

## Использование современных информационных технологий для оптимизации бизнес-процессов

И. Р. Кузнецова, А. С. Букунов, С. В. Букунов✉

Санкт-Петербургский государственный архитектурно-строительный университет,  
Санкт-Петербург, Россия

✉ [sergeybukunov@yandex.ru](mailto:sergeybukunov@yandex.ru)

**Аннотация.** Описывается разработанное на кафедре информационных технологий Санкт-Петербургского архитектурно-строительного университета веб-приложение, предназначенное для оптимизации бизнес-процессов клуба активных путешествий. Приложение может быть отнесено к классу программного обеспечения для бизнеса. Оно позволяет существенно облегчить руководству клуба ведение операций по управленческому и финансовому учету за счет автоматизации рутинных операций и визуализации информации, необходимой для принятия управленческих решений. Работа выполнялась в соответствии с техническим заданием, выданным руководством клуба. В приложении реализованы системы авторизации и аутентификации пользователей. Каждая группа пользователей обладает своим набором возможностей при работе с приложением, написанным на языке Python с использованием библиотеки Django. Для хранения всей необходимой информации была спроектирована и реализована реляционная база данных (БД). Работа с ней осуществляется с помощью системы управления базами данных (СУБД) MySQL. С помощью графической библиотеки Plotly реализована встроенная аналитическая панель, позволяющая визуализировать и анализировать информацию, хранящуюся в БД. Для решения ряда задач использовались языки HTML, CSS, JavaScript, технология AJAX. Для работы с приложением не требуется установка дополнительного программного обеспечения, достаточно наличия доступа к сети Интернет. Подходы и решения, реализованные в приложении, могут быть с успехом использованы при создании подобных программных продуктов для других организаций, занимающихся схожими видами деятельности.

**Ключевые слова:** бизнес-процесс, веб-приложение, база данных, визуализация данных, объектно-ориентированное программирование

**Для цитирования:** Кузнецова И. Р., Букунов А. С., Букунов С. В. Использование современных информационных технологий для оптимизации бизнес-процессов // Изв. СПбГЭТУ «ЛЭТИ». 2022. Т. 15, № 8. С. 57–68. doi: 10.32603/2071-8985-2022-15-8-57-68.

**Конфликт интересов.** Авторы заявляют об отсутствии конфликта интересов.

Original article

## Use of Modern Information Technologies to Optimization Business Processes

I. R. Kuznetsova, A. S. Bukunov, S. V. Bukunov✉

Saint Petersburg State University of Architecture and Civil Engineering, Saint Petersburg, Russia

✉ [sergeybukunov@yandex.ru](mailto:sergeybukunov@yandex.ru)

**Abstract.** A web-application developed at the Department of Information Technologies of the Saint Petersburg State University of Architecture and Civil Engineering, which makes it possible to optimize business processes of the active travel club, is described. The application can be categorized as business software. It allows the club's management to significantly facilitate the management of the club's management and financial accounting operations by automating of routine operations and visualization of the information necessary for making man-

agement decisions. The work was carried out according to the technical specification given by the club management. The application implements authorization and user authentication systems. Each user group has its own set of features when working with the application. The application is developed by Python programming language and Django framework. Relational data base was designed and implemented for data storage. MySQL data base management system (DBMS) is used to organize interaction with the database. A built-in analytical panel designed with the Plotly graphic library is implemented to visualize and analyze information stored in the data base. To solve some problems such languages as HTML, CSS, JavaScript and AJAX technology were used. The system does not require the installation of additional software. Access to the Internet need to use the system only. The approaches and technological solutions implemented in the application can be successfully used to create similar systems for other organizations and companies involved in similar activities.

**Keywords:** business process, web-application, data base, data visualization, object-oriented programming

**For citation:** Kuznetsova I. R., Bukunov A. S., Bukunov S. V. Use of Modern Information Technologies to Optimization Business Processes // LETI Transactions on Electrical Engineering & Computer Science. 2022. Vol. 15, no. 8. P. 57–68. doi: 10.32603/2071-8985-2022-15-8-57-68.

