

**КУРСОВОЙ ПРОЕКТ**

**По дисциплине: МДК 01.01 Разработка программных модулей**

**Тема: Разработка модуля «Работа с участниками» приложения «Молодежный форум»**

**Специальность 09.02.07 «Информационные системы и программирование»**

|  |  |  |
| --- | --- | --- |
| **Выполнила студентка группы 309ИС-22** | **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** | **Е.С. Зайцева** |
| **Руководитель** | **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** | **Л.Б. Гусятинер** |

**Москва 2024**

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc185521201)

[1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ И ОБОСНОВАНИЕ АКТУАЛЬНОСТИ ПРОЕКТА. 5](#_Toc185521202)

[1.1 Информационное обеспечение задачи и обоснование актуальности проекта 5](#_Toc185521203)

[1.2 Обзор и анализ существующих программных решений 6](#_Toc185521204)

[1.3 Процессы, представляемые к автоматизации и требования к функциональным характеристикам 7](#_Toc185521205)

[1.3.1 Постановка задачи 9](#_Toc185521206)

[1.3.2 Структура входной и выходной информации 10](#_Toc185521207)

[1.4 Выбор и обоснование выбора языка программирования, среды разработки и жизненного цикла ПО 10](#_Toc185521208)

[2. ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ ПРОГРАММНОГО МОДУЛЯ 13](#_Toc185521209)

[2.1 Построение модели системы 13](#_Toc185521210)

[2.1.1 Проектирование общей структуры программного продукта 13](#_Toc185521211)

[2.1.2 Описание сценария работы программы 15](#_Toc185521212)

[2.2 Разработка программного продукта 17](#_Toc185521213)

[2.2.1 Разработка и описание главного модуля 17](#_Toc185521214)

[2.2.2 Спецификация модулей 18](#_Toc185521215)

[2.2.3 Описание модулей 19](#_Toc185521216)

[2.3 Расчет сложности алгоритма 21](#_Toc185521217)

[3. ОТЛАДКА И ТЕСТИРОВАНИЕ МОДУЛЯ 22](#_Toc185521218)

[3.1 Описание тестовых наборов модулей 22](#_Toc185521219)

[Таблица 2.1Список функций, представляемых проверке в программном модуле и их расположение 22](#_Toc185521220)

[3.2 Описание применения средств отладки 23](#_Toc185521221)

[ЗАКЛЮЧЕНИЕ 26](#_Toc185521222)

ВВЕДЕНИЕ

Тема курсового проекта: Разработка программного модуля «Работа с участниками» для приложения «Молодежный форум».

Актуальность данной темы курсового проекта обусловлена динамикой потребности в автоматизации процессов управления в сфере организации мероприятий в целом и процессом работы с участниками мероприятия, в частности.

В современном мире, где значительная часть коммуникации, особенно после пандемии COVID-19, осуществляется посредством сети Интернет, сложно переоценить важность и эффективность живого общения между людьми. Поэтому многие организации проводят очные встречи для обмена опытом, знаниями и практиками.

При подготовке и проведении подобных мероприятий необходимо учитывать множество процессов, ведь впечатление от посещения события, напрямую влияет на имидж компании.

Однако, не всегда представляется возможным найти необходимое количество людей, для реализации мероприятия. Преодоление этой проблемы затрудняется ограниченным бюджетом и недостаточным количеством времени для обучения помощников.

Решением может стать применение информационных технологий, которые способны автоматизировать часть процессов, что позволит сократить состав штаба организаторов и время, затрачиваемое на подготовку.

Для форумов, относящихся к категории «молодежный», особенно важно наличие инновационных технологий в процессе участия. Молодежь с большим интересом относится к событию, если на нем присутствуют программные решения. Внедрение новых способов выполнения задач, также является фактором, повышающим лояльность участников.

Это выгодно проектам, которые стремятся проводить более масштабные мероприятия с регулярной частотой или с меньшими экономическими затратами.

Исходя из вышеперечисленных фактов, можно выделить основную цель и задачи проекта:

**Цель выполнения проекта** – разработка программного модуля приложения для автоматизации молодежного мероприятия.

**Объект исследования** – организация мероприятий.

**Предмет исследования** – процесс работы с участниками на молодежном форуме.

Для достижения поставленной цели курсового проекта необходимо решить следующие **задачи:**

- Проанализировать предметную область и обосновать актуальность проекта.

- Изучить существующие программные решения.

- Выделить процессы, которые необходимо автоматизировать.

- Выбрать язык программирования, среду разработки и ЖЦ ПО.

- Спроектировать общую структуру программного продукта.

- Продумать необходимые функции и сценарии работы программы.

- Разработать программное обеспечение.

- Провести отладку и тестирование программного обеспечения.

- Разработать программную документацию для сопровождения программного обеспечения.

- Подготовить презентацию проекта для защиты перед потенциальными инвесторами или покупателями.

1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ И ОБОСНОВАНИЕ АКТУАЛЬНОСТИ ПРОЕКТА.

1.1 Информационное обеспечение задачи и обоснование актуальности проекта

В данном курсовом проекте предметной областью для решения поставленных задач является организация молодежного форума.

**Форум** *–* это мероприятие, на котором люди собираются вместе для обсуждения определенной темы или проблемы. [1] А также для обмена знаниями, опытом и идеями, образования деловых связей, поиска инвесторов, обретения полезных знакомств или получения гранта на реализацию проекта.

Мероприятия данного типа проводятся повсюду: в больших корпорациях, в государственных органах власти, в школах и вузах; среди работников медицины, культуры, педагогики или бизнеса. Тематика и масштаб события может быть абсолютно разным.

К числу известных форумов относятся Петербургский международный экономический форум (ПМЭФ), Всероссийский молодежный форум «Машук» и др. Также, например, в рамках деятельности нашего колледжа, студенты и администрация регулярно устраивают различные форумы (II Межрегиональный форум «Предпринимательство: новые идеи молодежи»; Форум «Волонтерство в СПО» и др.).

*Основные посетители молодежного форума* – это люди в возрасте от 14 до 35 лет, каждый из которых неравнодушно относится к теме форума, хочет узнать что-то новое или поделится своими знаниями.

«Проведение масштабных «форумных кампаний» в России началось сравнительно недавно, но они активно популяризируются в молодежной среде, развиваются старые и появляются новые площадки, которые ежегодно посещают тысячи молодых людей.

С 2015 г. постоянно растет количество участников форумов, что говорит об охвате все большего количества молодежи и профильной работы с ней. В 2015 г. в молодежных форумах России приняли участие около 170 000 чел., в 2016 г.— 360 000 чел., в 2017 г.— 500 000 чел., в 2018 г.— 800 000 чел., в 2019 г.— 900 000 чел.» [2] Наглядное представление тенденции роста количества участников представлено на диаграмме:

Рисунок 1.1.1 – Динамика количества участников молодежной «форумной кампании»

Рост количества участников свидетельствует о необходимости внедрения новых решений в существующую организационную структуру, для поддержания высокого уровня обслуживания при увеличении числа посетителей.

У каждого форума есть собственный стиль или брендбук, который регламентируют дизайн афиш, мерча и помещения, поэтому важно учитывать пожелания заказчиков при создании шаблонов. Это влияет на уникальность и узнаваемость проекта.

1.2 Обзор и анализ существующих программных решений

Анализ существующих решений позволил найти сервис, предоставляющий схожие с планируемым функционалом программные решения. Сервис REGPART, команда которого с 2013 года занимается автоматизацией регистрации участников различных мероприятий. [4]

Также практически идеальным аналогом разрабатываемого модуля является ФГАИС «Молодежь России».[5]

Идеи для разработки ПО также можно найти, если изучить функционал различных CRM-систем, которые существуют для управления всевозможными процессами в бизнесе.

**CRM-система** — это программа для автоматизации и контроля взаимодействия компании с клиентами. Она хранит и структурирует информацию о заказах и покупателях, помогает оптимизировать маркетинг, повысить продажи и качество обслуживания. [6]

Дополнительная информация о проанализированных программных продуктах, находится в Приложении 3.

1.3 Процессы, представляемые к автоматизации и требования к функциональным характеристикам

Организация мероприятия представляет собой многоступенчатый процесс, требующий от дирекции комплексного подхода и координации следующих аспектов:

- Взаимодействие с поставщиками – обеспечение закупок надлежащего качества и исполнения установленных сроков.

- Закупка и выдача мерча – разработка дизайна, обеспечение достаточного количества продукции для всех участников.

- Организация места проведения форума – поиск и аренда подходящей площадки и необходимого оборудования.

- Составление программы мероприятия – составление детального плана, включающего все запланированные сессии, выступления и активности.

