

КАФЕДРА ИУ-7 «Программное обеспечение ЭВМ и информационные технологии»

Метод первичной обработки пользовательского поведения на веб-странице с использованием карты кликов

2025 год

УТВЕРЖДАЮ

Заведующий кафедрой ИУ7

И. В. Рудаков

23.01.2025 г.

ЗАДАНИЕ на выполнение выпускной квалификационной работы

Студент группы ИУ7-73Б

Дьяченко Артём Александрович

Тема выпускной квалификационной работы

**Метод первичной обработки пользовательского поведения на веб-странице с
использованием карты кликов**

При выполнении ВКР:

	Используются / Не используются	Да/Нет
1)	Литературные источники и документы, имеющие гриф секретности	Нет
2)	Литературные источники и документы, имеющие пометку «Для служебного пользования», иные пометки, запрещающие открытое опубликование	Нет
3)	Служебные материалы других организаций	Нет
4)	Результаты НИР (ОКР), выполняемой в МГТУ им. Н.Э. Баумана	Нет
5)	Материалы по незавершенным исследованиям или материалы по завершенным исследованиям, но ещё не опубликованные в открытой печати	Нет

Тема квалификационной работы утверждена распоряжением по факультету:	
Название факультета:	«Информатика и системы управления», ИУ
Дата и рег. номер распоряжения:	№ 03.03-03/1591 от 06.12.2024 г.

Часть 1. Аналитический раздел

Провести анализ предметной области обработки пользовательского поведения, в частности — на веб-страницах. Ввести ключевые понятия работы. Рассмотреть существующие методы сбора и анализа данных о взаимодействии пользователей с веб-страницами, включая тепловые карты. Выполнить сравнительный анализ существующих решений сбора и интерпретации данных о пользовательском поведении. Формализовать постановку задачи в виде функциональной модели, описывающей процесс обработки данных о кликах пользователей.

Часть 2. Конструкторский раздел

Разработать метод первичной обработки пользовательского поведения на основе карты кликов. Описать особенности предлагаемого метода. Определить структуры данных для хранения и обработки информации о кликах. Представить ключевые этапы метода в виде диаграмм IDEF0 или схем алгоритмов.

Часть 3. Технологический раздел

Обосновать выбор технологий и программных средств для реализации метода. Описать архитектуру разрабатываемого программного обеспечения, включая модули сбора, хранения и анализа данных. Разработать прототип системы, реализующей предложенный метод. Описать форматы входных (сырые данные кликов) и выходных (агрегированные карты кликов, статистика) данных. Провести тестирование системы.

Часть 4. Исследовательский раздел

Ввести критерии и провести исследование эффективности разработанного метода. Сравнить точность и скорость обработки данных с существующими аналогами.

Оформление квалификационной работы:

Расчетно-пояснительная записка на 50-80 листах формата А4.

Перечень графического (иллюстративного) материала (чертежи, слайды и т.п.):

Презентация к выпускной квалификационной работе на 12-20 слайдах.

Дата выдачи задания: «23» января 2025 года.

В соответствии с учебным планом выпускную квалификационную работу выполнить в полном объеме в срок до «28» мая 2025 года.

Руководитель ВКР	_____	<u>Гончаров С. С.</u>	<u>23.01.2025 г.</u>
Студент	_____	<u>Дьяченко А. А.</u>	<u>23.01.2025 г.</u>

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИУ
КАФЕДРА ИУ7
ГРУППА ИУ7-73Б

УТВЕРЖДАЮ

Заведующий кафедрой ИУ7
(индекс)
И. В. Рудаков
14.01.2025 г.
(подпись) (дата)

КАЛЕНДАРНЫЙ ПЛАН выполнения выпускной квалификационной работы

Студента Дьяченко Артёма Александровича

Тема квалификационной работы: Метод первичной обработки пользовательского поведения на веб-странице с использованием карты кликов

№ п.п.	Наименование этапов выпускной квалификационной работы	Сроки выполнения этапов		Отметка о выполнении	
		План	Факт	Должность	Подпись
	Задание на выполнение работы. Формулирование проблемы, цели и задач работы	23.01.2025 г.	23.01.2025 г.	Руководитель ВКР	Гончаров С. С.
	1 часть. Аналитический раздел			Руководитель ВКР	Гончаров С. С.
	Утверждение окончательных формулировок решаемой проблемы, цели работы и перечня задач			Заведующий кафедрой	Рудаков И. В.
	2 часть. Конструкторский раздел			Руководитель ВКР	Гончаров С. С.
	3 часть. Технологический раздел			Руководитель ВКР	Гончаров С. С.
	4 часть. Исследовательский раздел			Руководитель ВКР	Гончаров С. С.
	1-я редакция работы			Руководитель ВКР	Гончаров С. С.
	Подготовка доклада и презентации			Руководитель ВКР	Гончаров С. С.
	Отзыв руководителя			Руководитель ВКР	Гончаров С. С.
	Нормоконтроль			Нормоконтролер	
	Внешняя рецензия			Секретарь ГЭК	
	Защита работы на ГЭК			Секретарь ГЭК	

Руководитель ВКР

Гончаров С. С.

14.01.2025 г.

Студент

Дьяченко А. А.

14.01.2025 г.

РЕФЕРАТ

Расчетно-пояснительная записка 63 с., 3 табл., 36 ист., 1 прил.

Пользовательское поведение, Цифровая кафедра, LMS, системы электронного обучения, воспроизведение сеансов пользователей, тепловые карты, cookie, счётчики, журналирование действий, карты кликов.

Цель работы — выявление методов первичного анализа поведения студентов, обучающихся на базе системы электронного обучения ЦК МГТУ им. Н.Э. Баумана.

В процессе работы выполнен анализ предметной области, описаны решения и методы сбора данных для идентификации поведения пользователей, выполнен обзор методов анализа данных пользовательского поведения в цифровой среде. Разработан, реализован и протестирован метод сбора и первичной обработки информации о пользовательском поведении на основе карты кликов.

Содержание

ВВЕДЕНИЕ	8
1 Аналитический раздел	14
1.1 Методы анализа пользовательского поведения	14
1.1.1 Наблюдение	14
1.1.2 Опросы и интервью	14
1.1.3 Эксперимент	16
1.1.4 Анализ документов	16
1.2 Сбор и интерпретация данных	17
1.2.1 Методы сбора данных	17
1.2.2 Интерпретация данных	19
1.3 Анализ ограничений существующих решений	21
1.3.1 Требование публичного домена	21
1.3.2 Обработка данных на стороне третьих лиц	21
1.3.3 Ограниченность контроля и модифицируемости	21
1.3.4 Невозможность оптимизации производительности	21
2 Конструкторский раздел	23
2.1 Модули разрабатываемого ПО	24
2.2 Требования к разрабатываемому методу	27
2.3 Формат входных и выходных данных	27
2.4 Обработка данных	29
2.5 Особенности разрабатываемого метода	30
2.6 Сценарий применения метода	30
3 Технологический раздел	32
3.1 Обоснование выбора технологий и программных средств . .	32
3.2 Архитектура разрабатываемого программного обеспечения .	33
3.3 Сбор информации о кликах	36
3.4 Приём и сохранение данных	37
3.5 Обработка данных	38
3.6 Снятие снимков с помощью вебдрайвера	40
3.7 Генерация тепловых карт кликов	42

3.8	Графический интерфейс просмотра тепловых карт	43
3.9	Форматы входных и выходных данных	45
3.9.1	Проведение тестирования системы	46
3.10	Примеры работы	49
4	Исследовательский раздел	52
4.1	Предмет исследования	52
4.2	Характеристики вычислительной машины	52
4.3	Результаты исследования	52
4.3.1	Производительность метода	52
4.3.2	Средние значения параметров при различных объёмах данных	53
4.3.3	Графики зависимости производительности и использования ресурсов	54
4.4	Инструменты мониторинга и профилирования	57
4.5	Анализ вычислительной сложности алгоритма	57
4.6	Выводы	57
	ЗАКЛЮЧЕНИЕ	59
	СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	60
	ПРИЛОЖЕНИЕ А	63

ВВЕДЕНИЕ

В связи с развитием информационных технологий все больше образовательных программ создаются на базе систем электронного (также называемого цифрового) обучения (LMS — англ. Learning Management System) — программных приложений, используемых для помощи в учебном процессе при электронном обучении [1].

Система (с др.-греч. — целое, составленное из частей; соединение) — множество элементов, находящихся в отношениях и связях друг с другом, которое образует определенную целостность, единство [2].

LMS в общем случае может представлять собой некоторый веб-ресурс, онлайн портал, который предоставляет возможность обмена учебными материалами и проведением занятий. Он зачастую имеет функционал, который позволяет общаться обучающимся вне класса, ведя обсуждения через форумы, которые в противном случае могли бы занять слишком много времени, предполагаемого для очного обучения [3].

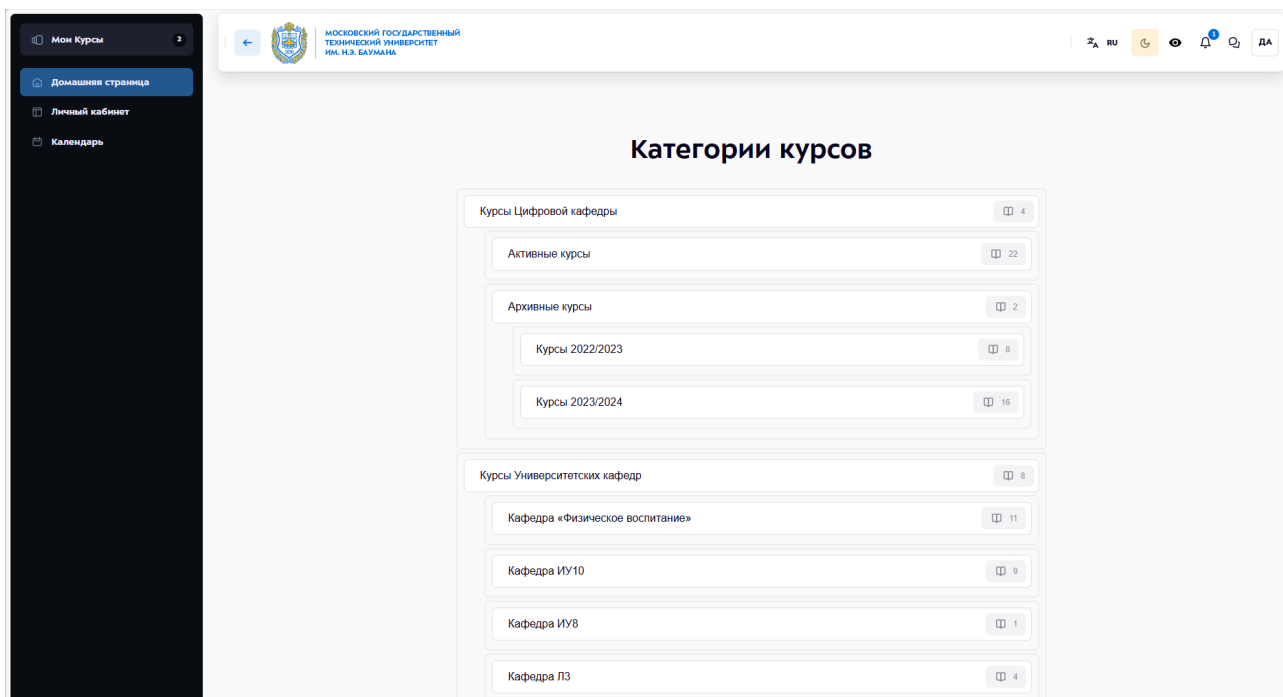


Рисунок 1 – Веб-сервис LMS BMSTU

Большинство учебных заведений в некоторой мере используют системы электронного обучения. Этому способствовала пандемия коронавируса и последующие запреты на проведение массовых мероприятий — в результате этого в школах и университетах обучение перешло в онлайн-формат, но даже

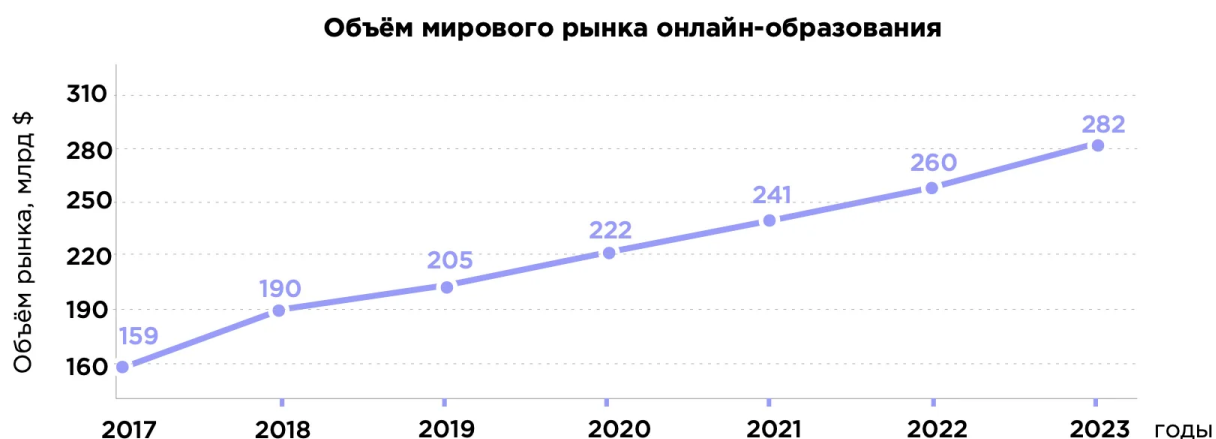
после снятия ограничений часть материалов время от времени преподается дистанционно.

