МИНИСТЕРСТВО ОБРАЗОВАНИЯ КИРОВСКОЙ ОБЛАСТИ

Кировское областное государственное профессиональное образовательное

бюджетное учреждение

«Слободской колледж педагогики и социальных отношений»

**ОТЧЕТ**

**по производственной практике**

**ПМ01. Разработка модулей программного обеспечения для компьютерных систем**

**Тема: «Разработка программного модуля «Дороги России»**

Студент

Платунов Павел Андреевич

Группа 21П-1

Специальность 09.02.07 Информационные системы и программирование

Руководитель практики от колледжа:

*Калинин Арсений Олегович*

Руководитель практики от организации:

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Петроченко Николай Юрьевич*

подпись

УТВЕРЖДАЮ:

Директор

\_ Олин Игорь Витальевич \_  
 Наименование организации

КОГОБУ Средняя школа пгт. Вахруши \_  
Слободского района \_  
\_\_\_\_\_\_\_\_\_\_\_\_\_ / \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись расшифровка

М. П.

2024-2025 уч. год

**Оглавление**

[1. Сведенья об организации (базе практики) 3](#_Toc194790466)

[2. Разработка и анализ технического задания 5](#_Toc194790467)

[2.1. Техническое задание 5](#_Toc194790468)

[2.2. Анализ технического задания 7](#_Toc194790469)

[3. Разработка программного модуля по алгоритму в соответствии с техническим заданием 10](#_Toc194790470)

[4. Оформление документации на программной модуль 17](#_Toc194790471)

[4.1. Руководство пользователя для настольного приложения 17](#_Toc194790472)

[4.2. Руководство пользователя для мобильного приложения 22](#_Toc194790473)

[5. Выполнение отладки программного модуля 26](#_Toc194790474)

[6. Выполнение тестирование программного модуля 28](#_Toc194790475)

[7. Оформление результатов тестирования 31](#_Toc194790476)

[8. Рефакторинг и оптимизация программного кода 36](#_Toc194790477)

[9. Разработка программного модуля для мобильного устройства 40](#_Toc194790478)

[10. Заключение 43](#_Toc194790479)

[Приложения 44](#_Toc194790480)

# Сведенья об организации (базе практики)

Производственную практику по модулю ПМ.01 «Разработка модулей программного обеспечения для компьютерных систем» я проходил в КОГОБУ СШ пгт. Вахруши Слободского района.

Данные компании:

Юридический адрес: 613110, Кировская область, Слободской район, пгт.Вахруши, ул.Ленина, д.4.

Контактные телефоны: (83362) 3-18-42 — приемная, 3-13-49 — бухгалтерия, 3-14-81 — вахта, учительская главного корпуса, 3-17-81 — вахта начальной школы.

Учебное заведение специализируется на общеобразовательных программах, а также предлагает:

• Профильные программы, обеспечивающие специализированное обучение;

• Углубленные программы изучения отдельных предметов для углубления знаний;

• Программы дополнительного образования для детей и взрослых, направленные на развитие навыков и знаний;

• Адаптированные программы для учеников с особыми образовательными потребностями, созданные с учетом их индивидуальных особенностей.

Структура организации:

1. Управление учреждением осуществляется на основе сочетания принципов единоначалия и коллегиальности.
2. Единоличным исполнительным органом учреждения является его руководитель – директор, который осуществляет текущее руководство деятельностью учреждения.
3. Коллегиальными органами управления учреждением являются совет учреждения, педагогический совет, общее собрание трудового коллектива.
4. Компетенции учредителя.
5. Директор учреждения.
6. Совет учреждения.
7. Педагогический совет.
8. Общее собрание трудового коллектива.

# Разработка и анализ технического задания

## 2.1. Техническое задание

1. Введение

Наименование проекта: Приложение для оптимизации работы организации «Дороги России».

Цель проекта: Разработка программного обеспечения для управления персоналом и документами, уведомления сотрудников о новостях и событиях внутри организации.

Целевая аудитория: Сотрудники организации.

2. Требования к функциональности:

2.1. Просмотр сотрудников

Просмотр списка сотрудников в том числе и подчинённых ниже отделов.

Отображение уволенного сотрудника (менее 30 дней серым цветом в списке, более 30 дней не отображать)

2.2. Управление сотрудниками

Добавление, увольнение, редактирование сотрудников.

Добавление, удаление событий (обучение, временное отсутствие, отпуск).

Фильтрация событий (прошедшие, текущие, будущие)

2.3. Управление документами

Авторизация с использованием JWT токена.

Просмотр документов.

Просмотр комментариев к документу.

Оставление комментария к документу.

2.4. Мобильное приложение

Просмотр новостей.

Оставление реакций на новость (положительная, отрицательная).

Просмотр событий.

Добавление события в системный календарь устройства.

3. Нефункциональные требования

3.1. Производительность

Поддержка одновременной работы до 50 пользователей.

Время отклика системы не более 2 секунд.

3.2. Надежность

Обеспечение бесперебойной работы 24/7.

Резервное копирование данных.

3.3. Безопасность

Защита документов (JWT токен).

3.4. Масштабируемость

Возможность увеличения количества пользователей и подразделений без потери производительность.

Возможность дальнейшего увеличения системы (внедрение веб-приложения).

3.5. Платформы

Настольное приложение для Windows.

Мобильное приложение для Android.

4. Технические требования

4.1. Стек технологий

Фронтенд и Бэкенд: XAML и C#.

База данных: MS Server.

Мобильное приложение: Xamarin Forms.

4.2. Инфраструктура

Хостинг на облачных серверах (Azure (ASP NET)).

4.3. API

Поддержка формата данных JSON.

5. Этапы разработки

5.1. Анализ и проектирование

Сбор требований.

Проектирование архитектуры системы.

Создание прототипов интерфейсов.

5.2. Разработка

Разработка базового функционала (управление сотрудниками и документами)

5.3. Тестирование

Модульное тестирование.

Интеграционное тестирование.

Нагрузочное тестирование.

5.4. Внедрение

Руководство пользователя.

5.5. Сопровождение

Техническая поддержка.

Исправление ошибок и доработка функционала.

6. Дополнительные требования

Приложения должны быть оформлены в фирменных цветах организации.

7. Заключение

Приложение должно быть интуитивно понятным, надежным и соответствовать современным стандартам разработки.

## 2.2. Анализ технического задания

После ознакомления с техническим заданием я составил UML диаграмму вариантов использования для большего понимания требований заказчика к системе (Рисунок 1). На основе анализа технического задания я спроектировал ER диаграмму для дальнейшего создания базы данных (Рисунок 2).



Рисунок 1 – UML Диаграмма вариантов использования



Рисунок 2 – ER диаграмма

# Разработка программного модуля по алгоритму в соответствии с техническим заданием

На основе анализа технического задания была разработана информационная система, состоящая из API для обмена данными между базой данных и приложениями, библиотека DLL для упрощения программирования и возможности дальнейшего развития и(или) исправления функционала программы без её полной переустановки, настольное приложение для доступа к системе на рабочем компьютере сотрудника.

API:

Взаимодействие пользователя с API построено через пользовательский интерфейс Swagger. Данный интерфейс позволяет просматривать все методы в API и тестировать их (Рисунок 3).

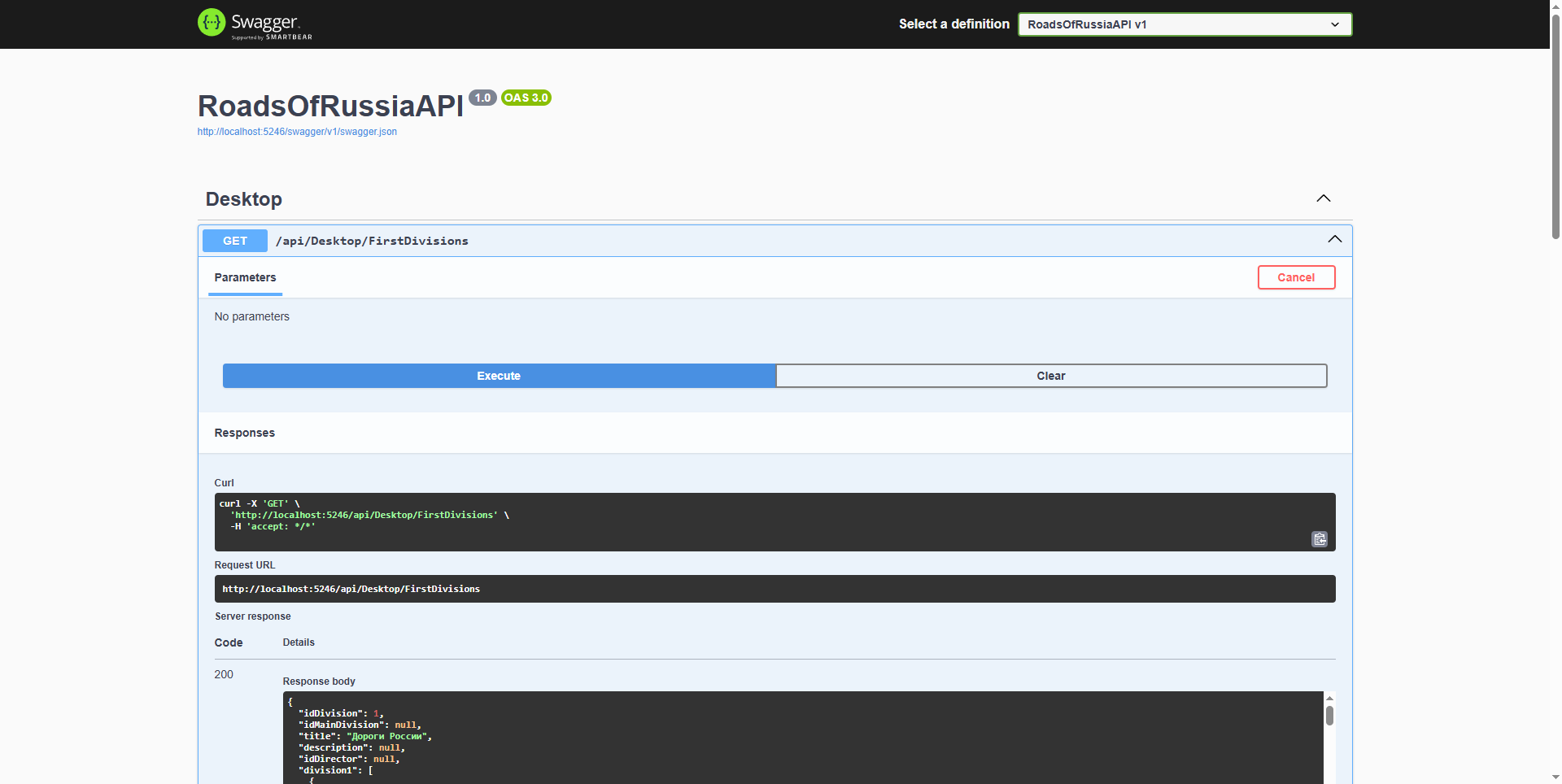


Рисунок 3 – Интерфейс Swagger

Само API содержит в себе следующие методы для работы основных модулей приложений:

Работа с документами организации:

* Авторизация с использованием JWT токена;
* Просмотр всех документов компании;
* Просмотр комментариев по выбранному документу;
* Комментирование выбранного документа.

Настольное приложение:

* Просмотр всех подразделений организации;
* Просмотр сотрудников по выбранному подразделению и дочерних подразделений;
* Просмотр сотрудников только по выбранному подразделению;
* Просмотр должностей компании;
* Увольнение выбранного сотрудника;
* Добавление(редактирование) выбранного сотрудника;
* Просмотр всех событий в календаре обучения выбранного сотрудника;
* Просмотр всех событий в календаре временного отсутствия выбранного сотрудника;
* Просмотр всех событий в календаре отпуска выбранного сотрудника;
* Удаление выбранного события из календаря обучения;
* Удаление выбранного события из календаря временного отсутствия;
* Удаление выбранного события из календаря отпуска;
* Добавление нового события в календарь обучения выбранного сотрудника;
* Добавление нового события в календарь временного отсутствия выбранного сотрудника;
* Добавление нового события в календарь отпуска выбранного сотрудника.

Мобильное приложение:

* Просмотр всех новостей организации;
* Оставление позитивной реакции на выбранную новость;
* Оставление негативной реакции на выбранную новость;
* Просмотр всех событий в организации;

В качестве примера представлен метод получения должности (Рисунок 3).

|  |
| --- |
| [HttpGet("Posts")]  public async Task<IActionResult> GetPost()  {  try  {  var post = await database.Post.ToListAsync();  if (post.Count == 0)  {  return NotFound(new { Message = "Post not found" });  }    return Ok(post);  }  catch (Exception ex)  {  return BadRequest(ex.Message);  }  } |

Рисунок 3 – Пример кода для получения всех должностей

Библиотека DLL:

Библиотека DLL создается для того, чтобы обеспечить дальнейшее развитие системы, исправление и оптимизацию кода, без непосредственного перекомпилирования приложения для её обновления. Посредством DLL реализовано взаимодействие приложения с API. Сама библиотека содержит методы, возвращающие набор данных полученные из API. Основные методы библиотеки:

Работа с документами организации:

* Авторизация с использованием JWT токена;
* Просмотр всех документов компании;
* Просмотр комментариев по выбранному документу;
* Комментирование выбранного документа.

