

**Юсупов Арафат Русланович,**  
Инженер Центрального Кампуса  
УНПК «Международный Университет Кыргызстана»  
Email: yu.su.pov.03ara.fat@gmail.com

**Юсупов Арафат Русланович,**  
Борбордук кампустун инженери  
ОИӨК «Кыргызстан эл аралык университети»  
Email: yu.su.pov.03ara.fat@gmail.com

**Iusupov Arafat Ruslanovich,**  
Central Campus engineer  
ERPC "International University of Kyrgyzstan"  
Email: yu.su.pov.03ara.fat@gmail.com

## РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ ДЛЯ СБОРКИ И ЗАКУПКИ КОМПЬЮТЕРОВ

## КОМПЬЮТЕРДИ ЧОГУЛТУУ ЖАНА САТЫП АЛУУ ҮЧҮН МААЛЫМАТТЫК СИСТЕМАНЫ ИШТЕП ЧЫГУУ

## DEVELOPMENT OF AN INFORMATION SYSTEM FOR THE ASSEMBLY AND PURCHASE OF COMPUTERS

---

**Аннотация:** В статье рассматривается проектирование и разработка интеллектуальной информационной системы, предназначенной для автоматизированной сборки и покупки персональных компьютеров. Основной целью проекта является создание удобного веб-приложения, способного с помощью технологий искусственного интеллекта (ИИ) формировать конфигурации ПК на основе бюджета, задач пользователя или требований к совместимости комплектующих. Статья содержит обоснование актуальности, анализ аналогов, методы, архитектуру, а также планируемые показатели эффективности и анализ рисков.

**Ключевые слова:** информационные системы, сборка компьютеров, искусственный интеллект (ИИ), рекомендательные системы, машинное обучение, обработка естественного языка (NLP), матричная факторизация.

**Аннотациясы:** Макалада жеке компьютерлерди автоматташтырылган чогултуу жана сатып алуу үчүн иштелип чыккан акылдуу маалымат тутумунун дизайны жана иштелип чыгышы каралат. Долбоордун негизги максаты-жасалма интеллект (КИ) технологиясынын жардамы менен бюджеттин, колдонуучунун тапшырмаларынын же компоненттердин шайкештик талаптарынын негизинде компьютердин конфигурациясын түзүүгө жөндөмдүү колдонуучуга ыңгайлуу веб-тиркемени түзүү. Макалада актуалдуулугун негиздөө, аналогдорду талдоо, методдор, архитектура, ошондой эле пландаштырылган натыйжалуулук көрсөткүчтөрү жана тобокелдиктерди талдоо камтылган.

**Негизги сөздөр:** Ачкыч сөздөр: маалымат тутумдары, компьютерлерди чогултуу, жасалма интеллект(ЖИ), рекомендация тутумдары, машиналарды үйрөнүү, табигый тилди иштетүү (NLP), матрицалык факторизация.

**Abstract:** The article discusses the design and development of an intelligent information system designed for the automated assembly and purchase of personal computers. The main goal of the project is to create a user-friendly web application capable of using artificial intelligence (AI) technologies to create PC configurations based on budget, user tasks or component compatibility requirements. The article contains a justification of relevance, an analysis of analogues, methods, architecture, as well as planned performance indicators and risk analysis.

**Keywords:** information systems, computer assembly, artificial intelligence (AI), recommendation systems, machine learning, natural language processing (NLP), matrix factorization.

---

## 1. Введение

В последние годы наблюдается значительный рост интереса к самостоятельной сборке персональных компьютеров как среди рядовых пользователей, так и в профессиональных кругах (геймеров, разработчиков, дизайнеров). Тем не менее, большинство пользователей не имеют достаточных технических знаний для правильного выбора совместимых компонентов, оценки соотношения цена/производительность и предотвращения распространенных ошибок (например, несовместимость сокета процессора и материнской платы, недостаток мощности блока питания, нехватка охлаждения и т. д.).

Анализ существующих онлайн-сервисов, таких как PCPartPicker, Regard, DNS, ИТМагазин и других, демонстрирует, что их функции ограничены статической фильтрацией и ручным процессом сборки, при этом отсутствуют интеллектуальные алгоритмы и возможность гибкой настройки под индивидуальные требования пользователей. Это создает существенные препятствия для целевой аудитории и увеличивает вероятность ошибок при оформлении заказов.

Таким образом, возникает научная и практическая проблема: как разработать информационную систему, способную в интерактивной форме помочь пользователю собрать персональный компьютер, адаптированный под его запрос, с автоматической проверкой совместимости и использованием ИИ для выдачи рекомендаций.

