

**Содержание**

[Введение 5](#_Toc153491512)

[1 Цель и задачи проекта 6](#_Toc153491513)

[1.1 Цель проекта 6](#_Toc153491514)

[1.2 Задачи проекта 6](#_Toc153491515)

[2 Проблематика и актуальность 7](#_Toc153491516)

[3 Обзор существующих программных средств 8](#_Toc153491517)

[3.1 Google Календарь 8](#_Toc153491518)

[3.2 Microsoft’s Outlook Calendar 9](#_Toc153491519)

[3.3 Apple iCloud Calendar 10](#_Toc153491520)

[4 Предпроектное обследование и анализ предметной области 11](#_Toc153491521)

[4.1 Процессы AS IS и TO BE 11](#_Toc153491522)

[4.2 Описание вариантов использования 13](#_Toc153491523)

[4.3 Описание сценариев использования 13](#_Toc153491524)

[5 Выработка требований к проекту 16](#_Toc153491525)

[5.1 Требования к структуре АС в целом 16](#_Toc153491526)

[5.1.1 Требования к режимам функционирования системы 16](#_Toc153491527)

[5.1.2 Требования по диагностированию системы 16](#_Toc153491528)

[5.1.3 Требования к надежности 16](#_Toc153491529)

[5.1.4 Требования к численности и квалификации персонала программы и режимы его работы 16](#_Toc153491530)

[5.1.5 Требования к защите информации от несанкционированного доступа 16](#_Toc153491531)

[5.1.6 Требования к эргономике и технической эстетике 17](#_Toc153491532)

[5.2 Требования к функциям (задачам), выполняемым АС 17](#_Toc153491533)

[5.3 Требования к видам обеспечения АС 18](#_Toc153491534)

[5.3.2 Требования к информационному обеспечению 18](#_Toc153491535)

[5.3.3 Требования к лингвистическому обеспечению 18](#_Toc153491536)

[5.3.4 Требования к программному обеспечению 18](#_Toc153491537)

[5.3.5 Требования к пользовательскому интерфейсу 18](#_Toc153491538)

[6 Выбор и обоснование средств проектирования и реализации 19](#_Toc153491539)

[7 Проектирование архитектуры приложения 22](#_Toc153491540)

[8 Проектирование хранилища данных 24](#_Toc153491541)

[9 Проектирование пользовательского интерфейса 27](#_Toc153491542)

[10 Разработка 35](#_Toc153491543)

[11 Тестирование 40](#_Toc153491544)

[Заключение 44](#_Toc153491545)

[Список использованных источников 45](#_Toc153491546)

# **Введение**

В современном мире технологий и инноваций, электронный календарь стал неотъемлемой частью нашей жизни. Он помогает нам организовывать свое время, планировать встречи и события, не забывать о важных датах и мероприятиях. Электронный календарь, в котором ведется учет коммуникаций играет важную роль в жизни пользователя, его работы и личной жизни. Он позволяет хранить информацию и управлять коммуникационными процессами.

АС по учёту коммуникаций обладает рядом преимуществ перед традиционными бумажными аналогами, такими как удобство использования, хранение информации, наглядность информации, а также позволяет автоматизировать некоторые процессы, например, напоминание о событиях или встречах. АС по учёту коммуникаций предоставляет возможность анализа данных о коммуникациях. Это позволяет сэкономить время пользователя и повысить эффективность его работы. Одним из преимуществ использования АС по учету коммуникаций является то, что он позволяет пользователям легко находить и просматривать информацию о предстоящих мероприятиях и встречах. Кроме того, он может помочь сократить количество ошибок и проблем, связанных с неправильным планированием мероприятий, поскольку он позволяет автоматически обновлять информацию о встречах и событиях.

Таким образом, создание АС по учёту коммуникаций является актуальным и важным проектом, который может значительно улучшить жизнь и работу пользователя.

# **Цель и задачи проекта**

## **1.1 Цель проекта**

Данный проект ставит перед собой целью помочь пользователю с планированием и анализом его рабочих и личных коммуникаций, и осуществлением напоминаний пользователю о запланированных коммуникациях.

## **1.2 Задачи проекта**

Данный проект ставит перед собой задачи:

1. Выполнить анализ предметной области
2. Ознакомиться с существующими похожими программными средствами
3. Изучить типы и способы коммуникаций пользователя, выявить информацию о фактических (состоявшихся) и запланированных коммуникациях
4. Спроектировать и реализовать веб-приложение:
   1. Осуществлять учёт информации о фактических (состоявшихся) и запланированных коммуникациях
   2. Реализовать расчет количества дней до запланированной коммуникации и количества дней с последней коммуникации с конкретным человеком
   3. Реализовать напоминания о запланированной коммуникации
   4. Осуществлять анализ статистики коммуникаций пользователя

# **Проблематика и актуальность**

АС по учёту коммуникаций позволит решить следующие проблемы:

* Контроль за взаимодействием: позволит отслеживать все коммуникации пользователя, что повысит уровень контроля за процессом добавления и удаления информации и поможет избежать недоразумений;
* Управление коммуникациями: предоставляет возможность планирования и координации различных коммуникаций и встреч;
* Экономия времени и ресурсов: благодаря автоматизации некоторых процессов, таких как напоминания о встречах и событиях, АС поможет сэкономить время пользователя и его личные ресурсы;
* Удобство использования: электронный календарь всегда под рукой, на любом устройстве, подключенном к интернету, что позволяет быстро и легко вносить изменения и обновления в расписание;
* Хранение информации: все данные хранятся в одном месте, в отличие от бумажного календаря, где информация может быть утеряна или забыта.

Актуальность проекта по созданию автоматизированной системы по учёту коммуникаций заключается в оптимизации планирования времени, улучшении эффективности работы и повышении уровня организации повседневной жизни человека.

В современном мире, переполненном информацией и задачами, электронный календарь может стать незаменимым помощником для каждого человека, стремящегося к упорядочиванию своего расписания и рациональному использованию своего времени.

# **Обзор существующих программных средств**

## **3.1 Google Календарь**

Google Calendar – это веб-приложение и мобильное приложение, доступное на Android и iOS, которое предлагает возможность планирования и организации событий, как личных, так и рабочих. Интерфейс веб-приложения представлен на рисунке 1.



Рисунок 1 – Интерфейс веб-приложения Google Calendar

Функции Google Calendar:

* Создание и редактирование событий: пользователи могут легко создавать события, указав название, дату, время, местоположение и другие детали;
* Приглашение участников: можно приглашать других пользователей присоединиться к событию через электронную почту или поделиться ссылкой на событие;
* Напоминания: Календарь автоматически отправляет напоминания о предстоящих событиях, чтобы помочь пользователям оставаться организованными;
* Интеграция с другими сервисами Google: Календарь интегрирован с Gmail, Google Meet и другими сервисами, что упрощает планирование событий и управление ими;
* Синхронизация данных: все данные календаря автоматически синхронизируются между устройствами, что позволяет пользователям всегда быть в курсе своих событий.

Однако, Google Calendar может быть сложным для понимания из-за его обширных функций и множества опций. Некоторые из минусов использования Google Calendar включают:

* Сложность в использовании: Google Calendar может показаться сложным для тех, кто только начинает использовать его, из-за обилия функций и возможностей;
* Зависимость от других сервисов: Google Calendar тесно связан с другими сервисами Google, такими как Gmail и Google Meet, что может усложнить его использование для пользователей, которые не используют эти сервисы.

## **3.2 Microsoft’s Outlook Calendar**

Microsoft Outlook Calendar – календарь, разработанный Microsoft и интегрированный в пакет Microsoft Office и Microsoft 365. Он предназначен для планирования и управления событиями, встречами и задачами, а также для обмена информацией с другими пользователями. Интерфейс приложения представлен на рисунке 2.

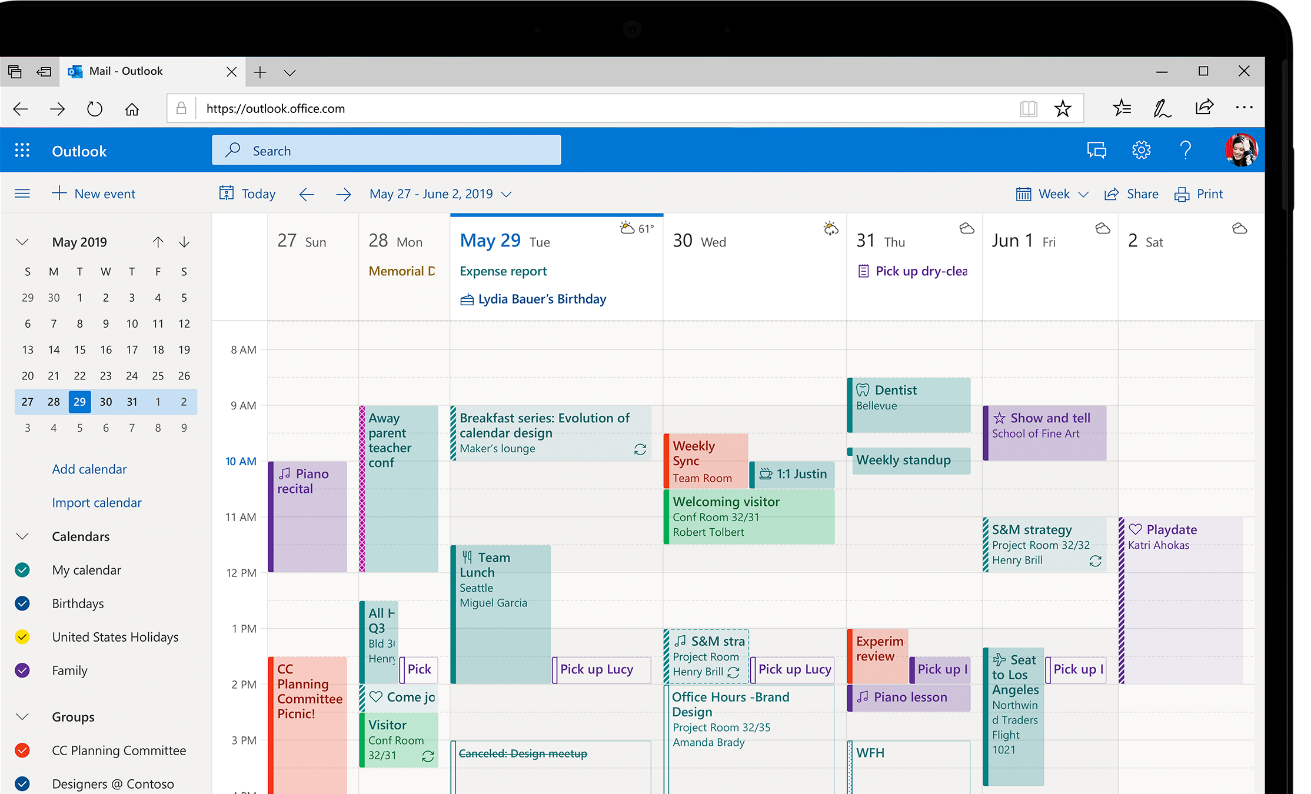


Рисунок 2 – Интерфейс приложения Microsoft Outlook Calendar

Функциями являются: создание событий, напоминаний, приглашений; планирование повторяющихся событий; интеграция с другими приложениями Microsoft; просмотр календаря на разных устройствах и другое. Он также имеет возможность синхронизации данных с другими устройствами через облачные сервисы Microsoft.

Минусами Microsoft Outlook Calendar являются:

* Outlook Calendar может быть дорогим для некоторых пользователей, особенно тех, кто не использует другие продукты Microsoft;
* Интерфейс может быть сложным и запутанным для новичков;
* Некоторые функции доступны только в премиум-версии, что может увеличить стоимость для некоторых пользователей.

## **3.3 Apple iCloud Calendar**

Apple iCloud Calendar – это календарь, созданный компанией Apple и доступный пользователям iCloud. Он предназначен для организации событий, встреч и напоминаний на устройствах Apple, таких как iPhone, iPad и Mac. iCloud Calendar предлагает широкий спектр функций, включая создание событий, приглашений, повторяющихся событий, синхронизацию данных между устройствами и интеграцию с другими сервисами Apple, такими как Mail, Messages и Reminders. Интерфейс приложения представлен на рисунке 3.

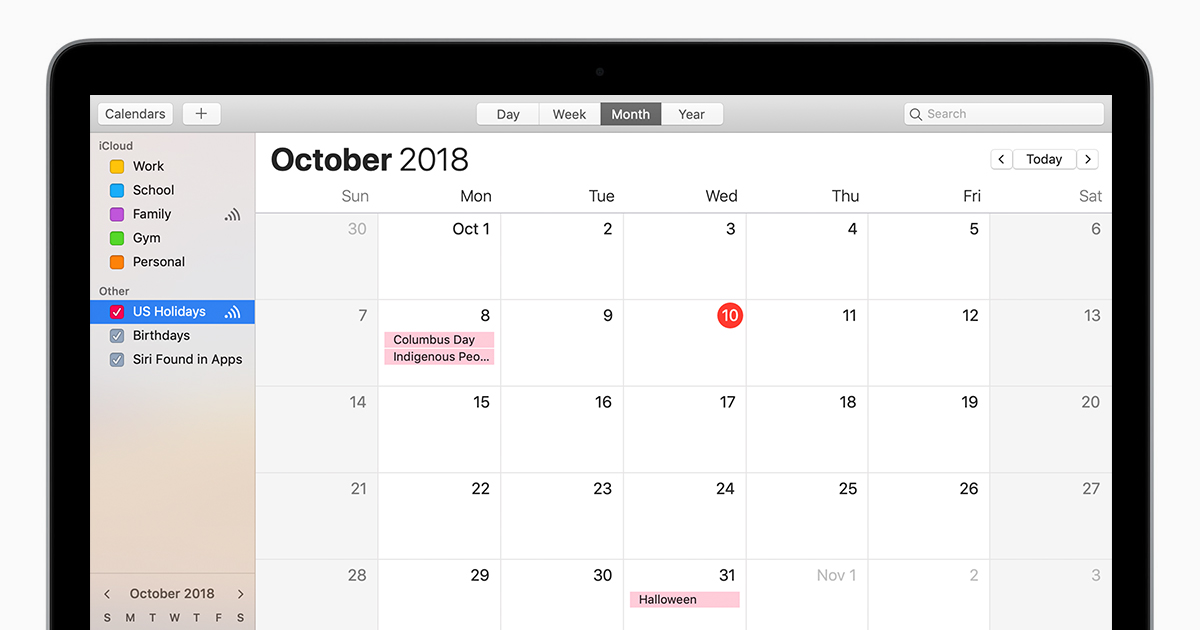


Рисунок 3 – Интерфейс приложения Apple iCloud Calendar

Минусами Apple’s iCloud Calendar являются:

* iCloud Calendar может быть менее знакомым и удобным для пользователей, не использующих устройства Apple;
* Некоторые функции могут быть ограничены в бесплатной версии iCloud;
* iCloud может быть дороже для некоторых пользователей, особенно если они не используют другие продукты Apple.

В процессе изучения и анализа существующих программных решений, таких как Google Calendar, Microsoft Outlook Calendar и Apple iCloud Calendar, был сделан вывод о необходимости создания программного обеспечения для учета коммуникаций, основываясь на определении основных функциональных возможностей этих средств, а также недостатков их использования.

# **Предпроектное обследование и анализ предметной области**

## **4.1 Процессы AS IS и TO BE**

BPMN технология (Business Process Model and Notation) – язык моделирования бизнес-процессов, который является промежуточным звеном между формализацией/визуализацией и воплощением бизнес-процесса. С помощью моделирования мы можем описать любые бизнес-процессы, и они могут выполняться в самых разных системах управления. BPMN в графическом виде отражает последовательность работ бизнес-процессов и логику их выполнения. Дальше эту схему используют для управления бизнес-процессами – ищут слабые участки и оптимизируют их.

Бизнес-процесс – это логически связанная последовательность действий, направленных на создание продуктов или услуг. К бизнес-процессам относятся все повторяющиеся операции, которые помогают решать задачи бизнеса и получать доход.

Основные преимущества использования BPMN:

* Стандартизация и визуализация бизнес-процессов: BPMN позволяет создать единое представление процесса для всех участников команды, что облегчает коммуникацию и понимание;
* Улучшение эффективности процессов: BPMN помогает выявить слабые места в процессах, определить возможные проблемы и найти оптимальные решения;
* Внедрение и управление изменениями: BPMN обеспечивает наглядное представление того, как изменения в одном элементе процесса могут повлиять на другие элементы.

AS IS (как есть) в BPMN – это текущее состояние или модель бизнес-процесса без каких-либо изменений или улучшений. Это базовая модель, которая отражает то, как процесс выполняется в настоящее время. AS IS модель может быть использована для анализа текущего процесса, выявления проблем и определения возможностей для улучшения.

На схеме AS IS (см. рисунок 4) показаны слабые места текущего состояния, такие как:

1. В бесплатной версии системы нет нужных функций – пользователь не хочет или не может платить за нужные ему функции.
2. Система сложна в использовании – пользователю будет трудно разобраться в системе.
3. Другой сервис, тесно связанный с выбранным, которым не обладает пользователь – пользователь не хочет или не может использовать другой связанный сервис.

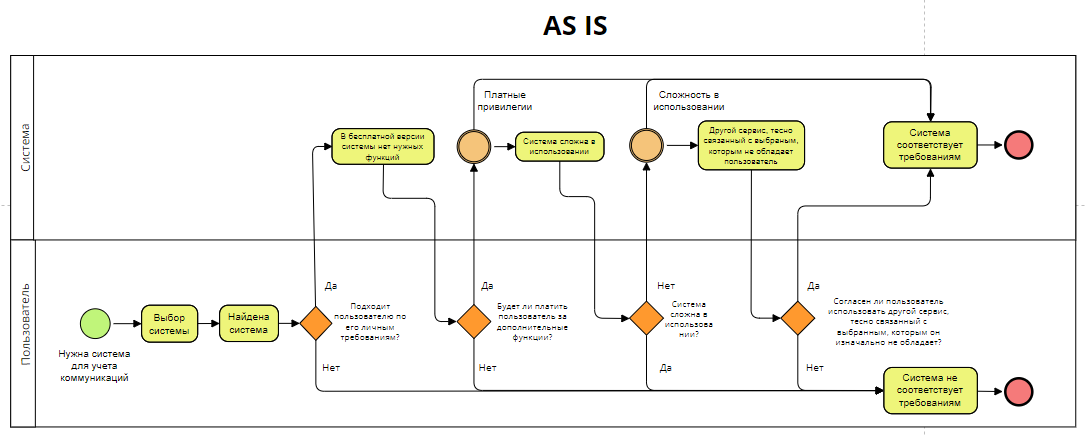


Рисунок 4 – AS IS

Данные примеры могут привести к несоответствию системы с личными требованиями пользователя, следовательно, пользователь не станет использовать данную систему.

TO BE (как должно быть) в BPMN – это модель бизнес-процесса после внесения изменений или улучшений. TO BE модель представляет собой желаемый или оптимальный вид процесса, который должен быть достигнут в результате изменений. Она может включать новые этапы, измененные последовательности действий, улучшенные потоки данных и другие изменения, направленные на повышение эффективности и результативности процесса.

