

Пермский филиал федерального государственного автономного образовательного учреждения высшего образования
«Национальный исследовательский университет
«Высшая школа экономики»

Вечерне-заочный факультет экономики и управления

**«РАЗРАБОТКА ПРОГРАММНОГО ПРОДУКТА ДЛЯ
ОБЕСПЕЧЕНИЯ РАБОТЫ СИСТЕМЫ ВЕНДИНГОВЫХ
АППАРАТОВ ДЛЯ КРАТКОСРОЧНОЙ СДАЧИ ПЛЕДОВ В
АРЕНДУ»**

КОНЦЕПЦИЯ ПРОЕКТА

Пермь, 2023

СПИСОК ИСПОЛНИТЕЛЕЙ

| | | |
|---------|-------|----------------|
| Студент | _____ | Д. А. Зубарев |
| Студент | _____ | Н. Е. Копылов |
| Студент | _____ | Н. А. Калиман |
| Студент | _____ | Н. А. Перминов |
| Студент | _____ | Е. А. Рудакова |

СОДЕРЖАНИЕ

| | | |
|----------|--|-----------|
| 1 | Необходимость проекта | 4 |
| 1.1 | Обоснование необходимости | 4 |
| 1.1.1 | Обоснование реализации стартапа | 4 |
| 1.2 | Видение проекта | 4 |
| 1.3 | Анализ выгод..... | 4 |
| 1.3.1 | Финансовые выгоды..... | 4 |
| 1.3.2 | Нематериальные выгоды | 5 |
| 2 | Концепция решения..... | 6 |
| 2.1 | Цели и Задачи | 6 |
| 2.2 | Предположения и Ограничения | 6 |
| 2.3 | Анализ использования | 7 |
| 2.3.1 | Пользователи | 7 |
| 2.3.2 | Сценарии использования | 8 |
| 2.4 | Требования | 11 |
| 2.4.1 | Требования пользователей..... | 11 |
| 2.4.2 | Системные требования | 13 |
| 3 | Рамки | 15 |
| 3.1 | Функциональность решения | 15 |
| 3.2 | За рамками решения | 16 |
| 3.3 | Критерии одобрения решения..... | 16 |
| 4 | Стратегии дизайна решения..... | 18 |
| 4.1 | Стратегия архитектурного дизайна | 18 |
| 4.2 | Стратегия технологического дизайна | 20 |

1 Необходимость проекта

1.1 Обоснование необходимости

Необходимость проекта обусловлена потребностью создания собственной программно-аппаратной системы для реализации коммерческого проекта по краткосрочной аренде через вендинговые аппараты.

Не существует аналогичного программного обеспечения, обладающего подобным функционалом, так как стартап реализует инновационную бизнес-модель.

1.1.1 Обоснование реализации стартапа

Для тех, кто хочет провести время на свежем воздухе, предлагается оптимальное решение проблемы – посуточная аренда пледов через приложение и вендинговые автоматы, не приобретая пледы, которые будут использоваться однократно. С помощью данного проекта проблема холода во время прогулки может быть решена на время, так как пледы обогревают на несколько часов, позволяя продолжить прогулку. >

1.2 Видение проекта

Проект направлен на разработку в период с 1 сентября 2023 года по 24 марта 2024 года программно-аппаратной системы, которая позволит реализовать автоматизированную краткосрочную аренду пледов через вендинговые аппараты, а именно выдачу и прием пледов, сбор использованных пледов.

1.3 Анализ выгод

1.3.1 Финансовые выгоды

Разработка программно-аппаратного продукта позволит реализовать коммерческий проект, по финансовой модели которого средний доход с 1 автомата

составит 120 тысяч рублей в месяц, при рентабельности до 50% в сезон или 25% за год.

Бизнес-модель заключается в создании нескольких тарифов, при этом себестоимость одной покупки составляет 56 рублей. Средний чек - двести рублей. При этом точкой безубыточности будет продажа 12 пледов в день на один автомат, а рассчитываемое среднее количество покупок 22.