Популярность электронного образования в том числе можно отследить по объёму инвестиций. Так, в 2019-ом году наблюдался закономерный скачок вложенных средств.



Рисунок 2 – Инвестиции в российский рынок онлайн образования [4]

А рынок электронного обучения на протяжении нескольких лет показывает стабильный рост:



Для годов с 2019 по 2023-ий даны расчётные значения при среднегодовом приросте на уровне 8,2%.
[EdTechXGlobal, Global Market Insights, Education International, экспертные оценки]

Рисунок 3 – Объём мирового рынка электронного обучения [4]

Цифровизация образовательных процессов является одним из методов

повышения эффективности передачи знаний, о чём указано в статьях [5—10]. Она также входит в ключевые направления развития высшего образования, правовой основой для которых являются:

- федеральный Закон от 29.12.2012 г. № 273—ФЗ «Об образовании в Российской Федерации», который закрепил право образовательных организаций на применение в их деятельности различных цифровых образовательных технологий;
- национальный проект «Цифровая экономика Российской Федерации». Согласно Указу Президента Российской Федерации к 2024 году должна быть создана цифровая образовательная среда, обеспечивающая качество и доступность образования;
- государственная программа Российской Федерации «Развитие образования» на 2018–2025 годы. Она ставит перед собой следующие цели: высокое качество, доступность образования, применение дистанционных технологий. В рамках данной программы предполагается реализация нескольких проектов: «Вузы как центры пространства создания новаций», «Создание современной образовательной среды для школьников» и т.п.

В рамках имплементации такого подхода системы электронного обучения находят свое применение и в рамках отечественных университетов. Так, реализуемые в рамках национального проекта «цифровая экономика» направления Цифровой кафедры МГТУ им. Н.Э. Баумана [11] (далее — ЦК) проводятся на одной из подобных площадок.

Потребитель (англ. — consumer) — индивидуум или группа, использующие товар, услугу, идею для удовлетворения своих потребностей [12].

В контексте систем электронного обучения (LMS) потребителем является человек, проходящий обучение на платформе. В случае обучения на ЦК таковым является студент. Поскольку упомянутые образовательные услуги предоставляются в цифровом виде на электронных ресурсах, которые в большинстве своём являются веб-приложениями, студент также является пользователем этих приложений. Поэтому в данной работе понятия пользователя и потребителя подразумеваются тождественными.

Согласно словарю Т.Ф. Ефремовой поведение представляет собой совокупность поступков и действий по отношению к окружающим [13; 14].

Социологический словарь определяет потребление как использование, употребление продукции, товаров, благ и услуг в процессе удовлетворения потребностей [14; 15].

В настоящее время понятие «потребительское поведение» используется для определения:

- действий людей по приобретению, потреблению товаров и услуг и избавлению от них;
- деятельности, непосредственно вовлеченной в обретение, потребление и избавление от продуктов, услуг, идей, включая процессы решений, предшествующие этой деятельности и следующие за ней;
- единства процесса принятия потребителем решения о покупке и факторов, определяющих его характер и направленность;
- процесса формирования рыночного спроса покупателей, осуществляющих выбор благ с учётом существующих цен;
- совокупности признаков и показателей, характеризующих действия потребителей, включая их потребительские предпочтения, спрос на товары и услуги, структуру потребления, способы использования доходов.

Существует множество определений пользовательского поведения, вот некоторые из них:

- действие, имеющее непосредственное отношение к получению, потреблению и распоряжению продуктами и услугами, включая процессы принятия решений, которые предшествуют этим действиям и следуют за ними;
- более или менее повторяющиеся решения по поводу выбора определенной торговой марки товара;
- деятельность, непосредственно вовлеченная в обретение и избавление от продуктов, услуг, идей, включая процессы решений, предшествующие этой деятельности и следующие за ней;

- ситуационные переменные, как физическая и социальная среда, настроение человека и время совершения покупки или использования товара;
- эмпирический процесс, в котором пользователи товаров находят интерес, посредством которого выражают свои чувства радости и удовольствия;
- процесс осознания потребителем необходимости получения определенного блага под воздействием внутренних и внешних факторов, в результате которого он делает определённый выбор.

Большинство авторов рассматривают понятие «потребительское поведение» как совокупность действий, реализуемых в сфере потребления. В работе Е.А. Лысова [14] приводится анализ термина «потребительское поведение», который приводит к следующей формулировке: «потребительское поведение – это действия потребителя на рынке конкретного товара или услуги, которые заключаются в принятии решения на совершение покупки на основе экономических выгод, под влиянием личного опыта и мнений его социального окружения, а также с учетом его психологических особенностей».

В текущей работе используется данное Е.А. Лысовой определение с поправкой на контекст образовательной деятельности ЦК и включение в него понятий потребления и отказа.

Таким образом, в настоящей работе под потребительским поведением подразумеваются действия, связанные с принятием решений относительно выбора, потребления или отказа от образовательных услуг на базе систем электронного обучения.

Актуальность работы обуславливается развитием и эффективностью электронного обучения, примером реализации которого является «Цифровая кафедра» МГТУ им. Н.Э. Баумана.

С помощью сбора и анализа цифровых следов пользователей платформы и последующей минимизации факторов, негативно влияющих на освоение образовательных программ, можно добиться повышения качества электронного обучения.

Цель работы — выявление методов первичного анализа поведения студентов, обучающихся на базе системы электронного обучения ЦК МГТУ им. Н.Э. Баумана.

Для выполнения работы были поставлены следующие задачи:

- анализ предметной области;
- разработка функциональной модели обработки данных о кликах и определение структур данных для их хранения;
- разработка предлагаемого метода;
- проектирование и реализация системы;
- тестирование и оценка эффективности;

1 Аналитический раздел

1.1 Методы анализа пользовательского поведения

Существуют три основных методологических подхода анализа пользовательского поведения: наблюдение, опросы и интервью, эксперимент. Стоит отметить, что некоторые авторы дополнительно выделяют другие подходы. Так, Александрова И.Ю. в работе «Методология маркетингового исследования интернет-пользователей» [16] рассматривает ещё один метод сбора первичных эмпирических данных — анализ документов.

1.1.1 Наблюдение

В работе Блэкуэлла, Роджера Д., Пола У. Миниарда, и Джеймса Ф. Энджела «Поведение потребителей» [12] говорится о том, что метод наблюдения состоит в отслеживании поведения потребителей в различных ситуациях. Исследователи могут прослеживать поведение потребителей в естественной обстановке, например то, как они пользуются продуктами или потребляют пищу у себя дома, но иногда они наблюдают за поведением в лабораторных условиях. Этот процесс может состоять в наблюдении с использованием исследовательских инструментов за реакцией потребителей на различные рекламные объявления, упаковку или цвета.

Авторы также выделяют некоторые реализации метода наблюдения:

- наблюдение в домашних условиях;
- экранирование или теневой метод;
- метод физиологического наблюдения.

В следующих главах будут рассмотрены приложения реализации метода наблюдения за пользовательским поведением с точки зрения цифровых пространств.

1.1.2 Опросы и интервью

Опрос — это метод сбора первичной информации на основе взаимодействия исследователя и опрашиваемого (респондента) [17].

Экспертный опрос — это разновидность опроса, в ходе которого респондентами являются эксперты — высококвалифицированные специалисты в определенной области деятельности [18].

Интервью – разновидность метода опроса, специальный вид целенаправленного общения с человеком или группой людей. В основе интервью лежит обычная беседа. Однако, в отличие от нее, роли собеседников закреплены, нормированы, а цели определяются замыслом и задачами проводимого исследования [19].

В источнике [12] опросы классифицируются как:

- персональные (личные, «перехваты»);
- телефонные;
- по почте;
- интернет-опросы.

При этом способ проведения опроса (онлайн или оффлайн) нельзя считать фактором, определяющим результаты исследования. В работе «Сравнение результатов онлайн-и оффлайн-опросов (на примере анкет разной сложности)» Некрасов С. И. отмечает, что «результаты грамотно составленного онлайн-опроса не отличаются от привычного бумажного».

В зависимости от целей интервью или опроса определяется выборка. Так, для «поверхностного» масштабного анализа пользователей подойдёт телефонный или почтовый опрос. В то же время для глубокого исследования специфических аспектов пользовательского поведения подходят фокус-группы [12].

Фокус-группа — это качественный метод исследования, который заключается в проведении группового глубинного интервью. Целью проведения фокус-группы является выяснение отношения участников к исследуемой в данной работе проблеме [20].

1.1.3 Эксперимент

Эксперимент — метод исследования, который заключается в попытке понять причинно-следственные связи путем манипуляций с независимыми переменными (например, набором рекламных объявлений, вариантов дизайна упаковки, методов коммуникации) с целью определить их влияние на зависимые переменные (такие как покупательское намерение или поведение) [12].

В работе «Поведение потребителей» [12] эксперименты классифицированы как:

- лабораторные, обеспечивающие максимальный контроль над изучаемыми переменными;
- полевые — т.е. имеющие место в естественных условиях, т.е. на дому или в магазине.

1.1.4 Анализ документов

Александрова И.Ю. в работе «Методология маркетингового исследования интернет-пользователей» [16] приводит следующее определение: анализ документов — метод социологического исследования, предполагающий качественный и/или количественный анализ различных информационных источников (в научно-исследовательской терминологии — документов), представленных в материальной или электронной форме.

Одной из основных методик анализа документов является анализ следов. Специфика данной методики заключается в том, что исследовательскому анализу подвергается вторичная информация, сформированная в результате обработки первичных данных. Если в качестве первичных данных в маркетинговом исследовании интернет-пользователей выступают различные результаты психологических и поведенческих реакций потребителей в веб-коммуникационном процессе, то есть собственно социальные факты, то под вторичными данными подразумевается аналитически обработанная информация об этих социальных фактах, зафиксированная в различных документах. В качестве таких документов исследователи могут использовать статистические отчеты веб-сайта, отчеты об инцидентах, отчеты об отказах от покупок в интернет-магазине, отчеты об эксплуатации веб-системы и т.п. [16].

1.2 Сбор и интерпретация данных

В этой главе рассматриваются методы и инструменты сбора данных и их первичной обработки.

Для последующей аналитической обработки данных необходимо определить способ сбора статистики. Выбор метода сбора данных является ключевым этапом в построении эффективной аналитической системы. От корректности и полноты собранных данных зависит точность и качество принимаемых решений на их основе.

1.2.1 Методы сбора данных

Обратившись к работе Прохоровой А.М. «Роль методов анализа и прогнозирования поведения пользователей на образовательном портале» [21] определим некоторые варианты:

- cookie;
- счётчики;
- лог-файлы (журналирование действий).

Методы сбора данных различаются по точности, сложности внедрения и другим важным характеристикам. Ниже приведены критерии, используемые для их оценки:

- точность данных — насколько полно и детально фиксируются пользовательские действия;
- сложность внедрения — какие ресурсы и навыки нужны для настройки метода;
- объем собираемых данных — количество и разнообразие информации, которую можно получить;
- конфиденциальность — обеспечение приватности пользователей;
- аналитические возможности — насколько эффективно данные используются для построения выводов.

Сравнение технологий cookie, лог-файлов и счетчиков по этим критериям представлено в таблице 1.1.

