

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

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

2025 год

РЕФЕРАТ

Расчетно-пояснительная записка 64 с., 4 табл., 36 ист., 1 прил.

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

Цель работы — разработка метода первичного анализа поведения студентов на веб-страницах.

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

РЕФЕРАТ	5
ВВЕДЕНИЕ	8
1 Аналитический раздел	9
1.1 Определение и назначение систем электронного обучения	9
1.2 Актуальность электронного обучения	10
1.3 Цифровизация образования в государственной политике	11
1.4 Определение потребления, потребителя и поведения	12
1.5 Методология анализа пользовательского поведения	14
1.6 Наблюдение	14
1.7 Опросы и интервью	15
1.8 Эксперимент	17
1.9 Анализ документов	17
1.10 Сбор и интерпретация данных	18
1.11 Методы сбора данных	18
1.12 Интерпретация данных	20
1.13 Анализ ограничений существующих решений	22
1.14 Требование публичного домена	22
1.15 Обработка данных на стороне третьих лиц	22
1.16 Ограниченность контроля и модифицируемости	22
1.17 Невозможность оптимизации производительности	22
1.18 Сравнительный анализ существующих решений	23
2 Конструкторский раздел	25
2.1 Модули разрабатываемого метода	26
2.2 Требования к разрабатываемому методу	29
2.3 Формат входных и выходных данных	30
2.4 Обработка данных	30
2.5 Особенности разрабатываемого метода	31
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	34
3.4	Приём и сохранение данных	36
3.5	Обработка данных	36
3.6	Снятие снимков с помощью вебдрайвера	38
3.7	Генерация тепловых карт кликов	40
3.8	Графический интерфейс просмотра тепловых карт	42
3.9	Форматы входных и выходных данных	44
3.10	Проведение тестирования системы	45
3.11	Примеры работы	49
4	Исследовательский раздел	51
4.1	Характеристики вычислительной машины	51
4.2	Результаты исследования	51
4.3	Средние значения параметров при различных объёмах данных	52
4.4	Инструменты мониторинга и профилирования	56
4.5	Выводы	57
	ЗАКЛЮЧЕНИЕ	58
	СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	59
	ПРИЛОЖЕНИЕ А	63
	ПРИЛОЖЕНИЕ Б	64

ВВЕДЕНИЕ

Развитие информационных технологий способствует широкому внедрению систем электронного обучения в образовательный процесс [1]. Электронные образовательные платформы обеспечивают доступ к учебным материалам и взаимодействие обучающихся с преподавателями и друг с другом вне традиционного аудиторного формата [2]. Рост популярности электронного обучения подтверждается как стабильным увеличением инвестиций в отрасль [3], так и активным переходом части обучения в дистанционный формат.

Актуальность работы определяется необходимостью повышения качества электронного обучения за счёт анализа и оптимизации поведения пользователей платформы на примере системы Цифровой кафедры МГТУ им. Н.Э. Баумана [4].

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 — разработка метода первичного анализа поведения пользователей на веб-страницах.

Для выполнения работы были поставлены следующие задачи:

- провести анализ предметной области;
- разработать метод, сформулировав и описав его ключевые этапы в виде IDEF0 и схем алгоритмов;
- разработать программное обеспечение, реализующее метод;
- исследовать зависимость потребляемых программой ресурсов от объёма данных.

1 Аналитический раздел

В настоящем разделе рассматриваются теоретические основы, касающиеся электронного обучения и анализа пользовательского поведения.

Определены ключевые понятия, такие как система электронного обучения, потребитель и пользователь, а также представлена роль цифровизации образования. Особое внимание уделено методологическим подходам к анализу пользовательского поведения: наблюдению, опросам, экспериментам и анализу документов.

Кроме того, описаны методы сбора и интерпретации данных, необходимые для последующей реализации системы анализа поведения пользователей в электронных образовательных системах.

1.1 Определение и назначение систем электронного обучения

В связи с развитием информационных технологий всё 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 определённую целостность, единство [5].

LMS в общем случае может представлять собой некоторый веб-ресурс, онлайн-портал, который предоставляет возможность обмена учебными материалами и проведения 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 [2].

На рисунке 1.1 представлена домашняя страница системы электронного обучения МГТУ им. Н.Э. Баумана.

