

БУ ВО
«СУРГУТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ХАНТЫ-МАНСКИЙСКОГО АВТОНОМНОГО ОКРУГА - ЮГРЫ»

ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ
Кафедра информатики и вычислительной техники

УТВЕРЖДАЮ:
Заведующий кафедрой ИВТ
Канд. физ.-мат. наук, доцент
_____ С.А.Лысенкова
" __ " _____ 2024г.

КУРСОВОЙ ПРОЕКТ
по дисциплине «**Методы и средства проектирования
информационных систем**»
бакалавра по направлению
09.03.02 - Информационные системы и технологии

на тему ***Информационная система по сборке персонального компьютера из
совместимых комплектующих***

Выполнил: бакалавр группы: 607-01
_____ Шумский Даниил Иванович
(фамилия, имя, отчество)
по специальности 09.03.02 - Информационные
системы и технологии

Проверил:
преподаватель Столбов Д.А. _____
подпись

Сургут – 2024

Реферат

Курсовой проект содержит 34 страницы, 10 рисунков, 4 таблицы, 8 источников, 2 приложение.

Информационная система по сборке персонального компьютера из совместимых комплектующих

Цель работы: проектирование ИС по сборке ПК из совместимых комплектующих.

Основные результаты: разработана проектная модель, охватывающая все необходимые аспекты проекта, подлежащие реализации в течение всего процесса разработки информационной системы по сборке персонального компьютера.

Оглавление

Словарь терминов	5
Введение.....	7
Основная часть	8
1. Постановка задачи	8
2. Описание предметной области	9
2.1. Заинтересованные лица	9
2.2. Модель предметной области.....	10
3. Обзор аналогов	11
4. Требования к системе	14
4.1. Функциональные требования	14
4.2. Требования по видам обеспечения.....	15
4.2.1. Требования к алгоритмическому обеспечению	15
4.2.3. Требования к информационному обеспечению.....	16
4.2.4. Техническое обеспечение	17
4.2.5. Лингвистическое обеспечение.....	17
4.2.6. Требования к методическому обеспечению.....	18
5. Проектирование системы	19
5.1. Диаграмма вариантов использования	19
5.2. Анализ прецедентов.....	19
5.2.1. Прецедент «Последовательный выбор комплектующих при сборке ПК».....	20
5.2.2. Прецедент «Альтернативный сценарий последовательного выбора комплектующих ПК».....	21
5.2. Диаграммы процессов BPMN	23
5.3. Диаграмма классов.....	24

5.4. Информационная модель данных	24
5.5. Архитектура системы	25
5.6. Прототип UI.....	26
6. Разработка информационной системы	28
6.1. Backlog	28
6.2. Функциональные блоки программы	28
Заключение	30
Список источников	31
Приложение А. Исходный код правил проверки совместимости.....	32
Приложение В. Таблица атрибутов сущностей.	34

Словарь терминов

Таблица 1. Словарь терминов

Термин	Определение	Формат	Синоним
Блок-схема	Графическое представление алгоритма, которое использует специальные символы и блоки...	Н/А	Алгоритм
Блок питания	Устройство, обеспечивающее электроэнергию для работы компьютерных компонентов.	Мощность	Питание, PSU
Видеокарта	Устройство, отвечающее за обработку и вывод графики на монитор.	Объем видеопамяти в гб	GPU, Графический адаптер
Геймер	Человек, играющий в видео-игры	Н/А	Игрок
Жесткий диск (ЖД)	Устройство для долгосрочного хранения данных, используемых компьютером.	Объем	HDD, накопитель
ИС	Информационная система	Н/А	Система обработки информации
Компоненты ПК	Отдельные части, из которых состоит компьютерная система.	Н/А	Элемент, тип, часть, девайс
Корпус	Оболочка, в которой размещаются и защищаются компоненты ПК.	Форм-фактор	Кейс, Case
Материнская плата	Основная печатная плата, на которой устанавливаются и взаимодействуют другие компоненты ПК.	Поддерживаемый сокет	МП, Motherboard
Оперативная память (ОЗУ)	Временное хранилище данных, используемое компьютером для выполнения операций в реальном времени.	Объем	ОЗУ, RAM
ПО	Программное обеспечение	Н/А	Программа, продукт
Процессор	Центральный вычислительный блок ПК, отвечающий за	Сокет	CPU

	выполнение операций и управление системой.		
Прямоугольник	Обозначает конкретные действия или операции, которые выполняются, такие как вычисления или...	Н/А	Блок действия
Ромб	Представляет проверку условия, и в зависимости от результата выполняется различное действие.	Н/А	Условный блок
Совместимость	Возможность различных компонентов ПК работать вместе без конфликтов.	Совместимо/несовместимо	Сочетаемость, несочетаемость
SSD (Solid State Drive)	Твердотельный накопитель, использующий флэш-память для хранения данных.	Объем	Накопитель, твердотельный накопитель

Введение

В настоящее время на рынке присутствует несколько программ для сборки ПК. PCPartPicker, Intel ARK, AMD Product Selector. Одним из основных недостатков этих программ является отсутствие локализации на различные языки, поскольку они предоставляются исключительно на английском языке. Это может создать преграды для пользователей, не владеющих английским языком, особенно в технической сфере, где использование специфичных терминов обычно требует хорошего владения языком. Кроме того, важно отметить, что данные программы не учитывают местоположение пользователей.

Таким образом, цены и доступность продуктов, представленных в этих программах, могут быть актуальными только для определенных регионов, таких как Европа или США, и не могут быть применимыми к пользователям из других частей мира. Это может ограничить возможности выбора и оптимизации конфигурации компьютера для пользователей из различных регионов.

Дополнительно, следует отметить, что некоторые другие программы, доступные на рынке, сосредоточены преимущественно на продвижении своих собственных продуктов и не стремятся к поддержке "кросс-платформинга". Это означает, что программы могут быть ограничены в своей способности предложить всесторонние решения, а именно интеграцию с продуктами от различных производителей или поддержку различных платформ.

Основная часть

1. Постановка задачи

Цель курсовой работы- анализ требований и проектирование информационной системы по сборке персонального компьютера с добавлением проверки совместимости комплектующих.

Задачи:

1. Изучить предметную область
2. Проанализировать существующие аналоги
3. Разработать требования к информационной системе
4. Разработать информационную систему и пользовательский интерфейс
5. Разработать набор правил проверки совместимости
6. Разработать базу данных

Цель информационной системы:

Обеспечение пользователям возможности создания совместимой конфигурации компьютера.

Задачи информационной системы:

Предоставить удобный и эффективный способ выбора комплектующих, проверить их совместимость и обеспечить информацию, необходимую для успешной сборки и работы созданной системы.

2. Описание предметной области

2.1. Заинтересованные лица

Таблица 2. Заинтересованные лица

Роль	Цель	Требование	Текущее решение
Энтузиасты сборки ПК	Легкий и точный выбор совместимых компонентов и создание оптимальных систем	ПО для выбора и сборки совместимых компонентов	Онлайн-ресурсы, форумы, калькуляторы совместимости
Геймеры	Выбор наиболее подходящих компонентов для игровой системы	ПО для выбора оптимальных компонентов с учетом требований игр	Онлайн-ресурсы, форумы, рекомендации от друзей
Профессиональные пользователи	Выбор и сборка компонентов, соответствующих специфическим требованиям	ПО для выбора и сборки компонентов с учетом профессиональных требований	Онлайн-ресурсы, специализированные магазины, консультации специалистов
Технические специалисты	Быстрый анализ совместимости компонентов при обновлении или расширении системы	ПО для анализа совместимости компонентов и предоставления рекомендаций	Ручной анализ, онлайн-ресурсы, форумы, консультации специалистов

2.2. Модель предметной области

Предметной областью данной работы является процесс согласованной компоновки и подключения различных аппаратных компонентов, таких как центральный процессор (CPU), материнская плата, оперативная память (RAM), жесткие диски (HDD/SSD), видеокарты, блоки питания, охлаждающие системы и другие устройства, с целью создания работоспособной системы обработки информации.

