|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |

Институт Информационных технологий

Кафедра Математического обеспечения и стандартизации информационных технологий

**Отчет по практическим работам №5-8**

по дисциплине «Системная и программная инженерия»

|  |  |
| --- | --- |
| **Выполнил:**  Студенты группыИКБО-20-19 | Городнов С.А.  Ильин А.Ю.  Московка А.А.  Николаев-Аксёнов И.С. |
| **Проверил:** | ассистент Трушин С.М. |

МОСКВА 2022 г.

**Оглавление**

[Практическая работа №5 3](#_Toc101127749)

[Практическая работа №6 8](#_Toc101127750)

[Практическая работа №7 12](#_Toc101127751)

[Практическая работа №8 21](#_Toc101127752)

[Заключение 31](#_Toc101127753)

Практическая работа №5

**Цель работы**

Получить навыки по анализу бизнес-процессов, при помощи их моделирования в нотациях BPMN или IDEF0. А также навыки по проектированию структуры проекта посредством построения диаграммы классов и диаграммы объектов в нотации UML.

**Задание на 5 практическую работу**

1. Построить структурные диаграммы своего проекта
   1. Диаграмма классов
   2. Диаграмма объектов
2. Разобрать процесс работы внутри проекта в нотации по выбору, IDEF0 или BPMN 2.0. Построить собственный процесс в выбранной нотации.

**Выполнение задания**

Построенные структурные диаграммы (диаграмма классов и диаграмма объектов) (Рисунок 1-2):

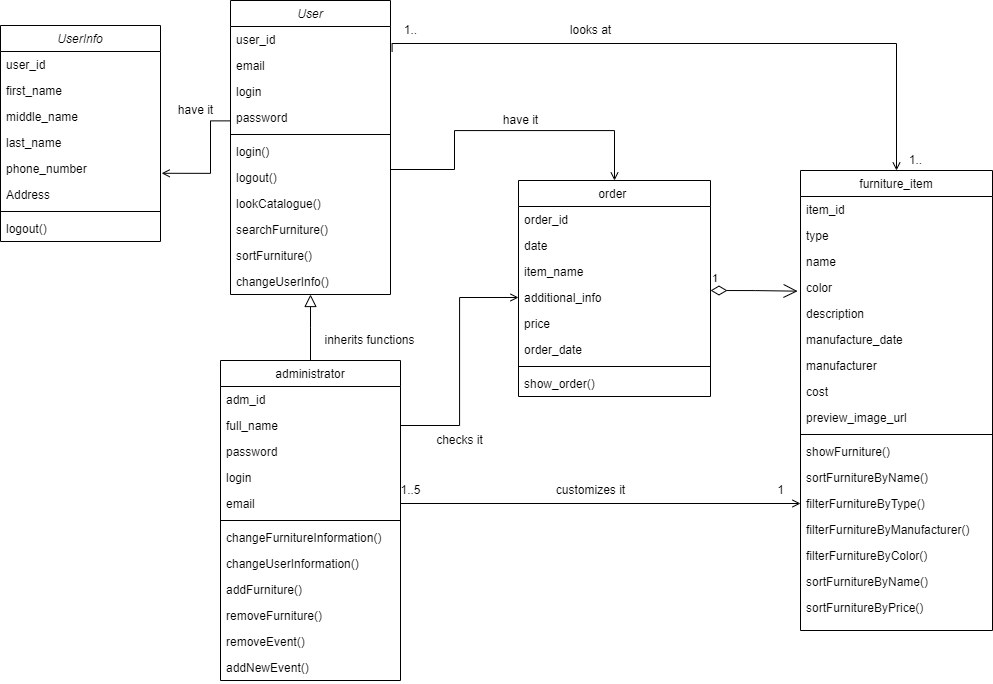


Рисунок 1 – Диаграмма классов разрабатываемой системы

На данной диаграмме имеются следующие связи:

1. User-Userinfo – пользователь имеет личные данные, отличные от данных входа.
2. User – furinture\_item - пользователи могут смотреть на товары. Связь является типом "многие ко многим" так как несколько пользователей могут смотреть на различные товары.
3. User-order – Связь один ко многим, пользователь может заказывать различные товары.
4. Administrator – order - администратор может проверять заказы на ошибки и прочее.
5. Administrator – user - администратор в дополнение к своим полям имеет поля пользователя.
6. Administrator – furniture\_item - Администратор может изменять, добавлять, удалять различные товары, связь является "Многие ко многим" с ограничениями от 1 до 5 со стороны администратора, по причине отсутствия необходимости в большем числе администраторов.
7. Order - furniture\_item – связь, показывающая что один заказ содержит в себе furniture\_item.

