1 Анализ структуры предприятия

1.1 Технико-экономическая характеристика предприятия.

Общество с ограниченной ответственностью «Ависта» занимается программным и бухгалтерским сопровождением бизнеса.

Структуру данной организации можно описать так, как показано на Рисунке 1. Ссылаясь на эту схему можно проследить следующую иерархию:

* учредитель;
* директор;
* начальники отдела;
* штатные работники.

Учредитель – лицо, учредившее предприятие и являющееся его непосредственным владельцем. Любые изменения в системе качества производятся только с его устного(если изменения не являются критичными) или письменного(если изменения носят достаточно глобальный характер) согласия. Данное лицо имеет полный контроль над организацией, и может давать указания всем элементам системы.

Директор координирует и систематизирует работы подконтрольных ему групп (в основном, общаясь с начальниками отделов, но в некоторых ситуациях непосредственно с рядовыми сотрудниками).

Начальники отделов осуществляют непосредственное руководство сотрудниками, а также, при необходимости, берут часть их нагрузки на себя.

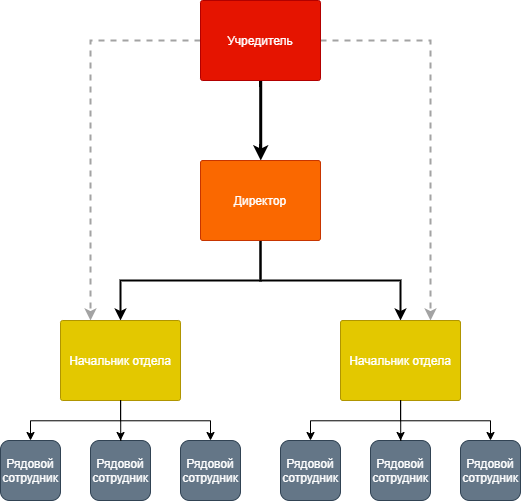


Рисунок 1 — Структура предприятия

Далее ознакомимся с кратким личным делом предприятия.

Название предприятия: Общество с Ограниченной Ответственностью «Ависта».

Зарегистрировано 13 апреля 2010 года. Регистрирующий орган: ИФНС России по Верх-Исетскому району города Екатеринбурга.

Вид деятельности: Разработка компьютерного программного обеспечения.

Дополнительные виды деятельности:

* Копирование записанных носителей информации.
* Деятельность агентов по оптовой торговле вычислительной техникой, телекоммуникационным оборудованием и прочим офисным оборудованием
* Торговля розничная компьютерами, периферийными устройствами к ним и программным обеспечением в специализированных магазинах
* Торговля розничная музыкальными записями.

Имеет два офиса по адресам:

* Улица Льва Толстого 2-а, город Новоуральск, Свердловская область, Россия, 624000
* Улица Фрунзе 7, город Новоуральск, Свердловская область, Россия, 624132

1.2 Диаграмма потоков данных. Описание информационных потоков

Общую и очень упрощённую схему потоков данных на предприятии можно наблюдать на рисунке 2. Здесь мы можем видеть, что клиент в первую очередь обращается к ответственному за работу с клиентами (но это не единственная его функция) сотруднику предприятия(это может быть как исполнитель, так и начальник отдела, так и директор), оформляет заявку, которая обрабатывается так. как показано на рисунке 3.



Рисунок 2 — Обобщённая диаграмма потоков данных.

При получении заявки сотрудник разделяет данные на данные непосредственно заявки(заказа) и данные заказчика, после чего формализует заявку в рамках стандартов предприятия и получает заказ (рисунок 3).

Разделённые данные сохраняются в соответствующие хранилища информации.

Далее начинается непосредственный процесс разработки.

