

**Министерство науки и высшего образования Российской Федерации**  
**Федеральное государственное бюджетное образовательное учреждение**  
**высшего образования**  
**«Владимирский государственный университет имени Александра**  
**Григорьевича и Николая Григорьевича Столетовых»**  
**(ВлГУ)**  
**Колледж инновационных технологий и предпринимательства**

**КАФЕДРА ФИЗИКИ И ПРИКЛАДНОЙ МАТЕМАТИКИ**

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К КУРСОВОЙ РАБОТЕ**  
**ПО ДИСЦИПЛИНЕ «Технология разработки программного обеспечения»**  
**на тему «Разработка веб-приложения: «Агрегатор календарей»»**

Студента    Рушева Алексея Михайловича

Направление подготовки    СПО    09.02.03    "Программирование    в  
компьютерных системах"

Группы ПКсп-117

*Руководитель:* преподаватель КИТП \_\_\_\_\_ Лексин А.Ю.

# ВЛАДИМИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

УТВЕРЖДАЮ

Зав. кафедрой \_\_\_\_\_

« \_\_\_\_\_ »

\_\_\_\_\_ 20 \_\_\_\_ г.

## ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ

по дисциплине «Технология разработки программного обеспечения»

Выдано студенту Рушеву Алексею Михайловичу группы ПКсп-117  
Направление подготовки СПО 09.02.03 «Программирование в  
компьютерных системах»

1. Тема работы «Разработка веб-приложения "Агрегатор календарей"»
2. Срок сдачи законченной работы 08.04.2021
3. Исходные данные к курсовой работе постановка задачи.
4. Содержание расчетно-пояснительной записки (перечень подлежащих разработке вопросов)

### Введение.

- 1 Анализ предметной области.
- 2 Постановка задачи
- 3 Проектирование приложения.
- 4 Разработка приложения.
- 5 Тестирование.
- 6 Руководство пользователя.

### Заключение.

## 5. Постановка задачи

### ЧАСТЬ 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1. Выполнить описание предметной области.
2. На основании описания предметной области провести обзор существующих аналогов.

### ЧАСТЬ 2. ПОСТАНОВКА ЗАДАЧИ

1. Определить требования и задачи для достижения поставленной цели.
2. Определить минимальные требования для работы программы.

### ЧАСТЬ 3. ПРОЕКТИРОВАНИЕ ПРИЛОЖЕНИЯ

1. На основании описания предметной области выполнить разработку ER-модели, реляционной модели данных.
2. Составить словарь данных, в котором представлены все поля таблиц базы данных с указанием ключа, наименования, типа, обязательности заполнения и замечок.
3. Представить диаграмму прецедентов, диаграмму активностей, диаграмму развертывания.
4. Расписать основные методы, используемые в проектировании интерфейса.

#### ЧАСТЬ 4. РАЗРАБОТКА ПРИЛОЖЕНИЯ

1. Описать выбор инструментальных средств разработки приложения.
2. На основании ER-диаграммы базы данных, описать физическую реализацию базы данных. Если в работе с базой данных используются представления, то описать подробно каждое из них.
3. Описать все классы приложения с пояснением для каких целей они предназначены и методы с описанием: назначения, входных значений и выходных значений каждого метода.
4. Кратко описать структуру и компоненты, используемые при разработке интерфейса.

#### ЧАСТЬ 5. ТЕСТИРОВАНИЕ

1. Составить тест-требования, определяющие, что должно быть протестировано, но не определяющие, как это должно быть сделано.
2. По тест-требованиям разработать тест-план, в каждом тестовом примере которого обязательно перечислены все входные значения и ожидаемые выходные значения, а также сценарий, описывающий последовательность действий, которые необходимо выполнить для выполнения тестового примера.
3. Указать результаты тестирования некоторого функционала и общую оценку от проведенного тестирования.

#### ЧАСТЬ 6. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

1. Описать действия пользователя в приложении, с момента установки его и до получения конечного результата, рассмотрев весь возможный функционал приложения.

Дата выдачи задания 03.02.2021

Руководитель \_\_\_\_\_ Лексин А.Ю.

Задание принял к исполнению \_\_\_\_\_ Рушев А.М.

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	6
1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ .....	7
1.1 Описание предметной области .....	7
1.2 Обзор существующих решений задачи .....	8
1.2.1 Google Календарь.....	8
1.2.2 Яндекс.Календарь. ....	9
2 ПОСТАНОВКА ЗАДАЧИ.....	11
2.1 Определение требований и задач .....	11
2.2 Минимальные требования для работы программы.....	11
3 ПРОЕКТИРОВАНИЕ ПРИЛОЖЕНИЯ .....	13
3.1 ER-диаграмма базы данных .....	13
3.2 Словарь данных.....	14
3.3 Диаграмма прецедентов .....	16
3.4 Диаграмма активностей.....	17
3.5 Диаграмма развертывания .....	18
3.6 Проектирование интерфейса .....	19
4 РАЗРАБОТКА ПРИЛОЖЕНИЯ .....	21
4.1 Инструментальные средства разработки приложения.....	21
4.2 Физическая реализация БД .....	24
4.2 Описание классов.....	24
4.3 Разработка интерфейса .....	27
5 ТЕСТИРОВАНИЕ .....	29
5.1 Тест-требования .....	29
5.2 Тест-план.....	30

5.3 Результаты тестирования .....	33
6 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ .....	39
ЗАКЛЮЧЕНИЕ .....	42
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ.....	43

## ВВЕДЕНИЕ

В настоящее время невозможно представить жизнь без компьютерных технологий. Каждый из нас пользуется календарем, для просмотра дат, заметок или создавая напоминание о событиях.

Сейчас существует проблема в том, что для каждого онлайн календаря используется определенный сервис, в котором этот календарь используется, но может возникнуть необходимость увидеть событие из другого календаря. Поэтому пользователю приходится заходить в другой сервис и это занимает много времени.

Следовательно, необходимо повысить автоматизацию, которая позволит увеличить производительность и качество работы пользователя в использовании календаря.

На основе этой проблемы целью курсовой работы стало: «создание веб-приложения «Агрегатор календарей».

В связи с поставленной целью в работе решаются следующие задачи:

- Анализ предметной области;
- Проектирование приложения;
- Разработка приложения;
- Тестирование приложения;
- Составление руководства пользователя.

# 1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

## 1.1 Описание предметной области

Целью разрабатываемого проекта является создание веб-приложения: «Агрегатор календарей», для отображения событий из календарей с разных сервисов (google, outlook и т.д.).

Онлайн календарь – это сервис для лёгкой организации дел, событий, их проверки, а также добавления новых мероприятий.

Сервис обладает данным функционалом:

- Добавление события в календарь;
- Редактирование события в календаре;
- Удаление события в календаре;
- Переключение между месяцами;
- Переключение на текущий день.

Пример календаря представлен ниже (Рисунок 1).

Август 2010						
Понедельник	Вторник	Среда	Четверг	Пятница	Суббота	Воскресенье
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

Рисунок 1 – Календарь

## 1.2 Обзор существующих решений задачи

На сегодняшний день существует большое количество календарей, с помощью которых можно просматривать напоминания, корректировать их и добавлять новые, а также удалять.

Примеры таких календарей представлены ниже:

### 1.2.1 Google Календарь

Данный календарь разработан компанией «Google» и он охватывает весь необходимый функционал для календарей.

Данный проект имеет следующий функционал:

- Создание нового напоминания для даты, а также в разное время;
- Просмотр календаря (целый месяц и по неделям);
- Выбор нужного месяца и года;
- Варианты просмотров (дни рождения, задачи, напоминания, праздники);
- Настройка напоминания;
- Поиск по ключевым словам в календаре;
- Настройки календаря.

Интерфейс Google календаря представлен ниже (Рисунок 1).



