

МИНИСТЕРСТВО ОБРАЗОВАНИЯ ОРЕНБУРГСКОЙ ОБЛАСТИ
ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ СРЕДНЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«ОРЕНБУРГСКИЙ КОЛЛЕДЖ ЭКОНОМИКИ И ИНФОРМАТИКИ»
(ГАПОУ СПО ОКЭИ)

КУРСОВОЙ ПРОЕКТ

ОКЭИ 09.02.07. 4323. 19 ПЗ

(код документа)

«Разработка веб-сайта для автоматизации процесса продажи настольных игр»

Количество листов 31
Дата готовности 28.12.23
Руководитель Гукасян А.Д.
Разработал Скороспелова А.В.
Защищен _____ с оценкой _____
(дата)

Оренбург 2023

Содержание

Введение.....	3
1 Анализ предметной области.....	6
2 Проектирование приложения.....	8
3 Разработка программного обеспечения.....	11
3.1 Описание технологического стека разработки.....	11
3.2 Описание алгоритма работы.....	14
3.3 Описание интерфейса пользователя.....	19
4 Тестирование приложения.....	22
4.1 План тестирования.....	22
4.2 Оценка результатов проведения тестирования.....	24
Заключение.....	26
Список использованных источников.....	28
Приложение А (обязательное) Диаграмма прецедентов.....	29
Приложение Б (обязательное) Диаграмма деятельности.....	30
Приложение В (обязательное) Диаграмма классов.....	31

					<i>ОКЭИ 09.02.07. 4323 19 ПЗ</i>			
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>				
Разраб.		<i>Скораспелова А.В.</i>						
Провер.		<i>Гцкасян А.Д.</i>						
Реценз.								
Н. Контр.								
Утверд.								
						<i>Лит.</i>	<i>Лист</i>	<i>Листов</i>
							<i>2</i>	<i>31</i>
						<i>Отделение — очное гр.4бд1</i>		

Введение

На современном этапе развития общества информация представляет собой важнейший ресурс, необходимый людям для принятия важных решений, выполнения поставленных перед ними задач, а также для познания окружающего их мира. От качества полученных данных, их актуальности и доступности напрямую зависит эффективность работы, скорость принятия каких-либо решений и, в целом, качество жизни каждого человека.

В связи с тем, какую роль в настоящее время играет информация, огромное внимание уделяется развитию информационных технологий, наиболее известным примером которых является сеть Интернет, ставшая неотъемлемой частью жизни людей 21 века. Данный ресурс представляет собой действующую по всему миру информационную систему, в которую объединены сети меньшего масштаба. Количество устройств, которые могут быть подключены к этой глобальной сети, практически неограниченно. С каждым годом число пользователей становится только больше. Сеть Интернет является самым быстроразвивающимся средством передачи информации за всю историю человечества.

В ходе развития информационных технологий появляется всё больше разнообразных Интернет-ресурсов, каждый из которых позволяет людям тем или иным образом удовлетворить свои потребности в какой-либо сфере их жизнедеятельности. Человеку предоставляется шанс найти необходимую информацию быстро и легко, ведь достаточно иметь для этого любое электронное устройство, обладающее доступом к сети Интернет. Возможности данного ресурса не ограничиваются лишь поиском каких-либо данных, глобальная сеть также предоставляет огромное количество инструментов, которые люди могут использовать для осуществления автоматизации каких-либо сфер их трудовой деятельности, а иногда и позволяют полностью заменить привычный им рабочий уклад. Именно Интернет в настоящий период времени способен быстро и массово донести до общества информацию, что является путём к успеху в области современного бизнеса.

К инструментам, которые были описаны выше, можно отнести веб-сайты, один из которых будет разработан в ходе данной работы. Веб сайт – это интерактивная веб-страница или набор страниц, которые содержат информацию, изображения, видео и другие мультимедийные элементы, доступные через Интернет. Существуют следующие виды веб-сайтов, каждый из которых имеет своё назначение и структуру:

- сайт-визитка. Данный вид представляет собой небольшой справочник о фирме, в котором содержится краткая основная информация о деятельности компании и контакты. Всё это располагается на одной странице;

- корпоративный. Основной задачей данного вида является повышение прибыли бизнеса. С помощью корпоративного сайта клиенты могут просматривать предоставляемые компанией товары или услуги, совершать заказы

и тому подобное. Этот вид сайта может иметь различное наполнение, в зависимости от организации, для которой он создаётся;

- промо-сайт. Такой веб-сайт используется в тех случаях, когда запускается промо-компания, конкурс или акция. По структуре он напоминает рекламный буклет или проспект, оформляется ярко и красочно, чтобы обязательно привлечь пользователей;

- сайт-витрина. Это оптимальный вариант для тех компаний, которые продают один товар или группу товаров одного типа. На таком сайте товары подробно описывают, размещают их фото и стоимость. Каталог организовывается таким образом, чтобы пользователь мог найти нужную позицию, совершив минимальное количество действий;

- интернет-магазин. Здесь можно не только просмотреть товар, но и заказать его, а на некоторых ресурсах предоставлена возможность сразу произвести оплату. Такой ресурс должен быть максимально простым и удобным, чтобы пользователь мог совершить покупку без каких-либо препятствий.

Разработка, которую планируется осуществить в ходе текущей работы, относится к категории интернет-магазинов, так как на веб-сайте будет предоставлена информация о доступных товарах, в данном случае о настольных играх, а в дальнейшем пользователи смогут выбрать нужные им позиции и оформить заказ.

Тема данной работы является актуальной, так как в настоящее время веб-сайты представляют собой один из самых распространённых ресурсов, предназначенных для быстрого, удобного и эффективного удовлетворения различных потребностей людей.

Целью данной работы является разработка веб-сайта для автоматизации продажи настольных игр «BGUniverse». В рамках разработки предполагается создание ресурса, на котором будет размещена вся необходимая для пользователей информация, которая позволит им ознакомиться с ассортиментом товаров, а также осуществить их заказ в режиме онлайн.

Исходя из поставленной цели, можно выделить следующие задачи, которые необходимо выполнить для её осуществления:

- создание прототипа и макета дизайна веб-сайта;
- создание сервера для хранения информации;
- адаптивная разработка страниц веб-сайта;
- создание системы авторизации;
- обеспечение возможности оформления заказа в режиме онлайн;
- осуществление тестирования готового веб-сайта.

Каждая из выдвинутых задач должна быть выполнена качественно, в рамках определённого для неё срока, для того, чтобы в дальнейшем стало возможным создать и ввести в эксплуатацию правильно функционирующую систему, которая будет обладать не только визуально приятным и простым, но и удобным и работоспособным интерфейсом. Это очень важно для удовлетворения потребностей пользователей, что и является одним из главных условий, которые выдвигаются перед разработчиками любого Интернет-ресурса.

Объектом текущей работы является оформление заказа настольных игр онлайн с помощью веб-сайта.

Предметом работы является разработка веб-сайта для автоматизации продажи настольных игр «BGUniverse».

Данная разработка является практически значимой, так как она позволит компании расширить свой бизнес, предоставив людям удобный ресурс, на котором будет размещена в доступном виде информация об организации и товарах, которые она предоставляет. Также, пользователи смогут оформлять заказ понравившихся им настольных игр с помощью соответствующего раздела веб-сайта, где они быстро и легко осуществят покупку. Создание данного продукта позволит привлечь больше клиентов и предоставить им надёжный и удобный ресурс, который поможет им приобрести средство, позволяющее скрасить своё свободное время.

Для реализации проекта понадобится компьютер, имеющий любую из актуальных операционных систем и обладающий ресурсами, которые позволят внедрять все необходимые программы, вести разработку веб-сайта, а также произвести его тестирование. В перечень программ, которые потребуется установить для того, чтобы осуществить создание системы, входят следующие:

- Visual Studio Code - редактор исходного кода, который работает на персональном компьютере и доступен для таких операционных систем, как Windows, macOS и Linux. Данная программа будет использоваться для написания кода, отвечающего за внешний вид и работоспособность веб-сайта;

- Node.js - кроссплатформенная среда исполнения с открытым исходным кодом, которая позволяет разработчикам создавать всевозможные серверные инструменты и приложения используя язык JavaScript;

- PostgreSQL - свободная объектно-реляционная система управления базами данных. Потребуется для создания сервера, на котором будут храниться все необходимые данные о товарах и пользователях веб-сайта;

- Postman - это сервис для создания, тестирования, документирования, публикации и обслуживания API. В данной работе он требуется для тестирования работы запросов к серверу, чтобы своевременно исправить ошибки, если таковые имеются, и внедрить в код программы корректный вариант написания запросов.

В дальнейшем функционал веб-сайта может быть расширен путём внедрения в него различных инструментов, которые помогут сделать взаимодействие с ресурсом ещё более удобным и эффективным. Примерами таких нововведений могут послужить фильтрация товаров, поиск по всем разделам веб-сервиса, а также совершенствование системы авторизации и оформления заказа.

1 Анализ предметной области

В современном мире информация играет всё более значимую роль в жизни каждого человека. В настоящий момент общество достигло этапа, на котором наблюдается значительный прогресс в области информационных технологий, предлагающих людям колоссальное количество новых возможностей.