На схеме TO BE (см. рисунок 5) показаны решения и улучшения для слабых мест в текущего состоянии, такие как:

1. Система бесплатна – пользователь будет пользоваться системой, потому что за нее не нужно платить.
2. Система проста в использовании – пользователю будет легко разобраться в системе.
3. У системы нет другого сервиса, тесно связанный с выбранным, которым не обладает пользователь – пользователь не обязан использовать другой связанный сервис.

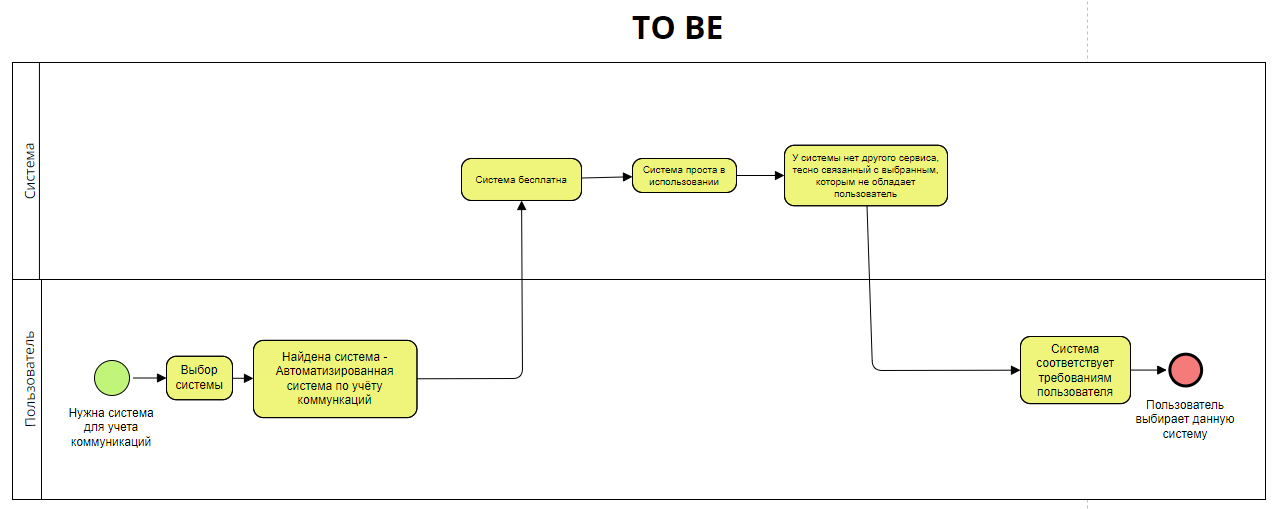


Рисунок 5 – TO BE

При решении данных примеров, пользователь скорее всего станет использовать данную систему.

## **4.2 Описание вариантов использования**

Диаграмма вариантов использования (англ. use-case diagram) – диаграмма, описывающая, какой функционал разрабатываемой программной системы доступен каждой группе пользователей.

Диаграмма помогает определить основные сценарии использования системы, выявить возможные варианты взаимодействия пользователей с системой и определить границы системы. Это важный инструмент для анализа требований и проектирования систем, который помогает обеспечить понимание и согласованность требований между различными заинтересованными сторонами.

UML (англ. "Unified Modeling Language") – стандартизированный язык моделирования при проектировании программ.

Диаграмма вариантов использования, отображающая функциональные требования к системе с точки зрения пользователя изображена на рисунке 6.

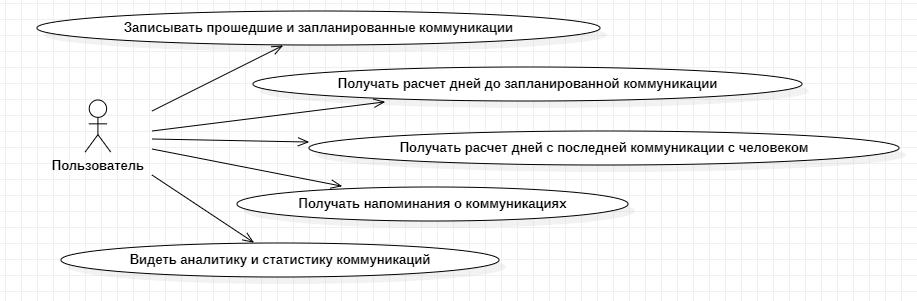


Рисунок 6 – Диаграмма вариантов использования

На диаграмме представлены функциональные требования к системе с точки зрения пользователя, такие как:

1. Записывать прошедшие и запланированные коммуникации
2. Получать расчет дней до запланированной коммуникации
3. Получать расчет дней с последней коммуникации с человеком
4. Получать напоминания о коммуникациях
5. Видеть аналитику и статистику коммуникаций

## **4.3 Описание сценариев использования**

Сценарии использования – это сценарий взаимодействия пользователя с программным продуктом для достижения конкретной цели.

Функциональные требования к системе с точки зрения пользователя:

1. Записывать прошедшие и запланированные коммуникации (см. таблица 1)

Предусловие: Пользователь находится на главной странице

Таблица 1 – Функциональное требование «Записывать прошедшие и запланированные коммуникации»

| **Пользователь** | **Система** |
| --- | --- |
| *Какое физическое действие произвел пользователь?* | *Как отреагировала система?* |
| Пользователь нажал на кнопку "+", означающая добавление новой записи о коммуникации | Система вывела форму для заполнения |
| Пользователь записывает данные о коммуникации (название, дата, с кем осуществляется коммуникация, тип, описание | В форму для заполнения вносятся данные |
| Пользователь нажимает кнопку "Сохранить" | Система заполняет базу данных |

1. Получать расчет дней до запланированной коммуникации (см. таблица 2)

Предусловие: Пользователь находится на главной странице

Таблица 2 – Функциональное требование «Получать расчет дней до запланированной коммуникации»

| **Пользователь** | **Система** |
| --- | --- |
| *Какое физическое действие произвел пользователь?* | *Как отреагировала система?* |
| Пользователь переходит во вкладку "Напоминания" | Система открывает вкладку "Напоминания" |
|  | Система ведет подсчет дней к записям в информационном поле страницы |

1. Получать расчет дней с последней коммуникации с человеком (см. таблица 3)

Предусловие: Пользователь находится на главной странице

Таблица 3 – Функциональное требование «Получать расчет дней с последней коммуникации с человеком»

| **Пользователь** | **Система** |
| --- | --- |
| *Какое физическое действие произвел пользователь?* | *Как отреагировала система?* |
| Пользователь переходит во вкладку "Люди" | Система открывает вкладку "Люди" |
| Пользователь отмечает людей, для которых будет осуществляться подсчет дней с последней коммуникации | Система отмечает и сохраняет в базе данных список людей |
| Пользователь переходит во вкладку "Напоминания" | Система открывает вкладку "Напоминания" |
|  | Система ведет подсчет дней к записям в информационном поле страницы |

1. Получать напоминания о коммуникациях (см. таблица 4)

Предусловие: Пользователь находится на главной странице

Таблица 4 – Функциональное требование «Получать напоминания о коммуникациях»

| **Пользователь** | **Система** |
| --- | --- |
| *Какое физическое действие произвел пользователь?* | *Как отреагировала система?* |
|  | Система выводит сообщение-напоминание о коммуникации |

1. Видеть аналитику и статистику коммуникаций (см. таблица 5)

Предусловие: Пользователь находится на главной странице

Таблица 5 – Функциональное требование «Видеть аналитику и статистику коммуникаций»

| **Пользователь** | **Система** |
| --- | --- |
| *Какое физическое действие произвел пользователь?* | *Как отреагировала система?* |
| Пользователь переходит во вкладку "Статистика" | Система открывает вкладку "Статистика" |
|  | Система выводит статистику и аналитику по коммуникациям в трёх разделах "Запланированные", "Состоявшиеся" и "Все коммуникации" |

# **Выработка требований к проекту**

* 1. **Требования к структуре АС в целом**
     1. **Требования к режимам функционирования системы**

АС должна поддерживать режим функционирования – основной режим, обеспечивающий работу пользователя каждый день и выполнение своих функций – сбор, обработка и загрузка данных, хранение данных и предоставление отчетности.

* + 1. **Требования по диагностированию системы**

Диагностика и профилактика технических средств, проводится раз в месяц. Проверка целостности данных и нарушений проводится по мере необходимости. Проверка программного и аппаратного обеспечения проводится по мере необходимости. Улучшение приложения будет осуществляться самими разработчиком.

* + 1. **Требования к надежности**

АС должна обладать устойчивостью к отказам оборудования и программных систем. Для надежной работы комплекса необходимы высоконадежные аппаратные и программные системы.

С целью обеспечения надежного функционирования в системе должны быть предусмотрены:

* Сохранение целостности данных в БД при нештатном завершении работы системы;
* Сохранение работоспособности программного обеспечения при некорректных действиях пользователя (ввод неверных данных, нештатное завершение действий и т.п.).

Также должна быть обеспечена работоспособность системы в целом в случае возникновения сбоев и аварий.

* + 1. **Требования к численности и квалификации персонала программы и режимы его работы**

Разработчик приложения – обеспечивает работоспособный режим функционирования и улучшает его в следующих обновлениях. Пользователь – тот, кто непосредственно пользуется данной веб-страницей и может вносить свои данные.

* + 1. **Требования к защите информации от несанкционированного доступа**

При работе с АС, необходимо, чтобы она была защищена от попыток изменения и разрушения, тем самым, она нуждается в защите информации от несанкционированного доступа. Осуществление защиты происходит посредством введения логинов и паролей, для пользователей и разработчика. ИС защищается паролем. Соответственно существует два вида доступа:

* Доступ пользователям – могут вносить, редактировать, удалять и просматривать свои данные о коммуникациях;
* Доступ разработчика – может вносить изменения в структуру АС.
  + 1. **Требования к эргономике и технической эстетике**

Требованиями являются:

* должно быть обеспечено наличие локализованного (русскоязычного) интерфейса пользователя;
* должен использоваться понятный и читабельный шрифт, соответствующий цветовой палитре;
* размер шрифта должен быть в пределах 12-16 пикселей;
* цветовая палитра должна быть приятной и не вызывающей, сочетающей в себе спокойные цвета;
* при возникновении ошибок на экран монитора должно выводиться сообщение с наименованием ошибки и с рекомендациями по её устранению на русском языке.
  1. **Требования к функциям (задачам), выполняемым АС**

Требования к функция представлены в таблице 6.

Таблица 6 – Требования к функция (задачам)

| **Задача** | **Требуемый результат** |
| --- | --- |
| Учёт информации о фактических (состоявшихся) и запланированных коммуникациях | Хранение информации в БД, вывод информации в понятном для пользователя виде на главной странице веб-ресурса |
| Расчет количества дней до запланированной коммуникации | Хранение информации в БД, вывод информации в понятном для пользователя виде на главной странице веб-ресурса |
| Расчет количества дней с последней коммуникации с конкретным человеком | Вывод информации в понятном для пользователя виде |
| Напоминание о запланированной коммуникации | Вывод информации в понятном для пользователя виде |
| Анализ статистики коммуникаций | Хранение информации в БД, вывод информации в понятном для пользователя виде |

* 1. **Требования к видам обеспечения АС**
     1. **Требования к информационному обеспечению**

АС должна предоставлять доступ к своей информационной базе, в состав которого входит БД. В качестве входной информации выступает информация, поступающая в БД, а выходной информацией служит изменение информации в БД.

* + 1. **Требования к лингвистическому обеспечению**

Шрифт ввода-вывода данных – кириллица.

* + 1. **Требования к программному обеспечению**

ПО должно соответствовать критериям:

* иметь доступ к сети Интернет.
  + 1. **Требования к пользовательскому интерфейсу**

Пользовательский интерфейс должен соответствовать следующим требованиям:

* Эффективные интерфейсы должны быть очевидными и внушать своему пользователю чувство контроля. Необходимо, чтобы пользователь мог одним взглядом окинуть весь спектр своих возможностей, понять, как достичь своих целей и выполнить работу;
* Эффективные интерфейсы не должны беспокоить пользователя внутренним взаимодействием с системой. Необходимо бережное и непрерывное сохранение работы, с предоставлением пользователю возможности отменять любые действия в любое время.

1. **Выбор и обоснование средств проектирования и реализации**

В дизайне используется:

Figma – сервис для разработки интерфейсов и прототипирования с возможностью организации совместной работы в режиме реального времени.

Figma имеет ряд преимуществ, которые делают ее одной из лучших платформ для дизайнеров и разработчиков:

1. Простота использования: Figma предлагает интуитивно понятный интерфейс, который упрощает навигацию и работу с различными инструментами. Даже новички могут быстро освоиться и начать создавать профессиональные дизайны.
2. Богатый функционал: Figma обладает широким спектром инструментов для работы с векторной графикой, текстом, слоями, формами и т. Д. Она также предлагает продвинутые функции, такие как анимация, прототипирование, адаптивный дизайн и возможность работы со сторонними шрифтами.
3. Совместная работа: Figma поддерживает совместную работу в реальном времени, что позволяет командам работать над проектом одновременно и обсуждать изменения в режиме реального времени. Это помогает избежать недопонимания и ускоряет процесс разработки.
4. Кроссплатформенность: Figma доступна на различных устройствах и операционных системах, включая Windows, macOS, Linux, iOS и Android.

Для хранения используется:

GitHub – популярный веб-сервис для хостинга IT-проектов и их совместной разработки. GitHub предоставляет возможность хранения кода, управления версиями, отслеживания ошибок, участия в обсуждении и совместной разработке проектов.

Выбор в пользу GitHub обусловлен множеством его преимуществ:

1. Гибкость: Гибкая система контроля доступа к коду.
2. Сообщество: Возможность взаимодействия с сообществом разработчиков.
3. Инструментарий: Наличие инструментария для непрерывной интеграции и развертывания кода.
4. Обсуждения: Наличие встроенной системы обсуждения и баг-трекинга.
5. Деление ответственности: Возможность организации коллективных проектов с разделением зон ответственности.

В веб-разработке используется:

HTML5 – язык гипертекстовой разметки для структурирования и представления содержимого Интернет-ресурса. Он определяет структуру документа, содержит теги (элементы) для выделения различных частей текста, таких как заголовки, абзацы, списки и т.д.

Выбор в пользу HTML обусловлен следующими преимуществами:

1. Широкая поддержка: HTML поддерживается большинством браузеров, что обеспечивает доступность и совместимость веб-страниц.
2. Простота: HTML имеет простой и понятный синтаксис, что облегчает его изучение и использование.
3. Эффективность: HTML-страницы обычно имеют небольшой размер, что ускоряет загрузку и снижает затраты на трафик.
4. Универсальность: HTML подходит для создания веб-страниц различных типов и размеров, от простых статических страниц до сложных динамических сайтов.

CSS3 – язык стилей, который позволяет определить внешний вид и отображение HTML-документов. Представляет собой формальный язык, реализованный с помощью языка разметки.

Выбор в пользу CSS обусловлен следующими причинами:

1. Гибкость: CSS позволяет стилизовать HTML-элементы различными способами, такими как изменение цвета, шрифта, размера и т.д. Это позволяет создавать уникальные и привлекательные веб-страницы.
2. Совместимость: CSS поддерживается всеми современными браузерами, что гарантирует, что ваш веб-сайт будет корректно отображаться на всех устройствах и платформах.
3. Простота в использовании: CSS имеет простой и понятный синтаксис, который легко освоить и использовать даже начинающим разработчикам.
4. Каскадирование: CSS позволяет применять стили к элементам в иерархическом порядке, что упрощает процесс стилизации веб-страницы и делает его более эффективным.
5. Наследование: CSS позволяет наследовать стили от родительских элементов к дочерним, что также упрощает процесс стилизации и делает его более гибким.

Javascript – язык программирования, используемый для создания интерактивных веб-сайтов и приложений. Он позволяет добавлять различные функции, такие как анимацию, обработку событий и взаимодействие с пользователем.

Выбор JavaScript обусловлен следующими причинами:

1. Кроссбраузерность: JavaScript работает во всех основных браузерах, что делает его универсальным решением для веб-разработки.
2. Интерактивность: JavaScript позволяет создавать динамические элементы на веб-страницах, такие как всплывающие окна, анимации и интерактивные элементы.
3. Простота изучения: JavaScript имеет простой и понятный синтаксис, что делает его доступным для изучения даже новичкам в программировании.
4. Поддержка сообщества: JavaScript имеет большое сообщество разработчиков, которые постоянно создают новые библиотеки, инструменты и руководства, что облегчает процесс обучения и разработки.

PHP – серверный язык программирования, который широко используется для создания динамических веб-приложений и сайтов.

Выбор PHP обусловлен следующими преимуществами:

1. Простота: PHP имеет простой и понятный синтаксис, что позволяет быстро освоить его даже новичкам.
2. Производительность: PHP является одним из самых быстрых языков программирования, что делает его идеальным для создания высоконагруженных проектов.
3. Поддержка: PHP имеет огромное сообщество разработчиков, что обеспечивает доступ к большому количеству документации и примеров кода.
4. Безопасность: PHP имеет встроенные функции для защиты от распространенных видов атак, что снижает риск взлома сайта.
5. Интеграция с другими технологиями: PHP легко интегрируется с другими популярными технологиями, такими как MySQL, JavaScript, HTML и CSS.

MySQL – система управления базами данных, используемая для хранения и обработки информации.

Выбор MySQL обусловлен рядом причин:

1. Простота использования: MySQL обладает простым и понятным интерфейсом, который позволяет даже новичкам быстро освоить основы работы с базами данных.
2. Поддержка: MySQL имеет огромное сообщество пользователей и разработчиков, которые предоставляют помощь и поддержку в случае возникновения проблем.
3. Производительность: MySQL является одной из самых быстрых систем управления базами данных, что позволяет обрабатывать большие объемы информации с высокой скоростью.
4. Безопасность: MySQL имеет встроенные механизмы защиты от различных видов атак, таких как SQL-инъекции и XSS-атаки.
5. Интеграция: MySQL может быть интегрирована с другими языками программирования и технологиями, такими как PHP, Python, Java и др.
6. **Проектирование архитектуры приложения**

Архитектура приложения – это структура или организация различных компонентов программного обеспечения, которые вместе выполняют определенные функции. Она определяет, как различные части приложения взаимодействуют друг с другом, и какие технологии и подходы используются для разработки.

Схема архитектуры приложения нужна для визуализации и понимания общей структуры приложения, а также для выявления возможных проблем и оптимизации производительности. Схема позволяет разработчикам и другим участникам проекта лучше понять требования к приложению и определить зависимости между компонентами.

Схема архитектуры приложения приведена на рисунке 7.