Сборка ПК является важным этапом в создании компьютерных систем различного назначения, будь то игровой ПК, рабочая станция для проектирования или сервер для облачных вычислений. В современном мире у сборки ПК существует широкий спектр возможностей для настройки и оптимизации, чтобы соответствовать конкретным потребностям и требованиям пользователей.

Данный процесс включает в себя выбор совместимых компонентов, на основе правил проверки совместимости комплектующих персонального компьютера. Например, материнская плата должна поддерживать процессор выбранной марки и поколения, оперативная память должна быть совместима с материнской платой по частоте и типу слотов и т.д.

На рисунке 1 изображена концептуальная модель предметной области.

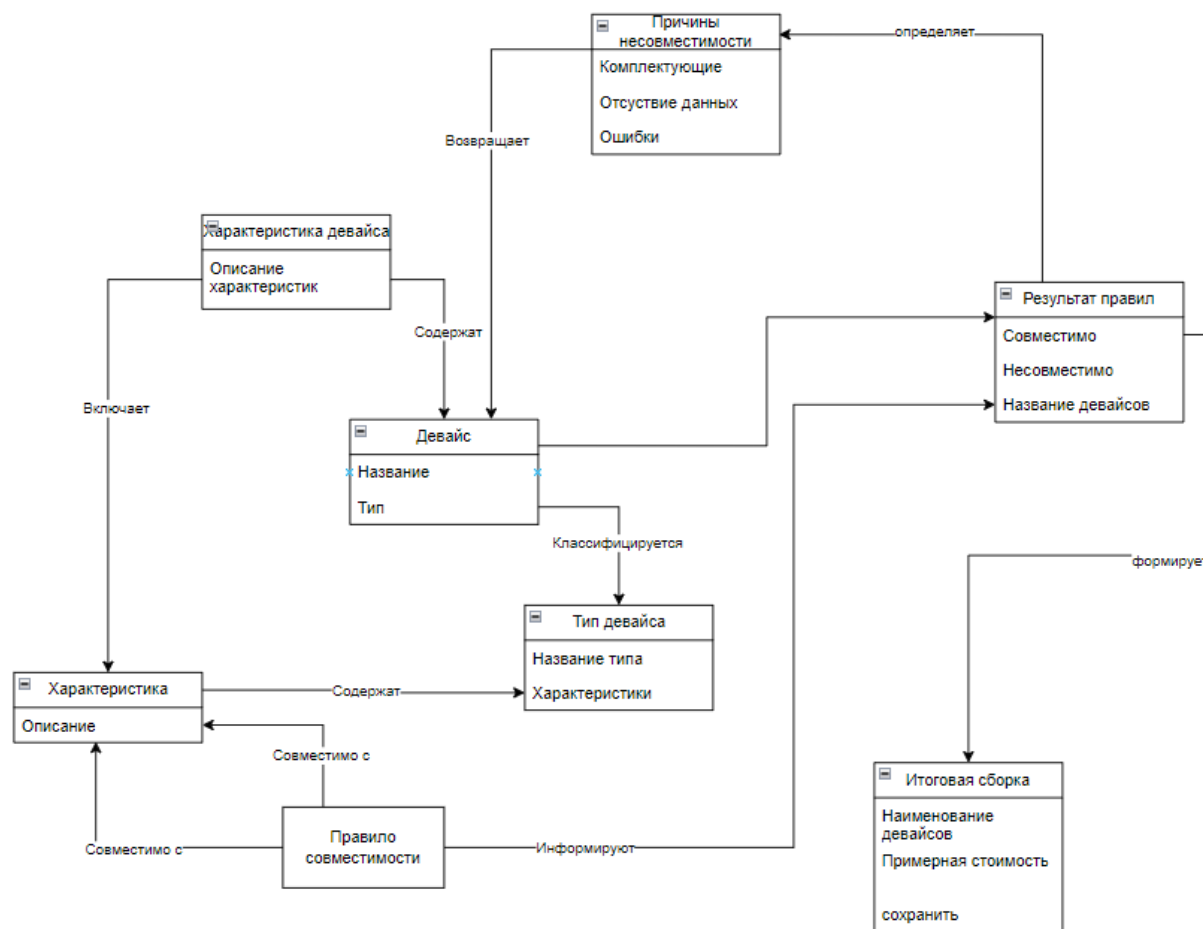


Рис.1. Концептуальная модель предметной области

3. Обзор аналогов

В качестве аналогов будут рассмотрены различные платформы для сборки персонального компьютера. Обзор аналогов представлен в виде таблицы.

В таблице 3 приведены сравнительные характеристики трех решений, аналогов, по различным критериям.

Таблица 3. Обзор аналогов

Критерий	PCPartPicker	Intel ARK	AMD Product Selector
Обширная база данных	PCPartPicker предлагает широкий выбор компонентов	Intel ARK предоставляет информацию только о процессорах	AMD Product Selector предоставляет информацию о продуктах AMD
Информация о ценах и характеристиках	PCPartPicker предоставляет подробные сведения о ценах, характеристиках и отзывах пользователей	Intel ARK предлагает точные и подробные данные о характеристиках процессоров Intel	AMD Product Selector предоставляет информацию о характеристиках продуктов AMD
Удобный интерфейс	Интерфейс PCPartPicker интуитивно понятен и прост в использовании	Инструмент Intel ARK позволяет пользователям легко искать и сравнивать процессоры по моделям	AMD Product Selector также предлагает удобный интерфейс для выбора и сравнения компонентов
Ошибки в совместимости	Возможны случаи неправильной классификации и ошибочной информации о совместимости между компонентами	Intel ARK является официальным ресурсом Intel, предоставляющим точные данные о совместимости процессоров Intel	AMD Product Selector предоставляет информацию о совместимости продуктов AMD
Официальный источник	PCPartPicker не является официальным ресурсом производителя	Intel ARK является офици-	AMD Product Selector является

Критерий	PCPartPicker	Intel ARK	AMD Product Selector
информации		альным ресурсом Intel, предоставляющим точные данные о процессорах Intel	официальным ресурсом AMD, предоставляющим информацию о продуктах AMD
Совместимость с другими производителями	PCPartPicker предлагает широкий выбор компонентов, но не всегда предоставляет полную информацию о совместимости с компонентами других производителей	Intel ARK предоставляет информацию только о процессорах Intel, не охватывая совместимость с компонентами других производителей	AMD Product Selector предоставляет информацию о совместимости продуктов AMD, но не охватывает совместимость с компонентами других производителей

ПО имеет хорошие перспективы в современной сфере компьютерных технологий. Вот несколько факторов, указывающих на его перспективы:

Растущий рынок сборки ПК: Сборка собственных компьютеров становится все более популярной среди энтузиастов и геймеров. ПО.

Постоянное развитие технологий: С появлением новых компонентов и обновлений старых, пользователи нуждаются в инструментах, которые помогут им ориентироваться во всей этой информации и выбирать наиболее подходящие компоненты.

Удобство и эффективность: Пользователи ищут решения, которые помогут им сэкономить время и усилия при сборке ПК. ПО, которое предоставляет удобный интерфейс, точную информацию о совместимости.

Потребность в профессиональных и оптимизированных системах: Профессионалы, такие как дизайнеры, разработчики и видеомонтажеры, нуждаются в

мощных и оптимизированных системах. ПО, которое помогает им выбрать и собрать подходящие компоненты, будет ценным инструментом для таких пользователей.

Возможности расширения функциональности: ПО может постепенно расширять свою функциональность, предлагая дополнительные сервисы, такие как интеграция с онлайн-магазинами компонентов, анализ производительности системы и другие инструменты оптимизации.

Однако следует отметить, что конкуренция в этой области также может быть интенсивной. Чтобы успешно развиваться, ПО должно предлагать уникальные и полезные функции, акцентировать внимание на удобстве использования и поддерживать актуальность своих данных и функциональности. Регулярные обновления и поддержка пользователей также могут способствовать успеху и долгосрочным перспективам такого ПО.

