда является привлекательным, когда дело касается создания так называемой модели важно, чтобы язык программирования включал большое количество соответствующих библиотек. Поскольку это система, она должна без проблем взаимодействовать с другими микросервисами. В таблице 2.1 приводится наглядное сравнение вышеуказанных языков программирования.

Таблица 2.1 – Сравнение языков программирования

Название ЯП/Название характеристики	Лицензия	Интеграция с другими инструментами/языкам	Скорость обработки данных	Безопасность	Библиотеки МО
-------------------------------------	----------	---	---------------------------	--------------	---------------

стики ЯП		и			
----------	--	---	--	--	--

Продолжение таблицы 2.1

R	Бесплатная	Простая	Высокая	Средняя	Имеются только для конкретных задач
Python	Бесплатная	Простая	Высокая	Высокая	Имеются в большом количестве
Java	Зависит от версии	Простая	Низкая	Высокая	Имеются в большом количестве
C++	Платная	Относительно сложная	Низкая	Средняя	Имеются в достаточном количестве
JavaScript	Бесплатная	Простая	Высокая	Высокая	Имеются, но мало

Исходя из результатов сравнения выбранных языков программирования друг с другом, становится ясным, что Python больше всего подходит для работы с машинным обучением, так как обладает большим выбором библиотек и фреймворков, безопасен в использовании, имеет высокую скорость обработки набора данных, легко интегрируется с другими инструментами и используется без затрат средств. Также он поддерживает фреймворк Flask, который можно использовать в написании веб-API. Поэтому обучение модели и создание API будет произведено на этом языке.

Согласно таблице 2.1 JavaScript как язык программирования, обладает хорошими характеристиками. Он дает возможность создавать богатые компоненты пользовательского интерфейса, такие как слайдеры изображений, всплывающие окна, мегаменю навигации по сайту, проверки форм, вкладки, аккордеоны и многое другое.

Для написания front-end части большим плюсом является наличие в JavaScript библиотеки React с открытым исходным кодом для разработки пользовательских интерфейсов, потому что она подходит для написания одностраничных приложений. Также позволяет веб-страницам реагировать на действия пользователя и динамически обновлять себя, и все это без необходимости перезагрузки страницы для изменения ее внешнего вида.

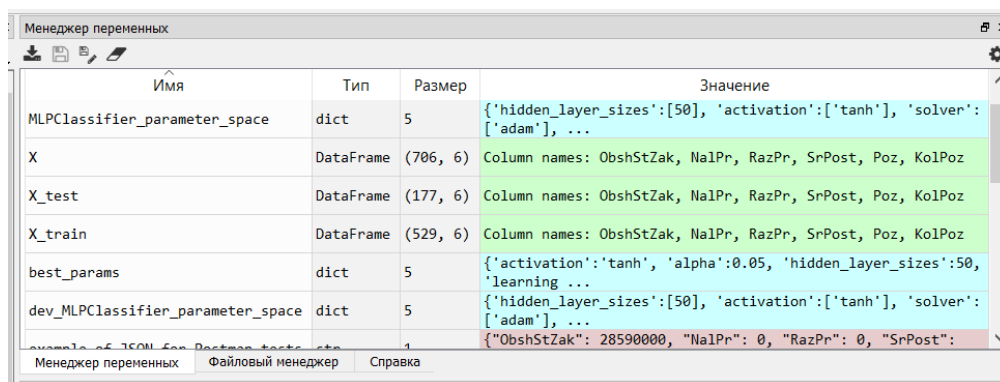
2.2.3 Выбор среды разработки и редактора кода

IDE – интегрированная среда разработки – это программа, предназначенная для разработки программного обеспечения. Spyder – свободная и кроссплатформенная интерактивная IDE для научных расчетов на языке Python, обеспечивающая простоту использования функциональных возможностей и легковесность программной части.

Возможности Spyder:

а) просмотр значений переменных;

В процессе выполнения программы они выводятся на панели в виде списка с возможностью просмотра их значений. Имеется даже возможность строить графики по данным массива. Это является удобным при отладке программы и поиске логических ошибок.

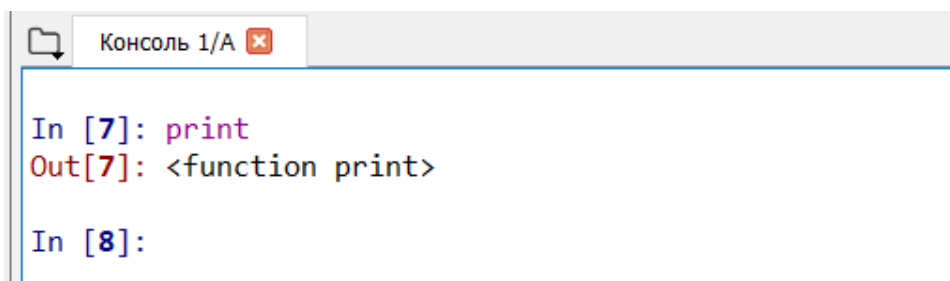


Имя	Тип	Размер	Значение
MLPClassifier_parameter_space	dict	5	{'hidden_layer_sizes':[50], 'activation':['tanh'], 'solver':['adam'], ...}
X	DataFrame	(706, 6)	Column names: ObshStZak, Na1Pr, RazPr, SrPost, Poz, KolPoz
X_test	DataFrame	(177, 6)	Column names: ObshStZak, Na1Pr, RazPr, SrPost, Poz, KolPoz
X_train	DataFrame	(529, 6)	Column names: ObshStZak, Na1Pr, RazPr, SrPost, Poz, KolPoz
best_params	dict	5	{'activation':'tanh', 'alpha':0.05, 'hidden_layer_sizes':50, 'learning ...}
dev_MLPClassifier_parameter_space	dict	5	{'hidden_layer_sizes':[50], 'activation':['tanh'], 'solver':['adam'], ...}
example_of_ICM_for_Deepnet_tests	dict	1	{"ObshStZak": 28590000, "Na1Pr": 0, "RazPr": 0, "SrPost": ...}

Рисунок 2.4 - Список переменных в готовой модели

б) встроенная консоль;

Spyder предоставляет большие возможности по работе с консолями Python или IPython. Можно создавать требуемое количество консолей и взаимодействовать с ними как с отдельными процессами. Программы могут запускаться в новой или в существующей консоли.



```
In [7]: print
Out[7]: <function print>

In [8]:
```

Рисунок 2.5 - Взаимодействие с консолью IPython

в) мгновенный анализ на ошибки.

С помощью ruflakes в реальном времени код программы проверяется на ошибки, и в случае их нахождения пользователь получает подсказки

```
8 import pandas as pd
9 import numpy as np
10 import seaborn as sns
11
12 from sklearn import metrics
13 from sklearn import metrics
14 #from joblib import dump, load
15 import pickle
16 import matplotlib.pyplot as plt
```

Анализ кода

'numpy as np' imported but unused

Рисунок 2.6 – Анализ кода в модели в процессе разработки

Редактор исходного кода – это приложение или встроенный в интегрированную среду редактор для создания и изменения исходного кода программ. Одним из них является Visual Studio Code, который был разработан для всех самых часто используемых операционных систем. Он был выбран для редактирования front-end части.

2.2.3.1 Возможности Visual Studio Code:

а) IntelliSense;

Это система автозавершения кода. Она считается одной из основных средств VS Code, обеспечивающих высокую производительность труда. Выбрать нужную функцию и переменную удобнее не вводя, я используя эту возможность.

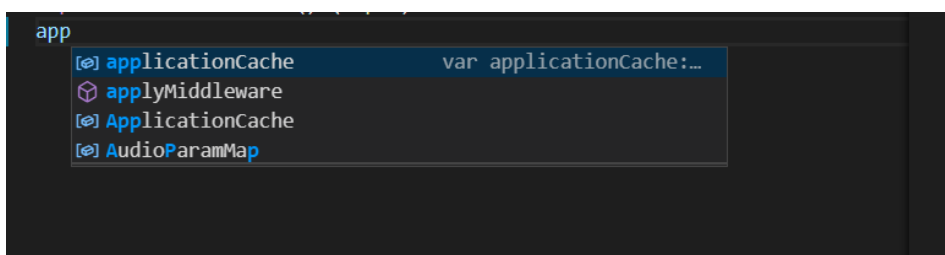


Рисунок 2.7 – Работа с IntelliSense

б) клавиатурные сокращения;

Каждый раз, когда программисту приходится убирать руки с клавиатуры, он теряет несколько секунд. Это удлиняет сроки работы над проектами. Панель клавиатурных сокращений можно посмотреть, воспользовавшись комбинацией клавиш CTRL + K + S.

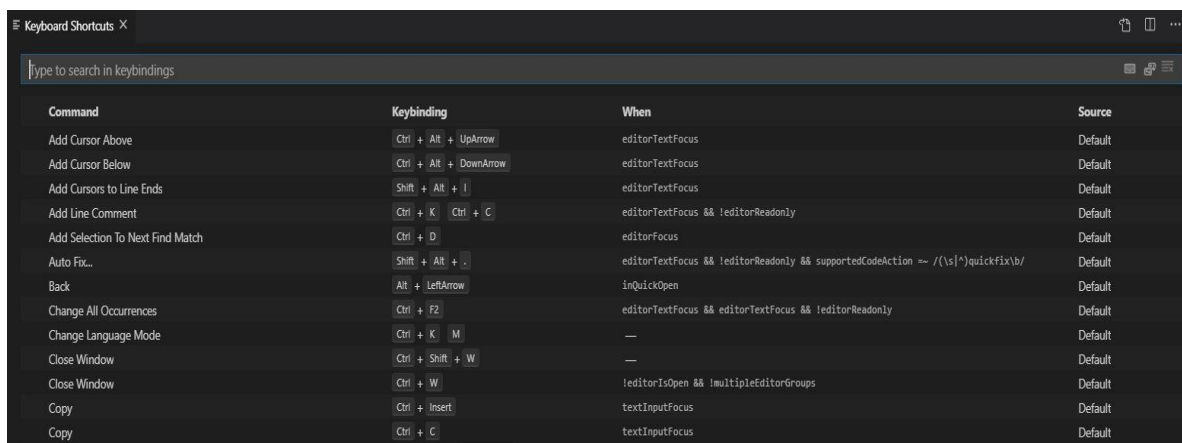


Рисунок 2.8 – Панель клавиатурных сокращений в VS Code

с) отладчик.