Одним из таких технологических достижений является сеть Интернет, которая представляет собой известное всем средство, которое расширяет возможности людей и во многом упрощает их жизнь, охватывая все её аспекты. Существует множество ресурсов, содержащих информацию о разных областях жизнедеятельности, и их число растёт с каждым днём.

Интернет позволяет людям общаться на расстоянии, находить друзей и партнёров по работе, а также обмениваться информацией на различных уровнях. Он также предоставляет доступ к огромному количеству знаний, образовательных ресурсов и развлекательному контенту. С помощью интернета можно совершать покупки, заказывать услуги, искать работу, получать образование и многое другое.

Темой данной работы является разработка веб-сайта для автоматизации продажи настольных игр «BGUniverse». Создание этого ресурса поможет расширить выбор настольных игр для людей, которые выбирают данный вид развлечений в качестве своего досуга, а также предпочитают быстрый и удобный доступ к различным разновидностям игр с дальнейшей возможностью оформления их заказа. Пользователям будущего веб-сайта будет предоставлен шанс не только просмотреть и выбрать понравившийся им товар, используя любое удобное для них устройство с доступом к сети Интернет, но и заказать его с доставкой на дом.

С появлением различных веб-сайтов стало возможным расширение деятельности различных организаций, автоматизация процессов, которые раньше выполнялись рабочими, а также упрощение жизни людей в различных сферах. Тем не менее, даже в таких условиях существует проблема нехватки действительно удобных и доступных ресурсов, которые помогут человеку качественно удовлетворить свои потребности.

Настольные игры существуют ещё с древних времён, а в данное время рынок настольных игр является одним из самых быстрорастущих в мире. С каждым годом количество их разновидностей растёт, появляется всё больше организаций, каждая из которых вносит свой вклад в расширение данной области. Целевая аудитория настольных игр включает в себя людей разных возрастов, статусов и с различными увлечениями. В связи с этим появляется необходимость расширения диапазона их распространения, что поможет не только самим компаниям, производящим эти игры, но и их клиентам.

Также стоит отметить, что настольные игры становятся все более популярными среди молодёжи, что способствует увеличению их продаж. Это связано с тем, что современные молодые люди предпочитают проводить

свободное время за интересными и увлекательными занятиями, которые не только приносят удовольствие, но и развивают их.

Помочь в расширении могут веб-сайты, один из которых и будет разработан в ходе данной работы. Для того, чтобы более понятно обосновать, в чём заключается проблема в сфере настольных игр, и пояснить, что позволит осуществить разработка специализированного ресурса, требуется рассмотреть эту область более подробно.

Наиболее распространённой из проблем, которые возникают при желании людей приобрести настольные игры, является наличие в их городах маленького количества физических магазинов, предоставляющих действительно большой ассортимент качественных и уникальных товаров. Это препятствует выбору интересных для человека игр, что приводит к его неудовлетворённости предоставленными услугами.

Ещё одной проблемой является высокая стоимость некоторых игр. Не все покупатели готовы тратить большие суммы на покупку настольной игры, особенно если они не уверены, что она им понравится.

Для того, чтобы решить проблемы, описанные выше, и существуют веб-сайты. Они позволяют не только предоставить пользователю намного более широкий выбор товаров, но и даёт им возможность из любого удобного им места, дом это или какое-либо заведение, получить доступ к этим товарам, выбрать нужные и оформить доставку.

Выполняемая разработка будет направлена на то, чтобы сделать процесс покупки настольных игр более удобным и доступным для людей. Воспользоваться веб-сайтом сможет любой человек, который обладает хотя бы базовыми знаниями в области использования современных устройств, а также различных Интернет-ресурсов. Пользователям будет предоставлена возможность просмотреть все товары, которые расположены в соответствующих разделах системы, узнать о них более подробную информацию, а также добавить в корзину заинтересовавшие их игры и приобрести их онлайн. Администратору веб-сайта, в свою очередь, будет доступен удобный ресурс для редактирования информации о товарах, их добавления и удаления, а также для просмотра данных об авторизованных пользователях и их заказах.

Подводя итоги, можно сказать, что разработка веб-сайта для автоматизации продажи настольных игр «BGUniverse» позволит не только решить проблему для людей, которые увлекаются настольными играми, но и имеют возможности найти что-то качественное и интересное на имеющихся ресурсах или в магазинах, но и расширить деятельность компаний, которые занимаются производством и поставкой данной категории товаров. Также, этот продукт поможет сделать работу сотрудника, отвечающего за управление заказами клиентов, более простой и удобной, так как ему будет доступна система, в которой будут предоставлены все необходимые ему данные, что позволит качественно и эффективно выполнить требуемые действия.

2 Проектирование приложения

Прежде, чем приступить к непосредственно разработке программного продукта, требуется наиболее точно и подробно разобраться с тем, какие требования к нему выдвигаются. Для того, чтобы сделать этот процесс результативным, необходимо рассмотреть требования к веб-сайту с двух сторон: функциональной и нефункциональной.

Функциональные требования представляют собой основные функциональные возможности или особенности, которыми должна обладать система, чтобы обеспечить достижение поставленной перед ней цели. Другими словами, эти требования определяют, какие действия должен осуществлять программный продукт. Они служат для описания взаимодействия между ресурсом и его пользователями, а также поведения программы в различных условиях.

Функциональные требования обычно обладают следующими характеристиками:

- подробное и конкретное описание, которое однозначно определяют, какие именно функции должна выполнять система, а также её входы и выходы;
- тестирование и проверка, выполняемые для того, чтобы гарантировать, что программный продукт корректно функционирует, в соответствии с его назначением;
- тесная связь с потребностями и ожиданиями пользователей, чтобы обеспечить качественное и полное удовлетворение их потребностей;
- способность меняться в ходе проекта по мере расширения потребностей пользователей или организации.

К веб-сайту настольных игр «BGUniverse» выдвигаются следующие функциональные требования:

- осуществление возможности авторизации для пользователей. Каждый из людей, которые взаимодействуют с веб-сайтом, должен иметь возможность создать свой личный аккаунт с дальнейшим доступом к своему профилю после ввода соответствующих данных. Все действия требуется выполнять при использовании непосредственно разрабатываемого ресурса, на котором будут размещены необходимые для этого страницы;
- предоставление доступа к корзине товаров после авторизации. После того, как пользователь введёт корректные данные, отвечающие за его вход в свой аккаунт, ему необходимо дать возможность добавлять понравившиеся товары в специальный раздел, роль которого в данной системе будет играть корзина товаров. На каждой из позиций требуется разместить элемент, при взаимодействии с которым выбранная настольная игра будет добавлена в соответствующий раздел, который сохраняется в аккаунте каждого конкретного пользователя для дальнейшего осуществления заказа;
- автоматизация процесса осуществления заказа. Пользователи будущей системы должны иметь возможность оформить заказ тех товаров, которые были добавлены ими в корзину. Это действие требуется выполнять посредством

нажатия на соответствующий элемент, который будет представлен в виде кнопки, размещённой на странице корзины пользователей. При взаимодействии с кнопкой должно появляться всплывающее окно, в которое клиент вносит все необходимые данные, которые затем будут отправляться на сервер для дальнейшей обработки и осуществления доставки товаров.

Нефункциональные требования выступают как дополнение к функциональным требованиям, определяя, каким образом информационная система должна выполнять требуемые функции. Они служат для описания качества, характеристик и ограничений ресурса, а не её конкретных функций.

Нефункциональные требования обладают следующими характеристиками:

- сосредоточенность на качественных аспектах, таких как производительность, надёжность и безопасность разрабатываемого программного продукта;
- применение к системе в целом и оказание воздействия на её общее поведение;
- большая стабильность на протяжении всего жизненного цикла проекта, при этом изменения происходят реже по сравнению с функциональными требованиями;
- возможность измерить и протестировать требования.

К веб-сайту настольных игр «BGUniverse» выдвигаются следующие нефункциональные требования:

- лаконичный дизайн, соответствующий тематике системы. Для того, чтобы пользователю было приятно и комфортно взаимодействовать с веб-сайтом, требуется подобрать такую цветовую гамму, которая не только будет вызывать ассоциацию с тематикой разработки, но и выглядеть просто и, в то же время, интересно. Для этого нужно выбрать в качестве основного цвета спокойный, неброский, и дополнить его акцентным цветом, который придаст дизайну особенный вид;
- адаптивный дизайн. Помимо того, что веб-сайт должен выглядеть приятно для пользователя, от него также требуется быть адаптированным под все размеры устройств, от экрана компьютера до небольшого смартфона. Это необходимо для того, чтобы обеспечить более удобное использование ресурса, так как он будет доступен клиентам в любое время и в любом месте, что повысит количество пользователей;
- конфиденциальность данных пользователей. В разрабатываемой системе будет возможность авторизации, которая предполагает внесение пользователями своей личной информации. Для того, чтобы клиенты могли безопасно эксплуатировать ресурс и не бояться за сохранность своих данных, необходимо обеспечить их конфиденциальность. Это возможно сделать путём разграничения доступа к данным, в зависимости от роли пользователя, а также различными методами шифрования данных, к которым относится хеширование паролей и использование уникальных токенов для клиентов сервиса;
- расширяемость. В будущей системе должна быть предусмотрена возможность расширения, которая позволит легко и быстро добавлять новые

функции и возможности в уже работающий продукт без необходимости полностью его переписывать. Это необходимо для того, чтобы быстро реагировать на изменения в требованиях к системе, а также для упрощения процесса разработки и тестирования новых функций;

- надёжность. Будущий веб-сайт должен работать стабильно и корректно для того, чтобы пользователи могли в любой момент им воспользоваться. Требуется предусмотреть пути решения проблем, которые могут возникнуть при функционировании ресурса, чтобы обеспечить его надёжность и устойчивость.

