**РОСЖЕЛДОР**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**ВЫСШЕГО ОБРАЗОВАНИЯ**

**СИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**ПУТЕЙ СООБЩЕНИЯ (СГУПС)**

|  |  |  |  |
| --- | --- | --- | --- |
| **К защите:** |  |  | |
| **Заведующий кафедрой** | **Информационные** | |
| **технологии транспорта** | | |
|  | д-р техн. наук, проф. | |
|  |  | В. И. Хабаров | |
| *подпись* |  | *инициалы, фамилия* | |
|  |  |  | |
| *дата* |  |  | |

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Тема:** | Защита web-ресурсов от кибер-атак и устранение уязвимостей | | | | | |
|  |  | | | | | |
|  | |  | БР.БИСТ.22.2022 |  |  |
|  | |  | *шифр документа* |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | **Выполнил** |  |  |  | **Руководитель** |
|  |  | С.А. Сыромукова |  |  |  | канд. техн. наук, доц.  С.А. Диденко |
| *подпись* |  | *инициалы, фамилия* |  | *подпись* |  | *инициалы, фамилия* |
|  |  |  |  |  |  |  |
| *дата* |  |  |  | *дата* |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Консультанты по разделам** |  |  |  |  |
| 1 |  |  |  |  |
|  |  |  |  |  |
| 2 |  |  |  |  |
|  |  |  |  |  |
| 3 |  |  |  |  |
|  |  |  |  |  |
| Нормоконтролер работы |  |  |  | ст. преп.  Т. А. Распопина |
|  |  | *подпись* |  |  |
|  |  |  |  |  |
|  |  | *дата* |  |  |

**2022 г.**

**СИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**ПУТЕЙ СООБЩЕНИЯ (СГУПС)**

Факультет: Бизнес-информатики

Кафедра: Информационные технологии транспорта

Направление: 09.03.02 «Информационные системы и технологии»

Профиль: Интеллектуальные транспортные системы

|  |  |
| --- | --- |
|  | ***УТВЕРЖДАЮ****: зав. кафедрой «Информационные технологии транспорта»*  д-р техн. наук, проф.  В. И. Хабаров |
|  | *«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_ 2022 г.* |

**З А Д А Н И Е**

**на выполнение выпускной квалификационной работы**

|  |  |  |  |
| --- | --- | --- | --- |
| студенту | Сыромуковой Софье Алексеевне | | |
|  |  | | |
| 1. Тема «Защита web-ресурсов от кибер-атак и устранение уязвимостей» утверждена приказом № 203/c от «30» мая 2022 г. | | | |
| 2. Задание выдано «12» мая 2022 г. | | | |
| 3. Срок сдачи законченной работы на кафедру «17» июня 2022 г. | | | |
| 4. Исходные данные: данные, полученные в ходе прохождения преддипломной практики | | | |
| 5. Содержание расчетно-пояснительной записки | | | |
| Наименование разделов и вопросов | | Примерное количество страниц | График (сроки) выполнения |
| Введение | | 1 | 12.05.2022 |
| Аналитическое исследование | | 5 | 18.05.2022 |
| Проектирование информационной системы | | 13 | 25.05.2022 |
| Создание приложения | | 37 | 07.06.2022 |
| Заключение | | 1 | 09.06.2022 |

6. Содержание и объемы графической части

|  |  |  |
| --- | --- | --- |
| Наименование графического документа (чертежа, схемы, графика) | Количество  листов  формата А1 | График  (сроки)  выполнения |
| Презентация PowerPoint | 22 | 07.05.2022 |

7. Консультанты по разделам

|  |  |  |
| --- | --- | --- |
| Наименование  раздела | Фамилия, И. О.  консультанта | Подпись консультанта,  дата выдачи задания |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| Руководитель |  | С.А. Диденко |
|  | *(подпись, фамилия, И.О.)* |  |
| Задание к использованию принял |  | С.А. Сыромукова |
|  | *(подпись студента)* |  |

УДК 004.41

**АННОТАЦИЯ**

В работе 50 страниц, 21 рисунков, 3 таблицы, 15 источников.

Ключевые слова: *веб-ресурс, уязвимость, информационная безопасность, бэкдор, инъекции, межсайтовый скриптинг, сканирование.*

Предметная область – защита веб-ресурсов от уязвимостей. В качестве информационной системы был создан сайт банка. Веб сайт работает с сервером, откуда берет информацию о клиентах.

Работа состоит из трёх разделов. В первом разделе представлена теоретическая информацию о уязвимостях, а также методологии, которые использованы при моделировании бизнес-процессов. Во втором разделе представлена информация о разработанном ПО и инструментах разработки. В третьем рассмотрен разработанный прототип ИС и мероприятия по его защите. В результате исследования был разработан защищенный сайт банка.

**ABSTRACT**

The work contains 50 pages, 21 figures, 3 tables, 15 sources.

Keywords: *web resource, vulnerability, information security, backdoor, injection, cross-site scripting, scanning.*

The subject area is the protection of web resources from vulnerabilities. The bank's website was created as an information system. The website works with a server, from where it takes information about clients.

The work consists of three sections. The first section provides theoretical information about vulnerabilities and methodologies that are used in modeling business processes. The second section provides information about the developed software and development tools. In the third, the already developed prototype of the IS and measures for its protection are considered. Result of the research is a secure bank website was developed.

**ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ**

ИС – информационная система.

ПО – программное обеспечение.

ПК – персональный компьютер.

ОС – операционная система.

БД – база данных.

ИБ – информационная безопасность.

Java – объектно-ориентированный язык программирования высокого уровня.

CSS — формальный язык описания внешнего вида документа, написанного с использованием языка разметки.

HTML — стандартизированный язык разметки документов для просмотра веб-страниц в браузере.

PHP — скриптовый язык общего назначения, интенсивно применяемый для разработки веб-приложений.

Php MyAdmin — веб-приложение с открытым кодом, написанное на языке PHP и представляющее собой веб-интерфейс для администрирования СУБД MySQL.

SQL (Structured Query Language) – декларативный язык программирования, применяемый для управления данными в реляционной базе данных.

URL (Uniform Resource Locator) – стандарт записи ссылок на объекты в Интернете.

ER (Entity Relationship) – модель «сущность-связь».

**СОДЕРЖАНИЕ**

[Введение 6](#_Toc106292465)

[1 Анализ предметной области 9](#_Toc106292466)

[1.1 Обзор существующих уязвимостей и описание защиты от них 10](#_Toc106292467)

[1.2 Обзор существующих инструментов выявления уязвимостей 15](#_Toc106292468)

[1.3 Основные понятия моделирования бизнес-процессов 16](#_Toc106292469)

[1.4 Методология объектного проектирования и анализа с применением языка UML 19](#_Toc106292470)

[1.5 Виды диаграмм, использующих при моделировании бизнес-процессов 21](#_Toc106292471)

[1.6 Моделирование бизнес-процессов предметной области 22](#_Toc106292472)

[1.7 Выводы по разделу 27](#_Toc106292473)

[2 Проектирование прототипа информационной системы 28](#_Toc106292474)

[2.1 Обоснование выбора языка программирования 28](#_Toc106292475)

[2.2 Среда разработки (язык программирования) 30](#_Toc106292476)

[2.3 Модель данных (ER-модель) 34](#_Toc106292477)

[2.4 Выводы по разделу 36](#_Toc106292478)

[3 Информационная система 37](#_Toc106292479)

[3.1 Описание разработки пользовательского интерфейса прототипа информационной системы 37](#_Toc106292480)

[3.2 Описание функционала разработанного прототипа информационной системы 42](#_Toc106292481)

[3.3 Авторизация пользователя 43](#_Toc106292482)

[3.4 Использование личного кабинета пользователем 43](#_Toc106292483)

[3.5 Защита прототипа ИС от атак и устранение в ней уязвимостей 43](#_Toc106292484)

[3.6 Выводы по разделу 46](#_Toc106292485)

[Заключение 48](#_Toc106292486)

[Список использованных источников 49](#_Toc106292487)

**ВВЕДЕНИЕ**

На сегодняшний день проблема безопасности информационных ресурсов обстоит особенно остро, поскольку существует огромное количество уязвимостей и атак, которые могут нанести серьезный вред и стать причиной убытков. Поскольку особое распространение получила такая технология как компьютерная сеть Интернет, которая крепко вошла во всевозможные сферы деятельности человека, особенное значение приобрела информационная безопасность данных, которая имеет различные степени открытости доступ. И особенно это касается web – ресурсов, которые тесно связаны воедино с этой компьютерной сетью, которая таит в себе множество опасностей, а в неумелых руках может являться рассадником различных угроз. Киберпространство не имеет каких-то границ и рамок, а государство имеет минимальное влияние на данное явление, по этой причине человек, который осуществляет несанкционированный доступ к информации, может находиться в любой точке мира. Ежедневно совершаются взломы сайтов и приложений, целых информационных систем, принадлежащих компаниям – от мелких контор до компаний – гигантов. Злоумышленники получают доступ к конфиденциальной и личной информации и используют её в корыстных целях. Периодически они находят новые способы достижения своих целей, находят новые уязвимости в системах – так называемые бэкдоры.

Поэтому была сформирована отдельная сфера области информационных технологии – информационная безопасность, которая включает в себя ряд способ и мероприятий по защите информации, а также различные существующие уязвимости и методы по их устранению. Обычные рядовые пользователи компьютеров на сегодняшний день используют в качестве защиты антивирусы, которых рынок может предложить огромное количество. К сожалению, данная мера не является панацей, поскольку вредоносные программы появляется зачастую слишком быстро и в больших количествах так, что разработчики данного программного обеспечения не успевают обновить свои базы. А некоторые трояны или малвари умеют встраиваться в систему, заражая уже и файлы операционный системы и антивирус, действуя скрытно, как правило, обнаружить их в системе достаточно сложно, почти невозможно и помогает только полная переустановка ОС. Крупные компании нанимают на работу специалистов по защите информации, а особо крупные компании как государственные, так и коммерческие могут иметь даже целые отделы таких специалистов.

Большинство современных информационных систем представляют собой web – сайты, именно поэтому их безопасности стоит уделить особо внимание. Например, интернет-магазины являются настоящим лакомым кусочком для хакеров, так как взлом может открыть злоумышленникам доступ к личным данным пользователей и их платежной информации. Банковские системы имеют свои сайты, которые содержат в себе конфиденциальную информацию о их клиентах, хотя это одни из самых защищенных систем. У взлома таких ресурсов есть одна особенность – он не персонифицирован, а автоматизирован, то есть производится массово при помощи специальных программ, если речь не идет о какой-то сложно защищенной системе с особым алгоритмом шифрования и командой специалистов по информационной безопасности. Полученная в результате взлома информация продается третьим лицам или используется в своих целях взломщиком.

Вопреки тому факту, что периодически появляются новых средств для защиты, количество атак на web -ресурсы растет с каждым годом, они всё также уязвимы для хакеров. Не смог противостоять взлому даже настолько, казалось, защищённый Twitter. Злоумышленники интересуются способами взлома web -приложений в различных целях – получить какого-либо ценный ресурс без оплаты, кража аккаунта в игре, изменение данных, иногда цели и вовсе весьма неожиданны. Например, в эпоху майнинга мошенники научились вставлять специальный скрипт в код web - страниц и майнить на компьютерах посетителей данного ресурса. Причиной тому, что разработчики не всегда уделяют должного внимания полной защите и безопасности созданных сайтов, хотя она и играет очень важную роль, потому что каждый сайт так или иначе может быть подвергнут атаке.

По всем вышеперечисленным причинам проблема наиболее актуальна для сфер, которые должны быть максимально защищены: финансовые, налоговые и государственные структуры. Поэтому в качестве предметной области была выбрана информационная система в банковской сфере.

Информационная система банковской сферы - это система, считывающая, хранящая и обрабатывающая данные о транзакциях клиентах банка. Такая информационная система используется для счета сумм, которые хранятся на счетах и картах клиента. Система хранит в себе все операции, которые были произведены клиентом со счетом за весь период существования счета.

*Объектом исследования* различные виды атак и уязвимостей, которые должны быть устранены в разрабатываемом прототипе информационной системе.