Примечание: Связи от администратора до товара напрямую и через order не образуют собой петлю, так как в первом случае администратор редактирует товар независимо от order (заказа), а во втором – проверяет заказ

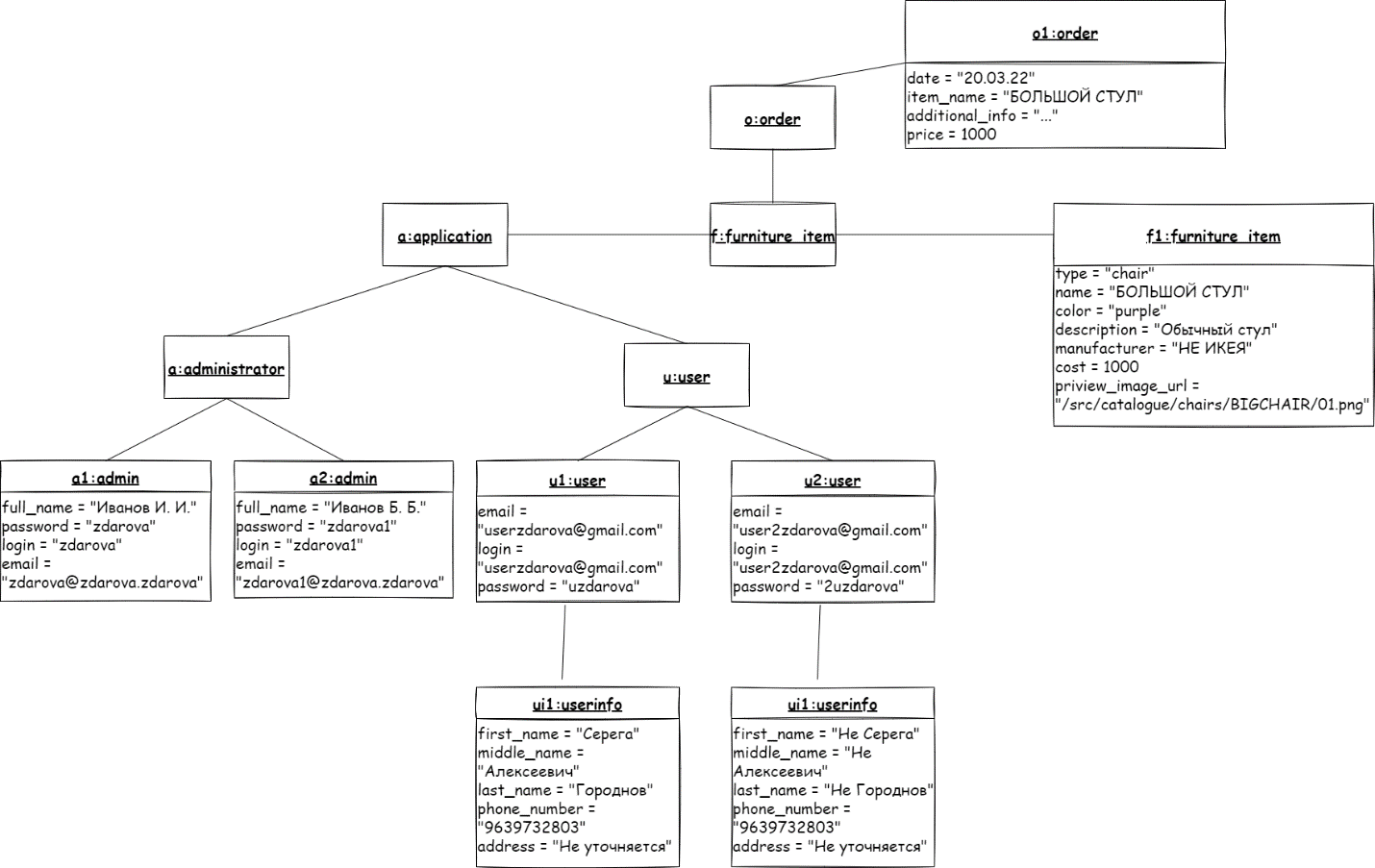


Рисунок 2 – Диаграмма объектов разрабатываемой системы

Выбранный и построенный в нотации IDEF0 основной процесс, протекающий внутри разрабатываемой информационной системы (Рисунок 3-4):

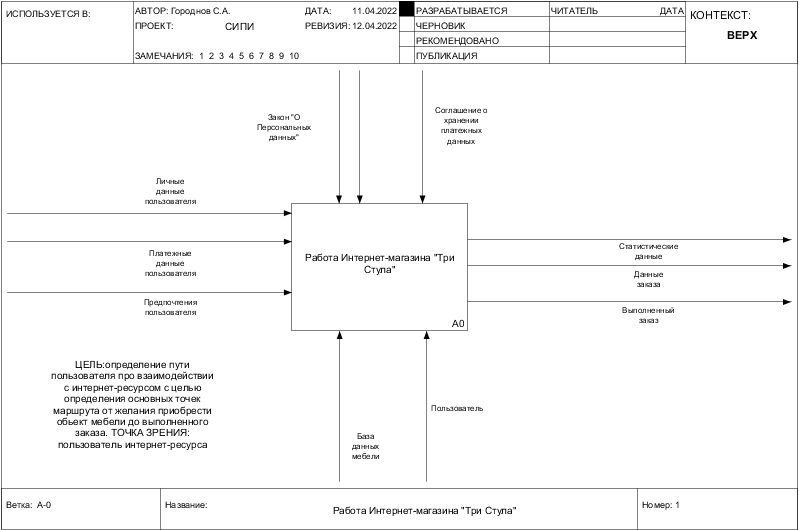


Рисунок 3 – Диаграмма основного процесса «Работа Интернет-магазина “Три Стула”» в нотации IDEF0