Настольное приложение:

* Просмотр всех подразделений организации;
* Просмотр сотрудников по выбранному подразделению и дочерних подразделений;
* Просмотр сотрудников только по выбранному подразделению;
* Просмотр должностей компании;
* Увольнение выбранного сотрудника;
* Добавление(редактирование) выбранного сотрудника;
* Просмотр всех событий в календаре обучения выбранного сотрудника;
* Просмотр всех событий в календаре временного отсутствия выбранного сотрудника;
* Просмотр всех событий в календаре отпуска выбранного сотрудника;
* Удаление выбранного события из календаря обучения;
* Удаление выбранного события из календаря временного отсутствия;
* Удаление выбранного события из календаря отпуска;
* Добавление нового события в календарь обучения выбранного сотрудника;
* Добавление нового события в календарь временного отсутствия выбранного сотрудника;
* Добавление нового события в календарь отпуска выбранного сотрудника.

Мобильное приложение:

* Просмотр всех новостей организации;
* Оставление позитивной реакции на выбранную новость;
* Оставление негативной реакции на выбранную новость;
* Просмотр всех событий в организации;

В качестве примера представлен метод получения должностей от API (Рисунок 4).

|  |
| --- |
| public async Task<List<PostModel>> GetPostAsync()  {  using (HttpClient httpClient = new HttpClient())  {  string apiUrl = $"http://localhost:5246/api/Desktop/Posts";  var response = await httpClient.GetStringAsync(apiUrl);  var post = JsonConvert.DeserializeObject<List<PostModel>>(response);  return post;  }  } |

Рисунок 4 – Пример кода для получения должностей от API

Настольное приложение:

Настольное приложение переназначается для работы сотрудников на IBM совместимой технике. И позволяет сотрудникам выполнять следующие действия в системе:

* Просматривать все подразделения организации с иерархией подразделений;
* Просматривать сотрудников выбранного подразделения и дочерних подразделений рекурсивно;
* Просматривать карточку сотрудника, которая состоит их данных сотрудника и событий в календарях обучения, временного отсутствия, отпуска;
* Добавлять нового сотрудника;
* Редактировать выбранного сотрудника;
* Увольнять сотрудника если у него незапланированно обучения;
* Добавлять новое событие в календарь обучения, временного отсутствия, отпуска в карточке сотрудника;
* Удалять событие из календаря обучения, временного отсутствия, отпуска в карточке сотрудника;
* Фильтровать события на прошедшие, текущие, будущие.

Интерфейс программы выполнен в фирменных цветах организации. Главное окно представлено списком подразделений слева и списком сотрудников справа (Рисунок 5).

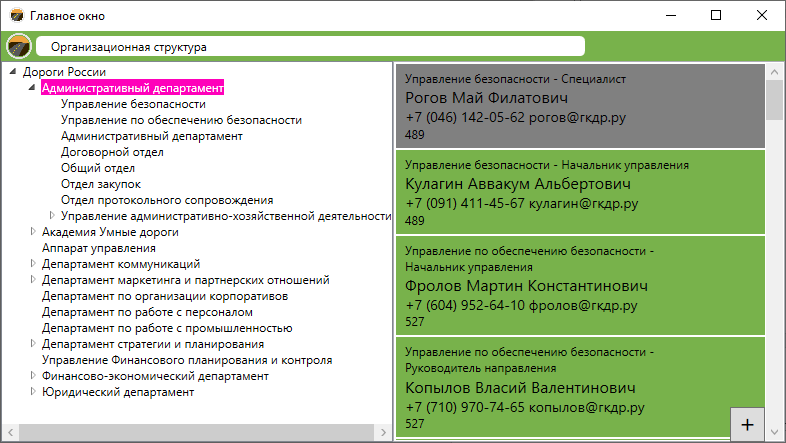


Рисунок 5 – Главное окно приложения

Карточка сотрудника представлена полями с данными сотрудника для редактирования слева и событиями с возможностью фильтрации по времени, кнопками добавления и удаления справа (Рисунок 6).

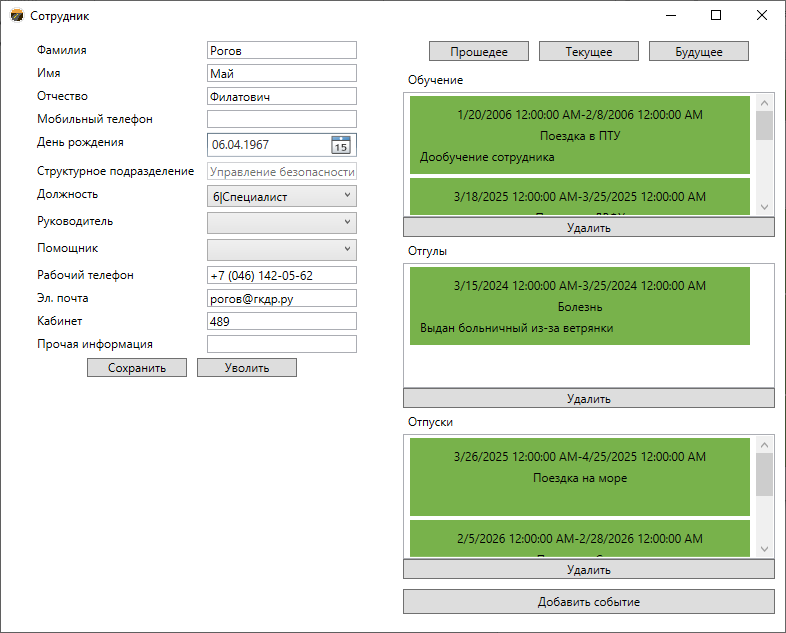


Рисунок 6 – Карточка сотрудника

Окно добавления события содержит поля для выбора типа, дат, названия и описания события (Рисунок 7).

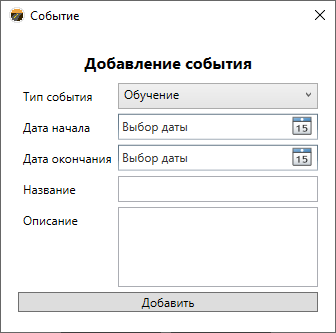


Рисунок 7 – Окно добавления события

# Оформление документации на программной модуль

## 4.1. Руководство пользователя для настольного приложения

**Введение**

Приложение предназначено для управления сотрудниками организации. Данное приложение позволяет повысить эффективность управления персоналов, позволяя работать с данными сотрудников, а также планировать обучение, отпуск или временное его отсутствие.

**Функционал приложения**

Основной функционал приложения:

• Просмотр подразделений в виде иерархического списка;

• Просмотр сотрудников выбранного подразделения (в том числе и подчинённых подразделений);

• Добавление, редактирование, увольнение сотрудника;

• Просмотр событий;

• Фильтрация событий (прошедшие, текущие, будущие);

• Добавление, удаление событий.

**Просмотр подразделений**

Приложение позволяет просматривать подразделения в иерархичном списке. В данном списке пользователь может просмотреть основное подразделение и его дочерние подразделения. Для этого пользователь должен нажать на стрелочку радом с название подразделения или дважды нажать на само подразделение для его раскрытия (Рисунок 8).

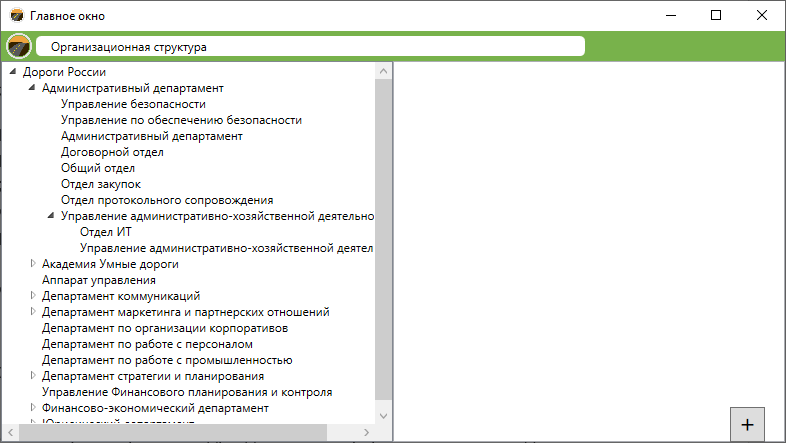


Рисунок 8 – Просмотр подразделений.

**Просмотр сотрудников**

Пользователь может просматривать сотрудников подразделений, при этом в список сотрудников выводиться и сотрудники нижестоящих подразделений. Для просмотра списка сотрудников пользователь должен нажать на подразделение, после чего сотрудники отобразятся в списке (Рисунок 9).

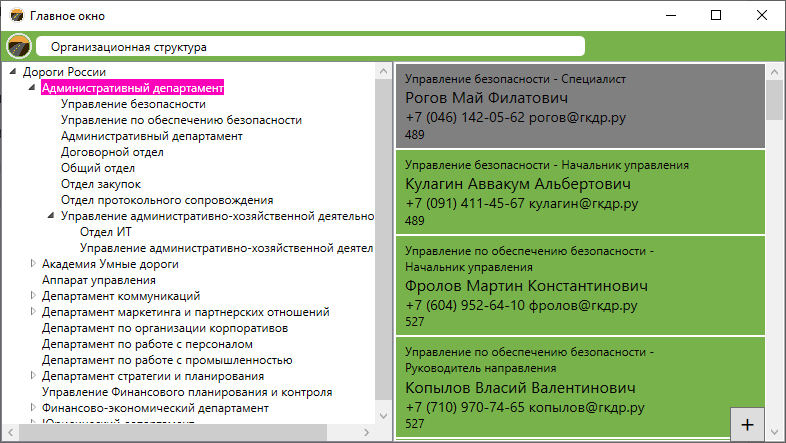


Рисунок 9 – Просмотр списка сотрудников

**Добавление, редактирование, увольнение сотрудников**

Пользователь может добавить или редактировать сотрудника. Для добавления нового сотрудника пользователь должен нажать на значок «+» в списке сотрудника после нажатия на кнопку откроется окно создания нового сотрудника (Рисунок 10). Для редактирования сотрудника пользователь должен дважды нажать на выбираемого сотрудника после чего откроется окно редактирования сотрудника (Рисунок 10).

Окно добавления(редактирования) представлено набором полей содержащие данные сотрудника, а также списками, отображающими обучение, временное отсутствие или отпуск сотрудника. После добавления новой информации или редактирование существующей необходимо нажать на кнопку сохранения (Рисунок 10).

Для увольнения сотрудника необходимо нажать на кнопку «Уволить», при этом стоит учитывать, что нельзя уволить сотрудника, у которого запланировано обучение. После нажатия на кнопку «Уволить» появиться окно подтверждения для увольнения сотрудника (Рисунок 11), в данном окне необходимо нажать кнопку «Да» для увольнения или кнопку «Нет» для отмены действия.

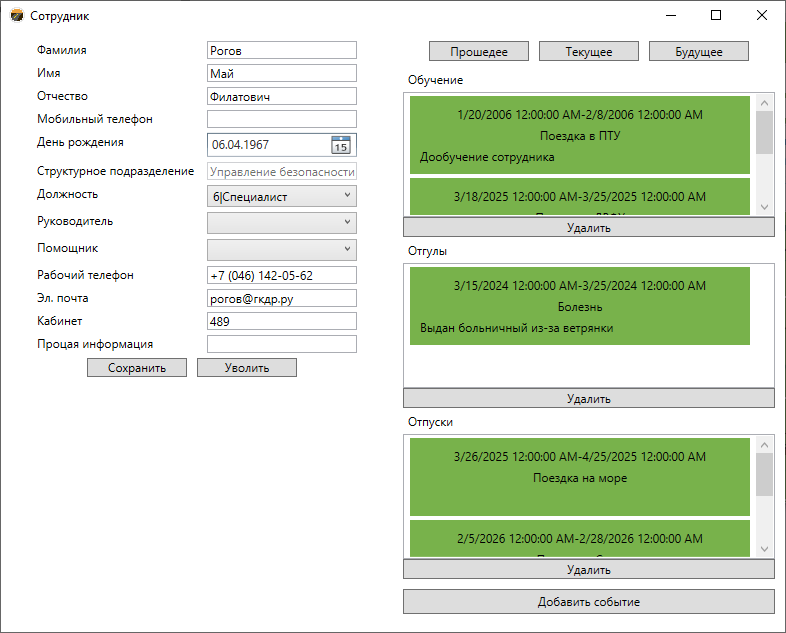


Рисунок 10 – Окно редактирования сотрудника

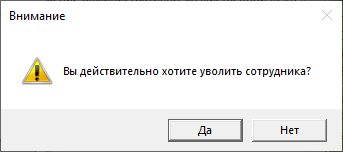


Рисунок 11 – Окно подтверждения увольнения сотрудника

**Просмотр событий**

Пользователь может просматривать запланированные события выбранного сотрудника в окне добавления(редактирования) сотрудников. События разделены на 3 основные группы: обучение, временное отсутствие(отгул), отпуск (Рисунок 10).

