**Санкт-Петербургское государственное бюджетное профессиональное образовательное учреждение**

**«Академия управления городской средой, градостроительства и печати»**

Допустить к защите:

Зам. директора по УМР

\_\_\_\_\_\_\_\_\_\_\_\_ Фомичева О.В.

«\_\_\_» \_\_\_\_\_\_\_\_\_ 2024 г.

**ДИПЛОМНЫЙ ПРОЕКТ**

Тема: Разработка веб-приложения для типографии

Студент группы 11ИСП-211 Герасимов В.А

№ группы подпись И.О. Фамилия

Специальность 09.02.07 «Информационные системы и программирование»

код наименование

Руководитель

подпись И.О. Фамилия

Санкт-Петербург

2024 г.

**СОДЕРЖАНИЕ**

[СОДЕРЖАНИЕ 1](#_Toc9021)

[ВВЕДЕНИЕ 2](#_Toc242)

[1 АНАЛИЗ ПРЕДМЕТНОЙОБЛАСТИ 4](#_Toc2310)

[1.1 Идентификация предметной области 4](#_Toc1062)

[1.2 Выбор методологии и технологии концептуального моделирования 6](#_Toc13278)

[1.3 Функциональное моделирование 14](#_Toc28636)

[1.4 Моделирование элементов структуры веб-приложения с использованием UML 18](#_Toc3874)

[2 ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ 21](#_Toc22913)

[2.1 Анализ требований и разработка спецификаций 21](#_Toc23553)

[2.2 Проектирование и разработка дизайна интерфейса 27](#_Toc24856)

[2.3 Разработка кода веб-приложения и программирование 28](#_Toc22461)

[2.4 Тестирование и отладка 36](#_Toc7934)

[2.5 Разработка руководства пользователя 40](#_Toc20697)

[ЗАКЛЮЧЕНИЕ 43](#_Toc18654)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 44](#_Toc2752)

**ВВЕДЕНИЕ**

Наше время называют «информационным веком». Это название возникло потому, что самым важным, ценным и необходимым ресурсом является информация. Обладание информацией и умение своевременно, быстро и четко донести необходимую информацию до клиентов – ключ к успеху в реалиях современного бизнеса. В настоящий момент именно Интернет, способен оперативно и массово передавать текст, звук, изображения и даже видео-ролики, т.е. практически любую информацию. C точки зрения пользователя, Интернет – это огромный информационный ресурс, в котором можно найти все, что угодно: от прогноза погоды до личных предпочтений голливудских кинозвезд. Аудитория сети Интернет растет с каждым годом, Интернет – самое быстро развивающееся средство передачи информации за всю историю человечества. Пользователи сети интернет являются целью особого интереса для рекламодателей, потому что факт остается фактом – среди пользователей сети куча потенциальных клиентов.

Сфера услуг являются одной из самых перспективных сфер экономики и экономической деятельности. Она охватывает широкий спектр экономической деятельности и по своему функциональному назначению не является единым комплексом. В процессе коммерциализации российской экономики и при Российских реалиях: большие расстояния, острая нехватка времени; особую значимость приобретает такая важная форма реализации товаров и услуг, как торговля и реклама через сеть Интернет. Существует огромное количество способов коммерческого подхода к сети интернет. В сети можно рекламировать услуги, продавать товары либо представлять потребителю и то и другое. В современном бизнесе многое зависит от самопрезентации компании, ее позиционирования на рынке оказываемых услуг и способности искать новых клиентов и рынки сбыта. Одним из инструментов, как имиджевых, так и маркетинговых, является наличие своего сайта и приложения в сети Интернет. С точки зрения бизнеса Интернет – это современная рекламная площадка, позволяющая обеспечить приток клиентов. Web-приложение превращает компанию в современный бизнес. Web-приложение компании является важнейшим источником информации для потенциальных клиентов и людей, чье мнение является общественно значимым.

Актуальность темы дипломного проекта обусловлена востребованностью качественного web-приложения для возможности оформления заказа в типографии.

Объектом разработки является web-приложение типографии.

Предметом разработки является разработка web-приложения типографии.

Цель настоящей работы: разработка web-приложения типографии, предназначенного для ознакомления потенциальных клиентов с компанией и спектром предоставляемых услуг, возможности воспользоваться услугой по печати в типографии, а также упорядочивания и ускорения деятельности сотрудников типографии.

Результатом работы будет являться web-приложение типографии.

Дипломный проект состоит из 2-х глав. В первой проводится анализ предметной области, во второй проектирование и разработка веб-приложения.

**1 АНАЛИЗ ПРЕДМЕТНОЙОБЛАСТИ**

**1.1 Идентификация предметной области**

В качестве предметной области выступает типография Санкт-Петербургского государственного бюджетного профессионального образовательного учреждения «Академия управления городской средой, градостроительства и печати». Типография оказывает полиграфические услуги, печать. Главная задача - это качественное и оперативное обслуживание клиентов, так же отдел осуществляет выполнение внутренних заказов, переходящих из других подразделений Академии.

Выполнение услуг осуществляется сотрудниками типографии. В задачи сотрудников входит качественное выполнение заказа и правильный расчет стоимости выполненных услуг.

Основные бизнес-процессы компании - закупки, складирование запасов, продажи, взаиморасчеты с поставщиками и клиентами.

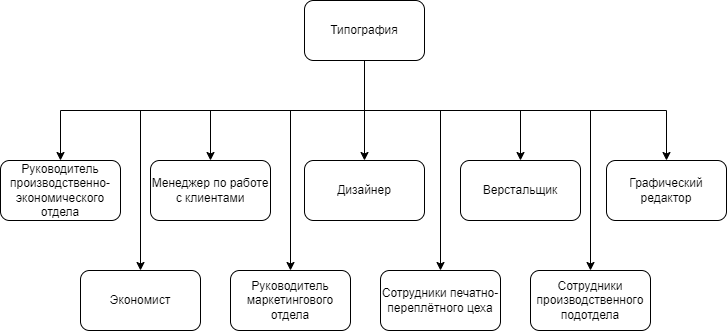
В типографии также присутствует руководитель, который назначает ответственного за принятие заказа и его выполнение, так же руководитель выдаёт необходимую продукцию по служебной записке, если услуга выполняется для нужд предприятия. Обязанностями сотрудников отдела являются:

* сдача статистики по каждому оператору отдела;
* учет рабочего времени персонала;
* обеспечение отдела расходными материалами;
* контроль за своевременным обслуживанием оргтехники и оборудования согласно нормам и правилам эксплуатации;
* сбор информации о постоянных клиентах и внесение их в клиентскую базу;
* интервьюирование клиентов об новых направлениях деятельности копи-центра для удовлетворения спроса;
* контроль внешнего вида сотрудников;
* обеспечение быстрого и качественного обслуживания клиентов;
* прием заказов для изготовления сувенирной продукции для организации выездной торговли;
* контроль оплаты клиентами услуг копи-центра.

Типография оказывает следующий спектр услуг:

* ксерокопирование цветное и черно-белое;
* сканирование;
* набор и распечатка текстов, чертежей, схем, бланков, листовок, европлакатов;
* цветная широкоформатная распечатка фотографий, афиш, плакатов;
* брошюровка документов, рефератов, дипломных проектов и т.д.;

Структура типографии представлена на рисунке 1.

Рисунок 1 Структура типографии

**1.2 Выбор методологии и технологии концептуального моделирования**

Проектирование системы, особенно на этапе концептуальной модели должно быть привязано к процессу: технологическому, или бизнес-процессу. Возрастание числа сущностей и связей в информационных системах (ИС) приводит к необходимости постоянного пересмотра методов и средств проектирования ИС, основанных на CASE-технологиях.