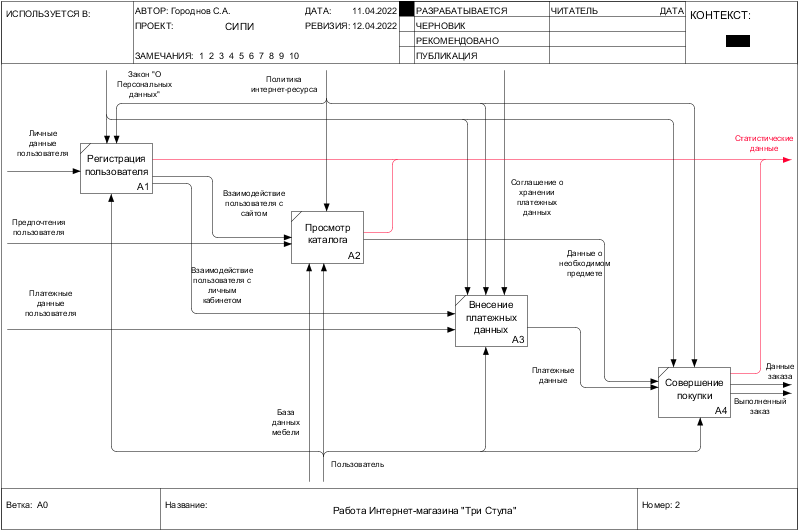


Рисунок 4 – Диаграмма декомпозиции основного процесса «Работа Интернет-магазина “Три Стула”» в нотации IDEF0

Практическая работа №6

**Цель работы:**

Получить навыки по определению информационного взаимодействия системы, отражению процессов обработки, хранения и передачи информации между компонентами системы и внешней средой, а также получить навыки по проектированию логической модели базы данных для своей разработки.

Задание на 6 практическую работу

1. Посредством анализа прошлых диаграмм и изучив материал практики, выполнить построение диаграммы в нотации DFD для своего проекта
2. Словесно описать информационное взаимодействие компонентов системы
3. Построить нормализованную логическую модель базы данных собственного проекта

**Выполнение задания**

Основной поток данных, протекающий внутри разрабатываемой информационной системы, и построенный поток в нотации DFD проекта (Рисунок 5-6):

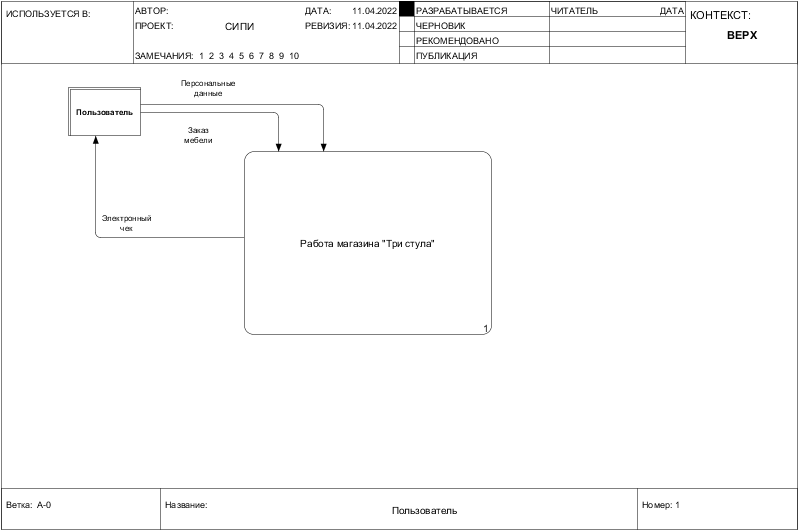


Рисунок 5 – Диаграмма основного потока данных, протекающего внутри разарабатываемой инофрмационной системы, построенная в нотации DFD

