

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий

(наименование института полностью)

Кафедра «Прикладная математика и информатика»

(наименование кафедры)

02.03.03 Математическое обеспечение и администрирование
информационных систем

(код и наименование направления подготовки, специальности)

Технология программирования

(направленность (профиль)/специализация)

БАКАЛАВРСКАЯ РАБОТА

на тему «Исследование методов построения VI-систем с web интерфейсом»

Студент

А.А. Морозов

(И.О. Фамилия)

(личная подпись)

Руководитель

Т.Г. Султанов

(И.О. Фамилия)

(личная подпись)

Консультанты

К.А. Селиверстова

(И.О. Фамилия)

(личная подпись)

Допустить к защите

Заведующий кафедрой к.т.н. доцент Очеповский А.В.

(ученая степень, звание, И.О. Фамилия)

(личная подпись)

« _____ » _____ 2019 г.

Тольятти 2019

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий
(наименование института полностью)

Кафедра «Прикладная математика и информатика»
(наименование кафедры)

УТВЕРЖДАЮ
Зав. кафедрой «Прикладная
математика и информатика»
_____ А.В. Очеповский
(подпись) (И.О. Фамилия)
« _____ » _____ 20__ г.

ЗАДАНИЕ
на выполнение бакалаврской работы

Студент Морозов Александр Алексеевич

1. Тема «Исследование методов построения BI систем с web интерфейсом»
2. Срок сдачи студентом законченной выпускной квалификационной работы 21.06.2018
3. Исходные данные к выпускной квалификационной работе: возможность визуализации аналитической информации по BigData, использование языка JS.
4. Содержание выпускной квалификационной работы:
 - Обзор технологий и применение BI,
 - Основные алгоритмы BI,
 - Реализация выбранного алгоритма;
5. Ориентировочный перечень графического и иллюстративного материала: диаграммы, модели, скриншоты, презентация.
6. Дата выдачи задания «25» октября 2018 г.

Руководитель выпускной
квалификационной работы

(подпись)

Т.Г. Султанов
(И.О. Фамилия)

Задание принял к исполнению

(подпись)

А.А. Морозов
(И.О. Фамилия)

Аннотация

Тема выпускной квалификационной работы «Исследование методов построения VI-систем с web интерфейсом».

Объектом исследования являются VI-системы.

Предметом данной работы является VI-приложение с web интерфейсом.

В ходе работы были выполнены следующие задачи:

- рассмотрение ключевых вопросов в области VI;
- анализ ключевых вопросов существующих VI-систем;
- проектирование и разработка web-приложения.

В первой главе рассматривается вопрос о разработке VI систем. Приведены основные понятия в данной области, актуальность VI. Также, приведен ряд основных задач, решаемых VI-системой.

Во второй главе рассмотрены основные алгоритмы VI. Также, между различными готовыми решениями проводится сравнение критериев, таких как производительность, доступность (цена), безопасность, многозадачность. Итогом этого сравнения является выбранная для решения поставленной задачи.

Третья глава посвящена концептуальному и логическому моделированию приложения для VI системы. Были приведены основные аспекты разработки, функциональные требования, архитектура разрабатываемого приложения, основные модули, используемые в разработке, а также, тестирование приложения.

В итоге работы реализовано web приложение, в котором была автоматизирована VI-система для анализа данных, полученных с сервера.

Данная работа состоит из 50 страниц, включая в себя введение, три главы, заключение, список используемой литературы из 25 источников и приложения.

ABSTRACT

The topic of the graduation work is research of methods for building BI-systems with a web-interface.

The aim of the work is to give some information about the algorithms of the BI systems. The object of research is a business intelligence. The subject of the graduation work is a web-application.

This graduation work consists of an explanatory note on 50 pages, including introduction, 20 figures and 1 appendix..

This graduation work may be divided into several logically connected parts, which are: Technology review and BI application, Basic BI algorithms and Implementation of the selected algorithm.

The study examines the problem of a big data visualization and reports. The aim of this work is to make the reports creation and analysis easier for unexperienced users.

The author dwells on BI-systems and algorithms of data visualization. The work touches upon comparison of the different BI development platforms, that could be shown as existing alternatives of the developing application. Also, the theoretical part contains different BI algorithms comparison.

The main issues of the third part are about the development process and implementation of the web-application. The reader's attention is also drawn to different useful technologies, that were used for the application development.

As a result, the web-application with big data visualization is developed. This application could be used at university as the student interest statistics report.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	6
1 ОБЗОР ТЕХНОЛОГИЙ И ПРИМЕНЕНИЕ ВІ.....	8
1.1 Понятие и история ВІ.....	8
1.2 Перспективы использования ВІ систем.....	10
1.3 Обзор задач, решаемых ВІ.....	12
1.4 Постановка задачи.....	14
2. ОСНОВНЫЕ АЛГОРИТМЫ ВІ.....	16
2.1 Концепция ВІ систем.....	16
2.2 Архитектура ВІ систем.....	17
2.2 Обзор ВІ платформ.....	19
2.2.1 QlikView.....	20
2.2.2 Klipfolio.....	21
3. РЕАЛИЗАЦИЯ ИНТЕРФЕЙСА ВІ СИСТЕМЫ.....	28
3.1 Функциональные требования к приложению.....	28
3.2 Архитектура приложения.....	31
3.3 Технология WebSocket.....	32
3.4 Реализация приложения.....	35
3.5 Тестирование приложения.....	40
ЗАКЛЮЧЕНИЕ.....	42
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ.....	43
ПРИЛОЖЕНИЕ А Листинг вывода гистограммы на экран.....	46

ВВЕДЕНИЕ

BI (Business Intelligence) является важным процессом в любой сфере деятельности, позволяющим в несколько раз увеличить производительность и улучшить его результаты. Организации, использующие BI, как показывает практика, значительно превосходят своих сверстников. Основная задача BI – выявить основные потребности в том бизнесе, а также, найти возможные проблемы и уязвимости и преодолеть их.

Цель BI состоит в интерпретации большого количества данных в понятную, читаемую форму. При этом, выделяются лишь основные факторы эффективности, позволяющие наиболее подробно представить текущее положение дел на предприятии. Выполнение задач BI, позволяет понять текущее состояние дел на том или ином предприятии, оценить его текущее состояние, а также, спрогнозировать состояние наперёд.

В рамках данной работы будут рассмотрены наиболее популярные решения для разработки BI систем. При этом, по каждому из них расписаны основные аспекты, преимущества и недостатки и основные алгоритмы их работы. Тема выпускной квалификационной работы – исследование методов построения BI-систем с Web-интерфейсом.

Предметом исследования данной работы является BI-приложение с web интерфейсом.

Объектом исследования являются BI-системы.

Целью работы является разработка веб-приложения для проведения анализа вовлеченности студентов в образовательный процесс и прогноз изменения данного показателя с учётом уровня регрессии на основе уже имеющихся данных.

Для достижения поставленной цели, необходимо рассмотреть и выполнить следующие задачи:

- рассмотреть вопрос о разработке веб-приложений;

- исследовать методы построения VI-систем;
- проанализировать основные алгоритмы и программные средства для реализации VI-систем с веб-интерфейсом;
- изучить литературу по визуализации большого количества данных;
- спроектировать и разработать веб-приложение;

В первой главе проведено исследование необходимости использования VI-систем в настоящее время, приведены основные понятия по данной теме и рассмотрены основные перспективы. Также, в данной главе приведена постановка задачи.

Во второй главе приводятся основные платформы для разработки систем VI, проводится сравнение между данными платформами, их функционалом, алгоритмами обработки данных, а также, пользовательским интерфейсом. На основании поставленной задачи, выявляется средство для дальнейшей разработки VI-системы с web-интерфейсом.

Третья глава посвящена реализации приложения с веб-интерфейсом, которое позволяет предоставить пользователю данные об эмоциональном состоянии студентов и их вовлеченности в процесс обучения. Данные должны быть представлены пользователю как в графическом формате, так и в виде обычной таблицы, с возможностью её сортировки по столбцам.

В заключении подводятся итоги выполнения данной работы.

1 ОБЗОР ТЕХНОЛОГИЙ И ПРИМЕНЕНИЕ ВІ

1.1 Понятие и история ВІ

Интеллектуальный анализ данных, Business Intelligence (ВІ) – набор инструментов и методов для перевода сырой, необработанной информации в удобную для человека форму [21]. Данное понятие в России используется довольно редко. Возможно, это является результатом отсутствия прозрачного и понятного перевода. Подобная неопределённость в первую очередь связана с тем, что слово «Intelligence» имеет сразу несколько значений и смыслов в английском языке:

- готовность понимать;
- знания, приобретенные или переданные в процессе опыта, исследования или обучения;
- состояние или действия в процессе познания;
- разведывательные данные или разведку.

Впервые термин Business Intelligence предложил Ханс Петер Лун (1896 – 1964). Выдающийся американский ученый являлся специалистом в данной сфере, несмотря на то, что в основном занимался индексацией текстов и статическими исследованиями.

В 1958 году Ханс Петер Лун опубликовал статью «A Business Intelligence System». При этом, он показал бизнес как различные виды деятельности в разных сферах. Именно в 50-е годы Лун разработал несколько методов работы с информацией, а именно, алгоритмы хеширования и полнотекстового поиска.

К сожалению, его работы были забыты на 30 лет, до тех пор, пока известный аналитик Ховард Дреснер не открыл их заново. Он несколько расширил трактовку ВІ, предложив использовать данное понятие как описание различных технологий для поддержки принятия решений [16]:

Именно точка зрения Дреснера была принята за основу VI-систем. Сегодня же, VI представляет из себя процесс сбора, обработки и анализа разрозненной информации, выработки интуиции и понимания для улучшенного принятия решений пользователем.

VI позволяет пользователю получить ответы на вопросы бизнеса, а также, выявлять тенденции или шаблоны в необходимой пользователю информации. Данное программное обеспечение позволяет пользователю наглядно видеть и использовать большое количество данных. Знания, основанные на данных, получаются из данных с использованием VI, а также, хранилища данных.

Средства VI включают в себя

- инструменты построения реляционных хранилищ данных, позволяющих выполнять запросы с высокой скоростью;
- средства построения многомерных хранилищ данных, содержащих агрегатные данные;
- пользовательские приложения, содержащие интерфейс для вывода информации на экран;
- средства генерирования отчётов по данным.

На рисунке 1.1 представлены основные типы средств VI [21].