- Приглашение и согласование выступающих спикеров.

- Управление регистрацией участников (при наличии конкурса на участие, включает в себя отбор поступающих заявок).

Возникает необходимость в компьютерной обработке данных, поступающих от разных категорий участников и организаторов, с целью централизованного хранения информации и дальнейшего анализа эффективности мероприятия.

**Управление регистрацией участников** — это комплексный процесс, включающий в себя все действия, необходимые для эффективного сбора, обработки и управления информацией об участниках мероприятия.

Варианты регистрации и пропуска участников на событие:

*- Фактически свободный вход* — когда участники формально проходят через некую входную группу, но их проверяет только служба безопасности. Могут возникнуть недопонимания с партнерами, из-за отсутствия системы отслеживания количества посетителей.

*- Проход по билетам через волонтеров* — когда на входе стоят члены команды и участники предъявляют им билеты (распечатанные или в электронном виде).

*- Получение бейджа из заранее напечатанных.* На основании данных из своей базы организаторы печатают бейджи или подписывают их вручную. На стойке регистрации участник называет свою фамилию, по ней в стопках находят его бейдж, и на основании этого он проходит на мероприятие. Однако, при таком варианте регистрации, кто-то может назвать чужую фамилию и пройти на мероприятие.

*- Печать бейджа на месте в обмен на предъявление документа, удостоверяющего личность*. Волонтер сверяется со списками, загруженными в компьютер на стойке регистрации, и, найдя нужную фамилию, печатает бейдж.

*- Самостоятельная регистрация* — сервис, где участник прикладывает штрихкод к считывателю или вводит цифровой код из билета и получает бейдж, распечатанный волонтером или автоматически.

*- Регистрация через приложение* — участник скачивает специально созданное для мероприятия приложение и после покупки билета проходит в нем регистрацию. На мероприятии через него он может распечатать бейдж. [5]

Последние два варианта наиболее предпочтительные, но при этом дорогостоящие, позволить себе их могут не все организации, что ещё раз показывает актуальность создания простого программного модуля, способного решить эту проблему и повысить количество компаний, использующих инновационные технологии.

1.3.1 Постановка задачи

Разрабатываемый программный модуль должен обеспечивать, реализацию следующих процессов:

*- Регистрация:* сбор и анализ личных данных участника необходимых для подтверждения личности и формирования итогового отчета о посещении мероприятия (Фамилия, имя и отчество, дата рождения, половая принадлежность, номер телефона, имя пользователя в социальной сети «Telegram»)

*- Подтверждение финальной явки на мероприятие и отслеживание текущего количества участников*

*- Визуализация информации об участниках:* этот процесс важен не только для составления отчета, но и для формирования представления о количестве участников разных категорий. Необходимо иметь возможность реализовать передачу этой информации другим организаторам или, в последствии, другим модулям программы. (Получение статистики обязательно, если на форуме предусмотрена выдача индивидуального комплекта мерча для каждого участника.)

*- Выдача индивидуального бейджа:* наличие бейджа поможет людям правильно обращаться друг к другу и понимать роль человека на мероприятии, упросит взаимодействие во время нетворкинга, а также позволит подтвердить свое участие, при выходе с площадки проведения форума.

Более подробно все процессы представлены на DFD – диаграмме потоков (Приложение 1, Рис.1.2)

1.3.2 Структура входной и выходной информации

Исходные данные поступают в систему с помощью ввода пользователем данных о пришедших участниках или из заранее составленного списка зарегистрированных, тогда в обязанности пользователя будет входить проставление настоящего статуса участника каждого зарегистрированного. После анализа и обработки поступившей информации они сохраняются в системе классов и файлах с соответствующими названиями.

Выходными данными являются отчет с информацией участников, список заготовленных бейджей в текстовом формате и непосредственно бейдж зарегистрированного участника. (Приложение, рис.)

1.4 Выбор и обоснование выбора языка программирования, среды разработки и жизненного цикла ПО

Существует множество инструментов для создания программ, одних только языков программирования насчитывается более 700, например:

- Python: язык, который поддерживает процедурное, объектно-ориентированное и функциональное программирование. Он известен своей простотой и читабельностью, которая достигается за счёт явного синтаксиса и автоматического управления памятью.

- C++: преимущественно, объектно-ориентированный язык, но также поддерживает процедурное и функциональное программирование. Он имеет расширенный контроль над памятью и ресурсами, но также требует более сложного и подробного синтаксиса.

Для разработки программного модуля «Регистрация участников» выбран ***язык программирования Python*** и интегрированная среда разработки (IDE) PyCharm.

Выбор языка программирования Python обусловлен следующими факторами:

*- Высокая читаемость и простота:* Python отличается ясным и лаконичным синтаксисом, что упрощает разработку, отладку и поддержку кода. Это особенно важно для командной работы и дальнейшего развития модуля, как часть приложения.

*- Богатый набор библиотек*: это особенно актуально для реализации функциональности регистрации участников, которая может включать в себя взаимодействие с различными системами.

*- Большое сообщество разработчиков:* сообщество Python-разработчиков обеспечивает доступ к обширной документации, поддержке и готовым решениям

Проект будет разрабатываться в ***среде PyCharm***, т.к она многофункциональна, совместима с языком Python и имеет следующие преимущества:

*- Удобство использования*: PyCharm имеет интуитивно понятный интерфейс, что повышает производительность разработчиков. Например, среда разработки сама подскажет имя переменной и каким типом данных она является

*- Инструменты для работы с данными:* PyCharm интегрируется с различными системами управления базами данных (СУБД), что в последствии упростит работу с данными участников.

Для разрабатываемого программного модуля «работа с участниками» подходит ***V-образная модель жизненного цикла***, которая характеризуется параллельным выполнением этапов разработки и тестирования.

Эта модель обеспечивает строгую последовательность и верификацию каждого этапа разработки, что полезно при работе с ограниченным количеством разработчиков и четко обозначенными требованиями.

В рамках V-образной модели, каждый этап разработки имеет соответствующий этап тестирования. Только после успешного прохождения тестирования на каждом этапе, проект переходит к следующему.

V-образная модель обладает рядом преимуществ:

* *- Раннее выявление ошибок.*
* *- Повышенное качество ПО*.
* *- Четкая документация.*
* *- Лучшее понимание требований.*

V-образная модель обеспечивает более предсказуемый процесс разработки и гарантирует высокое качество конечного продукта. Поэтому с уверенностью можно сказать, что данная модель жизненного цикла подходит для разработки программного модуля «Работа с участниками».

Таким образом, проанализировав предметную область, существующие программные решения и предоставив информационное обеспечение задачи, были выявлены процессы, которые необходимо автоматизировать, используя выбранные программные средства, в рамках работы над данным проектом.

2. ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ ПРОГРАММНОГО МОДУЛЯ

2.1 Построение модели системы

2.1.1 Проектирование общей структуры программного продукта

В процессе проектирования основное внимание уделяется логическому решению, обеспечивающему выполнение основных требований.

В данном описании проектирования представлена консольная версия приложения, которая может быть использована, как бесплатная демоверсия при продаже продукта, в дальнейшем будет реализован дополнительный функционал, который будет являться Full-версией приложения «Молодежный форум»

Рассмотрим построение ***системы классов***, где будет хранится информация о зарегистрированных участниках, категориях участия и мероприятиях, организованных пользователем(администратором-организатором). Информация будет использоваться дирекцией, другими организаторами молодежного форума и, непосредственно, его посетителями.

Выделим следующие классы, их элементы и методы:

*- Участник,* основной класс используемый как шаблон для заполнения отчета по каждому посетителю: ID, фамилия, имя, отчество, дата рождения, пол, телефон, имя пользователя в телеграм (для связи и отображения на бейдже), статус присутствия (для мониторинга присутствующих участников). Каждый участник будет принадлежать к определенной категории.

- Класс *Категории* состоит из ID, названия категории и списка участников, которые ей принадлежат

*- Мероприятие,* класс от которого будут зависеть другие классы. Необходим для функции, осуществляющую доступ для работы с несколькими мероприятиями, как отдельными проектами.

- *Заявка:* класс, для хранения дополнительных данных об участнике

*- Бейдж:* вспомогательный класс, в котором хранится необходимая для генерации бейджей информация. Зависим от трех других классов

Анализ объектов позволяет выделить, основные классы: Участник, Категории, Мероприятие, которые могут стать основополагающими при создании базы данных для программного модуля приложения.