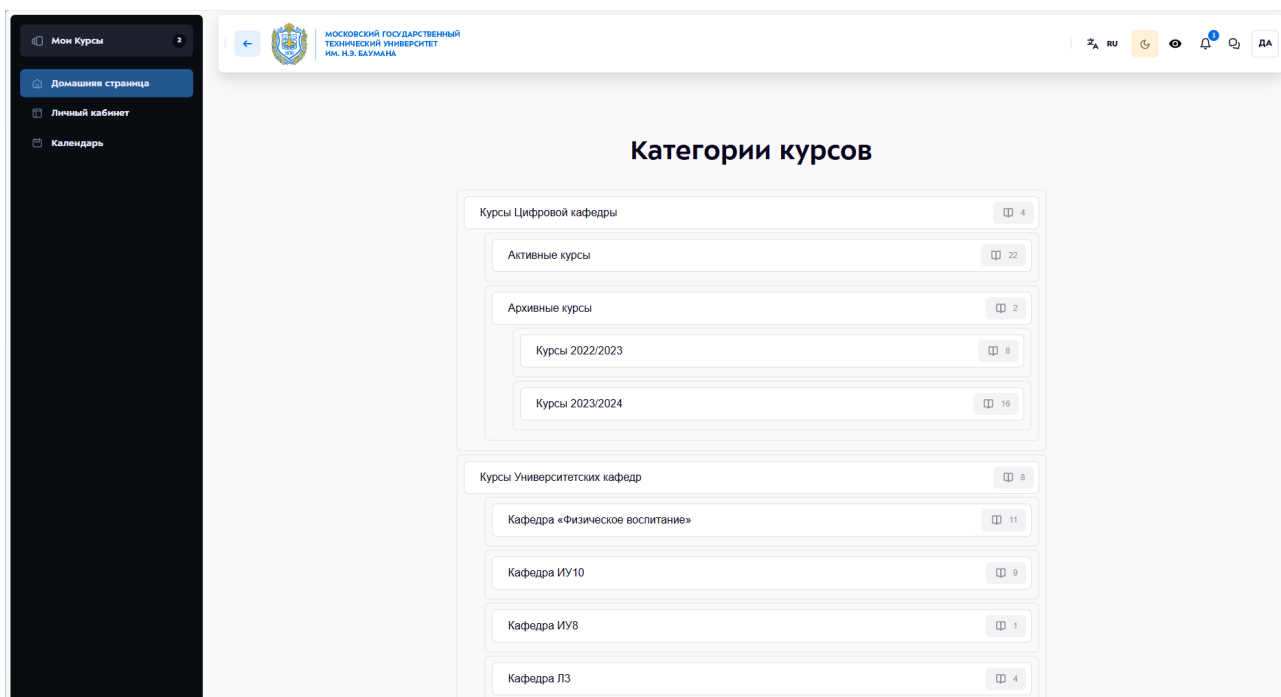


Рисунок 1.1 – Веб-сервис LMS BMSTU

1.2 Актуальность электронного обучения

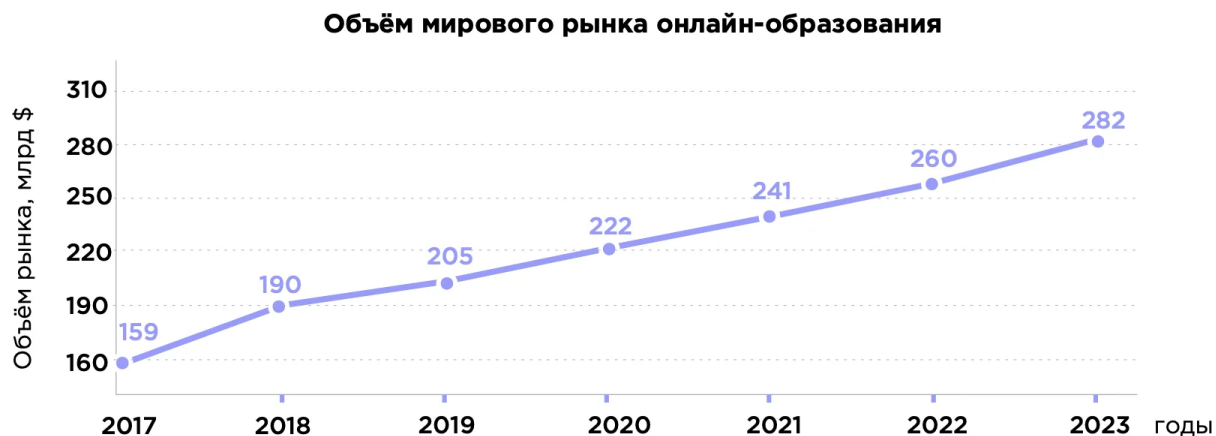
Школы и университеты в некоторой мере используют системы электронного обучения. Этому способствовала пандемия коронавируса и последующие запреты на проведение массовых мероприятий — в результате этого в школах и университетах обучение перешло в онлайн-формат, но даже после снятия ограничений часть материалов время от времени преподаётся дистанционно.

Популярность электронного образования, в том числе, можно отследить по объёму инвестиций. Так, по данным исследования [3], в 2019 году наблюдался закономерный скачок вложенных средств. Графическое представление объёма инвестиций представлено на рисунке 1.2.



Рисунок 1.2 – Инвестиции в российский рынок онлайн-образования

В то же время существуют зафиксированные факты того, что рынок электронного обучения на протяжении нескольких лет показывает стабильный рост. На рисунке 1.3 представлена динамика роста объёма соответствующего мирового рынка в результате исследования [3] рынка онлайн-образования.



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 1.3 – Объём мирового рынка электронного обучения

1.3 Цифровизация образования в государственной политике

Цифровизация образовательных процессов является одним из методов повышения эффективности передачи знаний, что подтверждает ряд исследований [6—11].

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 своё применение и в рамках отечественных университетов. Так, реализуемые в рамках национального проекта «Цифровая экономика» направления Цифровой кафедры МГТУ им. Н.Э. Баумана [4] (далее — ЦК) проводятся на одной из подобных площадок.

1.4 Определение потребления, потребителя и поведения

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 вовлечённой 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 определённой торговой марки товара;
- деятельность, непосредственно вовлечённая 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 определённого 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. Лысовой [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 учётом 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.

1.5 Методология 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.

1.6 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.

1.7 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. [20] 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 [21].

1.8 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 — имеющие место в естественных условиях, т.е. на дому или в магазине.

1.9 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].

1.10 Сбор и интерпретация данных

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.

1.11 Методы сбора данных

Обратившись к работе Прохоровой А.М. «Роль методов анализа и прогнозирования поведения пользователей на образовательном портале» [22] 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

1.12 Интерпретация данных

Полученные данные можно интерпретировать в нескольких форматах. Кушмар С.Е. и Пилецкий И.И. [23] 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.

1.13 Анализ ограничений существующих решений

Существующие сервисы для анализа пользовательского поведения, такие как Яндекс.Метрика [24] и Clarity [25], 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.

1.14 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.

1.15 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.

1.16 Ограниченность контроля и модифицируемости

Поскольку Яндекс.Метрика [24] и Clarity [25] 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.

1.17 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.

1.18 Сравнительный анализ существующих решений

Для оценки применимости популярных решений к задачам анализа пользовательского поведения в локальных системах проведено сравнение по ряду ключевых критериев. В таблице представлены данные по трём широко используемым сервисам: Вебвизор (часть Яндекс.Метрики [24]), Clarity [25] и Plerdy [26].

Таблица 1.3 – Сравнение существующих сервисов анализа поведения пользователей

Критерий	Вебвизор	Clarity	Plerdy
Наличие карт кликов	Есть	Есть	Есть
Требование публичного домена	Есть	Есть	Есть
Открытый исходный код	Нет	Есть	Нет

Как видно из таблицы, несмотря на наличие базовой функциональности по отображению карт кликов, все перечисленные решения требуют размещения сайта на публичном домене, что делает их непригодными для анализа локальных веб-систем.

Кроме того, только Clarity [25] предоставляет открытый исходный код, что теоретически позволяет его доработку, но это не решает проблему внешней обработки данных.

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.

На рисунке 1.4 представлена IDEF0-диаграмма, формализующая процесс сбора и интерпретации пользовательских данных с использованием указанных инструментов.

