|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | |  |  |  | | МИНОБРНАУКИ РОССИИ | | | | Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | | | |
| Институт информационных технологий (ИТ) | |
| Кафедра инструментального и прикладного программного обеспечения (ИиППО) | |

|  |  |  |  |
| --- | --- | --- | --- |
| **ОТЧЁТ ПО ПРАКТИЧЕСКИМ РАБОТАМ** | | | |
| **по дисциплине «Проектирование информационных систем»**  на тему  **«Интернет-магазин игровой валюты»** | | | |
|  | | | |
| Выполнил студент группы ИКБО-04-21 | | Даурбеков М.И. | |
|  | |  | |
| Принял  *Ассистент* | | Мельников Д.А. | |
| Практические работы выполнены | «\_\_\_»\_\_\_\_\_\_\_2024 г. | | (подпись студента) | |
| «Зачтено» | «\_\_\_»\_\_\_\_\_\_\_2024 г. | | (подпись руководителя) | |
|  |  | |  | |

Москва 2024

Table of Contents

[Введение 3](#_Toc167728197)

[1 Общие сведения 4](#_Toc167728198)

[1.1 Список терминов и определений 4](#_Toc167728199)

[1.2 Описание бизнес-ролей 5](#_Toc167728200)

[2 Требования к системе 6](#_Toc167728201)

[2.1 Требования к системе в целом 6](#_Toc167728202)

[2.2 Требования к функциям (задачам), выполняемым системой 9](#_Toc167728203)

[2.3 Требования к видам обеспечения 10](#_Toc167728204)

[Заключение 12](#_Toc167728205)

[Ответы на вопросы 13](#_Toc167728206)

[Введение 17](#_Toc167728207)

[1 Общие сведения 18](#_Toc167728208)

[**2.4** **Список терминов и определений** 18](#_Toc167728209)

[**2.5** **Описание бизнес-ролей** 18](#_Toc167728210)

[3 Проектирование диаграммы прецедентов 19](#_Toc167728211)

[Реализуем диаграмму прецедентов для информационной системы “Интернет магазин игровой валюты” в нотации UML (рис. 1). 19](#_Toc167728212)

[Заключение 20](#_Toc167728213)

[Ответы на вопросы 21](#_Toc167728214)

[Введение 24](#_Toc167728215)

[**1.1** **Полное наименование системы и ее условное обозначение** 25](#_Toc167728216)

[**1.2** **Номер договора** 25](#_Toc167728217)

[**1.3** **Наименование организаций – Заказчика и Разработчика** 25](#_Toc167728218)

[**1.4** **Основания для разработки системы** 25](#_Toc167728219)

[**1.5** **Плановые сроки начала и окончания работы по созданию системы** 25](#_Toc167728220)

[**1.6** **Источники и порядок финансирования работ** 25](#_Toc167728221)

[**1.7** **Порядок оформления и предъявления заказчику результатов работ по созданию системы** 26](#_Toc167728222)

[**1.8** **Перечень нормативно-технических документов, методических материалов, использованных при разработке ТЗ** 26](#_Toc167728223)

[**1.9** **Список терминов и определений** 27](#_Toc167728224)

[**8.1** **Описание бизнес-ролей** 28](#_Toc167728225)

[**9.1** **Назначение системы** 29](#_Toc167728226)

[**10.1** **Краткие сведения об объекте автоматизации** 30](#_Toc167728227)

[**10.2** **Сведения об условиях эксплуатации объекта автоматизации** 30](#_Toc167728228)

[**11.1** **Требования к системе в целом** 31](#_Toc167728229)

[**11.2** **Требования к функциям (задачам), выполняемым системой** 35](#_Toc167728230)

[**11.3** **Функциональная структура системы** 36](#_Toc167728231)

[**11.4** **Требования к видам обеспечения** 36](#_Toc167728232)

[**14.1** **Приведение поступающей в систему информации к виду, пригодному для обработки с помощью ЭВМ** 42](#_Toc167728233)

[**14.2** **Изменения, которые необходимо осуществить в объекте автоматизации** 42](#_Toc167728234)

[**14.3** **Создание условий функционирования объекта автоматизации, при которых гарантируется соответствие создаваемой системы требованиям, содержащимся в ТЗ** 42](#_Toc167728235)

[**14.4** **Создание необходимых для функционирования системы подразделений и служб** 42](#_Toc167728236)

[**14.5** **Сроки и порядок комплектования штатов и обучения персонала** 43](#_Toc167728237)

[Ответы на вопросы 46](#_Toc167728238)

[Введение 53](#_Toc167728239)

[1 Общие сведения 54](#_Toc167728240)

[**3.1** **Список терминов и определений** 54](#_Toc167728241)

[**3.2** **Описание бизнес-ролей** 55](#_Toc167728242)

[4 Проектирование контекстной диаграммы 56](#_Toc167728243)

[**4.1** **Цель создания ИС** 56](#_Toc167728244)

[Целю создания ИС «Интернет-магазин игровой валюты» является стремление упростить процесс покупки игровой валюты. 56](#_Toc167728245)

[Краткое описание 56](#_Toc167728246)

[**4.2** **Способ создания ИС** 56](#_Toc167728247)

[**4.3** **Средства создания ИС** 56](#_Toc167728248)

[**4.4** **Проектирование контекстной диаграммы функциональной модели ИС** 57](#_Toc167728249)

[Была спроектирована контекстная диаграмма A-0 в нотации IDEF0. 57](#_Toc167728250)

[В качестве управления было выбрано: 57](#_Toc167728251)

[1. ГОСТ Р 57489-2017; 57](#_Toc167728252)

[2. Политика компании. 57](#_Toc167728253)

[Входящие информационные потоки: 57](#_Toc167728254)

[1. Выбранная валюта из каталога; 57](#_Toc167728255)

[2. Способ оплаты; 57](#_Toc167728256)

[3. Данные аккаунты получателя. 57](#_Toc167728257)

[Механизмы: 57](#_Toc167728258)

[1. Авторизованный пользователь; 57](#_Toc167728259)

[2. Поставщик; 57](#_Toc167728260)

[3. Сервер. 57](#_Toc167728261)

[Выходящие информационные потоки: 57](#_Toc167728262)

[1. Результат передачи данных; 57](#_Toc167728263)

[На рисунке 1 представлена контекстная диаграмма информационной системы “Интернет-магазин игровой валюты” в нотации IDEF0. 57](#_Toc167728264)

[**Заключение** 59](#_Toc167728265)

[Введение 60](#_Toc167728266)

[1 Общие сведения 61](#_Toc167728267)

[**4.5** **Список терминов и определений** 61](#_Toc167728268)

[**4.6** **Описание бизнес-ролей** 62](#_Toc167728269)

[5 Проектирование декомпозиции диаграмм 63](#_Toc167728270)

[Реализуем декомпозицию контекстной диаграммы на 2 уровне для информационной системы “Интернет-магазин игровой валюты” в нотации IDEF0 (рис. 1). 63](#_Toc167728271)

[Функциональные блоки: 63](#_Toc167728272)

[ Оплата услуги поставщику; 63](#_Toc167728273)

[ Вход на аккаунт покупателя; 63](#_Toc167728274)

[ Покупка валюты. 63](#_Toc167728275)

[Потоки данных: 63](#_Toc167728276)

[ Выбранная валюта из каталога 63](#_Toc167728277)

[ Способ оплаты 63](#_Toc167728278)

[ Данные аккаунта получателя 63](#_Toc167728279)

[Реализуем декомпозицию диаграммы “Вход на аккаунт покупателя” на 3 уровне для информационной системы “Интернет-магазин игровой валюты” в нотации IDEF0 (рис. 2). 63](#_Toc167728280)

[Функциональные блоки: 64](#_Toc167728281)

[ Проверка оплаты; 64](#_Toc167728282)

[ Ввод данных пользователя. 64](#_Toc167728283)

[Потоки данных: 64](#_Toc167728284)

[ Чек оплаты; 64](#_Toc167728285)

[ Данные аккаунта покупателя. 64](#_Toc167728286)

[Заключение 65](#_Toc167728287)

[Введение 66](#_Toc167728288)

[1 Общие сведения 67](#_Toc167728289)

[**5.1** **Список терминов и определений** 67](#_Toc167728290)

[**5.2** **Описание бизнес-ролей** 67](#_Toc167728291)

[6 Проектирование DFD диаграммы 68](#_Toc167728292)

[Реализуем декомпозицию контекстной диаграммы для информационной системы “Интернет-магазин игровой валюты” в нотации DFD (рис. 1). 68](#_Toc167728293)

[Функциональные блоки: 68](#_Toc167728294)

[ Авторизация; 68](#_Toc167728295)

[ Оформление заказа; 68](#_Toc167728296)

[ Выполнение заказа. 68](#_Toc167728297)

[Хранилище данных: 68](#_Toc167728298)

[ База данных пользователей; 68](#_Toc167728299)

[Внешняя сущность: 68](#_Toc167728300)

[ Модератор. 68](#_Toc167728301)

[Потоки данных: 68](#_Toc167728302)

[ Авторизационные данные пользователя; 68](#_Toc167728303)

[ Авторизированный пользователь; 68](#_Toc167728304)

[ Идентификатор пользователя; 68](#_Toc167728305)

[ Баланс пользователя; 68](#_Toc167728306)

[ Заказ выполнен; 68](#_Toc167728307)

[Реализуем декомпозицию диаграммы “Авторизация” для информационной системы “Интернет-магазин игровой валюты” в нотации DFD (рис. 2). 68](#_Toc167728308)

[Функциональные блоки: 69](#_Toc167728309)

[ Ввод регистрационных данных; 69](#_Toc167728310)

[ Авторизация 69](#_Toc167728311)

[ Проверка уникальности регистрационных данных 69](#_Toc167728312)

[ Создание запроса на добавление пользователя. 69](#_Toc167728313)

[Внешняя сущность: 69](#_Toc167728314)

[ База данных пользователей. 69](#_Toc167728315)

[Потоки данных: 69](#_Toc167728316)

[ Авторизационные данные пользователя; 69](#_Toc167728317)

[ Данные пользователя на проверку; 69](#_Toc167728318)

[ Данные зарегистрированного пользователя; 69](#_Toc167728319)

[ Запрос об уникальности пользовательских данных; 69](#_Toc167728320)

[ Статус уникальности данных; 69](#_Toc167728321)

[ Запрос на создание нового пользователя; 69](#_Toc167728322)

[ Новый пользователь; 69](#_Toc167728323)

[ Авторизированный пользователь. 69](#_Toc167728324)

[Заключение 69](#_Toc167728325)

[Введение 82](#_Toc167728326)

[1 Общие сведения 83](#_Toc167728327)

[**6.1** **Список терминов и определений** 83](#_Toc167728328)

[**6.2** **Описание бизнес-ролей** 83](#_Toc167728329)

[7 Проектирование контекстной диаграммы 84](#_Toc167728330)

[**7.1** **План разработки модели БД** 84](#_Toc167728331)

[План: 84](#_Toc167728332)

[1. Планирование разработки базы данных. Включает определение объёма работ, ресурсов и стоимости проекта. 84](#_Toc167728333)

[2. Определение требований к системе. Включает выбор целей БД, выяснение информационных потребностей различных отделов и руководителей фирмы, требований к оборудованию и программному обеспечению. 84](#_Toc167728334)

[3. Сбор и анализ требований пользователей. На данном этапе необходимо создать модель движения важных нематериальных объектов и уяснить процесс товарообмена. 84](#_Toc167728335)

[4. Проектирование базы данных. Включает концептуальное, логическое и физическое проектирование. 84](#_Toc167728336)

[5. Разработка приложений. Включает проектирование транзакций и пользовательского интерфейса. 84](#_Toc167728337)

[6. Реализация. На данном этапе осуществляется физическая реализация базы данных и разработанных приложений. 84](#_Toc167728338)

[7. Загрузка данных. На этом этапе созданные в соответствии со схемой базы данных пустые файлы, предназначенные для хранения информации, должны быть заполнены данными. 84](#_Toc167728339)

[8. Тестирование. Для оценки законченности и корректности выполнения приложения базы данных может использоваться несколько различных стратегий тестирования. 84](#_Toc167728340)

[9. Эксплуатация и сопровождение. Включает анализ функционирования и поддержку исходного варианта БД, а также адаптацию, модернизацию и поддержку переработанных вариантов. 84](#_Toc167728341)

[**7.2** **Анализ предметной области** 85](#_Toc167728342)

[**7.3** **Проектирование ER-диаграммы** 86](#_Toc167728343)

[**7.4** **Тестовые SQL запросы** 86](#_Toc167728344)

[**Вывод** 87](#_Toc167728345)

[Введение 94](#_Toc167728346)

[1 Общие сведения 95](#_Toc167728347)

[**7.5** **Список терминов и определений** 95](#_Toc167728348)

[**7.6** **Описание бизнес-ролей** 96](#_Toc167728349)

[8 Создание диаграммы состояний 97](#_Toc167728350)

[Реализуем диаграмму состояний для информационной системы “Интернет-магазин игровой валюты” на языке UML (рис. 1). 97](#_Toc167728351)

[Заключение 99](#_Toc167728352)

[Ответы на вопросы 100](#_Toc167728353)

[Введение 103](#_Toc167728354)

[1 Общие сведения 104](#_Toc167728355)

[**8.1** **Список терминов и определений** 104](#_Toc167728356)

[**8.2** **Описание бизнес-ролей** 105](#_Toc167728357)

[9 Проектирование контекстной диаграммы 106](#_Toc167728358)

[**9.1** **Описание ЭСЕ** 106](#_Toc167728359)

[Элементарная семантическая единица (ЭСЕ) – неделимая единица информации, использующаяся в ИС. ЭСЕ представляет собой завершенную контекстную конструкцию, вызываемую в результате поиска по различным атрибутам или в результате тех или иных команд в виде отклика или отчета. В случае исследования настоящей системы за элементарную семантическую единицу была выбрана одна из характеристик поиска, а именно количество мест, возвращаемых на запрос. В нашем примере эта величина меняется случайным образом в пределах от 100 до 500 [продавцов]. 106](#_Toc167728360)

[**9.2** **Наполнение системы** 106](#_Toc167728361)

[**9.3** **Математические расчеты** 107](#_Toc167728362)

[**9.4** **Расчет математического ожидания блока системы** 107](#_Toc167728363)

[**9.5** **Расчет дисперсии информационного блока системы** 107](#_Toc167728364)

[2.6 Расчет среднеквадратичного отклонения 108](#_Toc167728365)

[2.7 Расчет энтропии системы 108](#_Toc167728366)

[**Вывод** 109](#_Toc167728367)

# Введение

Сегодня видеоигры стали занимать у людей большую часть свободного времени. Игры бывают так и платные так и бесплатные, но в основном транзакции игровых предметов проходят с помощью «внутриигровых валют».

Информационная система "Интернет-магазин игровой валюты" будет предоставлять возможность пользователям приобретать игровую валюту большинства игр с помощью онлайн-платежа.

Разработка информационной системы "Интернет-магазин игровой валюты" позволит сократить временные затраты на использования внутриигровых методов покупки валюты.

Целью практической работы является формирование требований к описанной выше системе. Заданием практической работы является описание объекта автоматизации, формулировка основных задач автоматизации объекта, описание основных параметров проектируемой информационной системы, описание путей достижения целей. Кроме того, необходимо сформулировать требования к информационной системе.

1 Общие сведения

### Список терминов и определений

Сервер — часть системы, являющаяся хостом и набором функций для сайта системы.

Клиент — часть системы, отображающая интерфейс сайта системы.

Коэффициент юзабилити — часть пользователей, которым удобно пользоваться сайтом к общему количеству пользователей системы.

Коэффициент интерактивности — часть пользователей, которая пользовалась системой за определенный промежуток времени к ожидаемому количеству пользователей за тот же промежуток времени.

БД (База Данных) — представленная в объективной форме совокупность самостоятельных материалов, систематизированных таким образом, чтобы эти материалы могли быть найдены и обработаны с помощью электронной вычислительной машины (ЭВМ).

ИС (Информационная Система) — система, предназначенная для хранения, поиска и обработки информации, и соответствующие организационные ресурсы (человеческие, технические, финансовые и т. д.), которые обеспечивают и распространяют информацию.

Python — высокоуровневый язык программирования общего назначения с динамической строгой типизацией и автоматическим управлением памятью, ориентированный на повышение производительности разработчика, читаемости кода и его качества, а также на обеспечение переносимости написанных на нём программ.

PEP8 — документ, описывающий соглашение о том, как писать код на языке Python.

Система контейнеризации — это технология абстракции, которая позволяет упаковывать и исполнять приложения вместе со всеми их зависимостями в изолированных средах, называемых контейнерами.

Система оркестрации — система автоматического размещения, координации и управления сложными компьютерными системами и службами.

### Описание бизнес-ролей

Пользователь — авторизованный человек, имеющий расширенный доступ к системе, позволяющая ему просматривать свой профиль и проводить оплату.

Оператор – авторизированный пользователь, имеющий доступ к информации о всех пользователях, служащий для помощи обычным пользователям при использовании системы.

Поставщик – авторизированный пользователь, который имеет собственную страницу на сайте с предоставляемыми услугами.

Администратор — авторизованный пользователь, имеющий полный доступ к системе, позволяющая ему устранять технический ошибки системы.

1. Требования к системе

### Требования к системе в целом

#### Требования к структуре и функционированию системы

Система имеет модульную структуру, включающую в себя следующие модули:

* модуль базы данных;
* модуль «Сервер»;
* модуль «Клиент»;

Система должна выполнять следующие функции:

* осуществлять возможность просмотра информации об клиентах;
* осуществлять транзакции с помощью сторонних онлайн-сервисов;
* осуществлять возможность связи между пользователем и оператором;
* осуществлять возможность создания, изменения, удаления из каталога товаров;
* осуществлять обработку трафика большого объема;
* осуществлять информирование о сбоях и ошибках при обработке транзакций;
* осуществлять возможность регистрации, авторизации пользователей;
* осуществлять возможность администратору назначения ролей пользователям.