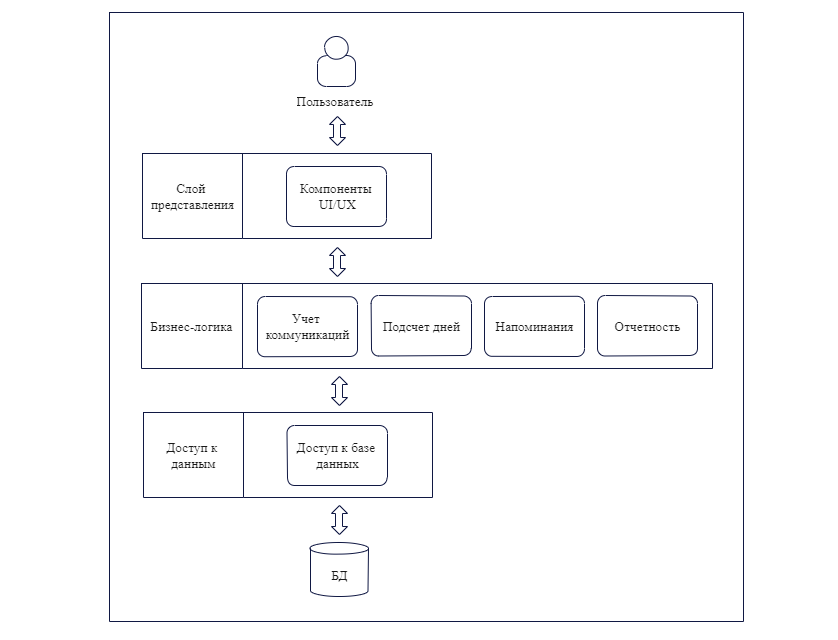
[](https://github.com/ArbakovaAnastasia/AccountingForCommunications/raw/main/%D0%9F%D1%80%D0%BE%D0%B5%D0%BA%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5/%D0%B0%D1%80%D1%85%D0%B8%D1%82%D0%B5%D0%BA%D1%82%D1%83%D1%80%D0%B0.png)

Рисунок 7 – Схема архитектуры приложения

На схеме представлена взаимосвязь частей архитектуры приложения: слоя представления, бизнес-логика, доступ к данных.

Слой представления в архитектуре приложения необходим для связи между пользовательским интерфейсом и бизнес-логикой. Он отвечает за отображение данных на экране, обработку пользовательского ввода и передачу информации на другие уровни приложения. Основные функции слоя представления включают:

* Отображение данных из модели на экране;
* Обработка событий, таких как нажатия кнопок, ввод текста и т. д.;
* Валидация пользовательского ввода;
* Передача данных на уровень бизнес-логики или модели;
* Обновление пользовательского интерфейса в ответ на изменения в данных или состоянии приложения.

Бизнес-логика приложения состоит из задач: учет коммуникаций, подсчет дней, напоминания, отчетность.

Слой бизнес-логики необходим для выполнения сложных операций и алгоритмов, связанных с обработкой данных и выполнением задач, определенных функциональностью приложения. Он включает в себя правила и процедуры, которые управляют работой приложения и обеспечивают выполнение бизнес-требований. Этот слой отделяет представление и доступ к данным от бизнес-правил и предоставляет возможность легко изменять и обновлять логику без влияния на другие компоненты системы. Он также обеспечивает гибкость и масштабируемость приложения, позволяя добавлять новые функции и возможности без необходимости изменения других частей системы.

Слой доступа к данным обеспечивает взаимодействие приложения с источниками данных, такими как базы данных, файловые системы, API и другие. В данной случае, с базой данных. Он отвечает за получение данных из источников, их преобразование и передачу на следующий уровень архитектуры. Основными функциями слоя доступа к данным являются:

* Подключение к источникам данных;
* Выполнение запросов к базам данных;
* Трансформация и форматирование данных для передачи на следующий уровень;
* Обеспечение безопасности и контроля доступа к данным;
* Обработка ошибок и исключений, возникающих при работе с данными.

1. **Проектирование хранилища данных**

Сущность БД – объект, который может быть однозначно идентифицирован в рамках предметной области. Сущности используются для организации данных в базе данных и для установления связей между ними. Они позволяют структурировать информацию, облегчают поиск и извлечение данных, а также позволяют контролировать целостность данных.

Для предметной области, определены сущности: пользователь; коммуникация; человек, с которым коммуникация.

Пара сущностей – это два объекта, которые связаны друг с другом и имеют определенные отношения между собой. Пары сущностей используются для описания связей между объектами в базе данных, что позволяет организовать информацию и упростить поиск и извлечение связанных данных.

На рисунке 8 изображены пары сущностей, между которыми можно установить связь.

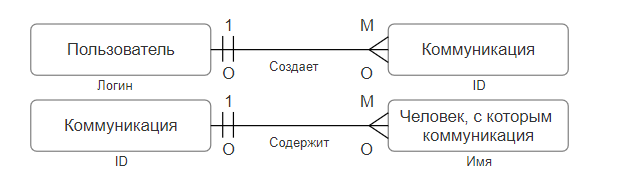


Рисунок 8 – Пары сущностей

В таблицах 7, 8 и 9 описаны данные хранящиеся для сущностей «Пользователь», «Коммуникация», «Человек с которым коммуникация».

Таблица 7 – Сущность «Пользователь»

| **Пользователь** |
| --- |
| Логин (PK) |
| Пароль |
| Электронная почта |
| Имя |
| Дата рождения |
| Дата регистрации |
| Коммуникация (FK) |

Таблица 8 – Сущность «Коммуникация»

| **Коммуникация** |
| --- |
| ID (PK) |
| Название |
| Дата планирования |
| Время планирования |
| Тип коммуникации |
| Человек, с которым коммуникация (FK) |
| Описание |
| Статус |
| Дата фактическая |
| Время фактическое |

Таблица 9 – Сущность «Человек с которым коммуникация»

| **Человек с которым коммуникация** |
| --- |
| Имя (PK) |
| Количество коммуникаций |

Логическая модель базы данных – это описание структуры данных, включая таблицы, поля и связи между ними, в терминах, не зависящих от конкретной СУБД. Она нужна для того, чтобы определить, как данные организованы и связаны друг с другом в приложении, и чтобы разработчики могли понять, какие данные требуются для выполнения определенных функций.

Логическая модель базы данных представлена на рисунке 9.

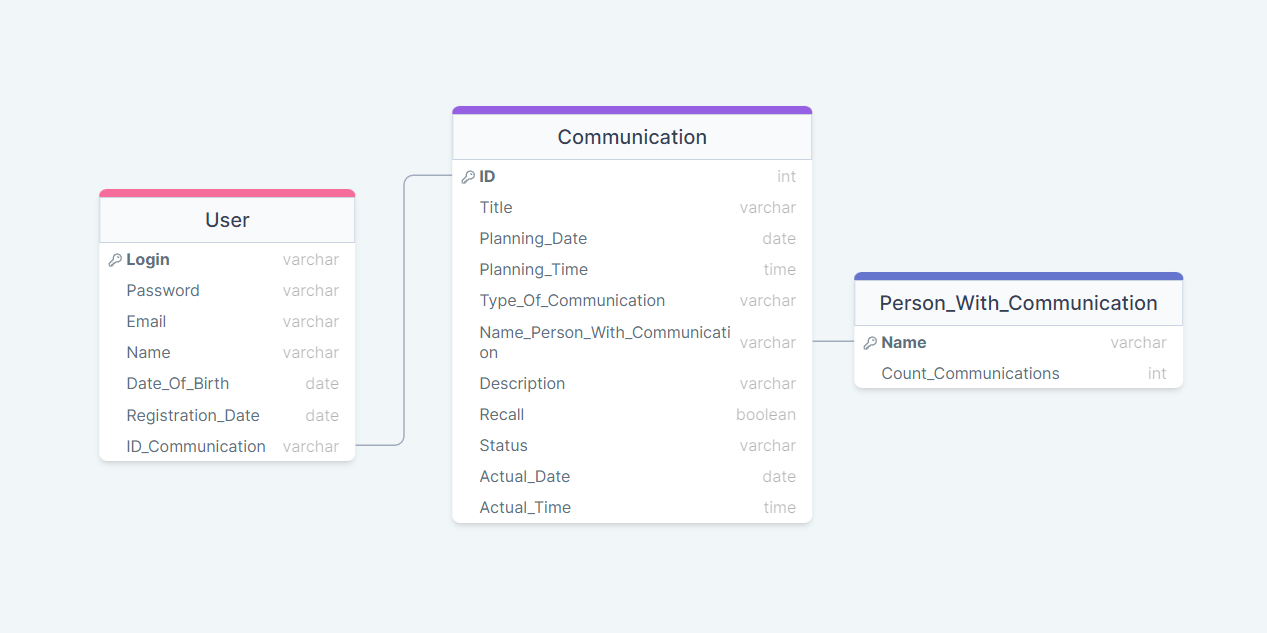


Рисунок 9 – Логическая модель БД

Функции, выполняемые логической моделью:

* Определение структуры данных: логическая модель определяет, какие таблицы, поля и отношения существуют в базе данных;
* Упрощение разработки: логическая модель упрощает процесс разработки, поскольку разработчики могут сосредоточиться на структуре данных, а не на деталях реализации;
* Обеспечение совместимости: логическая модель обеспечивает совместимость между различными системами, поскольку она описывает данные в абстрактных терминах, а не в терминах конкретных СУБД.

# **Проектирование пользовательского интерфейса**

Верхнеуровневое определение экранов – это процесс анализа и определения основных элементов пользовательского интерфейса, которые будут использоваться для взаимодействия пользователя с системой или приложением. Это включает в себя определение расположения и размера различных элементов на экране, таких как кнопки, текстовые поля, списки и таблицы, а также определение их функций и поведения.

Верхнеуровневое определение экранов является важным этапом в разработке пользовательского интерфейса и имеет следующие преимущества:

* Улучшение коммуникации: Верхнеуровневое определение экранов помогает разработчикам и дизайнерам четко понимать и согласовывать требования к пользовательскому интерфейсу, что снижает риск недопонимания и обеспечивает последовательный и качественный дизайн;
* Повышение эффективности: Определение основных элементов экрана на раннем этапе проекта помогает избежать добавления ненужных или дублирующих функций, что экономит время и ресурсы при разработке;
* Создание основы для тестирования: Верхнеуровневое определение экранов служит основой для тестирования пользовательского интерфейса. Оно помогает определить, какие элементы интерфейса должны быть проверены, и позволяет разработчикам сосредоточиться на важных функциях;
* Обеспечение качества: Верхнеуровневое определение экранов обеспечивает качество пользовательского интерфейса на протяжении всего процесса разработки, гарантируя, что каждый экран соответствует общим стандартам и требованиям.

Верхнеуровневое определение экранов представлено на рисунке 10.



Рисунок 10 – Верхнеуровневое определение экранов

Имеются разделы “Aссount” и “System”.

Раздел “Aссount” состоит из таких частей, как: вход (Login), регистрация (Register), профиль (Profile), редактирование профиля (Edit\_Profile).

Раздел “System” состоит из таких частей, как: календарь (Calendar), напоминания (Reminder), статистика (Statistics).

Из календаря (Calendar) вызывается форма ввода коммуникации (Input\_Form).

Назначение экранов – это процесс определения того, какой экран или экраны будут использоваться для отображения определенной функции или задачи в приложении или системе. Это может включать в себя выбор между различными типами экранов, такими как главный экран, экран настроек или экран регистрации, а также определение того, какие элементы должны быть на каждом экране для обеспечения эффективного взаимодействия с пользователем.

Назначение экранов с краткими названиями, полями ввода для валидации и описанием экрана и его поведения необходимо для создания четкой и понятной документации, которая описывает каждый экран в приложении. Эта документация помогает другим разработчикам и командам понять, как работает приложение, какие функции выполняет каждый экран и какие правила валидации применяются к полям ввода. Кроме того, краткое название экрана позволяет быстро идентифицировать его в коде и облегчает навигацию по приложению. Описание экрана и его поведения также помогает улучшить коммуникацию между разработчиками и повышает эффективность работы над проектом.

Таблица назначения экранов представлена на рисунке 11.

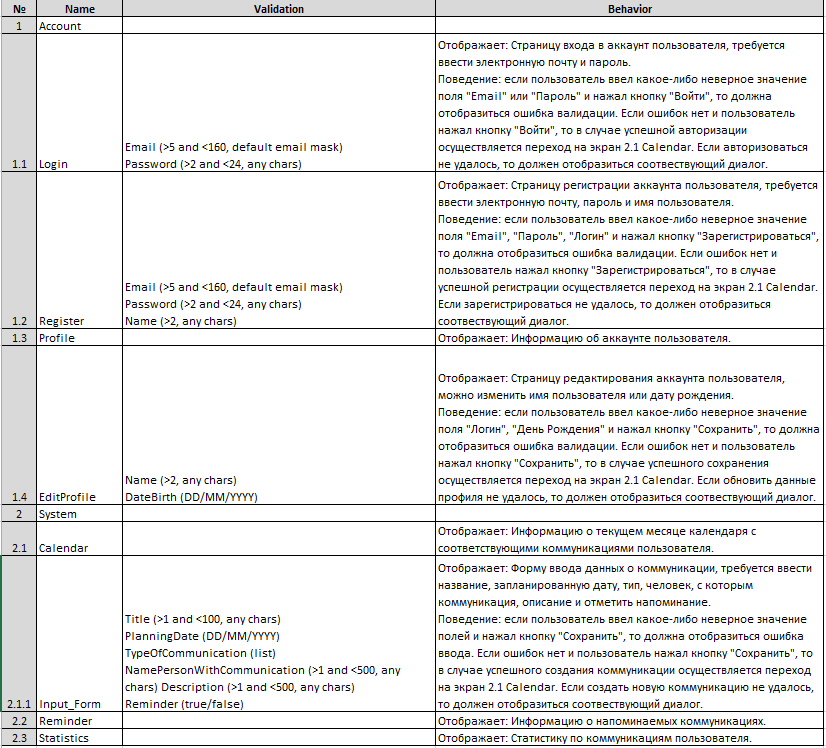


Рисунок 11 – Назначение экранов

Ключевые столбцы таблицы:

1. Номер экрана
2. Краткое название (Name)
3. Поля ввода для валидации (Validation)
4. Описание экрана и его поведения (Behavior)

Номер экрана – используется для упорядочивания экранов в приложении.

Краткое название (Name) - краткое и понятное название экрана, которое помогает другим разработчикам быстро понять, о каком экране идет речь.

Поля ввода для валидации (Validation) - список правил, которые проверяются при вводе данных пользователем. Это помогает гарантировать, что данные, введенные пользователем, соответствуют требованиям приложения.

Описание экрана и его поведения (Behavior) - подробное описание того, как работает экран, какие функции он выполняет и как взаимодействует с другими экранами приложения.

Отрисовка и описание макетов экранов – это процесс создания визуальных представлений каждого экрана в приложении, включая расположение и размер всех элементов пользовательского интерфейса. Это помогает разработчикам лучше понять, как будет выглядеть приложение для пользователей, и позволяет внести необходимые изменения до начала разработки. Описание макетов также включает в себя описание поведения каждого элемента, такие как кнопки, ссылки и другие интерактивные элементы.

* 1. Вход (Login) (см. рисунок 12)

Отображает: Страницу входа в аккаунт пользователя, требуется ввести электронную почту и пароль.

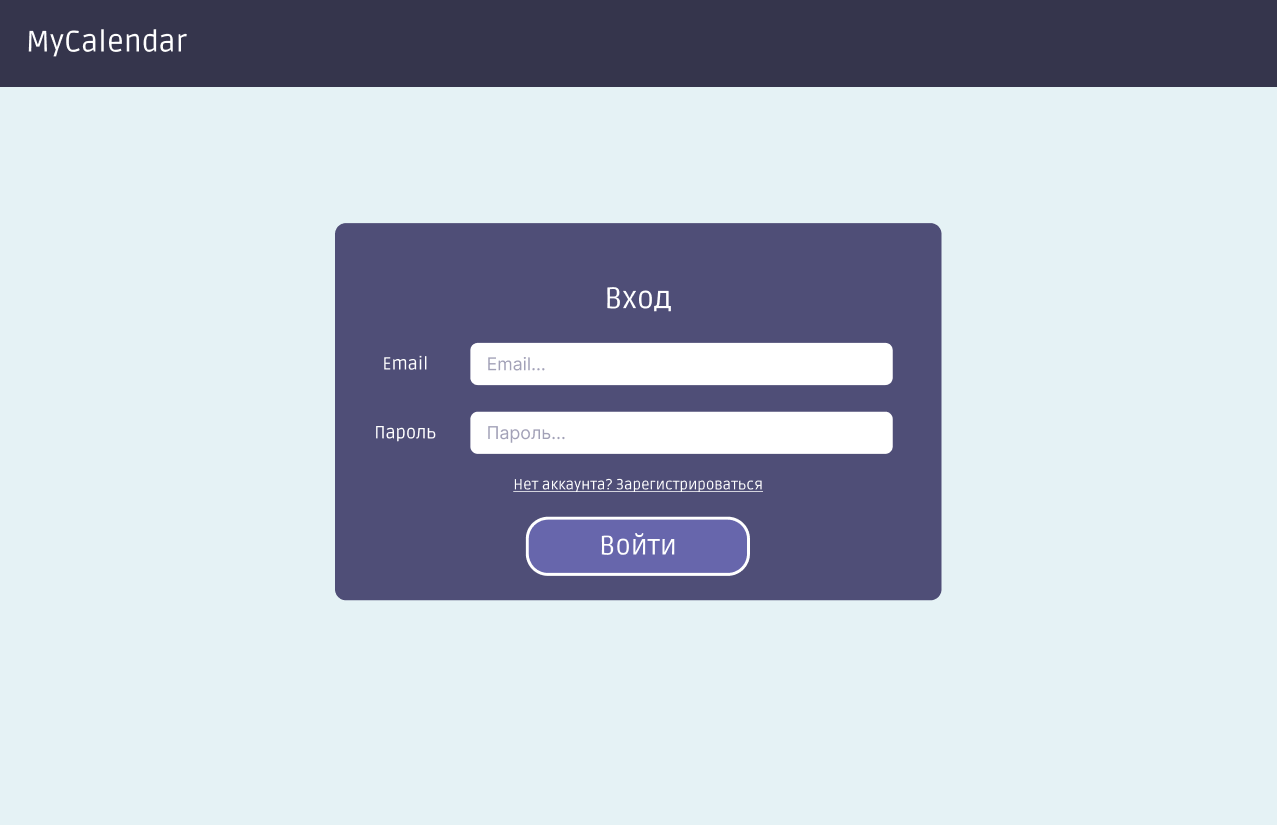
[](https://github.com/ArbakovaAnastasia/AccountingForCommunications/raw/main/%D0%9F%D1%80%D0%BE%D0%B5%D0%BA%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5/%D0%92%D1%85%D0%BE%D0%B4.png)

Рисунок 12 – Макет экрана «Вход»

* 1. Регистрация (Register) (см. рисунок 13)

Отображает: Страницу регистрации аккаунта пользователя, требуется ввести электронную почту, пароль и имя пользователя.