4. Требования к системе

4.1. Функциональные требования

Функциональные требования – это перечень сервисов, которые должна выполнять разрабатываемая информационная система, а также элементы этой системы.

Программа должна предоставлять данный функционал:

1. Подбирать комплектующие ПК

Эта функция позволяет пользователю выбирать компоненты компьютера, такие как процессор, видеокарта, оперативная память, жёсткий диск и т.д. Может включать фильтры по параметрам, таким как цена, производительность, бренд и т.д.

2. Проверять совместимость комплектующих ПК

Программа должна автоматически проверять, совместимы ли выбранные пользователем компоненты между собой. Например, убеждаться, что выбранный процессор подходит для выбранной материнской платы и т.д.

3. Выводить результат проверки совместимости

После проверки совместимости компонентов программа должна предоставить пользователю информацию о результате проверки, указав на какие-либо несовместимости.

4. Выводить итоговую сборку ПК

После успешного подбора и проверки совместимости компонентов, программа должна отобразить пользователю итоговую сборку ПК, включая все выбранные компоненты с их характеристиками и ценами.

5. Редактировать итоговую сборку

Пользователь должен иметь возможность вносить изменения в итоговую сборку, добавляя, удаляя или заменяя компоненты на своё усмотрение.

6. Сохранять итоговую сборку

4.2. Требования по видам обеспечения

4.2.1. Требования к алгоритмическому обеспечению

Блок-схема – это графическое представление алгоритма, которое использует специальные символы и блоки для описания последовательности шагов или операций. Она помогает визуализировать и структурировать алгоритм, делая его более понятным и легким для анализа.

Алгоритм работы ИС представлен на рисунке 2.

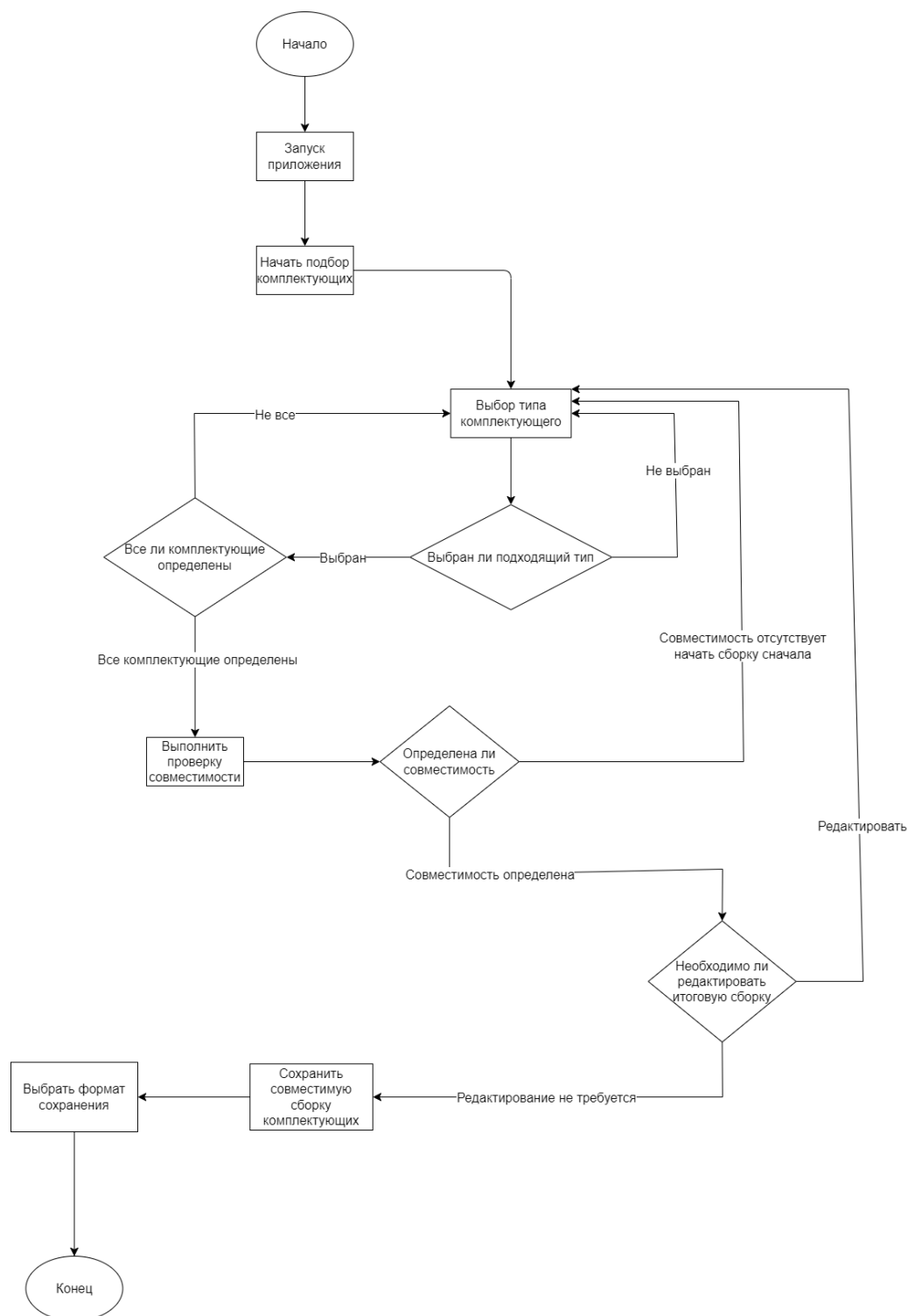


Рис. 2. Алгоритм работы ИС

4.2.3. Требования к информационному обеспечению

Источником информации ИС является excel файл с данными о комплектующих ПК. Входными данными системы являются данные, заполняемые пользо-

вателем при сборке ПК, которые в свою очередь берутся из Excel файла, представленного на рисунке 3. Выходными данными системы является готовая сборка ПК.

Наименование	Тип видеокарты	Память	Частота ядра	Цена в Плеер.ру	Цена в Олди	Цена в Ситилинк	Цена в CU	Цена в Регард	Цена в ДНС
Gigabyte GeForce RTX 4060 Ti GAMING OC 8Gb	GeForce RTX 4060 Ti	8GB	1822 MHz	+23 047 p.	+31 398 p.	+39 598 p.	23 432 p.	+28 678 p.	+24 567 p.
Palit GeForce RTX 3060 Dual OC 12GB	GeForce RTX 3060	12GB	1365 MHz	+12 915 p.	+21 062 p.	+24 672 p.	18 008 p.	+16 492 p.	+11 691 p.
Gigabyte GeForce RTX 4090 GAMING OC 24GB	GeForce RTX 4090	24GB	1842 MHz	+109 658 p.	+236 975 p.	+152 295 p.	100 995 p.	+201 115 p.	+135 004 p.
Palit GeForce RTX 4070 Ti GamingPro OC 12Gb	GeForce RTX 4070 Ti	12GB	1665 MHz	+35 562 p.	+44 358 p.	+44 578 p.	50 922 p.	+50 678 p.	+34 077 p.
Palit GeForce RTX 4070 Dual 12Gb	GeForce RTX 4070	12GB	1515 MHz	+27 748 p.	+32 657 p.	+48 617 p.	34 193 p.	+45 487 p.	+30 806 p.
Gigabyte GeForce RTX 4070 WINDFORCE OC 12Gb	GeForce RTX 4070	12GB	1670 MHz	+32 018 p.	+42 495 p.	+50 945 p.	34 415 p.	+37 405 p.	+36 884 p.

Рис.3. Excel файл с характеристиками комплектующих

4.2.4. Техническое обеспечение

Для обеспечения работоспособности ИС необходима клиентская часть. ПК с минимальными системными требованиями:

- Процессор: Intel Core i3 или выше, AMD Ryzen 3 или выше.
- Видеокарта: Дискретная видеокарта
- Размер, занимаемый на диске: от 2 ГБ
- Размер оперативной памяти: от 8 ГБ.