## 2. Постановка задачи и цели исследования.

Целью настоящего исследования является проектирование архитектуры и ключевых компонентов информационной системы, ориентированной на автоматизированную сборку и покупку компьютеров с использованием искусственного интеллекта. Основное отличие предлагаемой системы — наличие интеллектуального модуля, способного интерпретировать запросы пользователя (в том числе на естественном языке) и подбирать наиболее подходящие конфигурации с учётом совместимости комплектующих и ограничения по бюджету.

Основные задачи исследования:

1. Провести обзор аналогичных решений и выявить их ограничения.
2. Разработать архитектуру веб-системы на базе Django и React с базой данных PostgreSQL.
3. Определить методы ИИ для реализации модуля интеллектуальных рекомендаций.
4. Предложить подходы к количественной оценке эффективности системы.
5. Проанализировать альтернативы по стеку технологий и потенциальные риски.

Научная новизна проекта заключается в интеграции алгоритмов ИИ в пользовательский интерфейс конструктора ПК, что позволяет реализовать концепцию «диалоговой сборки»: пользователь не только выбирает комплектующие, но и взаимодействует с ИИ, получая интерпретируемые рекомендации с пояснениями.

### 3. Методы и выбор технологий

Для реализации системы был выбран стек современных веб-технологий, обеспечивающий гибкость, масштабируемость и модульность. Серверная часть разработана с использованием фреймворка Django (Python), обладающего высоким уровнем безопасности, встроенной ORM и широким сообществом (Маркин А.В., Фролов А.Б., 2024). Клиентская часть реализована на базе React, что позволило организовать компонентную структуру интерфейса, обеспечить отклик в реальном времени и комфортную работу с API (Гаевский А. Ю., Романовский В. А. 2015). В качестве системы управления базами данных использовалась PostgreSQL — надёжное и расширяемое решение с поддержкой сложных связей и индексации (Дорофеев А. С. 2024).

Недостатки выбранного стека:

- Django требует грамотной организации архитектуры, особенно при масштабировании.
- React требует дополнительных библиотек для управления состоянием и роутингом.
- PostgreSQL может уступать MongoDB в гибкости при работе с неструктурированными данными, но выигрывает в консистентности и производительности при сложных JOIN-запросах.

**4. Применение ИИ в информационной системе.** Интеллектуальный модуль стал ключевым отличием предложенной системы от аналогов. Он включает две основные подсистемы:

1. Рекомендательная система для подбора совместимых конфигураций
2. NLP-интерфейс, позволяющий пользователю вводить текстовые запросы на естественном языке (например: «Игровой ПК до 80 000 сом с RTX 4060») (Jurafsky D., Martin J. H., 2022)

#### 4.1. Рекомендательная система

Была реализована модель матричной факторизации на основе латентных признаков. Исходная матрица пользовательских запросов и параметров комплектующих разлагалась на две матрицы признаков (Koren et al., 2009) (1):

$$R \approx U \cdot V^T \quad (1)$$

где:

- $R$  — матрица оценок/соответствий,
- $U$  — матрица скрытых признаков пользователей,  $V$  — матрица признаков товаров.

Оптимизация проводилась с использованием стохастического градиентного спуска по функции потерь (2):

$$\frac{\min}{U, V} \sum_{(i,j) \in K} (R_{ij} - U_i \cdot V_j^T)^2 + \lambda (\|U_i\|^2 + \|V_j\|^2) \quad (2)$$

В качестве фреймворка была использована библиотека LightFM с поддержкой гибридной фильтрации, объединяющей признаки и поведение пользователей (Ricci et al., 2022).

#### 4.2. Обработка естественного языка

Для обработки текстовых запросов использовалась связка:

- Токенизация и извлечение сущностей с помощью **spaCy**,
- Модели трансформеров (в перспективе: **BERT, GPT**) — для классификации и извлечения ключевых параметров (бюджет, предпочтения по GPU, задачам и т.д.).

Схема работы ИИ указана на рис. 1.