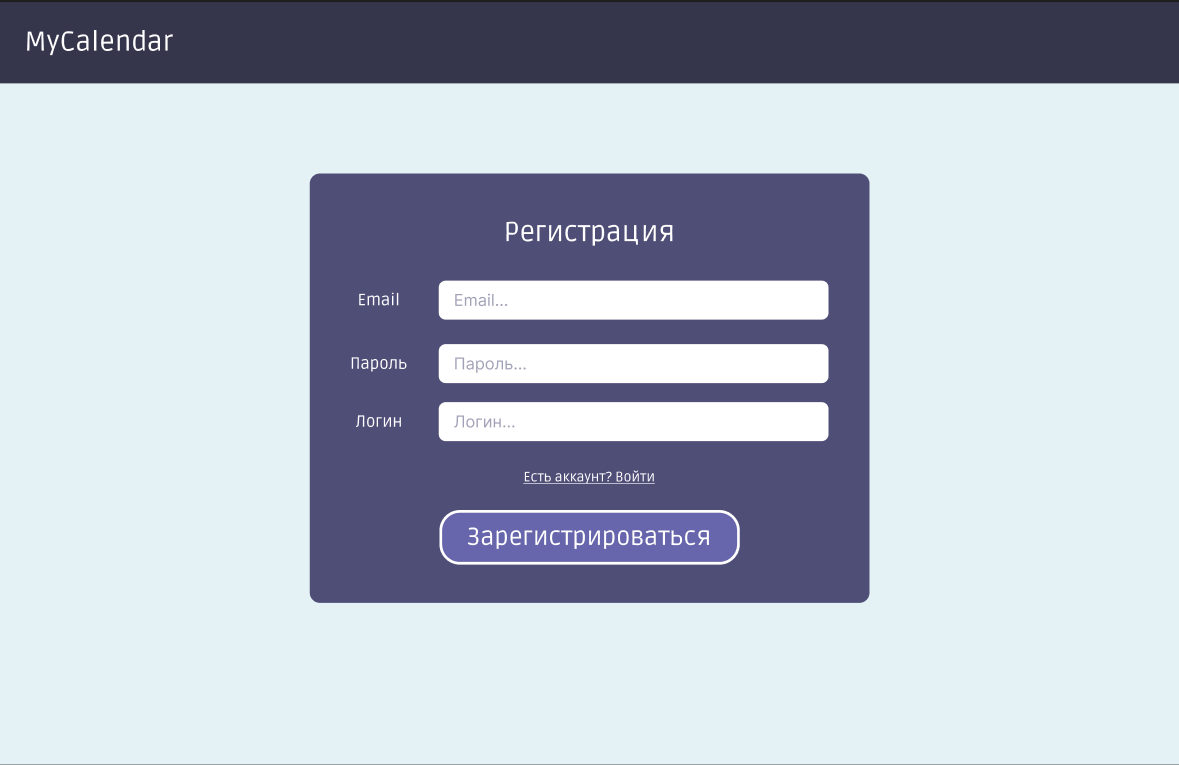
[](https://github.com/ArbakovaAnastasia/AccountingForCommunications/raw/main/%D0%9F%D1%80%D0%BE%D0%B5%D0%BA%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5/%D0%A0%D0%B5%D0%B3%D0%B8%D1%81%D1%82%D1%80%D0%B0%D1%86%D0%B8%D1%8F.png)

Рисунок 13 – Макет экрана «Регистрация»

* 1. Профиль (Profile) (см. рисунок 14)

Отображает: Информацию об аккаунте пользователя.

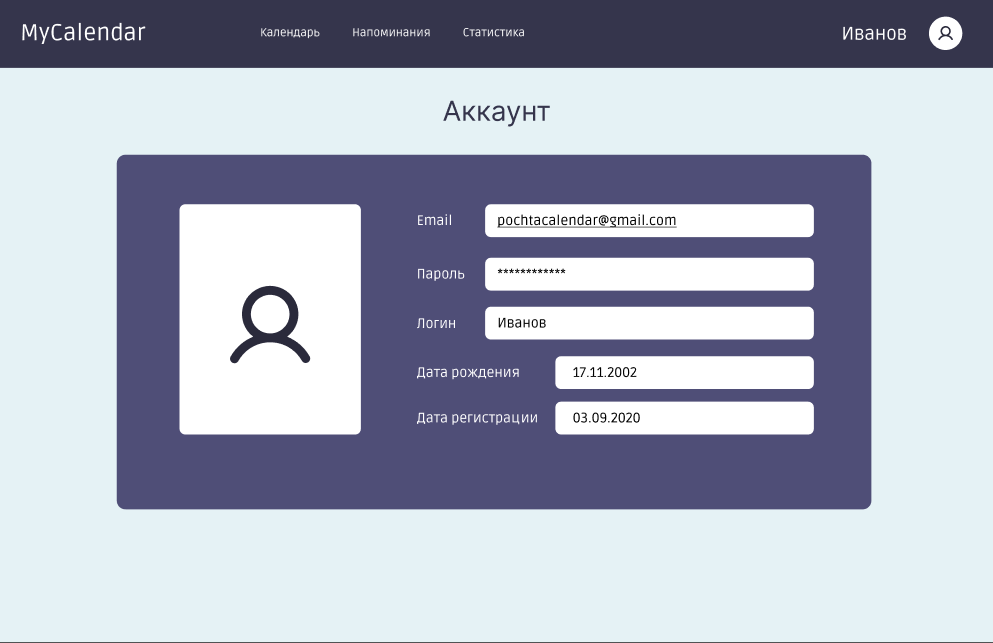
[](https://github.com/ArbakovaAnastasia/AccountingForCommunications/raw/main/%D0%9F%D1%80%D0%BE%D0%B5%D0%BA%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5/%D0%9F%D1%80%D0%BE%D1%84%D0%B8%D0%BB%D1%8C.png)

Рисунок 14 – Макет экрана «Профиль»

* 1. Редактирование профиля (EditProfile) (см. рисунок 15)

Отображает: Страницу редактирования аккаунта пользователя, можно изменить имя пользователя или дату рождения.

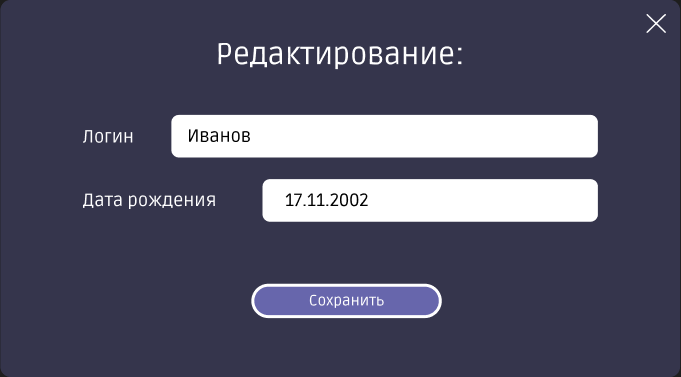


Рисунок 15 – Макет экрана «Редактирование»

2.1 Календарь (Calendar) (см. рисунок 16)

Отображает: Информацию о текущем месяце календаря с соответствующими коммуникациями пользователя.

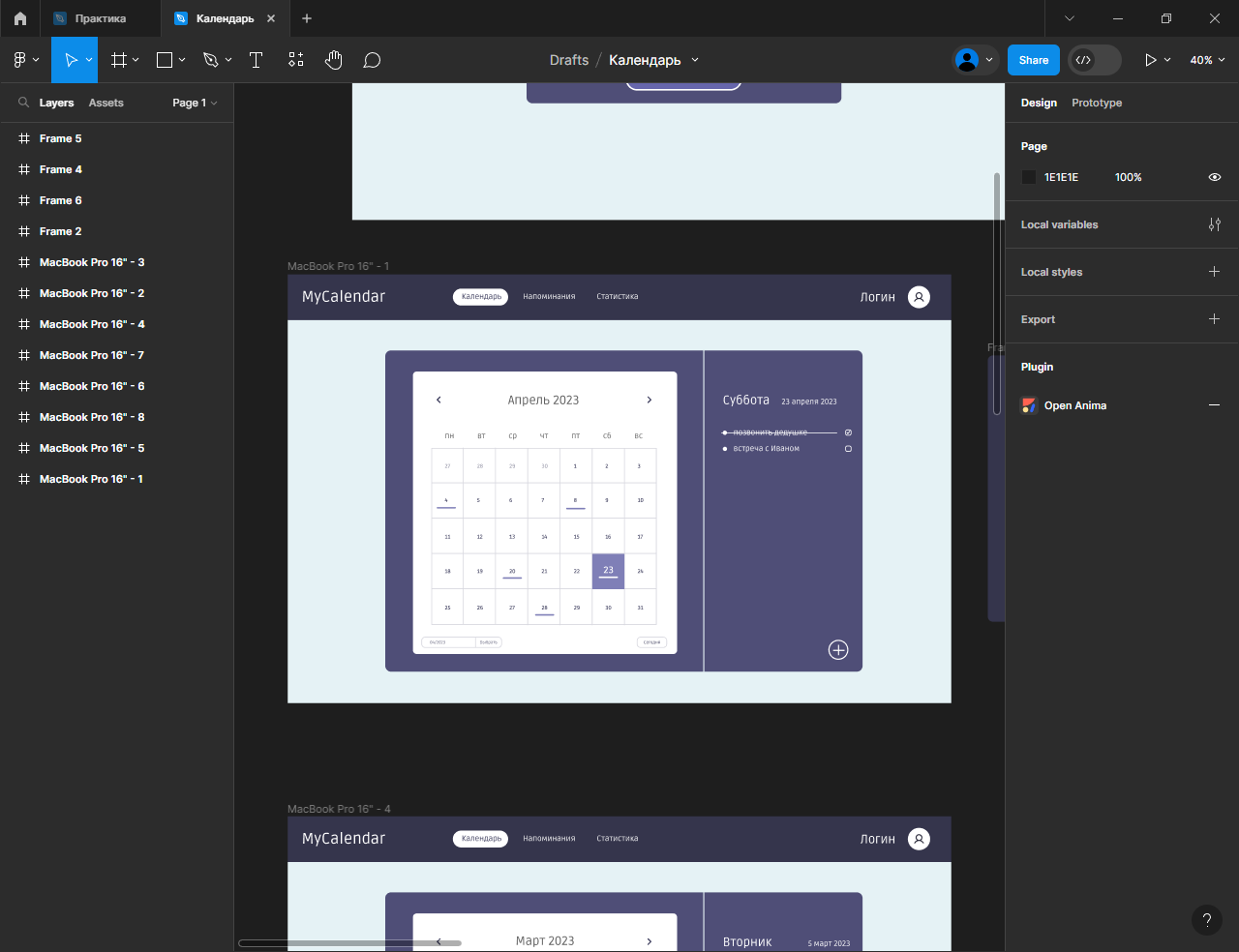


Рисунок 16 – Макет экрана «Календарь»

2.1.1 Форма ввода (Input\_Form) (см. рисунок 17)

Отображает: Форму ввода данных о коммуникации, требуется ввести название, запланированную дату, тип, человек, с которым коммуникация, описание.

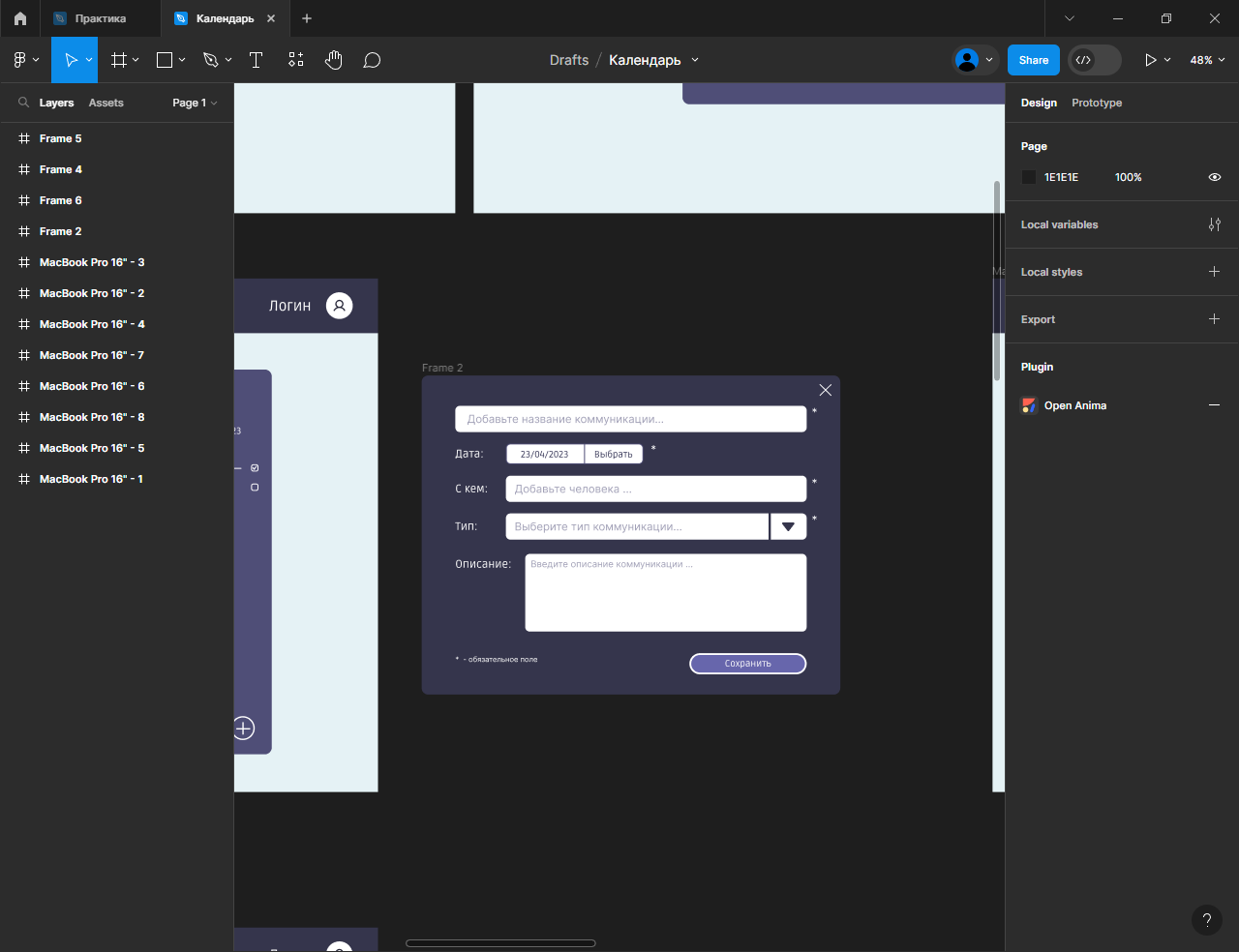


Рисунок 17 – Макет экрана «Форма ввода»

2.2 Напоминания (Reminder) (см. рисунок 18)

Отображает: Информацию о напоминаемых коммуникациях.

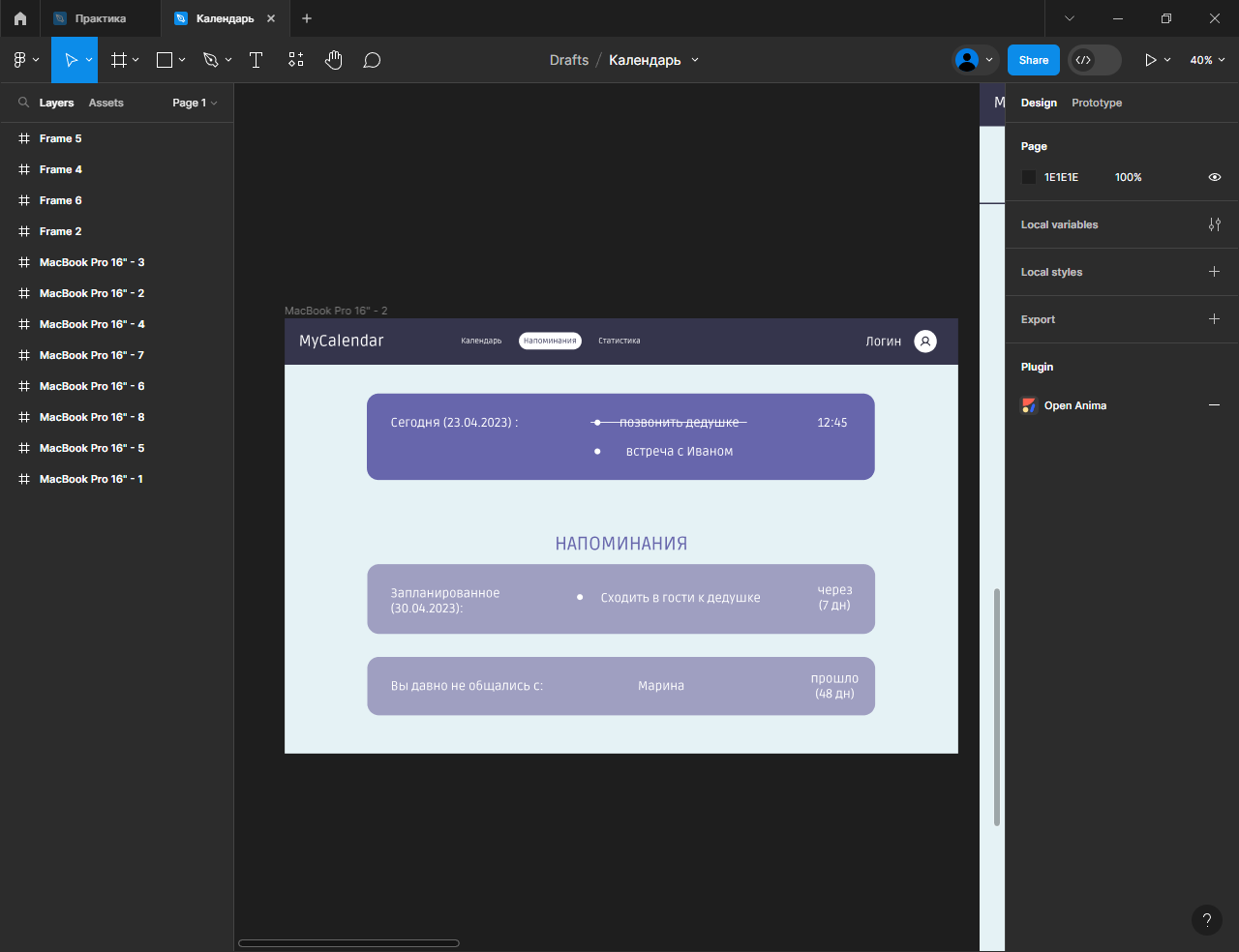


Рисунок 18 – Макет экрана «Напоминания»

2.3 Статистика (Statistics) (см. рисунках 19 и 20)

Отображает: Статистику по коммуникациям пользователя.