Для того, чтобы более наглядно продемонстрировать то, каким образом будет функционировать веб-сайт, требуется построить диаграмму вариантов использования (диаграмму прецедентов), которая позволит определить границы системы, основные функции и их последовательность, а также выявить потенциальных участников системы (актёров) и их взаимодействие с системой. Данный вид модели включает в себя следующие элементы:

- актёры (или действующие лица) представляют собой сущности, которые взаимодействуют с системой. В качестве данного элемента могут выступать люди, организации, устройства или другие системы;
- прецеденты (или варианты использования) это функциональные требования или сценарии использования для системы. Каждый прецедент представляет собой определённый способ использования системы актёром;
- отношения между актёрами и прецедентами, такие как «включает» и «расширяет».

Диаграмма прецедентов для разрабатываемого ресурса представлена в Приложении А (Рисунок А.1). На модели представлено 2 актёра: администратор и пользователь. Это демонстрирует разделение ролей на веб-сайте.

Администратор осуществляет вход в систему, после чего ему открывается доступ к управлению данными веб-сайта и просмотру данных пользователей. Управление данными включает в себя редактирование товаров, а также их добавление и удаление. Просмотр данных пользователей включает в себя доступ к списку имеющихся авторизованных клиентов и к их заказам.

Пользователь может просматривать информацию, которая представлена на веб-сайте. Ему доступны все страницы системы, кроме тех, которые открыты только для авторизованных пользователей. Также, пользователь может осуществить вход в систему, после чего ему так же, как и неавторизованному, доступен просмотр информации, а также оформление заказа, которое расширяется добавлением товара в корзину.

Таким образом, для данной работы были выдвинуты функциональные и нефункциональные требования, каждое из которых подробно и конкретно расписано. Также, была построена диаграмма прецедентов, которая позволила визуализировать все механизмы взаимодействия пользователей с будущей системой, что в дальнейшем поможет более качественно разработать веб-сайт, соответствующий всем требованиям.

3 Разработка программного обеспечения

3.1 Описание технологического стека разработки

Перед тем, как приступить к разработке веб-сайта, требуется определиться с тем, какие именно технологии будут применяться при написании программного кода. Для того, чтобы сделать этот выбор, нужно более подробно ознакомиться с такими понятиями, как стек технологий, фреймворк и арі, чтобы иметь представление о том, за что отвечает каждое из них.

Стек технологий это набор технологий, которые используются для разработки программного обеспечения, включающий в себя различные языки программирования, фреймворки, библиотеки, сторонние программы и другие инструменты, необходимые при создании программного продукта. Для того, чтобы качественно и в полном объёме произвести планируемую разработку, требуется подобрать такие технологии, которые предоставят весь необходимый перечень инструментов.

Фреймворк представляет собой набор правил, шаблонов и инструментов, которые используются для построения продуктов или процессов как в программировании, так и в других областях. Он предлагает готовые решения и методики, которые можно адаптировать и применять в различных проектах и областях деятельности.

Как правило, любой веб-сайт, который разрабатывается с целью автоматизации какого-либо процесса, состоит из двух частей: front-end (клиентская) и back-end (серверная) части, каждая из которых разрабатывается с применением разных технологий. Далее будет приведено более подробное объяснение того, что из себя представляют перечисленные выше компоненты.

Front-end отвечает за ту часть веб-сайта, которую видит пользователь, и с помощью которой он взаимодействует с программным продуктом. То есть, к данному блоку системы относятся все веб-страницы, на которые попадает пользователь при посещении какого-либо Интернет-ресурса. При написании программного кода части front-end требуется реализовать тот дизайн, который был создан для последующего его внедрения в автоматизированную систему.

Back-end обеспечивает правильное функционирование приложения, связанное с запросами пользователей. Back-end создаётся с помощью языков программирования, различных фреймворков и других инструментов, служащих для этих целей. В данной части веб-сайта хранится вся информация, которую в будущем необходимо воспроизводить на страницы ресурса, а также здесь сохранены данные пользователей, вводимые ими при регистрации, которые понадобятся для их дальнейшей авторизации в системе и взаимодействия с ней.

Помимо двух основных частей также существует связующее программное обеспечение, называемое Middleware. Оно не относится к инструментам разработки, а функционирует как скрытый слой, который служит для соединения front-end и back-end.

В области программирования предоставлено огромное количество разнообразных фреймворков, каждый из которых предназначен преимущественно для определённых программных продуктов. Существуют фреймворки, которые специально предназначены для разработки front-end части. В их число входят React и Vue.js. Как правило, фронтенд-фреймворки предоставляют разработчикам набор инструментов и шаблонов, которые позволяют ускорить и упростить процесс создания интерфейсов пользовательского взаимодействия. Они также обеспечивают стандартизацию разработки, что упрощает совместную работу и переход между разработчиками.

Для того, чтобы произвести разработку клиентской части веб-сайта для продажи настольных игр «BGUniverse», будет выбран React, так как он специально предназначен для создания интерфейсов пользовательского взаимодействия, что позволит более качественно, быстро и легко осуществить написание программного кода, который даст возможность, в конечном итоге, предоставить пользователю удобный для него веб-сайт. React.js представляет собой библиотеку для языка программирования JavaScript с открытым исходным кодом для разработки пользовательских интерфейсов. Данный фреймворк был выбран для текущей разработки по следующим причинам:

- простота создания продуктов. С помощью компонентов проекта появляется возможность быстро и с лёгкостью создать полноценно функционирующий интерфейс веб-сайта или приложения любой сложности. Это сокращает количество ошибок и способствует более качественному и результативному подходу к разработке создаваемого программного обеспечения. Компоненты веб-сайта могут использоваться многократно в любой части кода, что ещё больше облегчает процесс разработки;

- реактивность. Это одно из ключевых преимуществ данного фреймворка. Библиотека React оперативно реагирует на изменение компонента и автоматически отражает его обновления в структуре документа.;

- эффективность. Благодаря виртуальному DOM (специальная древовидная структура, которая позволяет управлять HTML-разметкой из JavaScript-кода), библиотека экономит ресурсы. Не нужно полностью перезагружать всё DOM-дерево для изменения состояния элементов, что позволяет избежать излишней траты трафика пользователя и нагрузки на браузер. Вносятся изменения только в конкретные элементы через виртуальное дерево DOM, что делает сайты и приложения более «лёгкими» и удобными;

- высокая скорость. React делает веб-сайты и приложения сверхскоростными благодаря виртуальному DOM. Структура данных занимает меньше пространства, быстрее обновляется, что также ускоряет работу программистов. Компоненты создаются и используются повторно с лёгкостью, поскольку между ними уже существует встроенная система обмена информацией;

- удобная отладка. Вероятность ошибок в React значительно ниже, чем при использовании «чистых» языков без библиотек, так как многие процессы автоматизированы и минимизируют роль человеческого фактора. Если же разработчик допустил ошибку, то её идентификация и исправление просты

благодаря прозрачной логике потока данных, вспомогательным инструментам и понятному синтаксису.

API (Application Programming Interface, или программный интерфейс приложения) представляет собой совокупность способов, протоколов и инструментов, с помощью которых программы обмениваются своими возможностями, данными и выполняют различные функции. API бывают 4 видов, каждый из которых предназначен для определённых целей и имеет свои особенности. В их число входят следующие:

- SOAP API. Данный вид дословно обозначает «простой протокол доступа к объектам». Обмен информации между программой и сервером осуществляется на языке XML;
- RPC API. Этот API выполняет удалённые вызовы процедур. Клиент запрашивает определённое действие на сервере и получает ответ, который выполняет функцию веб-сайта;
- Websocket API. С использованием данного вида при отправке информации клиенту или серверу применяется текстовый формат JSON;
- REST API. При применении данного API программа отправляет нужные данные на сервер, который выполняет встроенные функции и отправляет итоговую информацию клиенту.

Для текущей разработки будет использоваться REST API, так как он является наиболее подходящим для осуществления тех задач, выполнение которых поставлено перед веб-сайтом. Данный вид API обладает следующими преимуществами, которые и поспособствовали его выбору для создания системы в рамках данной работы:

- отсутствие необходимости полагаться на структуру проекта. Разделение между клиентом и сервером позволяет разработчикам модифицировать каждый из них независимо друг от друга. С использованием гибких REST API, они могут настраивать определённые аспекты приложения без воздействия на остальные;
- портативность и адаптивность. Функционирование REST API зависит от того, насколько корректно были переданы запрашиваемые данные. Такие API позволяют легко переносить работу с одного сервера на другой и модифицировать базу данных по мере необходимости;
- масштабируемость. Благодаря независимой работе клиента и сервера, разработчики могут значительно ускорить процесс создания продукта.