Современные методы проектирования ИС довольно разнообразны, и среди разработчиков существует определённое непонимание целесообразности и границ использования каждого из методов, поэтому появилась проблема выбора и обоснованного использования того или иного средства. В частности, речь идёт об областях применения структурного (функционального) и объектно-ориентированного моделирования.

Методология SADT (Structured Analisys and Design Technique - технология структурного анализа и проектирования) разработана Дугласом Т. Россом в 1969-1973 годах. Технология изначально создавалась для проектирования систем более общего назначения по сравнению с другими структурными методами, выросшими из проектирования программного обеспечения. SADT - одна из самых известных и широко используемых методик проектирования. Новое название методики, принятое в качестве стандарта - IDEF0 (Icam DEFinition) - часть программы ICAM (Integrated Computer-Aided Manufacturing - интегрированная компьютеризация производства), проводимой по инициативе ВВС США[1].

Процесс моделирования в SADT включает сбор информации об исследуемой области, документирование полученной информации и представление ее в виде модели и уточнение модели. Кроме того, этот процесс подсказывает вполне определенный путь выполнения согласованной и достоверной структурной декомпозиции, что является ключевым моментом в квалифицированном анализе системы. SADT уникальна в своей способности обеспечить как графический язык, так и процесс создания непротиворечивой и полезной системы описаний.

В IDEF0 система представляется как совокупность взаимодействующих работ (или функций). Связи между работами определяют технологический процесс или структуру взаимосвязи внутри организации. Модель SADT представляет собой серию диаграмм, разбивающих сложный объект на составные части.

Каждый блок IDEF0-диаграммы может быть представлен несколькими блоками, соединенными интерфейсными дугами, на диаграмме следующего уровня. Эти блоки представляют подфункции (подмодули) исходной функции. Каждый из подмодулей может быть декомпозирован аналогичным образом. Число уровней не ограничивается, зато рекомендуется на одной диаграмме использовать не менее 3 и не более 6 блоков.

Диаграммы потоков данных DFD (Data Flow Diagrams)

Диаграммы потоков данных используются для описания движения документов и обработки информации как дополнение к IDEF0. В отличие от IDEF0, где система рассматривается как взаимосвязанные работы и стрелки представляют собой жесткие взаимосвязи, стрелки в DFD показывают лишь то, как объекты (включая данные) движутся от одной работы к другой. DFD отражает функциональные зависимости значений, вычисляемых в системе, включая входные значения, выходные значения и внутренние хранилища данных. DFD - это граф, на котором показано движение значений данных от их источников через преобразующие их процессы к их потребителям в других объектах[1].

DFD содержит процессы, которые преобразуют данные, потоки данных, которые переносят данные, активные объекты, которые производят и потребляют данные, и хранилища данных, которые пассивно хранят данные.Методология объектного проектирования на языке UML (UML-диаграммы)

При моделировании сложной информационной системы, как правило, ее делят на части. Каждую частьв дальнейшем рассматривают и разрабатывают отдельно, при этом используется либо функциональное деление системы, либо применяется объектная декомпозиция.

При объектной декомпозиции система разбивается на объекты или компоненты, которые взаимодействуют друг с другом, обмениваясь сообщениями. Сообщения описывают или представляют собой некоторые события. Получение объекты сообщения активизируют его и побуждают выполнять определенные действия. Инициаторами событий могут быть не только объекты системы, но и ее внешнее окружение, например, пользователи.

Объекты имеют определенные свойства и методы.

В объективно-ориентированных моделях применяется совершенно другой принцип проектирования ИС. Сначала выделяются классы объектов, а далее в зависимости от возможных состояний объектов (жизненного цикла объектов) определяются методы обработки (функциональные процедуры), что обеспечивает наилучшую реализацию динамического поведения информационной системы. То есть главным структурообразующим компонентом является класс объектов с набором функций, которые могут обращаться к атрибутам этого класса.

В настоящее время для объектно-ориентированного моделирования предметной области используется унифицированный язык моделирования UML (Unified Modeling Language), который является стандартом по объектно-ориентированным технологиям, в нем разработаны графические методы моделирования предметной области.

Совокупность объектов, имеющих общий набор свойств и характеризующихся одинаковым поведением, называются классом. Классы могут строиться по иерархическому принципу, когда один класс может быть подклассом другого класса. Из определения класса следует, что каждый класс объект является экземпляром одного определенного класса.

Как правило, в качестве критерия при выборе методики моделирования предметной области рассматривается степень ее динамичности. Для более регламентированных задач больше подходят функциональные модели, для более адаптивных бизнес-процессов (управления рабочими потоками, реализации динамических запросов к информационным хранилищам) – объективно-ориентированные модели.

На начальном этапе проектирования строится укрупненную диаграмма деятельности организации в виде диаграммы вариантов использования. Диаграммы прецендентов (вариантов использования) – диаграммы поведения, на которой показаны множества прецедентов, которые представляют собой законченную последовательность действий и субъектов (исполнителей) и отношения между ними.

Для детального описания выполнения бизнес-процессов используются диаграммы взаимодействия. Диаграммы взаимодействия подразделяются на диаграммы последовательностей – диаграмма поведения, на которой показано взаимодействие и подчеркнута временная последовательность событий и кооперативные диаграммы – диаграмма поведения, на которой показано взаимодействие и подчеркнута структурная организация объектов, посылающих и принимающих сообщения.

Диаграмма активности (видов деятельности) – отражает динамические аспекты поведения системы, то есть отображается потоки работ во взаимосвязанных вариантов использования. По существу, эта диаграмма представляет собой блок-схему, которая наглядно показывает, как поток управления переходит от одной деятельности к другой.

В диаграмме классов отражается структура базы данных, которая является необходимым компонентомдля создания физической модели предметной области и для дальнейшего развёртывания информационной системы.

Помимо прочего, язык UML применяется для проектирования реляционных БД. Для этого используется небольшая часть языка (диаграммы классов), да и то не в полном объеме. С точки зрения проектирования реляционных БД модельные возможности не слишком отличаются от возможностей ER-диаграмм.

Диаграммой классов в терминологии UML называется диаграмма, на которой показан набор классов (и некоторых других сущностей), не имеющих явного отношения к проектированию БД), а также связей между этими классами. Ограничения могут неформально задаваться на естественном языке или формулироваться на языке объектных ограничений OCL (Object Constraints Language).

Атрибутом класса называется именованное свойство класса, описывающее множество значений, которые могут принимать экземпляры этого свойства. Класс может иметь любое число атрибутов (в частности, не иметь ни одного атрибута).

Операцией класса называется именованная услуга, которую можно запросить у любого объекта этого класса. Операция - это абстракция того, что можно делать с объектом. Класс может содержать любое число операций (в частности, не содержать ни одной операции). Набор операций класса является общим для всех объектов данного класса.

IDEF3

IDEF3 является стандартом документирования технологических процессов, происходящих на предприятии, и предоставляет инструментарий для наглядного исследования и моделирования их сценариев. Сценарием называется описание последовательности изменений свойств объекта, в рамках рассматриваемого процесса (например, описание последовательности этапов обработки детали в цеху и изменение её свойств после прохождения каждого этапа). Исполнение каждого сценария сопровождается соответствующим документооборотом, который состоит из двух основных потоков: документов, определяющих структуру и последовательность процесса (технологических указаний, описаний стандартов и т.д.), и документов, отображающих ход его выполнения (результатов тестов и экспертиз, отчетов о браке, и т.д.). Для эффективного управления любым процессом, необходимо иметь детальное представление об его сценарии и структуре сопутствующего документооборота. Средства документирования и моделирования IDEF3 позволяют выполнять следующие задачи:[1]

* Документировать имеющиеся данные о технологии процесса, выявленные, скажем, в процессе опроса компетентных сотрудников, ответственных за организацию рассматриваемого процесса.
* Определять и анализировать точки влияния потоков сопутствующего документооборота на сценарий технологических процессов.
* Определять ситуации, в которых требуется принятие решения, влияющего на жизненный цикл процесса, например изменение конструктивных, технологических или эксплуатационных свойств конечного продукта.
* Содействовать принятию оптимальных решений при реорганизации технологических процессов. Разрабатывать имитационные модели технологических процессов, по принципу "КАК БУДЕТ, ЕСЛИ..."

