Министерство образования и науки Республики Башкортостан Государственное автономное профессиональное образовательное учреждение

Уфимский колледж статистики, информатики и вычислительной техники

ОТЧЁТ

по производственной практике

ПП.04.01 Производственная практика

по модулю ПМ.04 Сопровождение и обслуживание программного обеспечение компьютерных систем

Специальность

09.02.07 Информационные системы и программирование Квалификация

Программист

Заведующий Центром	Руководитель практики			
профессиональной карьеры	от предприятия			
/ А.Г. Анастасьев	/ И.В. Ванов			
подпись ФИО	подпись ФИО			
	М.П.			
_				
Руководитель практики	Студент группы 21П-2			
от учебного заведения	/ Д.Р.Исламов			
/ О.В.Фатхулова	подпись ФИО			
подпись ФИО	«» 2024 год			

СОДЕРЖАНИЕ

	лист
Введение	3
1 Характеристика организационной и функциональной структуры	<mark>4</mark>
системы управления предприятия с перечнем задач	
2 Сопровождение и обслуживание программного обеспечения	<mark>6</mark>
предприятия	
2.1 Анализ аппаратного и программного обеспечения	<mark>6</mark>
2.2 Анализ сетевого обеспечения предприятия	<mark>7</mark>
2.3 Анализ различных антивирусных программ	<mark>7</mark>
2.4 Настройка защиты системы стандартными средствами	8
операционной системы	
3 Проектирование программного обеспечения для решения	14
прикладной задачи	
3.1 Постановка задачи. Техническое задание на разработку	14
программного продукта	
3.2 Описание программы	<mark>20</mark>
3.3 Протокол тестирования разработанного программного продукта	<mark>16</mark>
3.4 Руководство пользователя	27
Заключение	<mark>30</mark>
Список использованных источников	<mark>31</mark>

ВВЕДЕНИЕ

Практика составной основной студентов является частью образовательной программы высшего профессионального образования, а обучения теоретического также средством связи практической обеспечивающим деятельностью, прикладную направленность специализацию обучения.

Целью прохождения практики является углубление знаний, полученных в процессе обучения и приобретение практических навыков.

Место прохождение практики – Общество с ограниченной ответственностью «Класс информационные Технологии».

Цель практики — получение навыков сопровождения и обслуживания программного обеспечения компьютерных систем.

Задачи производственной практики:

□ ознакомлени	іе с	предп	риятием,	видами	деятельност	гь,
организационной ст	груктурой	;				
□ проведение	функцион	нального	анализа	деятельности	предприятия	И
предметной області	и;					

□ озна	комление с	ИМ	еющимися	В	отделе	автома	гизиро	ванными
системами,	аппаратным	И	программн	ΙЫΜ	обеспе	чением	ПК,	сетевым
обеспечение	ем, антивируст	ным:	и программа	ами;				

□ освоение методик	разработки	технической	документации	на создани	e
программного продукта;					

□ освоение	методов	разработки	программного	продукта	согласно
задачам, выполня	немым в от	деле;			

□ создание	тестовых	элементов	для	проверки	работоспособности
создаваемого про	ограммного	продукта;			

					<i>#####################################</i>	
∌ ←	% →	<i>⊕ זע\⇔7</i>	とリオク	₹→ <i>∅</i>	- ·	

1 Характеристика организационной и функциональной структуры системы управления предприятия с перечнем задач

Общество с ограниченной ответственность «Класс Информационные Технологии» основано в 2012 году в г. Уфа. Компания занимает одно из лидирующих мест по разработке программного обеспечения для бизнеса и разработки SaaS сервисов. Является ведущим разработчиком облачных сервисов для управления торговлей и складом. Более 12 лет занимается разработкой онлайн-сервисов для автоматизации бизнес-процессов, основала один из крупнейших порталов для предпринимателей с одноименным названием.

Разработанные компанией облачные сервисы закрывают 95% всех потребностей современного бизнеса и полностью соответствуют законам РФ (54-ФЗ, ЕГАИС, Маркировка товаров)

На сегодняшний день одним из ключевых приоритетов компании является развитие продуктов для бизнеса.

Действующая в компании система менеджмента качества позволяет выходить со своими решениями на международный рынок. Для эффективной продажи программного обеспечения в компании запущен официальный сайт по продаже необходимых услуг разработки программного обеспечения и готовых решений.

Организационная структура организации представлена на рисунке 1.1.

						%→
					######################################	1
∌ ←	%→	⊕ ⊿₹₹₩	& Y 7	♦→ <i>∅</i>		4

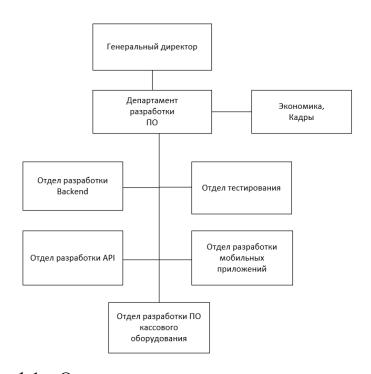


Рисунок 1.1 – Организационная структура организации Схема расположения рабочих мест в отделе разработки ПО представлена на рисунке 1.2.

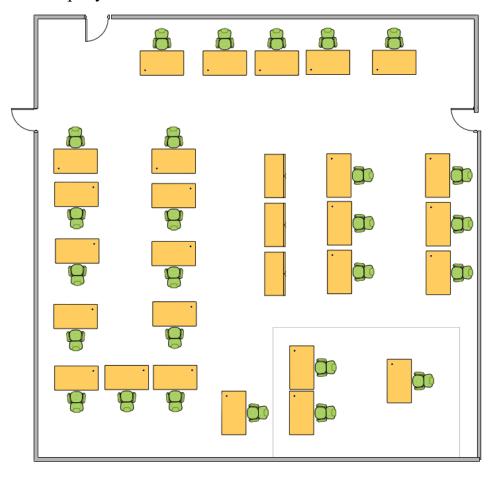


Рисунок 1.2 – Схема расположения рабочих мест в отделе

						% →
·					<i></i>	_
∌ ←	%→	⊕ ⊿⊻♦≉≉	& Y 7&	$\Leftrightarrow \rightarrow \emptyset$		5

2 Сопровождение и обслуживание программного обеспечения предприятия

2.1 Анализ аппаратного и программного обеспечения

В компаии для оптимизации процессов управления проектами и внутренней коммуникации используются две основные программы: Jira и Discord. Эти инструменты помогают решать широкий спектр задач, связанных с планированием, организацией командной работы и улучшением взаимодействия сотрудников.