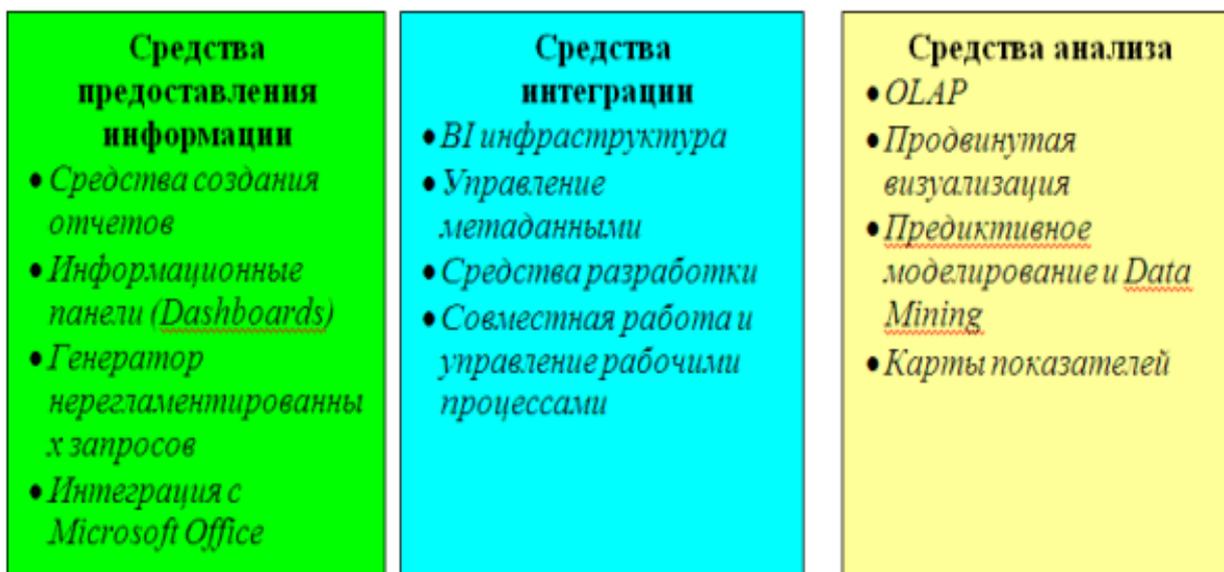


Рисунок 1.1 - Типы средств BI

1.2 Перспективы использования BI систем

BI является важным набором инструментов, позволяющим вывести создание отчетов на новый уровень. Данный набор инструментов включает в себя: интеллектуальный анализ данных, аналитическую обработку, запросы и отчетность.

Актуальность BI можно оценить по следующим признакам:

- удовлетворённость клиентов – BI позволяет повысить способность анализировать тенденции рынка и потребителей. С помощью данного набора инструментов можно легко определить тенденции спроса и предложения, оценить статистику того или иного предприятия. Это выгодно, потому что позволяет спланировать дальнейшие шаги к увеличению эффективности.

- снижение рисков и управление рисками – BI позволяет бизнесу получить доступ к подробной аналитике о бизнесе. Такая аналитика позволяет понять основные риски и неэффективные элементы в развитии бизнеса. Данный процесс может занимать значительное время, однако принесёт плодотворный результат, так как выявление основных рисков позволяет принимать более обоснованные решения.

- получение конкурентного преимущества – помимо всего вышеперечисленного, ВІ также позволяет бизнесу получить преимущество перед другими подобными компаниями на рынке. Данное преимущество в основном обусловлено постоянным отслеживанием тенденций и спроса на тот или иной товар или услугу. Зная данные тенденции, можно с лёгкостью принимать решения, позволяющие занять лидирующие места в рассматриваемой сфере.

Что касается российского рынка ВІ, согласно ресурсу Tadviser [13], он продолжает развиваться и расти. Повышение роли ВІ в оптимизации рабочих процессов различных компаний, в совокупности с ростом количества данных и развитием инструментов для их обработки, способствуют увеличению продаж аналитического программного обеспечения.

На рисунке 1.2 [12] можно увидеть выручку от проектов по 2016-2017 годам.

№	Компания	Выручка от BI-проектов в 2017 году, млн руб. с НДС	в т.ч. от продаж лицензий	в т.ч. от услуг*	Выручка от BI-проектов в 2016 году, млн руб. с НДС	в т.ч. от продаж лицензий	в т.ч. от услуг*	Динамика 2017/2016	Крупнейшие заказчики (проекты) в 2016-2017 годах
1	ГК GlowByte	2 193,9	н/д	н/д	1 711	н/д	н/д	28,2	н/д
2	Техносерв	1 525,5	н/д	н/д	н/д	н/д	н/д	н/д	Аэрофлот, ВТБ Лизинг, Газпромнефть
3	Крок	1 377,9	н/д	н/д	1 308,4	н/д	н/д	5,3	н/д
4	ГК Айтеко**	513,5	99,1	413,8	252	н/д	н/д	103,7	н/д
5	Инфосистемы Джет	400	н/д	н/д	н/д	н/д	н/д	н/д	н/д
6	БАРС Груп	386	57	329	331	70	261	16,6	Росавиация, Минздрав РФ (ЕГИСЗ), Главгосэкспертиза России (ИАС ЦС), Алабуга
7	Форсайт	313,3	264,9	48,4	60,4	н/д	н/д	418,5	Транснефть
8	Navicon	312,9	122,4	190,5	105	н/д	н/д	198	Heineken Россия, Акрихин
9	РДТЕХ	311,9	147,6	161,8	300,5	144,2	155,3	3,8	н/д
10	ГК Корус Консалтинг	299	н/д	н/д	62	н/д	н/д	382,3	н/д
11	Sapiens solutions	173	-	173	145	-	145	19,3	МегаФон, М.Видео, Детский мир, Газпром, Лента, Сбербанк
12	ГК Форс	157,3	31,5	125,8	159,4	71,7	87,7	-1,3	н/д
13	Полиматика (Polymatica)***	128,9	108,2	20,8	20,2	н/д	н/д	539,3	н/д
14	АйДи - Технологии управления	110,9	н/д	н/д	80,4	н/д	н/д	37,9	Россети, Атомэнергопром, МОЭСК, ФСК ЕЭС
15	Норбит	79,4	33,7	45,7	49,5	23,7	25,8	60,4	н/д

Рисунок 2 - выручка участников российского рынка BI-систем

Также, можно отметить, что одним из самых популярных инновационных направлений на аналитическом рынке является интеграция BI-систем с инструментами машинного обучения. Применение данных инструментов позволяет найти скрытые закономерности в больших объёмах данных. Таким образом, данные инструменты будут полезны при настройке рекомендательных сервисов.

1.3 Обзор задач, решаемых BI

BI-системы имеют довольно широкий спектр применения и задач, из которых можно выделить следующие:

- поддержка развития бизнес-процессов;
- анализ и корректировка оперативных целей;
- возможность моделирования различных бизнес-ситуаций;

- проведение оперативного анализа;
- снижение нагрузки на сотрудников, таким образом освобождая их время для решения более сложных задач;

- возможность масштабирования данных.

Также, в области поддержки стратегического развития предприятия:

- оценка эффективности бизнеса;
- оценка достижимости поставленных задач;
- бизнес-моделирование и оценка инвестиционных проектов.

ВІ-системы используются в производстве, а именно в прогнозировании продаж и формировании оптимального производственного плана по тому или иному типу товара.

В финансовом секторе, ВІ-системы используются, чтобы проанализировать базу клиентов, а также, спрогнозировать спрос на услуги банка. При помощи данных систем можно выявить потенциального потребителя, провести анализ по потенциальным заёмщикам.

В государственном секторе, ВІ-системы используются, чтобы накапливать данные и проводить статистические исследования по городам и регионам той или иной страны.

В области энергетики можно отметить, что ВІ-системы могут выявить и спрогнозировать расходование электроэнергии по регионам, также, можно проводить анализ состояния приборов и формировать отчётную документацию.

В области медицины, ВІ так же играет немаловажную роль. Системы позволяют формировать прогноз по тому или иному заболеванию, формировать отчёты и закономерности по различным болезням.

В логистике, ВІ-системы используют в основном для того, чтобы контролировать своевременность погрузки/выгрузки различных товаров перевозчиком, отслеживать и прогнозировать сроки доставки того или иного товара.

Подведя итог, можно сказать что ВІ-системы могут выполнять рассмотренный набор задач в самых разных сферах деятельности. В рамках данной работы, были изучены все из перечисленных задач для дальнейшей разработки приложения.

1.4 Постановка задачи

Отталкиваясь от задач, существующих и решаемых ВІ-системами, можно сформулировать задачу на данную работу. Основной задачей данной работы является разработка приложения, позволяющего выводить на экран статистику эмоционального состояния студентов в определённый интервал времени как в виде таблицы, так и в графическом виде (гистограмма, график, диаграмма). Данное приложение должно иметь веб-интерфейс, а также, использовать современный инструментарий для визуализации полученных с сервера данных.

Также для разработки приложения будет необходим сервер, позволяющий отправлять и принимать запросы из удалённой базы данных.

Выводы по первой главе

Таким образом, на основании рассмотренного теоретического материала, можно сделать следующие выводы.

Современные информационные системы предоставляют пользователям всё больше возможностей в области анализа текущей ситуации компании через анализ основных показателей. Также, они помогают в принятии правильных и своевременных решений, мер по улучшению положения бизнеса при возникновении определённых трудностей и проблем.

ВІ является одним из самых главных факторов, способствующих эффективной работе любого предприятия в любой сфере. Использование ВІ-систем позволяет не только проанализировать эффективность работы, но и спрогнозировать дальнейший рост или падение эффективности.

Была показана актуальность ВІ-технологий на примере построения отчётов предприятия, приведения статистики предприятия, прогнозирования различных событий, в том числе и связанных с экономикой, медициной, логистикой и энергетикой.

Также, были приведены основные отличительные особенности ВІ, и преимущества. Были рассмотрены задачи ВІ, а также их применяемость в разных сферах деятельности.

2. ОСНОВНЫЕ АЛГОРИТМЫ BI

2.1 Концепция BI систем

Сегодня на рынке существует несколько мощных BI-платформ, предоставляющих большие функциональные возможности пользователю. Согласно рисунку 2.1[16], концепция платформы BI содержит в себе источник данных OLTP, базу данных (хранилище), OLAP-куб, а также, клиентскую часть с соответствующим пользовательским интерфейсом.