Существуют два типа диаграмм в стандарте IDEF3, представляющие описание одного и того же сценария технологического процесса в разных ракурсах. Диаграммы относящиеся к первому типу называются диаграммами Описания Последовательности Этапов Процесса (Process Flow Description Diagrams, PFDD), а ко второму - диаграммами Состояния Объекта в и его Трансформаций Процессе (Object State Transition Network, OSTN).

Сравнительный анализ методологий представлен на таблице 1.

Таблица 1 – Сравнительный анализ методологий

|  |  |  |  |
| --- | --- | --- | --- |
| Методология | Основной фокус | Преимущества | Недостатки |
| IDEF0 | Функции и процессы | Чёткое и подробное описание процессов, многоуровневая детализаци | Сложность в обучении, ограниченные динамические возможности |
| DFD | Потоки данных | Простота и наглядность, четкое разделение данных и процессов | Не подходит для сложных бизнес-правил, ограниченная динамика |
| UML | Объекты и их взаимодействия | Универсальность, гибкость, моделирование статических и динамических аспектов | Избыточность для простых проектов, значительные знания и навыки |
| IDEF3 | Процессы и события | Эффективность для динамических процессов, моделирование сценариев | Сложность для больших систем, требует времени на изучение |

**1.3 Функциональное моделирование**

Целью данной части работы является изложение результатов анализа предметной области и разработка концептуальной модели проектируемой информационной системы.

Концептуальная модель основывается на результатах анализа предметной области и представляет собой ее описание, выполненное с использованием естественного языка, математических выражений, таблиц, графов и других средств.

На рисунке 2 представлена контекстная диаграмма работа Типографии

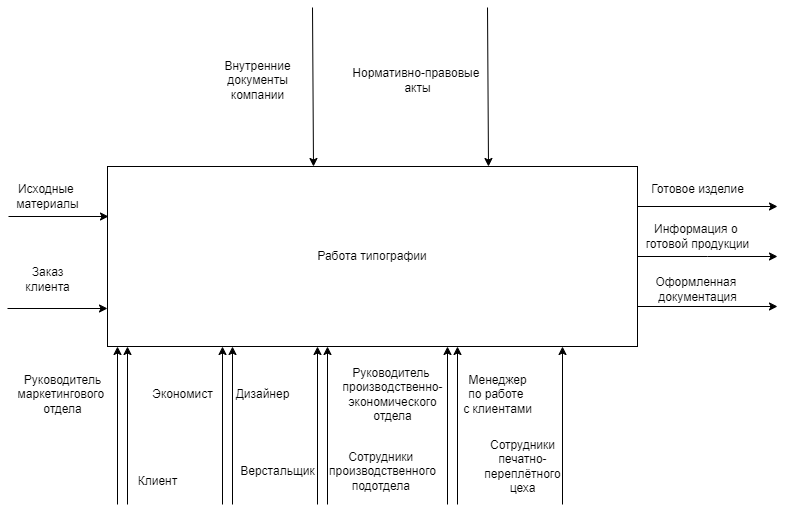


Рисунок 2 Контекстная диаграмма работы типографии

На рисунке 3 изображена диаграмма декомпозиции работы типографии, она состоит из 5 функциональных блоков:

* работа маркетингового отдела;
* работа экономического подотдела;
* работа производственного подотдела;
* работа цеха допечатных процессов;
* работа печатно-переплетного цеха

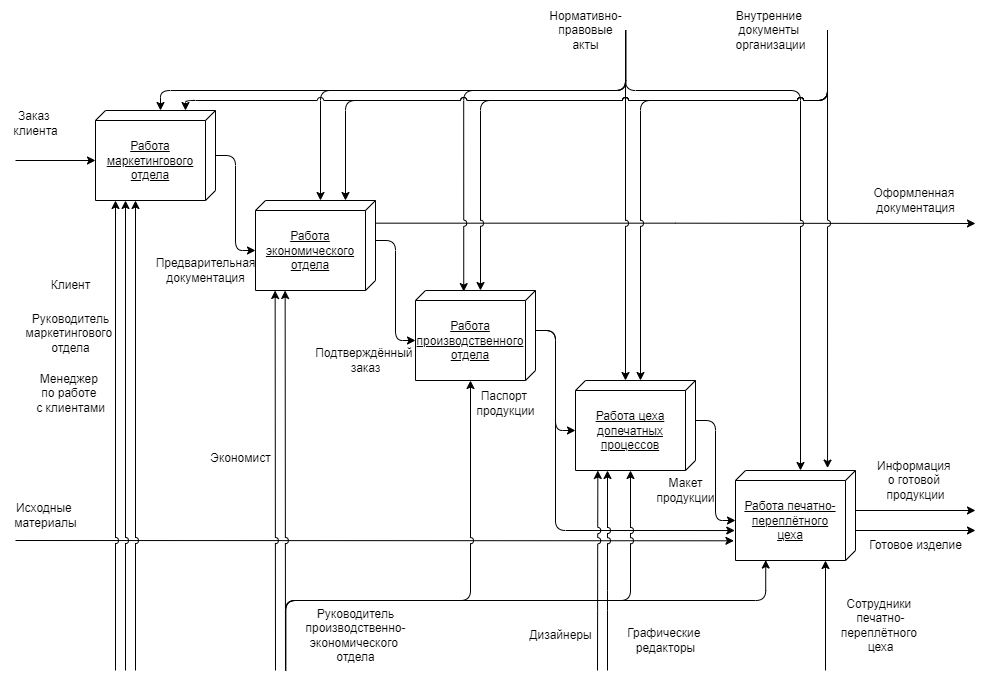


Рисунок 3 Диаграмма декомпозицииработы типографии

Обычно когда поступает заказ, прежде всего, его обрабатывают сотрудники маркетингового отдела. Они согласовывают заказ и оформляют предварительную документацию. Затем, если заказ подтвердили, он переходит в экономический подотдел, где рассчитывают стоимость заказа и оформляют всю необходимую документацию. Далее заказ переходит в производственный отдел, где после составление составления паспорта продукции его передают в цех допечатных процессов. После того как получился готовый макет, его передают в печатно-переплетный цех, в котором получается готовое изделие.

Необходимо рассмотреть работу маркетингового отдела.