#### Требования к численности и квалификации персонала системы и режиму его работы

Для поддержания приложения и эксплуатации интерфейса системы управления персонал должен обладать навыками работы в информационных организациях, общими навыками работы с персональным компьютером, а также опытом общения с клиентами.

#### Показатели назначения

Подсистемы, разработанные и доработанные в рамках данного раздела должны отвечать следующим требованиям:

1. Время на запуск или перезапуск системы и компонентов системы должно составлять не более 60 минут.
2. Коэффициент юзабилити не менее 75%.
3. Коэффициент интерактивности не менее 60%.
4. Коэффициент достоверности информации не менее 100%.
5. В режиме отправки/приемки сообщений, подсистема должна поддерживать интенсивность минимум 100 запросов в секунду при среднем размере пакета 1 Мб. Интенсивность должна быть обеспечена разработанным SDK.

Требования к аппаратной части и масштабированию для обеспечения перечисленных показателей должны быть определены на этапе технического проектирования

#### Требования к надёжности

Программное обеспечение не должно выходить из строя более чем на 60 дней.

Для устойчивости к потере данных необходимо регулярно производить регулярное резервное копирование БД примерно раз в 10 дней.

При этом необходимо использовать соответствующие требованиям программно-аппаратные средств. В частности, можно использовать следующие базовые подходы:

* системное и базовое ПО и технические средства, соответствующие классу решаемой задачи;
* четкое соблюдение правил эксплуатации, а также регламентных сроков обслуживания используемых программно-аппаратных средств;

#### Требования к безопасности

Безопасность данных пользователей должна обеспечиваться шифрованием методом MD5 для паролей от учетных записей и методом AES-256 для обратного шифрования персональной информации, а также обеспечением устойчивости программно-технических средств к возможным кибератакам.

#### Требования к эргономике и технической эстетике

Взаимодействие пользователей с системой должно производиться с помощью интерфейса сайта. Интерфейс сайта должен быть понятным и удобным, не должен быть перегружен графическими.

#### Требования к транспортабельности для подвижных АС

Требования к транспортабельности не предъявляются.

#### Требования к эксплуатации, техническому обслуживанию, ремонту и хранению компонентов системы

Техническим обслуживанием, ремонтом и хранением сервера АС занимаются сетевые инженеры-техники, специалисты по серверным и сетевым технологиям, а также мастера по ремонту компьютерного и другого технического оборудования.

#### Требования к защите информации от несанкционированного доступа

При работе с системой необходимо, чтобы данные могли быть восстановлены из резервных копий в случае потери, информация пользователей была защищена от доступа или модификации несанкционированными лицами.

#### Требования по сохранности информации при авариях

Серверное программное обеспечение системы должно восстанавливать свое функционирование при перезапуске аппаратных средств. Для обеспечения сохранности данных требуется предусмотреть резервное копирование в энергонезависимые ячейки памяти.

#### Требования к защите от влияния внешних воздействий

В случае техногенных или природных аварий серверное программное обеспечение должно продолжать работать на резервных серверах, обеспечивая бесперебойную работу системы.

#### Требования к патентной чистоте

Требования не предъявляются.

#### Требования по стандартизации и унификации

Для реализации статических страниц и шаблонов должен использоваться язык Python. Исходный код должен разрабатываться в соответствии со стандартами PEP8. Для реализации интерактивных элементов клиентской части должен использоваться язык Python. Для реализации динамических страниц должен использоваться язык Python. Приложение должно запускаться на любом устройстве, отвечающем техническим ограничениям, что должно быть реализовано посредством технологий докеризации и оркестрации.

#### Дополнительные требования

Дополнительные требования не предъявляются.

### Требования к функциям (задачам), выполняемым системой

Таблица 2.1 — Требования к функциям, выполняемым системой

Таблица 2.1 – Требования к функциям, выполняемым системой

|  |  |
| --- | --- |
| **Функция** | **Задача** |
| Обнаружение устройств | Система должна обнаруживать активное сетевое оборудование в сети |
| Возможность автоматического обновления списка обнаруженных устройств |
| Сбор информации о состоянии устройств | Сбор данных о состоянии и параметрах работы оборудования. Предусмотреть механизмы для периодического опроса устройств и сбора данных |
| Обработка, хранение и поддержка БД | Создание резервных копий в энергонезависимые ячейки памяти |
| Работа с пользователями | Регистрация пользователя |
| Авторизация пользователя |
| Мониторинг событий и оповещения | Мониторинг событий и аномалий в работе оборудования |
| Оповещение администраторов о возможных проблемах или аварийных ситуациях |
| Отслеживание изменений конфигурации | Автоматическое отслеживание изменения в конфигурации сетевого оборудования |
| Возможность восстановления предыдущих версий конфигураций |
| Анализ производительности и загрузки сети | Анализ производительности сетевого оборудования и загрузки сети |
| Отчёты и статистика по использованию ресурсов |

Продолжение таблицы 2.1

|  |  |
| --- | --- |
| Интеграция с другими системами | Интеграция со смежными информационными системами для обмена данными и событиями |
| Автоматизация задач и сценариев | Поддержка автоматизации рутинных задач и сценариев (автоматическое устранение проблем) |

### Требования к видам обеспечения

#### Требования к математическому обеспечению системы

Математическое обеспечение системы должно обеспечивать реализацию перечисленных в данном документе функций, а также выполнение операций конфигурирования, программирования, управления базами данных и документирования. Алгоритмы должны быть разработаны с учетом возможности получения некорректной входной информации и предусматривать соответствующую реакцию на такие события.

#### Требования к информационному обеспечению системы

Данные, используемые системой, должны храниться в реляционной СУБД. Структура базы данных определяется с учетом особенностей внутренней модели информационной системы.

Информационный обмен между серверной и клиентской частями системы должен осуществляться по протоколу TCP/IP.

#### Требования к лингвистическому обеспечению системы

Информационная система должна быть реализована на русском и английском языке.

#### Требования к программному обеспечению системы

Программное обеспечение клиентской части должно удовлетворять следующим требованиям:

* Установлено браузерное ПО типа Google Chrome, Safari, Yandex Browser;
* Брандмауэр должен пропускать входящий трафик от сайта.

#### Требования к техническому обеспечению системы

Платформа, на которой будет развернута серверная часть системы, должна удовлетворять следующим минимальным требованиям:

* не менее 4 GB оперативной памяти;
* поддерживаемый протокол передачи данных HTTP / HTTPS, скорость передачи данных 15 Мбит/с;

#### 2.3.6 Требования к метрологическому обеспечению системы

Система должна иметь возможность оценивать передачу данных, согласно критериям по ГОСТ Р МЭК 870—5—2.

#### 2.3.7 Требования к организационному обеспечению системы

Рабочее место системного администратора должно быть обустроено согласно ГОСТ 12.1.005-88.

#### 2.3.8 Требования к методическому обеспечению системы

Требования к методическому обеспечению системы не предъявляются

# Заключение

В ходе выполнения работы были сформулированы функциональные и нефункциональные требования к рассматриваемой информационной системе. Сформированы пункты, необходимые для выполнения дальнейших работ, такие как бизнес-роли.

# Ответы на вопросы

1. Что такое требование к информационной системе?

это утверждение, которое описывает ожидаемые характеристики, возможности или поведение информационной системы. Требования могут быть функциональными, не функциональными, пользовательскими, системными или организационными.

1. Какие нормативные документы регламентируют сбор и управление требованиями к информационной системе?

ГОСТ 34.602-89 "Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы"

ГОСТ Р ИСО/МЭК 12207-2010 "Информационная технология. Системная и программная инженерия. Процессы жизненного цикла программного обеспечения"

1. Что является важным при формировании нефункциональных требований к информационной системе?

При формировании нефункциональных требований к информационной системе важно учитывать следующие аспекты:

* Измеримость: нефункциональные требования должны быть измеримыми, чтобы можно было оценить их выполнение. Например, требование "система должна быть надежной" можно измерить с помощью показателя доступности системы.
* Реалистичность: нефункциональные требования должны быть реалистичными и достижимыми с учетом имеющихся ресурсов и технологий. Например, требование "система должна обрабатывать 1000 запросов в секунду" может быть нереалистичным для небольшой системы с ограниченными ресурсами.
* Согласованность: нефункциональные требования должны быть согласованы между собой и не должны противоречить друг другу. Например, требование "система должна быть надежной" может противоречить требованию "система должна быть гибкой и легко адаптируемой".
* Приоритетность: нефункциональные требования должны быть расставлены по приоритетам, чтобы можно было определить, какие требования являются наиболее важными и должны быть выполнены в первую очередь. Например, требование "система должна быть безопасной" может иметь более высокий приоритет, чем требование "система должна быть удобной в использовании".

Кроме того, при формировании нефункциональных требований важно учитывать специфику конкретной информационной системы и ее целевую аудиторию. Например, для системы, которая будет использоваться в критически важной области, такие как здравоохранение или финансы, требования к надежности и безопасности будут иметь более высокий приоритет, чем для системы, которая будет использоваться для развлечения.

1. Какие средства существуют для формирования функциональных требований к информационной системе?

* Анализ предметной области: анализ предметной области позволяет определить основные функции и задачи, которые должна выполнять информационная система. Этот анализ проводится путем изучения бизнес-процессов, интервьюирования экспертов и анализа существующих документов.
* Моделирование бизнес-процессов: моделирование бизнес-процессов позволяет визуализировать и документировать бизнес-процессы, которые должна поддерживать информационная система. Это помогает выявить основные функции и задачи системы, а также определить их взаимосвязи.
* Использование прецедентов: использование прецедентов позволяет описать функциональные требования в виде сценариев взаимодействия пользователя с системой. Каждый прецедент описывает конкретную задачу, которую пользователь должен выполнить с помощью системы.
* Использование языка требований: использование языка требований позволяет формально описать функциональные требования. Язык требований должен быть понятным и недвусмысленным, чтобы его могли понимать как разработчики, так и пользователи системы.

1. При помощи каких диаграмм можно описать функциональные требования к информационной системе?

* Диаграммы вариантов использования: диаграммы вариантов использования описывают функциональность системы с точки зрения пользователя. Они показывают, какие задачи пользователь может выполнять с помощью системы, и какие актёры участвуют в этих задачах.
* Диаграммы потоков данных: диаграммы потоков данных показывают, как данные перемещаются между различными компонентами системы. Они помогают выявить основные функции системы и определить, какие данные необходимы для выполнения этих функций.
* Диаграммы состояний: диаграммы состояний описывают поведение системы в различных состояниях. Они показывают, как система реагирует на различные события и как она переходит из одного состояния в другое.
* Диаграммы классов: диаграммы классов описывают структуру системы и ее основные компоненты. Они показывают, какие классы объектов существуют в системе и как они связаны между собой.
* Диаграммы последовательностей: диаграммы последовательностей описывают порядок выполнения действий в системе. Они показывают, какие объекты участвуют в выполнении действий и в какой последовательности эти действия выполняются.

1. Для чего выполняется ранжирование требований к информационной системе?

Ранжирование требований к информационной системе выполняется для того, чтобы определить, какие требования являются наиболее важными и должны быть реализованы в первую очередь. Это позволяет разработчикам расставить приоритеты и сосредоточить свои усилия на реализации наиболее важных функций системы.

1. Что значит связные требования к информационной системе?

Связные требования к информационной системе означают, что требования согласованы и непротиворечивы друг с другом. Это важно, потому что противоречивые требования могут привести к проблемам при разработке и реализации системы.

Например, если одно требование гласит, что система должна быть доступна 24/7, а другое требование гласит, что система должна быть отключена для технического обслуживания каждую ночь, то эти требования являются противоречивыми. Такая ситуация может привести к проблемам при разработке и реализации системы, поскольку разработчики не смогут удовлетворить оба требования одновременно.

1. Приведите критерии, определяющие качество собранных требований к информационной системе.

* Корректность. Требования должны быть корректными, то есть они должны точно отражать потребности заинтересованных сторон.
* Полнота. Требования должны быть полными, то есть они должны охватывать все аспекты информационной системы, которые важны для заинтересованных сторон.
* Согласованность. Требования должны быть согласованными, то есть они не должны противоречить друг другу.
* Однозначность. Требования должны быть однозначными, то есть они не должны допускать различных толкований.
* Приоритетность. Требования должны быть расставлены по приоритету, то есть должны быть определены наиболее важные требования, которые необходимо реализовать в первую очередь.
* Измеримость. Требования должны быть измеримыми, то есть должна быть возможность проверить, насколько хорошо они реализованы.
* Достижимость. Требования должны быть достижимыми, то есть они должны быть реализуемы с использованием имеющихся ресурсов и технологий.
* Релевантность. Требования должны быть релевантными, то есть они должны соответствовать целям и задачам информационной системы.

# Введение

Целью практической работы является проектирование диаграммы прецедентов информационной системы “Интернет-магазин игровой валюты” в нотации UML.

**1 Общие сведения**

* 1. **Список терминов и определений**

Сервер — часть системы, являющаяся хостом и набором функций для сайта системы.

Клиент — часть системы, отображающая интерфейс сайта системы.

Коэффициент юзабилити — часть пользователей, которым удобно пользоваться сайтом к общему количеству пользователей системы.

Коэффициент интерактивности — часть пользователей, которая пользовалась системой за определенный промежуток времени к ожидаемому количеству пользователей за тот же промежуток времени.

БД (База Данных) — представленная в объективной форме совокупность самостоятельных материалов, систематизированных таким образом, чтобы эти материалы могли быть найдены и обработаны с помощью электронной вычислительной машины (ЭВМ).

* 1. **Описание бизнес-ролей**

Пользователь — авторизованный человек, имеющий расширенный доступ к системе, позволяющая ему просматривать свой профиль и проводить оплату.

Оператор – авторизированный пользователь, имеющий доступ к информации о всех пользователях, служащий для помощи обычным пользователям при использовании системы.

Поставщик – авторизированный пользователь, который имеет собственную страницу на сайте с предоставляемыми услугами.

Администратор — авторизованный пользователь, имеющий полный доступ к системе, позволяющая ему устранять технический ошибки системы.

1. **Проектирование диаграммы прецедентов**

Реализуем диаграмму прецедентов для информационной системы “Интернет магазин игровой валюты” в нотации UML (рис. 1).

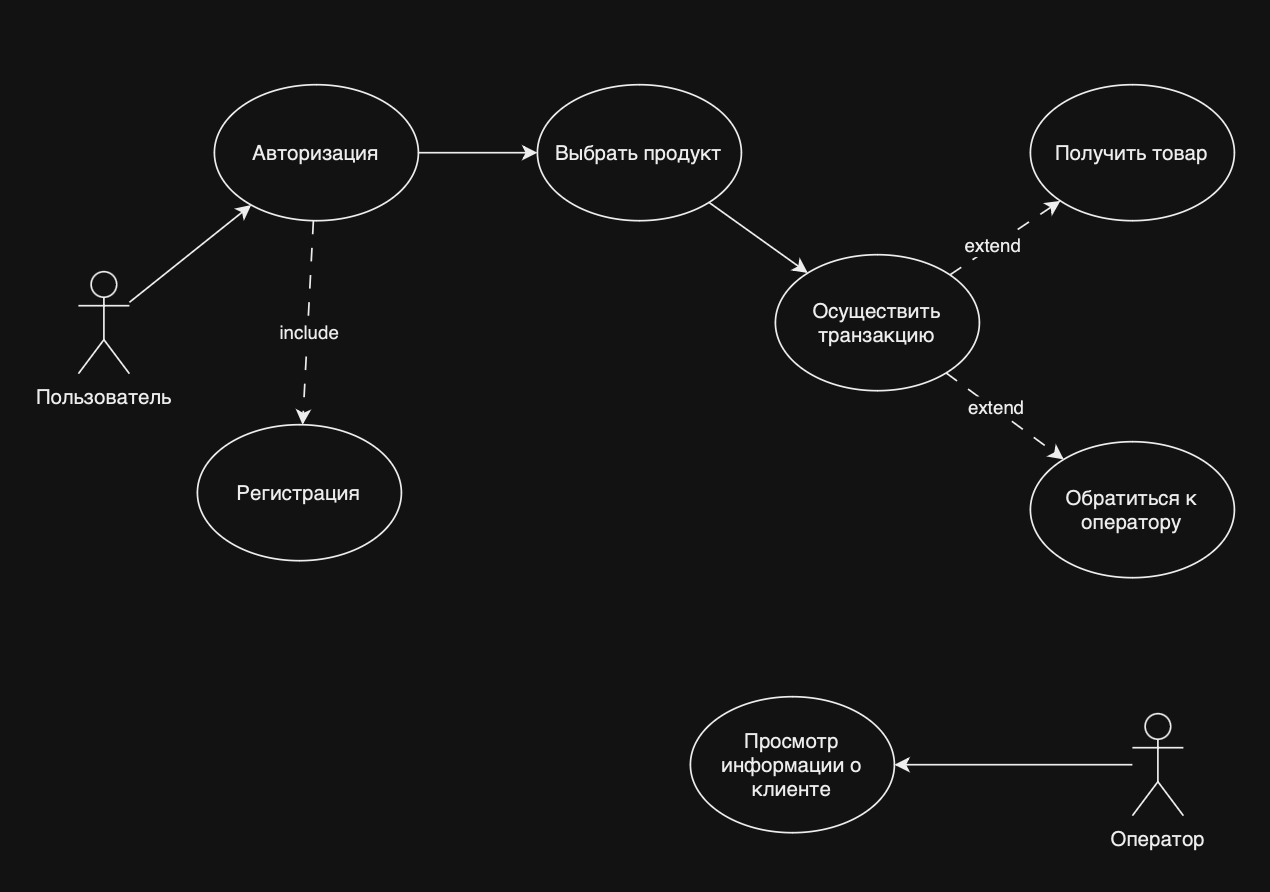


Рисунок 1 – Диаграмма прецедентов системы рекомендаций мест

Действующие субъекты: пользователь, оператор.