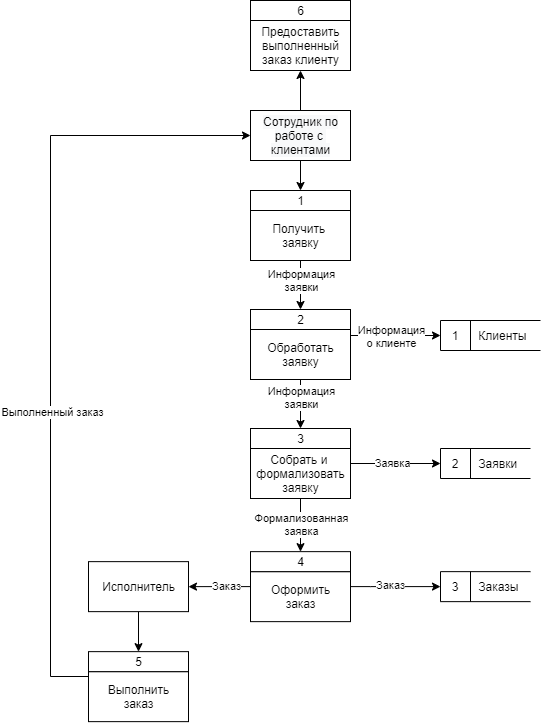


Рисунок 3 — Детализированная диаграмма потоков данных.

2 План разработки программного продукта

В ходе проведения анализа предстоящей разработки была выявлена необходимость создания плана дальнейшей разработки , который включает в себя:

1. Сбор информации о предприятии.
2. Формализация данных, полученных при опросе сотрудников предприятия и анализе его информационных потоков.
3. Анализ и осознание необходимой архитектуры для построения приложения.
4. Проведение теоретического и практического (в случае наиболее перспективных средств) исследования технических средств, которые могут быть полезны при разработке.
5. Разработка технического задания на основе полученных данных.
6. Систематизация и анализ полученных практических и теоретических знаний.
7. Непосредственная реализация программного продукта на основе выработанных данных и навыков.
8. Подведение итогов производственной практики и освоения технических средств.

3 анализ технических средств

В ходе поиска и анализа были выявлены следующие наиболее благоприятные для выполнения дальнейшей разработки технические средства.

Для клиентской части были выявлены следующие модули:

* JavaScript

Мультипарадигменный язык программирования. Поддерживает объектно-ориентированный, императивный и функциональный стили. Является реализацией спецификации ECMAScript.

* ReactJS

JavaScript-библиотека с открытым исходным кодом для разработки пользовательских интерфейсов. React разрабатывается и поддерживается Facebook, Instagram и сообществом отдельных разработчиков и корпораций [2].

* NextJS

JavaScript фреймворк, разработанный на основе ReactJS, обладающий технологией серверного рендеринга, созданный компанией Vercel. Исходный код находится в открытом доступе.

* ReactSpring

Это библиотека анимации на основе физики пружины. Данная библиотека охватывает большинство пользовательских потребностей в анимации, связанных с анимированием графического интерфейса.

* Axios

Основанный на JS Promise HTTP-клиент для браузера и Node.js.

* TailwindCSS

Библиотека для быстрой разработки пользовательских интерфейсов.

Вместо заранее разработанных компонентов, Tailwind предлагает использовать низкоуровневые служебные классы, которые позволяют создавать полностью индивидуальные проекты, используя только HTML.

* PostCSS

Данный продукт представляет собой программу на языке JavaScript, для автоматизации рутинных действий с CSS. Используется в TailwindCSS.

Для серверной части было решено использовать следующие разработки

* Golang

Компилируемый статически типизированный язык, разработанный компанией Google в 2007 году, имеющий простой синтаксис и высокую производительность.

* Gorilla

Комплект библиотек для веб разработки, который включает в себя библиотеки для роутинга запросов, работы по протоколу RPC, работы с GET-параметрами, организации web-сокет соединиения, а также реализации механизма аутентификации с cookie и т.д.

* PG

ORM для языка Golang, специализирующаяся на базе данных PostgreSQL, за счёт чего обеспечивается наибольшая возможная производительность и наиболее комфортное использование.