Таблица 1.1 – Сравнение технологий сбора

Критерий	Cookie	Счётчики	Лог-файлы
Точность данных	Зависят от браузеров, нельзя отслеживать одного пользователя на разных устройствах	Зависят от корректности роботов скриптов	Предоставляют данные о действиях на сервере
Сложность внедрения	Требуется настройка политики конфиденциальности и обработки cookie	Требуется разработка программного кода	Необходима настройка сервера для ведения журналов
Объем собираемых данных	Ограничен информацией, которая передается через браузер	Ограничен событиями, зафиксированными программным кодом	Все запросы к серверу, включая ошибки, хакерские атаки, скачиваемый контент
Конфиденциальность	Могут быть удалены или заблокированы пользователем	Зависит от настроек веб-сайта и наличия пользовательских согласий на сбор данных	Высокий уровень контроля, но требует обработки данных на стороне сервера
Аналитические возможности	Используются для персонализации контента, построения сценариев	Подходят для оперативного анализа посещаемости	Подходят для детального анализа: ошибки, атаки, пользовательские сценарии
Влияние на производительность	Обрабатываются браузером и почти не нагружают сервер	Зависит от частоты вызовов и объема данных	Может увеличивать нагрузку на сервер при большом объеме данных

1.2.2 Интерпретация данных

Полученные данные можно интерпретировать в нескольких форматах. Кушмар, С.Е. и И.И. Пилецкий [22] приводят тепловые карты (heat map) и воспроизведение сеанса пользователя (sessions replay). Эти методы различаются по своим характеристикам:

- цель анализа: какие задачи решает метод, например, поиск популярных элементов интерфейса или изучение последовательности действий пользователя;
- формат представления данных: как визуализируются данные для анализа (цветовые карты, видео, текстовые отчеты);
- точность анализа: уровень детализации предоставляемых данных;
- влияние на производительность: нагрузка на систему при использовании метода.

Сравнение видов тепловых карт по этим критериям представлено в таблице 1.2.

Таблица 1.2 – Сравнение тепловых карт и воспроизведения сеансов пользователей

Критерий	Тепловые карты	Воспроизведение сеансов пользователей
Цель анализа	Определение наиболее «кликабельных» элементов интерфейса, поведения пользователей на определенных частях страницы.	Реконструкция полного пути пользователя, анализ пошагового взаимодействия с интерфейсом.
Формат представления	Цветовая карта (от холодных до теплых оттенков), визуализация статистики кликов, переходов, скроллинга.	Видеозаписи действий пользователя, доступ к событиям (клики, ввод текста, перемещения курсора).
Точность анализа	Ограничена уровнем агрегации данных (тепловая карта дает усредненный результат).	Позволяет увидеть конкретные действия, но не подходит для анализа больших объемов данных.
Влияние на производительность	Низкое, так как собирают агрегированные данные без записи детальных действий в реальном времени.	Может создавать сильную нагрузку при сборе и воспроизведении большого количества сеансов.

1.3 Анализ ограничений существующих решений

Существующие сервисы для анализа пользовательского поведения, такие как Яндекс.Метрика [23] и Clarity [24], предоставляют широкие возможности для сбора и визуализации карты кликов. Однако их применение сопряжено с рядом ограничений, которые делают их непригодными для использования в случае локальных веб-ресурсов.

1.3.1 Требование публичного домена

Основным недостатком данных сервисов является необходимость размещения анализируемого сайта на публичном домене. Поскольку их функционал основан на удалённом сборе и обработке данных, локальные веб-приложения, работающие в закрытых сетях или без доступа к интернету, не могут быть проанализированы с их помощью. Это существенно сужает область применения подобных инструментов, исключая возможность их использования, например, во внутренних корпоративных системах или на этапе разработки и тестирования.

1.3.2 Обработка данных на стороне третьих лиц

Другим критическим аспектом является передача и хранение данных о пользовательских действиях на серверах внешних сервисов. Несмотря на наличие соглашений о конфиденциальности, некоторые организации или частные пользователи могут быть не заинтересованы в раскрытии подобной информации третьим лицам. Это особенно актуально для проектов, работающих с персональными или коммерчески чувствительными данными, где соблюдение требований информационной безопасности является приоритетом.

1.3.3 Ограниченность контроля и модифицируемости

Поскольку Яндекс.Метрика [23] и Clarity [24] являются проприетарными решениями с закрытым исходным кодом, пользователь не может самостоятельно модифицировать или расширять их функциональность. Это исключает возможность кастомизации интерфейса, добавления новых функций, интеграции с внутренними системами или изменения логики сбора данных под специфические требования проекта.

1.3.4 Невозможность оптимизации производительности

Также отсутствует возможность управления производительностью сервисов со стороны пользователя. В случае возникновения задержек, перегрузки или неэффективного использования ресурсов нельзя внести изменения в ар-

хитектуру системы или распределение нагрузки, так как серверная часть полностью контролируется поставщиком услуги. Это может стать узким местом при масштабировании или в условиях повышенных требований к скорости отклика и стабильности.

Вывод

На основании проведённого анализа сделан вывод, что для изучения поведения пользователей в цифровой среде наиболее рациональным является комбинированный подход, сочетающий методы наблюдения, опросов и анализа документов.

Среди инструментов сбора данных, рассмотренных в разделе, лог-файлы показали высокую точность, широкий охват информации и приемлемую сложность внедрения, тогда как тепловые карты, особенно в формате карты кликов, предоставляют наглядную визуализацию взаимодействия пользователей с интерфейсом, позволяя выявить ключевые зоны внимания.

В результате анализа критериев выбора, таких как точность, аналитические возможности и уровень влияния на производительность, в качестве основных средств были выбраны лог-файлы и карта кликов.

Ниже представлена IDEF0-диаграмма, формализующая процесс сбора и интерпретации пользовательских данных с использованием указанных инструментов.

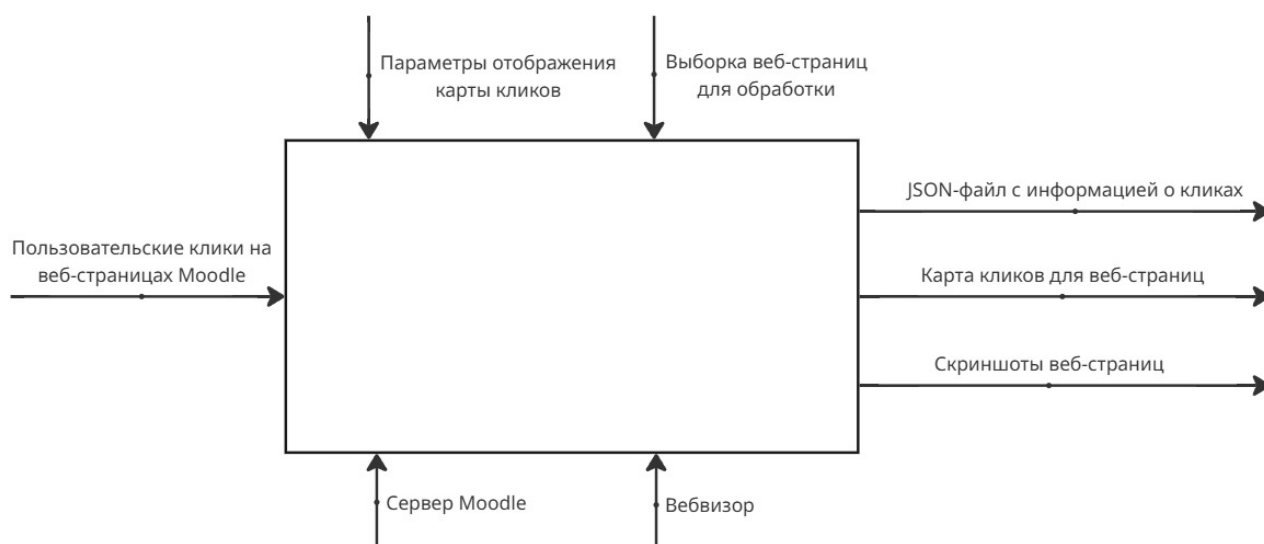


Рисунок 1.1 – Процесс сбора и интерпретации данных

2 Конструкторский раздел

В данном разделе

- рассмотрен разрабатываемый метод первичной обработки пользовательского поведения на основе карты кликов;
- описаны особенности предлагаемого метода;
- определены структуры данных для хранения и обработки информации о кликах;
- представлены ключевые этапы метода.

Анализ существующих решений для сбора и анализа пользовательского взаимодействия, таких как Яндекс.Метрика [23] и Microsoft Clarity [24], выявил ряд существенных ограничений:

- требование публичного домена для размещения анализируемого ресурса;
- передачу конфиденциальных данных на серверы третьих лиц;
- невозможность работы в локальных сетях без интернет-доступа.

Эти ограничения делают существующие сервисы неприменимыми для:

- некоторых корпоративных внутренних систем;
- веб-приложений на этапе разработки и тестирования;
- проектов с повышенными требованиями к информационной безопасности.

В рамках данной работы предлагается модульная система анализа пользовательского поведения, призванная учесть ограничения приведённых сервисов.

2.1 Модули разрабатываемого ПО

В рамках работы реализуется модульная архитектура системы первичной обработки пользовательского поведения на основе карты кликов.

Модуль – (от лат. *modulus* — мера) в широком смысле — любая отделяемая, относительно самостоятельная часть системы, устройства, организации [25].

Система включает следующие основные модули (рисунок 2.2):

- модуль сбора — отвечает за регистрацию событий кликов на страницах веб-приложения;
- модуль обработки — выполняет форматирование собранных данных, на их основе составляет карты кликов;
- модуль отображения — служит для навигации и просмотра созданных карт кликов.

Такая архитектура позволяет обеспечить конфиденциальность собранных данных и предоставить разработчикам возможность оптимизации разрабатываемого решения.

Верхнеуровневая схема приложения приведена на рисунке 2.1:

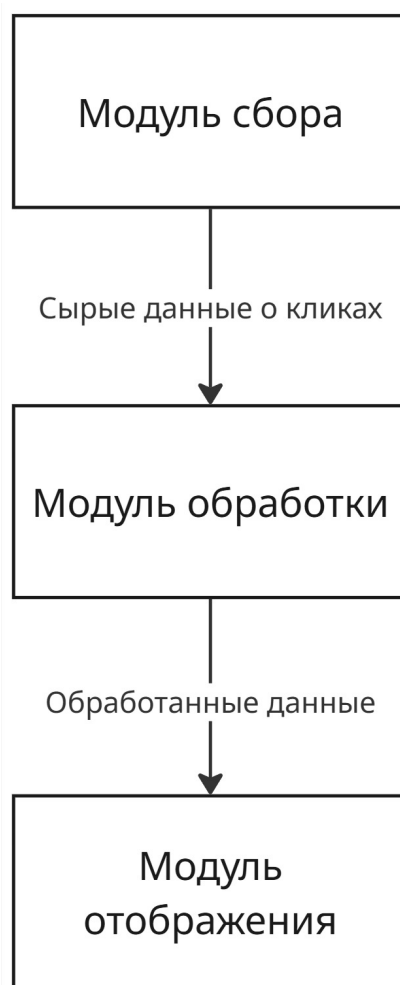


Рисунок 2.1 – Верхнеуровневая схема модулей приложения

Взаимодействие между модулями более подробно представлено на рисунке 2.2:

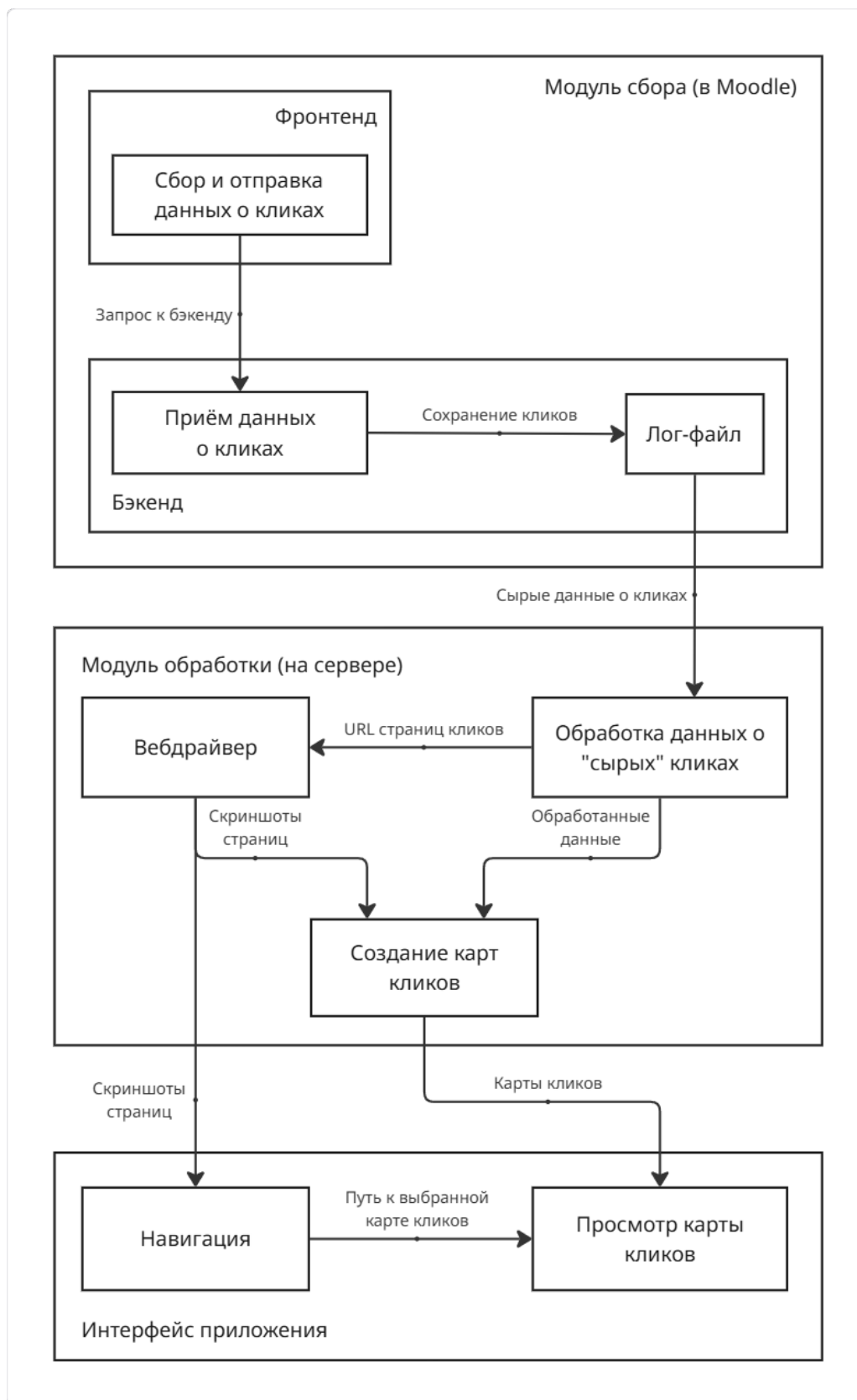


Рисунок 2.2 – Схема модулей разрабатываемого ПО

2.2 Требования к разрабатываемому методу

При разработке метода первичной обработки пользовательского поведения на основе карты кликов были сформулированы следующие требования:

- локальность — вся обработка данных может осуществляться на стороне клиента;
- изменяемость — возможность внесения пользователем изменений в систему;
- визуальная наглядность — результат должен быть представлен в виде карты кликов.