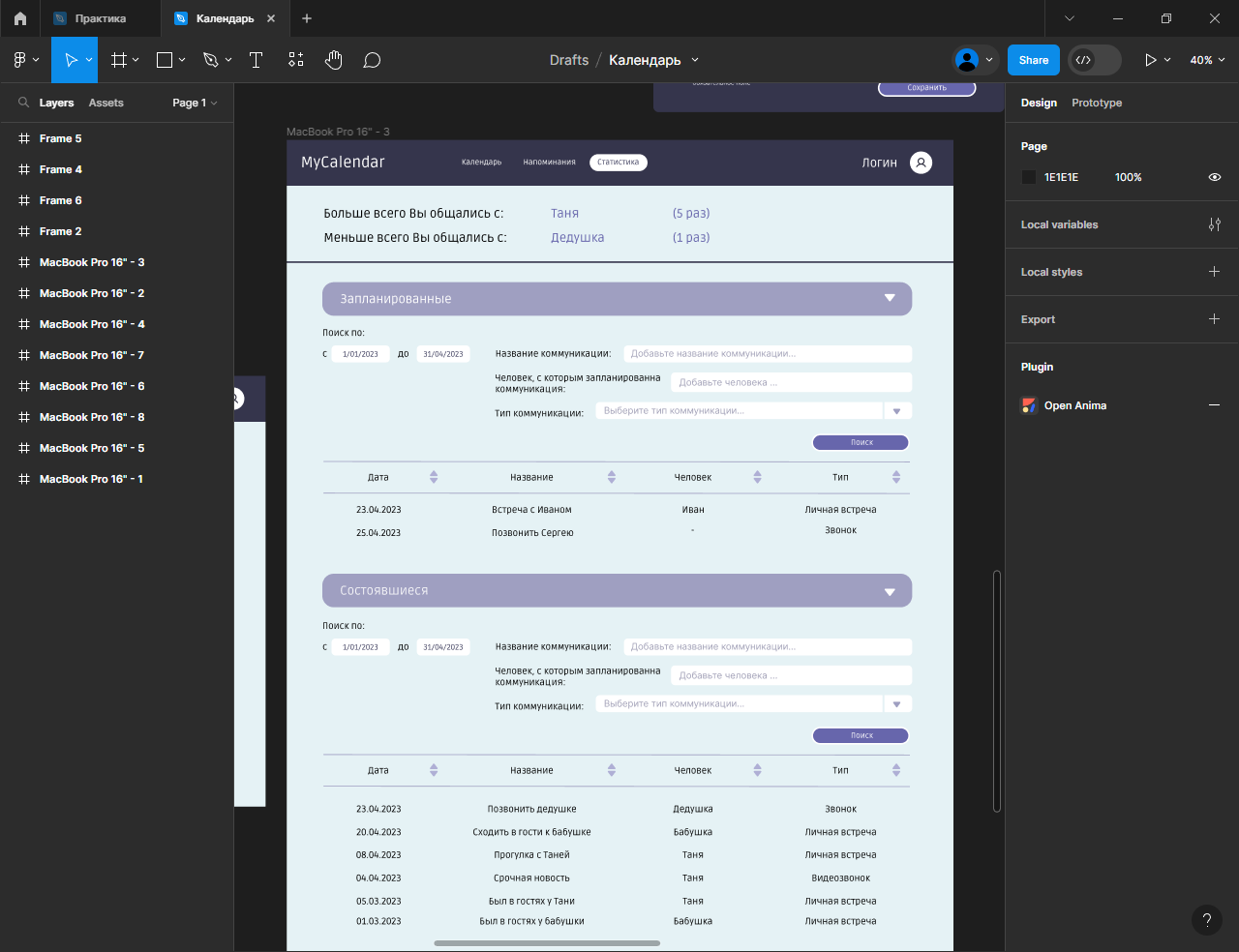


Рисунок 19 – Макет экрана «Статистика» (часть 1)

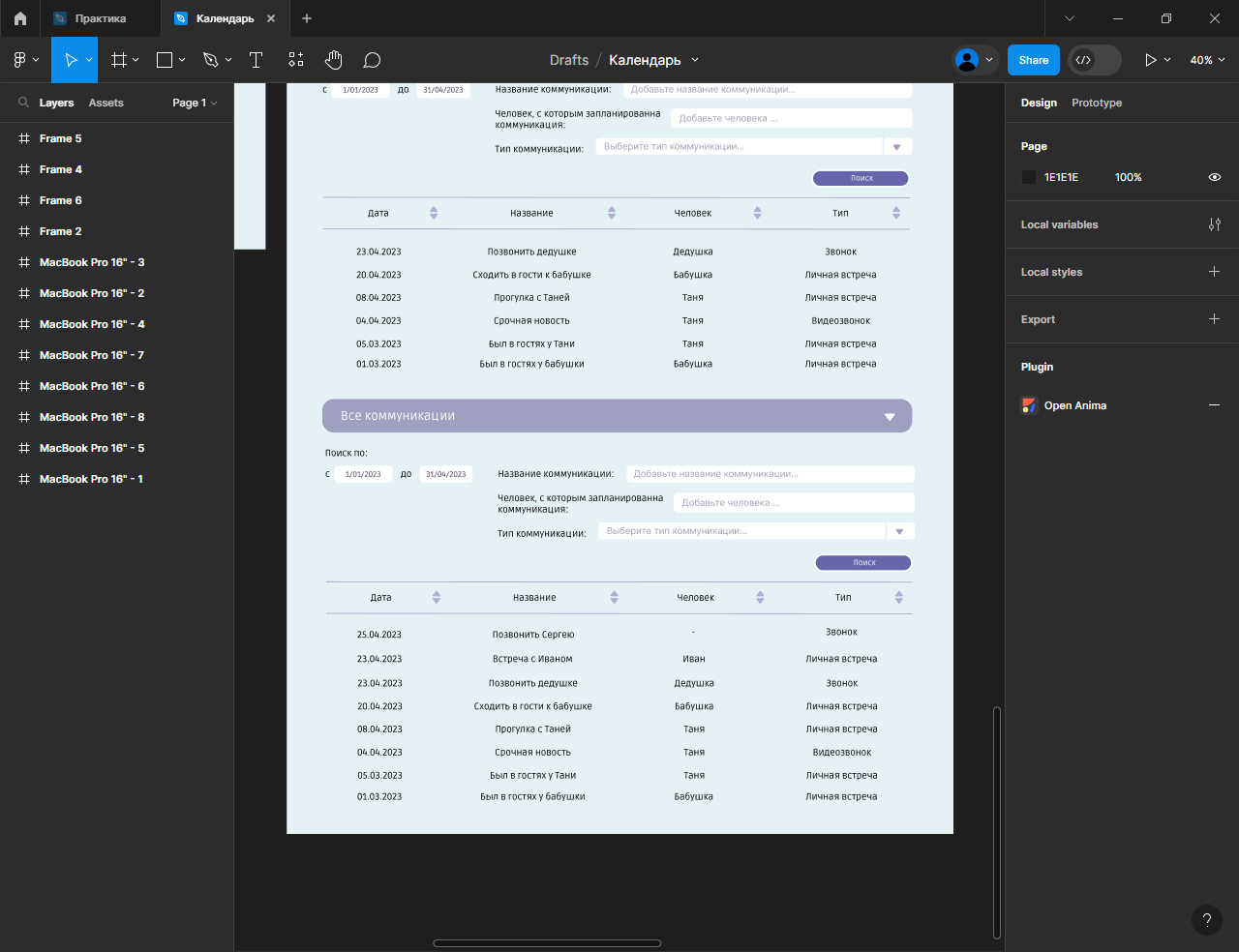


Рисунок 20 – Макет экрана «Статистика» (часть 2)

Карта экранов – это визуальное представление всех экранов в приложении или системе, упорядоченных и связанных друг с другом. Она помогает разработчикам понять, как различные экраны взаимодействуют друг с другом, какие данные передаются между ними и какие функции доступны на каждом экране. Карта экранов также может использоваться для создания прототипов пользовательского интерфейса и тестирования различных сценариев использования приложения.

Карта экранов представлена на рисунке 21.

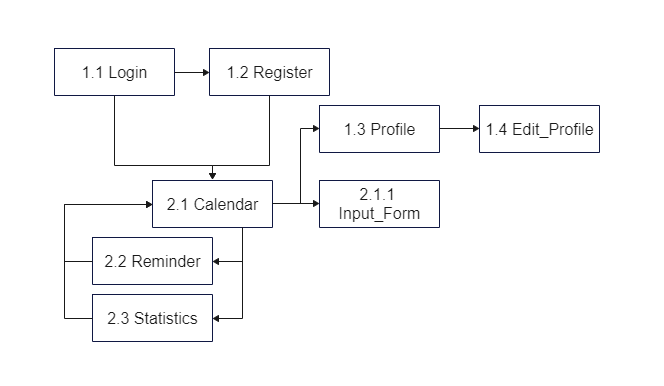
[](https://github.com/ArbakovaAnastasia/AccountingForCommunications/raw/main/%D0%9F%D1%80%D0%BE%D0%B5%D0%BA%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5/%D0%BA%D0%B0%D1%80%D1%82%D0%B0%20%D1%8D%D0%BA%D1%80%D0%B0%D0%BD%D0%BE%D0%B2.png)

Рисунок 21 – Карта экранов

На карте экранов изображены переходы из экрана входа на экран регистрации и на экран календаря.

Из экрана регистрации также имеется переход на экран календаря.

Из экрана календаря имеются переходы на экраны формы ввода коммуникации, профиля, напоминаний и статистики.

Из формы профиля имеется переход на экран редактирования профиля.

Из экранов напоминаний и статистики можно вернуться на экран календаря.

1. **Разработка**

Этап разработки включает в себя создание ряда ключевых страниц, необходимых для полноценного функционирования современного веб-ресурса. Были созданы такие страницы, как: вход, регистрация, профиль пользователя, календарь, статистика и напоминания.

Начальными страницами из них являются страницы входа и регистрации, которые обеспечивают пользователям возможность доступа к основному функционалу сайта. Страница профиля предоставляет инструменты для настройки личного кабинета и управления персональными данными. Календарь помогает организовывать мероприятия и события, а также следить за их проведением. Статистика и напоминания позволяют анализировать активность пользователей и своевременно уведомлять их о важных событиях.

На рисунке 22 представлена страница входа, где пользователю нужно ввести данные электронной почты и пароль, и после нажать кнопку «Войти». При отсутствии аккаунта у пользователя, он может зарегистрироваться.

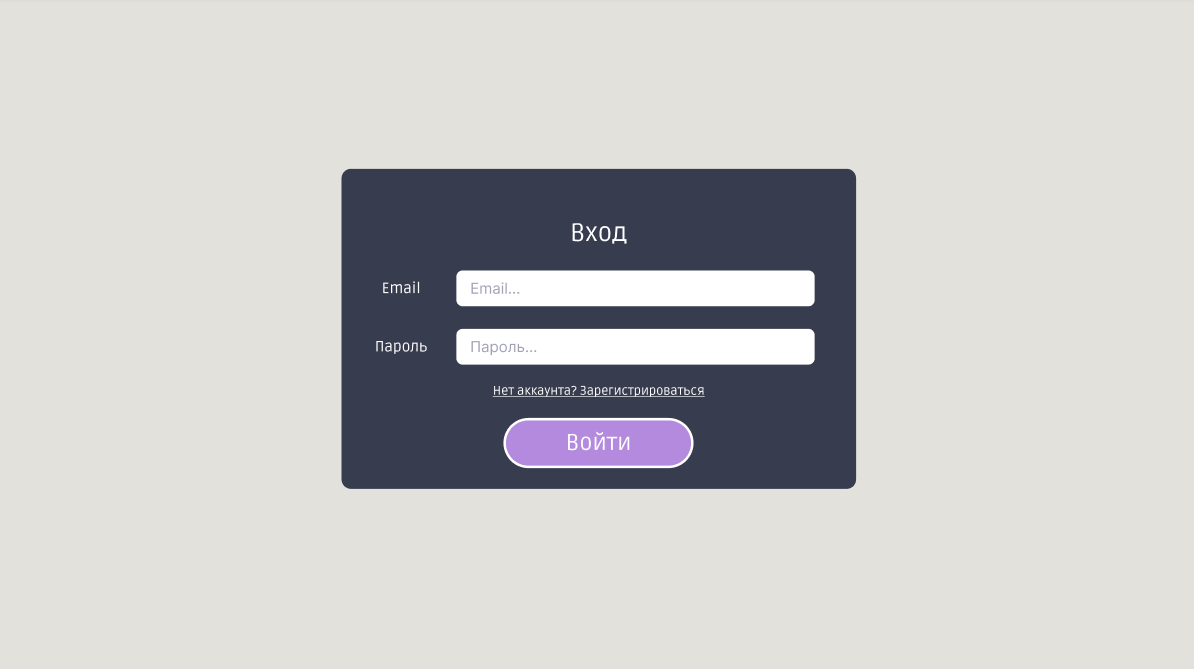


Рисунок 22 – Страница входа

Со страницы входа можно перейти на страницу регистрации, изображенной на рисунке 23.

Регистрация подразумевает под собой создание нового аккаунта пользователя, для чего нужно ввести электронную почту, придумать пароль и логин, и нажать кнопку «Зарегистрироваться». При наличии аккаунта у пользователя, он может перейти по ссылке и войти в аккаунт.

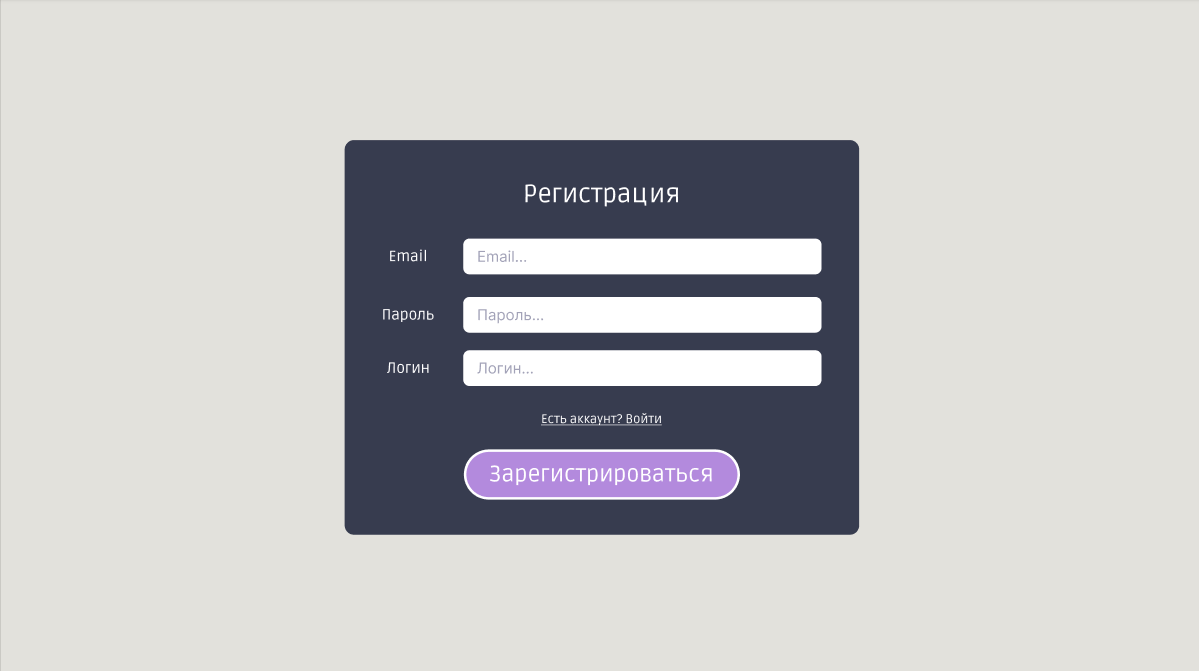


Рисунок 23 – Страница регистрации

При удачной авторизации для пользователя откроется окно календаря. В верхнем меню (см. рисунок 24), содержащее разделы «Календарь», «Статистика», «Напоминания» и «Профиль», пользователь может перейти во вкладку «Пользователь» (см. рисунок 25), в котором можно увидеть данные пользователя: электронную почту, пароль, логин, дата рождения и дату регистрации.

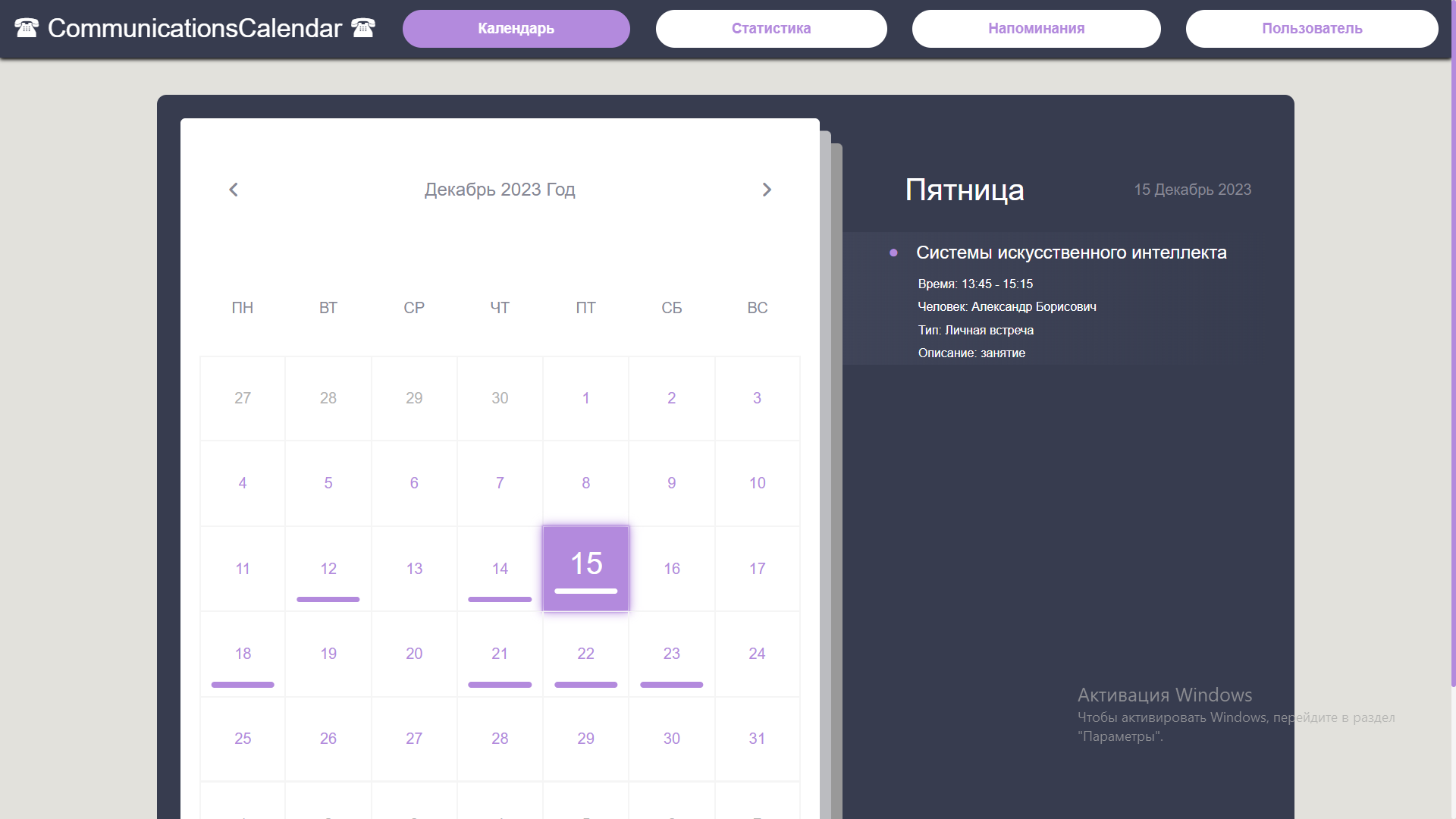


Рисунок 24 – Меню



Рисунок 25 – Страница профиля

Главной страницей системы является раздел «Календарь», представленный на рисунке 26. На данной странице можно увидеть интерактивный календарь, показывающий текущий месяц, год и день. Текущий день выделен фиолетовый цветом с увеличенным шрифтом числа.