На рисунке 4 изображена диаграмма декомпозиции Работа маркетингового отдела

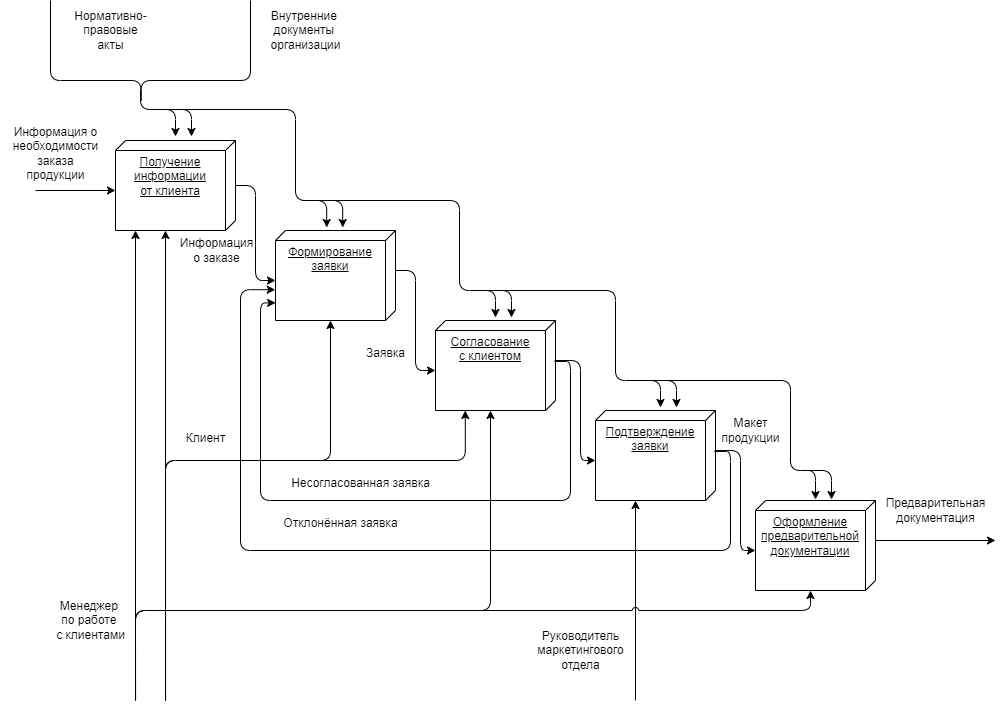


Рисунок 4 диаграмма декомпозиции Работа маркетингового отдела

На данном этапе в отдел поступает информация о необходимости заказа, данный заказ формируется, подтверждается и оформляется предварительная документация, которая передается в экономический подотдел.

Далее рассмотрим диаграмму работы экономического подотдела на рисунке 5.

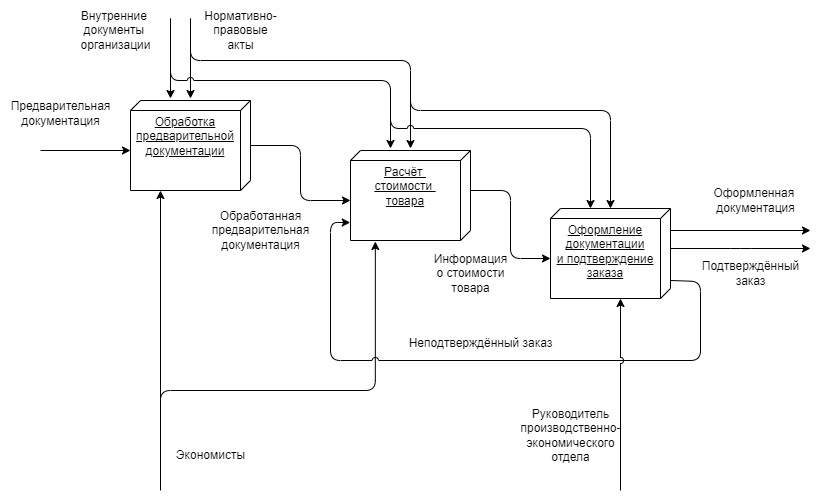


Рисунок 5 Диаграмма декомпозиции работы экономического отдела

Данная диаграмма иллюстрирует процесс, который начинается с получения предварительной документации и заканчивается оформленной готовой документацией и окончательно подтвержденным заказом.

Далее информация передается в производственный подотдел, цех допечатных процессов и печатно-переплётный цех. Декомпозицию этих функциональных блоков рассматривать нет необходимости, поэтому как разрабатываемое веб приложение не будет взаимодействовать с этими отделами.

**1.4 Моделирование элементов структуры веб-приложения с использованием UML**

Вместо того, чтобы сразу разрабатывать функции, объектно-ориентированные модели (ООМ) сначала определяют классы объектов, составляющих систему. Затем, на основе возможных состояний (жизненного цикла) этих объектов, определяются методы обработки (функции), что обеспечивает гибкую реализацию динамического поведения системы.

Ключевым компонентом в ООМ является класс объектов, который обладает набором функций, способных работать с его атрибутами.

Для унифицированного моделирования предметной области в ООМ используется язык UML (Unified Modeling Language). Он является общепринятым стандартом в этой сфере и предлагает набор графических методов для моделирования предметной области.

Другими словами, ООМ фокусируется на классах объектов и их взаимодействии, а не на функциях, а UML – это язык, помогающий визуально представить и описать систему в терминах ООМ.

Например, представим систему онлайн услуг. В ООМ сначала определим классы объектов, например:

* услуга(с такими свойствами, как название, описание, цена и количество),
* клиент (с такими свойствами, как имя, адрес и история заказов),
* заказ (с такими свойствами, как предметы и способ оплаты).

Далее мы определим методы обработки для каждого класса, например:

* товар (добавить в корзину, удалить из корзины, изменить количество),
* клиент (зарегистрироваться, войти в систему, оформить заказ),
* заказ (подтвердить заказ, отменить заказ).

Используя UML, мы можем создать диаграммы, которые иллюстрируют классы объектов, их свойства, методы и связи между ними.

Преимущества ООМ:

* повышает гибкость и модульность системы,
* упрощает разработку и сопровождение системы,
* улучшает читаемость и понятность кода;
* способствует повторному использованию кода.

Совокупность объектов, имеющих общий набор свойств и характеризующихся одинаковым поведением, называются классом. Классы могут строиться по иерархическому принципу, когда один класс может быть подклассом другого класса. Из определения класса следует, что каждый объект является экземпляром одного определенного класса.

Как правило, в качестве критерия при выборе методики моделирования предметной области рассматривается степень ее динамичности. Для более регламентированных задач больше подходят функциональные модели, для более адаптивных бизнес-процессов (управления рабочими потоками, реализации динамических запросов к информационным хранилищам) – объективно-ориентированные модели.

На начальном этапе проектирования строится укрупненная диаграмма деятельности организации в виде диаграммы вариантов использования. Диаграммы прецендентов (вариантов использования) – диаграммы поведения, на которой показаны множества прецедентов, которые представляют собой законченную последовательность действий и субъектов (исполнителей) и отношения между ними.

Для детального описания выполнения бизнес-процессов используются диаграммы взаимодействия. Диаграммы взаимодействия подразделяются на диаграммы последовательностей – диаграмма поведения, на которой показано взаимодействие и подчеркнута временная последовательность событий, и кооперативные диаграммы – диаграмма поведения, на которой показано взаимодействие и подчеркнута структурная организация объектов, посылающих и принимающих сообщения.

Диаграмма активности (видов деятельности) отражает динамические аспекты поведения системы, то есть отображает потоки работ во взаимосвязанных вариантах использования. По существу, эта диаграмма представляет собой блок-схему, которая наглядно показывает, как поток управления переходит от одной деятельности к другой.

В диаграмме классов отражается структура базы данных, которая является необходимым компонентом для создания физической модели предметной области и для дальнейшего развёртывания информационной системы.

Для разработки концептуальной модели выполняется моделирование и анализ (обследование) бизнес-процессов на основе реинжиниринга.

2 **ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ВЕБ-ПРИЛОЖЕНИЯ**

**2.1 Анализ требований и разработка спецификаций**

Требования разделяются на 2 вида: на функциональные и нефункциональные. Функциональные требования описывают необходимые действия, которые может выполнять клиент в веб-приложении, при работе непосредственно с ним. Нефункциональные требования отражают те условия, которые должны исходить от самого веб-приложения.

Функциональные требования веб-приложения:

Поиск услуг: Покупатели должны иметь возможность легко находить нужные услуги. Система должна предоставлять удобный интерфейс для поиска.

Просмотр информации: Для каждой услуги на сайте должна быть доступна подробная информация и актуальная цена.

Добавление в корзину: Клиенты должны иметь возможность без труда добавлять в корзину услуги, которыми хотят воспользоваться. Система должна чётко отображать содержимое корзины, позволяя покупателям просматривать выбранные товары, менять их количество и удалять ненужные позиции.

Оформление заказа: Процесс оформления заказа должен быть максимально простым и понятным. Покупатели должны иметь возможность указать свои данные, выбрать способ оплаты и доставки.