Для того, чтобы осуществить разработку веб-сайта для продажи настольных игр «BGUniverse» также будет использован Node.js, представляющий собой приложение на C++, которое получает на входе код JavaScript и выполняет его. Для обеспечения взаимодействия с устройствами ввода и вывода на компьютере, платформа предлагает собственный интерфейс, написанный на языке C++. Это позволяет преобразовывать узкоспециализированный скриптовый язык JavaScript в полноценный, на котором можно создавать любые компьютерные приложения.

Для того, чтобы разработать серверную часть системы, понадобится использовать такой продукт, как PostgreSQL, включающий в себя программу для

управления базами данных PgMyAdmin. PostgreSQL представляет собой объектно-реляционную систему управления базами данных (СУБД), наиболее развитую из открытых в мире. Она имеет открытый исходный код и является альтернативой коммерческим базам данных. Данная СУБД была выбрана по ряду причин, к числу которых относятся следующие:

- объектно-реляционная модель. Наиболее распространённые системы управления базами данных - это реляционные системы, где информация представлена в виде записей со связями между ними. Другой широко используемой моделью является объектная. В ней данные представляются в виде объектов, их характеристик, методов и классов. PostgreSQL - это объектно-реляционная система управления базами данных, что означает поддержку как объектного, так и реляционного подходов;

- поддержка множества типов данных. Речь идёт не только о стандартных для баз данных типах, но и о специализированных форматах, таких как денежные значения, геометрические данные, перечисления, двоичные данные и других. Различные типы данных, поддерживаемые этой системой управления базами данных, позволяют разработчикам хранить и обрабатывать информацию о пользователях, товарах, их стоимости и других данных в структурированном и эффективном виде;

- работа с большими объёмами информации. Во многих системах управления базами данных, предназначенных для средних и малых проектов, существуют ограничения по размеру базы данных и количеству записей. В отличие от них, PostgreSQL не имеет таких ограничений;

- написание функций на нескольких языках. PostgreSQL позволяет создавать свои функции - уникальные блоки кода, реализующие определённые операции. Практически все СУБД имеют такую возможность, но PostgreSQL выделяется тем, что поддерживает больше языков программирования, чем его аналоги. Помимо стандартного SQL, PostgreSQL позволяет писать код на C, C++, Java, Python, PHP, Ruby и Lua. Он также поддерживает V8 - один из JavaScript-движков, поэтому и JS может быть использован вместе с PostgreSQL;

- одновременная модификация базы. СУБД использует клиент-серверную архитектуру, где данные хранятся на сервере и доступны для клиентских устройств. Это позволяет реализовать различные веб-сайты.

Таким образом, были рассмотрены ключевые понятия, связанные с технологиями, требуемыми для разработки различных программных продуктов, а также выбран тот стек технологий, который будет применён при разработке веб-сайта для продажи настольных игр «BGUniverse».

3.2 Описание алгоритма работы

После того, как были выдвинуты требования к системе, а также описаны и выбраны технологии, с применением которых будет происходить разработка веб-

					ОКЭИ 09.02.07. 4323. 19 ПЗ	Лист 14
Изм.	Лист	№ докум.	Подпись	Дата		

сайта, можно приступать к описанию алгоритма функционирования будущей автоматизированной системы.

Для того, чтобы наиболее понятно и наглядно представить то, каким образом будет осуществляться работа с создаваемым веб-сайтом, необходимо познакомиться с диаграммой деятельности, которую планируется применить для наглядной демонстрации функционирования системы.

Диаграмма деятельности представляет собой блок-схему, которая служит для визуализации того, каким образом поток управления переходит от одной деятельности к другой. Данный вид диаграмм отражает динамические аспекты поведения системы. Как и любая другая модель, диаграмма деятельности состоит из различных элементов, каждый из которых изображается соответствующе на схеме и отвечает за конкретные действия.

Первым рассматриваемым элементом диаграммы деятельности являются плавательные дорожки, изображаемые на схеме в виде параллельных линий. Они применяются для описания последовательности операций и отображения параллельных процессов. Дорожки помогают разработчикам и аналитикам визуализировать выполнение функций, определять ключевые точки принятия решений и учитывать возможные исключения. С их применением можно понять, как задача разделяется на подзадачи и как они связаны друг с другом.

Ещё одним из основных элементов диаграммы деятельности является начальный узел, который изображается на схеме в виде круга чёрного цвета. Данный объект служит для обозначения начального состояния процесса или потока управления в рамках системы. Начальный узел не имеет входящих связей и указывает на момент, с которого запускаются действия или операции, представленные в диаграмме. Он позволяет визуализировать отправную точку процесса и способствует более точному пониманию того, с чего начинается выполнение действий в системе.

Другим элементом диаграммы, который понадобится для построения данной модели для разрабатываемого веб-сайта, является ветвление, изображаемое на схеме с помощью узлов решения или шлюзов соответствующих исходящих стрелок, которые отображают различные траектории выполнения процесса в зависимости от определённого условия. Ветвление может быть простым при наличии только двух путей или сложным, когда имеется более двух исходящих стрелок. Данный элемент применяется для наглядного представления альтернативных вариантов выполнения действий.

Операция является ещё одним элементом диаграммы деятельности. Данный объект представляет собой действие или процесс, выполняемый системой или подсистемой. Это основная составляющая диаграммы деятельности, которая показывает один из шагов, необходимых для достижения конечного результата. На схеме операция изображается в виде прямоугольника со скруглёнными краями, внутри которого содержится название действия.

В целом, операции имеют важное значение в диаграмме деятельности, так как они устанавливают основные этапы достижения цели.

Следующим элементом рассматриваемой модели является поток, или ребро, изображаемый на схеме в виде стрелки, соединяющей между собой операции. Данный элемент представляет собой последовательность действий, выполняемых в процессе работы системы. Поток в диаграмме деятельности применяется для того, чтобы визуализировать и описать порядок функций, производимых веб-сайтом, а также, он помогает выявить возможные точки ветвления и альтернативные пути выполнения.

Решение в диаграмме деятельности изображается, как правило, в виде ромба, который обозначает точку принятия решения. Внутри этого символа указываются условия, при которых происходит ветвление потока управления. Решение необходимо для указания на принятие решений в процессе выполнения действий. У решения может быть несколько возможных вариантов выполнения, каждый из которых может привести к различным результатам или действиям. Также, его применяют для обозначения необходимости принятия решений на основании вводимых пользователем данных или внешних факторов.

Следующим элементом диаграммы деятельности, который будет использован при построении модели для разрабатываемой системы, является объединение. Данный элемент изображается на схеме в виде прямоугольника, имеющего несколько входящих и исходящих стрелок. Количество входящих стрелок соответствует количеству объединяемых элементов, а количество исходящих - количеству элементов, полученных в результате объединения. Объединение в диаграмме деятельности используется для группировки смежных действий или решений для упрощения визуализации и понимания потока управления в системе.

Последним рассматриваемым элементом диаграммы деятельности является конец деятельности, изображаемый на схеме в виде чёрного круга, окружённого границей на небольшом расстоянии. Данный объект служит для обозначения завершения процесса или этапа и перехода управления к следующему уровню или системе. Он указывает на то, что все относящиеся к нему действия окончены и ресурс готов к исполнению следующих задач. Конец деятельности также используют для отображения завершения работы системы или возврата к исходному состоянию.

После того, как была рассмотрена диаграмма деятельности, а также описаны её основные элементы, которые необходимо использовать при построении модели для текущей разработки, можно приступить к её созданию и описанию. Готовая диаграмма деятельности представлена в Приложении Б (Рисунок Б.1). Далее будет приведено описание функционирования веб-сайта для продажи настольных игр «BGUniverse» на основе построенной модели.

На построенной диаграмме расположено 3 плавательные дорожки, каждая из которых соответствует определённой части системы, выполняющей свои функции. Дорожка «Пользователь» отображает всех пользователей системы, которые осуществляют с ней взаимодействие. «Веб-сайт» представляет фронтенд-часть ресурса, которая представляет собой интерфейс, с помощью которого клиент работает с веб-сайтом. Клиентская часть в данной системе реализуется с

использованием React. «Сервер» отвечает за бэкенд-часть системы, которая содержит в себе все данные, необходимые для функционирования ресурса.

Неавторизованный пользователь начинает работу с веб-сайтом, посещая данный ресурс. Это действие отображено в модели посредством расположения на соответствующей дорожке начального узла, соединённого ребром с первой операцией. Далее, пользователь осуществляет вход в систему, вводя определённые данные, требуемые для его авторизации. В свою очередь, веб-сайт, на соответствующей странице которого вводится информация, при нажатии клиента на кнопку отправляет на сервер запрос, проверяющий наличие данных этого пользователя в базе. Затем, сервер осуществляет авторизацию пользователя и, в зависимости от его роли, даёт доступ к определенным страницам ресурса. Разделение ролей изображено на диаграмме в виде решения, которое имеет 2 условия, при которых происходит ветвление потока управления. В данном случае пользователь авторизуется либо как клиент, либо как администратор. Функция авторизации реализуется в системе с применением запросов к базе данных в PostgreSQL посредством REST API.