Связи, методы и атрибуты классов представлены на Диаграмме классов:

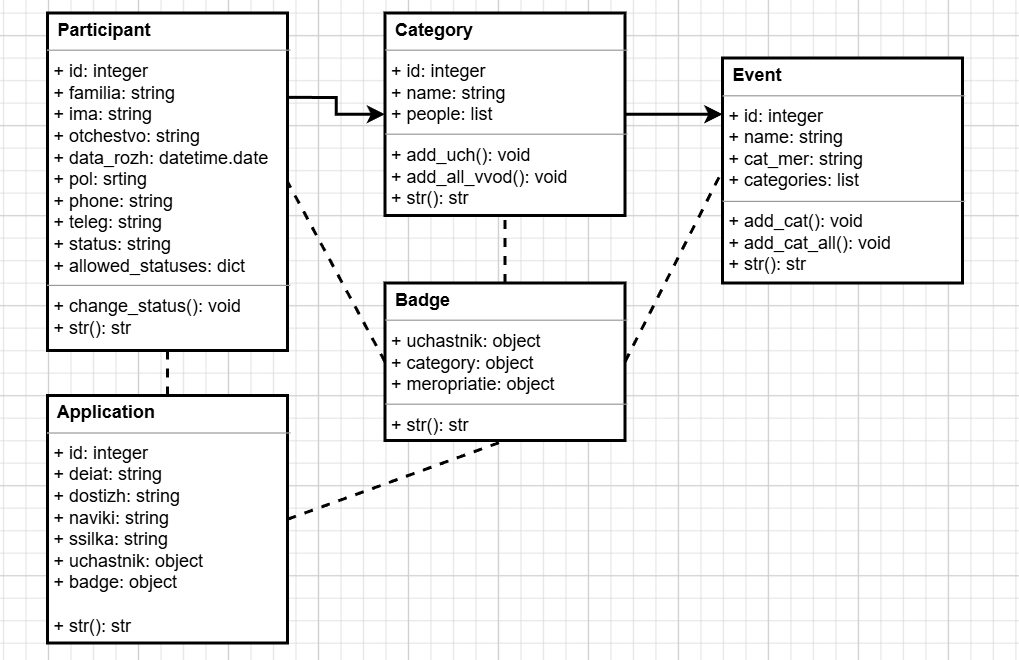


Рисунок 2.1 – Диаграмма классов

Для расширения возможностей приложения и создания базы данных, необходимо будет воспользоваться ER – диаграммой. (Приложение 1, рис.2.2)

В программном модуле предусмотрено наличие двух ролей пользователей, это «организатор», который выполняет роль администратора, имеет доступ к отчетам, файлам спискам и полному набору функций программы и «участник», задача которого, как пользователя программного продукта, пройти регистрацию и заполнить заявку на участие.

2.1.2 Описание сценария работы программы

Краткое описание сценария роли Участник:

* UC-1 Выбор мероприятия
* UC-2 Регистрация на мероприятие
* UC-3 Заполнение заявки
* UC-4 Получение бейджа

Таблица 2.1.2.1 – Основное описание сценария для пользователя Участник

|  |  |
| --- | --- |
| Пользователь | Система |
| Вводит название мероприятия в поле ввода | - Выводит название мероприятия  - Предоставляет доступ к регистрации |
| Вводит требуемые данные в поля ввода | - Сохраняет данные в личном кабинете и списке участников  - Предоставляет доступ к заполнению заявки |
| Заполняет заявку на участие | - Выводит документ с заполненными данными для организатора  - Сообщает пользователю об успешно заполненной заявке |
| Запрашивает вывод бейджа | - Передает данные организатору |

Краткое описание сценария роли Организатор:

* UC-1 Создание или выбор мероприятия
* UC-2 Ввод категорий участников
* UC-3 Получение отчета по каждой категории
* UC-4 Получение общего списка участников
* UC-5 Выдача бейджей на мероприятии

Таблица 2.1.2.2 – Основное описание сценария для пользователя Организатор

|  |  |
| --- | --- |
| Пользователь | Система |
| Вводит название мероприятия в поле ввода | - Выводит название мероприятия  - Предоставляет доступ к созданию категорий |
| Вводит данные о категориях через консоль или обращается к системе и файлам | - Сохраняет категории в список  - Предоставляет доступ к добавлению участников |
| Вводит данные нового участника (при необходимости) | - Передает данные в список участников |
| Запускает функцию вывода бейджа | - Экспортирует бейдж в Word |
| Запускает функцию вывода всех бейджей | - Выводит список участников в файл |
| Запускает функцию вывода списка | - Экспортирует список в Excel |
| Изменяет ID статуса участия | - Изменяет статус участия участика |

Более наглядно все процессы можно представить на диаграмме прецедентов:

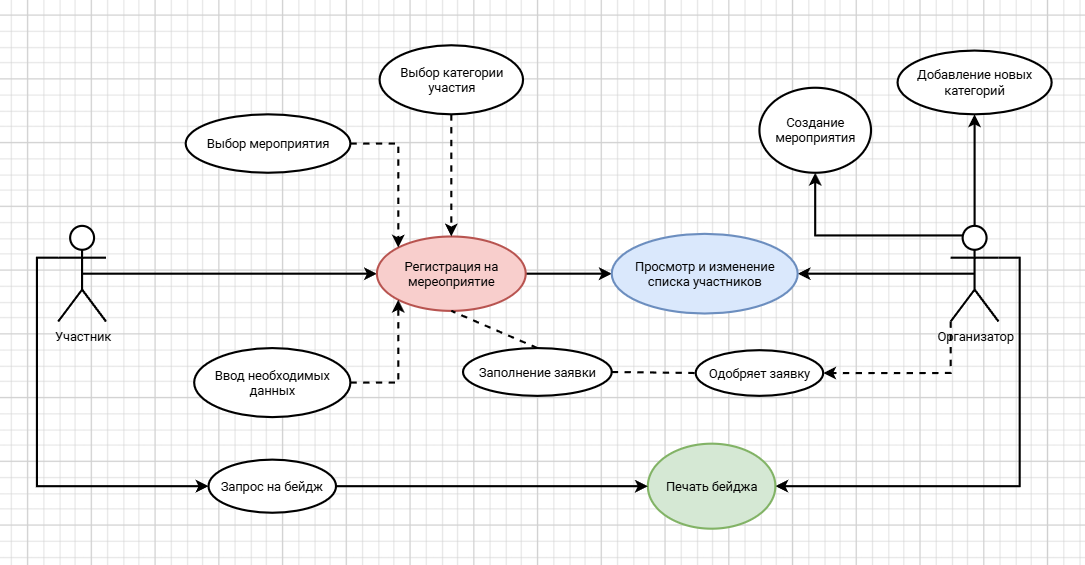


Рисунок 2.1.2.1 – Диаграмма прецедентов

В данной программе также могут существовать альтернативные сценарии. Т.к при проработке основного сценария, все варианты действий пользователя и поведения системы, отличных от основного сценария необходимо выносить в альтернативный сценарий. То есть, везде, где можно будет указать «если» –будет создан альтернативный сценарий.

«Альтернативный сценарий должен ссылаться только на один успешный сценарий. Недопустимо прописывать альтернативный сценарий для альтернативного сценария.» [8]

2.2 Разработка программного продукта

2.2.1 Разработка и описание главного модуля

Главный модуль предназначен для управления данными о участниках, категориях, мероприятиях, бейджах и заявках. Он предоставляет консольный интерфейс для взаимодействия с пользователем, запускает функции и вызывает классы из других модулей программы, позволяя добавлять, изменять, удалять и просматривать информацию. Модуль использует классы Uchastnik, Category, Meropriatie, Badge, и Zaiavka для организации, хранения и экспорта данных.

Функциональность главного модуля:

Управление участниками (Uchastnik), категориями (Category), мероприятиями (Meropriatie), бейджами (Badge), заявками (Zaiavka)

* Экспорт данных: экспорт данных в форматы Excel и Word для дальнейшей обработки или отчетности.
* Интерфейс пользователя: Главный модуль использует консольный интерфейс (CLI), в дальнейшем – графический интерфейс (GUI) и веб-интерфейс для взаимодействия с пользователем.

Модуль также обеспечивает связь между другими модулями и классами. Например: Zaiavka будет содержать ссылки на объекты Uchastnik и Badge. А Badge будет содержать ссылки на объекты Uchastnik, Meropriatie, и Category.

Поэтому в начале кода данного модуля обязательно прописывается импорт из всех остальных файлов приложения:

from {Название файла с классом} import \*

для каждого модуля программы.

Глобальной переменный не объявлено, однако в нижеописанном модуле присутствует список MER = [], индексы и элементы которого используются во время работы всей программы.