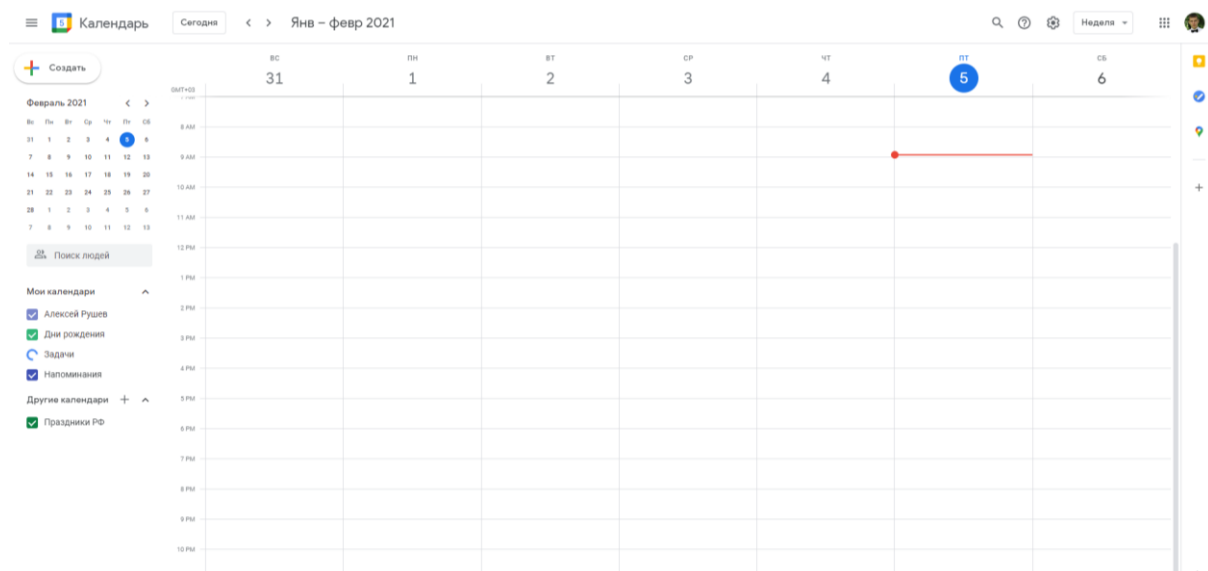


Рисунок 1 – Google календарь

### 1.2.2 Яндекс.Календарь

Данный проект имеет следующий функционал:

- Создание нового напоминания для даты, а также в разное время;
- Просмотр календаря (целый месяц, по неделям, дням);
- Выбор нужного месяца и года;
- Варианты просмотров (дни рождения, задачи, напоминания, праздники);
- Поиск людей (для показа календаря);
- Настройка напоминания;
- Настройки календаря;
- При создании напоминания есть выбор куда отправится уведомление (почта, смс, CalDAV).

Интерфейс Яндекс календаря представлен ниже (Рисунок 2).

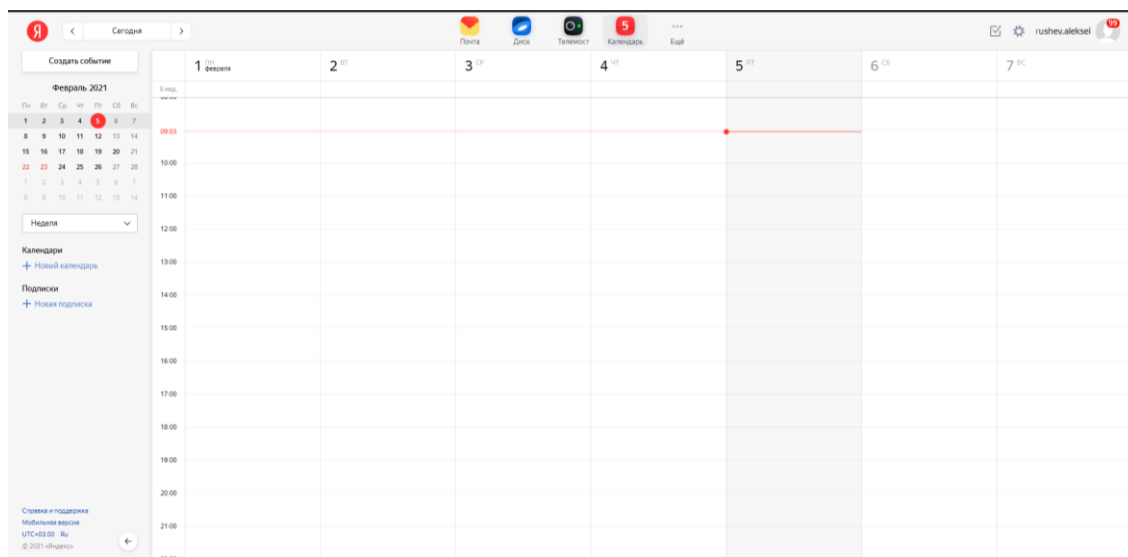


Рисунок 2 - Яндекс.Календарь

Первый календарь удовлетворяет многим функциональным возможностям. Второй же календарь аналогичен первому, но у него есть возможность уведомлять по смс о напоминании в календаре.

Рассмотренные аналоги обладают многим нужным требованиям для работы с календарем обычному пользователю. Оба приведенные в качестве примера. Аналоги, являются бесплатными, для них лишь нужно иметь почту того сервиса, чей календарь используешь. Исходя из вышеперечисленного, можно сделать вывод, что требуется разработка проекта, удовлетворяющему большинству требованиям обоих аналогов.

## 2 ПОСТАНОВКА ЗАДАЧИ

### 2.1 Определение требований и задач

Из описания предметной области, можно прийти к тому, что разрабатываемое приложение должно формировать сетку календаря, навигацию по нему и добавление события, удаления, а также изменения события.

Веб-приложение должно выполнять следующие задачи:

- Формирование сетки календаря;
- Переключение между месяцами;
- Переключение на текущий день в календаре;
- Добавление события в календарь;
- Удаление события из календаря;
- Изменение события в календаре.

### 2.2 Минимальные требования для работы программы

Для работы программы необходимо предоставить место на устройстве:

- 54МБ потребуется для готовой БД;
- 1ГБ потребуется для веб-приложения;
- ОС: Windows 7 и выше;
- Любой веб-браузер, поддерживающий HTML5.

ПО разрабатывается при помощи основных технологий:

- Microsoft SQL Server Management Studio 18;

- Microsoft Visual Studio Code;
- Node.js.

### 3 ПРОЕКТИРОВАНИЕ ПРИЛОЖЕНИЯ

#### 3.1 ER-диаграмма базы данных

Для разрабатываемого приложения была разработана ER-модель схемы данных, которая включает в себя 4 сущности предметной области.

Данная ER-диаграмма представлена ниже (Рисунок 3).

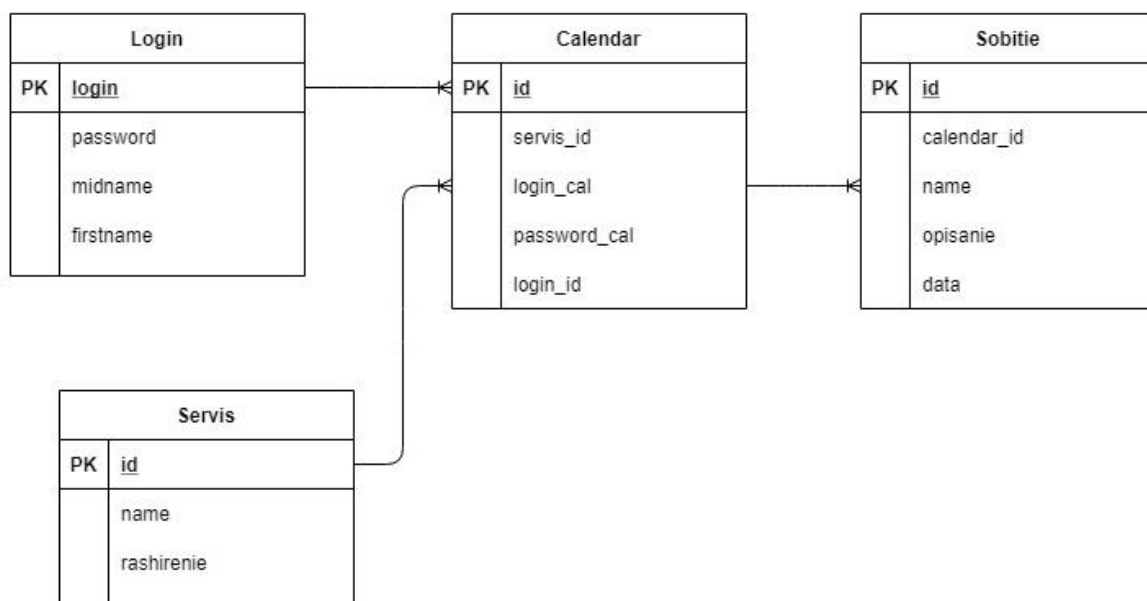


Рисунок 3 – ER-диаграмма приложения

### 3.2 Словарь данных

Словарь данных представлен в таблице 3.1. Данный словарь отражает сущности и поля из ER-диаграммы, в нём представлены все поля таблиц с указанием типов, обязательное заполнение и замечок.