Оценка рынка: TAM составляет 20 млрд рублей в год, SAM - 2 млрд рублей, SOM - 1 млрд рублей.

1.3.2 Нематериальные выгоды

В ходе реализации проекта команда разработки получает опыт работы с коммерческим проектом.

Backend-разработчики получают опыт участия в разработке перспективного продукта, развивают и закрепляют навыки работы с неизвестными технологиями, заносят крупный проект в портфолио.

Full-stack разработчики получают опыт участия в разработке перспективного продукта, развитие и закрепление навыков работы с неизвестными технологиями, разработки клиентской и серверной частей коммерческого продукта, создания полноценного продукта с нуля, заносят крупный проект в портфолио.

2 Концепция решения

2.1 Цели и Задачи

Сделать возможным установку на улицах и в парках города вендинговых аппаратов для автоматизированной автономной краткосрочной аренды (шеринга) пледов. Для этого необходимо:

- провести анализ программных решений, позволяющих реализовать распределенное управление множеством устройств
- провести анализ программного обеспечения вендинговых аппаратов
- провести анализ программных возможностей реализации сайта, адаптированного под мобильные устройства
- провести анализ архитектурных решений, позволяющих распределить нагрузку (на сайт заходит много пользователей, покупку осуществляет только 1%)
- спроектировать систему, позволяющую реализовать бизнес-модель
- разработать информационную систему
- протестировать информационную систему
- запустить в тестовую эксплуатацию информационную систему с использованием одного вендингового аппарата
- применить систему в рамках бизнес-модели

2.2 Предположения и Ограничения

Старт происходит в городе Москва из-за высокой концентрации капитала.

Источник финансирования – Средства гранта “Студенческого стартапа”; ограничение бюджета – 1 000 000 рублей. Технологии, используемые в разработке, должны быть свободно распространяемые и не должны быть подвержены влиянию санкций. Временные ограничения обусловлены техническим заданием,

они не должны быть позже 31 марта 2024 года и должны соответствовать календарному плану проекта.

Разработка технического задания должна осуществляться в соответствии с законодательством РФ: № 152-ФЗ “О персональных данных”, № 395-1 “О банках и банковской деятельности”, № 115-ФЗ “О противодействии легализации (отмыванию) доходов, полученных преступным путем, и финансированию терроризма”.

Санкционные ограничения не позволяют выложить мобильное приложение в сервисы *play market* и *app store*, поэтому было принято решение разработать веб-решение с интерфейсом мобильного приложения.

На этапе разработки системы отсутствует доступ к вендинговому аппарату. Применение аппарата должно производиться на этапе тестовой эксплуатации.

Необходимо наладить взаимодействие разработанного программного обеспечения и аппаратов.

2.3 Анализ использования

2.3.1 Пользователи

Определим пользователей системы, которые будут задавать сценарии использования.

Клиенты коммерческого проекта – люди, гуляющие вечерами, когда на улице начинает холодать, а форма одежды была выбрана на более теплое время суток.

Специалисты технического обслуживания автоматов – специалисты, задачами которых является поддержка рабочего состояния автоматов и выкладка/сбор пледов.

Системные администраторы – специалисты, занимающиеся поддержкой работы программно-аппаратного комплекса.

Аналитики – специалисты, занимающиеся анализом статистики взятий и возвратов пледов за определенный период.

2.3.2 Сценарии использования

После определения сценариев использования будет возможно построение диаграмм прецедентов.

2.3.2.1 *Для клиентов*

Были выделены следующие сценарии использования для не аутентифицированного клиента коммерческого проекта:

- Отобразить информацию о местоположении автоматов на карте.
- Отобразить информацию об остатке пледов в автомате на карте.