2.2.2 Спецификация модулей

*- Модуль Participant.py*

Содержит класс Uchastnik для представления информации об участнике мероприятия. Предоставляет функции для изменения статуса участника.

*- Модуль Category.py*

Содержит класс Category для представления категорий участников мероприятия. Предоставляет функции для управления списками участников в этих категориях. Класс включает в себя методы для добавления участников (как напрямую через объекты, так и через ввод данных пользователем), а также методы для вывода информации о категории и её участниках. Основные функции: запись списка участников в файл, очистка файла и обновление списка.

*- Модуль Event.py*

Содержит класс Meropriatie для представления информации о мероприятиях. Модуль предоставляет функции для добавления категорий, объявления мероприятия и работы с файлами, содержащими информацию о категориях.

*- Модуль Badge.py*

Содержит класс Badge для вывода бейджей в файл и Word. Предоставляет функции для генерации и вывода информации о бейджах участников. Включает в себя функцию для записи информации о бейджах в текстовый файл, функцию для очистки файла и функцию для обновления файла с информацией о бейджах. Также модуль содержит функцию для генерации бейджа в формате Word документа.

*- Модуль Application.py*

Содержит класс Zaiavka для представления информации о заявках участников на мероприятие. Класс включает в себя атрибуты для хранения данных о заявке, таких как идентификатор заявки, информация о деятельности участника, его достижениях, навыках, ссылке на портфолио, а также ссылку на объект Uchastnik и необязательную ссылку на объект Badge.

2.2.3 Описание модулей

Основные функции, которые автоматизируют процессы при управлении регистрацией участников это Spisok, Generate\_badges, Badge\_w и Student\_excel

***- Функция Spisok (модуль Category)***

***Создает текстовый файл «spisok\_text.txt»***

# функция записывает в файл список участников по категории

def Spisok(ct, filename="spisok\_text.txt"):

try:

with open(filename, "a", encoding="utf-8") as file: #открытие файла с помощью конструкции with на добавление

file.write(f'{ct.name\_c}:\n')

for uchastnik in ct.people:

file.write(str(uchastnik) + "\n")

except Exception as fall: #обработка ошибки

print(f"Ошибка при записи в файл: {fall}")

***- Generate\_badges (модуль Badge):***

# функция для генерации бейджей  
def Generate\_badges(meropriatie, category):  
 with open('badges.txt', 'a', encoding='utf-8') as file: #открытие файла с помощью конструкции with на добавление for people in category.people:  
 badge = Badge(people, meropriatie, category)  
 file.write(str(badge) + '\n')

***- Badge\_w (Модуль Badge)***

Для работы функции Badge\_w необходимо установить библиотеку:

pip install docxtpl

А также прописать в коде программы импорт из этой библиотеки:

from docxtpl import DocxTemplate

Функция:

# функция выводит бейдж участника в Word  
def Badge\_w(MR):  
 doc = DocxTemplate("бейдж.docx")  
 c = int(input('Генерация бейджа. Введите ID категории: ')) - 1  
 u = int(input('Генерация бейджа. Введите ID участника: ')) - 1

context = {'name\_m': MR.name\_m, 'cat\_mer': MR.cat\_mer,  
 'familia': MR.categories[c].people[u].familia,'ima': MR.categories[c].people[u].ima , 'teleg': MR.categories[c].people[u].teleg, 'name\_c':MR.categories[c].name\_c}  
 doc.render(context)  
 doc.save("бейдж-final.docx")

***- Student\_to\_excel (Модуль Event)***

Для работы функции Student\_to\_excel необходимо установить библиотеку:

pip install openpyxl

А также прописать в коде программы импорт из этой библиотеки:

from openpyxl import Workbook

Функция:

def Students\_to\_excel(self):  
 try:  
 with open('spisok\_excel.txt', 'r', encoding='utf-8') as file:  
 lines = file.readlines()  
 wb = Workbook()  
 ws = wb.active  
 today = date.today()  
 ws['A1'] = f"Список участников на {today}"  
 ws['A1'].font = Font(bold=True, size=14)  
 ws['A1'].alignment = Alignment(horizontal='center')  
 ws.merge\_cells('A1:H1')  
 headers = ["ID", "Фамилия", "Имя", "Отчество", "Дата рождения", "Пол", "Телефон", "Телеграм", "Статус участия", "Категория"]  
 ws.append(headers)  
 for line in lines:  
 st = line.strip().split(' ')  
 ws.append(st)  
 wb.save('new\_excel.xlsx')  
 print(f"Данные экспортированы в файл new\_excel.xlsx")  
 except Exception as e:(  
 print(f"Ошибка: {e}"))

Подробнее работа модулей представлена с помощью блок-схем в Приложении

2.3 Расчет сложности алгоритма

***Расчёт сложности алгоритма*** — это оценка его производительности по времени выполнения или по используемой памяти. Например: O(1) –

константная сложность, O(n) – линейная сложность.[9]

Оценка сложности методов класса Uchastnik, модуль Partisipant:

\_\_init\_\_(self, id\_uch, familia, ima, otchestvo, data\_rozh, pol, phone, teleg, status="registered"): этот метод выполняет присваивание значений атрибутам объекта. Количество операций не зависит от размера входных данных, следовательно сложность – O(1).

change\_status(self, new\_status): этот метод выполняет поиск по словарю allowed\_statuses, который имеет фиксированный размер, поэтому сложность можно считать O(1).

По итогам проделанной работы, был создан программный модуль «Работа с участниками» приложения «Молодженый форум», отвечающий требованиям, заявленным в техническом задании.

3. ОТЛАДКА И ТЕСТИРОВАНИЕ МОДУЛЯ

3.1 Описание тестовых наборов модулей

Отладка кода предназначена для поиска ошибок и дефектов в коде программы.

Таблица 2.1Список функций, представляемых проверке в программном модуле и их расположение

|  |  |  |  |
| --- | --- | --- | --- |
| *№* | *Название* | *Функция* | *Класс или файл* |
| 1 | change\_status() | Изменение статуса участника. | Uchastnik |
| 2 | add\_uch() | Добавление участника в категорию. С помощью объявленного объекта в программе. | Category |
| 3 | add\_uch\_vvod() | Добавление участника в категорию. Пользовательский ввод. | Category |
| 4 | add\_cat() | Добавление категории в мероприятие. Пользовательский ввод. | Meropriatie |
| 5 | add\_all\_cat() | Добавление сразу всех категорий в мероприятие. Пользовательский ввод. | Meropriatie |
| 6.1 | Spisok\_cat() | Запись в файл список категорий на мероприятии | Event.py |
| 6.2 | clear\_Spisok\_cat() | Очистка списка с категориями | Event.py |
| 7.1 | Spisok() | Запись в файл списка участников по категории | Category.py |
| 7.2 | clear\_Spisok() | Очистка списка участников | Category.py |
| 7.3 | New\_spisok() | Обновление списка участников | Category.py |
| 8.1 | Generate\_badges() | Генерация бейджей | Badge.py |
| 8.2 | clear\_Badges() | Очистка файла с бейджами | Badge.py |
| 8.3 | New\_badges() | Обновление файла с бейджами | Badge.py |
| 9 | Badge\_w() | Вывод индивидуального бейджа на печать | Badge.py |
| 10 | Mer\_vvod() | Добавляет мероприятие(проект) в список мероприятий | Event.py |

Формат ввода тестовых данных для каждой функции:   
ct1 = (1 Волонтер)

uch1 = Uchastnik(1, 'Зайцева', 'Елена', 'Сергеевна', '2006-5-26', 'ж', '89913002586','liberty\_lena')

MR = (1, Молодежный Форум, СпортСтройФест)

Для каждого модуля были написаны отдельные тест-кейсы, для тестирования методов и функций программы