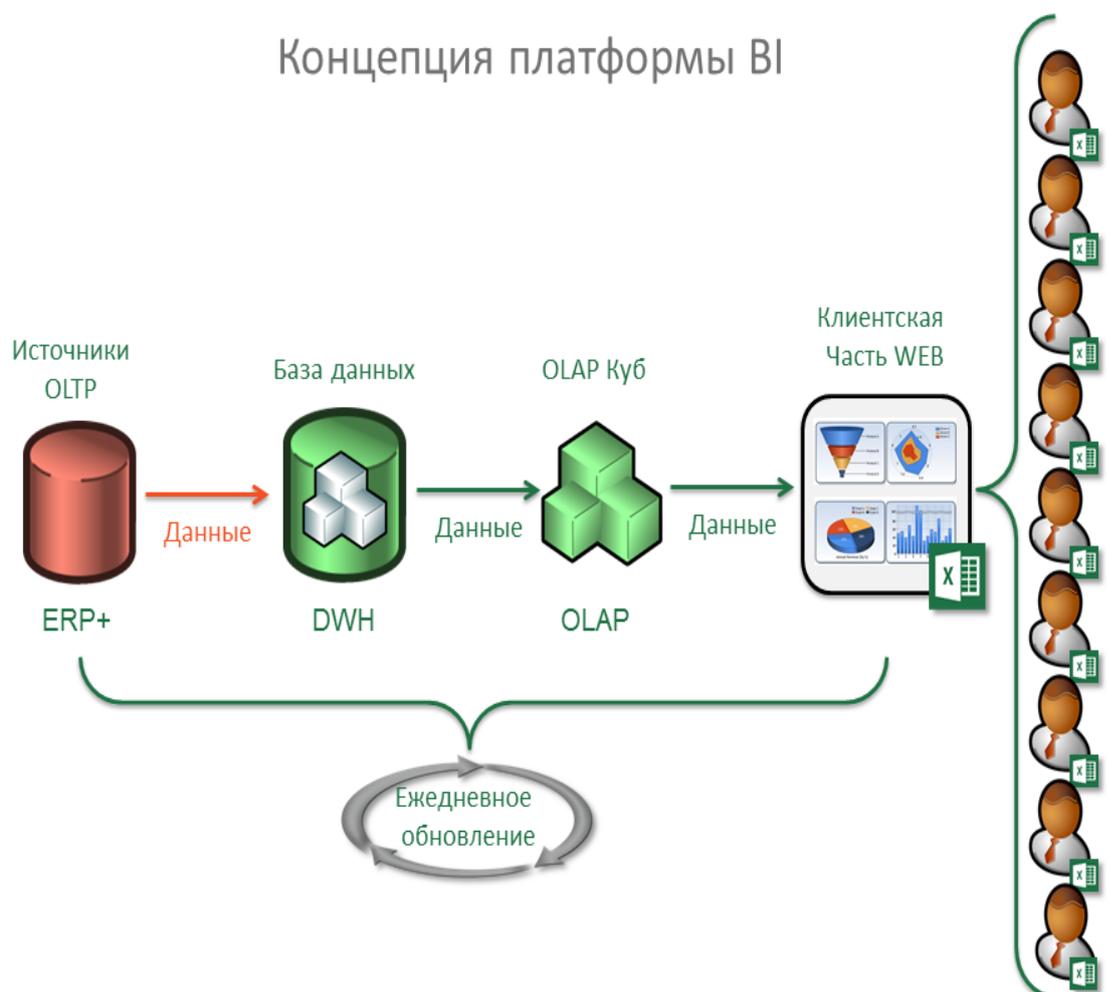


Рисунок 2.1 – Концепция платформы BI

- **OLTP:** OLTP (Online Transaction Processing) – метод организации баз данных, который позволяет работать с небольшими по размерам транзакциями, что значительно ускоряет процесс обработки данных.

Источники OLTP представляют из себя приложения, позволяющие выполнять различные задачи в разных отраслях деятельности, в том числе, банковские системы, системы ERP (Enterprise Resource Planning, планирование ресурсов предприятия) и тд. Такие приложения позволяют автоматизировать повторяющиеся задачи, такие как фиксация документов, учёт расходов.

- **DWH-хранилище:** Data Warehouse – хранилище, представляющее собой информационную базу данных, разработанную для реализации задач BI

- **OLAP-кубы:** OLAP (Online Analytical Processing, интерактивная аналитическая обработка) – система обработки данных, позволяющая делать в реальном времени (1-5 секунд) любые отчеты и проводить полноценный анализ данных. Главным преимуществом является высокая скорость работы благодаря реализации OLAP-куба.

- **клиентская часть WEB:** как правило, для детального анализа данных и построения динамических отчетов пользователи используют Сводные таблицы Microsoft Excel, подключенные к OLAP-кубам. Для поверхностного анализа и визуализации ключевых показателей также используются WEB-приложения, которые должны поддерживать доступ к отчетам с любого устройства: компьютер, планшет, телефон.

2.2 Архитектура BI систем

В центре BI-системы находится модель данных (также называемая репозиторием). Модель данных представляет из себя описание модели и её привязку к различным источникам данных. Данная модель обычно состоит из трёх слоёв: презентационный слой, бизнес-слой и физический слой.

Презентационный слой модели позволяет разбить логическую модель на разные предметные области, таким образом ограничив доступ пользователей к показателям.

Бизнес-слой позволяет отразить логическую модель в форме многомерной модели, содержащей факты и измерения. Также, в данном слое описывается привязка логических атрибутов к их физическим источникам.

Физический слой содержит в себе описание источников данных – таблиц, полей, ключей, кубов данных.

При построении отчёта пользователем, сервер презентаций генерирует запрос к серверу BI. В последствии, сервер разбирает данный запрос и переводит его в SQL запрос, таким образом позволяя обратиться в базу данных и извлечь данные. Далее, результат возвращается на сервер презентаций, который в свою очередь генерирует отчёт, представляя полученные данные в графическом формате.

В качестве примера BI-системы, использующей такую архитектуру, была выбрана Oracle Business Intelligence. Это популярная платформа для решения BI-задач, создания интерактивных отчётов, мониторинга бизнес-процессов.

На рисунке 2.2 показана архитектура BI-системы Oracle Business Intelligence [15].

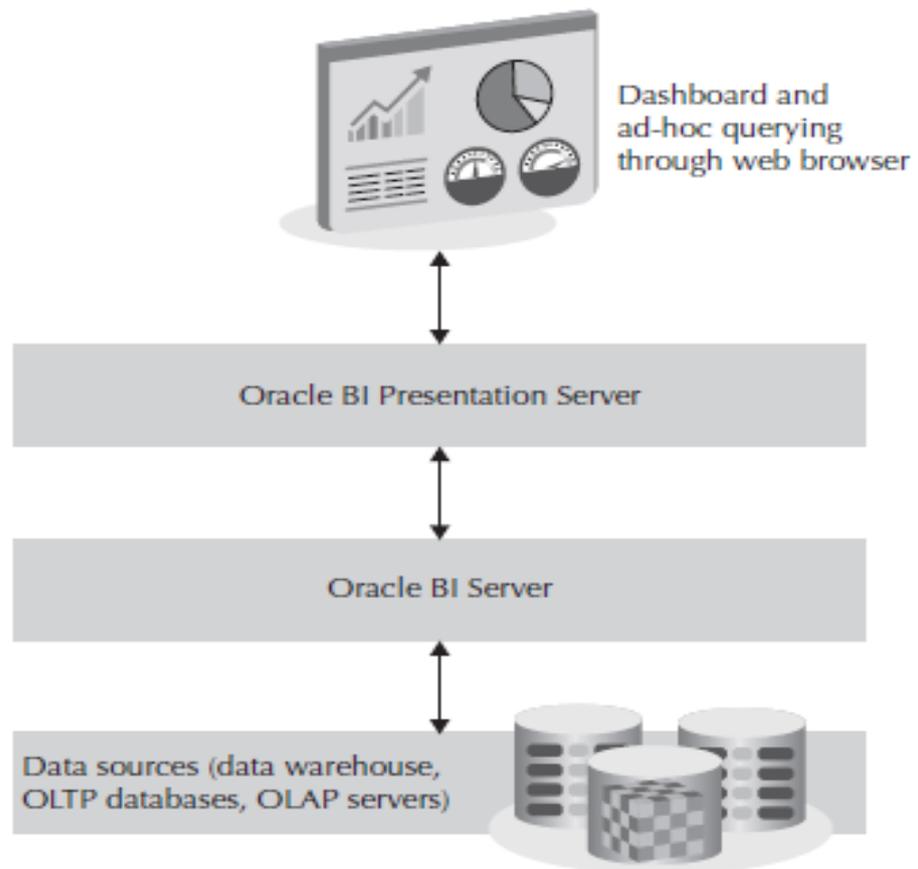


Рисунок 3.2 - Архитектура BI-системы Oracle BI

В данном случае, сервер OBIEE включает в себя несколько разных компонентов, которыми управляет сервер приложений Weblogic. Также, показана реляционная база (Oracle, MS Sql, MySQL).

Всё управление OBIEE берёт на себя веб-интерфейс Weblogic Administration Console.

2.2 Обзор BI платформ

На сегодняшний день доступно большое количество платформ для разработки BI систем. Все они позволяют в полной мере отобразить огромное количество данных на экране таким образом, чтобы пользователь имел возможность не только посмотреть эти данные, но и вовремя спрогнозировать дальнейшие изменения.

Каждая из представленных платформ имеет как преимущества, так и недостатки. В данной главе будут рассмотрены несколько BI-систем, работающих по различным алгоритмам. Некоторые из них более

эффективны, но обычно и цена на такие платформы выше, чем у конкурентов.

2.2.1 QlikView

Данная платформа позволяет пользователю получать данные, при этом сохраняя связи между ними. Таким образом, помимо непосредственно данных, запрошенных пользователем, платформа также предоставляет информацию, имеющую к ним отношение и возможно также будет полезна. Система также содержит в себе встроенное ETL приложение, с помощью которого можно легко управлять подключенным хранилищем данных.

Пример пользовательского интерфейса, а именно, построенного пользователем отчёта, представлен на рисунке 2.2 [20]

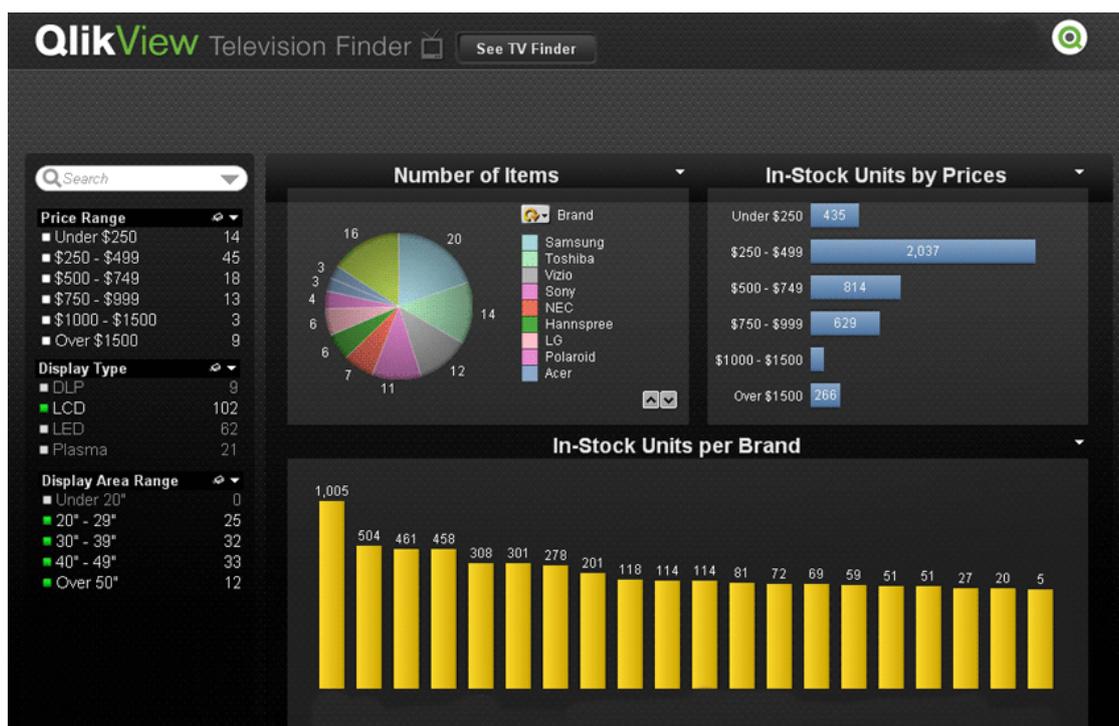


Рисунок 2.2 - Отчёт, построенный при помощи QlikView

Данная система имеет ряд особенностей, которые и выделяют её на фоне остальных. К примеру, огромное количество разнообразных функций для построения дашбордов. Именно благодаря этому, процесс создания

отчёта становится быстрее и удобнее для пользователя. Также, дашборды QlikView очень просты для начинающего пользователя.