*Предмет исследования* - прототип информационной системы банковской сферы, представленный в виде веб-ресурса.

*Цель работы* - создание максимально защищённой информационной системы в банковской сфере.

Для достижения поставленной цели необходимо выполнить следующие задачи:

* провести анализ предметной области;
* построить модели бизнес-процессов для выбранной предметной области;
* выбрать и обосновать инструменты разработки приложения;
* реализовать, настроить и протестировать web-приложение;
* разработать руководство пользователя приложением.

# **1 Анализ предметной области**

Анализ предметной области или этап концептуального проектирования является первым шагом при проектировании и реализации любой информационной системы. Он включает в себя сбор, анализ и редактирование требований к используемым данным, позволяет сформулировать ключевые требования заказчика к проектируемой информационной системе, проанализировать функции системы для автоматизации и улучшения. Также происходит анализ пользовательских запросов, используемых информационных объектов и их характеристик, которые составляют содержание системы. Анализ предметной области не зависит от программной и технической сред, он заканчивается построением концептуальной схемы.

Банковская система представляет из себя определенную совокупность, состоящую из различных видов национальных банков и кредитных организаций, которая функционирует в строгих рамках общего денежно-кредитного механизма. Данная система включает в себя ряд организаций, таких как центральный банк, сеть коммерческих банков и других кредитно-расчётных центров.

В рамках данной предметной области необходимо разработать в информационной системе следующие действия, которые бы полностью отражали работу любого стандартного банка:

* наличие различного рода денежных операций со счетами клиентов;
* возможность создания счетов и выпуска банковских карт для клиентов;
* возможность оставить заявку на различные банковские услуги: кредит, ипотека, выпуск карты и так далее;
* манипулирование денежными средствами в системе;
* создание личного кабинета для каждого клиента.
  1. **Обзор существующих уязвимостей и описание защиты от них**

Безопасность сайта - ключевая составляющая часть разработки сайта, от которой зависит его успешное существование, поэтому оно требует бдительности во всех аспектах дизайна и использования сайта. Написание сайта с нуля намного снижает вероятность взлома, нежели создание сайта с помощью CMS - системы управления контентом сайта или движком сайта. К ним относятся WordPress, Joomla!, Drupal, Wix, Shopify, Squarespace и тому подобные.

DoS и DDoS атака - это атака с целью доведения системы до полного отказа. Добросовестный пользователь, который не нарушил никаких правил, не может войти в систему, поскольку она просто падает. Различие DoS и DDoS в том, что вторые выполняются с нескольких устройств.

Методы защиты от данной атаки делятся на пассивные и активные:

* предотвращение - проведение профилактических мер причин, заставляющих третьих лиц производить атаки;
* ответные меры, то есть проведение ряда мер, которые повлияют на источники или организатора атак, используя различные методы;
* специализированное программное и аппаратное обеспечение.

На сегодняшний момент существует множество программного и аппаратного обеспечения для защиты от DDoS-атак. Далее были более подробно рассмотрены все эффективные методы защиты:

* фильтрация (или в некоторых случаях вовсе блокировка) трафика атакующей системы, помогает снизить или ликвидировать атаку. Существует два способа: маршрутизация по спискам ACL и использование межсетевых экранов;
* обратный DDoS - это перенаправление трафика на атакующего, если достаточно мощности. В некоторых случаях может вывести оборудование, которое атакует;
* устранение уязвимостей и наращивание ресурсов нацелено на устранение ошибок в системах или службах, не работает против флуд – атак. Второй способ не дает абсолютной защиты, но с помощью программного и аппаратного обеспечения можно удачно справиться с атакой;
* построение распределенных систем - распределенные системы или её дублирование на другой территории позволяют справиться практически с любым типом атак при правильном архитектурном проектировании;
* уклонение - разделение атаки на другие ресурсы, которые расположены на общей площадке;
* мониторинг - установка системы, которая производит мониторинг. В случаи атаки происходит оповещения, которое позволяет вовремя среагировать и принять меры;
* покупка сервиса по защите от DDoS-aтак. Один из самых простых методов.

Также во внимание были приняты инъекции нескольких видов. SQL-инъекция - это атака, при которой злоумышленником производится вставка вредоносного кода в строки, передающиеся на сервер СУБД для синтаксического анализа и выполнения [1]. Устраняется с помощью связываемых переменных (prepared statements) или экранизация спецсимволов слэшами.

CRLF (внедрение символа возврата каретки (англ. Carriage Return, CR) и перевода строки (англ. Line Feed, LF) в запрос к серверу (например, с помощью протокола HTTP). Устранение: санитизация данных.

Брутфорс - достаточно старый способ взлома, который подразумевает собой рандомный подбор паролей. Зачастую происходит с помощью словарей и требует большое количество времени и ресурсов системы. Устраняется достаточно просто, стоит ввести жесткие ограничения на попытки авторизации на единицу времени, так называемую, капчу.

Межсайтовый скриптинг - уязвимость, что подразумевает под собой внедрение кода злоумышленника в код страницы или приложения, что влечет за собой установку вредоносного ПО напрямую в браузер жертвы. Устраняется с помощью кодирования данных (или дополнительная валидация), а также установкой инструментов для сканирования данной уязвимости.

SSRF представляет собой подделку запроса на стороне сервера. Стоит заметить, что данную уязвимость используют в совокупности с DoS-атакой, чтобы замаскировать источник нападения.

В качестве устранения необходимо ограничить доступ к внутренней инфраструктуре для потенциально подверженных SSRF серверов:

* ввести межсервисную аутентификацию;
* использовать внешний прокси;
* ограничить поддерживаемые схемы;
* отключить поддержку перенаправлений или проверять каждый шаг;
* валидировать доменные имена;
* разобраться, как используемая библиотека обрабатывает адреса.

IDOR – с английского Insecure Direct Object Reference или небезопасные прямые ссылки на объекты. Возникает при отсутствии контроля и проверки входных данных пользователей. Метода устранения вытекают из причин появления:

* использование одного из способов контроля доступа, например, ролевая авторизация;
* создание косвенную ссылочную карту, чтобы ограничить злоумышленникам доступ к прямым ссылкам;
* проверять входные данные каждого пользователя и что важно на стороне сервера, а не клиента.

Remote Code Execution или удаленное выполнение кода - это достаточно опасная уязвимостей, поскольку гарантирует успешный взлома сайта и веб приложения.

Различные недочеты в работе или грубые ошибки в самой разработке веб-ресурса приводят к тому, что появляется возможность эксплуатации данной уязвимости. Также причиной могут стать полное отсутствие фильтрации параметров, которые передаются на вход или вход или же наоборот, бездумное использование небезопасных функций или приемов в написании кода. Устранение: санитизация данных, проверка валидности запросов, а также данных в передающих параметрах.

Межсайтовая трассировка (XST) - это уязвимость, использующая метод HTTP TRACE, также называется атакой одним щелчком или обкрытием сеанса, потому что она использует преимущества сеанса пользователя, что был до неё и был проверен. Потому что веб-браузеры автоматически отправляют некоторые типы токенов проверки подлинности при каждом запросе на веб-сайт, такие атаки представляются возможными. В качестве полной гарантии защиты от XST, отключении JavaScript или ActiveX в браузерах, однако, это приведет к большому урезанию функционала, поэтому рекомендации, следующие:

* устранение известных недостатков обхода ограничений домена в браузерах;
* отключение или запрет метода запроса TRACE в производстве и разработке (кроме случаев, когда необходимо) веб-сервера;
* поставщики веб-серверов должны обновить свои пакеты веб-серверов, чтобы отключить TRACE, по дефолту;
* поставщики веб-серверов должны проинформировать своих пользователей о том, как отключить или запретить TRACE на существующих веб-серверах;
* элементы ActiveX, поддерживающие произвольные HTTP-запросы, пометить как небезопасные скрипты по умолчанию.

Clickjacking - это уязвимость, при которой вредоносные ссылки как бы нанесены поверх настоящих интерактивных кнопках. Это приводит к тому, что вводит пользователей в полное заблуждение и тем самым заставляет активировать какое-либо действие, которое выгодны мошеннику, своим щелчком по элементу.

Советы по защите от кликджекинга:

* параметры X-Frame. Это решение является самым эффективным. Вы можете включить HTTP-заголовок X-Frame-Options на все свои веб страницы. Это предотвратит размещение вашего сайта во фрейме;
* перемещение элементов на странице. Злоумышленник не будет знать о текущем расположении элементов, а заражение элементов происходит по начальному расположению элементов;
* одноразовые URL. Это довольно продвинутый метод защиты, значительно усложняет атаку;
* framebuster Javascript. Заключается в проверке кода Javascript на предмет обнаружения - фрамбастинге.

LFI-уязвимости (Local File Inclusion; Включение локальных файлов). Принцип данной уязвимости заключается в опасности того, что она допускает удаленное подключение файлов на сервер через браузер с корыстной целью. Данный бэкдор имеет место быть если корректная обработка входящих данных отсутствует, это дает возможность хакеру проводить манипуляции с входящим трафиком, инжектировать символы типа path traversal и включать другие файлы с веб-сервера.

Устранение:

* фильтрация параметров, передающих данные;
* фильтрация или экранирование нулевого байта;
* проверка валидности запросов, серверного пути.

Remote File Inclusion – это весьма опасная уязвимость наравне с RCE, представляет собой выполнение удаленного файла на сервере. Когда входящие данные в коде сайта не проходят должную проверку, то злоумышленник может выполнить через скрипт на сервере удаленно файл. Главные причины возникновения данной уязвимости: грубые ошибки на стадии разработки сайта, отсутствия фильтрации передающих параметров и проверки серверного пути.

Устранение:

* фильтрация параметров, передающих данные в include(); include\_once() и так далее;
* проверка валидности запросов, серверного пути.

## **1.2 Обзор существующих инструментов выявления уязвимостей**

На сегодняшний день рынок имеет огромное количество различного программного обеспечения для сканирования веб-ресурсов на уязвимости, есть как платные, так и бесплатные. Далее представлены самые популярные из них.

Web Stress Tester - данная программа предназначена для стресс-тест веб-сайт или серверы с 100000 HTTP (S) соединений на машине Windows. То есть иммитиурет DDos атаку. Она проверяет производительность и стабильность любой веб-сервер и прокси-сервер с нашим SSL / TLS с поддержкой «Fastream веб Стресс тестер» инструмент, доступный с OpenSSL бинарных файлов здесь.

SQLmap - инструмент для тестирования веб-ресурсов на SQL -инъекции с открытым исходным кодом и утилита работает на большинстве платформ и написана на Python'e. Позволяет автоматизировать работу обнаружения и эксплуатации SQL Инъекций. Поддерживает такие базы данных, как: MySQL, MS SQL, PostgerSQL, Oracle.

IBM Security AppScan Standard - это программа, которая помогает организациям снизить вероятность атак на веб-приложения и дорогостоящих утечек данных за счет автоматизации тестирования уязвимостей безопасности приложений. Он обеспечивает широкий охват для сканирования и тестирования широкого спектра уязвимостей безопасности приложений.

Acunetix Web Vulnerability Scanner - данная программа представляет собой сканнер для обнаружения уязвимостей различного рода. Принцип её действия таков: происходит проверка объекта с помощью тестирования, главная цель которого выявить возможные угрозы безопасности, как результат данного теста будет создан подробный отчет, где описано, каким уязвимостям подвержен ресурс. У данной программы большое множество инструментов: различные пентесты, сканирование архитектуры ресурса и большей части контента, глубокое сканирование одностраничных приложений, проверка на внедрение SQL-кода и межсайтовый скриптинг, а также отправка и перехват различных HTTP/HTTPS запросов и анализирование ответов от сервера. Также имеется некоторое количество особенных проверок для одной из самых популярных CMS - Wordpress: безопасность пароля администратора, устойчивость к брутфорсу, проверка плагинов (в базе свыше 1200 известных плагинов), а также обнаружение замаскированных вредоносных модулей.