В VS Code имеется отладчик, который позволяет запускать код и устанавливать в нём точки останова, указывая те места, в которых выполнение программы должно быть приостановлено. Это позволяет, не покидая редактора, отлаживать программы, анализируя их внутреннее состояние во время выполнения.

Отладка – это нечто большее, нежели вывод сообщений в консоль инструментов разработчика Chrome. В ходе отладки можно узнавать о том, что происходит внутри программы и, в результате, обнаруживать и устранять источники самых разных проблем.

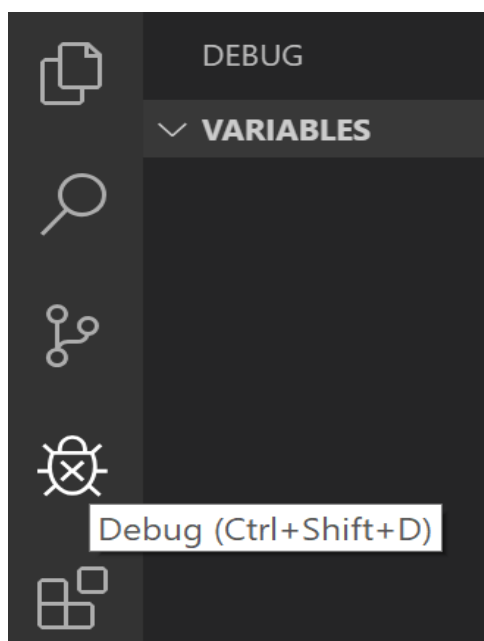


Рисунок 2.9 – Кнопка Debug и панель отладки

2.3 Разработка микрослужб прогнозирования системы

Создание спроектированной системы состоит из реализации трех выбранных составляющих системы, то есть микрослужбы «Модель», «API» и «Frontend».

2.3.1 Создание микрослужбы «Модель»

Создание этой микрослужбы начинается с выбора набора данных. Было определено, что существует 6 основных параметров, по которым можно определить коридор значений, при соблюдении которого заявка на проведение процедуры конкурса с высокой долей вероятности совершится:

а) Стоимость закупки. Стоимость всегда является важной частью любого закупочного процесса, поскольку формируется так, чтобы соответствовать возможностям компании-организатора и интересам поставщиков;

б) Наличие предоплаты. Предоплата обычно интересна Поставщикам, поскольку является привлекательной гарантией того, что сделка точно будет завершена (хотя это и так гарантируется документами, прикладываемыми к заявке на участие);

в) Размер предоплаты. Размер зависит от платежеспособности Закупщика, чаще всего встречается 50% предоплата;

г) Срок поставки. Сроки поставки зависят от процесса, для которого создается закупка – иногда нужно доставить товар в течение недели, иногда в течение полугода;

д) Попозиционно. Этот критерий отвечает на вопрос: будет ли закупка целой (когда нужно предоставить все лоты или позиции разом) или можно подать заявку на попозиционную доставку товара. Например, когда закупаются продукты для влажной обработки помещений, компания-организатор может искать резиновые перчатки, химические средства для мытья полов и швабры. И есть возможность дать разрешение на попозиционную доставку лотов;

е) Количество позиций. Критерий определяет не количество продуктов или услуг, но количество типов продуктов.

Далее происходит создание набора данных:

Был создан датасет - файл 'tender.csv', в котором из архива всех сделок по тендерам вручную была собрана информация.

1	ObshStZak,NalPr,RazPr,SrPost,Poz,KolPoz,Outcome			
2	2004000,1,50,25,0,1,1			
3	3725109,0,0,45,1,2,1			
4	59600,0,0,20,0,1,0			
5	3414850,1,100,60,0,1,1			
6	4258200,0,0,360,0,10,0			
7	26738578,0,0,45,1,3,1			
8	1909403,1,50,60,1,15,1			

Рисунок 2.10 – Набор данных ‘tender.csv’

После этого происходит создание и обучение модели машинного обучения. Ее программное название «Модель.ру».

Импортирование набора данных происходит с помощью функции `read_csv`, которая используется для чтения или загрузки данных из файлов CSV

```
import pandas as pd
tender = pd.read_csv('tender.csv')
```

Кадр данных, импортированный в программу представлен на рисунке 1.10. В нем 8 столбцов: Index – порядковый номер примера закупки, ObshStZak – стоимость закупки, NalPr – наличие предоплаты, RazPr – размер предоплаты, SrPost – срок поставки, Poz – Попозиционно, KolPoz – количество позиций, Outcome – исход.

Index	ObshStZak	NalPr	RazPr	SrPost	Poz	KolPoz	Outcome
265	1737790	0	0	60	0	1	1
266	1338000	1	80	150	0	1	0
267	12911565	0	0	30	1	53	0
268	20208837	0	0	30	0	12	0
269	2141071	0	0	90	1	1	0
270	705080	1	100	45	0	1	1
271	1149206	1	100	45	0	1	1
272	16424000	1	50	30	1	3	1
273	773381	1	100	45	0	1	1
274	529217	1	100	45	0	1	1
275	3542573	0	0	30	0	34	1

Рисунок 2.11 – Кадр данных в файле ‘tender.csv’

Создание обучающих и проверочных наборов и импортирование их в файлы `X_test.csv` и `y_test.csv`:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y)
X_test.to_csv("X_test.csv", index=False)
y_test.to_csv("y_test.csv", index=False)
```

Импортирование и инициализация `MLPClassifier`, а также выбор его параметров:

`hidden_layer_sizes`: этот параметр позволяет нам установить количество слоев и количество узлов, которые мы хотим иметь в классификаторе нейронной сети. Каждый элемент в кортеже представляет количество узлов в i -й позиции, где i - индекс кортежа. Таким образом, длина кортежа обозначает общее количество скрытых слоев в сети.

`max_iter`: обозначает количество эпох.

`activation`: функция активации для скрытых слоев.

`solver`: этот параметр определяет алгоритм оптимизации веса по узлам.

`alpha`: параметр срока регуляции

`random_state`: параметр позволяет установить начальное значение для воспроизведения тех же результатов

```
from sklearn.neural_network import MLPClassifier
```

```
mlp = MLPClassifier()
```

```
MLPClassifier__parameter_space = {  
    'hidden_layer_sizes': [(50,50,50), (50,100,50), (100,)],  
    'activation': ['tanh', 'relu'],  
    'solver': ['sgd', 'adam'],  
    'alpha': [0.0001, 0.05],  
    'learning_rate': ['constant','adaptive'],  
}
```

Импортирование и инициализация `GridSearchCV`

```
from sklearn.model_selection import GridSearchCV
```

```
clf = GridSearchCV(mlp, MLPClassifier__parameter_space, n_jobs=1, cv=3)
```

Предоставление данных для обучения нейронной сети

```
clf.fit(X_train, y_train)
```

Создание файла `.joblib` в котором будет храниться модель.

```
model_filename = model_name + '.joblib'
```

```
model_file = open(model_filename, 'wb')
```

```
pickle.dump(clf, model_file)
```

```
model_file.close
```

```
model_file = open(model_filename, 'rb')
```

```
clf = pickle.load(model_file)
```

```
model_file.close
```

2.3.2 Описание работы главных элементов микрослужбы «API»

Микрослужба «API» описана в файле «API.py». В части кода, изображенном на рисунке 2.12 представлен класс `api_logic`, который состоит из функций: `_init_`, `add_prediction`, `get_prediction`, `load_model`.


```

class api_logic:

    def __init__(self):
        self.current_model_name = "tender_clf"
        model_file = open(self.current_model_name+".joblib", "rb")
        #self.mlp = load(self.current_model_name+".joblib")
        self.mlp = pickle.load(model_file)
        model_file.close()
        self.predictions_store = {}

    def add_prediction(self, test_sample_as_dict):
        test_sample_as_df_row = convert_sample_from_dict_to_df_row(test_sample_as_dict)
        #prediction = int(self.mlp.predict(test_sample_as_df_row)[0])
        prediction = self.mlp.predict_proba(test_sample_as_df_row)[0][0]
        print(prediction)
        return prediction

    def get_prediction(self, test_sample_as_dict):
        sample_as_dict_hashed = get_hash_based_ID(test_sample_as_dict)
        if sample_as_dict_hashed in self.predictions_store:
            prediction = self.predictions_store[sample_as_dict_hashed]
        else:
            prediction = self.add_prediction(test_sample_as_dict)
            self.predictions_store[sample_as_dict_hashed] = prediction
        return prediction, sample_as_dict_hashed

    def load_model(self, new_model_file_name):
        #self.mlp = load(new_model_file_name+".joblib")
        model_file = open(self.current_model_name+".joblib", "rb")
        self.mlp = pickle.load(model_file)
        model_file.close()
        self.current_model_name = new_model_file_name
        self.predictions_store = {}
        model_loading_result_dict = {}
        model_loading_result_dict["model_loading_result"]="The model " + new_model_file_name + "
        return model_loading_result_dict

```

Рисунок 2.12 – Фрагмент кода микрослужбы «API»

Функции этого класса нужны для работы с моделью. Например, на рисунке 2.13 объявляется функция `add_prediction`, которая конвертирует данные для расчетов.

```

def add_prediction(self, test_sample_as_dict):
    test_sample_as_df_row = convert_sample_from_dict_to_df_row(test_sample_as_dict)
    #prediction = int(self.mlp.predict(test_sample_as_df_row)[0])
    prediction = self.mlp.predict_proba(test_sample_as_df_row)[0][0]
    print(prediction)
    return prediction

```

Рисунок 2.13 – Фрагмент кода микрослужбы «API»

Объект `api_logic` выполняет все, что описано в классе `api_logic`.

```

api_logic=api_logic()

```

Рисунок 2.14 – Строка кода микрослужбы

На рисунке 2.15 описываются функции, в которых данные передаются между моделью и API, используя методы GET и POST.

```

7 @app.route("/customer-record", methods=["POST"])
8 @cross_origin(supports_credentials=True)
9 def new_rating():
10     new_test_sample = request.get_json()
11     new_prediction, customer_ID = api_logic.get_prediction(new_test_sample)
12     customer_record = {}
13     customer_record["customer_ID"]=customer_ID
14     return json.dumps(customer_record), 201
15
16 @app.route("/customer-prediction/<customer_ID>", methods=["GET"])
17 def get_prediction_for_customer(customer_ID):
18     prediction_dict = {}
19     prediction_dict["probability-of-tender"] = api_logic.predictions_store[customer_ID]
20     return json.dumps(prediction_dict), 200
21

```

Рисунок 2.15 – Фрагмент кода микрослужбы «API»

В строке кода, изображенной на рисунке 2.16, `app.run` запускает API на локальном хосте `127.0.0.1:9875`.