Были выделены следующие сценарии использования для аутентифицированного клиента коммерческого проекта:

- Отобразить информацию о пледях, находящихся в аренде у клиента в данный момент: тариф, время использования, общая цена аренды.
- Взять плед в аренду: выбрать автомат, выбрать ячейку, открыть ячейку, забрать плед, закрыть ячейку.
- Добавить банковскую карту в список карт для оплаты: выбрать операцию добавления новой карты, ввести данные карты (номер, cvv, дата истечение срока действия), подтвердить добавление карты пробным списанием.
- Определить банковскую карту как стандартную для оплаты: выбрать операцию определения карты по умолчанию, выбрать необходимую карту.
- Вернуть плед: выбрать автомат, выбрать ячейку, открыть ячейку, положить плед, закрыть ячейку.

По итогу выделения сценариев использования была построена UML диаграмма прецедентов представленная на рисунке [2.1](#).



Рисунок 2.1 – Диаграмма прецедентов, описывающая взаимодействие с системой клиентом коммерческого проекта

2.3.2.2 Для специалистов технического обслуживания автоматов

Были выделены следующие сценарии использования для специалиста технического обслуживания автоматов:

- Изъять использованные пледы из автомата: выбрать автомат, открыть ячейки, отметить наличие/отсутствие пледов в ячейке, забрать пледы, закрыть ячейки.
- Вложить чистые пледы в автомат: выбрать автомат, открыть ячейки, положить чистые пледы, закрыть ячейки.

- Открыть конкретную ячейку автомата: выбрать автомат, в рамках автомата выбрать ячейку по номеру, запустить операцию открытия ячейки.
- Заккрыть конкретную ячейку автомата: выбрать автомат, в рамках автомата выбрать ячейку по номеру, выбрать ячейку, запустить операцию закрытия ячейки.

По итогу выделения сценариев использования была построена UML диаграмма прецедентов представленная на рисунке 2.2.

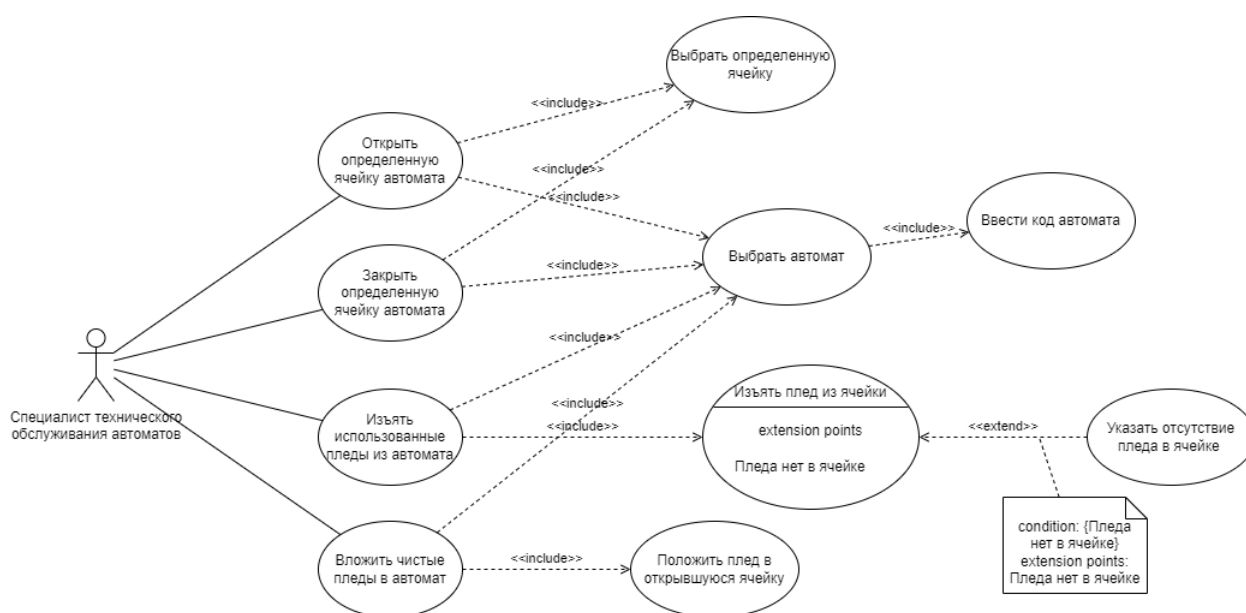


Рисунок 2.2 – Диаграмма прецедентов, описывающая взаимодействие с системой специалистом технического обслуживания