Wapiti - консольный сканер веб-ресурсов, который базируется на принципе BlackBox тестирования (анализируются не исходники приложения, а ответы сервера на запросы с измененными параметрами). Утилита проводит анализ структуры сайта, поиск доступные сценарии, анализ параметры и по итогу использует свой фаззер (инструмент динамического анализа, необходимый на всех этапах разработки, тестирования и эксплуатации программного обеспечения).

Wapiti работает со следующими типами уязвимостей: File disclosure, всевозможными инъекциями, межсайтовым скриптингом, CRLF и XXE инъекциями.

**1.3 Основные понятия моделирования бизнес-процессов**

Во втором разделе рассмотрены положения бизнес-процессов, их моделирование, а также схематические варианты представления прообраза информационной системы в банковской сфере, которая выполняет ряд функций. Но для начала необходимо ввести основные понятия, относящиеся к бизнес-процессам.

В наши дни в такой сфере как менеджменте образовалось множество терминов, определений, постулатов, правил и притязаний к системе менеджмента качества организаций и предприятий, руководство по достижению устойчивого результата. Всё это влилось в серию международных стандартов ISO 9000, сконцентрировавший мировой опыт по обеспечению качества продукции.

В настоящую бытность бурно развивающегося мира происходят одновременно множество процессов развития, которые оказывают влияние на весь мир, на бизнес и на людей: организациям приходится работать в условиях аморфности внешней среды, где посекундно растет конкуренция и запросы потребителей. Вследствие этого сейчас остро встает вопрос – в силах ли системы управления в организации обеспечить приспосабливаемость к новым условиям и удовлетворить растущие требования и запросы потребителей, обеспечив на этой основе свое развитие? Это стало причиной предпосылки к пересмотру и принятию новой редакции стандартов ISO серии 9000:2000, содержащий рассматриваемые нами понятия.

Бизнес-процесс является логически закаченной совокупностью, что представляет собой цепь взаимозависимых, а также циклических разновидностей работы, вследствие каковых средства компании либо компании задействуются с целью свершения конкретных измеримых итогов.

Термин «бизнес-правила» подразумевают методы осуществлении бизнес-функций в рамках бизнес-процесса, а кроме того свойства, а также требование исполнения бизнес-процесса. Элементы воздействия имеют все шансы быть выполнены людьми либо быть абсолютно автоматизированы. Проблемой любой компании, тяготящейся к модернизированию собственной работы, считается создание подобных бизнес-процессов, какие существовали с успешными результатами, а также содержали только лишь на самом деле необходимые действия.

Моделирование бизнес-процессов представляет из себя некий процесс конструирования модели, который рассмотрен как подобие оригинала, интерпретирующего особенно важные его черты, функции и свойства, сам процесс изучения, анализа системы с точки зрения её функций. Целью моделирования является изложение реального хода бизнес-процессов организации, но сперва стоит предопределить, что именно служит итогом реализации процесса, кем и какие действия выполняются, каков их алгоритм, каково движение документов в ходе реализации процесса, а также какова вероятность неудачного выполнения, и как он может быть расширен или модифицирован в будущем. Это поможет осуществить прозрачность хода бизнес-процессов: как идут дела в компании.

Бизнес-модель может состоять из одной диаграммы, но вследствие сложных и многоаспектных бизнес-процессов в организации их подразумевается некоторое множество. Они состоят из специализированных элементов:

* представлений, которые отражают определенную точку зрения и скрывают малозначимые в данный момент детали;
* диаграмм, которые каждое представление которых состоит из ряда диаграмм, описывающих структурные и динамические аспекты бизнес-процессов;
* объектов и процессов, которые реализуются по определенным бизнес-правилам, совместно выполняют поставленную задачу в пределах одной организационной единицы, нескольких единиц или нескольких организаций.

К целям моделирования также относится выявление проблем организации и поиск путей их решения, понимание целей и задач организации заказчиком и разработчиком модели, подготовка технического задания для дальнейшего составления программного обеспечения для автоматизации определенных процессов, которое должно автоматизировать бизнес-процессы организации.

Используя методы моделирования, проводят реинжиниринг различных процессов на предприятиях. Он позволяет работать над изменениями текущей модели деятельности из вида «AS-IS» (как есть) в перспективную «AS-TO-BE» (как следует быть).

Анализируя вышесказанное, создание моделей бизнес-процессов банковской сферы даст возможность понять принцип движения денежных средств внутри системы, а также поможет в создании личного кабинета. Все это поможет сформулировать требуемый функционал к проектируемому приложению.

**1.4 Методология объектного проектирования и анализа с применением языка UML**

Под методом или методологией моделирования подразумевается алгоритм, состоящий из ряда последовательных действий, которые необходимо выполнить для построения модели. В нее входит сама процедура моделирования и используемая нотация - язык, на котором формулируется модель. Он содержит собственный синтаксис - относительные обозначения разных компонентов, а также принципы их сочетания, а кроме того, семантику – принципы объяснения модификаций, а также их компонентов.

Необходимо выделить наиболее полуполярные и актуальные методологии бизнес-моделирования, которые используют на сегодняшний день. Разумеется, первое, что приходит на ум и что у всех на слуху, так это ARIS, однако, существуют также еще такие методологии, как Catalyst компании CSC, Business Genetics, SCOR (Supply \ Chain Operations Reference), POEM (Process Oriented Enterprise Modeling), а также множество других менее популярных.

Однако, на сегодняшний день имеются различные методы и подходы к построению, а также отображению моделей бизнес-процессов, в качестве основных можно выделить: функциональный и объектно-ориентированный. В первом обозначенном подходе главной мыслью является то, что структурообразующий элемент есть функция. То есть это может быть операция, действие или, например, бизнес-функция. В рассматриваемом случае система имеет образ некой иерархии, которая состоит из ряда функций, которые имеют взаимосвязь между собой. Второй же подход базируется на том, что раздроблен на некий набор объектов, которые отражают какие-либо объекты из реального мира. Стоит уточнить, что эти объекты имеют взаимодействие между собой при помощи рассылки неких сообщений.

В данный момент необходимо рассмотреть именно объектно-ориентированный подход моделирования.

Следующим шагом необходимо рассмотреть сам язык UML (аббревиатура расшифровывается с английского как Unified Modeling Language), а также его определенные свойства, важные для того, чтобы правильно создать модели бизнес-процессов. К данным свойствам можно отнести следующее: различная стандартизация, различные графические определения, принятые для обозначения различных объектов диаграмм, различными вариациями диаграмм, предназначенных для эффективного моделирования отдельных статистических и динамических процессов и подпроцессов.

Язык UML имеет одну важную особенность, о которой необходимо упомянуть – он является слабо типизированным языком. А также включает в себя различные механизмы для расширения, которые позволяют реализовывать разнообразные интерпретации семантики всех элементов моделей. Язык UML обладает следующим рядом важных преимуществ нотации:

* относится к типу объектно-ориентированного подхода, а способы его представления результатов, полученных как итог анализа и реализации проектирования, которые близки по смыслу с объектно-ориентированным программированием, что позволяет осуществлять;
* позволяет рассмотреть систему практически со всех возможных сторон деятельности и проанализировать разные аспекты ее поведения;
* UML-диаграммы со всеми элементами элементарны для чтения и понимания;
* является масштабируемым и активно развивающимся.

**1.5 Виды диаграмм, использующих при моделировании бизнес-процессов**

В пункте 1.4 было рассмотрено, как реализуется моделирование бизнес-процессов. Следующим шагом необходимо выделить виды диаграмм, которые используются при моделировании бизнес-процессов.

В наши дни используется широкий спектр диаграмм и графических методов для отображения процессов, поэтому для каждого случая можно найти свой наиболее эффективный способ. Ниже были перечислены виды диаграмм:

* диаграмма потока работ (Flow Chart Diagram) с целью отображения операций, данных, оборудования процесса и других элементов специальными символами, позволяет осуществлять построение логической последовательности действий процесса. Её главное преимущество заключено в том, что она обладает достаточной гибкостью, которая полностью позволяет представить процесс и подпроцесс множеством самых различных способов;
* диаграмма потока данных (Data Flow Diagram) отражает процесс передачи информации между операциями процесса, показывает входы и выходы информации, какие действия ее изменяют и где она хранится. Описывает взаимосвязь этих операций за счет данных, лежит в основе структурного анализа процессов, которые можно разложить этим методом на логические уровни, где у каждого процесса можно полностью рассмотреть, а также выделить его подпроцессы с высочайшим уровнем детализации всех элементов;
* диаграмма ролей (Role Activity Diagram) рассматривает и моделирует процесс, основываясь на отдельных ролях (абстрактном элементе процесса, выполняющего организационную функцию) и их группах, рассматривает взаимодействие ролей в процессе. Диаграмма демонстрирует ответственность каждого за выполняемый процесс и его операции;
* IDEF (Integrated Definition for Function Modeling) реализует набор методов описания частей бизнес-процессов с помощью диаграмм (IDEF0, IDEF1, IDEF1X, IDEF2, IDEF3, IDEF4, IDEF5) основанных на методологии SADT;
* IDEF0 представляет модель функций процесса, определяет основные функции рассматриваемого процесса, его входы и выходы, управляющие воздействия и ресурсы, взаимосвязанные с основными функциями. Процесс может быть рассмотрен на более низком уровне;
* IDEF3 отражает модель поведения процесса и состоит из двух видов моделей, где первый вид описывает поток работ, а второй - состояния перехода объектов;
* цветные сети Петри - метод, представляющий процесс в виде графа, имеющим вершины - действия процесса, и дуги - события перехода процесса из состояния в состояние, применяется для динамического моделирования поведения процесса.

## **1.6 Моделирование бизнес-процессов предметной области**

Методы моделирования в пункте 1.5 реализованы в виде отдельных специальных программ для моделирования.

Рассмотрев преимущества методов и нотации UML, для моделирования бизнес-процессов банковской сферы было выбрано CASE-средство и инструмент моделирования – Visual Paradigm, являющийся полноценной платформой упрощения процесса разработки ПО и поддерживающий языки программирования.

Для моделирования деятельности организации используется диаграмма вариантов использования (Use Case в UML), которая отражает отношения между актёрами (логически связанные роли в UML, исполняемые при взаимодействии с прецедентами или сущностями; актером может быть человек, другая система, подсистема класс, представляющий нечто вне сущности) и прецедентами (спецификациями последовательностей действий в моделируемой системе или частях её функциональности, в которых пользователь получает измеримый и конкретный результат), а также позволяет рассмотреть систему на концептуальном уровне.

Диаграммы предметной области представлены на рисунках ниже.

Диаграмма UseCase представлена на рисунке 1.1.

На диаграмме, которая представлена на рисунке 1.2 можно увидеть 3 основных структурной единицы, а именно: клиент – наш основной пользователь, ИС – в данном случае наше приложение, сервер – приложения для обработки запросов, приходящих от ИС, БД – наша база данных всех пользователей системы и информационных блоков.

Для описания состояний каждой из систем используем диаграмму активностей, которая представлена на рисунке 1.3.