Нефункциональные требования веб-приложения:

Доступность: Интернет-магазин должен быть доступен пользователям круглосуточно, 7 дней в неделю, без перерывов на техническое обслуживание или плановые работы. Это позволит покупателям совершать покупки в любое удобное для них время, независимо от часового пояса или их собственного графика.

Безопасность: Интернет-магазин должен соответствовать самым высоким стандартам безопасности, чтобы защитить личные данные пользователей, платежную информацию и конфиденциальную информацию о заказах. Необходимо использовать надежные методы шифрования, протоколы аутентификации и защиты от взлома, чтобы предотвратить несанкционированный доступ к данным и кибератаки.

Производительность: Интернет-магазин должен работать быстро и отзывчиво, обеспечивая бесперебойную работу даже при пиковых нагрузках. Страницы должны загружаться быстро, а все функции должны быть доступны без задержек. Это позволит покупателям совершать покупки без каких-либо разочарований и сохранить их положительный опыт взаимодействия с магазином.

Простота использования: Интернет-магазин должен быть простым и понятным в использовании даже для неопытных пользователей. Интерфейс должен быть интуитивно понятным, навигация должна быть простой, а все функции должны быть легко доступны. Покупатели должны иметь возможность легко находить нужные им товары, добавлять их в корзину, оформлять заказы и отслеживать их статус без необходимости изучения сложных инструкций или обращения в службу поддержки.

На основе данного анализа требований формируется следующая модель варианта использований представленная на рисунке 6.

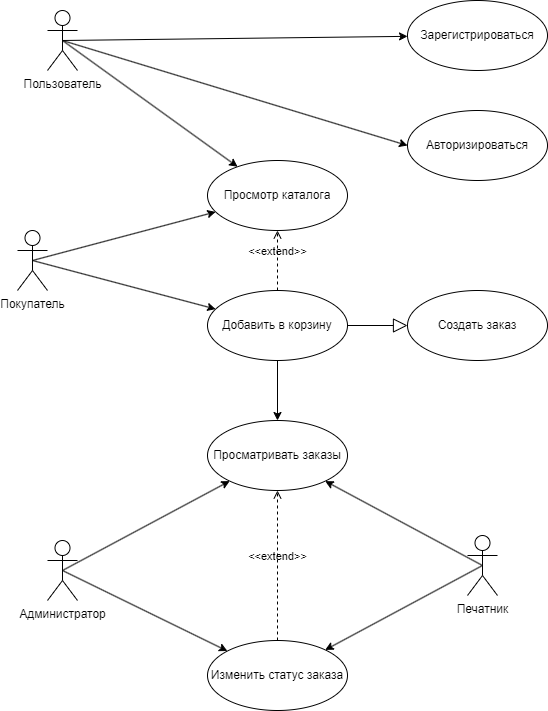


Рисунок 6 - Диаграмма вариантов использования

Список терминов и определений

Пользователь — человек, который имеет, имел, или, возможно, будет иметь доступ в систему для совершения операций.

Операция — совокупность действий, составляющих содержание одного акта бизнес-деятельности. Совокупность операций одного модуля представляет интерфейс взаимодействия клиент-сервер этого модуля.

Описание бизнес-ролей

Покупатель — просматривает возможные варианты заказа, создаёт заказ.

Сотрудник — человек, который имеет доступ в систему для получения информации о заказе для принятия его в работу, изменяет статус заказа.

Администратор — человек, который имеет доступ в систему для совершения операций над аккаунтами пользователей, принимает или отклоняет заказы, убирает или добавляет возможность заказа определённого товара(услуги), изменяет стоимость товаров(услуг).

Варианты использования:

1. Поиск услуги.

Название: Просмотр каталога.

Описание:

1. Система отображает каталог услуг, доступных для заказа.
2. Покупатель может выбрать услугу из списка для просмотра подробной информации.

Предпосылки: Покупатель находится на сайте интернет-магазина.

Постусловия: Покупатель видит каталог услуг.

1. Просмотр информации об услуге:

Название: Просмотр подробной информации об услуге.

Описание:

1. Покупатель выбирает услугу из списка.
2. Система отображает подробную информацию об услуге, включая форму для выбора спецификаций(например, выбор цветности, количества копий и формат бумаги).
3. Покупатель может добавить позицию услуги в корзину.

Предпосылки: Покупатель выбрал услугу из каталога.

Постусловия: Покупатель видит подробную информацию о услуге.

1. Добавление услуги в корзину:

Название: Добавление услуги в корзину.

Описание:

1. Покупатель нажимает кнопку "Добавить в корзину".
2. Система добавляет позицию услуги в корзину покупателя.
3. Система отображает сообщение о том, что услуга добавлена в корзину.

Предпосылки: Покупатель просмотрел информацию об услуге и выбрал необходимые спецификации.

Постусловия: Услуга добавлена в корзину покупателя.

1. Оформление заказа:

Название: Оформление заказа услуги.

Описание:

1. Покупатель переходит в корзину.
2. Покупатель вводит свои данные и отправляет заказ в систему.
3. Система подтверждает заказ и отправляет его на обработку.

Предпосылки: В корзине покупателя есть хотя бы одна услуга.

Постусловия: Заказ оформлен и отправлен на обработку.

1. Регистрация в веб-приложении:

Название: Регистрация пользователя в веб-приложении.

Описание:

1. Пользователь переходит на страницу регистрации.
2. Покупатель вводит данные для своего нового аккаунта (имя, почту, логин, пароль, подтверждение пароля.)
3. После успешной регистрации клиента отправляет сразу на страницу авторизации.

Предпосылки: Пользователь находится на сайте.

Постусловия: Покупатель зарегистрирован и отправлен на авторизацию.

1. Авторизация в веб-приложении:

Название: Авторизация пользователя в веб-приложении.

Описание:

1. Пользователь переходит на страницу авторизации.
2. Пользователь вводит свои данные от аккаунта (логин и пароль).
3. Пользователь нажимает кнопку «Войти».
4. Система авторизует пользователя.

Предпосылки: Пользователь должен быть зарегистрирован.

Постусловия: Пользователь использует веб-приложение со своего личного аккаунта.

**2.2 Проектирование и разработка дизайна интерфейса**

Основные принципы, которые следует учитывать при разработке пользовательского интерфейса сайта типографии:

1. Интуитивность и простота.

Интерфейс должен быть интуитивно понятным, чтобы пользователь мог легко найти и использовать необходимые функции; не должен перегружать память пользователя, чтобы он мог сконцентрироваться на выполнении своих задач;

1. Совместимость с опытом пользователя

Интерфейс должен быть эргономичным и совместимым с опытом пользователя, чтобы он мог легко адаптироваться к новым функциям и элементам.

1. Типографика и визуальная привлекательность

Необходимо учитывать основные правила типографики для сайтов, интерфейсов и веб-дизайна, такие как выбор шрифтов, гарнитуры и определение размеров. Интерфейс должен быть визуально привлекательным, чтобы пользователь чувствовал себя комфортно и хотел возвращаться на сайт.

1. Обеспечение безопасности данных и конфиденциальности

Интерфейс должен обеспечивать безопасность данных, чтобы защищать информацию пользователей от несанкционированного доступа и обеспечивать конфиденциальность, чтобы защищать личную информацию пользователей.

1. Обеспечение доступности

Интерфейс должен обеспечивать доступность для пользователей с ограниченными возможностями, чтобы быть максимально инклюзивным.

1. Обеспечение доступности для всех устройств, браузеров и операционных систем

Интерфейс должен обеспечивать доступность для всех устройств, чтобы быть доступным для пользователей с различных устройств, браузеров и операционных систем.

**2.3 Разработка кода веб-приложения и программирование**

Исходя из требований и спецификаций была создана структура базы данных представленная на рисунке 7 .