Такой подход позволяет анализировать взаимодействие пользователя с интерфейсом без привлечения сторонних сервисов.

2.3 Формат входных и выходных данных

Модуль сбора данных реализуется с использованием JavaScript-скрипта, внедрённого в код Moodle с помощью функции "Дополнительного HTML" в настройках администрирования.

При каждом клике пользователя фиксируются следующие параметры:

- относительные координаты клика **relX**, **relY** — безразмерные значения от 0 до 1, характеризующие положение клика относительно ширины и высоты всего веб-сайта;
- абсолютные координаты **absX**, **absY** — пиксельные координаты относительно начала страницы;
- URL текущей страницы **url**;
- заголовок страницы **title**;
- ширина и высота страницы в пикселях **pageWidth**, **pageHeight**.

Сформированные данные отправляются на сервер с использованием HTTP-запроса и сохраняются в лог-файл для каждого клика в следующем виде:

Array

```
(  
  [relX] => 0.025913621262458  
  [relY] => 0.012054244098443  
  [absX] => 39  
  [absY] => 24  
  [url] => http://192.168.1.102/moodle/admin  
  [title] => Appearance | Administration  
  [pageWidth] => 1505  
  [pageHeight] => 1991  
)
```

Модуль обработки данных конвертирует каждую запись в лог-файле в формат JSON для дальнейшей работы:

```
{  
  "relX": 0.025913621262458,  
  "relY": 0.012054244098443,  
  "absX": 39,  
  "absY": 24,  
  "url": "http://192.168.1.102/moodle/admin",  
  "title": "Appearance | Administration",  
  "pageWidth": 1505,  
  "pageHeight": 1991  
}
```

Каждой тепловой карте соответствует URL-адрес страницы, запись хотя бы об одном клике по которой есть в лог-файле.

Выходными данными модуля визуализации являются изображения экранов соответствующих страниц с нанесёнными точками кликов, формирующие тепловые карты.

2.4 Обработка данных

Обработка данных выполняется в несколько этапов, реализуемых модулем обработки:

- 1) чтение лог-файла, содержащего записи о кликах в виде массивов;
- 2) преобразование каждой записи в формат JSON с сохранением всех ключевых параметров: координат, URL, заголовка страницы и размеров;
- 3) с помощью браузерного вебдрайвера (автоматизированного средства управления браузером, например Selenium [26]) выполняется автоматическое открытие каждой уникальной страницы;
- 4) производится снимок каждой страницы;
- 5) на снимок наносятся визуальные маркеры кликов в соответствии с координатами;
- 6) сгенерированные изображения сохраняются в виде тепловых карт и ассоциируются (посредством наименования) с соответствующими URL.

Модуль отображения (пользовательский интерфейс приложения) организует доступ к сформированным тепловым картам с помощью древовидной структуры URL-адресов, позволяющей выбрать интересующую страницу и отобразить соответствующую карту кликов.

Такая обработка позволяет формировать наглядное представление о пользовательской активности на различных страницах веб-ресурса.

2.5 Особенности разрабатываемого метода

Предлагаемый метод обладает следующими особенностями:

- автономность работы — возможность функционирования без интернет-соединения;
- локальная обработка данных без передачи информации третьим лицам;
- применимость к локальным веб-приложениям и тестовым средам;
- модульность архитектуры, позволяющая использовать компоненты отдельно;
- открытая реализация — возможность модификации и встраивания;
- гибкость — адаптируемость под различные типы веб-приложений.

2.6 Сценарий применения метода

Рассмотрим типичный сценарий использования системы анализа пользовательского поведения:

- пользователь взаимодействует с веб-страницей, клики фиксируются с помощью JS-обработчика;
- события отправляются на локальный сервер и сохраняются в лог-файле;
- администратор запускает обработку данных на сервере;
- генерируется карта кликов;
- дизайнер анализирует карту и принимает решения о корректировке интерфейса.

Выводы

В данном разделе были разработаны и обоснованы ключевые компоненты системы первичной обработки пользовательского поведения на основе карты кликов.

Рассмотрена архитектура, включающая модули сбора, обработки и отображения данных, обеспечивающая автономность, безопасность и гибкость решения.

Определены требования к системе, формат входных и выходных данных, а также последовательность этапов обработки информации.

Особое внимание уделено возможности локальной работы без передачи данных третьим лицам, что повышает конфиденциальность и расширяет сферу применения метода.

Предложенный подход обеспечивает наглядное представление пользовательской активности и служит основой для дальнейшего анализа и улучшения интерфейсов веб-страниц.

3 Технологический раздел

3.1 Обоснование выбора технологий и программных средств

Для реализации системы на платформе были выбраны следующие технологии и средства:

- серверная платформа: Ubuntu 22.04.5 LTS [27] — стабильная операционная система с широким сообществом и поддержкой серверных решений;
- веб-платформа: Moodle 5.0+ [28] — современная бесплатная система управления обучением с открытым исходным кодом, которая используется МГТУ Н.Э. Баумана;
- интерпретатор серверного кода: PHP 8.2 [29] — совместим с последними версиями Moodle, обеспечивает приём данных от клиентского скрипта;
- сервер приложений: Apache/2.4.52 [30] — надёжный HTTP-сервер, часто используемый в связке с Moodle;
- система управления базами данных: PostgreSQL 14.17 [31] — обеспечивает хранение структурированных логов пользовательской активности;
- язык программирования: Python 3.11 [32] — используется для разработки модуля обработки и пользовательского интерфейса;
- язык клиентской логики: JavaScript [33] — необходим для сбора данных о действиях пользователя на веб-странице;
- среда разработки: PyCharm 2025.1.11 [34] — используется для написания и отладки Python-кода, позволяет удалённо подключаться к серверу и редактировать на нём файлы.

Такой стек технологий выбран на основе требований к функциональности, расширяемости и совместимости компонентов системы.

3.2 Архитектура разрабатываемого программного обеспечения

Ранее в конструкторском разделе уже была приведена архитектура разработанного ПО. На самом деле, её можно представить в нескольких видах. Например, архитектуру можно поделить на следующие функциональные блоки:

- модуль сбора — реализуется на клиентской стороне и осуществляет запись координат кликов;
- модуль хранения — обрабатывает поступающие с клиента данные и сохраняет их в лог-файл;
- модуль анализа — агрегирует и структурирует данные, готовит их к визуализации.

Графически архитектура будет выглядеть так:

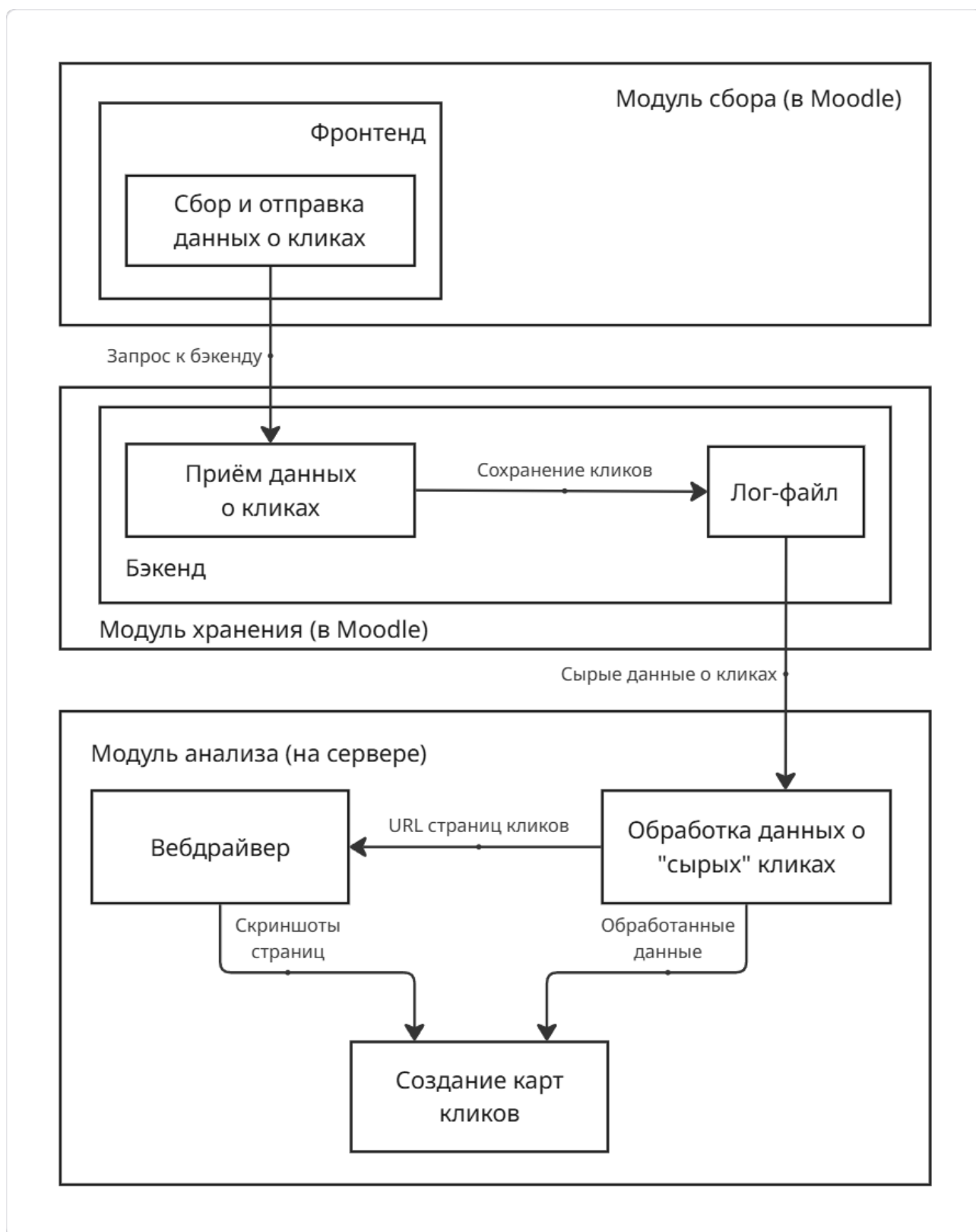


Рисунок 3.1 – Другое представление архитектуры ПО

Для более лёгкого сравнения обоих вариантов архитектуры отобразим их на схеме 3.2.