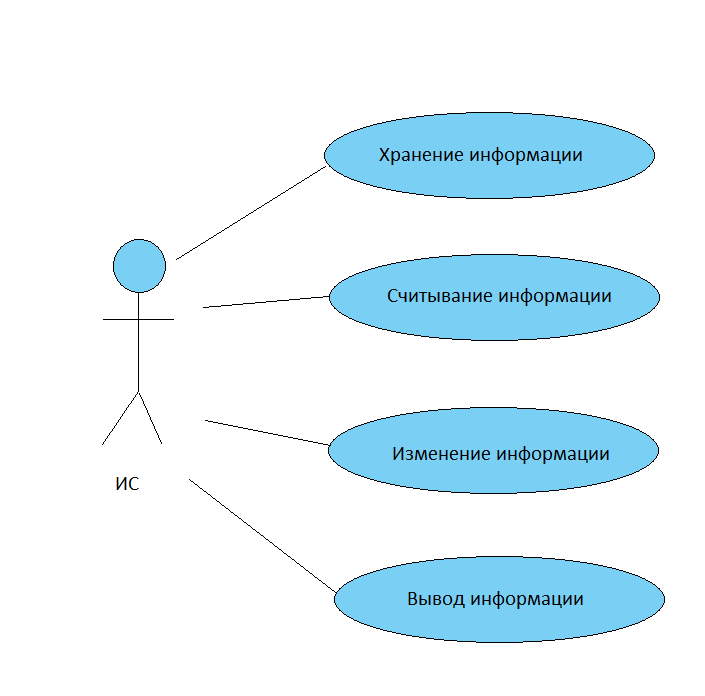


Рисунок 1.1 – UseCase – диаграмма «Основные структуры и функциональности»

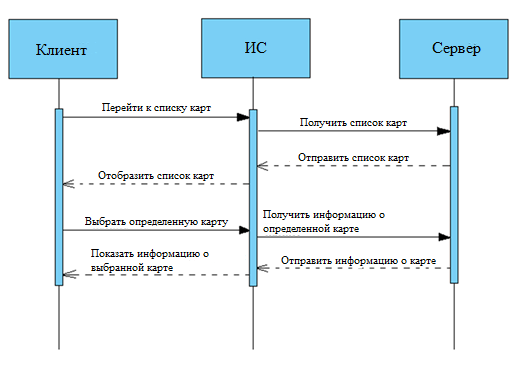


Рисунок 1.2 – Диаграмма последовательности для процесса «Просмотр информации о банковской карте»

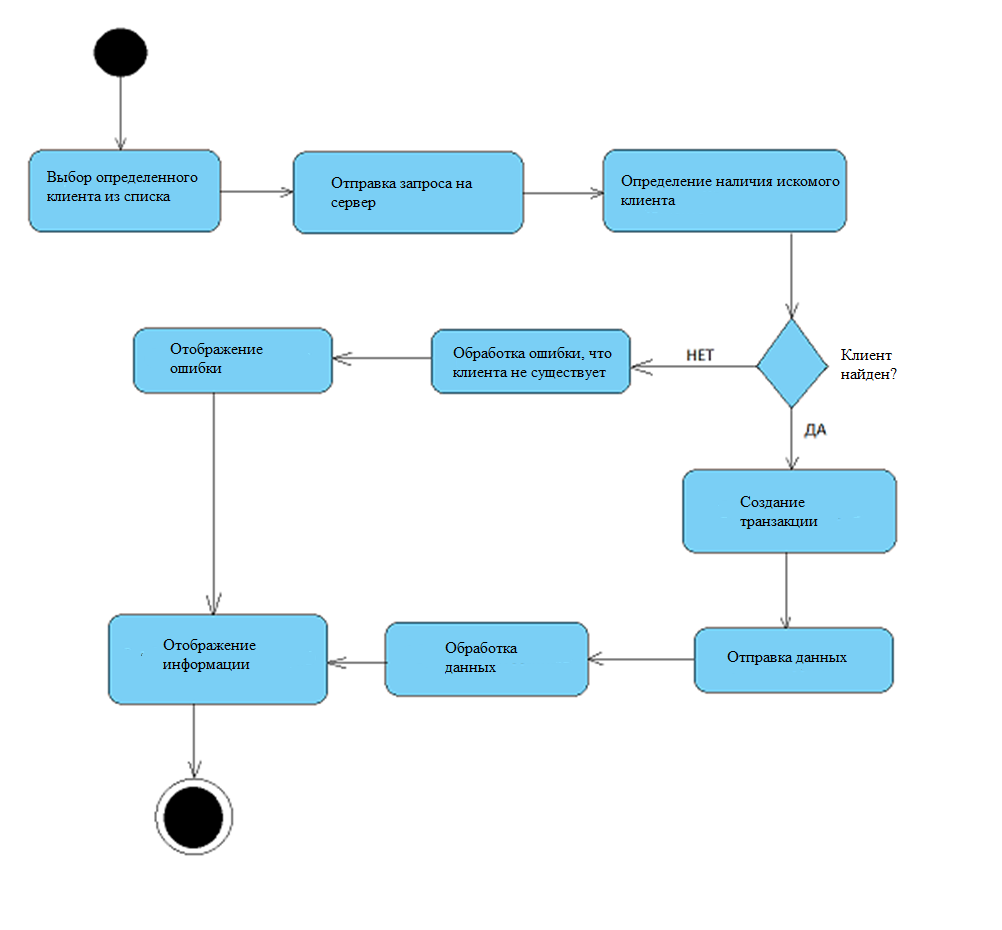


Рисунок 1.3 – Диаграмма активностей процесса «Просмотр информации о клиенте»

Для того, чтобы отобразить основные состояния информационной системы на экране просмотра информации о клиенте. В данном случае для описания данной процесса была использована диаграмма состояний.

Диаграмма состояний процесса просмотра информации о клиенте представлена на рисунке 1.4.

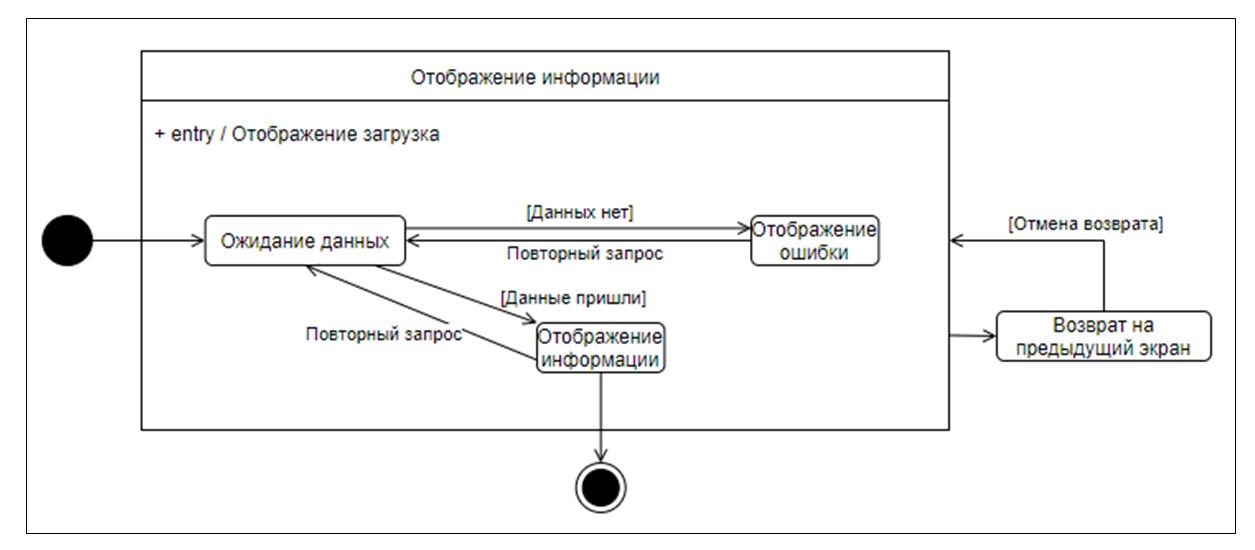


Рисунок 1.4 – Диаграмма состояний «Просмотр информации о клиенте»

Можно заметить, что диаграммы UML демонстрируют основные бизнес-процессы информационной системы для того, чтобы разработчики в процессе написания программного продукта ориентировались на них и время разработки увеличивалась.

На рисунке 1.5 представлена диаграмма классов. На данной диаграмме можно увидеть следующие список связей между сущностями, которые представлены ниже:

* несколько пользователей может пользоваться приложением;
* несколько приложений пользуется одним сервером;
* один пользователь может иметь несколько карт;
* один пользователь может иметь только один личный кабинет с личными данными;
* один сервер взаимодействует только с одним приложением;
* один сервер взаимодействует с одной базой данных.

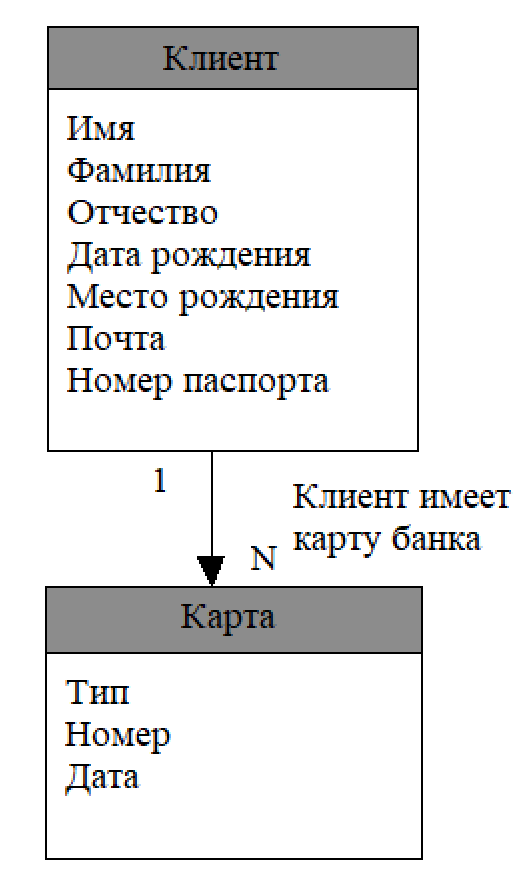


Рисунок 1.5 – Диаграмма классов

Также на рисунке 1.6 представлена диаграмма обзора взаимодействия при сценарии, когда пользователь отправляет свой логин и пароль.

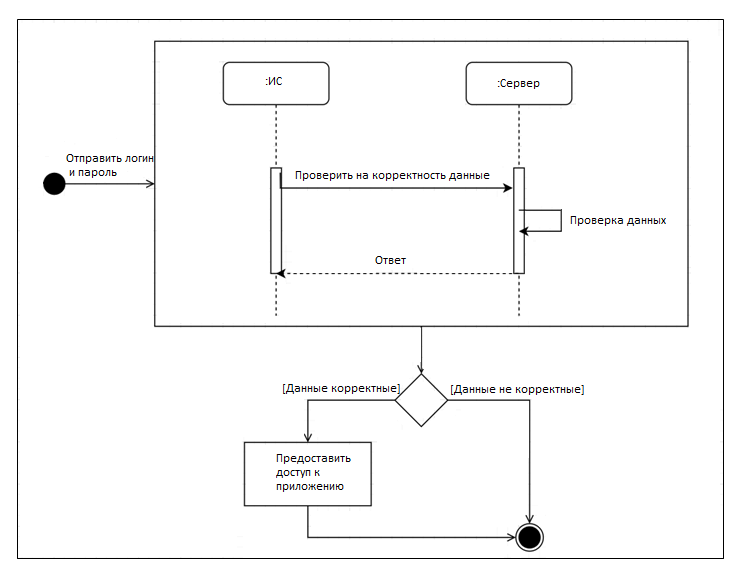


Рисунок 1.6 – Диаграмма обзора взаимодействия при отправке данных авторизации

**1.7 Выводы по разделу**

В данном разделе работы:

* проанализированы два подхода к построению и отображению моделей бизнес-процессов;
* выбран подход к построению и отображению моделей бизнес-процессов;
* рассмотрен процесс моделирования бизнес-процессов и проанализированы его цели;
* рассмотрена методология проектирования и анализа с применением языка UML;
* изучена методология проектирования и анализа с применением языка UML;
* выполнено моделирование бизнес-процессов рассматриваемой предметной области. Построены следующие диаграммы: диаграмма состояний, диаграмма обзора взаимодействия при отправке данных авторизации, диаграмма классов, диаграмма активностей, а также диаграмма UseCase;
* создана диаграмма классов, на основе которой будет создана база данных;
* проведен анализ предметной области.

1. **Проектирование прототипа информационной системы**

Сделав вывод из анализа, пришли к выводу о том, что создание прототипа информационной системы банковской сферы в качестве защищенной системы для тестирования её на различные уязвимости и атаки, будет наиболее актуальным.

Функциональность продукта: главная функция разрабатываемого ПО – это проведение банковских операций в режиме реального времени, как со счетами пользователей, так и с банковскими картами. Также дополнительные функции для персонала банка, например, просмотре всех транзакций, если сотрудник имеет допустимый уровень безопасности для данного действия, в том время как клиент банка может посмотреть только свои операции.