Также, на скорость обработки данных влияет и то, что при построении отчёта все данные хранятся в оперативной памяти сервера, что ускоряет выполнение того или иного запроса в несколько раз и, как следствие, отчёт будет сформирован быстрее. Конечно, может показаться, что данный подход неэффективен, однако QlikView сжимает все хранимые данные, вследствие чего они не занимают много места.

Более того, платформа позволяет агрегировать данные на лету, в процессе формирования запроса. Это намного быстрее, нежели сначала агрегировать данные, и только потом получать их.

Наиболее ярко выраженные преимущества и недостатки данной платформы представлены на рисунке 2.3 [20]

QlikView Pros & Cons		
Со стороны:	⊕ PROS	⊖ CONS
ПОЛЬЗОВАТЕЛЯ	<ul style="list-style-type: none"> + Привлекательный интерфейс + Легко фильтровать данные при любой визуализации + Скорость создания графиков и таблиц + Возможность отправлять отчеты в удобном формате .pdf 	<ul style="list-style-type: none"> - При фильтрации может комбинировать некоторые типы данных, когда это не требуется - Нет возможности объединять результаты в закладках - Пользователь без технической базы чаще всего будет испытывать трудности
РАЗРАБОТЧИКА	<ul style="list-style-type: none"> + Поддерживает импорт данных из множества источников + Удобство создания любых графиков, таблиц и фильтров + Великолепная скорость загрузки и обработки данных, даже при миллионах строк + Позволяет совместную разработку 	<ul style="list-style-type: none"> - Синтаксис неочевиден, а выражения выглядят необычно - Некоторые базовые функции очень трудно либо невозможно реализовать - Сложности с использованием в качестве рабочего инструмента для всей компании - Отправлять отчеты просто ужасно

Рисунок 2.3 – QlikView - основные преимущества и недостатки платформы

2.2.2 Klipfolio

Данная платформа была разработана канадской компанией Klipfolio Inc. (Оттава). Так же, как и остальные BI платформы, позволяет строить дашборды. Однако в отличие от предыдущей, является полностью онлайн

платформой, так как все вычисления и хранение данных располагается на серверной части. Платформа позволяет создавать дашборды с графиками, диаграммами и тд. К плюсам можно отнести отсутствие программирования, таким образом даже начинающий пользователь сможет через web-интерфейс работать с программой и с легкостью строить необходимые ему отчёты.

Большим плюсом является то, что программа поддерживает различные источники, как онлайн (Google Sheets), так и оффлайн (MS Excel, XML, JSON). Также, имеется возможность подключения сторонних сервисом.

Пример пользовательского интерфейса данной платформы представлен на рисунке 2.4 [20]

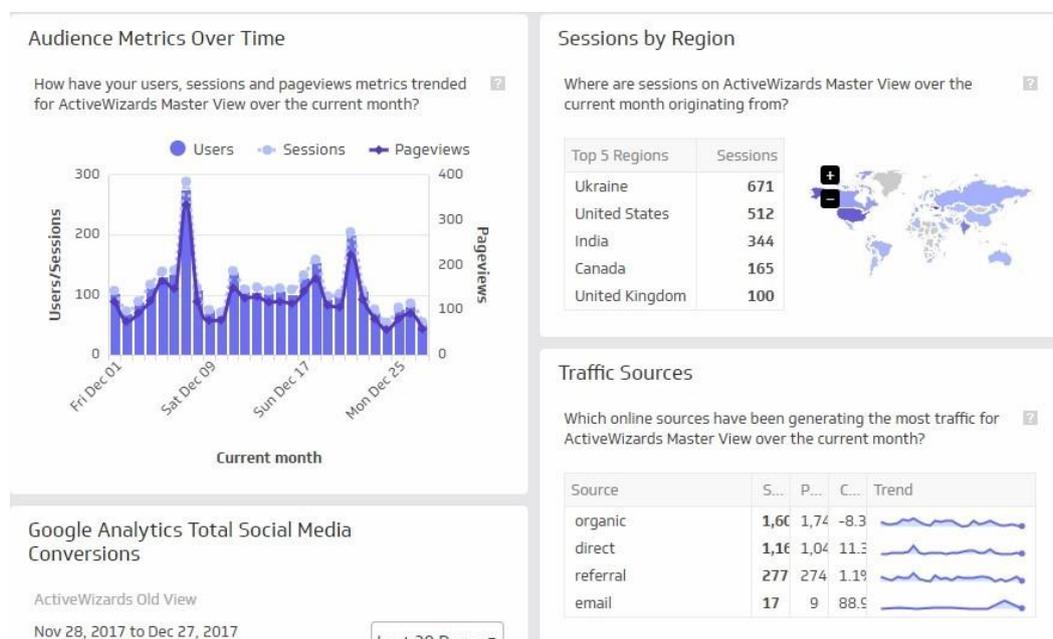


Рисунок 2.4 - построение дашборда в Klipfolio

Klipfolio позволяет сразу нескольким пользователям в режиме реального времени создавать бизнес-дэшборды без каких-либо профессиональных навыков программирования. Данная платформа полностью ориентирована на повышение эффективности предприятий в разных сферах.

Благодаря возможности подключения сторонних сервисов, платформа становится практически идеальным вариантом для составления бизнес отчётов.

Основные преимущества и недостатки данной платформы представлены на рисунке 2.5 [20]

Klipfolio Pros & Cons	
⊕ PROS	⊖ CONS
<ul style="list-style-type: none">+ Чрезвычайно гибкий по части дизайна, интерфейса, вычислений и совместимости+ Привлекательный интерфейс для создания дашбордов+ Качественная и быстрая поддержка+ Постоянно совершенствуется, регулярно добавляются новые функции и опции+ Поддержка API и многих популярных форматов данных+ Анализ данных в реальном времени+ Легко интегрировать сторонние приложения и ПО, включая Adobe, Salesforce и CampaignMonitor	<ul style="list-style-type: none">- Учиться как базовым, так и сложным операциям придется довольно долго- Дополнительная плата за кастомизированные CSS стили- Плата за пользователя остается неизменной только для менее чем 10 человек- Нет функций для ad hoc анализа- Нет функций для предиктивной аналитики и прогнозирования- Все расчеты проводятся через извилистые формулы в отличие от простых drag-and-drop решений конкурентов

Рисунок 2.5 - плюсы и минусы Klipfolio

2.2.3 D3.js

D3 расшифровывается как Data-Driven Documents. При этом акцентируется внимание именно на работе с данными, несмотря на то, что ключевым аспектом при выборе данной библиотеки являются именно мощные возможности для визуализации.

Данный набор инструментов состоит из нескольких десятков модулей, решающих различные задачи. Универсальной данную библиотеку делает также то, что кроме непосредственно модулей для построения диаграмм и графиков, в составе есть модули для управления элементами на веб-странице (аналог jQuery), загрузки данных (аналог ajax), форматирования и масштабирования данных, математических функций. На рисунке 2.5 показан пример построенного при помощи библиотеки D3.js графика. Хотелось бы отметить, что данный пример используется как образец при создании более сложных графиков.

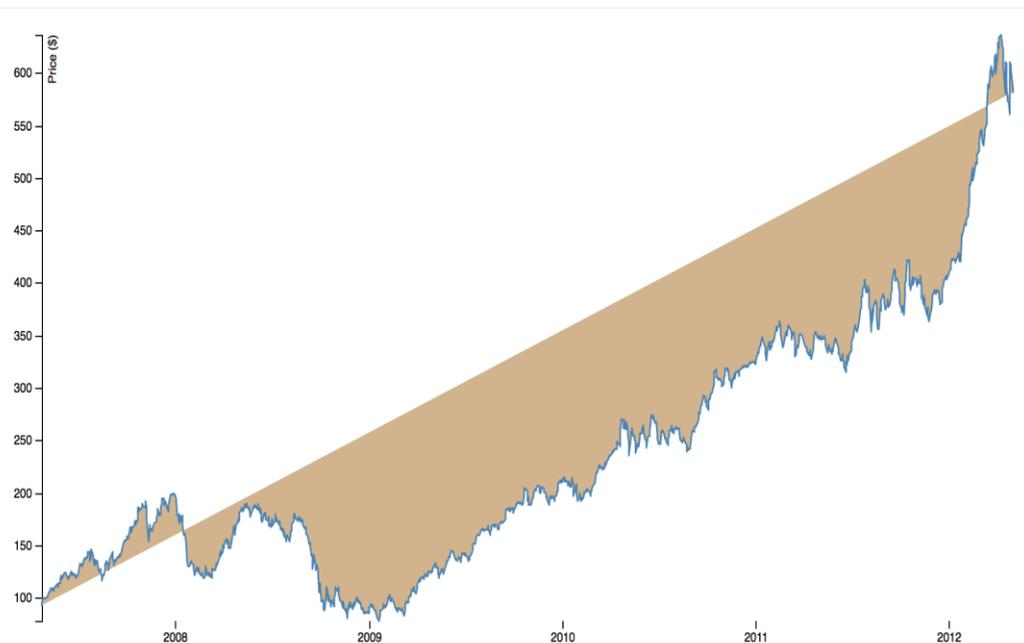


Рисунок 2.5 – Пример графика на D3.js

D3 может работать с любым типом и набором входных данных (Json, csv, tsv и другие). В то время как другие подобные решения используют элемент canvas, данная библиотека прежде всего построена на использовании Javascript, SVG и CSS. Основное отличие состоит в том, что canvas использует пиксели, а SVG использует векторы. Таким образом, D3.js работает с векторной графикой, что является одним из основных преимуществ.

Однако данная библиотека имеет достаточно высокий порог вхождения, что и является большим и основным недостатком данного инструмента. Начинающему разработчику BI будет необходимо время, чтобы понять как создаются сложные интерактивные дэшборды для визуализации.

На рисунке 2.6 приведён сводный анализ рассмотренных систем по основным критериям.

	QlikView	KlipFolio	D3
Поддерживаемые операционные системы			
Windows	X	X	X
Android		X	
iOS	X	X	X
MacOS	X	X	X
Web	X	X	X
Поддержка			
Онлайн	X	X	
Документация		X	X
Видео уроки	X	X	X
Интеграция			
REST API		X	X
Поддержка языков	JavaScript		JavaScript
Лицензия			
	Personal - Free	Пробный период 2 недели	Free

Рисунок 2.6 - сводный анализ приведённых систем

Рассмотрев данные платформы, а также, проанализировав их основные преимущества и недостатки, можно принять решение о выборе необходимой для выполнения задачи данной работы платформы.