```

) app.run(host="127.0.0.1", port=9875)

```

Рисунок 2.16 – Строка кода микрослужбы «API»

2.3.3 Описание микрослужбы «Frontend»

Эта микрослужба создана как приложение «frontend». Приложение «frontend» состоит из папок «node_modules», «public», «src».

компьютер > Локальный диск (C:) > Пользователи > Alika > front

Имя	Дата изменения	Тип	Размер
node_modules	02.04.2020 13:17	Папка с файлами	
public	02.04.2020 13:10	Папка с файлами	
src	02.04.2020 13:10	Папка с файлами	

Рисунок 2.17 – компоненты приложения frontend

Папка `node_modules` содержит все модули приложения. Модули — это законченные программные блоки, состоящие из моделей, представлений, контроллеров и других вспомогательных компонентов.

Папка `public` содержит иконки и изображения приложения.

Папка `src` состоит из 3 папок: `components`, `misc`, `redux` и различных файлов, которые составляют важную часть приложения.

Папка `components` хранит в себе файлы, составляющие логическую и стилевую оболочки приложения.

Папка `misc` содержит сервис и стили. В файле «`dataService.js`» объявляется метод `POST`, который передает введенные пользователем данные в микросервис «API» и после успешной обработки отправляет `GET`, который принимает результаты обработки.

```
return axios({
  method: 'POST',
  url: 'http://localhost:9875/customer-record',
  data: data,
  headers: { 'Content-Type': 'application/json' }
})
.then(response => {
  return axios({
    method: 'GET',
    url: 'http://localhost:9875/customer-prediction/' + response
  })
  .then(response => response.data)
})
}
```

Рисунок 2.18 – Скрипты инструмента Redux

В папке `redux` находятся скрипты инструмента Redux для создания и сохранения изменений в приложении.

3 Эксплуатационная часть

3.1 Интерфейс

Основной интерфейс системы включает в себя:

- область ввода данных;
- кнопку передачи данных;
- область вывода вероятности.

На рисунке 3.1 представлена область ввода данных. Область ввода данных состоит из подписанных окон.



The image shows a light blue rectangular area containing six white input fields arranged in a 3x2 grid. Each field has a label in Russian: 'Стоимость закупки' (Purchase cost), 'Наличие предоплаты' (Advance payment), 'Размер предоплаты' (Advance payment amount), 'Срок поставки' (Delivery term), 'Попозиционно' (Positionally), and 'Количество позиций' (Number of positions).

Рисунок 3.1 – Область ввода данных

На рисунке 3.2 показана кнопка, которая позволяет отправить введенные данные для подсчета вероятности.

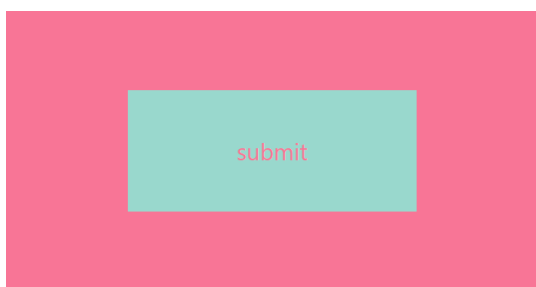


Рисунок 3.2 – Кнопка «submit»

На рисунке 3.3 показана область вывода вероятности успешности тендера.

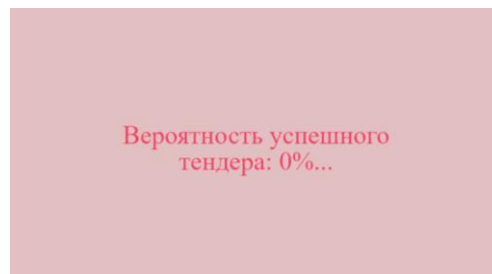


Рисунок 3.3 – Область вывода прогноза

3.2 Алгоритм действий для работы с системой

Для получения результата прогнозирования нужно выполнить такие действия:

- 1) Запуск API.py для того, чтобы он мог передавать набор данных;

```
In [1]: runfile('C:/Users/Alika/Desktop/Система/API.py', wdir='C:/Users/Alika/Desktop/Система')
* Serving Flask app "API" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:9875/ (Press CTRL+C to quit)
```

Рисунок 3.4 –Результат запуска API.py

- 2) Запуск приложения «front» через командную строку;
 - а) переход в папку на локальном диске, где установлено приложение с помощью команды cd;

```
C:\Users\Alika>cd front
```

Рисунок 3.5 – Переход папку с приложением

- б) старт приложения с помощью команды npm start.