Таблица 3.1 – Словарь данных

Словарь данных			
Login			
Ключ	Поле	Обязательное	Примечание
Первичный	login	Да	Логин пользователя в приложении
	password	Да	Пароль пользователя в приложении
	midname	Нет	Фамилия пользователя
	firstname	Нет	Имя пользователя
Servis			
Ключ	Поле	Обязательное	Примечание
Первичный	id	Да	Уникальный идентификатор таблицы
	name	Да	Наименование сервиса
	rashirenie	Да	Расширение сервиса (@google.com)

Продолжение таблицы 3.1.

<b>Sobitie</b>			
<b>Ключ</b>	<b>Поле</b>	<b>Обязательное</b>	<b>Примечание</b>
Первичный	id	Да	Уникальный идентификатор таблицы
Внешний	calendar_id	Да	Внешний ключ к таблице Calendar
	name	Да	Наименование события
	opisanie	Нет	Описание события
	data	Да	Дата события
<b>Calendar</b>			
<b>Ключ</b>	<b>Поле</b>	<b>Обязательное</b>	<b>Примечание</b>
Первичный	id	Да	Уникальный идентификатор таблицы
Внешний	servis_id	Да	Внешний ключ к таблице Servis
	login_cal	Да	Логин пользователя в календаре
	password_cal	Да	Пароль пользователя в календаре
Внешний	login_id	Да	Внешний ключ к таблице Login

### 3.3 Диаграмма прецедентов

Диаграмма прецедентов – диаграмма, на которой отражены отношения, существующие между актерами и прецедентами и являющиеся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне.

На рисунке 4. представлена диаграмма, на которой показаны все возможные функции, который реализован в приложении для пользователя. На диаграмме представлено описание, того, что может делать пользователь.

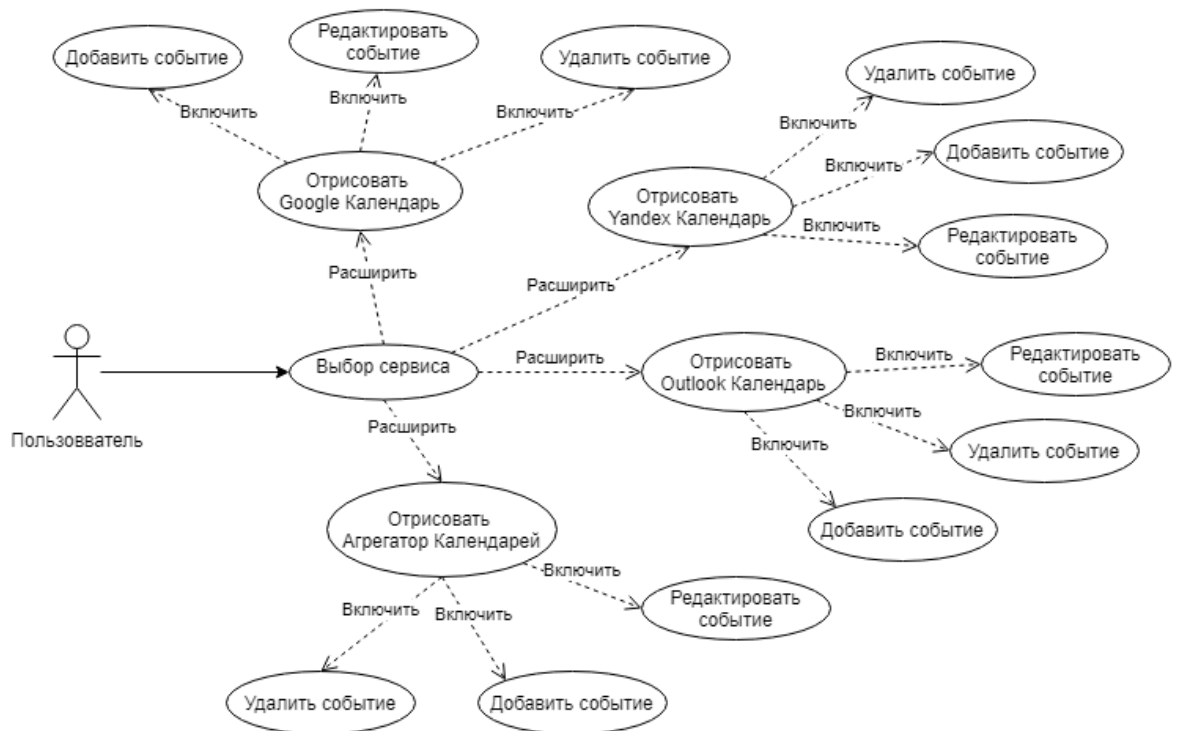


Рисунок 4 – Диаграмма прецедентов



### 3.4 Диаграмма активностей

Диаграмма активности – UML-диаграмма, на которой показаны действия, состояния которых описано на диаграмме состояний. На рисунке 5 изображена диаграмма активностей.

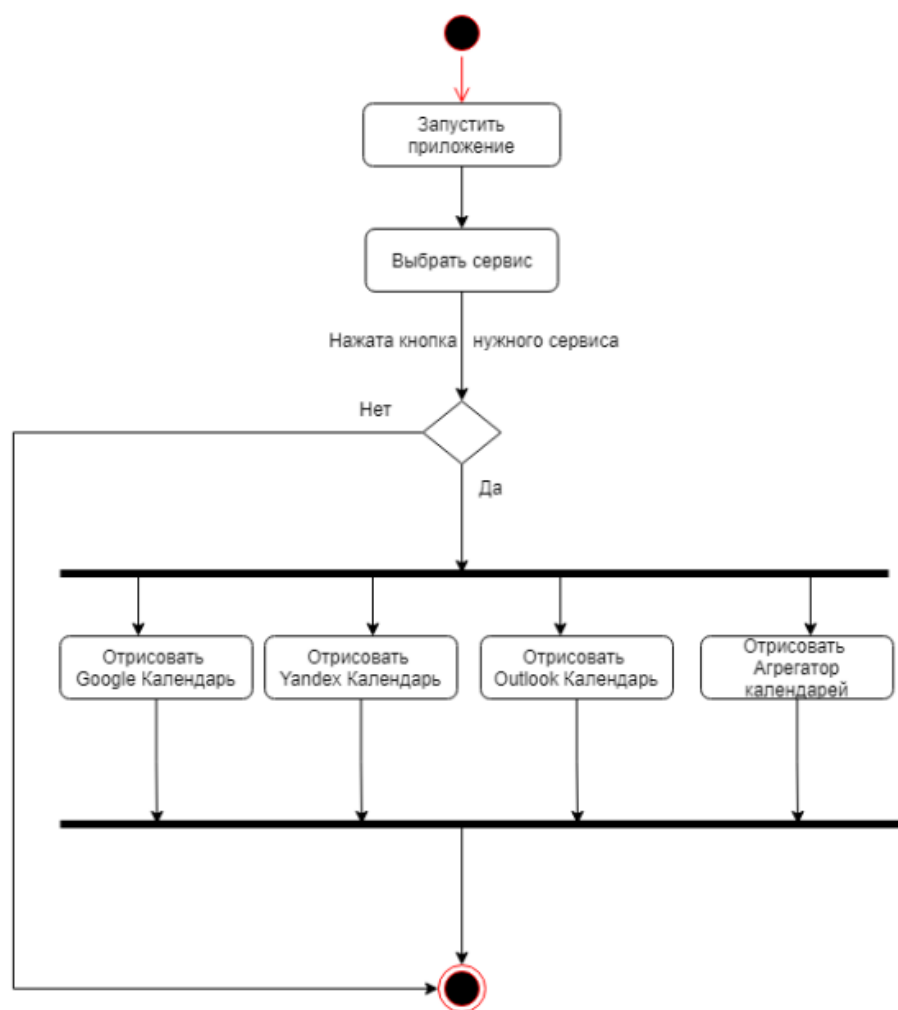


Рисунок 5 – Диаграмма активностей

### 3.5 Диаграмма развертывания

Диаграмма развертывания – диаграмма, на которой представлены узлы выполнения программных компонентов реального времени, а также процессов и объектов. На рисунке 6 представлена диаграмма, на которой показаны связи модулей приложения с базой данных.