В правой части показан соответствующий день недели и полная дата, и соответствующие записи о коммуникациях при наличии. При наличии записей на определённый день на календаре отображается полоска фиолетового или белого цвета, под числом дня.

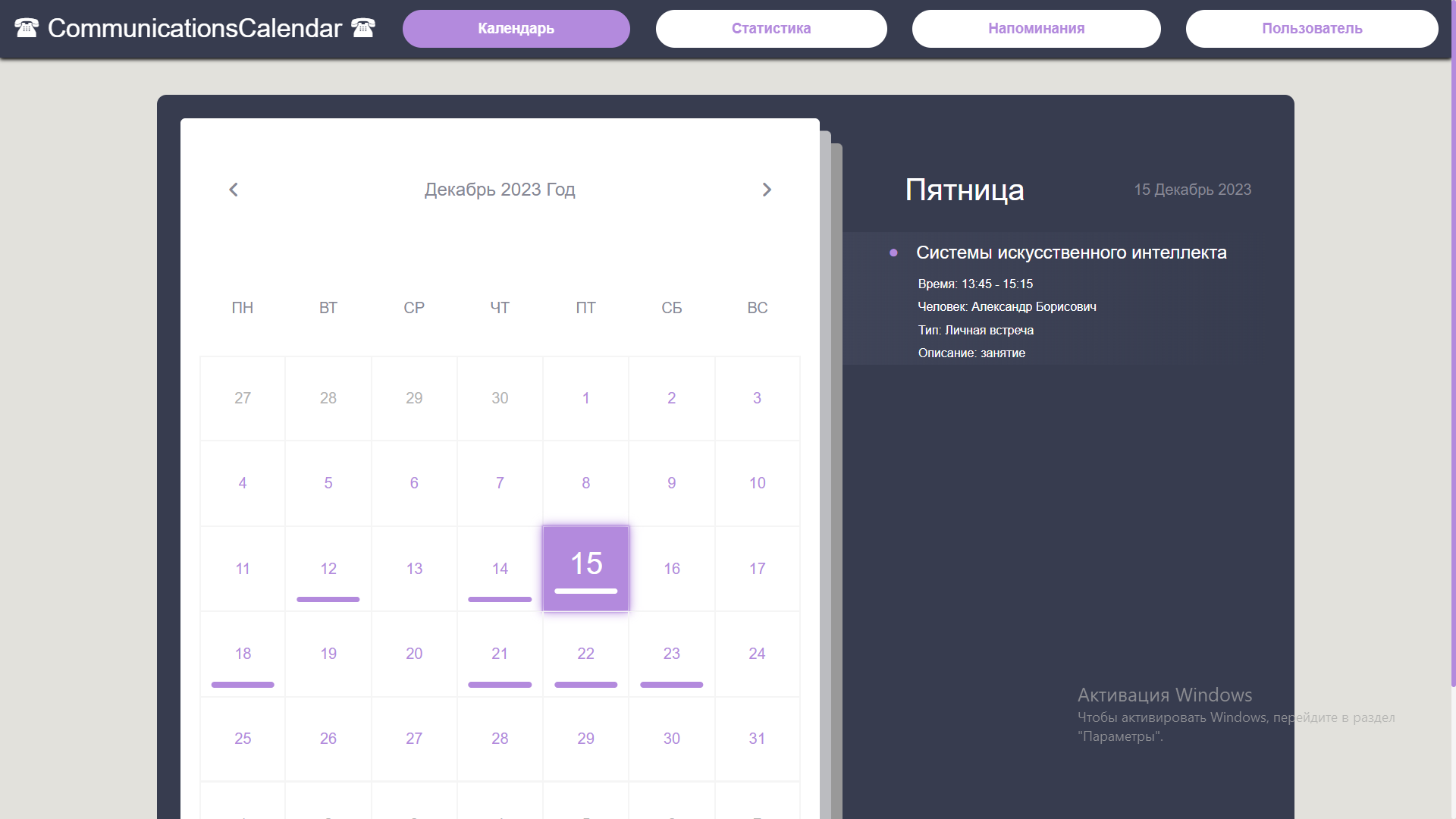


Рисунок 26 – Страница календаря

Для добавления новой коммуникации имеется кнопка со знаком «+» (см. рисунок 27).

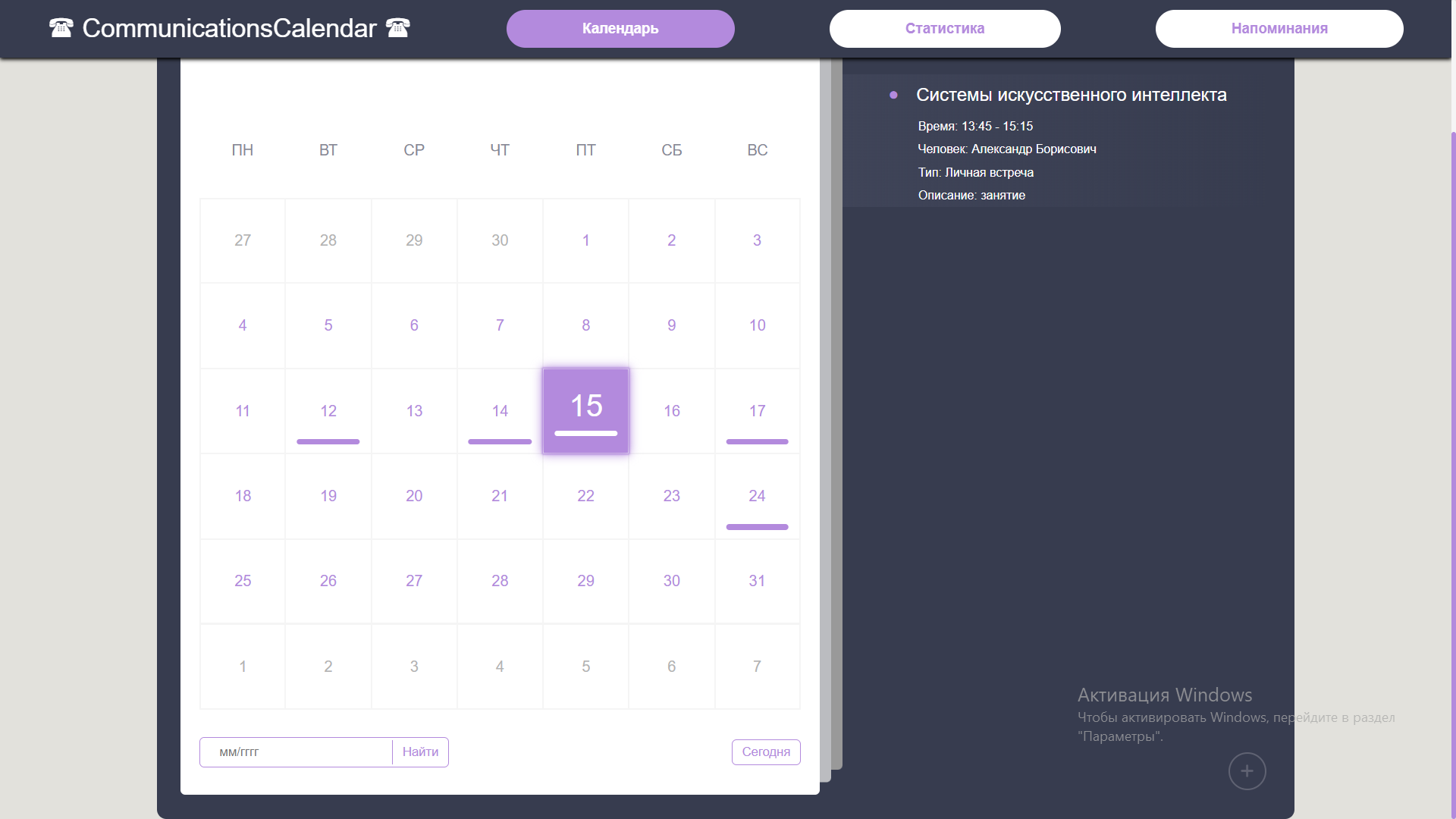


Рисунок 27 – Кнопка добавления новой записи

Для перехода на следующий и предыдущий месяц имеются соответственно кнопки с символами «<» и «>» (см. рисунок 28).



Рисунок 28 – Кнопки для перехода на предыдущий и следующий месяц

Для поиска определенного месяца за какой-либо год имеется поле ввода информации и кнопка «Найти» (см. рисунок 29).

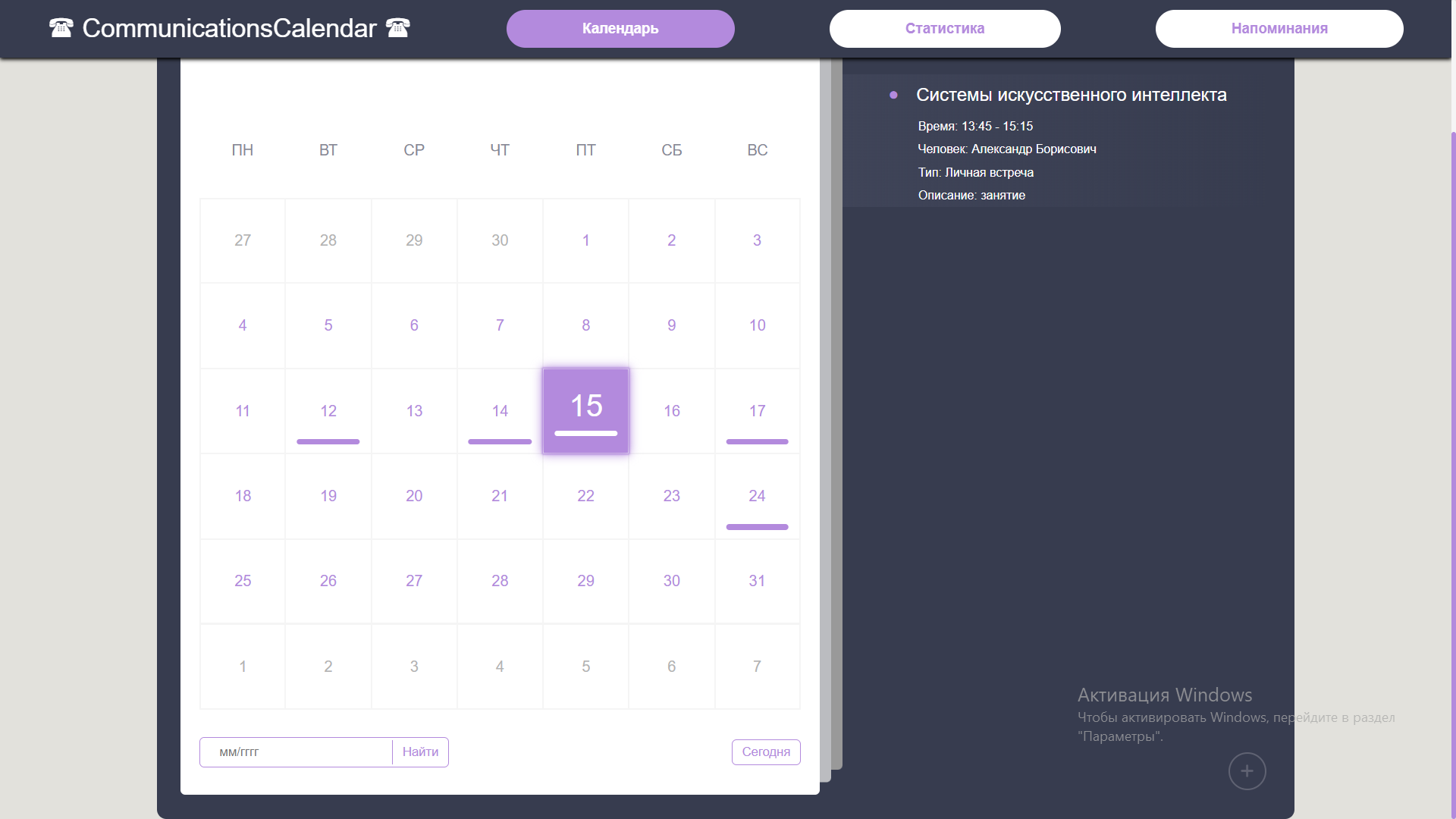


Рисунок 29 – Поиск определенного месяца

При наведении на запись о коммуникации меняется цвет (см. рисунок 30). Коммуникация содержит название, время, человека, с которым коммуникация, тип коммуникации и описание.



Рисунок 30 – Запись о коммуникации

При нажимании на кнопку добавления новой записи открывается форма для заполнения информации о коммуникации, изображённой на рисунке 31. Форма содержит поля для заполнения: наименование, время, человек, тип и описание.

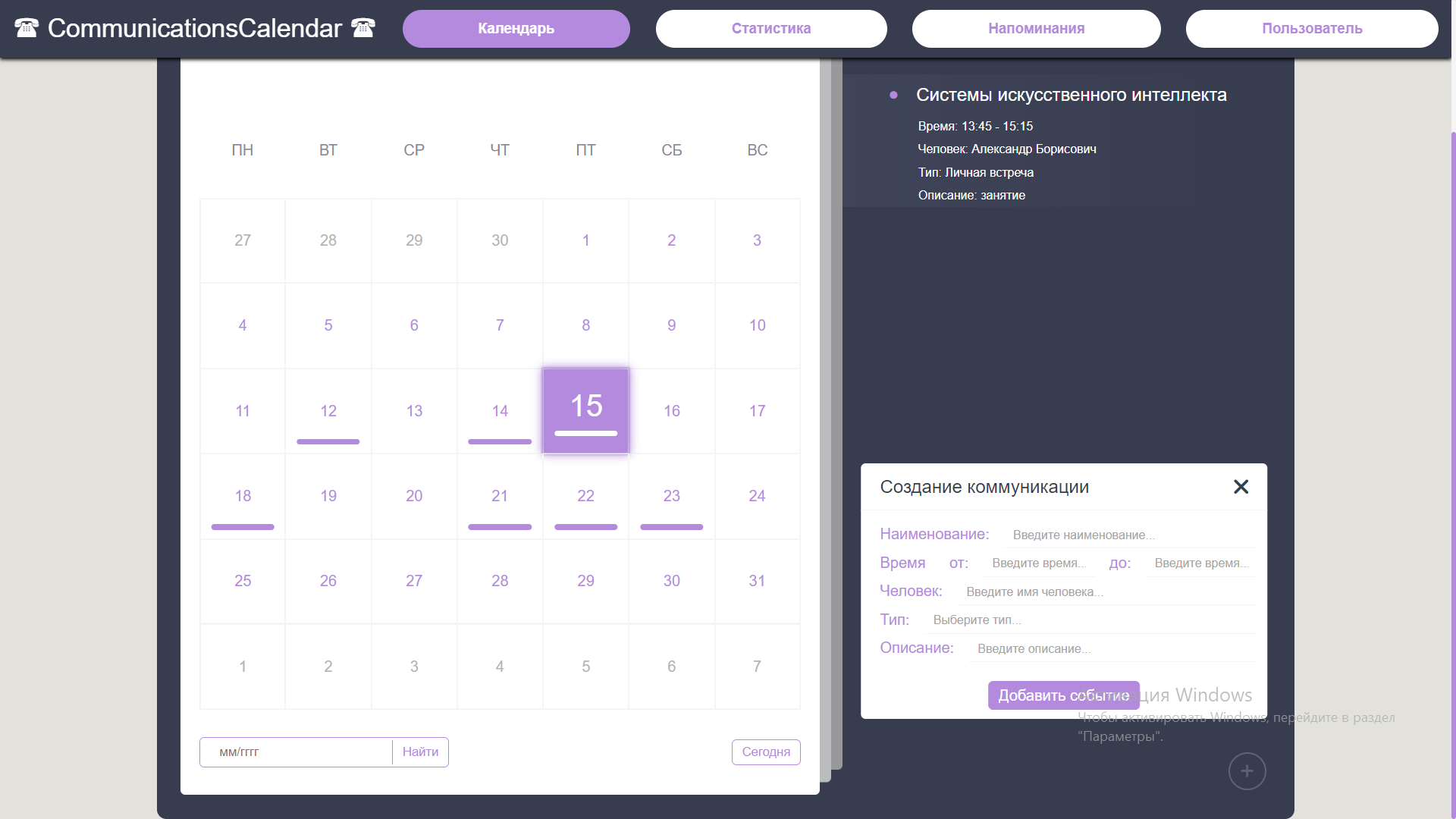


Рисунок 31 – Форма добавления новой записи о коммуникации

При выборе раздела «Статистика», для пользователя откроется окно с запланированными, состоявшимися и всеми коммуникациями (см. рисунок 32).

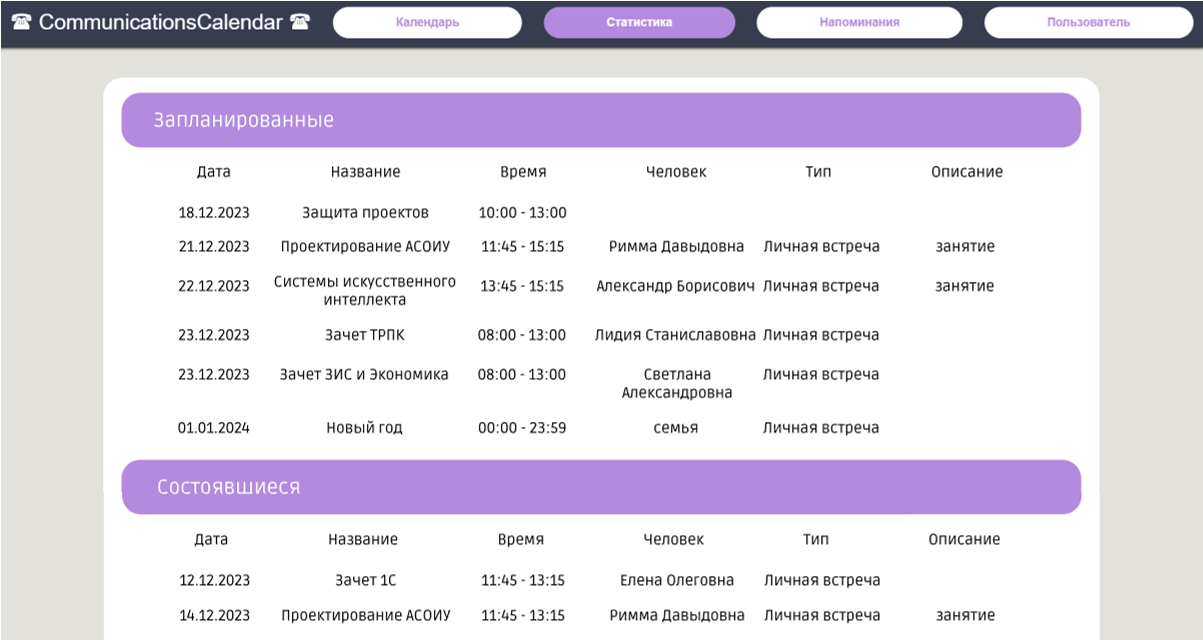


Рисунок 32 – Страница статистики

При выборе раздела «Напоминания», для пользователя откроется окно с запланированными коммуникациями и количеством дней до коммуникации (см. рисунок 33).