* PostgreSQL

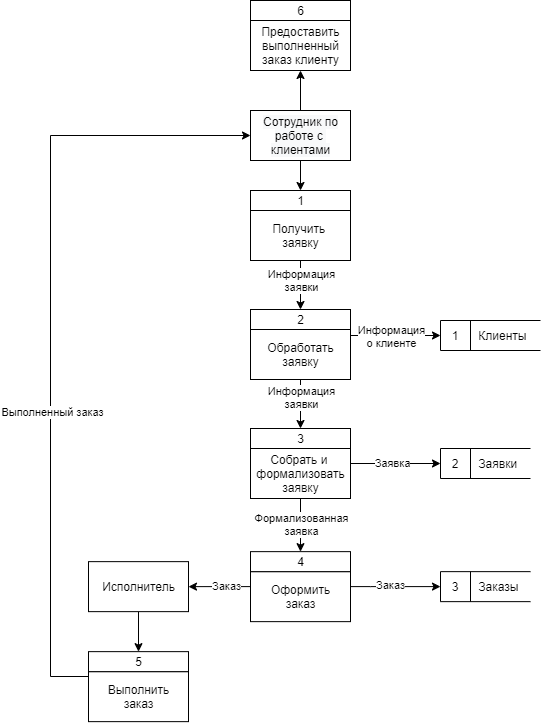
Объектно-реляционная система управления базами данных, которая зарекомендовала себя как стандарт в области разработки СБ и использования СУБД.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. ГОСТ 19.101-77 Единая система программной документации (ЕСПД). Виды программ и программных документов [Электронный ресурс] — Режим доступа: <https://docs.cntd.ru/document/1200007627>
2. ГОСТ 19.201-78 Единая система программной документации (ЕСПД). Техническое задание. Требования к содержанию и оформлению [Электронный ресурс] — Режим доступа: <https://docs.cntd.ru/document/1200007648>
3. React [Электронный ресурс] — Режим доступа: <https://ru.wikipedia.org/wiki/React>
4. Next.JS [Электронный ресурс] — Режим доступа: <https://ru.wikipedia.org/wiki/Next.js>
5. axios [Электронный ресурс] — Режим доступа: <https://github.com/axios/axios>
6. TailwindCSS [Электронный ресурс] — Режим доступа: <https://tailwindcss.com/>
7. Gorilla Web Toolkit [Электронный ресурс] — Режим доступа: <https://www.gorillatoolkit.org/>
8. go-pg/pg [Электронный ресурс] — Режим доступа: <https://github.com/go-pg/pg>
9. PostgreSQL https://ru.wikipedia.org/wiki/PostgreSQL
10. Что такое DFD (диаграммы потоков данных) [Электронный ресурс] — Режим доступа: <https://habr.com/ru/company/trinion/blog/340064/>
11. Зачем нам UML? Или как сохранить себе нервы и время [Электронный ресурс] — Режим доступа: https://habr.com/ru/post/458680/
12. Пять простых шагов для понимания JSON Web Tokens (JWT) [Электронный ресурс] — Режим доступа: <https://habr.com/ru/post/340146/>
13. JWT простым языком: что такое JSON токены и зачем они нужны [Электронный ресурс] — Режим доступа: <https://proglib.io/p/json-tokens/>
14. Локальное хранилище или куки? Безопасное хранение JWT на клиенте [Электронный ресурс] — Режим доступа: <https://habr.com/ru/company/ruvds/blog/512866/>
15. LocalStorage vs Cookies: все, что нужно знать о безопасном хранении токенов JWT во Front-End [Электронный ресурс] — Режим доступа: <https://webdevblog.ru/localstorage-vs-cookies-vse-chto-nuzhno-znat-o-bezopasnom-hranenii-tokenov-jwt-vo-front-end/>
16. В чем различия между HTTP и HTTPS [Электронный ресурс] — Режим доступа: <https://sweb.ru/journal/article/v-chem-razlichiya-mezhdu-http-i-https/>
17. Динамический рендеринг на Next.js [Электронный ресурс] — Режим доступа: <https://habr.com/ru/post/468153/>
18. Next JS Docs Getting Started [Электронный ресурс] — Режим доступа: <https://nextjs.org/docs/getting-started>
19. React Spring Introducing v9.0 [Электронный ресурс] — Режим доступа: <https://aleclarson.github.io/react-spring/v9/#Platform-packages>
20. react-spring: bring your components to life with simple spring animation primitives [Электронный ресурс] — Режим доступа: <https://react-spring.io/docs>
21. Создание анимированных приложений React с помощью React Spring [Электронный ресурс] — Режим доступа: <https://internet--technologies-ru.turbopages.org/internet-technologies.ru/s/articles/sozdanie-animirovannyh-prilozheniy-s-react-spring.html>
22. SASS - CSS с суперсилой [Электронный ресурс] — Режим доступа: <https://sass-scss.ru/>
23. PostCSS, полное погружение: Все, что Вам нужно знать [Электронный ресурс] — Режим доступа: <https://webdesign.tutsplus.com/ru/tutorials/postcss-deep-dive-what-you-need-to-know--cms-24535>
24. Лучшие WYSIWYG-редакторы - [Электронный ресурс] — Режим доступа: <https://timeweb.com/ru/community/articles/obzor-luchshih-wysiwyg-redaktorov>
25. Небезопасный cross-origin resource sharing / Блог компании OWASP / Хабр [Электронный ресурс] — Режим доступа: <https://habr.com/ru/company/owasp/blog/337146/>
26. Resources for developers, by developers. [Электронный ресурс] — Режим доступа: <https://developer.mozilla.org/ru/>
27. WebDAV API Диска — Технологии Яндекса [Электронный ресурс] — Режим доступа: <https://yandex.ru/dev/disk/webdav/>
28. WebDAV: что это такое, где с ним можно столкнуться, и существующие альтернативы [Электронный ресурс] — Режим доступа: <https://codeby.net/threads/webdav-chto-ehto-takoe-gde-s-nim-mozhno-stolknutsja-i-suschestvujuschie-alternativy.67246/>
29. Изучаем WebDav — «Хакер» [Электронный ресурс] — Режим доступа: <https://xakep.ru/2014/09/09/webdav/#toc01>.

