

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН  
Некоммерческое акционерное общество  
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ  
им. ГУМАРБЕКА ДАУКЕЕВА»

Кафедра IT - инжиниринг

«ДОПУЩЕН К ЗАЩИТЕ»

Зав. кафедрой \_\_\_\_\_

(ученая степень, звание, Ф.И.О.)

\_\_\_\_\_ «\_\_» \_\_\_\_\_ 2020 г.

(подпись)

**ДИПЛОМНЫЙ ПРОЕКТ**

На тему: Разработка информационной системы для автоматизации  
деятельности ресторана

Специальность: Информационные системы – 5В070300

Выполнил: Ануарбеков Ш.Н.

Группа: ИС 16-2

Научные руководитель: PhD, доцент Кожамкулова Ж.Ж.

Консультанты:

по экономической части: к.э.н., профессор Габелашвили К.Р

(учёная степень, звание, Ф.И.О.)

\_\_\_\_\_ «\_\_» \_\_\_\_\_ 2020 г.

по безопасности жизнедеятельности: ассистент Тыщенко Е.М

(учёная степень, звание, Ф.И.О.)

\_\_\_\_\_ «\_\_» \_\_\_\_\_ 2020 г.

по программному обеспечению: ст.преп. Майкотов М.Н

(учёная степень, звание, Ф.И.О.)

\_\_\_\_\_ «\_\_» \_\_\_\_\_ 2020 г.

Нормоконтролер: ст.преп. Абсатарова Б.Р

(учёная степень, звание, Ф.И.О.)

\_\_\_\_\_ «\_\_» \_\_\_\_\_ 2020 г.

Рецензент: \_\_\_\_\_

(учёная степень, звание, Ф.И.О.)

\_\_\_\_\_ «\_\_» \_\_\_\_\_ 2020 г.

Алматы 2020

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН  
Некоммерческое акционерное общество  
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ  
им. ГУМАРБЕКА ДАУКЕЕВА»

Институт систем управления и информационных технологий

Кафедра IT-инжиниринг

Специальность 5B070300 – Информационные системы

**ЗАДАНИЕ**

на выполнение дипломного проекта

Студенту Ануарбекову Шынғыс Нұрланұлы

Тема проекта: Проектирование автоматизированной информационной системы в ресторанном бизнесе

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): руководство по программированию PHP, интегрированная среда Visual Studio Enterprise, данные преддипломной практики.

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

- а) анализ и исследование предметной области;
- б) проектирование приложения;
- в) реализация приложения;
- г) экономическое обоснование эффективности;
- д) вопросы безопасности жизнедеятельности и охраны труда.

Перечень графического материала (с точным указанием обязательных чертежей): отсутствует

Основная рекомендуемая литература:

- 1 Скляр, Дэвид. Изучаем PHP 7: руководство по созданию интерактивных веб-сайтов: Пер. с англ. - СПб.: ООО “Альфа-книга”, 2017. - 464 с.

Консультация по проекту с указанием относящихся к ним разделов проекта

Раздел	Консультант	Сроки	Подпись
Экономическая часть	Габелашвили К.Р.	21.04.2020	
Безопасности жизнедеятельности	Тыщенко Е.М.	30.04.2020	
Программная часть	Майкотов М.Н.	14.05.2020	
Нормконтролер	Абсатарова Б.Р.	13.05.2020 – 18.05.2020	

ГРАФИК  
подготовки дипломной работы (проекта)

Наименования разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю	Примечания
Анализ и исследование предметной области	17.02.2020 – 24.02.2020	
Проектирование приложения	25.02.2020 – 31.03.2020	
Программная реализация	01.04.2020 – 01.05.2020	

Дата выдачи задания « \_\_\_\_ » \_\_\_\_\_ 2020 г.

Заведующий кафедрой \_\_\_\_\_ А.А.Досжанова  
(подпись) (Ф.И.О.)

Научный руководитель \_\_\_\_\_ Ж.Ж. Кожамкулова  
проекта (подпись) (Ф.И.О.)

Задание принял к исполнению \_\_\_\_\_ Ш.Н. Ануарбеков  
студент (подпись) (Ф.И.О.)

## **Аннотация**

В данном дипломном проекте показан ход работы над реализацией программного продукта для оптимизации работы ресторана. Разработанное приложение помогает сотрудникам в работе ресторана, а также оптимизирует отчетность их деятельности. Возможности программного продукта позволяют непосредственно с приложения оформить заказ, осмотр всех имеющихся на данный момент категорий и продукции а также их редактирование, просмотр всех заказов за различный период времени. В программе также заложены возможности по аналитике реализации продукта в течение всего существования ресторана.

Приложение было разработано на языке программирования HTML, CSS, PHP, JavaScript с использованием редактора кода SublimeText3 и фреймворка CodeIgniter3. В разработке были использованы различные плагины для оптимизации работы приложения.

Также в работе также рассматриваются возможности и сравнение различных типов СУБД, экономическая сторона разработки данного проекта, и рассчитываются условия безопасности жизнедеятельности.

## **Annotation**

This graduation project shows the progress on the implementation of a software product to optimize the restaurant. The developed application helps employees in the work of the restaurant, and also optimizes the reporting of their activities. The capabilities of the software product allow you to place an order directly from the application, inspect all currently available categories and products, as well as edit them, view all orders for a different period of time. The program also includes analytic capabilities for product sales throughout the restaurant's existence.

The application was developed in the programming language HTML, CSS, PHP, JavaScript using the SublimeText3 code editor and the CodeIgniter3 framework. In the development, various plugins were used to optimize the application.

The work also discusses the possibilities and comparison of various types of DBMS, the economic side of the development of this project, and calculates the conditions of life safety.

## **Аннотация**

Бұл дипломдық жоба мейрамхананы оңтайландыруға арналған бағдарламалық өнімдерді енгізу барысын көрсетеді. Әзірленген қосымша қызметкерлерге мейрамхана жұмысында көмектеседі, сонымен қатар олардың қызметі туралы есеп беруді оңтайландырады. Бағдарламалық жасақтаманың мүмкіндіктері сізге тікелей қосымшадан тапсырыс беруге, барлық қол жетімді санаттар мен өнімдерді тексеруге, оларды өңдеуге, әртүрлі уақыт кезеңіндегі барлық тапсырыстарды қарауға мүмкіндік береді. Бағдарламада сонымен қатар мейрамхананың бүкіл кезеңінде өнімді сатуға арналған аналитикалық мүмкіндіктер бар.

Бағдарлама HTML, CSS, PHP, JavaScript бағдарламалау тілінде SublimeText3 кодтық редакторы мен CodeIgniter3 жүйесін қолдана отырып жасалған. Әзірлеуде қосымшаны оңтайландыру үшін әртүрлі плагиндер қолданылды.

Сондай-ақ, жұмыста ДББЖ-дің әртүрлі мүмкіндіктері мен салыстыруы, осы жобаны әзірлеудің экономикалық жағы және өмір қауіпсіздігі жағдайлары есептеледі

## Содержание

Введение .....	8
1 Аналитическая часть .....	10
1.1 Содержательное описание предметной области.....	10
1.2 Обоснование необходимости автоматизации объекта исследования.....	14
1.3 Обзор программных и инструментальных средств.....	16
2 Проектная часть.....	17
2.2 Этапы создания ПП.....	18
2.3 Case-проектирование ПП .....	20
2.3.1 Диаграмма вариантов использования .....	20
2.3.2 Диаграмма последовательности .....	23
2.4 Выбор средств для разработки.....	25
2.5 Разработка базы данных.....	26
2.6 Структура программного продукта.....	36
2.7 Разработка программного продукта .....	39
2.8 Размещение Веб-приложения .....	57
2.9 Тестирование Веб-приложения.....	58
3 Экономическая часть.....	60
3.1 Резюме .....	60
3.2 Трудоемкость разработки ПП.....	61
3.3 Расчет затрат на разработку ПП.....	62
3.4 Расчет сравнительной экономической эффективности ПП.....	69
3.5 Вывод по экономическому разделу.....	71
4 Безопасность жизнедеятельности.....	72
4.1 Анализ потенциально опасных и вредных факторов воздействия на персонал в процессе работы .....	72
4.2 Расчетная часть.....	74
4.2.1 Исходные данные.....	74
4.2.2 Расчет тепловых нагрузок в помещении .....	75
4.2.3 Расчет теплового баланса помещения.....	78
4.2.4 Выбор кондиционера. Схема расположения.....	79
4.2.5 Расчет искусственного освещения.....	82
Заключение .....	84
Список литературы.....	85
Приложение А.....	86
Приложение Б.....	120

## Введение

Информационная система – интегрированный набор компонентов для сбора, хранения и обработки данных, а также для предоставления информации, знаний и цифровых продуктов. Коммерческие фирмы и другие организации полагаются на информационные системы для выполнения и управления своей деятельностью, взаимодействуя со своими клиентами и поставщиками, а также конкурировать на рынке. Информационные системы используются для управления внутриорганизационных систем поставок и обмена информацией. Например, корпорации используют информационные системы для обработки финансовых счетов, управления своими человеческими ресурсами и привлечения потенциальных клиентов с помощью онлайн-акций. Многие крупные компании построены исключительно вокруг информационных систем. К ним относятся eBay, в основном аукционный рынок; Amazon, расширяющийся электронный торговый центр и поставщик услуг облачных вычислений; Alibaba, электронная торговая площадка для бизнеса; и Google, компания поисковых систем, которая получает большую часть своего дохода от рекламы по ключевым словам в поисковых запросах в Интернете. Правительства внедряют информационные системы для предоставления гражданам экономически эффективных услуг. Цифровые товары, такие как электронные книги, видеопродукция и программное обеспечение, и онлайн-сервисы, такие как игры и социальные сети, поставляются с информационными системами. Люди полагаются на информационные системы, обычно основанные на веб-сервисах, для ведения большей части своей личной жизни: для общения, учебы, покупок, банковских операций и развлечений.

Поскольку информационные системы способствовали более разнообразной человеческой деятельности, они оказали глубокое влияние на общество. Эти системы ускорили темпы повседневной деятельности, позволили людям развивать и поддерживать новые и зачастую более выгодные отношения, влияли на структуру и состав организаций, меняли тип покупаемых продуктов и влияли на характер работы. Информация и знания стали жизненно важными экономическими ресурсами. Большая категория информационных систем включает системы, предназначенные для поддержки управления организацией. Эти системы основаны на данных, полученных системами обработки транзакций, а также данными и информацией, полученными вне организации (например, в Интернете) и предоставляемыми деловыми партнерами, поставщиками и клиентами.

Информационные системы поддерживают все уровни управления, от тех, кто отвечает за краткосрочные графики и бюджеты для небольших рабочих групп, до тех, кто занимается долгосрочными планами и бюджетами для всей организации. Системы управленческой отчетности предоставляют рутинные, подробные и объемные информационные отчеты, специфичные для областей ответственности каждого менеджера. Эти системы обычно используются руководителями первого уровня. Как правило, такие отчеты фокусируются на



прошлой и настоящей деятельности, а не на прогнозировании будущих результатов. Во избежание информационной перегрузки отчеты могут автоматически отправляться только при исключительных обстоятельствах или по специальному запросу менеджера.

В настоящее время информационную систему все больше и больше используют для автоматизации всевозможных процессов, наблюдаемых в разных предприятиях, компаниях и корпорациях. IT технологии очень прочно вошли в работу бизнес сферы.

При автоматизации бизнеса ускоряется работа всего предприятия, что естественно положительно сказывается на порядок и организованность работы, а также повышении денежного притока.

Но ввести систему в бизнес это очень долгий и трудоемкий процесс. Очень много факторов влияющий на это. Один из самых распространенных проблем внедрения информационных систем – высокий процент провалов. Внедрение системы замедляется, или приостанавливается из-за отсутствия взаимопонимания между программистами и заказчиками, и появляются случаи, при которых готовое решение никем не используется. Проблема в понятии информационной системы, заказчики нередко не знают, что именно им нужно в их ситуации. Информационная система в большинстве случаев автоматизирует лишь краткий аспект работы предприятия заказчика, а не полную работу.

Одно из сфер в котором все больше и больше автоматизированных предприятий является сфера общепита. Общественное питание играет все возрастающую роль в жизни современного общества. Это обеспечивается изменением технологий переработки продуктов питания, развитием коммуникаций, средств доставки продукции и сырья, интенсификацией многих производственных процессов. В HoReCa много мелких деталей которые существенно замедляют процесс приготовления продукта что в конечном итоге сказывается на качестве сервиса и управлении предприятия в целом.

Правильное внедрение автоматизации помогает упростить и ускорить работу.

## 1 Аналитическая часть

### 1.1 Содержательное описание предметной области

NoReCa – это емкое понятие, включающее в себя множество организаций, которые так или иначе причастны к оказанию услуг или продаже товаров в сфере гостеприимства [2].

Ресторан - это заведение, где закуски и блюда могут быть приобретены населением. Общественная столовая, которая в конечном итоге стала называться рестораном, возникла во Франции, и французы продолжают вносить большой вклад в развитие ресторана.

Предполагается, что первым владельцем ресторана был один из торговцев супом господин Буланже, который открыл свой бизнес в Париже в 1765 году. На табличке над его дверью были изображены ресторативы, в которых ссылались на супы и бульоны. Заведение получило свое название от этого знака, и «ресторан» теперь обозначает место общественного питания на английском, французском, голландском, датском, норвежском, румынском и многих других языках, с некоторыми вариациями. Несмотря на то, что в гостиницах и общежитиях гости часто оплачивали еду, а напитки продавались в кафе, ресторан Буланже был, вероятно, первым общественным местом, где любой посетитель мог заказать еду из меню, предлагающего выбор блюд.

Заведения общепита является очень распространённым бизнесом в больших мегаполисах. Самым выгодным представителем общепита является сеть заведений, успешно работающих по отложенной схеме обслуживания клиентов. Сетью может быть любой формат заведения – от фаст-фуда до дорогого ресторана. Есть разные формы в сфере общепита:

1) *Рестораны*. Средний чек колеблется в зависимости от местоположения и статуса заведения. Звеном заведения может быть разные элементы: тематика, атмосфера, интерьер, особые предложения. Имеет достаточно дорогую отделку, высокий уровень сервиса, особый внешний вид персонала, изысканные блюда и элитные алкогольные напитки. Также имеет довольно высокий ценник, но предложения по бизнес-ланчам может конкурировать с форматами заведений поменьше [2].

2) *Бистро и кафе*. Заведение меньше по площади чем классический ресторан, предлагающий демократичные цены. Может быть оборудовано ресторанным оборудованием [2].

3) *Пивные бары и пабы*. Это заведения вечернего режима, с контингентом людей, употребляющих пиво с закусками. Ассортимент блюд скромнен, что компенсируется большим количеством сортов пива [2].

4) *Кофейни и кондитерские*. Именно здесь основной элемент заведения кофе, разные десерты и выпечка. Ориентирован на молодых, продвинутых людях. Ассортимент «продает» кофе и чай [2].

5) *Бары*. Это также заведение вечернего времени, в котором люди приходят выпить и поговорить. Обычно в таких заведениях приготовление

пищи может быть минимальным, так как направленность больше на разнообразие алкоголя. Также могут отсутствовать посадочные места. Средний чек колеблется в зависимости от местоположения и статуса заведения [2].

6) *Фаст-фуды*. В основном это филиалы крупных сетей с одинаковыми предложениями во всех точках. На кухне, как правило, проходит доготовка продуктов, что увеличивает скорость готовки. Средний чек от 1500 до 3000 тг [2].

7) *Столовые*. Заведение с низкими ценами с самообслуживанием, являющимся наследием советского времени. Средний чек в районе 2000 тг [2].

8) *Суши-бары*. Очень модное на сегодня японское направление подающие суши и азиатские закуски. Средний чек в районе 5000 тг [2].

9) Микроформаты, где обычно отсутствуют посадочные места. В иностранных обозначениях называются Take-Out. Средний чек от 300 до 1200 тг [2].

Ресторан является предприятием с полным производственным циклом, на котором выполняются все стадии технологического процесса приготовления пищи, организуются универсальные рабочие места, характерные для бесцеховой структуры предприятия.

Они построены из комплексных систем для покупки, хранения, приготовления и продажи продуктов питания. Благосостояние ресторана зависит от его информационных систем управления, которые координируют все от планирования персонала до обслуживания клиентов. Информационные системы управления рестораном должны сделать ресторан более прибыльным, а также лучшим местом для клиентов.

Каждому ресторану нужна стратегия приема заказов, доставки информации на кухню и взимания платы за еду. Эти системы могут быть такими же простыми, как рукописные заметки, или такими же сложными, как компьютерные системы, которые отправляют заказы на кухню и подсчитывают продажи для каждой смены. Простые системы менее подвержены техническим трудностям, но они не могут обрабатывать информацию так же эффективно, как слаженно работающие компьютерные системы. Системы ресторанных точек продаж должны также включать инфраструктуру для обработки платежей по кредитным картам [1].

Рестораны зависят от обмена информацией между различными подразделениями, как серверы, передающие заказы кухонному персоналу и кухонный персонал, уведомляющий серверы, что их заказы готовы. Кроме того, системы ресторанного общения должны позволять сотрудникам связывать готовые блюда с заказчиками, а также передавать подробную информацию об особых запросах и особых потребностях. Менеджмент ресторана должен также разработать информационные системы для связи как с передней, так и с задней стороны дома о таких проблемах, как низкий запас по конкретным пунктам меню или ингредиентам [1].

Укомплектование ресторана персоналом может быть сложным делом, потому что спрос на еду, вероятно, будет сильно колебаться, часто из-за

переменных, которые вы не можете отследить. Важно определять любые переменные, которые влияют на трафик в ресторане, такие как погода и день недели. Составление еженедельного расписания для персонала ресторана в соответствии с этими переменными, такими как планирование дополнительного персонала в субботу вечером, если это самая загруженная смена также является частью ресторана, которую приходится анализировать. Скомпилирование данных о продажах и рабочих часах, для определения выгодного соотношения рабочих часов и общих продаж. Системы обучения персонала ресторанов также имеют жизненно важное значение для успеха, обеспечивая сотрудников знаниями протоколов и системы ресторана, и были способны предоставить высококачественные услуги.

В типовом формате ресторана в обслуживании гостя взаимодействуют почти все члены персонала и многие процессы работы. Вся деятельность персонала должна быть по правилам ресторана. Управляющий или менеджер должен следить за всеми, и поправлять недочеты сотрудников.

Стандартными сотрудниками почти любого общепита являются:

- 1) Директор;
- 2) Администратор (менеджер);
- 3) Кассир;
- 4) Официант;
- 5) Повар;
- 6) Бармен (только в заведениях с алкоголем).

В заведениях в сфере питания бухгалтерский учет, налоговый учет, кадровый учет, ревизии и инициативный аудит может выполнять аутсорсинговая фирма, предоставляющая такие услуги. В большинство заведений малого бизнеса нанимать таких сотрудников финансово не является выгодным, и они прибегают к услугам аутсорсинга.

В заведении у каждого сотрудника своя роль, которую только он должен выполнять.

Директор – контролирует деятельность работников ресторана, проводит анализ информации о получаемой прибыли, и в соответствии с этим осуществляет постановку целей и задач на следующий период деятельности ресторана.

Администратор – главный человек в коллективе управляющий персоналом. Большинство нетипичных задач как ошибки персонала, неудобный клиент, проблемы оборудования решает этот сотрудник.

Кассир – сотрудник, стоящий за кассовым аппаратом и открывающий новые чеки. В основу его работы входит открытие счета, расчет клиентов.

Официант – чаще всего является лицом ресторана так как первый встречает гостей. При входе опрятно выглядящий официант с хорошими манерами может только улучшить настроение гостей. Он же и проводит гостей к столу и принимает заказы. В мелких ресторанах один официант выполняет большую роль – он же принимает заказ, приносит блюда, и убирает стол после гостей. На западе официанты имеют собственную иерархию в одном заведении.

У нас же такую иерархию имеют лишь крупные, дорогие рестораны. Старший официант лишь следит за остальными, и каждый выполняет лишь одно действие: один только принимает заказы, следующий только приносит блюда, а другой убирает стол и забирает посуду.

Повар – персонал, готовящий блюда. Должен хорошо разбираться в продуктах в зависимости от тематики ресторана: в ресторанах с морскими блюдами должен уметь разбираться в свежей рыбе, а также способе разделки. Хороших поваров очень трудно найти в мелких форматах. В дорогих заведениях, как и среди официантов есть своя иерархия на кухне, и они делают только одну работу. Есть также шеф-повар, который не готовит, но следит за всем персоналом кухни.

Бармен – в зависимости от формата бара имеет разные обязанности. Обычно форматов бара 2: контактный и сервисный. Самый распространенный сервисный бар, этот имеется в ресторанах, но в них бармен принимает заказ только через официантов. В контактном же баре гости напрямую обращаются к бармену, делают заказ, и также оплачивают бармену. В контактных барах бармены также являются кассирами в заведении.

## **1.2 Обоснование необходимости автоматизации объекта исследования**

Улучшение производительности различных предприятий за счет «Информационных технологий» осуществляется путем оснащения различных структурных подразделений с различными профессиональными программными продуктами.

Использование современных информационных технологий в сфере управления обеспечивает повышение качества экономической информации, ее точности, объективности и как следствие этого, возможности принятия своевременных управленческих решений.

Автоматизация деятельности кафе является необходимым и перспективным процессом. Комплексное использование современных информационных технологий позволяет значительно облегчить работу в сфере общепита.

Функциональные возможности программы должны охватывать все возможные ситуации в работе кафе. Программа должна поддерживать ведение базы продуктов, блюд, клиентов и сотрудников, а также обеспечивать правильное взаимодействие этих баз. Кроме того, необходимо обеспечение всех пользователей программы нужной им информацией в удобной форме.

Основные задачи – делать все возможное, чтобы доставить удовольствие гостям, повышать уровень персонала, перевыполнять задачи, поставленные инвесторами.

Целью этого проекта является разработка системы программного обеспечения, которая устранил необходимость традиционного подхода бумажной волокиты для частных ресторанов. Проект направлен на то, чтобы сделать ресторан полностью автоматизированным, чтобы было проще координировать различные виды деятельности, которые выполняются в типичном ресторане. Основные особенности проекта включают в себя:

- организация базы данных для ресторана среднего размера;
- координация работы различных участников - хозяин, официант, повар, автобусный менеджер и менеджер;
- повышение эффективности за счет минимизации времени между размещением заказа и выставлением счетов;
- увеличение прибыли за счет снижения эксплуатационных расходов и увеличения доходов за счет повышения эффективности;
- архивирование информации рабочих и отработанных часов;

Что дает автоматизация ресторана:

- оптимизация товарного производственного бухгалтерского учета;
- резкое снижение количества ошибок персонала;
- лучшее распределение временных ресурсов;
- оптимизация условий и качества всех выполняемых работ;
- отсутствие ошибок в работе персонала;
- отчетность по эффективности всех сотрудников.

Традиционный подход с бумажной работой имеет следующие недостатки:

- для отслеживания пустых столов хост постоянно отслеживает состояние столов, если это небольшой ресторан;
- официант оформляет заказ на бумаге и должен передать избыточную информацию в терминальную систему. Это занимает некоторое время и снижает эффективность в часы пик обслуживания клиентов, также, возможно, не хватит терминалов в ресторане, так как есть несколько столов, которые требуют, чтобы определенные официанты ждали, пока другие не закончат вводить свои заказы в системы;
- повара не могли сообщить официанту, что еда была готова;
- сохранение счетов и другой статистической информации было проблемой.

В этом дипломном проекте разрабатывается программное решение для вышеуказанных проблем, которое позволило бы упростить управление рестораном и улучшить координацию повседневной работы. Персонал будет использовать технику выполнения желаемой задачи.

### 1.3 Обзор программных и инструментальных средств

Использование инструментальных средств для проектирования и разработки продукта помогает уменьшить сроки на разработку, сократить затраты, улучшить качество, минимизировать ошибки при работе. Инструментальные средства предлагают:

- структурирование данных о проекте и его составляющих, что облегчает изменение во время работы;
- визуализация проекта, что помогает формировать проект графически в интерактивном режиме с помощью средств (диаграммы, блок-схемы, графы);
- анализирование всего проекта, помогающее при подсчете характеристик процессов;
- использование уже имеющихся данных для реализации проекта;
- автоматическая генерация компонентов системы.

При проектировании и разработке используются специальные CASE – средства. CASE – это Computer Aided System Engineering, компьютерная поддержка проектирования систем.

Все действующие CASE – средства можно разделить на группы:

*Средство управления проектом* – используется для планировки и сопровождения продукта. Основные функции: формирование графиков работы, управление ресурсами для распределения, построение диаграмм, управление расходами.

*Средства создания диаграмм* – используется для визуализации проекта на этапе проектирования. Основные функции: формирование информационной модели, анализ эффективности организации проекта.

*Средства имитационного моделирования* – средство используется на этапах визуализации. Основные функции: построение потоковых диаграмм, изменение характеристик потоков и распределения ресурсов, при этом используются анимационные эффекты для демонстрации работы модели.

*Средства создания информационных систем* – используется для разработки информационных систем. Основные функции: формирования функциональной структуры информационной системы; структурирование (моделирование) данных, в том числе: создание концептуальной структуры базы данных, автоматическая генерация физической модели БД и др.; быстрая разработка приложений (визуальное программирование).

*Интегрированные многофункциональные средства* – автоматизирует все основные этапы проекта. Основные функции: возможности имитационного моделирования, включение средств разработки приложений, поддерживают многопользовательский доступ к инструментарию.



## **2 Проектная часть**

### **2.1 Постановка задачи**

Предлагается программное решение для проблем, сопровождаемое традиционным управлением общепита, которое позволило бы упростить управление баром и улучшить координацию повседневной работы. Персонал будет использовать сенсорный экран для входа в систему и выполнения желаемой задачи.

Поддерживаемые роли сотрудников: Kassir, Администратор, Директор. Различные сотрудники имеют учетные записи пользователей и входят в систему, используя свои аккаунты, которые они должны помнить. Вход и выход будут использоваться как иницилирующие события для обновления и организации данных.

Когда человек входит в бар, хозяин приветствует клиента и предлагает пройти к стойке. Kassir предлагает клиенту меню и принимает заказ. После заказа бармен готовит заказ видя его на экране. После того, как клиент оплатил выставленный счет, заказ архивируется в базе данных для расчета доходов бара за этот день / месяц / год. Это также позволяет легко подготовить статистику, касающуюся большого количества часов обслуживания клиентов и т.д.

Для Администратор имеет административную власть над профилями сотрудников. Они могут сделать следующее:

- 1) Возможность создавать и изменять профили
- 2) Отслеживать деятельность сотрудников
- 3) Проверка статистики.

Будет учтено количество кликов, необходимых для выполнения отдельных задач, и целью будет свести к минимуму количество кликов для эффективного развертывания системы.

Далее задачами являются:

- вход по аккаунтам;
- выход из программы;
- общий просмотр данных (виджеты);
- управление пользователями;
- управление групп пользователей;
- организация точек продаж;
- управление меню;
- добавление продукции;
- управление заказами;
- просмотр статистики предприятия;
- просмотр статистики точки продажи;
- просмотр информации предприятия;
- просмотр данных аккаунта;
- изменение данных аккаунта;
- доступ в любое время;
- доступ с любого устройства.

## 2.2 Этапы создания ПП

Разработка данного программного продукта состоит в основном из нескольких последовательных этапов, определяющих каждый последующий шаг, для того чтобы учесть все возможные ошибки и недочеты, и исправить их, а также для удобного, быстрого проектирования:

Начальный этап состоит из следующих пунктов:

- согласовать функциональность с заказчиком. Утверждение пунктов структуры заявки, сбор информации;
- анализ собранной информации. Затем определитесь с инструментами разработки (редактор кода, графические редакторы, выбор хостинга);
- приобретение программного обеспечения;
- структурная презентация приложения.

Следующий шаг:

- брендинг. Для этого проекта был создан логотип, выбор фирменных цветов, определение стиля, дизайн интерфейса. Структура каждой страницы должна быть связана с предыдущей;
- если проект разрабатывается подрядчиком, необходимо согласовать его с заказчиком;
- макет (если не используются утилиты для создания шаблонов веб-страниц, готовых шаблонов или встроенных шаблонов для систем управления контентом).
- тестирование в разных браузерах. Исправление HTML и CSS кода.

Результатом работы на этапе разработки дизайна и архитектуры проекта станут эскизы всех типовых страниц сайта.