2.3.2.3 Для аналитиков и системных администраторов

Были выделены следующие сценарии использования для аналитика и системного администратора:

- Отобразить статистику взятий в аренду из автоматов за определенный период.
- Отобразить статистику возвратов пледов в автоматы за определенный период.
- Экспорт статистики в файл формата CSV.

По итогу выделения сценариев использования была построена UML диаграмма прецедентов представленная на рисунке 2.3.



Рисунок 2.3 – Диаграмма прецедентов, описывающая взаимодействие с системой аналитиком и системным администратором

2.4 Требования

2.4.1 Требования пользователей

2.4.1.1 Для клиентов

На основе выделенных сценариев использования были выявлены требования пользователей клиентов коммерческого проекта:

- Ячейка должна закрываться по указанию пользователя.
- Выбор автомата должен производиться посредством перехода по уникальной ссылке.
- Система должна отображать номер ячейки автомата при взаимодействии клиента с ней.
- Система должна отображать карту с указанием местоположения автоматов и пользователя.

- Система должна отображать информацию о количестве доступных пледов в автоматах на карте.
- Система должна отображать информацию о пледах, находящихся в аренде пользователем на данный момент, а именно тариф, время пользования, общая цена аренды.
- Система должна отображать данные пользователя (номер телефона) на основной странице.
- Система должна позволять добавлять банковскую карту для оплаты с основной страницы.
- Система должна позволять указывать основную банковскую карту для оплаты.
- Пользователь должен иметь возможность выбрать автомат посредством ввода уникального кода автомата.
- Пользователь должен иметь возможность взять в аренду до 5 пледов.
- Пользователь должен иметь возможность убедиться в том, что он выбрал правильный автомат посредством отображения системой фотографии автомата.
- Система должна отображать список возможных тарифов для аренды пледа.
- Система должна позволять выбрать один из возможных тарифов для аренды пледа.
- Система должна позволять выбрать основную банковскую карту до оплаты тарифа.

2.4.1.2 *Для специалистов технического обслуживания автоматов*

На основе выделенных сценариев использования были выявлены требования пользователей специалистов технического обслуживания автоматов:

- Система должна отображать номер ячейки автомата при взаимодействии специалиста с ней.

- Система должна позволять выбирать автомат один раз для выполнения нескольких операций.
- Система должна организовать изъятие использованных пледов из автомата.
- Система должна позволять отмечать отсутствие пледа в ячейке (данное действие требует подтверждения).
- Система должна организовать выкладывание чистых пледов в автомат.
- Система должна позволять открывать определенную ячейку автомата.
- Система должна позволять закрывать определенную ячейку автомата.

2.4.1.3 *Для аналитиков и системных администраторов*

На основе выделенных сценариев использования были выявлены требования пользователей аналитиков и системных администраторов:

- Система должна отображать графики по количеству взятий пледов в аренду из определенного автомата за определенный период.
- Система должна отображать графики по количеству возвратов пледов в определенный автомат за определенный период.
- Система должна иметь возможность экспортировать данные в формате CSV.

2.4.2 Системные требования

На основе выделенных сценариев использования и анализа концепции решения были выявлены следующие системные требования:

- Система должна позволять блокировать доступ к определенной ячейке автомата.
- Система должна проводить авторизацию специалистов технического обслуживания, аналитиков, системных администраторов.
- Система должна идентифицировать пользователя с помощью номера телефона.
- Система должна подтвердить, что номер телефона пользователя принадлежит ему.