4.2.5. Лингвистическое обеспечение

1. Разработка desktop приложения:

- Использование языка программирования Python.
- Использование языка программирования C#.
- Использование Excel в качестве базы данных.

2. Описание предметной области:

- Использование языка моделирования UML.
- Использование языка моделирования BPMN.
- Использование диаграммы компонентов.
- Использование Use Case диаграммы.

4.2.6. Требования к методическому обеспечению

При разработке данной информационной системы и создании документации на неё, необходимо руководствоваться следующими нормативными документами:

- ГОСТ 34.201-89. Информационная технология. Комплекс стандартов на автоматизированные системы. Виды, комплексность и обозначение документов при создании автоматизированных систем.
- ГОСТ Р 50.1.028-2001 Информационные технологии поддержки жизненного цикла продукции. Методология функционального моделирования.
- ГОСТ Р 51583-2014 Защита информации. Порядок создания автоматизированных систем в защищенном исполнении. Общие положения.
- ГОСТ Р ИСО/МЭК 13335 Информационная технология. Методы и средства обеспечения безопасности.
- ГОСТ 19.402-78 Описание программы.

5. Проектирование системы

5.1. Диаграмма вариантов использования

Use case-диаграмма (диаграмма прецедентов) является графическим инструментом моделирования, который используется для описания функциональности системы и ее взаимодействия с акторами (пользователями или другими системами).

На рисунке 4 представлены диаграммы вариантов использования, которые хорошо визуализируют функциональные требования.

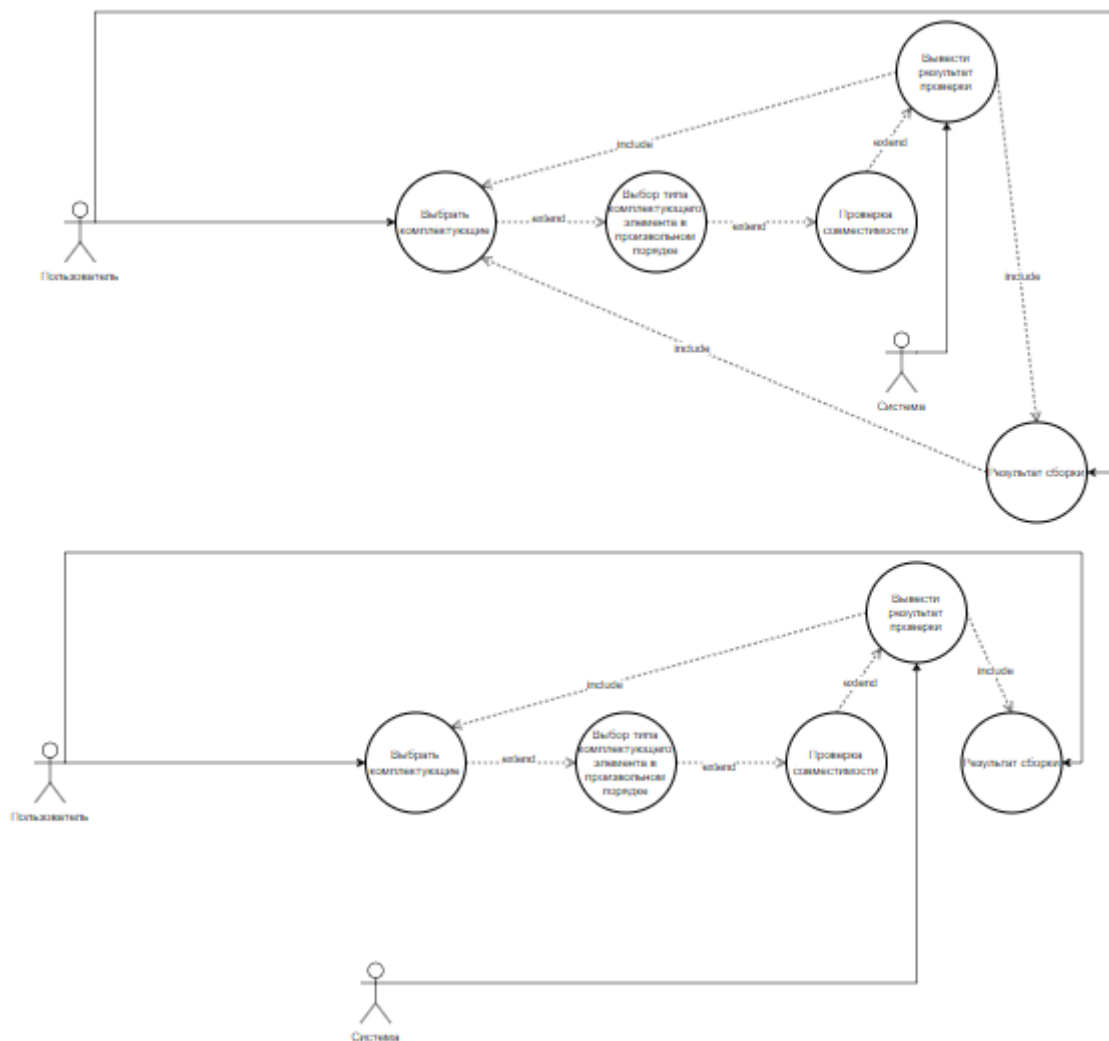


Рис.4. Диаграммы вариантов использования

5.2. Анализ прецедентов

Анализ прецедентов является процессом выделения, анализа и документирования прецедентов для разработки программного продукта. Данный анализ

позволяет понять, какие функциональные требования должны быть удовлетворены системой, какие шаги и взаимодействия между актерами необходимы для достижения этих целей.

5.2.1. Прецедент «Последовательный выбор комплектующих при сборке ПК»

Участники:

Пользователь

Система по сборке ПК (часть программного обеспечения)

Описание:

Данный прецедент описывает ситуацию, в которой пользователь взаимодействует с системой по сборке ПК для последовательного выбора комплектующих при создании персонального компьютера. Прецедент позволяет пользователю быть уверенным в совместимости комплектующих элементов.

Предусловия:

Пользователь запустил программу системы по сборке ПК.

Основной сценарий:

1. Пользователь запускает систему по сборке ПК.
2. Система приветствует пользователя и предлагает начать процесс выбора комплектующих.
3. Пользователь указывает свои требования или предпочтения для ПК (например, цель использования, бюджет).
4. Система анализирует введенную информацию и предлагает пользователю выбрать первый комплектующий из списка (например, процессор).
5. Пользователь выбирает комплектующий из предложенных вариантов.
6. Система анализирует выбор пользователя и переходит к следующему комплектующему (например, материнская плата).
7. Шаги 5-6 повторяются для каждого комплектующего, пока не будут выбраны все необходимые компоненты для ПК.
8. Система проверяет совместимость(совместимо).
9. Система предоставляет пользователю итоговую конфигурацию ПК, основанную на выбранных комплектующих.

10. Пользователь может сохранить конфигурацию или запросить дополнительные рекомендации и изменения (собрать заново).

Постусловия:

Пользователь получает итоговую конфигурацию ПК на основе выбранных комплектующих.

Исключения:

Если пользователь собирает ПК и в нем присутствует элемент, который со сборкой работать не будет, то система предложит перейти к редактированию сборки.

В случае, если система не может предложить подходящие варианты комплектующих на основе предоставленных требований, она может предложить пользователю пересмотреть свои предпочтения или обратиться за дополнительной помощью.

Комплектующие могут иметь некоторые особенности использования

Этот прецедент описывает основной сценарий использования системы по сборке ПК, где пользователь взаимодействует с программным обеспечением для последовательного выбора комплектующих при создании персонального компьютера.