Если вход был осуществлён под аккаунтом клиента, веб-сайт предоставляет ему основные страницы системы, после чего пользователь может просмотреть каталог, в котором содержатся различные настольные игры, распределённые по жанрам. Клиент добавляет в корзину понравившиеся ему товары, а затем оформляет заказ, после чего завершает работу с веб-сайтом.

При авторизации пользователя в системе под ролью администратора, веб-сайт предоставляет ему соответствующую страницу, с помощью которой осуществляется просмотр данных пользователей. Также, администратор может редактировать информацию о товарах, такую как название, стоимость, изображение и описание. При необходимости ему также предоставлена возможность добавлять новые товары и удалять старые или вышедшие из продажи. После выполнения всех требуемых действий администратор завершает работу с веб-сайтом.

Таким образом, для веб-сайта для продажи настольных игр «BGUniverse» была создана диаграмма деятельности, описание которой помогло более наглядно представить процесс функционирования системы. Перед этим было проведено изучение необходимых составных элементов модели, каждый из которых был использован по назначению при построении диаграммы.

Сервер в данной системе выполняет функцию хранилища данных. Происходит это посредством использования системы управления базами данных PostgreSQL. В базе данных хранится несколько таблиц, каждая из которых содержит в себе соответствующую информацию о пользователях, их заказах, а также о товарах, предоставленных на веб-сайте.

Для того, чтобы наиболее понятно и наглядно представить информацию о том, что именно хранит в себе сервер, требуется построить диаграмму классов. Прежде, чем приступить к созданию данной модели для веб-сайта для продажи настольных игр «BGUniverse», необходимо рассмотреть основные элементы, из

которых она состоит, а также выяснить, для чего используется тот или иной объект.

Диаграмма классов - это модель, которая описывает систему путём визуализации различных типов объектов внутри системы и видов статических связей, существующих между ними. Она помогает облегчить процесс создания базы данных в будущем, так как наглядно демонстрирует, какие именно сущности необходимо будет добавить, какими атрибутами будут обладать классы, а также, какие связи нужно установить между ними. Диаграмма классов включает в себя следующие основные элементы:

- класс (Class). Представляет собой логический блок, который содержит множество объектов, которые имеют общие атрибуты, операции и семантику. На модели изображается в виде прямоугольника с именем класса внутри;
- атрибут (Attribute). Отражает свойство класса, описывающее некоторую характеристику объектов этого класса. На диаграмме изображается в виде поля внутри класса с именем атрибута;
- операция (Operation). Это метод или функция, определяющая действие, которое выполняется над объектами класса. На модели изображается как метод внутри класса с именем операции;
- отношение (Relation). Представляет собой связь между классами, показывающую взаимодействие объектов одного класса с объектами другого. На диаграмме изображается в виде линии, ограниченной с двух сторон определёнными символами в зависимости от вида связи.

Для того, чтобы определить, какие связи устанавливаются между объектами в базе данных, необходимо рассмотреть виды этих связей. Всего выделяют 3 вида, к которым относятся:

- один-к-одному (1:1). В таком типе связи каждому объекту одного класса соответствует ровно один объект другого класса, и наоборот;
- один-ко-многим (1:N). При данной связи каждому объекту одного класса может соответствовать много объектов другого класса, но каждому объекту другого класса соответствует только один объект первого класса;
- многие-ко-многим (N:N). В данном виде связи каждому объекту одного класса может соответствовать несколько объектов другого класса и наоборот.

Готовая диаграмма классов для веб-сайта для продажи настольных игр «BGUniverse» представлена в Приложении В (Рисунок В.1). Далее, необходимо описать данную модель для того, чтобы понять, каким образом будет организована работа с серверной частью ресурса.

На диаграмме классов для разрабатываемой системы представлено 5 классов, каждый из которых имеет свои атрибуты и отношения с другими классами. Первым из них является «user», который представляет собой таблицу в базе данных, которая содержит в себе информацию о пользователях, которые зарегистрированы на веб-сайте. Здесь находится несколько атрибутов, отвечающих за то, какие именно данные представлены: «id» (уникальный идентификатор), «email» (адрес электронной почты), «password» (пароль,

хранящийся в закодированном виде), «role» (роль пользователя, позволяющая в дальнейшем осуществить разграничение доступа), «fio» (фамилия, имя и отчество) и «phone» (номер телефона). Класс «user» связан со следующим классом «cart» видом «один-к-одному». «Cart» представляет собой связующее звено между пользователем и его корзиной, хранящей в себе добавленные туда товары. Данный класс состоит из двух атрибутов: «id» и «user_id» (внешний ключ, связывающий его с классом «user»). Далее, класс корзины «cart_game», состоящий из атрибутов «id», «game_id» (внешний ключ, связывающий его с классом товаров видом «один-ко-многим») и «cart_id» (внешний ключ, связывающий его с классом «cart»). Данная сущность связана с таблицей товаров «game» видом «один-к-одному». Класс «game» содержит в себе все данные о товарах: «id», «title» (название), «price» (стоимость), «description» (описание), «img» (изображение) и «genreId» (внешний ключ, связывающий его с классом жанров видом «один-ко-многим»). Последний класс «genre» состоит из жанров, по которым разделяют все товары, и содержит в себе: «id» и «title».

Таким образом, были рассмотрены и созданы диаграмма деятельности и классов для веб-сайта для продажи настольных игр «BGUniverse», а также описано функционирование системы на их основе.

3.3 Описание интерфейса пользователя

При разработке веб-сайта необходимо продумать и создать элементы, с помощью которых будет происходить работа с системой. Для реализации этого процесса применяется пользовательский интерфейс, также называемый UI (User Interface), представляющий собой среду, позволяющую взаимодействовать с программой или сайтом.

UI, как правило, обеспечивает работу клиента с системой за счёт применения таких элементов, как клавиатура, компьютерная мышь, джойстик, дисплей и стилус. Пользовательский интерфейс может выглядеть различным образом, в зависимости от операционной системы. Графический интерфейс является наиболее распространённым среди всех типов UI, к числу которых относятся:

- интерфейс командной строки и текстовый интерфейс (Command Line Interface или CLI), где пользователь, используя клавиатуру, вписывает какие-либо команды в терминале, после нажимая на Enter;
- графический пользовательский интерфейс (Graphical User Interface GUI). Данный вид позволяет взаимодействовать с веб-сайтом посредством различных контроллеров, таких как компьютерная мышь, джойстик и другие;
- жестовый, голосовой, тактильный, нейронный. В качестве примера можно привести голосовые помощники.

Для того, чтобы реализовать взаимодействие пользователя с веб-сайтом для продажи настольных игр «BGUniverse», используется GUI, как и для множества

других подобных ресурсов. При создании интерфейса применяют принципы атомарного дизайна, представляющего собой подход к разделению системы любой сложности на маленькие части - атомы, которые можно использовать несколько раз и комбинировать друг с другом. Составными частями данного подхода являются следующие элементы интерфейса:

- атомы. Являются мельчайшими частями интерфейса, к которым относят различные кнопки, поля ввода, чекбоксы, радиокнопки и стили для типографики;
- молекулы (группы атомов). Представляют собой объединение атомов, при котором сочетается несколько мельчайших элементов, например, поле ввода и кнопка;
- организмы. Состоят из объединения молекул, которое образует крупную часть интерфейса, например, навигационную панель;
- шаблоны. Представляют собой полноценный интерфейс для решения типовых задач. Состоят из организмов, объединённых друг с другом.

На разрабатываемом веб-сайте пользователю будет доступно несколько страниц, каждая из которых содержит в себе соответствующую информацию и создаётся с добавлением различных элементов графического пользовательского интерфейса, с помощью которых реализуется взаимодействие клиента или администратора с ресурсом.

При начале работы с системой пользователю предоставляется главная страница, которая содержит в себе основную информацию о компании, её преимущества, а также некоторое количество популярных товаров, которые чаще всего заказываются клиентами. В самом верху располагается панель навигации, которая содержит в себе логотип, позволяющий попасть на главную страницу из любого раздела, а также ссылки на все страницы веб-сайта, доступные пользователю, и кнопка, направляющая в раздел входа в систему. Данная кнопка после авторизации клиента изменяется и отвечает за выход из аккаунта. С помощью неё происходит перемещение по различным блокам ресурса, в зависимости от роли пользователя. Страницы профиля и корзины будут доступны только после авторизации, при нажатии на них неавторизованным клиентом он будет перенаправлен на вход в систему. Данный элемент доступен также на страницах каталога, профиля и корзины. Ниже располагается блок, содержащий в себе краткую информацию о назначении ресурса. Здесь также есть кнопка, при нажатии на которую пользователь сможет быстро переместиться на страницу каталога и просмотреть товары. Далее, клиент пролистывает главную страницу вниз и просматривает информацию. В блоке с популярными товарами также расположена кнопка, которая перенесёт пользователя в раздел каталога, содержащий в себе большее количество данной категории товаров. В самом низу находится подвал, который также выполняет функцию навигации по веб-сайту. Данный объект также размещён на страницах каталога, профиля и корзины.