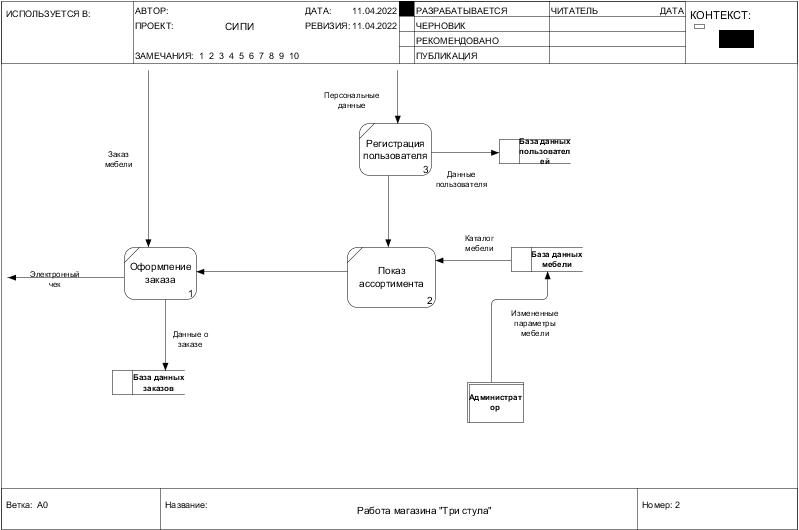


Рисунок 6 – Диаграмма декомпозиции основного потока данных, протекающего внутри разарабатываемой инофрмационной системы, построенная в нотации DFD

Словесное описание информационного взаимодействия компонентов системы:

Когда пользователь обращается к разрабатываемому сервису, он предоставляет персональные данные на портале. Персональные данные используются в процессе Регистрации пользователя, в результате которого Данные пользователя сохраняются в Базе данных пользователей, а также у зарегистрированного пользователя появляется доступ к ассортименту, который наполняет Администратор изменением параметров мебели и добавлением новых записей в действующей базе данных моделей. После выбора желаемой к приобретению мебели зарегистрированный пользователь участвует в процессе Оформления заказа. В результате Оформления заказа в Базу данных заказов вносятся новые данные о заказе, а также печатается электронный чек, который возвращается к пользователю, оформившему заказ.

Нормализованная логическая схема базы данных (Рисунок 7):

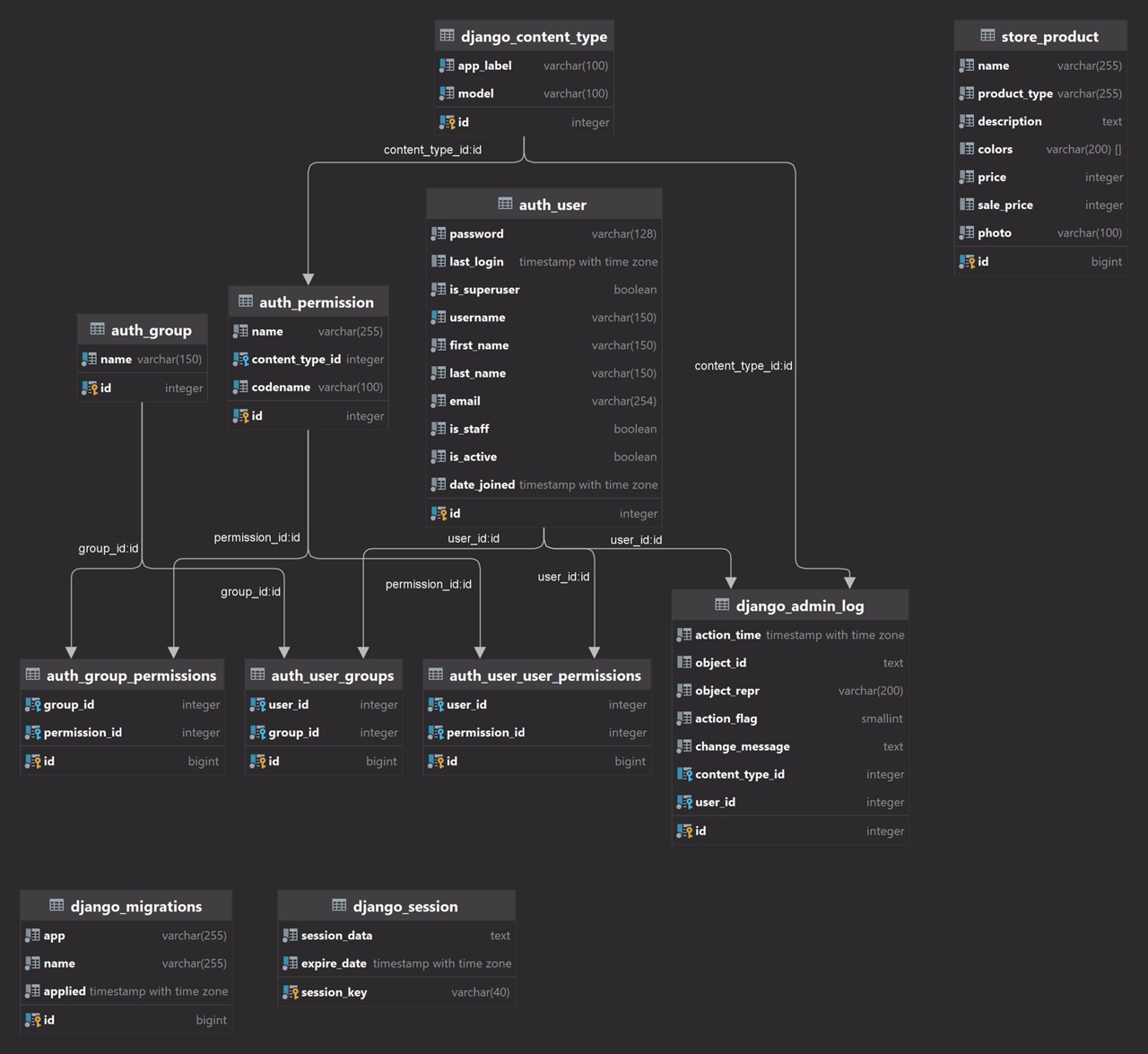


Рисунок 7 – Нормализованная логическая схема базы данных разрабатываемой информационной системы