5.2.2. Прецедент «Альтернативный сценарий последовательного выбора комплектующих ПК»

Название прецедента:

Последовательный выбор комплектующих при сборке ПК (Несовместимо)

Участники:

Пользователь

Система по сборке ПК (часть программного обеспечения)

Описание:

Данный прецедент описывает ситуацию, в которой пользователь взаимодействует с системой по сборке ПК для последовательного выбора комплектующих при создании персонального компьютера. Прецедент позволяет пользователю быть уверенным в совместимости комплектующих элементов.

Предусловия:

Пользователь запустил программу системы по сборке ПК.

Основной сценарий:

1. Пользователь запускает систему по сборке ПК.
2. Система приветствует пользователя и предлагает начать процесс выбора комплектующих.
3. Пользователь указывает свои требования или предпочтения для ПК (например, цель использования, бюджет).
4. Система анализирует введенную информацию и предлагает пользователю выбрать первый комплектующий из списка (например, процессор).
5. Пользователь выбирает комплектующий из предложенных вариантов.
6. Система анализирует выбор пользователя и переходит к следующему комплектующему (например, материнская плата).
7. Шаги 5-6 повторяются для каждого комплектующего, пока не будут выбраны все необходимые компоненты для ПК.
8. Система проверяет совместимость(несовместимо).
9. Система сообщает о несовместимости сборки ПК
10. Возвращает к шагу 5 до тех пор, пока не выполнится первый сценарий

Постусловия:

Пользователь получает итоговую конфигурацию ПК на основе выбранных комплектующих.

Исключения:

1. Если пользователь собирает ПК и в нем присутствует элемент, который со сборкой работать не будет, то система предложит перейти к редактированию сборки.

2. В случае, если система не может предложить подходящие варианты комплектующих на основе предоставленных требований, она может предложить пользователю пересмотреть свои предпочтения или обратиться за дополнительной помощью.
3. Комплектующие могут иметь некоторые особенности использования

5.2. Диаграммы процессов BPMN

BPMN — система условных обозначений и их описания для моделирования бизнес-процессов.

Для демонстрации был выбран процесс последовательной сборки ПК, изображенный на рисунке 5.

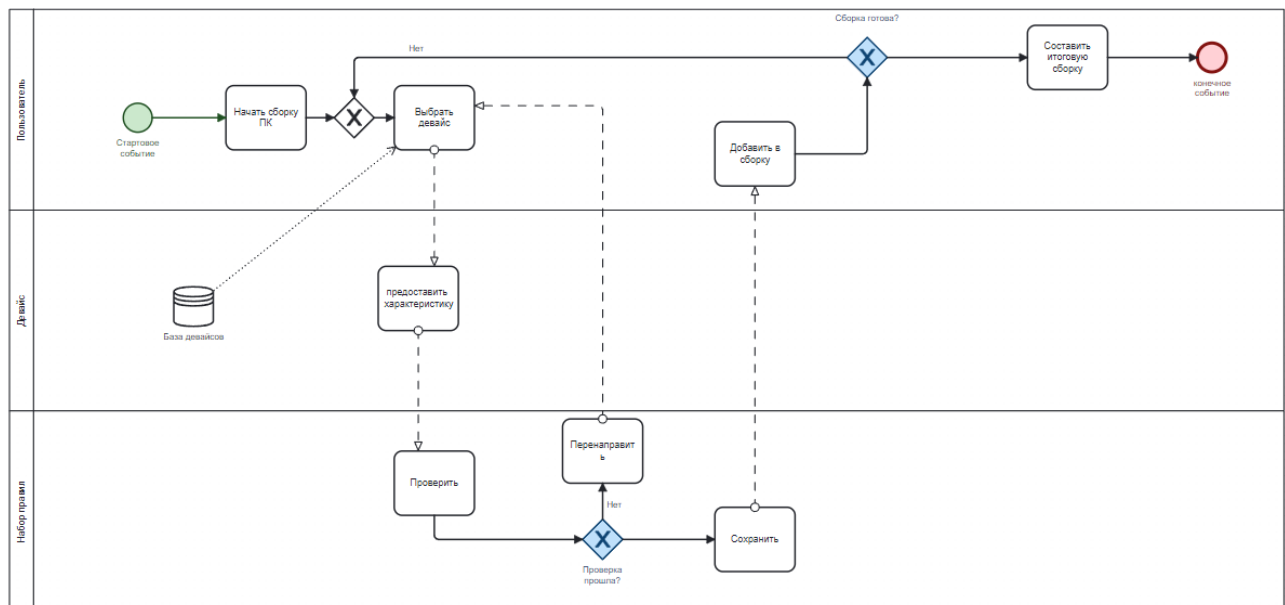


Рис.5. Диаграмма процессов (BPMN)

Пользователь начинает процесс с указания своих параметров, таких как цель использования ПК, бюджет и желаемые характеристики. Система, в свою очередь, анализирует вводимую информацию и предлагает пользователю выбрать каждый компонент по очереди, начиная с процессора и продолжая выбор материнской платы, видеокарты и других необходимых элементов. Повторение шагов выбора и анализа продолжается до тех пор, пока все компоненты не будут выбраны, после чего система проверяет их совместимость и предоставляет итоговую конфигурацию ПК.

Система предоставляет пользователю возможность сохранить конфигурацию или запросить дополнительные рекомендации и изменения для ее доработки. В случае необходимости пользователь может начать процесс сборки заново, внося коррективы в выбор комплектующих. Этот подход позволяет пользователям удобно и эффективно подбирать компоненты для сборки персонального компьютера, обеспечивая оптимальное сочетание требований, бюджета и желаемых характеристик.