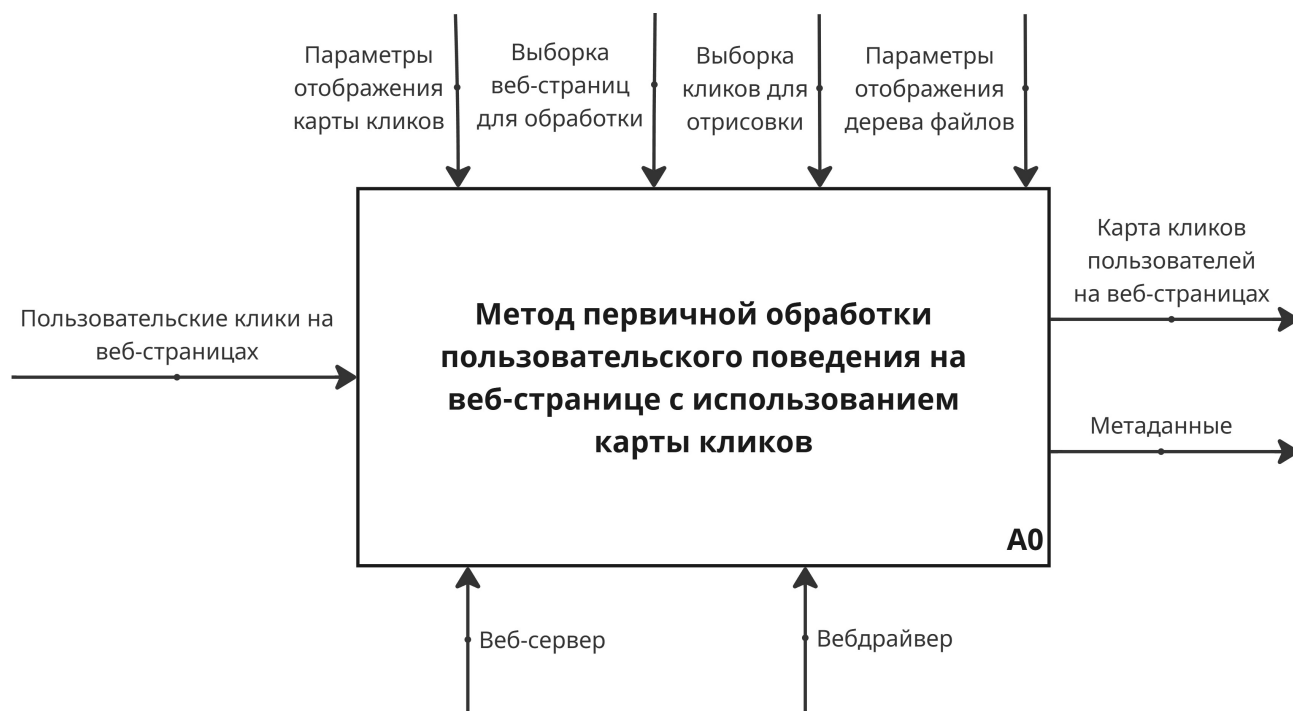


Рисунок 1.4 – Процесс сбора и интерпретации данных

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 [24] и Microsoft Clarity [25], 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 метода

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 [27].

Система включает следующие основные модули (рисунок 3.1):

- 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.

Несмотря на локальное развертывание, архитектура системы остаётся клиент-серверной: клиентский модуль отвечает за сбор и отправку данных, серверный — за их хранение, обработку и визуализацию.

В связи с локальным развертыванием такая архитектура позволяет избежать хранения собираемых данных на серверах третьих лиц и предоставляет конечным пользователям возможность оптимизации разрабатываемого решения.

Верхнеуровневая схема модулей приведена на рисунке 2.1:



Рисунок 2.1 – Верхнеуровневая модульная схема

Таким образом, IDEF0 диаграмма второго уровня приведена на рисунке 2.2.

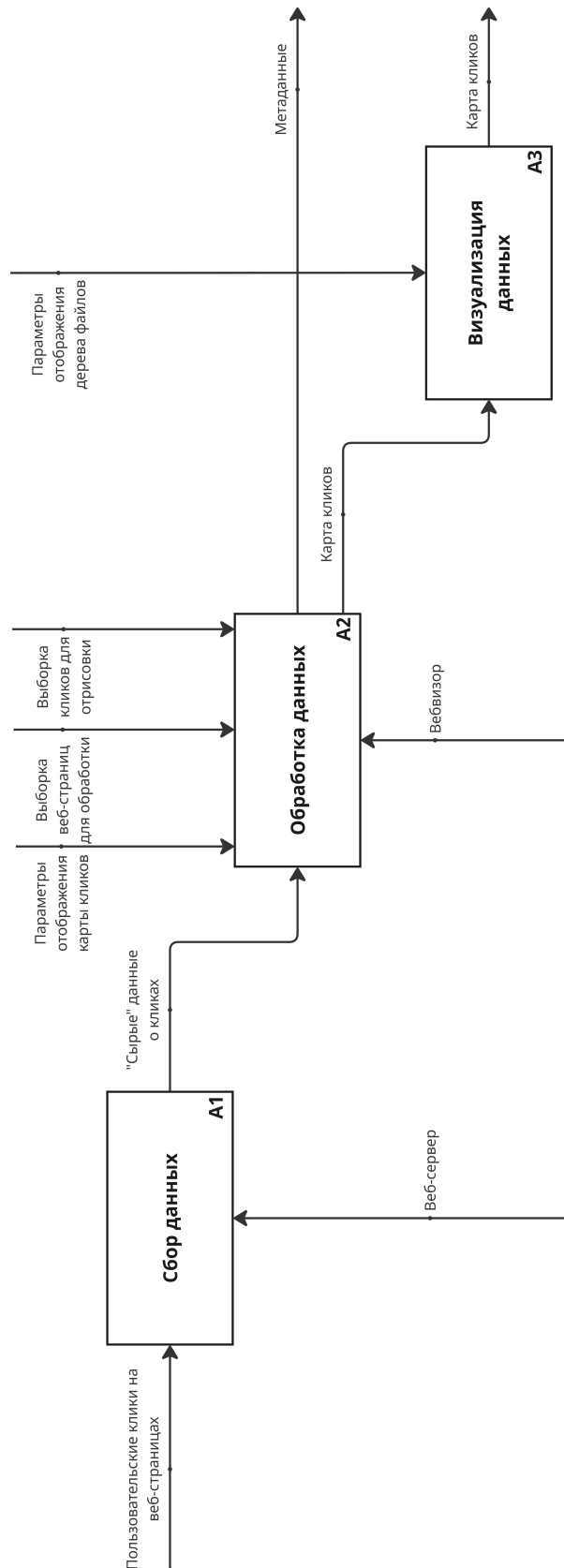


Рисунок 2.2 – Процесс сбора и интерпретации данных на уровне модулей

Схема разрабатываемого метода приведена на рисунке 2.3.