- Открытие ячейки должно происходить не позднее, чем через 1 секунду после оплаты.
- Система должна позволять пользователю открывать только одну ячейку за один запрос.
- Система должна позволять устанавливать таймеры для операций взятия и сдачи пледа, которые не будут считаться в время аренды.
- Система должна быть реализована в соответствии с федеральными законами РФ: № 152-ФЗ “О персональных данных”, № 395-1 ”О банках и банковской деятельности”, № 115-ФЗ “О противодействии легализации (отмыванию) доходов, полученных преступным путем, и финансированию терроризма”.
- Авторизация персонала должна производиться посредством протокола *LDAP*.

3 Рамки

Рассмотрим границы решения:

- Функциональные границы: автоматизация процесса аренды пледов.
- Географические границы: вендинговые аппараты, выдающие и принимающие пледы. Аппараты находятся в парках города Москва. Автоматы берутся в аренду с ПО от производителя, далее это ПО может быть модифицировано или заменено. Разрабатываемое ПО подключается к арендуемому автомату. Тестирование будет производиться в зависимости от того, удастся ли получить экземпляр вендингового аппарата. Если не удастся, то тестирование будет производиться посредством моделирования реального поведения вендингового автомата. В случае, если вендинговый аппарат получить удастся, то тестирование будет производиться непосредственно на нем. Второй вариант наиболее благоприятен, так как показывает результат test case в реальных условиях.
- Организационные границы: техник обслуживания(забирает/раскладывает пледы по аппаратам, техническое обслуживание), посетители парков.

3.1 Функциональность решения

Рассмотрим функциональность решения:

- Функция краткосрочной аренды пледа из автомата.
- Возможность не возврата пледа в автомат.
- Возможность вернуть плед в любой автомат (независимо от того, из какого автомата был взят плед).
- Возможность просмотра автоматов на карте с указанием количества пледов в них.
- Функция изъятия использованных пледов из автомата.
- Функция выкладывания чистых пледов в автомат.

- Функция сбора информации о количестве взятий пледов в аренду из определенного автомата за определенный период.
- Функция сбора информации о количестве возвратов пледов в автоматы за определенный период.
- Функция просмотра графиков о количестве взятий пледов в аренду из определенного автомата за определенный период.
- Функция просмотра графиков о количестве возвратов пледов в автоматы за определенный период.
- Возможность создания графиков со статистическими данными.
- Возможность экспорта статистических данных в формате CSV.

3.2 За рамками решения

За рамками разрабатываемого решения остаётся:

- организация обработки пледов;
- организация сбора использованных и выкладывания чистых пледов;
- создание бухгалтерских отчетов и обработка данных;
- организация маркетинговых акций;
- общая организация бизнес-процессов компании ООО “ГЭТ Э БЛАНКЕТ”;
- управление персоналом.

3.3 Критерии одобрения решения

ООО “ГЭТ Э БЛАНКЕТ”:

- Техническая работоспособность: Программный продукт выполняет все функции описанные в разделе 2.4 и подключен к вендинговому аппарату.
- User experience: Фокус-группа показывает не менее 70% удовлетворенности интерфейсом продукта (среднее значение по сумме вопросов анкетирования фокус-группы).

- Проведено приемочное (эксплуатационное) тестирование, результаты которого сопоставляются с бизнес-логикой проекта: покрытие тестов составляет 100% функциональных и нефункциональных требований, успешно завершается 100% всех тестов (примерное описание).
- Отчетность: В срок до 25 марта 2024 года представлены: концепция проекта, функциональная спецификация, спецификации и сценарии тестов, тестирование и отчеты об ошибках, отчет о пилотном внедрении, отчет о завершении проекта.