Jira — это система управления проектами и задачами, разработанная компанией Atlassian, которая позволяет эффективно организовать рабочие процессы и контролировать выполнение задач в команде. Платформа особенно полезна для предприятий, работающих по методологиям Agile и Scrum, так как она предлагает гибкие инструменты для управления спринтами и канбан-досками. Jira позволяет:

□ планировать и отслеживать задачи и проекты — создание, назначение и контроль выполнения задач с возможностью группировки по проектам и отделам;

□ управлять рабочими процессами — автоматизация и настройка последовательности действий, что способствует более точному распределению задач и эффективному соблюдению сроков;

□ отслеживать и анализировать прогресс — мощные инструменты отчетности и аналитики позволяют руководству видеть текущий статус проекта, загруженность сотрудников и потенциальные риски;

□ обеспечивать интеграцию с другими инструментами — поддержка интеграций с различными программами, такими как Confluence, Slack, GitHub и многими другими, что позволяет настраивать совместную работу и обмен данными.

					<i> </i>	
∌ ←	% →	⊕ ЛУ ∳\$ Л	& ¥ 7<₽	♦→ ∅	- ·	

Discord — это универсальная платформа для общения и командного взаимодействия, которая помогает поддерживать эффективную коммуникацию среди сотрудников предприятия. Хотя изначально Discord создавался для игровой индустрии, его функции и возможности сделали его удобным инструментом для корпоративного использования, особенно для удаленных и гибридных команд. Discord предоставляет возможности для: □ организации текстового и голосового общения сотрудникам обмениваться сообщениями в текстовых каналах или проводить голосовые и видеоконференции; 🗆 создания специализированных каналов по темам и проектам каждая команда или проект могут иметь свои собственные каналы, что помогает поддерживать порядок и лучше организовать рабочий процесс; □ интеграции с другими инструментами — поддерживает интеграцию с Jira, Trello, Google Drive и другими популярными программами, что упрощает обмен данными и централизует коммуникацию; □ обеспечения оперативной связи — возможность получать мгновенные уведомления и обмениваться информацией в реальном времени, что ускоряет процесс принятия решений и повышения вовлеченности сотрудников; □ настройки доступа и ролей — можно назначать права доступа к различным каналам и контенту, что повышает безопасность и позволяет настроить доступ к информации для каждой команды. Использование Jira и Discord в тандеме позволяет эффективно управлять проектами, обеспечивать прозрачность выполнения задач и поддерживать быструю и надежную коммуникацию на предприятии. Это производительность, помогает повысить **УЛУЧШИТЬ** командное взаимодействие и обеспечить более гибкое управление процессами в развивающейся компании. Мой рабочий компьютер имеет следующую конфигурацию: □ процессор – Intel Core i5-10400 Six-Core Processor 2.9 GHz;

- □ материнская плата MSI PRO H610M-E;
- □ O3Y 32 Γ B, DDR 4;
- □ SSD NVME GAMMIX 512GB;
- □ операционная система Ubuntu 24.04 LTS.
- 2.2 Анализ сетевого обеспечения предприятия

Локальная сеть департамента разработки состоит из: коммутаторов, серверов, рабочих станций (ПК). Ядро сети – комплекс сетевых устройств (маршрутизаторов и коммутаторов), обеспечивающих высокоскоростную передачу данных и подключение к глобальной сети Интернет. Такая архитектура сети позволяет сокращать время простоя сети, минимизировать потери рабочего времени, а также создает возможность внедрения дополнительных приложений и сервисов. Основная цель - обеспечение высокой надежности и производительности. Сеть отдела департамента разработки изображена на рисунке 2.2.1.

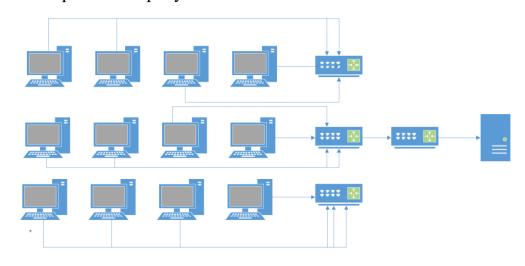


Рисунок 2.2.1 – Локальная сеть департамента разработки ПО

2.3 Анализ различных антивирусных программ

Для защиты данных и минимизации рисков кибератак в компании используется комплексное решение на основе систем Symantec Endpoint Protection и Malwarebytes. Эти программы предназначены для защиты корпоративных компьютеров и сетей от широкого спектра угроз, включая

						26-3
					<i>#####################################</i>	0
£	%→	⊕ 7¥ ₩ 7	& ¥ 7¢=	♦→ <i>∅</i>	- ·	0

вредоносное ПО, фишинговые атаки и сетевые угрозы, не снижая при этом производительность системы.

Symantec Endpoint Protection — это многоуровневая платформа кибербезопасности, которая обеспечивает защиту как локальных, так и удалённых рабочих мест. Она использует передовые технологии машинного обучения и анализа поведения для обнаружения угроз.

Malwarebytes — это дополнительный уровень защиты, который эффективно устраняет сложные угрозы, такие как шпионские программы, рекламное ПО и руткиты, которые могут быть пропущены основным антивирусом. Программа подходит для глубокого сканирования системы и удаления вредоносного ПО, не замеченного на предыдущих этапах.

2.4 Настройка защиты системы стандартными средствами операционной системы

Брандмауэр (фаервол) — это программное средство, которое обеспечивает линию обороны компьютера от людей или программ, которые несанкционированно пытаются подключиться к компьютеру или просто охранник вашего компьютера, как и антивирус.