- разработка программного алгоритма (скриптовая часть кода), алгоритма взаимодействия с базой данных;
- написание программы;
- разработка административной части сайта (если система управления контентом не используется);
- тестирование и отладка приложений;
- проверка кода.

На этом этапе будет создана рабочая версия сайта, готовая к наполнению текстовым и графическим материалом.

- наполнение сайта информацией (контентом). После создания соответствующих шаблонов и текстовых механизмов файлы и необходимые иллюстрации будут составлены или перенесены в базу данных;
- проверьте в разных браузерах. При необходимости измените стили и форматирование контента;

Третий, последний этап:

- подбор оптимальной конфигурации оборудования и программного обеспечения для переноса сайта на платформу хостинга;
- перевод сайта на хостинг;

- запуск проекта. На данном этапе необходимые документы подписываются при приемке и доставке клиенту;

Обычно в разработке веб-проектов принимают участие несколько специалистов, выполняющих следующие функции:

1) Дизайнер

- разрабатывает элементы сайта и эскизы типичных страниц сайта;
- создает графические формы и элементы навигации;
- актуализирует элементы дизайна.

2) Веб-разработчик

- дизайн базы данных;
- скриптинг;
- разработка, установка и настройка интерактивных сервисов;
- разработка интерфейса и механизмов системы администрирования сайта.

3) HTML-кодировщик

- верстка и адаптация содержания текста;
- оптимизация HTML кода;
- размещение иллюстраций и графических элементов.

## 2.3 Case-проектирование ПП

### 2.3.1 Диаграмма вариантов использования

UML (Unified Modeling Language – унифицированный язык моделирования) – это язык графического описания для моделирования объектов в области разработки программного обеспечения, для моделирования бизнес-процессов, проектирования систем и отображения организационных структур.

UML – это язык на широкой основе; это открытый стандарт, который использует графические обозначения для создания абстрактной модели системы, называемой моделью UML. UML был создан для определения, визуализации, проектирования и документирования в основном программных систем. UML не является языком программирования, но код может быть сгенерирован на основе моделей UML.

Структура программного продукта универсально в предоставлении выбора функционала для пользователя:

*Выбрать вкладку.*

- 1) Виджеты
- 2) Пользователи
- 3) Группы
- 4) Точки продаж
- 5) Таблицы
- 6) Категории
- 7) Продукция
- 8) Заказы
- 9) Уведомления
- 10) Компания
- 11) Профайл
- 12) Настройки
- 13) Выйти

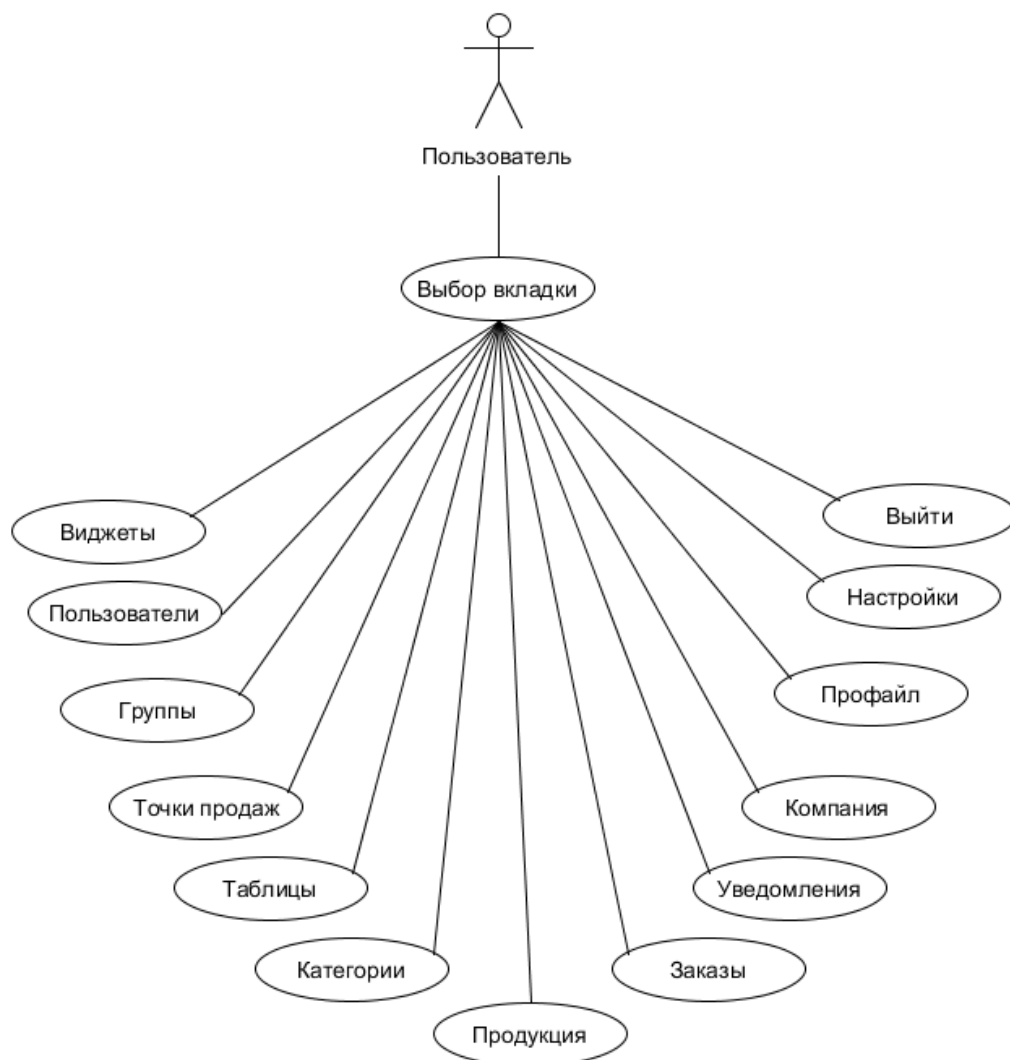


Рисунок 2.3.1 – Диаграмма вариантов использования для действий со стороны пользователя

Как видно из диаграммы (Рис. 2.3.1), пользователю доступны все вкладки, а также выход из приложения. Но эти множества выбора объясняется тем, что администратор может давать каждому пользователю индивидуальные разрешения на тот или другой раздел. Это учтено, для того чтобы можно было давать разные права разным сотрудникам, при том что сотрудники могут быть с одной группе пользователей.

Как видно из рис. 2.3.2, при взаимодействии пользователя с приложением, он получает свой ответ от сервера.

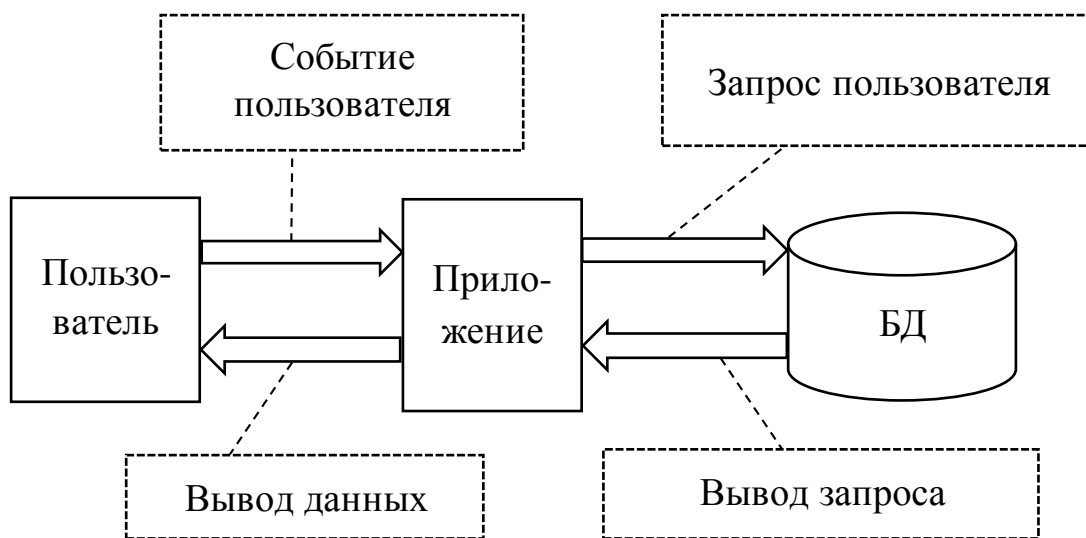


Рисунок 2.3.2 – Функциональная модель использования

### 2.3.2 Диаграмма последовательности

Диаграммы последовательностей, вероятно, являются наиболее важными диаграммами UML не только среди компьютерного сообщества, но и в качестве моделей уровня проектирования для разработки бизнес-приложений. Они стали популярны в изображении бизнес-процессов, из-за их визуальной самоочевидной природы.

Как следует из названия, диаграммы последовательности описывают последовательность сообщений и взаимодействий, которые происходят между субъектами и объектами. Актеры или объекты могут быть активны только тогда, когда это необходимо или, когда другой объект хочет общаться с ними. Все сообщения представлены в хронологическом порядке.

Как следует из названия, структурные диаграммы используются для отображения структуры системы. Более конкретно, он используется в разработке программного обеспечения для представления архитектуры системы и того, как различные компоненты взаимосвязаны.

На рисунке 2.3.2 представлена диаграмма с актером «Пользователь», отображающая последовательность действий пользователя при входе в программу и дальнейшие действия.

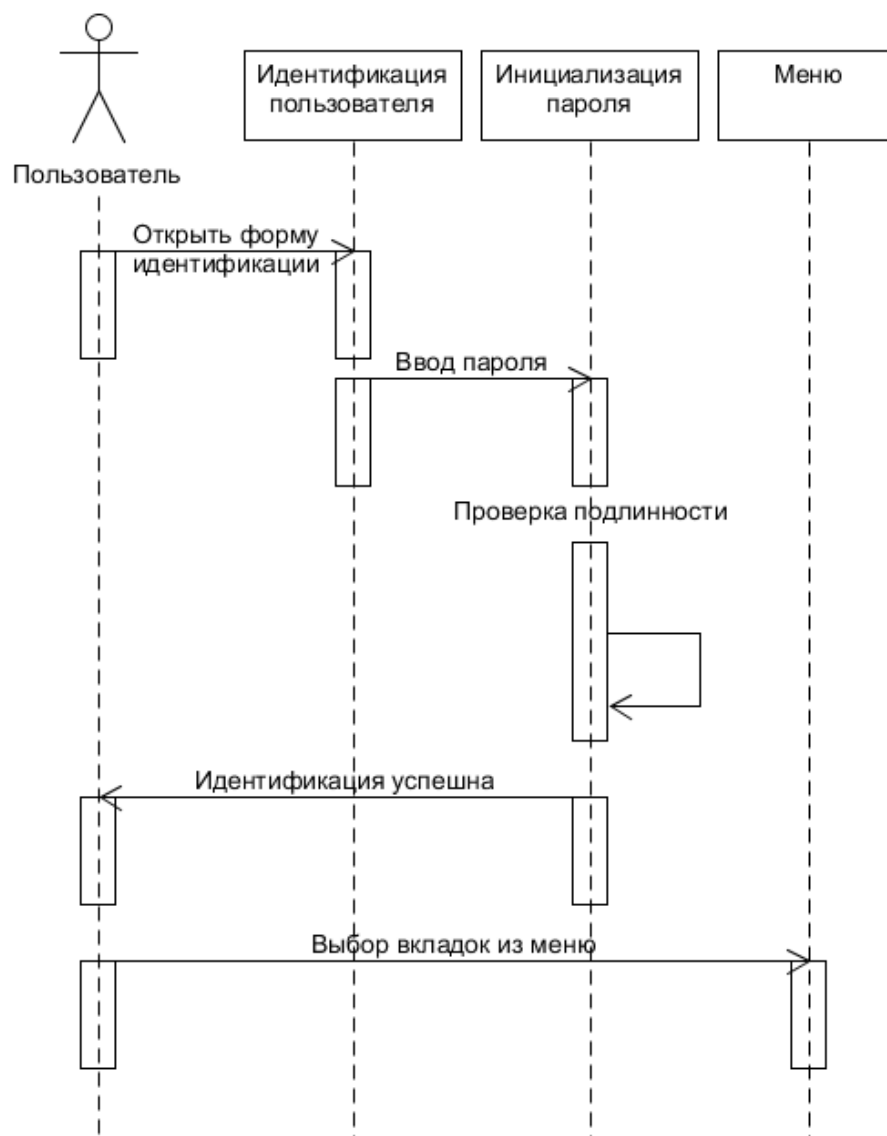


Рисунок 2.3.3 – Диаграмма последовательности с актером «Пользователь»



## 2.4 Выбор средств для разработки

Для обеспечения функциональности почти на любом устройстве будет разрабатываться веб-приложение на PHP, HTML.

HTML – является языком кодирования, который поддерживает большинство веб-страниц в Интернете.

CMS (content management system) – это программная платформа, работающая на базе данных и предоставляющая пользователям простой в использовании интерфейс для создания и поддержки веб-сайта.

HTML является очень мощным и эффективным языком кодирования, который относительно легко выучить. Однако, хотя новичок может создать простой веб-сайт за день или два, результат будет очень простым и понятным. Тем, кто хочет создать профессионально выглядящий веб-сайт для малого бизнеса с высоким уровнем функциональности, потребуется нанять опытного веб-разработчика или дизайнера.

Однако, используя CMS, владельцы веб-сайтов, не имеющие опыта программирования, могут создавать, проектировать и поддерживать свой собственный веб-сайт. Это может быть идеально для новых начинающих компаний, которые хотят сэкономить, создавая собственные сайты для малого бизнеса без необходимости нанимать дорогого веб-разработчика.

Есть также варианты использования специальных фреймворков, с конструкцией MVC.

## 2.5 Разработка базы данных

При запуске нового проекта корпоративной базы данных, одним из наиболее важных шагов является выбор правильной базы данных. С появлением больших данных появилось гораздо больше возможностей для управления данными. Выбор правильной базы данных будет означать следующее:

- прежде всего, нужно понимание, как база данных будет использоваться в соответствии с требованиями проекта;
- с одним типом базы данных, будет выполняться только некоторые из потребностей базы данных;
- производительность достигается только после того, как были успешно сопоставлены все потребности создаваемой базы данных с нужным видом базы данных;
- всегда существует компромисс между последовательностью, доступностью и допустимостью разбиения.

Нужно понимание компромисса. Причина, по которой у нас сегодня много вариантов базы данных, связана с теоремой CAP. CAP означает согласованность, доступность и устойчивость к разделению (consistency, availability, partition tolerance).

- согласованность означает, что любой запрос на чтение вернет самую последнюю запись;
- доступность означает, что не отвечающий узел должен ответить в течение разумного периода времени;
- устойчивость к разделению означает, что система продолжит работать, несмотря на сбои сети или узла.

В любой момент времени только два из этих трех требований могут быть выполнены одновременно.

Реляционные базы данных традиционно отличаются высокой согласованностью и высокой доступностью за счет допусков на разделы. Примеры: SQL Server, MySQL, база данных Oracle, PostgreSQL, IBM DB2

Нереляционные базы данных были разработаны для удовлетворения потребностей доступности и устойчивости к разделению или согласованности и устойчивости к разделению. Примеры: Memcached, Redis, Coherence, Hbase, BigTable, Accumulo, MongoDB, CouchDB

Для сложных систем, которые интенсивно читают и пишут, может быть важно иметь комбинацию реляционных и нереляционных баз данных, чтобы разделить задачи чтения и записи для оптимизации CAP.

Следующим шагом при выборе базы данных является список пунктов, которые необходимо учесть в соответствии с требованиями создаваемой базы данных бизнеса:

- сколько связей в данных;
- каков уровень сложности данных;
- как часто меняются данные;

- как часто приложение запрашивает данные;
- как часто приложение запрашивает отношения, лежащие в основе данных
- как часто пользователи обновляют данные;
- как часто пользователи обновляют логику в данных;
- насколько критично будет вести себя приложение в случае непредвиденных обстоятельств.

Также нужно понимание преимуществ и недостатков того или иного вида базы данных. Реляционные базы данных оптимизированы для записи. Они оптимизированы для согласованности и доступности.

Преимущества реляционных баз данных включают простоту, простоту извлечения данных, целостность данных и гибкость.

Недостатки реляционных баз данных включают в себя:

- дороговизну - дорого устанавливать и поддерживать базу данных;
- структурированные ограничения - реляционные базы данных имеют ограничения по длине поля. Это может быть неудобно для хранения большого количества информации в одном поле;
- изоляция - несколько реляционных баз данных могут легко стать «островками информации». Может быть трудно соединить базы данных, где они могут общаться друг с другом.

Нереляционные базы данных оптимизированы для чтения. Они удовлетворяют требованиям доступности и допустимости разбиения или согласованности и допустимости разбиения.

Преимущества нереляционных баз данных включают в себя:

- гибкость - хранение больших объемов структурированных, полуструктурированных и неструктурированных данных;
- гибкое программирование - может приспособить быстрые итерации спринтов и нажатий кода;
- недорогая масштабируемость - может эффективно масштабировать архитектуру без дорогостоящих накладных расходов.

Недостатки нереляционных баз данных включают в себя:

- согласованность данных - нереляционные базы данных не выполняют транзакции ACID (Atomicity, Consistency, Isolation, Durability). Вместо этого они полагаются на «возможную последовательность». Преимущества производительности этих баз данных означают потерю согласованности;
- стандартизация. Не существует специального интерфейса программирования для разных баз данных. Каждый из них отличается в языке запросов с другим;

- масштабируемость. Не все нереляционные базы данных хороши для автоматизации процесса разделения или распределения базы данных по нескольким узлам. Это создает ограничения для возможности увеличения или уменьшения в зависимости от колебания спроса.

В наши дни существуют разные виды нереляционных баз данных. Они попадают в конкретные категории. Каждая категория нереляционной базы данных служит определенной цели.

Ключ-значение — эти базы данных лучше всего работают с простой схемой базы данных. Оно лучше всего подходит для чтения, записи и немногих обновлений. Он работает лучше всего, когда нет сложных запросов или бизнес-логики. Примеры: Redis, Dynamo DB и Cosmos DB.

Документоориентированная СУБД — эти базы данных работают лучше всего, если вам нужна гибкая схема. Данные хранятся в формате XML или JSON. Вы можете жить с высокой производительностью чтения и балансировать производительность чтения с производительностью записи. Вы можете использовать индексы, чтобы максимизировать свою производительность с этими базами данных. Примеры: MongoDB, DynamoDB и Couchbase.

Графовая СУБД — Эти базы данных отлично подходят, когда у вас есть сложная схема базы данных. Вам необходимо часто отображать бизнес-логику между узлами. Граф Базы данных позволят вам перемещаться между узлами. Примеры: Neo4j, Cosmos Db и Amazon Neptune.

В данном проекте был выбран MySQL. База данных разработана для работы на веб сервере. База данных будет продемонстрирована в phpMyAdmin. Были учтены все функциональные зависимости приложения. В базе данных «restaurant» содержатся следующие таблицы:

- category;
- company;
- groups;
- orders;
- order\_items;
- products;
- stores;
- tables;
- users;
- user\_group.

Вид структуры показан на рисунке 2.5.1:

Сервер: 127.0.0.1 > База данных: restaurant

Структура SQL Поиск Запрос по шаблону Экспорт Импорт Операции Привилегии Ещё

Фильтры

Содержит слово:

Таблица	Действие	Строки	Тип	Сравнение	Размер	Фр
<input type="checkbox"/> category	Обзор Структура Поиск Вставить Очистить Удалить	7	InnoDB	utf8_general_ci	16.0 Киб	
<input type="checkbox"/> company	Обзор Структура Поиск Вставить Очистить Удалить	1	InnoDB	utf8_general_ci	16.0 Киб	
<input type="checkbox"/> groups	Обзор Структура Поиск Вставить Очистить Удалить	3	InnoDB	utf8_general_ci	16.0 Киб	
<input type="checkbox"/> orders	Обзор Структура Поиск Вставить Очистить Удалить	0	InnoDB	utf8_general_ci	16.0 Киб	
<input type="checkbox"/> order_items	Обзор Структура Поиск Вставить Очистить Удалить	0	InnoDB	utf8_general_ci	16.0 Киб	
<input type="checkbox"/> products	Обзор Структура Поиск Вставить Очистить Удалить	31	InnoDB	utf8_general_ci	16.0 Киб	
<input type="checkbox"/> stores	Обзор Структура Поиск Вставить Очистить Удалить	1	InnoDB	utf8_general_ci	16.0 Киб	
<input type="checkbox"/> tables	Обзор Структура Поиск Вставить Очистить Удалить	1	InnoDB	utf8_general_ci	16.0 Киб	
<input type="checkbox"/> users	Обзор Структура Поиск Вставить Очистить Удалить	2	InnoDB	utf8_general_ci	16.0 Киб	
<input type="checkbox"/> user_group	Обзор Структура Поиск Вставить Очистить Удалить	2	InnoDB	utf8_general_ci	16.0 Киб	
10 таблиц	Всего	48	InnoDB	utf8mb4_general_ci	160.0 Киб	

Рисунок 2.5.1 – Структура базы данных «restaurant»

Структура таблицы category:

- 1) id;
- 2) name;
- 3) active.

Сервер: 127.0.0.1 > База данных: restaurant > Таблица: category

Обзор Структура SQL Поиск Вставить Экспорт Импорт Привилегии Операции

Структура таблицы Связи

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/> 1	id	int(11)			Нет	Нет		AUTO_INCREMENT	Изменить Удалить Ещё
<input type="checkbox"/> 2	name	varchar(255)	utf8_general_ci		Нет	Нет			Изменить Удалить Ещё
<input type="checkbox"/> 3	active	int(11)			Нет	Нет			Изменить Удалить Ещё

☐ Отметить все С отмеченными: Обзор Изменить Удалить Первичный Уникальный Индекс

Полнотекстовый Добавить к центральному столбцам Удалить из центральных столбцов

Печать Анализ структуры таблицы Отслеживать таблицу Переместить поля Нормировать

Добавить 1 поле(я) после active Вперёд

Рисунок 2.5.2 – Вид таблицы category

Структура таблицы company:

- 1) id;
- 2) company\_name;
- 3) service\_chage\_value;
- 4) vat\_charge\_value;
- 1) address;
- 2) phone;

- 3) country;
- 4) message;
- 5) currency.

Сервер: 127.0.0.1 » База данных: restaurant » Таблица: company

Обзор Структура SQL Поиск Вставить Экспорт Импорт Привилегии Операции Ещё

Структура таблицы Связи

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 id	int(11)			Нет	Нет		AUTO_INCREMENT	Изменить Удалить Ещё
<input type="checkbox"/>	2 company_name	varchar(255)	utf8_general_ci		Нет	Нет			Изменить Удалить Ещё
<input type="checkbox"/>	3 service_charge_value	varchar(255)	utf8_general_ci		Нет	Нет			Изменить Удалить Ещё
<input type="checkbox"/>	4 vat_charge_value	varchar(255)	utf8_general_ci		Нет	Нет			Изменить Удалить Ещё
<input type="checkbox"/>	5 address	varchar(255)	utf8_general_ci		Нет	Нет			Изменить Удалить Ещё
<input type="checkbox"/>	6 phone	varchar(255)	utf8_general_ci		Нет	Нет			Изменить Удалить Ещё
<input type="checkbox"/>	7 country	varchar(255)	utf8_general_ci		Нет	Нет			Изменить Удалить Ещё
<input type="checkbox"/>	8 message	text	utf8_general_ci		Нет	Нет			Изменить Удалить Ещё
<input type="checkbox"/>	9 currency	varchar(255)	utf8_general_ci		Нет	Нет			Изменить Удалить Ещё

Отметить все С отмеченными: Обзор Изменить Удалить Первичный Уникальный Индекс

Полнотекстовый Добавить к центральным столбцам Удалить из центральных столбцов

Печать Анализ структуры таблицы Отслеживать таблицу Переместить поля Нормировать

Добавить 1 поле(я) после currency Вперёд

Рисунок 2.5.3 – Вид таблицы company

Структура таблицы groups:

- 1) id;
- 2) group\_name;
- 3) permission.

Сервер: 127.0.0.1 » База данных: restaurant » Таблица: groups

Обзор Структура SQL Поиск Вставить Экспорт Импорт Привилегии Операции Ещё

Структура таблицы Связи

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 id	int(11)			Нет	Нет		AUTO_INCREMENT	Изменить Удалить Ещё
<input type="checkbox"/>	2 group_name	varchar(255)	utf8_general_ci		Нет	Нет			Изменить Удалить Ещё
<input type="checkbox"/>	3 permission	text	utf8_general_ci		Нет	Нет			Изменить Удалить Ещё

Отметить все С отмеченными: Обзор Изменить Удалить Первичный Уникальный Индекс

Добавить к центральным столбцам Удалить из центральных столбцов

Печать Анализ структуры таблицы Отслеживать таблицу Переместить поля Нормировать

Добавить 1 поле(я) после permission Вперёд

Индексы

Рисунок 2.5.4 – Вид таблицы groups

Структура таблицы orders:

- 1) id;
- 2) bill\_no;
- 3) date\_time;
- 4) gross\_amount;
- 5) service\_charge\_rate;
- 6) service\_charge\_amount;
- 7) vat\_charge\_rate;
- 8) vat\_charge\_amount;
- 9) discount;
- 10) net\_amount;
- 11) user\_id;
- 12) table\_id;
- 13) paid\_status;
- 14) store\_id;

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
1	id	int(11)			Нет	Нет		AUTO_INCREMENT	Изменить Удалить Ещё
2	bill_no	varchar(255)	utf8_general_ci		Нет	Нет			Изменить Удалить Ещё
3	date_time	varchar(255)	utf8_general_ci		Нет	Нет			Изменить Удалить Ещё
4	gross_amount	varchar(255)	utf8_general_ci		Нет	Нет			Изменить Удалить Ещё
5	service_charge_rate	varchar(255)	utf8_general_ci		Нет	Нет			Изменить Удалить Ещё
6	service_charge_amount	varchar(255)	utf8_general_ci		Нет	Нет			Изменить Удалить Ещё
7	vat_charge_rate	varchar(255)	utf8_general_ci		Нет	Нет			Изменить Удалить Ещё
8	vat_charge_amount	varchar(255)	utf8_general_ci		Нет	Нет			Изменить Удалить Ещё
9	discount	varchar(255)	utf8_general_ci		Нет	Нет			Изменить Удалить Ещё
10	net_amount	varchar(255)	utf8_general_ci		Нет	Нет			Изменить Удалить Ещё
11	user_id	int(11)			Нет	Нет			Изменить Удалить Ещё
12	table_id	int(11)			Нет	Нет			Изменить Удалить Ещё
13	paid_status	int(11)			Нет	Нет			Изменить Удалить Ещё
14	store_id	int(11)			Нет	Нет			Изменить Удалить Ещё

Отметить все С отмеченными: Обзор Изменить Удалить Первичный Уникальный Индекс

Полнотекстовый Добавить к центральным столбцам Удалить из центральных столбцов

Печать Анализ структуры таблицы Отслеживать таблицу Переместить поля Нормировать

Добавить 1 поле(я) после store\_id Вперёд

Рисунок 2.5.5 – Вид таблицы orders

Структура таблицы order\_items:

- 1) id;

- 2) order\_id;
- 3) product\_id;
- 4) qty;
- 5) rate;
- 6) amount;

Сервер: 127.0.0.1 » База данных: restaurant » Таблица: order\_items

Обзор Структура SQL Поиск Вставить Экспорт Импорт Привилегии Операции Ещё

Структура таблицы Связи

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 id	int(11)			Нет	Нет		AUTO_INCREMENT	Изменить Удалить Ещё
<input type="checkbox"/>	2 order_id	int(11)			Нет	Нет			Изменить Удалить Ещё
<input type="checkbox"/>	3 product_id	int(11)			Нет	Нет			Изменить Удалить Ещё
<input type="checkbox"/>	4 qty	varchar(255)	utf8_general_ci		Нет	Нет			Изменить Удалить Ещё
<input type="checkbox"/>	5 rate	varchar(255)	utf8_general_ci		Нет	Нет			Изменить Удалить Ещё
<input type="checkbox"/>	6 amount	varchar(255)	utf8_general_ci		Нет	Нет			Изменить Удалить Ещё

↑ ☐ Отметить все С отмеченными: Обзор Изменить Удалить Первичный Уникальный Индекс

Полнотекстовый Добавить к центральным столбцам Удалить из центральных столбцов

Печать Анализ структуры таблицы Отслеживать таблицу Переместить поля Нормировать

Добавить 1 поле(я) после amount Вперёд

Рисунок 2.5.6 – Вид таблицы order\_items

Структура таблицы products:

- 1) id;
- 2) category\_id;
- 3) store\_id;
- 4) name;
- 5) price;
- 6) description;
- 7) image;
- 8) active.