На странице каталога, помимо верхней навигационной панели, также располагается боковое меню, которое позволяет пользователю перемещаться между категориями настольных игр. При нажатии на любой из пунктов этого

элемента клиенту будут предоставлены товары, которые соответствуют выбранному жанру. При наведении на объект с конкретной настольной игрой появляется элемент, который содержит в себе краткое описание позиции. Также, на каждом товаре есть кнопка, при нажатии на которую неавторизованный пользователь будет направлен на страницу входа, а авторизованный сможет добавить понравившуюся позицию в корзину. В самом низу страницы расположен подвал.

Страница входа содержит в себе логотип, расположенный на самом верху и позволяющий вернуться на главную. Ниже находится форма, в которой есть поля ввода, которые пользователю необходимо заполнить корректными данными для авторизации. Здесь также есть ссылка, которая направляет на страницу регистрации, если у клиента ещё нет аккаунта. Последним элементом формы является кнопка, отправляющая введённые параметры на сервер для входа в систему.

Страница регистрации выглядит и функционирует по тому же принципу, что и страница входа. Отличие состоит в количестве полей ввода в форме - здесь их больше. Также, ссылка в форме направляет пользователя на страницу входа, а при нажатии на кнопку происходит внесение данных в базу для осуществления регистрации.

После авторизации клиенту доступны страница профиля, на которой он может просмотреть свои данные, и страница корзины. В разделе с корзиной пользователь может увидеть все товары, которые он сохранил, и оформить их заказ с помощью соответствующей кнопки. На обеих страницах имеются навигационная панель, расположенная в самом верху, подвал, находящийся внизу.

При входе в систему администратора, ему открывается страница, на которой он может просмотреть данные о всех пользователях, зарегистрированных на веб-сайте, а также об их заказах. В верхней части размещён логотип и кнопка, при нажатии на которую можно выйти из системы. Здесь также расположена форма, с помощью которой он может добавить новый товар. Она содержит в себе поля ввода, каждое из которых соответствует характеристике, которую необходимо внести в базу данных. В нижней части формы находится кнопка, отправляющая параметры на сервер. Также, администратор видит общую таблицу всех товаров, где он может редактировать данные и сохранять изменения, а также, при необходимости удалять товары. Ему также доступны главная страница и страница каталога.

Таким образом, был рассмотрен пользовательский интерфейс и его виды, что было необходимым для дальнейшего его внедрения в текущую разработку. Также, было проведено ознакомление с одним из принципов его создания, включающего различные элементы, которые использовались при реализации графического пользовательского интерфейса для веб-сайта для продажи настольных игр «BGUniverse».

4 Тестирование приложения

4.1 План тестирования

Заключительным этапом работы над созданием веб-сайта для продажи настольных игр «BGUniverse» является проведение тестирования, которое необходимо для того, чтобы оценить полноту и корректность функционирования готового продукта. Прежде чем приступить непосредственно к его осуществлению, требуется рассмотреть основные его понятия и виды, чтобы качественно и правильно проверить систему на работоспособность, а также, необходимо составить план проведения.

Тестирование представляет собой способ проверки того, выполняет ли программа или её отдельные компоненты предназначение, для которого они разрабатывались. Основной целью его проведения является обнаружение любых проблем или ошибок и определение того, что программное обеспечение высокого качества. Существуют различные виды тестирования, каждый из которых предназначен для того, чтобы осуществить определённый тип проверки веб-сайта. Далее будут описаны те виды, которые требуется применить по отношению к данной системе.

Первым типом проверки, который будет применён к разработанному ресурсу, является unit-тестирование, или модульное тестирование, который представляет собой проверку отдельных элементов процессов, кода и модулей. Низкоуровневое тестирование играет важную роль в предотвращении накопления ошибок и, следовательно, снижает риск серьёзных сбоев в программном обеспечении. Unit-тестирование позволяет проверять отдельные элементы, помогая избежать ситуации, когда после объединения всех компонентов приложения оно начинает работать нестабильно. Основными задачами являются:

- обнаружение и устранение ошибок на ранних стадиях разработки, что позволяет сократить затраты в будущем;
- гарантия работоспособности кода: успешное прохождение тестов подтверждает качество разработанного продукта;
- простота и скорость внесения корректировок в систему благодаря лучшему пониманию разработчиками структуры кода;
- повторное использование кода: unit-тесты используются при переносе частей кода в другие приложения;
- применение в проектной документации: это облегчает понимание проекта сторонними специалистами.

В целом, все эти задачи направлены на снижение затрат на разработку и уменьшение риска совершения серьёзных ошибок в процессе написания программного кода.

Unit-тестирование веб-сайта для продажи настольных игр «BGUniverse» будет состоять из следующих этапов:

					ОКЭИ 09.02.07. 4.323. 19 ПЗ	Лист
						22
Изм.	Лист	№ докум.	Подпись	Дата		

- выбор кода для тестирования. В данном случае тесту подвергается код, который отвечает за регистрацию пользователя в системе;
- определение способа проведения теста. Тестирование выбранного фрагмента программы будет происходить внутри системы, так как необходимо проверить также взаимодействие с другими элементами кода;
- написание теста. Прописывается условие, которое будет при успешном выполнении выводить результат проведенных действий, а при не успешном - сообщение об ошибке и её причине;
- запуск теста;
- интерпретация результатов. Если в тесте использовались неправильные данные, а результат положительный, то это будет ошибкой.

Для того, чтобы провести тестирование, необходимо также применить тест-кейс, представляющий собой форму записи проверки, которую проводит тестировщик. Другими словами, это алгоритм действий, по которому предполагается тестировать уже написанную программу. В нём подробно прописаны шаги, которые необходимо сделать для подготовки к тесту, сама проверка и ожидаемый результат. Как правило, один тест-кейс описывает небольшую последовательность действий с одним конкретным результатом.

Существует три вида тест-кейсов:

- позитивные, или положительные. Они проверяют, как система реагирует на правильные данные;
- негативные, или отрицательные. Показывают, что система умеет работать с некорректными данными;
- деструктивные. Служат для проверки прочности системы. Например, позволяют убедиться, что в поле для email нельзя ввести команду, которая удалит базу данных зарегистрированных пользователей.

В данном случае будут использованы позитивные тест-кейсы. Для того, чтобы протестировать разработанную систему с их помощью потребуется создать тест-кейс из следующих элементов:

- заголовок. В данном случае это будет «Авторизация пользователя в системе»;
- предусловия. Для того, чтобы провести данный тест пользователю необходимо быть зарегистрированным в системе и находиться на странице входа;
- окружение. Тестирование планируется провести на персональном компьютере с ОС Windows 10, в браузере Yandex;
- шаги. Для авторизации необходимо зайти на страницу входа, заполнить все поля ввода в форме и нажать на кнопку;
- ожидаемый результат. При вводе корректных данных пользователь направляется на главную страницу, после чего может переходить по страницам, которые были недоступны до авторизации.

В последнюю очередь, необходимо осуществить интеграционное тестирование продукта, чтобы окончательно убедиться в его работоспособности. Интеграционное тестирование представляет собой тип проверки, при котором программные модули объединяются логически и тестируются как группа. Целью

проведения является выявление багов при взаимодействии между программными модулями системы и проверка обмена данными между этими самими модулями.

Существуют следующие подходы к интеграционному тестированию:

- подход «Большого взрыва». Все составляющие или модули собираются вместе и затем проверяются. В этом случае объединённый комплект компонентов рассматривается как единое целое. В случае, если не все компоненты в комплекте завершены, их объединение может оказаться невозможным;

- инкрементальный подход. Инкрементное тестирование заключается в объединении двух или более модулей, которые имеют логическую связь друг с другом. После объединения проверяется корректность работы приложения. Затем продолжают добавлять другие модули, связанные с уже объединёнными, и продолжают этот процесс до тех пор, пока не объединят и не протестируют все логически связанные модули.

- подход сэндвич-тестирования предполагает совместную проверку модулей верхнего уровня с модулями нижнего, а также их интеграцию с последующей проверкой как единой системы.

Для того, чтобы провести интеграционное тестирование необходимо определить этапы, которые будут выполнены для этого. Порядок данного вида тестов для веб-сайта для продажи настольных игр «BGUniverse» следующий:

- определение цели теста. В данном случае целью является проверка интерфейсной связи между модулем авторизации и корзиной;

- определение методов. Будет использован подход большого взрыва, при котором все компоненты собираются вместе, а затем тестируются;

- написание условия для тестирования. Потребуется персональный компьютер, с которого нужно войти на сайт, авторизоваться в системе и попробовать добавить товары в корзину;

- описание ожидаемого результата. После авторизации пользователь должен успешно добавлять товары в корзину и иметь возможность просматривать их;

- проведение тестирования;

- оценка результатов.

Подводя итоги, был составлен план тестирования с использованием различных его видов, что поможет проверить готовый продукт на работоспособность и исправить ошибки в функционировании до внедрения продукта в использование.

4.2 Оценка результатов проведения тестирования

После того, как был создан план тестирования, по которому проводилась проверка работы определённых модулей системы, необходимо проанализировать результаты каждого теста и сделать выводы на их основе.

					ОКЭИ 09.02.07. 4.323. 19 ПЗ	Лист
						24
Изм.	Лист	№ докум.	Подпись	Дата		

Первым видом выступает unit-тестирование, целью которого была проверка функционирования системы регистрации на сайте. Тест проводился внутри системы путём написания кода с условием, который содержал в себе сам модуль, который подвергается оценке, а также исходы поведения системы при корректной работе и некорректной.