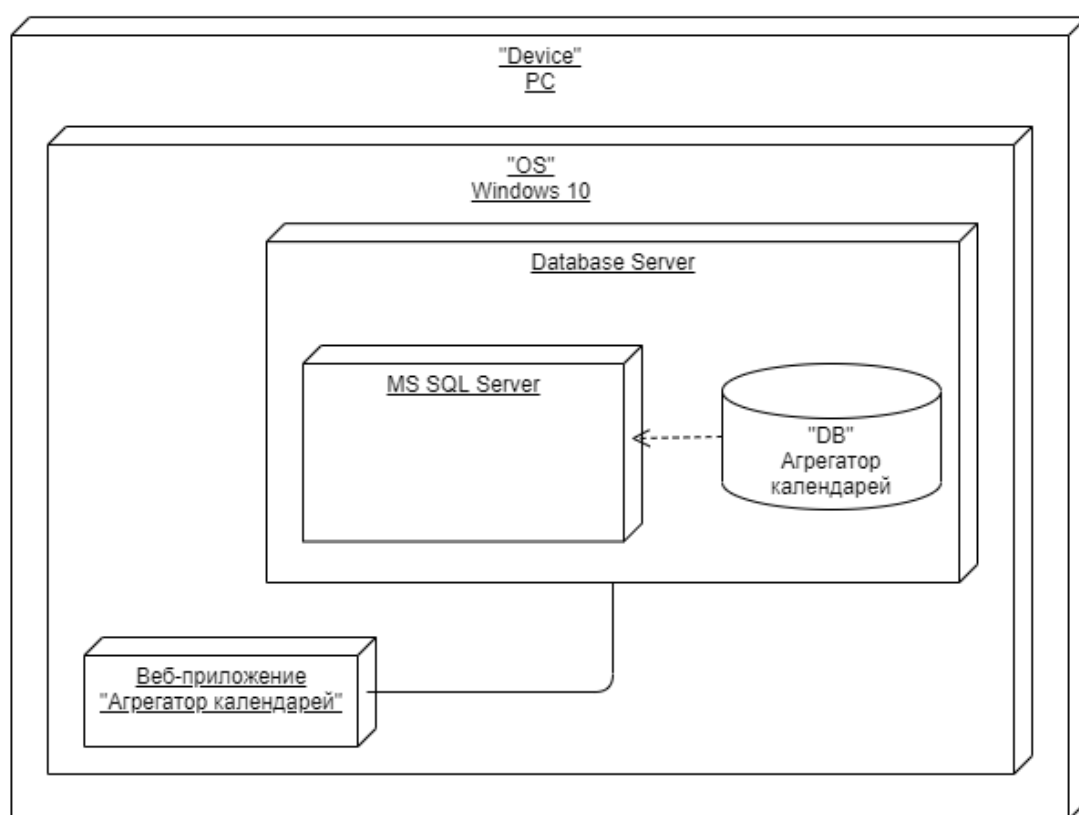


Рисунок 6 – Диаграмма развертывания

### 3.6 Проектирование интерфейса

В настоящее время много онлайн календарей и все они похожи между собой, но у них настолько большой функционал, что иногда не сразу приходит понимание интерфейса у пользователя. Поэтому необходимо разработать интерфейс, который будет максимально прост для понимания пользователя. Чтобы пользователю не потребовалось объяснять, как им пользоваться.

На рисунке 7 представлен макет страницы сайта, разрабатываемого приложения, которая динамически обновляется, что позволяет пользоваться новыми технологиями.

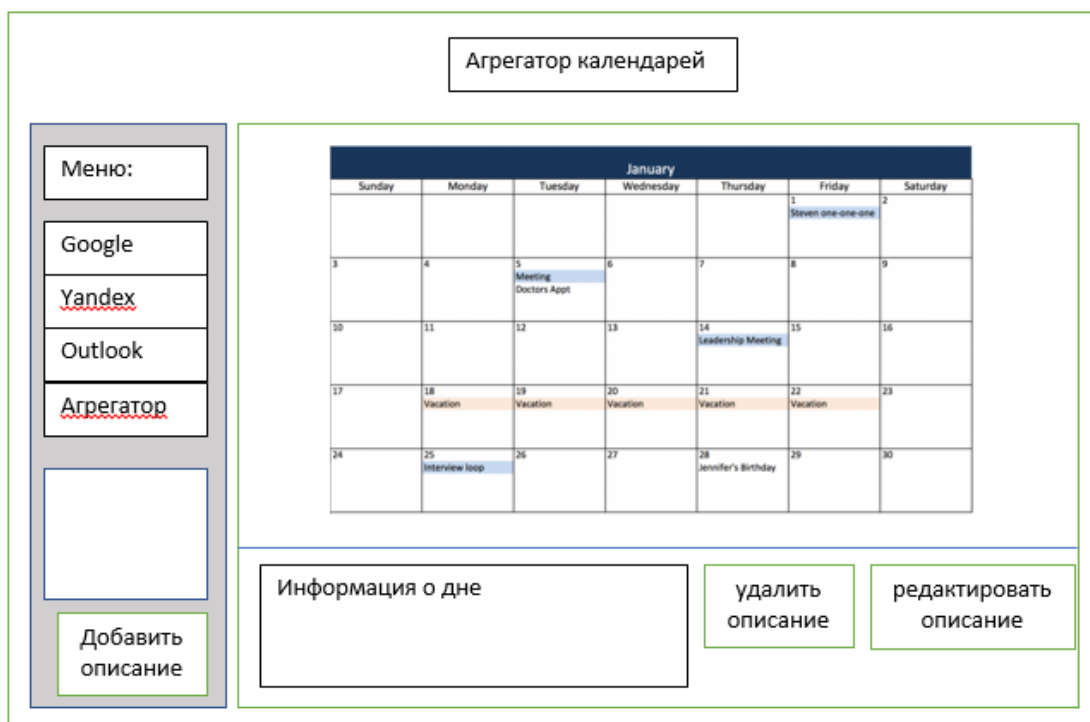


Рисунок 7 – Макет приложения

На рисунке 8 представлена карта сайта.



Рисунок 8 – Карта сайта

## 4 РАЗРАБОТКА ПРИЛОЖЕНИЯ

### 4.1 Инструментальные средства разработки приложения

Для разработки веб-приложения «Агрегатор календарей», было задействованы нижеперечисленные средства разработки.

В качестве реализации БД была выбрана система Microsoft SQL Server Management Studio 18.

Microsoft SQL Server 2017 Express – это надежная система управления данными, обеспечивающая множество функций, защиту данных и высокую производительность для внедренных приложений-клиентов, «легких» вебприложений и локальных хранилищ данных. Microsoft SQL Server 2017 с пакетом обновления 1 Express — это бесплатная версия SQL Server с широкими функциональными возможностями, идеально подходящая для обучения, разработки и обеспечения работы классических, небольших серверных приложений и веб-приложений, а также для распространения независимыми поставщиками программного обеспечения.

SQL Server 2017 упрощает развёртывание, управление и оптимизацию данных предприятия и аналитических приложений. Как платформа управления данными, он предоставляет единую консоль управления, которая позволяет администраторам данных, находящимся в любом месте, отслеживать, управлять и настраивать все базы данных и связанные службы по всему предприятию. Он предоставляет расширяемую инфраструктуру управления, которая может быть легко запрограммирована при помощи SQL Management Objects (SMO), позволяя пользователям переделывать и 28 расширять их среду управления и независимым поставщикам программных продуктов (ISV) создавать дополнительные инструменты и функциональность

для дальнейшего расширения возможностей, поставляемых по умолчанию. Данная система не содержит и не является средой разработки клиентских приложений.

В качестве среды разработки была выбрана Microsoft Visual Studio Code. Среда разработки Visual Studio Code – это редактор исходного кода, он позиционируется как «лёгкий» редактор кода для кроссплатформенной разработки веб- и облачных приложений. Включает в себя отладчик, инструменты для работы с Git, подсветку синтаксиса, IntelliSense и средства для рефакторинга.