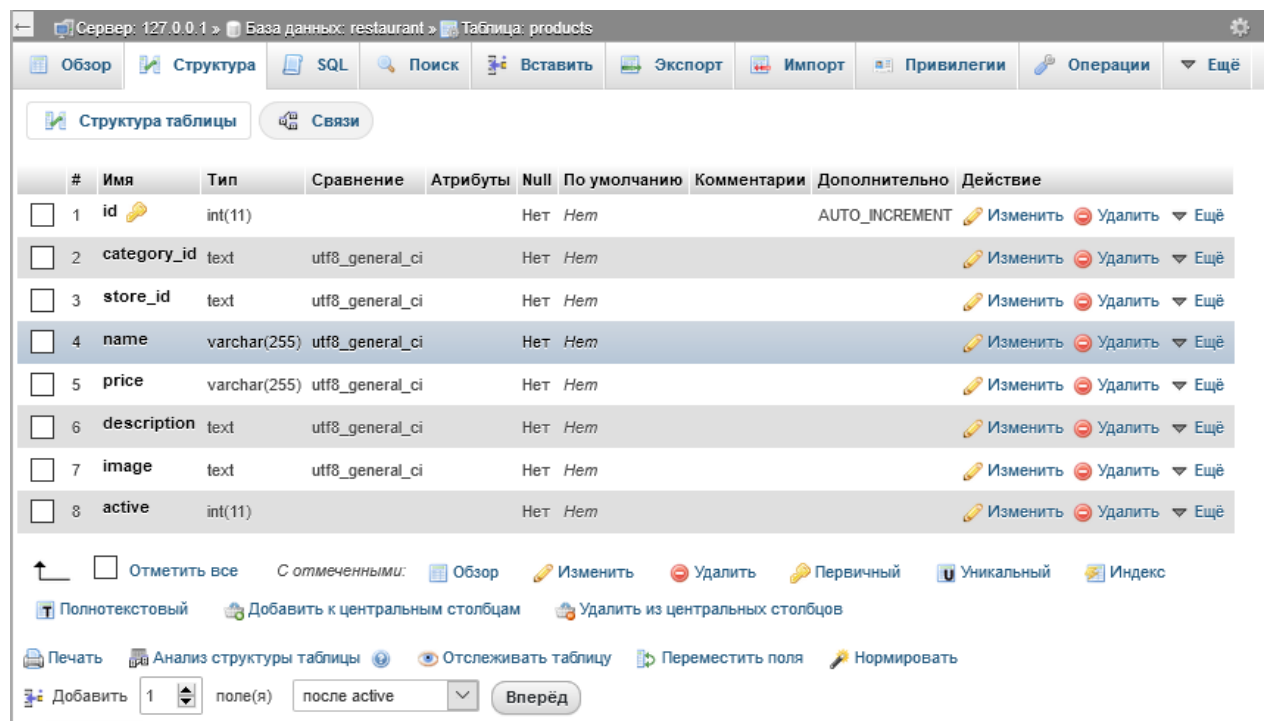


Рисунок 2.5.7 – Вид таблицы products

Структура таблицы stores:

- 1) id;
- 2) name;
- 3) active.

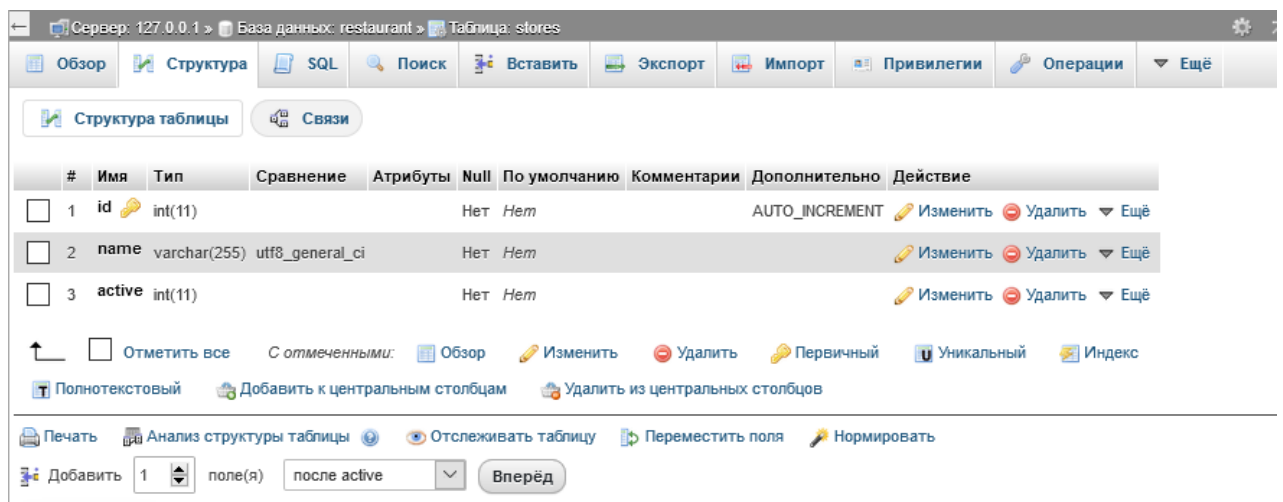


Рисунок 2.5.8 – Вид таблицы stores

Структура таблицы tables:

1. id;
2. table\_name;
3. capacity;

4. available;
5. active;
6. store\_id.

Сервер: 127.0.0.1 » База данных: restaurant » Таблица: tables

Обзор Структура SQL Поиск Вставить Экспорт Импорт Привилегии Операции Ещё

Структура таблицы Связи

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 id	int(11)			Нет	Нет		AUTO_INCREMENT	Изменить Удалить Ещё
<input type="checkbox"/>	2 table_name	varchar(255)	utf8_general_ci		Нет	Нет			Изменить Удалить Ещё
<input type="checkbox"/>	3 capacity	varchar(255)	utf8_general_ci		Нет	Нет			Изменить Удалить Ещё
<input type="checkbox"/>	4 available	int(11)			Нет	Нет			Изменить Удалить Ещё
<input type="checkbox"/>	5 active	int(11)			Нет	Нет			Изменить Удалить Ещё
<input type="checkbox"/>	6 store_id	int(11)			Нет	Нет			Изменить Удалить Ещё

☐ Отметить все С отмеченными: Обзор Изменить Удалить Первичный Уникальный Индекс  
 Полнотекстовый Добавить к центральным столбцам Удалить из центральных столбцов

Печать Анализ структуры таблицы Отслеживать таблицу Переместить поля Нормировать  
 Добавить 1 поле(я) после store\_id Вперёд

Рисунок 2.5.9 – Вид таблицы tables

Структура таблицы users:

- 1) id;
- 2) username;
- 3) password;
- 4) email;
- 5) firstname;
- 6) lastname;
- 7) phone;
- 8) gender;
- 9) store\_id.

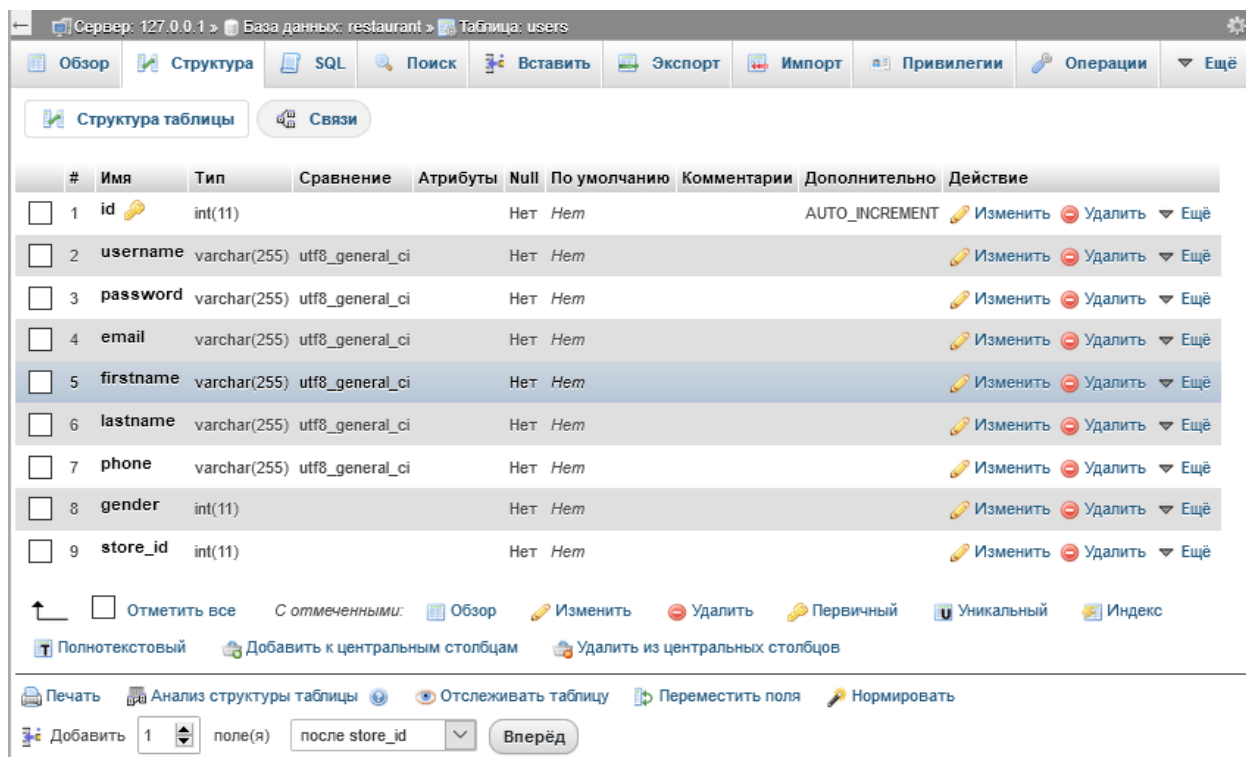


Рисунок 2.5.10 – Вид таблицы users

Структура таблицы user\_group:

- 1) id;
- 2) user\_id;
- 3) group\_id.

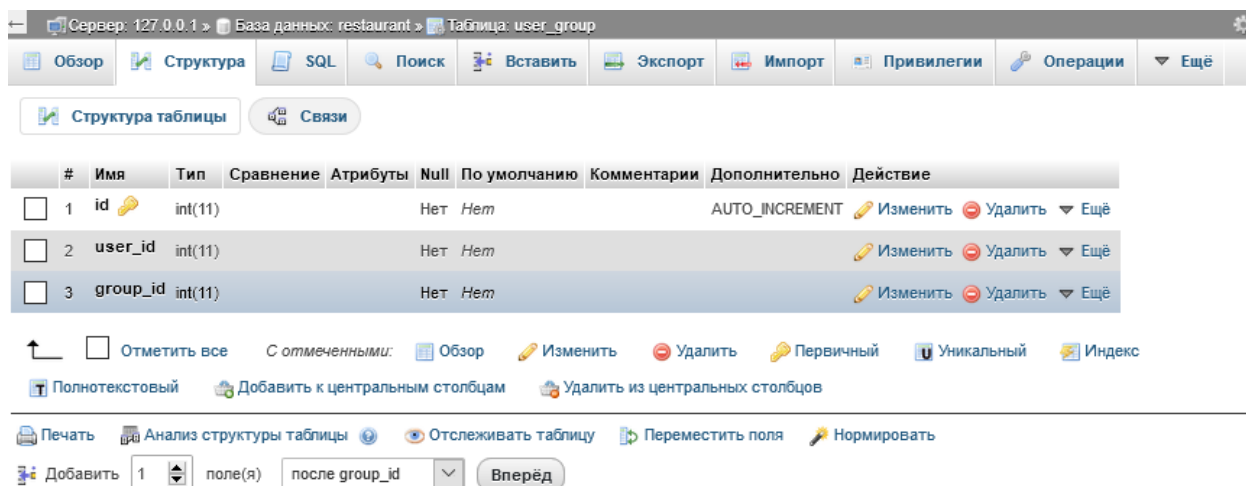


Рисунок 2.5.11 – Вид таблицы user\_group

## 2.6 Структура программного продукта

Разработка программного продукта начинается с его структуры. Конструкция ПО должна быть удобна для выполнения задач, поставленных перед разработчиком.

Модель конструкции Model-View-Control (MVC), первоначально сформулированный в конце 1970-х годов, представляет собой паттерн архитектуры программного обеспечения, построенный на основе разделения представления данных от методов, взаимодействующих с данными. Теоретически, хорошо разработанная система MVC должна позволять внешнему разработчику и внутреннему разработчику работать в одной и той же системе без вмешательства, совместного использования или редактирования файлов, над которыми работает любая сторона.

Несмотря на то, что MVC изначально был разработан для персональных компьютеров, он был адаптирован и широко используется веб-разработчиками из-за его акцента на разделение интересов и, следовательно, косвенного повторного использования кода. Шаблон поощряет разработку модульных систем, позволяя разработчикам быстро обновлять, добавлять или даже удалять функциональные возможности.

Название паттерна представляет собой совокупность трех его основных частей: Модель, Вид и Контроллер. Визуальное представление полного и правильного паттерна MVC выглядит следующим образом:

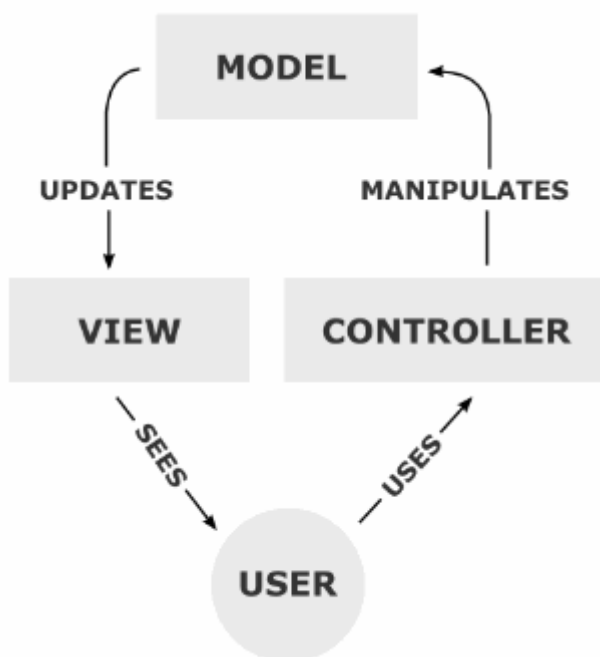


Рисунок 2.6.1 – Паттерн MVC

Model – это имя, данное постоянному хранилищу данных, используемых в общем проекте. Он должен предоставлять доступ для просмотра, сбора,

записи данных и является мостом между компонентом View и компонентом Controller в общем паттерне.

В компоненте View просматриваются данные, запрошенные у модели, и определяется их окончательный вывод. Традиционно в веб-приложениях, созданных с использованием MVC, представление является частью системы, где генерируется и отображается HTML-код. Представление также запускает реакции пользователя, который затем взаимодействует с контроллером. Основным примером этого является кнопка, сгенерированная представлением, которое пользователь щелкает и запускает действие в контроллере.

Последний компонент триады – Controller. Его задача – обрабатывать данные, которые пользователь вводит или отправляет, и соответственно обновлять модель. Двигатель Controller-a – это пользователь; без взаимодействия с пользователем контроллер не имеет цели. Это единственная часть паттерна, с которой пользователь должен взаимодействовать.

Для использования паттерна MVC был выбран фреймворк CodeIgniter. Данный фреймворк очень популярен в западных странах в отличие от постсоветского пространства. Одним из причин использовать именно этот фреймворк это его бесплатность. Данный пункт снижает стоимость конечного программного продукта.

Пользовательская структура показана на рисунке 2.6.2.



## Рисунок 2.6.2 – Пользовательская структура программного продукта

Боковое меню позволяет переключаться между другими функциями в любое время.

Меню «Виджеты» - страница, на которой общий взгляд на имеющуюся информацию.

Меню «Пользователи» - страница, на которой просматриваются имеющиеся пользователи, а также возможность добавить новых и изменить их права.

Меню «Группы» - обзор групп одинаковых пользователей, и соответствующие права для групп.

Меню «Точки продаж» - обзор и редактирование точек продаж филиала

Меню «Категории» - добавление и изменение номенклатуры

Меню «Продукция» - добавление и изменение имеющейся продукции.

Меню «Заказы» - добавление заказа и просмотр всех заказов.

Меню «Уведомления» - аналитика и отчетность.

Меню «Компания» - просмотр информации о компании и также их редактирование.

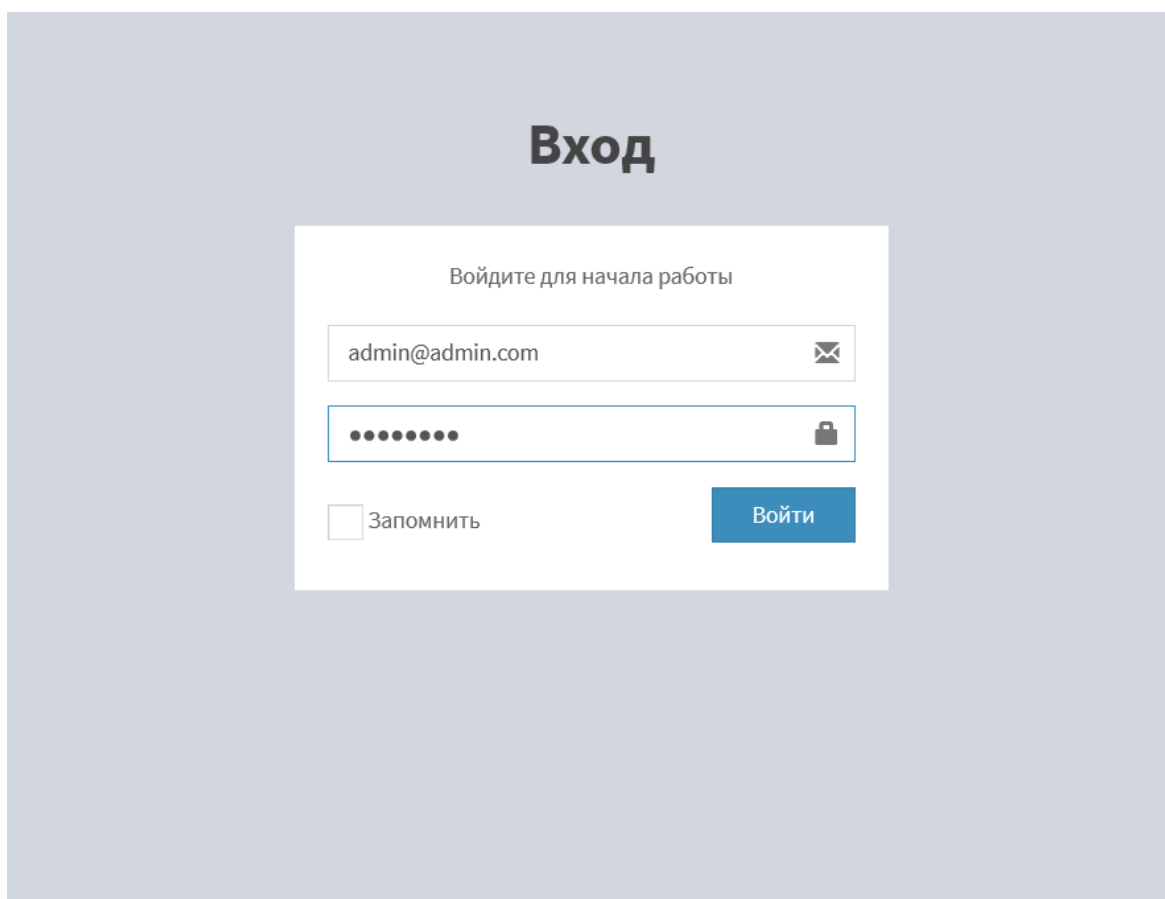
Меню «Профайл» - просмотр информации об аккаунте вошедшего в программу.

Меню «Настройки» - настройки аккаунта, вошедшего в программу.

Меню «Выход» - кнопка выхода из программы.

## 2.7 Разработка программного продукта

Вход в программу требует наличие аккаунта в базе данных. После ввода соответствующих данных инициируется вход в меню приложения.



The image shows a login interface with a light gray background. At the top center, the word "Вход" (Login) is displayed in a large, bold, dark gray font. Below it, the text "Войдите для начала работы" (Log in to get started) is centered in a smaller font. The login form consists of two input fields: the first for an email address, containing "admin@admin.com", and the second for a password, represented by eight dots. Both fields have icons on the right (an envelope for email and a padlock for password). Below the password field is a checkbox labeled "Запомнить" (Remember me). To the right of the checkbox is a blue button with the text "Войти" (Login) in white.

Рисунок 2.7.1 – Меню входа в приложение

Вид пользовательской части показан на рисунке 2.7.2. боковое меню состоит из других подменю.

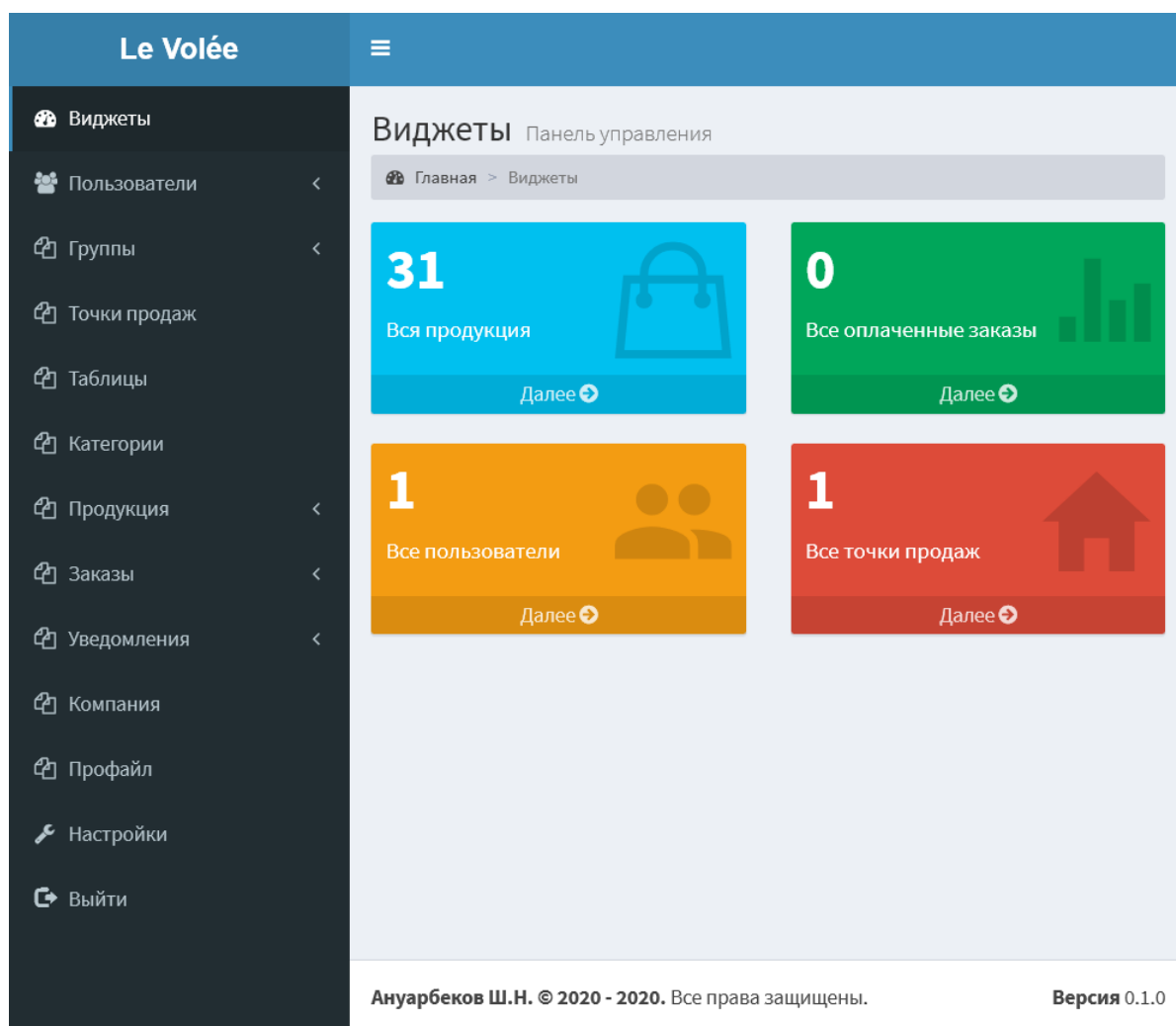


Рисунок 2.7.2 – Вид пользовательского меню «Виджеты»



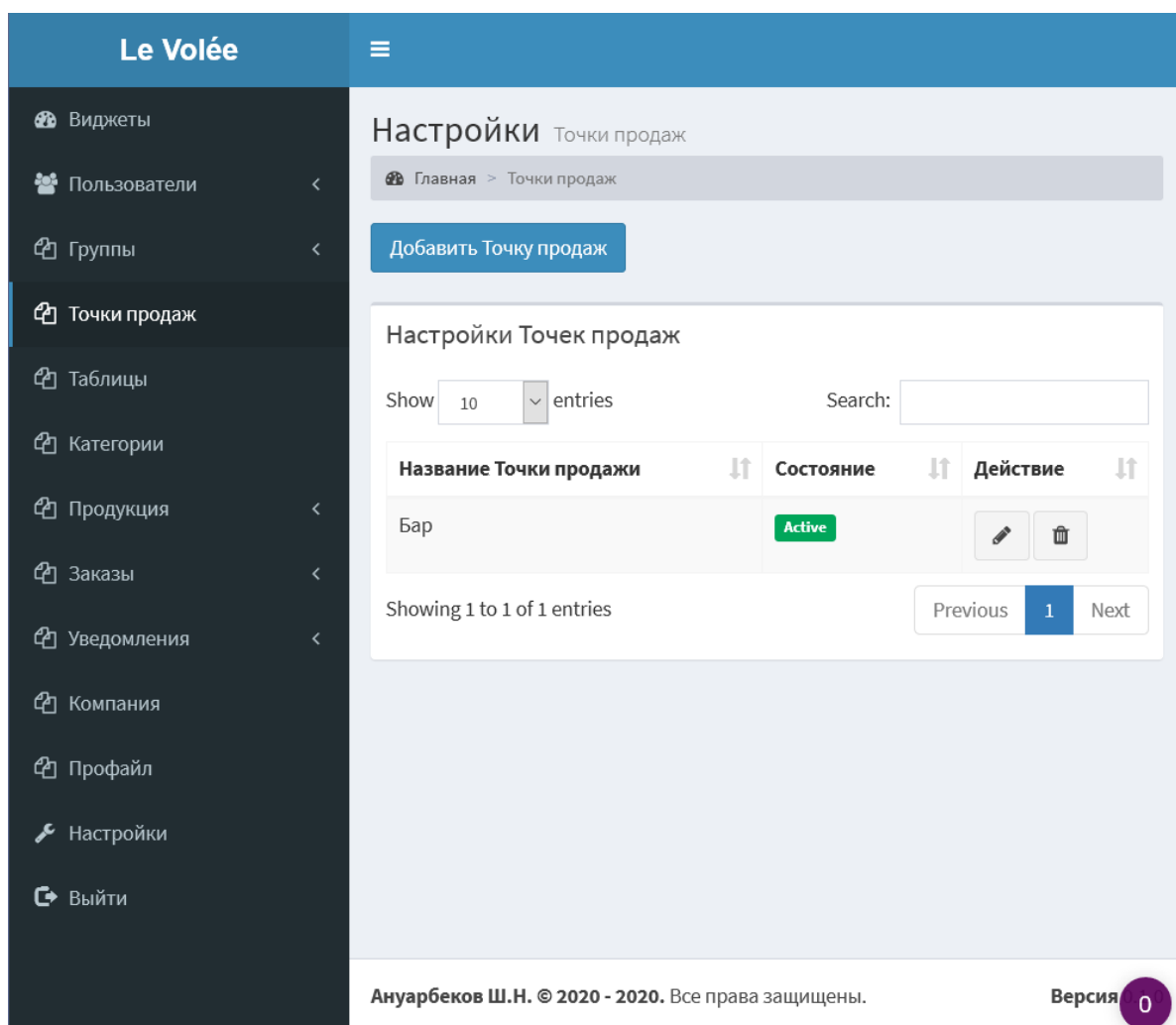


Рисунок 2.7.3 – Пункт меню «Точки продаж»