**Conflict of interest.** The authors declare no conflicts of interest.

**Введение.** Сложившаяся в настоящее время достаточно жесткая конкурентная среда, нарастающая сложность решаемых задач и стремительно увеличивающиеся информационные потоки заставляют организации вне зависимости от их размера и формы собственности использовать современные информационные технологии и научные методы обработки данных для оптимизации своих бизнес-процессов. Еще совсем недавно для решения задач такого рода использовались в основном настольные приложения, которые необходимо было устанавливать на стационарные компьютеры пользователей. Однако в последние годы широкое распространение получили информационные технологии, связанные с сетью Интернет, – так называемые веб-технологии. Использование веб-приложений позволяет компаниям существенно расширить круг потенциальных заказчиков и повысить эффективность управления своими бизнес-процессами за счет повышения оперативности и снижения расходов. К очевидным преимуществам веб-приложений относятся возможность работы с ними из любой точки мира с любого устройства, имеющего выход в сеть Интернет.

Разработка данного приложения осуществлялась на основании технического задания, сформулированного руководством клуба активных путешествий INSLED. Клуб INSLED занимается проведением различных активных мероприятий для широкого круга лиц – пешеходов и лыжных походов, велопрогулок, интеллектуальных игр и т. п. Для информационного взаимодействия с участниками мероприятий и оперативного учета своей деятельности руководство клуба использовало

различные программные продукты: электронные таблицы Excel, Google-таблицы, социальные сети и др. Однако по мере увеличения количества участников и проводимых мероприятий организация хранения данных, обмена данными между различными программными продуктами, обработка данных стали требовать значительных затрат. Это и послужило причиной для принятия решения о создании нового собственного программного продукта в виде веб-приложения, представляющего собой универсальный инструмент для решения всех стоящих перед руководством клуба задач. Универсальность такого приложения должна заключаться в том, что, с одной стороны, оно должно представлять собой некий единый информационный ресурс, позволяющий работать с ним разным группам авторизованных пользователей, а с другой стороны, служить расчетно-аналитическим инструментом для руководства клуба.

**Постановка задачи.** Основные подходы, используемые для реализации клиентской части разработанного веб-приложения, системы авторизации и аутентификации пользователей, функциональности пользователей разного уровня представлены в [1].

Цель настоящей статьи заключается в расширении функциональных возможностей пользователей приложения, имеющих статус организатора или администратора, за счет автоматизации операций по бюджетированию проводимых мероприятий, статистической обработки и визуализации информации, необходимой для анализа деятельности клуба и принятия обоснованных управленческих решений.

**Основная функциональность приложения.**

Все пользователи приложения подразделяются на два типа:

- 1) внешние (неавторизованные) пользователи;
- 2) внутренние (авторизованные) пользователи.

В свою очередь, внутренние пользователи могут относиться к одному из трех видов:

- 1) участник;
- 2) организатор;
- 3) администратор.

Функциональность пользователей со статусом *участника* ограничена просмотром и редактированием личной информации и информации о мероприятиях, в которых он принимал участие. Детальное описание функциональных возможностей пользователей этого типа представлено в [1].

Пользователи со статусом *организатора* обладают более расширенными возможностями. Помимо полного доступа к личной информации и информации об участниках мероприятий, организованных ими, в этом приложении такие пользователи могут работать с информацией, касающейся бюджета проводимого ими мероприятия. В частности, они могут:

- добавлять, редактировать и удалять данные о контрагентах мероприятия;
- создавать, редактировать и удалять категории расходов на мероприятие;
- распределять категории расходов по участникам мероприятия;
- заносить и редактировать информацию о выставлении счета участнику мероприятия, организованного пользователем;
- заносить информацию об оплате счетов, выставленных контрагентами мероприятия;
- просматривать совершенные финансовые операции в рамках мероприятия;
- просматривать задолженности клуба перед контрагентами в рамках мероприятия;
- просматривать задолженности участников клуба в рамках мероприятия;
- просматривать статистическую информацию по мероприятиям и их бюджетам с помощью аналитической панели.

Пользователи со статусом *администратора* обладают максимальным набором возможностей при работе с данным приложением, а именно имеют полный доступ к редактированию (с возможностью удаления) любой информации, хранящейся в базе данных, в частности:

- редактировать сведения о пользователях;
- редактировать информацию о мероприятиях;
- редактировать информацию о клубных картах;
- редактировать информацию о контрагентах;
- редактировать информацию о совершенных финансовых операциях;
- просматривать статистическую информацию о мероприятиях и их бюджетах с помощью аналитической панели.