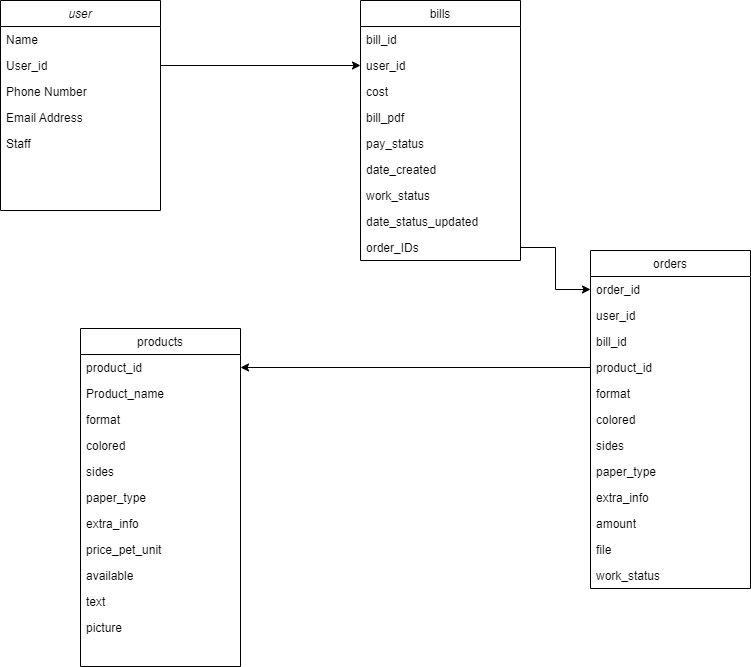


Рисунок 7 Структура базы данных информационной системы Типографии

Взаимодействие сущностей пользователя и системы можно представить на диаграммах процессов регистрации на рисунке 8 и авторизации на диаграмме 8.

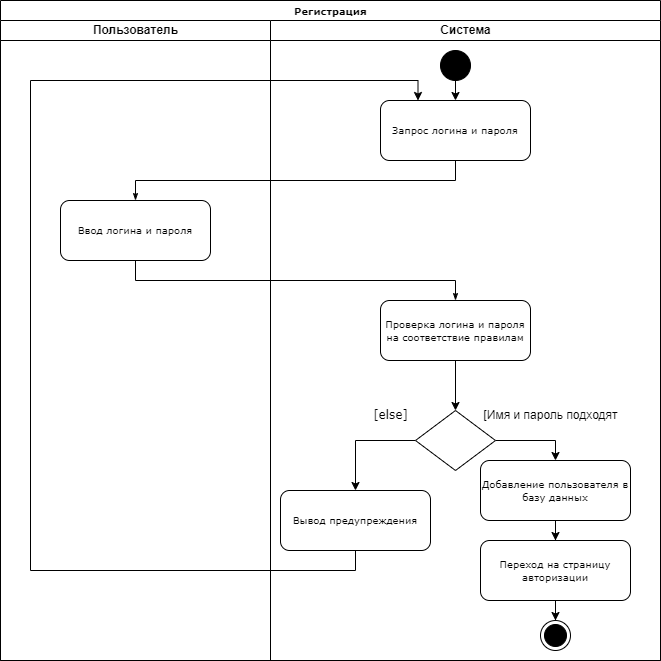


Рисунок 8 Диаграмма процесса регистрации пользователя

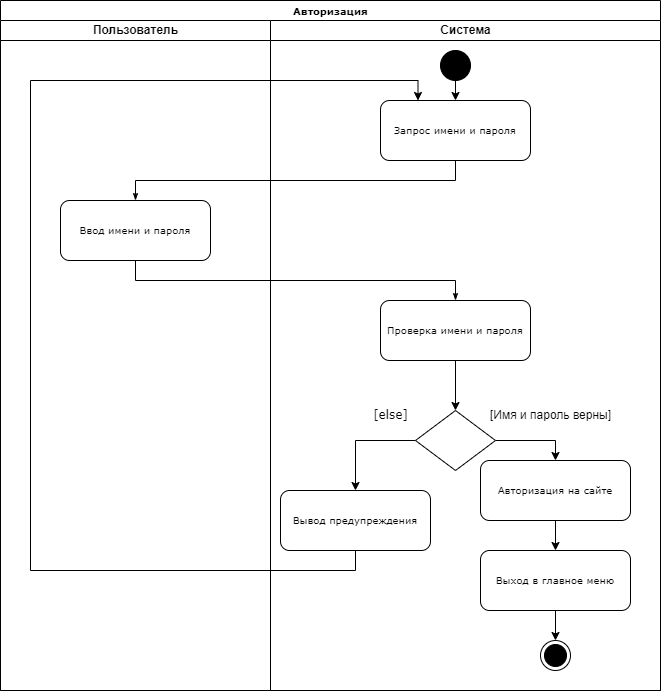


Рисунок 9 Диаграммапроцесса авторизации пользователя

Выбор технического обеспечения и средств разработки

Для реализации хранения выбрана SQLite3 - компактная встраиваемая СУБД с открытым исходным кодом. Для работы SQLite не нужны сторонние библиотеки или службы. По функциональности SQLite3 — программа-клиент для клиент-серверных приложений. С ее помощью можно вводить и передавать запросы к базе данных: создавать, модифицировать, получать или удалять таблицу. Разница в том, что она обращается не к отдельному процессу-серверу, а ко встроенному в приложение движку SQLite. В SQLite3 можно писать SQL-код: утилита отправит запрос к ядру, получит и отобразит результат[5].

В качестве основного языка программирования веб-приложения выбран Python. это высокоуровневый язык программирования, отличающийся эффективностью, простотой и универсальностью использования. Он широко применяется в разработке веб-приложений и прикладного программного обеспечения, а также в машинном обучении и обработке больших данных. За счет простого и интуитивно понятного синтаксиса является одним из распространенных языков для обучения программированию. Этот язык позволяет вести веб-разработку на стороне сервера, потому что его обширная библиотека включает множество решений как раз для реализации сложных серверных функций.

Характеристики Python

Интерпретируемость. В языке Python операторы кода исполняются последовательно с помощью программы-интерпретатора. Если по ходу исполнения программы встречается ошибка, оно сразу же прекращается. Это позволяет Python-разработчику быстро обнаружить и устранить недочеты, но в то же время снижает производительность.

Динамическая типизация. Это автоматическое связывание переменной и типа в момент, когда ей присваивается определенное значение. Такой механизм ускоряет написание программы в различных ситуациях.

Язык высокого уровня. Python по своему синтаксису и грамматике близок к естественным языкам. Благодаря этому программисту с его помощью легче описать различные структуры данных и операции, что также ускоряет и упрощает написание кода. Кроме того, это делает ПО, написанное на Python, менее зависимым от платформы.

Объектно-ориентированность. Написанная на Python программа представляет собой совокупность объектов, каждому из которых присвоены определенный класс и место в иерархии. Таким образом проще управлять процессом программирования, что особенно важно при создании сложных проектов[2].

Преимущества Python

Простота и воспринимаемость. Язык «Питон» специально создан похожим по своему синтаксису на естественные языки (прежде всего английский). Блоки кода в нем отделяются друг от друга пробельными отступами. Это делает код, написанный на Python, более удобочитаемым и понятным для программистов. Минималистичный синтаксис также позволяет реже обращаться к документации и писать меньше строк кода, что, в свою очередь, ускоряет работу над проектом.

Обширные библиотеки. Программисты и сообщество пользователей «Питона» создали для этого языка множество библиотек, в которых содержатся оптимизированные и многократно используемые фрагменты кода для решения практически любых задач. Благодаря этому программисту не нужно писать код полностью с нуля, что также ускоряет работу над проектом или анализом данных.