Прецеденты: Регистрация, Авторизация, Выбрать продукт, Осуществить транзакцию, Получить товар, Обратиться к оператору, Просмотр информации о клиенте.

**Заключение**

В ходе выполнения работы была спроектирована диаграмма прецедентов информационной системы “Интернет магазин игровой валюты” в нотации UML.

**Ответы на вопросы**

1. Для чего создается диаграмма прецедентов?

Диаграмма прецедентов создается в рамках процесса разработки программного обеспечения для моделирования функциональности системы. Она предназначена для описания взаимодействия между системой и ее окружением, включая пользователей, внешние системы и другие агенты. Диаграмма прецедентов помогает идентифицировать основные функции системы, специфицировать их взаимодействие с актерами и обеспечивает понимание того, как пользователи будут взаимодействовать с системой.

1. Что позволяет определить использование диаграммы прецедентов в процессе проектирования информационной системы?

- Взаимодействие между пользователями (актерами) и системой: Диаграмма прецедентов помогает определить, какие действия могут выполнять пользователи (актеры) в системе и как эти действия взаимодействуют с различными функциями системы.

- Определение функциональных требований: путем анализа диаграммы прецедентов можно выявить основные функции и действия, которые система должна поддерживать. Это помогает определить функциональные требования к системе.

- Определение приоритетов: Диаграмма прецедентов также может помочь в определении приоритетов функций системы на основе их важности для пользователей и бизнеса.

1. Назовите главный элемент диаграммы прецедентов.

Главным элементом диаграммы прецедентов является "прецедент".

1. Дайте определение, что такое «прецедент» или «вариант использования» в рамках диаграммы прецедентов UML.

Прецедент представляет собой определенное действие или функцию, которую система выполняет в ответ на запрос от актера (пользователя или другой системы). Прецеденты описываются как функциональные требования или действия, которые могут быть выполнены в системе. В контексте диаграммы прецедентов UML, прецеденты представляют собой основные функции системы и описывают, как система взаимодействует с внешними актерами.

1. На какой вопрос должна отвечать созданная диаграмма прецедентов?

Диаграмма прецедентов должна отвечать на следующий вопрос: "Какие функции и действия выполняет система и как она взаимодействует с внешними актерами (пользователями или другими системами)?" Другими словами, диаграмма прецедентов помогает идентифицировать основные функции системы, определить, как пользователи будут взаимодействовать с системой и как система будет реагировать на различные запросы от внешних актеров.

1. Для чего необходимы связи на диаграмме прецедентов UML, что они показывают?

Связи на диаграмме прецедентов UML необходимы для показа взаимодействия между различными прецедентами и актерами в системе. Они позволяют определить, как различные прецеденты взаимодействуют между собой и с внешними актерами. Кроме того, связи на диаграмме прецедентов могут показывать наследование между прецедентами, а также включение и расширение функциональности других прецедентов. Таким образом, связи на диаграмме прецедентов UML позволяют определить взаимосвязь между различными функциями и действиями системы, что помогает лучше понять ее функциональность.

1. Назовите два типа связей прецедентов UML.

- Включение (Include): Связь включения показывает, что один прецедент включает в себя функциональность другого прецедента. То есть, если один прецедент должен выполняться в рамках другого, это отображается с помощью связи включения. Это позволяет избежать дублирования функциональности, так как один прецедент может включать в себя выполнение другого.

- Расширение (Extend): Связь расширения показывает, что один прецедент может быть расширен другим прецедентом. Это позволяет учитывать альтернативные сценарии использования, где один прецедент может быть расширен другим при определенных условиях или событиях.

**Введение**

Сегодня видеоигры стали занимать у людей большую часть свободного времени. Игры бывают так и платные так и бесплатные, но в основном транзакции игровых предметов проходят с помощью «внутриигровых валют».

Информационная система "Интернет-магазин игровой валюты" будет предоставлять возможность пользователям приобретать игровую валюту большинства игр с помощью онлайн-платежа.

Разработка информационной системы "Интернет-магазин игровой валюты" позволит сократить временные затраты на использования внутриигровых методов покупки валюты.

Целью практической работы является формирование требований к описанной выше системе. Заданием практической работы является описание объекта автоматизации, формулировка основных задач автоматизации объекта, описание основных параметров проектируемой информационной системы, описание путей достижения целей. Кроме того, необходимо сформулировать требования к информационной системе.

**1 Общие сведения**

* 1. **Полное наименование системы и ее условное обозначение**

Наименование системы: Интернет-магазин игровой валюты.

Условное обозначение: ИМИВ.

* 1. **Номер договора**

Шифр темы: C-NM.

Номер контракта: №1/11-11-11-001 от 09.02.2024.

* 1. **Наименование организаций – Заказчика и Разработчика**

Заказчиком системы является РТУ МИРЭА.

Адрес заказчика: Проспект Вернадского, д. 78

Разработчиком системы является ООО “ГеймингИзи”.

* 1. **Основания для разработки системы**

Работа по созданию системы, обеспечивающей удобство в покупке игровой валюты.

* 1. **Плановые сроки начала и окончания работы по созданию системы**

Плановый срок начала работ по созданию системы – 17 февраля 2024 года.

Плановый срок окончания работ по созданию системы – 30 мая 2024 года

* 1. **Источники и порядок финансирования работ**

Собственные средства разработчика.

* 1. **Порядок оформления и предъявления заказчику результатов работ по созданию системы**

Результаты работ передаются Заказчику в порядке, определенном контрактом в соответствии с Календарным планом работ контракта на основании

Актов сдачи-приемки выполненных работ (этапа работ).

Документация ММ передается на бумажных (два экземпляра, один экземпляр после подписания Заказчиком должен быть возвращен Исполнителю) и на машинных носителях (USB Flash) (в двух экземплярах). Текстовые документы, передаваемые на машинных носителях, должны быть представлены в форматах PDF.

Все материалы передаются с сопроводительными документами Исполнителя.

* 1. **Перечень нормативно-технических документов, методических материалов, использованных при разработке ТЗ**

При разработке автоматизированной системы и создании проектно-эксплуатационной документации Исполнитель должен руководствоваться требованиями следующих нормативных документов:

* ГОСТ 19.106-78. Единая система программной документации. Требования к программным документам, выполненным печатным способом.
* ГОСТ 34.602 – 2020 Техническое задание на создание автоматизированной системы
* ГОСТ Р 59793-2021. Информационные технологии. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания.
* ГОСТ 34.201–2020. Информационные технологии. Комплекс стандартов на автоматизированные системы. Виды, комплектность и обозначение документов при создании автоматизированных систем.
* ГОСТ Р 59795-2021. Информационные технологии. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Требования к содержанию документов
  1. **Список терминов и определений**

1. Сервер — часть системы, являющаяся хостом и набором функций для сайта системы.
2. Клиент — часть системы, отображающая интерфейс сайта системы.
3. Коэффициент юзабилити — часть пользователей, которым удобно пользоваться сайтом к общему количеству пользователей системы.
4. Коэффициент интерактивности — часть пользователей, которая пользовалась системой за определенный промежуток времени к ожидаемому количеству пользователей за тот же промежуток времени.
5. БД (База Данных) — представленная в объективной форме совокупность самостоятельных материалов, систематизированных таким образом, чтобы эти материалы могли быть найдены и обработаны с помощью электронной вычислительной машины (ЭВМ).
6. ИС (Информационная Система) — система, предназначенная для хранения, поиска и обработки информации, и соответствующие организационные ресурсы (человеческие, технические, финансовые и т. д.), которые обеспечивают и распространяют информацию.
7. Python — высокоуровневый язык программирования общего назначения с динамической строгой типизацией и автоматическим управлением памятью, ориентированный на повышение производительности разработчика, читаемости кода и его качества, а также на обеспечение переносимости написанных на нём программ.
   1. **Описание бизнес-ролей**

Пользователь — авторизованный человек, имеющий расширенный доступ к системе, позволяющая ему просматривать свой профиль и проводить оплату.

Оператор – авторизированный пользователь, имеющий доступ к информации о всех пользователях, служащий для помощи обычным пользователям при использовании системы.

Поставщик – авторизированный пользователь, который имеет собственную страницу на сайте с предоставляемыми услугами.

Администратор — авторизованный пользователь, имеющий полный доступ к системе, позволяющая ему устранять технический ошибки системы.

1. **Назначение и цели создания (развития) системы**
   1. **Назначение системы**

Система должна обеспечить удобство в покупке игровой валюты.

1. **Характеристика объекта автоматизации**
   1. **Краткие сведения об объекте автоматизации**

Объектом автоматизации является веб-страница.

* 1. **Сведения об условиях эксплуатации объекта автоматизации**

Условия эксплуатации комплекса технических средств Системы должны соответствовать условиям эксплуатации группы 2 ГОСТ 21552-84 «Средства вычислительной техники. Общие технические требования, приемка, методы испытаний, маркировка, упаковка, транспортировка, хранение».

Условия эксплуатации персональных компьютеров Системы соответствуют Гигиеническим требованиям к видео-дисплейным терминалам, персональным электронно-вычислительным машинам и организации работы (Санитарные правила и нормы. СанПиН 2.2.2.542-96).

Разработчик должен проверить соблюдение условий эксплуатации комплекса технических средств на этапе технического проектирования.

1. **Требования к системе**
   1. **Требования к системе в целом**
      1. **Требования к структуре и функционированию системы**

Система имеет модульную структуру, включающую в себя следующие модули:

* модуль базы данных;
* модуль «Сервер»;
* модуль «Клиент»;

Система должна выполнять следующие функции:

* осуществлять возможность просмотра информации об клиентах;
* осуществлять возможность связи между пользователями;
* осуществлять возможность создания, изменения, удаления товара;
* обработка трафика среднего объема;
* информирование о сбое, ошибках передачи данных;
* осуществлять возможность регистрации, авторизации пользователей, с помощью администратора;
* осуществлять возможность администратору назначения ролей пользователям.
  + 1. **Требования к численности и квалификации персонала системы и режиму его работы**

Для поддержания приложения и эксплуатации интерфейса системы управления персонал должен обладать навыками работы в информационных организациях, а также общими навыками работы с персональным компьютером.

* + 1. **Показатели назначения**

Подсистемы, разработанные и доработанные в рамках данного раздела должны отвечать следующим требованиям:

1. Время на запуск или перезапуск системы и компонентов системы должно составлять не более 45 минут.
2. Коэффициент юзабилити не менее 80%.
3. Коэффициент интерактивности не менее 50%.
4. Коэффициент достоверности информации не менее 100%.
5. В режиме отправки/приемки сообщений, подсистема должна поддерживать интенсивность минимум 200 запросов в секунду при среднем размере пакета 10 Мб. Интенсивность должна быть обеспечена разработанным SDK.

Требования к аппаратной части и масштабированию для обеспечения перечисленных показателей должны быть определены на этапе технического проектирования

* + 1. **Требования к надёжности**

Программное обеспечение не должно выходить из строя более чем на 30 дней.

Для устойчивости к потере данных необходимо регулярно производить выгрузку хранимой информации.

Надежность требуемого уровня достигается путем комплексного применения организационных и организационно-технических мероприятий. При этом необходимо использовать соответствующие требованиям программно-аппаратные средств. В частности, можно использовать следующие базовые подходы:

* системное и базовое ПО и технические средства, соответствующие классу решаемой задачи;
* четкое соблюдение правил эксплуатации, а также регламентных сроков обслуживания используемых программно-аппаратных средств;
* допуск к информационной системе только пользователей, прошедших предварительное обучение.
  + 1. **Требования к безопасности**

Безопасность данных пользователей должна обеспечиваться шифрованием методом SHA-256, а также обеспечением устойчивости программно-технических средств к возможным кибератакам.

* + 1. **Требования к эргономике и технической эстетике**

Взаимодействие пользователей с прикладным программным обеспечением, входящим в состав системы должно осуществляться посредством визуального графического интерфейса (GUI). Интерфейс системы должен быть понятным и удобным, не должен быть перегружен графическими элементами и должен обеспечивать быстрое отображение экранных форм. Более точное описание интерфейса будет представлено в дизайн-документе.

* + 1. **Требования к транспортабельности для подвижных АС**

Требования к транспортабельности не предъявляются.

* + 1. **Требования к эксплуатации, техническому обслуживанию, ремонту и хранению компонентов системы**

Техническим обслуживанием, ремонтом и хранением сервера АС занимаются сетевые инженеры-техники, специалисты по серверным и сетевым технологиям, а также мастера по ремонту компьютерного и другого технического оборудования.

* + 1. **Требования к защите информации от несанкционированного доступа**

При работе с системой необходимо, чтобы данные могли быть восстановлены из резервных копий в случае потери, информация пользователей была защищена от доступа или модификации несанкционированными лицами.

* + 1. **Требования по сохранности информации при авариях**

Серверное программное обеспечение системы должно восстанавливать свое функционирование при перезапуске аппаратных средств. Для обеспечения сохранности данных требуется предусмотреть резервное копирование в энергонезависимые ячейки памяти.

* + 1. **Требования к защите от влияния внешних воздействий**

В случае техногенных или природных аварий серверное программное обеспечение должно продолжать работать на резервных серверах, обеспечивая бесперебойную работу системы.

* + 1. **Требования к патентной чистоте**

Требования к патентной чистоте не предъявляются.

* + 1. **Требования по стандартизации и унификации**

Для реализации статических страниц и шаблонов должен использоваться язык Python. Исходный код должен разрабатываться в соответствии со стандартами PEP8. Для реализации интерактивных элементов клиентской части должен использоваться язык Python. Для реализации динамических страниц должен использоваться язык Python. Приложение должно запускаться на любом устройстве, отвечающем техническим ограничениям, что должно быть реализовано посредством технологий докеризации и оркестрации.

* + 1. **Дополнительные требования**

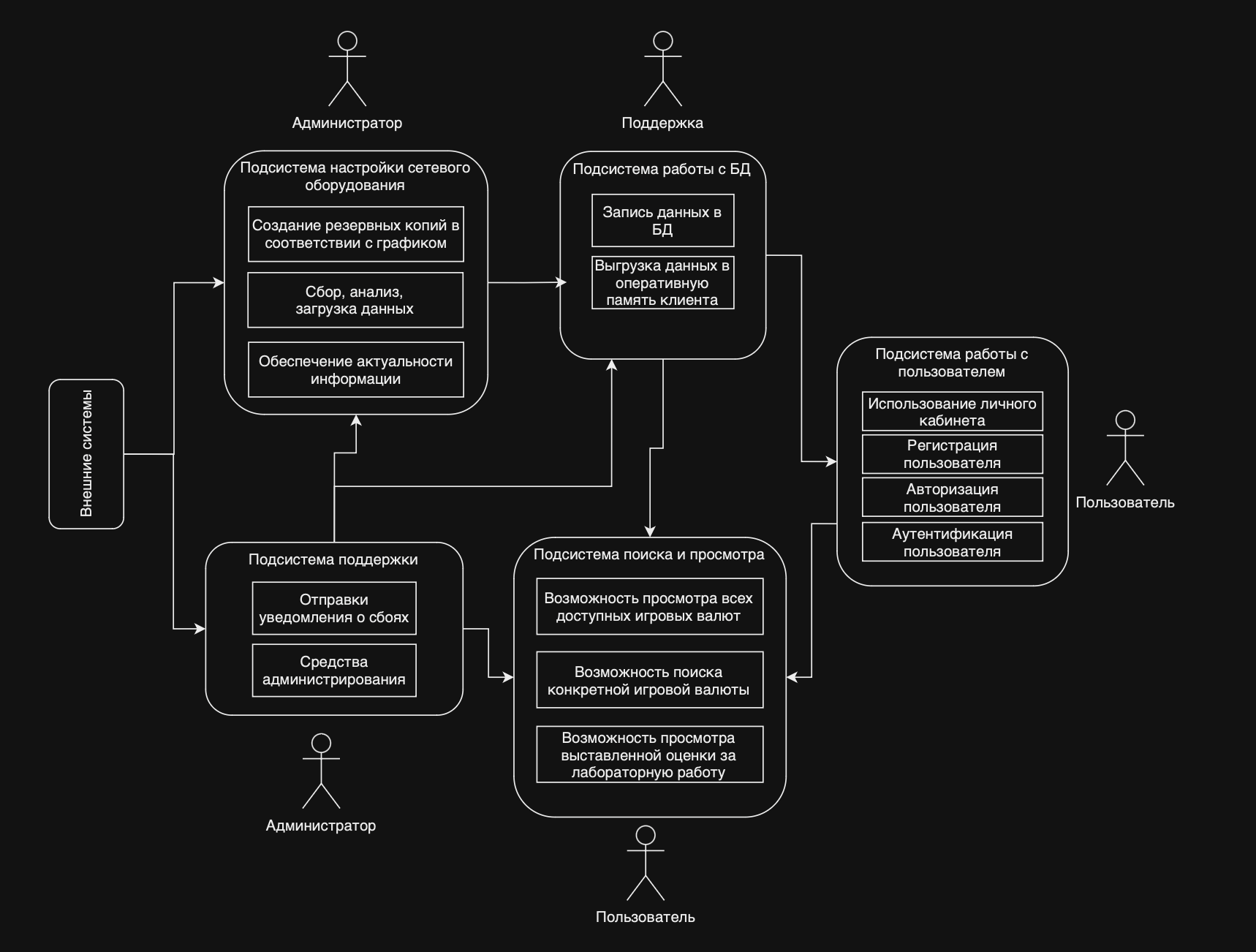
Дополнительные требования не предъявляются.