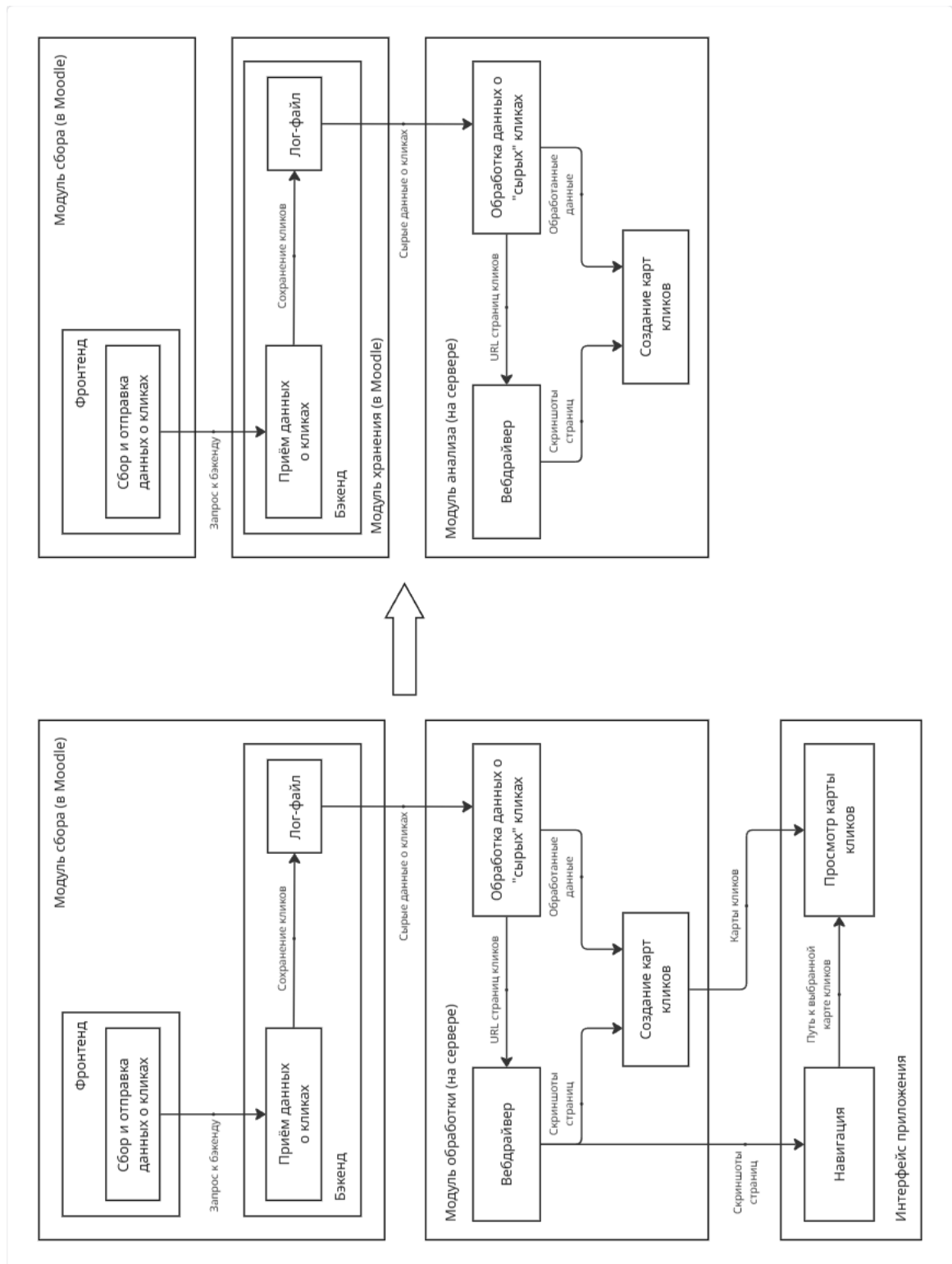


Рисунок 3.2 – Сравнение двух представлений архитектуры

Как видно на схеме, интерфейс во втором представлении отсутствует в качестве модуля. Это потому, что интерфейс не выполняет самостоятельных вычислительных и функций хранения, а лишь предоставляет доступ к функциональности других модулей. С архитектурной точки зрения он представляет собой надстройку над модулем анализа и не входит в состав логической обработки данных.

Тем не менее, для реализации метода в рамках данной работы пользовательский интерфейс является необходимым компонентом, так как обеспечивает:

- выбор файлов для анализа;
- запуск модулей обработки и анализа;
- отображение результатов анализа в наглядной форме.

Таким образом, обе представленные схемы архитектуры являются эквивалентными с функциональной точки зрения и различаются лишь по критерию группировки компонентов.

3.3 Сбор информации о кликах

Модуль сбора информации реализован на языке JavaScript и встраивается в HTML-код страницы, открываемой пользователем. Скрипт обрабатывает событие `click` и извлекает координаты клика относительно всей страницы, а также заголовок и адрес текущей страницы. Для последующей агрегации координаты нормализуются по ширине и высоте страницы, чтобы затем отображать их на общей тепловой карте независимо от разрешения экрана.

Листинг 3.1 – Регистрация события клика на странице

```
1 document.addEventListener('DOMContentLoaded', function() {  
2     document.addEventListener('click', function(event) {  
3         const scrollWidth = document.documentElement.scrollWidth;  
4         const scrollHeight =  
5             document.documentElement.scrollHeight;  
6         // Рассчитываем нормализованные координаты (0—1)  
7         const relX = event.pageX / scrollWidth;  
8         const relY = event.pageY / scrollHeight;
```

Полученные данные сериализуются в формат JSON и передаются на сервер с помощью HTTP-запроса методом POST. Передача осуществляется по

адресу `/moodle/local/receiver.php`, где они сохраняются для последующей обработки.

Листинг 3.2 – Отправка данных о клике на сервер

```
1      const eventData = {
2          relX: relX ,
3          relY: relY ,
4          absX: event.clientX ,
5          absY: event.clientY ,
6          url: window.location.href ,
7          title: document.title
8      };
9
10     fetch( '/moodle/local/receiver.php' , {
11         method: 'POST' ,
12         headers: { 'Content-Type': 'application/json' },
13         body: JSON.stringify(eventData)
14     })
15     .then(response => {
16         if (!response.ok) throw new Error('Ошибка HTTP: ' +
17             response.status);
18         return response.text();
19     })
20     .then(data => console.log('Ответ сервера:', data))
21     .catch(error => console.error('Ошибка:', error));
22 });
```

3.4 Приём и сохранение данных

Полученные от клиента данные о кликах обрабатываются серверным скриптом, написанным на языке PHP. Этот скрипт размещается по адресу `/moodle/local/receiver.php` и предназначен для приёма HTTP-запросов с содержимым в формате JSON.

Сначала происходит чтение тела POST-запроса и его декодирование в ассоциативный массив:

Листинг 3.3 – Чтение и декодирование данных

```
1 <?php
2 $data = json_decode(file_get_contents('php://input'), true);
```

Если данные успешно получены, они сохраняются в текстовый файл `logs/log.txt` с помощью функции `file_put_contents`. Каждая запись до-

бавляется в конец файла, что обеспечивает накопление истории всех кликов пользователей.

Листинг 3.4 – Сохранение в лог и отправка ответа

```
1 if ($data) {  
2     file_put_contents('logs/log.txt', print_r($data, true),  
3     FILE_APPEND);  
4     echo "Получено: " . json_encode($data);  
5 } else {  
6     echo "Ничего не пришло или не JSON.";
```

Таким образом, данный модуль реализует начальный этап хранения кликов в виде текстового журнала, пригодного для последующей обработки.

3.5 Обработка данных

Модуль обработки реализует конвертацию собранных данных о кликах в формат JSON, пригодный для дальнейшего анализа и генерации визуализаций. Исходные данные поступают в виде лог-файла, содержащего текстовые записи в формате, аналогичном выводу функции *print_r* в PHP. Для перевода этих данных в структурированный формат используется скрипт на языке Python.

Общая структура работы скрипта включает следующие этапы:

- загрузка лог-файла с удалённого сервера по URL;
- разбор содержимого лог-файла на отдельные записи;
- извлечение ключей и значений из каждой записи;
- автоматическое приведение значений к подходящим типам (целые, вещественные, булевы и т. д.);
- сохранение результата в файл формата JSON.

Ниже приведён ключевой фрагмент кода, отвечающий за парсинг лог-файла и преобразование его содержимого:

Листинг 3.5 – Извлечение данных из лог-файла

```
1 array_pattern =  
    re.compile(r'Array\s*(\s*(.*?)\s*)\s*(?=Array|$)', re.DOTALL)  
2 field_pattern =  
    re.compile(r'\[\s*(.*?)\s*\]\s*=>\s*(.*?)(?=\s*\[|$)',  
        re.DOTALL)  
3 entries = []  
4  
5 for array_block in array_pattern.finditer(log_content):  
6     block_content = array_block.group(1)  
7     entry = defaultdict(str)  
8  
9     for field in field_pattern.finditer(block_content):  
10        key = field.group(1).strip()  
11        value = field.group(2).strip()  
12        ...
```

Для повышения точности анализа выполняется попытка автоматически привести значения к числовому типу или булевому:

Листинг 3.6 – Преобразование типов значений

```
1 if value.isdigit():  
2     value = int(value)  
3 elif re.match(r'^-?\d+\.\d+$', value):  
4     value = float(value)  
5 elif value.lower() == 'null':  
6     value = None  
7 elif value.lower() == 'true':  
8     value = True  
9 elif value.lower() == 'false':  
10    value = False
```

После обработки все записи сохраняются в формате JSON:

Листинг 3.7 – Сохранение в JSON

```
1 with open(output_json_path, 'w') as json_file:  
2     json.dump(entries, json_file, indent=2, ensure_ascii=False)
```

Выполнение всего процесса сопровождается выводом статистики:

Листинг 3.8 – Вывод количества записей

```
1 print(f"Успешно обработано {len(entries)} записей. Результат сохра  
   нён в {output_json_path}")
```

Таким образом, данный модуль осуществляет подготовку данных о пользовательских кликах для дальнейшей визуализации и анализа.

3.6 Снятие снимков с помощью вебдрайвера

Для автоматического создания скриншотов веб-страниц, на которых фиксировались клики пользователей, применяется скрипт с использованием библиотеки Selenium.

Основные функции и логика работы скрипта заключаются в следующем:

- чтение обработанных данных о кликах из файла `clicks.json` и выделение уникальных URL веб-страниц;
- настройка параметров браузера Chrome для работы в режиме без графического интерфейса (headless);
- создание временного профиля пользователя для изоляции сессии;
- автоматический вход в систему Moodle с использованием заданных учётных данных;
- последовательный переход по всем уникальным URL, ожидание полной загрузки страницы;
- формирование путей для сохранения скриншотов с безопасными для файловой системы именами директорий и файлов;
- выполнение скрипта для получения полноэкранного скриншота с помощью Chrome DevTools Protocol и сохранение его в указанный путь;
- обработка возможных исключений с выводом ошибок.

Ниже приведён пример функции, которая отвечает за создание скриншота всей страницы:

Листинг 3.9 – Создание полноэкранного скриншота

```
1 def take_fullpage_screenshot(driver, path):
2     driver.execute_script("window.scrollTo(0, 0);")
3     screenshot = driver.execute_cdp_cmd("Page.captureScreenshot",
4         {
5             "fromSurface": True,
6             "captureBeyondViewport": True
7         })
8     with open(path, "wb") as f:
9         f.write(base64.b64decode(screenshot['data']))
```

Функция `sanitize_path_part` используется для очистки частей URL, чтобы получить валидные имена папок и файлов:

Листинг 3.10 – Очистка строки для файловой системы

```
1 def sanitize_path_part(s):
2     return re.sub(r'[^a-zA-Z0-9_]', '_', s)
```

Для входа в Moodle применяется следующий блок кода:

Листинг 3.11 – Автоматизация входа в систему

```
1 driver.get(login_url)
2 WebDriverWait(driver,
3     15).until(EC.presence_of_element_located((By.ID, "username")))
4 time.sleep(1)
5 driver.find_element(By.ID, "username").clear()
6 driver.find_element(By.ID, "username").send_keys(USERNAME)
7 driver.find_element(By.ID, "password").clear()
8 driver.find_element(By.ID, "password").send_keys(PASSWORD)
9 driver.find_element(By.ID, "loginbtn").click()
10 WebDriverWait(driver, 10).until(
11     EC.presence_of_element_located((By.XPATH,
12         "//a[contains(@href, 'logout.php')]"))
13 )
```

Данный модуль обеспечивает автоматизированный сбор изображений страниц, позволяя в дальнейшем визуализировать распределение кликов пользователей в виде тепловых карт на этих снимках.

3.7 Генерация тепловых карт кликов

Часть модуля обработки, скрипт `heatmaps.py` отвечает за визуализацию распределения кликов пользователей в виде тепловых карт, наложенных на полученных на прошлом этапе скриншоты веб-страниц.

Основные этапы работы скрипта:

- загрузка данных кликов из JSON-файла `clicks.json`;
- группировка кликов по URL страниц;
- загрузка ранее сохранённого кэша для оптимизации повторной генерации тепловых карт;
- обход всех URL с проверкой изменения данных кликов по сравнению с кэшем;
- для каждого URL поиск соответствующего скриншота страницы;
- создание прозрачного слоя, на который наносятся цветные точки, соответствующие кликам;
- вычисление плотности кликов в окрестности каждой точки с применением гауссова распределения для плавного перехода цвета;
- определение цвета в зависимости от плотности от синего (одионые клики) к красному (наиболее плотные области);
- наложение слоя тепловой карты на исходный скриншот;
- сохранение итогового изображения в директорию с сохранением структуры URL.