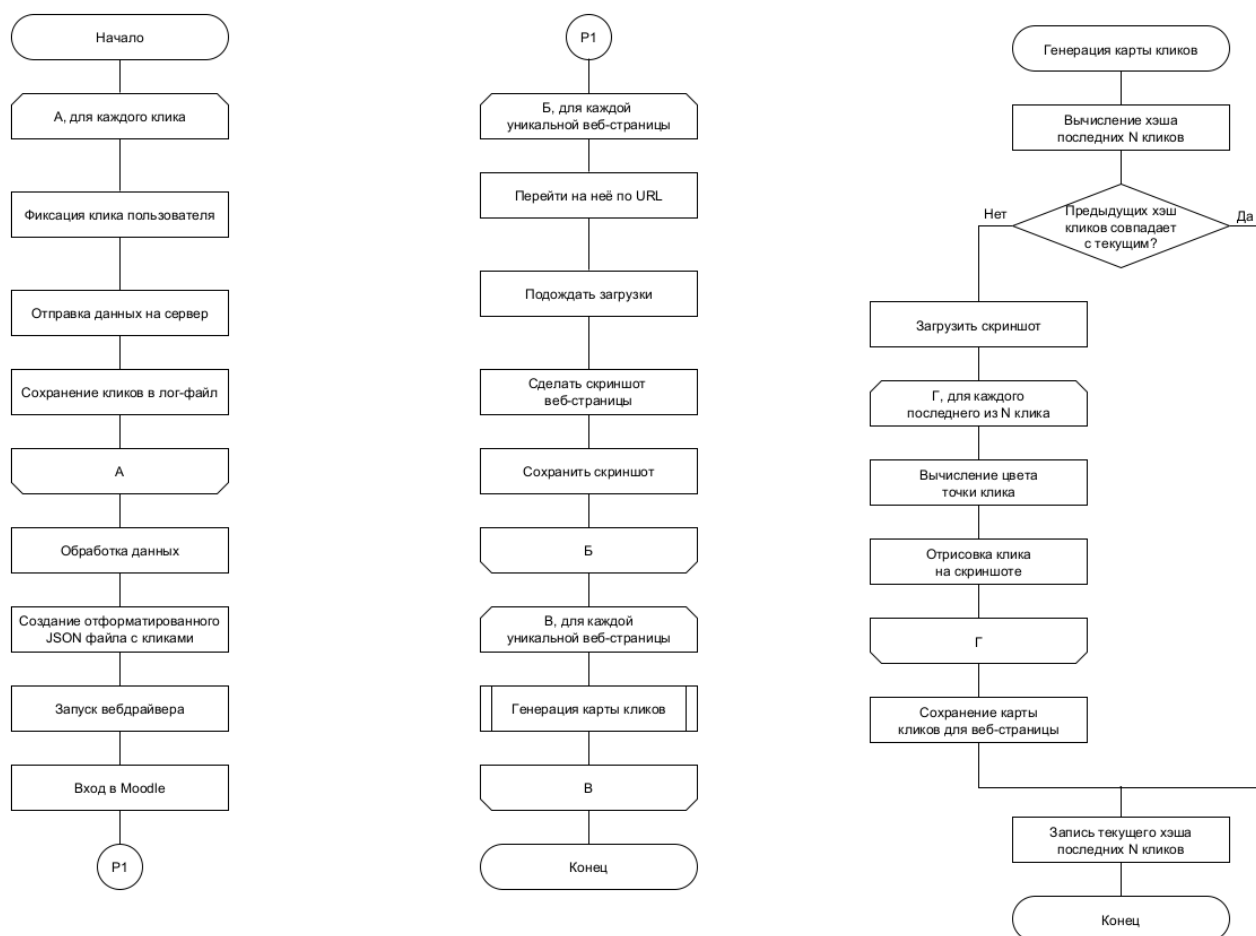


Рисунок 2.3 – Схема разрабатываемого метода

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 осуществляется в пределах той же локальной сети, в которой находится клиент, без передачи информации на внешние серверы;
- 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

Для корректной обработки пользовательского поведения необходимо фиксировать следующие параметры при каждом клике:

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

Каждой тепловой карте соответствует URL-адрес страницы, запись хотя бы об одном клике по которой содержится в источнике данных.

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

2.4 Обработка данных

Обработка данных выполняется поэтапно, с разделением на логические шаги:

- 1) агрегация записей событий кликов;
- 2) трансформация данных в единый формат;
- 3) сопоставление страниц и событий;
- 4) генерация изображений с нанесением маркеров кликов;
- 5) формирование выходного визуального представления.

Модуль отображения организует доступ к сгенерированным тепловым картам с возможностью выбора интересующей страницы и просмотра соответствующих данных.

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.

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

В данном разделе рассматриваются этапы реализации разработанного программного обеспечения, предназначенного для сбора, хранения, обработки и визуализации данных о кликах пользователей в системе Moodle.

Описаны используемые технологии и программные средства, приведена структура системы и модулей.

Каждый модуль сопровождается примерами кода и объяснением применённых технических решений.

3.1 Обоснование выбора технологий и программных средств

Для реализации системы были выбраны следующие технологии и средства:

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

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 верхнеуровневая схема модулей разработанного ПО. Взаимодействие между модулями более подробно представлено на рисунке 3.1.

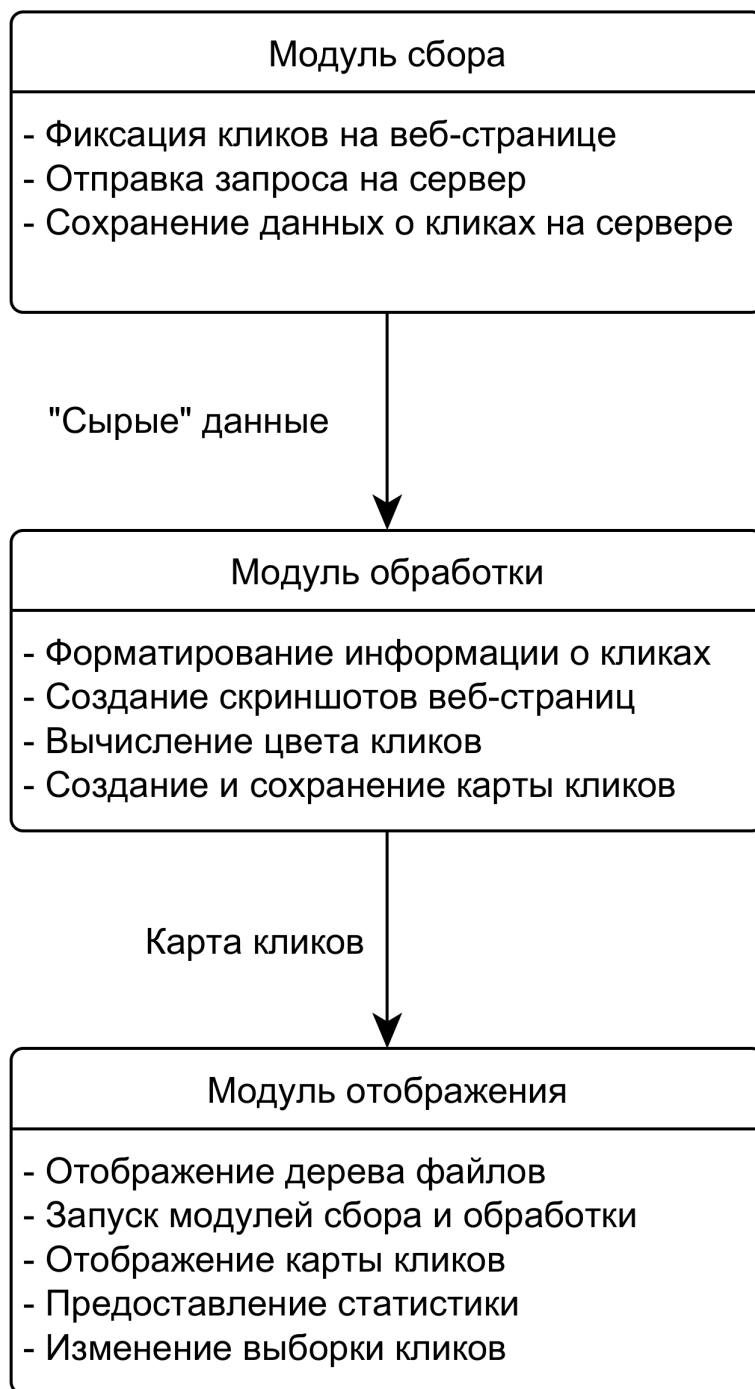


Рисунок 3.1 – Подробная модульная схема

3.3 Сбор информации о кликах

Модуль сбора данных реализуется с использованием кода JavaScript, внедрённого в код Moodle с помощью функции «Дополнительного HTML» в настройках администрирования. Пользователь взаимодействует с веб-страницей, клики фиксируются с помощью JS-обработчика.