Планировалось, что при вводе пользователем некорректных данных или наличии в базе данных клиента с таким же email будет выводиться ошибка, следовательно, регистрации не произойдёт. При вводе правильных данных в консоль должен был возвращаться токен пользователя, что служит подтверждением успешной регистрации.

При проведении теста было проверено оба исхода. При неверных или дублирующихся данных выводилась ошибка, что является одним из доказательств того, что модуль функционирует так, как нужно, не позволяя создавать пользователей с одинаковым адресом электронной почты. При вводе правильной и уникальной информации возвращался токен, что является подтверждением осуществления регистрации. Это является вторым доказательством корректности работы тестируемого модуля.

По результатам unit-тестирования можно сделать вывод о том, что модуль регистрации пользователя в системе был разработан правильно и допускается к внедрению его в готовую систему.

Следующим подходом к проверке системы являются тест-кейсы. В данном случае требовалось протестировать авторизацию пользователя в системе. При первом посещении сайта была осуществлена регистрация путём ввода соответствующих данных на странице, а затем осуществлён переход ко входу в систему. Требовалось ввести в форму те же данные, что были использованы при создании аккаунта. Данные были верными, поэтому произошла авторизация, после чего недоступные обычному пользователю страницы стали доступными.

После проведения ранее описанного тест-кейса можно сделать вывод о том, что система авторизации на веб-сайте работает так, как нужно, и позволяет осуществить разделение ролей пользователей.

Последним проводимым тестированием является интеграционное. Оно объединяет в себе ранее проведённые тесты и дополняет их проверкой функции добавления пользователем товаров в корзину. Для начала, была осуществлена авторизация в системе, после чего произошёл переход на страницу каталога, содержащую товары, которые необходимо было попытаться добавить в корзину. Действие планировалось проводить путём нажатия на соответствующую кнопку, расположенную на каждой позиции каталога. Было осуществлено добавление трёх товаров, после чего, при переходе в раздел корзины, они отобразились на её странице.

По итогам интеграционного тестирования можно сделать вывод, что вся клиентская часть веб-сайта функционирует так, как и было предусмотрено, что свидетельствует о качестве разработки продукта и возможности его эксплуатации для пользователей.

Заключение

В ходе выполнения данной курсовой работы производилась разработка автоматизированной системы, которая происходила на базе Государственного автономного образовательного учреждения «Оренбургский колледж экономики и информатики».

Целью курсового проекта была разработка веб-сайта для автоматизации процесса продажи настольных игр «BGUniverse». Требовалось создать продукт, который позволит облегчить работу людей, которые занимаются предоставлением людям товаров, необходимых для удовлетворения их потребностей в области проведения своего досуга. Создание программного продукта предполагало решение проблем, связанных с отсутствием удобного ресурса с функцией оформления заказа в режиме онлайн.

Для того, чтобы успешно и правильно достигнуть поставленной цели, были выдвинуты определённые задачи, которыми являлись:

- создание прототипа и макета дизайна веб-сайта;
- создание сервера для хранения информации;
- адаптивная разработка страниц веб-сайта;
- создание системы авторизации;
- обеспечение возможности оформления заказа в режиме онлайн;
- осуществление тестирования готового веб-сайта.

В ходе работы был создан прототип дизайна системы, который послужил основой для полноценного макета того, как должна выглядеть готовая система для пользователя. При разработке прототипа был подобран способ расположения контента и основных элементов интерфейса на странице, а также их размер. После этого создавался сам дизайн, выполненный в подобранной согласно тематике веб-сайта цветовой гамме. Также, был выбран шрифт, который подходит по теме и хорошо выглядит в совокупности с остальными элементами. Это являлось первой задачей, которая была успешно и полноценно достигнута.

Затем, необходимо было произвести разработку серверной части системы, которая должна выполнять роль хранилища данных, а также реализовывать в дальнейшем все требуемые функции, связанные с ней. В программе, используемой для написания кода, был создан проект, в котором и производилось создание сервера. Были прописаны все требуемые модели для базы данных и написан код, который отвечает за реализацию всех функций, которые в будущем должны быть доступны пользователю. Вторая задача была также успешно выполнена.

Следующей задачей являлась адаптивная разработка страниц веб-сайта, которую требовалось произвести на основе созданного ранее макета дизайна. Клиентская часть была объединена в один проект с серверной, после чего поэтапно происходило написание кода, который позволял создать для страниц элементы, а затем задать им стилизацию, с помощью чего и обеспечивается адаптивный дизайн системы. Задача была полноценно достигнута.

После этого требовалось внедрить первую функцию, которая должна отвечать за авторизацию пользователя в системе. Серверная и клиентская часть были дополнены соответствующими фрагментами кода, что позволило реализовать возможность клиента регистрироваться на веб-сайте, а также, используя те же данные, входить в свой аккаунт. Система авторизации пользователя была успешно внедрена в продукт.

Для обеспечения возможности оформления заказа онлайн, что являлось одной из самых важных задач, также были внесены дополнительные фрагменты в код, написанный при разработке веб-сайта. Была реализована функция добавления авторизованным пользователем товаров в корзину, с помощью которой и происходит заказ необходимых позиций.

После создания дизайна системы, разработки страниц веб-сайта и внедрения различных функций требовалось провести тестирование системы, чтобы иметь представление о том, нет ли ошибок в работе продукта и готов ли он к использованию. Все предусмотренные тесты были проведены, а их результаты проанализированы.

Все поставленные задачи были выполнены, что позволило полноценно и корректно достигнуть поставленной цели. В ходе работы был создан веб-сайт «BGUniverse», который позволит автоматизировать процесс продажи настольных игр.

Функционал приложения реализован в полном объёме и предоставляет пользователям системы все возможности, которые им требуются для просмотра настольных игр и их заказа в режиме онлайн.

В ходе реализации системы добавления товаров в корзину возникли некоторые сложности, связанные с созданием уникальных сущностей в базе данных на сервере, требуемых для хранения в них информации о корзине каждого конкретного пользователя, который был зарегистрирован на веб-сайте. Эти сложности были решены, что привело к созданию полноценной системы, которая качественно и правильно выполняет свою работу.

В дальнейшем имеются перспективы развития разработанной системы, которые позволят сделать её ещё более удобной и привлекательной для пользователя. Одной из них является добавление поиска по ресурсу, который поможет сделать взаимодействие с веб-сайтом быстрее и проще. Также, имеется возможность добавить более удобную фильтрацию товаров не только по жанрам, но и по стоимости. Не менее хорошей перспективой для развития является внедрение системы отзывов и оценок, что позволит облегчить процесс выбора той или иной игры на основе мнения других клиентов, а также вызовет у них больше доверия к компании.

Подводя итоги, все задачи и главная цель разработки были выполнены, после чего проводилось тестирование, по результатам которого стало понятно, что система пригодна к её введению в эксплуатацию. Также, в дальнейшем имеется множество возможностей для расширения системы, что позволит сделать её ещё лучше и эффективнее.

Список использованных источников

1 Веб-сайт в интернете: что это такое, зачем нужен, как работает, как его создать. - [Электронный ресурс]. - Режим доступа: <https://semantica.in/blog/что-такое-veb-sajt.html> (дата обращения: 1.12.2023)

2 Веб сайт это? Что такое веб-сайт и для чего он нужен? - [Электронный ресурс]. - Режим доступа: <https://pumpkinstudio.ru/blog/development/veb-sajt/> (дата обращения: 1.12.2023)

3 Диаграмма вариантов использования. Актуальное руководство. - [Электронный ресурс]. - Режим доступа: https://itonboard.ru/analysis/629-diagramma_variantov_ispolzovanija_use_case_diagram/ (дата обращения: 4.12.2023)

4 Диаграмма деятельности UML — Планёрка — креативность для бизнеса. - [Электронный ресурс]. - Режим доступа: <https://planerka.info/item/diagrammy-deyatelnosti-uml/> (дата обращения: 9.12.2023)

5 Тестирование сайта: что это, способы тестирования веб-сайтов. - [Электронный ресурс]. - Режим доступа: <https://optimalgroup.ru/blog/testirovanie-sajta/> (дата обращения: 24.12.2023)

6 Функциональные и нефункциональные требования - решения Visure. - [Электронный ресурс]. - Режим доступа: <https://visuresolutions.com/ru/руководство-по-отслеживаемости-управления-требованиями/функциональные-и-нефункциональные-требования/> (дата обращения: 4.12.2023)

7 Что такое пользовательский интерфейс: ценность, виды UI и элементы. - [Электронный ресурс]. - Режим доступа: <https://ux-journal.ru/что-такое-polzovatelskij-interfejs.html> (дата обращения: 12.12.2023)

8 Что такое фреймворк | Медиа Нетологии. - [Электронный ресурс]. - Режим доступа: <https://netology.ru/blog/06-2023-framework> (дата обращения: 7.12.2023)

9 Node.JS — что это такое и для чего нужно. - [Электронный ресурс]. - Режим доступа: <https://blog.skillfactory.ru/glossary/node-js/> (дата обращения: 7.12.2023)