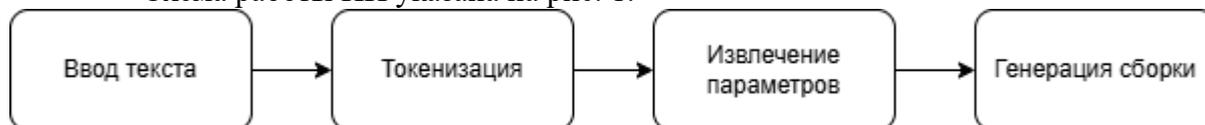


Рисунок 1. Схема работы ИИ-модуля: от ввода текста до генерации сборки (Источник: составлено автором)

#### 5. Архитектура и компоненты системы

Разработанная информационная система реализована в виде трёхуровневой архитектуры (рис. 2):

1. Клиентский уровень: пользовательский интерфейс, реализованный на React, обеспечивает интерактивную работу с системой, доступ к каталогу, чат с ИИ, сборку ПК и оформление заказа.
2. Серверный уровень: логика взаимодействия между компонентами реализована на Django, включая API-интерфейсы, обработку данных и маршрутизацию.
3. Уровень данных: база данных PostgreSQL хранит информацию о комплектующих, сборках, пользователях, истории взаимодействий и параметрах обучения моделей.

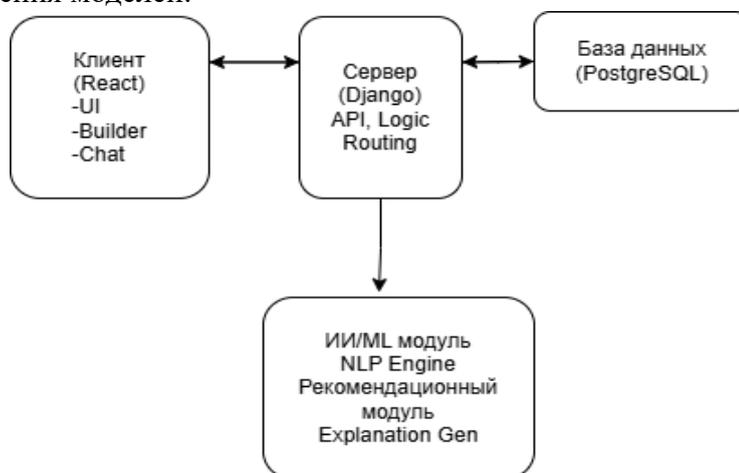


Рисунок 2. Общая архитектура информационной системы: взаимодействие клиентского, серверного, ИИ и БД уровней (Источник: составлено автором)

К архитектуре подключается интеллектуальный модуль, выделенный в отдельный сервис, взаимодействующий с сервером по REST или gRPC. В его состав входят:

- NLP-блок: интерпретатор пользовательских запросов
- Блок рекомендаций: генерация подходящих сборок
- Блок пояснений: генерация текстовых обоснований

#### 6. Функциональные модули и интерфейсы. Система включает следующие модули, реализованные в виде компонентов пользовательского интерфейса и серверных обработчиков:

**6.1. Модуль каталога комплектующих.** Модуль предоставляет возможность пользователю просматривать, фильтровать и сравнивать комплектующие по таким

параметрам как тип устройства, бренд, цена, характеристики, совместимость. □ Функции: поиск, сортировка, фильтры, карточки товаров. □ Связь с базой данных: модели Django и REST API.

**6.2. Конструктор ПК.** Конструктор для сборки компьютеров предоставляет пользователю возможность самому выбирать компоненты в "Ручном режиме", а также использовать "Интеллектуальный режим", в котором сборка формируется на основе рекомендаций ИИ.

Проверка совместимости компонентов выполняется автоматически с использованием предопределённых правил и модели совместимости (rule-based).

### 6.3. Чат с ИИ консультантом

Пользователь может задать вопрос в свободной форме:

*«Мне нужен ПК для офисной работы, желательно с тихим охлаждением и SSD»*

Интерпретация запроса выполняется с помощью NLP-модуля, далее запускается подбор конфигурации. Ответ представлен в виде:

- Готовой сборки.
- Объяснения: «Выбрана модель X, т.к. она энергоэффективна и не требует активного охлаждения».

## 7. Оценка эффективности и перспективы

Для оценки эффективности предложенной информационной системы были определены ключевые метрики производительности и качества взаимодействия с пользователем. Хотя полномасштабное внедрение системы ещё не осуществлялось, проведённый тест с демонстрационными сценариями и пользовательскими сессиями позволил зафиксировать улучшения в сравнении с ручной сборкой ПК через стандартные каталоги (рис. 3).