2.3 Выбор платформы для разработки BI-системы с web интерфейсом

В ходе данной работы была сформулирована и поставлена задача – разработать веб-приложение для визуализации данных, получаемых с удалённого сервера. В качестве визуального представления были выбраны

таблицы, гистограммы и диаграммы. Для решения данной задачи, необходимо выбрать подходящий набор инструментов.

Рассмотрев различные варианты платформ для визуализации данных, было установлено, что в рамках данной работы, для разработки веб-приложения больше всего подходит библиотека D3.js.

Данная библиотека является бесплатной, существует много обучающего материала и примеров на официальном сайте разработчика, D3 предоставляет неплохой функционал для визуализации данных и может использовать данные из самых различных источников.

D3 представляет из себя мощный набор инструментов для визуализации данных. Он состоит из множества модулей, позволяющих решать различный набор задач визуализации. Помимо модулей построения фигур, также, есть модули для работы с элементами страницы, загрузки данных, форматированием и масштабированием данных, математическими функциями.

В D3 визуализация строится в виде векторной графики, в формате SVG. Данный формат позволяет строить и работать с простейшими фигурами. Благодаря данным фигурам можно строить более сложные графики, используя примитивы.

На рисунке 2.7 изображена простая линия, к которой применены дополнительные функции трансформации [19]. Подобный график является более сложным по сравнению с обычными, так как использует дополнительные инструменты.

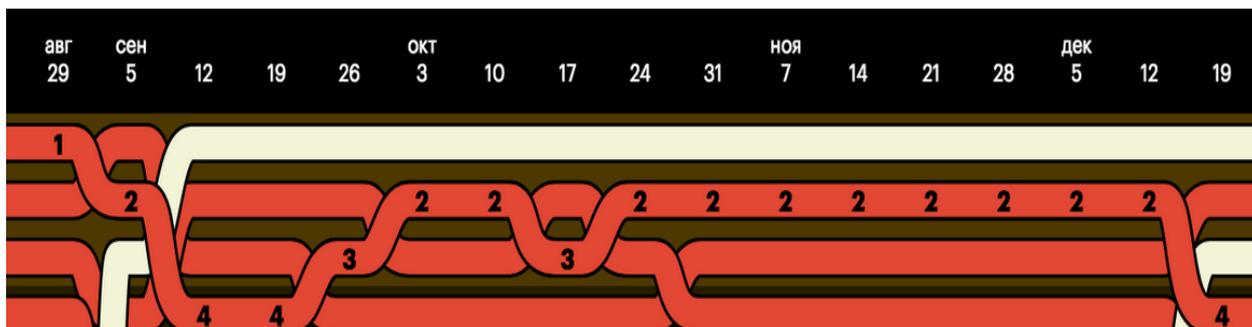


Рисунок 2.7 - Модернизированная линия, построенная на D3.js

На рисунке 2.8 показан более сложный график, состоящий из нескольких элементов, гистограммы и круговой диаграммы [19]. Данный график содержит в себе несколько svg-элементов и предоставляет интерактивность с каждым из них.

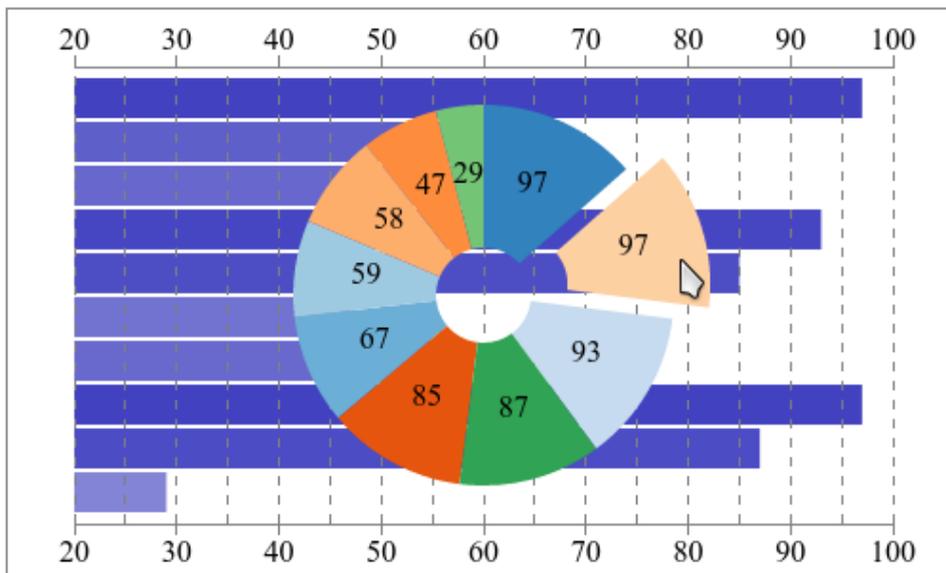


Рисунок 2.8 - График, построенный на D3.js

Представленные на рисунке 2.8 графики – это лишь малая часть из возможностей D3.js. Данные графики можно комбинировать и совершенствовать, усложняя их. Это возможно благодаря мощным средствам визуализации.

Выводы по второй главе

Во второй главе была рассмотрена концепция BI-систем, была приведена их архитектура на примере такой популярной платформы как Oracle Business Intelligence. Также, были представлены основные платформы для разработки, и был проведен их анализ.

В результате, для решения задачи данной работы, была выбрана платформа D3, так как именно она позволяет встроить в разработанное на JavaScript веб-приложение графические элементы для визуализации данных.

3. РЕАЛИЗАЦИЯ ИНТЕРФЕЙСА VI СИСТЕМЫ

3.1 Функциональные требования к приложению

В рамках данной работы должно быть реализовано VI-приложение с веб интерфейсом.

Функциональные требования к приложению:

- на главной странице расположен интерфейс, с которым взаимодействует пользователь;
- интерфейс содержит: кнопки выбора входных данных, окно выбора запрашиваемого пользователем визуального представления данных (таблица, гистограмма, круговая диаграмма), кнопки сортировки графического представления и таблицы;
- при нажатии на конкретный элемент графического представления данных, появляется подсказка-окно с уточняющей информацией по данному элементу;
- гистограмма может быть отмасштабирована по желанию пользователя;

Диаграмма вариантов использования приложения представлена на рисунке 3.1.

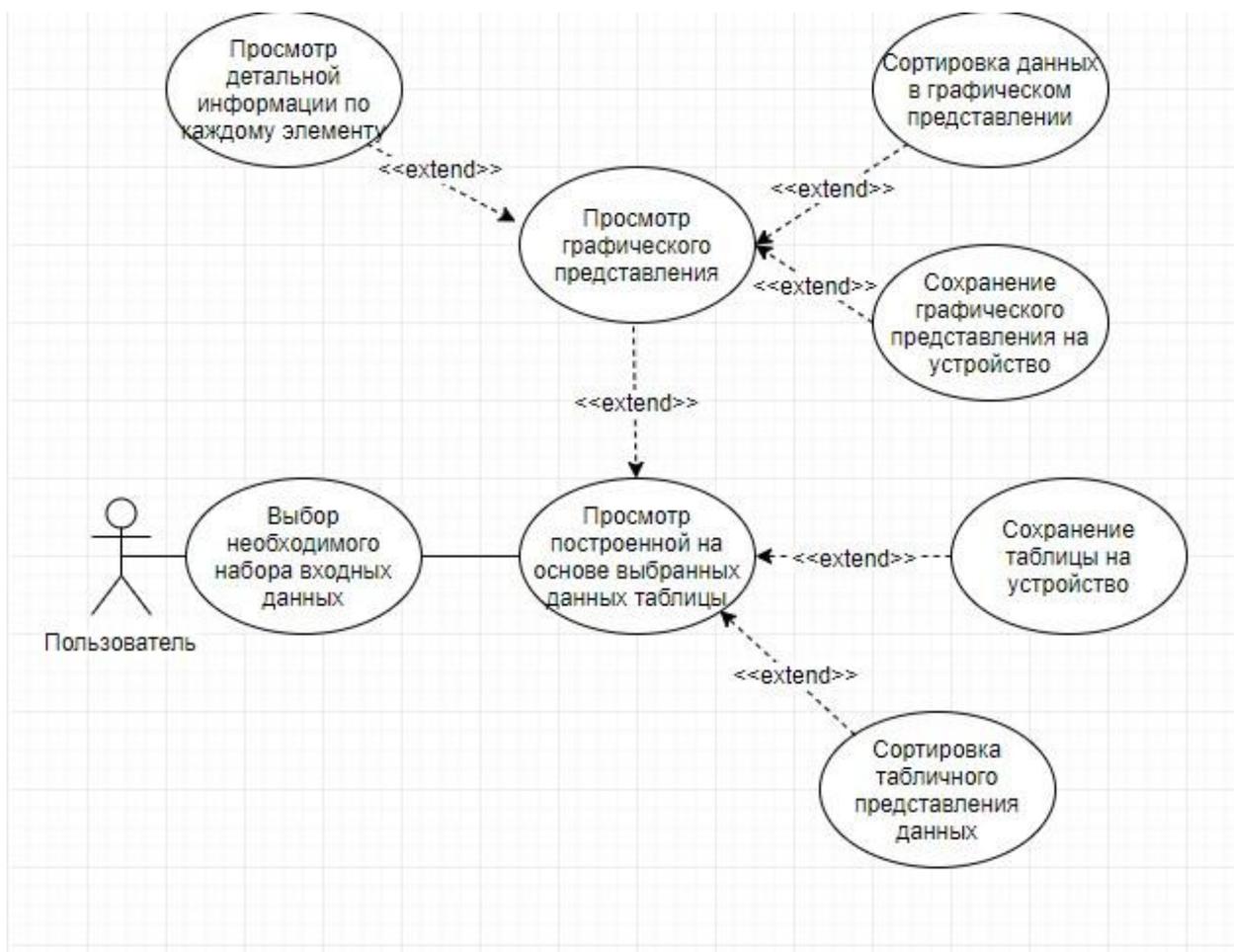


Рисунок 3.1 - диаграмма вариантов использования приложения

На диаграмме можно увидеть следующие варианты использования:

- выбор необходимого набора входных данных:

Краткое описание. При запуске веб-приложения, пользователь получает возможность выбора набора входных данных, например, студент из какой-либо группы.

- просмотр построенной на основе выбранных данных таблицы:

Краткое описание. После выбора набора входных данных, приложение строит таблицу на их основе.

- просмотр графического представления:

Краткое описание. После построения таблицы, пользователь имеет возможность построить графическое представление выбранных им данных и занесённых приложением в таблицу. Графическое представление

представляет из себя гистограмму, содержащую нормализованное значение эмоционального состояния студента в диапазоне от 0 до 100.

- просмотр детальной информации по каждому элементу:

Краткое описание. После просмотра графического представления, пользователь имеет возможность просмотреть детальную информацию по каждому из элементов построенной на основе выбранных им данных гистограммы.