Настройка брандмауэра на Windows.

Для того, чтобы оказаться в разделе с настройкой можно использовать сочетание клавиш Win+R после чего ввести команду «control» или же нажав клавишу Пуск ввести текст «Брандмауэр защитника Windows».

Активировать брандмауэр или отключить его можно, перейдя по соответствующей ссылке в разделе настроек. Представлено на рисунке 2.4.1.

						26-
					######################################	0
∌ ←	%→	<i>⊕ זעע⇔ז</i>	& ¥ 7¢	♦→ ∅		9

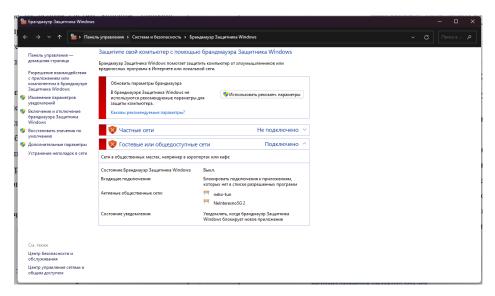


Рисунок 2.4.1 – Раздел «Включение и отключение брандмауэра Защитника Windows»

Здесь достаточно поставить переключатель в нужное положение и нажать ОК. Блокировка подразумевает запрет всех входящих подключений, то есть любые приложения, в том числе и браузер, не смогут загружать данные из сети. Уведомления представляют собой особые окна, возникающие при попытках подозрительных программ выйти в интернет или локальную сеть, её можно отключить и подключить в любое время. Представлено на рисунке 2.4.2.

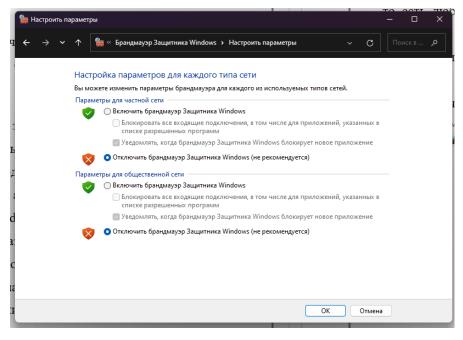


Рисунок 2.4.2 – Настройка параметров для каждого типа сети

						%→
						7
₽€	% →	⊕ ⊿⊻♦≉≉	& ¥ 7¢	♦→ <i>∅</i>	<u>.</u> .	10

Также можно настроить дополнительные параметры Брандмауэра, создав дополнительные правила для абсолютно всех файлов и приложений, которые присутствуют на компьютере. Представлено на рисунке 2.4.5

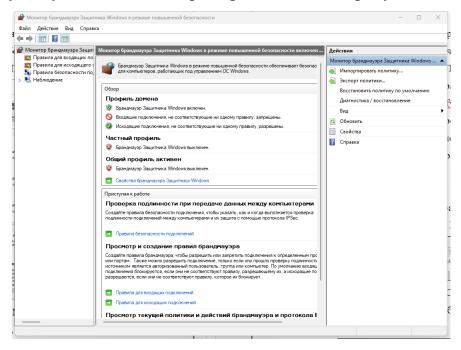


Рисунок 2.4.5 – Раздел «Мониторинг брандмауэра»

Для того, чтобы создать правило, нужно в пункте Действие, которое расположено в левом верхнем углу окна, выбрать «Создать правило», после чего откроется новое окно с именем мастер создания правила для нового входящего сетевого подключения в брандмауэре, где пользователь выбирает конкретный тип правила. Представлено на рисунке 2.4.6.

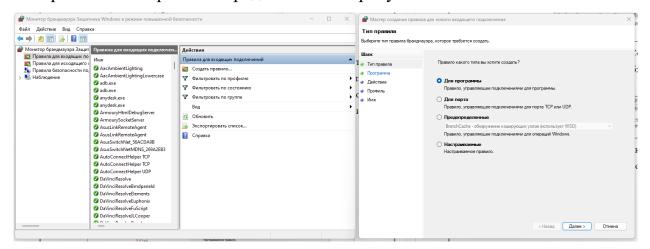


Рисунок 2.4.6 – Раздел «Мастер создания правила для входящего подключения»

							% →
						<i></i>	11
I	∌ ←	%→	⊕ ⊿⊻♦≉≉	& Y 7 (=	♦→ <i>∅</i>		11

Настройка брандмауэра на Linux UBUNTU

Для управления возможностями брандмауэра используется одноимённая команда - ufw. Синтаксис команды: \$ ufw опции действие параметры.

Опции определяют общие настройки поведения утилиты, действие указывает, что нужно сделать, а параметры - дополнительные сведения для действия, например, IP-адрес или номер порта.

Включение UFW

Сначала нужно отметить, что в серверной версии Ubuntu UFW по умолчанию включён, а в версии для рабочего стола он отключён. Поэтому нужно посмотреть состояние фаервола. Представлено на рисунке 2.4.5.

```
Файл Правка Вид Поиск Терминал Справка danil.bru@business.r:~$ sudo ufw status Status: inactive
```

Рисунок 2.4.5 – Состояние фаервола

Если он не включён, то его необходимо включить. Представлено на рисунке 2.4.6.

```
Файл Правка Вид Поиск Терминал Справка
danil.bru@business.r:-$ sudo ufw enable
Command may disrupt existing ssh connections. Proceed with operation (y|n)?
Firewall is active and enabled on system startup
```

Рисунок 2.4.6 – Включение фаервола

Затем снова можно посмотреть состояние. Представлено на рисунке 2.4.7.

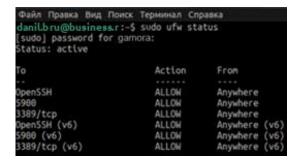


Рисунок 2.4.7 – Состояние включенного фаервола

Перед переходом к добавлению правил, необходимо указать политику по умолчанию. Какие действия будут применяться к пакетам, если они не

						% →
					<i> </i>	12
∌ ←	%→	<i>⊕ זעע⇔ז</i>	& ¥7¢	$\Leftrightarrow \rightarrow \emptyset$		12

подпадают под созданные правила ufw. Все входящие пакеты будем отклонять. Представлено на рисунке 2.4.8.