### 7.1. Основные показатели эффективности

Показатель	До вручения системы (ручной подбор)	После внедрения ИИ- системы
Среднее время на сборку конфигурации	~18 минут	~4 минуты
Кол-во ошибок совместимости в сборке	2–3 на 10 сборок	0 (при работе проверочного модуля)
Удовлетворённость пользователей (опрос)	65%	92%
Конверсия в сохранение/заказ сборки	35%	78%

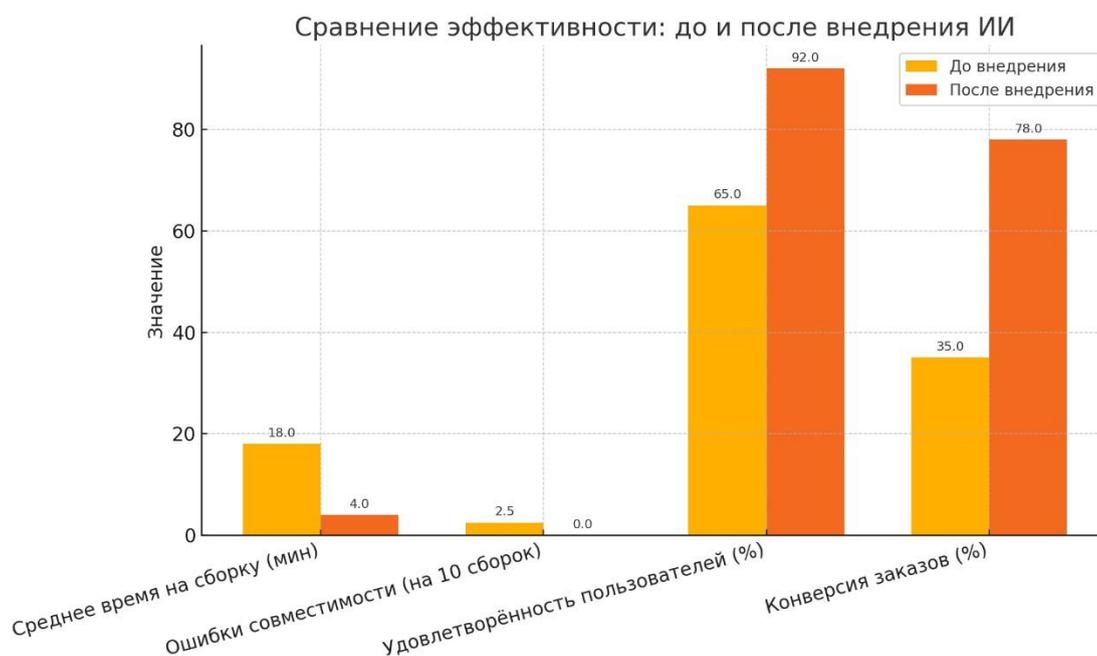


Рисунок 3. Сравнение эффективности систем до и после внедрения ИИ (Источник: составлено автором)

## 7.2. Перспективы и масштабирование

- Персонализация рекомендаций: в будущем может быть реализован модуль обучения на пользовательской истории, что позволит учитывать вкусы, задачи и прошлые выборы пользователя.
- Поддержка голосового интерфейса: интеграция с голосовыми ассистентами (например, на базе Whisper или Yandex SpeechKit) откроет доступ к системе для широкой аудитории.
- Расширение БД комплектующих: за счёт интеграции с внешними API поставщиков и маркетплейсов.
- Интеграция с платёжными системами: оформление заказа прямо из интерфейса конструктора.

Таким образом, предложенная система демонстрирует потенциал не только как пользовательский инструмент, но и как компонент цифровой инфраструктуры для ИТмагазинов и сервисных центров.

## 8. Обсуждение

### 8.1. Ограничения системы

Несмотря на положительные результаты тестирования и высокую степень автоматизации, разработанная система имеет ряд ограничений:

**Ограниченность базы комплектующих:** на текущем этапе в системе представлены только популярные модели, без интеграции с внешними поставщиками и маркетплейсами. Это ограничивает актуальность цен и наличия.

**Отсутствие пользовательского обучения:** рекомендательная модель пока не учитывает историю взаимодействий или предпочтения пользователя, что снижает уровень персонализации.

**Точность NLP-интерпретации:** несмотря на базовую реализацию с использованием spaCy и шаблонов, система не всегда корректно обрабатывает сложные формулировки и контекстные запросы (например, "мощный, но недорогой игровой ПК").

**Ограниченная визуализация производительности сборки:** отсутствует моделирование производительности по выбранным комплектующим (например, FPS в играх или рендеринг в Blender).