**Фильтрация событий**

Пользователь может отфильтровать события по трем категориям: прошедшие, текущие, будущие. Для фильтрации необходимо в окне добавления(редактирования) нажать на соответствующую кнопку вверху правой панели (Рисунок 10). После нажатия на кнопку списки событий отобразят события по выбранной категории (Рисунок 12).

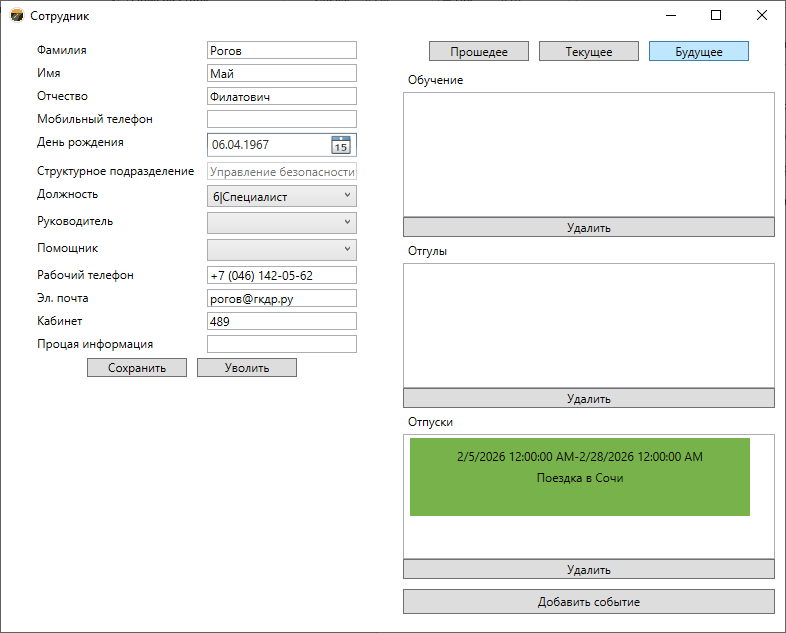


Рисунок 12 – Просмотр будущих событий

**Добавление, удаление событий**

Пользователь может добавлять события, для этого пользователь должен нажать на кнопку «Добавить событие» в окне добавления(редактирования) сотрудника (Рисунок 10). После нажатия на кнопку откроется окно добавления события (Рисунок 13). В данном окне можно выбрать тип обучения (обучение, временное отсутствие, отпуск), ввести название, описание, а также даты начала и окончания события.

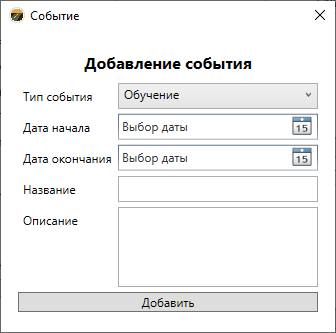


Рисунок 13 – Окно добавления сотрудника

**Требования к ПО**

Технические требования к АРМ:

* Процессор: 1 гигагерц (ГГц) или быстрее с двумя или более ядрами на совместимом 64-разрядном процессоре;
* ОЗУ: 1 гигабайта (ГБ) или больше;
* ПЗУ: 512 мегабайт (МБ) или больше свободного места на диске;
* Графическая карта: совместим с DirectX 9 или более поздней версии с драйвером WDDM 2.0;
* Дисплей: дисплей с разрешением (1280\*720).
* Устройства ввода: клавиатура, компьютерная мышь.
* ОС: Windows 7 или более поздний.

## 4.2. Руководство пользователя для мобильного приложения

**Введение**

Приложение предназначено для уведомления пользователей о происходящем в организации через новостную ленту ил список событий. Для большего удобства сотрудников они могут автоматически занести события в системный календарь своего устройства

**Функционал приложения**

Основной функционал приложения:

• Просмотр новостей;

• Оставление реакции на новость (положительную, отрицательную);

• Просмотр событий;

• Добавление события в системный календарь.

**Просмотр новостей**

Пользователь может просматривать новости на вкладке «Новости» (Рисунок 14).

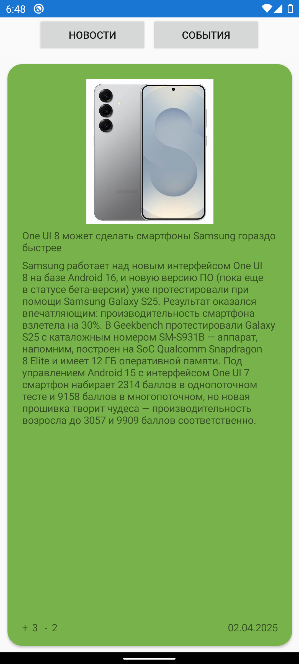


Рисунок 14 – Вкладка «Новости»

**Оставление реакции на новость**

Пользователь может оставить реакцию на новость как положительную, так и отрицательную. Для этого пользователь должен нажать на новость после чего откроется окно с выбором реакции (Рисунок 15).

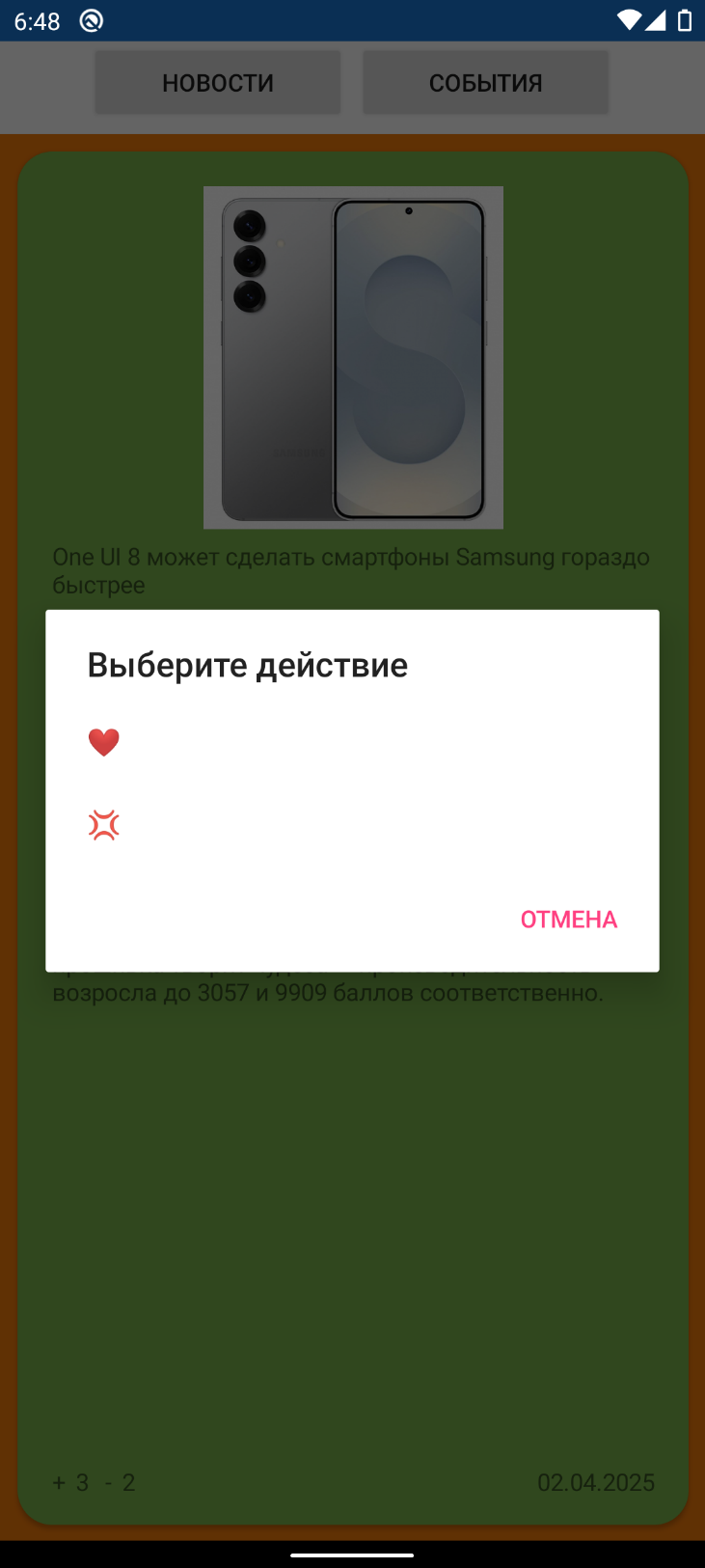


Рисунок 15 – Окно выбора реакции

**Просмотр событий**

Пользователь может просматривать события на вкладке «События» (Рисунок 16).

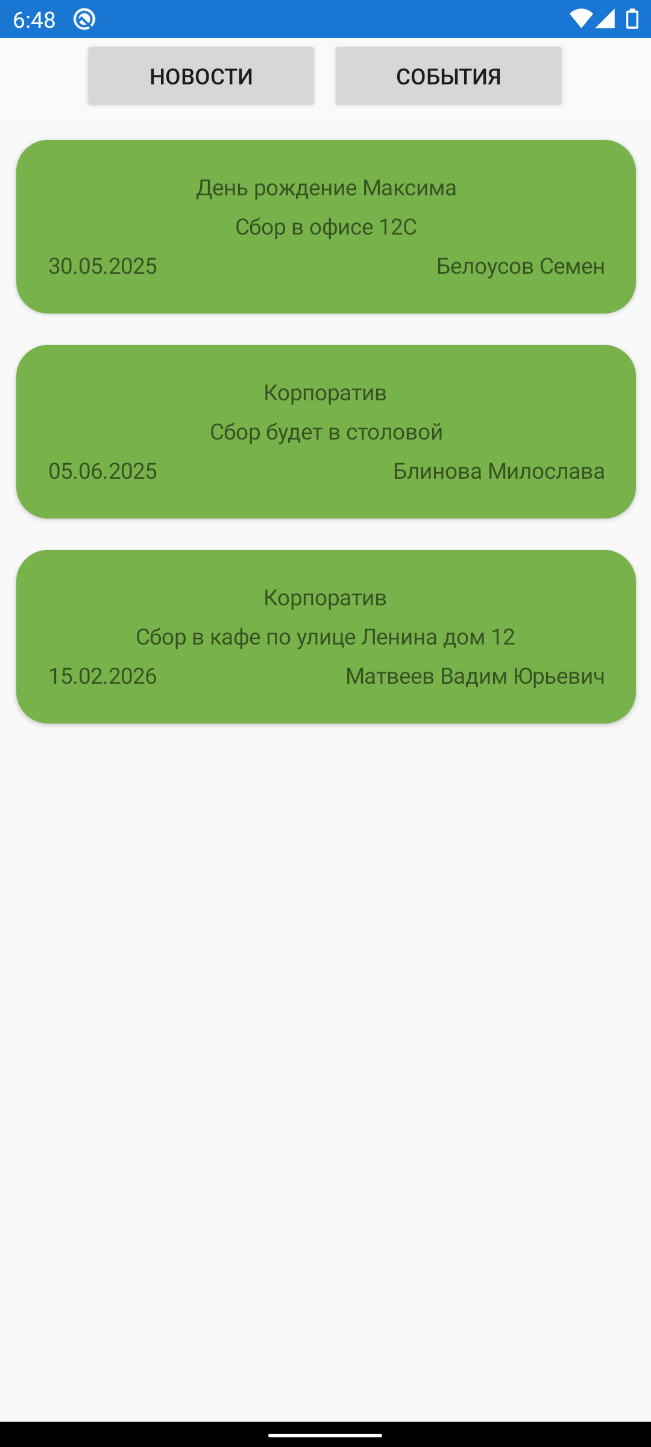


Рисунок 16 – Вкладка «События»

Добавление события в системный календарь

Пользователь может добавить событие в системный календарь для этого необходимо нажать на событие после чего событие добавиться в системный календарь (Рисунок 17).