Рамки и ограничения проекта: требования в разрабатываемому ПО должны быть реалистичны, временной промежуток должен быть достаточен для разработки внедрения. Функционал должен быть строго ограничен предметной областью и не содержать в себе лишних операции, что повлечет за собой увеличение ресурсов на разработку и конечный размер ПО. То есть ПО должно быть емким, простым и понятным интуитивно, но при этом отвечать всем функциям.

* 1. **Обоснование выбора языка программирования**

Для реализации информационной системы был выбран язык JavaScript.

JavaScript - это «клиентский» скриптовый интерпретируемый браузером, язык программирования. Его основная задача создания интерактивных веб-страниц анимационных эффектов и работы с элементами DOM. Ни один современный сайт не мыслим без JavaScript с его может происходить полноценная загрузка страниц, так называемые AJAX сайт, что ускоряет работу сайта особенно если он обладает большим количеством данных и сложным функционалом.

Основные архитектурные черты: динамическая типизация, слабая типизация, автоматическое управление памятью, прототипное программирование, функции как объекты первого класса.

На JavaScript оказали влияние многие языки, при разработке была цель сделать язык похожим на Java. Языком JavaScript не владеет какая-либо компания или организация, что отличает его от ряда языков программирования, используемых в веб-разработке.

Название «JavaScript» является зарегистрированным товарным знаком корпорации Oracle в США.

Стандартом языка JavaScript является ECMAScript. По состоянию на 2012 год, все современные браузеры полностью поддерживают ECMAScript 5.1. Старые версии браузеров поддерживают по крайней мере – ECMAScript 3. 17 июня 2015 года состоялся выпуск шестой версии ECMAScript. Эта версия официально называется ECMAScript 2015, которую чаще всего называют ECMAScript 2015 или просто ES2015. С недавнего времени стандарты ECMAScript выпускаются ежегодно. Эта документация относится к последней версии черновика, которой является ECMAScript 2018.

Первоначально JavaScript обладал довольно небольшими возможностями. Его цель состояла лишь в том, чтобы добавить немного поведения на веб-страницу. Например, обработать нажатие кнопок на веб-странице, произвести какие-нибудь другие действия, связанные, прежде всего с элементами управления.

Однако развитие веб-среды, появление HTML5 и технологии Node.js открыло перед JavaScript гораздо большие горизонты. Сейчас JavaScript продолжает использоваться для создания веб-сайтов, только теперь он предоставляет гораздо больше возможностей.

Также он применяется как язык серверной стороны. То есть если раньше JavaScript применялся только на веб-странице, а на стороне сервера нам надо было использовать такие технологии, как PHP, ASP.NET, Ruby, Java, то сейчас благодаря Node.js мы можем обрабатывать все запросы к серверу также с помощью JavaScript.

В последнее время переживает бум сфера мобильной разработки. И JavaScript опять же не остается в стороне: увеличение мощности устройств и повсеместное распространение стандарта HTML5 привело к тому, что для создания приложений для смартфонов, планшетов и настольных компьютеров мы также можем использовать JavaScript. То есть JavaScript уже перешагнул границы веб-браузера, которые ему были очерчены при его создании.

И что вообще раньше казалось фантастикой, но сегодня стало реальностью - javascript может использоваться для набирающего популярность направления разработки для IoT (Internet of Things или Интернет вещей).

Таким образом, вы можете встретить применение JavaScript практически повсюду. Сегодня это действительно один из самых популярных языков программирования, и его популярность еще будет расти.

* 1. **Среда разработки (язык программирования)**

Для разработки банковской системы, которая будет разработана на основе языка JavaScript необходимо выбрать среду разработки, которая будет соответствовать необходимым требованиям и будет удобна для выполнения данной задачи.

Поскольку JS является один из популярных высокоуровневых языков программирования, то существует большой выбор сред разработки.

Также необходимо учесть тот факт, что среда разработки должна быть удобна для верстки сайта на HTML и CSS и для создания кода на языке PHP.

Для анализа было выбрано несколько популярных вариантов сред разработки, которые будут сравнены между собой в таблице 2.1 ниже.

Таблица 2.1 – Сравнение сред разработки

|  |  |  |  |
| --- | --- | --- | --- |
| Критерий сравнения | Atom | Visual Studio Code | PHPStorm |
| Открытый код | Да. | Да. | Нет. |
| Процесс установки | Очень легкая установка. | Простая установка. | Простая установка. |
| Цена | Бесплатный. | Бесплатный редактор. | 129 $ за первый год работы |
| Поддержка ОС | Linux, macOS, Windows с поддержкой плагинов, написанных на JavaScript, | Microsoft Windows, macOS и Linux | Windows, macOS или Linux. |
| Горячие клавиши | Редактирование и навигация при помощи горячих клавиш. | Присутствуют. | Присутствуют. |
| Системные требования | Достаточно приемлемые как для среды разработки. | Необходима достаточно мощная система для поддержки данной среды. | Достаточно приемлемые как для среды разработки. |
| Автосохранение | Можно включить пакет автосохранения. | Можно включить режим автосохранения. | Присутствует, но не всегда срабатывает. |
| Управление | Клавиатура и мышь. | Клавиатура и мышь. | Клавиатура и мышь. |

Продолжение таблицы 2.1

|  |  |  |  |
| --- | --- | --- | --- |
| Встроенные тесты | Нет. | Нет. | Больше сотни различных встроенных тестов для обнаружения ошибок для эффективного написания кода. |
| Подсветка синтаксиса | Присутствует. | Присутствует. | Присутствует. |
| Дебаггер | Присутствует. | Дебаггер с точками останова, стеком вызовов, интерактивной консолью. | Полномасштабный дебаггер для эффективной отладки кода, происходящей как на серверной, так и на клиентской сторонах. |
| Интерфейс | Удобный в использовании интерфейс, гибкие настройки редактора, подключаемых пакетов. | Удобный и простой интерфейс. | Навигация для одновременной работы с несколькими файлами.  Возможность изменения темы: её цвета (на темный или светлый) или шрифта. |

Продолжение таблицы 2.1

|  |  |  |  |
| --- | --- | --- | --- |
| Функционал | Более 50 открытых модулей, однако, пустая комплектация «из коробки». | Мало плагинов. Однако, поскольку имеет хороший набор функционала «из коробки», без необходимости устанавливать дополнительные плагины. | Для неопытных программистов функционал избыточен. |
| Автодополнение кода | Присутствует. | Контекстное автодополнение, как синтаксиса, так и используемых переменных, модулей, функций и так далее. | Присутствует. |

Исходя из таблицы, в которой приведены критерии, по которым сравнивались несколько сред разработок, можно сделать следующей выбор. Самым лучшим выбором будет среда разработки Atom, поскольку Atom и WebStorm имеют похожие характеристики, однако, первая среда разработки полностью бесплатна и данный аргумент является слишком весомым для выбора. Среда Visual Studio Code имеет более невыгодные параметры по сравнению с двумя другими выборами: отсутствие встроенных тестов и достаточно большое ресурсопотребление. Также была выбрана среда PHPStorm, поскольку часть движка сайта была написана на языке PHP, поэтому было удобнее использовать сразу несколько сред разработки.

## **2.3 Модель данных (ER-модель)**

Для предметной области банковской системы, работающей в реальном времени, можно спроектировать следующие сущности: Клиент, Карта. В таблицах 2.2-2.3 дано описание каждой сущности.

Таблица 2.2 – Описание сущности «Клиент»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Тип данных | Обязательное | PK | Описание |
| KlientID | int |  |  | Уникальный идентификатор клиента |
| First\_Name | varchar(256) |  |  | Полное имя клиента |
| Middle\_Name | varchar(256) |  |  | Отчество клиента |
| Last\_Name | varchar(256) |  |  | Фамилия клиента |
| PhoneNumber | bigint |  |  | Номер телефона клиента |
| Passport | bigint |  |  | Паспорт клиента |
| Password | varchar(256) |  |  | Пароль клиента |
| DateBirth | Date |  |  | Дата рождения клиента |
| PlaceBirth | varchar(256) |  |  | Место рождения клиента |

Продолжение таблицы 2.2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Email | varchar(256) |  |  | Особые отметки о клиенте |
| Avavtar | varchar(256) |  |  | Аватарка клиента |

Таблица 2.3 – Описание сущности «Карта»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя поля | Тип данных | Обязательное | PK | Описание |
| TypeCard | integer |  |  | Уникальный идентификатор счета клиента |
| NumberCard | integer |  |  | Номер карты, состоящий из 16 символов |
| DateCard | integer |  |  | Дата окончания действия карты |
| CardHolder | Varchar(50) |  |  | Имя держателя карты |

На рисунке 2.1 показана ER-диаграмма, представляющая сущности Клиент и Карта и их взаимосвязи. Числа, описывающие типы бинарных связей (1:1, 1:M, M:N), обозначают максимальное количество сущностей на каждой стороне связи. Эти числа называются максимальными кардинальными числами, а соответствующая пара чисел называется максимальной кардинальностью.

Исходя из таблиц с описанием сущностей и построенной ER-диаграммы проекта, была создана база данных для веб-сайта банка. В качестве СУБД был использован phpMyAdmin.

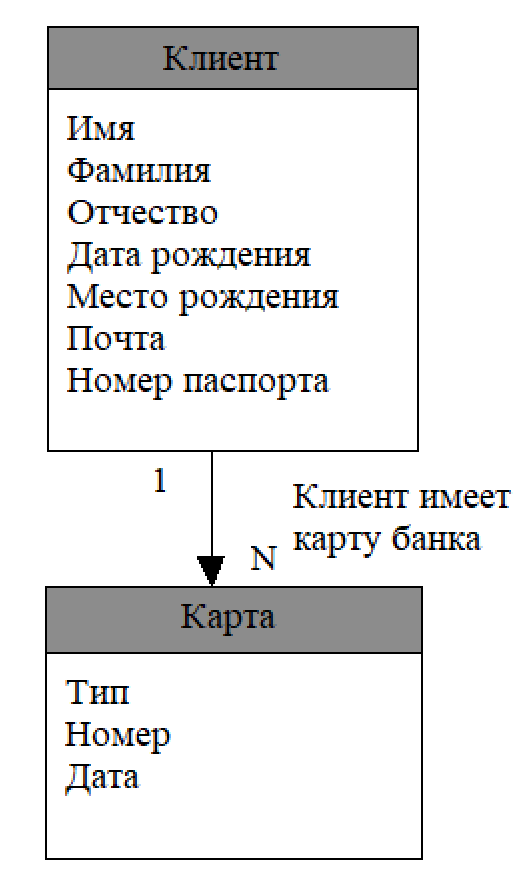


Рисунок 2.1 – ER-диаграмма проекта

## **2.4 Выводы по разделу**

Во втором разделе была рассмотрена интегрированная среда разработки. Были рассмотрены и проанализированы существующие IDE по разработанным критериям оценки. Для реализации программы был выбран язык JS и проанализирован.

Также были выделены сущности и связи для БД, на основе которых они и была создана с помощью СУБД phpMyAdmin.

# **Информационная система**

**3.1 Описание разработки пользовательского интерфейса прототипа информационной системы**

Основной интерфейс содержит главная страница, на которой предоставлен почти весь спектр возможностей: обширное меню имеет все необходимые возможности для пользования система. Главная страница сайта представлена на рисунке 3.1. Она содержит меню с выпадающими списками, кнопки, которые направляют на страницы отзывов и новостей банка.