* 1. **Требования к функциям (задачам), выполняемым системой**

Таблица 2.1 — Требования к функциям, выполняемым системой

|  |  |
| --- | --- |
| **Функция** | **Задача** |
| Обработка трафика среднего объема | Запись данных в БД |
| Обработка данных на клиенте |
| Графическое отображение данных |
| Информирование о сбоях | Отправление уведомлений о сбое администратору |
| Обработка, хранение и поддержка БД | Создание резервных копий в энергонезависимые ячейки памяти |
| Работа с пользователями | Регистрация пользователя |
| Авторизация пользователя |
| Выдача роли пользователю |
| Функция просмотра информации | Возможность просмотра отзывов |
| Возможность просмотра профилей |
| Возможность просмотра личных сообщений |
| Функция изменения информации | Возможность изменения профиля пользователя |
| Возможность изменения данных о товаре |
| Возможность изменения роли пользователя |

* 1. **Функциональная структура системы**

Рисунок 1 - Макет модели информационного обмена между компонентами системы

* 1. **Требования к видам обеспечения**
     1. **Требования к математическому обеспечению системы**

Математическое обеспечение системы должно обеспечивать реализацию перечисленных в данном документе функций, а также выполнение операций конфигурирования, программирования, управления базами данных и документирования. Алгоритмы должны быть разработаны с учетом возможности получения некорректной входной информации и предусматривать соответствующую реакцию на такие события.

* + 1. **Требования к информационному обеспечению системы**

Данные, используемые системой, должны храниться в реляционной СУБД. Структура базы данных определяется с учетом особенностей внутренней модели информационной системы.

Информационный обмен между серверной и клиентской частями системы должен осуществляться по протоколу HTTP.

* + 1. **Требования к лингвистическому обеспечению системы**

Информационная система должна быть реализована на русском, английском и казахском языках.

* + 1. **Требования к программному обеспечению системы**

Программное обеспечение клиентской части должно удовлетворять следующим требованиям:

* Установленный браузер Chrome, Safari, Yandex Browser или Opera;  
  + 1. **Требования к техническому обеспечению системы**

Платформа, на которой будет развернута серверная часть системы, должна удовлетворять следующим минимальным требованиям:

* не менее 16 GB оперативной памяти;
* не менее 512 GB свободного места на жестком диске;
* OC на базе Linux или ОС Windows;
* поддерживаемый протокол передачи данных HTTP / HTTPS, скорость передачи данных 20 Мбит/с;
* процессор с тактовой частотой не менее 4.6 GHz.
  + 1. **Требования к метрологическому обеспечению системы**

Система должна иметь возможность оценивать передачу данных, согласно критериям и нормам оценивания обучающихся по ГОСТ Р МЭК 870—5—2.

* + 1. **Требования к организационному обеспечению системы**

Рабочее место оператора должно быть обустроено согласно ГОСТ Р 50923-96 от 01.01.2008.

* + 1. **Требования к методическому обеспечению системы**

Необходимо разработать несколько типов руководств:

* руководство пользователя для клиентов системы;
* руководство программиста для администраторов, обеспечивающих поддержку системы.

1. **Состав и содержание работ по созданию (развитию) системы**

Разработка системы предполагается по укрупненному календарному плану, приведенному в таблице 5.1.

Таблица 5.1 — Требования к функциям, выполняемым системой

|  |  |  |
| --- | --- | --- |
| **Этапы работ** | **Содержание работ** | **Сроки** |
| 1. Исследование и обоснование создания АС | 1.1. Обследование (сбор и анализ данных) автоматизированного объекта, включая сбор сведений о зарубежных и отечественных аналогах | 17.02.2024- 23.02.2024 |
| 2. Составление технического задания | 2.1. Разработка функциональных и нефункциональных требований к системе | 24.02.2024- 28.02.2024 |
| 3. Эскизное проектирование | 3.1. Разработка предварительных решений по выбранному варианту АС и отдельным видам обеспечения | 01.03.2024- 09.03.2024 |
| 4. Техническое проектирование | 4.1. Разработка диаграмм | 10.03.2024- 17.03.2024 |
| 4.2. Разработка макетов интерфейса | 18.03.2024- 31.03.2024 |
| 5. Разработка программной части | 5.1 Разработка модуля «Авторизация» | 01.04.2024- 5.2. 2024 |
| 5.2 Разработка модуля «Поиск» |
| 5.3 Разработка модуля «Работа с данными» |
| 5.4 Разработка модуля «Уведомления» |
| 5.5 Разработка модуля «Отзывы и рейтинг» |
| 6. Предварительные комплексные испытания | 6.1. Проверка работоспособности системы в условиях, приближенных к реальным | 26.04.2024- 03.05.2024 |
| 7. Опытная эксплуатация | 7.1. Эксплуатация с привлечением небольшого количества участников | 04.05.2024- 10.05.2024 |
| 7.2. Устранение замечаний, выявленных при эксплуатации, АС | 11.05.2024- 15.05.2024 |
| 8. Ввод в промышленную эксплуатацию | 8.1. Приемка АС в промышленную эксплуатацию (внедрение АС) | 16.05.2024- 30.05.2024 |

1. **Порядок контроля и приемки системы**

В соответствии с разделом 5 необходимо на каждой стадии создания системы установить контроль и приемку результатов работ.

На стадии 5 происходит прием готовой версии программного продукта (модели), а остальные результаты работ представляются в виде документов согласно таблице 5.1.

Приемка этапа включает в себя рассмотрение и оценку объема работ и предоставленной технической документации в соответствии с требованиями технического задания.

Организацию и проведение приемки системы должен осуществлять заказчик, а приемка системы должна производиться только после того, как будут выполнены все задачи системы.

Заказчик обязан предоставить материальную часть (технические средства), проектную документацию и специально выделенный персонал.

Последним этапом при приемке системы является составление акта приемки.

1. **Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие**

Для обеспечения готовности объекта к вводу системы в действие провести комплекс мероприятий:

* приобрести компоненты программного обеспечения, заключить договора на их лицензионное использование;
* завершить работы по установке технических средств;
* провести диагностику устойчивости сети к нагрузкам;
* провести обучение сотрудников.
  1. **Приведение поступающей в систему информации к виду, пригодному для обработки с помощью ЭВМ**

Информация вводится пользователем в разработанные экранные формы компонентов системы.

* 1. **Изменения, которые необходимо осуществить в объекте автоматизации**

Изменений не требуется.

* 1. **Создание условий функционирования объекта автоматизации, при которых гарантируется соответствие создаваемой системы требованиям, содержащимся в ТЗ**

Для функционирования создаваемой системы требуется платформа, технические характеристики которой соответствуют предъявленным.

* 1. **Создание необходимых для функционирования системы подразделений и служб**

Для функционирования системы не требуется дополнительных подразделений и служб.

* 1. **Сроки и порядок комплектования штатов и обучения персонала**

Комплектование штатов подразделений и служб, необходимых для функционирования системы, а также подготовка их сотрудников должны быть завершены до начала опытной эксплуатации системы.

1. **Требования к документированию**

Проектная документация должна быть разработана в соответствии с ГОСТ 34.201-2020 и ГОСТ 7.32-2017.

Отчетные материалы должны включать в себя текстовые материалы (представленные в виде бумажной копии и на цифровом носителе в формате MS Word) и графические материалы. Предоставить документы:

1. схема функциональной структуры автоматизируемой деятельности;
2. описание технологического процесса обработки данных;
3. описание информационного обеспечения;
4. описание программного обеспечения АС;
5. схема логической структуры БД;
6. руководство пользователя;
7. описание контрольного примера (по ГОСТ 24.102);
8. протокол испытаний (по ГОСТ 24.102).
9. **Источники разработки**

* ГОСТ 34.602-2020. Информационные технологии. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы.
* ГОСТ Р 59793-2021. Информационные технологии. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания.
* ГОСТ 34.201-2020. Информационные технологии. Комплекс стандартов на автоматизированные системы. Виды, комплектность и обозначение документов при создании автоматизированных систем.
* ГОСТ Р 59795-2021. Информационные технологии. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Требования к содержанию документов.
* ГОСТ 19.106-78. Единая система программной документации. Требования к программным документам, выполненным печатным способом.
* ГОСТ 19.105-78. Единая система программной документации. Общие требования к программным документам.

**Ответы на вопросы**

1. Что такое требование к информационной системе?

Требование к информационной системе — это формализованное описание функциональных, производственных, технических, процессуальных или бизнес-ориентированных условий, которые должны быть удовлетворены или реализованы в информационной системе. Требования могут включать в себя описания функциональности системы, производительности, надежности, безопасности, совместимости, а также другие аспекты, которые определяют желаемое поведение или характеристики информационной системы.

1. Какие нормативные документы регламентируют сбор и управление требованиями к информационной системе?

* ISO/IEC/IEEE 29148:2018 - Стандарт о системной и программной инженерии, который содержит рекомендации по управлению требованиями.
* ISO/IEC/IEEE 15288:2015 - Еще один стандарт по системной и программной инженерии, который также охватывает процессы управления требованиями.
* BABOK (A Guide to the Business Analysis Body of Knowledge) - Этот документ предоставляет руководство по анализу бизнес-требований, включая сбор и управление требованиями.
* PMBOK (A Guide to the Project Management Body of Knowledge) - Стандарт по управлению проектами, который включает в себя процессы управления требованиями в контексте проектов.

1. Что является важным при формировании нефункциональных требований к информационной системе?

* Конкретность: Нефункциональные требования должны быть конкретизированы и измеримы, чтобы обеспечить их понимание и проверяемость.
* Консистентность: Нефункциональные требования должны быть согласованы друг с другом и с функциональными требованиями, чтобы избежать противоречий и недопонимания.
* Релевантность: Требования должны быть релевантными для области применения системы и соответствовать бизнес-потребностям и ожиданиям пользователей.
* Измеримость: Нефункциональные требования должны быть измеримыми, чтобы обеспечить возможность оценки и проверки их выполнения.
* Управление рисками: Нефункциональные требования могут включать в себя аспекты безопасности, производительности, надежности и другие параметры, связанные с управлением рисками, которые должны быть учтены.
* Прозрачность: Нефункциональные требования должны быть понятными и прозрачными для всех заинтересованных сторон, чтобы обеспечить их правильное восприятие и интерпретацию.

1. Какие средства существуют для формирования функциональных требований к информационной системе?

* Интервью и обсуждения с заинтересованными сторонами: Проведение интервью и обсуждение с заказчиками, конечными пользователями и другими заинтересованными сторонами помогает понять их потребности и желания, что может стать основой для формирования функциональных требований.
* Структурированные методы сбора информации: Методы, такие как анализ документации, наблюдение за рабочим процессом, анкетирование и т.д. могут помочь в сборе информации о функциональных требованиях.
* Прототипирование и моделирование: Создание прототипов и моделей информационной системы позволяет лучше понять функциональность, которая требуется от системы, и помогает в формировании соответствующих требований.
* Использование стандартных нотаций и языков моделирования: Нотации, такие как UML (Unified Modeling Language) и BPMN (Business Process Model and Notation), могут быть использованы для визуализации и документирования функциональных требований.
* Работа с прецедентами и кейсами использования: Анализ типичных сценариев использования системы и создание прецедентов помогает выявить функциональные требования, которые могут быть документированы.

1. При помощи каких диаграмм можно описать функциональные требования к информационной системе?

Кроме этого, функциональные требования можно объединять в use case или, по-русски, варианты использования, сценарии использования или прецеденты использования. Сценарий использования – это некая абстракция, которая включает диаграммы, составленные по определенным правилам, и текстовые документы, поясняющие диаграммы. Сценарий использования позволяет описать поведение системы при взаимодействии с объектами внешней среды. В нотации UML 2.4.12 прецеденты являются средством для определения требуемых использований системы

1. Для чего выполняется ранжирование требований к информационной системе?

Для расставления приоритетов их реализации в первой и последующих версиях ИС.

1. Что значит связные требования к информационной системе?

В процессе классификации требований все собранные на предыдущем шаге требования трансформируются в группы, связанные логически и технологически.

1. Приведите критерии, определяющие качество собранных требований к информационной системе.

Полнота, точность и четкость формулирования требований определяют эффективность работы команды проекта и успех всего проекта, в том числе соблюдение сроков внедрения информационной системы и стоимость разработки, и ее качество

1. Приведите достоинства и недостатки итеративного макетирования.

Достоинства итеративного макетирования:

* Позволяет быстро получить обратную связь: благодаря созданию прототипов и макетов рано в процессе разработки, проектные команды могут быстро получить обратную связь от заказчиков и конечных пользователей, что позволяет быстро вносить изменения и уточнять требования.
* Улучшает понимание требований: Создание макетов помогает заказчикам и разработчикам лучше понять требования и ожидания в отношении функциональности системы.
* Снижает риск: Итеративный подход позволяет выявить проблемы и риски на ранних этапах, что позволяет быстро реагировать и вносить изменения, тем самым снижая риск для проекта.
* Повышает удовлетворенность заказчика: Раннее представление прототипов и макетов позволяет заказчикам лучше представить себе будущую систему и вносить коррективы до завершения проекта.

Недостатки итеративного макетирования:

* Дополнительные затраты времени и ресурсов: Создание итеративных макетов может потребовать дополнительного времени и ресурсов, особенно в начальной фазе проекта.
* Риск недопонимания: несмотря на возможность улучшить понимание требований, существует риск недопонимания или неправильной интерпретации требований при создании макетов.
* Необходимость управления изменениями: поскольку итеративный подход подразумевает частые изменения, требуется эффективное управление изменениями и контроль версий макетов.
* Неэффективность для некоторых типов проектов: Для некоторых типов проектов, особенно технически сложных или с жесткими требованиями, итеративное макетирование может оказаться неэффективным.

1. Опишите процесс создания и оформления пояснительной записки эскизного проекта информационной системы.

* Введение: В начале пояснительной записки следует предоставить краткое введение, в котором описывается цель и область применения информационной системы. Здесь также можно включить обзор существующих проблем и потребностей, которые предполагается решить с помощью новой системы.
* Описание бизнес-требований: В этом разделе следует описать бизнес-требования к информационной системе. Это может включать в себя описание основных функций и возможностей системы, требования к производительности, безопасности, управлению данными и другие аспекты, которые важны для бизнеса.
* Техническое описание: здесь следует предоставить техническое описание предполагаемой архитектуры системы, используемых технологий, интеграционных аспектов, требований к инфраструктуре и т.д.
* Описание пользовательского интерфейса: если информационная система имеет пользовательский интерфейс, в пояснительной записке следует предоставить описание ключевых элементов пользовательского интерфейса, принципов взаимодействия с системой и т.д.
* Описание процессов и бизнес-логики: В этом разделе следует описать ключевые бизнес-процессы, которые поддерживаются системой, бизнес-правила, алгоритмы и другие аспекты бизнес-логики.
* Оценка рисков: здесь следует провести оценку рисков, связанных с проектом, и предложить возможные стратегии управления рисками.
* Бюджет и график: В пояснительной записке также рекомендуется предоставить оценку бюджета и графика проекта, а также описать план действий для реализации проекта.
* Заключение: В заключительной части пояснительной записки можно подвести итоги, подчеркнуть ключевые моменты и выделить основные рекомендации.

1. Какие нормативные документы регламентируют процесс создания и оформления эскизного проекта?

* ISO/IEC/IEEE 12207:2017 - Стандарт о жизненном цикле процессов разработки и сопровождения программных систем, который охватывает процессы начального проектирования, включая создание эскизного проекта.
* Отраслевые стандарты - В различных отраслях могут существовать свои стандарты и руководства по созданию и оформлению эскизных проектов для информационных систем, включая стандарты, разработанные организациями или ассоциациями в данной области.
* Внутренние стандарты организации - Многие организации имеют свои внутренние стандарты, которые регламентируют процессы создания и оформления документации, включая эскизные проекты.
* Методологии разработки - Методологии разработки, такие как Agile, Waterfall, и другие, могут также включать стандарты и рекомендации по созданию и оформлению эскизных проектов.

1. Какие действия должны быть выполнены, чтобы пояснительная записка эскизного проекта информационной системы стала программным документом?

1. Оформить пояснительную записку в соответствии с требованиями к программным документам. Это включает в себя детальное описание целей и задач проекта, функциональные требования к информационной системе, архитектурные решения, технологии, используемые в проекте, и т.д.

2. Преобразовать пояснительную записку в формат программного документа, который обычно включает в себя структурированные разделы, описание алгоритмов и структур данных, технические спецификации и другую необходимую информацию.

3. Проверить документ на наличие ошибок и неточностей, а также на соответствие требованиям и ожиданиям заказчика.

4. Утвердить программный документ и передать его в разработку информационной системы.

5. В процессе разработки системы следует регулярно обновлять программный документ и вносить любые изменения, которые могут возникнуть в процессе работы над проектом.

1. Какие нотации можно использовать для описания в графическом виде функциональной модели информационной системы?

1. Диаграммы активностей (Activity Diagrams) - для описания последовательности действий и процессов в системе.

2. Диаграммы случаев использования (Use Case Diagrams) - для описания взаимодействия пользователей и системы, а также функциональных требований.

3. Диаграммы последовательностей (Sequence Diagrams) - для визуализации взаимодействия объектов и компонентов системы.

4. Диаграммы классов (Class Diagrams) - для описания структуры системы и ее компонентов.

5. Диаграммы коммуникации (Communication Diagrams) - для описания взаимодействия объектов в системе и передачи сообщений.

**Введение**

Сегодня видеоигры стали занимать у людей большую часть свободного времени. Игры бывают так и платные так и бесплатные, но в основном транзакции игровых предметов проходят с помощью «внутриигровых валют».