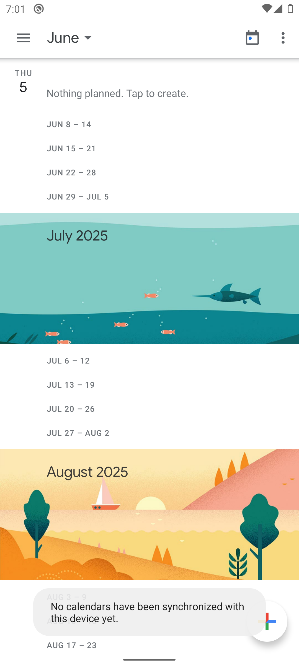


Рисунок 17 – Добавление события в системный календарь

**Требования к ПО**

Технические требования к АРМ:

* Процессор: 1,8 гигагерц (ГГц) или быстрее с четырьмя или более ядрами;
* ОЗУ: 2 гигабайта (ГБ) или больше;
* ПЗУ: 200 мегабайт (МБ) или больше свободного места на диске;
* ОС: Android 8 или более поздний.

# Выполнение отладки программного модуля

В процессе разработки программного модуля была проведена его отладка для устранения ошибок в коде и оптимизации его работы (Рисунок 18,19,20,21).

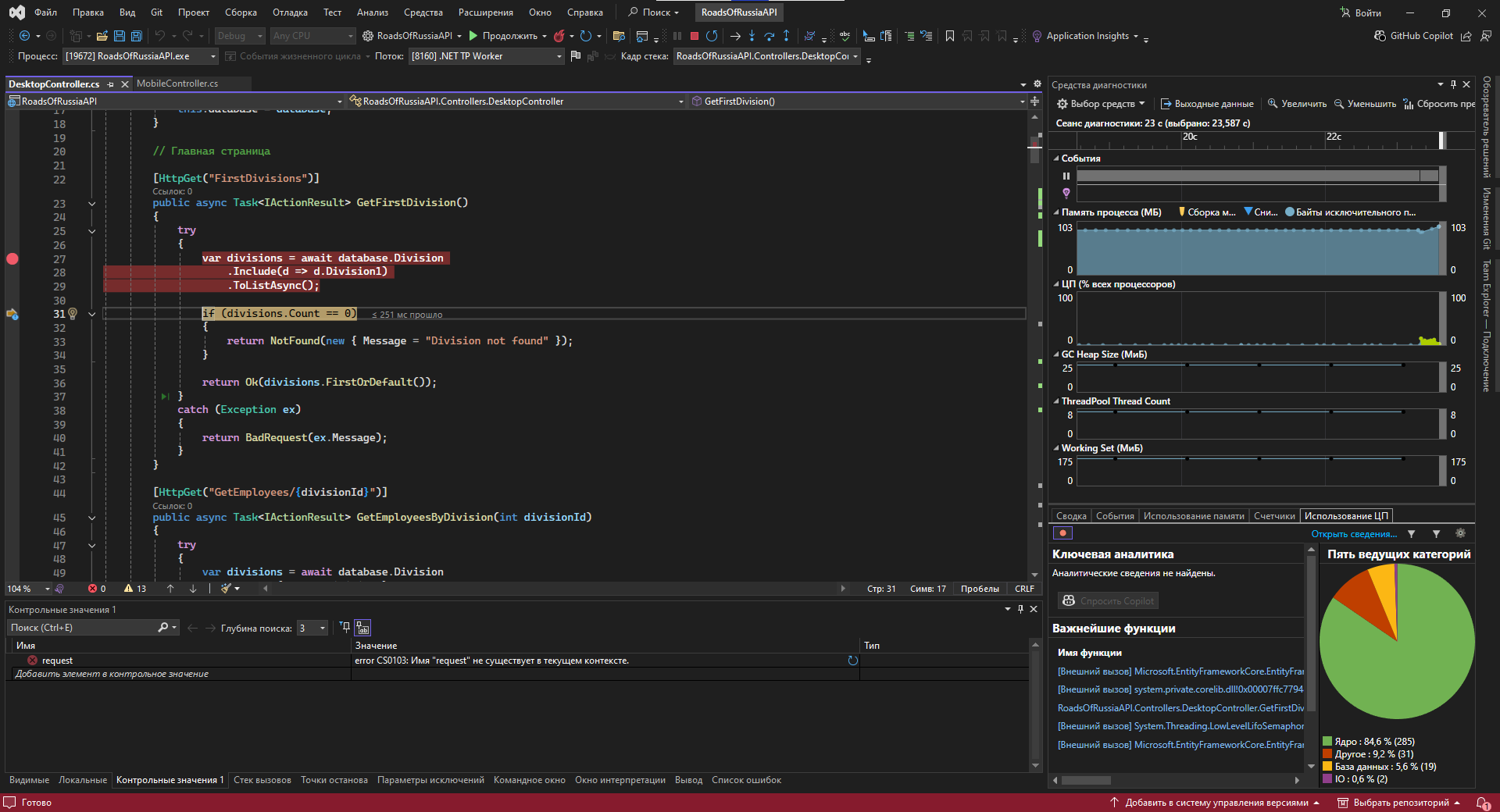


Рисунок 18 – Отладка API

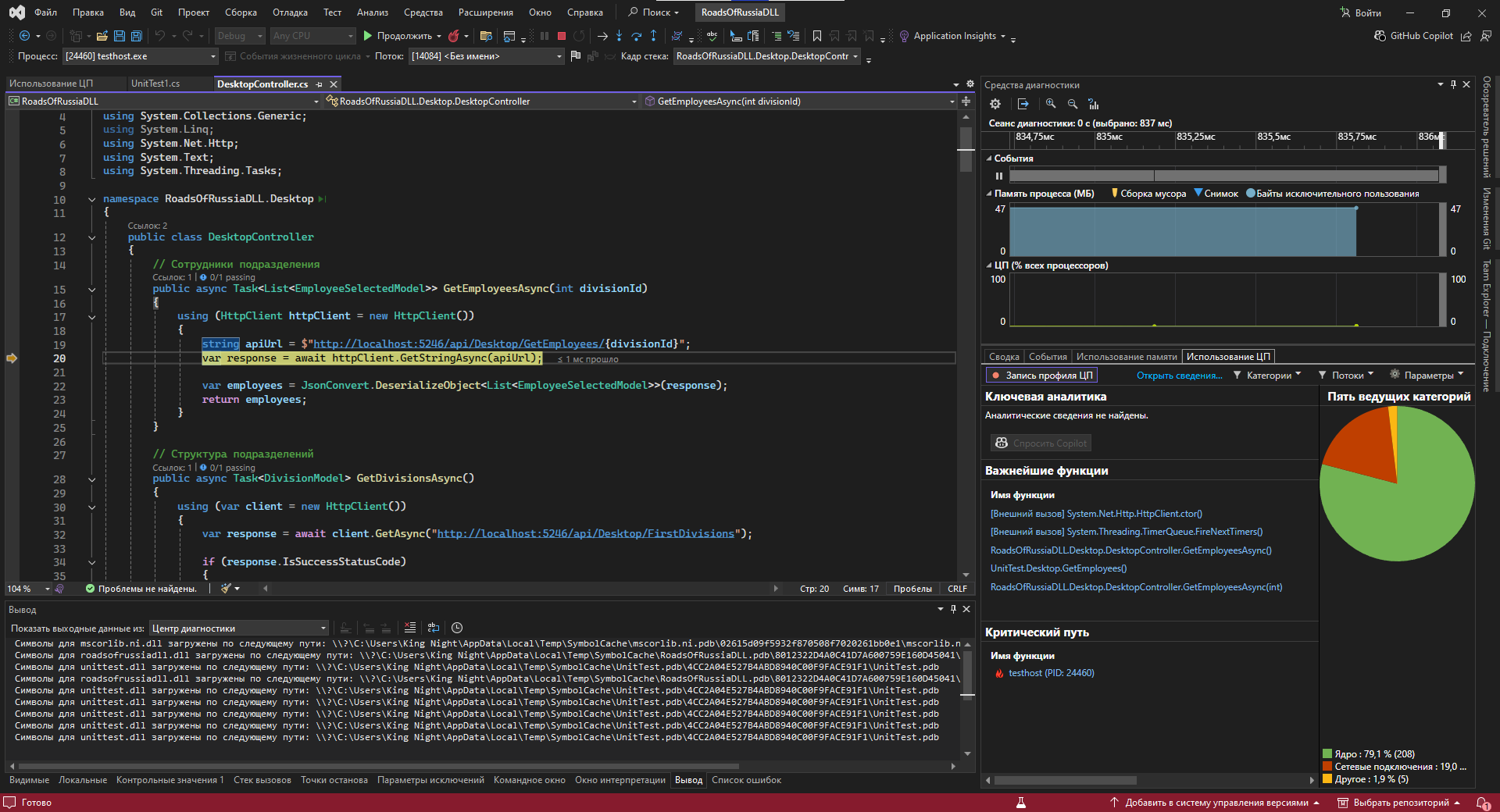


Рисунок 19 – Отладка DLL

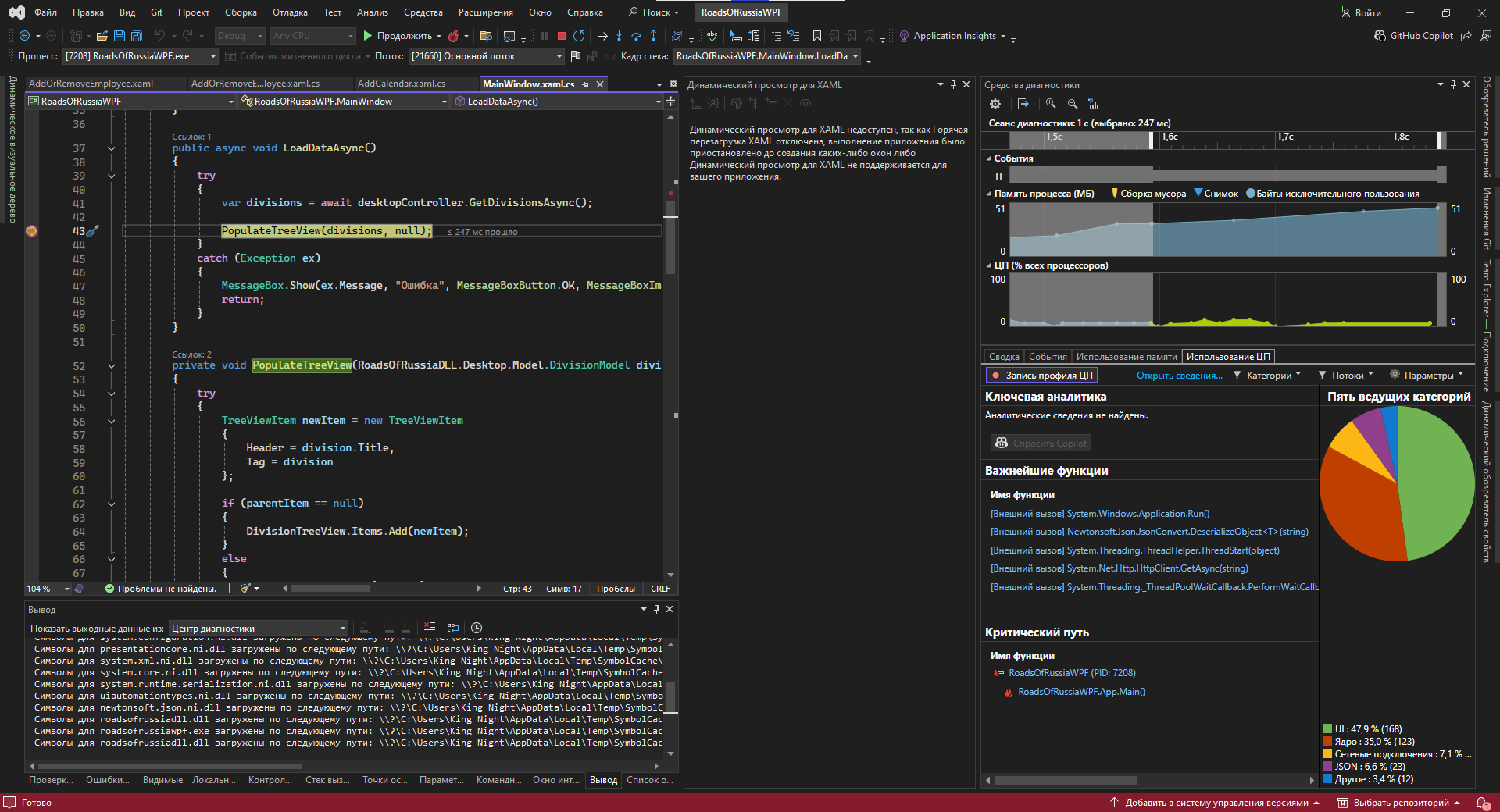


Рисунок 20 – Отладка настольного приложения

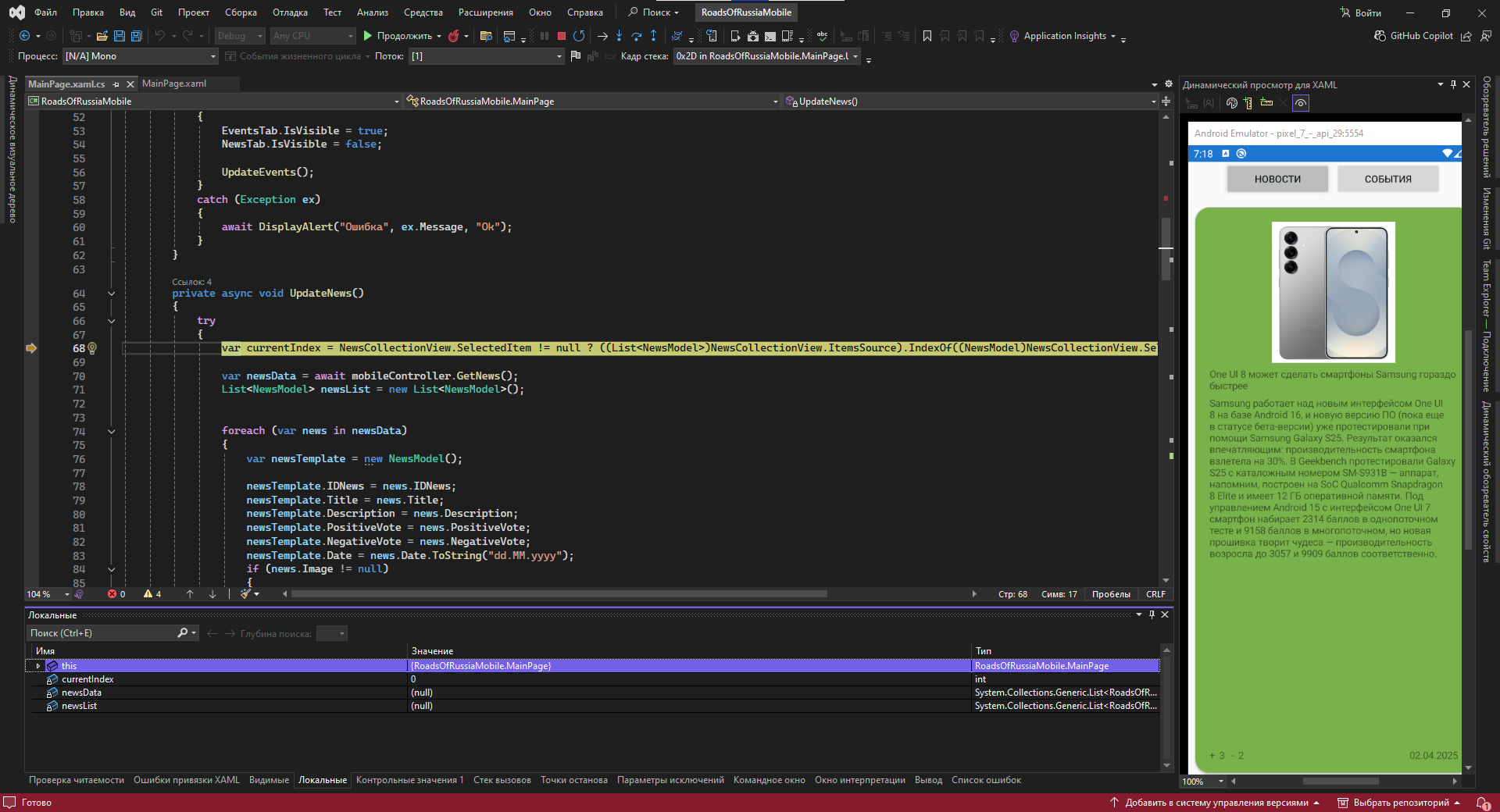


Рисунок 21 – Отладка мобильного приложения

# Выполнение тестирование программного модуля

При тестирование программного модуля были использованы следующие тесты: Unit тестирование для проверки кода на работоспособность, интеграционное тестирование для проверки успешности работы DLL библиотеки