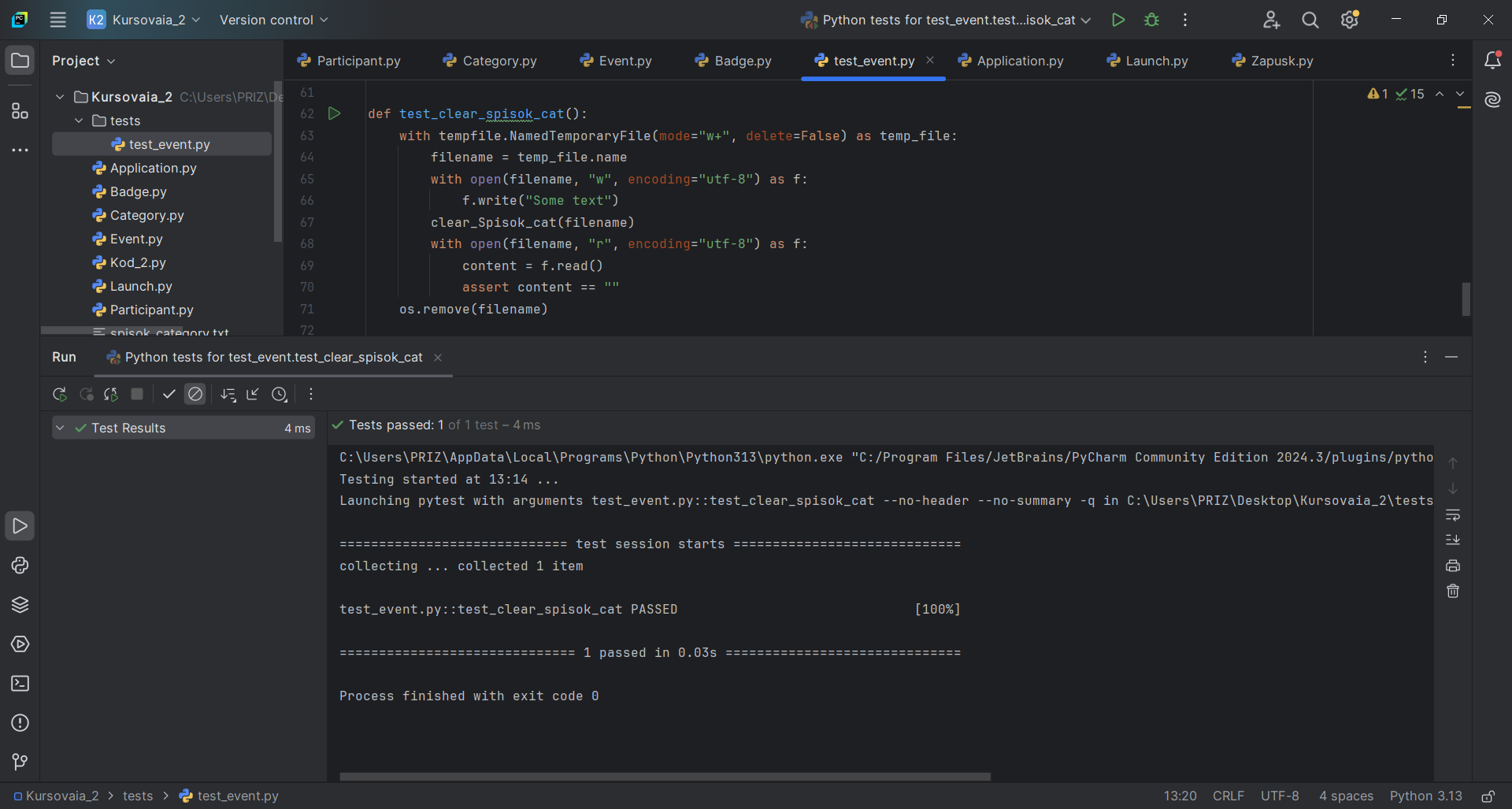
****

Рисунок 3.1.1 – Окно Pycharm c функцией тестирования и выводом результата

3.2 Описание применения средств отладки

В данном проекте были применены инструменты тестирования среды разработки PyCharm – PyTest и следующие методы тестирования:

*Модульное тестирование (Unit Testing)*– это тип тестирования программного обеспечения, при котором тестируются отдельные модули или компоненты программного обеспечения. Его цель заключается в том, чтобы проверить, что каждая единица программного кода работает должным образом.

*Use case тестирование* — это техника тестирования на основе вариантов использования, которая рассматривает взаимодействие системы и пользователя через призму конкретных сценариев. Сценарии описываются с точки зрения пользователя, а не системы.

Пример работы программы с консольным интерфейсом:

Приведенные ниже действия выполняются ролью «Организатор»:

- Создание нового мероприятия, как проекта, в котором будут выполняться остальные функции и храниться данные. Также выводит номер и название мероприятия (Рис.2.1.2.2):



Рисунок 2.2.2.2 – Создание нового мероприятия, как проекта.

- Вводится количество категорий участников, которые будут задействованы на мероприятии (Рис.2.4):



Рисунок 2.4 Ввод количества категорий.

- Вводится ID и названия всех категорий (Рис.2.5):

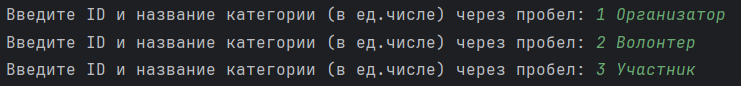


Рисунок 2.5

- Вводятся новые участники, которые пришли на мероприятие без предварительной регистрации (Рис.2.6):

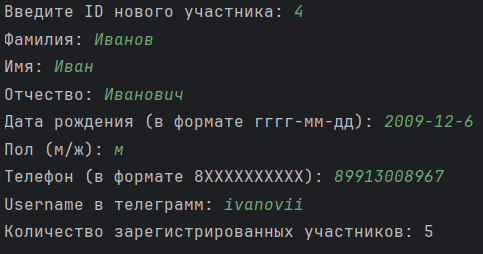


Рисунок 2.6

- Генерируется бейдж пришедшего участника для печати (Рис.2.7):

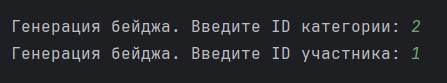


Рисунок 2.7

Таким образом, были протестированы функции программы включающие в себя анализ и обработку входных данных, формирование отчета, вывод в ворд документ для печати индивидуального бейджа.

Также был проведен рефакторинг кода, на добавление комментариев и исправление ненормированного нейминга, относительно названий модулей.

Стоит отметить, что рефакторинг ещё продолжается, так как в данной версии приложения в некоторых частях программы все ещё присутствует ненормированный нейминг, это было сделано разработчиком программы для ускорения процесса написания кода в условиях ограниченного времени, выделенного на разработку.

ЗАКЛЮЧЕНИЕ

В ходе данной курсовой работы был разработан программный модуль «Работа с участниками» для приложения «Молодежный форум», подтверждая актуальность автоматизации процессов управления в сфере организации мероприятий.

Анализ предметной области показал значительную потребность в подобных решениях, особенно в свете растущей популярности онлайн-коммуникаций и необходимости оптимизации ресурсов при организации масштабных мероприятий.

Изучение существующих программных решений позволило определить ключевые функции и сценарии работы, которые были реализованы в разработанном модуле.

Разработанный модуль успешно прошел отладку и тестирование, подтверждая свою работоспособность и соответствие заявленным требованиям.

Разработанная программная документация обеспечит простоту сопровождения и дальнейшего развития программного обеспечения.

Презентация проекта, подготовленная для потенциальных инвесторов или покупателей, демонстрирует значительный потенциал модуля для повышения эффективности работы с участниками молодежных форумов, сокращения затрат и увеличения лояльности участников, что отвечает основным целям проекта и подтверждает его актуальность и востребованность.

В целом, разработанное программное обеспечение решает проблему автоматизации процессов работы с участниками на молодежном форуме, способствуя развитию инновационных технологий в сфере организации молодежных мероприятий.

**ПРИЛОЖЕНИЕ 1**

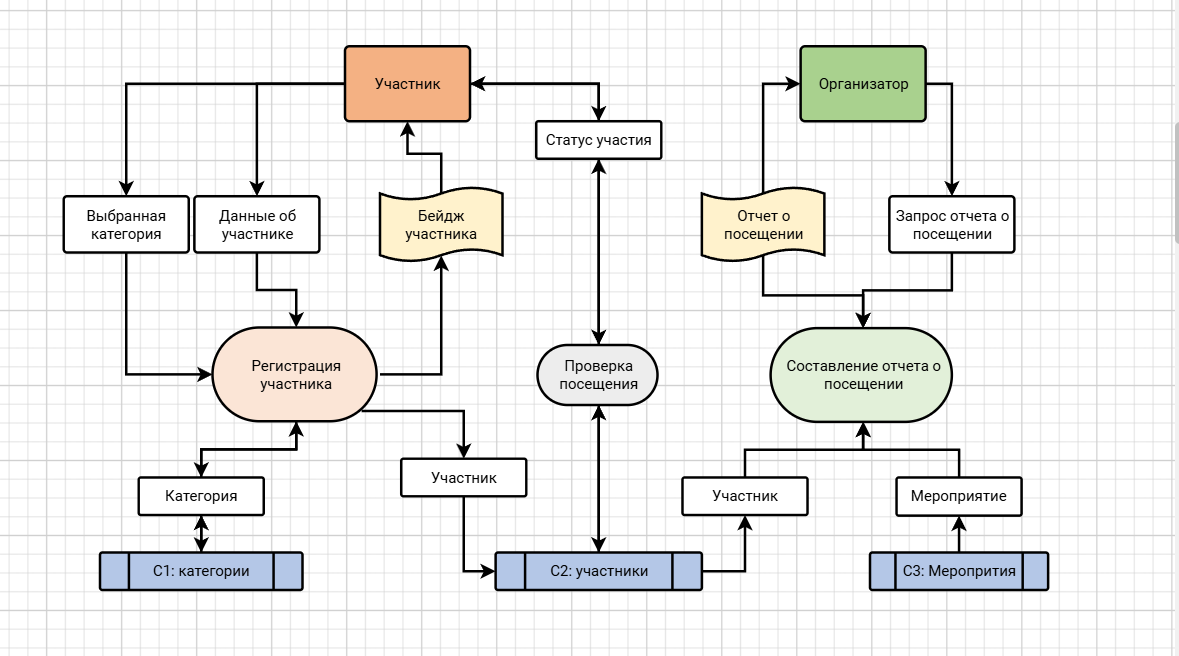
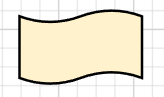