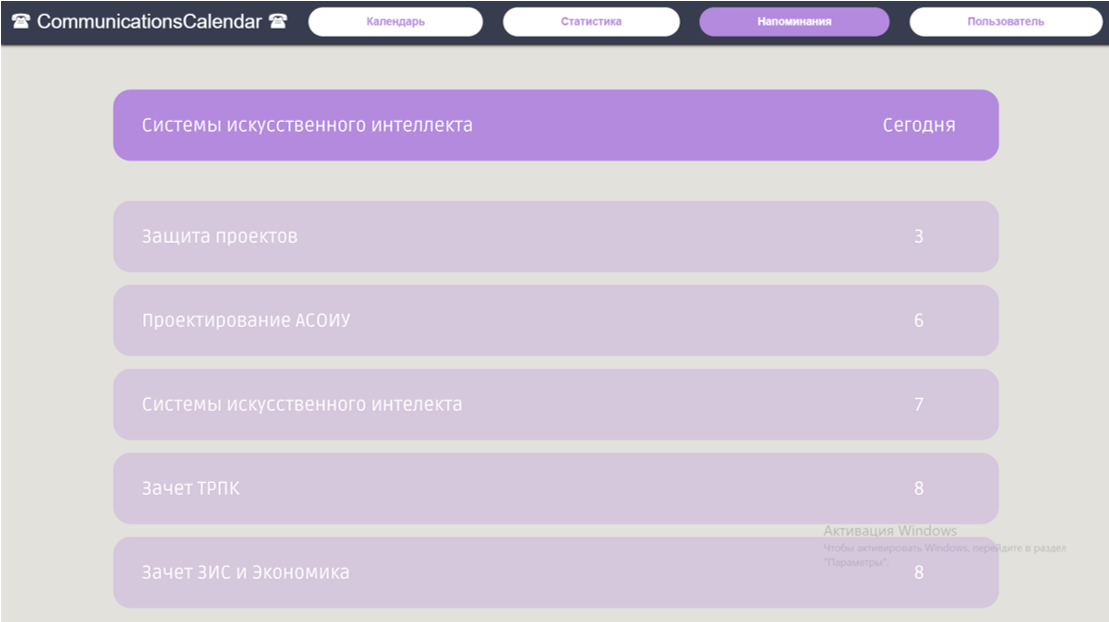


Рисунок 33 – Страница напоминаний

# **Тестирование**

Выберем случайный день для добавления новой коммуникации (см. рисунок 34).

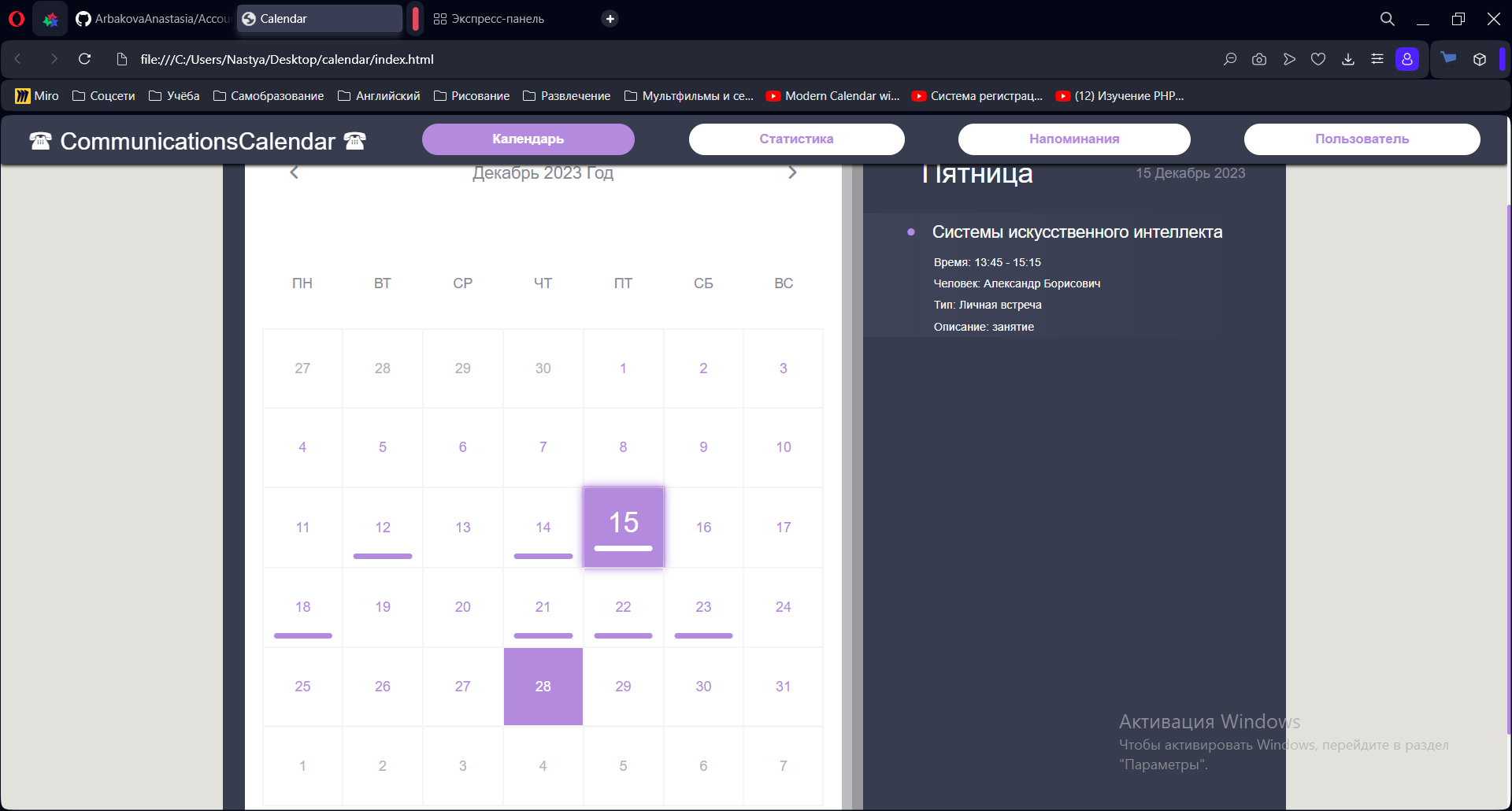


Рисунок 34 – Выбор дня

При нажатии на кнопку добавления новой записи «+», откроется форма для ввода информации о коммуникации. Заполним её, как на рисунке 35. Нажимаем кнопку «Добавить событие».

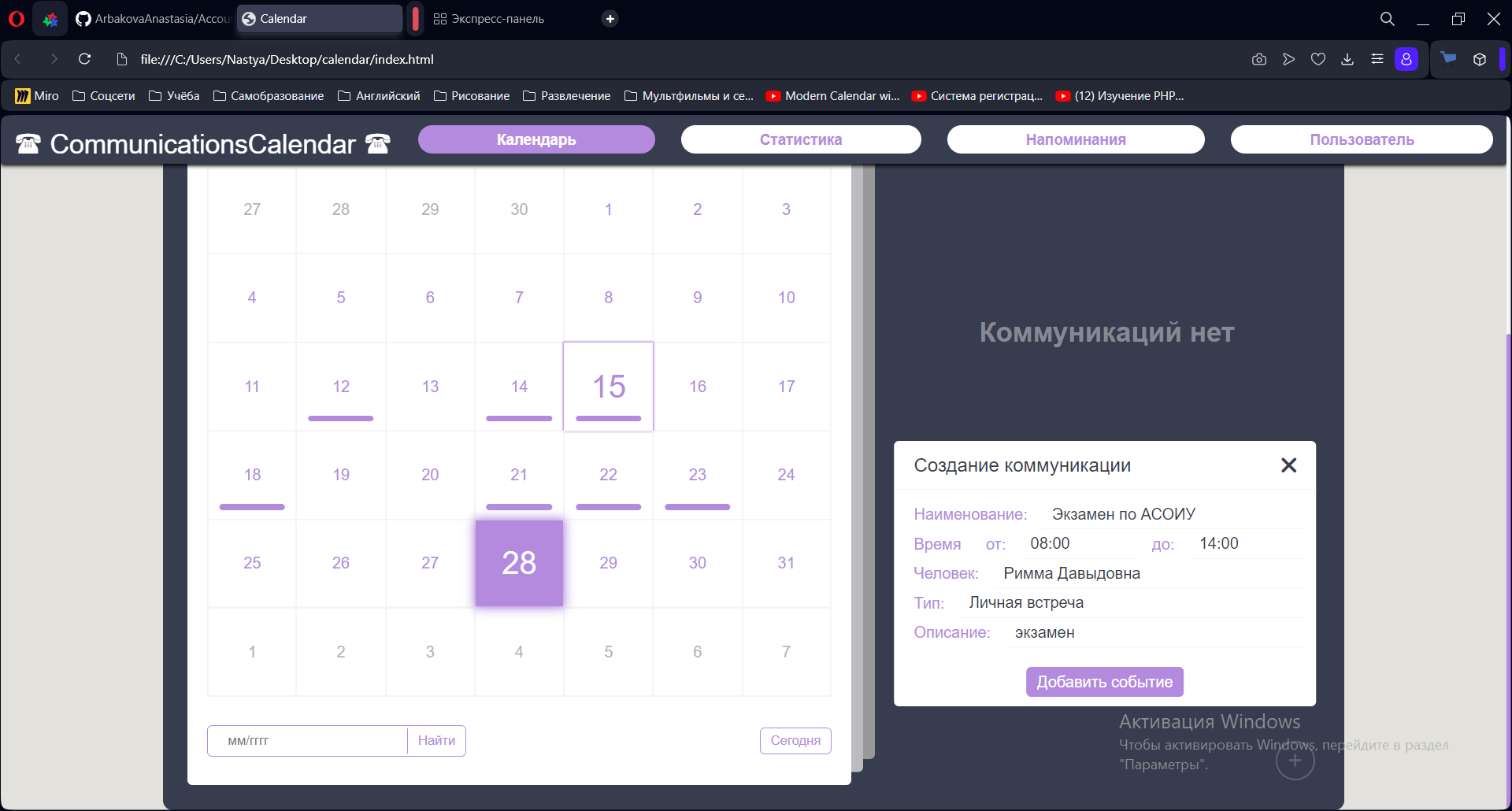


Рисунок 35 – Заполнение формы ввода

При успешном правильном вводе, коммуникация появится в списке коммуникаций (см. рисунок 36).

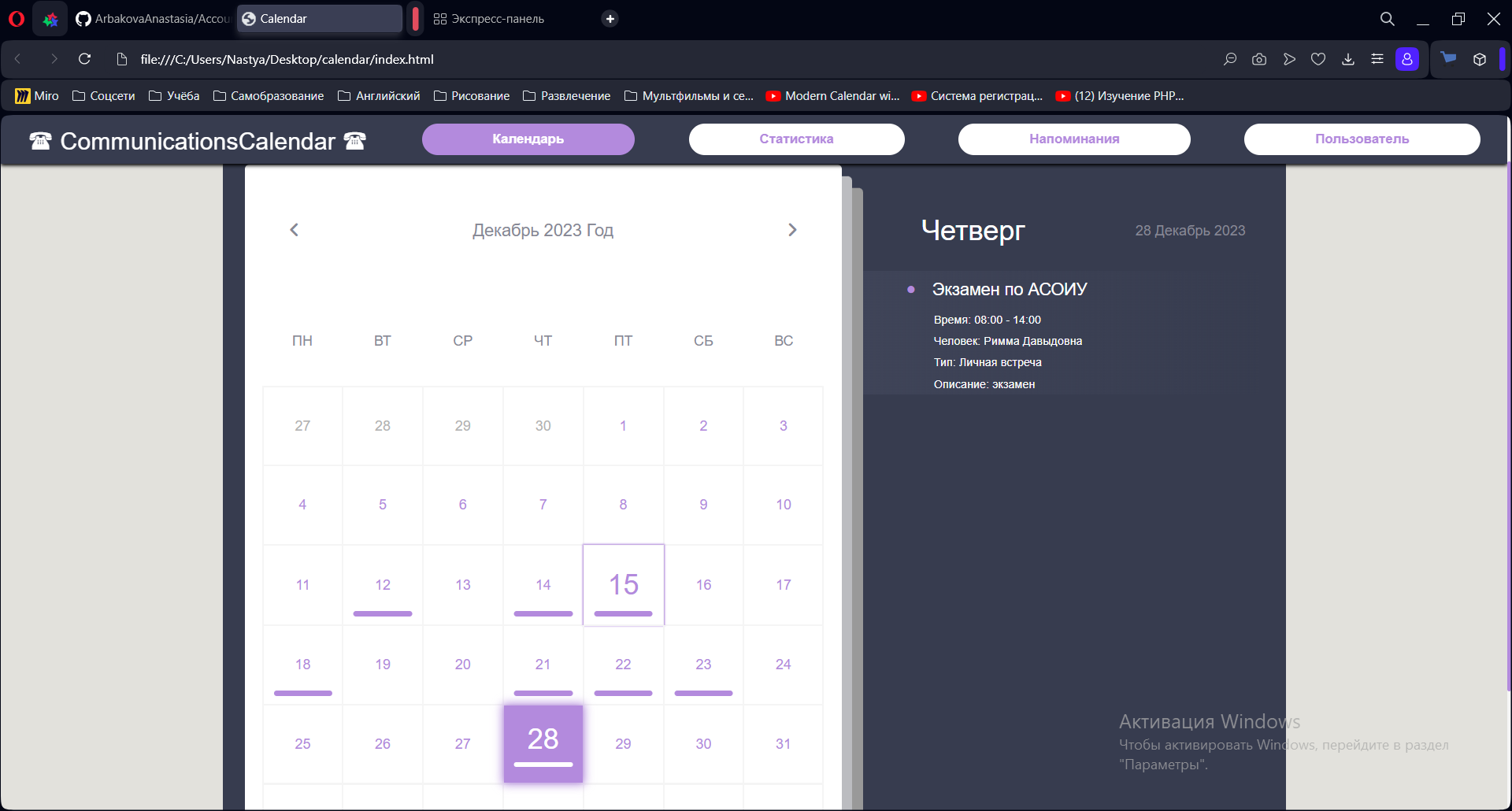


Рисунок 36 – Отображение новой коммуникации в списке

При неверном вводе появится оповещающее окно с сообщением (см. рисунок 37).

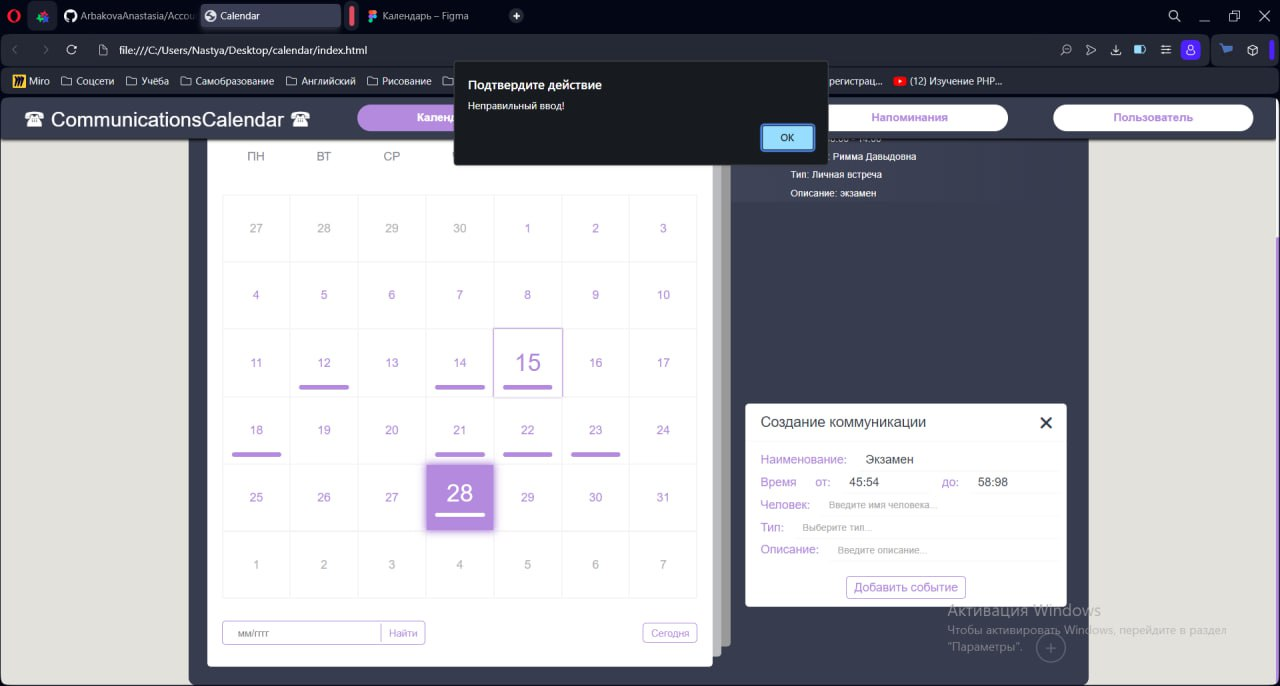


Рисунок 37 – Сообщение о неправильном вводе

На календаре отобразится полоска, обозначающая о наличии коммуникации на данный день (см. рисунок 38).