Фонд содействия инновациям:

- Финансовая эффективность: исполнитель эффективно использовал бюджет, продемонстрировал экономическую целесообразность проекта.
- До 04.12.2023 предоставить техническое задание программного обеспечения системы шеринга пледов.
- До 04.10.2024 обеспечить коммерческой модели стартап-проекта, разработать систему шеринга пледов, в том числе программного обеспечения для внутреннего функционирования системы, разработать прототип мобильного сайта для клиентов, позволяющего взять плед в аренду, используя QR-код расположенный на автомате.

4 Стратегии дизайна решения

4.1 Стратегия архитектурного дизайна

Для реализации выбран подтип микросервисной архитектуры: микроядерная архитектура, так как это обеспечивает защиту модулей от взлома, так как они не доступны не авторизованным пользователям сети интернет. Диаграмма архитектуры представлена на рисунке 4.1.

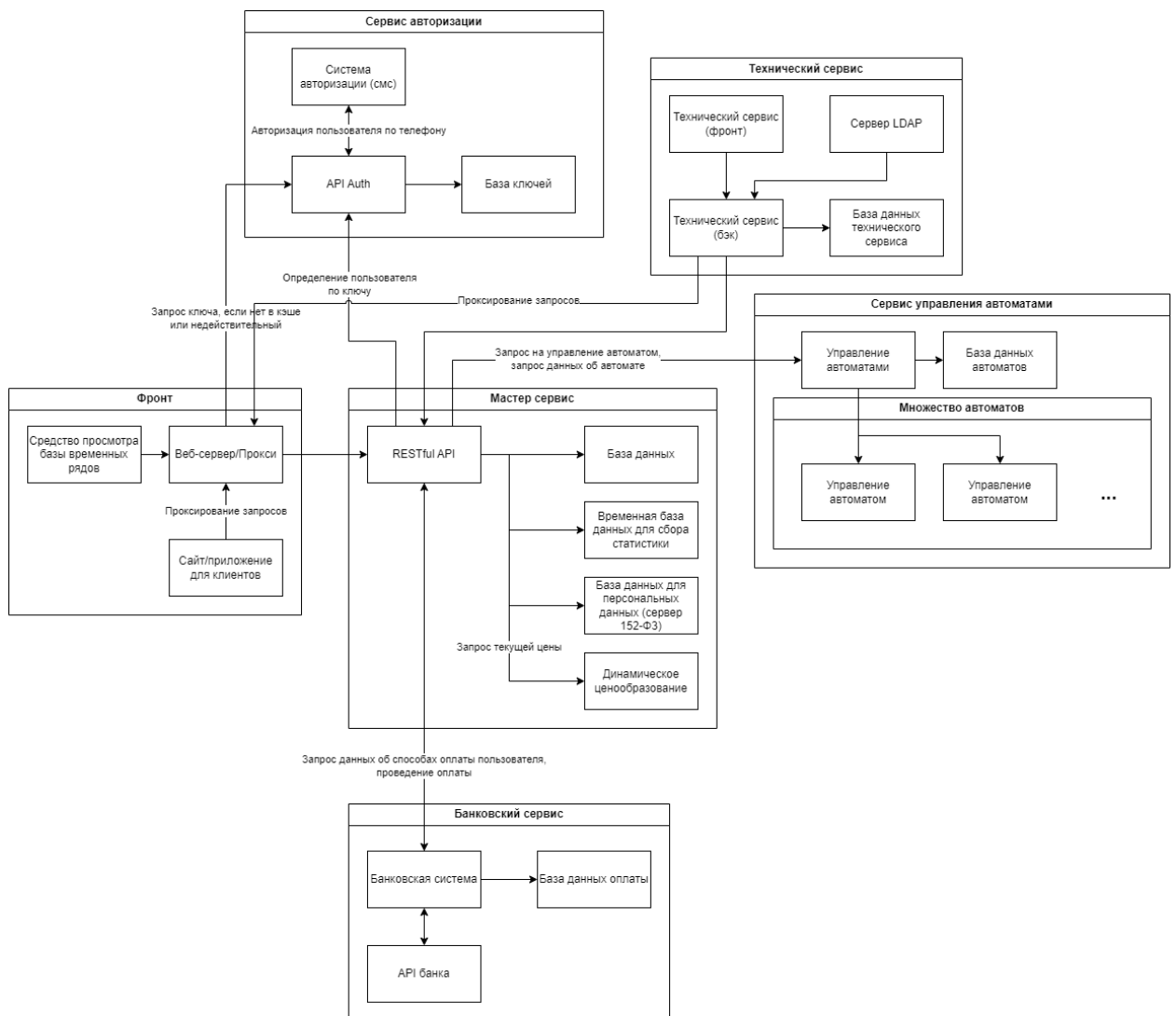


Рисунок 4.1 – Диаграмма архитектуры программного решения

Доступ к системе осуществляется посредством веб-сервера, выполняющего также роль прокси. Таким образом, исключается возможность несанкционированного доступа к серверам.

4.2 Стратегия технологического дизайна

В ходе разработки архитектурного дизайна, рассматривалась возможность реализации данной архитектуры существующими программными средствами.

Для разработки модуля “Фронт” было принято решение использовать стек технологий, представленный ниже:

- Для реализации “Сайт/приложение для клиентов” выбрано программное решение на основе библиотеки Flutter на языке Dart. Это решение обосновано тем, что данный фреймворк позволяет компилировать приложение под платформы: Web, Android, iOS. Это значительно сокращает потенциальные трудозатраты на разработку отдельных приложений под каждую платформу.
- Для реализации “Веб-сервер/Прокси” выбрано программное решение nginx, так как данное решение имеет возможность полного проксирования запросов на другую машину и разработчики уже имеют опыт его использования.
- Реализация “Средства просмотра базы временных рядов” связана с выбором стека технологий ядра системы (“Мастер сервис”). Для реализации “Временной базы данных для сбора статистики” была выбрана база данных InfluxDB, так как данная база данных имеет гибкую настройку автоматического удаления устаревших данных, а также к данной СУБД возможно подключение программного решения “Grafana”, позволяющего выполнять все поставленные задачи, описанные в разделе 2.4.