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

Листинг 3.1 – Пример записи о клике в исходном массиве

```
1 Array
2 (
3     [relX] => 0.025913621262458
4     [relY] => 0.012054244098443
5     [absX] => 39
6     [absY] => 24
7     [url] => http://192.168.1.102/moodle/admin
8     [title] => Appearance | Administration
9     [pageWidth] => 1505
10    [pageHeight] => 1991
11 )
```

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

Листинг 3.2 – Преобразованная JSON-запись для дальнейшей обработки

```
1 {
2     "relX": 0.025913621262458,
3     "relY": 0.012054244098443,
4     "absX": 39,
5     "absY": 24,
6     "url": "http://192.168.1.102/moodle/admin",
7     "title": "Appearance | Administration",
8     "pageWidth": 1505,
9     "pageHeight": 1991
10 }
```

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 3.3 – 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
7         // Рассчитываем нормализованные координаты (0—1)
8         const relX = event.pageX / scrollWidth;
9         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 3.4 – 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 3.5 – Чтение и декодирование данных

```
1 $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 добавляется в конец файла, что обеспечивает накопление истории всех кликов пользователей.

Листинг 3.6 – 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.";
7 }
```

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.

В листинге 3.7 приведён ключевой фрагмент кода, отвечающий за чтение лог-файла и преобразование его содержимого:

Листинг 3.7 – 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 в листинге 3.8 выполняется попытка автоматически привести значения к числовому типу или булевому.

Листинг 3.8 – 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 3.9 – 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, представленной в листинге 3.10.

Листинг 3.10 – Вывод количества записей

```
1 print(f"Успешно обработано {len(entries)} записей. Результат сохранён 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 [35].

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.

В листинге 3.11 представлен пример функции, которая отвечает за создание скриншота всей страницы:

Листинг 3.11 – 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`, предназначенная для очистки частей URL с целью получения валидных имён папок и файлов, представлена в листинге 3.12.

Листинг 3.12 – Очистка строки для файловой системы

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

Блок кода, автоматизирующий процесс входа в систему Moodle, представлен в листинге 3.13.

Листинг 3.13 – 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 (одионочные клики) к красному (наиболее плотные области);
- наложение слоя тепловой карты на исходный скриншот;
- сохранение итогового изображения в директорию с сохранением структуры URL.

Функция, используемая для очистки частей URL и формирования корректных путей, представлена в листинге 3.14.

Листинг 3.14 – 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, принимающая параметры для путей к файлу с кликами, скриншотам, папке сохранения и числу последних кликов, представлена в листинге 3.15.

Листинг 3.15 – 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 построение двумерного массива плотности $D(x, y)$, в который добавляются значения плотности с гауссовым весом от каждого клика.

Пусть

- (x_k, y_k) — координаты k -го клика;
- r_k — радиус влияния, определяемый по формуле:

$$r_k = \min(r_0 + 5 \cdot \sqrt{n_k}, r_{\max}), \quad (3.1)$$

где

- r_0 — базовый радиус;
- n_k — количество повторных кликов в точке (x_k, y_k) ;
- r_{\max} — максимально допустимый радиус.

Для каждой точки изображения (i, j) вычисляется евклидово расстояние до точки клика:

$$d = \sqrt{(i - x_k)^2 + (j - y_k)^2}. \quad (3.2)$$

Если $d \leq r_k$, плотность $D(i, j)$ увеличивается на вес w , заданный ядром Гаусса:

$$w = \exp\left(-\frac{d^2}{2 \cdot \sigma^2}\right), \quad \sigma = \frac{r_k}{2}. \quad (3.3)$$

Максимальная плотность D_{\max} используется для нормализации цветовой шкалы.

Цвет пикселя определяется функцией `get_color_for_density`, где используются интервалы по значению плотности:

- если $D(i, j) \leq 1$, цвет синий: $(0, 0, 255, 160)$;
- если $1 < D(i, j) < D_{\max}/2$, цвет изменяется от синего к фиолетовому:

$$\text{ratio} = \frac{D(i, j)}{D_{\max}/2}, \quad (3.4)$$

$$\text{color} = (255 \cdot \text{ratio}, 0, 255 \cdot (1 - \text{ratio}), 180); \quad (3.5)$$

- если $D(i, j) \geq D_{\max}/2$, цвет переходит от фиолетового к красному:

$$\text{ratio} = \frac{D(i, j) - D_{\max}/2}{D_{\max}/2}, \quad (3.6)$$

$$\text{color} = (255, 0, 255 \cdot (1 - \text{ratio}), 200). \quad (3.7)$$

Таким образом, итоговая тепловая карта визуализирует распределение плотности кликов, отражая как одиночные события, так и области высокой активности пользователей.

Таким образом, итоговая тепловая карта визуализирует распределение плотности кликов, отражая как одиночные события, так и области высокой активности пользователей.

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.

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

Верхняя панель включает кнопки «Раскрыть всё», «Свернуть всё», чек-бокс скрывания queгу-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. Кнопка «Обновить» запускает последовательно все три скрипта, обновляет дерево файлов и текущее изображение, а также обновляет статистику кликов.

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.

Листинг 3.16 – 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.

Код обработки выбора узла дерева отображает соответствующее изображение в центре с автоматическим масштабированием и приведена на листинге 3.17.

Листинг 3.17 – 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.

3.10 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 3.2.



Рисунок 3.2 – 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.

Ключевой фрагмент этого скрипта представлен в листинге 3.18.

Листинг 3.18 – 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()
```

Блок, представленный в листинге 3.19 запускает экземпляр браузера Chrome в фоновом режиме и осуществляет вход от имени администратора.