- сортировка данных в графическом представлении:

Краткое описание. После построения графического представления данных, пользователь имеет возможность отсортировать гистограмму по возрастанию значений/убыванию значений.

- сохранение графического представления на устройство:

Краткое описание. После построения графического представления данных, пользователь имеет возможность сохранения гистограммы на устройство (в формате png).

- сортировка табличного представления данных:

Краткое описание. После построения таблицы, пользователь имеет возможность отсортировать таблицу по определённому столбцу по возрастанию значений/убыванию значений.

- сохранение таблицы на устройство:

После построения таблицы, пользователь имеет возможность сохранения таблицы на устройство (в формате xls).

3.2 Архитектура приложения

При разработке данного приложения от выбора его архитектуры зависит реализация и скорость его работы. Таким образом, выбор архитектуры приложения является одним из главных процессов в его разработке. В настоящее время подавляющее число подобных приложений имеют трёхуровневую архитектуру. Именно такая архитектура и была использована в конечном результате при разработке веб-приложения.

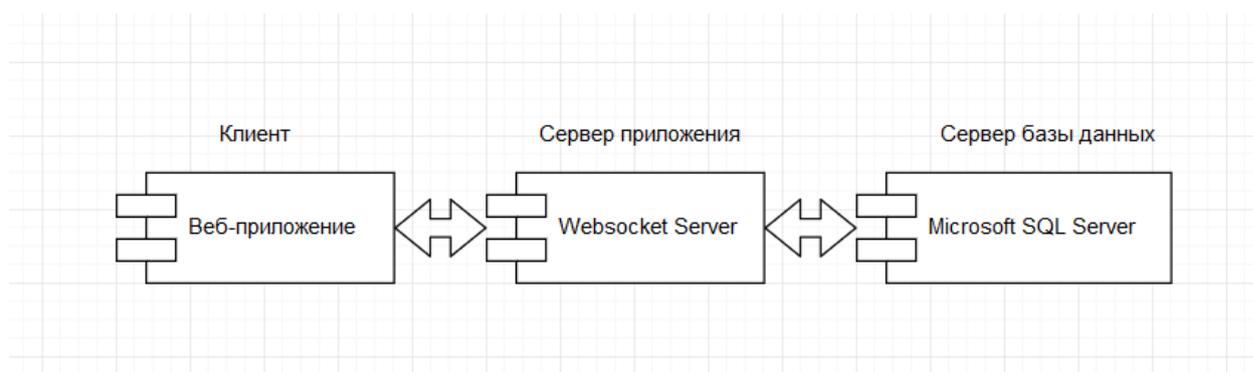


Рисунок 3.5 - Трёхуровневая архитектура приложения

При разработке данного приложения за основу взят язык JavaScript. Для визуализации получаемых с сервера данных (построения таблиц, гистограмм и диаграмм), была использована библиотека D3.js. В качестве клиента в данном случае представлена веб-страница, отправляющая запрос на сервер.

Сервер базы данных представляет собой внешнюю систему, подключение к которой производится удалённо. В качестве сервера базы данных используется Microsoft SQL Server. Данный инструмент был выбран в качестве сервера базы данных, так как в отличие от других СУБД, Microsoft SQL Server гарантирует повышенную безопасность и производительность средств разработки, а также, содержит более мощные инструменты для работы с данными. Кроме того, можно также выделить следующие преимущества:

- масштабируемость;

- производительность;
- менее уязвимая база данных;
- наличие инструментов бизнес-аналитики.

Структура запроса представляет из себя JSON файл с набором данных, необходимых серверу для идентификации клиента и дальнейшей отправки необходимых ему данных.

```

{
  "IdList": [
    1
  ],
  "MessageType": 1,
  "From": "2019-04-01T12:21:44.7252327+04:00",
  "Till": "2019-05-11T12:21:44.7254368+04:00"
}

```

Рисунок 3.6 - Структура JSON-файла, запроса клиента к серверу

В качестве сервера приложения был использован WebSocket-сервер, позволяющий подключаться к удалённому серверу базы данных, отправляя запросы и получая ответы по протоколу WebSocket.

3.3 Технология WebSocket

WebSocket – представляет из себя протокол для обмена сообщениями между клиентской частью и сервером. В роли клиентской части может выступать как веб-браузер, так и любое другое приложение, которое при помощи соответствующего API имеет возможность подключения к WebSocket-серверу.

Именно на основе данной технологии была проведена разработка веб-приложения.

Данный протокол основан на протоколе TCP. При помощи данной технологии клиент получает возможность взаимодействовать клиенту с сервером в реальном времени.

Изначально WebSocket разрабатывался как синхронный протокол, который был основан на модели «Запрос-ответ» (рисунок 3.2), однако теперь

он является полностью асинхронным, так как больше нет WebSocket-сервера, как такового. При использовании данной технологии, мы получаем двух равноценных участника обмена данными, каждый из которых работает независимо от другого и может отправлять данные в любой момент времени.

Данная технология получила широкое применение в следующих разработках:

- веб-приложения с постоянным обменом большого количества данных,
- приложения, имеющие множество асинхронных элементов на странице;

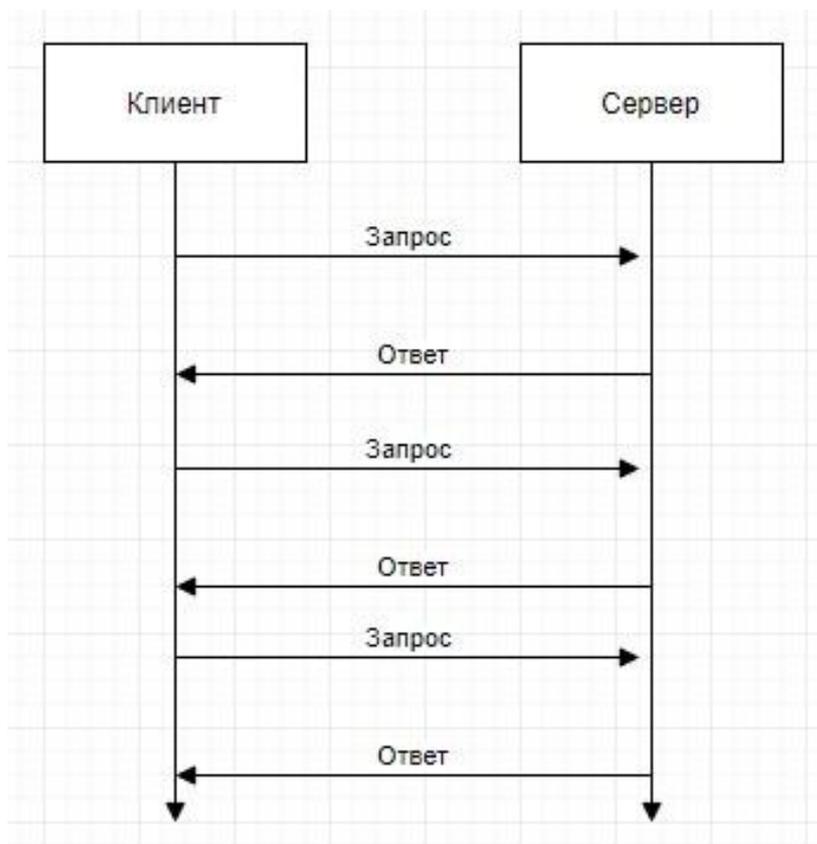


Рисунок 3.2 - Модель "Запрос - ответ"

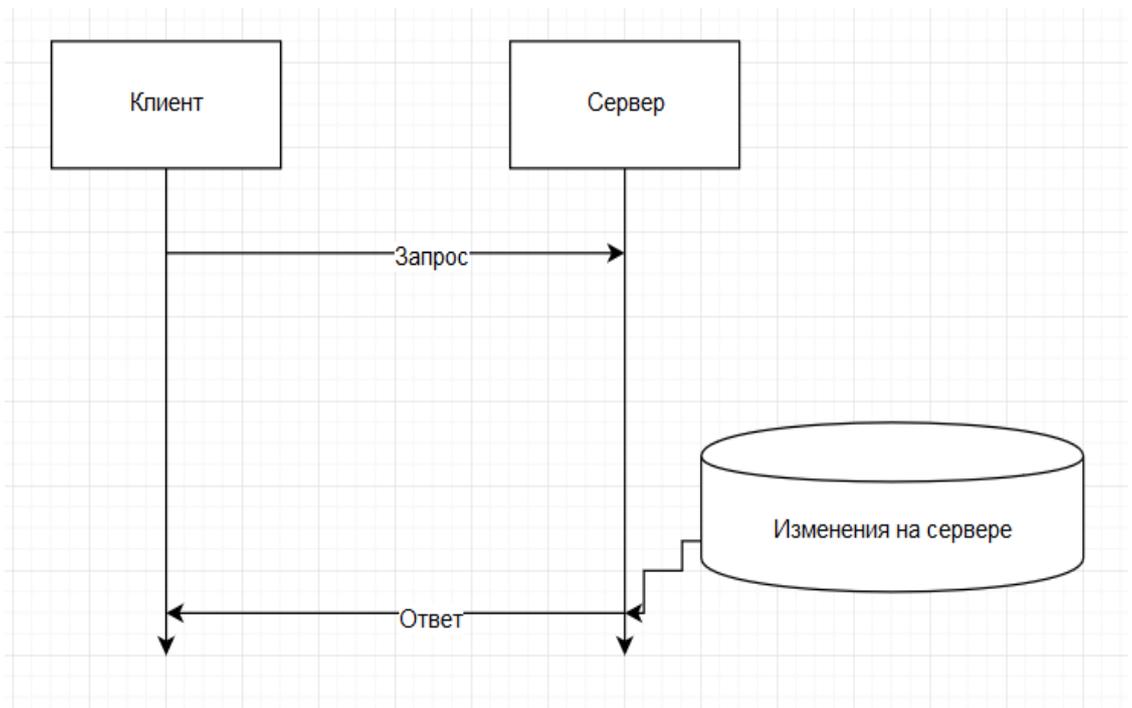


Рисунок 3.3 - Асинхронная модель приложения

```

<script>
  ws = new WebSocket("ws://localhost");

  // Соединение установлено
  ws.onopen = function() { alert("Connection opened...") };

  // Соединение закрыто
  ws.onclose = function() { alert("Connection closed...") };

  // Получено сообщение
  ws.onmessage = function(evt) { $("#msg").append("<p>"+evt.data+"</p>"); };
</script>

```

Рисунок 3.4 - Простейшая реализация WebSocket-клиента

На рисунке можно увидеть то, как в JavaScript реализован WebSocket-клиент. Реализация представляет из себя несколько функций:

Ws.onopen – данная функция будет вызвана как только будет установлено новое соединение с WebSocket-клиентом.

Ws.onclose – данная функция будет вызвана как только соединение с WebSocket-клиентом будет разорвано.

Ws.onmessage – данная функция будет вызвана как только будет получено новое сообщение по протоколу ws. В теле данной функции можно реализовать обработку входного сообщения и, как вариант, разбор принятого

файла на строки для дальнейшего использования в таблице или ином представлении данных.