При помощи Visual Studio Code вы можете создавать:

- приложения и игры, которые выполняются не только на платформе Windows, но и на Android и IOS;
- веб-сайты и веб-службы на основе ASP.NET, JQuery, AngularJS и других популярных платформ;
- приложения для самых разных платформ и устройств, включая, но не ограничиваясь: Office, Sharepoint, Hololens, Kinect и «Интернета вещей»;
- игры и графические приложения для разных устройств Windows, включая Xbox, с поддержкой DirectX.

Также основной средой разработки является: Node или Node.js — это среда выполнения JavaScript. Окружение Node.js включает все, что вам нужно для выполнения программы, написанной на JavaScript. Окружение Node.js включает все, что вам нужно для выполнения программы, написанной на JavaScript.

Данное средство разработки следует использовать по следующим причинам:

- Node.js — среда выполнения JavaScript, основанная на JavaScript движке V8 из Chrome;
- Node.js использует управляемую событиями, неблокирующую модель ввода-вывода, которая делает ее легкой и эффективной;

– Пакетная экосистема Node.js, npm, является самой большой экосистемой библиотек с открытым исходным кодом в мире.

Еще одна необходимая технология при разработке приложения – это Redux. Она является библиотекой для JavaScript с открытым исходным кодом, предназначенная для управления состоянием приложения. Чаще всего используется в связке с React или Angular для разработки клиентской части. Содержит ряд инструментов, позволяющих значительно упростить передачу данных хранилища через контекст.

Самым главным средством разработки приложения является JavaScript — язык программирования. Поддерживает объектно-ориентированный, императивный и функциональный стили. Является реализацией спецификации ECMAScript. JavaScript обычно используется как встраиваемый язык для программного доступа к объектам приложений. Наиболее широкое применение находит в браузерах как язык сценариев для придания интерактивности веб-страницам. Основные архитектурные черты: динамическая типизация, слабая типизация, автоматическое управление памятью, прототипное программирование, функции как объекты первого класса.

Вторым по значимости средством разработки приложения является React (иногда React.js или ReactJS) — это JavaScript-библиотека с открытым исходным кодом для разработки пользовательских интерфейсов. React разрабатывается и поддерживается Facebook, Instagram и сообществом отдельных разработчиков и корпораций. React может использоваться для разработки одностраничных и мобильных приложений. Его цель: предоставить высокую скорость, простоту и масштабируемость. В качестве библиотеки для разработки пользовательских интерфейсов React часто используется с другими библиотеками, такими как MobX, Redux и GraphQL.

Для создания календаря необходимо такое средство разработки, как: Moment.js – это библиотека JS, которая проста в использовании. Она содержит

разнообразные функции для операций с датами. Moment.js была создана для обработки даты и времени, и их различных сочетаний.

## 4.2 Физическая реализация БД

Исходя из ER-диаграммы базы данных, была составлена физическая реализация базы данных в Microsoft SQL Server, она представлена на рисунке 9. Скрипт создания базы данных представлен в Приложении А.

\*Скрин БД\*

## 4.2 Описание классов

Разрабатываемое приложение предназначено для создания календаря, возможности навигации в нем и формировании событий в нем. Исходя из этого, формы приложения должны иметь следующие возможности:

- Отрисовка главной страницы;
- Добавление события в календарь;
- Удаление события из календаря;
- Редактирование события из календаря;
- Переключение на следующий месяц в календаре;
- Переключение на предыдущий месяц в календаре;
- Отображение текущего дня.

Рассмотрим методы, которые используются в компонентах: App.js, redrow.jsx, index.js, basis.jsx, Main.jsx, MonitorCalendar.jsx.



### **let redrawPage (basis)**

Назначение: перерисовка страницы.

Входные значения: слушатель компонента.

Выходные значения: отрисовка страницы.

Содержание: для начала происходит возвращение в DOM-элемент и отслеживание последних изменений на странице, для того чтобы были произведены последние изменения, затем идет роутинг сайта, чтобы было отслеживание по URL адресу, после идет отрисовка самой страницы.

### **function App (props)**

Назначение: отрисовка страницы.

Входные значения: параметры, передающиеся из redraw.jsx

Выходные значения: отрисовка страницы.

Содержание: для начала идет отрисовка «Шапки» сайта, после отрисовка «Меню», где передается весь роутинг сайта, затем отрисовка «Информации о календаре».

### **redrawPage (basis)**

Назначение: отрисовка страницы.

Входные значения: методы basis.jsx.

Выходные значения: нет.

Содержание: при запуске приложения компонента просматривает компоненту redraw.jsx.

### **let addText ()**

Назначение: добавление события.

Входные значения: нет.

Выходные значения: нет.

Содержание: происходит заполнение новой переменной, в которую записывается событие и после передается в redrawPage, для перерисовки страницы.

### **let removeText (textSobitie)**

Назначение: удаление события.

Входные значения: текущее событие.

Выходные значения: нет.

Содержание: происходит удаление содержимого переменной, в которую было записано событие и после передается в `redrowPage`, для перерисовки страницы.

#### **let editText (textSobitie)**

Назначение: изменение события.

Входные значения: текущее событие.

Выходные значения: нет.

Содержание: происходит изменение переменной, в которой записано событие и после передается в `redrowPage`, для перерисовки страницы.

#### **const Main ({ startDay,today, nextMath, todayMath, previeMath })**

Назначение: отрисовка главного блока страницы.

Входные значения: начало отсчета, следующий месяц, текущий день, предыдущий месяц.

Выходные значения: навигация в календаре.

Содержание: происходит отрисовка шапки календаря, навигации по нему и отрисовка самого календаря.

#### **const nextMath ()**

Назначение: переключение на следующий месяц.

Входные значения: нет.

Выходные значения: нет.

Содержание: происходит переключение на следующий месяц в календаре.

#### **const todayMath ()**

Назначение: переключение на текущий день.

Входные значения: нет.

Выходные значения: нет.

Содержание: происходит переключение на текущий день в календаре.

### **const previeMath ()**

Назначение: переключение на предыдущий месяц.

Входные значения: нет.

Выходные значения: нет.

Содержание: происходит переключение на предыдущий месяц в календаре.

## **4.3 Разработка интерфейса**

Комплекс средств, который соединяет две системы друг с другом, называется интерфейсом. В качестве таких систем может выступать что угодно, от людей до искусственного интеллекта. Интерфейс приложения необходим пользователям для удобного использования приложения, не требуя специальных навыков.

Статус хорошего приложения относится к тому приложению, у которого удобный и простой интерфейс, так как пользователь должен на интуитивном уровне понимать, что сделает программа, после выполнения пользователем определенной команд.

Интерфейс состоит из: одной страницы, на которой отображено меню, где по нажатию на определенный интерфейс отобразится календарь в центральной части экрана.

На рисунке 9 изображена страница сайта, где расположены разные формы: слева расположено меню, вверху расположена шапка сайта, в центральной части сам календарь и информация о нем.