после интеграции с приложением, нагрузочное тестирование для проверки работоспособности API при большой нагрузке со стороны пользователей, тестирование интерфейса для проверки корректности отображения графических элементов интерфейса.

Unit тестирование:

При Unit тестирование были протестированы все методы для работы приложения с API (Рисунок 22).

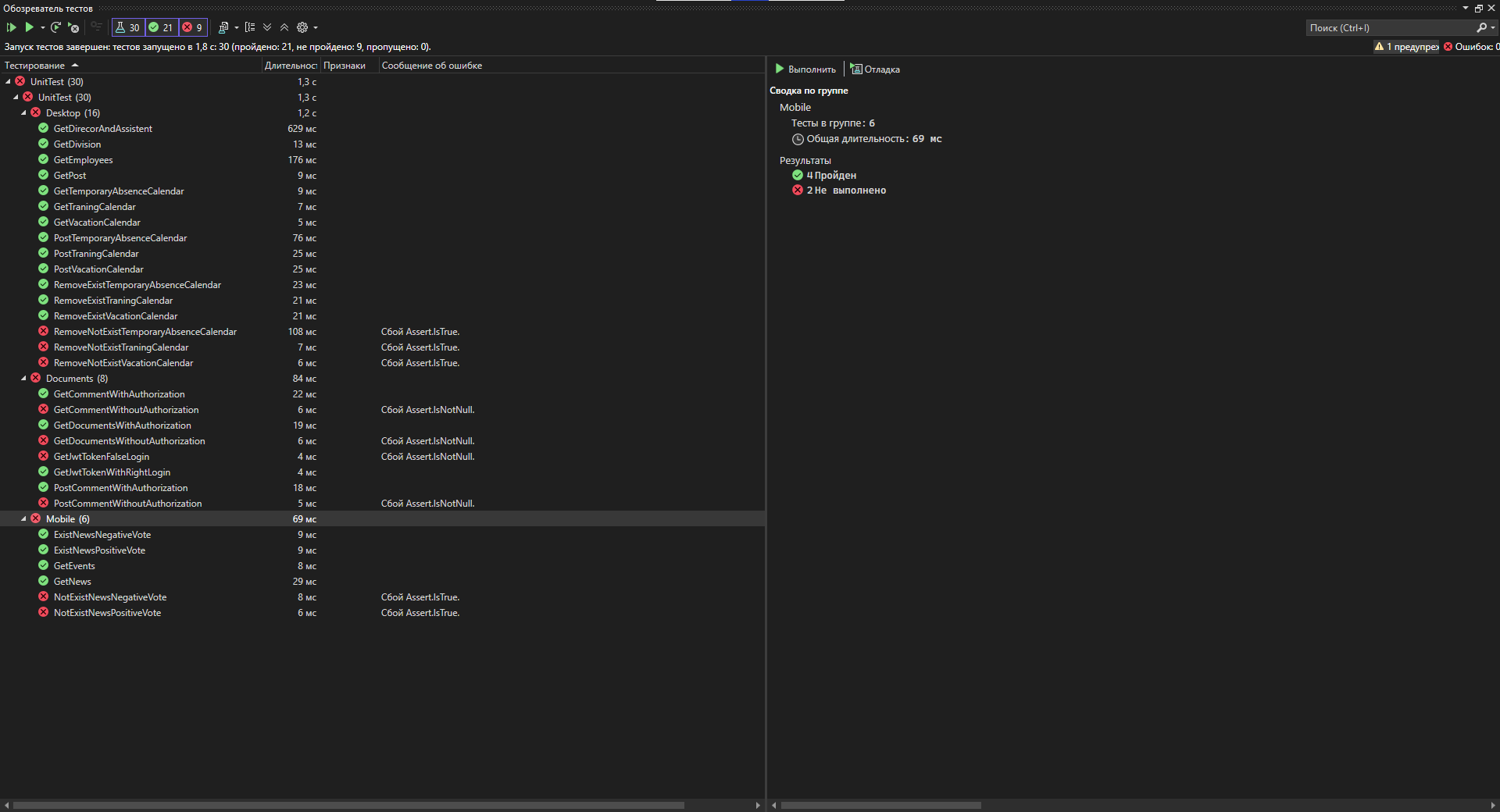


Рисунок 22 – Unit тестирование всех методов для работы с API

Интеграционное тестирование:

При интеграционном тестирование было протестированные интегрирование библиотеки DLL с приложением (Рисунок 23).

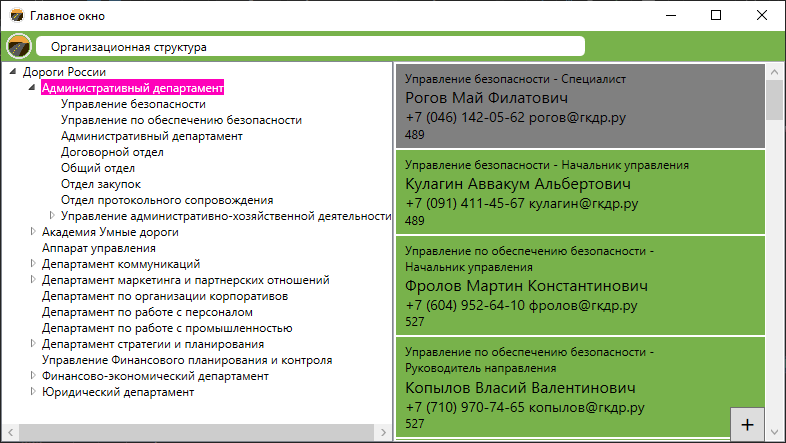


Рисунок 23 – Интеграционное тестирование

Нагрузочное тестирование:

При нагрузочном тестировании было протестированные стойкость API к большим нагрузкам со стороны пользователей (Рисунок 24).

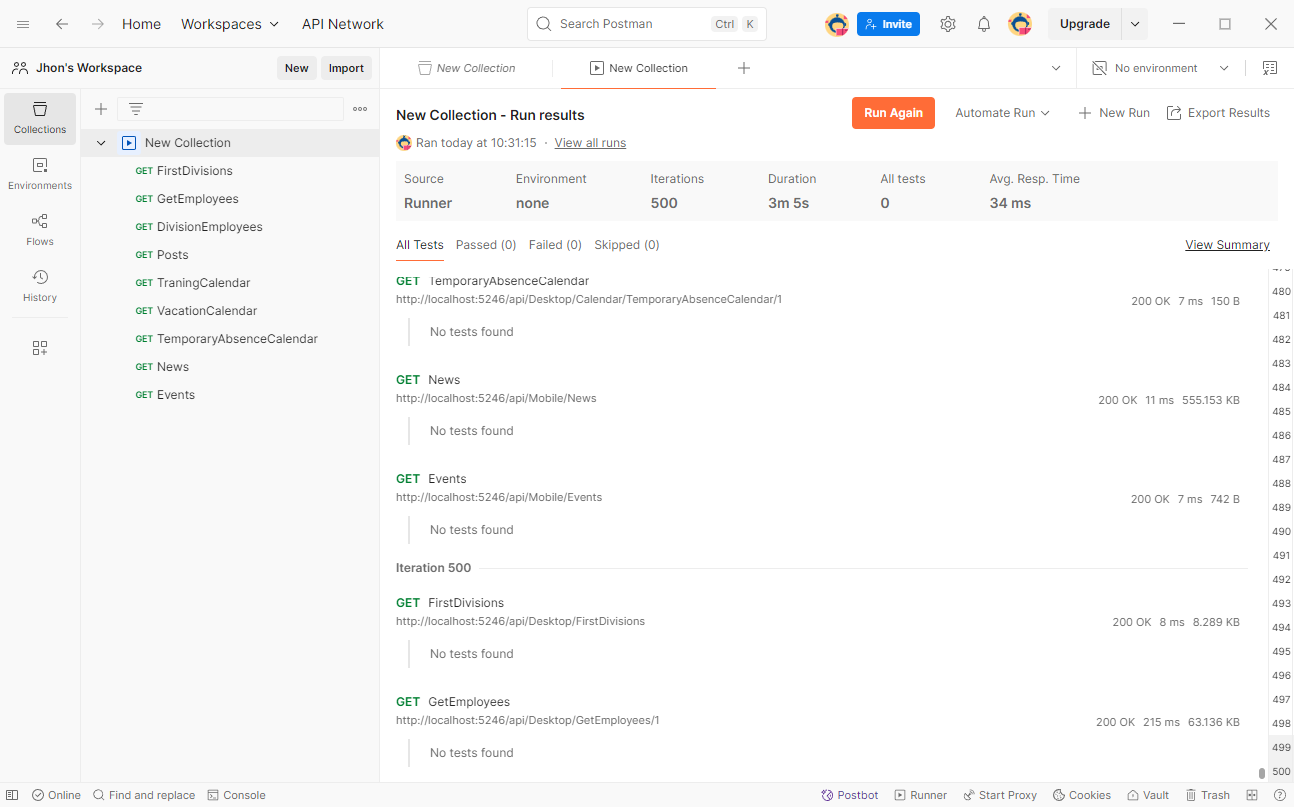


Рисунок 24 – Нагрузочное тестирование

Тестирование интерфейса:

При тестировании интерфейса был протестирован интерфейс мобильного приложения на наличие некорректности в отображение графических элементов интерфейса (Рисунок 25).