Для разработки ядра системы (модуль “Мастер сервис”) было принято решение использовать стек технологий представленный ниже:

- Основная часть модуля (“Restful API”) будет разработана с использованием библиотек ASP.NET Core, Entity Framework Core для языка C#, так как с этими технологиями знакомы все разработчики, при этом оно обеспечивает достаточную производительность. Использование данного стека позволяет автоматически генерировать документацию в соответствии с OpenAPI.

- Для реализации базы данных будет использовано СУБД PostgreSQL, так как данная СУБД распространяется бесплатно по свободной лицензии, а также у разработчиком есть опыт работы с ней.
- Для реализации “Временной базы данных для сбора статистики” была выбрана база данных InfluxDB, обоснование представлено выше.
- Реализация отдельной базы данных для персональных данных обусловлена законодательством РФ, которое обязывает хранить персональные данные на серверах, имеющих дополнительные степени защиты, из-за чего их использование обходится дороже, поэтому было принято решение вынести базу персональных данных на отдельный сервер. Реализация – СУБД PostgreSQL.
- Реализация “Динамического ценообразования” будет представлена в виде библиотеки (пространства имен), написанной на языке C#, так как это обеспечивает наиболее быстрый запрос текущей цены.

Для разработки модуля “Сервис авторизации” было принято решение использовать стек технологий, представленный ниже:

- Основная часть модуля (“API Auth”) будет представлена в виде Restful API и разработана с использованием библиотек ASP.NET Core, Entity Framework Core для языка C#, так как с этими технологиями знакомы все разработчики, при этом оно обеспечивает достаточную производительность. Использование данного стека позволяет автоматически генерировать документацию в соответствии с OpenAPI.
- Для реализации “Базы ключей” будет использовано СУБД PostgreSQL, так как данная СУБД распространяется бесплатно по свободной лицензии, а также у разработчиков есть опыт работы с ней.
- Реализация “Сервиса авторизации (смс)” представляет собой внешний сервис, общение с которым производится посредством Restful API.