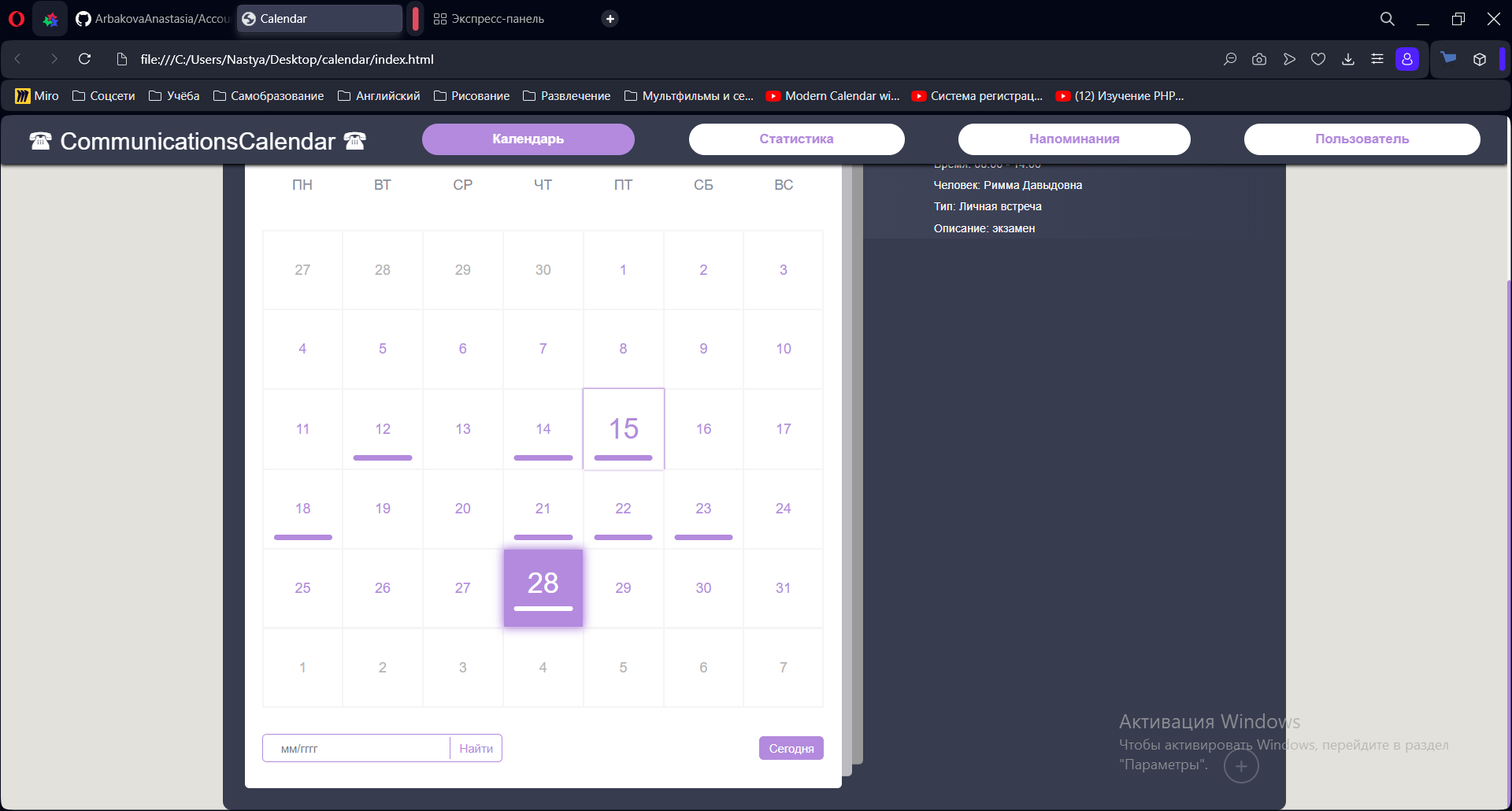


Рисунок 38 – Отображение наличия коммуникации

Во вкладке «Статистика» появилось новая запись (см. рисунок 39).

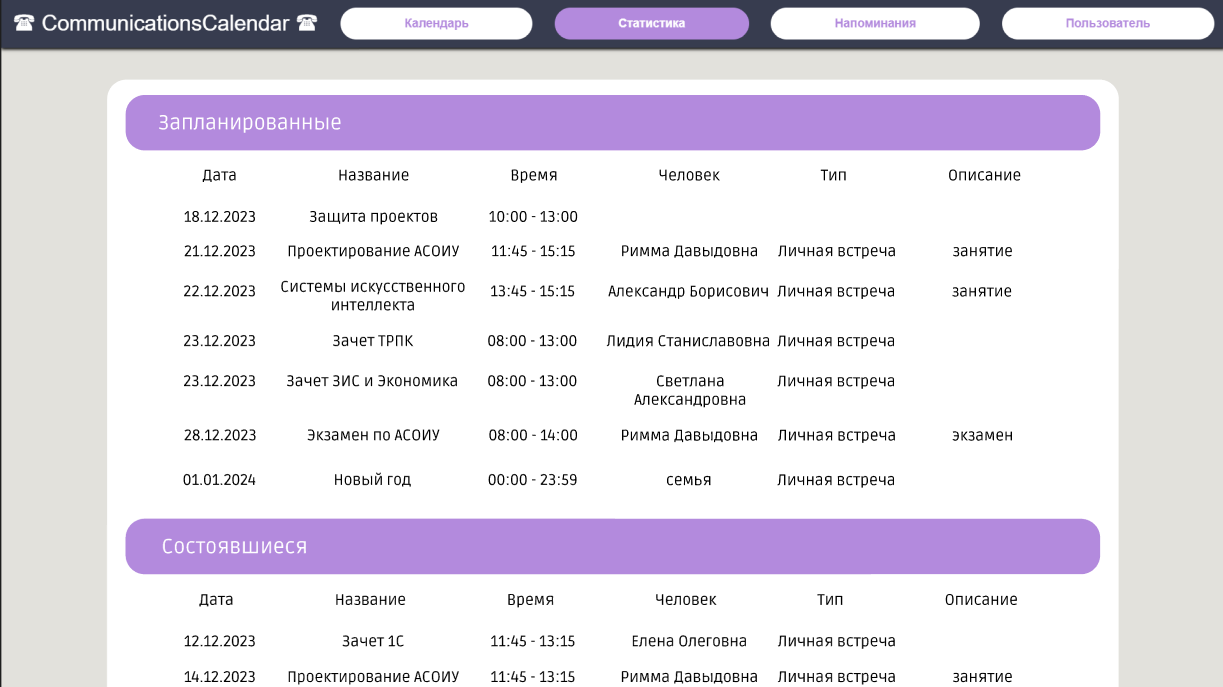


Рисунок 39 – Отображение коммуникации во вкладке «Статистика»

Во вкладке «Напоминания» также появилось новая запись (см. рисунок 40).

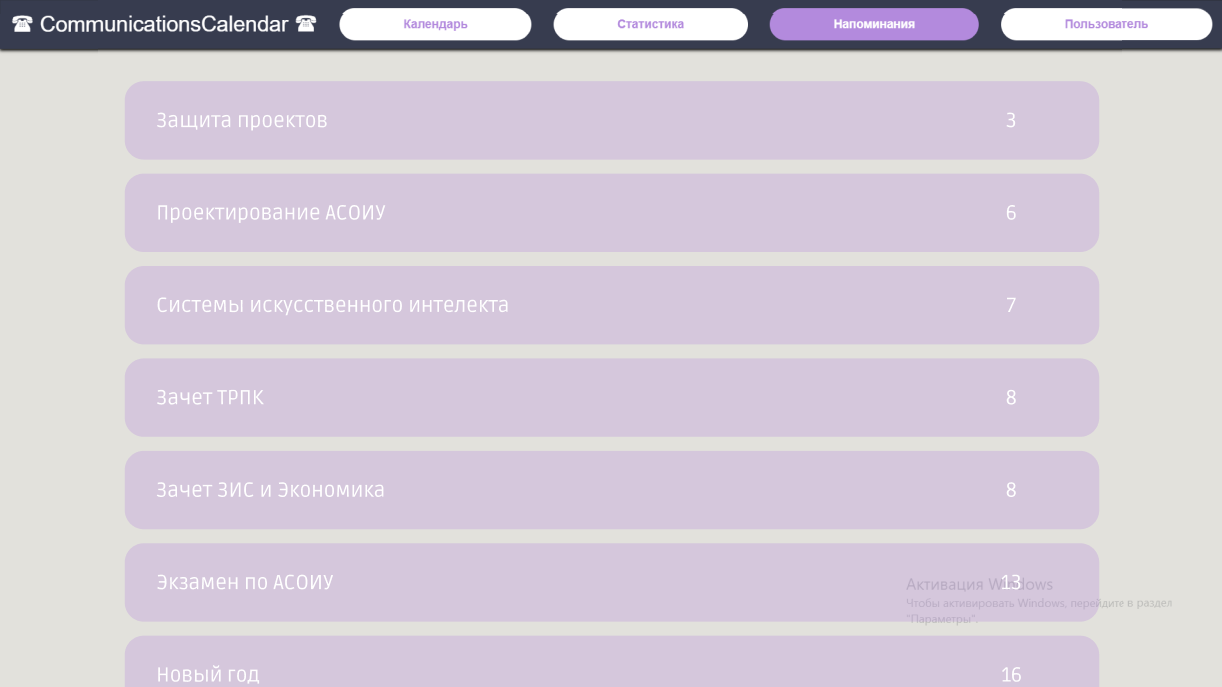


Рисунок 40 – Отображение коммуникации во вкладке «Напоминания»

Проверим поиск конкретного месяца за какой-либо год. Для этого введем в строку поиска месяц и год – «10/2025», и нажмем кнопку «Найти» (см. рисунок 41).

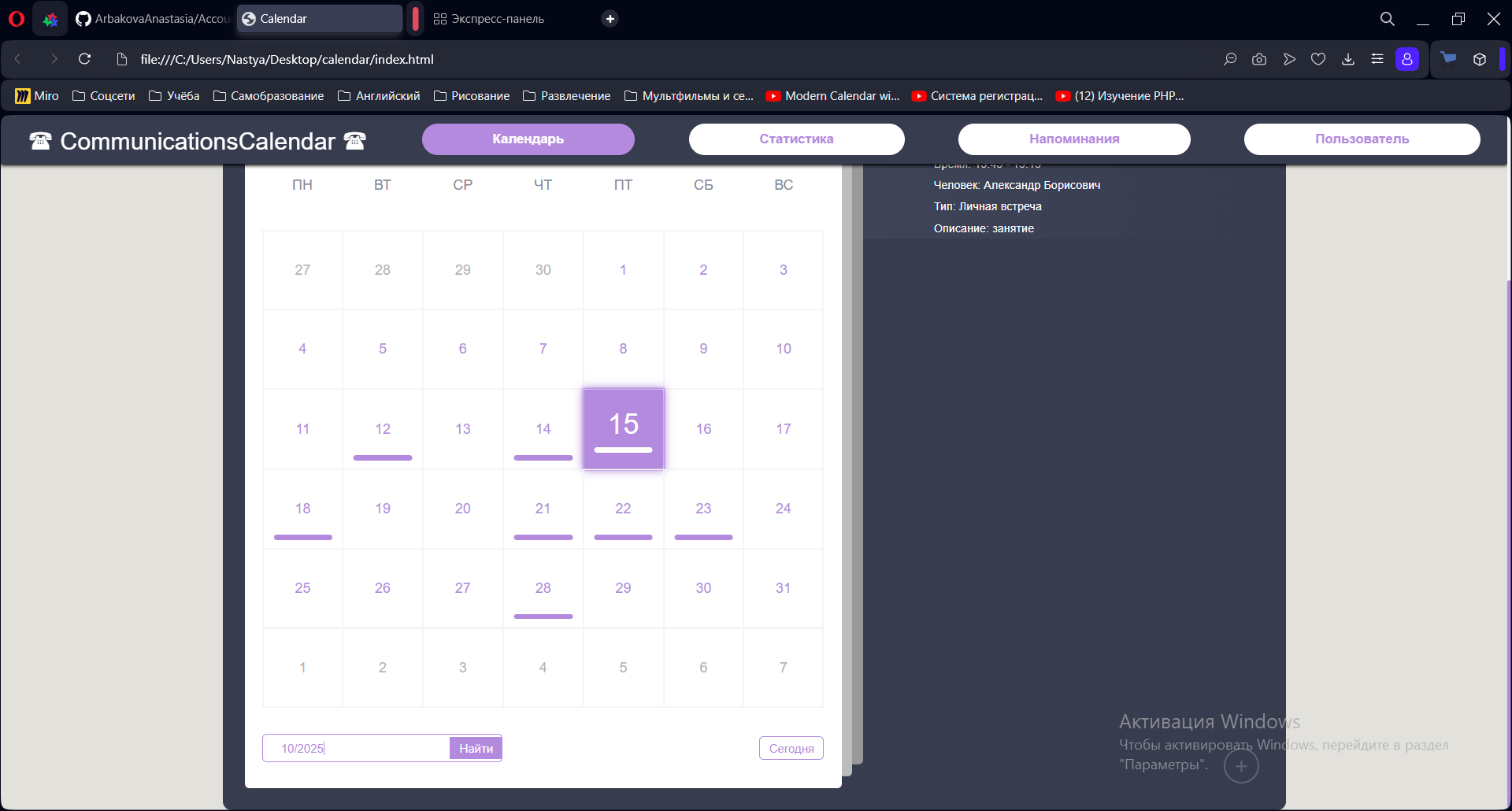


Рисунок 41 – Поиск конкретного месяца

В результате календарь перелистнется на октябрь 2025 года (см. рисунок 42).

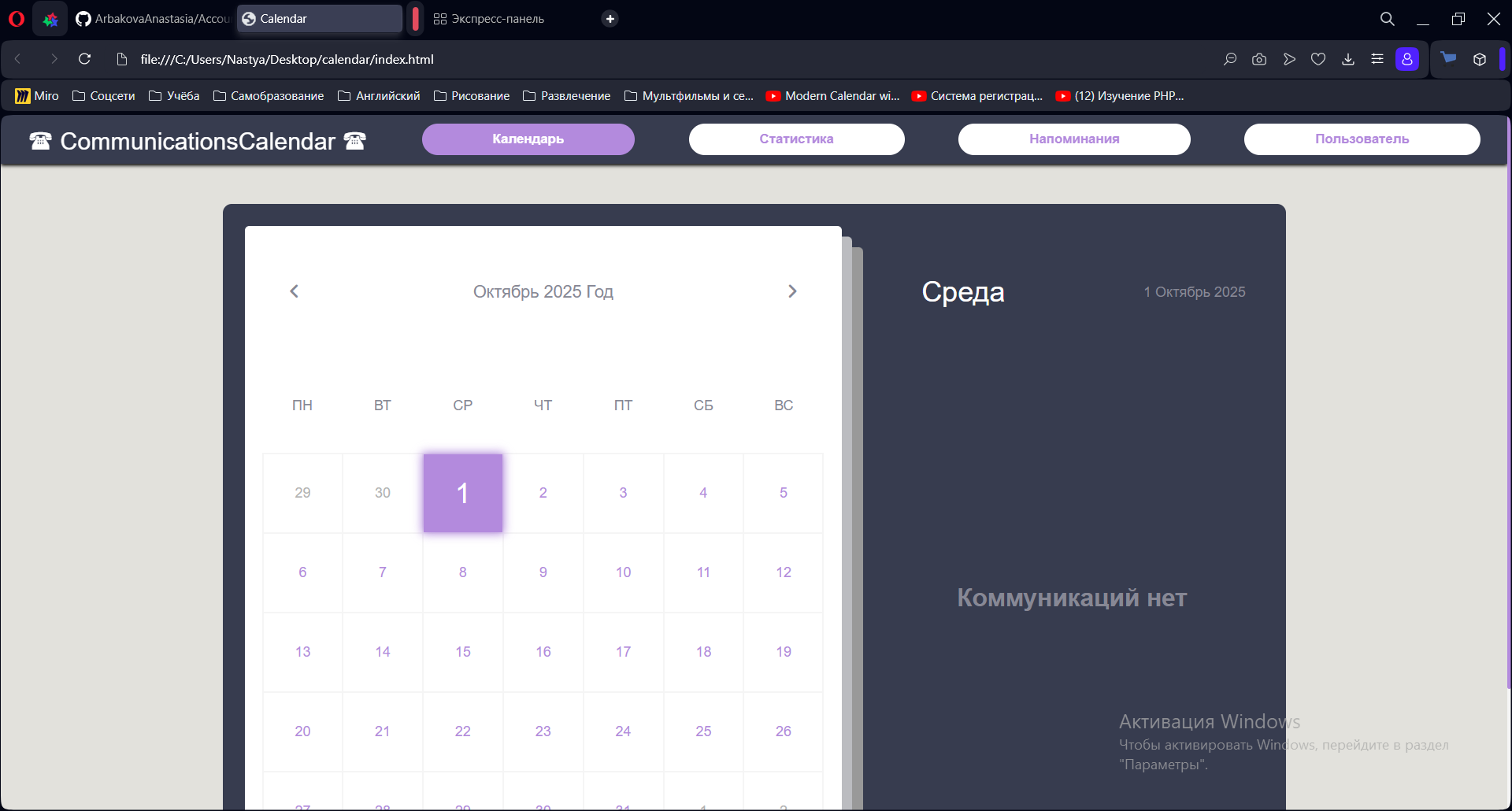


Рисунок 42 – Отображение заданного месяца и года

# **Заключение**

В ходе выполнения курсового проекта было разработано веб-приложение по учету коммуникаций, для которого первостепенно были выявлены цель и задачи проекта, его проблематика и актуальностью, был проведен анализ существующих программных средств, определение их основных функций и недостатков, было проведено предпроектное обследование и проектирование. На этапе проектирования были определены структура приложения и элементы графического интерфейса, с дальнейшим построением макета экранов приложения.

Для разработки приложения были использованы такие средства, как: сервис для разработки интерфейсов и прототипирования Figma; веб-сервис для хостинга IT-проектов GitHub; язык гипертекстовой разметки для структурирования и представления HTML5; язык стилей CSS3; язык программирования Javascript; серверный язык программирования PHP и система управления базами данных MySQL.

Результатом курсового проекта является веб-приложение по учету коммуникаций, который может помочь пользователю организовывать свое время, планировать встречи и события, не забывать о важных датах и мероприятиях. Электронный календарь встреч, в котором ведется учет коммуникаций может выполнять важную роль в жизни пользователя, его работы и личной жизни. Он позволяет хранить информацию и управлять коммуникационными процессами. Система по учёту коммуникаций стремится быть проектом, который может значительно улучшить жизнь и работу пользователя.

# **Список использованных источников**

* + 1. Вейцман В. М. Проектирование информационных систем – М: Издательство Лань, 2022. – 316 с.
    2. Гутгарц Р.Д Проектирование автоматизированных систем обработки информации и управления: учебное пособие для академического бакалавриата. – М.: Издательство Юрайт, 2019.
    3. Рочев К. В. Информационные технологии. Анализ и проектирование информационных систем – М: Издательство Лань, 2022. - 128 с.
    4. Проектирование информационных систем – М. / А. В. Остроух, Н.Е. Суркова - Лань, 2021. - 164 с.
    5. Проектирование информационных систем: учебник и практикум для академического бакалавриата / Д. В. Чистов, П. П. Мельников, А. В. Золотарюк, Н. Б. Ничепорук; под общ. ред. Д. В. Чистова. – М.: Издательство Юрайт, 2016.
    6. Документация по HTML URL: https://developer.mozilla.org/ru/docs/Web/HTML (дата обращения: 20.11.2023).
    7. Документация по CSS URL: https://developer.mozilla.org/ru/docs/Web/CSS/Reference (дата обращения: 20.11.2023).
    8. Документация по JavaScript URL: https://developer.mozilla.org/en-US/docs/Web/JavaScript (дата обращения: 20.11.2023).
    9. Документация по PHP URL: https://www.php.net/manual/ru/index.php (дата обращения: 20.11.2023).
    10. Документация по MySQL URL: https://dev.mysql.com/doc/ (дата обращения: 20.11.2023).