Рисунок 1.2 DFD Диаграмма потоков для модуля работа с участниками

Примечание 1 - схема составлена на основе, нотации Йордона-де Марко. Введено дополнительное обозначение  для более наглядного представления, которое означает итоговый выводимый документ.

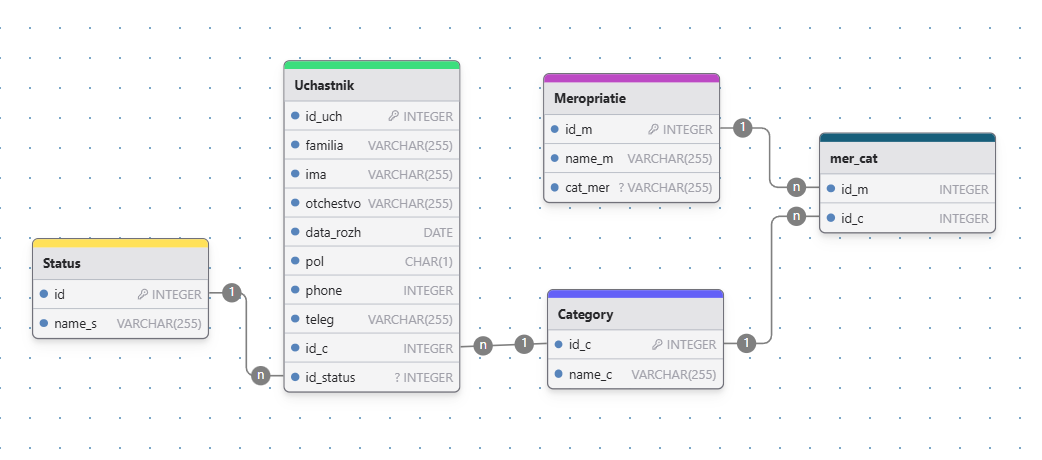


Рисунок 2 – ER – модель базы данных программного модуля

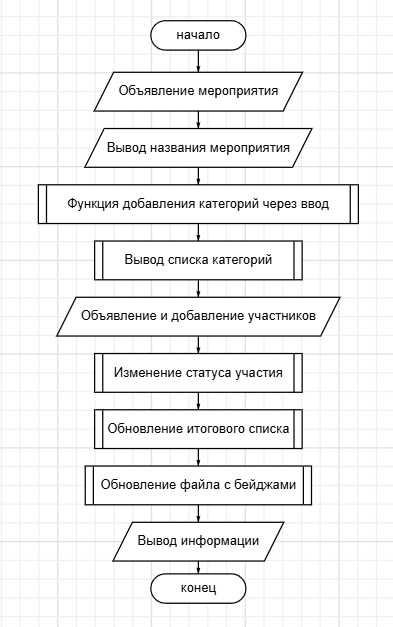


Рисунок 3 –Блок – схема модуля Launch

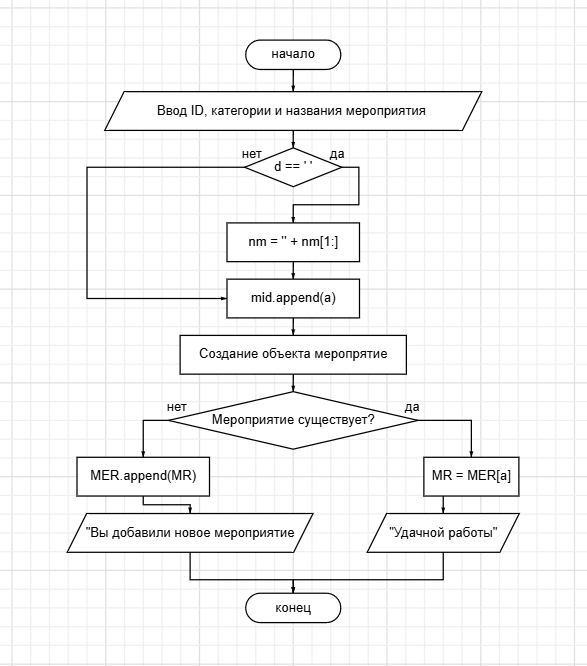
****

Рисунок 4 – Блок-схема функции Mer\_vvod

**ПРИЛОЖЕНИЕ 2**

****

Примечание 2 – Пример сгенерированного на основе шаблона бейджа. Выполнен в фирменном стиле Департамента городского имущества города Москвы

**ПРИЛОЖЕНИЕ 3 Анализ существующих программных решений**

1. ***REGPART***

Это не конкретный программный продукт, а целый сервис электронного сопровождения мероприятий. От электронной регистрации и изготовления бейджей до регистрации гостей на мероприятии, системы контроля доступа и подробной аналитики.

По функционалу данных сервис подходит, чтобы использовать его, как прототип для модуля приложения «Молодежный форум»

ПО предоставляет для решения различных задач связанных с регистрацией участников на мероприятии и действительно в комплексе способно упростить организацию.

Основные возможности REGPART:

* Электронная регистрация
* Печать бейджей
* Очная регистрация на мероприятии
* Предоставление оперативной информации
* Поддержка авторизации через соцсети
* Создание различных видов регистрации
* Система обмена сообщениями
* Поддержка разных языков и др.

***Ссылка на регистрацию и покупку сопровождения:***

[***https://www.regpart.ru/***](https://www.regpart.ru/) ***(19.12.2024г.)***

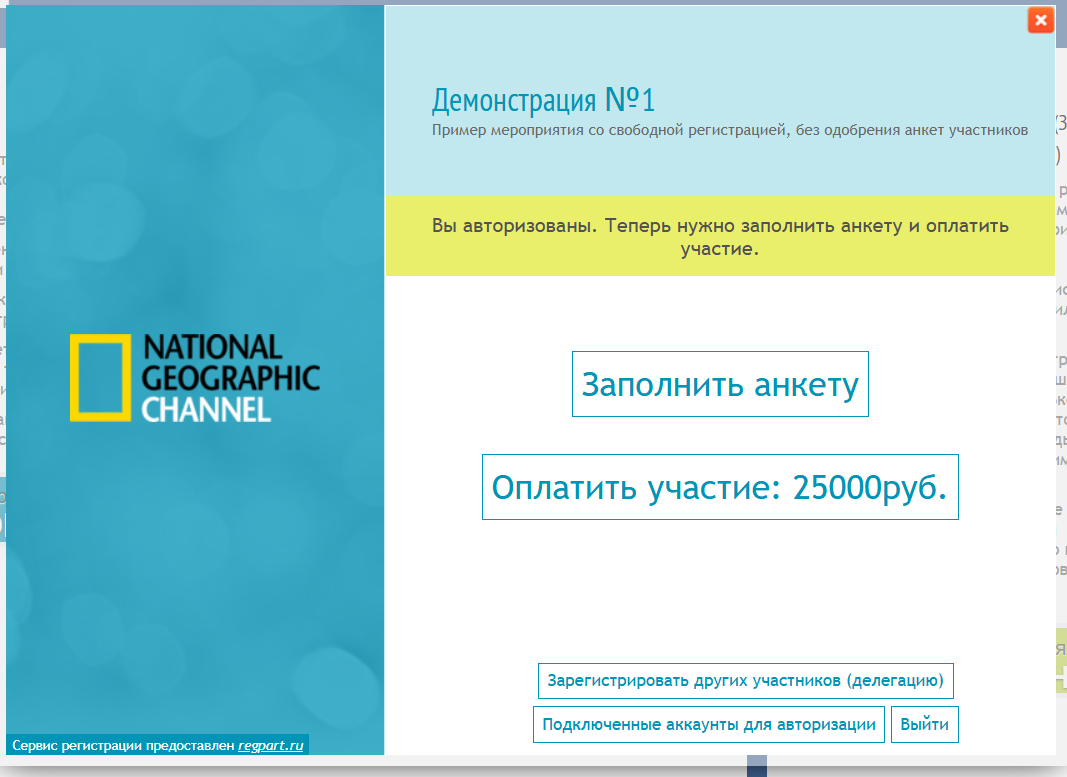


Рисунок 1 –Пример окна для регистрации.