Для очистки частей URL и формирования корректных путей используется функция:

Листинг 3.12 – Очистка части пути URL

```
1 def sanitize_path_part(s):  
2     return re.sub(r'^a-zA-Z0-9_', '_', s)
```

Основная функция генерации тепловых карт принимает параметры для путей к кликам, скриншотам, папке для сохранения и количеству последних кликов:

Листинг 3.13 – Основная функция генерации тепловых карт

```
1 def draw_clicks_on_screenshots(clicks_path="clicks.json",
2     screenshots_root="webpages/screenshots",
3     output_root="heatmaps",
4     last_n_clicks=None, force=True):
5     ...
```

Визуализация плотности кликов реализована через двухмерный массив, в который суммируются веса каждого клика с гауссовой функцией ядра. Цвет каждого пикселя зависит от рассчитанной плотности:

Листинг 3.14 – Определение цвета по плотности кликов

```
1 def get_color_for_density(density, max_density):
2     if density <= 1:
3         return (0, 0, 255, 160)
4     elif density < max_density / 2:
5         ratio = density / (max_density / 2)
6         return (int(255 * ratio), 0, int(255 * (1 - ratio)), 180)
7     else:
8         ratio = (density - max_density / 2) / (max_density / 2)
9         return (255, 0, int(255 * (1 - ratio)), 200)
```

Данный модуль позволяет наглядно представить интенсивность взаимодействия пользователей с элементами интерфейса сайта, что полезно для анализа юзабилити и улучшения дизайна.

3.8 Графический интерфейс просмотра тепловых карт

Модуль `interface.py` реализует полноэкранное GUI-приложение на основе библиотеки `tkinter` для просмотра сгенерированных тепловых карт кликов.

Основные возможности и структура интерфейса:

Левая панель содержит дерево файловой структуры папки с тепловыми картами (`heatmaps`), позволяя раскрывать и сворачивать каталоги, а также скрывать query-параметры в именах файлов для удобства просмотра.

Верхняя панель включает кнопки «Раскрыть всё», «Свернуть всё», чек-бокс скрывания query-параметров и поле ввода для указания количества последних кликов, которое передаётся в скрипт генерации тепловых карт.

Центральная область — `canvas` для отображения выбранного изображения (тепловой карты). Изображение масштабируется с сохранением пропорций под размер окна.

Нижняя консоль выводит логи запускаемых вспомогательных скриптов (`clicks.py`, `webdriver.py`, `heatmap.py`) в отдельном потоке, не блокируя интерфейс.

Правая панель отображает статистику по кликам, вычисленную из файла `clicks.json`, включая общее число кликов, количество уникальных URL, среднее число кликов на URL, средние координаты кликов, диапазон размеров страниц и самое популярное разрешение.

Кнопка «Обновить» запускает последовательно все три скрипта, обновляет дерево файлов и текущее изображение, а также обновляет статистику кликов.

Функция преобразования имени файла в URL с восстановлением query-параметров и PHP-расширений:

Листинг 3.15 – Преобразование имени файла в URL

```
1 def filename_to_url(name):
2     base = remove_png_extension(name)
3     if base.count('_') == 3 and all(part.isdigit() for part in
4         base.split('_')):
5         return base.replace('_', '.')
6     parts = base.split('_')
7     if len(parts) >= 2 and parts[1] == 'php':
8         filename = f"{parts[0]}.php"
9         if len(parts) > 2:
10             params = parts[2:]
11             query_parts = []
12             i = 0
13             while i < len(params):
14                 if i + 1 < len(params):
15                     param = params[i].replace('.', '=')
16                     if '=' not in param:
17                         param = f"{param}={params[i + 1]}"
18                     i += 1
19                     query_parts.append(param)
20                 else:
21                     query_parts.append(params[i])
22                 i += 1
23             return f"{filename}?{'&'.join(query_parts)}"
24     return filename
25     return base.replace('_', '.')
```

Для запуска внешних скриптов используется многопоточность с классом `threading.Thread` и перенаправлением вывода в текстовое поле консоли. Это сделано для бесшовного использования интерфейса во время работы остальных модулей системы.

Обработка выбора узла дерева отображает соответствующее изображение в центре с автоматическим масштабированием:

Листинг 3.16 – Отображение выбранного изображения на canvas

```
1 def on_tree_select(self , event):
2     ...
3 def display_image(self):
4     ...
5 def on_canvas_resize(self , event):
6     self.display_image()
```

Таким образом, модуль обеспечивает пользовательский интерфейс для просмотра и анализа тепловых карт кликов, а также интеграцию с процессом их генерации.

3.9 Форматы входных и выходных данных

Формат входных данных представляет собой необработанные клики, содержащие следующую информацию:

- URL страницы;
- координаты X и Y;
- разрешение экрана;

Формат выходных данных включает:

- файл с координатами кликов в json-формате;
- изображение страницы с наложенной тепловой картой;
- файл со статистикой (количество кликов, частота, наиболее активные зоны).

Для анализа и визуализации данных разработан графический интерфейс, который автоматически обновляет и отображает тепловые карты на основе сгенерированных файлов. Интерфейс поддерживает просмотр структуры файлов, раскрытие и сворачивание каталогов, фильтрацию по query-параметрам URL, а также масштабирование изображений под размер окна.

3.9.1 Проведение тестирования системы

В рамках данной работы было проведено только функциональное тестирование системы тепловой визуализации пользовательских кликов, т.к. система состоит из связанных компонентов, основная задача которых заключается в обеспечении целостного сценария сбора, обработки и визуализации пользовательских кликов.

Модульное тестирование не проводилось, так как реализация системы ориентирована на непосредственную интеграцию всех модулей.

Тестирование графического интерфейса не проводилось, так как пользовательский интерфейс проверялся визуально в процессе разработки, без применения инструментов автоматизированного GUI-тестирования.

Функциональное тестирование проводилось по следующему алгоритму:

- 1) автоматическая инициализация скриптов сбора и приёма данных о кликах в Moodle;
- 2) генерация тестовых кликов вручную и с использованием автоматизированного скрипта;
- 3) сбор и сохранение координат в формате JSON;
- 4) запуск серверного модуля генерации скриншотов;
- 5) формирование тепловой карты на основе собранных кликов;
- 6) отображение результата в пользовательском интерфейсе.

Графически алгоритм тестирования приведён на рисунке 3.3.



Рисунок 3.3 – Алгоритм тестирования

Для проверки корректности функционирования были выделены следующие классы эквивалентности:

- количество кликов: 0, от 1 до 10, более 50;
- координаты: внутри видимой области и за её пределами;
- уровень активности на страницах: отсутствующая (0 кликов в минуту), умеренная (10 кл./мин.), высокая (50 кл./мин.).

Во всех случаях система показала стабильную и корректную работу:

- данные сохранялись в файл и обрабатывались на стороне сервера;
- при отсутствии кликов карта не формировалась, интерфейс отображал пустую страницу без ошибок;
- визуализация адаптировалась под различные размеры экрана;
- фильтрация URL по query-параметрам функционировала корректно;
- пользовательский интерфейс отображал изображения тепловых карт без сбоев.

Для автоматизированного создания тестовых кликов использовался скрипт, реализованный на языке Python с применением библиотеки Selenium. Он выполняет авторизацию в Moodle, переходит на целевую страницу, вычисляет размеры области документа и запускает цикл генерации кликов по случайным координатам.

Ниже приведены ключевые фрагменты этого скрипта.

Листинг 3.17 – Инициализация браузера и авторизация

```

1 options = Options()
2 options.add_argument('--headless=new')
3 options.add_argument('--window-size=1920,1080')
4 driver = webdriver.Chrome(options=options)
5
6 driver.get("http://192.168.1.102/moodle/login/index.php")
7 driver.find_element(By.ID, "username").send_keys(USERNAME)
8 driver.find_element(By.ID, "password").send_keys(PASSWORD)
9 driver.find_element(By.ID, "loginbtn").click()
```

Данный блок запускает экземпляр браузера Chrome в фоновом режиме и осуществляет вход от имени администратора.

Листинг 3.18 – Генерация и фиксация случайных кликов

```
1 for i in range(NUM_CLICKS):
2     absX = random.randint(10, width - 10)
3     absY = random.randint(10, height - 10)
4
5     driver.execute_script("""
6         const x = arguments[0], y = arguments[1];
7         const el = document.elementFromPoint(x, y) ||
8             document.body;
9
10        el.dispatchEvent(new PointerEvent('pointerdown', {...}));
11        el.dispatchEvent(new PointerEvent('pointerup', {...}));
12        el.dispatchEvent(new MouseEvent('click', {...}));
13    """, absX, absY)
```

Этот фрагмент запускает последовательность событий, имитирующих пользовательский клик мыши по элементам на странице. Координаты выбираются случайным образом в пределах доступной области.

Листинг 3.19 – Сохранение результата в JSON

```
1 results.append({
2     "relX": relX,
3     "relY": relY,
4     "absX": absX,
5     "absY": absY,
6     "url": url,
7     "title": title,
8     "pageWidth": width,
9     "pageHeight": height
10 })
11
12 with open("autoclicks.json", "w", encoding="utf-8") as f:
13     json.dump(results, f, indent=2, ensure_ascii=False)
```

Собранные данные сохраняются в файл *autoclicks.json* и используются далее для сравнения с клиентскими логами.

3.10 Примеры работы

Ниже приведены примеры работы системы, включая скриншоты веб-страниц, поверх которых наложена тепловая карта активности пользователей.