Файл Правка Вид Поиск Терминал Справка danilbru@business.r:~\$ sudo ufw default deny incoming Default incoming policy changed to 'deny' (be sure to update your_rules accordingly)

Рисунок 2.4.8 – Указание политики по умолчанию

Чтобы создать разрешающее правило, используется команда allow. Вместо allow могут использоваться и запрещающие правила ufw - deny и reject. Они отличаются тем, что для deny компьютер отсылает отправителю пакет с уведомлением об ошибке, а для reject просто отбрасывает пакет и ничего не отсылает. Для добавления правил можно использовать простой синтаксис:

- □ \$ ufw allow имя службы
- □ \$ ufw allow порт
- □ \$ ufw allow порт/протокол

Посмотреть доступные имена приложений можно с помощью команды: \$ sudo ufw app list. Представлено на рисунке 2.4.9.

```
Файл Правка Вид Поиск Терминал Справка danil.bru@business.r:~$ sudo ufw app list Available applications:
CUPS
OpenSSH
```

Рисунок 2.4.9 – Доступные имена приложений

Можно также указать направление следования трафика с помощью слов оut для исходящего и in для входящего. Например, разрешим только исходящий трафик на порт 80, а входящий запретим. Представлено на рисунке 2.4.10.

```
Файл Правка Вид Поиск Терминал Справка
danil.bru@business.r :~$ sudo ufw allow out 80/tcp
Rule added
Rule added (v6)
danil.bru@business.r :~$ sudo ufw deny in 80/tcp
Rule added
Rule added
Rule added (v6)
```

Рисунок 2.4.10 – Указание следования трафика

						%→
					<i></i>	10
∌ ←	% →	<i>⊕ זע\⇔Я</i>	& ¥7¢	♦→ <i>∅</i>		13

Чтобы удалить правило ufw, используется команда delete. Представлено на рисунке 2.4.11.

```
Файл Правка Вид Поиск Терминал Справка
danil.bru@business.r:~$ sudo ufw delete allow out 80/tcp
Rule deleted
Rule deleted (v6)
danil.bru@business.r:~$ sudo ufw delete deny in 80/tcp
Rule deleted
Rule deleted
Rule deleted (v6)
```

Рисунок 2.4.11 – Удаление правил

Для того, чтобы полностью отключить UFW, используется команда disable, а для того, чтобы сбросить настройки до состояния по умолчанию – reset.

						% →
					<i>acolore</i>	11
£ €	%→	⊕ ⊿⊻∜⇔Я	& YA	$Arr \rightarrow \emptyset$		14

- 3 Проектирование программного обеспечения для решения прикладной задачи
- 3.1 Постановка задачи. Техническое задание на разработку программного продукта
 - 3.1.1 Введение
 - 3.1.1.1 Наименование программы

Наименование программы – «Эво-тест».

3.1.1.2 Краткая характеристика области применения

Программа «Эво-тест» предназначена для диагностики и обслуживания кассовых аппаратов.

3.1.2 Основания для разработки

Разработка выполняется на основании задания по практике.

3.1.3 Назначение разработки

Программа будет использоваться потребителями ККТ компании.

3.1.3.1 Функциональное назначение

Пользователям программа позволяет отправлять заявку на консультацию, а также диагностику ККТ. Просматривать текущий тариф обслуживания.

3.1.3.2 Эксплуатационное назначение

Программа должна эксплуатироваться пользователями при проблемах с использованием ККТ. Конечными пользователями программы являются пользователи.

- 3.1.4 Требования к программе или программному изделию
- 3.1.4.1 Требования к функциональным характеристикам
- 3.1.4.1.1 Требования к составу выполняемых функций

						%→
					<i> </i>	1 -
∌ ←	%→	<i>⊕ זעע⇔ז</i>	& Y 7¢	$\Leftrightarrow \rightarrow \emptyset$		15

обеспечивать Программа должна выполнения возможность перечисленных ниже функций: □ функции отправки заявки об консультации пользователя; □ функции отправки заявки об диагностике ККТ; □ функции просмотра текущего тарифа пользователя. 3.1.4.1.2 Требования к организации входных и выходных данных Данные к заявке формируются на основании введенных данных пользователя. Данные приложения хранятся локально. 3.1.4.1.3 Требования к временным характеристикам Временны требования на разработку приложения – 1 месяц. 3.1.4.2 Требования к надежности 3.1.4.2.1 Требования К обеспечению надежного (устойчивого) функционирования программы Надежное (устойчивое) функционирование программы должно быть обеспечено выполнением Заказчиком совокупности организационнотехнических мероприятий, перечень которых приведен ниже: □ использованием лицензионного программного обеспечения; 🗆 регулярным выполнением рекомендаций Министерства труда и социального развития РФ, изложенных в Постановлении от 23 июля 1998 г. «Об утверждении межотраслевых типовых норм времени на работы по сервисному обслуживанию ПЭВМ оргтехники сопровождению И И программных средств»; □ регулярным выполнением требований ГОСТ 51188-98. информации. Испытания программных средств на наличие компьютерных вирусов. 3.1.4.2.1 Время восстановления после отказа

Время восстановления после отказа, вызванного сбоем электропитания технических средств (иными внешними факторами), не фатальным сбоем (не крахом) операционной системы, не должно превышать 10 минут при условии соблюдения условий эксплуатации технических и программных средств.

Время восстановления после отказа, вызванного неисправностью технических средств, фатальным сбоем (крахом) операционной системы, не должно превышать времени, требуемого на устранение неисправностей технических средств и переустановки программных средств.

3.1.4.2.3 Отказы из-за некорректных действий оператора

Отказы программы возможны вследствие некорректных действий пользователя при взаимодействии с операционной системой. Во избежание возникновения отказов программы по указанной выше причине следует обеспечить работу пользователя без предоставления ему административных привилегий.