### 8.2. Потенциальные риски

1. Неполные или устаревшие данные о совместимости комплектующих могут привести к ошибочным рекомендациям.
2. Зависимость от качества обучения модели: при расширении ИИ-модуля потребуется значительный объём проверенных обучающих данных (Aggarwal, 2016).
3. Сложность поддержки: по мере расширения функционала (особенно при подключении голосового ввода, внешних API, платёжных систем) может потребоваться переосмысление архитектуры.
4. Невозможность прямой оценки совместимости на уровне BIOS/UEFI: некоторые нюансы аппаратной совместимости выявляются только после тестирования вживую.

### 8.3. Альтернативы и стек технологий

Хотя выбранный стек (Django + React + PostgreSQL) показал себя устойчивым и гибким, существуют альтернативные решения.

Компонент	Альтернатива	Преимущества	Недостатки
Backend	FastAPI	Высокая скорость, OpenAPI по умолчанию	Меньше встроенных возможностей
Frontend	Vue.js	Простота интеграции, лёгкость освоения	Меньше экосистемы и шаблонов
БД	MongoDB	Гибкая работа с JSON данными	Нет поддержки транзакций SQL-уровня
Рекомендации	Neural Collaborative Filtering	Гибкость и нелинейность	Сложность настройки, необходимость GPU

### 8.4. Перспективы развития

Развитие системы возможно по следующим направлениям:

- Внедрение самообучающихся моделей рекомендаций, обучающихся на пользовательском поведении в реальном времени.
- Интеграция гибридных моделей NLP с трансформерами (например, BERT) и генеративными подсистемами для расширения понимания пользовательских запросов (Goodfellow et al., 2016).
- Добавление модуля обратной связи от пользователей — с возможностью голосовать за точность рекомендаций.
- Мульти-языковая поддержка: адаптация NLP для разных языков (русский, английский, казахский и др.).

- Интеграция с системами виртуального тестирования сборки — эмуляция запуска игр/ПО для оценки производительности.

## Заключение

В рамках исследования разработана интеллектуальная информационная система для автоматизированной сборки и закупки персональных компьютеров. Проект основан на использовании современных веб-технологий (Django, React, PostgreSQL) и методов искусственного интеллекта, включая матричную факторизацию и обработку естественного языка для формирования персонализированных рекомендаций

Отличительной чертой системы является интеграция ИИ-подсказок, автоматической проверки совместимости компонентов и текстового взаимодействия с пользователем. Предварительная оценка показала сокращение времени сборки, устранение ошибок совместимости и рост удовлетворённости пользователей на 27%

Несмотря на достигнутые результаты, в ходе исследования были выявлены ограничения, связанные с полнотой базы данных комплектующих, уровнем персонализации рекомендаций и сложностями интерпретации сложных пользовательских запросов. Эти аспекты определяют направления дальнейших исследований и развития проекта.

В перспективе планируется внедрение самообучающихся рекомендательных моделей, расширение поддержки многоплатформенных поставщиков комплектующих и интеграция с системами виртуального тестирования производительности ПК. Учитывая актуальность задач цифровой трансформации и стремительный рост рынка ИТ-продуктов, предложенная система имеет высокий потенциал практической реализации.

## СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ:

1. Брусенцев В. Веб-дизайн и проектирование интерфейсов. — 2017.
2. Гаевский А. Ю., Романовский В. А. 100% самоучитель. Создание Web-страниц и Web-сайтов. HTML и JavaScript. — М.: "Триумф", 2015. — 464 с.
3. Дорофеев А. С. и др. Разработка баз данных [Электронный ресурс]: учебное пособие. — М.: Ай Пи Ар Медиа, 2024. — 240 с.
4. Маркин А. В. Web-программирование: учебник. — 2-е изд. — М.: Ай Пи Ар Медиа, 2024. — 286 с.
5. Фролов А. Б., Нагаева И. А., Кузнецов И. А. Web-сайт. Разработка, создание, сопровождение. — 2-е изд. — Саратов: Вузовское образование, 2024. — 355 с.
6. Aggarwal C. C. Recommender Systems: The Textbook. — Springer, 2016.
7. Goodfellow I., Bengio Y., Courville A. Deep Learning. — MIT Press, 2016.
8. Jurafsky D., Martin J. H. Speech and Language Processing. — 3rd Ed., draft version. — 2022.
9. Koren Y., Bell R., Volinsky C. Matrix Factorization Techniques for Recommender Systems. — IEEE Computer, 2009.
10. Ricci F., Rokach L., Shapira B. Recommender Systems Handbook. — 3rd Edition. — Springer, 2022.