Практическая работа №7

**Цель работы**

Получить навыки по описанию архитектуры своего решения, проектированию архитектурной диаграммы, распределению сформированных требований по различным компонентам системы с последующим обновлением матрицы требований.

**Задание на 7 практическую работу**

1. Необходимо описать предлагаемую архитектуру системы и обосновать выбор определенных программных решений для реализации ее компонентов. Программные решения должны включать в себя языки реализации будущей системы, планируемые к применению фреймворки, движки, базы данных и т.д.
2. Построить архитектурную диаграмму своей разработки.
3. Отразить в матрице требований соответствие каждого требования компоненту архитектуры, в котором оно реализуется.

**Выполнение работы**

Описание архитектуры системы и обоснованный выбор программных решений:

Используется трехуровневая архитектура

Трехуровневая архитектура — это широко применяемая архитектура программного обеспечения, в которой приложения разделены на три логических и физических уровня: уровень представления (пользовательский интерфейс), уровень приложения, на котором осуществляется обработка данных, и уровень данных, предназначенный для хранения и управления данными, относящимися к приложению.

Основное преимущество трехуровневой архитектуры заключается в том, что поскольку каждый уровень имеет собственную инфраструктуру, разработкой каждого уровня может заниматься отдельная команда разработчиков. Кроме того, каждый уровень можно обновлять и масштабировать по мере необходимости, не затрагивая другие уровни.

**Уровень представления** (клиент) разрабатывается при помощи HTML, CSS, JavaScript, а также CSS фреймворк Tailwind. Представленные технологии являются стандартом написания клиентской части веб-приложения, также для работы и отображения уровня обмена данных используется Django template language.

**Уровень приложений** (сервер приложений) разработан при помощи языка программирования Python и фреймворка Django, данный фреймворк позволяет создать приложение, основанное на трехуровневой архитектуре быстро и эффективно, также он предоставляет админ-панель и форму авторизации «из коробки».

**Уровнем данных** (Сервер БД) в разрабатываемом приложении является свободная объектно-реляционная система управления базами данных PostgreSQL, ввиду ее надежности, распространенности и относительной легкости использования и внедрения.

**Преимущества трехуровневой архитектуры**

Итак, главным преимуществом трехуровневой архитектуры является логическое и физическое разделение функциональных возможностей. Каждый уровень можно запустить в отдельной операционной системе и серверной платформе (веб-сервер, сервер приложений, сервер базы данных), наилучшим образом соответствующей его функциональным требованиям. Поскольку каждый уровень выполняется по крайней мере на одном выделенном аппаратном или виртуальном сервере, уровни можно настраивать и оптимизировать независимо друг от друга.

Другие преимущества (по сравнению с одноуровневой и двухуровневой архитектурой):

* Более быстрая разработка: поскольку разработкой уровней одновременно занимаются разные команды, организация может быстрее вывести приложение на рынок, а программисты могут использовать наилучшим образом подходящие языки и инструменты для каждого уровня.
* Улучшенная масштабируемость: каждый уровень можно масштабировать независимо от других в соответствии с потребностями.
* Повышенная надежность: сбой одного из уровней не повлияет на доступность и производительность других уровней.
* Высокий уровень безопасности: поскольку уровень представления и уровень данных не могут напрямую взаимодействовать друг с другом, хорошо спроектированный уровень приложения может выполнять роль своего рода внутреннего брандмауэра, предотвращая внедрение кода SQL и другие вредоносные действия.

Архитектурная диаграмма разрабатываемого проекта:

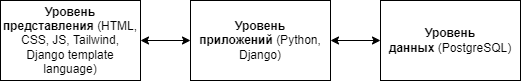


Рисунок 8 – Архитектурная диаграмма разрабатываемого проекта

Матрица требований:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **№** | **Требование** | **Суть** | **Автор** | **Ссылки** | **Критерий проверки** | **Компоненты архитектуры** |
| 1 | Веб-интерфейс | | | | | |
| 1.1 | Регистрация пользователя | Приложение должно иметь функцию регистрации нового пользователя | Николаев-Аксенов И. С. | https://tristula.ga/registration/ | Регистрация нового пользователя | Ввод данных реализован на устройстве клиента в браузере, обработка происходит на сервере, хранение в единой базе данных. |
| 1.2 | Авторизация пользователя | Приложение должно иметь функцию авторизации зарегистрированного пользователя | Николаев-Аксенов И. С. | https://tristula.ga/login/ | Авторизация зарегистрированного пользователя | Ввод данных реализован на устройстве клиента в браузере, обработка происходит на сервере, хранение в единой базе данных. |
| 1.3 | Демонстрация каталога | Приложение должно иметь функцию демонстрации каталога товаров | Николаев-Аксенов И. С. | https://tristula.ga/catalog/ | Открытие каталога товара | Ввод данных реализован на устройстве клиента в браузере, обработка происходит на сервере, хранение в единой базе данных. |
| 1.4 | Демонстрация товара | Приложение должно иметь функцию демонстрации конкретного товара | Николаев-Аксенов И. С. | https://tristula.ga/catalog/{id} | Открытие страницы конкретного товара | Ввод данных реализован на устройстве клиента в браузере, обработка происходит на сервере, хранение в единой базе данных. |
| 1.5 | Добавление товара в корзину | Приложение должно иметь функцию добавление выбранного товара в корзину | Николаев-Аксенов И. С. | https://tristula.ga/cart/ | Добавление товара в корзину | Ввод данных реализован на устройстве клиента в браузере, обработка происходит на сервере, хранение в единой базе данных. |
| 1.6 | Отображение содержимого корзины | Приложение должно иметь функцию демонстрации содержимого корзины пользователю | Николаев-Аксенов И. С. | https://tristula.ga/cart/ | Открытие страницы корзины | Ввод данных реализован на устройстве клиента в браузере, обработка происходит на сервере, хранение в единой базе данных. |
| 1.7 | Осуществление заказа | Приложение должно иметь функцию осуществления заказа посредством получения данных от пользователя и последующим их сохранением | Николаев-Аксенов И. С. | https://tristula.ga/cart/ | Осуществление заказа | Ввод данных реализован на устройстве клиента в браузере, обработка происходит на сервере, хранение в единой базе данных. |
| 1.8 | Изменение информации о пользователе | Приложение должно иметь функцию редактирование данных о пользователе | Николаев-Аксенов И. С. | https://tristula.ga/account/ | Изменение данных в личном кабинете | Ввод данных реализован на устройстве клиента в браузере, обработка происходит на сервере, хранение в единой базе данных. |
| 1.9 | Отображение событий | Приложение должно иметь функцию отображение страницы событий магазина | Николаев-Аксенов И. С. | https://tristula.ga/events/ | Открытие страницы событий | Ввод данных реализован на устройстве клиента в браузере, обработка происходит на сервере, хранение в единой базе данных. |
| 1.10 | Отправление отзывов и предложений | Приложение должно иметь функцию получения и сохранения отзывов и предложений пользователей | Николаев-Аксенов И. С. | https://tristula.ga/feedback/ | Отправка отзыва и предложения | Ввод данных реализован на устройстве клиента в браузере, обработка происходит на сервере, хранение в единой базе данных. |
| 2 | Правовые нормы регулирование деятельности компании | | | | | |
| 2.1 | Закон "О защите персональных данных" | Программная система, при получении, хранении и обработки персональных данных клиентов и сотрудников компании, должна руководствоваться нормами закона Федерального закона "О персональных данных" от 27.07.2006 N 152-ФЗ (последняя редакция) | ??? | http://www.consultant.ru/document/cons\_doc\_LAW\_61801/ | Прохождение теста на невозможность получения закрытых персональных данных третьими лицами | Выполнение норм должны обеспечивать все компоненты программной системы. |
| 3 | Серверная часть и БД | | | | | |
| 3.1 | Выдерживание нагрузок на сервер | Серверная часть и БД должны выдерживать нагрузки до 1000 одновременно открытых заказов, до 300 пользователей | ??? | https://firstvds.ru/technology/nagruzka-na-server-opredelenie-prichin | Система должна выдерживать тесты по пиковой нагрузке, которые соответствуют данным в описании требования | Увеличение допустимых пиковых нагрузок на сервер достигается с помощью улучшения характеристик серверного оборудования, а также улучшения методов работы с данными |
| 3.2 | Стрессоустойчивость системы к аварийным ситуациям | Время восстановление системы после аварийной ситуации не должно превышать 30 минут | ??? | https://m.habr.com/ru/company/dataline/blog/501848/ | Восстановление серверной части системы не превышает 30 минут | Стрессоустойчивость системы достигается подведением системы к сертификации дата-центра по стандарту Tier от Uptime Institute (уровень не ниже Tier II) |

Практическая работа №8

**Цель работы:**

Получение навыков по оформлению технического задания в соответствии с ГОСТ 34.602-2020 или ГОСТ 19.201-78.

**Задание на 8 практическую работу:**

1. Необходимо обосновать выбор ГОСТа для разработки технического задания, почему был выбран тот или иной стандарт, по каким разделам данный ГОСТ подходит больше и т.д.
2. Составить в соответствии с выбором техническое задание по ГОСТу.
3. Оформить единый отчет по 2 блоку практик (с 5 по 8 практику).