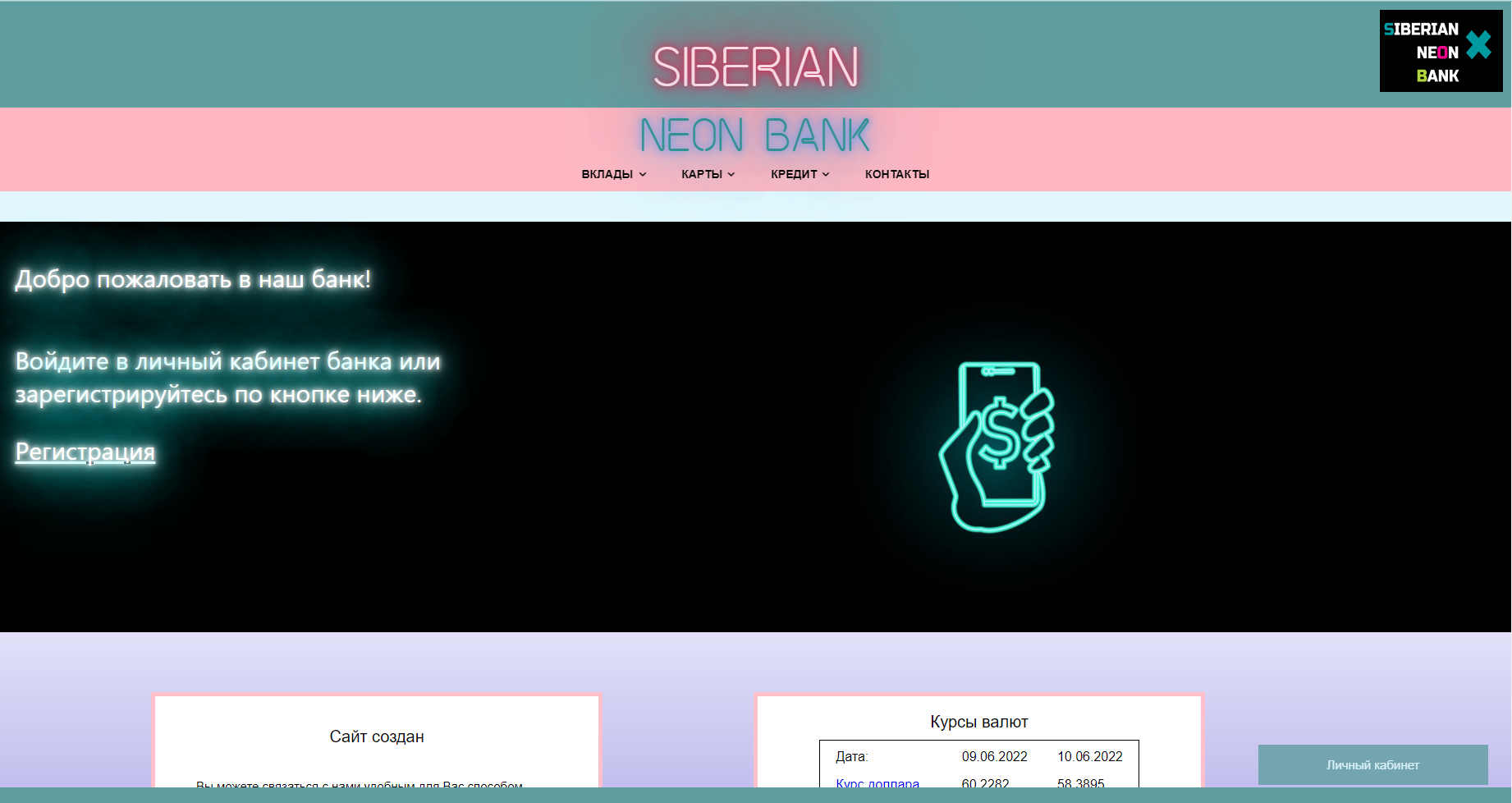


Рисунок 3.1 – Главная страница сайта

Внизу главной страницы находятся четыре текстовых блока. Первый блок имеет ссылки на связь с банком.

Второй блок имеет в себе конвертер валют, который обновляется каждый день по курсу Центробанка.

Третий блок имеет кнопку на переход с отзывами банка.

А четвертый содержит новости о банке. Данные блоки представлены на рисунке 3.2.

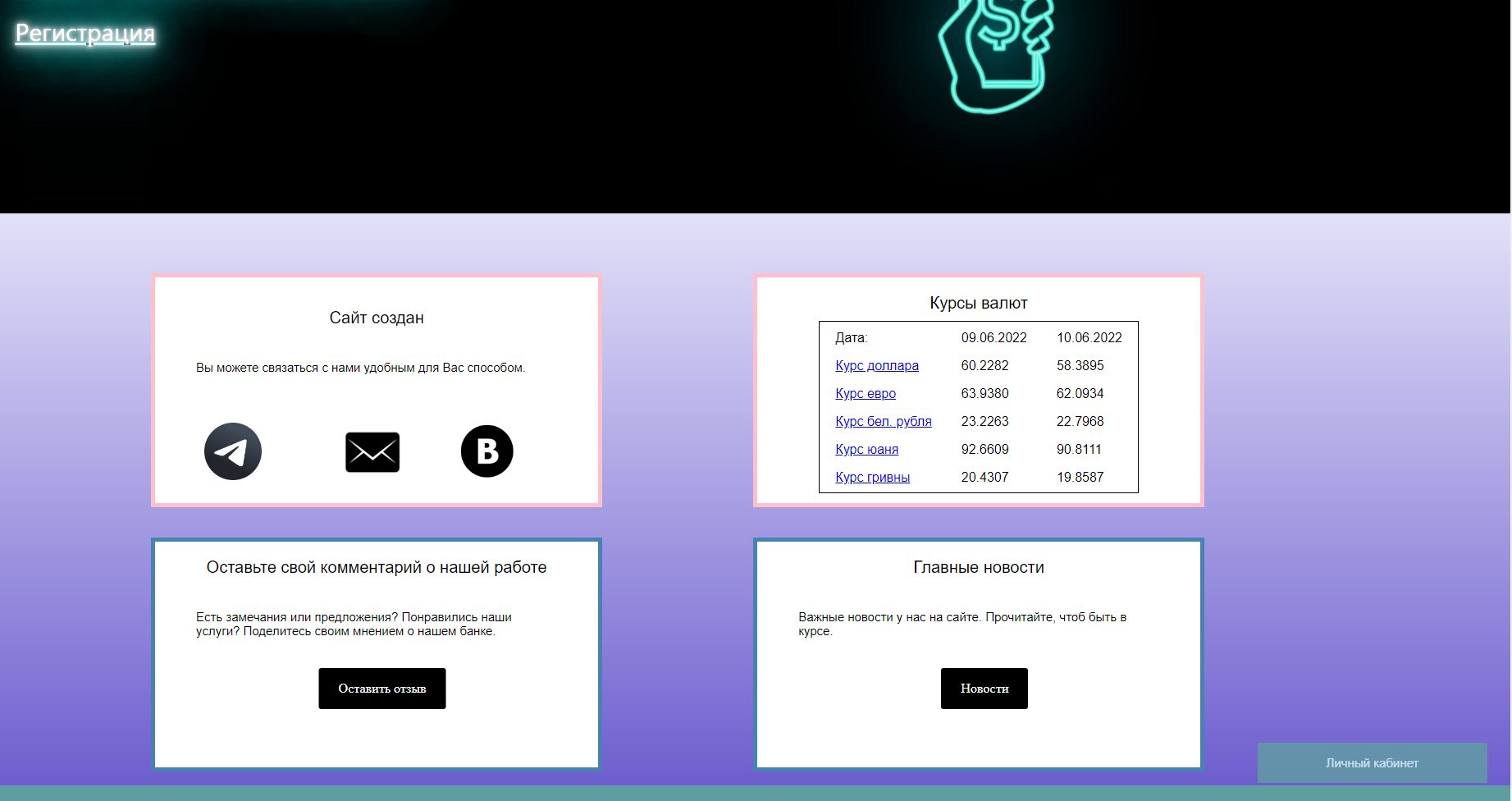


Рисунок 3.2 – Главная страница сайта

Страница сайта с отзывами представлена на рисунке 3.3.



Рисунок 3.3– Страница с отзывами

Страница с новостями банка, представлена на рисунке 3.4.

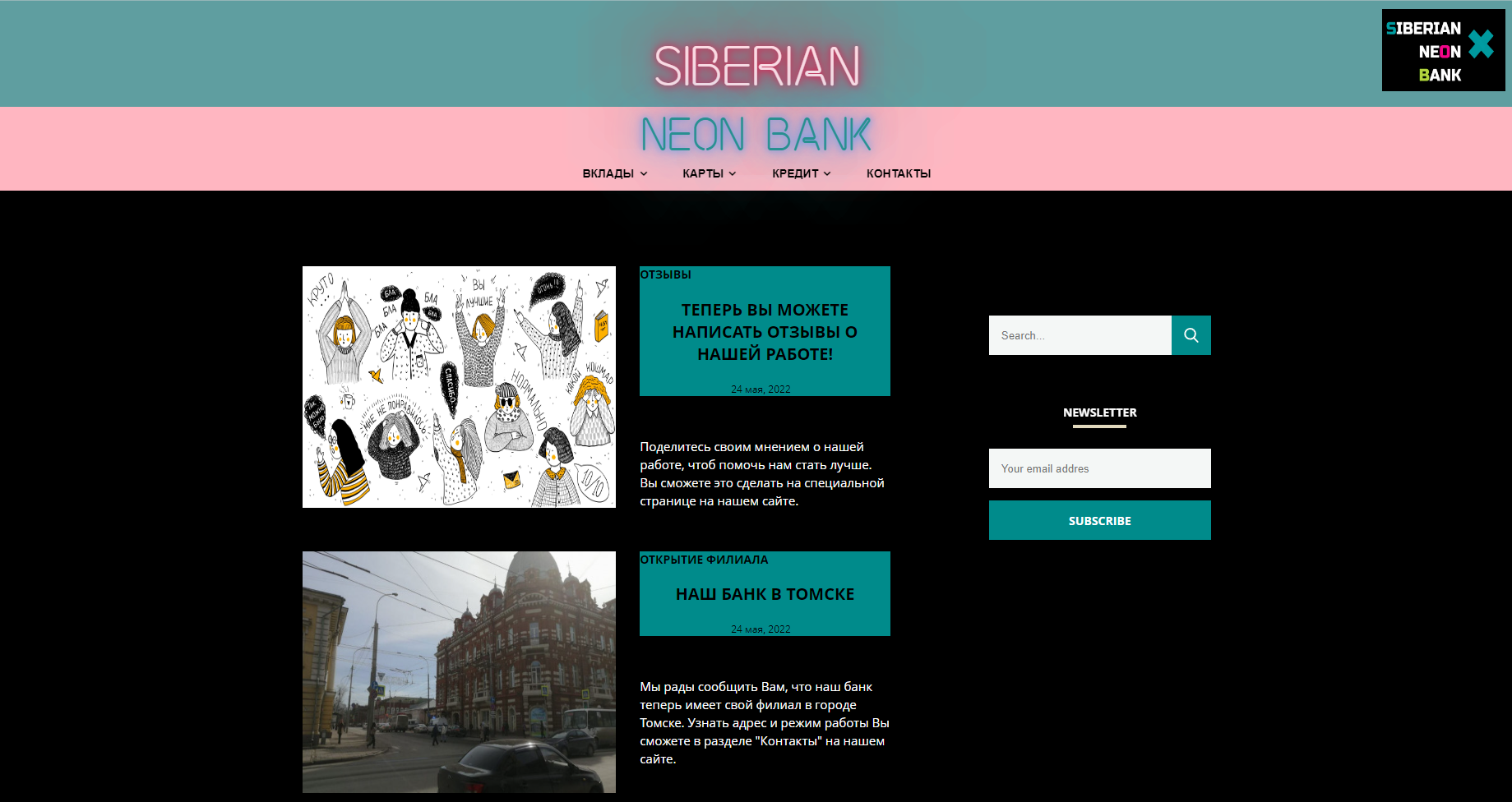


Рисунок 3.4 – Новости банка

Меню с выпадающими списками предоставлено на рисунке 3.6. Вкладка «Вклады» содержит форму для открытия вклада, а также информацию по вкладам. «Карты» содержат форму для отправки заявки на оформление карты, а также информацию по картам. Вкладка «Кредит» позволяет подать заявку на кредит.



Рисунок 3.6 – Меню с выпадающими списками

Внизу страницы располагается форма для входа в личный кабинет, рисунок 3.7.