Таким образом, чтобы создать простое WebSocket-приложение для обмена данными, нет необходимости в постоянной отправке запросов от клиента на сервер. Технология WebSocket позволяет легко обойтись без этого при помощи асинхронного протокола ws.

3.4 Реализация приложения

После исследования, согласно поставленной задаче на данную работу, необходимо разработать веб-приложение. Был выбран язык JavaScript, как основа для данного приложения, а также, мощный инструмент для визуализации D3.js.

Сначала была создана веб-страница с помощью HTML-разметки. Были подготовлены все необходимые элементы для последующего размещения в них блоков с графическим представлением данных.

Далее, для построения таблицы были использованы стандартные средства языка JavaScript. На рисунке 3.5 приведён листинг алгоритма создания таблицы. Функция «CreateTable» создаёт таблицу на основе шаблона «tableTemplate».

```
var tableTemplate=$( [
  "<table class='table table-stripped'>",
  "<caption></caption>",
  "  <thead><tr/></thead>",
  "  <tbody></tbody>",
  "</table>"
].join('\n'));
CreateTable = function(data,variablesInTable,title){
  //console.log(variablesInTable);
  var table=tableTemplate.clone();
  var ths=variablesInTable.map(function(v){return $("<th>").text(v)});

  $('caption',table).text(title);
  $('thead tr',table).append(ths);
}
```

Рисунок 3.5 - листинг алгоритма создания таблицы

Для реализации отправки запроса на сервер, необходимо установить WebSocket-соединение, позволяющее асинхронно обмениваться

сообщениями между серверной и клиентской частью. На рисунке 3.6 приведён листинг алгоритма установки WebSocket-соединения с сервером.

```
var socket = new WebSocket("ws://127.0.0.1");
socket.onopen = function () {
    console.log("Соединение открыто");
};
socket.onclose = function () {
    console.log ("Соединение закрылось");
};
socket.onmessage = function (event) {
    console.log ("Пришло сообщение с содержанием:", event.data);
    var json = '{"student":studentId,"time":time}',
        inputdata = JSON.parse(json);
});
```

Рисунок 3.6 - Листинг алгоритма Websocket-соединения

При получении ответа от удалённого сервера, клиент получает JSON-файл с данными, соответствующими эмоциональному состоянию студентов на протяжении выбранного периода времени. Изначально данный показатель представляет собой дробное число и находится в диапазоне от 0 до 1. При построении таблицы и гистограммы, для удобства восприятия человеком, полученные числа были умножены на 100. В отчёте, показатель эмоционального состояния студента представляет собой число в диапазоне от 0 до 100. Чем выше данное число, тем выше уровень эмоционального состояния студента.

Таким образом, была построена таблица (рисунок 3.7), позволяющая отследить зависимость эмоционального состояния определённого студента в зависимости от времени и посещаемого студентом занятия.

Эмоциональное состояние объекта 'студент-1'		
Lesson	Time	Emotion
1	8:30	90
1	9:00	85
1	9:30	84
2	10:00	82
2	10:30	75
2	11:00	63
3	11:30	88
3	12:00	86
3	12:30	80
4	13:00	79
4	13:30	61
4	14:00	50
5	14:30	43
5	15:00	42
5	15:30	25
5	16:00	50

Рисунок 3.7 - построенная таблица

Далее, необходимо построить графическое представление данной таблицы для наглядности отображаемых данных. Для этого, были использованы средства D3.js, а именно, элемент `svg`, позволяющий работать с векторной графикой. На рисунке 3.9 показан листинг алгоритма построения `svg`-элемента, а также, заполнение его гистограммой. В качестве элементов гистограммы выступают прямоугольники (элемент «`rect`»), для которых задаются их основные параметры, такие как длина, ширина и их координаты.

```

var buildSvg=function() {
  if(!built){
    xScale=d3.scaleBand()
      .domain(d3.range(dataset.length))
      .rangeRound([0,w])
      .padding(0.05);
    svg=d3.select("div.bars").append("svg");
    svg.attr("width",w).attr("height",h);
    svg.selectAll("rect").data(dataset)
      .enter().append("rect")
      .attr("x",function(d,i){
        //return i*(w/dataset.length);
        return xScale(i);
      })
      .attr("y",function(d){return h-d*2;})
      .attr("width",w/dataset.length-barPadding)
      // .attr("width",xScale.rangeRound())
      .attr("height",function(d){return d*4;})
      .attr("fill",function(d){return "rgb(0,0,"+(d*2.5)+")"});
    //var texts=svg.selectAll("text").enter();
  }
}

```

Рисунок 3.9 - Листинг алгоритма построения svg-элемента и его заполнение

Так как данная гистограмма также должна иметь возможность сортировки, необходимо написать функцию, позволяющую перемещать построенные прямоугольники в соответствие с их значением.

На рисунке 3.10 показан листинг алгоритма сортировки гистограммы.

```

var sortBars=function() {
  sortOrder=!sortOrder;
  svg.selectAll("rect").sort(function(a,b) {
    if(sortOrder){return d3.ascending(a,b);}else{return d3.descending(a,b);}
  }).transition().delay(function(d,i){return i*50;}).duration(1000).attr("x",function(d,i){
    return xScale(i);
  })

  svg.selectAll("text.title").sort(function(a,b) {
    if(sortOrder){return d3.ascending(a,b);}else{return d3.descending(a,b);}
  }).transition().delay(function(d,i){return i*50;}).duration(1000).attr("x",function(d,i){
    return i*(w/dataset.length)+(w/dataset.length)/2-5;
  })
}

```

Рисунок 3.10 - Листинг алгоритма сортировки гистограммы

Таким образом, графическое представление данных в виде гистограммы было реализовано. На рисунке 3.11 показана данная гистограмма.

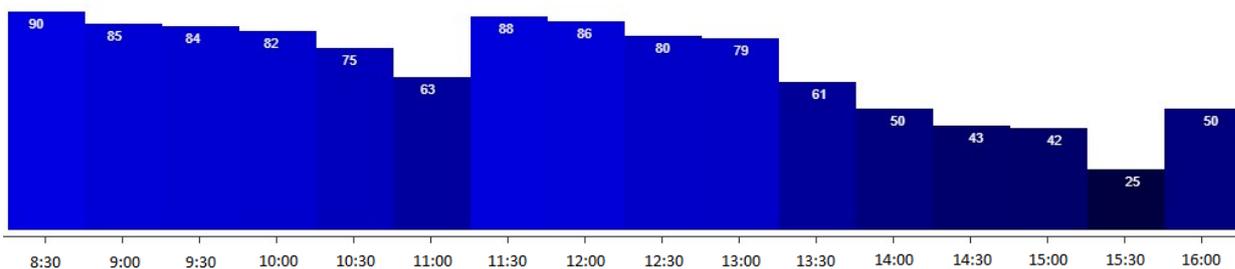


Рисунок 3.11 - Построенная гистограмма

Также, пользователь имеет возможность построить круговую диаграмму, согласно которой можно отследить статистику эмоционального состояния студентов после посещения каждого конкретного предмета. Для этого необходимо также воспользоваться инструментарием D3.js, для построения круговой диаграммы. На рисунке 3.12 приведён листинг алгоритма построения круговой диаграммы. Здесь функция buildPie позволяет заполнить элемент svg круговой диаграммой.

```
var buildPie=function() {
  var peidataset=[35,40,200,68],
  pie = d3.pie(),
  piew=300, pieh=300,
  outerRadius=piew/2, innerRadius=0;

  var svg=d3.select("div.pie").append("svg").attr("width",piew).attr("height",pieh),
  arc=d3.arc()
    .innerRadius(innerRadius).outerRadius(outerRadius);
  var color = d3.scaleOrdinal(['#4daf4a', '#377eb8', '#ff7f00', '#984ea3', '#e41a1c']);

  var arcs=svg.selectAll("g.arc")
    .data(pie(peidataset)).enter().append("g")
    .attr("class", "arc")
    .attr("transform", "translate("+outerRadius+", "+outerRadius+)");

  arcs.append("path")
    .attr("fill", function(d, i) {
      return color(i);
    })
    .attr("d", arc);

  arcs.append("text").attr("transform", function(d) {return "translate("+arc.centroid(d)+")";})
    .attr("text-anchor", "middle").text(function(d) {return d.value;});
}
```

Рисунок 4 - Листинг алгоритма построения круговой диаграммы

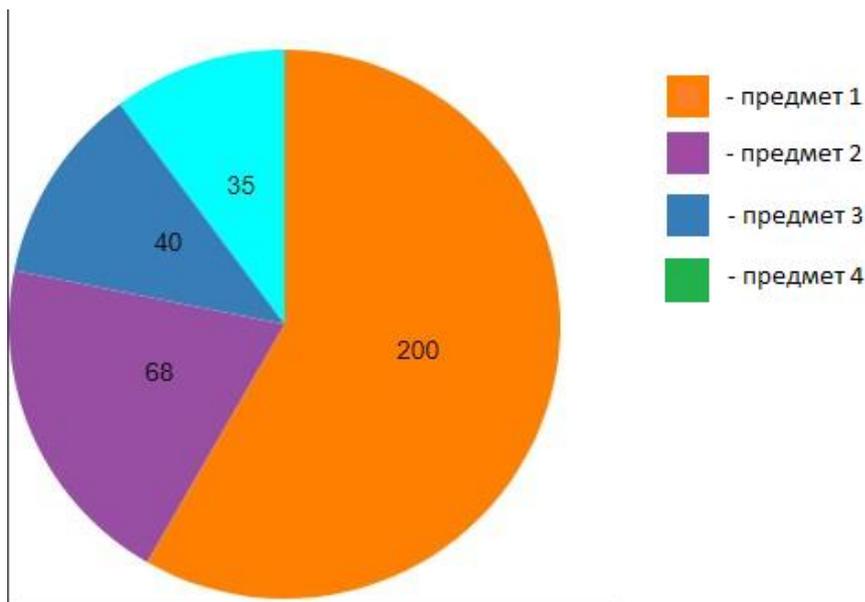


Рисунок 5 - Построенная круговая диаграмма, отображающая статистику заинтересованности предметами студентами

Таким образом, поставленное задание было выполнено, основные функции приложения были реализованы и протестированы.

3.5 Тестирование приложения

После разработки данного ВІ-приложения, предоставляющего пользователю возможность строить отчёты на основе Big Data, необходимо протестировать приложение с разными источниками, типами, набором и количеством входных данных. Ведь именно от этого будет зависеть в дальнейшем работоспособность и быстродействие разработанного веб-приложения.

Таким образом, для тестирования приложения были использованы как данные, полученные с сервера, так и данные, воссозданные вручную.

Количество студентов/групп	Время на отправку/получение запроса	Время построения отчёта
50/500	4ms	10ms
100/1000	6ms	11ms
5000/50000	10ms	21ms

Рисунок 3.13 - время, затраченное на построение отчёта

Таким образом по рисунку 3.13 можно увидеть, что основная нагрузка на клиентской части приходится именно на построение графической составляющей, а именно: гистограммы и круговой диаграммы.

По рисунку также видно, что время на отправку и получение запроса варьируется в зависимости от количества запрашиваемых элементов. Это связано с подключением к удалённому серверу по WebSocket-протоколу.