Совместимость. Интерпретатор Python может интегрироваться с другими языками программирования — например, с Java, C и C++. Благодаря этому Python-разработчики могут расширять функциональность своего проекта, подключая к нему их возможности. Верно и обратное: программы, написанные на указанных языках программирования, можно сочетать с модулями, созданными с помощью «Питона».

Мультиплатформенность. Программы, написанные на Python, адаптированы ко всем существующим аппаратным и программным платформам. Язык используется для работы на разных устройствах, от мейнфреймов до смартфонов и карманных компьютеров под управлением операционных систем Microsoft Windows, MacOS, iOS, Android, UNIX, Linux, Symbian, PalmOS и т.д. Это создает широчайшие возможности для разработки программных продуктов различного назначения, адаптированных под разные платформы.

Мультипарадигменность. На «Питоне» можно программировать в различных парадигмах: объектно-ориентированной, функциональной, императивной, процедурной, структурной и т.д. Это достигается за счет подключения дополнительных библиотек, интерпретаторов и фреймворков. Мультипарадигменность позволяет расширить выразительные способности языка программирования, решать с его помощью больший спектр задач в рамках одного проекта[7].

В качестве фреймворка для создания веб-приложения на языке Python выбран фреймворк Django — это высокоуровневый Python веб-фреймворк, который позволяет быстро создавать безопасные и поддерживаемые веб-сайты. Созданный опытными разработчиками, Django берёт на себя большую часть веб-разработки, разработчик может сосредоточиться на написании веб-приложения без необходимости изобретать велосипед. Фреймворк является бесплатным и имеет открытый исходный код, растущее и активное сообщество, отличную документацию[3].

Django помогает писать программное обеспечение, которое будет:

Полным. Django следует философии «Всё включено» и предоставляет почти всё, что разработчики могут захотеть сделать «из коробки». Поскольку всё, что вам нужно, является частью единого «продукта», всё это безупречно работает вместе, соответствует последовательным принципам проектирования и имеет обширную и актуальную документацию.

Разносторонним. Django может быть использован для создания практически любого типа веб-сайтов — от систем управления контентом и wiki до социальных сетей и новостных сайтов. Он может работать с любой клиентской средой и может доставлять контент практически в любом формате (включая HTML, RSS-каналы, JSON, XML и т. д.). Внутренне он также может быть расширен сторонними компонентами, если это необходимо.

Безопасным. Django помогает разработчикам избежать многих распространённых ошибок безопасности, предоставляя фреймворк, разработанный чтобы «делать правильные вещи» для автоматической защиты сайта. Например, Django предоставляет безопасный способ управления учётными записями пользователей и паролями, избегая распространённых ошибок, таких как размещение информации о сеансе в файлы cookie, где она уязвима (вместо этого файлы cookie содержат только ключ, а фактические данные хранятся в базе данных) или непосредственное хранение паролей вместо хэша пароля. Django, по умолчанию, обеспечивает защиту от многих уязвимостей, включая SQL-инъекцию, межсайтовый скриптинг, подделку межсайтовых запросов и кликджекинг.

Масштабируемым. Django использует компонентную "shared-nothing" архитектуру (каждая её часть независима от других и, следовательно, может быть заменена или изменена, если это необходимо). Чёткое разделение частей означает, что Django может масштабироваться при увеличении трафика, путём добавления оборудования на любом уровне: серверы кеширования, серверы баз данных или серверы приложений. Одни из самых загруженных сайтов успешно масштабировали Django (например, Instagram и Disqus, если назвать только два из них).

Удобным в сопровождении. Код Django написан с использованием принципов и шаблонов проектирования, которые поощряют создание поддерживаемого и повторно используемого кода. В частности, в нём используется принцип «Don't Repeat Yourself» (DRY, «не повторяйся»), поэтому нет ненужного дублирования, что сокращает объём кода. Django также способствует группированию связанных функциональных возможностей в повторно используемые «приложения» и, на более низком уровне, группирует связанный код в модули[6].

Переносным. Django написан на Python, который работает на многих платформах. Это означает, что вы не привязаны к какой-либо конкретной серверной платформе и можете запускать приложения на многих версиях Linux, Windows и Mac OS X. Кроме того, Django хорошо поддерживается многими веб-хостингами, которые часто предоставляют определённую инфраструктуру и документацию для размещения сайтов Django.

Bootstrap 5 — это последняя версия популярного фреймворка для разработки адаптивных и мобильных интерфейсов веб-приложений. Он предоставляет набор инструментов для создания современных, удобных и кроссбраузерных пользовательских интерфейсов. Включает в себя готовые компоненты CSS и JavaScript для типографики, форм, кнопок, навигации и других элементов интерфейса[4].

Основные области применения включают:

* Разработка адаптивных сайтов и веб-приложений.
* Быстрое прототипирование интерфейсов.
* Создание консистентного дизайна благодаря стандартным компонентам и стилям.
* Обеспечение кроссбраузерной совместимости.

Преимущества Bootstrap 5

1. Адаптивность:

Bootstrap 5 позволяет создавать интерфейсы, которые автоматически адаптируются к различным устройствам и экранам, используя сеточную систему и медиазапросы.

1. Быстрое развитие:

С помощью готовых компонентов и стилей разработчики могут быстро создавать сложные интерфейсы, не тратя время на разработку каждого элемента с нуля.

1. Удобство использования:

Хорошо организованная и подробная документация делает Bootstrap доступным даже для начинающих разработчиков.

1. Кроссбраузерная совместимость:

Bootstrap 5 обеспечивает совместимость с основными браузерами, что гарантирует одинаковый вид и поведение интерфейсов на различных платформах.

1. Модульность:

Bootstrap 5 позволяет подключать только необходимые компоненты и утилиты, что помогает оптимизировать производительность и уменьшить размер загружаемых файлов.

В качестве среды разработки выбран Visual Studio Code(VS Code) — это редактор кода для разных языков программирования. Он относительно немного весит, гибкий и удобный. В нем можно писать, форматировать и редактировать код на разных языках.

Преимущества Visual Studio Code

Простота освоения. У редактора интуитивно понятный интерфейс, его легко установить и настроить. А после этого им сразу можно начинать пользоваться. Единственной проблемой поначалу могут стать горячие клавиши и дополнительные настройки — они необычно организованы. Но со временем легко разобраться, особенно если вначале держать под рукой памятку или пользоваться туториалами.

Малый вес. По сравнению с IDE редактор быстро работает и мало весит, не так требователен к ресурсам компьютера. При этом он довольно функционален, а огромное количество расширений позволяет добавлять в него новые возможности.

Кроссплатформенность. Visual Studio Code есть для всех популярных операционных систем, так что с ним можно работать вне зависимости от ОС, которую вы предпочитаете. К тому же это позволяет быстрее и удобнее переносить проекты с одного устройства на другое.

Поддержка разных языков. Есть инструменты, которые предназначены специально для какого-то языка, например Python или C++. Visual Studio Code поддерживает огромное количество языков программирования, и они легко подключаются. Поэтому с его помощью можно вести проекты, написанные на нескольких языках одновременно.

Гибкость. У VS Code множество расширений и удобных функций. Помимо вещей, стандартных для редактора кода, вроде подсветки синтаксиса. Его можно легко и гибко настроить под себя в зависимости от потребностей разработчика и технологий, с которыми он работает.

**2.4 Тестирование и отладка**

Тестирование программного обеспечения (ПО) — это процесс выявления ошибок в программном обеспечении. Современные методы тестирования не позволяют однозначно и полностью устранить все дефекты и ошибки и установить корректность функционирования анализируемой программы. Все существующие методы тестирования действуют в рамках формального процесса проверки исследуемого или разрабатываемого ПО. Формальная проверка или верификация может доказать, что дефекты отсутствуют с точки зрения используемого метода, но нет никакой возможности точно установить или гарантировать отсутствие дефектов в программном продукте, учитывая человеческий фактор на всех этапах жизненного цикла ПО.