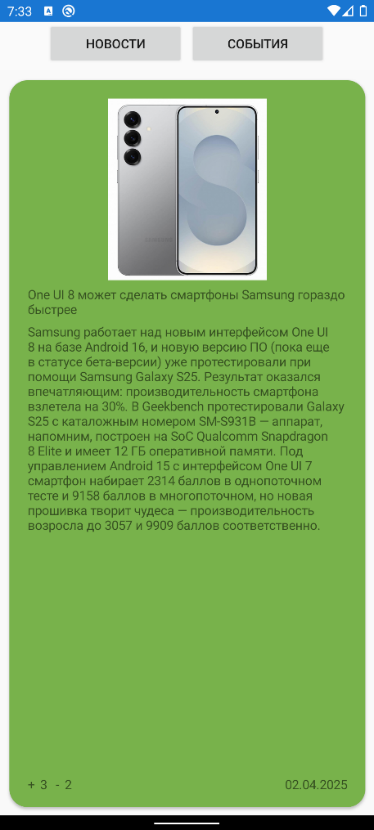


Рисунок 25 – Тестирование интерфейса

# Оформление результатов тестирования

Для проведения тестирования корректности работы приложения было проведено Unit тестирование (Рисунок 22). Результатом проведения тестирования было тестовое покрытие на все модули из 30 тестов как положительных, так и отрицательных, связанные с получение и работой с данными на стороне API, так как приложение полностью работает за счет API. Результатом тестирования можно отметить полную работоспособность методов приложения с обработкой ошибок, которые могут возникнуть при неправильной работе с приложением.

Проведенное нагрузочное тестирование при помощи Postman показало высокую надежность написанного API, и малое время ответа при большой нагрузке на API в среднем 35 миллисекунд (Рисунок 24). Само тестирование состояло из 9 методов и 500 итераций, что дало полную картину работы API при большой нагрузке со стороны пользователей, стоит отметить, что данной нагрузки не будет в реальных условиях использования программного обеспечения, он тестирование дает понять что есть возможность дальнейшего расширения программного обеспечения и увеличения количества пользователей системы.

Тестирование интеграции проверяет работу настольного приложения с DLL библиотекой и API через эту библиотеку. Для проведения тестирования интеграции было выбрано ручное тестирование (тест кейсы). По результатам тестирования было установлено что интеграция DLL библиотеки прошла успешно и все методы работают корректно. Некоторые тест кейсы интеграционного тестирования приведены ниже, подробное тестирование расписано в приложение 2.

|  |  |
| --- | --- |
| TestCase # | Division\_1 |
| Приоритет | Высокий |
| Название тестирования/Имя | Загрузка подразделений |
| Резюме испытания | Корректная загрузка всех подразделений с иерархией в список |
| Шаги тестирования | 1. Раскрыть первый элемент списка |
| Данные тестирования | Подразделений организации |
| Ожидаемый результат | Присутствуют все подразделения с иерархией |
| Фактический результат | Присутствуют все подразделения с иерархией |
| Предпосылки | Запуск приложения |
| Постусловия |  |
| Статус (Pass/Fail) | Pass |
| Комментарии |  |

|  |  |
| --- | --- |
| TestCase # | Employee\_1 |
| Приоритет | Высокий |
| Название тестирования/Имя | Загрузка сотрудников подразделения |
| Резюме испытания | Корректная загрузка всех сотрудников выбранного подразделения и дочерних подразделений |
| Шаги тестирования | 1. Раскрыть первое подразделение 2. Нажать на любое подразделение |
| Данные тестирования | Подразделений и сотрудники организации |
| Ожидаемый результат | Присутствуют все сотрудники, работающие в выбранном подразделение и дочерних подразделениях |
| Фактический результат | Присутствуют все сотрудники, работающие в выбранном подразделение и дочерних подразделениях |
| Предпосылки | Запуск приложения |
| Постусловия |  |
| Статус (Pass/Fail) | Pass |
| Комментарии |  |

Тестирование интерфейса проверяет корректность отображения графических элементов в интерфейсе программы в сценариях повседневного использования приложения, по результатам тестирования было выявлено неправильное отображение данных в карточке новостей, описание при перевороте экрана накладывается на реакции и дату. Тестирование интерфейса было проведено вручную и оформлено в виде тест кейсы. Некоторые тест кейсы тестирования интерфейса приведены ниже, более подробное тестирование приведено в приложении 3.

|  |  |
| --- | --- |
| TestCase # | ScreenRotation\_1 |
| Приоритет | Низкий |
| Название тестирования/Имя | Смена ориентации экрана на горизонтальный во вкладке «Новости» |
| Резюме испытания | Поворот устройства на 90 влево |
| Шаги тестирования | 1. Перевернуть устройство на 90 влево |
| Данные тестирования | Новости организации |
| Ожидаемый результат | Переворот элементов на 90 влево |
| Фактический результат | Переворот элементов на 90 влево |
| Предпосылки | Открытие вкладки «Новости» |
| Постусловия | Обновление интерфейса |
| Статус (Pass/Fail) | Pass |
| Комментарии |  |

|  |  |
| --- | --- |
| TestCase # | ScreenRotation\_2 |
| Приоритет | Низкий |
| Название тестирования/Имя | Смена ориентации экрана на горизонтальный во вкладке «События» |
| Резюме испытания | Поворот устройства на 90 влево |
| Шаги тестирования | 1. Перевернуть устройство на 90 влево |
| Данные тестирования | События организации |
| Ожидаемый результат | Переворот элементов на 90 влево |
| Фактический результат | Переворот элементов на 90 влево |
| Предпосылки | Открытие вкладки «События» |
| Постусловия | Обновление интерфейса |
| Статус (Pass/Fail) | Pass |
| Комментарии |  |

|  |  |
| --- | --- |
| TestCase # | ScreenRotation\_3 |
| Приоритет | Средний |
| Название тестирования/Имя | Правильное отображение новостей при повороте экрана в горизонтальный режим |
| Резюме испытания | Проверка правильности отображения новостей после смены ориентации экрана на горизонтальный |
| Шаги тестирования | 1. Перевернуть устройство на 90 влево |
| Данные тестирования | Новости организации |
| Ожидаемый результат | Элементы интерфейса не наслаиваются друг на друга |
| Фактический результат | Описание наслаивается на реакции и дату публикацию |
| Предпосылки | Открытие вкладки «Новости» |
| Постусловия | Обновление интерфейса |
| Статус (Pass/Fail) | Fail |
| Комментарии |  |

# Рефакторинг и оптимизация программного кода

После написания первоначальной версии программы и проведения первичного тестирования на проверку корректной работы методов приложения следует рефакторинг кода для повышения читаемости кода, сам рефакторинг включает в себя упрощение методов с сохранением функциональности метода и комментирование кода для дальнейшей разработки (Рисунок 26, 27, 28, 29). Сам рефакторинг включает в себя и оптимизацию кода приложения для минимизации нагрузки на систему, так, например в карточке сотрудника события подгружаться с API 1 раз при переходе на модальное окно, после чего все форматирование выполняется на стороне настольного приложения (Рисунок 30, 31).

|  |
| --- |
| // Новости  // Получение новости  [HttpGet("News")]  public async Task<IActionResult> GetNews()  {  try  {  var newsData = new List<NewsModel>();  foreach (var news in await database.News.ToListAsync())  {  var singleNews = new NewsModel  {  IDNews = news.IDNews,  Title = news.Title,  Description = news.Description,  Image = news.Image,  PositiveVote = news.PositiveVote,  NegativeVote = news.NegativeVote,  Date = news.Date,  };  newsData.Add(singleNews); |

Рисунок 26 – Метод для получения новостей до рефакторинга

|  |
| --- |
| }  bool isEmpty = newsData.Count == 0;  if (isEmpty)  {  return NotFound(new { Status = "Error", Message = "No news found", Code = 404 });  }  var response = newsData.Select(n => new  {  Id = n.IDNews,  Title = n.Title,  Description = n.Description,  ImageUrl = n.Image,  Votes = $"{n.PositiveVote}-{n.NegativeVote}",  PublicationDate = n.Date.ToString("yyyy-MM-dd HH:mm:ss")  }).ToList();  return Ok(response);  }  catch (Exception ex)  {  var errorResponse = new  {  ErrorMessage = ex.Message,  ExceptionType = ex.GetType().Name,  StackTrace = ex.StackTrace,  SolutionHint = "Please contact administrator"  };  return BadRequest(errorResponse);  }  } |

Рисунок 27 – Продолжение метода для получения новостей до рефакторинга

|  |
| --- |
| // Новости  // Получение новости  [HttpGet("News")]  public async Task<IActionResult> GetNews()  {  try |

Рисунок 28 – Метод для получения новостей после рефакторинга

|  |
| --- |
| {  var newsData = await (from news in database.News  select new NewsModel  {  IDNews = news.IDNews,  Title = news.Title,  Description = news.Description,  Image = news.Image,  PositiveVote = news.PositiveVote,  NegativeVote = news.NegativeVote,  Date = news.Date,  }).ToListAsync();  if (newsData.Count == 0)  {  return NotFound(new { Message = "News not found" });  }  return Ok(newsData);  }  catch (Exception ex)  {  return BadRequest(ex.Message);  }  } |

Рисунок 29 – Продолжение метода для получения новостей после рефакторинга

|  |
| --- |
| // Общее объявление коллекции для дальнейшего использование  private List<RoadsOfRussiaDLL.Desktop.Model.CalandarModel> traningCalendar;  private List<RoadsOfRussiaDLL.Desktop.Model.CalandarModel> vacationCalendar;  private List<RoadsOfRussiaDLL.Desktop.Model.CalandarModel> temporaryAbsenceCalendar;  // Получение данных из API  traningCalendar = await desktopController.GetTraningCalendar(selectedEmployee.IDEmployee);  vacationCalendar = await desktopController.GetVacationCalendar(selectedEmployee.IDEmployee);  temporaryAbsenceCalendar = await desktopController.GetTemporaryAbsenceCalendar(selectedEmployee.IDEmployee);  // Работа с календарями  private void PastTime\_Click(object sender, RoutedEventArgs e) |

Рисунок 30 – Оптимизированный код для работы с календарями событий

|  |
| --- |
| {  try  {  if(traningCalendar != null)  TraningCalendarListView.ItemsSource = traningCalendar.Where(x => x.EndDate < DateTime.Now.Date).ToList();  if(vacationCalendar != null)  VacationCalendarListView.ItemsSource = vacationCalendar.Where(x => x.EndDate < DateTime.Now.Date).ToList();  if(temporaryAbsenceCalendar != null)  TemporaryAbsenceListView.ItemsSource = temporaryAbsenceCalendar.Where(x => x.EndDate < DateTime.Now.Date).ToList();  }  catch (Exception ex)  {  MessageBox.Show(ex.Message, "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);  return;  }  } |

Рисунок 31 – Продолжение кода для работы с календарем событий

# Разработка программного модуля для мобильного устройства

Для поддержания сотрудников в курсе событий организации было разработано мобильное приложение. Данное приложение призвано предоставлять сотрудникам актуальную информацию через новостную ленту и список событий. Основной функционал мобильного приложения:

* Просмотр новостей;
* Оставление реакции на новость;
* Просмотр событий;
* Добавление события в системный календарь.

Интерфейс приложения выполнен в фирменных цветах организации. Главное окно содержит 2 вкладки: новости (Рисунок 32) и события (Рисунок 33).