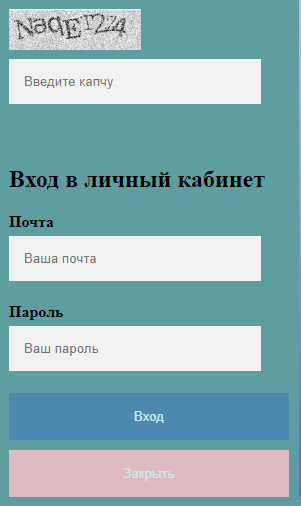


Рисунок 3.7 – Форма для личного кабинета

Вкладка меню «Контакты» содержит контактную информацию банка: на карте отмечены цветом регионы, в которых работает банк. При наведении на область будут показаны адреса банка, что показано на рисунке 3.8.



Рисунок 3.8 – Вкладка «Контакты»

Также в данной меню присутствует вкладка с информацией о картах банка, что представлена на рисунке 3.9.



Рисунок 3.9 – Информация о картах банка

Для того, чтобы оформить карту банка необходимо зайти на страницу регистрации, ссылка на которую располагается на главной странице. Необходимо заполнить все данные, а в конце выбрать карту, которая необходима. Форма регистрации представлена на рисунках 3.10-3.11.

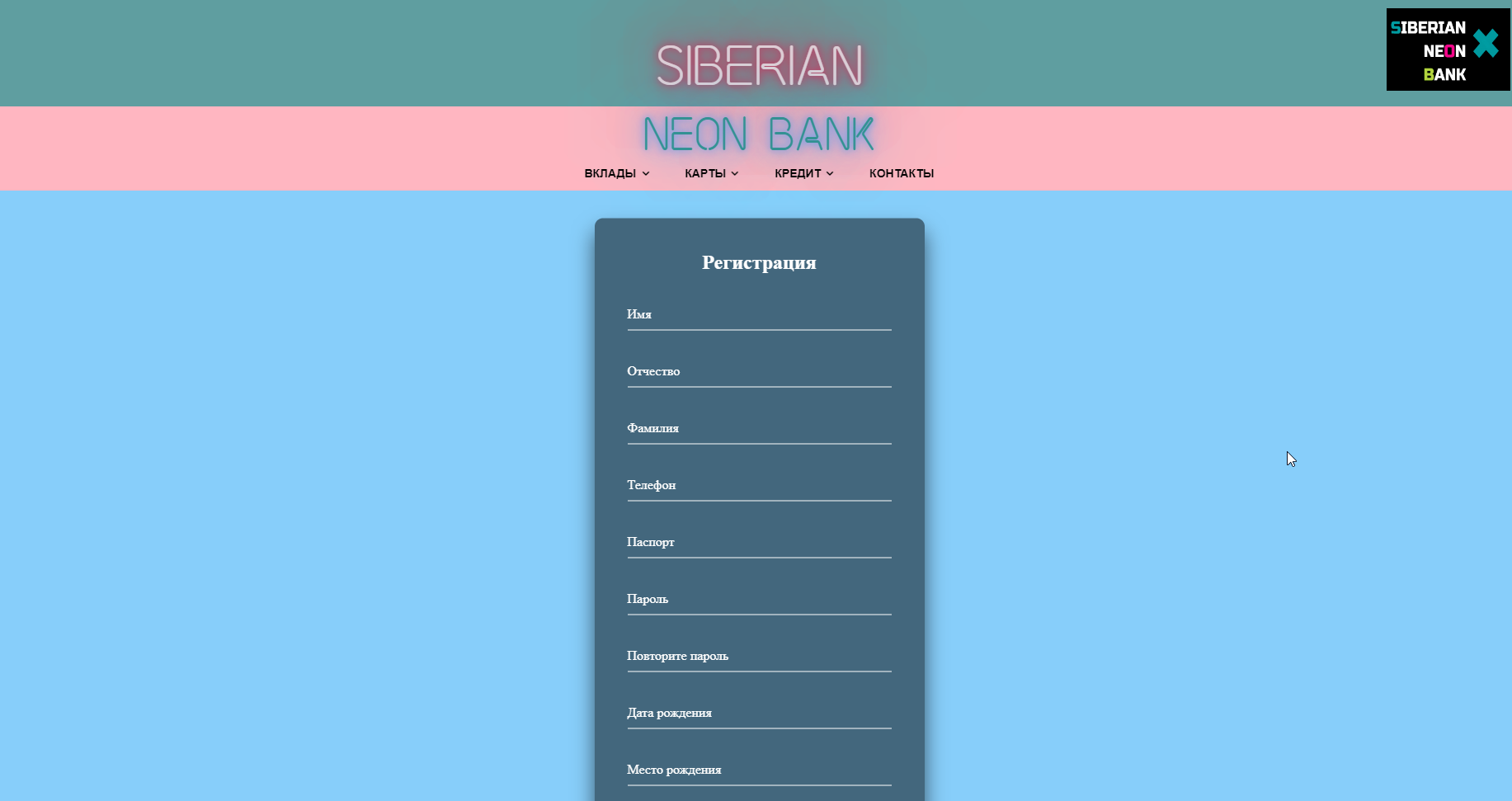


Рисунок 3.10– Форма регистрации

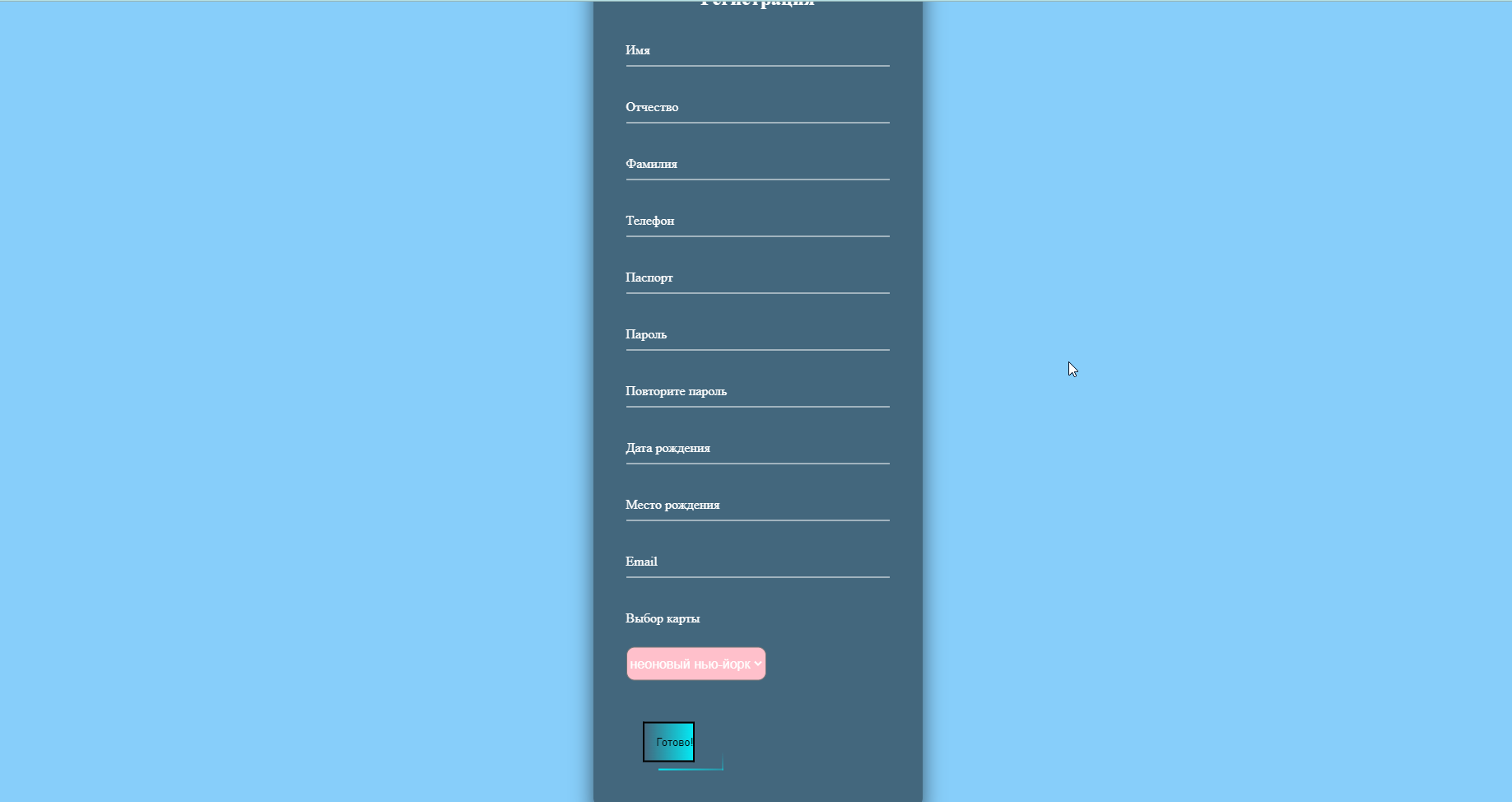


Рисунок 3.11– Форма регистрации

Зайти в личный кабинет можно по форме на главной странице, интерфейс личного кабинета изображен на рисунке 3.12.

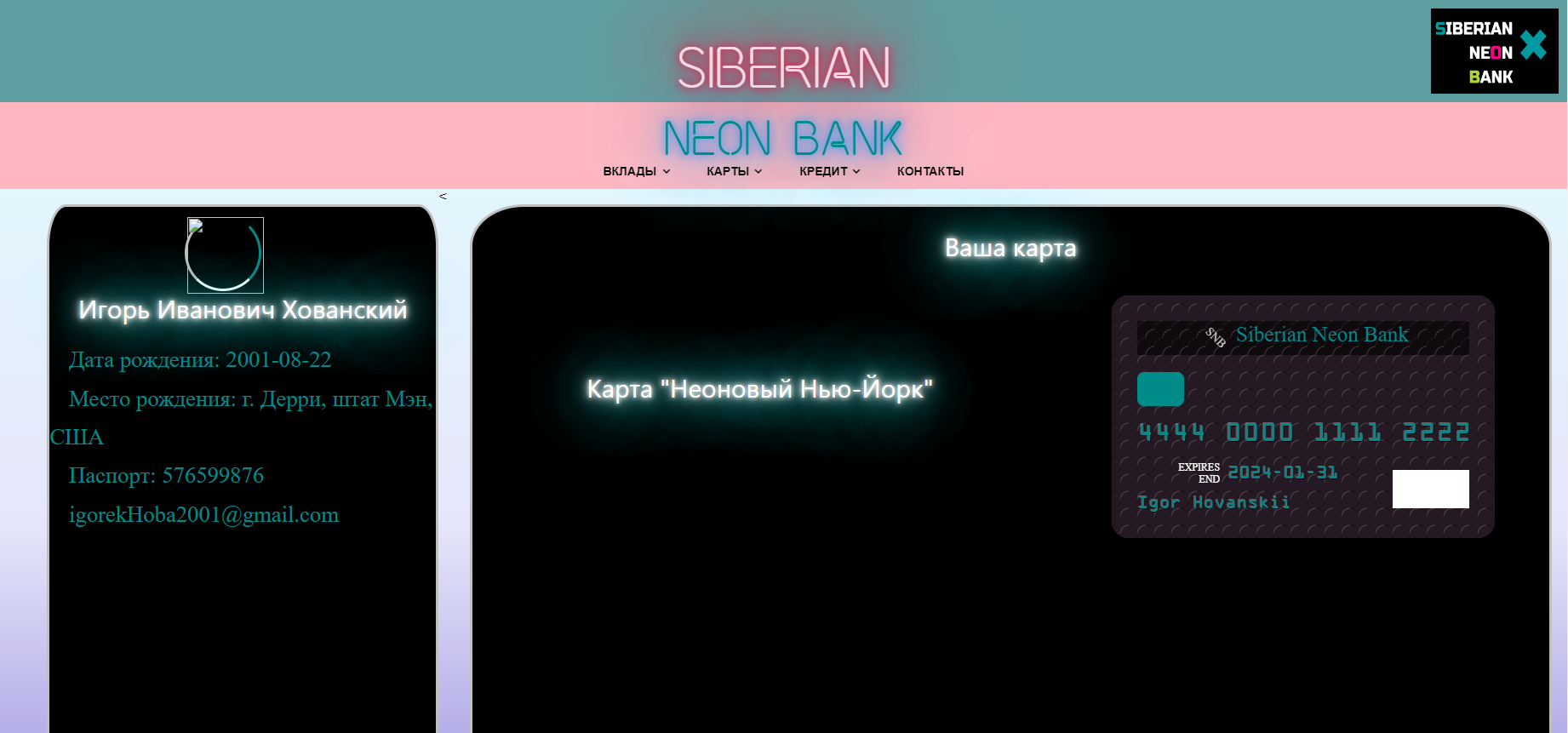


Рисунок 3.12 – Личный кабинет

3.2 Описание функционала разработанного прототипа информационной системы

Прототип информационной системы предназначен для информирования клиентов банка. Прототип ИС содержит личный кабинет пользователя с информацией о карте. Также сайт содержит различную информацию о банке и его услугах. Например, тарифы карта, контактная информация и прочую информацию.