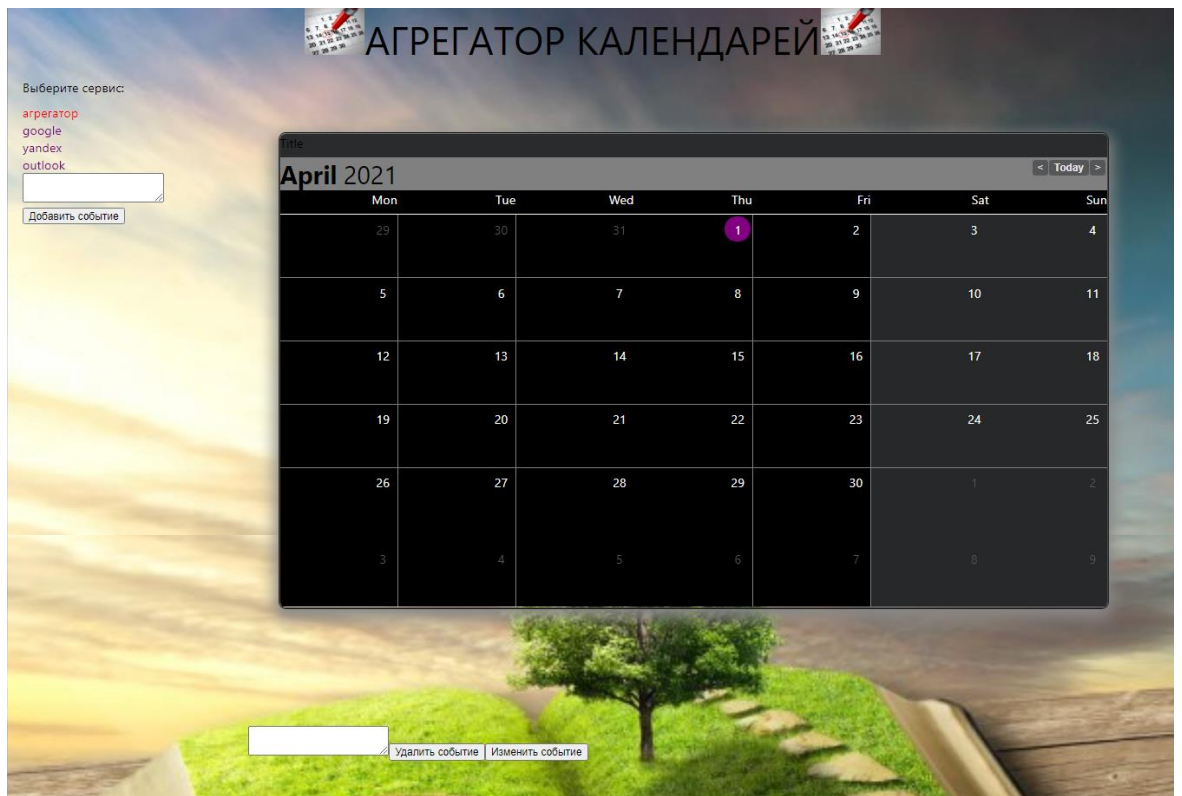


Рисунок 9 Главная страница сайта

## 5 ТЕСТИРОВАНИЕ

### 5.1 Тест-требования

1. Проверка поведения программы при удалении события.

Необходимо проверить, что будет, если удалить событие в выбранном дне.

2. Проверка поведения программы при редактировании информации о событии.

Необходимо проверить, что будет, если редактировать событие в выбранном дне.

3. Проверка поведения программы при добавлении события.

Необходимо проверить, что будет, если добавить событие в выбранный день.

4. Проверка поведения программы при выборе сервиса Google Календаря.

Необходимо проверить, что будет, если нажать в меню «google».

5. Проверка поведения программы при выборе сервиса Yandex Календаря.

Необходимо проверить, что будет, если нажать в меню «yandex».

6. Проверка поведения программы при выборе сервиса Outlook Календаря.

Необходимо проверить, что будет, если нажать в меню «outlook».

7. Проверка поведения программы при выборе сервиса Агрегатор календарей.

Необходимо проверить, что будет, если нажать в меню «агрегатор»

8. Проверка навигации календаря по текущему дню.

Необходимо проверить, что будет, если нажать на календаре «Today».

9. Проверка навигации календаря по месяцам.

Необходимо проверить, что будет, если нажать на календаре кнопку «<» или «>».

На основе тест-требований был составлен тест-план, состоящий из 5 тестов. Тест-план представлен ниже.

## 5.2 Тест-план

Тестовый пример 1

Номер тест-требования 1

Описание теста: в данном тесте проверяется переключение между месяцами в календаре.

Входные данные:

1. Календарь

Выходные данные:

1. В случае удачного выполнения произойдет переключение на предыдущий или следующий месяц в календаре.

2. В случае неудачного выполнения месяц в календаре останется прежним.

Сценарий теста:

1. Запустить программу;
2. Выбрать сервис;
3. Нажать на кнопку «<»;
4. Нажать на кнопку «>»;
5. Увидеть отображение другого месяца.

## Тестовый пример 2

### Номер тест-требования 2

Описание теста: в данном тесте проверяется переключение с любого месяца или дня к текущему дню.

Входные данные:

1. Календарь

Выходные данные:

1. В случае удачного выполнения переключится на текущий день.
2. В случае неудачного выполнения календарь не отобразит текущий день.

Сценарий теста:

1. Запустить программу;
2. Выбрать сервис;
3. Нажать кнопку «<» или «>»;
4. Нажать кнопку «Today»;
5. Увидеть текущий день.

## Тестовый пример 3

### Номер тест-требования 3

Входные данные:

1. Календарь

Выходные данные:

1. В случае удачного выполнения в календарь добавится событие.
2. В случае неудачного выполнения в календарь не добавится событие.

Сценарий теста:

1. Запустить программу;
2. Выбрать сервис;
3. Выбрать день в календаре;
4. Написать в пустое окно текст;

5. Нажать кнопку «Добавить событие»;
6. Увидеть под календарем текст события в выбранном дне.

#### Тестовый пример 4

##### Номер тест-требования 4

##### Входные данные:

1. Календарь

##### Выходные данные:

2. В случае удачного выполнения в календарь удалится событие.
3. В случае неудачного выполнения в календарь не удалится событие.

##### Сценарий теста:

1. Запустить программу;
2. Выбрать сервис;
3. Выбрать день в календаре;
4. Увидеть под календарем текст события;
5. Нажать кнопку «Удалить событие»;
6. Увидеть под календарем пустое окно для текста в выбранном дне.

#### Тестовый пример 5

##### Номер тест-требования 5

##### Входные данные:

1. Календарь

##### Выходные данные:

1. В случае удачного выполнения в календарь изменится событие.
2. В случае неудачного выполнения в календарь не изменится событие.

##### Сценарий теста:

1. Запустить программу;
2. Выбрать сервис;
3. Выбрать день в календаре;



4. Увидеть под календарем текст события;
5. Нажать кнопку «Изменить событие»;
6. Увидеть под календарем измененный текст события в выбранном дне.

### 5.3 Результаты тестирования

Тестовый пример 1.

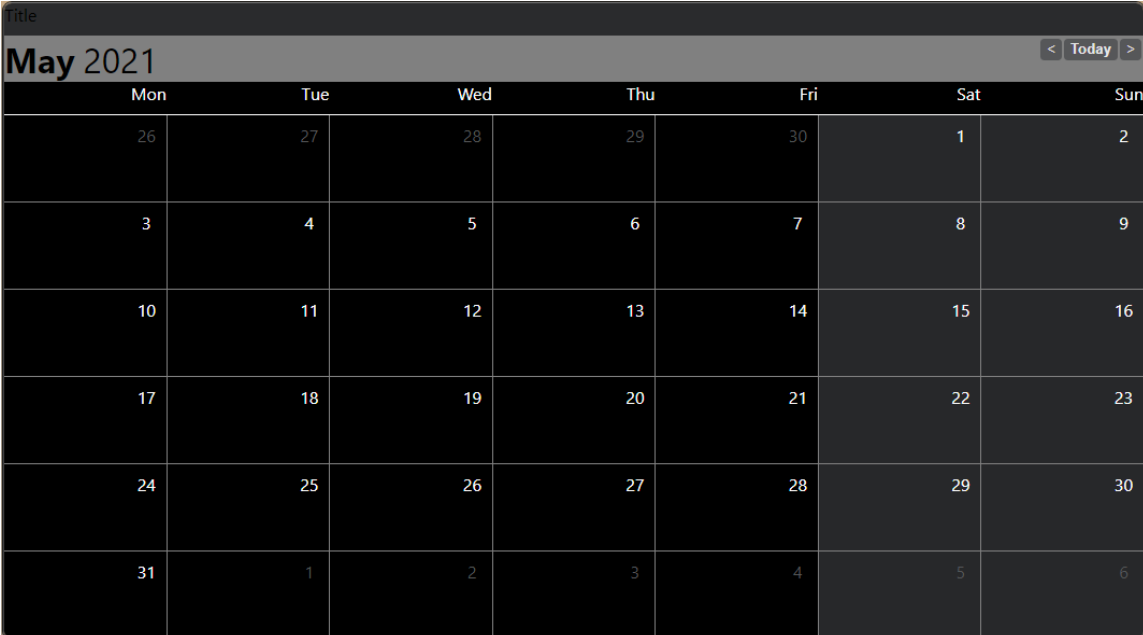
Специалист по тестированию: Рушев Алексей Михайлович

Дата и время тестирования: 01.04.2021, 23:15.

Входные данные:

1. Выбрать сервис;
2. Нажать кнопку «<» в календаре;
3. Нажать кнопку «>» в календаре.