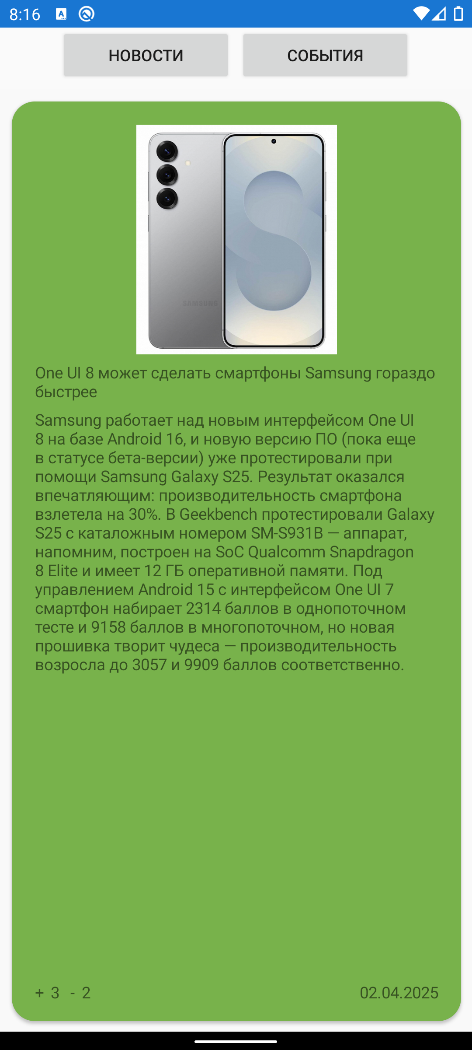


Рисунок 32 – Открытая вкладка «Новости»

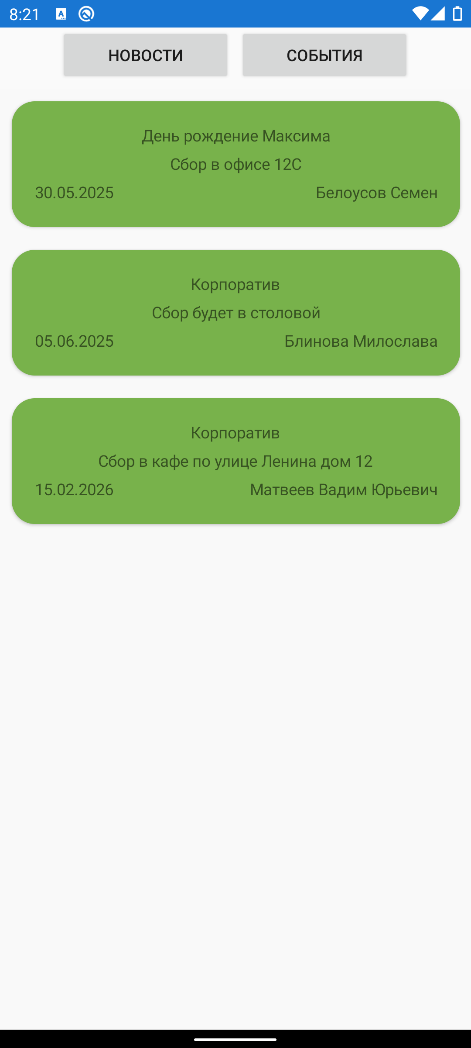


Рисунок 33 – Открытая вкладка «События»

На вкладке новости при нажатии на новость откроется окно с выбором реакции на новость (Рисунок 34). После выбора реакции счетчик реакций будет обновлён.

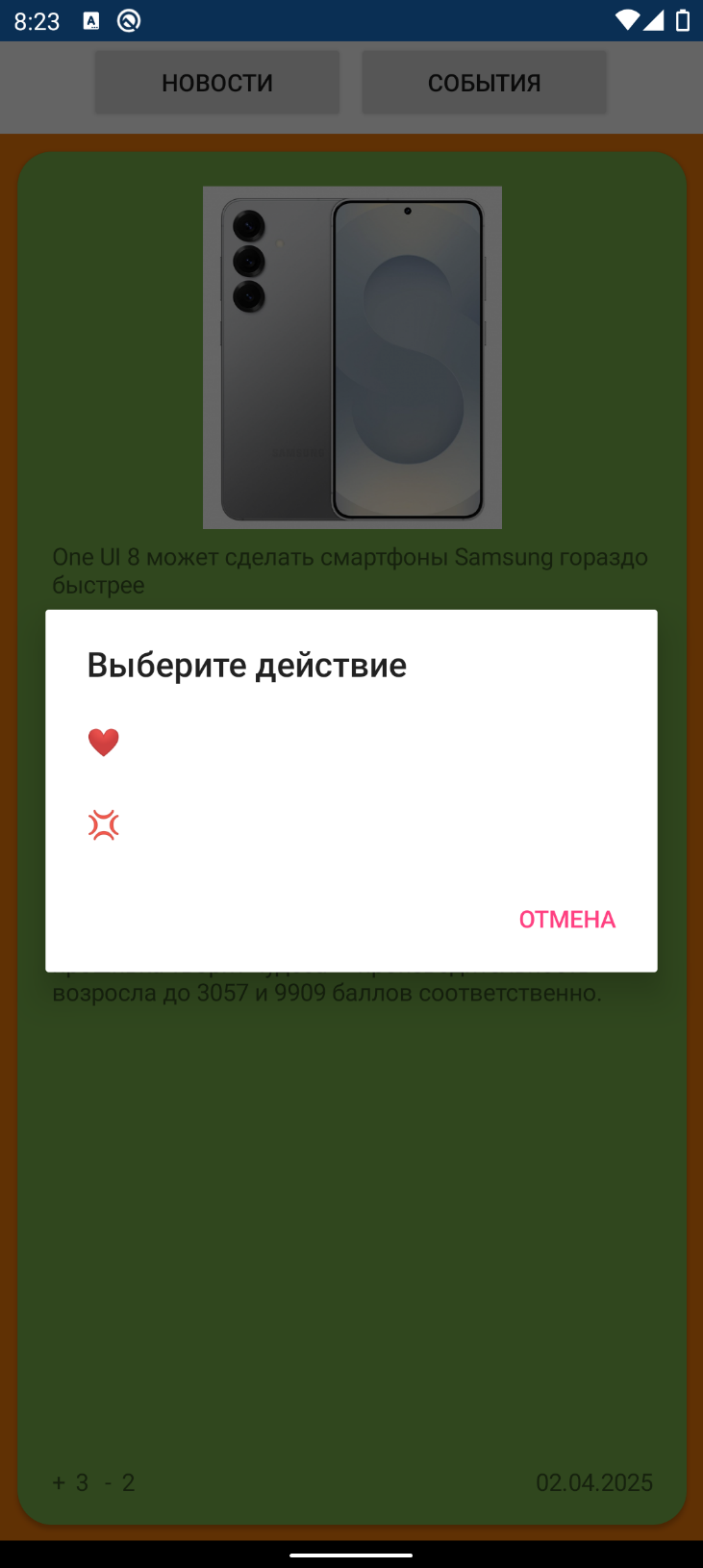


Рисунок 34 – Окно выбора реакции

На вкладке события при нажатии на событие оно будет автоматически добавлено в календарь устройства (Рисунок 35).



Рисунок 35 – Системный календарь

# Заключение

В период с 20 марта по 21 апреля 2025 года я проходил производственную практику по специальности 09.02.07 Информационные системы и программирование по модулю ПМ.01 «Разработка модулей программного обеспечения для компьютерных систем» в КОГОБУ СШ пгт. Вахруши Слободского района.

Перед началом производственной практики я был ознакомлен с техникой безопасности в кабинете и при работе с оборудованием.

При прохождении производственной практики я применил теоретические и практические навыки разработки, тестирования, рефакторинга и оптимизации программных модулей.

При прохождении производственной практики я разработал программные модули, которые помогут организации увеличить эффективность управления персоналом и поддержания сотрудников в курсе последних новостей. После разработки программных модулей было проведено тестирование с целью выявления ошибок, допущенных при написание модулей. На основе данных тестирования был проведен рефакторинг и оптимизация кода. На протяжение практики промежуточные результаты разработки программных модулей сохранялись в Git репозиторий.

# Приложения

**Приложение 1**

Документы и разработанное программное обеспечение можно просмотреть в облачном хранилище – https://cloud.mail.ru/public/DAa8/qmzB5CJ7q.

Также материалы практики можно просмотреть в Git репозитории – https://github.com/ALTERRA4546/p1mpp.

**Приложение 2**

Тест кейсы интеграционного тестирования:

|  |  |
| --- | --- |
| TestCase # | Division\_1 |
| Приоритет | Высокий |
| Название тестирования/Имя | Загрузка подразделений |
| Резюме испытания | Корректная загрузка всех подразделений с иерархией в список |
| Шаги тестирования | 1. Раскрыть первый элемент списка |
| Данные тестирования | Подразделений организации |
| Ожидаемый результат | Присутствуют все подразделения с иерархией |
| Фактический результат | Присутствуют все подразделения с иерархией |
| Предпосылки | Запуск приложения |
| Постусловия |  |
| Статус (Pass/Fail) | Pass |
| Комментарии |  |

|  |  |
| --- | --- |
| TestCase # | Employee\_1 |
| Приоритет | Высокий |
| Название тестирования/Имя | Загрузка сотрудников подразделения |
| Резюме испытания | Корректная загрузка всех сотрудников выбранного подразделения и дочерних подразделений |
| Шаги тестирования | 1. Раскрыть первое подразделение 2. Нажать на любое подразделение |
| Данные тестирования | Подразделений и сотрудники организации |
| Ожидаемый результат | Присутствуют все сотрудники, работающие в выбранном подразделение и дочерних подразделениях |
| Фактический результат | Присутствуют все сотрудники, работающие в выбранном подразделение и дочерних подразделениях |
| Предпосылки | Запуск приложения |
| Постусловия |  |
| Статус (Pass/Fail) | Pass |
| Комментарии |  |

|  |  |
| --- | --- |
| TestCase # | Employee\_Card\_1 |
| Приоритет | Высокий |
| Название тестирования/Имя | Загрузка данных сотрудника |
| Резюме испытания | Корректная загрузка данных сотрудника |
| Шаги тестирования | 1. Раскрыть первое подразделение 2. Нажать на любое подразделение 3. Дважды нажать на любого сотрудника |
| Данные тестирования | Подразделений и сотрудники организации |
| Ожидаемый результат | Данные сотрудника заполнены |
| Фактический результат | Данные сотрудника заполнены |
| Предпосылки | Запуск приложения |
| Постусловия |  |
| Статус (Pass/Fail) | Pass |
| Комментарии |  |

|  |  |
| --- | --- |
| TestCase # | Save\_Employee\_1 |
| Приоритет | Высокий |
| Название тестирования/Имя | Сохранение изменённых данных сотрудника |
| Резюме испытания | Проверка корректности сохранённых данных сотрудника |
| Шаги тестирования | 1. Изменение данных сотрудника 2. Нажатие на кнопку Сохранить |
| Данные тестирования | Фамилия: Рогов  Имя: Май  Отчество: Филатович  Мобильный телефон:  День рождения: 06.04.1967  Структурное подразделение: Управление безопасностью  Должность: Специалист  Руководитель:  Помощник:  Рабочий телефон: +7 (046) 142-05-62  Электронная почта: [рогов@гкдр.ру](mailto:рогов@гкдр.ру)  Кабинет: 489  Прочая информация: |
| Ожидаемый результат | Данные сотрудника успешно сохранились |
| Фактический результат | Данные сотрудника успешно сохранились |
| Предпосылки | Открыта карточка сотрудника |
| Постусловия | Сообщение о успешном сохранение |
| Статус (Pass/Fail) | Pass |
| Комментарии |  |

|  |  |
| --- | --- |
| TestCase # | Save\_Employee\_2 |
| Приоритет | Высокий |
| Название тестирования/Имя | Добавление нового сотрудника |
| Резюме испытания | Проверка корректности добавления нового сотрудника |
| Шаги тестирования | 1. Нажатие на кнопку «+» 2. Заполнение данных о новом сотруднике 3. Нажатие на кнопку «Сохранить» |
| Данные тестирования | Фамилия: Иванов  Имя: Иван  Отчество: Витальевич  Мобильный телефон: +7 (488) 888-78-78  День рождения: 06.04.1962  Структурное подразделение: Управление безопасностью  Должность: Курьер  Руководитель:  Помощник:  Рабочий телефон: +7 (046) 758-14-62  Электронная почта: [иванов@гкдр.ру](mailto:иванов@гкдр.ру)  Кабинет: 489  Прочая информация: |
| Ожидаемый результат | Данные сотрудника успешно сохранились |
| Фактический результат | Данные сотрудника успешно сохранились |
| Предпосылки | Запуск приложения |
| Постусловия | Сообщение о успешном сохранение |
| Статус (Pass/Fail) | Pass |
| Комментарии |  |

|  |  |
| --- | --- |
| TestCase # | Add\_event\_1 |
| Приоритет | Высокий |
| Название тестирования/Имя | Добавление события обучение |
| Резюме испытания | Проверка корректности добавления события обучение |
| Шаги тестирования | 1. Нажатие на кнопку «Добавить событие» 2. Заполнение данных о событие 3. Нажатие на кнопку «Сохранить» |
| Данные тестирования | Тип события: Обучение  Дата начала: 03.04.2025  Дата окончания: 15.05.2025  Название: Повышение квалификации сотрудника  Описание: Обучение в ДВФУ |
| Ожидаемый результат | Событие успешно добавилось |
| Фактический результат | Событие успешно добавилось |
| Предпосылки | Открытие карточки сотрудника |
| Постусловия | Сообщение о успешном сохранение |
| Статус (Pass/Fail) | Pass |
| Комментарии |  |