- 3.1.4.3 Условия эксплуатации
- 3.1.4.3.1 Климатические условия эксплуатации

Специальные условия не требуются.

3.1.4.3.2 Требования к видам обслуживания

Программа не требует проведения каких-либо видов обслуживания.

3.1.4.3.3 Требования к численности и квалификации персонала

Требований к квалификациям конечного пользователя нет

3.1.4.4 Требования к составу и параметрам технических средств

В состав технических средств должен входить смартфон на операционной системе Android или IOS.

- 3.1.4.5 Требования к информационной и программной совместимости
- 3.1.4.5.1 Требования к информационным структурам и методам решения

						%→
					<i>]_@£406266666666666666666666666666666666666</i>	17
∌ ←	%→	⊕ ⊿⊻♦≉≉	& Y 7¢	$\Leftrightarrow \rightarrow \emptyset$		17

Требования к информационным структурам на входе и выходе, а также к методам решения не предъявляются.

3.1.4.5.2 Требования к исходным кодам и языкам программирования

Исходные коды программы должны быть реализованы на языке Dart с использованием фреймворка Flutter и стейт менеджером GetX в среде разработки Visual Studio.

3.1.4.5.3 Требования к программным средствам, используемым программой

Системные программные средства, используемые программой, должны быть представлены лицензионной локализованной версией операционной системы.

3.1.4.5.4 Требования к защите информации и программ

Такие данные пользователей как ИНН, номер телефона и код терминала должны быть защищены специальным шифрованием.

3.1.4.5.5 Требования к транспортированию и хранению

Требования не предъявляются.

3.1.4.5.6 Специальные требования.

Программа должна обеспечивать взаимодействие с пользователем посредством графического пользовательского интерфейса.

3.1.5 Требования к программной документации

В состав программнои документации должны входить:
□ техническое задание;
□ руководство пользователя;
П примеры тестирования программы.

3.1.6 Технико-экономические показатели

						26
					<i>#####################################</i>	1
∌ ←	% →	⊕ ЛУ ₩ Я	& Y 7 (=	$\Leftrightarrow \rightarrow \emptyset$		1

Ориентировочная экономическая эффективность не рассчитываются.
Предполагаемое число использования программы в год – 12 сеансов работы.
3.1.7 Стадии и этапы разработки
Разработка должна быть проведена в три стадии:
□ техническое задание;
□ проектирование и разработка;
□ внедрение.
На стадии «Техническое задание» должен быть выполнен этап
разработки, согласования и утверждения настоящего технического задания.
На стадии «Проектирование и разработка» должны быть выполнены
перечисленные ниже этапы работ:
□ разработка программы;
□ разработка программной документации;
□ испытания программы.
На стадии «Внедрение» должен быть выполнен этап разработки
«Подготовка и передача программы».
Содержание работ по этапам:
На этапе разработки технического задания должны быть выполнены
перечисленные ниже работы:
□ постановка задачи;
□ определение и уточнение требований к техническим средствам;
□ определение требований к программе;
□ определение стадий, этапов и сроков разработки программы и
документации на нее;
□ согласование и утверждение технического задания.
На этапе разработки программы должна быть выполнена работа по
программированию (кодированию) и отладке программы.

 \oplus \wedge

∌← %→

 \mathcal{H}

19

На этапе разработки программной документации должна быть выполнена разработка программных документов в соответствии с требованиями ГОСТ 19.101-77.

На этапе испытаний программы должны быть выполнены перечисленные ниже виды работ:

□ разработка, согласование и утверждение порядка и методики испытаний;

□ проведение приемо-сдаточных испытаний;

□ корректировка программы и программной документации по результатам испытаний.

На этапе подготовки и передачи программы должна быть выполнена работа по подготовке и передаче программы и программной документации в эксплуатацию на объектах заказчика.

3.1.8 Порядок контроля и приемки

Приемосдаточные испытания программы должны проводиться согласно разработанной исполнителем и согласованной заказчиком «Программы и методики испытаний».

Ход проведения приемо-сдаточных испытаний заказчик и исполнитель документируют в протоколе испытаний.

На основании протокола испытаний исполнитель совместно с заказчиком подписывают акт приемки-сдачи программы в эксплуатацию.

3.2 Описание программы

Программа Эво-Тест написана на языке Dart с использованием фреймворка Flutter. Программа имеет модульную структуру. Модульная схема представлена на рисунке 3.2.1.

						%-
						20
∌ ←	% →	⊕ ЛУ У⇔ Л	& Y 70	♥→ ∅		20

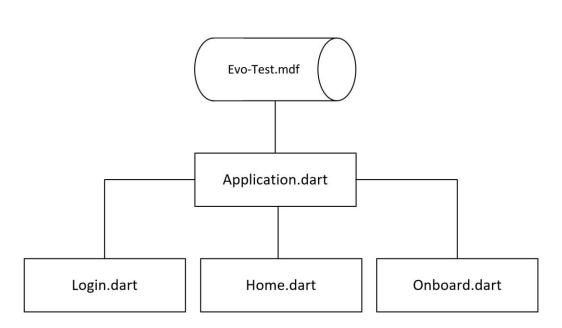


Рисунок 3.2.1 – Модульная схема

Описание модулей и процедур указано в таблице 3.2.1.

Таблица 3.2.1 – Описание модулей и процедур

Процедуры	Назначение	
1	2	
Модуль Application.dart	-Модуль приложения	
initServiceManager()	Инициализация сервисов	
registerPages	Регистрация страниц	
GetMaterialApp	Создание виджета приложения	

Продолжение таблицы 3.2.1

1	2			
Модуль Onboard.dart – Модуль онбординга				
nextPage()	Переход на следующее окно			
Redirect()	Переход на окно авторизации			
Модуль Login.dart – М				
Login()	Авторизация пользователя			
DatabaseSave()	Локальное сохранение			
	пользователя			
SaveInn()	Сохранение ИНН пользователя			
Skip()	Пропустить окно			
FillQr()	Заполнение данных с QR			

						%→
					######################################	21
∌ ←	%→	<i>⊕ זע\⇔ז</i>	& Y 7 &	♦→ <i>∅</i>	 	21

Модуль Home.dart – Модуль основного окна				
Refresh()	Обновление данных пользователя			
SendConsultation()	Отправка заявки на консультацию			
SendDiagnostic()	Отправка заявки на диагностику			
Save()	Сохранение данных			

3.3 Протокол тестирования разработанного программного продукта

В ходе тестирования программного продукта на корректных и некорректных данных не было обнаружено ошибок, которые влияли бы на работу самого программного продукта и всей системы.