Методология функционального тестирования

Функциональное тестирование — это тип тестирования, который проверяет работу ПО в соответствии с его функциональными требованиями.

Основные преимущества функционального тестирования:

Имитирует фактическое использование системы: Функциональное тестирование позволяет тестировать ПО в условиях, близких к реальным, что помогает выявить проблемы, которые могут возникнуть у конечных пользователей.

Выявление системных ошибок: Позволяет своевременно выявить системные ошибки, избегая множества проблем при работе ПО в дальнейшем.

Экономия за счет раннего выявления ошибок: Исправление ошибок на ранних этапах жизненного цикла ПО обходится дешевле и занимает меньше времени.

Тестирование клиентской части и административной панели

Для веб-приложения типографии, разработанного с помощью Python Django, необходимо провести тестирование как клиентской части, так и административной панели. Эти два компонента имеют разные функциональные требования и сценарии использования, поэтому тестирование для них будет отличаться. Тестируемые аспекты их описание, методы и инструменты представлены в таблице 2 Тестирование клиентской части и таблице 3 тестирование административной части

Таблица 2 – Тестирование клиентской части

|  |  |  |  |
| --- | --- | --- | --- |
| Тестируемый аспект | Описание | Методы тестирования | Инструменты |
| Функциональность | Проверка соответствия функциональных требований | Функциональное тестирование | Selenium, PyTest |
| Юзабилити | Оценка удобства использования интерфейса | Тестирование удобства использования | Пользовательское тестирование, A/B тестирование |
| Кросс-браузерность | Проверка работы приложения в разных браузерах | Кросс-браузерное тестирование | BrowserStack, Sauce Labs |
| Отзывчивость | Проверка корректного отображения на различных устройствах и экранах | Тестирование отзывчивости | Chrome DevTools, Responsive Design Mode |
| Производительность | Оценка времени загрузки страниц и отклика | Нагрузочное тестирование | JMeter, Locust |
| Безопасность | Проверка на наличие уязвимостей | Тестирование безопасности | OWASP ZAP, Burp Suite |

Таблица 3 – Тестирование административной панели

|  |  |  |  |
| --- | --- | --- | --- |
| Тестируемый  аспект | Описание | Методы  тестирования | Инструменты |
| Функциональность | Проверка функциональных возможностей админ-панели (CRUD операции, фильтрация, поиск) | Функциональное тестирование | Selenium, Django Test Framework |
| Права доступа | Проверка корректности работы системы прав доступа | Тестирование безопасности | PyTest, Django  Test Framework |
| Производительность | Оценка производительности админ-панели при высоких нагрузках | Нагрузочное тестирование | JMeter, Locust |
| Юзабилити | Проверка удобства использования для администраторов | Тестирование удобства использования | Пользовательское тестирование, A/B тестирование |
| Безопасность | Проверка на уязвимости, такие как SQL инъекции, XSS и CSRF | Тестирование безопасности | OWASP ZAP, Burp Suite |

Тестирование и отладка являются критически важными этапами разработки веб-приложения. Функциональное тестирование позволяет выявить системные ошибки и улучшить качество ПО, имитируя его реальное использование. Тестирование клиентской части и административной панели требует различных подходов и инструментов, но в совокупности они обеспечивают надежность и стабильность веб-приложения.

**2.5 Разработка руководства пользователя**

1. Введение
   1. Область применения

Данное руководство пользователя предназначено для ознакомления с веб-приложением типографии. В нем описаны структура, функции и возможности приложения, а также даны инструкции по его использованию.

* 1. Краткое описание возможностей

Приложение позволяет пользователям:

Просматривать каталог доступных услуг для печати.

Создавать онлайн заказ печати в типографии.

Оплачивать заказы онлайн.

Отслеживать статус заказа.

Связываться со службой поддержки.

* 1. Уровень подготовки пользователя

Данное руководство предполагает, что пользователь имеет базовые навыки работы с компьютером и Интернетом.

* 1. Перечень эксплуатационной документации, с которой необходимо ознакомиться пользователю

Данное руководство пользователя.

Документация на сайте разработчика приложения.

1. Назначение и условия применения ПО

Веб-приложение предназначено для оказания услуг печати конечным пользователям.

Приложение может использоваться на любых устройствах с доступом к Интернету, включая компьютеры, планшеты и смартфоны.

1. Подготовка к работе
   1. Состав и содержание дистрибутивного носителя данных

Веб-приложение не требует установки на компьютер. Доступ к нему осуществляется через браузер в сети интернет.

* 1. Порядок загрузки данных и программ

Для работы с приложением необходимо иметь браузер с поддержкой cookies.

* 1. Порядок проверки работоспособности

Для проверки работоспособности приложения необходимо открыть веб-приложение на локальном или удалённом сервере и попробовать выполнить любую из доступных операций.

1. Описание операций
   1. Регистрация пользователя

Для регистрации пользователя необходимо перейти на страницу регистрации и ввести следующие данные:

* ФИО
* Email
* Логин
* Пароль
* Подтверждение пароля
  1. Авторизация пользователя

Для авторизации пользователя необходимо ввести его логин и пароль на странице авторизации.

* 1. Просмотр каталога услуг

Для просмотра каталога услуг необходимо перейти на страницу "Услуги". На этой странице можно выбрать услугу и перейти на страницу для просмотра подробностей и вариантах заказа.

* 1. Оформление заказа

Для оформления заказа необходимо выбрать необходимые варианты услуги(например, выбрать цветность, количество копий и формат бумаги) и добавить услугу в корзину и перейти к оформлению заказа. На странице оформления заказа(корзине) необходимо:

Выбрать способ оплаты.

Подтвердить заказ.

* 1. Оплата заказа

Оплатить заказ можно следующими способами:

Наличными или банковской картой при получении заказа.

* 1. Связь со службой поддержки

Для связи со службой поддержки необходимо воспользоваться формой обратной связи.

**ЗАКЛЮЧЕНИЕ**

В результате выполнения дипломного проекта было разработано веб-приложение для онлайн заказа в типографии.

Разработанное приложение имеет следующую структуру… «логическое описание работы веб приложения».

Функции, которые существуют в приложении.

Регистрация и авторизация пользователей

Просмотр информации об организации и каталогов услуг доступных для заказа

Просмотр информации об услуге, выбор спецификаций, возможность создания онлайн заказа в типографии

Веб-приложение реализовано с использованием следующих технологий и языков: HTML5, CSS, Bootstrap5, Python, Python Django, SQLite3.

Таким образом, разработанное веб-приложение автоматизирует заказ печати в типографии.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Резник, В.Г. Проектирование информационных систем [Текст]: учебное пособие / В.Г. Резник. – Томск : Томск. гос. ун-т систем упр. и радиоэлектроники, 2022. – 180 с. .
2. Задорожный, С.С. Объектно-ориентированное программирование на языке Python [Текст]: учебно-методическое пособие / С.С. Задорожный, Е.П. Фадеев. – Физический факультет МГУ им. М. В. Ломоносова, 2022. – 40с.
3. Django Project Documentation [Электронный ресурс] (дата обращения: 30.05.2024). – Режим доступа: https://docs.djangoproject.com/
4. Bootstrap Documentation [Электронный ресурс] (дата обращения: 30.05.2024). – Режим доступа: https://getbootstrap.com/docs/
5. SQLite Documentation [Электронный ресурс] (дата обращения: 30.05.2024). – Режим доступа: https://sqlite.org/docs.html
6. W3Schools: Web Development Tutorials [Электронный ресурс] (дата обращения: 30.05.2024). – Режим доступа: https://www.w3schools.com/
7. Mozilla Developer Network (MDN) Web Docs [Электронный ресурс] (дата обращения: 30.05.2024). – Режим доступа: https://developer.mozilla.org/