Информационная система "Интернет-магазин игровой валюты" будет предоставлять возможность пользователям приобретать игровую валюту большинства игр с помощью онлайн-платежа.

**1 Общие сведения**

* 1. **Список терминов и определений**

Сервер — часть системы, являющаяся хостом и набором функций для сайта системы.

Клиент — часть системы, отображающая интерфейс сайта системы.

Коэффициент юзабилити — часть пользователей, которым удобно пользоваться сайтом к общему количеству пользователей системы.

Коэффициент интерактивности — часть пользователей, которая пользовалась системой за определенный промежуток времени к ожидаемому количеству пользователей за тот же промежуток времени.

БД (База Данных) — представленная в объективной форме совокупность самостоятельных материалов, систематизированных таким образом, чтобы эти материалы могли быть найдены и обработаны с помощью электронной вычислительной машины (ЭВМ).

ИС (Информационная Система) — система, предназначенная для хранения, поиска и обработки информации, и соответствующие организационные ресурсы (человеческие, технические, финансовые и т. д.), которые обеспечивают и распространяют информацию.

Python — высокоуровневый язык программирования общего назначения с динамической строгой типизацией и автоматическим управлением памятью, ориентированный на повышение производительности разработчика, читаемости кода и его качества, а также на обеспечение переносимости написанных на нём программ.

PEP8 — документ, описывающий соглашение о том, как писать код на языке Python.

Система контейнеризации — это технология абстракции, которая позволяет упаковывать и исполнять приложения вместе со всеми их зависимостями в изолированных средах, называемых контейнерами.

Система оркестрации — система автоматического размещения, координации и управления сложными компьютерными системами и службами.

* 1. **Описание бизнес-ролей**

Пользователь — авторизованный человек, имеющий расширенный доступ к системе, позволяющая ему просматривать свой профиль и проводить оплату.

Оператор – авторизированный пользователь, имеющий доступ к информации о всех пользователях, служащий для помощи обычным пользователям при использовании системы.

Поставщик – авторизированный пользователь, который имеет собственную страницу на сайте с предоставляемыми услугами.

Администратор — авторизованный пользователь, имеющий полный доступ к системе, позволяющая ему устранять технический ошибки системы.

1. **Проектирование контекстной диаграммы** 
   1. **Цель создания ИС**

Целю создания ИС «Интернет-магазин игровой валюты» является стремление упростить процесс покупки игровой валюты.

**Краткое описание**

ИС «Интернет-магазин игровой валюты» представлена в виде серверной части.

Одно из важных достоинств проектируемой ИС – большой функционал для зарегистрированных пользователей. Авторизованные пользователи могут выкладывать писать комментарии и взаимодействовать с поставщиками. Это даст дополнительный интерес пользователям к регистрации на платформе, которая позволит им получить полный доступ к функциям портала.

* 1. **Способ создания ИС**

В качестве способа определения требований была выбрана методология «последовательных приближений», которая основана на том, что все расчеты и графические построения, связанные с определением основных элементов, разбиваются на несколько более мелких элементов, в которых происходит их уточнение. Данный метод также хорошо сочетается с нотацией IDEF0, которая основана на декомпозиции каждого блока на более мелких с уточнением деталей.

* 1. **Средства создания ИС**

В качестве средств создания ИС были использованы языки программирования Python, СУБД PostgreSQL и сервис для развёртывания сервера Ubuntu Server. Для моделирования проектируемой ИС будет использоваться нотация IDEF0 в CASE-средстве Ramus Educational.

* 1. **Проектирование контекстной диаграммы функциональной модели ИС**

Была спроектирована контекстная диаграмма A-0 в нотации IDEF0.

В качестве управления было выбрано:

1. ГОСТ Р 57489-2017;
2. Политика компании.

Входящие информационные потоки:

1. Выбранная валюта из каталога;
2. Способ оплаты;
3. Данные аккаунты получателя.

Механизмы:

1. Авторизованный пользователь;
2. Поставщик;
3. Сервер.

Выходящие информационные потоки:

1. Результат передачи данных;

На рисунке 1 представлена контекстная диаграмма информационной системы “Интернет-магазин игровой валюты” в нотации IDEF0.

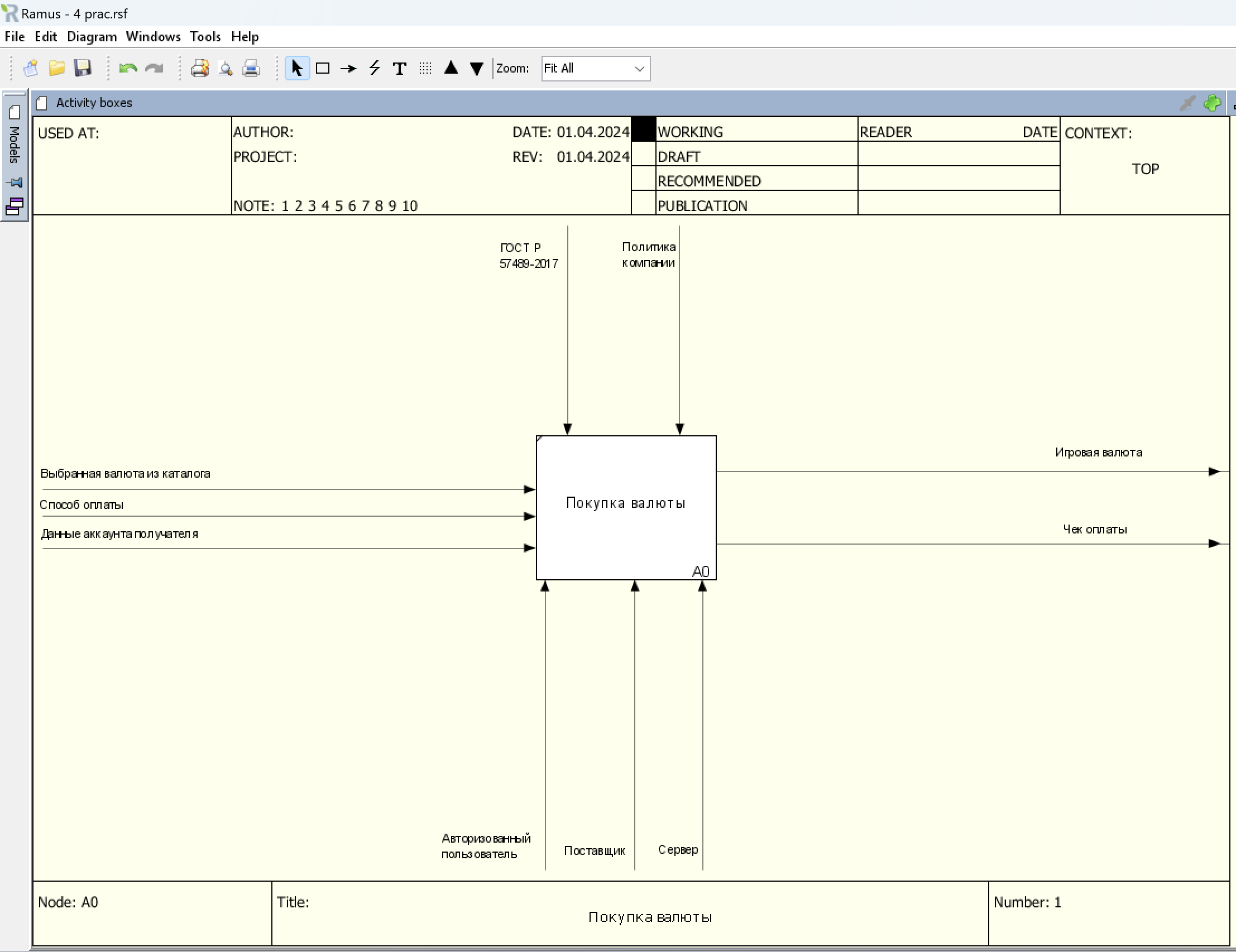


Рисунок 1 – Контекстная диаграмма

**Заключение**

В ходе выполнения работы было реализовано функциональное проектирование модели информационной системы “Интернет-магазин игровой валюты” с использованием методологии SADT в нотации IDEF0, составление краткого описания ИС, включая цель, способ и средства её создания.

**Введение**

Сегодня видеоигры стали занимать у людей большую часть свободного времени. Игры бывают так и платные так и бесплатные, но в основном транзакции игровых предметов проходят с помощью «внутриигровых валют».

Информационная система "Интернет-магазин игровой валюты" будет предоставлять возможность пользователям приобретать игровую валюту большинства игр с помощью онлайн-платежа.

**1 Общие сведения**

* 1. **Список терминов и определений**

Сервер — часть системы, являющаяся хостом и набором функций для сайта системы.

Клиент — часть системы, отображающая интерфейс сайта системы.

Коэффициент юзабилити — часть пользователей, которым удобно пользоваться сайтом к общему количеству пользователей системы.

Коэффициент интерактивности — часть пользователей, которая пользовалась системой за определенный промежуток времени к ожидаемому количеству пользователей за тот же промежуток времени.

БД (База Данных) — представленная в объективной форме совокупность самостоятельных материалов, систематизированных таким образом, чтобы эти материалы могли быть найдены и обработаны с помощью электронной вычислительной машины (ЭВМ).

ИС (Информационная Система) — система, предназначенная для хранения, поиска и обработки информации, и соответствующие организационные ресурсы (человеческие, технические, финансовые и т. д.), которые обеспечивают и распространяют информацию.

Python — высокоуровневый язык программирования общего назначения с динамической строгой типизацией и автоматическим управлением памятью, ориентированный на повышение производительности разработчика, читаемости кода и его качества, а также на обеспечение переносимости написанных на нём программ.

PEP8 — документ, описывающий соглашение о том, как писать код на языке Python.

Система контейнеризации — это технология абстракции, которая позволяет упаковывать и исполнять приложения вместе со всеми их зависимостями в изолированных средах, называемых контейнерами.

Система оркестрации — система автоматического размещения, координации и управления сложными компьютерными системами и службами.

* 1. **Описание бизнес-ролей**

Пользователь — авторизованный человек, имеющий расширенный доступ к системе, позволяющая ему просматривать свой профиль и проводить оплату.

Оператор – авторизированный пользователь, имеющий доступ к информации о всех пользователях, служащий для помощи обычным пользователям при использовании системы.

Поставщик – авторизированный пользователь, который имеет собственную страницу на сайте с предоставляемыми услугами.

Администратор — авторизованный пользователь, имеющий полный доступ к системе, позволяющая ему устранять технический ошибки системы.

1. **Проектирование декомпозиции диаграмм**

Реализуем декомпозицию контекстной диаграммы на 2 уровне для информационной системы “Интернет-магазин игровой валюты” в нотации IDEF0 (рис. 1).

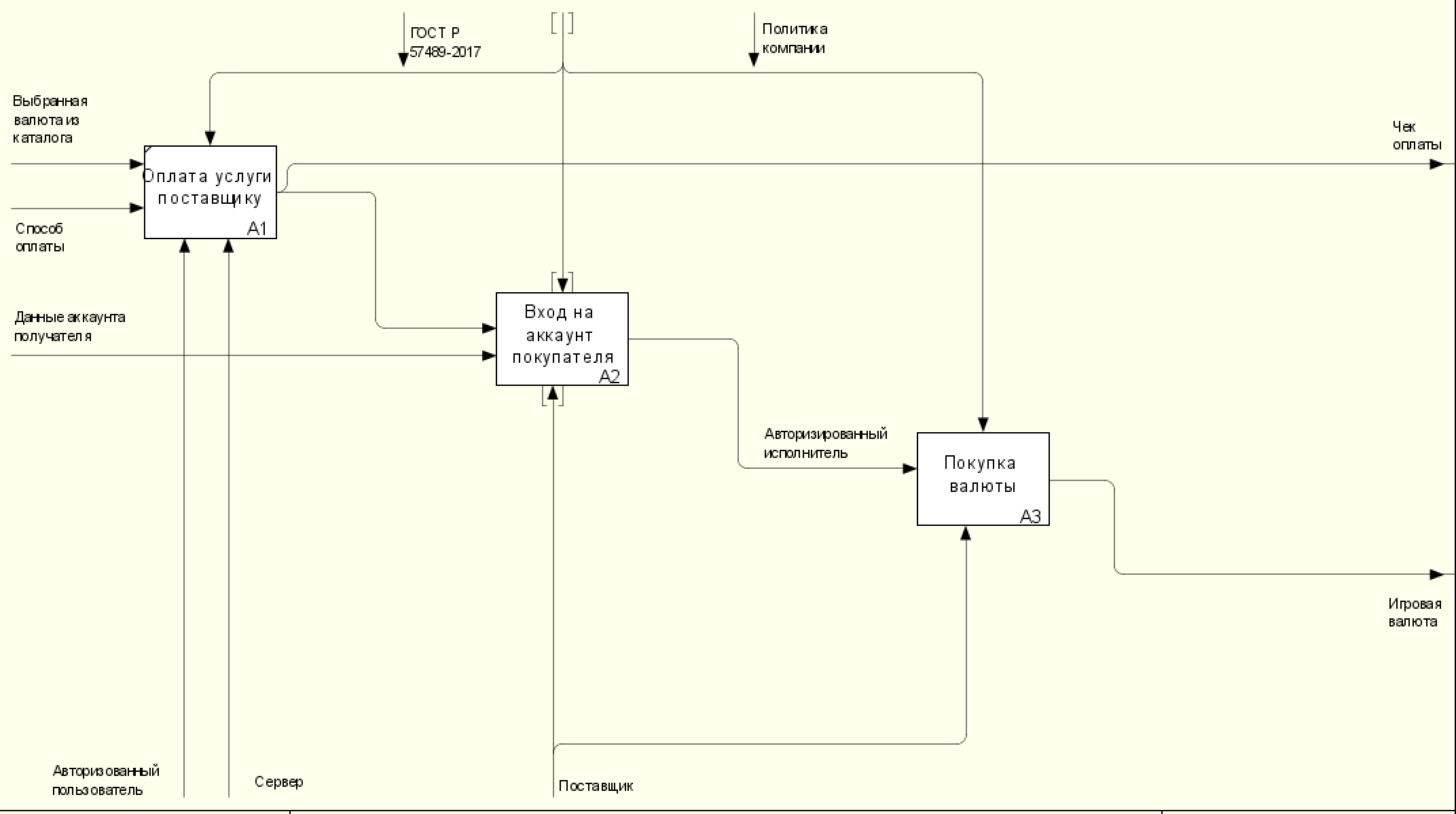


Рисунок 1 – Декомпозиция контекстной диаграммы ИС

Функциональные блоки:

* Оплата услуги поставщику;
* Вход на аккаунт покупателя;
* Покупка валюты.

Потоки данных:

* Выбранная валюта из каталога
* Способ оплаты
* Данные аккаунта получателя

Реализуем декомпозицию диаграммы “Вход на аккаунт покупателя” на 3 уровне для информационной системы “Интернет-магазин игровой валюты” в нотации IDEF0 (рис. 2).

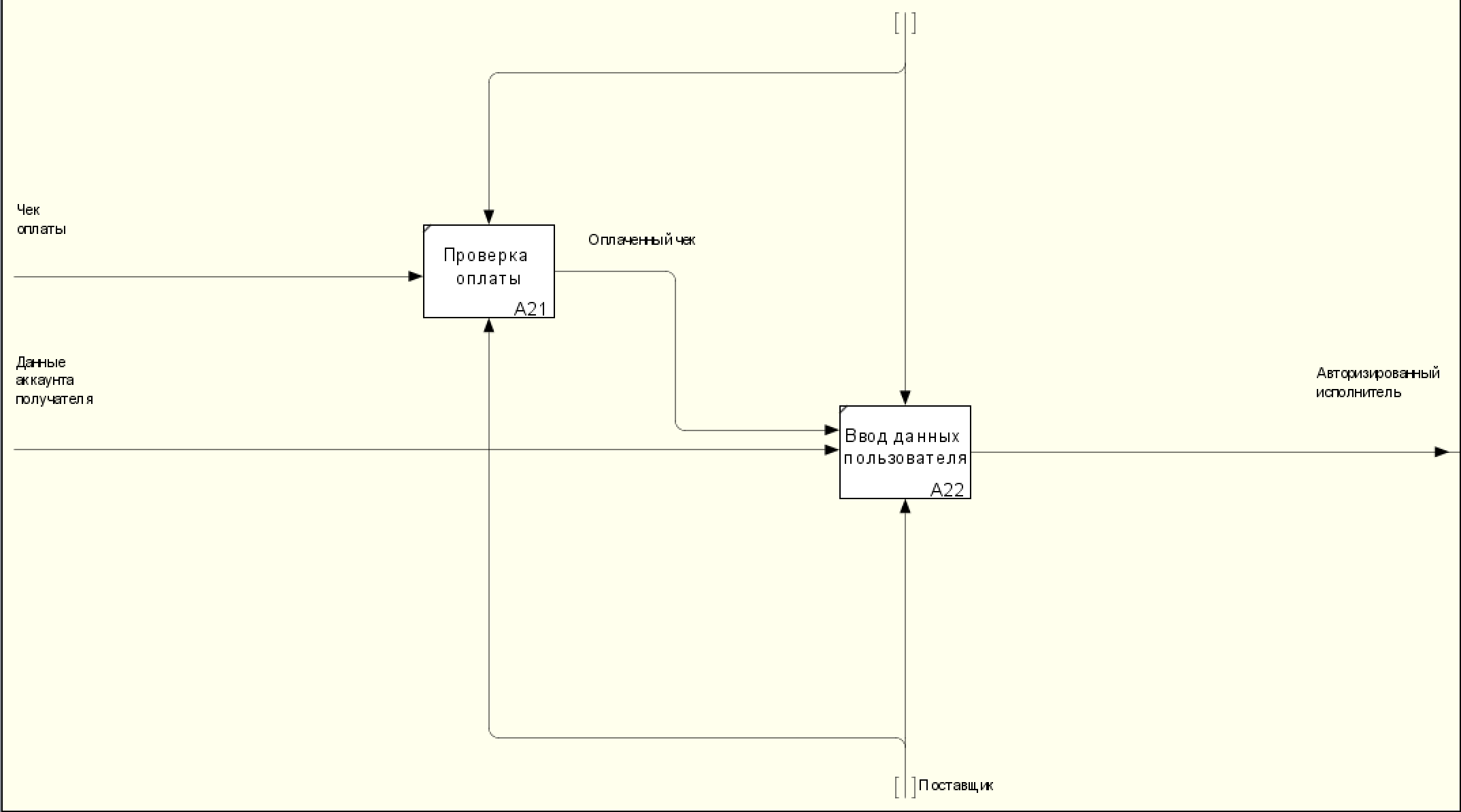


Рисунок 2 – Декомпозиция диаграммы “Вход на аккаунт покупателя”ИС

Функциональные блоки:

* Проверка оплаты;
* Ввод данных пользователя.