Данный программный продукт удовлетворяет всем предъявленным требованиям, имеет комфортный интерфейс и интуитивно понятный функционал, исключает появления системных ошибок.

В таблице 3.3.1 представлена общая информация о тестировании. В таблицах 3.3.2-3.3.4 представлены протоколы тестирования: тестирование авторизации сотрудника на корректных данных, тестирование авторизации сотрудника на некорректных данных, тестирование на получение статистики за выбранный день, в котором есть данные, тестирование на получение статистики за выбранный день, в котором нет данных.

Таблица 3.3.1 – Общая информация о тестировании

Общая информация о тестировании				
Название проекта	Evo_test_flutter			
Номер версии	1.0			
Имя тестера	Исламов Данил Рустамович			
Даты тестирования	01.11.2024			

Таблица 3.3.2 – Протокол тестирования добавления на корректных данных

Описание информационных полей для тестирования		
Наименование	Описание	

						8
					######################################	2.
₽	% →	<i>⊕ זעע⇔ז</i>	& ¥ 7<₽	♦→ <i>∅</i>		22

1	2
Test Case #	testcase #1
Приоритет	Высокий
тестирования	BBCORIN

Продолжение таблицы 3.3.2

1	2
Наименование	Проверка корректности ввода ИНН
тестирования	
Резюме испытания	Необходимо добиться корректного поведения
1 CSIONE PICTIBITATIVA	программы при вводе корректных данных
Шаги тестирования	Ввод корректных данных в текстовые поля;
шаги гестирования	Нажатие кнопки «Активировать».
Данные тестирования	ИНН: 123456789111
Ожидаемый результат	Вывод сообщения с названием организации.
Фактический результат	Вывод сообщения названия организации «тест
Фактический результат	123»
Предпосылки	Переход на окно авторизации
Постусловия	Активация кнопки «Активировать»
Статус (Pass/Fail)	Pass
Комментарии	Организация найдена корректно

Результат тестирования добавления на корректных данных представлен на рисунке 3.3.2.

						H
					<i></i>	١,
∌ ←	%→	⊕ ⊿⊻∜⇔Я	& Y 74	$\Leftrightarrow \rightarrow \emptyset$		



Рисунок 3.3.2— Результат тестирования добавления записи с корректными данными

Таблица 3.3.3— Протокол тестирования добавления на некорректных данных

Описание информационных полей для тестирования				
Наименование	Описание			
1	2			
Test Case #	testcase #2			
Приоритет	Высокий			
тестирования	Высокии			
Наименование	Проверка добавления кода терминала			
тестирования	посредством QR и текстового ввода			
Резюме испытания	Необходимо добиться корректного поведения			
т сложе испытания	программы при вводе корректных данных			

						% →
					<i> </i>	24
∌ ←	% →	<i>⊕ זעע⇔ז</i>	& Y 7&	♦→ ∅		24

	Ввод данных, проверка QR данных в текстовые
Шаги тестирования	поля;
	Нажатие кнопки «Подключить».
Данные тестирования	Код терминала: 12341567894687

Продолжение таблицы 3.3.3

1	2
Ожидаемый результат	Касса подключена
Фактический результат	Касса подключена и интерфейс соответствует статусу
Предпосылки	Авторизация и ввод данных
Постусловия	Переход на основное окно
Статус (Pass/Fail)	Pass
Комментарии	Работает

Результат тестирования добавления на некорректных данных представлен на рисунке 3.3.3.

						9
					<i>#####################################</i>	_
∌ ←	% →	⊕ 7¥ \⇔7	& Y 74	♦→		-

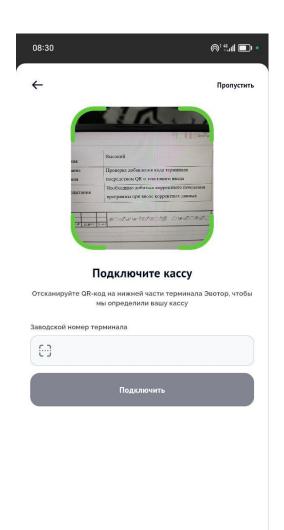


Рисунок 3.3.3— Результат тестирования добавления записи с некорректными данными

Таблица 3.3.4— Протокол тестирования добавления на некорректных данных

Описание информационных полей для тестирования							
Наименование	Описание						
1	2						
Test Case #	testcase #2						
Приоритет	Высокий						
тестирования	DDICORPIP						

Продолжение таблицы 3.3.4

						%→
					<i> </i>	26
∌ ←	% →	⊕ ⊿₹₹₩	& Y 7	♥→ ∅		26

1	2
Наименование	Проверка корректности ввода номера телефона
тестирования	
Резюме испытания	Необходимо добиться корректного поведения
T CSTOME HORBITATION	программы при вводе некорректных данных
Шаги тестирования	Ввод некорректных данных в текстовые поля.
Данные тестирования	Номер телефона: +7(999)-11
Ожидаемый результат	Неактивная кнопка
Фактический результат	Кнопка «Продолжить» неактивна
Предпосылки	Авторизация пользователя
Постусловия	Переход на основное окно
Статус (Pass/Fail)	Pass
Комментарии	Работает

Результат тестирования добавления на некорректных данных представлен на рисунке 3.3.4.