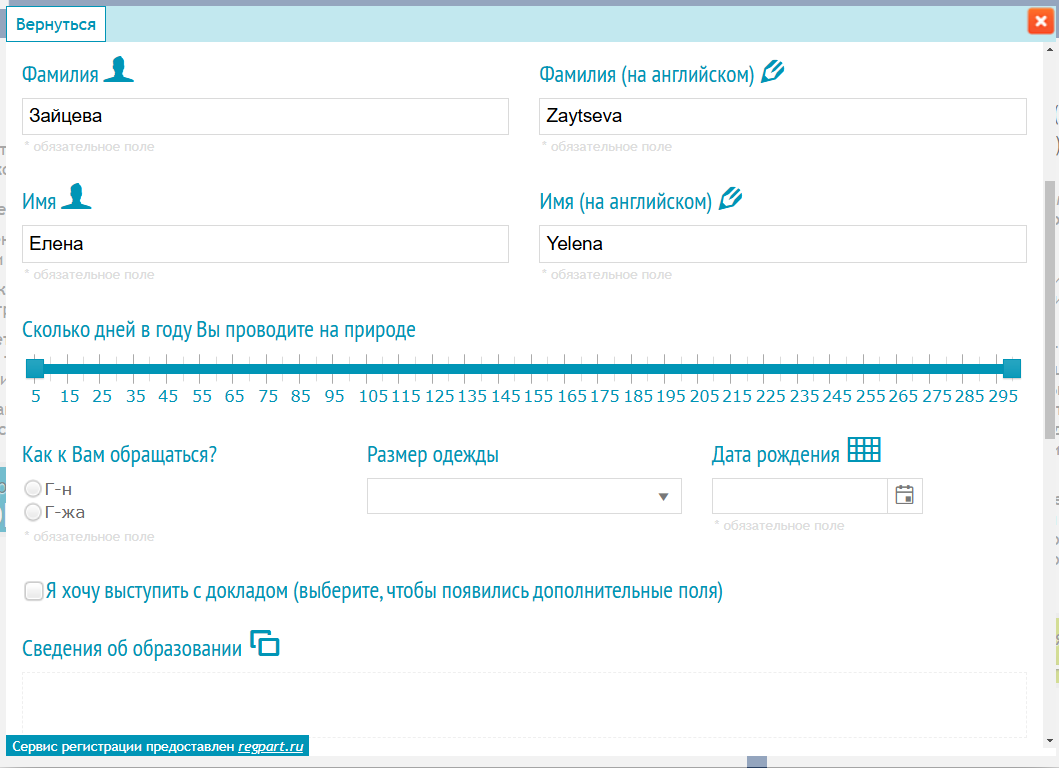


Рисунок 2 – Пример окна для заполнения анкеты участника.

1. ***ФГАИС Молодежь России***

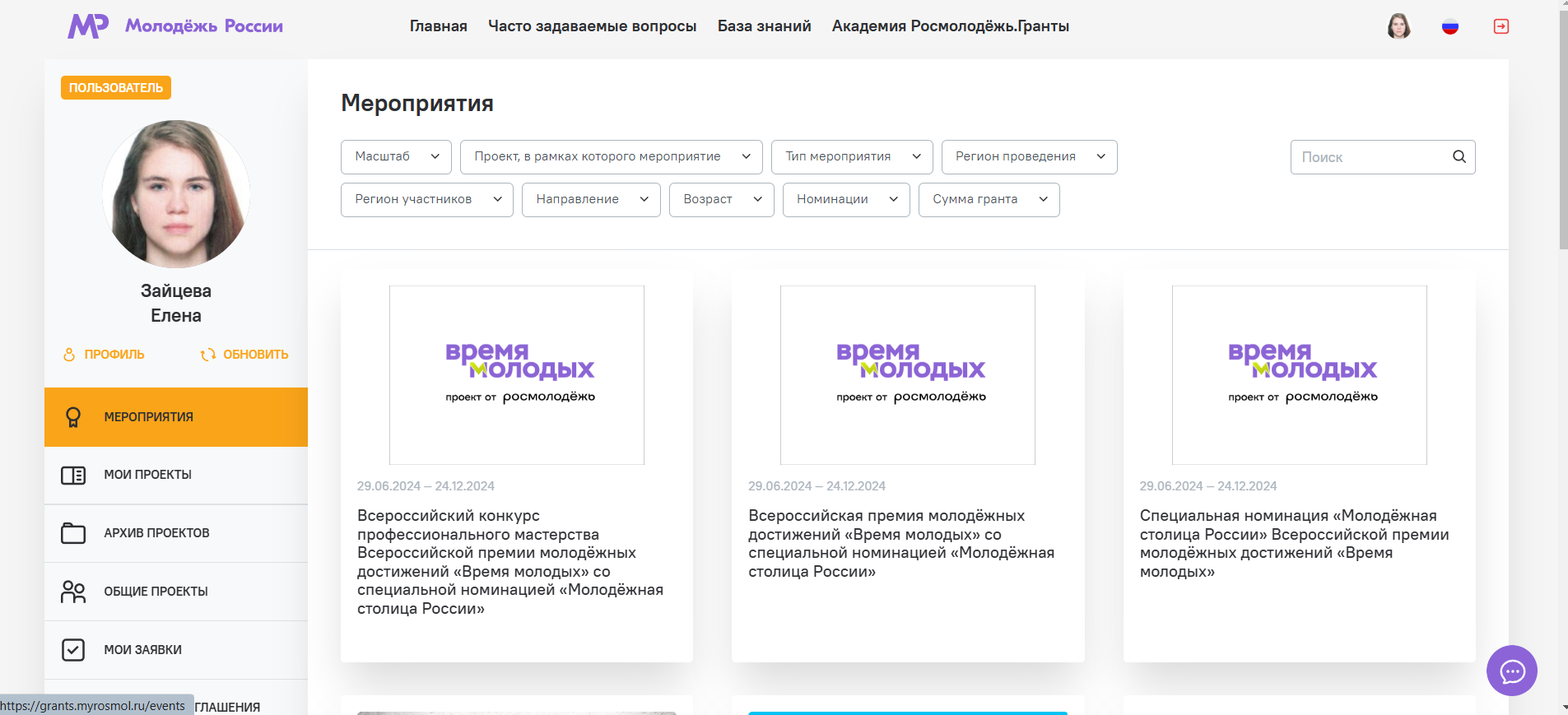


Рисунок 3 – Пример окна личного кабинета пользователя ФГАИС «Молодежь России». Выбор мероприятия.

ФГАИС «Молодёжь России» – это автоматизированная информационная система, через которую проходит регистрация на самые крупные молодежные мероприятия страны. Грантовые конкурсы, федеральные молодежные образовательные форумы и фестивали, региональные мероприятиятия. Система «Молодёжь России» - это удобная стартовая площадка для социально активного молодого человека.

Чтобы зарегистрироваться на платформе, необходимо:

1. Перейти по ссылке: [myrosmol.ru/](https://vk.com/away.php?to=https%3A%2F%2Fmyrosmol.ru%2F) (19.12.2024г.)
2. Выбрать либо «Войти», либо «Быстрая регистрация».

У пользователей системы есть возможность создавать и оформлять собственные портфолио с информацией об участии в общественных мероприятиях.

Данная ИС стала основой для создания функции заявка в программном модуле «Работа с участниками»

1. ***ADVANTA***

Российская система управления портфелями проектов. Функции для управления проектами:

* Автоматизированный сбор статусов от исполнителей
* Планирование платежей, контроль оплат, анализ БДР и БДДС
* Балансировка ресурсов, анализ загрузки подразделений
* Сводная отчетность по портфелю: прогресс проектов, качество управления, показатели
* Жизненный цикл проекта
* Удобная и функциональная диаграмма Ганта
* Отчет по проекту в один клик
* Управление трудовыми ресурсами
* Разграничение доступа к данным по ролям
* Управление договорами
* Приемка задач и контроль поручений
* Управление рисками

Функция вывода списка пользователей, стала основой для разработки функции вывод участников в проекте.