Потоки данных:

* Чек оплаты;
* Данные аккаунта покупателя.

**Заключение**

В ходе выполнения работы были реализованы декомпозиции контекстных и функциональных диаграмм информационной системы “Проектирование сети” в нотации IDEF0.

**Введение**

Сегодня видеоигры стали занимать у людей большую часть свободного времени. Игры бывают так и платные так и бесплатные, но в основном транзакции игровых предметов проходят с помощью «внутриигровых валют».

Информационная система "Интернет-магазин игровой валюты" будет предоставлять возможность пользователям приобретать игровую валюту большинства игр с помощью онлайн-платежа.

**1 Общие сведения**

* 1. **Список терминов и определений**

Сервер — часть системы, являющаяся хостом и набором функций для сайта системы.

Клиент — часть системы, отображающая интерфейс сайта системы.

Коэффициент юзабилити — часть пользователей, которым удобно пользоваться сайтом к общему количеству пользователей системы.

Коэффициент интерактивности — часть пользователей, которая пользовалась системой за определенный промежуток времени к ожидаемому количеству пользователей за тот же промежуток времени.

БД (База Данных) — представленная в объективной форме совокупность самостоятельных материалов, систематизированных таким образом, чтобы эти материалы могли быть найдены и обработаны с помощью электронной вычислительной машины (ЭВМ).

* 1. **Описание бизнес-ролей**

Пользователь — авторизованный человек, имеющий расширенный доступ к системе, позволяющая ему просматривать свой профиль и проводить оплату.

Модератор – авторизированный пользователь, имеющий доступ к информации о всех пользователях, служащий для помощи обычным пользователям при использовании системы.

Поставщик – авторизированный пользователь, который имеет собственную страницу на сайте с предоставляемыми услугами.

Администратор — авторизованный пользователь, имеющий полный доступ к системе, позволяющая ему устранять технический ошибки системы.

1. **Проектирование DFD диаграммы**

Реализуем декомпозицию контекстной диаграммы для информационной системы “Интернет-магазин игровой валюты” в нотации DFD (рис. 1).

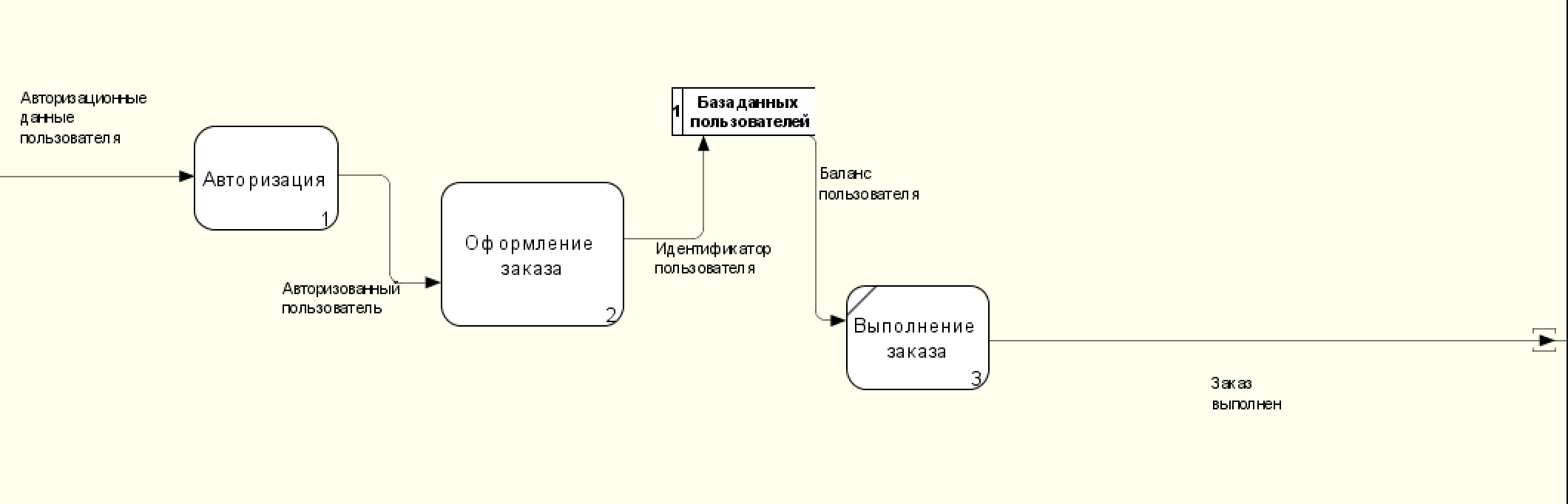


Рисунок 1 – Декомпозиция контекстной диаграммы ИС

Функциональные блоки:

* Авторизация;
* Оформление заказа;
* Выполнение заказа.

Хранилище данных:

* База данных пользователей;

Внешняя сущность:

* Модератор.

Потоки данных:

* Авторизационные данные пользователя;
* Авторизированный пользователь;
* Идентификатор пользователя;
* Баланс пользователя;
* Заказ выполнен;

Реализуем декомпозицию диаграммы “Авторизация” для информационной системы “Интернет-магазин игровой валюты” в нотации DFD (рис. 2).

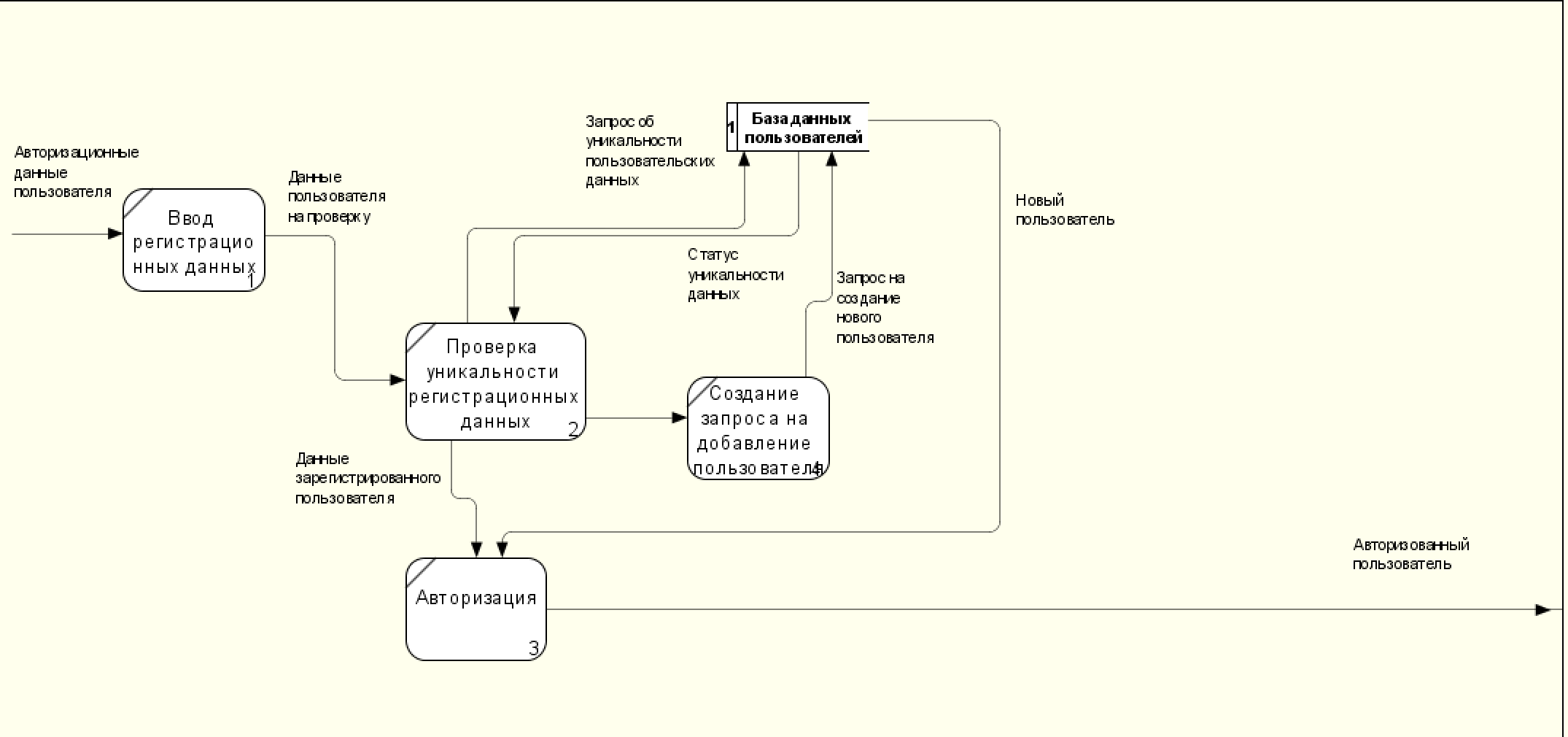


Рисунок 2 – Декомпозиция диаграммы “Авторизация”ИС

Функциональные блоки:

* Ввод регистрационных данных;
* Авторизация
* Проверка уникальности регистрационных данных
* Создание запроса на добавление пользователя.

Внешняя сущность:

* База данных пользователей.

Потоки данных:

* Авторизационные данные пользователя;
* Данные пользователя на проверку;
* Данные зарегистрированного пользователя;
* Запрос об уникальности пользовательских данных;
* Статус уникальности данных;
* Запрос на создание нового пользователя;
* Новый пользователь;
* Авторизированный пользователь.

**Заключение**

В ходе выполнения работы были реализованы декомпозиции контекстных и функциональных диаграмм информационной системы “Интернет-магазин игровой валюты” в нотации DFD.

**Ответы на вопросы**

1. Что такое функциональная модель информационной системы?

Функциональная модель информационной системы представляет собой концептуальное представление о функциях, процессах и операциях, которые выполняются в информационной системе. Она описывает, как различные компоненты системы взаимодействуют для выполнения определенных задач и обеспечения определенной функциональности. В функциональной модели могут быть представлены основные бизнес-процессы, операции, взаимодействие пользователей с системой, а также потоки данных и информации в системе.

2. Какие нотации можно использовать для создания функциональной модели информационной системы?

* Диаграммы потоков данных (DFD): Это одна из наиболее распространенных нотаций, используемых для моделирования функциональной модели. DFD позволяют описать потоки данных в информационной системе и процессы их обработки.
* Диаграммы прецедентов: Эта нотация широко используется в методологии Unified Modeling Language (UML) и описывает функциональность системы через взаимодействие акторов (пользователей) и прецедентов (действий или событий).
* Блок-схемы: Это классический метод моделирования, который позволяет представить функциональность системы в виде блоков, связанных стрелками, чтобы показать поток выполнения операций.
* IDEF0 (Integration Definition for Function Modeling): Это стандартная нотация, используемая для анализа и моделирования функциональных аспектов системы, позволяющая описать функциональные обязанности, связи и потоки данных.

3. Как строится функциональная модель информационной системы в нотации IDEF0?

* Идентификация функций: определяются основные функции (процессы, операции), которые выполняет информационная система.
* Построение блоков: Функции представляются в виде блоков (ящиков) на диаграмме. Каждый блок описывает определенную функцию системы.
* Определение потоков данных: Стрелки между блоками показывают потоки данных и управления между функциями.
* Аннотации: Блоки и стрелки подписываются для ясного определения функций и потоков данных.
* Декомпозиция: Функции могут быть более подробно декомпозированы на более низкие уровни, что позволяет уточнить процессы и операции.
* Документация: Каждая функция описывается детально, включая входы, выходы и условия выполнения.

4. Что является самым важным при создании диаграммы IDEF0 уровня А0?

При создании диаграммы IDEF0 уровня A0 самым важным является правильное определение основных функций и их взаимосвязей. Важно четко и точно определить функции системы, их входы и выходы, а также потоки данных и управления между ними. Кроме того, необходимо обеспечить грамотное документирование каждой функции, включая ее описание, входы, выходы и условия выполнения.

5. Какие нормативные документы регламентируют создание диаграмм IDEF0?

* Federal Information Processing Standards (FIPS) PUB 183: Этот стандарт утвержден Национальным институтом стандартов и технологий (NIST) и определяет методику IDEF0 для моделирования функций, процессов и потоков данных в системах.
* MIL-STD-499: Этот военный стандарт устанавливает требования для системной инженерии в области военной промышленности, включая методологию IDEF0.
* ISO/IEC 14019: Этот международный стандарт устанавливает рекомендации для применения методики IDEF0.

6. Приведите основные требования к названию функционального блока диаграммы IDEF0 уровня А-0.

* Описание функции: Название функционального блока должно четко и точно отражать выполняемую функцию или процесс.
* Краткость: Название должно быть кратким, но информативным, чтобы ясно указать основную функцию блока.
* Однозначность: Название должно быть однозначным и понятным для всех заинтересованных сторон, чтобы избежать недопонимания и разногласий по поводу того, что конкретно выполняет этот функциональный блок.
* Согласованность: Названия функциональных блоков должны быть согласованы с терминологией и описаниями в других частях модели, чтобы избежать путаницы при взаимодействии с другими моделями или документацией.

7. Приведите основные требования к названиям функциональных блоков диаграммы IDEF0.

* Однозначность: Названия функциональных блоков должны быть четкими и однозначными, чтобы представлять конкретные функции или процессы в информационной системе, и чтобы не допускать двусмысленности в их интерпретации.
* Описательность: Названия должны содержать достаточно информации, чтобы понять назначение функционального блока без необходимости обращения к дополнительным пояснениям.
* Краткость: Названия должны быть лаконичными, несмотря на их описательность, чтобы упростить восприятие и понимание модели.
* Структурированность: Названия должны быть структурированными и логичными, чтобы отражать иерархию функциональных блоков и их взаимосвязи.
* Согласованность: Названия функциональных блоков должны быть согласованы с терминологией и описаниями в других частях модели, чтобы избежать путаницы при взаимодействии с другими моделями или документацией.

8. Какие атрибуты отличают диаграмму IDEF0, отвечающую требованиям нормативных документов, от простой картинки?

* Стандартизированные символы: В диаграмме IDEF0 используются стандартизированные символы, которые четко определены и описаны в нормативных документах. Это позволяет обеспечить единообразие в представлении функциональных блоков и потоков данных.
* Систематизированная структура: Диаграмма IDEF0 следует определенной структуре, которая включает в себя уровни декомпозиции функциональных блоков, а также потоки данных и управления между ними. Это обеспечивает систематизированное и логическое представление функций информационной системы.
* Документирование: Каждый функциональный блок на диаграмме IDEF0 должен быть документирован, включая описание его функции, входы, выходы и другую соответствующую информацию. Это позволяет обеспечить полное понимание представленных элементов.
* Согласованность с терминологией: Диаграмма IDEF0 должна быть согласована с установленной терминологией и описаниями в других частях модели, что обеспечивает ее целостность и согласованность.
* Информативность: Диаграмма IDEF0 должна быть информативной и вместе с тем лаконичной, чтобы предоставлять необходимую информацию о функциональности системы.

9. Приведите требования нормативных документов по определению числа стрелок на диаграмме IDEF0. Являются ли эти требования выполнимыми? Как правильно интерпретировать эти требования?

Нормативные документы, такие как FIPS PUB 183 и MIL-STD-499, не устанавливают конкретное число стрелок на диаграмме IDEF0. Вместо этого, эти документы предоставляют стандартизированные символы и соглашения для представления потоков данных и управления между функциональными блоками.

Требования по определению числа стрелок на диаграмме IDEF0 не устанавливают конкретного количества, а вместо этого они уделяют внимание правильному использованию стрелок для отображения потоков данных и управления. Это означает, что каждая стрелка должна явно представлять поток данных или управления между соответствующими функциональными блоками.

Таким образом, важно интерпретировать эти требования как руководство по использованию стрелок для точного и понятного представления потоков данных и управления в информационной системе. Количество стрелок будет зависеть от конкретной модели и не будет ограничено нормативами, но их использование должно быть логичным, понятным и соответствовать целям моделирования.

10. Что необходимо для однозначного и полного понимания диаграммы IDEF0?

Для однозначного и полного понимания диаграммы IDEF0 необходимо:

* Знание основных символов и правил построения диаграммы IDEF0.
* Понимание контекста и цели моделирования процесса.
* Изучение всех элементов диаграммы, включая блоки функций, входы и выходы, управляющие связи и механизмы управления.
* Обязательное использование текстовых описаний функций и связей для уточнения смысла диаграммы.
* Коммуникация с участниками процесса или специалистами, чтобы получить дополнительные объяснения и уточнения.
* Проверка согласованности и соответствия диаграммы реальным процессам и операциям.
* Внимание к деталям и учет особенностей каждой функции для полного понимания взаимосвязей и зависимостей.