**Выполнение задания**

Обоснованный выбор ГОСТа проекта:

Для оформления ТЗ был выбран **ГОСТ 34.602**-**2020**, поскольку наиболее знаком благодаря другим дисциплинам данного курса и наиболее полно и точно позволяет описать требования к разрабатываемому технологическому и информационному комплексу.

ТЗ по выбранному ГОСТу:

1. Общие сведения:
   1. Полное наименование системы и ее условное обозначение:
      1. Полное наименование системы: «Интернет-магазин мебели “Три стула”»
      2. Условное обозначение системы: «ИММТС»
   2. Наименование разработчиков системы и реквизиты заказчика:
      1. Заказчик – кафедра МОСИТ
      2. Разработчики – студенты группы ИКБО-20-19 Городнов С.А., Ильин А.Ю., Московка А.А., Николаев-Аксёнов И.С.
   3. Основания для разработки АС: практическая работа по дисциплине «Системная и программная инженерия»
   4. Плановые сроки начала и окончания работы по созданию системы:
      1. Начало работ по созданию системы – февраль 2022
      2. Окончание работ по созданию системы – конец мая 2022
   5. Источник финансирования работ по созданию АС: собственные средства разработчиков.
   6. Порядок оформления и предъявления заказчику результатов работ по созданию системы:
      1. К результатам труда разработчика относится:
         1. Оригинальное аппаратное обеспечение;
         2. Оригинальное программное обеспечение;
         3. Уникальные структуры данных;
         4. Типовые проектные решения и особенности построения распределенной системы;
         5. Проектная и рабочая документация.
      2. Результаты передаются заказчику частями по завершении каждой стадии работы по созданию соистемы
2. Назначение и цели создания (развития) системы:
   1. Назначение системы: ИММТС предназначен для просмотра и онлайн покупки мебели.
   2. Цели создания системы:
      1. Увеличение доступности большого разнообразия мебели в сети Интернет;
      2. Повышение числа заказчиков у производителей мебели;
      3. Устранение необходимости покупателям посещать физические магазины и склады мебели с возможностью просмотра мебели на сайте.
3. Характеристика объектов автоматизации:
   1. Краткие сведения об объекте автоматизации: сведения не предъявлены.
   2. Сведения об условиях эксплуатации объекта автоматизации:
      1. ИММТС используется пользователями, администраторами.
      2. Функционирование системы должно происходить в требуемых условиях: при конструктивной температуре, давлении и допустимом уровне запыленности;
4. Требования к системе:
   1. Требования к системе в целом:
      1. Требования к структуре и функционированию системы:
         1. Добавление товара из каталога в корзину;
         2. Опции при выборе товара;
         3. Визуальное сопоставление товаров для различия и сравнения;
         4. Сортировка товаров в каталоге по стоимости;
         5. Наличие адреса магазина;
         6. Возможность клиентов оставлять отзывы в виде сообщений по качеству работы сотрудников или сайта;
      2. Требования к численности и квалификации персонала системы и режиму его работы:
         1. Минимум два системных администратора;
         2. Минимум два разработчика для контроля работы системы и баз данных;
         3. Минимум два front-end разработчика.
         4. Минимум два тестировщика;
         5. Минимум два сотрудника тех. поддержки для приема заявок и обработки сообщений об ошибках;
         6. Минимум один специалист по сетевым технологиям.
      3. Требования к надежности:
         1. Сохранение резервной копии системы раз в неделю;
         2. Сохранение резервной копии баз данных раз в неделю;
         3. Обеспечение бесперебойного электропитания к всем частям системы.
      4. Требования к безопасности:
         1. Шифрование паролей;
         2. Конфиденциальность аккаунтов заказчиков;
      5. Требования к эргономике и технической эстетике:
         1. Интуитивно-понятный интерфейс для пользователей;
         2. Комфортная для глаз цветовая гамма оформления страниц;
      6. Требования к транспортабельности для подвижных АС: требования не представлены.
      7. Требования к эксплуатации, тех. обслуживанию, ремонту и хранению компонентов системы:
         1. Необходимо выделять время на обслуживание и профилактику аппаратных систем комплекса (1 день в месяц).
         2. Сеть энергоснабжения должна иметь следующие параметры: напряжение – 220В; частота – 50Гц.
         3. Для обслуживания и профилактики аппаратных систем комплекса необходимо привлечение инженера-электронщика либо специалиста по сетевым технологиям. Его образование должно быть исключительно высшее техническое, связанное с отладкой локальных или структурированных кабельных сетей. Специалист по плану должен уделять 1 день в месяц обслуживанию аппаратных систем комплекса, либо в случае непредвиденного выхода аппаратных систем из строя по заявке персонала компании.
         4. Специалист по сетевым технологиям с высшим образованием должен проводить обслуживание программных систем комплекса в следующих случаях: выход из строя программных систем; при неправильном использовании программных систем; по плану 1 день в месяц для проведения тестирования программных систем.