***Ссылка на скачивание ДЕМОВЕРСИИ:*** [***https://promo.advanta-group.ru/***](https://promo.advanta-group.ru/)

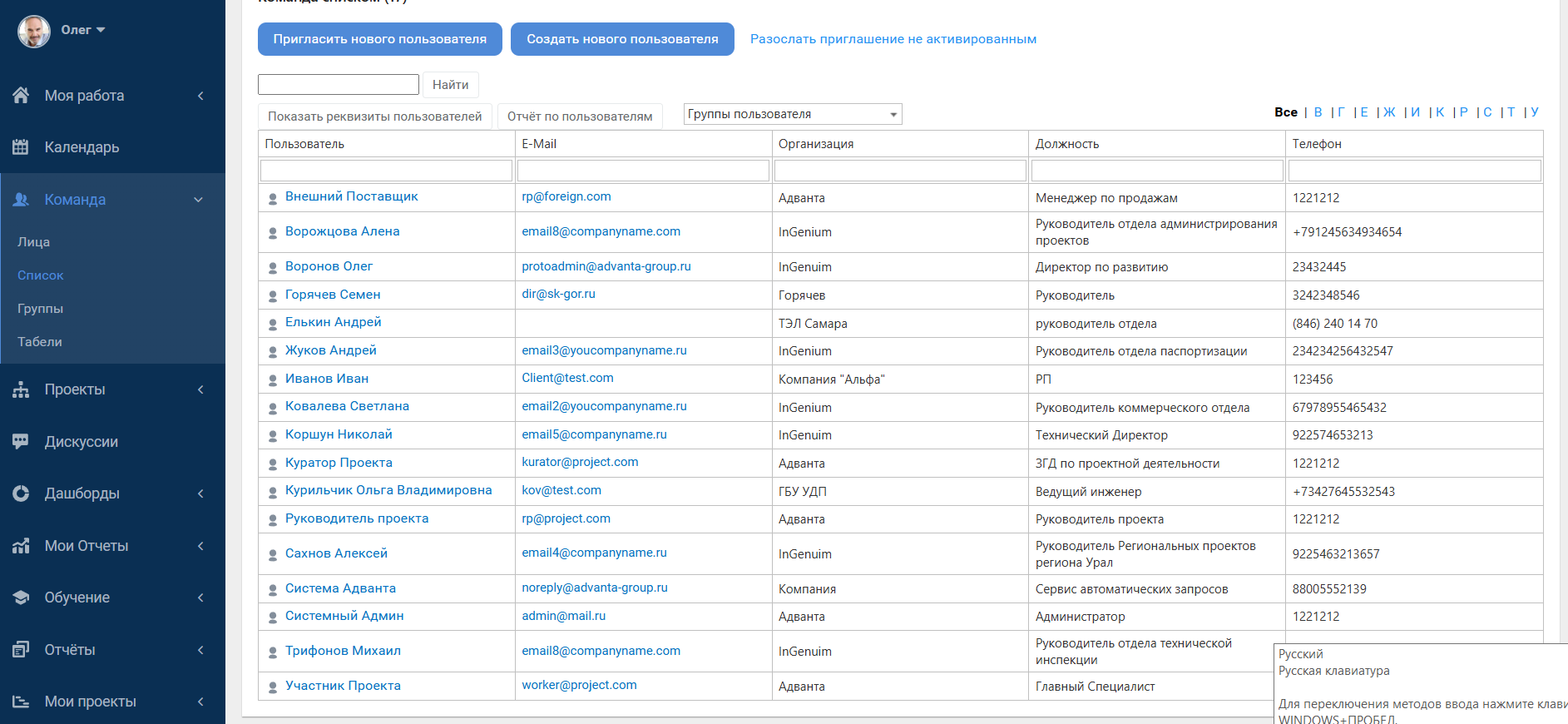
******

Рисунок 5 – Пример окна списка пользователей ADVANTA.

1. ***CREATIO CRM***

Один из лидеров среди CRM. Продукт отечественной компании Terrasoft.

Платформа для управления бизнес-процессами и CRM-система, предоставляющая широкий спектр инструментов для автоматизации различных аспектов бизнеса,

Возможности приложения (Creatio):

* Управление взаимоотношениями с клиентами (CRM)
* Управление маркетингом, продажами, сервисом, проектами
* Мобильный доступ
* Интеграция
* Настраиваемость
* BPM (Business Process Management)

В настоящее время эта программа недоступна для скачивания и сотрудничества. Прямую ссылку на установку Creatio дать нельзя, поскольку это платное решение, устанавливается на собственных серверах или в облаке поставщика. ***Для установки необходимо было бы обратиться к официальному сайту Creatio*** [***https://www.creatio.com/***](https://www.creatio.com/) ***(нет доступа)***

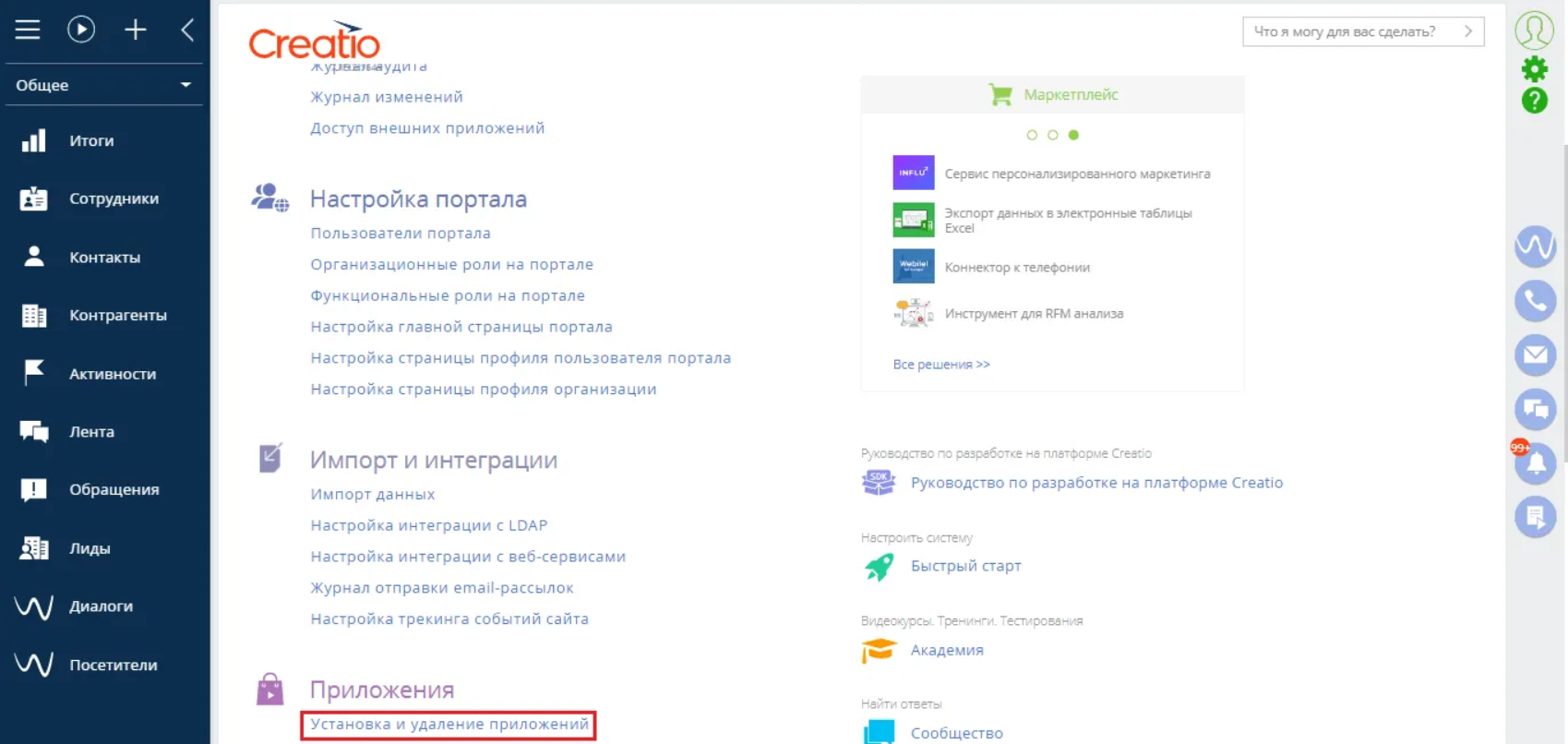


Рисунок 4 – Пример окна пользователя CREATIO