Полученные выходные данные:



Title						
May 2021						
Mon	Tue	Wed	Thu	Fri	Sat	Sun
26	27	28	29	30	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

Рисунок 10 – Результат нажатия на кнопку «>»

Title						
March 2021						
< Today >						
Mon	Tue	Wed	Thu	Fri	Sat	Sun
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4
5	6	7	8	9	10	11

Рисунок 11 – Результат нажатия на кнопку «<>»

Вывод: соответствует ожидаемым выходным данным.

## Тестовый пример 2

Специалист по тестированию: Рушев Алексей Михайлович

Дата и время тестирования: 01.04.2021, 23:22.

Входные данные:

1. Выбрать сервис;
2. Нажать кнопку «<>» или «>>» в календаре;
3. Нажать кнопку «Today» в календарею.

Полученные выходные данные:

Title						
October 2020						
< Today >						
Mon	Tue	Wed	Thu	Fri	Sat	Sun
28	29	30	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	1
2	3	4	5	6	7	8

Рисунок 12 – Выбор любого месяца

Title						
April 2021						
< Today >						
Mon	Tue	Wed	Thu	Fri	Sat	Sun
29	30	31	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	1	2
3	4	5	6	7	8	9

Рисунок 13 – Результат нажатия на кнопку «Today»

Тестовый пример 3.

Специалист по тестированию: Рушев Алексей Михайлович

Дата и время тестирования: 01.04.2021, 23:31.

Входные данные:

1. Выбор сервиса;

2. Выбор нужного дня;
3. Ввод в пустое окно в блоке меню текст события (слева от календаря);
4. Нажатие на кнопку «Добавить событие»;
5. Увидеть текст события в нужном дне (внизу календаря).

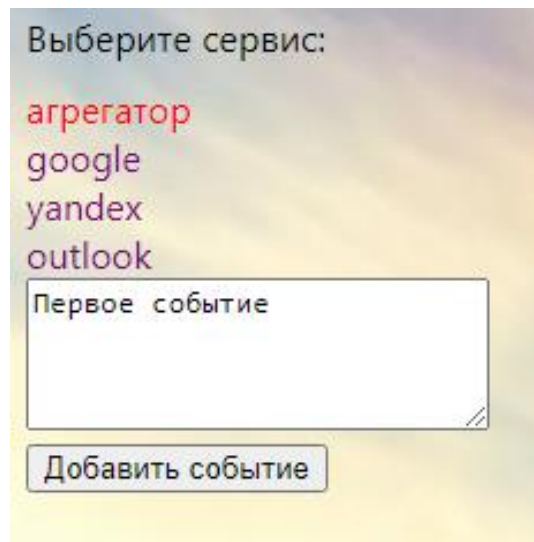


Рисунок 14 – Добавление события

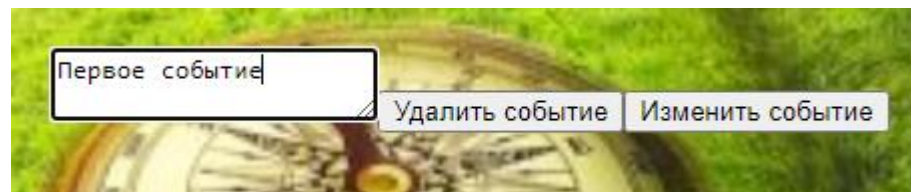


Рисунок 15 – Отображение события

Вывод: соответствует ожидаемым выходным данным.

Тестовый пример 4.

Специалист по тестированию: Рушев Алексей Михайлович

Дата и время тестирования: 01.04.2021, 23:40.

Входные данные:

1. Выбор сервиса;
2. Выбор нужного дня;

3. Просмотр сообщения события (под календарем);
4. Нажатие на кнопку «Удалить событие»;
5. Ожидание очистки окна вывода информации.

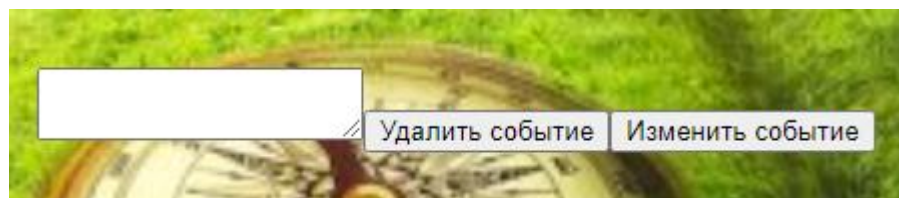


Рисунок 16 – Результат нажатия на кнопку «Удалить событие»

Вывод: соответствует ожидаемым выходным данным.

Тестовый пример 5.

Специалист по тестированию: Рушев Алексей Михайлович

Дата и время тестирования: 01.04.2021, 23:44.

Входные данные:

1. Выбор сервиса;
2. Выбор нужного дня в календаре с нужным событием;
3. Просмотр события (под календарем);
4. Изменение текста события;
5. Нажатие на кнопку «Изменить событие».

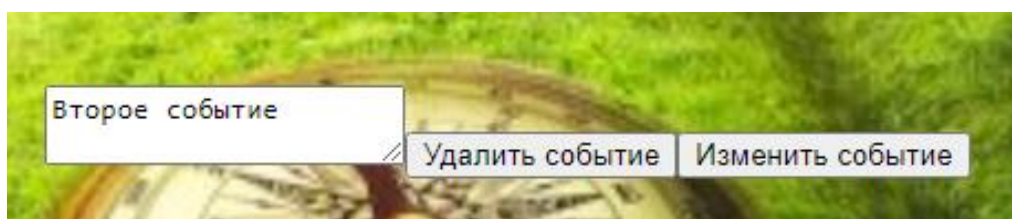


Рисунок 17 – Просмотр события

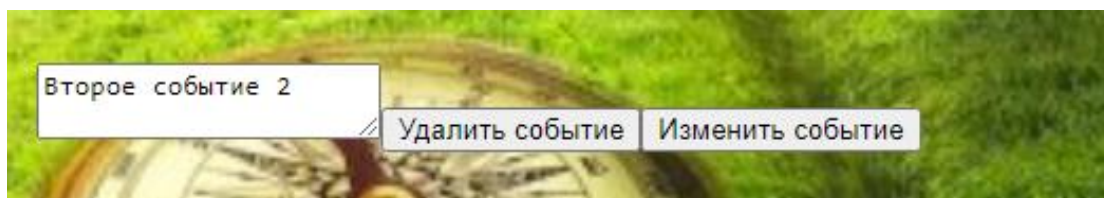


Рисунок 18 – Результат нажатия на кнопку «Изменить событие»

Вывод: соответствует ожидаемым выходным данным.

В ходе тестирования все тесты были успешно проведены и выполнены, весь функционал сайта успешно проведены и результаты тестирования проиллюстрирован на рисунках 10 – 18.

## 6 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

Веб-приложение «Агрегатор календарей» предназначено для формирования единого календаря с различных сервисов. Также в приложении есть навигация по календарю, можно переключаться между месяцами и видеть текущий день, а также добавлять/удалять или изменять событие в календаре.

На главной странице (Рисунок 19) можно выбрать сервис, либо отдельный сервис, либо сам агрегатор. Выбранный сервис будет выделен в блоке меню (Рисунок 20).