Далее будет описан функционал прототипа информационной системы.

* 1. **Авторизация пользователя**

Описание: авторизация пользователя по имени и паролю.

Причинно-следственные связи, алгоритмы: происходит при запуске прототипа информационной системы.

Функциональные требования: при запуске прототипа информационной системы происходит подключение к базе данных. Проверяется на соответствие вводимое имя пользователя и пароль с тем, который содержится в базе данных. Если проверка пройдена, то открывается главное окно программы со всеми функциональными возможностями. Если совпадений не найдено, то программа не осуществит вход в систему.

* 1. **Использование личного кабинета пользователем**

Описание: получение информации из личного кабинета.

Причинно-следственные связи, алгоритмы: происходит после авторизации пользователя в прототипе информационной системы.

Функциональные требования: после авторизации пользователя в прототипе информационной системы происходит вход в личный кабинет пользователя. В личном кабинете из базы данных подгружается личная информация пользователя: личные данные, банковские карты.

* 1. **Защита прототипа ИС от атак и устранение в ней уязвимостей**

Поскольку главной задачей, которая стояла перед нами, была задача создания максимально защищенного прототипа информационной системы, была выбрана область банковской сферы, где данное свойство было бы максимально актуально. В пункте 1.3 был представлен ряд инструментов для выявления уязвимостей. Эти инструменты являются сканерами, которые тестируют веб-ресурсы на все возможные уязвимости и подверженность атакам. Актуальные на сегодняшний день атаки и уязвимости были описаны в пункте 1.2, там же были описаны возможные способы устранения каждой уязвимости.

Последним этапом в разработке прототипа информационной системы была совокупность мероприятий по информационной безопасности – защите web-ресурсов.

Исходя из информации, описанной в этих двух частях, прототип был просканирован инструментами для выявления уязвимостей, то есть упомянутыми в пункте 1.2 сканерами, далее были найдены уязвимости, которые могли бы послужить в будущем причиной убытков. И уже после этого данная информация была проанализирована и все уязвимости были устранены.

Первым шагом необходимо протестировать разработанный сайт с помощью инструментов, описание которых было дано в предыдущем пункте.

С помощью программы Webserver stress test проведен стресс тест, то есть сымитируем DDos атаку: проверим стабильность и производительность веб-сервера. Результаты представлены на рисунке 3.13.

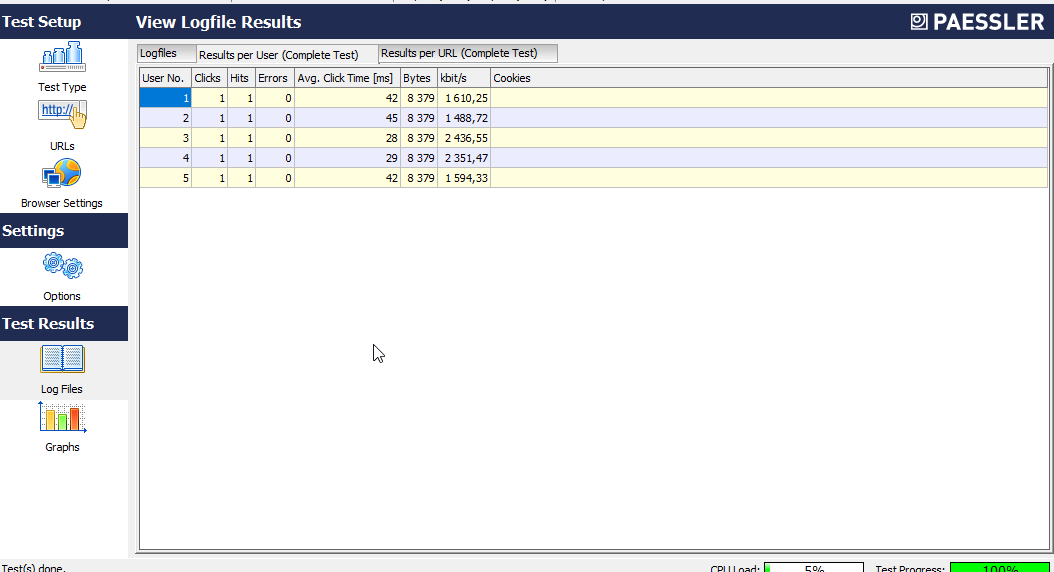


Рисунок 3.13 – Стресс-тест разработанного веб-ресурса

Несмотря на то, что сайт достаточно стрессоустойчивый, была введена, создана, так называемая капча, для защита от DDos атак, а также для защиты от такого метода взлома, как брутфорс. Капча (от CAPTCHA — англ. Completely Automated Public Turing test to tell Computers and Humans Apart — полностью автоматизированный публичный тест Тьюринга для различения компьютеров и людей) — компьютерный тест, используемый для того, чтобы определить, кем является пользователь системы: человеком или компьютером. Она представлена на рисунке 3.14.

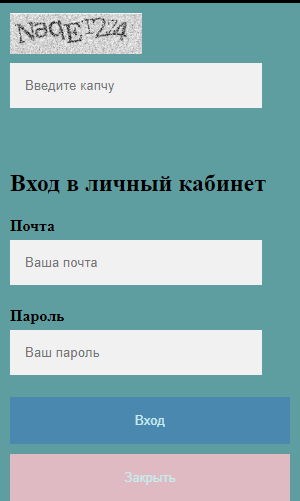


Рисунок 3.14 – CAPTCHA

Следующим шагом была устранена возможность какого-либо ввода в код SQl-инъекций. Данные входа или выхода, которые имеют числовые типы были защищены с помощью функции is\_numeric. Также было использовано PDO, так называемые, параметризированные запросы. PDO позволяют разработчика первым делом определить весь код SQL, а уже после передать каждый параметр в запрос. Это позволяет БД различать код и данные независимо от того, что вводится пользователем. Подготовленные операторы гарантируют, что злоумышленник не сможет изменить логику запроса, даже, если атакующий вставляет команды SQL.

Insecure Direct Object Reference уязвимость была устранена с помощью проверки каждого пользователя: всё данные проходят проверку на стороне сервера.

LFI-уязвимости (с английского расшифровывается как Local File Inclusion и переводится как включение локальных файлов) была устранена с помощью проверки валидности запросов, серверного пути. В PHP использование суперглобального массива $\_SERVER. Например, на рисунке 3.15 показан код, в котором авторизация пользователя происходит с помощью данного массива.



Рисунок 3.15 – Код с использованием суперглобального массива $\_SERVER

Также была создана фильтрация параметров входных данных, которые передают методы GET и POST. Данная мера является полноценной защитой от межсайтового скриптинга и PHP injection, что на сегодняшний день представляет достаточно большую угрозу для безопасности какого-либо веб-ресурса.

## **3.6 Выводы по разделу**

В данном разделе был полностью рассмотрен функционал сайта с подробным описанием.

Сайт позволяет создавать индивидуальные личные кабинеты для каждого клиента с личной информацией о банковских картах. А также для открытия\закрытия банковских карт и информировании клиентов. На разработанном сайте можно узнать адреса банка, узнать последние новости банка, а также оставить отзыв.

Вывод и ввод информации происходит из созданной ранее базы данных, подробное описание которой было предоставлено в предыдущем разделе.

Также были проведен ряд мероприятий по безопасности прототипа информационной системы, который поможет предотвратить незаконное и несанкционированное проникновение злоумышленников к личной информации, которая принадлежит клиентам банка. Данная задача проделанной работы являлась главной целью, которая была успешно выполнена.

Исходя из последнего раздела, был получен прототипа информационной системы банковской сферы, которые является максимально защищенным с точки зрения информационной безопасности.

# **ЗАКЛЮЧЕНИЕ**

В результате выполнения выпускной квалификационной работы была спроектирован прототип информационной системы банковской сферы, промоделированы бизнес-процессы, рассмотрены преимущества введения программы.

В процессе освоения дисциплины:

* изучен язык программирования Javascript;
* изучение язык программирования PHP;
* создана база данных для прототипа информационной системы;
* изучены основные типы уязвимостей и получены методы защиты от них;
* изучены инструменты защиты от уязвимостей;
* изучены способы совершенствования бизнес-процессов;
* получены навыки определения объектов автоматизации, входные и выходные данных создаваемой ИС;
* получены навыки создания документации рабочих проектов;
* изучены инструментальные средства моделирования бизнес-процессов и баз данных;
* проведен ряд мероприятий по защите web-ресурсов на разработанном прототипе информационной системы.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. И. Н. Васильева, Д. Ю. Федоров WEB-ТЕХНОЛОГИИ: Учебное пособие/ И. Н. Васильева, Д. Ю. Федоров. – издательство санкт-петербургского государственного экономического университета, 2018 - 48 с.
2. М. Егоров Выявление и эксплуатация SQl -инъекций в приложениях// Журнал «Защита информации» №1, 2019. – С. 2-8.
3. Стасышин В.М. Проектирование информационных систем и баз данных: учебное пособие / В.М. Стасышин - Новосибирск: НГТУ, 2019. – 100 c.
4. Малышев Е.Н. Проектирование информационных систем (Раздел 5. Индустриальное проектирование информационных систем. Объектно-ориентированная Caseтехнология проектирования информационных систем): учебное пособие / Е.Н. Малышев – Кемерово: КемГУ, 2019. – 70 с.
5. Букатов А.А. Методы и средства интеграции независимых баз данных в распределенных телекоммуникационных сетях: монография - А.А. Букатов, А.В. Пыхалов - Ростов-н/Д: Издательство Южного федерального университета, 2018. – 160 с.
6. Куликов, С. Тестирование программного обеспечения. Базовый курс. EPAM Systems, 2018. – 296 с.
7. Е.В. Побединский, В.В. Побединский Проектирование веб-сайтов с использованием технологий PHP, HTML, CSS И WORDPRES: Учебное пособие/ Е.В. Побединский, В.В. Побединский. – Екатеринбург, 2018 – 114 с.
8. DDoS–атаки. Причины возникновения, классификация и защита от DDoS-атак [Электронный ресурс] URL https://efsol.ru/articles/ddos-attacks.html (дата обращения 07.06.2022).
9. RCE - Remote code execution - удаленное выполнение кода на сервере [Электронный ресурс] URL https://insafety.org/rce.php (дата обращения 07.06.2022).
10. Буч Г., Рамбо Д., Якобсон И. Введение в UML от создателей языка. 2-е изд.: Пер. с англ. Мухин Н. – М.: ДМК Пресс, 2018. – 496 с.
11. Крэг Ларман: Применение UML 2.0 и шаблонов проектирования - Диалектика, 2019 г. – 736 с.
12. DDoS-атаки в I квартале 2021 года [Электронный ресурс] URL https://securelist.ru/ddos-attacks-in-q1-2021/101390/ (дата обращения 07.06.2022).
13. Статистика уязвимостей веб-приложений в 2018 году [Электронный ресурс] URL https://www.ptsecurity.com/ru-ru/research/analytics/web-application-vulnerabilities-statistics-2019/ (дата обращения 08.06.2022)
14. Смертные грехи безопасности сайта: что мы узнали из статистики сканера уязвимостей за год [Электронный ресурс] URL https://habr.com/ru/company/dataline/blog/505274/ (дата обращения 08.06.2022).
15. Информационная безопасность: вчера, сегодня, завтра учебное пособие / Арутюнов В. В. - Москва: РГГУ, 2019. – 219 c.