Рисунок 3.3.4 – Результат тестирования с некорректным номером телефона

						% →
					<i></i>	27
∌ ←	% →	<i>⊕ זעע⇔ז</i>	& Y 7&	♦→ ∅		2/

3.4 Руководство пользователя

Цель руководства заключается в расписании подробной инструкции и информации для пользователя, чтобы он смог самостоятельно пользоваться программой и правильно ее эксплуатировал.

Программа «Эво-тест» предназначена для пользователей использующих ККТ компании.

Для запуска программы необходимо запустить приложение Эво-Тест из системного каталога устройства, в котором установлен данный программный продукт. После этого открывается окно онбординга в котором есть возможность просмотра основных возможностей программы. Это показано на рисунке 3.4.1.

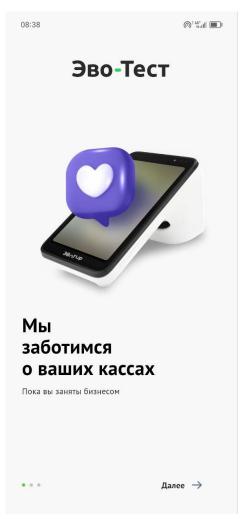


Рисунок 3.4.1- Окно онбординга

						%)
					######################################	20
∌ ←	%→	⊕ ЛУ ₩ Я	& ¥ 7¢	♦→ <i>∅</i>	= .	28

Для авторизации пользователя необходимо перейти на следующее окно соглашения обработки персональных данных, а на нем нажать кнопку «Принять и продолжить». После соглашения пользователь перейдет на окно ввода ИНН организации. Необходимо ввести ИНН организации для продолжения. Окно авторизации показано на рисунке 3.4.2.



Рисунок 3.4.2—Окно добавления ИНН

После ввода ИНН необходимо нажать на кнопку «Активировать». После загрузки данных пользователя перебросит на окно добавления данных терминала, нужно ввести данные терминала вручную или отсканировать по QR. После этого приложение попросит ввести контактных номер телефона для обратной. Это показано на рисунке 3.4.3.

						% →
					<i> </i>	20
∌ ←	%→	⊕ ⊿⊻♦≉≉	& Y 7¢	♦→ <i>∅</i>		29

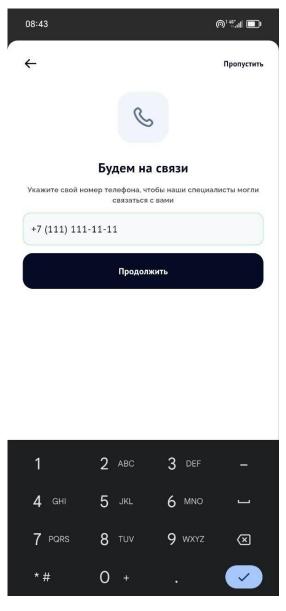


Рисунок 3.4.3- Окно добавления номера телефона

После ввода номера телефона пользователь попадет н основное окно, где может увидеть статус подключения кассы и свой текущий тариф. Справа сверху можно перейти на окно настроек. Если потянуть белую панель вниз, то произойдет обновление данных. Основное окно тестового аккаунта на рисунке 3.4.4.

					<i> </i>	
₽€	%→	⊕ ⊿⊻♦≉≉	& Y 7¢	♦→ <i>∅</i>		

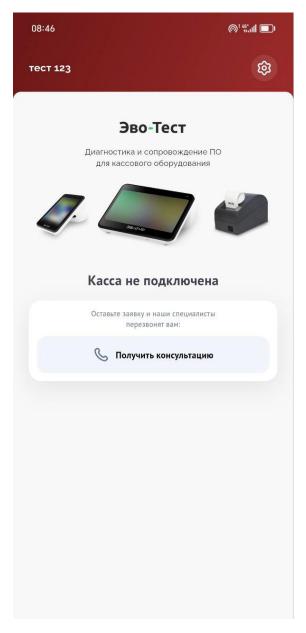


Рисунок 3.4.4— Основное окно

 \mathcal{H}

31

ЗАКЛЮЧЕНИЕ

В ходе прохождения учебной практики в компании ООО «Класс Информационные Системы» были получены знания о деятельности предприятия, была разработана программа на тему «Эво-Тест», которая была выполнена в соответствии с поставленной задачей.

Перед началом практики были пройдены инструктажи по технике безопасности на заводе и на рабочем месте.

Более подробно ознакомилась с деятельностью департамента разработки узнал о его информационных системах, используемых в работе, и задачах, которые они выполняют. Так же познакомилась с системой информационной безопасности, существующей в компании.

Задачи производственной практики:

□ ознан	сомление с	пре	едпр	иятием (отд	дело	м), его ф	ункциям	и, стр	уктурой;
□ ознан	комление	c	име	еющимися	В	отделе	автома	тизирс	ванными
системами,	аппаратн	ЫМ	И	программі	ным	обеспе	чением	ПК,	сетевым
обеспечение	ем, антивиј	русн	НЫМІ	и программ	ами;				

□ настройка защиты системы стандартными средствами операционной системы;

□ разработка программного продукта, согласно задачам, выполняемым в отделе.

При выполнении заданий от руководителя практики от организации использовала знания, полученные в ходе обучения по специальности. В ходе прохождения производственной практики были решены все задачи, поставленные цели достигнуты.

					<i></i>	
₽€	% →	<i>⊕ זעע⇔ז</i>	& ¥ 7<₽	♦→ <i>∅</i>	- .	