```
C:\Users\Alika\front>npm start
```

Рисунок 3.6 – команда для запуска приложения

- 3) Ввод данных в окна;

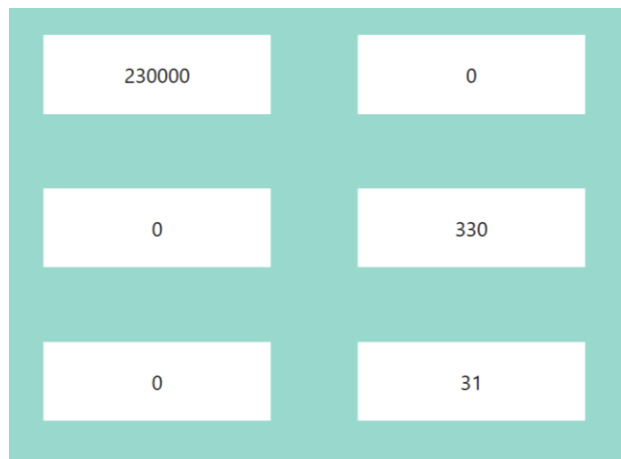


Рисунок 3.7 – Окна с введенными данными

4) Нажатие кнопки «submit»;

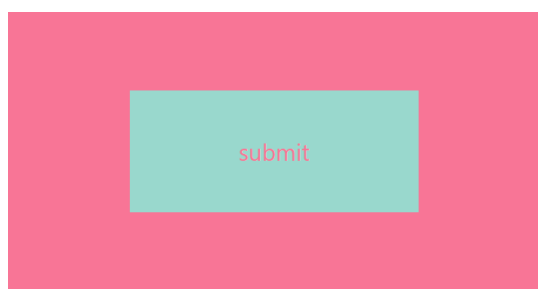


Рисунок 3.8 – Кнопка «submit»

5) Получение результата.



Рисунок 3.9 – Результат подсчета вероятности

3.3 Системные требования

Рекомендуемые системные требования:

- операционная система: Windows 10 — 64-bit;
- процессор (CPU): Intel Core i5 (Desktop);
- оперативная память (RAM): 1 ГБ (или больше);
- свободное место на жёстком диске: ~200 МВ.

Минимальные системные требования:

- операционная система: Windows 8/10 — 64-bit;
- процессор (CPU): Intel Celeron (Desktop);
- оперативная память (RAM): 512 МВ (или больше);
- свободное место на жёстком диске: ~193 МВ.

4 Экономическая часть

4.1 Техничко-экономическое обоснование проекта

В экономической части данного дипломного проекта будет приведено технико-экономическое обоснование разработки системы “Микрослужбы прогнозирования веб-API”. Она была проведена для электронной торговой площадки ТОО «ЕТС-Тендер».

В ходе деятельности организации была собрана статистика по количеству поданных на сайт заявок и количеству законченных по ним процедур. За период с июля 2017 года по 1 января 2020 года из 100 % поданных заявок, только 64% были завершены. При этом средний объем одной заявки составляет около 600 тысяч тенге.

Таким образом, очевидно, что необходимо понять, в чем причина незавершения процедур и минимизация таких случаев. Недопустимо, чтобы на каждую заявку, процедура по которой длится в среднем около 7 -10 дней, тратилось время компании, Заказчика и Поставщика. Кроме того, при незавершении процедуры клиенты могут потерять предполагаемый доход, не получив в назначенное время необходимых товаров, которые им нужны в ходе их деятельности [9].

Была поставлена задача создать программный продукт, который на этапе создания заявки сразу мог показать, состоится ли завершение процедуры или нет, на основе анализа основных параметров заявки.

Система микрослужб прогнозирования помогает менеджерам и самим Заказчикам, определяя вероятность успешности тендера основываясь на информации о прошлых сделках. В ее основе лежит машинное обучение и современные методы запроса и передачи информации.

4.2 Расчет трудоёмкости разработки системы

Для определения трудоемкости разработки ПП составлен перечень всех основных этапов и видов работ, которые должны быть выполнены.

Таблица 4.1 – Распределение работ по этапам и видам и оценка их трудоемкости

Этап разработки ПП	Вид работы на данном этапе	Трудоемкость разработки ПП, чел.× ч.
1. Подготовка	Постановка задачи, выбор набора данных и определение составляющих системы для	1×8

	эффективной микрослужб	работы	
--	---------------------------	--------	--

Продолжение таблицы 4.1

2. Создание модели	Сбор и анализ данных, обучение модели	1×11 2
3. Backend-разработка	Создание внутренней части системы	1×80
4. Frontend-разработка	Создание внешней части системы	1×56
5. Тестирование	Проверка правильности работы каждой части и работы микрослужб между собой, выявление ошибок	1×8
6. Доработка	Исправление ошибок и доведение работы до конечного состояния	1×40
ИТОГО трудоемкость выполнения работы		304

В общем на работу над системой ушло 304 часа, рабочий день составляет 8 часов, следовательно, было потрачено 38 рабочих дней.

4.3 Расчет затрат на разработку информационной системы

4.3.1 Затраты на спецоборудование

Таблица 4.2 – Затраты на спецоборудование

Наименование материального ресурса	Единица измерения	Количество израсходованного материала	Цена за единицу, тг	Сумма, тг
Ноутбук Lenovo	шт	1	280000	280000
Компьютерная мышь Razer	шт	1	18990	18990
ОС Windows 10 Pro (ключ активации)	шт	1	2 000	2 000
Дистрибутив языков Anaconda	шт	1	Бесплатно	0

Редактор кода VS Code	шт	1	Бесплатно	0
ИТОГО затраты на материальные ресурсы				300990

4.3.2 Затраты на электроэнергию

Таблица 4.3 – Затраты на электроэнергию

Наименование оборудования	Паспортная мощность, кВт	Коэффициент использования мощности	Время работы оборудования для разработки ПП, ч	Цена на электроэнергию, тг /кВт*ч	Сумма, тг
Ноутбук Lenovo	0,135	0,7	360	17,79	605,2
ИТОГО затраты на электроэнергию					605,2



Рисунок 4.1 – Данные о напряжении на выходе и потребляемом токе ноутбука

Взяв данные с задней крышки устройства (ноутбука), рассчитывается мощность:

$$20 \text{ (В)} \times 6,75 \text{ (А)} = 135 \text{ Вт} = 0,135 \text{ кВт}$$

Общая сумма затрат на электроэнергию (Z_3) рассчитывается по формуле:

$$Z_3 = \sum M_i * K_i * T_i * Ц$$

где M_i - паспортная мощность i -го электрооборудования, кВт;

K_i - коэффициент использования мощности i -го электрооборудования (принимается $K_i=0.7, 0.9$);

T_i - время работы i -го оборудования за весь период разработки ПП ч;

C - цена электроэнергии, тг/кВт×ч;

i - вид электрооборудования;

n - количество электрооборудования.

$Z_3 = 0,135 \times 0,7 \times 360 \times 17,79 = 605,2$ тенге

4.3.3 Затраты на оплату труда

Базой для расчета основной заработной платы являются общая трудоемкость, плановая численность работников и плановые сроки разработки ПО.

По данным о специфике и сложности выполняемых функций составляется штатное расписание группы специалистов-исполнителей, участвующих в разработке системы, с определением образования, специальности, квалификации и должности. Заработная плата разработчиков была определена по данным на headhunter.kz, путем нахождения средней ставки для каждого разработчика (при 8-часовом рабочем дне).

Таблица 4.4 – Сведения по работникам, задействованным в проекте

Специалист - Исполнитель	Количество, человек	Заработная плата в месяц, тенге
Программист машинного обучения	1	300000
Backend-разработчик	1	250000
Frontend-разработчик	1	250000
Итого		800000

Часовая тарифная ставка рассчитывается путем деления месячной тарифной ставки на установленную при 40-часовой недельной норме рабочего времени расчетную среднемесячную норму рабочего времени в часах (Φ_p):

$$T_{\text{ч}} = \frac{T_{\text{м}}}{\Phi_p} = \frac{300000}{8 \cdot 21} = 1785 \text{ тг/ч};$$

$$T_{\text{ч}} = \frac{T_{\text{м}}}{\Phi_p} = \frac{250000}{8 \cdot 21} = 1488 \text{ тг/ч},$$

где $T_{\text{ч}}$ - часовая тарифная ставка (тыс. тенге);

$T_{\text{м}}$ - месячная тарифная ставка (тыс. тенге).

Таблица 4.5 - Затраты на оплату труда

Категория работника	Трудоемкость разработки ПП, чел.×ч	Часовая ставка, тг/ч	Сумма, тг
Программист машинного обучения	138	1785	246330
Backend-разработчик	95	1488	145824
Frontend-Разработчик	71	1488	105648
ИТОГО затраты на оплату труда			497802

4.3.4 Социальные налоги

Социальный налог для юридических лиц рассчитывается:

$$З_{СН} = (З_{ФОТ} - ОПВ - ВОСМС) \cdot 0,095 - СО$$

где $З_{ФОТ}$ – общий фонд оплаты труда разработчиков, тенге;

ОПВ – обязательный пенсионный взнос (10% от заработной платы);

ВОСМС – взнос на ОСМС (2% от заработной платы);

СО – социальные отчисления (3,5% от заработной платы с вычетом пенсионного взноса).

$$ОПВ = 497\,802 \cdot 0,1 = 49\,780$$

$$ВОСМС = 497\,802 \cdot 0,02 = 9\,956$$

$$СО = (497\,802 - 49\,780) \cdot 0,035 = 15\,680$$

$$З_{СН} = (497\,802 - 49\,780 - 9\,956) \cdot 0,095 - 15\,680 = 25\,936 \text{ тенге}$$

$З_{сзи}$ – отчисления по социальному налогу, тенге:

$$З_{сзи} = З_{СН} + СО + ВОСМС = 25\,936 + 15\,680 + 9\,956 = 51\,572 \text{ тенге}$$

Таблица 4.6 - Смета затрат на разработку системы

Статьи затрат	Сумма, тг
1) Затраты на спецоборудование	300 990
2) Затраты на электроэнергию	605
3) Затраты на оплату труда	497 802
4) Социальные налоги	51 572
ИТОГО по смете	850 969

Рентабельность и прибыль по создаваемому ПО ($П_{сi}$) определяются исходя из результатов анализа рыночных условий, переговоров с заказчиком (потребителем) и согласования с ним отпускной цены, включающей дополнительно налог на добавленную стоимость [11]. В случае разработки ПО для использования внутри организации оценка программного продукта производится по действующим правилам и показателям внутреннего

хозрасчета (по ценам, устанавливаемым для расчета за услуги между подразделениями). Прибыль рассчитывается по формуле:

$$P_{oi} = C_{ni} \cdot \frac{Y_{pni}}{100}$$

где P_{oi} - прибыль от реализации ПО заказчику (тыс. тенге);

Y_{pni} - уровень рентабельности ПО (%), в дипломной работе брать 40-60%;

C_{ni} - себестоимость ПО (тыс. тенге).

$$P_{oi} = 850\,969 \cdot 0.6 = 510\,581 \text{ тенге}$$

Прогнозируемая цена ПО без налогов (C_{pi}):

$$C_{pi} = C_{ni} + P_{oi} = 850\,969 + 510\,581 = 1\,361\,550 \text{ тенге}$$

Прогнозируемая отпускная цена (C_{o1}):

$$C_{o1} = C_{ni} + \text{НДС}$$

Ставка налога на добавленную стоимость НДС в РК на 2020 год составляет 12% от отпускной цены ПО.

$$C_{o1} = 1\,361\,550 + \frac{1\,361\,550 \cdot 12}{100} = 1\,524\,936 \text{ тенге}$$

Организация-разработчик участвует в освоении ПО и несет соответствующие затраты, на которые составляется смета, оплачиваемая заказчиком по договору[11]. Затраты на освоение определяются по нормативу ($H_o=10\%$) от себестоимости ПО в расчете на 3 месяца и рассчитываются по формуле:

$$P_{oi} = C_{ni} \cdot \frac{H_o}{100} = 850\,969 \cdot 0,1 = 85\,297 \text{ тенге}$$

Затраты на сопровождение ПО (P_{ci}). Организация-разработчик осуществляет сопровождение ПО и несет соответствующие расходы, которые оплачиваются заказчиком в соответствии с договором и сметой на сопровождение. Затраты на сопровождение определяются по установленному нормативу ($H_c=20\%$) от себестоимости ПО (в расчете на год) и рассчитываются по формуле:

$$P_{ci} = C_{ni} \cdot \frac{H_c}{100} = 850\,969 \cdot 0,2 = 170\,593 \text{ тенге}$$

Капиталовложения программного обеспечения с учетом затрат на освоение и сопровождение будет:

$$K = 1\,524\,936 + 85\,297 + 170\,593 = 1\,780\,830 \text{ тенге.}$$

4.4 Расчет результатов от создания и использования Системы.

Программный продукт предназначен для исключения заявок с высоким риском неисполнения. Система упрощает процесс принятия решения – создавать заявку или нет. Снижает риск создания заявок, которые не приносят обороты и доход.

Оценка результатов от создания и использования системы делится на две части:

1) Экономия компании за счет увеличения объемов оборота и дохода за счет исключения создания и сопровождения заявок с высоким риском незавершения процедур;

2) Экономия компании за счет эффективного использования времени сотрудников за счет исключения создания и сопровождения заявок с высоким риском незавершения процедур.

Для оценки результативности компании за счет эффективного использования получение оборотов, дохода и времени сотрудников и от использования разработанной Системы необходимо сравнить эти параметры с применением Системы и без [10].