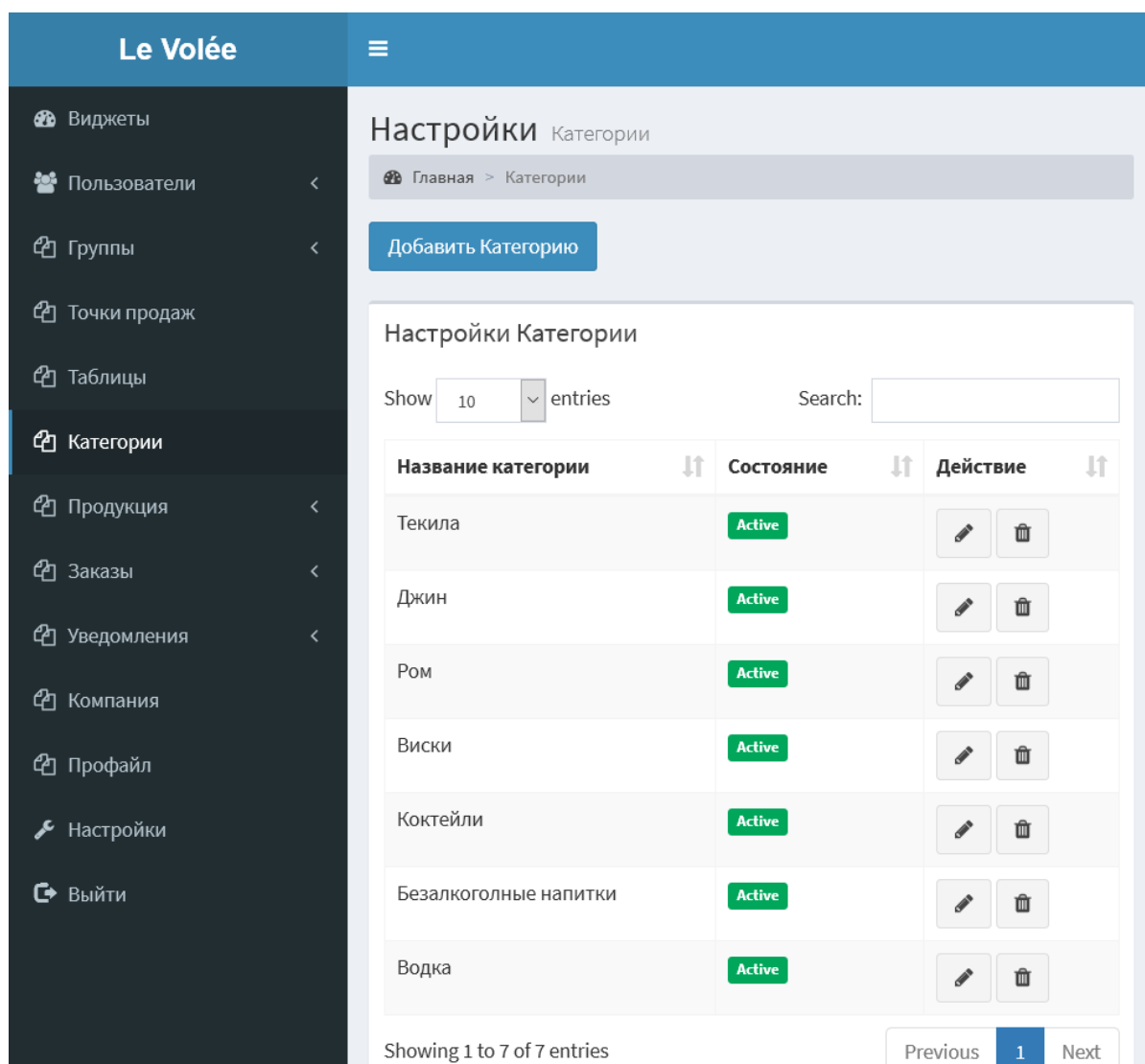


Рисунок 2.7.4 – Пункт меню «Категории»

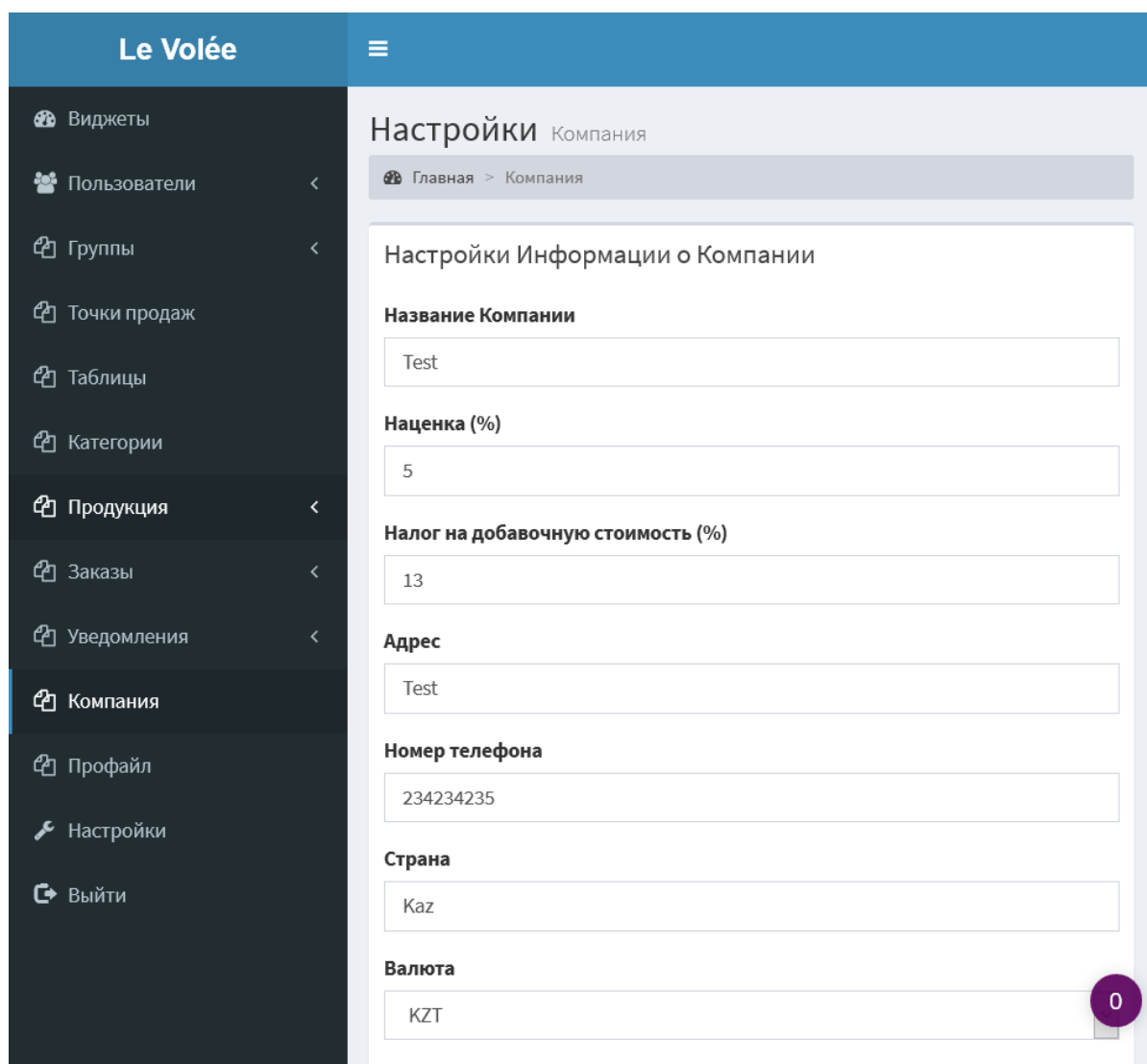


Рисунок 2.7.5 – Пункт меню «Компания»

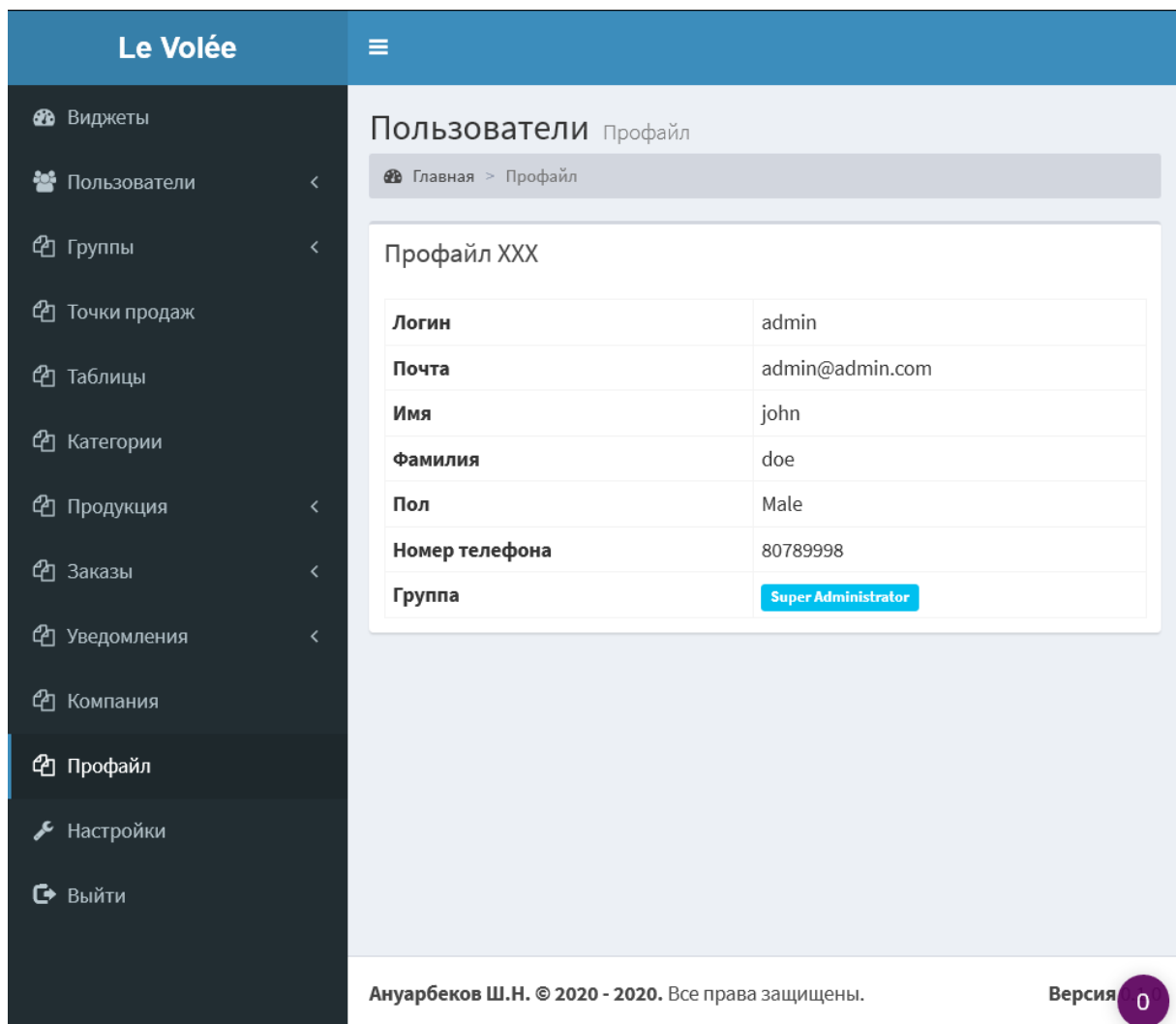


Рисунок 2.7.6 – Пункт меню «Профиль»

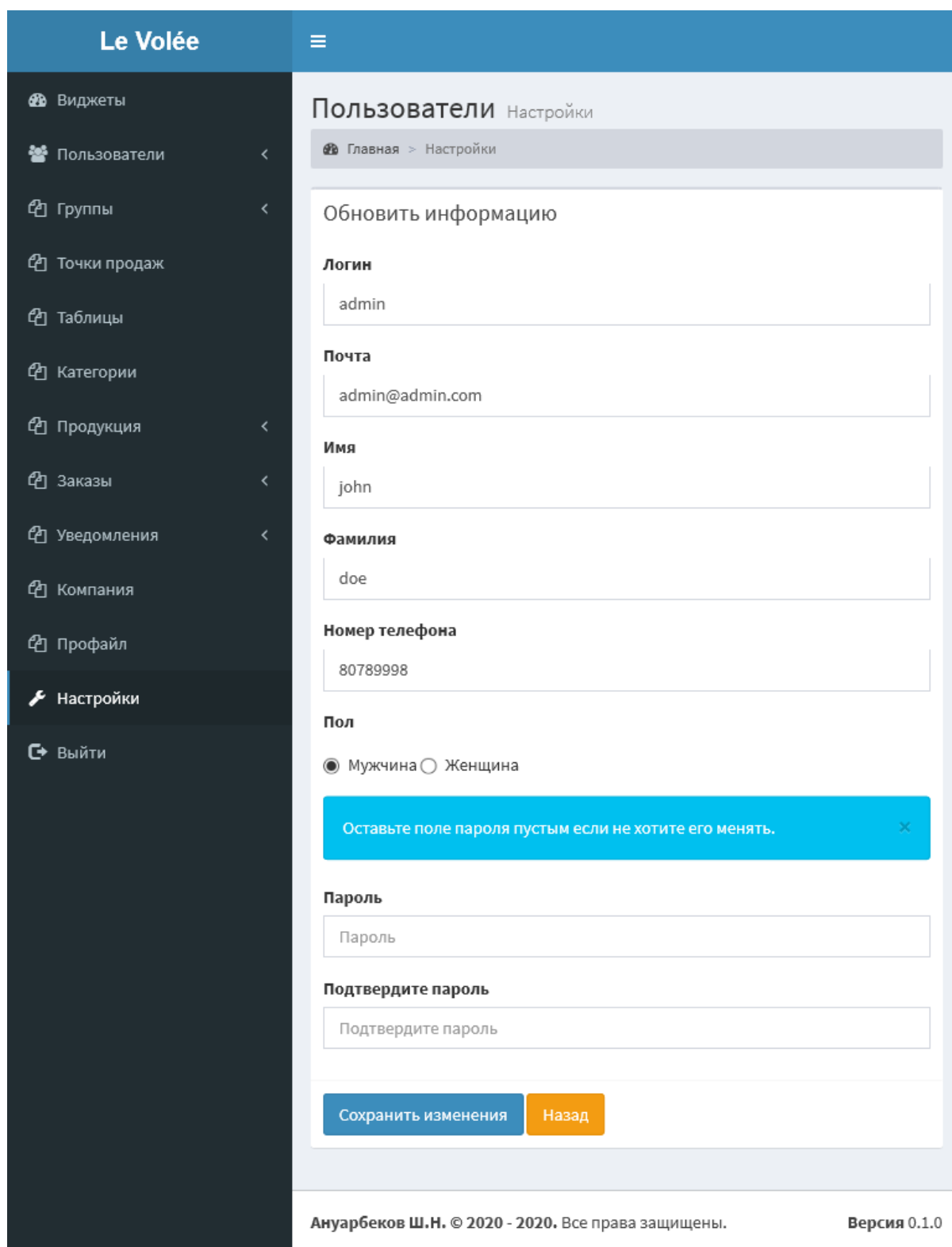


Рисунок 2.7.7 – Пункт меню «Настройки»

Также у некоторых пунктов меню, есть подменю для разных возможностей. Такие подменю есть у пунктов Пользователи, Группы, Продукция, Заказы, Уведомления.

**Le Volée**

Виджеты

**Пользователи**

Добавить пользователя

Управлять пользователями

Группы

Точки продаж

Таблицы

Категории

Продукция

Заказы

Уведомления

Компания

Профайл

Настройки

Выйти

**Настройки Пользователи**

Главная > Пользователи

**Добавить Пользователя**

**Группы**

Выбрать Группы

**Точка продаж**

Выбрать Точку продаж

**Логин**

Логин

**Почта**

Почта

**Пароль**

Пароль

**Подтвердить пароль**

Подтвердить пароль

**Имя**

Имя

**Фамилия**

Рисунок 2.7.8 – Подпункт меню «Пользователи» – «Добавить пользователя»

**Фамилия**

Фамилия

**Номер телефона**

Номер телефона

**Пол**

☐ Мужчина ☐ Женщина

**Сохранить изменения** **Назад**

Ануарбеков Ш.Н. © 2020 - 2020. Все права защищены. **Версия 0.1.0**

Рисунок 2.7.9 – Продолжение подпункта меню «Пользователи» – «Добавить пользователя»

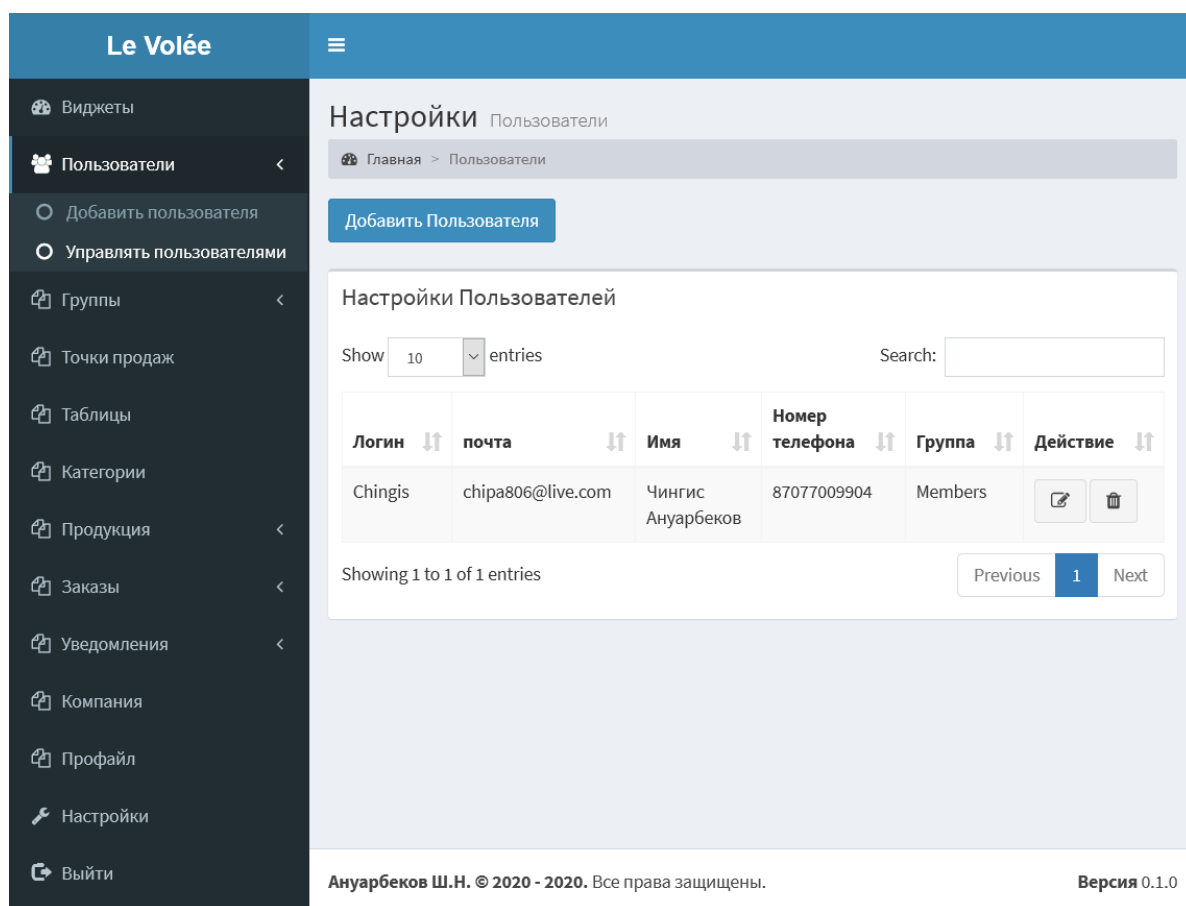


Рисунок 2.7.10 – Подпункт меню «Пользователи» – «Управлять пользователями»

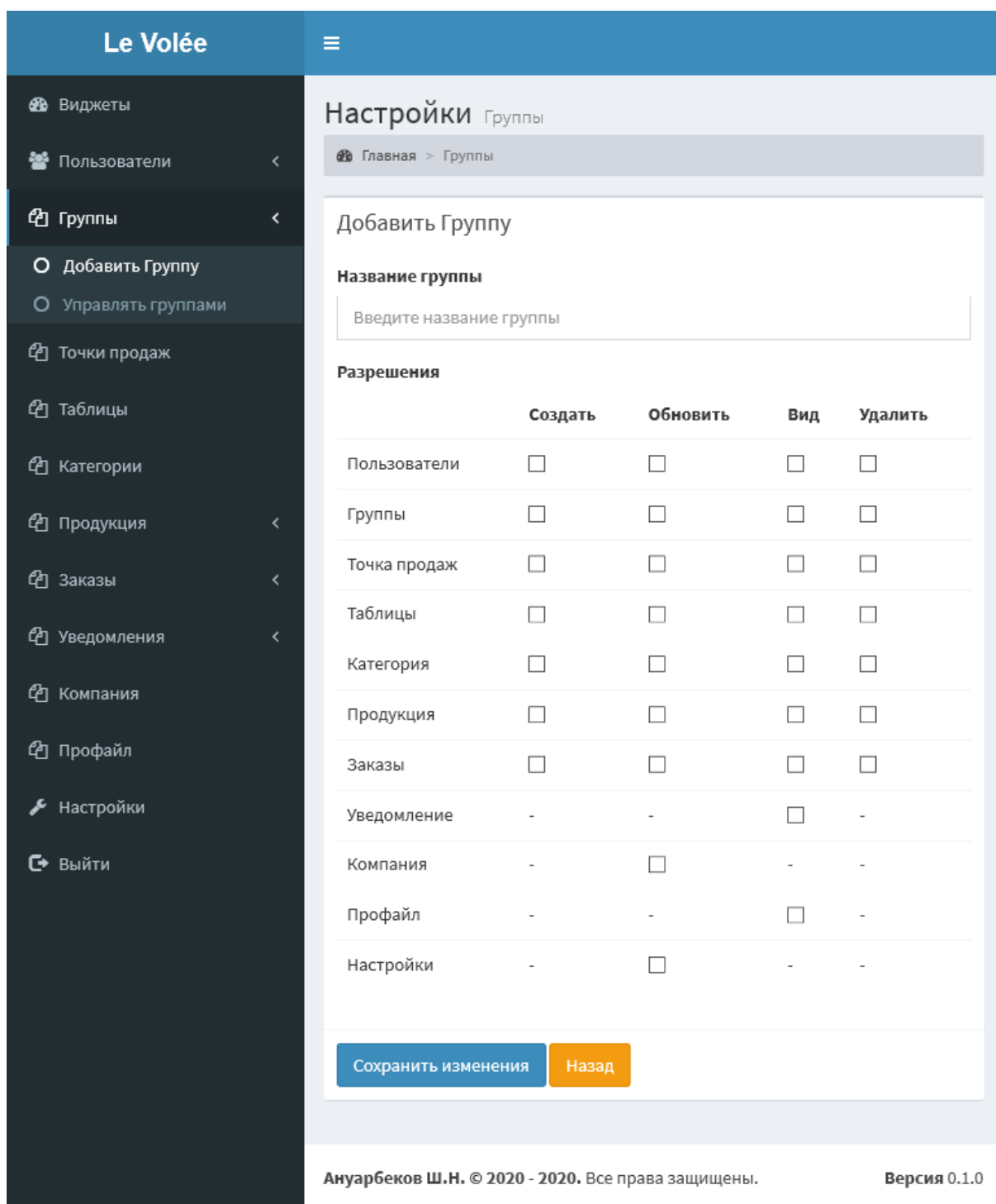


Рисунок 2.7.11 – Подпункт меню «Группы» – «Добавить Группу»



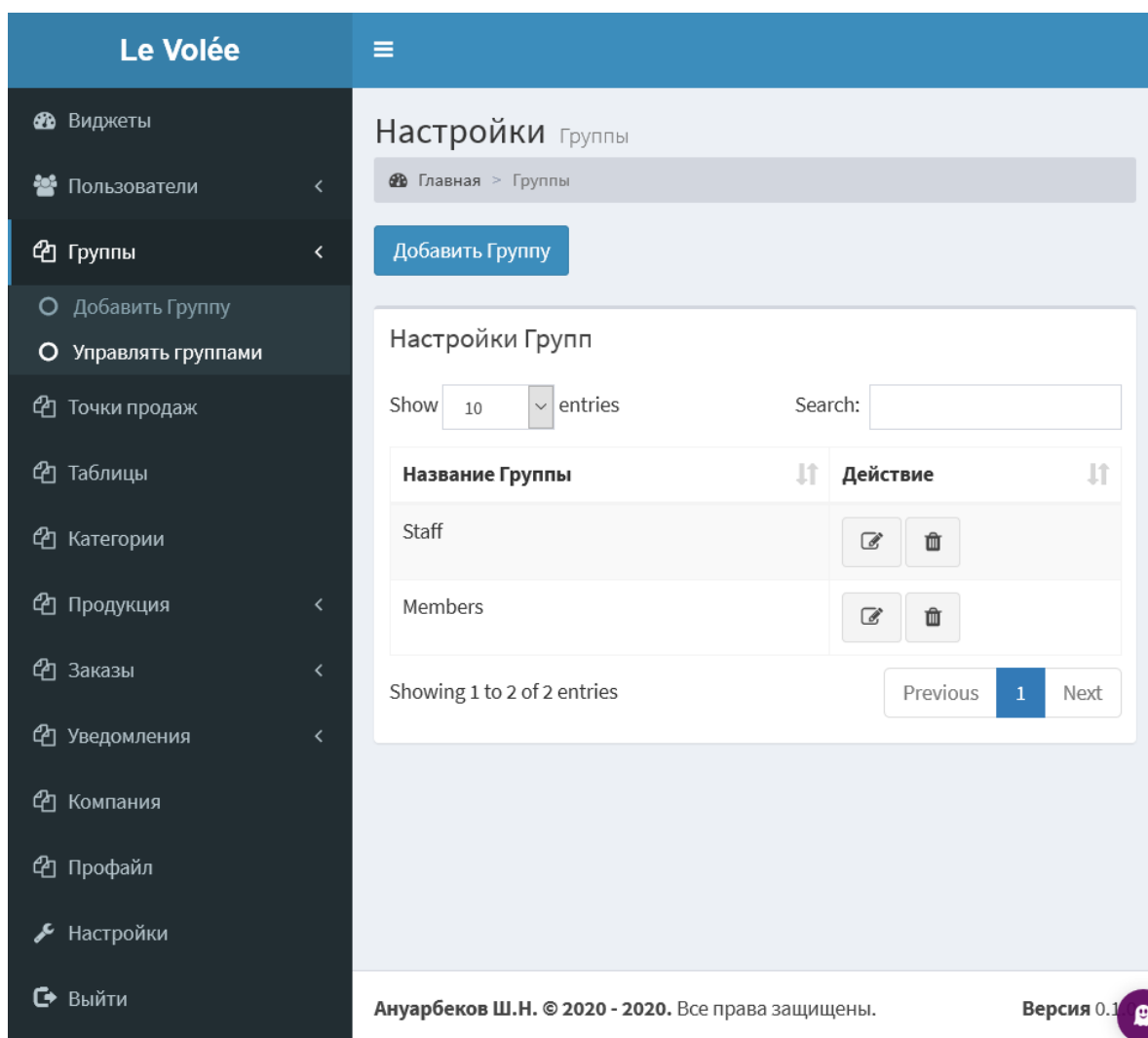


Рисунок 2.7.12 – Подпункт меню «Группы» – «Управлять группами»

Le Volée

Виджеты

Пользователи

Группы

Точки продаж

Таблицы

Категории

Продукция

Добавить продукт

Управлять продукцией

Заказы

Уведомления

Компания

Профайл

Настройки

Выйти

Настройки

Продукция

Главная > Продукция

Добавить Продукт

Изображение

Название продукта

Введите название

Цена

Введите цену

Описание

Normal text

Bold

Italic

Underline

Small

“”

Введите описание

Категория

Точка продажи

Активно

Да

Сохранить изменения

Назад

Ануарбеков Ш.Н. © 2020 - 2020. Все права защищены.Версия 0.1.0

Рисунок 2.7.13 – Подпункт меню «Продукция» – «Добавить продукт»

Le Volée

Виджеты

Пользователи

Группы

Точки продаж

Таблицы

Категории

Продукция

Добавить продукт

Управлять продукцией

Заказы

Уведомления

Компания

Профайл

Настройки

Выйти

Настройки

Продукция

Главная > Продукция

Добавить Продукт

Посмотреть Продукт

Настройки Продуктов

Show

10

entries

Search:

Изображение	Название	Цена	Точка продаж	Состояние
	Olmeca Gold	1200	Бар	Active
	Olmeca Blanco	1000	Бар	Active
	Hendrick's	2000	Бар	Active
	Beefeater	1200	Бар	Active
	Sailor Jerry	1500	Бар	Active
	Havana club Especial	1000	Бар	Active
	Havana club 3YO	1000	Бар	Active
	Tullamore Dew	1500	Бар	Active
	Chivas Regal 12YO	2000	Бар	Active
	Glenfiddich	2500	Бар	Active

Showing 1 to 10 of 31 entries

Previous

1

2

3

4

Next

Ануарбеков Ш.Н. © 2020 - 2020. Все права защищены.