Листинг 3.19 – 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)
```

Фрагмент в листинге 3.20 запускает последовательность событий, имитирующих пользовательский клик мыши по элементам на странице. Координаты выбираются случайным образом в пределах доступной области.

Листинг 3.20 – 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.

3.11 Примеры работы

На рисунках 3.3 и 3.4 приведены примеры работы системы, включая скриншоты веб-страниц, поверх которых наложена тепловая карта кликов пользователей.

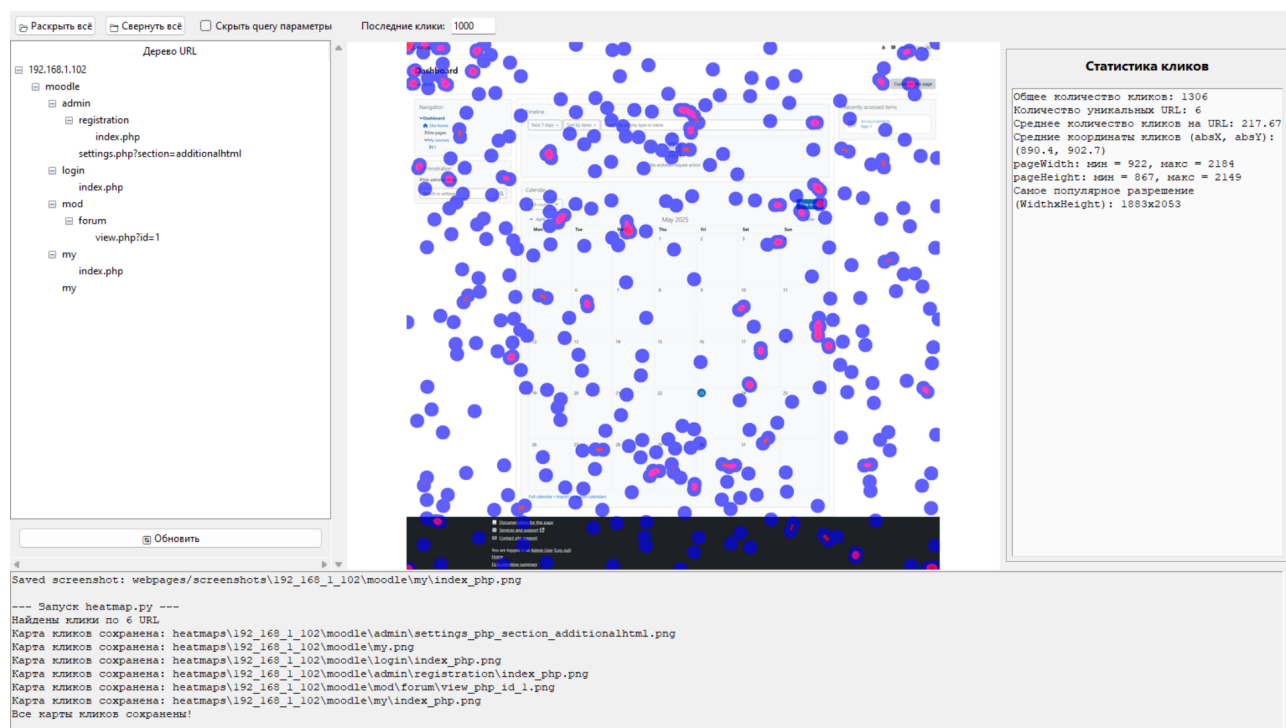


Рисунок 3.3 – Тепловая карта, 1000 кликов

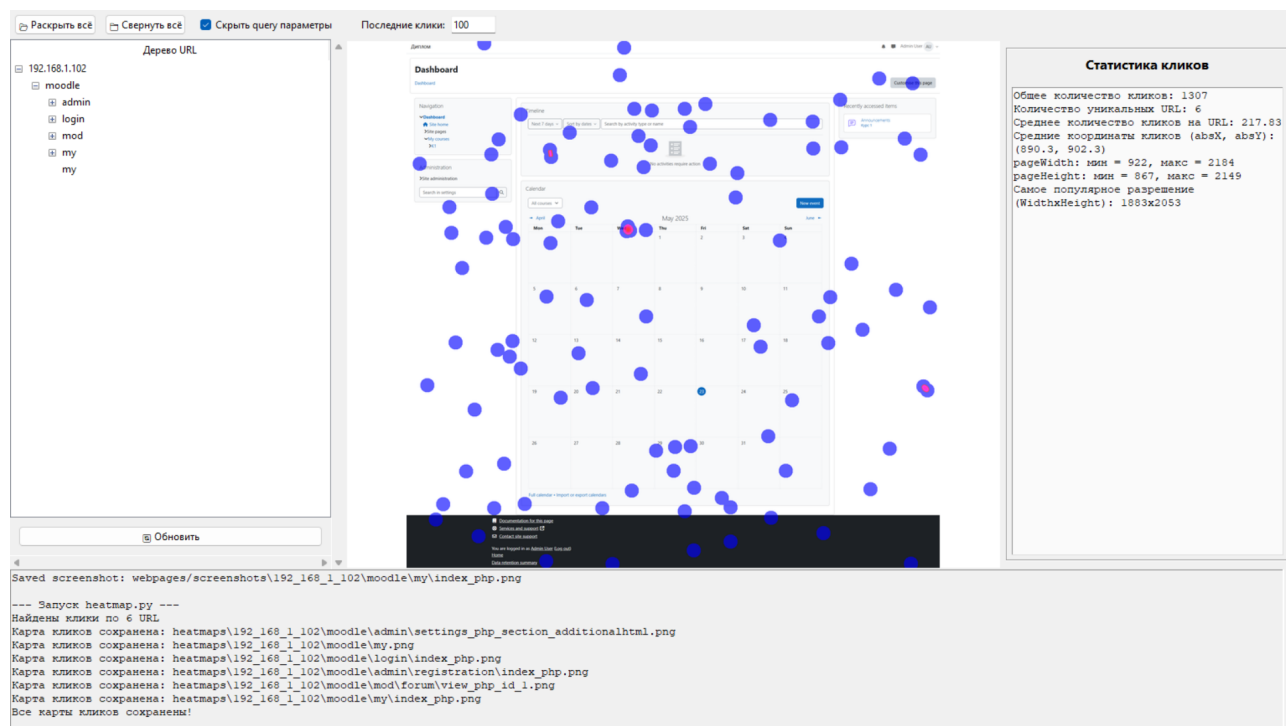


Рисунок 3.4 – Тепловая карта, 100 кликов

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

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.

4.1 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.

4.2 Результаты исследования

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 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 (ОЗУ) — усреднённое значение объёма используемой памяти в процессе выполнения алгоритма, измеряется в мегабайтах (Мбайт);
- пиковое потребление оперативной памяти (ОЗУ) — максимальный объём памяти, занятый в ходе работы метода, измеряется в мегабайтах (Мбайт);
- средняя загрузка процессора (ЦПУ) — усреднённый процент использования процессора во время выполнения обработки.

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

Клики, шт.	Время	ОЗУ, среднее	ОЗУ, пиковое	ЦПУ, среднее
0	1.07	34.82	62.26	31.63
1000	5.17	134.88	177.19	14.42
2000	8.21	168.17	214.81	32.94
3000	11.41	183.50	224.07	35.75
4000	14.09	195.82	236.04	36.06
5000	16.98	206.87	245.30	29.66
6000	19.29	211.93	247.02	34.90
7000	22.20	217.49	262.30	39.17
8000	24.64	222.01	252.75	33.99
9000	27.32	225.30	254.36	37.75
10000	30.06	228.58	256.04	31.11

На рисунке 4.1 представлена зависимость времени обработки от количества кликов.

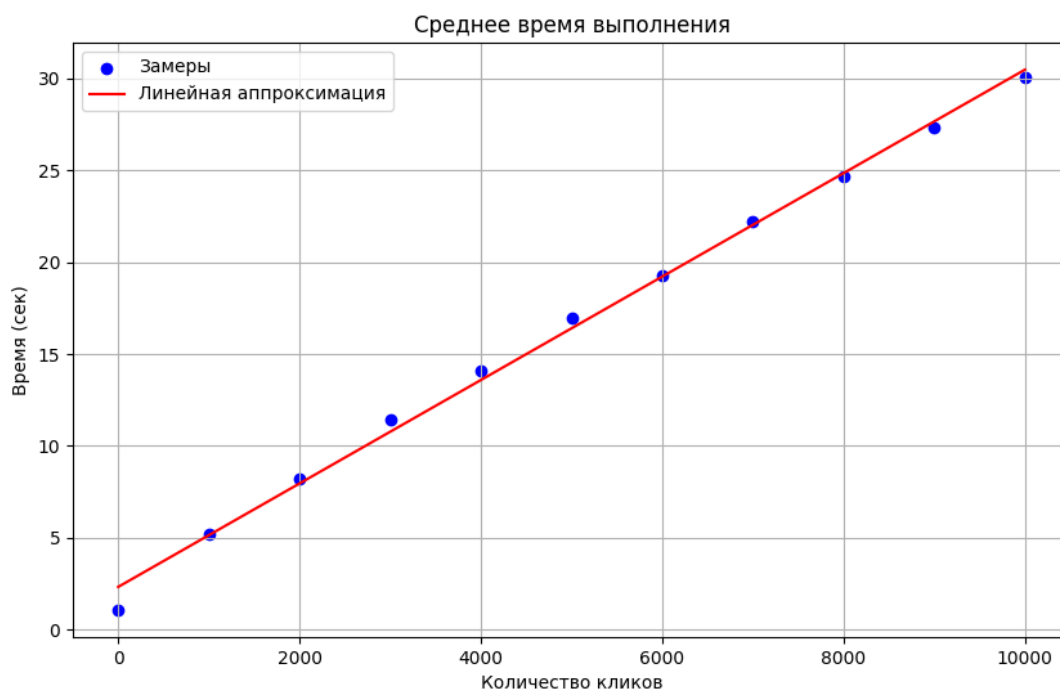


Рисунок 4.1 – Время генерации тепловых карт от количества кликов

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 аппроксимировать.

На рисунке 4.2 и 4.3 приведены графики среднего и максимального потребления оперативной памяти в зависимости от количества обрабатываемых кликов.

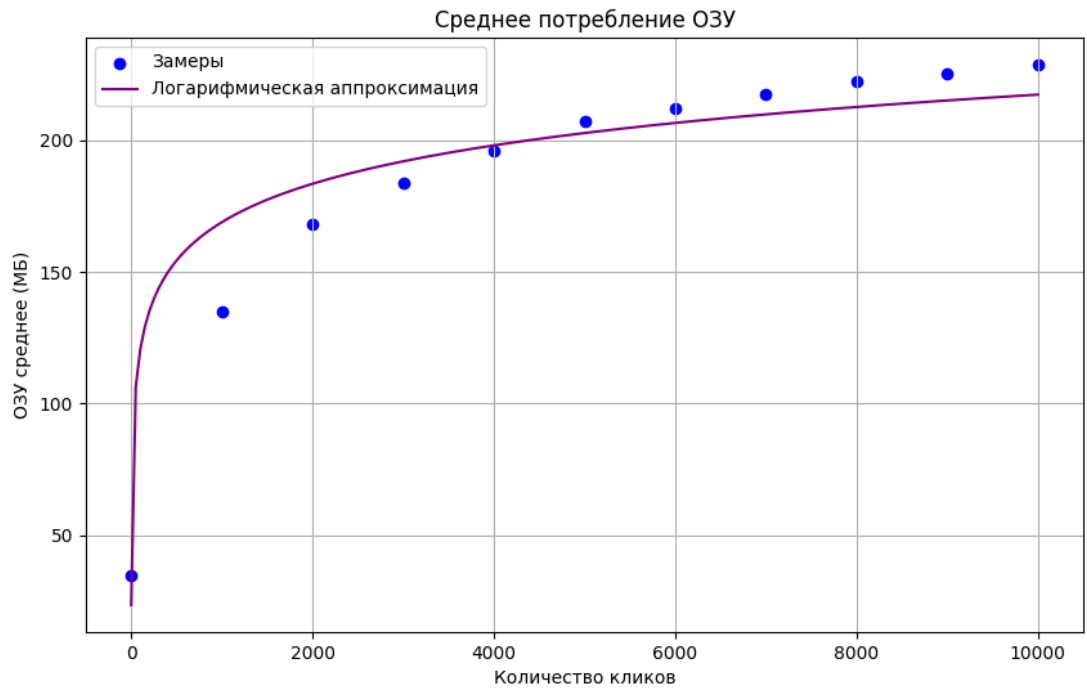


Рисунок 4.2 – Среднее потребление оперативной памяти от количества кликов

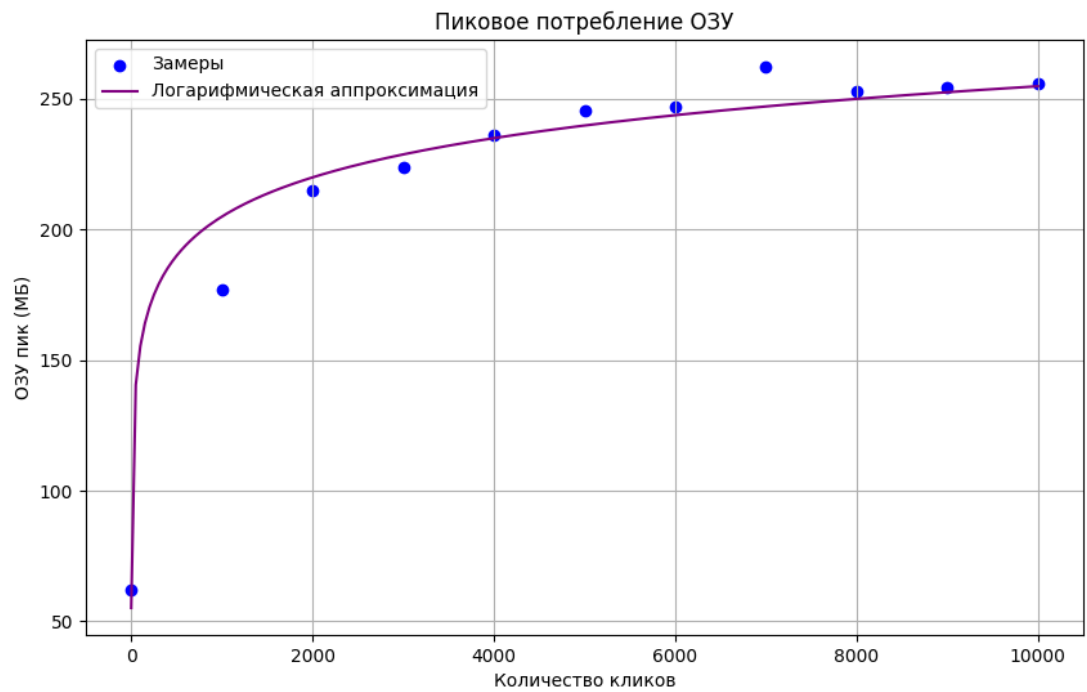


Рисунок 4.3 – Пиковое потребление оперативной памяти от количества кликов