В настоящее время для формирования информационных отчетов в бизнес-аналитике широкое распространение получили так называемые аналитические панели или дашборды (dashboard), представляющие собой лаконичные отчеты с визуально представленными данными [2], [3]. Как правило, в качестве элементов таких отчетов выступают различного типа графики, текст, цифры, таблицы, элементы инфографики. Пользователями таких компактных отчетов могут быть руководители компаний, специалисты подразделений, клиенты. При генерации отчета необходимые данные автоматически подгружаются из интегрированных сервисов и выводятся на экран. В настоящее время существует достаточно большое количество программных продуктов для создания таких аналитических панелей, например QlikView [4], Klipfolio [5], Power BI [6], Tableau [7]. Однако большинство сервисов для создания дашбордов, во-первых, платные, а во-вторых, требуют дополнительных трудозатрат для их интеграции с собственными информационными системами или базами данных.

Для реализации возможности проведения детального анализа деятельности клуба в разработанном приложении средствами графической библиотеки Plotly [8] была реализована собственная специальная аналитическая панель, которая позволяет визуализировать различную информацию, хранящуюся в базе данных. Эта панель позволяет, в частности, выводить на экран:

- диаграмму распределения мероприятий по типам;
- диаграмму распределения мероприятий по организаторам;
- график пройденной дистанции для мероприятий различного типа;
- диаграмму распределения мероприятий по направлениям;
- диаграмму распределения участников по типам мероприятий;
- зависимость количества мероприятий от уровня сложности;

- диаграмму распределения прибыли по типам мероприятий;
- график распределения прибыли по мероприятиям различного уровня сложности;
- диаграмму распределения прибыли по организаторам мероприятий;
- диаграмму распределения прибыли по различным направлениям проведения мероприятий.

**База данных.** Исходя из разработанной функциональности приложения была спроектирована и реализована реляционная база данных [9], основное назначение которой заключается в хранении всех данных, необходимых для его полноценной работы. Для унифицированного представления данных, хранящихся в БД, как правило, используется так называемая ER-модель (Entity-relationship model), которая представляет собой схему «сущность – связь» [10]. Под сущностью при этом понимают класс однотипных объектов, а связи характеризуют отношения между сущностями разного типа.

На этапе концептуального проектирования базы данных приложения было выделено пятнадцать сущностей:

- 1) пользователь;
- 2) тип информации;
- 3) дополнительная информация о пользователе;
- 4) клубная карта;
- 5) мероприятие;
- 6) тип мероприятия;
- 7) участник мероприятия;
- 8) организатор мероприятия;
- 9) счет пользователя/контрагента;
- 10) тип расходов;
- 11) расходы на мероприятие;
- 12) расходы пользователя на мероприятие;
- 13) счет, выставленный участнику мероприятия;
- 14) счет, выставленный контрагентом;
- 15) финансовая операция.

ER-модель разработанной базы данных представлена на рис. 1.

На этапе логического проектирования ER-модель была конвертирована в схему базы данных на основе реляционной модели данных. При этом была реализована нормализация отношений между сущностями и сформированы требования поддержки целостности данных. В результате для каждой сущности была создана соответствующая таблица в базе данных, определены типы данных для всех полей каждой таблицы и реализованы необходимые связи между таблицами.

**Используемые технологии.** Для создания веб-приложения использовался язык программирования Python [11] и его популярный фреймворк Django [12], [13]. Приложение реализует технологию «клиент–сервер», т. е. состоит из клиентской и серверной частей.

Клиентская часть веб-приложения представляет собой интерфейс пользователя, с помощью которого формируются запросы к серверу и обрабатываются ответы от него. Для разработки клиентской части приложения в качестве основного инструмента использовался язык разметки документов HTML [14]. Для улучшения представления информации и добавления интерактивности в работу приложения дополнительно были использованы формальный язык описания внешнего вида документа CSS [14] и язык программирования JavaScript (jQuery) [15]. Для обмена информацией с сервером асинхронными запросами использовалась технология AJAX [16]. Вся интерактивная графика приложения реализована средствами графической библиотеки Plotly [8].

Серверная часть веб-приложения представляет собой программу, написанную на языке Python, которая обрабатывает запросы браузера. Эта часть приложения была реализована на базе встроенного в Django административного сайта.

Для работы с БД использовалась система управления базами данных MySQL [17].

**Архитектура приложения.** Приложение было спроектировано на основе реализованного в библиотеке Django шаблона проектирования MVT (Model-View-Template) [12], основная идея которого заключается в разделении данных, хранящихся в базе данных, логики по обработке этих данных и шаблонов для их отображения на отдельные компоненты.