Для подсчета экономии за счет Системы подсчитываем примерное число заявок, создаваемых за 1 год – 5000 заявок. Известно, что без применения системы завершаются 3000 заявок и 2000 заявок – не завершаются. Создаваемая Система должна снизить число заявок с высоким риском не завершения процедур на 20% при том же количестве заявок. При применении Системы должны завершаться 4000 заявок и 1000 заявок – не завершаться.

Средний объем оборота 1 заявки высчитан путем деления общего оборота компании (3 млрд тенге) на количество всех заявок (5000 заявок). Все данные по ТОО «ЕТС-Тендер» в данной работе не носят точный характер, так как реальный цифры являются коммерческой тайной.

Доход высчитывается также по формуле: $D=O*T$, где D – доход, O – оборот, T – тариф, $T=0,02\%$.

		С применением Системы	Без применения Системы	Результативность (экономия)
Количество заявок в год	с высоким риском не завершения процедуры	1 000	2 000	
	с низким риском не завершения процедуры	4 000	3 000	
Средний объем оборота 1 заявки (тенге)		600 000	600 000	
Объем оборота в год (тенге)		2 400 000 000	1 800 000 000	600 000 000
Доход		24 000 000	18 000 000	6 000 000

Кол-во часов на создание заявок с низким риском не завершения процедуры в месяц	33,6	25,2	
Кол-во часов на сопровождение заявок низким риском не завершения процедуры в месяц	195,2	146,4	
Оплата 1 человеко-часа	1071	1071	
Оплата человеко-часов за 1 месяц	245044,8	183783,6	61 261
Оплата человеко-часов за 1 год	2940537,6	2205403,2	735 134

Таблица 4.7.- Результативность системы

Данная Система не предполагает расхода каких-либо материалов. Накладные расходы при применении Системы увеличиваются незначительно:

$$НР = ЗП \cdot \frac{Н_{НР}}{100} = 432000 \cdot 0,7 = 302\,400 \text{ тенге,}$$

где ЗП – оплата за труд за 1 год, $Н_{НР}$ – норма накладных расходов.

Расчетный коэффициент экономической эффективности капитальных вложений составляет:

$$E_p = \frac{\text{Э}_{уг}}{K}$$

где E_p - расчетный коэффициент экономической эффективности капитальных вложений;

$\text{Э}_{уг}$ — ожидаемая условно-годовая экономия, тенге;

K — капитальные вложения на создание системы, тенге.

$$E_p = \frac{6\,735\,134}{1\,780\,830} = 3,78$$

Расчетный срок окупаемости капитальных вложений составляет:

$$T_p = \frac{1}{E_p}$$

где E_p - коэффициент экономической эффективности капитальных вложений.

$$T_p = \frac{1}{3,78} = 0,27 \text{ года} \approx 4 \text{ месяцев}$$

Таблица 4.8 – Показатели сравнительной экономической эффективности от внедрения программного продукта

Наименование показателей	Значение
Условная годовая экономия затрат, тенге	6 735 134

Коэффициент экономической эффективности капитальных вложений (Ер)	1,16
Срок окупаемости капитальных вложений (Тр)	0,27

Изначально Система была создана как некий «калькулятор», размещенный в свободном доступе на сайте, которым могут пользоваться как клиенты ТОО «ЕТС-Тендер», так и его сотрудники.

Таким образом, при использовании Системы компания экономит 6 735 134 тенге, в случае, если работать с Системой будут сотрудники компании, то из-за накладных расходов экономия снизится на 302 400 тенге в год и составит 6 432 734 тенге.

При этом капиталовложения на ее создание составляют - 1 780 830 тенге. Таким образом, Система себя оправдывает через 4 месяца.

5 Безопасность жизнедеятельности

Темой данной дипломной работы является «Разработка системы «Микрослужбы прогнозирования веб-API». В результате разработки Системы, которая представляет собой некий калькулятор, предназначенный для пользования одной из категорий клиентов ТОО «ЕТС-Тендер».

Поясним, что ТОО «ЕТС-Тендер» является электронной торговой площадкой, на которой Компании-Заказчики (Заказчик) размещают заявки на покупку, а Компании-Поставщики (Поставщик), увидев подходящую для них заявку, выставляют в ответ свои предложения. Заказчик выбирает среди предложений нескольких Поставщиков самый выгодный для себя вариант и совершает сделку на приобретение товаров или услуг.

Система, разработанная в рамках дипломной работы, должна размещаться на сайте ets-tender.kz и использоваться Заказчиками. Таким образом, предполагается, что сами сотрудники ТОО «ЕТС-Тендер» не будут им пользоваться, поэтому не имеет смысла проектирование рабочих мест.

Однако, не исключена возможность того, что Заказчики могут просить совершать проверки своих заявок через Систему персоналом ТОО «ЕТС-Тендер». Если такой вариант осуществится, то ежедневно сотрудники ТОО «ЕТС-Тендер» будут тратить около 45 минут своего рабочего времени на проверку заявок Заказчика через Систему, а значит пользоваться персональным компьютером.

В настоящее время Заказчиков обслуживает два менеджера ТОО «ЕТС-Тендер». Таким образом, необходимо спроектировать два рабочих места для пользователей персональным компьютером.

5.1 Охрана труда в ТОО «ЕТС-Тендер»

Проанализируем системы «человек-машина-среда» (ЧМС) в ТОО «ЕТС-Тендер», в следующей последовательности «среда-машина-человек».

5.1.1 Риски для безопасной жизнедеятельности человека в помещении (среде)

Для обеспечения безопасной жизнедеятельности человека при работе в офисе и с использованием ПК необходимо основываться на государственных требованиях. Проанализируем помещение, в котором будет располагаться Система.

Важными факторами риска являются:

- параметры микроклимата;
- освещение на рабочем месте;
- класс помещения по опасности поражения электротоком;

- категорию помещения по взрыво- и пожароопасности.

Вредных производственных факторов в помещении не обнаружено. Помещение класса «Элит» (<https://www.elitstroy.kz/projects/pioner-3/>) расположено в бизнес-центре «Пионер-3», спроектированном строительной компанией «Элит-строй» специально для государственной компании - Национальный банк Республики Казахстан в 2008 году. Здание имеет серверную, компьютерную сеть, интернет, видеонаблюдение, пожарную сигнализацию, круглосуточную охрану, паркинг, витражные окна, климатические системы, не имеет цокольного этажа.

5.1.2 Соответствие Требованиям микроклимата помещения

Влажность воздуха и температура – одни их основополагающих факторов хорошего здоровья и самочувствия персонала.

Благодаря тому, что изначально здание оборудовано техническими средствами, то операторы ПК чувствуют себя комфортно круглый год в течение всего рабочего дня. Климатическое оборудование дает возможность установлены величины параметров микроклимата в соответствии с санитарными нормами СН-245-71, так как есть возможность и выставлять температуру воздуха + 22.24°C, поддерживать относительную влажность воздуха в пределах 40,60% и поддерживать скорость движения воздуха до 0,1м/с.

Поскольку в помещении изначально установлена климатическая система, то сотрудники могут самостоятельно менять интенсивность работы системы кондиционирования летом и системы отопления зимой, просто повернув стрелку климатической системы на нужную величину. Также в потолок встроена система вентилирования, которая позволяет круглые сутки иметь в здании свежий воздух. В потолке, к слову, также находится система пожаротушения.

Поскольку здание относится к бизнес центрам элит класса, то его обслуживают высококвалифицированные специалисты в области вентиляционного и климатического оборудования.

Таким образом, помещение ТОО «ЕТС-Тендер» соответствует пп. 23 и 24 пункта 3 Требованиям, предъявляемых к микроклимату.

5.1.3 Соответствие Требованиям к освещению помещения

Созданию комфортных условий способствует также определенные нормы освещения, которые не дают человеку испортить зрение или получить иной ущерб для своего здоровья.

Для обеспечения освещения помещения ТОО «ЕТС-Тендер» без бликов на рабочей поверхности монитора целесообразно использовать естественную и искусственную общую систему освещения.

Для обеспечения такого освещения в помещении размером 10 квадратных метров необходимо произвести расчет, сколько ламп (осветительных установок), какого светового потока они должны быть.

Рассчитаем необходимый световой поток одной лампы.

Чтобы определить необходимую величину светового потока (в люменах, Лм), используем формулу $X * Y * Z$, где X - норма освещенности объекта (Лк) (см. таблицу №1), Y – площадь помещения в м², Z – коэффициент поправки на высоту потолков.

Таблица 5.1 – Нормы освещенности

Тип помещения	Норма освещенности	Высота потолков	Коэффициент поправки на высоту
Офис общего назначения	300 Лк	2,5-2,7 м	1
Офис, где производятся чертежные работы	500 Лк	2,7-3 м	1,2
Конференц-зал, переговорные	200 Лк	3-3,5 м	1,5
Лестницы, подсобные помещения	50-75 Лк	3,5-4,5 м	2

X – равен 500 Лк, согласно пп 28 пункта 4 Требований, минимальная нормированная освещенность (E_{min}) рабочей поверхности стола должна быть не менее 400 люкс (Лк).

Y – 75 м²

Z – коэффициент поправки на высоту будет равен 1,2.

Таким образом, подставим в формулу цифры:

$500\text{Лк} * 10\text{м}^2 * 1,2 = 6000$ Люменов (Лм)

Чтобы рассчитать необходимое количество светильников, нужно разделить полученную цифру на величину светового потока выбранного типа лампы.

Мы выбрали светодиодную лампу трубку Т8 600 мм.8 Вт, так как является отличной заменой люминесцентных ламп ЛБ 20 Вт, 36 Вт. Для бизнес-центра типичным и целесообразным источником искусственного освещения до 2016 года являлись люминесцентные лампы, однако сейчас все чаще применяют светодиодные лампы. Они наиболее эффективные – светоотдача до 75 лм/Вт, хорошая цветопередача, низкая температура и долговечные - срок службы до 10000 ч.

Административный персонал здания может обеспечить хранение и утилизацию таких ламп, а также своевременную замену мигающих или шумящих экземпляров, а также имеют в штате электриков, которые могут починить их сложную пусковую аппаратуру. При меньшем потреблении они имеют дополнительные преимущества: вибростойкие, не мерцают, в светильнике не нужен дроссель и ПРА, долговечные, нет специальных требований к утилизации, не содержат вредных веществ и ртути.

Модернизация освещения с применением светодиодных ламп Т 8 600 мм очень проста, и состоит в замене лампы ЛБ на светодиодную в существующем светильнике типа ЛПО. Следует обратить внимание, что современные светодиодные лампы по размеру соответствуют люминесцентным и подходят для светильников ЛПО (Люминесцентный Потолочный для Общественного здания). Как раз светильники ЛПО уже используются в здании Бизнес-центра «Пионер-3».

Световой поток данной лампы в 8 Вт, согласно маркировке и данным производителя, равен 800 Лм. В одном светильнике используются четыре лампы, таким образом, световой поток одного светильника равен $800 \cdot 4 = 3200$ Лм. Следовательно, нам потребуется 2 светильника.