11. Что подразумевается под декомпозицией диаграмм в нотации IDEF0? Как можно осуществлять поиск функций-потомков и функций-родителей в нотации IDEF0? Какие CASE-средства поддерживают правильную нумерацию диаграмм и функциональных блоков IDEF0?

Декомпозиция диаграмм в нотации IDEF0 подразумевает разбиение функциональной модели на более мелкие и более детальные уровни. Это позволяет более детально изучить функции системы и их взаимосвязи.

Для поиска функций-потомков и функций-родителей в нотации IDEF0 используются специальные способы обозначения функций на диаграммах. Для функций-потомков обычно используется специальная стрелка, указывающая на более детальное разбиение функции, а для функций-родителей используется обратная стрелка, указывающая на более общее представление функции.

Некоторые CASE-средства, такие как Enterprise Architect, Sparx Systems и Visio, имеют встроенные инструменты для создания и редактирования диаграмм IDEF0. Они также поддерживают правильную нумерацию диаграмм и функциональных блоков IDEF0, что позволяет легко ориентироваться и работать с большими моделями.

12. Назовите основные правила декомпозиции функциональных моделей согласно методологии SADT.

* Разбиение на уровни абстракции: функциональная модель разбивается на отдельные уровни абстракции для более детального изучения процессов.
* Иерархическая структура: функциональная модель должна иметь иерархическую структуру, где каждая функция может разбиваться на более детальные подфункции.
* Идентификация функциональных связей: необходимо определить связи между различными функциями и подфункциями для понимания взаимодействия между ними.
* Процесс декомпозиции: функциональная модель декомпозируется на все более детальные уровни, разделяя процессы на более простые и понятные элементы.
* Определение границ функциональных элементов: необходимо определить границы каждой функциональной единицы для четкого разграничения их функций и ответственностей.
* Оценка взаимодействия: оценка взаимодействия между функциями и подфункциями позволяет выявить проблемы внутри процессов и оптимизировать их работу.
* Построение иерархии: функциональная модель строится как иерархия, где каждая функция может быть разбита на более низкие уровни до достижения наиболее детального описания процессов.

13. Какие требования предъявляются к стрелкам, входящим в блок функциональной модели IDEF0?

* Стрелки должны иметь уникальное название или идентификатор для их идентификации в рамках модели.
* Стрелки должны быть явно и четко направлены от исходного элемента к целевому элементу, обозначая поток информации, материалов или энергии между ними.
* Стрелки должны быть адекватно подписаны, чтобы обозначить характер передаваемой информации или материалов.
* Стрелки должны иметь метку или масштаб, указывающий на количество передаваемых единиц информации или материалов.
* Стрелки должны быть соединены с другими элементами модели, чтобы показать их взаимосвязь и зависимость друг от друга.
* Стрелки должны быть корректно размещены на диаграмме, с учетом логического порядка передачи информации или материалов.

14. Какие процессы отражает диаграмма в нотации DFD?

Диаграмма в нотации DFD (Data Flow Diagram) отражает процессы обработки данных в системе. На диаграмме отображаются процессы, которые преобразуют входные данные в выходные данные, данные, которые передаются между процессами, а также данные, которые хранятся и используются системой. Кроме того, на диаграмме могут быть отражены внешние агенты, которые взаимодействуют с системой, и данные, которые вводятся в систему или выводятся из нее.

15. На каком уровне декомпозиции можно использовать нотацию DFD?

Нотация DFD (диаграмма потоков данных) обычно используется на более высоком уровне декомпозиции, таком как уровень системы или подсистемы. На этих уровнях DFD помогает визуализировать потоки данных через систему, идентифицировать входы и выходы, а также сгруппировать функции системы по их взаимодействию с данными. На более низких уровнях декомпозиции обычно используются другие типы диаграмм, такие как диаграммы последовательности или диаграммы классов.

16. Сколько уровней описания в нотации DFD рекомендуется выполнить в работе?

В нотации DFD (Диаграммы потоков данных) рекомендуется выполнить три уровня описания:

* Уровень 0 - общий уровень, на котором показывается взаимодействие системы с внешними сущностями и основные процессы системы.
* Уровень 1 - более детальное разбиение процессов на подпроцессы и уточнение взаимодействия потоков данных между ними.
* Уровень 2 и выше - еще более детальное разбиение процессов и уточнение потоков данных.
* Количество уровней описания зависит от сложности системы. В больших и сложных проектах может потребоваться создание более высоких уровней детализации, чтобы полностью описать все процессы и взаимодействия системы.

17. Как рекомендуется создавать DFD-диаграмму с использованием CASE-средства Ramus Educational?

Для создания DFD-диаграммы с использованием CASE-средства Ramus Educational рекомендуется следующий подход:

* Откройте Ramus Educational и создайте новый проект.
* Выберите тип диаграммы "Диаграмма потоков данных" (DFD).
* Создайте основные элементы диаграммы, такие как процессы, внешние сущности и данные.
* Установите связи между элементами, отображая потоки данных между ними.
* Добавьте названия и описания к элементам диаграммы, чтобы обеспечить понимание их функциональности.
* Проверьте диаграмму на наличие ошибок, корректность и полноту.
* Экспортируйте готовую диаграмму в нужный формат для дальнейшего использования или публикации.

18. Как расшифровывается аббревиатура DFD?

DFD расшифровывается как "Диаграмма потоков данных" (Data Flow Diagram).

19. Как расшифровывается аббревиатура IDEF0?

DEF0 расшифровывается как Integrated DEFinition for Function Modeling, что означает Интегрированное Определение для Моделирования Функций.

20. Как расшифровывается аббревиатура SADT?

SADT расшифровывается как Structured Analysis and Design Technique, что означает Техника структурированного анализа и проектирования.

21. Назовите ограничения нотации методологии SADT.

* Стандартная диаграмма активности SADT может быть сложной и трудночитаемой при описании сложных процессов.
* Нотация SADT требует определенных навыков и знаний для понимания и использования.
* SADT может быть неэффективной при моделировании более динамичных и сложных процессов.
* Ограничение на количество уровней моделирования может затруднять адаптацию к изменяющимся условиям и требованиям.
* SADT не предоставляет механизмов для анализа и оптимизации процессов в реальном времени.

22. Назовите достоинства методологии SADT.

* Структурированность: методология SADT обладает четкой структурой, которая позволяет организовать процесс анализа системы и ее функций.
* Наглядность: SADT использует графические диаграммы и символы для представления информации, что делает процесс анализа более понятным и наглядным.
* Универсальность: методология SADT может быть применена для анализа различных типов систем и процессов, что делает ее универсальным инструментом для моделирования и анализа.
* Высокая точность: благодаря строгой структуре и методам анализа, SADT обеспечивает более точные результаты, что позволяет выявить проблемы и улучшить процессы.
* Использование стандартных нотаций: методология SADT использует стандартные нотации и символы, что упрощает взаимопонимание и обмен информацией между специалистами.

23. Поясните связь между методологией SADT и нотациями IDEF0 и DFD.

Основной целью SADT является выделение и описание функциональности системы, ее компонентов и взаимосвязей.

Нотации IDEF0 (Integration Definition for Function Modeling) и DFD (Data Flow Diagram) являются графическими нотациями, которые используются в рамках методологии SADT для моделирования функций и потоков данных в системе.

IDEF0 используется для создания диаграмм, которые показывают функциональную структуру системы, отображая взаимосвязи между функциями и их компонентами. Каждая функция представлена блоком, внутри которого указывается ее описание и входные/выходные данные.

DFD используется для моделирования потоков данных в системе, и показывает, как данные передаются между различными компонентами системы. Диаграмма DFD состоит из процессов, потоков данных, хранилищ данных и внешних сущностей.

**Введение**

Сегодня видеоигры стали занимать у людей большую часть свободного времени. Игры бывают так и платные так и бесплатные, но в основном транзакции игровых предметов проходят с помощью «внутриигровых валют».

Информационная система "Интернет-магазин игровой валюты" будет предоставлять возможность пользователям приобретать игровую валюту большинства игр с помощью онлайн-платежа.

**1 Общие сведения**

* 1. **Список терминов и определений**

Сервер — часть системы, являющаяся хостом и набором функций для сайта системы.

Клиент — часть системы, отображающая интерфейс сайта системы.

Коэффициент юзабилити — часть пользователей, которым удобно пользоваться сайтом к общему количеству пользователей системы.

Коэффициент интерактивности — часть пользователей, которая пользовалась системой за определенный промежуток времени к ожидаемому количеству пользователей за тот же промежуток времени.

БД (База Данных) — представленная в объективной форме совокупность самостоятельных материалов, систематизированных таким образом, чтобы эти материалы могли быть найдены и обработаны с помощью электронной вычислительной машины (ЭВМ).

* 1. **Описание бизнес-ролей**

Пользователь — авторизованный человек, имеющий расширенный доступ к системе, позволяющая ему просматривать свой профиль и проводить оплату.

Оператор – авторизированный пользователь, имеющий доступ к информации о всех пользователях, служащий для помощи обычным пользователям при использовании системы.

Поставщик – авторизированный пользователь, который имеет собственную страницу на сайте с предоставляемыми услугами.

Администратор — авторизованный пользователь, имеющий полный доступ к системе, позволяющая ему устранять технический ошибки системы.

1. **Проектирование контекстной диаграммы** 
   1. **План разработки модели БД**

План:

1. Планирование разработки базы данных. Включает определение объёма работ, ресурсов и стоимости проекта.
2. Определение требований к системе. Включает выбор целей БД, выяснение информационных потребностей различных отделов и руководителей фирмы, требований к оборудованию и программному обеспечению.
3. Сбор и анализ требований пользователей. На данном этапе необходимо создать модель движения важных нематериальных объектов и уяснить процесс товарообмена.
4. Проектирование базы данных. Включает концептуальное, логическое и физическое проектирование.
5. Разработка приложений. Включает проектирование транзакций и пользовательского интерфейса.
6. Реализация. На данном этапе осуществляется физическая реализация базы данных и разработанных приложений.
7. Загрузка данных. На этом этапе созданные в соответствии со схемой базы данных пустые файлы, предназначенные для хранения информации, должны быть заполнены данными.
8. Тестирование. Для оценки законченности и корректности выполнения приложения базы данных может использоваться несколько различных стратегий тестирования.
9. Эксплуатация и сопровождение. Включает анализ функционирования и поддержку исходного варианта БД, а также адаптацию, модернизацию и поддержку переработанных вариантов.
   1. **Анализ предметной области**

В базе данных должны находится таблицы, которые хранят подробную информацию о пользователях, модераторах, запросах и местах в городах. Так же должны быть обобщающие таблицы, например: таблица “Person”, которая хранит данные применимые как к пользователям, так и к модераторам (имя, фамилия, номер телефона и роль)

Таблица Пользователей имеет такие поля как: Имя пользователя, Денежный баланс, Идентификатор роли, Уникальный идентификатор пользователя.

Таблица Заказов: Метод оплаты, Идентификатор товара на маркетплейсе, Идентификатор пользователя, Уникальный идентификатор заказа.

Таблица Товаров: Цена, Идентификатор соответствующей игры, Уникальный идентификатор товара.

Таблица Игр: Название игры, Уникальный идентификатор игры.

* 1. **Проектирование ER-диаграммы**

Реализуем ER-диаграмму для проектируемой БД (рис. 1):



Рисунок 1 – ER-диаграмма БД

* 1. **Тестовые SQL запросы**

Придумаем тестовые SQL запросы для получения данных (рис. 2):

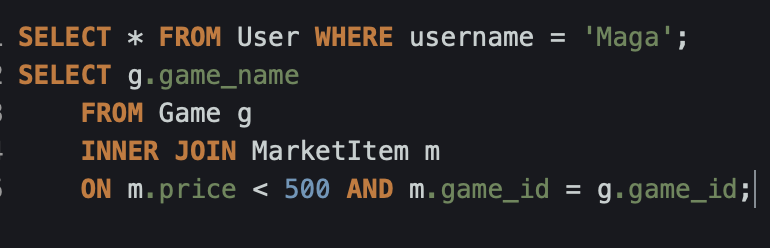


Рисунок 2 – Тестовые SQL запросы

**Вывод**

В ходе выполнения работы была реализована БД для ИС, план реализации БД, ER-диаграмма БД и тестовые SQL запросы.

**Ответы на вопросы**

1. Что такое база данных информационной системы? Дайте определение.

База данных информационной системы (БД ИС) – это организованная структура, предназначенная для хранения и управления большим объемом структурированных данных. Она обеспечивает доступ к информации, ее обновление, хранение и обработку в соответствии с заданными правилами и структурой.

1. Что такое модель данных? Дайте определение.

Модель данных — это абстрактное представление данных и их отношений в базе данных. Она определяет структуру и организацию данных в базе данных, и представляет собой некий шаблон или рамки, в которых данные могут быть хранены, обрабатываться и использоваться.

1. Что такое «сущность» в рамках ERD-моделирования?

В рамках ER-моделирования сущности — это объекты или понятия, несущие важную информацию.

1. Что такое «связь» в рамках ERD-моделирования?

В контексте ERD-моделирования "связь" (relationship) представляет собой ассоциацию между двумя или более сущностями (entities) в базе данных.

1. Какую роль играет ER-диаграмма в проекте информационной системы?

Роль ER-диаграммы в проекте информационной системы включает:

* Визуализация структуры данных: ER-диаграмма позволяет визуально представить структуру базы данных, отображая сущности (entities) и их атрибуты, а также связи между сущностями.
* Определение сущностей и их атрибутов: ER-диаграмма помогает определить сущности, их атрибуты и типы данных, которые будут храниться в базе данных.
* Определение связей между сущностями: ER-диаграмма помогает определить связи между различными сущностями в базе данных, указывая их типы (например, один к одному, один ко многим, многие ко многим).
* Помощь в коммуникации с заказчиком и разработчиками: ER-диаграмма является удобным средством для обсуждения структуры базы данных с заказчиком и разработчиками, чтобы уточнить требования к информационной системе.
* Определение целостности данных: ER-диаграмма помогает определить ограничения целостности данных, такие как уникальность ключей, ограничения на значения атрибутов и другие правила для обеспечения целостности данных в базе данных.

1. Какую задачу надо решить в процессе выполнения работы?

* Идентификация сущностей: определение всех сущностей, которые будут храниться в базе данных, и их атрибутов (характеристик).
* Определение связей между сущностями: установление связей между сущностями, указание их типов (один к одному, один ко многим, многие ко многим) и определение обязательности участия каждой сущности в связи.
* Установление ограничений целостности данных: определение правил и ограничений, которые должны быть соблюдены для обеспечения целостности данных в базе данных (например, уникальность ключей, ограничения на значения атрибутов).
* Разработка структуры ER-диаграммы: создание графического представления структуры базы данных с использованием стандартных символов и нотаций для сущностей, атрибутов и связей.
* Проверка и уточнение диаграммы: обсуждение и проверка ER-диаграммы с заказчиком и другими заинтересованными сторонами для уточнения требований и обеспечения соответствия модели базы данных потребностям информационной системы.

1. В каком аспекте рассматривается БД информационной системе в рамках данной работы?

База данных (БД) информационной системы рассматривается в рамках данной работы с точки зрения проектирования структуры данных, их организации и взаимосвязей между различными сущностями.

1. Каким образом определяются сущности в процессе создания ER-диаграммы?

В процессе создания ER-диаграммы сущности определяются на основе анализа предметной области, которая моделируется. Сущности представляют собой объекты или концепции, которые имеют свои атрибуты и связи с другими сущностями. Для определения сущностей следует проанализировать бизнес-процессы, данные и взаимодействия в предметной области, выделить основные сущности и их характеристики (атрибуты).

1. Кто из авторов предложил наиболее часто используемую нотацию ERD-моделирования?

Более часто используемая нотация ERD-моделирования была предложена Питером Ченом.

1. Что такое «экземпляр сущности»? Дайте определение.

Экземпляр сущности - это конкретный объект или элемент данной сущности, который представляет собой конкретную реализацию или экземпляр данного типа сущности.

1. Каков принцип группировки объектов в общую сущность ERD?

Принцип группировки объектов в общую сущность в ERD основан на их схожих характеристиках и отношениях между ними. Группировка происходит на основе атрибутов, которые объекты имеют в общей сущности, и их взаимосвязи друг с другом. Например, если у объектов есть одинаковые атрибуты и они имеют тесное отношение друг с другом, то их можно объединить в одну сущность.

1. Какие элементы входят в модель ERD?

* Сущности (Entity): представляют отдельные объекты, о которых мы хотим хранить информацию.
* Атрибуты (Attributes): характеристики сущностей, которые описывают их свойства.
* Связи (Relationships): определяют взаимосвязь между сущностями.
* Ключи (Keys): идентификаторы уникальных записей в базе данных.
* Ограничения (Constraints): определяют условия, которым должны соответствовать данные в базе данных.

1. Какие CASE-средства рекомендованы для разработки ER-диаграммы?

* ER Studio
* Oracle SQL Developer Data Modeler
* Microsoft Visio
* Lucidchart
* DbDesigner

1. Какие типы связей доступны при создании ER-диаграммы?

* Один ко многим
* Многие ко многим
* Один к одному
* Частичное участие
* Полное участие
* Смешанные связи
* Связи с композицией
* Связи с агрегацией
* Связи с ослаблением

1. Назовите правила и приведите пример использования связей один-ко-многим ER-диаграммы?

Правила для связей один-ко-многим в ER-диаграмме:

* У каждого элемента в сущности "один" связь может быть только один элемент в сущности "много".
* У каждого элемента в сущности "много" связь может быть любое количество элементов в сущности "один".