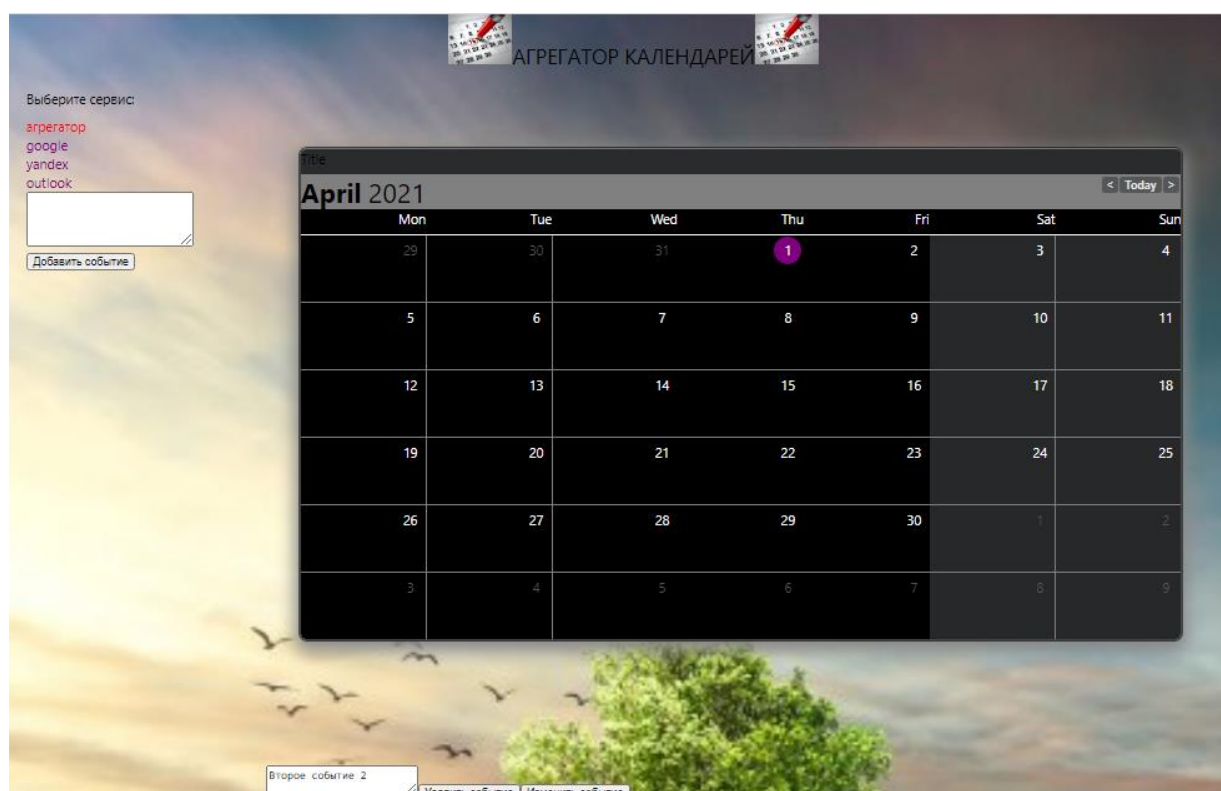


Рисунок 19 – Главная страница

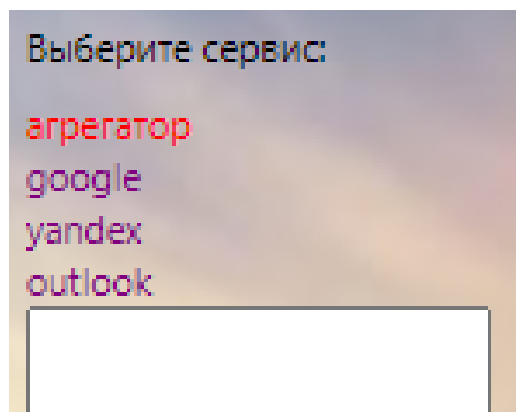


Рисунок 20 – Выбор сервиса

В календаре присутствует навигация по переходу между месяцами. Для этого нужно нажать на одну из кнопок: «<» или «>» в правом верхнем углу календаря, в зависимости от того, в какую сторону хотите «двигаться» (Рисунок 21-22).



Рисунок 21 – Кнопка переключения на предыдущий месяц



Рисунок 22 – Кнопка переключения на следующий месяц

Также в календаре возможно посмотреть текущий день, вне зависимости от того какой у вас показывает месяц на данный момент. Для этого нужно нажать на кнопку «Today», которая также расположена в правом верхнем углу календаря (Рисунок 23).

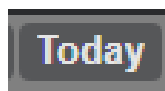


Рисунок 23 – Кнопка отображения текущего дня



В календаре есть возможность добавлять/удалять или изменять событие в выбранном дне.

Для добавления события нужно выбрать день в календаре и затем ввести текст в пустое окно, которое расположено слева от календаря (Рисунок 24), после нажать на кнопку «Добавить событие» и увидеть отображение текста события внизу календаря, в аналогичном окне (Рисунок 25).

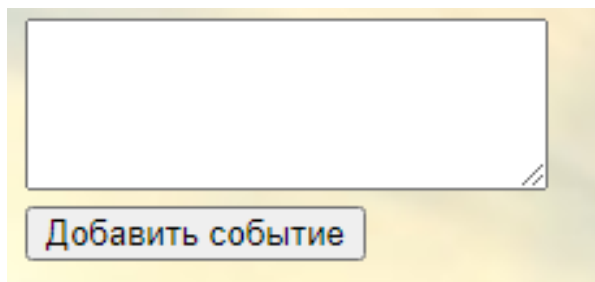


Рисунок 24 – Окно добавления события

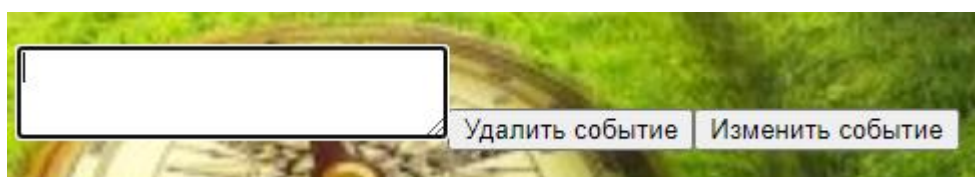


Рисунок 25 – Окно отображения события

Для изменения или удаления события из выбранного дня, нужно: выбрать день, проверить текст события в окне отображения информации (Рисунок 25). Затем нажать на кнопку «Удалить событие», если необходимо его удалить, если же хотите изменить текущее событие, то можно корректировать текст в самом блоке и после нужно нажать кнопку «Изменить событие».

## ЗАКЛЮЧЕНИЕ

Была решена задача по автоматизации календаря. Теперь можно увидеть события без дополнительных действий по переходу в другой сервис.

Были решены следующие задачи:

- Анализ предметной области;
- Проектирование приложения;
- Разработка приложения;
- Тестирование приложения;
- Составление руководства пользователя.

Проектирование, разработка и тестирование приложения позволило увеличить производительность и качество работы пользователя. Теперь пользователю не нужно переходить между сервисами и у него есть возможность делать необходимые действия в одном календаре, что экономит время.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1) Календарь [Электронный ресурс]. URL: <https://ru.wikipedia.org/wiki/Календарь> (дата обращения: 16.03.2021);
- 2) Google Календарь [Электронный ресурс]. URL: <https://calendar.google.com/calendar/u/0/r?tab=uc&pli=1> (дата обращения: 16.03.2021);
- 3) Яндекс.Календарь [Электронный ресурс]. URL: <https://calendar.yandex.ru/week?uid=531691259> (дата обращения: 16.03.2021);
- 4) Outlook календарь [Электронный ресурс]. URL: <https://office.live.com/start/Calendar.aspx?ui=ru%2DRU&rs=US> (дата обращения: 16.03.2021).
- 5) JavaScript [Электронный ресурс]. URL: <https://ru.wikipedia.org/wiki/JavaScript> (дата обращения: 16.03.2021);
- 6) React.js [Электронный ресурс]. URL: <https://ru.wikipedia.org/wiki/React> (дата обращения: 16.03.2021);
- 7) Redux [Электронный ресурс]. URL: <https://ru.wikipedia.org/wiki/Redux> (дата обращения: 16.03.2021);
- 8) Moment.js [Электронный ресурс]. URL: <https://techrocks.ru/2019/02/03/momentjs-and-other-js-libraries-for-moment-js> (дата обращения: 16.03.2021).
- 9) Node.js [Электронный ресурс]. URL: <https://ru.wikipedia.org/wiki/Node.js> (дата обращения: 16.03.2021).
- 10) ASP.NET [Электронный ресурс]. URL: <https://ru.wikipedia.org/wiki/ASP.NET> (дата обращения: 16.03.2021).
- 11) SQL Server [Электронный ресурс]. URL: [https://ru.wikipedia.org/wiki/Microsoft\\_SQL\\_Server](https://ru.wikipedia.org/wiki/Microsoft_SQL_Server) (дата обращения: 16.03.2021).

12) Entity Framework [Электронный ресурс]. URL: [https://ru.wikipedia.org/wiki/ADO.NET\\_Entity\\_Framework](https://ru.wikipedia.org/wiki/ADO.NET_Entity_Framework) (дата обращения: 16.03.2021).

13) VS Code [Электронный ресурс]. URL: [https://ru.wikipedia.org/wiki/Visual\\_Studio\\_Code](https://ru.wikipedia.org/wiki/Visual_Studio_Code) (дата обращения: 16.03.2021).