Кроме того, окна в комнате оборудованы жалюзи, которые применяются в случае избытка естественного света.

Отсюда следует, что помещение соответствует пп. 28-30 п. 4 Требований.

Также следует не забывать, что окна нужно мыть не реже двух раз в год, согласно пункту 6.15. «СанПиН 2.2.2/2.4.1340-03» для обеспечения требуемых норм естественного освещения.

5.1.4 Опасность поражения электрическим током

Переменный ток обладает напряжением в 220 В и частотой 50 Гц и изолирован прорезиненной обмоткой на проводах. Небрежное обращение с проводами может привести к нарушению изоляции проводки и может поразить электрическим током. Ущерб человеку не ограничивается только термическими ожогами, но и электрическим, автоматическим и биоповреждениями тканей и крови человека.

Как правило, офисные помещения по степени поражения людей электрическим током относятся к помещениям без повышенной опасности.

Согласно пункту 3.7 «СанПиН 2.2.2/2.4.1340-03» и пп 8 пункта 2 Требований [12], помещение ТОО «ЕТС-Тендер» оборудовано защитным заземлением (занулением) в соответствии с техническими требованиями по эксплуатации. Вблизи, в соответствии с пп 4 пункта 2 Требований, нет силовых кабелей и вводов, высоковольтных трансформаторов, технологического оборудования, создающего помехи в работе Системы.

5.1.5 Опасность возникновения пожара

Нарушения в электропроводке и короткие замыкания – основная причина возникновения пожара, риск которого увеличивается при скачках напряжения, перегрузке электросети, накоплении статического электричества.

Офисное помещение ТОО «ЕТС-Тендер» относится к категории «В» по взрыво- и пожароопасности, и является пожароопасным, имеет кабели и твердые пожароопасные вещества и материалы, которые могут возгореться.

В связи с пожароопасностью, комната оснащена, кроме пожарной системы, встроенной в потолок, двумя ручными углекислотными огнетушителями ОУ-5 и оснащена и аптечкой первой помощи. Их расположение указано на рисунке 5.1.

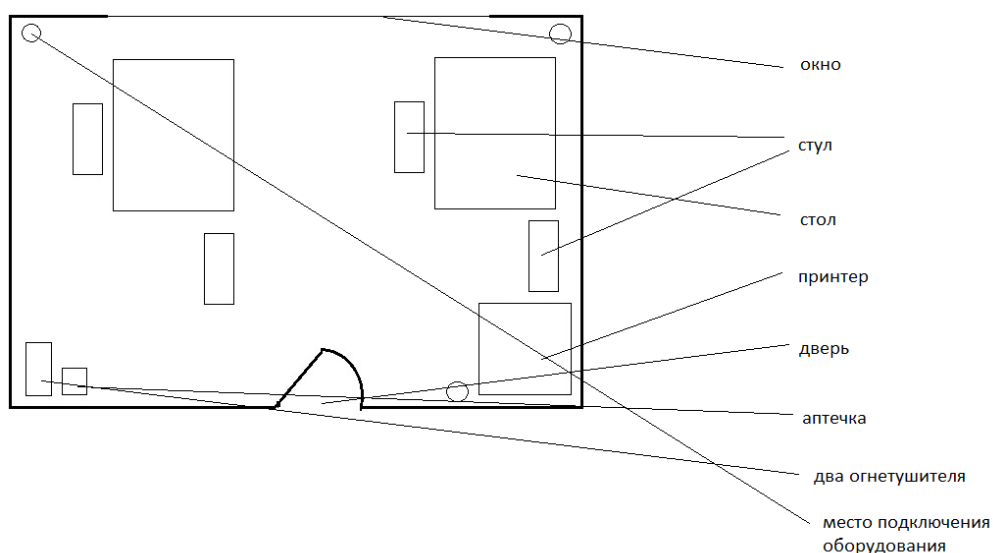


Рисунок 5.1 – Расположение мебели и защитных средств в помещении

5.1.6 Риски для безопасной жизнедеятельности человека, связанные с компьютером (машиной)

Работа на персональных компьютерах несет определенные риски для здоровья человека и при проектировании рабочего места для персонала, который будет на них работать, необходимо минимизировать вред и осуществлять охрану его труда, обеспечить безопасность его жизнедеятельности.

Перечислим известные факторы, которые следует сделать безопасными для здоровья человека при работе на ПК:

- уровни электромагнитных полей;
- акустического шума;
- концентрация вредных веществ в воздухе;
- визуальные показатели видеодисплейного терминала (ВДТ);

- мягкое рентгеновское излучение (для ВДТ с использованием электронно-лучевых трубок);
- превышение нормы излучения электромагнитных волн, акустического шума и концентрации вредных веществ в воздухе.

ПК излучают электромагнитные волны, доза которых может быть избыточной и привести к ухудшению здоровья. Звуки, издаваемые одним компьютером, не значительно влияют на человека, если находятся в диапазоне 30-68 дБ. Но более высокий шум обладает свойством кумуляции, иными словами, он может накапливаться и влиять на нервную систему, ускорять наступление утомления, может отвлекать и даже снижать интерес к работе, ухудшать память и слух. Концентрация вредных веществ в воздухе также может привести к неприятным для человека последствиям.

5.1.7 Соответствие Требованиям по нормам излучения электромагнитных волн, акустического шума и концентрации вредных веществ в воздухе

Современные персональные компьютеры приобретаются, согласно пп 20 п 2 Требованиям только при наличии документов, подтверждающих качество и соответствие всем нормам, предъявляемым им пп 31-33 п 5 Требованиям, в частности по допустимым значениям уровней неионизирующих электромагнитных излучений, допустимые уровни звукового давления в октавных полосах частот и уровня звука, создаваемого компьютерами и видеотерминалами и допустимым уровням концентраций аэроионов и коэффициента униполярности. Тем не менее, в предусмотренных Требованиями случаях, проводятся проверки оборудования на предмет их соответствия нормам.

Такой параметр, как мягкое рентгеновское излучение (для ВДТ с использованием электронно-лучевых трубок) в данной работе не рассматриваем, поскольку электронно-лучевые трубки в помещении не используются и не могут принести вред.

5.1.8 Меры, принимаемые сотрудниками для безопасной жизнедеятельности человека и охраны его труда (человек)

Монотонность труда. Действительно, труд на ПК очень часто автоматизирован и персоналу приходится заниматься рутинным, повторяющимся изо дня в день делом.

Повышенные психофизиологические нагрузки. Работа на ПК требует от юзера сосредоточенности и сконцентрированности. Постоянное сидение за столом и незначительное число поз, которые можно использовать при этом также ведет к физиологическим нагрузкам, при несоблюдении правил

безопасности жизнедеятельности может привести к изменению осанки, слабости мускулатуры, при стечении дополнительных факторов даже к ожирению, снижению зрения и многим другим последствиям.

В этой связи, согласно Требованиям пп 15,18 и 21 п 2, сотрудникам организуются 15 минутные перерывы через каждые 2 часа, соблюдаются нормы гигиены, влажной уборке на постоянной основе подвергаются все рабочие поверхности, пол.

5.2 Проектирование рабочего места оператора ТОО «ЕТС-Тендер»

5.2.1 Соответствие Требованиям размещения и эксплуатации ПК

Помещение находится на предпоследнем этаже 12-ти этажного здания, с окнами, выходящими на запад. Поэтому столы в помещении расставлены так, чтобы операторы сидели лицом на юг, таким образом, естественный свет освещал рабочий стол оператора с левой стороны (Схема №1).

Проектирование двух рабочих мест для сотрудников, которые имеют к доступ к Системе в помещении площадь на одного человека равна $10/2=5 \text{ м}^2$, что соответствует пп.5, пункта Требованиям (не менее 4 м^2 на одного человека), так как используют вычислительную технику (ВТ) на базе плоских дискретных экранов (жидкокристаллические, плазменные).

Таким образом, рабочие места двух операторов, выполняющих работу с Системой, расположены рядом в помещении на 11 этаже, что соответствует пп4 и пп9 пункта 2 Требованиям). Эти Требования позволяют избежать пагубного влияния неионизирующих электромагнитных излучений от монитора компьютера. Так как согласно СанПиНом 2.2.2.542-96, при 50-ти сантиметровом расстоянии между человеком и монитором компьютера напряженность электрической составляющей электромагнитного поля ниже или равна 10 В/м , напряженность магнитной составляющей электромагнитного поля на расстоянии $0,3 \text{ А/м}$ и эти показатели являются допустимыми. Кроме того, у современных компьютеров напряженность электростатического поля не превышает 20 кВ/м , что допустимо для взрослых людей.

Столешницы стоят в $0,5 \text{ м}$ от окна, а стол и принтер возле стены отстоят от него на расстоянии $0,4 \text{ м}$.

Согласно пп.6 и 7 пункта 2 Требованиям, помещение имеет евроремонт и покрытия, допускающие влажную уборку с применением моющих средств.

5.2.2 Выбор офисной мебели

Офисная мебель подобрана в соответствии пп.10-13 пункта 2 Требованиям. У каждого оператора есть свой отдельный стол и стул, один общий принтер стоит на отдельном столе, в удалении от каждого сотрудника.

Параметры мебели позволяют сотрудникам в течении дня легко менять позу, подбирать оптимальную высоту стула, в соответствии со своими физическими данными, расслабляться или наоборот концентрироваться.

В частности, выбрана высота стола – 750 мм, ширина рабочей поверхности – 1400 мм, пространство для ног высотой не менее 580 мм, шириной - не менее 500 мм, глубиной - не менее 450 мм. Экран находится также на оговоренном в нормах уровне 600 мм от глаз пользователя. Высота стула, наклон спинки регулируется и подбирается в соответствии с физическими данными пользователя. Кроме того, возле каждого стола стоит один дополнительный стул для посетителей, который не имеет колес, а также регулировки спинки и высоты как у оператора.

В комнате на отдельной тумбе стоит принтер. Так как в компании ЕТС-Тендер большинство процессов документооборота осуществляется в электронном виде, то необходимость в сканировании вообще отпадает (при этом в крайнем случае сканер стоит в общей доступности в приемной), а принтер нужен от случая к случаю, поэтому один аппарат вполне удовлетворяет потребности двух сотрудников, работающих в одном не большом помещении.

У каждого компьютера и принтера есть свой источник питания, а также помещение оборудовано защитным заземлением.

5.2.3 Выбор оборудования

Характеристики системного блока должны соответствовать современным требованиям: четырёхъядерный процессор (Intel Core i5), объем оперативной памяти – 4 ГБ, объем жесткого диска – 500 ГБ, наличие кулера задней панели, монитор подойдет жидкокристаллический, с угловым размером знака 30 угл.мин. Для использования системы не нужно дополнительное оборудование. Источник беспроводного питания в компании имеется, поскольку там все время проводятся онлайн операции, его номинальное выходное напряжение 230 В, эффективная мощность батареи 300 Ватт, имеется защита от перегрузки и высоковольтных импульсов, но он подключен к системному блоку в комнате технической поддержки.

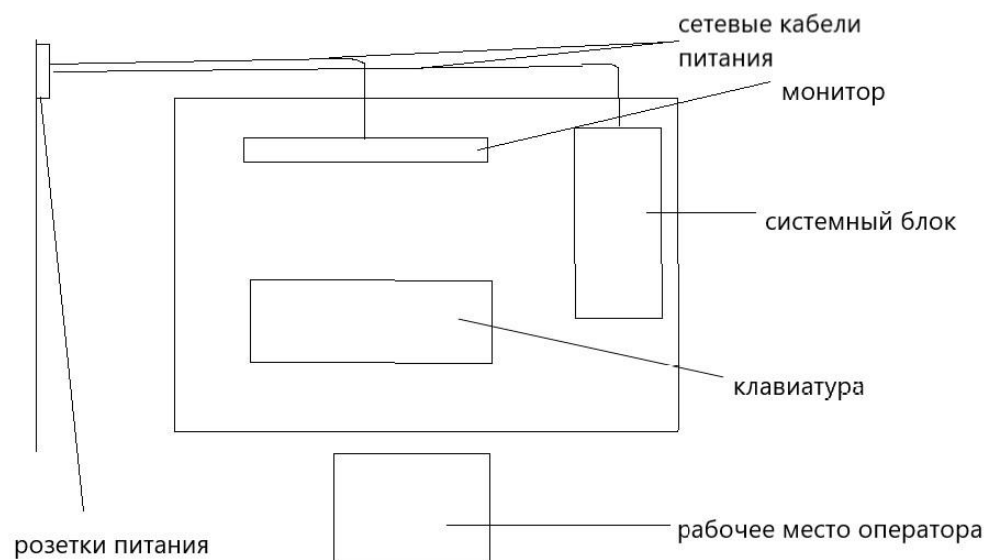


Рисунок 5.2 – Подключение оборудования

В офисном помещении три места для подключения оборудования, по две розетки на каждый стол для оператора и одна розетка для принтера.

В качестве средства защиты от прямого прикосновения к токоведущим частям оборудования используются ограждения, вход за которые только после вскрытия его специальным ключом.

Заключение

Благодаря специальности «Информатика» и наличию навыков разработки систем, удалось создать систему микрослужбы прогнозирования завершения процедур закупок, которая смогла улучшить экономические показатели компании, а в дальнейшем и получить прибыль. Это стало возможным благодаря качественному обслуживанию клиентов и созданию нового сервиса, который предвосхищает потребности клиентов. Создание такого конкурентного преимущества и укрепление лояльного отношения клиентов к компании – один из важных факторов электронной коммерции, поскольку это один из самых высококонкурентных рынков.

Первым этапом работы был анализ первичных данных, собранных компанией за два с половиной года работы. Как и предполагалось изначально, он показал, что часть незавершенных процедур заведомо были некачественными и их можно было бы отбраковать на этапе создания. Таким образом, стала очевидной необходимость создания системы прогнозирования завершения процедур закупок.