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 ГОСТ 19.701-90 ЕСПД. Схемы алгоритмов, программ, данных и систем. Обозначения условные и правила выполнения [Текст]. Введ. 1992–01–01. М.: Изд-во стандартов, 1992. 24 с.
- 2 ГОСТ 2.105 95. Общие требования к текстовым документам. Единая система конструктивной документации [Текст]. Введ. 1996 07 01. М.: Стандартинформ, 2007.
- 3 ГОСТ 34.601-90 Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания [Текст]. Введ. 1990 01 01. М.: Стандартинформ, 2007.
- 4 ГОСТ Р 51583-2014. Защита информации. Порядок создания автоматизированных систем в защищенном исполнении. Общие положения. [Текст]. Введ. 2014- 09 01. М.: Стандартинформ, 2007.
- 5 Базовая модель угроз безопасности персональных данных при их обработке в информационных системах персональных данных [Электронный ресурс]: утв. ФСТЭК России 15 февраля 2008 // Информационно-правовая система КонсультантПлюс.
- 6 Руководящий документ. Концепция защиты средств вычислительной техники и автоматизированных систем от несанкционированного доступа к информации. [Электронный ресурс]: утв. решением Государственной технической комиссии при Президенте РФ от 30 марта 1992 // Информац.-правовая система Гарант.
- 7 Белов, Е.Б. Организационно-правовое обеспечение информационной безопасности [Текст]: учеб. пособ. для студ. учрежд. СПО / Е.Б. Костров, В.Н. Пржегорлинский. М.: Издательский центр "Академия", 2017. 336 с.
- 8 Якобсон А. Язык UML. Руководство пользователя. Второе издание. ДМК, 2016, 496 с.

						26-
						22
∌ ←	%→	<i>⊕ זעע⇔ז</i>	& Y 7¢	♥→ ∅		33

- 9 Википедия [Электронный ресурс] // Свободная энциклопедия. Режим доступа: http://ru.wikipedia.org/wiki/, свободный.
- 10 METANIT [Электронный ресурс] // Сайт о программировании. Режим доступа: https://metanit.com/, свободный.
 - 11 Dart [Электронный ресурс]. Режим доступа: https://dart.dev
- 12 Sound null safety [Электронный ресурс]. Режим доступа: https://dart.dev/null-safety
 - 13 Flutter [Электронный ресурс]. Режим доступа: https://flutter.dev
- 14 Effective Dart: Documentation [Электронный ресурс]. Режим доступа: https://dart.dev/guides/language/effective-dart/documentation
- 15 String class [Электронный ресурс]. Режим доступа: https://api.dart.dev/stable/3.1.3/dart-core/String-class.html
- 16 List<E> class [Электронный ресурс]. Режим доступа: https://api.dart.dev/stable/3.1.3/dart-core/List-class.html
- 17 Set<E> class [Электронный ресурс]. Режим доступа: https://api.dart.dev/stable/3.1.3/dart-core/Set-class.html
- 18 Map<K, V> class [Электронный ресурс]. Режим доступа: https://api.dart.dev/stable/3.1.3/dart-core/Map-class.html
- 19 Ожегов С. И. Толковый словарь русского языка: около 100 000 слов, терминов и фразеологических выражений [Текст] / С. И. Ожегов; под ред. Л. И. Скворцова. 26-е изд., испр. и доп. М.: Оникс [и др.], 2009. 1359 с.
- 20 Горский Д. П. Вопросы абстракции и образование понятий [Текст]. М.: Изд-во Академии наук СССР, 1961. 353 с.
- 21 ГОСТ 28397-89 (ИСО 2382-15-85) Языки программирования. Термины и определения : Межгосударственный стандарт [Текст] / Федеральное агентство по техническому регулированию. Изд. официальное. Москва : Стандартинформ, 2010. 8 с.
- 22 Typedefs [Электронный ресурс]. Режим доступа: https://dart.dev/language/typedefs

						% →
					<i> </i>	24
∌ ←	%→	<i>⊕ זעע⇔ז</i>	& ¥7¢	$\Leftrightarrow \rightarrow \emptyset$		34

- 23 PREFER inline function types over typedefs [Электронный ресурс]. Режим доступа: https://dart.dev/effective-dart/design#prefer-inline-function-types-over-typedefs
- 24 Creating packages [Электронный ресурс]. Режим доступа: https://dart.dev/guides/libraries/create-library-packages
- 25 Dart testing [Электронный ресурс]. Режим доступа: https://dart.dev/guides/testing
- 26 dart compile [Электронный ресурс]. Режим доступа: https://dart.dev/tools/dart-compile
- 27 Concurrency in Dart [Электронный ресурс]. Режим доступа: https://dart.dev/language/concurrency
- 28 Streams and Sinks in Dart and Flutter [Электронный ресурс]. Режим доступа: https://dart.academy/streams-and-sinks-in-dart-and-flutter/
- 29 Dart asynchronous programming: Isolates and event loops [Электронный ресурс]. Режим доступа: https://medium.com/dartlang/dart-asynchronous-programming-isolates-and-event-loops-bffc3e296a6a
- 30 spawnUri static method [Электронный ресурс]. Режим доступа: https://api.dart.dev/stable/3.1.4/dart-isolate/Isolate/spawnUri.html
- 31 send abstract method [Электронный ресурс]. Режим доступа: https://api.dart.dev/stable/3.1.4/dart-isolate/SendPort/send.html
- 32 Isolate class [Электронный ресурс]. Режим доступа: https://api.dart.dev/stable/3.1.4/dart-isolate/Isolate-class.html
- 33 Zone class [Электронный ресурс]. Режим доступа: https://api.dart.dev/stable/3.1.4/dart-async/Zone-class.html
- 34 HTTP Messages [Электронный ресурс]. Режим доступа: https://developer.mozilla.org/en-US/docs/Web/HTTP/Messages
- 35 Wei-Meng Lee. Go Programming Language For Dummies [Текст] / John Wiley & Sons, Inc., 2019. 336 с. ISBN: 978-1-119-78619-1

						% →
					######################################	25
∌ ←	% →	⊕ ७४५ ₽	& ¥ 7¢	♥→ ∅		33

36 REST API Tutorial [Электронный ресурс]. — Режим доступа: https://restfulapi.net

37 REST API conventions [Электронный ресурс]. — Режим доступа: https://www.ibm.com/docs/en/urbancode-release/6.1.1?topic=reference-rest-api-conventions

						% →
					<i>]_@D\$vo^\$\$D</i> }_	26
∌ ←	%→	⊕ ⊿⊻↓⇔Я	& YA	$\Diamond \rightarrow \emptyset$		36