10 PostgreSQL: что это за СУБД, основы и преимущества. - [Электронный ресурс]. - Режим доступа: <https://blog.skillfactory.ru/glossary/postgresql/> (дата обращения: 7.12.2023)

Приложение А (обязательное)

Диаграмма прецедентов

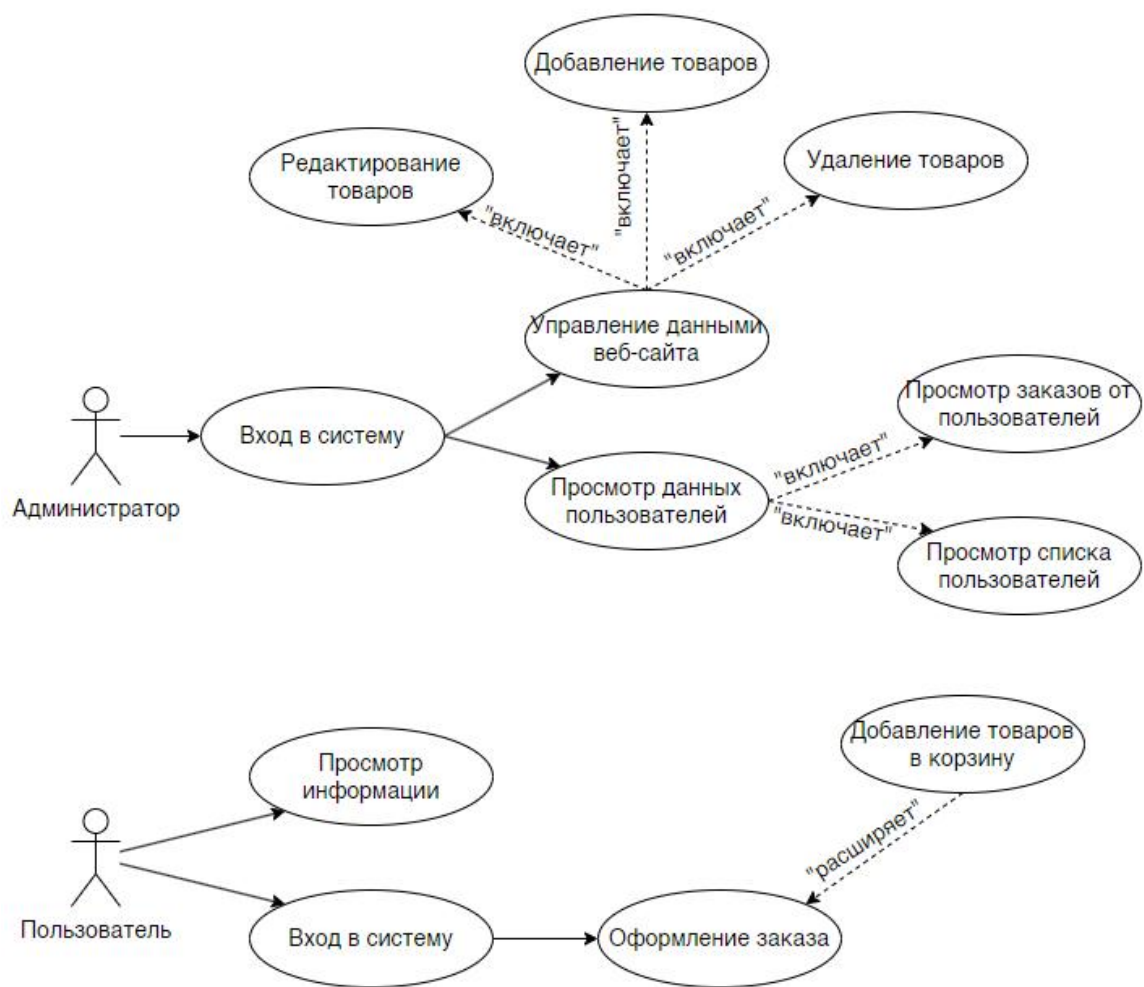


Рисунок А.1 - Диаграмма прецедентов

Приложение Б (обязательное)

Диаграмма деятельности

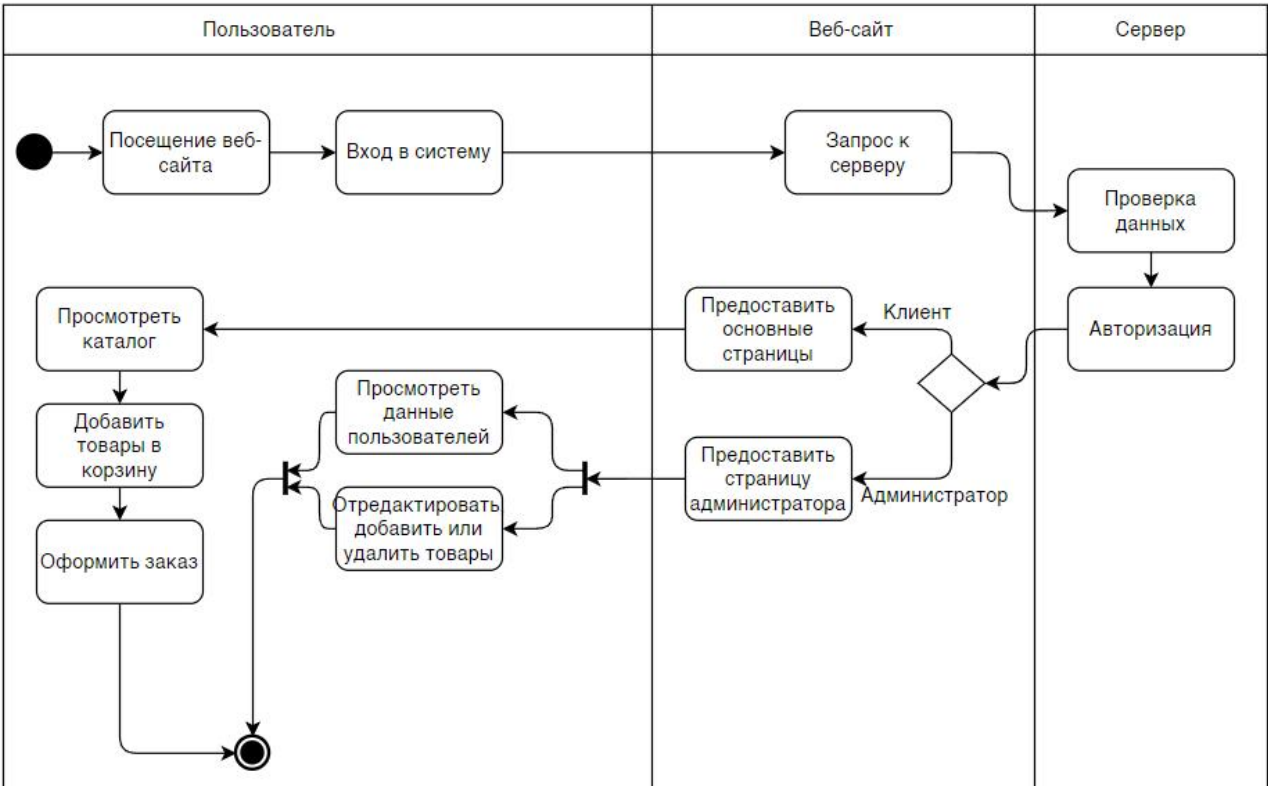


Рисунок Б.1 - Диаграмма деятельности

Приложение В
(обязательное)

Диаграмма классов

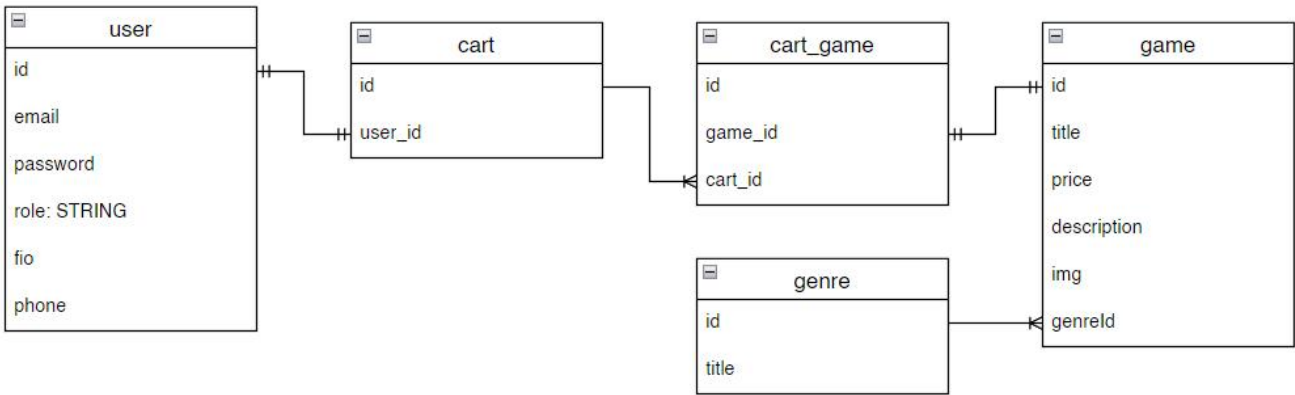


Рисунок В.1 - Диаграмма классов