Версия 0.1.0

Рисунок 2.7.14 – Подпункт меню «Продукция» – «Управлять продукцией»

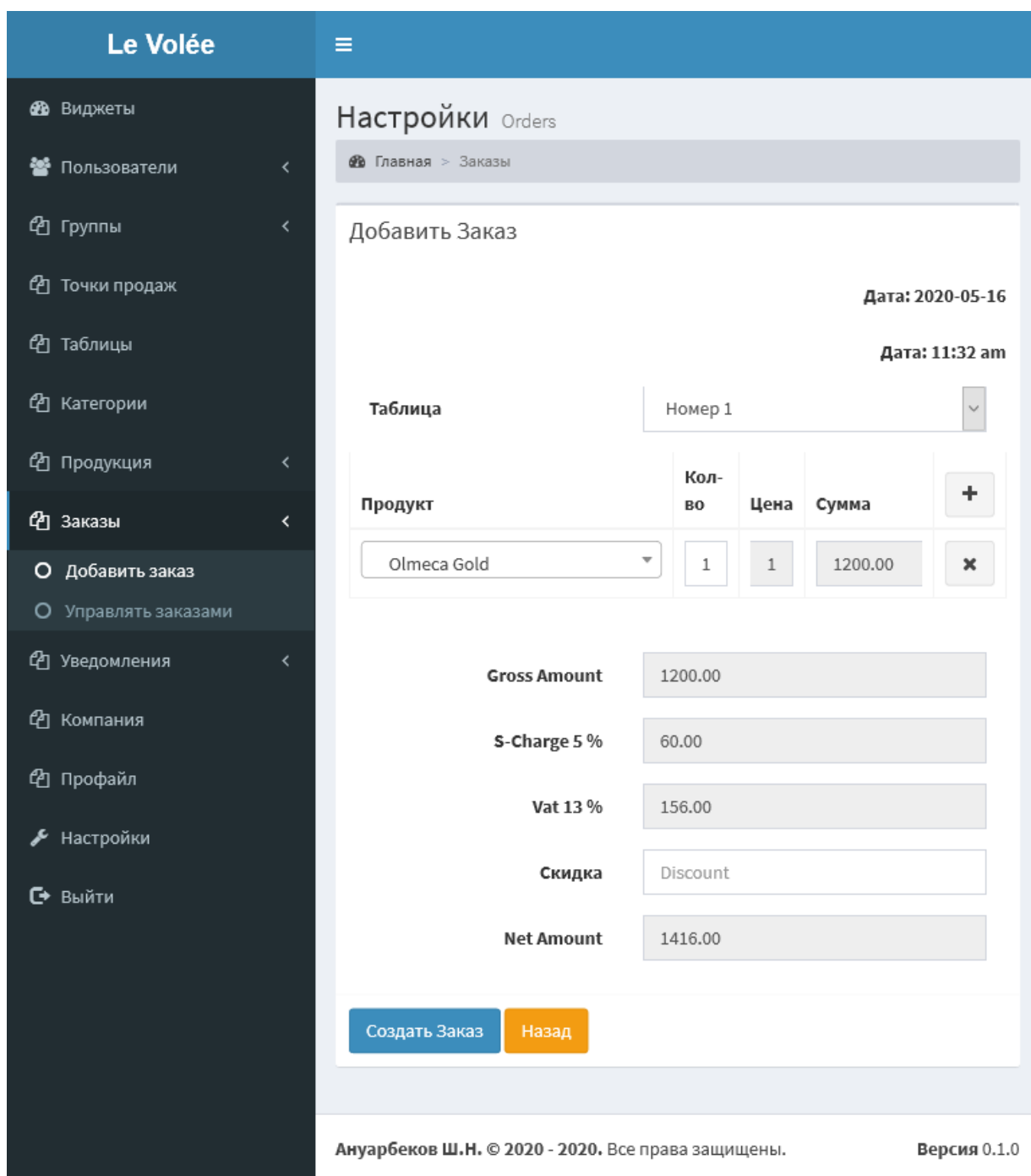


Рисунок 2.7.15 – Подпункт меню «Заказы» – «Добавить заказ»

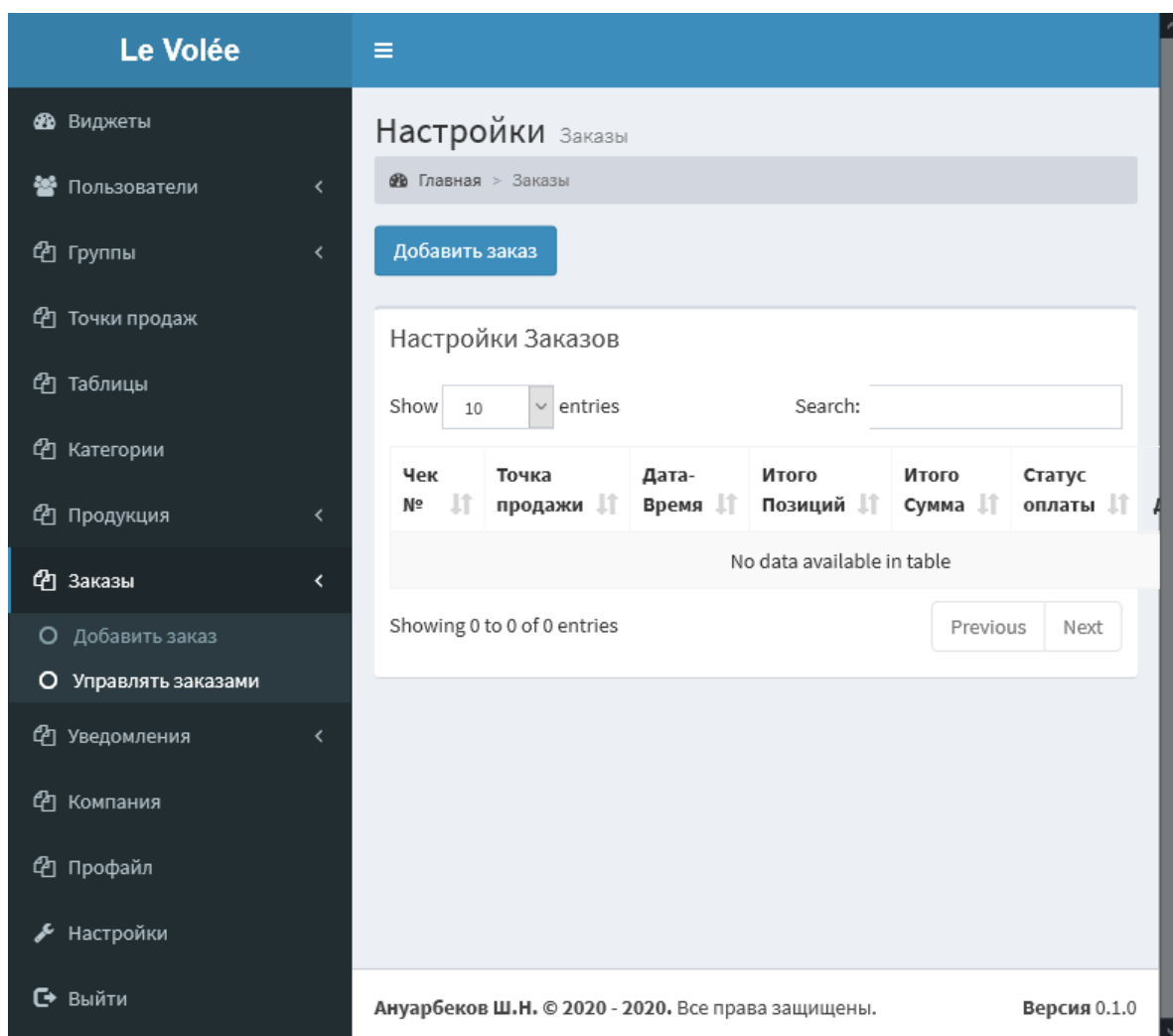


Рисунок 2.7.16 – Подпункт меню «Заказы» – «Управлять заказами»

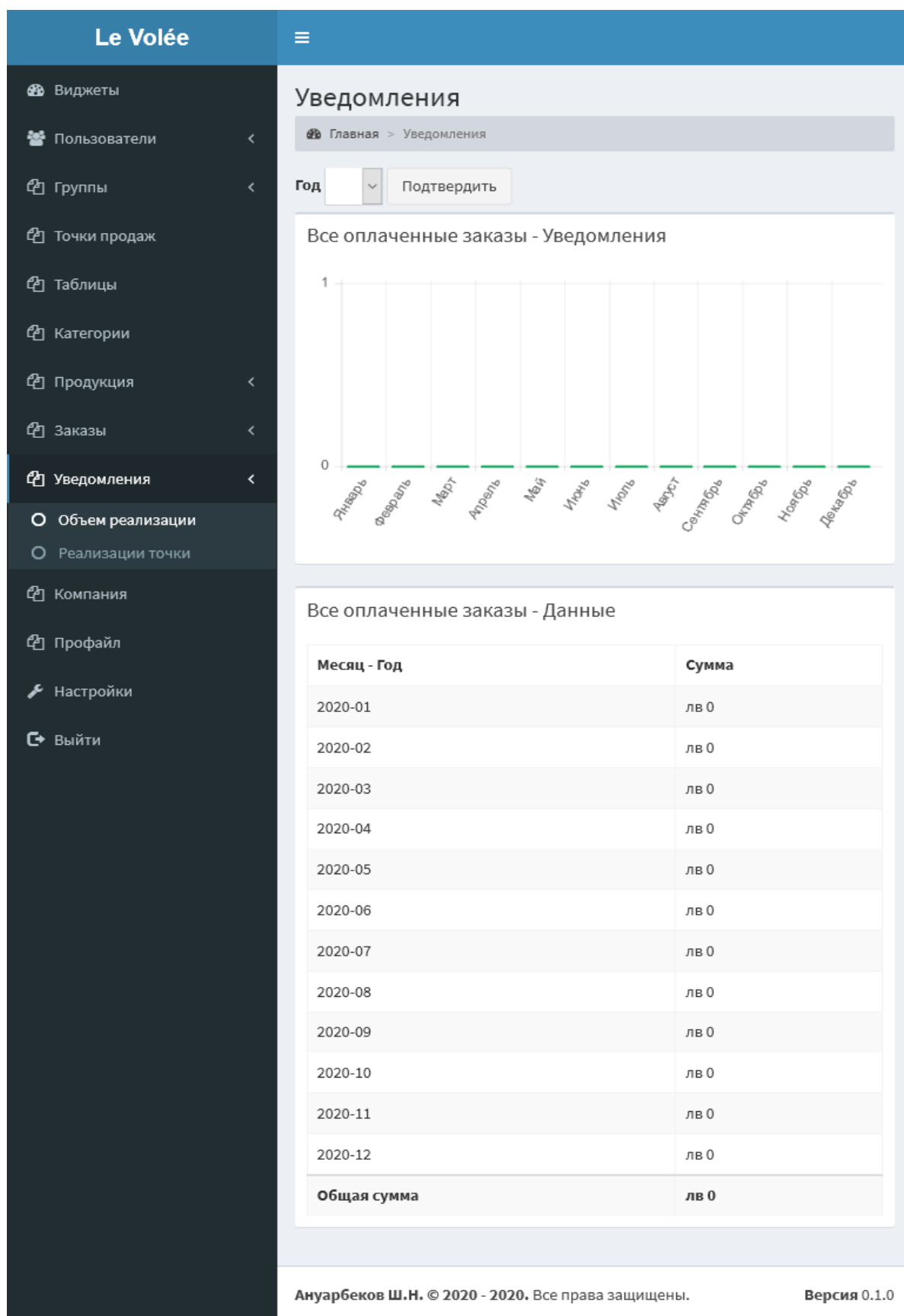


Рисунок 2.7.17 – Подпункт меню «Уведомления» – «Объем реализации»

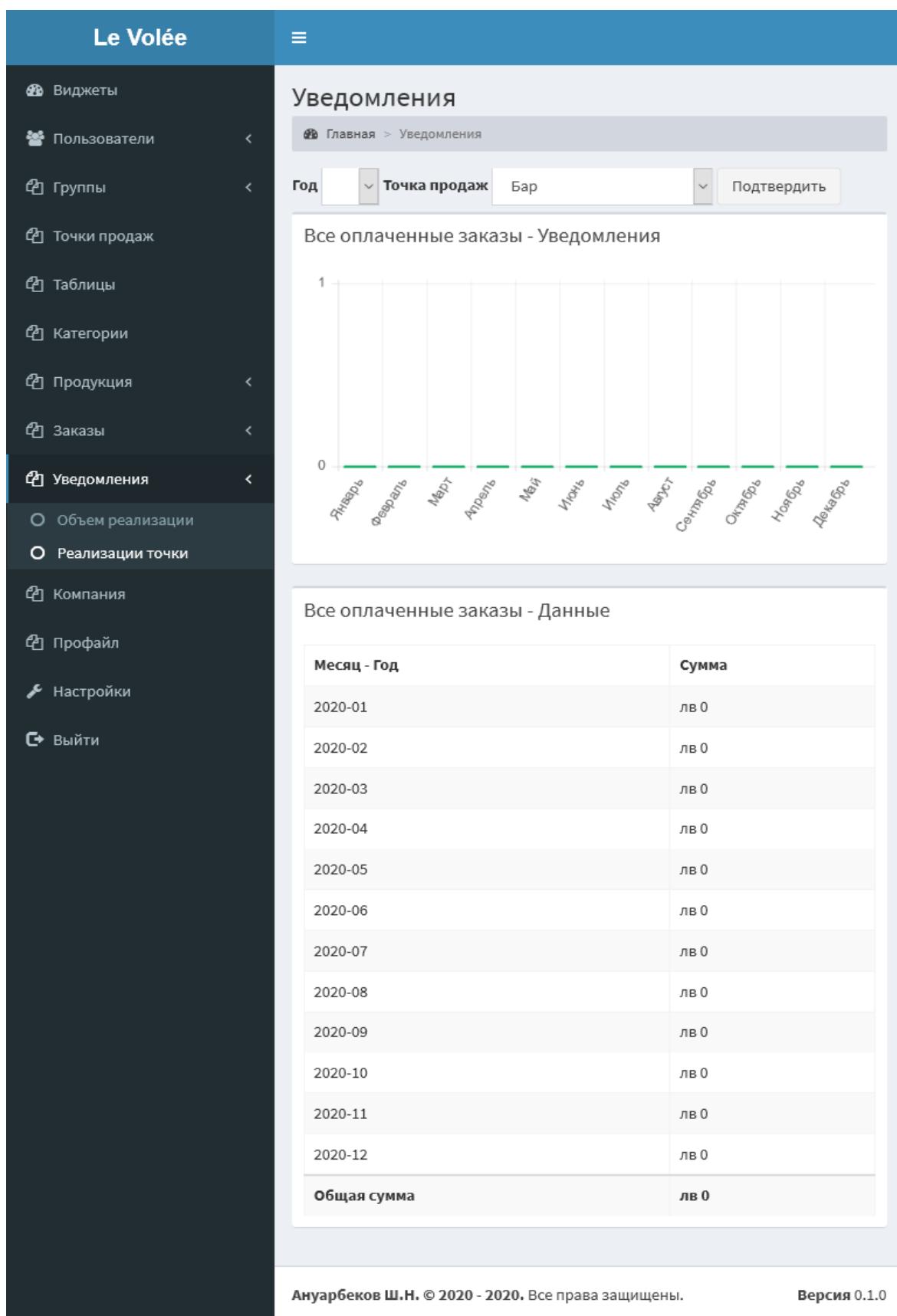


Рисунок 2.7.18 – Подпункт меню «Уведомления» – «Реализации точки»

Также боковое меню имеет два вида: открытый и скрытый. При скрытом варианте меню полностью сдвигается в бок, пряча все пункты, и оставляя лишь логотип приложения.

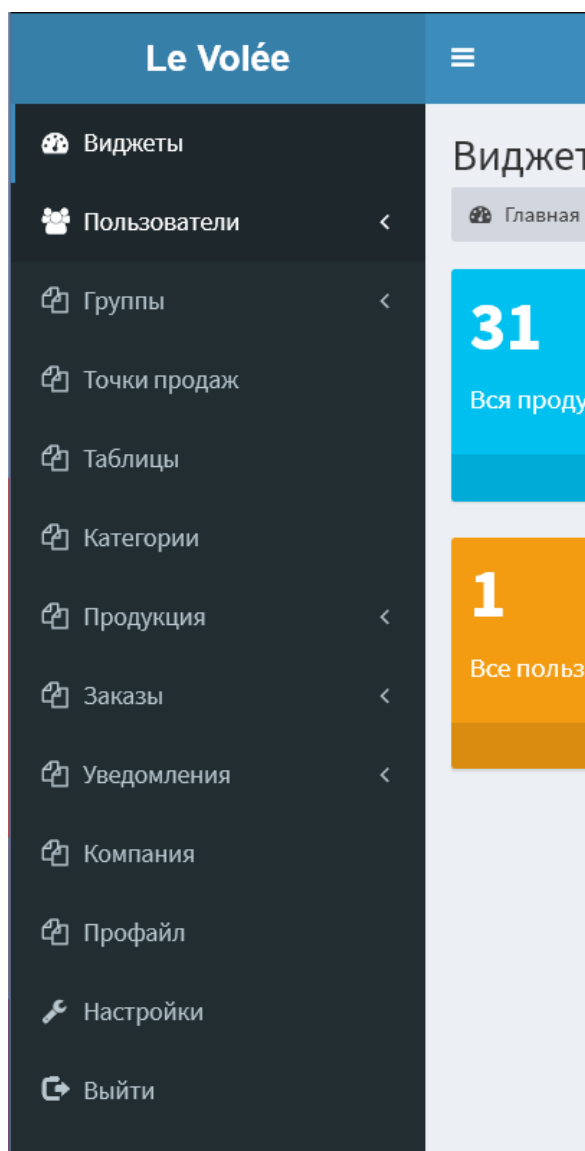


Рисунок 2.7.18 – Открытый вариант бокового меню

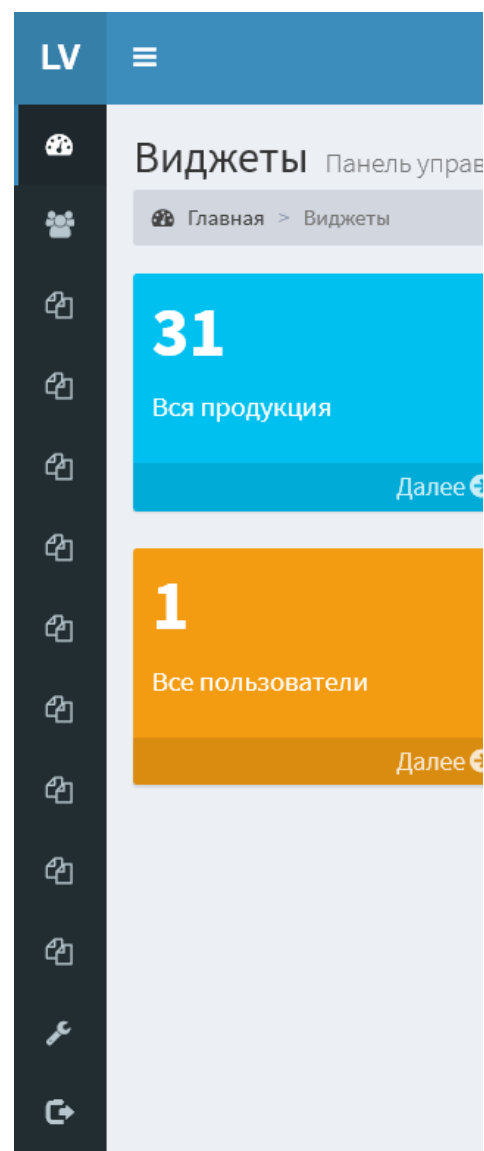


Рисунок 2.7.19 – Закрытый вариант бокового меню



## 2.8 Размещение Веб-приложения

Веб-хостинг — это услуга, которая позволяет организациям и частным лицам размещать веб-сайт или веб-страницу в Интернете. Веб-хостинг или поставщик услуг веб-хостинга — это бизнес, который предоставляет технологии и услуги, необходимые для просмотра веб-сайта или веб-страницы в Интернете. Веб-сайты размещаются или хранятся на специальных компьютерах, называемых серверами. Когда пользователям Интернета нужно посетить нужный сайт, все, что им нужно сделать, это ввести адрес веб-сайта или домен в свой браузер. Затем их компьютер подключится к нужному серверу, и требуемые веб-страницы будут доставлены им через браузер.

Большинство хостинговых компаний требуют, чтобы владелец сайта владел своим доменом, чтобы размещать их вместе с ними. Если домен отсутствует, хостинговые компании могут помочь приобрести его.

Вот некоторые функции, которые следует ожидать от хостинг-провайдера:

Учетные записи электронной почты. Как упоминалось ранее, большинство хостинг-провайдеров требуют, чтобы пользователи имели свое собственное доменное имя. С помощью доменного имени (например, `www.yourwebsite.com`) и функций учетной записи электронной почты, предоставляемых хостинговой компанией, можно создавать учетные записи электронной почты домена (например, `yourname@yourwebsite.com`).

Доступ по FTP. Использование FTP позволяет загружать файлы с локального компьютера на веб-сервер. Если создается собственный веб-сайт, используя собственные HTML-файлы, можно передавать файлы собственного компьютера на веб-сервер через FTP, что позволяет получить доступ к веб-сайту через Интернет.

Поддержка WordPress WordPress - это инструмент для создания веб-сайтов. Это мощная система ведения блога и управления сайтом, которая является удобным способом создания и управления сайтом. WordPress поддерживает более 25% веб-сайтов в Интернете. Большинство хостинг-провайдеров сразу сообщат, совместимы ли их планы с WordPress или нет. Простые требования для размещения сайтов WordPress включают: PHP версии 7 или выше; MySQL версии 5.6 или выше.

При решении создать и разместить сайт на Website.com(например), в дополнение к доступу к менеджеру сайта, можно получить собственный домен, адреса электронной почты и веб-хостинг, объединенные в одну подписку.

Прежде чем начинать искать хостинг, нужно продумать, что именно нужно получить от хоста. К примеру веб-сайт личного портфолио не нуждается в выделенном сервере, как интернет-магазин с большим трафиком и большим объемом трафика не должен выходить за рамки с самым дешевым планом совместного хостинга.

## 2.9 Тестирование Веб-приложения

Тестирование веб-приложений, методика тестирования программного обеспечения предназначена исключительно для тестирования приложений, размещенных в Интернете, в которых тестируются интерфейсы приложений и другие функциональные возможности.

Тестирование веб-приложений – методы:

1) Проверка работоспособности - ниже приведены некоторые проверки, которые выполняются, но не ограничиваются приведенным ниже списком:

- нужно убедиться что нет мертвых страниц или недействительных перенаправлений;

- ввод неправильных данных для выполнения отрицательного тестирования;

- проверка рабочего процесса системы;

- проверка целостности данных.

2) Юзабилити-тестирование – для проверки простоты использования приложения:

- проверка навигации и элементов управления;

- проверка контента;

- проверка интуитивности пользовательских элементов.

3) Тестирование интерфейса - выполняется для проверки интерфейса и потока данных из одной системы в другую.

4) Тестирование на совместимость. Тестирование на совместимость выполняется в зависимости от контекста приложения:

- совместимость браузера;

- совместимость с операционной системой;

- совместим с различными устройствами, такими как ноутбук, мобильный телефон и т. Д.

5) Тестирование производительности – выполняется для проверки времени отклика сервера и пропускной способности при различных условиях нагрузки:

- нагрузочное тестирование - это самая простая форма тестирования, проводимая для понимания поведения системы при определенной нагрузке. Нагрузочное тестирование приведет к измерению важных бизнес-критических транзакций, а также будет отслеживаться нагрузка на базу данных, сервер приложений и т. д.;

- стресс-тестирование - выполняется для определения верхнего предела пропускной способности системы, а также для определения того, как система работает, если текущая нагрузка значительно превышает ожидаемый максимум;

- испытание на текучесть - испытание на текучесть, также известное как испытание на прочность, проводится для определения параметров системы при постоянной ожидаемой нагрузке. Во время тестов выдержки контролируются такие параметры, как использование памяти, чтобы обнаружить утечки памяти

или другие проблемы с производительностью. Основной целью является выявление производительности системы при устойчивом использовании;

- spike-тестирование - spike-тестирование выполняется путем внезапного увеличения количества пользователей на очень большую величину и измерения производительности системы. Основная цель - определить, сможет ли система выдержать рабочую нагрузку.

б) Тестирование безопасности – выполняется для проверки того, защищено ли приложение в Интернете, поскольку кража данных и несанкционированный доступ являются более распространенными проблемами, и ниже приведены некоторые методы для проверки уровня безопасности системы:

- инъектирование;
- сломанная аутентификация и управление сессиями;
- межсайтовый скриптинг (XSS);
- небезопасные прямые ссылки на объекты;
- неправильная настройка безопасности;
- конфиденциальная информация;
- отсутствует контроль доступа на уровне функций;
- подделка межсайтовых запросов (CSRF);
- использование компонентов с известными уязвимостями;
- непроверенные перенаправления и пересылки.

### **3 Экономическая часть**

#### **3.1 Резюме**

В наше время большое количество информации в основном храниться на электронных носителях, являясь наиболее защищенным местом хранения. К тому же большинство вычислений и операций выполняется на компьютерах. С целью улучшения работы ресторана также используют информационные технологии.

Дипломный проект посвящён разработке продукта, которое будет вести учет продаж и инвентаризацию бара «Шустов». Целью работы является создание максимально удобного приложения для улучшения и ускорения работы предприятия в сфере общепита. Данное приложение будет помощником для сотрудников бара, и будет внедрен с призванием облегчить работу.

На даны момент во внедрении и разработке программного продукта стоит его окупаемость и востребованность, так как оно должно стоит вложенных в него средств. Необходим правильный расчет стоимости затраченных работ в произведение продукта.

### 3.2 Трудоемкость разработки ПП

Главными задачами этапа определить и разделить объем работ и их последовательность.

Таблица 3.1 – Распределение работ по этапам, видам и оценка их трудоемкости

№	Этап разработки ПП	Вид работы на данном этапе	Трудоемкость разработки ПП	
			Чел. x час	Час x день
1	Анализ требований	Анализ предметной области, установление целей и задач.	1 x 24	8 x 3
2	Анализ рынка	Анализ конкурентов, альтернативных продуктов, потребность в данном программном продукте, выявление преимуществ перед подобными товарами.	1 x 24	8 x 3
3	Проектирование	Разработка технического задания, формирование требований к проекту, определение набора средств для разработки.	1 x 40	8 x 5
4	Реализация	Построение диаграмм, построение базы данных, ее заполнение, реализация интерфейса, разработка ПП.	1 x 96	8 x 12
5	Тестирование	Тестирование, исправление ошибок и неполадок продукта.	1 x 16	8 x 2
6	Внедрение и поддержка	Установка программного обеспечения, исправление выявленных ошибок, ознакомление персонала с продуктом и его сопровождение.	1 x 24	8 x 3
Итого трудоемкость выполненного проекта			1 x 224	8 x 28

### 3.3 Расчет затрат на разработку ПП

Расчет полных затрат на разработку проектного решения будет осуществлен по формуле:

$$C_{ni} = Z_{тр} + Z_{сзи} + M_i + P_{ci} + P_{зи} + P_{ни} \quad (3.1)$$

где  $Z_{тр}$  – затраты на оплату труда разработчика, тенге;

$Z_{сзи}$  – отчисления по социальному налогу, тенге;

$M_i$  – затраты на материалы, тенге;

$P_{ci}$  – затраты на специальные программные средства необходимые для разработки, тенге;

$P_{зи}$  – прочие затраты, тенге;

$P_{ни}$  – накладные расходы, тенге.