Предположим, у нас есть сущности "Компания" и "Сотрудник". Каждая компания имеет множество сотрудников, но каждый сотрудник работает только в одной компании. Таким образом, мы можем нарисовать связь один-ко-многим между сущностями "Компания" (один) и "Сотрудник" (много).

1. Назовите правила и приведите пример использования связей один-к-одному ER-диаграммы?

Правила использования связей один-к-одному в ER-диаграмме:

* У каждой сущности может быть только одна связь с другой сущностью.
* Каждому экземпляру сущности A соответствует только один экземпляр сущности B, и наоборот.

Предположим, у нас есть сущности "Студент" и "Книжный билет". У каждого студента может быть только один книжный билет, и у каждого книжного билета может быть только один студент. В этом случае мы можем установить связь один-к-одному между сущностями "Студент" и "Книжный билет" на ER-диаграмме

1. Назовите правила и приведите пример использования связей многие-ко-многим ER-диаграммы?

Правила связей многие-ко-многим в ER-диаграмме:

* В ER-диаграмме связь многие-ко-многим обозначается двумя ромбами, один со стороны сущности A, а другой со стороны сущности B.
* Для реализации связей многие-ко-многим используется таблица-связь, которая содержит первичные ключи сущностей A и B в качестве внешних

Предположим, у нас есть две сущности: "Студент" и "Предмет". Студент может изучать несколько предметов, и один предмет может быть изучен несколькими студентами. В этом случае у нас возникает связь многие-ко-многим.

**Введение**

Целью практической работы является проектирование диаграммы прецедентов информационной системы “Интернет-магазин игровой валюты” в нотации UML.

**1 Общие сведения**

* 1. **Список терминов и определений**

Сервер — часть системы, являющаяся хостом и набором функций для сайта системы.

Клиент — часть системы, отображающая интерфейс сайта системы.

Коэффициент юзабилити — часть пользователей, которым удобно пользоваться сайтом к общему количеству пользователей системы.

Коэффициент интерактивности — часть пользователей, которая пользовалась системой за определенный промежуток времени к ожидаемому количеству пользователей за тот же промежуток времени.

БД (База Данных) — представленная в объективной форме совокупность самостоятельных материалов, систематизированных таким образом, чтобы эти материалы могли быть найдены и обработаны с помощью электронной вычислительной машины (ЭВМ).

ИС (Информационная Система) — система, предназначенная для хранения, поиска и обработки информации, и соответствующие организационные ресурсы (человеческие, технические, финансовые и т. д.), которые обеспечивают и распространяют информацию.

Python — высокоуровневый язык программирования общего назначения с динамической строгой типизацией и автоматическим управлением памятью, ориентированный на повышение производительности разработчика, читаемости кода и его качества, а также на обеспечение переносимости написанных на нём программ.

PEP8 — документ, описывающий соглашение о том, как писать код на языке Python.

Система контейнеризации — это технология абстракции, которая позволяет упаковывать и исполнять приложения вместе со всеми их зависимостями в изолированных средах, называемых контейнерами.

Система оркестрации — система автоматического размещения, координации и управления сложными компьютерными системами и службами.

* 1. **Описание бизнес-ролей**

Пользователь — авторизованный человек, имеющий расширенный доступ к системе, позволяющая ему просматривать свой профиль и проводить оплату.

Оператор – авторизированный пользователь, имеющий доступ к информации о всех пользователях, служащий для помощи обычным пользователям при использовании системы.

Поставщик – авторизированный пользователь, который имеет собственную страницу на сайте с предоставляемыми услугами.

Администратор — авторизованный пользователь, имеющий полный доступ к системе, позволяющая ему устранять технический ошибки системы.

1. **Создание диаграммы состояний**

Реализуем диаграмму состояний для информационной системы “Интернет-магазин игровой валюты” на языке UML (рис. 1).



Рисунок 1 – Диаграмма состояний системы Интернет-магазин игровой валюты

**Заключение**

В ходе выполнения работы была спроектирована диаграмма состояний информационной системы “Интернет-магазин игровой валюты” на языке UML.

**Ответы на вопросы**

1. Для чего в язык UML введена диаграмма состояний?

Диаграмма состояний в UML введена для моделирования поведения объекта или системы в зависимости от его состояния и внешних событий. Она позволяет описать все возможные состояния, переходы между ними и события, вызывающие эти переходы. Диаграмма состояний удобна для представления дискретных изменений в системе и является полезным инструментом для анализа и проектирования систем, где важно моделировать поведение объектов или систем в зависимости от их состояния.

1. Для каких целей используется диаграмма состояний UML?

* Моделирование поведения: Диаграмма состояний позволяет моделировать поведение объекта или системы в зависимости от его состояния и внешних событий.
* Описание жизненного цикла: Диаграмма состояний помогает в описании всех возможных состояний, переходов между ними и событий, вызывающих эти переходы, что позволяет лучше понять жизненный цикл объекта или системы.
* Анализ и проектирование систем: Эта диаграмма является полезным инструментом для анализа и проектирования систем, где важно моделировать поведение объектов или систем в зависимости от их состояния.
* Визуализация: Диаграмма состояний помогает визуализировать различные состояния и переходы объекта или системы, что упрощает понимание и взаимодействие с ними.

1. Что выступает в качестве объекта моделирования диаграммы состояний UML?

В диаграмме состояний UML объектом моделирования выступает объект или система, поведение которого зависит от его состояния и внешних событий. На диаграмме состояний моделируются различные состояния, переходы между ними и события, вызывающие эти переходы для данного объекта или системы. Таким образом, объект, чье поведение моделируется, является объектом моделирования на диаграмме состояний UML.

1. Каким образом передается основная информация, заложенная в диаграмме состояний UML?

* Состояния: Основной информацией являются различные состояния, в которых может находиться объект или система. Эти состояния отображаются на диаграмме и описывают поведение объекта в определенных условиях.
* Переходы: Диаграмма состояний показывает переходы между различными состояниями и событиями, вызывающими эти переходы. Это позволяет передать информацию о том, как объект или система реагирует на внешние события и изменяет свое состояние.
* События: События, вызывающие переходы между состояниями, также являются частью основной информации на диаграмме состояний. Они описывают воздействия, которые приводят к изменению состояния объекта или системы.
* Действия: Диаграмма состояний также может включать информацию о действиях, которые выполняются при переходах между состояниями, что позволяет передать информацию о поведении объекта или системы в различных состояниях.

1. Какие типы отношений чаще всего применяются в диаграмме состояний UML?

* Переходы: отображаются стрелками и показывают переходы между различными состояниями объекта или системы в ответ на определенные события. Переходы представляют собой важную часть диаграммы состояний, так как они определяют поведение объекта в зависимости от его текущего состояния и внешних событий.
* Вложенные состояния: Этот тип отношений позволяет моделировать иерархию состояний, где одно состояние может содержать в себе другие состояния. Вложенные состояния позволяют более подробно описывать состояния объекта или системы и их взаимосвязи.
* История состояний: Этот тип отношений используется для обозначения возвращения к предыдущему состоянию, в зависимости от истории переходов. Это позволяет моделировать поведение, где объект возвращается к определенному состоянию в зависимости от предыдущих переходов.

1. Как можно представить диаграмму состояний UML в аналитическом виде?

Диаграмма состояний UML может быть представлена в аналитическом виде с использованием следующей структуры:

* Название диаграммы: (название диаграммы состояний)
* Цель диаграммы: (описание цели и назначения диаграммы)
* Объекты: (перечисление всех объектов, которые участвуют в состояниях)
* Состояния: (перечисление всех возможных состояний объектов)
* Переходы: (описание условий и действий, которые приводят к изменению состояний объектов)
* Начальное состояние: (начальное состояние объекта при запуске системы)
* Финальное состояние: (конечное состояние объекта при завершении работы системы).

**Введение**

Сегодня видеоигры стали занимать у людей большую часть свободного времени. Игры бывают так и платные так и бесплатные, но в основном транзакции игровых предметов проходят с помощью «внутриигровых валют».

Информационная система "Интернет-магазин игровой валюты" будет предоставлять возможность пользователям приобретать игровую валюту большинства игр с помощью онлайн-платежа.

**1 Общие сведения**

* 1. **Список терминов и определений**

Сервер — часть системы, являющаяся хостом и набором функций для сайта системы.

Клиент — часть системы, отображающая интерфейс сайта системы.

Коэффициент юзабилити — часть пользователей, которым удобно пользоваться сайтом к общему количеству пользователей системы.

Коэффициент интерактивности — часть пользователей, которая пользовалась системой за определенный промежуток времени к ожидаемому количеству пользователей за тот же промежуток времени.

БД (База Данных) — представленная в объективной форме совокупность самостоятельных материалов, систематизированных таким образом, чтобы эти материалы могли быть найдены и обработаны с помощью электронной вычислительной машины (ЭВМ).

ИС (Информационная Система) — система, предназначенная для хранения, поиска и обработки информации, и соответствующие организационные ресурсы (человеческие, технические, финансовые и т. д.), которые обеспечивают и распространяют информацию.

Python — высокоуровневый язык программирования общего назначения с динамической строгой типизацией и автоматическим управлением памятью, ориентированный на повышение производительности разработчика, читаемости кода и его качества, а также на обеспечение переносимости написанных на нём программ.

PEP8 — документ, описывающий соглашение о том, как писать код на языке Python.

Система контейнеризации — это технология абстракции, которая позволяет упаковывать и исполнять приложения вместе со всеми их зависимостями в изолированных средах, называемых контейнерами.

Система оркестрации — система автоматического размещения, координации и управления сложными компьютерными системами и службами.

* 1. **Описание бизнес-ролей**

Пользователь — авторизованный человек, имеющий расширенный доступ к системе, позволяющая ему просматривать свой профиль и проводить оплату.

Оператор – авторизированный пользователь, имеющий доступ к информации о всех пользователях, служащий для помощи обычным пользователям при использовании системы.

Поставщик – авторизированный пользователь, который имеет собственную страницу на сайте с предоставляемыми услугами.

Администратор — авторизованный пользователь, имеющий полный доступ к системе, позволяющая ему устранять технический ошибки системы.

1. **Проектирование контекстной диаграммы** 
   1. **Описание ЭСЕ**

Элементарная семантическая единица (ЭСЕ) – неделимая единица информации, использующаяся в ИС. ЭСЕ представляет собой завершенную контекстную конструкцию, вызываемую в результате поиска по различным атрибутам или в результате тех или иных команд в виде отклика или отчета. В случае исследования настоящей системы за элементарную семантическую единицу была выбрана одна из характеристик поиска, а именно количество мест, возвращаемых на запрос. В нашем примере эта величина меняется случайным образом в пределах от 100 до 500 [продавцов].

* 1. **Наполнение системы**

Проектируемая информационная система может быть наполнена практически любым количеством элементов базы данных. Их количество ограничиваются только параметрами сервера.

В рамках данной система была наполнена работы Система была наполнена 100 ЭСЕ. В рамках ограничений объема данной работы, невозможно привести полный перечень всех записей ЭСЕ, поэтому пример первых пяти записей приведен в таблице 1.

Структуризация ведется по количеству сертификатов, возвращаемых на

запрос.

Таблица 1 – Список элементарных семантических единиц

|  |  |
| --- | --- |
| **Наименование** | **Параметр** |
| Robux | 420 |
| Fortnite V-Bucks | 380 |
| FIFA point | 150 |
| Destiny 2 Silver | 230 |
| Rocket League Credits | 300 |

* 1. **Математические расчеты**

Для дальнейшего исследования проектируемой ИС необходимо рассчитать вероятности, с которыми ЭСЕ принимает то или иное значение. Для оценки этих вероятностей было принято решение разбить весь диапазон значений на 5 дискретных величин. Расчеты ведутся с помощью формулы P(ξ)=n/N, где n – благоприятное число исходов (в данном случае число мест, попадающих в данный диапазон), а N – общее число исходов. В таблице 2 приведены возможные значения, принимаемые ЭСЕ и их вероятности.

Таблица 2 – Ряд распределения

|  |  |  |
| --- | --- | --- |
| **№** | **x** | **P(x)** |
| 1 | 102,1 | 14/100=0,14 |
| 2 | 181,5 | 23/100=0,23 |
| 3 | 260,9 | 21/100=0,21 |
| 4 | 340,3 | 29/100=0,29 |
| 5 | 419,7 | 13/100=0,13 |

* 1. **Расчет математического ожидания блока системы**

Используя данные, полученные в таблице 2, сделаем расчёт математического ожидания по формуле 1 и получим:

М (5) = 264.07 [продавцов].

* 1. **Расчет дисперсии информационного блока системы**

Используя данные, полученные в таблице 2, получаем:

D (5) = 10076.8 [].

**2.6 Расчет среднеквадратичного отклонения**

**2.7 Расчет энтропии системы**

За основание логарифма **a** возьмем двоичную систему счисления и получаем:

**Вывод**

В данной практической работе был осуществлен расчет основных характеристик проектируемой ИС, и получены следующие результаты (см. таблицу 3):

Таблица 3 – Параметры проектируемой ИС

|  |  |
| --- | --- |
| Параметр | Значение |
| Математическое ожидание | 158,46 [мест] |
| Дисперсия | 1315,90 [] |
| Среднеквадратичное отклонение | 657,95 [мест] |
| Энтропия информационного наполнения | 1,563 [бит] |

**Приложения**

Приложение А. Код математических расчетов.

import math

g\_x = [

102.1,

181.5,

260.9,

340.3,

419.7,

]

g\_p = [

0.14,

0.23,

0.21,

0.29,

0.13,

]

def mat\_ozhidanie(x, p):

return sum([x[i] \* p[i] for i in range(5)])

def dispers(x, p):

return sum([p[i] \* x[i]\*\*2 for i in range(5)]) - mat\_ozhidanie(x, p) \*\* 2

def otkl(x, p):

return math.sqrt(dispers(x, p))

def entropy(p):

return -sum([p[i] \* math.log2(p[i]) for i in range(5)])

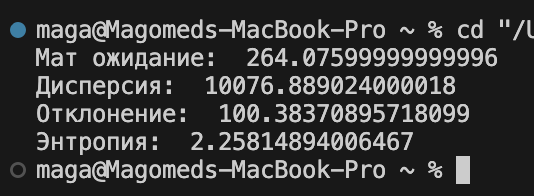
print('Мат ожидание: ', mat\_ozhidanie(g\_x, g\_p))

print('Дисперсия: ', dispers(g\_x, g\_p))

print('Отклонение: ', otkl(g\_x, g\_p))

print('Энтропия: ', entropy(g\_p))

Результат работы программы:

****

**Ответы на вопросы**

1. Что такое энтропия информационной системы?

Энтропия информационной системы — это мера неопределенности или неожиданности в ней. Чем больше энтропия, тем больше неопределенность и неожиданность в системе. Меньшая энтропия означает более предсказуемую и структурированную систему, в то время как большая энтропия указывает на хаос и беспорядок. В информационной теории энтропия определяется как количество информации, которое требуется для описания состояния системы.

1. Дайте определение термину «Семантическая информация».

Семантическая информация — это информация, содержащая в себе значение и смысл, которые позволяют описать объект или явление в мире и понять его суть. Это информация, которая не просто описывает факты или данные, но и передает их значения и связи между ними. В контексте информационных технологий семантическая информация отражает смысл и содержание данных, позволяя им быть интерпретированными и использованными для принятия решений.

1. Что такое энтропия Шеннона?

Энтропия Шеннона — это мера неопределенности или случайности в системе информации. Она используется для измерения количества информации, содержащейся в сообщении или данном.

1. Что такое дисперсия информации?

Дисперсия информации — это мера разброса значений данных относительно их среднего значения. Это показатель, который позволяет оценить, насколько данные отличаются друг от друга. В статистике дисперсией информации называется среднее квадратичное отклонение значений от их среднего значения. Величина дисперсии позволяет оценить степень разнообразия данных в наборе информации.

1. Взаимосвязь реляционных данных, корреляция данных.

Взаимосвязь реляционных данных отражает какие-либо связи или зависимости между переменными в базе данных. Это может быть как простая связь между двумя таблицами, так и более сложные отношения, такие как один ко многим или многие ко многим.

Корреляция данных отражает степень взаимосвязи между двумя переменными. Если две переменные коррелируют, это означает, что изменение одной переменной может быть связано с изменением другой переменной. Корреляция может быть положительной (обе переменные меняются в одном направлении), отрицательной (одна переменная увеличивается, а другая уменьшается) или нулевой (нет корреляции).

1. Что такое модели данных?

Модель данных — это абстрактная структура, которая описывает способ организации и хранения данных в базе данных. Она определяет типы данных, их отношения и способы доступа к данным. Модель данных является основой для проектирования баз данных и позволяет эффективно организовывать и обрабатывать информацию.

1. Что такое мера информации?

Мера информации — это количественное выражение степени неопределенности или неизвестности, содержащейся в сообщении. Она позволяет оценить количество информации, которое содержится в передаваемом сообщении или сигнале. Измеряется обычно в битах или байтах и может использоваться для оценки эффективности передачи информации или управления ею.

1. Как выполнить анализ параметров проектируемой информационной системы?

Для выполнения анализа параметров проектируемой информационной системы необходимо провести следующие шаги:

* Определить цель создания информационной системы и ее основные задачи.
* Изучить бизнес-процессы организации, на основе которых будет функционировать информационная система.
* Определить основные требования к информационной системе, такие как функциональные и нефункциональные требования, безопасность, надежность, производительность и т.д.
* Провести анализ рынка и конкурентов для определения требований к информационной системе, которые позволят организации оставаться конкурентоспособной.
* Выявить ограничения и ожидания пользователей от информационной системы.
* Разработать структуру и архитектуру информационной системы на основе выявленных требований.
* Провести тестирование и оценку информационной системы, чтобы удостовериться, что она соответствует заданным параметрам.
* Определить критерии успешности информационной системы и планы дальнейшего развития.