ПРИЛОЖЕНИЕ B





ПРИЛОЖЕНИЕ C

1. ГОСТ 19.101-77 Единая система программной документации (ЕСПД). Виды программ и программных документов [Электронный ресурс] — Режим доступа: <https://docs.cntd.ru/document/1200007627>
2. ГОСТ 19.201-78 Единая система программной документации (ЕСПД). Техническое задание. Требования к содержанию и оформлению [Электронный ресурс] — Режим доступа: <https://docs.cntd.ru/document/1200007648>
3. ГОСТ 19.301-79 Единая система программной документации (ЕСПД). Программа и методика испытаний [Электронный ресурс] — Режим доступа: <https://docs.cntd.ru/document/1200007650>
4. Приложение. Федеральный государственный образовательный стандарт среднего профессионального образования по специальности 09.02.03 Программирование в компьютерных системах [Электронный ресурс] — Режим доступа: <https://base.garant.ru/70731880/53f89421bbdaf741eb2d1ecc4ddb4c33/>
5. Что такое DFD (диаграммы потоков данных) [Электронный ресурс] — Режим доступа: <https://habr.com/ru/company/trinion/blog/340064/>
6. Зачем нам UML? Или как сохранить себе нервы и время [Электронный ресурс] — Режим доступа: https://habr.com/ru/post/458680/
7. Техника безопасности при работе с компьютером [Электронный ресурс] — Режим доступа: https://compsch.com/obzor/texnika-bezopasnosti-pri-rabote-s-kompyuterom.html
8. Различия REST и SOAP [Электронный ресурс] — Режим доступа: <https://habr.com/ru/post/483204/>
9. Что же такое этот GraphQL? [Электронный ресурс] — Режим доступа: <https://habr.com/ru/post/326986/>
10. Что такое GraphQL: с основ до первых запросов [Электронный ресурс] — Режим доступа: <https://ru-hexlet-io.turbopages.org/ru.hexlet.io/s/blog/posts/chto-takoe-graphql-s-osnov-do-pervyh-zaprosov>
11. gRPC и все-все-все: Часть I. Введение [Электронный ресурс] — Режим доступа: <https://proglib.io/p/grpc-i-vse-vse-vse-chast-i-vvedenie-2021-03-26>
12. Сравнение архитектурных стилей API: SOAP vs REST vs GraphQL vs RPC [Электронный ресурс] — Режим доступа: <https://zen.yandex.ru/media/nuancesprog/sravnenie-arhitekturnyh-stilei-api-soap-vs-rest-vs-graphql-vs-rpc-60197fcbb19d82187ee3d704>
13. Пять простых шагов для понимания JSON Web Tokens (JWT) [Электронный ресурс] — Режим доступа: <https://habr.com/ru/post/340146/>
14. JWT простым языком: что такое JSON токены и зачем они нужны [Электронный ресурс] — Режим доступа: <https://proglib.io/p/json-tokens/>
15. Локальное хранилище или куки? Безопасное хранение JWT на клиенте [Электронный ресурс] — Режим доступа: <https://habr.com/ru/company/ruvds/blog/512866/>
16. LocalStorage vs Cookies: все, что нужно знать о безопасном хранении токенов JWT во Front-End [Электронный ресурс] — Режим доступа: <https://webdevblog.ru/localstorage-vs-cookies-vse-chto-nuzhno-znat-o-bezopasnom-hranenii-tokenov-jwt-vo-front-end/>