Размер оплаты труда разработчиков рассчитывается по формуле:

$$Z_{фот} = \sum_{i=1}^n ЧС_i \times T_i \quad (3.2)$$

где  $T_i$  – трудоемкость разработки;

$ЧС_i$  – часовая ставка, тенге/час.

Затраты на оплату труда записаны в таблицу 3.2.

Таблица 3.2 – Затраты на оплату труда

Исполнитель	Трудоемкость, чел. х час	Часовая ставка, тенге/час	Сумма, тенге
Разработчик - программист	1 х 224	700	156800
Итого затрат на оплату труда			156800

Социальный налог составляет 9,5% от заработной платы сотрудника и рассчитывается по формуле:

$$Z_{сзи} = (Z_{тр} - ПО - ВОСМС) \times 9,5\% - Z_{coi} \quad (3.3)$$

где ПО – пенсионные отчисления, которые составляют 10% от заработной платы.

$$ПО = Z_{тр} \times 10\% \quad (3.4)$$

Социальные отчисления составляют 3,5% от дохода разработчика, рассчитывается по формуле:

$$З_{соi} = (З_{тр} - ПО) \times 3,5\% \quad (3.5)$$

Отчисления на взнос по обязательному социальному медицинскому страхованию (ВОСМС) составляют 2% от заработной платы работника и рассчитываются по формуле:

$$ВОСМС = З_{тр} \times 2\% \quad (3.6)$$

Тогда,

$$\begin{aligned} ПО &= 156800 \times 10\% = 15680 \text{ тенге;} \\ ВОСМС &= 156800 \times 2\% = 3136 \text{ тенге;} \\ З_{соi} &= (156800 - 15680) \times 3,5\% = 4939 \text{ тенге;} \\ З_{сзi} &= (156800 - 15680 - 1568) \times 9,5\% - 5116 = 8318 \text{ тенге.} \end{aligned}$$

Сумма всех налогов равна 10,46% от заработной платы. Отсюда:

$$З_{сзi} = 156800 \times 10,46\% = 16401 \text{ тенге}$$

Затраты на материалы определяются по формуле:

$$M_i = \frac{З_{тр} \times H_{мз}}{100\%} \quad (3.7)$$

где  $H_{мз}$  – норма расхода материалов от заработной платы (3-5%). Поэтому затраты на материалы составляют,

$$M_i = \frac{156800 \times 5\%}{100\%} = 7840 \text{ тенге.}$$

Для разработки понадобятся средства, программное и техническое обеспечение, представленное в таблице 3.3

Таблица 3.3 – Инструментарий

Наименование	Описание	Кол-во, шт.	Цена за единицу, тг	Сумма, тг
Среда для разработки программного обеспечения	Visual Studio	1	0	0
Программа для редактирования кода	SublimeText 3	1	0	0
Программа для проектирования базы данных	MySQL Workbench	1	0	0
Фреймворк для разработки интерфейса	CodeIgniter	1	0	0
Ноутбук Asus n550jk	Core i7-4710HQ/RAM 8Gb/ HDD 930GB/ Nvidia 850M 4Gb/ 1920x1080	1	254000	254000
Итого				254000

Затраты на электроэнергию рассчитываются по формуле 3.8:

$$З_э = \sum_{i=1}^n M_i \times K_i \times T_i \times Ц \quad (3.8)$$

где  $M_i$  – паспортная мощность электрооборудования, кВт;

$K_i$  – коэффициент использования мощности,  $K_i$  равен 0,8;

$T_i$  – время работы оборудования;

$Ц$  – цена электроэнергии, тг/кВт\*ч.

Для работы ноутбука по паспорту необходимо 120 Вт или же 0,120 кВт.

Цена электроэнергии с 1 января 2020 года составляет 19,17 тенге с НДС за 1 кВт/ч.

Следовательно затраты на электроэнергию равны:

$$З_э = 0,120 \times 0,8 \times 224 \times 19,17 = 412 \text{ тенге.}$$

Сумма годовых амортизационных отчислений определяется методом уменьшающегося остатка по формуле:

$$A = \frac{\Phi \times H_a}{100} \quad (3.9)$$



где  $\Phi$  – первоначальная стоимость основных производственных фондов;  
 $H_a$  – норма амортизации.

При использовании метода уменьшающегося остатка применяется удвоенная ставка амортизации.

Годовые нормы амортизации основных фондов принимаются по налоговому кодексу РК или определяется, на основе срока полезного использования основных фондов по формуле:

$$H_a = \frac{100}{T} \quad (3.10)$$

где  $T$  – возможный срок использования, год.

Для программного продукта срок полезного использования – 4 года.

$$H_a = \frac{100}{4} \times 2 = 50\%$$

Амортизационные отчисления приведены в таблице 3.4

Таблица 3.4 – Амортизация основных фондов

Период	Норма амортизации	Амортизация за период	Накопленная амортизация	Балансовая стоимость на конец года
				254000
2020 год	50%	127000	127000	127000
2021 год	50%	63500	190500	63500
2022 год	50%	31750	222250	31750
Продолжение таблицы Таблица 3.4				
2023 год	50%	15875	254000	0

Амортизация в последний период равен балансовой стоимости на конец предыдущего периода.

Для вычисления общих затрат на разработку программного продукта необходимо рассчитать амортизационные отчисления за период разработки по формуле:

$$З_{ам} = \frac{\Phi \times H_a \times N}{100\% \times 12 \times t} \quad (3.11)$$

где  $N$  – время использования программного продукта, дни;

$t$  – количество рабочих дней в месяце.

Основные производственные фонды используются только на протяжении периода разработки и тестирования продукта, что значит время использования равно 224 часам или 28 дням.

Следовательно,

$$З_{ам} = \frac{254000 \times 50\% \times 28}{100\% \times 12 \times 21} = 14111 \text{ тенге.}$$

Статья «Прочие затраты» включает затраты на содержание аппарата управления, вспомогательных хозяйств и опытных (экспериментальных) производств, а также расходы на общехозяйственные нужды, относятся на конкретное ПО по нормативу в процентном соотношении к заработной плате исполнителей. Норматив устанавливается в целом организацией:

$$P_{hi} = З_{тр} \times \frac{H_{нр}}{100} \quad (3.12)$$

где  $P_{hi}$  – накладные расходы на конкретное ПО (тыс. тенге);

$H_{нр}$  – норматив накладных расходов в целом по организации и равен 70%.

Следовательно,

$$P_{hi} = 156800 \times \frac{70}{100} = 109760 \text{ тенге.}$$

Расходы по статье «Прочие расходы» на конкретное ПО включает затраты на приобретение и подготовку специальной научно-технической информации и специальной литературы. Определяется по нормативу, разрабатываемому в целом по организации, в процентах к заработной плате:

$$П_{зи} = З_{тр} \times \frac{H_{рнк}}{100} \quad (3.13)$$

где  $H_{рнк}$  – норматив прочих затрат в целом по организации и равен 20%.

Получается,

$$П_{зи} = З_{тр} \times \frac{20}{100} = 31360 \text{ тенге.}$$

Результаты выполненных расчетов записаны в таблице 3.5. Диаграмма по затратам проиллюстрирована на рисунке 3.1.

Таблица 3.5 – Результаты выполненных расчетов

Затраты на разработку	Условное обозначение	Значение, тенге	В процентах от общей суммы
Заработная плата	$Z_{mp}$	156800	26%
Налоги от заработной платы	$Z_n$	16401	3%
Материалы	$M_i$	7840	1%
Спецоборудование	$P_{ci}$	254000	45%
Затраты на электроэнергию	$Z_э$	412	0%
Амортизация основных фондов	$Z_{ам}$	14111	3%
Прочие затраты	$P_{zi}$	31360	5%
Накладные расходы	$P_{ni}$	109760	18%
Итого	$C$	590684	100%

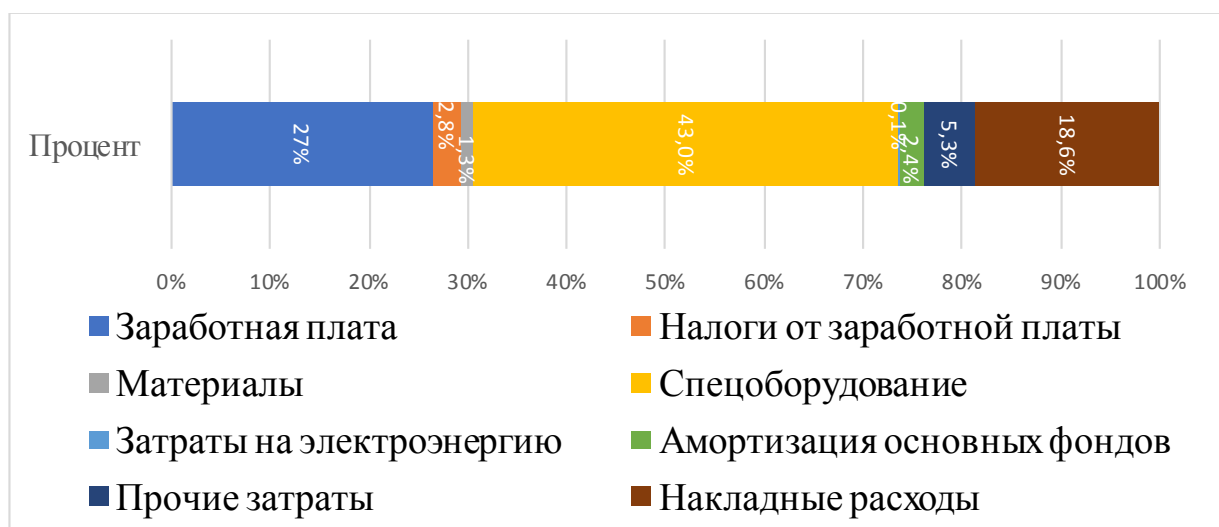


Рисунок 3.1 – Диаграмма затрат

Первоначальную цену продукта можно рассчитать по формуле

$$C_0 = C \left( 1 \times \frac{P}{100} \right),$$

где  $P$  – рентабельность (50%).

Цена готовой продукции с учетом НДС, которая равна 12%, рассчитывается по формуле:

$$C_p = C_0 + \text{НДС} \quad (3.14)$$

Отсюда, конечная цена программного продукта равна:

$$C_0 = 590684 \left( 1 \times \frac{50}{100} \right) = 886026 \text{ тенге}$$

$$C_p = 886026 + (886026 \times 12\%) = 992349 \text{ тенге.}$$

### 3.4 Расчет сравнительной экономической эффективности ПП

Разработка программы призвана ускорить работу менеджмента ресторана, а также скорость обслуживания и в последствии увеличении прибыли.

Величина ожидаемого годового эффекта от внедрения ПП рассчитывается по формуле

$$\mathcal{E}_r = \mathcal{E}_{yg} - K \times E_n \quad (3.15)$$

где  $\mathcal{E}_r$  – ожидаемый годовой экономический эффект, тенге;

$\mathcal{E}_{yg}$  – ожидаемая условно-годовая экономия, тенге;

$K$  – капитальные вложения, тенге;

$E_n$  – нормативный коэффициент эффективности капитальных вложений.

$E_n$  рассчитывается по формуле:

$$E_n = \frac{1}{T_n} \quad (3.16)$$

где  $T_n$  – нормативный срок окупаемости капитальных вложений, лет.

Для программных продуктов срок окупаемости равен 4 годам.

Благодаря программному продукту можно сократить персонал на 1 человека, который вел учет деятельности ресторана. Заработная плата менеджера за месяц составляет 250000 тенге, или же 3000000 тенге в год.

Следовательно,

$$\mathcal{E}_r = 3000000 - 992349 \times (0,25) = 2751913 \text{ тенге.}$$

Расчетный коэффициент экономической эффективности капитальных вложений составляет:

$$E_p = \frac{\mathcal{E}_{yg}}{K} \quad (3.17)$$

Расчетный срок окупаемости капитальных вложений происходит по формуле:

$$T_p = \frac{1}{E_p} \quad (3.18)$$

Отсюда,

$$E_p = \frac{3000000}{992349} = 3,02;$$

$$T_p = \frac{1}{3,02} = 0,33 \text{ года.}$$

Срок окупаемости составляет 0,33 года или же чуть меньше 4 месяцев.  
Результаты расчетов внесены в таблицу 3.5.

Таблица 3.5 – Показатели сравнительной экономической эффективности от внедрения программного продукта

Наименование показателей	Значение
Условная годовая экономия затрат, тенге	3000000
Коэффициент экономической эффективности капитальных вложений	3,02
Сроки окупаемости капитальных вложений, год	0,33

### **3.5 Вывод по экономическому разделу**

В данном разделе были рассмотрены экономические вопросы по разрабатываемому программному продукту для бара «Шустов». Данное программное обеспечение было выбрано в рамках заказа от предприятия ТОО «Шустов».

Были рассмотрены все этапы данного проекта и определение их трудоемкости. Были рассмотрены все расходы на создание данного продукта. Была рассчитана его стоимость и срок окупаемости, равный чуть меньше 4 месяцев.

Программа облегчает работу учета и обслуживания, что должно увеличить скорость работы персонала, и также увеличить прибыль. Также программа заменяет некоторую работу персонала, чем может быть причиной снижения затрат на их содержания.

При просмотре аналогичных продуктов, то можно отметить, что стоимость конкурентов ниже, чем стоимость данного программного продукта. Но также стоит заметить, что доступ сведениям бара можно будет получать не только лишь на ПК, но также и на других платформах, так как бэк-офис будет представлять собой сайт, на который можно будет зайти где угодно, и когда угодно. Отслеживать деятельность ресторана можно будет в реальном времени что в некотором смысле очень удобно для владельцев бизнеса.

## **4 Безопасность жизнедеятельности**

### **4.1 Анализ потенциально опасных и вредных факторов воздействия на персонал в процессе работы**

В данном дипломном проекте разрабатывается программный продукт для бара «Шустов». Необходимо рабочее место, где будет использоваться данный продукт, в данном случае помещение самого бара, где будут работать пять человек: администратор, кассир, бармен, барбэк, раннер. В помещении находится разное оборудование, включая терминал кассира. Само помещение имеет длину  $A=13\text{м}$ , ширину  $B=9\text{м}$ , и высоту потолка  $H=2,8\text{м}$ . В данном помещении отсутствуют окна, на потолках установлены светодиодные лампы Эдисона: мощ.  $N$  ос. уст.,  $\text{Вт/м}^2$  50. Также установлено видеосистема наблюдения, не нарушающая допустимую норму шума.

Планирование играет особо важную роль в рамках условий труда. Как видно из опыта, правильное планирование рабочего места обязано соответствовать всем правилам потребности персонала в комфорте и удобстве при выполнении своей рабочей деятельности и разумном использовании энергии, времени, рациональном подходе к использованию площади. Немало важную роль играет соблюдение правил техники безопасности с электронным оборудованием [11].

При работе на компьютере могут возникать опасные и вредные факторы, воздействие которых на организм человека может принести ему вред и привести к травматизму. Основные факторы с местом их возникновения и нормами изложены в ГОСТ 12.1.003-74/80 [11].

Основные опасные и вредные факторы:

- повышенное значение напряжения электрической сети возникающее на рабочем месте оператора, или в помещении с оборудованием, которое может вызвать электротравму;
- электрическая дуга, которая может произойти в распределительном щите, и приводящее к ожогам и пожару;
- повышенная напряженность электрического поля и электромагнитного излучения, происходящая на электроустановках 220В, в помещении, и может вызывать профессиональные заболевания, электротравмы, пожары;
- повышенный уровень статического электричества, которое также может возникнуть на электроустановках и рабочем месте, и может привести к серьезным последствиям как пожар, взрыв, электрический удар;
- повышенная или пониженная температура воздуха, влажность, подвижность воздуха рабочей зоны возникающее на рабочем месте и в помещении вызывают перегрев или переохлаждение организма;
- недостаточная освещенность рабочей зоны может вызвать утомляемость, дискомфорт, опасность травматизма, и ухудшение зрения
- Повышенный уровень шума на рабочем месте может вызвать нервно-психическую перегрузку, заболевания органов слуха;



Основным средством управления программным продуктом является ПЭВМ типа IBM PC оснащенная сенсорным монитором SVGA, которая также является источником негативного воздействия на здоровье человека. Следовательно нужно также учитывать время работы оператора за компьютером, дабы не вызывать проблем со здоровьем.

В течение времени, воздух скопившийся в помещении за время работы постоянно изменяет свой химический состав, температуру и влажность под действием достаточно большого количества факторов: переменность параметров внешнего воздуха, выделения теплоты, влажности, пыли и вредных газов от людей и электронных девайсов. Воздух в помещении под воздействием этих факторов может плохо сказаться на здоровье людей. Дабы избежать случаев изменения качества воздуха, следует периодически осуществлять воздухообмен, то есть производить смену воздуха в помещении.

Для этого предназначены системы кондиционирования. Технологические системы кондиционирования предназначены для обеспечения параметров воздуха (температуры, влажности и подвижности), в максимальной степени отвечающих требованиям определенного производственного или технологического процесса.

Проектирование теплового и влажностного баланса для кондиционируемого помещения производится общепринятыми методами, принятыми в отопительно-вентиляционной технике. Здесь должны быть учтены все факторы, влияющие на изменение состояния воздушной среды помещения. Также по причине отсутствия естественного освещения, условия искусственного освещения на промышленном предприятии оказывают большое влияние на зрительную работоспособность, физическое и моральное состояние людей, а, следовательно, на производительность труда, качество продукции и производственный травматизм.

Для создания благоприятных условий труда производственное освещение должно отвечать следующим требованиям:

- освещенность на рабочем месте должна соответствовать характеру выполняемой работы по СН и СНиП РК 2.04-05-2002 «Естественное и искусственное освещение. Общие требования»;
  - яркость на рабочей поверхности и в пределах окружающего пространства должна распределяться по возможности равномерно;
  - резкие тени на рабочей поверхности должны отсутствовать;
  - освещение должно обеспечивать необходимый спектральный состав света для правильной цветопередачи;
  - система освещения не должна являться источником других вредных факторов (шум и т.д.), а также должна быть электро- и пожаробезопасной.
- Искусственное освещение применяется при отсутствии или недостаточности естественного освещения, осуществляется путем использования таких источников света как лампы накаливания, газоразрядные лампы, плоские и щелевые светодиоды.

## 4.2 Расчетная часть

### 4.2.1 Исходные данные

Город: Алматы;

Параметры помещения (Д x Ш x В), м: 13 x 9 x 2.8;

Данные по оборудованию: 1 шт.;

Мощность  $P_{об}$ , 0,1 кВт/ч;

КПД  $\eta$  0,9;

Данные по ист. света: мощ. N ос.уст., 50 Вт/м<sup>2</sup>;

Вид ист. св. люминесцентные лампы

Число сотрудников, из них: мужчины 4

Расчетное время суток, ч.: 13-14

Температура в помещении, °С: летом 25  
зимой 20

Вид положения работы: стоя

#### 4.2.2 Расчет тепловых нагрузок в помещении

В помещениях различного назначения действуют в основном тепловые нагрузки, возникающие снаружи помещения (наружные); а также тепловые нагрузки, возникающие внутри зданий (внутренние).

Данные нагрузки представлены следующими составляющими:

- теплопоступления или теплопотери в результате разности температур снаружи и внутри здания через стены, потолки, полы, окна и двери;
- разность температур снаружи здания и внутри него летом является положительной, в результате чего имеет место приток тепла снаружи во внутрь помещения; и наоборот – зимой эта разность отрицательна и направление потока тепла меняется;
- теплопоступления от солнечного излучения через застекленные площади; данная нагрузка проявляется в форме ощущаемого тепла;
- теплопоступления от инфильтрации.

В зависимости от времени года и времени суток наружные тепловые нагрузки могут быть положительными.

Теплопоступления и теплопотери в результате разности температур определяются по формуле:

$$Q_{огр} = V_{пом} \cdot X_o \cdot (t_{Нрасч} - t_{Врасч}), \text{ Вт}$$

где  $V_{пом}$  – объем помещения,  $\text{м}^3$ :

$$V_{пом} = 13 \cdot 9 \cdot 2,8 = 327,6 \text{ м}^3;$$

$X_o$  – удельная тепловая характеристика,  $\text{Вт}/\text{м}^3 \text{ } ^\circ\text{C}$ :

$$X_o = 0,42 \text{ Вт}/\text{м}^3 \text{ } ^\circ\text{C}$$

$t_{Нрасч}$  – наружная температура. Для холодного периода – средняя температура самого холодного месяца в 13 часов, для теплого периода – средней температуре самого жаркого месяца в 13 часов.

$t_{Врасч}$  – внутренняя температура, выбирается с учетом комфортных условий или технологических требований, предъявляемых к производственным процессам.

Для теплого времени года

$$t_{Нрасч} = 29,4 \text{ } ^\circ\text{C}$$

$$t_{Врасч} = 25 \text{ } ^\circ\text{C}$$

$$Q_{огр} = 327,6 \cdot 0,42 \cdot (29,4 - 25) = 605,4 \text{ Вт}$$

Для холодного времени года

$$t_{Нрасч} = -9 \text{ } ^\circ\text{C}$$

$$t_{Врасч} = 20 \text{ } ^\circ\text{C}$$

$$Q_{\text{огр}} = 327,6 * 0,42 * |(-9 - 20)| = 3990,1 \text{ Вт}$$

Избыточная теплота солнечного излучения в зависимости от типа стекла почти до 90% поглощается средой помещения, остальная часть отражается. Максимальная тепловая нагрузка достигается при максимальном уровне излучения, которое имеет прямую и рассеянную составляющие. Интенсивность излучения зависит от ширины местности, времени года и времени суток.

Теплопоступление от солнечного излучения через остекление определяется по формуле:

$$Q_p = (q^I F_0^I + q^{II} F_0^{II}) \cdot \beta_{c.з.}$$

где  $q^I, q^{II}$  – тепловые потоки от прямой и рассеянной солнечной радиации, Вт/м<sup>2</sup>;

$F_0^I, F_0^{II}$  – площади светового проема, облучаемые и необлучаемые прямой солнечной радиацией, м<sup>2</sup>;

Поскольку помещение не имеет окон, тогда:

$$Q_p = 0 \text{ Дж}$$

Внутренние нагрузки в жилых, офисных или относящихся к сфере обслуживания помещениях складываются в основном из тепла:

- выделяемого людьми;
- выделяемого лампами и осветительными, электробытовыми приборами;
- выделяемого компьютерами, печатающими устройствами фотокопировальными машинами пр.;

В производственных и технологических помещениях различного назначения дополнительными источниками тепловыделений могут быть: нагретое производственное оборудование, горячие материалы, в том числе жидкости и различного рода полуфабрикаты, продукты сгорания и химических реакций.

Теплопоступления от людей зависят от интенсивности выполняемой работы и параметров окружающего воздуха. Тепло, выделяемое человеком, складывается из ощутимого (явного), то есть передаваемого в воздух помещения путем конвекции и лучеиспусканий, и скрытого тепла, затрачиваемого на испарение влаги с поверхности кожи и из легких.

Летом при 25 °С один мужчина выделяет явного тепла 61 Вт, а общего – 102 Вт. Женщина выделяет 85% от нормы тепловыделений взрослого мужчины. Тогда выделение явного тепла в помещении составит:

$$Q_{\text{л}}^{\text{я}} = 61 * 4 = 244 \text{ Вт}$$

А выделение общего тепла:

$$Q_{\text{л}}^{\text{я}} = 102 * 4 = 408 \text{ Вт}$$

По таблице [8] зимой при  $18^{\circ}\text{C}$  один мужчина выделяет явного тепла 85 Вт, а общего – 105 Вт. Женщина выделяет 85% от нормы тепловыделений взрослого мужчины. Тогда выделение явного тепла в помещении составит:  
 $Q_{\text{л}}^{\text{я}} = 85 \cdot 4 = 340 \text{ Вт}$

А выделение общего тепла:

$$Q_{\text{л}}^{\text{я}} = 105 \cdot 4 = 420 \text{ Вт}$$

Теплопоступление от осветительных приборов, оргтехники и оборудования рассчитывается следующим образом. Теплопоступление от ламп определяется по формуле:

$$Q_{\text{осв}} = \eta \cdot N_{\text{уст}} \cdot F_{\text{пол}}, \text{ Вт}$$

где  $\eta$  – коэффициент перехода электрической энергии в тепловую (для люминесцентных ламп  $\eta=0.5-0.6$ );

$N_{\text{осв}}$  – установленная мощность ламп ( $N=50 \text{ Вт/м}^2$ );

$F_{\text{пол}}$  – площадь пола:  $F_{\text{пол}} = 13 \cdot 9 = 117 \text{ м}^2$

Тогда:

$$Q_{\text{осв}} = 0,5 \cdot 50 \cdot 117 = 2925 \text{ Вт}$$

Тепло, выделяемое производственным оборудованием, определяется по формуле:

$$Q_{\text{об}} = N_{\text{уст}} \cdot K$$

$$Q_{\text{об}} = 0,1 \cdot 0,9 \cdot 1 = 0,09 \text{ кВт}$$

Теплопритоки, возникающие за счет находящейся оргтехники, - это 30% мощности оборудования:

$$Q_{\text{орг}} = 0,3 \cdot 0,1 \cdot 1 = 0,03 \text{ кВт.}$$

### 4.2.3 Расчет теплового баланса помещения

На основании выполненных расчетов составим баланс тепlopоступлений в помещении:

$$Q_{\text{вс}} = Q_{\text{д}} + Q_{\text{в}} + Q_{\text{ин}} + Q_{\text{из}} + Q_{\text{ид}} + Q_{\text{ид}}$$

Лето:

$$Q_{\text{изб}} = 605,4 + 0 + 244 + 2925 + 90 + 30 = 3894,4 \text{ Дж}$$

Зима:

$$Q_{\text{изб}} = 3990,1 + 0 + 340 + 2925 + 90 + 30 = 7375,1 \text{ Дж}$$

Так как тепловой баланс для лета больше зимнего теплового баланса, то рассчитаем теплonaпряженность воздуха по формуле:

$$Q_{\text{н}} = Q_{\text{изб.зима}} \cdot 860 / V_{\text{пом}} = 7375,1 \cdot 860 / 327,6 = 19,3 \text{ ккал/м}^3.$$

При  $Q_{\text{н}} > 20 \text{ ккал/м}^3$ ,  $\Delta t = 8 \text{ }^{\circ}\text{C}$ ,

Определение количества воздуха, необходимое для поступления в помещение:

$$L = Q_{\text{изб}} \cdot 860 / (C \cdot \Delta t \cdot \gamma) = 7375,1 \cdot 860 / (0,24 \cdot 1000 \cdot 8 \cdot 1,206) = 2739,2 \text{ м}^3/\text{час}$$

где  $C=0,24 \text{ ккал/(кг}^{\circ}\text{C)}$  – теплоемкость воздуха,

$\gamma=1,206 \text{ кг/м}^3$  – удельная масса приточного воздуха.