      8. Требования по защите информации от несанкционированного доступа: требования не предъявлены.
      9. Требования по сохранности информации при авариях: требования не предъявлены.
      10. Требования к защите от влияния внешних воздействий: требования не предъявлены.
      11. Требования к патентной чистоте: требования не предъявлены.
      12. Требования по стандартизации и унификации: требования не предъявлены.
      13. Дополнительные требования не предъявлены.
   2. Требования к функциям (задачам), выполняемым системой: требования не предъявлены.
   3. Требования к видам обеспечения:
      1. Требования к информационному обеспечению:
         1. В качестве входной информации выступают: данные пользователя, изменения параметров мебели, данные о заказе.
         2. В качестве выходной информации выступают: Каталог мебели.
      2. Требования к лингвистическому обеспечению:
         1. Шрифт ввода-вывода данных - кириллица;
         2. Пользовательский интерфейс должен соответствовать следующим требованиям:
            1. Эффективные интерфейсы должны быть очевидными и внушать своему пользователю чувство контроля. Необходимо, чтобы пользователь мог одним взглядом окинуть весь спектр своих возможностей, понять, как достичь своих целей и выполнить работу.
            2. Эффективные интерфейсы не должны беспокоить пользователя внутренним взаимодействием с системой. Необходимо бережное и непрерывное сохранение работы, с предоставлением пользователю возможности отменять любые действия в любое время.
      3. Требования к программному обеспечению:
         1. На сервере ИММТС должна быть установлена ОС Microsoft Windows не ниже 7 версии
         2. На сервере ИММТС должна быть установлена СУБД PostgreSQL.
         3. На устройствах пользователей должен быть доступ к сети Интернет и иметься браузер, по типу Google Chrome, Safari, Opera.
      4. Требования к техническому обеспечению:
         1. Для функционирования ИС необходимо: локальная вычислительная сеть на основе протокола TCP/IP с пропускной способностью 10/100 Мбит/с.
         2. Сервер должен удовлетворять следующим минимальным требованиям:
            1. процессор Celeron-500MHz или аналогичный,
            2. 1 Gb и более оперативной памяти;
            3. 80 Gb – жесткий диск
            4. Монитор – SVGA;
            5. Клавиатура – 101/102 клавиши;
            6. Манипулятор типа «мышь».
         3. Требования, предъявляемые к конфигурации клиентских станций:
            1. процессор, с тактовой частотой не менее 400 MHz,
            2. 256 Mb оперативной памяти;
            3. Монитор – SVGA;
            4. Клавиатура – 101/102 клавиши;
            5. Манипулятор типа «мышь».
         4. Альтернативно пользователь может использовать мобильное устройство на базе iOS, Android.
      5. Требования к методическому обеспечению:
         1. Необходимо создать инструкцию по регистрации и оформлению заказа для пользователей.
5. Состав и содержание работ по созданию системы:
   1. По окончании работ должен быть предоставлен отчет о проделанных работах и успешной работоспособности системы.
6. Порядок контроля и приемки системы:
   1. Приемка этапа заключается в рассмотрении и оценке проведенного объема работ и предъявленной технической документации в соответствии с требованиями настоящего технического задания.
   2. Ответственность за организацию и проведение приемки системы должен нести заказчик. Приемка системы должна производиться по завершению приемки всех задач системы. При этом необходимо предоставить обеспечение материальной частью (технические средства), проектной документацией и специально выделенным персоналом.
   3. Заказчик должен предъявлять систему ведомственной приемочной комиссии, при этом он обязан обеспечить нормальные условия работы данной комиссии в соответствии с принятой программой приемки.
   4. Завершающим этапом при приемке системы должно быть составление акта приемки.
7. Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие:
   1. Для обеспечения готовности объекта к вводу системы в действие провести комплекс мероприятий:
      1. приобрести компоненты технического и программного обеспечения, заключить договора на их лицензионное использование;
      2. завершить работы по установке технических средств;
      3. провести обучение пользователей.
8. Требования к документированию: отчетные материалы должны включать в себя скриншоты интерфейса системы, листинги наиболее важных элементов кода, а также доступные функции для каждой роли, участвующей в системе.
9. Источники разработки: не представлены.

Заключение

В результате выполнения второго блока практических работ (с пятой по восьмую включительно) были построены диаграмма классов, диаграмма объектов, диаграмма основного процесса в нотации IDEF0, диаграмма основного потока в нотации DFD со словесным описанием информационного взаимодействия компонентов системы, построена нормализованная логическая схема базы данных, описана архитектура системы и обоснован выбор определенных программных решений, построена архитектурная диаграмма разработки, отражены соответствия каждого требования компоненту архитектуры в матрице требований, обоснован выбор ГОСТа для проекта, а также составлено подробное ТЗ по выбранному ГОСТу.