В ходе анализа также были выявлены ключевые факторы, по которым становится ясно, что существуют коридор, выйдя за рамки которого

Была выбрана технология создания системы в виде системы микрослужб. Мониторинг информации о современных технологиях на которых работают передовые мировые компании показал, что очень часто их выбор останавливался на использовании систем микрослужб.

Архитектура систем микрослужб отвечает всем требованиям, которые изначально были поставлены перед системой. Во-первых, нет необходимости ее встраивать в существующий монолит системы электронных торгов. Во-вторых, она может справиться с большим числом новых заявок и не требует масштабирования, а значит в дальнейшем не потребует дополнительных работ по усовершенствованию в этом плане.

Система микрослужб создана так, чтобы отвечать поставленной задаче, выявить на этапе создания закупок те, прогноз по завершению которых не высок. Она работает отдельно от электронной системы торгов и не потребовала встройки. Сама технология создания микрослужб позволяет в дальнейшем легко справляться с большим потоком заявок.

Все данные были распределены на несколько категорий и выяснены причины, по которым значительное число заявок, по которым начинается процедура совершения сделки, так и не завершается. Выделенные параметры выведены в интерфейс микрослужбы.

Кроме того, немаловажно, чтобы интерфейс веб-страницы, к которой привязана микрослужба, имела дружелюбный характер, спокойные цветовые решения, поскольку сотрудники должны проводить в ней около 30-40 минут своего рабочего времени, находясь в сосредоточенном сконцентрированном состоянии.

Сервис представляет собой форму, в которую можно внести ряд основных данных, которые микрослужба анализирует и выдает ответ, в котором указывается с какой величиной вероятности будет завершена тендерная процедура закупки. При этом форма эта очень простая – состоит из шести окон, названных так, как и сами данные, благодаря этому интуитивно понятно пользование системой. И в форме есть кнопка, нажав на которую юзер может получить тут же ответ, который сервис дает автоматически. Таким образом, воплощена важная задача – создание простого удобного сервиса для пользователя.

Исключая на этапе создания некачественные заявки, любая торговая электронная площадка не будет больше тратить напрасно время и труд сотрудников. В результате внедрения системы микрослужб повысится эффективность деятельности электронной торговли из-за существенного повышения объема обрабатываемых заявок, по которым будет выхлоп: процесс заключения сделки завершится, при этом оценка качества создаваемой заявки проводится за считанные минуты. Создание микрослужбы ведет к существенному повышению конкурентоспособности компании и лояльности клиентов к ней. А это значит, что компания будет получать прибыль, а это главная цель любой коммерческой структуры.

Список литературы

- 1 Ричардсон К. Микросервисы. Паттерны разработки и рефакторинга. - М.: Питер, 2019. - 44 с.
- 2 Ньюмен С. Создание микросервисов - М.: Питер, 2016 – 78 с.
- 3 Рассел С. Искусственный интеллект: современный подход. - М.: Вильямс, 2016. - 578 с.
- 4 Лутц М. Программирование на Python, I том. - М.: Символ-плюс, 2015. - 992 с.
- 5 Крокфорд Д. JavaScript. Сильные стороны. - М.: Питер, 2016. - 262 с.
- 6 Кабаков Р. R в действии. - М.: ДМК-Пресс, 2014. - 588 с.
- 7 Страуструп Б. Язык программирования C++. Краткий курс. - М.: Бином, 2019. - 320 с.
- 8 Блох Д. Java. Эффективное программирование. - М.: Диалектика, 2019. - 464 с.
- 9 Гайсин Р.С. Экономическая теория. - М.: НИЦ ИНФРА-М, 2013. - 152 с.
- 10 Гродский В.С. Экономикс: учебное пособие для бакалавров. - М.: ИНФРА-М, 2013. - 220 с.
- 11 Боканова Г.Ш. – Методические указания по выполнению экономической части дипломных работ для студентов специальностей 5В070400 – «Вычислительная техника и программное обеспечение», 5В060200 – «Информатика», 5В070300 – «Информационные системы». Алматы, АУЭС, 2020. - 35 с.
- 12 СЭТ РК 21-01-2015 Санитарно-эпидемиологические требования к условиям работы с источниками физических факторов (компьютеры и видеотерминалы), оказывающих воздействие на человека.
- 13 Артамонов И. В. Показатели производительности микросервисных систем // Вестник НГИЭИ. 2018. № 8 (87). с 24–33.
- 14 Martin Fowler, James Lewis // Microservices // URL: <https://martinfowler.com/articles/microservices.html> (дата обращения: 25.03.2014).
- 15 Аналитический отчет об электронной торговле в Казахстане // Официальный информационный ресурс премьер-министра РК // URL: <https://primeminister.kz/ru/news/v-2019-godu-obem-pokupok-v-kazahstanskih-internet-magazinah-sostavil-422-mlrd-tenge> (дата обращения: 18.02.2020).
- 16 "Неофлекс" создала цифровую логистическую платформу для компании ПЭК // COMNEWS.RU: Цифровая экономика // URL: <https://www.comnews.ru/digital-economy/content/119834/2019-05-28/neofleks-sozdala-cifrovuyu-logisticheskuyu-platformu-dlya-kompanii-pek> (дата обращения: 28.05.2019).
- 17 Genesys использует архитектуры на базе микросервисов // club.cnews.ru: Блоги экспертов и ИТ компаний // URL:

https://club.cnews.ru/blogs/entry/bolee_1200_kompanij_po_vsemu_miru_pereshli_na_resheniya_genesys (дата обращения: 27.03.2019).

18 Lloyds готовится к переходу к открытому банкингу // computerworld.ru:ИТ-индустрия//URL:

<https://www.computerworld.ru/articles/Lloyds-gotovitsya-k-perehodu-k-otkrytomu-bankingu> (дата обращения: 18.04.2017).

Приложение А

Техническое задание

1 Общие сведения

1.2 Наименование системы

1.2.1 Полное наименование системы

Полное наименование:

Разработка системы «Микрослужбы прогнозирования веб-API»

1.2.2 Краткое наименование системы

Краткое наименование:

«Микрослужбы прогнозирования веб-API».

1.2.3 Основания для проведения работ

Работа выполняется на основании договора от 18.05.2019 между руководством ТОО «ЕТС-Тендер» и Унгаровой А.Б.

1.2.4 Наименование организаций Заказчика и Разработчика

1.2.4.1 Заказчик

Заказчик: ТОО «ЕТС-Тендер»

1.2.4.2 Разработчик

Разработчик: Унгарова Алиман Болатовна

1.2.5 Источники и порядок финансирования

Финансирование для данной системы необходимо для разработки и составляет 1 780 830 тенге. Финансирование также необходимо при последнем этапе проекта – при тестировании и отладке. Источником финансирования является ТОО «ЕТС-Тендер».

Продолжение приложения А

1.2.6 Плановые сроки начала и окончания работы

Работа проводится в соответствии графику, представленному в таблице

1.

Таблица А.1 – Плановые сроки работы

№	Наименование работы	Начало работы	Конец работы
1	Анализ существующих микрослужб	09.01.2020	21.01.2020
2	Проектирование функциональной структуры системы	22.01.2020	02.02.2020
3	Разработка бизнес-модели	03.02.2020	12.02.2020
4	Проектирование информационного обеспечения системы	13.02.2020	24.02.2020
5	Разработка прототипа системы	25.02.2020	29.02.2020
6	Разработка программного обеспечения системы	01.03.2020	07.03.2020
7	Тестирование и отладка программного обеспечения системы	08.03.2020	10.03.2020

1.2.7 Порядок оформления и предъявления заказчику результатов работ

Работы по созданию данной системы предоставляются Разработчиком согласно календарному плану проекта поэтапно.

Результаты работ предъявлялись заказчику в виде документов, которые содержат скриншоты проделанных работ, а также исполняемый файл программы, который устанавливался на компьютеры заказчика с целью демонстрации функционала программы.

1.3 Назначение и цели создания (развития) системы

1.3.1 Назначение системы

На этапе создания заявок Заказчика, зная ее основные данные, понять, будет ли процесс закупки завершен или нет. Исключая на этапе создания некачественные заявки, компания ТОО «ЕТС-Тендер» экономит время и труд

сотрудников, зарабатывает, использовав сохраненные часы на создание качественных заявок.

Продолжение приложения А

1.3.2 Цели создания системы

Повышение эффективности деятельности электронной торговли. Практическая ценность заключается в существенном повышении объема обрабатываемых заявок, по которым процесс заключения сделки завершается. При этом оценка качества создаваемой заявки проводится за считанные минуты. Создание микрослужбы ведет к существенному повышению конкурентоспособности компании и лояльности клиентов к ней. Ведь тендерная процедура занимает в среднем семь-десять дней.

1.4 Характеристика объектов автоматизации

Объектом автоматизации является обработка заявок Заказчика, часть которых не доходит до логического завершения и сделка не происходит. Данная же система экономит время сразу трёх сторон: заказчика, поставщика и самой компании, автоматизируя процесс обработки заявок и узнавая, будет ли процесс закупки завершён. В результате внедрения микрослужбы повысится эффективность деятельности электронной торговли.

1.5 Требования к системе

1.5.1 Требования к системе в целом

Взаимодействия с программой происходят с помощью управления программой мышью или клавиатурой, и, вводя исходные данные для расчётов.

Основной интерфейс системы включает в себя:

- область ввода данных;
- кнопку передачи данных;
- область вывода вероятности.

Область ввода данных состоит из следующих окон:

- стоимость закупки;
- наличие предоплаты;
- размер предоплаты;
- срок поставки;
- попозиционно;

- количество позиций.

После ввода всех исходных данных пользователем нажимается кнопка «Submit», которая позволяет отправить введенные данные для подсчета вероятности. Если все данные введены, то выводится окно успешного тендера с прогнозом.

Продолжение приложения А

1.5.2 Требования к функциям, выполняемым системой

1) Ввод данных.

Ввод данных происходит в области ввода данных на главном окне программы.

2) Вывод данных.

Вывод данных происходит в виде итогового результата прогноза тендера, то есть вероятность успешного тендера.

1.5.3 Требования к ПО

Микрослужбы прогнозирования веб-API на платформу Windows необходимо разработать в среде разработки «Spyder» на языке программирования Python. Минимальная версия операционной системы для работы программы – Windows 8 64-bit. Рекомендуемая версия операционной системы – Windows 10 64-bit.

1.6 Состав и содержание работ по созданию системы

Создание системы выполняется в несколько этапов:

- разработка эскиза проекта;
- проектирование проекта;
- разработка технической части проекта.

Более подробные сведения об этапах разработки программы можно узнать в пункте 1.2.6 – «Плановые сроки начала и окончания работы».

1.7 Источник разработки

Настоящее Техническое Задание разработано на основе следующих документов и информационных материалов:

- договор от 18.05.2019 между руководством ТОО «ЕТС-Тендер» и Унгаровой А.Б.

Приложение Б

Листинг программы