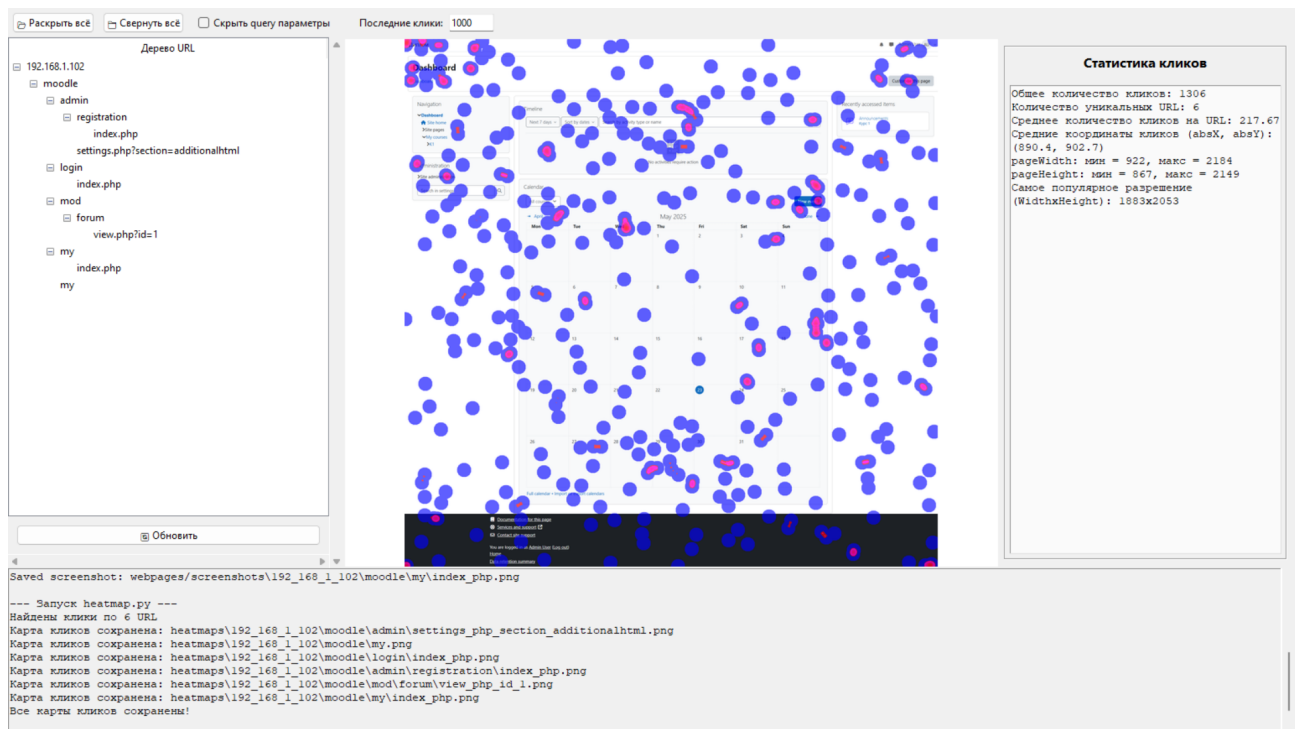


Рисунок 3.4 – Тепловая карта, 1000 кликов

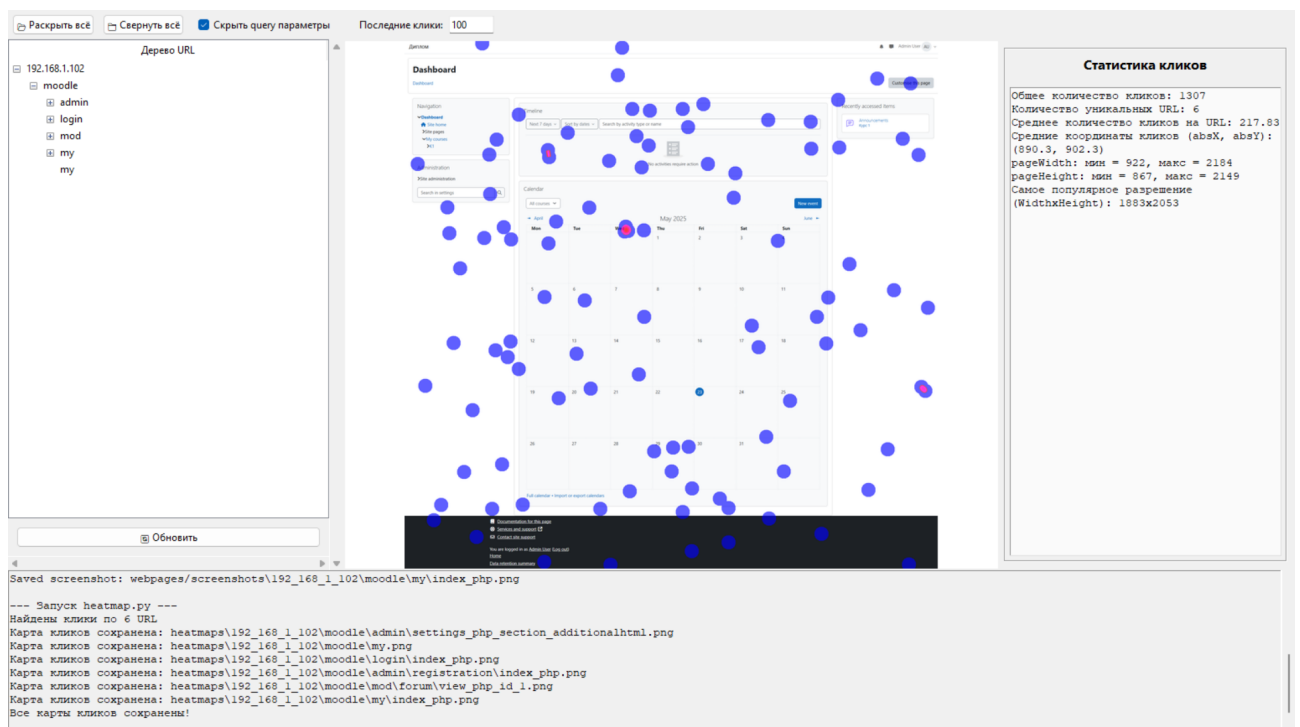


Рисунок 3.5 – Тепловая карта, 100 кликов

Выводы

Выбраны технологии и разработана архитектура с тремя модулями: сбор, хранение и анализ данных. Реализован клиентский сбор данных на JavaScript и серверный приём логов на PHP. Создан Python-скрипт для преобразования логов в JSON. Внедрён автоматический захват скриншотов с Selenium для тепловых карт. Разработан модуль генерации тепловых карт и интерфейс на tkinter для их просмотра. Реализован комплекс для сбора, обработки и анализа кликов.

4 Исследовательский раздел

4.1 Предмет исследования

В настоящем разделе проводится исследование характеристик разработанного метода генерации тепловых карт кликов на веб-страницах. Целью исследования является установление зависимости производительности метода от объёма обрабатываемых данных.

4.2 Характеристики вычислительной машины

Для проведения исследований использовалась вычислительная машина с параметрами:

- операционная система: Windows 11;
- объём оперативной памяти: 16 Гб;
- процессор—AMD Ryzen 5 5500U, 2100 МГц, 6 ядер.

4.3 Результаты исследования

4.3.1 Производительность метода

Производительность оценивается по времени выполнения обработки данных при различных объёмах кликов.

Для измерения производительности и построения всех графиков использовался следующий подход:

- число кликов изменяется в диапазоне от 0 до 10000 с шагом 1000;
- для каждого заданного объёма данных формируются или корректируются JSON-файлы так, чтобы они содержали точное количество записей;
- запускается скрипт, при этом фиксируются параметры работы с момента старта и до завершения генерации тепловой карты;
- для каждого объёма данных выполняется 20 повторных запусков с одинаковыми настройками;
- итоговые значения параметров усредняются по всем 20 замерам и используются для построения графиков.

Данный алгоритм усреднения применялся не только при замерах времени выполнения, но и при сборе данных о среднем и пиковом потреблении оперативной памяти, а также средней загрузке процессора. Это обеспечило получение статистически обоснованных и устойчивых результатов для всех исследуемых метрик.

4.3.2 Средние значения параметров при различных объёмах данных

Измеряемые критерии производительности:

- время выполнения — суммарное время, затраченное на обработку заданного объёма кликов и генерацию тепловой карты, измеряется в секундах;
- среднее потребление оперативной памяти (RAM) — усреднённое значение объёма используемой памяти в процессе выполнения алгоритма, измеряется в мегабайтах;
- пиковое потребление оперативной памяти (RAM) — максимальный объём памяти, занятый в ходе работы метода, измеряется в мегабайтах;
- средняя загрузка процессора (CPU) — усреднённый процент использования процессора во время выполнения обработки.

Таблица 4.1 – Средние значения времени выполнения, потребления памяти и загрузки CPU

Клики, шт.	Время	RAM, среднее	RAM, пиковое	CPU, среднее
0	1.05	120.00	130.00	23.0
1000	4.12	130.45	145.22	24.0
2000	7.98	145.77	160.14	24.7
3000	11.95	160.85	175.88	25.0
4000	15.87	175.32	190.76	25.2
5000	19.85	190.10	205.23	25.4
6000	23.76	202.44	218.14	25.5
7000	27.80	212.68	230.01	25.6
8000	31.70	219.95	240.55	25.7
9000	35.60	224.80	247.40	25.8
10000	39.50	228.42	257.19	25.9

4.3.3 Графики зависимости производительности и использования ресурсов

На рисунке 4.1 представлена зависимость времени обработки от количества кликов.

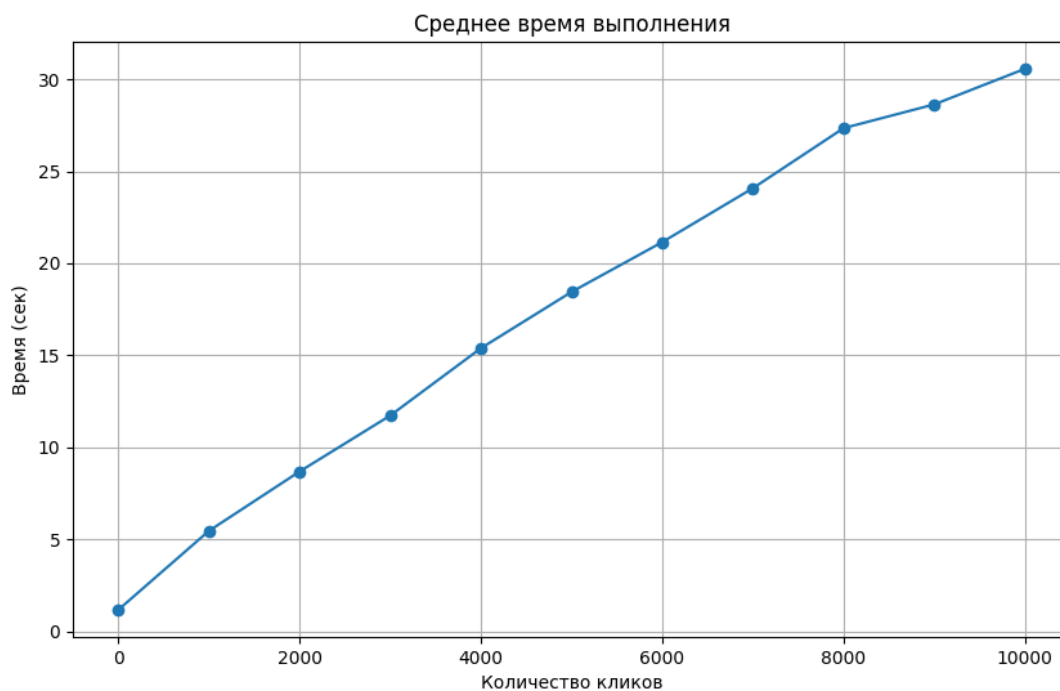


Рисунок 4.1 – Время генерации тепловых карт от количества кликов

Время запуска скрипта без обработки кликов можно посчитать как время, потраченное при нулевом количестве кликов. Оно составляет чуть больше одной секунды. При последующем замере, где клики уже присутствуют, наблюдается резкий рост времени работы скрипта. Далее график принимает линейную зависимость, что позволяет его интерполировать.

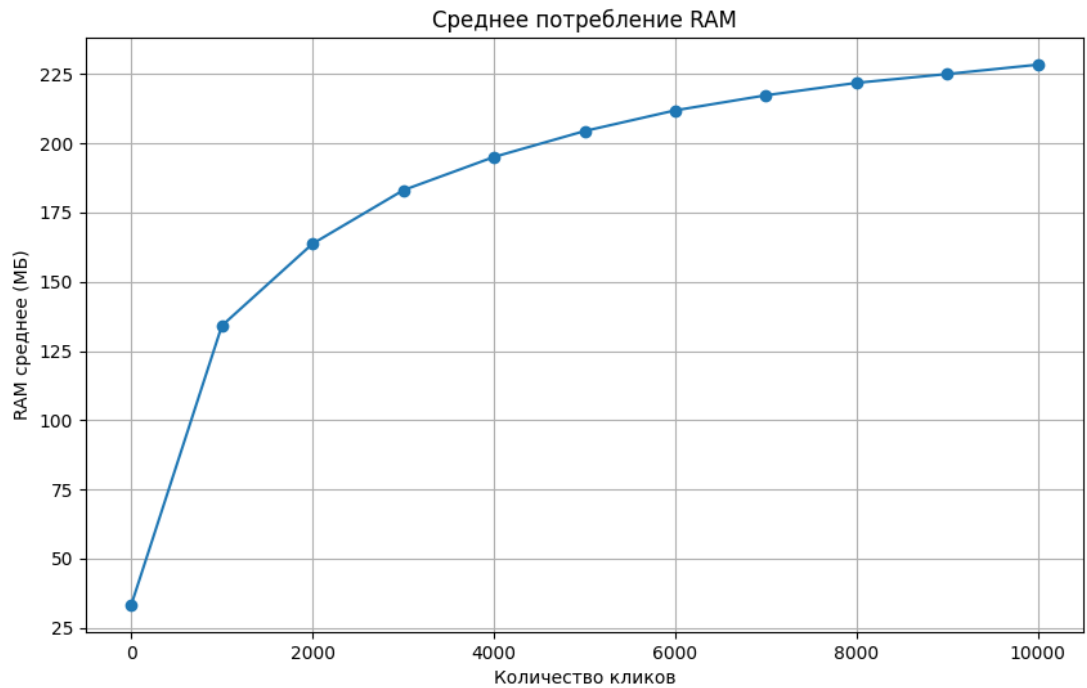


Рисунок 4.2 – Среднее потребление оперативной памяти от количества кликов

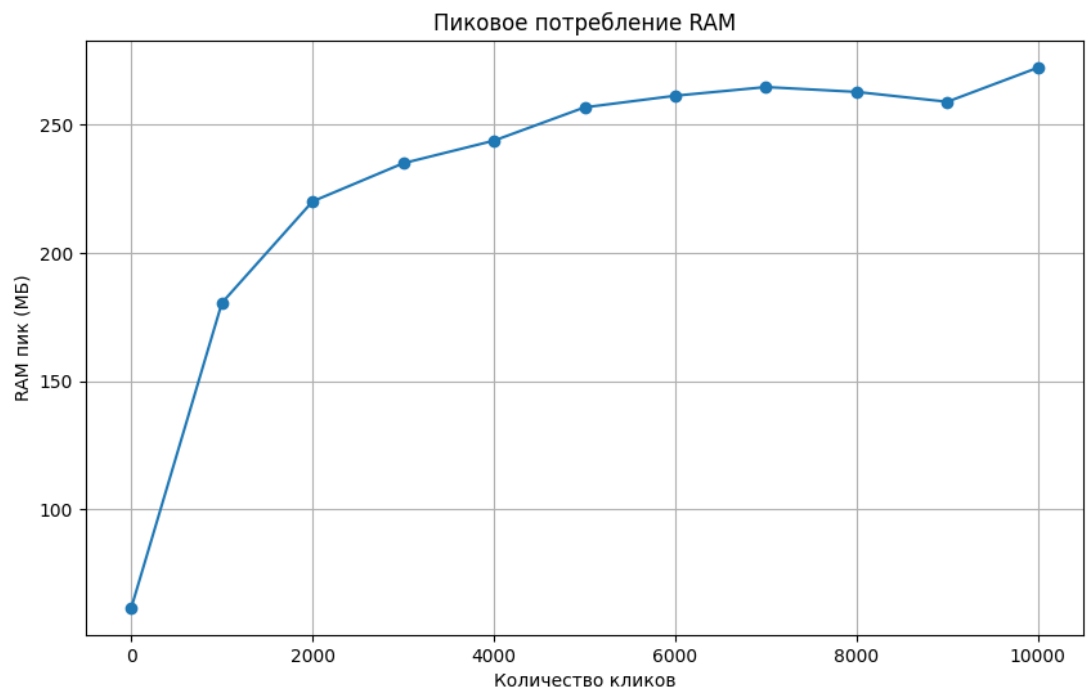


Рисунок 4.3 – Пиковое потребление оперативной памяти от количества кликов

Измерения пикового и среднего потребления оперативной памяти в процессе выполнения метода показали, что оба параметра зависят от объёма

данных нелинейным образом. На графиках (рисунки 4.2 и 4.3) наблюдается логарифмический рост как среднего, так и пикового потребления RAM при увеличении количества кликов.