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 ОЗУ 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.

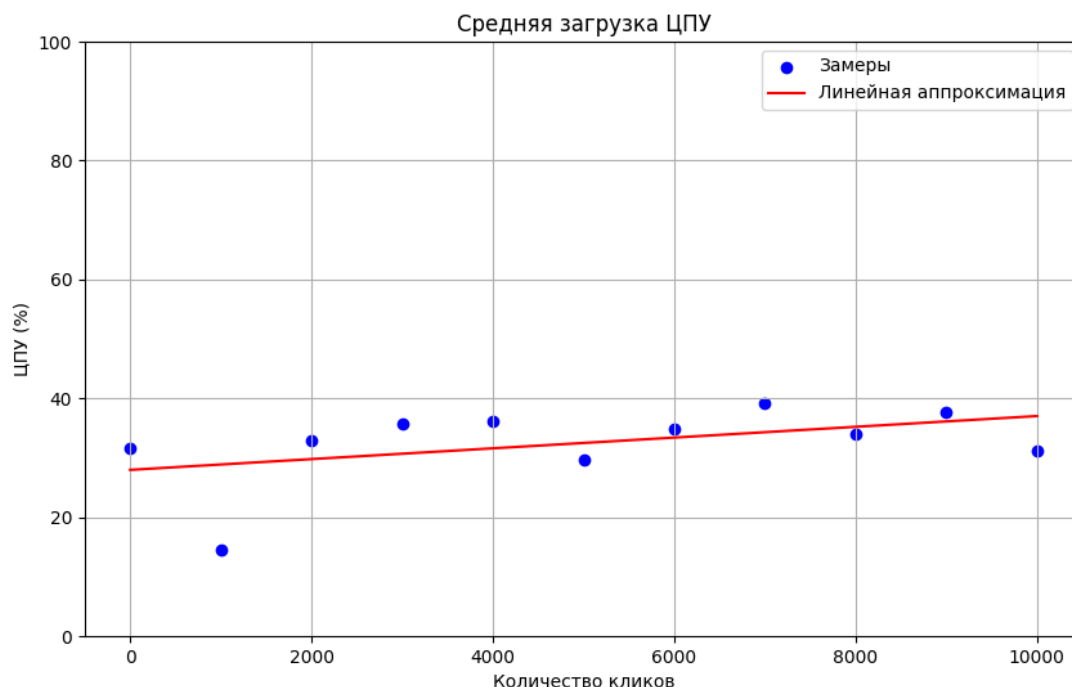


Рисунок 4.4 – Средняя загрузка процессора от количества кликов

График загрузки ЦПУ, представленный на рисунке 4.4, показывает относительно стабильный уровень средней загрузки процессора на протяжении всего диапазона исследуемых объёмов данных и принимает линейный характер.

При этом средняя загрузка ЦПУ ниже 40% свидетельствует о том, что процессор не является главным ограничивающим фактором в данном приложении.

4.4 Инструменты мониторинга и профилирования

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

- модуль `time` [26] языка Python для измерения времени генерации данных и общего времени выполнения;
- библиотека `psutil` [36] для мониторинга использования оперативной памяти (ОЗУ) и загрузки ЦПУ 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 Выводы

В ходе исследования установлено, что разработанный метод демонстрирует:

- линейную зависимость времени обработки от количества обрабатываемых кликов;
- логарифмический рост среднего и пикового потребления оперативной памяти при увеличении объёма данных;
- линейную зависимость средней загрузки процессора 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. Полученная зависимость потребления памяти и умеренная загрузка ЦПУ свидетельствуют об 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 выполнены 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 достигнута.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

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

13. Ефремова, Т. Ф. Толковый словарь русского языка. [Электронный ресурс]. — Режим доступа — 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. [Электронный ресурс]. — Режим доступа — URL: <http://vslovar.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.* Социологический опрос как метод маркетингового исследования. — 2007.
18. Социология : энциклопедия. — 2003.
19. *Щвец Е. В.* Интервью в системе методов опроса. — 2008.
20. *Некрасов С. И.* Сравнение результатов онлайн-и оффлайн-опросов (на примере анкет разной сложности). — 2011.
21. *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.
22. *Прохорова А. М.* Роль методов анализа и прогнозирования поведения пользователей на образовательном портале. — 2017.
23. *Кушмар С. Е., Пилецкий И. И.* Некоторые аналитические методы анализа поведения пользователей сайта. — 2015.