В случае с локально воссозданными данными, время на отправку и получение данных не рассматривается, так как задержек не наблюдалось.

Выводы по третьей главе

В третьей главе было разработано и протестировано ВІ приложение с веб-интерфейсом, позволяющее просматривать отчёт по вовлеченности студентов в процесс обучения. С помощью данного приложения были выведены, обработаны и представлены графически данные, полученные из базы.

Для визуализации полученных данных была изучена и в дальнейшем применена мощная библиотека, позволяющая строить различные графические элементы. Данная библиотека является основой для некоторых других подобных библиотек, что и объясняет её популярность.

В результате проведения исследования, была изучена документация как на русском, так и на английском языке по созданию веб-приложений, для приведения статистики, были использованы различные электронные ресурсы, в том числе и иностранные.

Задачи, поставленные на данную выпускную квалификационную работу, были выполнены.

ЗАКЛЮЧЕНИЕ

Целью данной выпускной квалификационной работы являлась разработка веб-приложения для анализа степени вовлеченности в учебный процесс на основе эмоционального состояния участников процесса обучения. В приложении реализованы функции вывода на экран статистики, как в табличном, так и в графическом формате.

В процессе выполнения данной работы был исследован материал из нескольких источников. Данный теоретический материал позволил расширить возможности в разработке ВІ-приложений, научиться построению графических способов представления больших объёмов данных.

Был проведён анализ различных средств разработки ВІ-приложений, построения веб-интерфейса и алгоритмов извлечения и обработки данных для дальнейшей визуализации. При создании приложения, были учтены уже существующие аналоги, рассмотренные во второй главе.

Разработанное приложение с веб-интерфейсом позволяет пользователям составлять отчёты за различные промежутки времени и отслеживать вовлеченность студентов в процесс обучения на основе их эмоционального состояния на протяжении дня. Важным достоинством данного приложения является удобство чтения информации для пользователя. Это было достигнуто благодаря использованию современных средств разработки и визуализации.

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

Научная и методическая литература

1. Берд, Дж. Веб-дизайн.Руководство разработчика. / Дж. Берд. - СПб.: Питер, 2012. - 224 с.
2. Вотролл, Э. Изучаем веб-дизайн / Э. Вотролл, Дж. Сьярто. - М.: Эксмо, 2010. - 496 с.
3. Голомбински, К. Добавь воздуха! Основы визуального дизайна для графики веб и мультимедиа / К. Голомбински, Р. Хаген; Пер. с англ. Н.А. Римицан.. - СПб.: Питер, 2013. - 272 с
4. Дунаев, Вадим Сценарии для Web-сайта. PHP и JavaScript / Вадим Дунаев. - М.: "БХВ-Петербург", 2012. – 576с.
5. Книга веб-программиста. Секреты профессиональной разработки веб-сайтов / Б. Хоган и др. - Москва: Мир, 2013. - 288 с.
6. Макнейл, П. Веб-дизайн. Книга идей веб-разработчика / П. Макнейл. - СПб.: Питер, 2017. - 480 с.
7. Маркотт, И. Отзывчивый веб-дизайн: № 1 / И. Маркотт. - М.: Манн, Иванов и Фербер, 2012. - 176 с.
8. Мэтью, Дэвид HTML5. Разработка веб-приложений / Дэвид Мэтью. - М.: Рид Групп, 2012. - 320 с.
9. Самков, Г. А. jQuery. Сборник рецептов / Г.А. Самков. - М.: БХВ-Петербург, 2011. - 416 с.
10. Силен Дэви, Мейсман Арно, Али Мохамед. Основы Data Science и Big Data. Python и наука о данных / СПб.: Питер,2019. – 336с.
11. Фельке-Моррис, Терри Большая книга веб-дизайна (+ CD-ROM) / Терри Фельке-Моррис. - М.: Эксмо, 2012. - 608 с.

Электронные ресурсы

12. Business Intelligence (рынок России), 2018 // Портал Tadviser [Электронный ресурс]. URL:

[http://www.tadviser.ru/index.php/статья:Business_Intelligence_\(рынок_России\)](http://www.tadviser.ru/index.php/статья:Business_Intelligence_(рынок_России))

(дата обращения 14.05.2019)

13. Business Intelligence Бизнес-аналитика, 2018 // Портал Tadviser [Электронный ресурс]. URL: <http://www.tadviser.ru/index.php/BI> (дата обращения 05.05.2019)

14. Data-Driven Documents, 2019 // Официальный сайт D3 [Электронный ресурс]. URL: <https://d3js.org/> (дата обращения 07.05.2019)

15. Oracle Business Intelligence 2018 // Tadviser [Электронный ресурс]. URL: http://www.tadviser.ru/index.php/Продукт:Oracle_Business_Intelligence (дата обращения 07.05.2019)

16. The Importance of Business Intelligence, 2018 // Портал BBN Times [Электронный ресурс]. URL: <https://www.bbntimes.com/en/companies/the-importance-of-business-intelligence> (дата обращения 08.05.2019)

17. Введение в D3.js, 2015 // Сайт metanit [Электронный ресурс]. URL: <https://metanit.com/web/d3js/1.1.php> (дата обращения 08.05.2019)

18. Основы Javascript, 2018 // HTML5book [Электронный ресурс]. URL: <https://html5book.ru/osnovy-javascript/> (дата обращения 09.05.2019)

19. Просто о D3.js, 2017 // Сайт habr [Электронный ресурс]. URL: <https://habr.com/ru/post/342106/> (дата обращения 13.05.2019)

20. Сравнение топ-4 популярных BI платформ. Какую выбрать?, 2018 // Сайт habr [Электронный ресурс]. URL: <https://habr.com/ru/company/newprolab/blog/349186/> (дата обращения 10.05.2019)

21. Что такое Business Intelligence, 2016 // Блог ViWeb [Электронный ресурс]. URL: <http://biweb.ru/chto-takoe-business-intelligence.html> (дата обращения 10.05.2019)

Литература на иностранном языке

22. Dustin Adkison – IBM Cognos Business Intelligence – Packt Publishing Ltd. 2013

23. Familiar B., Barnes J. - Business in Real-Time Using Azure IoT and Cortana Intelligence Suite Driving Your Digital Transformation – 2017

24. Karl Wieggers – Successful Business Analysis Consulting Strategies and Tips for Going It Alone – J. Ross Publishing – 2019

25. Yuli Vasiliev. Oracle Business Intelligence: The Condensed Guide to Analysis and Reporting / Packt Publishing Ltd. 2010

ПРИЛОЖЕНИЕ А

Листинг вывода гистограммы на экран:

```
var dataset=[],les=[],lessonsetForPie=[],  
  
i=0,sortOrder=false,svg,  
  
xScale,xAxis,svgAxis,w=1024,h=250,barPadding=1,built=false;  
  
d3.csv("students/students.csv").then(function(data){  
  
    //les=data;  
  
    data.forEach(function(d){  
  
        d.Emotion = +d.Emotion;  
  
        dataset[i]=d.Emotion;  
  
        timeset[i]=d.Time;  
  
        i++;  
  
    })  
  
    })  
  
    console.log(timeset);
```

```

var buildSvg=function(){
    if(!built){
        xScale=d3.scaleBand()
                                .domain(d3.range(dataset.length))
                                .rangeRound([0,w])
                                .padding(0.05);
        svg=d3.select("div.bars").append("svg");
        svg.attr("width",w).attr("height",h);
        svg.selectAll("rect").data(dataset)
            .enter().append("rect")
            .attr("x",function(d,i){
                //return i*(w/dataset.length);
                return xScale(i);
            })
            .attr("y",function(d){return h-d*2;})
            .attr("width",w/dataset.length-barPadding)
            // .attr("width",xScale.rangeRound())
            .attr("height",function(d){return d*4;})
            .attr("fill",function(d){return "rgb(0,0,"+(d*2.5)+")"});
        //var texts=svg.selectAll("text").enter();
    }
}

```

```
svg.selectAll("text.title")
```

```
    .data(dataset)
```

```
    .enter()
```

```
    .append("text")
```

```
    .text(function(d){
```

```
        return d;
```

```
    })
```

```
    .attr("x",function(d,i){
```

```
        return i*(w/dataset.length)+(w/dataset.length)/2-5;
```

```
    })
```

```
    .attr("y",function(d){
```

```
        return h-(d*2)+14;
```

```
    })
```

```
    .attr("fill","white")
```

```
    .attr("font-size","11px");
```

```
svg.selectAll("text.value")
```

```
    .data(timeset)
```

```
    .enter()
```

```
    .append("text")
```

```
    .text(function(d){
```

```
        return d;
```

```
    })
```

```
    .attr("x",function(d,i){
```

```

        return i*(w/dataset.length)+(w/dataset.length)/2-5;
    })
    .attr("y",function(d){
        return h-(10*2)+5;
    })
    .attr("fill","white")
    .attr("font-size","11px");

    console.log("Dataset: "+dataset);

    console.log("Timeset: "+timeset.length);

svgAxis=d3.select("div.bars").append("svg");

svgAxis.attr("width",w).attr("height",100).attr("margin-top","200px");

scaleX = d3.scaleLinear()

        .domain([d3.min(dataset),d3.max(dataset)])

        .range([0,w]);

xAxis = d3.axisBottom().scale(xScale);

svgAxis.append("g").attr("class","axis")

        .attr("transform","translate(0,0)")

        .call(xAxis);

```

```

        built=true;
    }
}

var sortBars=function(){

    sortOrder=!sortOrder;

    svg.selectAll("rect").sort(function(a,b){

        if(sortOrder){return d3.ascending(a,b);}else{return d3.descending(a,b);}

    }).transition().delay(function(d,i){return
i*50;}).duration(1000).attr("x",function(d,i){

        return xScale(i);

    })

    svg.selectAll("text.title").sort(function(a,b){

        if(sortOrder){return d3.ascending(a,b);}else{return d3.descending(a,b);}

    }).transition().delay(function(d,i){return
i*50;}).duration(1000).attr("x",function(d,i){

        return i*(w/dataset.length)+(w/dataset.length)/2-5;

    })

}

var buildPie=function(){

    var peidataset=[35,40,200,68],

```

```

pie = d3.pie(),
piew=300,pieh=300,
outerRadius=piew/2,innerRadius=0;

var svg=d3.select("div.pie").append("svg").attr("width",piew).attr("height",pieh),
arc=d3.arc()

    .innerRadius(innerRadius).outerRadius(outerRadius);
var color = d3.scaleOrdinal(['#4daf4a','#377eb8','#ff7f00','#984ea3','#e41a1c']);

var arcs=svg.selectAll("g.arc")

    .data(pie(peidataset)).enter().append("g")

    .attr("class","arc")

    .attr("transform","translate("+outerRadius+", "+outerRadius+)");

arcs.append("path")

    .attr("fill",function(d,i){

        return color(i);

    })

    .attr("d",arc);

arcs.append("text").attr("transform",function(d){return"translate("+arc.centroid(d
)+")";})

    .attr("text-anchor","middle").text(function(d){return d.value;});

}

```