Определение кратности воздухообмена:

$$n = L / V_{\text{пом}} = 2739,2 / 327,6 = 8,4 \text{ час}^{-1}.$$

#### 4.2.4 Выбор кондиционера. Схема расположения

Исходя из полученных данных, выберем кондиционер кассетного типа.  
*Кондиционер Shivaki SCH-604BE*



Рисунок 4.1 – Вид кондиционера

Таблица 4.1 – Основные характеристики

Разновидность	комплект
Площадь помещения	160 м <sup>2</sup>
Напряжение	220 В
Мощность охлаждения	16.1 кВт
Потребляемая мощность при охлаждении	5.6 кВт
Режим обогрева	
Мощность обогрева	17.7 кВт
Потребляемая мощность при обогреве	4.8 кВт
Режим вентиляции	
Расход воздуха	2000 м <sup>3</sup> /ч
Макс. уровень шума (внутр. блок)	43 дБ
Макс. уровень шума (внеш. блок)	47 дБ
Хладагент	R410a
Ширина (внутр. блок)	1200 мм
Глубина (внутр. блок)	800 мм
Высота (внутр. блок)	300 мм

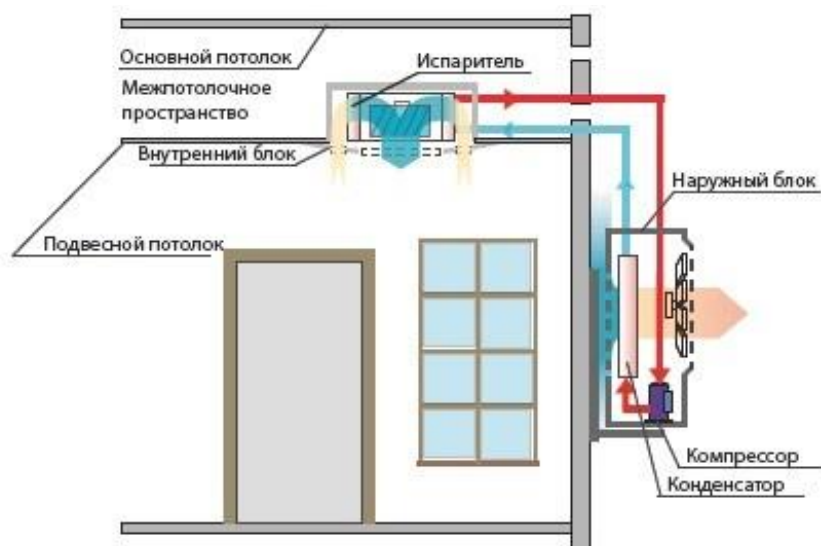
Продолжение таблицы 4.1.

Вес (внутр. блок)	37 кг
Ширина (внеш. блок)	940 мм
Глубина (внеш. блок)	340 мм
Высота (внеш. блок)	1250 мм
Вес (внеш. блок)	105 кг
Страна-производитель	Ю. Корея

Во внешнем блоке находятся компрессор, конденсатор и вентилятор. Внешний блок можно установить на стене здания, на крыше или на чердаке, в подсобном помещении или на балконе, то есть в таком месте, где горячий конденсатор может продуваться атмосферным воздухом более низкой температуры.

Внутренний блок устанавливается непосредственно в кондиционируемом помещении и предназначен для охлаждения или нагревания воздуха, фильтрации его и создания необходимой подвижности воздуха в помещении. Внутренние блоки поддерживают заданную температуру, обеспечивают равномерное распределение воздуха в помещении и работают практически бесшумно.

Управление работой настенного кондиционера производится с дистанционного пульта, который позволяет задать режим работы кондиционера: обогрев, охлаждение, осушку, вентиляцию, ночной режим; задать требуемую температуру, которую должен поддерживать автоматически; выбрать режим работы вентилятора: настроить таймер, который включит или выключит кондиционер в заданное время; автоматически регулировать положение направляющих шторок и изменить таким образом направление воздушного потока.





#### Рисунок 4.2 – Схема расположения кондиционера

#### 4.2.5 Расчет искусственного освещения

Для расчета искусственного освещения используют один из трех методов: по коэффициенту использования светового потока, точечный и метод удельной мощности.

При расчете общего равномерного освещения основным является метод использования светового потока, создаваемого источником света, и с учетом отражения от стен, потолка, пола.

Рассматриваемый метод заключается в определении значения коэффициента  $\eta$ , равного отношению светового потока, падающего на поверхность, к полному потоку осветительного прибора.

В практике расчетов значения коэффициентов  $\eta$  находятся из таблиц, связывающих геометрические параметры помещения (индекс помещения) с их оптическими характеристиками.

Индекс помещения определяется по формуле :

$$i = \frac{L \cdot B}{Hc \cdot (L + B)}$$
$$i = \frac{13 \cdot 9}{1,5 \cdot (13 + 9)} = 3,5 \Rightarrow$$

где  $L$  - длина помещения, м;

$B$  - ширина помещения, м;

$h_{расч}$  - расчетная высота, м.

Определим коэффициент использования светового потока по таблице ( $\eta=60\%$ ).

Определим число светильников по формуле:

$$N = \frac{E_n \cdot K_3 \cdot S \cdot Z}{n \cdot \Phi \cdot \eta}$$

$E_n = 200$  лк - заданное номинальное освещение, по таблице 3.12[5];

$K_3$  –коэффициент запаса, принимают по таблице 3.11[5].  $K_3=1,1$ ;

$S = 117 \text{ м}^2$  – площадь помещения;

$Z = 1,1$ - коэффициент неравномерности освещения;

$n$  – количество ламп в светильнике.

$$N = \frac{200 \cdot 1,1 \cdot 117 \cdot 1,1}{1 \cdot 4800 \cdot 0,6} = 9,83 \approx 10_{шт.}$$

Вывод: для обеспечения необходимой освещенности нужно 10 светильников Philips MASTER TL5 HO Xtra 1SL 3000K.

Планировка с расположением светильников:

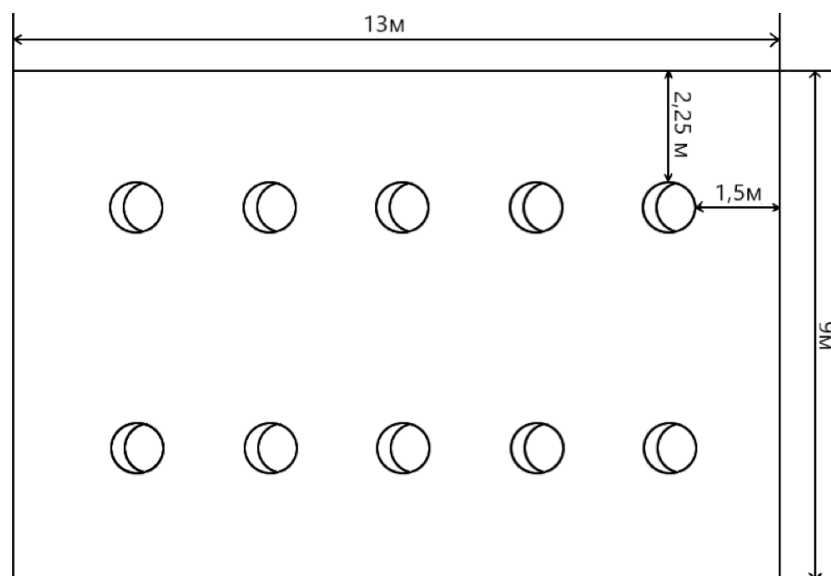


Рисунок 4.3 – Планировка светильников

## Заключение

Была проделана огромная работа по созданию продукта для упрощения работы бара. Была проанализирована деятельность данного предприятия и условия в которых она работает. Много задач были решены на программном уровне и создана аналитика для непрерывной работы. Был сделан веб доступ с любой точки и с любого места.

Настройка и реализация общей архитектуры была одной из самых сложные задачи в начале этого проекта. Возникли проблемы с достижением решение о необходимых функциональных возможностях, необходимых для разных пользователей бара модель автоматизации.

Используя MVC, получилась возможность реализовать варианты использования и изменить их в соответствии с индивидуальными потребностями владельца, администратора, кассира. Модель взаимодействия компонентов имела решающее значение в связывании всего. Стало очевидно, что все кратко и понятно, и большая часть кода может быть легко использована для различных целей функции.

При начале реализации базы данных и кодирования, структуры и организация класса, а также их функции были очень неясны. Были сомнения в обязательных параметрах и информации, необходимой для реализации базы данных. Различные функциональные возможности разных пользователей вызывали ненужную повторяемость в база данных.

Используя диаграмму классов и спецификацию интерфейса, которые обсуждались с заказчиком, получена возможность внедрять организованную базу данных, содержащую только необходимые, информация и устранение ненужной повторяемости. Это также сделало распределение задачи внутри конструкции легче, так как функциональные возможности, необходимые сейчас стали понятны для реализации.

Продукт не был сосредоточен на базе данных при реализации проекта. Хотя был проделан общий обзор над работай, там не было никакого особого внимания к техническим характеристикам для эффективной работы базы данных. Знания от прошедших практик и рабочего опыта очень помогли с реализацией этой программы. Эти знания могут быть очень полезны для многих сфер в будущем, при работе с аналогичными проектами.

Эта модель программного продукта может быть легко переделана для личного нужд потребителей (пункты меню, настройки управления и т.д.). Тем не менее, реализация продукта как веб-приложения для всех устройств может стать в будущем перспективной из-за их разнообразности и повсеместного использования.

## Список литературы

- 1 Ольга Курочкина. Ресторан: работа над ошибками. Москва: ИГ «Ресторанные ведомости», 2018.
- 2 Горелкина Е. Продажи в сфере HoReCa. – Вершина – 55 с.
- 3 Тусупова Б.Б. Конспект лекций. Проектирование и оптимизация информационных систем. Для студентов всех форм обучения всех специальностей. – Алматы: АУЭС, 2019.
- 4 Купарова А.Т. Методические указания к выполнению дипломных работ (проектов) для студентов специальностей: 5B060200-Информатика, 5B070300 - Информационные системы, 5B070400-Вычислительная техника и программное обеспечение. – Алматы: АУЭС, 2019 – 41 с.
- 5 [https://ontask.ru/start-future/horeca-chto-eto-takoe.html#\\_HoReCa](https://ontask.ru/start-future/horeca-chto-eto-takoe.html#_HoReCa)
- 6 Абдимуратов Ж.С., Мананбаева С.Е. Безопасность жизнедеятельности. Методические указания к выполнению раздела «Расчет производственного освещения» в выпускных работах для всех специальностей. Бакалавриат – Алматы: АИЭС, 2009. - 20 с.
- 7 СНиП РК 2.04-05-2002. Естественное и искусственное освещение. Общие требования. – М.: Стройиздат, 2002.
- 8 СНиП II - 4 - 79. Естественное и искусственное освещение. Нормы проектирования.-М.: Стройиздат, 1980.- 48 с.
- 9 Пособие 2.91 к СНиП 2.04.05-91 «Расчет поступления теплоты солнечной радиации в помещения».
- 10 Хакимжанов Т.Е. Расчет аспирационных систем. Дипломное проектирование. Для студентов всех форм обучения всех специальностей. – Алматы: АИЭС, 2002. – 30 с.
- 11 СНиП РК 2.04-01-2001. Общие строительные нормы и правила устройства систем вентиляции и кондиционирования воздуха. Административные и бытовые здания
- 12 <https://www.kazedu.kz/referat/132726/22>
- 13 Скляр, Дэвид. Изучаем РНР 7: руководство по созданию интерактивных веб-сайтов. : Пер. с англ. — СПб. : ООО “Альфа-книга”, 2017. — 464 с. : ил. — Парал. тит. англ.
- 14 <https://www.kazedu.kz/referat/195775>

## Приложение А

### Листинг

*Views/login.php:*

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <title>Вход</title>
  <meta content="width=device-width, initial-scale=1, maximum-scale=1, user-
scalable=no" name="viewport">
  <link          rel="stylesheet"          href="<?php          echo
base_url('assets/bower_components/bootstrap/dist/css/bootstrap.min.css') ?>">
  <link          rel="stylesheet"          href="<?php          echo
base_url('assets/bower_components/font-awesome/css/font-awesome.min.css')
?>">
  <link          rel="stylesheet"          href="<?php          echo
base_url('assets/bower_components/Ionicons/css/ionicons.min.css') ?>">
  <link          rel="stylesheet"          href="<?php          echo
base_url('assets/dist/css/AdminLTE.min.css') ?>">
  <link          rel="stylesheet"          href="<?php          echo
base_url('assets/plugins/iCheck/square/blue.css') ?>">
  <link          rel="stylesheet"          href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,600,70
0,300italic,400italic,600italic">
</head>
<body class="hold-transition login-page">
<div class="login-box">
  <div class="login-logo">
    <a href="<?php echo base_url('auth'); ?>"><b>Вход</b></a>
  </div>
  <!-- /.login-logo -->
  <div class="login-box-body">
    <p class="login-box-msg">Войдите для начала работы</p>
    <?php echo validation_errors(); ?>
    <?php if(!empty($errors)) {
      echo $errors;
    } ?>
    <form action="<?php echo base_url('auth/login') ?>" method="post">
```

```
<div class="form-group has-feedback">
  <input type="email" class="form-control" name="email" id="email"
placeholder="Электронная почта" autocomplete="off">
  <span class="glyphicon glyphicon-envelope form-control-
feedback"></span>
</div>
<div class="form-group has-feedback">
  <input type="password" class="form-control" name="password"
id="password" placeholder="Пароль" autocomplete="off">
  <span class="glyphicon glyphicon-lock form-control-feedback"></span>
</div>
<div class="row">
  <div class="col-xs-8">
    <div class="checkbox icheck">
      <label>
        <input type="checkbox"> Запомнить
      </label>
    </div>
  </div>
  <div class="col-xs-4">
    <button type="submit" class="btn btn-primary btn-block btn-
flat">Войти</button>
  </div>
<!-- /.col -->
</div>
</form>
</div>
</div>
<script src="<?php echo
base_url('assets/bower_components/jquery/dist/jquery.min.js') ?>"></script>
<script src="<?php echo
base_url('assets/bower_components/bootstrap/dist/js/bootstrap.min.js')
?>"></script>
<script src="<?php echo base_url('assets/plugins/iCheck/icheck.min.js')
?>"></script>
<script>
$(function () {
  $('input').iCheck({
    checkboxClass: 'checkbox_square-blue',
```

```
        radioClass: 'iradio_square-blue',
        increaseArea: '20%' // optional
    });
});
</script>
</body>
</html>
```

*Views/Dashboard.php*

```
<div class="content-wrapper">
  <section class="content-header">
    <h1>
      Виджеты
      <small>Панель управления</small>
    </h1>
    <ol class="breadcrumb">
      <li><a href="#"><i class="fa fa-dashboard"></i> Главная</a></li>
      <li class="active">Виджеты</li>
    </ol>
  </section>
  <section class="content">
    <?php if($is_admin == true): ?>
      <div class="row">
        <div class="col-lg-3 col-xs-6">
          <div class="small-box bg-aqua">
            <div class="inner">
              <h3><?php echo $total_products ?></h3>
              <p>Вся продукция</p>
            </div>
            <div class="icon">
              <i class="ion ion-bag"></i>
            </div>
            <a href="<?php echo base_url('products/') ?>" class="small-box-
footer">Далее <i class="fa fa-arrow-circle-right"></i></a>
          </div>
        </div>
        <div class="col-lg-3 col-xs-6">
          <div class="small-box bg-green">
```



```
<div class="inner">
  <h3><?php echo $total_paid_orders ?></h3>
  <p>Все оплаченные заказы</p>
</div>
<div class="icon">
  <i class="ion ion-stats-bars"></i>
</div>
<a href="<?php echo base_url('orders/') ?>" class="small-box-
footer">Далее <i class="fa fa-arrow-circle-right"></i></a>
</div>
</div>
<div class="col-lg-3 col-xs-6">
  <div class="small-box bg-yellow">
    <div class="inner">
      <h3><?php echo $total_users; ?></h3>
      <p>Все пользователи</p>
    </div>
    <div class="icon">
      <i class="ion ion-android-people"></i>
    </div>
    <a href="<?php echo base_url('users/') ?>" class="small-box-
footer">Далее <i class="fa fa-arrow-circle-right"></i></a>
    </div>
  </div>
  <div class="col-lg-3 col-xs-6">
    <div class="small-box bg-red">
      <div class="inner">
        <h3><?php echo $total_stores ?></h3>
        <p>Все точки продаж</p>
      </div>
      <div class="icon">
        <i class="ion ion-android-home"></i>
      </div>
      <a href="<?php echo base_url('stores/') ?>" class="small-box-
footer">Далее <i class="fa fa-arrow-circle-right"></i></a>
      </div>
    </div>
  </div>
<?php endif; ?>
```

```
</section>
</div>
<script type="text/javascript">
  $(document).ready(function() {
    $("#dashboardMainMenu").addClass('active');
  });
</script>
```

*Views/Index.html*

```
<!DOCTYPE html>
<html>
<head>
  <title>403 Forbidden</title>
</head>
<body>
<p>Directory access is forbidden.</p>
</body>
</html>
```

*Views/templates/header.php:*

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <title><?php echo $page_title; ?></title>
  <meta content="width=device-width, initial-scale=1, maximum-scale=1, user-
scalable=no" name="viewport">
  <link          rel="stylesheet"          href="<?php          echo
base_url('assets/bower_components/bootstrap/dist/css/bootstrap.min.css') ?>">
  <link          rel="stylesheet"          href="<?php          echo
base_url('assets/bower_components/font-awesome/css/font-awesome.min.css')
?>">
  <link          rel="stylesheet"          href="<?php          echo
base_url('assets/bower_components/Ionicons/css/ionicons.min.css') ?>">
  <link          rel="stylesheet"          href="<?php          echo
base_url('assets/dist/css/AdminLTE.min.css') ?>">
```

```
<link rel="stylesheet" href="<?php echo base_url('assets/dist/css/skins/_all-
skins.min.css') ?>">
<link rel="stylesheet" href="<?php echo
base_url('assets/bower_components/morris.js/morris.css') ?>">
<link rel="stylesheet" href="<?php echo
base_url('assets/bower_components/jvectormap/jquery-jvectormap.css') ?>">
<link rel="stylesheet" href="<?php echo
base_url('assets/bower_components/bootstrap-datepicker/dist/css/bootstrap-
datepicker.min.css') ?>">
<link rel="stylesheet" href="<?php echo
base_url('assets/bower_components/bootstrap-
daterangepicker/daterangepicker.css') ?>">
<link rel="stylesheet" href="<?php echo base_url('assets/plugins/bootstrap-
wysihtml5/bootstrap3-wysihtml5.min.css') ?>">
<link rel="stylesheet" href="<?php echo
base_url('assets/bower_components/datatables.net-
bs/css/dataTables.bootstrap.min.css') ?>">
<link rel="stylesheet" href="<?php echo
base_url('assets/bower_components/select2/dist/css/select2.min.css') ?>">
<link rel="stylesheet" href="<?php echo
base_url('assets/plugins/fileinput/fileinput.min.css') ?>">
<link rel="stylesheet"
href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,600,70
0,300italic,400italic,600italic">
<script src="<?php echo
base_url('assets/bower_components/jquery/dist/jquery.min.js') ?>"></script>
<script src="<?php echo base_url('assets/bower_components/jquery-ui/jquery-
ui.min.js') ?>"></script>
<script>
$.widget.bridge('uibutton', $.ui.button);
</script>
<script src="<?php echo
base_url('assets/bower_components/bootstrap/dist/js/bootstrap.min.js')
?>"></script>
<script src="<?php echo
base_url('assets/bower_components/raphael/raphael.min.js') ?>"></script>
<script src="<?php echo
base_url('assets/bower_components/morris.js/morris.min.js') ?>"></script>
```

```
<script src="<?php echo base_url('assets/bower_components/jquery-
sparkline/dist/jquery.sparkline.min.js') ?>"></script>
<script src="<?php echo base_url('assets/plugins/jvectormap/jquery-jvectormap-
1.2.2.min.js') ?>"></script>
<script src="<?php echo base_url('assets/plugins/jvectormap/jquery-jvectormap-
world-mill-en.js') ?>"></script>
<script src="<?php echo base_url('assets/bower_components/jquery-
knob/dist/jquery.knob.min.js') ?>"></script>
<script src="<?php echo base_url('assets/bower_components/moment/min/moment.min.js') ?>"></script>
<script src="<?php echo base_url('assets/bower_components/bootstrap-
daterangepicker/daterangepicker.js') ?>"></script>
<script src="<?php echo base_url('assets/bower_components/bootstrap-
datepicker/dist/js/bootstrap-datepicker.min.js') ?>"></script>
<script src="<?php echo base_url('assets/plugins/bootstrap-
wysihtml5/bootstrap3-wysihtml5.all.min.js') ?>"></script>
<script src="<?php echo base_url('assets/bower_components/jquery-
slimscroll/jquery.slimscroll.min.js') ?>"></script>
<script src="<?php echo base_url('assets/bower_components/fastclick/lib/fastclick.js') ?>"></script>
<script src="<?php echo base_url('assets/plugins/fileinput/fileinput.min.js')
?>"></script>
<script src="<?php echo base_url('assets/bower_components/select2/dist/js/select2.full.min.js')
?>"></script>
<script src="<?php echo base_url('assets/bower_components/Chart.js/Chart.js')
?>"></script>
<script src="<?php echo base_url('assets/dist/js/adminlte.min.js') ?>"></script>
<script src="<?php echo base_url('assets/dist/js/demo.js') ?>"></script>
<script src="<?php echo base_url('assets/bower_components/datatables.net/js/jquery.dataTables.min.js')
?>"></script>
<script src="<?php echo base_url('assets/bower_components/datatables.net-
bs/js/dataTables.bootstrap.min.js') ?>"></script>
</head>
<body class="hold-transition skin-blue sidebar-mini">
<div class="wrapper">
```

*Views/templates/header\_menu.php:*

```
<header class="main-header">
  <a class="logo">
    <span class="logo-mini"><b>LV</b></span>
    <span class="logo-lg"><b>Le Volée</b></span>
  </a>
  <nav class="navbar navbar-static-top">
    <a href="#" class="sidebar-toggle" data-toggle="push-menu" role="button">
      <span class="sr-only">Панель навигации</span>
    </a>
  </nav>
</header>
```

*Views/templates/side\_menubar.php:*

```
<aside class="main-sidebar">
  <section class="sidebar">
    <ul class="sidebar-menu" data-widget="tree">
      <li id="dashboardMainMenu">
        <a href="<?php echo base_url('dashboard') ?>">
          <i class="fa fa-dashboard"></i> <span>Виджеты</span>
        </a>
      </li>
      <?php if($user_permission): ?>
        <?php if(in_array('createUser', $user_permission) || in_array('updateUser',
$user_permission) || in_array('viewUser', $user_permission) ||
in_array('deleteUser', $user_permission)): ?>
          <li class="treeview" id="userMainNav">
            <a href="#">
              <i class="fa fa-users"></i>
              <span>Пользователи</span>
              <span class="pull-right-container">
                <i class="fa fa-angle-left pull-right"></i>
              </span>
            </a>
            <ul class="treeview-menu">
              <?php if(in_array('createUser', $user_permission)): ?>
```

```
<li id="createUserSubNav"><a href="php echo
base_url('users/create') ?&gt;"&gt;&lt;i class="fa fa-circle-o"&gt;&lt;/i&gt; Добавить
пользователя&lt;/a&gt;&lt;/li&gt;
&lt;?php endif; ?&gt;
&lt;?php if(in_array('updateUser', $user_permission) ||
in_array('viewUser', $user_permission) || in_array('deleteUser',
$user_permission)): ?&gt;
&lt;li id="manageUserSubNav"&gt;&lt;a href="<?php echo base_url('users')
?&gt;"&gt;&lt;i class="fa fa-circle-o"&gt;&lt;/i&gt; Управлять пользователями&lt;/a&gt;&lt;/li&gt;
&lt;?php endif; ?&gt;
&lt;/ul&gt;
&lt;/li&gt;
&lt;?php endif; ?&gt;
&lt;?php if(in_array('createGroup', $user_permission) ||
in_array('updateGroup', $user_permission) || in_array('viewGroup',
$user_permission) || in_array('deleteGroup', $user_permission)): ?&gt;
&lt;li class="treeview" id="groupMainNav"&gt;
&lt;a href="#"&gt;
&lt;i class="fa fa-files-o"&gt;&lt;/i&gt;
&lt;span&gt;Группы&lt;/span&gt;
&lt;span class="pull-right-container"&gt;
&lt;i class="fa fa-angle-left pull-right"&gt;&lt;/i&gt;
&lt;/span&gt;
&lt;/a&gt;
&lt;ul class="treeview-menu"&gt;
&lt;?php if(in_array('createGroup', $user_permission)): ?&gt;
&lt;li id="createGroupSubMenu"&gt;&lt;a href="<?php echo
base_url('groups/create') ?&gt;"&gt;&lt;i class="fa fa-circle-o"&gt;&lt;/i&gt; Добавить
Группу&lt;/a&gt;&lt;/li&gt;
&lt;?php endif; ?&gt;
&lt;?php if(in_array('updateGroup', $user_permission) ||
in_array('viewGroup', $user_permission) || in_array('deleteGroup',
$user_permission)): ?&gt;
&lt;li id="manageGroupSubMenu"&gt;&lt;a href="<?php echo
base_url('groups') ?&gt;"&gt;&lt;i class="fa fa-circle-o"&gt;&lt;/i&gt; Управлять
группами&lt;/a&gt;&lt;/li&gt;
&lt;?php endif; ?&gt;
&lt;/ul&gt;
&lt;/li&gt;</pre
```

```

        <?php endif; ?>
        <?php if(in_array('createStore', $user_permission) || in_array('updateStore',
        $user_permission) || in_array('viewStore', $user_permission) ||
        in_array('deleteStore', $user_permission)): ?>
            <li id="storesMainNav"><a href="<?php echo base_url('stores/') ?>"><i
            class="fa fa-files-o"></i> <span>Точки продаж</span></a></li>
        <?php endif; ?>
        <?php if(in_array('createTable', $user_permission) || in_array('updateTable',
        $user_permission) || in_array('viewTable', $user_permission) ||
        in_array('deleteTable', $user_permission)): ?>
            <li id="tablesMainNav"><a href="<?php echo base_url('tables/') ?>"><i
            class="fa fa-files-o"></i> <span>Таблицы</span></a></li>
        <?php endif; ?>
        <?php if(in_array('createCategory', $user_permission) ||
        in_array('updateCategory', $user_permission) || in_array('viewCategory',
        $user_permission) || in_array('deleteCategory', $user_permission)): ?>
            <li id="categoryMainNav"><a href="<?php echo base_url('category/')
            ?>"><i class="fa fa-files-o"></i> <span>Категории</span></a></li>
        <?php endif; ?>
        <?php if(in_array('createProduct', $user_permission) ||
        in_array('updateProduct', $user_permission) || in_array('viewProduct',
        $user_permission) || in_array('deleteProduct', $user_permission)): ?>
            <li class="treeview" id="productMainNav">
                <a href="#">
                    <i class="fa fa-files-o"></i>
                    <span>Продукция</span>
                    <span class="pull-right-container">
                        <i class="fa fa-angle-left pull-right"></i>
                    </span>
                </a>
                <ul class="treeview-menu">
                    <?php if(in_array('createProduct', $user_permission)): ?>
                        <li id="createProductSubMenu"><a href="<?php echo
                        base_url('products/create') ?>"><i class="fa fa-circle-o"></i> Добавить
                        продукт</a></li>
                    <?php endif; ?>
                    <?php if(in_array('updateProduct', $user_permission) ||
                    in_array('viewProduct', $user_permission) || in_array('deleteProduct',
                    $user_permission)): ?>

```

```
<li id="manageProductSubMenu"><a href="php echo
base_url('products') ?"><i class="fa fa-circle-o"></i> Управлять
продукцией</a></li>
<?php endif; ?>
</ul>
</li>
<?php endif; ?>
<?php if(in_array('createOrder', $user_permission) ||
in_array('updateOrder', $user_permission) || in_array('viewOrder',
$user_permission) || in_array('deleteOrder', $user_permission)): ?>
<li class="treeview" id="OrderMainNav">
<a href="#">
<i class="fa fa-files-o"></i>
<span>Заказы</span>
<span class="pull-right-container">
<i class="fa fa-angle-left pull-right"></i>
</span>
</a>
<ul class="treeview-menu">
<?php if(in_array('createOrder', $user_permission)): ?>
<li id="createOrderSubMenu"><a href="php echo
base_url('orders/create') ?"><i class="fa fa-circle-o"></i> Добавить
заказ</a></li>
<?php endif; ?>
<?php if(in_array('updateOrder', $user_permission) ||
in_array('viewOrder', $user_permission) || in_array('deleteOrder',
$user_permission)): ?>
<li id="manageOrderSubMenu"><a href="php echo
base_url('orders') ?"><i class="fa fa-circle-o"></i> Управлять
заказами</a></li>
<?php endif; ?>
</ul>
</li>
<?php endif; ?>
<?php if(in_array('viewReport', $user_permission)): ?>
<li class="treeview" id="ReportMainNav">
<a href="#">
<i class="fa fa-files-o"></i>
<span>Уведомления</span>
```



<span class="pull-right-container">

*Продолжение приложения А*

<i class="fa fa-angle-left pull-right"></i>

</span>

</a>

<ul class="treeview-menu">

<?php if(in\_array('viewReport', \$user\_permission)): ?>

<li id="productReportSubMenu"><a href="<?php echo base\_url('reports') ?>"><i class="fa fa-circle-o"></i> Объем реализации</a></li>

<li id="storeReportSubMenu"><a href="<?php echo base\_url('reports/storewise') ?>"><i class="fa fa-circle-o"></i> Реализации точки</a></li>

<?php endif; ?>

</ul>

</li>

<?php endif; ?>

<?php if(in\_array('updateCompany', \$user\_permission)): ?>

<li id="companyMainNav"><a href="<?php echo base\_url('company/') ?>"><i class="fa fa-files-o"></i> <span>Компания</span></a></li>

<?php endif; ?>

<?php if(in\_array('viewProfile', \$user\_permission)): ?>

<li id="profileMainNav"><a href="<?php echo base\_url('users/profile/') ?>"><i class="fa fa-files-o"></i> <span>Профайл</span></a></li>

<?php endif; ?>

<?php if(in\_array('updateSetting', \$user\_permission)): ?>

<li id="settingMainNav"><a href="<?php echo base\_url('users/setting/') ?>"><i class="fa fa-wrench"></i> <span>Настройки</span></a></li>

<?php endif; ?>

<?php endif; ?>

<li><a href="<?php echo base\_url('auth/logout') ?>"><i class="glyphicon glyphicon-log-out"></i> <span>Выйти</span></a></li>

</ul>

</section>

</aside>

*Views/templates/footer.php:*

<footer class="main-footer">

<div class="pull-right hidden-xs">

<b>Версия</b> 0.1.0

*Продолжение приложения А*

</div>

<b>Ануарбеков Ш.Н. &copy; 2020 - <?php echo date('Y'); ?>.</b> Все права защищены.