5.3. Диаграмма классов

На рисунке 6 приведена диаграмма классов для предметной области

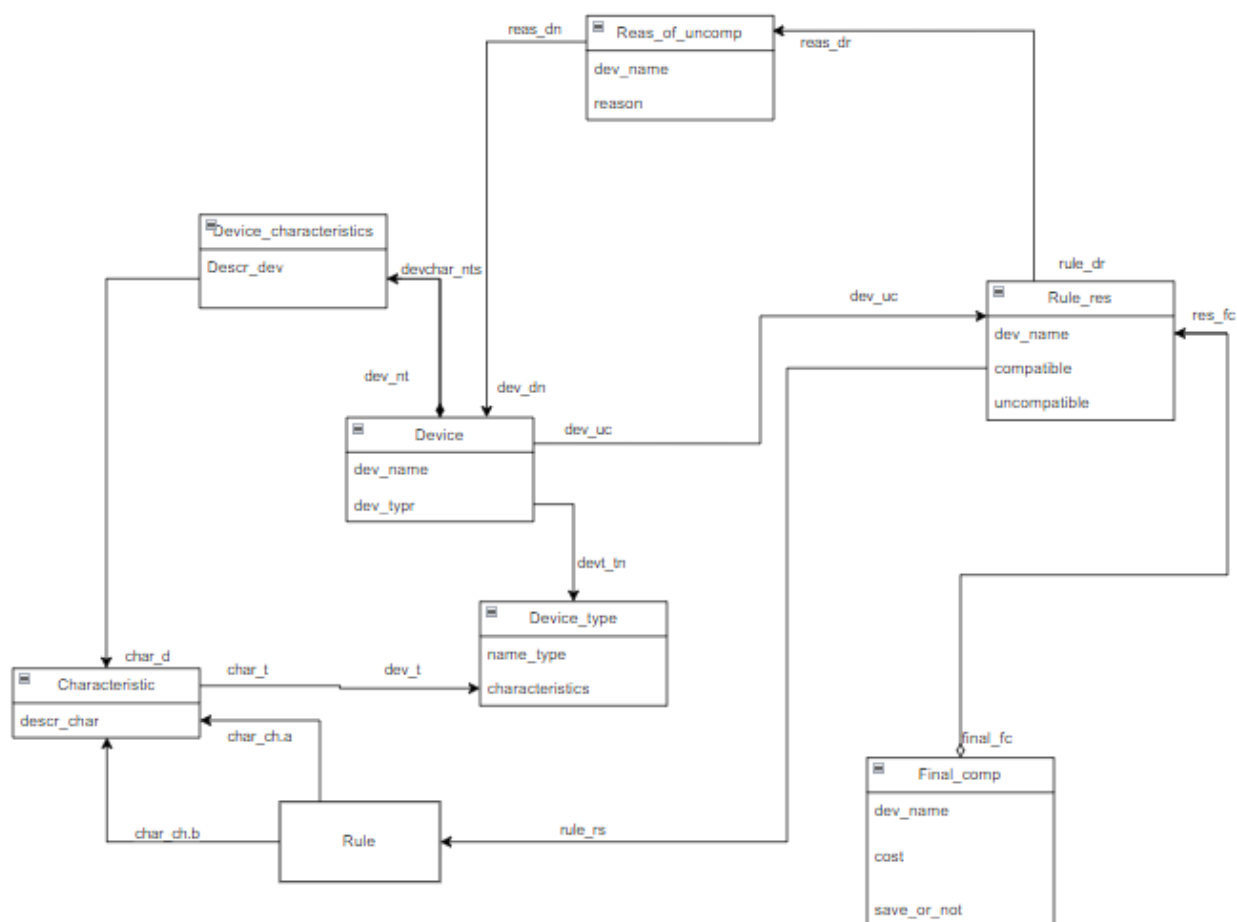


Рис.6.Диаграмма классов

5.4. Информационная модель данных

В Таблице 4 приведено соотношение сущностей в диаграмме классов с их фактическими названиями.

Таблица 4. «Сущность – Наименование»

Сущность	Наименование
Characteristics	Характеристики
Device	Девайс
Device_characteristics	Характеристика девайса
Device_type	Тип девайса
Rules	Правила совместимости
Rule_res	Результат правил
Reas_of_uncomp	Причины несовместимости
Final_comp	Итоговая сборка

Данная таблица составлена на основании диаграммы классов.

В приложении В представлена таблица атрибутов и ассоциаций сущностей.

5.5. Архитектура системы

Диаграмма компонентов — элемент языка моделирования UML, статическая структурная диаграмма, которая показывает разбиение программной системы на структурные компоненты и связи между компонентами. В качестве физических компонентов могут выступать файлы, библиотеки, модули, исполняемые файлы, пакеты и т. д.

На рисунке 7 представлена архитектура системы в виде диаграммы компонентов.

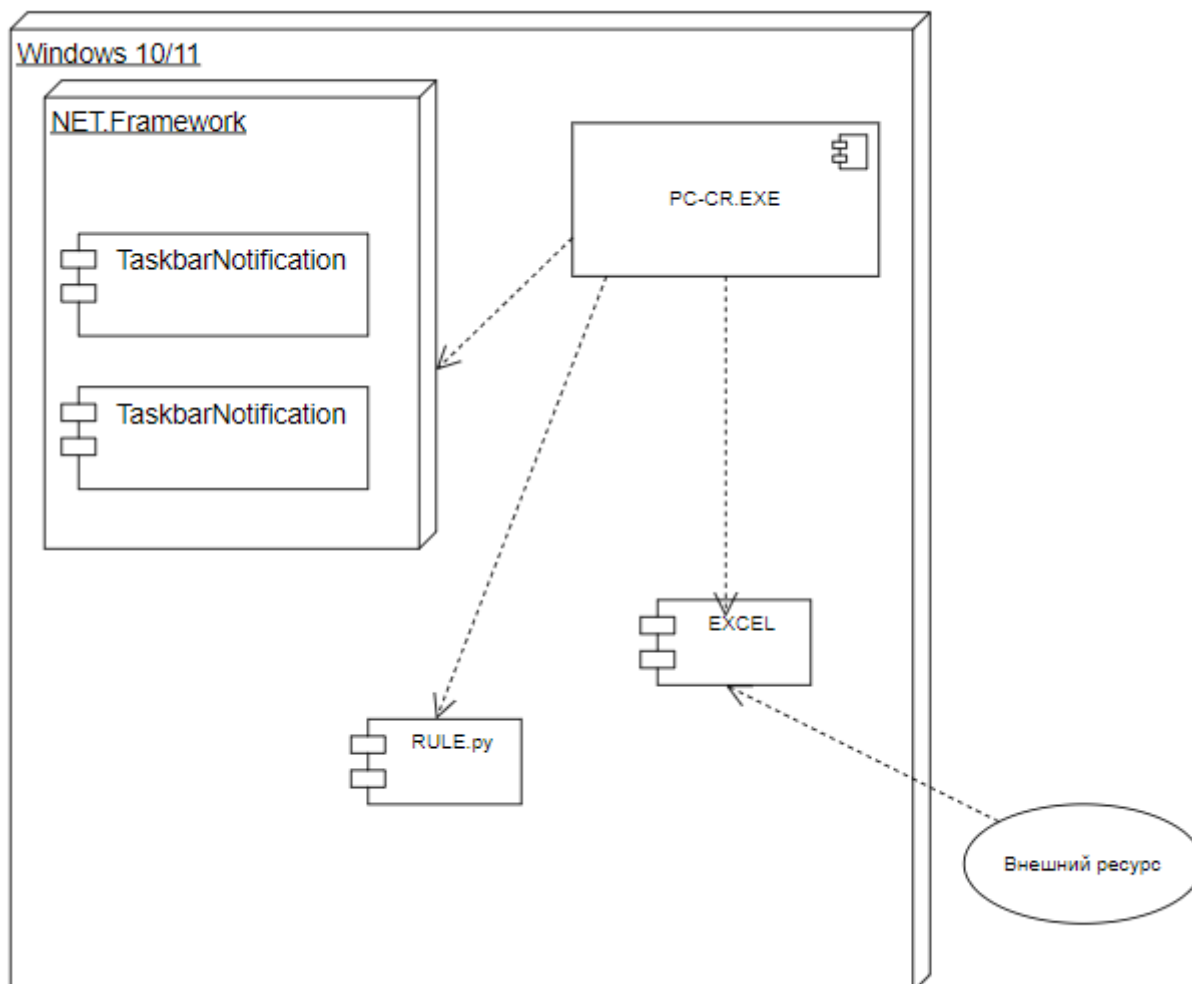


Рис. 7 Диаграмма компонентов

Диаграмма компонентов состоит из Desktop приложения на базе Windows. На данной модели представлены компоненты:

1. PC-CR.exe – исполняемый файл, при открытии которого запускается приложение.
2. Excel – база данных с использованием excel для хранения данных комплектующих
3. Taskbar Notification – библиотека для отображения приложения в панели задач.
4. Внешний ресурс – ресурс из которого берутся данные для excel
5. RULE – файл с набором правил проверки совместимости

5.6. Прототип UI

Макет дизайна разрабатывался в Figma и выглядит следующим образом. Главное меню на рисунке 8 состоит из нескольких элементов:

1. Список комплектующих для выбора
2. Проверка на совместимость
3. Итоговая сборка

При нажатии на кнопку «Проверить совместимость» пользователь в разделе «Итоговая сборка» увидит список комплектующих, а напротив списка будет отображаться, совместим ли элемент ПК или же нет. Также при нажатии на изображения комплектующих, можно попасть к выбору компонентов компьютера.