Для разработки модуля “Технический сервис” было принято решение использовать стек технологий, представленный ниже:

- Для реализации клиентской части будет использован язык TypeScript благодаря его статической типизации, что позволит избежать многих ошибок компиляции и позволит более четко определять структуры данных.
- В качестве целевого фреймворка будет использован React актуальной версии. Популярность данного фреймворка обеспечивает релевантность получаемого опыта, а также широкую поддержку сообщества так, что даже самые узконаправленные проблемы или сложности уже испытывали более опытные коллеги и можно будет быстро найти решение.
- В качестве сборщика пакетов используется webpack, так как разработчики имеют наибольший опыт работы именно с ним, а также:
 - Webpack может обрабатывать любые файлы интерфейса, такие как файлы .html , .css, .js, .scss , изображения и другие ресурсы.
 - Объединяет все зависимости, а не только модули Node.js и загружает их как статические ресурсы в браузер пользователя.
 - Конфигурируемость – webpack имеет несколько вариантов конфигурации.
 - Разделение кода на более мелкие части, которые могут загружаться асинхронно.
 - Огромная экосистема с богатым интерфейсом плагинов.
- Бэкенд для системы будет написан на TypeScript для соблюдения единообразия частей приложения.
- В качестве целевого фреймворка будет использован Nest.js, так как это один из популярнейших фреймворков, то он имеет крупное сообщество и актуальную всестороннюю документацию. Также Nest.js имеет ряд преимуществ:
 - Легко расширяется – может использоваться с другими библиотеками.
 - Предоставляет API фреймворков, которые помогают разработчику использовать различные сторонние модули, доступные для любой платформы.

- Позволяет разработчикам использовать чистый TS.
- Для реализации “База данных технического сервиса” будет использовано СУБД PostgreSQL, так как данная СУБД распространяется бесплатно по свободной лицензии, а также у разработчиком есть опыт работы с ней.

Для разработки модуля “Сервис управления автоматами” было принято решение использовать стек технологий, представленный ниже:

- “Управление автоматами” представляет собой Restful API, взаимодействующий с ядром системы и ПО непосредственно установленном на вендинговом аппарате. Разработка модуля будет с использованием библиотек ASP.NET Core, Entity Framework Core для языка C#, так как с этими технологиями знакомы все разработчики, при этом оно обеспечивает достаточную производительность. Использование данного стека позволяет автоматически генерировать документацию в соответствии с OpenAPI.
- Для реализации “Базы данных автоматов” будет использовано СУБД PostgreSQL, так как данная СУБД распространяется бесплатно по свободной лицензии, а также у разработчиков есть опыт работы с ней.
- “Управление автоматом” представляет собой Restful API, взаимодействующий с модулем “Управление автоматами”, открывающий и закрывающий автомат. Реализован будет с использованием библиотеки FastAPI для языка Python, так как данный язык отличается наибольшей мультиплатформенностью из фреймворков известных разработчикам.

Для разработки модуля “Банковский сервис” было принято решение использовать стек технологий, представленный ниже:

- Основная часть модуля (“Банковская система”) будет представлена в виде Restful API, взаимодействующим с ядром системы и “API банка”, и разработана с использованием библиотек ASP.NET Core, Entity Framework Core для языка C#, так как с этими технологиями знакомы все разработчики, при этом оно обеспечивает достаточную производительность. Использование данного

стека позволяет автоматически генерировать документацию в соответствии с OpenAPI.

- Для реализации “Базы данных оплаты” будет использовано СУБД PostgreSQL, так как данная СУБД распространяется бесплатно по свободной лицензии, а также у разработчиков есть опыт работы с ней.
- Реализация “API банка” представляет собой внешний сервис, общение с которым производится посредством Restful API.