|  |  |
| --- | --- |
| TestCase # | Add\_event\_2 |
| Приоритет | Высокий |
| Название тестирования/Имя | Добавление события временное отсутствие |
| Резюме испытания | Проверка корректности добавления события временное отсутствие |
| Шаги тестирования | 1. Нажатие на кнопку «Добавить событие» 2. Заполнение данных о событие 3. Нажатие на кнопку «Сохранить» |
| Данные тестирования | Тип события: Временное отсутствие  Дата начала: 03.04.2025  Дата окончания: 15.05.2025  Название: Опоздание на работу  Описание: Опоздание на 5 часов |
| Ожидаемый результат | Событие успешно добавилось |
| Фактический результат | Событие успешно добавилось |
| Предпосылки | Открытие карточки сотрудника |
| Постусловия | Сообщение о успешном сохранение |
| Статус (Pass/Fail) | Pass |
| Комментарии |  |

|  |  |
| --- | --- |
| TestCase # | Add\_event\_3 |
| Приоритет | Высокий |
| Название тестирования/Имя | Добавление события отпуск |
| Резюме испытания | Проверка корректности добавления события отпуск |
| Шаги тестирования | 1. Нажатие на кнопку «Добавить событие» 2. Заполнение данных о событие 3. Нажатие на кнопку «Сохранить» |
| Данные тестирования | Тип события: Отпуск  Дата начала: 03.04.2025  Дата окончания: 15.05.2025  Название: Поездка в Сочи  Описание: Отпуск с семьей в Сочи |
| Ожидаемый результат | Событие успешно добавилось |
| Фактический результат | Событие успешно добавилось |
| Предпосылки | Открытие карточки сотрудника |
| Постусловия | Сообщение о успешном сохранение |
| Статус (Pass/Fail) | Pass |
| Комментарии |  |

|  |  |
| --- | --- |
| TestCase # | Show\_Events\_1 |
| Приоритет | Высокий |
| Название тестирования/Имя | Загрузка событий |
| Резюме испытания | Корректная загрузка событий сотрудника в списки |
| Шаги тестирования |  |
| Данные тестирования | Данные сотрудника и его события |
| Ожидаемый результат | Корректное отображение событий |
| Фактический результат | Корректное отображение событий |
| Предпосылки | Открытие карточки сотрудника |
| Постусловия |  |
| Статус (Pass/Fail) | Pass |
| Комментарии |  |

|  |  |
| --- | --- |
| TestCase # | Filtering\_Event\_1 |
| Приоритет | Высокий |
| Название тестирования/Имя | Фильтрация событий на прошедшие |
| Резюме испытания | Корректная фильтрация событий по прошедшим |
| Шаги тестирования | 1. Нажатие на кнопку «Прошедшее» |
| Данные тестирования | Данные сотрудника и его события |
| Ожидаемый результат | Отображение событий, которые прошли |
| Фактический результат | Отображение событий, которые прошли |
| Предпосылки | Открытие карточки сотрудника |
| Постусловия |  |
| Статус (Pass/Fail) | Pass |
| Комментарии |  |

|  |  |
| --- | --- |
| TestCase # | Filtering\_Event\_2 |
| Приоритет | Высокий |
| Название тестирования/Имя | Фильтрация событий на текущие |
| Резюме испытания | Корректная фильтрация событий по текущим |
| Шаги тестирования | 1. Нажатие на кнопку «Текущее» |
| Данные тестирования | Данные сотрудника и его события |
| Ожидаемый результат | Отображение актуальных событий |
| Фактический результат | Отображение актуальных событий |
| Предпосылки | Открытие карточки сотрудника |
| Постусловия |  |
| Статус (Pass/Fail) | Pass |
| Комментарии |  |

|  |  |
| --- | --- |
| TestCase # | Filtering\_Event\_3 |
| Приоритет | Высокий |
| Название тестирования/Имя | Фильтрация событий на будущее |
| Резюме испытания | Корректная фильтрация событий по будущим |
| Шаги тестирования | 1. Нажатие на кнопку «Будущее» |
| Данные тестирования | Данные сотрудника и его события |
| Ожидаемый результат | Отображение будущих событий |
| Фактический результат | Отображение будущих событий |
| Предпосылки | Открытие карточки сотрудника |
| Постусловия |  |
| Статус (Pass/Fail) | Pass |
| Комментарии |  |

|  |  |
| --- | --- |
| TestCase # | Remove\_Event\_1 |
| Приоритет | Высокий |
| Название тестирования/Имя | Удаление события обучение |
| Резюме испытания | Проверка удаления события обучение |
| Шаги тестирования | 1. Выбрать удаляемое событие 2. Нажать на кнопку «Удалить» |
| Данные тестирования | Данные сотрудника и его события |
| Ожидаемый результат | Сообщение о успешном удаление |
| Фактический результат | Сообщение о успешном удаление |
| Предпосылки | Открытие карточки сотрудника |
| Постусловия | Сообщение о успешном увольнение |
| Статус (Pass/Fail) | Pass |
| Комментарии |  |

|  |  |
| --- | --- |
| TestCase # | Remove\_Event\_2 |
| Приоритет | Высокий |
| Название тестирования/Имя | Удаление события временное отсутствие |
| Резюме испытания | Проверка удаления события временное отсутствие |
| Шаги тестирования | 1. Выбрать удаляемое событие 2. Нажать на кнопку «Удалить» |
| Данные тестирования | Данные сотрудника и его события |
| Ожидаемый результат | Сообщение о успешном удаление |
| Фактический результат | Сообщение о успешном удаление |
| Предпосылки | Открытие карточки сотрудника |
| Постусловия | Сообщение о успешном увольнение |
| Статус (Pass/Fail) | Pass |
| Комментарии |  |

|  |  |
| --- | --- |
| TestCase # | Remove\_Event\_3 |
| Приоритет | Высокий |
| Название тестирования/Имя | Удаление события отпуск |
| Резюме испытания | Проверка удаления события отпуск |
| Шаги тестирования | 1. Выбрать удаляемое событие 2. Нажать на кнопку «Удалить» |
| Данные тестирования | Данные сотрудника и его события |
| Ожидаемый результат | Сообщение о успешном удаление |
| Фактический результат | Сообщение о успешном удаление |
| Предпосылки | Открытие карточки сотрудника |
| Постусловия | Сообщение о успешном увольнение |
| Статус (Pass/Fail) | Pass |
| Комментарии |  |

|  |  |
| --- | --- |
| TestCase # | Fired\_Employee |
| Приоритет | Высокий |
| Название тестирования/Имя | Увольнение сотрудника |
| Резюме испытания | Проверка увольнения сотрудника |
| Шаги тестирования | 1. Нажать на кнопку «Уволить» |
| Данные тестирования | Фамилия: Рогов  Имя: Май  Отчество: Филатович  Мобильный телефон:  День рождения: 06.04.1967  Структурное подразделение: Управление безопасностью  Должность: Специалист  Руководитель:  Помощник:  Рабочий телефон: +7 (046) 142-05-62  Электронная почта: [рогов@гкдр.ру](mailto:рогов@гкдр.ру)  Кабинет: 489  Прочая информация: |
| Ожидаемый результат | Сообщение о успешном увольнение |
| Фактический результат | Сообщение о успешном увольнение |
| Предпосылки | Открытие карточки сотрудника |
| Постусловия | Сообщение о успешном увольнение |
| Статус (Pass/Fail) | Pass |
| Комментарии |  |

**Приложение 3**

Тест кейсы тестирования интерфейса:

|  |  |
| --- | --- |
| TestCase # | ScreenRotation\_1 |
| Приоритет | Низкий |
| Название тестирования/Имя | Смена ориентации экрана на горизонтальный во вкладке «Новости» |
| Резюме испытания | Поворот устройства на 90 влево |
| Шаги тестирования | 1. Перевернуть устройство на 90 влево |
| Данные тестирования | Новости организации |
| Ожидаемый результат | Переворот элементов на 90 влево |
| Фактический результат | Переворот элементов на 90 влево |
| Предпосылки | Открытие вкладки «Новости» |
| Постусловия | Обновление интерфейса |
| Статус (Pass/Fail) | Pass |
| Комментарии |  |

|  |  |
| --- | --- |
| TestCase # | ScreenRotation\_2 |
| Приоритет | Низкий |
| Название тестирования/Имя | Смена ориентации экрана на горизонтальный во вкладке «События» |
| Резюме испытания | Поворот устройства на 90 влево |
| Шаги тестирования | 1. Перевернуть устройство на 90 влево |
| Данные тестирования | События организации |
| Ожидаемый результат | Переворот элементов на 90 влево |
| Фактический результат | Переворот элементов на 90 влево |
| Предпосылки | Открытие вкладки «События» |
| Постусловия | Обновление интерфейса |
| Статус (Pass/Fail) | Pass |
| Комментарии |  |

|  |  |
| --- | --- |
| TestCase # | ScreenRotation\_3 |
| Приоритет | Средний |
| Название тестирования/Имя | Правильное отображение новостей при повороте экрана в горизонтальный режим |
| Резюме испытания | Проверка правильности отображения новостей после смены ориентации экрана на горизонтальный |
| Шаги тестирования | 1. Перевернуть устройство на 90 влево |
| Данные тестирования | Новости организации |
| Ожидаемый результат | Элементы интерфейса не наслаиваются друг на друга |
| Фактический результат | Описание наслаивается на реакции и дату публикацию |
| Предпосылки | Открытие вкладки «Новости» |
| Постусловия | Обновление интерфейса |
| Статус (Pass/Fail) | Fail |
| Комментарии |  |

|  |  |
| --- | --- |
| TestCase # | ScreenRotation\_4 |
| Приоритет | Средний |
| Название тестирования/Имя | Правильное отображение событий при повороте экрана в горизонтальный режим |
| Резюме испытания | Проверка правильности отображения событий после смены ориентации экрана на горизонтальный |
| Шаги тестирования | 1. Перевернуть устройство на 90 влево |
| Данные тестирования | События организации |
| Ожидаемый результат | Элементы интерфейса не наслаиваются друг на друга |
| Фактический результат | Элементы интерфейса не наслаиваются друг на друга |
| Предпосылки | Открытие вкладки «События» |
| Постусловия | Обновление интерфейса |
| Статус (Pass/Fail) | Pass |
| Комментарии |  |

|  |  |
| --- | --- |
| TestCase # | DataOverflow\_1 |
| Приоритет | Средний |
| Название тестирования/Имя | Правильное отображение новостей при переполнение данными полей |
| Резюме испытания | Проверка правильности отображения новостей при переполнение данными полей |
| Шаги тестирования |  |
| Данные тестирования | Новости организации |
| Ожидаемый результат | Элементы интерфейса не наслаиваются друг на друга |
| Фактический результат | Описание наслаивается на реакции и дату публикацию |
| Предпосылки | Открытие вкладки «Новости» |
| Постусловия | Обновление интерфейса |
| Статус (Pass/Fail) | Fail |
| Комментарии |  |

|  |  |
| --- | --- |
| TestCase # | DataOverflow\_2 |
| Приоритет | Средний |
| Название тестирования/Имя | Правильное отображение событий при переполнение данными полей |
| Резюме испытания | Проверка правильности отображения событий при переполнение данными полей |
| Шаги тестирования |  |
| Данные тестирования | События организации |
| Ожидаемый результат | Элементы интерфейса не наслаиваются друг на друга |
| Фактический результат | Элементы интерфейса не наслаиваются друг на друга |
| Предпосылки | Открытие вкладки «События» |
| Постусловия | Обновление интерфейса |
| Статус (Pass/Fail) | Pass |
| Комментарии |  |

|  |  |
| --- | --- |
| TestCase # | ScrollingList\_1 |
| Приоритет | Высокий |
| Название тестирования/Имя | Пролистывание списка новостей |
| Резюме испытания | Проверка пролистывания списка новостей |
| Шаги тестирования | 1. Пролистывание списка до упора влево 2. Пролистывание списка до упора вправо |
| Данные тестирования | Новости организации |
| Ожидаемый результат | Отсутствие артефактов и дефектов отображения новостей |
| Фактический результат | Отсутствие артефактов и дефектов отображения новостей |
| Предпосылки | Открытие вкладки «Новости» |
| Постусловия | Обновление интерфейса |
| Статус (Pass/Fail) | Pass |
| Комментарии |  |

|  |  |
| --- | --- |
| TestCase # | ScrollingList\_2 |
| Приоритет | Высокий |
| Название тестирования/Имя | Пролистывание списка событий |
| Резюме испытания | Проверка пролистывания списка событий |
| Шаги тестирования | 1. Пролистывание списка до упора влево 2. Пролистывание списка до упора вправо |
| Данные тестирования | События организации |
| Ожидаемый результат | Отсутствие артефактов и дефектов отображения новостей |
| Фактический результат | Отсутствие артефактов и дефектов отображения новостей |
| Предпосылки | Открытие вкладки «События» |
| Постусловия | Обновление интерфейса |
| Статус (Pass/Fail) | Pass |
| Комментарии |  |