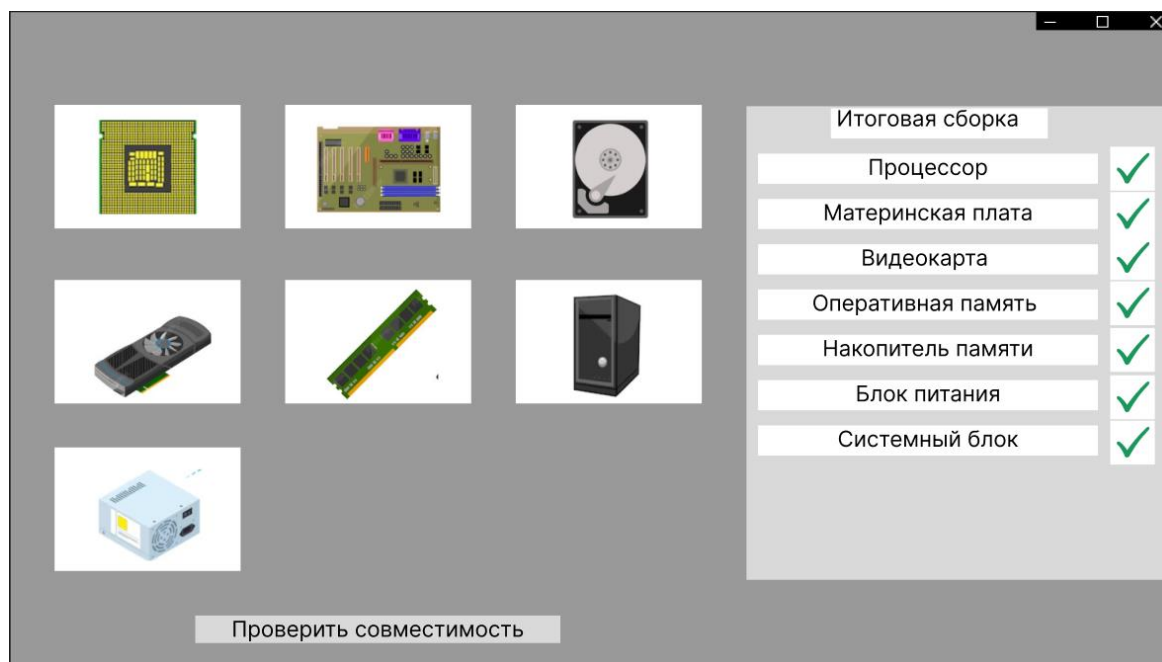


Рис.8. Главное меню

На рисунке 9 изображен выбор непосредственно элемента ПК, в данном случае указан процессор, но помимо него можно будет выбирать и другие компоненты персонального компьютера. На представленном прототипе пользовательского интерфейса можно выбрать компонент из предложенного списка, просмотреть их характеристики, возможность добавить данный компонент в сборку, который в дальнейшем будет отображен на рисунке 8.

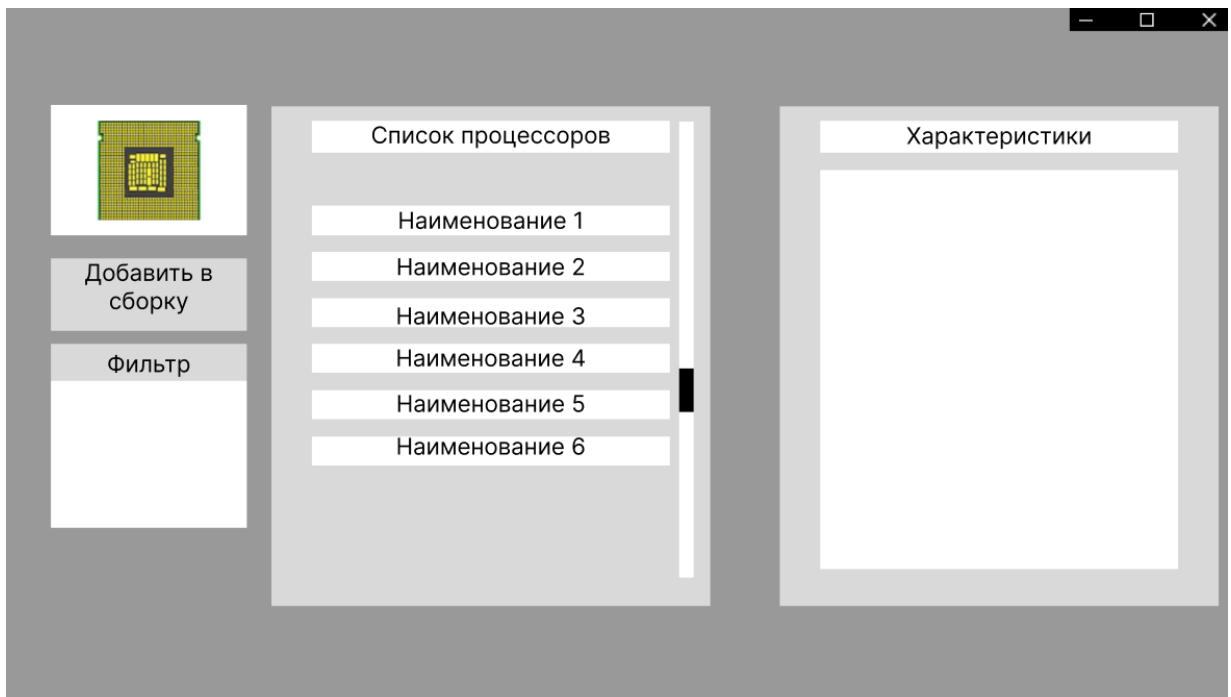


Рис.9. Выбор комплектующих

6. Разработка информационной системы

6.1. Backlog

Список задач по разработке информационной системы может выглядеть следующим образом:

1. Разработка набора правил проверки совместимости
2. Разработка базы данных с характеристиками комплектующих.
3. Разработка дизайна
4. Реализация интерфейса.
5. Реализация функционала программы под ОС Windows.
6. Оптимизация ИС

6.2. Функциональные блоки программы

Рассмотрим набор правил проверки совместимости с заранее заданными тестовыми значениями. Набор правил представлен на рисунке 10.

```

# Операция: Проверка совместимости компонентов ПК
class PCCompatibilityChecker:
    def check_compatibility(self):
        # Ввод сокета материнской платы
        socket_mb = input("Введите сокет материнской платы ('lga 1151', 'am5', 'lga 1200', 'am4', 'lga 1700', 'lga1851'): ").lower()
        while socket_mb not in ['lga 1151', 'am5', 'lga 1200', 'am4', 'lga 1700', 'lga1851']:
            socket_mb = input("Введите верный сокет материнской платы (lga 1151, 'am5', 'lga 1200', 'am4', 'lga 1700', 'lga1851'): ").lower()

        # Ввод поддерживаемой оперативной памяти
        ram_type = input("Введите поддерживаемую оперативную память (DDR4 или DDR5): ").lower()
        while ram_type not in ['ddr4', 'ddr5', 'ddr2', 'ddr3']:
            ram_type = input("Введите верный тип оперативной памяти ('ddr4', 'ddr5', 'ddr2', 'ddr3'): ").lower()

        # Ввод форм-фактора материнской платы
        mb_form_factor = input("Введите форм-фактор материнской платы (ATX или mATX): ").lower()
        while mb_form_factor not in ['atx', 'matx', 'ddr4', 'ddr5', 'ddr2', 'ddr3']:
            mb_form_factor = input("Введите верный форм-фактор материнской платы (ATX или mATX): ").lower()

        # Ввод сокета процессора
        socket_cpu = input("Введите сокет процессора (lga 1151, 'am5', 'lga 1200', 'am4', 'lga 1700', 'lga1851'): ").lower()
        while socket_cpu not in ['lga 1151', 'am5']:
            socket_cpu = input("Введите верный сокет процессора (LGA 1151 или AM5): ").lower()

        # Проверка совместимости сокетов материнской платы и процессора
        if socket_cpu == socket_mb:
            print("Совместимо")
        else:
            print("Не совместимо")
            continue # Перезапуск цикла ввода значений

        # Ввод типа оперативной памяти
        ram_type_input = input("Введите тип оперативной памяти (DDR4 или DDR5): ").lower()
        while ram_type_input not in ['ddr4', 'ddr5']:
            ram_type_input = input("Введите верный тип оперативной памяти (DDR4 или DDR5): ").lower()

        # Проверка совместимости типа оперативной памяти с поддерживаемой материнской платой
        if ram_type_input == ram_type:
            print("Совместимо")
        else:
            print("Не совместимо")
            continue # Перезапуск цикла ввода значений

        # Ввод мощности блока питания и потребления системы
        power_supply = int(input("Введите мощность блока питания: "))
        power_consumption = int(input("Введите общее потребление системы: "))

        # Проверка рекомендуемости блока питания
        if power_supply > power_consumption:
            print("Рекомендуется")
        else:
            print("Не рекомендуется")
            continue # Перезапуск цикла ввода значений

# Использование класса для проверки совместимости компонентов ПК
checker = PCCompatibilityChecker()
checker.check_compatibility()

```

Рис.10. Набор правил

Данный набор правил включает в себя проверку совместимости каждого комплектующего. Проверка совместимости будет закончена только тогда, когда все элементы компьютера будут совместимы. В приложении А представлена полная версия кода.