</footer>

<div class="control-sidebar-bg"></div>

</div>

</body>

</html>

*Views/users/edit.php:*

<div class="content-wrapper">

<!-- Content Header (Page header) -->

<section class="content-header">

<h1>

Настройки

<small>Пользователи</small>

</h1>

<ol class="breadcrumb">

<li><a href="#"><i class="fa fa-dashboard"></i> Главная</a></li>

<li class="active">Пользователи</li>

</ol>

</section>

<section class="content">

<div class="row">

<div class="col-md-12 col-xs-12">

<?php if(\$this->session->flashdata('success')): ?>

<div class="alert alert-success alert-dismissible" role="alert">

<button type="button" class="close" data-dismiss="alert" aria-label="Close"><span aria-hidden="true">&times;</span></button>

<?php echo \$this->session->flashdata('success'); ?>

</div>

<?php elseif(\$this->session->flashdata('error')): ?>

<div class="alert alert-error alert-dismissible" role="alert">

<button type="button" class="close" data-dismiss="alert" aria-label="Close"><span aria-hidden="true">&times;</span></button>

<?php echo \$this->session->flashdata('error'); ?>

</div>

<?php endif; ?>

*Продолжение приложения А*

```
<div class="box">
  <div class="box-header">
    <h3 class="box-title">Изменить Пользователя</h3>
  </div>
  <form role="form" action="<?php base_url('users/create') ?>"
method="post">
    <div class="box-body">
      <?php echo validation_errors(); ?>
      <div class="form-group">
        <label for="groups">Группы</label>
        <select class="form-control" id="groups" name="groups">
          <option value="">Выбрать Группы</option>
          <?php foreach ($group_data as $k => $v): ?>
            <option value="<?php echo $v['id'] ?>" <?php if($user_group['id']
== $v['id']) { echo 'selected'; } ?> ><?php echo $v['group_name'] ?></option>
          <?php endforeach ?>
        </select>
      </div>
      <div class="form-group">
        <label for="groups">Точка продаж</label>
        <select class="form-control" id="store" name="store">
          <option value="">Выбрать точку продаж</option>
          <?php foreach ($store_data as $k => $v): ?>
            <option value="<?php echo $v['id'] ?>" <?php
if($user_data['store_id'] == $v['id']) { echo "selected='selected'"; } ?>><?php
echo $v['name'] ?></option>
          <?php endforeach ?>
        </select>
      </div>
      <div class="form-group">
        <label for="username">Логин</label>
        <input type="text" class="form-control" id="username"
name="username" placeholder="Логин" value="<?php echo
$user_data['username'] ?>" autocomplete="off">
      </div>
      <div class="form-group">
        <label for="email">Почта</label>
```

## Продолжение приложения А

```
<input type="email" class="form-control" id="email" name="email"
placeholder="Почта" value="<?php echo $user_data['email'] ?>"
autocomplete="off">
</div>
<div class="form-group">
    <label for="fname">Имя</label>
    <input type="text" class="form-control" id="fname" name="fname"
placeholder="Имя" value="<?php echo $user_data['firstname'] ?>"
autocomplete="off">
</div>
<div class="form-group">
    <label for="lname">Фамилия</label>
    <input type="text" class="form-control" id="lname" name="lname"
placeholder="Фамилия" value="<?php echo $user_data['lastname'] ?>"
autocomplete="off">
</div>
<div class="form-group">
    <label for="phone">Номер телефона</label>
    <input type="text" class="form-control" id="phone" name="phone"
placeholder="Номер телефона" value="<?php echo $user_data['phone'] ?>"
autocomplete="off">
</div>
<div class="form-group">
    <label for="gender">Пол</label>
    <div class="radio">
        <label>
            <input type="radio" name="gender" id="male" value="1" <?php
if($user_data['gender'] == 1) {
    echo "checked";
} ?>>
            Мужчина
        </label>
        <label>
            <input type="radio" name="gender" id="female" value="2" <?php
if($user_data['gender'] == 2) {
    echo "checked";
} ?>>
            Женщина
        </label>
    </div>
</div>
```

</label>

## *Продолжение приложения А*

</div>

</div>

<div class="form-group">

<div class="alert alert-info alert-dismissible" role="alert">

<button type="button" class="close" data-dismiss="alert" aria-label="Close"><span aria-hidden="true">&times;</span></button>

Оставьте поле пароля пустым если не хотите его менять.

</div>

</div>

<div class="form-group">

<label for="password">Пароль</label>

<input type="password" class="form-control" id="password" name="password" placeholder="Пароль" autocomplete="off">

</div>

<div class="form-group">

<label for="cpassword">Подтвердите пароль</label>

<input type="password" class="form-control" id="cpassword" name="cpassword" placeholder="Подтвердите пароль" autocomplete="off">

</div>

</div>

<div class="box-footer">

<button type="submit" class="btn btn-primary">Сохранить изменения</button>

<a href="<?php echo base\_url('users/') ?>" class="btn btn-warning">Назад</a>

</div>

</form>

</div>

</div>

</div>

</section>

</div>

<script type="text/javascript">

\$(document).ready(function() {

\$("#groups").select2();

\$("#userMainNav").addClass('active');

\$("#manageUserSubNav").addClass('active');

});

</script>

## Продолжение приложения А

*Views/users/index.php:*

```
<div class="content-wrapper">
  <section class="content-header">
    <h1>
      Настройки
    <small>Пользователи</small>
    </h1>
    <ol class="breadcrumb">
      <li><a href="#"><i class="fa fa-dashboard"></i> Главная</a></li>
      <li class="active">Пользователи</li>
    </ol>
  </section>
  <section class="content">
    <div class="row">
      <div class="col-md-12 col-xs-12">
        <?php if($this->session->flashdata('success')): ?>
          <div class="alert alert-success alert-dismissible" role="alert">
            <button type="button" class="close" data-dismiss="alert" aria-
label="Close"><span aria-hidden="true">&times;</span></button>
            <?php echo $this->session->flashdata('success'); ?>
          </div>
          <?php elseif($this->session->flashdata('error')): ?>
            <div class="alert alert-error alert-dismissible" role="alert">
              <button type="button" class="close" data-dismiss="alert" aria-
label="Close"><span aria-hidden="true">&times;</span></button>
              <?php echo $this->session->flashdata('error'); ?>
            </div>
          <?php endif; ?>
          <?php if(in_array('createUser', $user_permission)): ?>
            <a href="<?php echo base_url('users/create') ?>" class="btn btn-
primary">Добавить Пользователя</a>
            <br /> <br />
          <?php endif; ?>
          <div class="box">
            <div class="box-header">
              <h3 class="box-title">Настройки Пользователей</h3>
            </div>
```

<!-- /.box-header -->

*Продолжение приложения А*

```
<div class="box-body">
  <table id="userTable" class="table table-bordered table-striped">
    <thead>
      <tr>
        <th>Логин</th>
        <th>почта</th>
        <th>Имя</th>
        <th>Номер телефона</th>
        <th>Группа</th>
        <?php if(in_array('updateUser', $user_permission) ||
in_array('deleteUser', $user_permission)): ?>
          <th>Действие</th>
        <?php endif; ?>
      </tr>
    </thead>
    <tbody>
      <?php if($user_data): ?>
        <?php foreach ($user_data as $k => $v): ?>
          <tr>
            <td><?php echo $v['user_info']['username']; ?></td>
            <td><?php echo $v['user_info']['email']; ?></td>
            <td><?php echo $v['user_info']['firstname'] . ' ' .
$v['user_info']['lastname']; ?></td>
            <td><?php echo $v['user_info']['phone']; ?></td>
            <td><?php echo $v['user_group']['group_name']; ?></td>
            <?php if(in_array('updateUser', $user_permission) ||
in_array('deleteUser', $user_permission)): ?>
              <td>
                <?php if(in_array('updateUser', $user_permission)): ?>
                  <a href="<?php echo base_url('users/edit/'. $v['user_info']['id'])
?>" class="btn btn-default"><i class="fa fa-edit"></i></a>
                <?php endif; ?>
                <?php if(in_array('deleteUser', $user_permission)): ?>
                  <a
                      href="<?php echo
base_url('users/delete/'. $v['user_info']['id']) ?>" class="btn btn-default"><i
class="fa fa-trash"></i></a>
                <?php endif; ?>
              </td>
            </tr>
          </td>
        </tr>
      </tr>
    </tbody>
  </table>
</div>
```

```

<?php endif; ?>
</tr>

```

*Продолжение приложения А*

```

<?php endforeach ?>
<?php endif; ?>
</tbody>
</table>
</div>
</div>
</div>
</div>
</section>
</div>
<script type="text/javascript">
$(document).ready(function() {
    $('#userTable').DataTable({
        'order' : [],
    });

    $("#userMainNav").addClass('active');
    $("#manageUserSubNav").addClass('active');
});
</script>

```

*Views/users/profile.php:*

```

<div class="content-wrapper">
    <section class="content-header">
        <h1>
            Пользователи
            <small>Профайл</small>
        </h1>
        <ol class="breadcrumb">
            <li><a href="#"><i class="fa fa-dashboard"></i> Главная</a></li>
            <li class="active">Профайл</li>
        </ol>
    </section>
    <section class="content">
        <div class="row">
            <div class="col-md-12 col-xs-12">

```



```

<div class="box">
  <div class="box-header">
    Продолжение приложения А

    <h3 class="box-title">Профайл XXX</h3>
  </div>
  <div class="box-body">
    <table class="table table-bordered table-condensed table-hovered">
      <tr>
        <th>Логин</th>
        <td><?php echo $user_data['username']; ?></td>
      </tr>
      <tr>
        <th>Почта</th>
        <td><?php echo $user_data['email']; ?></td>
      </tr>
      <tr>
        <th>Имя</th>
        <td><?php echo $user_data['firstname']; ?></td>
      </tr>
      <tr>
        <th>Фамилия</th>
        <td><?php echo $user_data['lastname']; ?></td>
      </tr>
      <tr>
        <th>Пол</th>
        <td><?php echo ($user_data['gender'] == 1) ? 'Male' : 'Gender';
?></td>
      </tr>
      <tr>
        <th>Номер телефона</th>
        <td><?php echo $user_data['phone']; ?></td>
      </tr>
      <tr>
        <th>Группа</th>
        <td><span class="label label-info"><?php echo
$user_group['group_name']; ?></span></td>
      </tr>
    </table>
  </div>
</div>

```

```

    </div>
  </div>
</section>

```

### Продолжение приложения А

```

</div>
<script type="text/javascript">
  $(document).ready(function() {
    $('#profileMainNav').addClass('active');
  });
</script>

```

*Views/category/index.php:*

```

<div class="content-wrapper">
  <section class="content-header">
    <h1>
      Настройки
    <small>Категории</small>
    </h1>
    <ol class="breadcrumb">
      <li><a href="#"><i class="fa fa-dashboard"></i> Главная</a></li>
      <li class="active">Категории</li>
    </ol>
  </section>
  <section class="content">
    <div class="row">
      <div class="col-md-12 col-xs-12">
        <div id="messages"></div>
        <?php if(in_array('createCategory', $user_permission)): ?>
          <button class="btn btn-primary" data-toggle="modal" data-
target="#addModal">Добавить Категорию</button>
          <br /> <br />
        <?php endif; ?>
        <div class="box">
          <div class="box-header">
            <h3 class="box-title">Настройки Категории</h3>
          </div>
          <div class="box-body">
            <table id="manageTable" class="table table-bordered table-striped">
              <thead>

```

```

        <tr>
            <th>Название категории</th>
            <th>Состояние</th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <td>
                <?php if(in_array('updateCategory', $user_permission) ||
in_array('deleteCategory', $user_permission)): ?>
                    <th>Действие</th>
                    <?php endif; ?>
                </td>
            </tr>
        </tbody>
    </table>
</div>
</div>
</div>
</div>
</section>
</div>
<?php if(in_array('createCategory', $user_permission)): ?>
<div class="modal fade" tabindex="-1" role="dialog" id="addModal">
    <div class="modal-dialog" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <button type="button" class="close" data-dismiss="modal" aria-
label="Close"><span aria-hidden="true">&times;</span></button>
                <h4 class="modal-title">Добавить Категорию</h4>
            </div>
            <form role="form" action="<?php echo base_url('category/create') ?>"
method="post" id="createForm">
                <div class="modal-body">
                    <div class="form-group">
                        <label for="brand_name">Название категории</label>
                        <input type="text" class="form-control" id="category_name"
name="category_name" placeholder="Введите название категории"
autocomplete="off">
                    </div>
                    <div class="form-group">
                        <label for="active">Состояние</label>
                        <select class="form-control" id="active" name="active">

```

```

        <option value="1">Активна</option>
        <option value="2">Деактивирована</option>
    </select>

```

*Продолжение приложения А*

```

    </div>
</div>
<div class="modal-footer">
    <button type="button" class="btn btn-default" data-
dismiss="modal">Закрыть</button>
    <button type="submit" class="btn btn-primary">Сохранить
изменения</button>
</div>
</form>
</div>
</div>
</div>
<?php endif; ?>
<?php if(in_array('updateCategory', $user_permission)): ?>
<div class="modal fade" tabindex="-1" role="dialog" id="editModal">
    <div class="modal-dialog" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <button type="button" class="close" data-dismiss="modal" aria-
label="Close"><span aria-hidden="true">&times;</span></button>
                <h4 class="modal-title">Изменить Категорию</h4>
            </div>
            <form role="form" action="<?php echo base_url('category/update') ?>"
method="post" id="updateForm">
                <div class="modal-body">
                    <div id="messages"></div>
                    <div class="form-group">
                        <label for="brand_name">Название Категории</label>
                        <input type="text" class="form-control" id="edit_category_name"
name="edit_category_name" placeholder="Введите название категории"
autocomplete="off">
                    </div>
                    <div class="form-group">
                        <label for="active">Состояние</label>
                        <select class="form-control" id="edit_active" name="edit_active">
                            <option value="1">Активна</option>

```

```

        <option value="2">Деактивирована</option>
    </select>
</div>

```

### *Продолжение приложения А*

```

</div>
<div class="modal-footer">
    <button type="button" class="btn btn-default" data-
dismiss="modal">Закрыть</button>
    <button type="submit" class="btn btn-primary">Сохранить
изменения</button>
</div>
</form>
</div>
</div>
</div>
<?php endif; ?>
<?php if(in_array('deleteCategory', $user_permission)): ?>
<div class="modal fade" tabindex="-1" role="dialog" id="removeModal">
    <div class="modal-dialog" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <button type="button" class="close" data-dismiss="modal" aria-
label="Close"><span aria-hidden="true">&times;</span></button>
                <h4 class="modal-title">Удалить Категорию</h4>
            </div>
            <form role="form" action="<?php echo base_url('category/remove') ?>"
method="post" id="removeForm">
                <div class="modal-body">
                    <p>Вы действительно хотите удалить?</p>
                </div>
                <div class="modal-footer">
                    <button type="button" class="btn btn-default" data-
dismiss="modal">Закрыть</button>
                    <button type="submit" class="btn btn-primary">Сохранить
изменения</button>
                </div>
            </form>
        </div>
    </div>
</div>
</div>
</div>

```

```

<?php endif; ?>
<script type="text/javascript">
var manageTable;

```

*Продолжение приложения А*

```

var base_url = "<?php echo base_url(); ?>";
$(document).ready(function() {
    $('#categoryMainNav').addClass('active');
    // initialize the datatable
    manageTable = $('#manageTable').DataTable({
        'ajax': base_url + 'category/fetchCategoryData',
        'order': []
    });
    // submit the create form
    $("#createForm").unbind('submit').on('submit', function() {
        var form = $(this);
        // remove the text-danger
        $(".text-danger").remove();
        $.ajax({
            url: form.attr('action'),
            type: form.attr('method'),
            data: form.serialize(), // /converting the form data into array and sending it to
server
            dataType: 'json',
            success: function(response) {
                manageTable.ajax.reload(null, false);
                if(response.success === true) {
                    $("#messages").html('<div class="alert alert-success alert-dismissible"
role="alert">'+
                        '<button type="button" class="close" data-dismiss="alert" aria-
label="Close"><span aria-hidden="true">&times;</span></button>'+
                        '<strong> <span class="glyphicon glyphicon-ok-sign"></span>
</strong>'+response.messages+
                        '</div>');
                    // hide the modal
                    $("#addModal").modal('hide');
                    // reset the form
                    $("#createForm")[0].reset();
                    $("#createForm .form-group").removeClass('has-error').removeClass('has-
success');
                } else {

```

```

    if(response.messages instanceof Object) {
        $.each(response.messages, function(index, value) {
            var id = $("#"+index);
            id.closest('.form-group')
                .removeClass('has-error')

                Продолжение приложения А

                .removeClass('has-success')
                .addClass(value.length > 0 ? 'has-error' : 'has-success');
            id.after(value);
        });
    } else {
        $("#messages").html('<div class="alert alert-warning alert-dismissible"
role="alert">'+
            '<button type="button" class="close" data-dismiss="alert" aria-
label="Close"><span aria-hidden="true">&times;</span></button>'+
            '<strong><span class="glyphicon glyphicon-exclamation-sign"></span>
</strong>'+response.messages+
            '</div>');
    }
}
});
return false;
});
});
// edit function
function editFunc(id)
{
    $.ajax({
        url: base_url + 'category/fetchCategoryDataById/'+id,
        type: 'post',
        dataType: 'json',
        success:function(response) {
            $("#edit_category_name").val(response.name);
            $("#edit_active").val(response.active);
            // submit the edit from
            $("#updateForm").unbind('submit').bind('submit', function() {
                var form = $(this);
                // remove the text-danger
                $(".text-danger").remove();

```

```
$.ajax({
  url: form.attr('action') + '/' + id,
  type: form.attr('method'),
```

*Продолжение приложения А*

```
    data: form.serialize(), // converting the form data into array and sending it
    to server
    dataType: 'json',
    success: function(response) {
      manageTable.ajax.reload(null, false);
      if(response.success === true) {
        $("#messages").html('<div class="alert alert-success alert-dismissible"
role="alert">'+
          '<button type="button" class="close" data-dismiss="alert" aria-
label="Close"><span aria-hidden="true">&times;</span></button>'+
          '<strong> <span class="glyphicon glyphicon-ok-sign"></span>
</strong>'+response.messages+
          '</div>');
        // hide the modal
        $("#editModal").modal('hide');
        // reset the form
        $("#updateForm .form-group").removeClass('has-
error').removeClass('has-success');
      } else {
        if(response.messages instanceof Object) {
          $.each(response.messages, function(index, value) {
            var id = $("#"+index);
            id.closest('.form-group')
              .removeClass('has-error')
              .removeClass('has-success')
              .addClass(value.length > 0 ? 'has-error' : 'has-success');
            id.after(value);
          });
        } else {
          $("#messages").html('<div class="alert alert-warning alert-dismissible"
role="alert">'+
            '<button type="button" class="close" data-dismiss="alert" aria-
label="Close"><span aria-hidden="true">&times;</span></button>'+
            '<strong> <span class="glyphicon glyphicon-exclamation-
sign"></span> </strong>'+response.messages+
            '</div>');
```



```

    }
  }
}

```

### *Продолжение приложения А*

```

    });
    return false;
  });
}
});
}
function removeFunc(id)
{
  if(id) {
    $("#removeForm").on('submit', function() {
      var form = $(this);
      $(".text-danger").remove();
      $.ajax({
        url: form.attr('action'),
        type: form.attr('method'),
        data: { category_id:id },
        dataType: 'json',
        success:function(response) {
          manageTable.ajax.reload(null, false);
          // hide the modal
          $("#removeModal").modal('hide');
          if(response.success === true) {
            $("#messages").html('<div class="alert alert-success alert-dismissible"
role="alert">'+
              '<button type="button" class="close" data-dismiss="alert" aria-
label="Close"><span aria-hidden="true">&times;</span></button>'+
              '<strong> <span class="glyphicon glyphicon-ok-sign"></span>
</strong>'+response.messages+
              '</div>');
          } else {
            $("#messages").html('<div class="alert alert-warning alert-dismissible"
role="alert">'+
              '<button type="button" class="close" data-dismiss="alert" aria-
label="Close"><span aria-hidden="true">&times;</span></button>'+
              '<strong> <span class="glyphicon glyphicon-exclamation-sign"></span>
</strong>'+response.messages+

```

```

        '</div>');
    }
}

```

### *Продолжение приложения А*

```

    });
    return false;
  });
}
}
</script>

```

### *Controllers/category.php*

```

<?php
class Category extends Admin_Controller
{
    public function __construct()
    {
        parent::__construct();
        $this->not_logged_in();
        $this->data['page_title'] = 'Пользователи';
        $this->load->model('model_category');
    }
    public function index()
    {
        $this->render_template('category/index', $this->data);
    }
    public function fetchCategoryData()
    {
        $result = array('data' => array());
        $data = $this->model_category->getCategoryData();
        foreach ($data as $key => $value) {
            // button
            $buttons = "";
            if(in_array('updateCategory', $this->permission)) {
                $buttons = '<button type="button" class="btn btn-default"
onclick="editFunc('.$value['id'].')" data-toggle="modal" data-
target="#editModal"><i class="fa fa-pencil"></i></button>';
            }
            if(in_array('deleteCategory', $this->permission)) {

```

```

$buttons .= ' <button type="button" class="btn btn-default"
onclick="removeFunc('.$value['id'].')"          data-toggle="modal"          data-
target="#removeModal"><i class="fa fa-trash"></i></button>';

```

*Продолжение приложения А*

```

    }
    $status = ($value['active'] == 1) ? '<span class="label label-
success">Active</span>' : '<span class="label label-warning">Inactive</span>';
    $result['data'][$key] = array(
        $value['name'],
        $status,
        $buttons
    );
} // foreach
echo json_encode($result);
}
public function create()
{
    // if(!in_array('createCategory', $this->permission)) {
    //     redirect('dashboard', 'refresh');
    // }
    $response = array();
    $this->form_validation->set_rules('category_name', 'Category name',
'trim|required');
    $this->form_validation->set_rules('active', 'Active', 'trim|required');
    $this->form_validation->set_error_delimiters('<p          class="text-
danger">','</p>');
    if ($this->form_validation->run() == TRUE) {
        $data = array(
            'name' => $this->input->post('category_name'),
            'active' => $this->input->post('active'),
        );
        $create = $this->model_category->create($data);
        if ($create == true) {
            $response['success'] = true;
            $response['messages'] = 'Успешно создано';
        }
        else {
            $response['success'] = false;
            $response['messages'] = 'Ошибка в базе данных во время
создания';

```

```

    }
}
else {

```

*Продолжение приложения А*

```

    $response['success'] = false;
    foreach ($_POST as $key => $value) {
        $response['messages'][$key] = form_error($key);
    }
}
echo json_encode($response);
}
public function fetchCategoryDataById($id = null)
{
    if($id) {
        $data = $this->model_category->getCategoryData($id);
        echo json_encode($data);
    }
}

public function update($id)
{
    // if(!in_array('updateStore', $this->permission)) {
    //     redirect('dashboard', 'refresh');
    // }
    $response = array();
    if($id) {
        $this->form_validation->set_rules('edit_category_name',
'Category name', 'trim|required');
        $this->form_validation->set_rules('edit_active',          'Active',
'trim|required');
        $this->form_validation->set_error_delimiters('<p      class="text-
danger">','</p>');
        if ($this->form_validation->run() == TRUE) {
            $data = array(
                'name' => $this->input->post('edit_category_name'),
                'active' => $this->input->post('edit_active'),
            );
            $update = $this->model_category->update($id, $data);
            if($update == true) {
                $response['success'] = true;

```

```

        $response['messages'] = 'Успешно создано';
    }
    else {
        Продолжение приложения А

        $response['success'] = false;
        $response['messages'] = 'Ошибка в базе данных во время
создания';
    }
}
else {
    $response['success'] = false;
    foreach ($_POST as $key => $value) {
        $response['messages'][$key] = form_error($key);
    }
}
}
else {
    $response['success'] = false;
    $response['messages'] = 'Ошибка, пожалуйста перезагрузите
страницу!';
}
echo json_encode($response);
}
public function remove()
{
    // if(!in_array('deleteStore', $this->permission)) {
    //     redirect('dashboard', 'refresh');
    // }
    $category_id = $this->input->post('category_id');
    $response = array();
    if($category_id) {
        $delete = $this->model_category->remove($category_id);
        if($delete == true) {
            $response['success'] = true;
            $response['messages'] = "Успешно удалено";
        }
        else {
            $response['success'] = false;
            $response['messages'] = "Ошибка в базе данных во время
удаления";

```

```

    }
}
else {

```

*Продолжение приложения А*

```

    $response['success'] = false;
    $response['messages'] = "Перезагрузите страницу опять!";
}
echo json_encode($response);
}
}
Models/Model_category.php

```

<?php

```

class Model_category extends CI_Model
{
    public function __construct()
    {
        parent::__construct();
    }
    public function getCategoryData($id = null)
    {
        if($id) {
            $sql = "SELECT * FROM category WHERE id = ?";
            $query = $this->db->query($sql, array($id));
            return $query->row_array();
        }
        $sql = "SELECT * FROM category ORDER BY id DESC";
        $query = $this->db->query($sql);
        return $query->result_array();
    }
    public function create($data = array())
    {
        if($data) {
            $create = $this->db->insert('category', $data);
            return ($create == true) ? true : false;
        }
    }
    public function update($id = null, $data = array())
    {

```

```

        if($id && $data) {
            $this->db->where('id', $id);
            $update = $this->db->update('category', $data);
            Продолжение приложения А

            return ($update == true) ? true : false;
        }
    }

    public function remove($id = null)
    {
        if($id) {
            $this->db->where('id', $id);
            $delete = $this->db->delete('category');
            return ($delete == true) ? true : false;
        }
    }

    public function getActiveCategory()
    {
        $sql = "SELECT * FROM category WHERE active = ?";
        $query = $this->db->query($sql, array(1));
        return $query->result_array();
    }
}

```

## Приложение Б

### Акт внедрения

Утверждаю  
Директор ТОО «Шустов»  
Кадирбеков С.А.  
«11» мая 2020г.

### АКТ

внедрения программы «Le Volée»

Разработанная студентом 4-курса НАО АУЭС гр.ИС-16-2 Ануарбековым Ш.Н. программа «Le Volée» была передана в эксплуатацию в ТОО «Шустов» в мае 2020 года для использования в качестве программного комплекса.

Назначение программы:

- Автоматизация работы по организации Шустов;
- Обеспечение возможности сбора аналитических данных.

Показатели эффективности программы:

- Сокращение времени на отчетность предприятия;
- Оптимизация процессов отслеживания заказов;
- Сокращение издержек за счет автоматизации номенклатуры, и вывода

статистики;

- Экономический эффект.

Дополнительные показатели эффективности, достигаемые за счет внедрения:

- Сокращение сроков подготовки отчетов на 69%.
- Общее снижение количества ошибок на 23 %.

Директор ТОО «Шустов»

Кадирбеков С.А.

Исполнитель

Ануарбеков Ш.Н.