17. В чем различия между HTTP и HTTPS [Электронный ресурс] — Режим доступа: <https://sweb.ru/journal/article/v-chem-razlichiya-mezhdu-http-i-https/>
18. Динамический рендеринг на Next.js [Электронный ресурс] — Режим доступа: <https://habr.com/ru/post/468153/>
19. Next JS Docs Getting Started [Электронный ресурс] — Режим доступа: <https://nextjs.org/docs/getting-started>
20. React Spring Introducing v9.0 [Электронный ресурс] — Режим доступа: <https://aleclarson.github.io/react-spring/v9/#Platform-packages>
21. react-spring: bring your components to life with simple spring animation primitives [Электронный ресурс] — Режим доступа: <https://react-spring.io/docs>
22. Создание анимированных приложений React с помощью React Spring [Электронный ресурс] — Режим доступа: <https://internet--technologies-ru.turbopages.org/internet-technologies.ru/s/articles/sozdanie-animirovannyh-prilozheniy-s-react-spring.html>
23. SASS - CSS с суперсилой [Электронный ресурс] — Режим доступа: <https://sass-scss.ru/>
24. PostCSS, полное погружение: Все, что Вам нужно знать [Электронный ресурс] — Режим доступа: <https://webdesign.tutsplus.com/ru/tutorials/postcss-deep-dive-what-you-need-to-know--cms-24535>
25. Лучшие WYSIWYG-редакторы - База Знаний Timeweb Community [Электронный ресурс] — Режим доступа: <https://timeweb.com/ru/community/articles/obzor-luchshih-wysiwyg-redaktorov>
26. Небезопасный cross-origin resource sharing / Блог компании OWASP / Хабр [Электронный ресурс] — Режим доступа: <https://habr.com/ru/company/owasp/blog/337146/>
27. Resources for developers, by developers. [Электронный ресурс] — Режим доступа: <https://developer.mozilla.org/ru/>
28. WebDAV API Диска — Технологии Яндекса [Электронный ресурс] — Режим доступа: <https://yandex.ru/dev/disk/webdav/>
29. WebDAV: что это такое, где с ним можно столкнуться, и существующие альтернативы [Электронный ресурс] — Режим доступа: <https://codeby.net/threads/webdav-chto-ehto-takoe-gde-s-nim-mozhno-stolknutsja-i-suschestvujuschie-alternativy.67246/>
30. Изучаем WebDav — «Хакер» [Электронный ресурс] — Режим доступа: <https://xakep.ru/2014/09/09/webdav/#toc01>.
31. Чем хорош язык Go и зачем его изучать? Все плюшки Golang [Электронный ресурс] — Режим доступа: <https://proglib.io/p/language-go/>
32. Зачем программисту учить Go [Электронный ресурс] — Режим доступа: <https://skillbox.ru/media/code/yazyk_go_chto_pod_kapotom_i_zachem_programmistu_uchit_ego_kak_vtoroy/>
33. Библиотека React: особенности, перспективы, ситуация на рынке труда [Электронный ресурс] — Режим доступа: <https://ru-hexlet-io.turbopages.org/ru.hexlet.io/s/blog/posts/biblioteka-react-review-article>
34. Что такое React и как его освоить? [Электронный ресурс] — Режим доступа: <https://academy.yandex.ru/posts/chto-takoe-react-i-kak-ego-osvoit>
35. Что такое файлы cookies? [Электронный ресурс] — Режим доступа: https://trends.rbc.ru/trends/industry/5f4e8d719a794788d1c8b49f