Заключение

На протяжении двух семестров был получен теоретический материал и приобретены практические навыки проектирования информационных систем. В рамках курса были детально изучены UML диаграммы и различные модели. В рамках выполнения курсового проекта была успешно спроектирована информационная система по сборке ПК с проверкой совместимости комплектующих.

Результаты работы могут быть использованы для дальнейшего совершенствования и развития информационной системы по сборке ПК с проверкой совместимости комплектующих.

Список источников

1. А.А. Буров, А.В. Буров. "Система сборки компьютера." URL: <https://www.elibrary.ru/item.asp?id=24413653> (Дата обращения 2023)
2. К.Ю. Береговой. "Сборка компьютера: шаг за шагом." URL: <https://www.elibrary.ru/item.asp?id=32846091> (Дата обращения 2023)
3. В.А. Петров, А.С. Смирнов. "Особенности сборки компьютеров с использованием процессоров Intel и AMD." URL: <https://www.elibrary.ru/item.asp?id=32526497> (Дата обращения 2023)
4. А.Г. Кузнецов, А.В. Мальцев. "Выбор комплектующих для сборки игрового компьютера." URL: <https://www.elibrary.ru/item.asp?id=32846052> (Дата обращения 2023)
5. Ю.А. Костюков, А.В. Лобанов. "Сборка компьютера для работы с большими объемами данных." URL: <https://www.elibrary.ru/item.asp?id=32526617> (Дата обращения 2023)
6. Л.Д. Иванова, А.В. Колесников. "Основные принципы сборки компьютера." URL: <https://www.elibrary.ru/item.asp?id=32526608> (Дата обращения 2023)
7. А.А. Петров, Н.П. Смирнова. "Сборка компьютера для домашнего использования." URL: <https://www.elibrary.ru/item.asp?id=32526581> (Дата обращения 2023)
8. К.А. Баранов, А.В. Голубев. "Сборка компьютера для работы с графикой и видео." URL: <https://www.elibrary.ru/item.asp?id=32526592> (Дата обращения 2023)

Приложение А. Исходный код правил проверки совместимости.

Операция: Проверка совместимости компонентов ПК

class PCCompatibilityChecker:

```
    def check_compatibility(self):

        # Ввод сокета материнской платы

        socket_mb = input("Введите сокет материнской платы ('lga 1151', 'am5', 'lga 1200', 'am4', 'lga 1700', 'lga1851'): ").lower()

        while socket_mb not in ['lga 1151', 'am5', 'lga 1200', 'am4', 'lga 1700', 'lga1851']:

            socket_mb = input("Введите верный сокет материнской платы (lga 1151, 'am5', 'lga 1200', 'am4', 'lga 1700', 'lga1851'): ").lower()

        # Ввод поддерживаемой оперативной памяти

        ram_type = input("Введите поддерживаемую оперативную память (DDR4 или DDR5): ").lower()

        while ram_type not in ['ddr4', 'ddr5', 'ddr2', 'ddr3']:

            ram_type = input("Введите верный тип оперативной памяти ('ddr4', 'ddr5', 'ddr2', 'ddr3'): ").lower()

        # Ввод форм-фактора материнской платы

        mb_form_factor = input("Введите форм-фактор материнской платы (ATX или mATX): ").lower()

        while mb_form_factor not in ['atx', 'matx', 'ddr4', 'ddr5', 'ddr2', 'ddr3']:

            mb_form_factor = input("Введите верный форм-фактор материнской платы (ATX или mATX): ").lower()

        # Ввод сокета процессора

        socket_cpu = input("Введите сокет процессора (lga 1151, 'am5', 'lga 1200', 'am4', 'lga 1700', 'lga1851'): ").lower()

        while socket_cpu not in ['lga 1151', 'am5']:

            socket_cpu = input("Введите верный сокет процессора (LGA 1151 или AM5): ").lower()

        # Проверка совместимости сокетов материнской платы и процессора

        if socket_cpu == socket_mb:

            print("Совместимо")

        else:

            print("Не совместимо")

            continue # Перезапуск цикла ввода значений

        # Ввод типа оперативной памяти

        ram_type_input = input("Введите тип оперативной памяти (DDR4 или DDR5): ").lower()

        while ram_type_input not in ['ddr4', 'ddr5']:

            ram_type_input = input("Введите верный тип оперативной памяти (DDR4 или DDR5): ").lower()

        # Проверка совместимости типа оперативной памяти с поддерживаемой материнской платой

        if ram_type_input == ram_type:

            print("Совместимо")

        else:

            print("Не совместимо")

            continue # Перезапуск цикла ввода значений
```



```

# Ввод мощности блока питания и потребления системы
power_supply = int(input("Введите мощность блока питания: "))
power_consumption = int(input("Введите общее потребление системы: "))

# Проверка рекомендуемости блока питания
if power_supply > power_consumption:
    print("Рекомендуется")
else:
    print("Не рекомендуется")
    continue # Перезапуск цикла ввода значений

# Ввод сокета кулера
socket_cooler = input("Введите сокет кулера (LGA 1151 или AM5): ").lower()
while socket_cooler not in ['lga 1151', 'am5']:
    socket_cooler = input("Введите верный сокет кулера (LGA 1151 или AM5): ").lower()

# Проверка совместимости сокетов процессора и кулера
if socket_cooler == socket_cpu:
    print("Совместимо")
else:
    print("Не совместимо")
    continue # Перезапуск цикла ввода значений

# Ввод форм-фактора системного блока
case_form_factor = input("Введите форм-фактор системного блока (ATX или mATX): ").lower()
while case_form_factor not in ['atx', 'matx']:
    case_form_factor = input("Введите верный форм-фактор системного блока (ATX или mATX): ").lower()

# Проверка совместимости форм-фактора материнской платы и системного блока
if case_form_factor == mb_form_factor:
    print("Совместимо")
else:
    print("Не совместимо")
    continue # Перезапуск цикла ввода значений

# Использование класса для проверки совместимости компонентов ПК
checker = PCCompatibilityChecker()
checker.check_compatibility()

```

Приложение В. Таблица атрибутов сущностей.

Сущность	Ключ	Атрибут	Обязательность	Тип	Отношение	Описание
Characteristic	+	descr_char	+	String		Описание
Characteristic		dev_t	+	String	1...*	
Device_characteristic		descr_dev	+	String		Описание
Device_characteristic		char_d	+	String	1...*	
Device_characteristic		dev_nt	+	String	1...*	
Rule	+	rule_rs	+	String	1...*	
Rule	+	char_ch	+	String	1...*	
Device_type	+	name_type	+	String		Имя
Device_type	+	characteristics	+	String		Описание
Device_type	+	devc_tn	+	String	1...*	
Device		dev_name	+	String		Имя
Device		dev_typr	+	String		Тип
Device		dev_uc	+	String	1...*	
Rule_res		dev_name	+	String		Имя
Rule_res		compatible	+	String		Совместимо
Rule_res		uncompatible	+	String		Несовместимо
Rule_res		final_fc	+	String	1...*	
Rule_res		reas_dr	+	String	1...*	
Reas_of_uncomp		dev_name	+	String		Имя
Reas_of_uncomp		reason	+	String		Причина
Reas_of_uncomp		dev_dn	+	String	1...*	
Final_comp		dev_name	+	String		Имя
Final_comp		cost	+	Integer		Цена
Final_comp		save_or_not	+	String		Сохранить или нет