24. Яндекс. Документация Яндекс.Метрики. [Электронный ресурс]. — Режим доступа — URL: <https://yandex.ru/support/metrika/ru/> — (дата обращения: 26.05.2025).
25. Clarify. Microsoft Clarify Documentation. [Электронный ресурс]. — Режим доступа — URL: <https://learn.microsoft.com/en-us/clarify/> — (дата обращения: 26.05.2025).

26. The Python Software Foundation. time — Time access and conversions. [Электронный ресурс]. — Режим доступа — URL: <https://docs.python.org/3/library/time.html> — (дата обращения: 26.05.2025).
27. *Жеребило Т. В.* Словарь лингвистических терминов. — ООО «Пилигрим», 2010.
28. Ubuntu. Ubuntu 22.04.5 LTS (Jammy Jellyfish). [Электронный ресурс]. — Режим доступа — URL: <https://releases.ubuntu.com/22.04.5/> — (дата обращения: 26.05.2025).
29. Moodle. Moodle Documentation. [Электронный ресурс]. — Режим доступа — URL: https://docs.moodle.org/500/en/Main_page — (дата обращения: 26.05.2025).
30. The PHP Documentation Group. Руководство по PHP. [Электронный ресурс]. — Режим доступа — URL: <https://www.php.net/manual/ru/> — (дата обращения: 26.05.2025).
31. The Apache Software Foundation. Apache HTTP Server Version 2.4 Documentation. [Электронный ресурс]. — Режим доступа — URL: <https://httpd.apache.org/docs/current/> — (дата обращения: 26.05.2025).
32. The PostgreSQL Global Development Group. PostgreSQL 14.17 Documentation. [Электронный ресурс]. — Режим доступа — URL: <https://www.postgresql.org/docs/14/index.html> — (дата обращения: 26.05.2025).
33. The Python Software Foundation. Python 3.11.12 documentation. [Электронный ресурс]. — Режим доступа — URL: <https://docs.python.org/3.11/index.html> — (дата обращения: 26.05.2025).
34. Mozilla Foundation. JavaScript | MDN. [Электронный ресурс]. — Режим доступа — URL: <https://developer.mozilla.org/en-US/docs/Web/JavaScript> — (дата обращения: 26.05.2025).
35. Selenium. The Selenium Browser Automation Project. [Электронный ресурс]. — Режим доступа — URL: <https://www.selenium.dev/documentation/> — (дата обращения: 26.05.2025).

36. Giampaolo Rodola. psutil documentation. [Электронный ресурс]. — Режим доступа — URL: <https://psutil.readthedocs.io/en/latest/index.html> — (дата обращения: 26.05.2025).

ПРИЛОЖЕНИЕ А

Программный код с реализацией разработанного метода содержится на цифровом носителе.

ПРИЛОЖЕНИЕ Б

Презентация к научно-исследовательской работе состоит из 14 слайдов.