```
import pandas as pd
from flask import Flask, request
import json
import hashlib
import pickle
from flask_cors import CORS, cross_origin

def convert_sample_from_dict_to_df_row(sample_as_dict):
    sample_as_dict_convertable_to_df = { }
    for sample_as_dict_key in sample_as_dict:
        sample_as_dict_convertable_to_df[sample_as_dict_key] =
[sample_as_dict[sample_as_dict_key]]
    df_row = pd.DataFrame.from_dict(sample_as_dict_convertable_to_df)
    return df_row

def get_test_samples_features_df():
    test_df = pd.read_csv("X_test.csv")
    return test_df

def integerFromString(inputString):
    inputStringHashed=hashlib.sha224(inputString.encode("utf-
8")).hexdigest()
    inputStringHashedAsInt=int(inputStringHashed,16)
    return inputStringHashedAsInt % 10000

def get_hash_based_ID(dict_to_hash):
    temp_hash = integerFromString(str(dict_to_hash))
    return str(temp_hash)

class api_logic:

    def __init__(self):
        self.current_model_name = "tender_clf"
        model_file = open(self.current_model_name+".joblib", "rb")
        self.mlp = pickle.load(model_file)
        model_file.close()
        self.predictions_store = { }

    def add_prediction(self, test_sample_as_dict):
        test_sample_as_df_row =
convert_sample_from_dict_to_df_row(test_sample_as_dict)
        prediction = self.mlp.predict_proba(test_sample_as_df_row)[0][0]
```

```

print(prediction)
return prediction

def get_prediction(self, test_sample_as_dict):
    sample_as_dict_hashed = get_hash_based_ID(test_sample_as_dict)
    if sample_as_dict_hashed in self.predictions_store:
        prediction = self.predictions_store[sample_as_dict_hashed]

        Продолжение приложения Б
    else:
        prediction = self.add_prediction(test_sample_as_dict)

self.predictions_store[sample_as_dict_hashed] = prediction
return prediction, sample_as_dict_hashed

def load_model(self, new_model_file_name):
    model_file = open(self.current_model_name+".joblib", "rb")
    self.mlp = pickle.load(model_file)
    model_file.close()
    self.current_model_name = new_model_file_name
    self.predictions_store = { }
    model_loading_result_dict = { }
    model_loading_result_dict["model_loading_result"]="The model " +
new_model_file_name + " has been loaded successfully."
    return model_loading_result_dict

api_logic=api_logic()
app = Flask(__name__)
CORS(app, supports_credentials=True)

@app.route("/customer-record", methods=["POST"])
@cross_origin(supports_credentials=True)
def new_rating():
    new_test_sample = request.get_json()
    new_prediction, customer_ID =
api_logic.get_prediction(new_test_sample)
    customer_record = { }
    customer_record["customer_ID"]=customer_ID
    return json.dumps(customer_record), 201

@app.route("/customer-prediction/<customer_ID>", methods=["GET"])
def get_prediction_for_customer(customer_ID):
    prediction_dict = { }

```

```

        prediction_dict["probability-of-tender"]
api_logic.predictions_store[customer_ID]
        return json.dumps(prediction_dict), 200

@app.route("/model", methods=["PUT"])
def set_model():
    new_model_metadata = request.get_json()
    new_model_file_name = new_model_metadata["new_model_file_name"]
    model_loading_result_dict
api_logic.load_model(new_model_file_name)
    return json.dumps(model_loading_result_dict), 200

app.run(host="127.0.0.1", port=9875)
import React, { useState } from 'react'
import HomeStyled from '../styled/pages/HomeStyled'
import Form from '../molecules/Form'
import Input from '../atoms/Input'
import Text from '../atoms/Text'
import { connect } from 'react-redux'
import passData from '../../misc/services/dataService'
        Продолжение приложения Б
import {
    costChange,
    prePayChange,
    prePayAmountChange,
    timeChange,
    popositionsChange,
    amountOfPositionsChange
} from '../../redux/actions/inputActions'
const Home = ({
    cost,
    prePay,
    prePayAmount,
    time,
    popositions,
    amountOfPositions
}) => {
    const [chance, setChance] = useState(0)
    console.log(chance)
    const handleSubmit = () => {
        passData(
            cost,
            prePay,
            prePayAmount,

```

```

    time,
    popositions,
    amountOfPositions
  )
  .then(response => {
    console.log(response)
    setChance(Math.ceil(response['probability-of-tender'] * 100))
  })
}
return (
  <HomeStyled>
    <div style={{ gridRow: '1 / 3', backgroundColor: '#99D8CD' }}>
      <Form>
        <Input
          type = 'text'
          placeholder = 'Стоимость закупки'
          callback = { value => costChange(value) }
        />
        <Input
          type = 'text'
          placeholder = 'Наличие предоплаты'
          callback = { value => prePayChange(value) }
        />
        <Input
          type = 'text'
          placeholder = 'Размер предоплаты'
          callback = { value => prePayAmountChange(value) }
        /> <Input

```

Продолжение приложения Б

```

type = 'text'
  placeholder = 'Срок поставки'
  callback = { value => timeChange(value) }
/>
<Input
  type = 'text'
  placeholder = 'Попозиционно'
  callback = { value => popositionsChange(value) }
/>
<Input
  type = 'text'
  placeholder = 'Количество позиций'
  callback = { value => amountOfPositionsChange(value) }
/>
</Form>

```

```

    </div>
    <div style={{ backgroundColor: '#E2C0C1', boxSizing: 'border-
box', display: 'flex', justifyContent: 'center', alignItems: 'center' }}>
      <Text
        size = 'large'
      >
        Вероятность успешного тендера: { chance }%...
      </Text>
    </div>
    <div style={{ backgroundColor: '#F87596', gridArea: '2 / 2 / 2 / 2', dis
play: 'flex' }}>
      <Input
        clickCallback = { handleSubmit }
        type = 'button'
        text = 'submit'
        height = '40%'
        width = '50%'
      />
    </div>
  </HomeStyled>
)
}

```

```

const mapStateToProps = (state) => {
  return {
    cost: state.inputChange.cost,
    prePay: state.inputChange.prePay,
    prePayAmount: state.inputChange.prePayAmount,
    time: state.inputChange.time,
    popositions: state.inputChange.popositions,
    amountOfPositions: state.inputChange.amountOfPositions,
  }
}

```

```

export default connect(mapStateToProps) (Home)

```

Приложение В

Акт внедрения

«ETS-Тендер»
жауапкершілігі
шектелуі серіктестірі



Товарищество с ограниченной
ответственностью
«ETS-Тендер»

Қазақстан Республикасы, 050051
Алматы қ., Достық д-ы, 136
Тел.: +7 727 244 57 80
Факс.: +7 727 244 57 79
e-mail: ets@ets.kz
www.ets-tender.kz

Республика Казахстан, 050051
г. Алматы, пр. Достык, 136
Тел.: +7 727 244 57 80
Факс.: +7 727 244 57 79
e-mail: ets@ets.kz
www.ets-tender.kz

Republic of Kazakhstan, 050051
Almaty, Dostyk ave., 136
Phone: +7 727 244 57 80
Fax: +7 727 244 57 79
e-mail: ets@ets.kz
www.ets-tender.kz

Исх. № _____

на № _____ от _____

АКТ о внедрении системы

Настоящий Акт свидетельствует, что система, разработанная Унгаровой Алиман Болатовной, внедрена компанией ТОО «ETS-Tender»

Процесс внедрения проходил с 20 января по 10 марта 2020 г.

Заявленные характеристики системы предполагали наличие следующих основных возможностей:

- интерфейс в виде сайта-одностраничника;
- прогнозирование завершения процесса закупки;

В ходе эксплуатации системы подтверждено, что она обладает всеми заявленными возможностями и позволяет активно пользоваться программой.

На момент подписания настоящего Акта система установлена и применяется в работе компании.

Директор

ТОО «ETS-Tender»



Н.А. Ахтырская