Данная зависимость свидетельствует о том, что при увеличении объёма входных данных нагрузка на память растёт более медленно по сравнению с линейным ростом времени обработки. Это можно объяснить эффективным использованием структур данных и алгоритмов, которые позволяют ограничить объём занимаемой памяти.

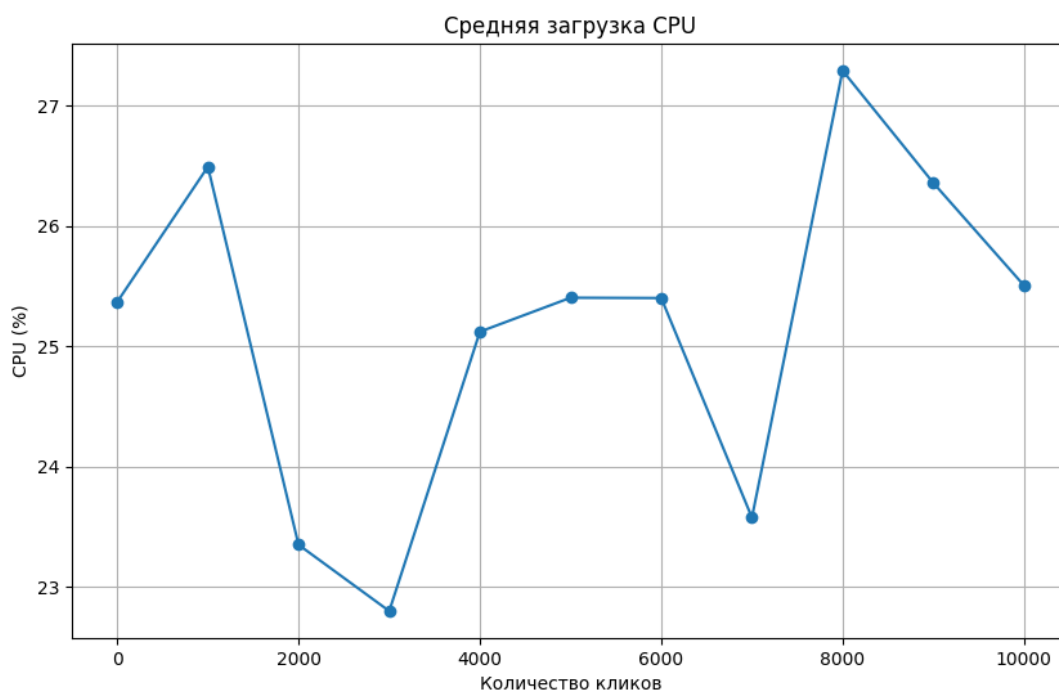


Рисунок 4.4 – Средняя загрузка процессора от количества кликов

График загрузки CPU, представленный на рисунке 4.4, показывает относительно стабильный уровень средней загрузки процессора на протяжении всего диапазона исследуемых объёмов данных.

Средняя загрузка CPU колеблется в пределах умеренных значений (примерно от 23% до 27%) вне зависимости от количества обрабатываемых кликов. Отсутствие выраженной корреляции между количеством кликов и загрузкой процессора указывает на то, что основное время выполнения скрипта может быть связано с операциями, не требующими высокой вычислительной мощности, такими как ввод-вывод или работа с памятью. При этом средняя загрузка CPU ниже 30% свидетельствует о том, что процессор не является

главным ограничивающим фактором в данном методе.

4.4 Инструменты мониторинга и профилирования

Для измерения временных характеристик, использования памяти и загрузки процессора применялись следующие инструменты и библиотеки:

- модуль `time` [35] языка Python для измерения времени генерации данных и общего времени выполнения;
- библиотека `psutil` [36] для мониторинга использования оперативной памяти (RAM) и загрузки CPU в режиме реального времени;
- собственный скрипт, реализующий многократные прогоны алгоритма для получения усреднённых значений параметров производительности.

Собранные данные включают среднее и пиковое значение потребляемой памяти, среднюю загрузку процессора и время выполнения каждого прогона.

4.5 Анализ вычислительной сложности алгоритма

С точки зрения теории вычислительных затрат временная сложность алгоритма оценивается как $O(n)$, где n — количество обработанных кликов. Основная часть работы представляет собой последовательную обработку каждого элемента входного набора без вложенных циклов, растущих с n .

Пространственная сложность алгоритма также является линейной, $O(n)$, поскольку объём используемой оперативной памяти пропорционален объёму входных данных.

Таким образом, алгоритм обладает линейной временной и пространственной сложностью, что обеспечивает его масштабируемость при увеличении объёма данных.

4.6 Выводы

В ходе исследования установлено, что разработанный метод демонстрирует:

- линейную зависимость времени обработки от количества данных в типичном диапазоне;
- логарифмический рост среднего и пикового потребления оперативной памяти при увеличении объёма данных;

- стабильный уровень средней загрузки процессора вне зависимости от объёма обрабатываемых кликов.

Полученные результаты позволяют рекомендовать метод для использования в условиях средней нагрузки, с учётом возможных ограничений при масштабировании, связанных с ростом времени обработки. Эффективное потребление памяти и умеренная загрузка CPU свидетельствуют о хорошей адаптивности метода к увеличению объёма данных.

ЗАКЛЮЧЕНИЕ

В результате проведённой работы раскрыто понятие пользовательского поведения. Рассмотрены основные методологические подходы к анализу пользовательского поведения. Выявлены основные способы, методы и технологии сбора статистики о пользователях веб-приложений: cookie, счётчики, журналирование действий. Проанализированы преимущества и недостатки перечисленных подходов и их технологических реализаций. В результате анализа пользовательского поведения рассмотрена технология тепловых карт, включающая в себя карты кликов, ссылок, скроллинга, пути по сайту. В качестве метода сбора информации рассмотренной платформы электронного обучения выбрана технология лог-файлов, а в качестве метода анализа этих данных — тепловая карта кликов.

В ходе работы достигнуты все поставленные задачи:

- выполнен анализ предметной области;
- разработана функциональная модель обработки данных о кликах и определены структуры данных для их хранения;
- разработан предлагаемый метод;
- спроектирована и реализована соответствующая система;
- проведено тестирование и оценка эффективности.

Цель данной работы выполнена.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. *Rusli M., Dadang H., Ni-Nyoman S.* Innovative Learning Multimedia. — 2017.
2. *Роскош М. В.* Город как система // Научный диалог. — 2013. — 12 (24). — С. 48—57.
3. Learning management system (LMS) among university students: Does it work / N. A. Adzharuddin, L. H. Ling [и др.] // International Journal of e-Education, e-Business, e-Management and e-Learning. — 2013. — Т. 3, № 3. — С. 248—252.
4. *TalentTech.* Исследование рынка онлайн-образования. — URL: <https://main.talenttech.ru/research/issledovanie-rynka-onlajn-obrazovaniya/> ; Дата обращения: 26.05.2025.
5. The impact of the COVID-19 pandemic on higher education: Assessment of student performance in computer science.
6. *E. M.* Effectiveness of traditional and online learning: comparative analysis from the student perspective. — 2021.
7. Comparing student outcomes in traditional vs intensive, online graduate programs in health professional education / K. Harwood [и др.]. — 2018.
8. Students' perception of online learning during the COVID-19 pandemic: a survey study of Polish medical students / M. Bączek [и др.]. — 2021.
9. Influence of COVID-19 confinement on students' performance in higher education / T. Gonzalez [и др.]. — 2020.
10. Are students performing the same in E-learning and In-person education? An introspective look at learning environments from an Iranian medical school standpoint / H. Mastour [и др.]. — 2023.
11. Цифровая кафедра МГТУ им. Н.Э. Баумана. — URL: <https://dc.bmstu.ru/>.
12. *Блэжуэлл Р. Д., Миниард П. У., Энджел Д. Ф.* Поведение потребителей. — 2007.
13. *Ефремова Т. Ф.* Толковый словарь русского языка. — URL: <http://www.efremova.info/> ; Дата обращения: 26.05.2025.

14. *Лысова Е. А.* Поведение потребителей: содержание понятия, социально-экономическая сущность. — 2019.
15. Социологический словарь. — URL: [http : / / vslovare . ru / slovo / sotziologicheskii-j-slovar / potreblenie](http://vslovare.ru/slovo/sotziologicheskii-j-slovar/potreblenie) ; Дата обращения: 26.05.2025.
16. *Александрова И. Ю.* Методология маркетингового исследования интернет-пользователей. — 2019.
17. *Петрова Н. Г. е. а.* Социологический опрос как метод маркетингового исследования. — 2007.
18. Социология : энциклопедия. — 2003.
19. *Щвец Е. В.* Интервью в системе методов опроса. — 2008.
20. *Копцева Н.* Проведение экспериментального прикладного культурного исследования межкультурной коммуникации: фокус-группы, личное интервью, анкетирование, получение экспертного мнения (на материале исследования Красноярского края) // Современные проблемы науки и образования. — 2013. — № 3. — С. 410—410.
21. *Прохорова А. М.* Роль методов анализа и прогнозирования поведения пользователей на образовательном портале. — 2017.
22. *Кушмар С. Е., Пилецкий И. И.* Некоторые аналитические методы анализа поведения пользователей сайта. — 2015.
23. *Яндекс.* Документация Яндекс.Метрики. — URL: <https://yandex.ru/support/metrika/ru/> ; Дата обращения: 26.05.2025.
24. *Clarify.* Microsoft Clarify Documentation. — URL: [https : / / learn . microsoft.com/en-us/clarify/](https://learn.microsoft.com/en-us/clarify/) ; Дата обращения: 26.05.2025.
25. *Жеребило Т. В.* Словарь лингвистических терминов. — ООО «Пилигрим», 2010.
26. *Selenium.* The Selenium Browser Automation Project. — URL: [https : //www.selenium.dev/documentation/](https://www.selenium.dev/documentation/) ; Дата обращения: 26.05.2025.
27. *Ubuntu.* Ubuntu 22.04.5 LTS (Jammy Jellyfish). — URL: <https://releases.ubuntu.com/22.04.5/> ; Дата обращения: 26.05.2025.

28. *Moodle*. Moodle Documentation. — URL: https://docs.moodle.org/500/en/Main_page ; Дата обращения: 26.05.2025.
29. *Group T. P. D.* Руководство по PHP. — URL: <https://www.php.net/manual/ru/> ; Дата обращения: 26.05.2025.
30. *Foundation T. A. S.* Apache HTTP Server Version 2.4 Documentation. — URL: <https://httpd.apache.org/docs/current/> ; Дата обращения: 26.05.2025.
31. *Group T. P. G. D.* PostgreSQL 14.17 Documentation. — URL: <https://www.postgresql.org/docs/14/index.html> ; Дата обращения: 26.05.2025.
32. *Foundation T. P. S.* Python 3.11.12 documentation. — URL: <https://docs.python.org/3.11/index.html> ; Дата обращения: 26.05.2025.
33. *Foundation M.* JavaScript | MDN. — URL: <https://developer.mozilla.org/en-US/docs/Web/JavaScript> ; Дата обращения: 26.05.2025.
34. *JetBrains*. Getting started | PyCharm Documentation. — URL: <https://www.jetbrains.com/help/pycharm/getting-started.html> ; Дата обращения: 26.05.2025.
35. *Foundation T. P. S.* time — Time access and conversions. — URL: <https://docs.python.org/3/library/time.html> ; Дата обращения: 26.05.2025.
36. *Rodola G.* psutil documentation. — URL: <https://psutil.readthedocs.io/en/latest/index.html> ; Дата обращения: 26.05.2025.

ПРИЛОЖЕНИЕ А

Презентация к научно-исследовательской работе состоит из *** слайдов.