Все приложение делится на модули по их функциональному назначению. К основным модулям приложения относятся:

- файл `models.py` – содержит модели приложения – абстракции в виде Python-классов для структурирования и управления данными приложения;
- файл `views.py` – содержит представления приложения – Python-классы, содержащие всю логику обработки данных (принимают веб-запросы и возвращают веб-ответы);
- шаблоны (`templates`) – отображают сгенерированный контент в форматах HTML, JSON, XML и им подобным на страницах приложения;
- файл `urls.py` – содержит URL-адреса всех веб-страниц приложения;

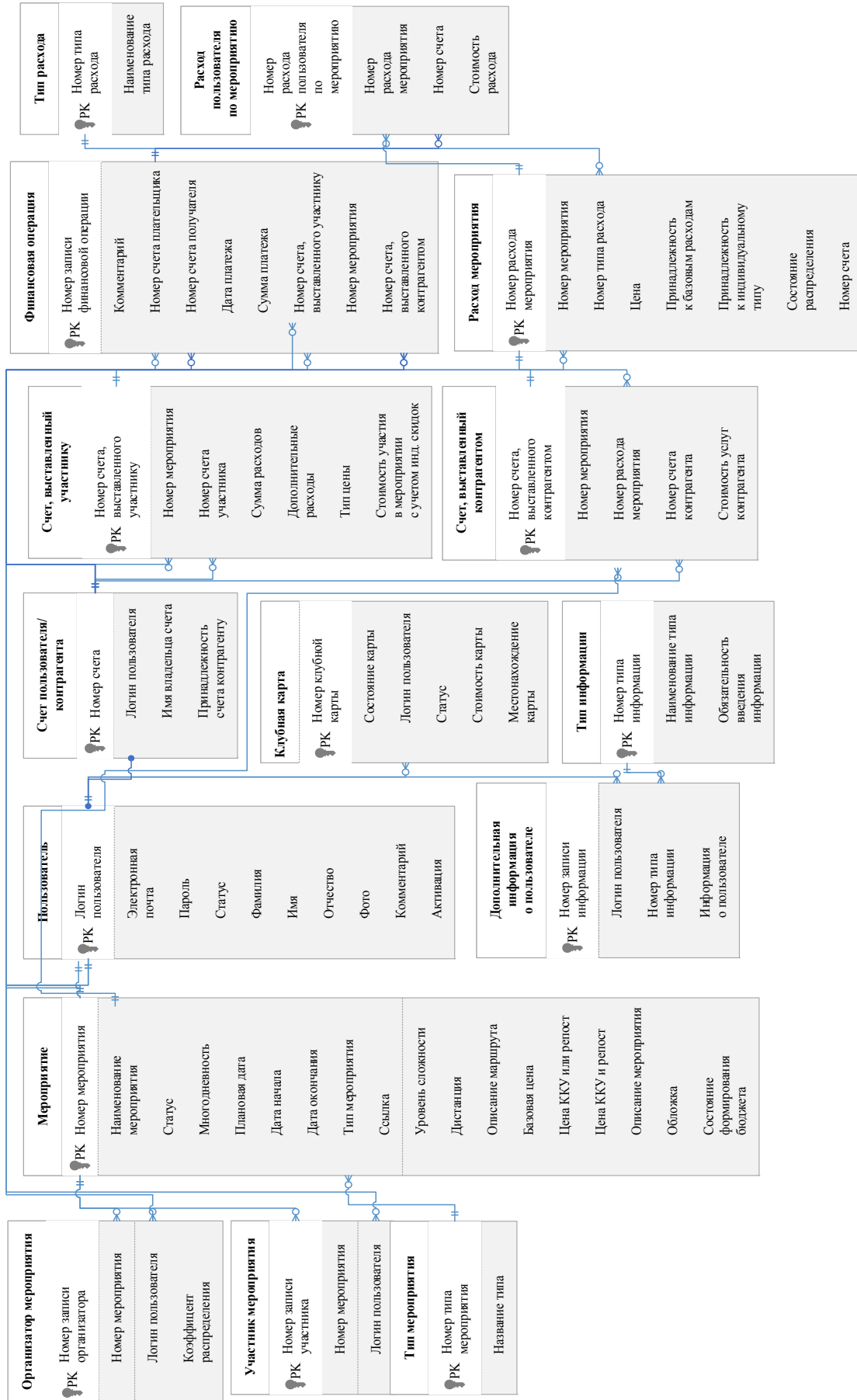


Рис. 1. ER-модель базы данных  
Fig. 1. Database ER-model

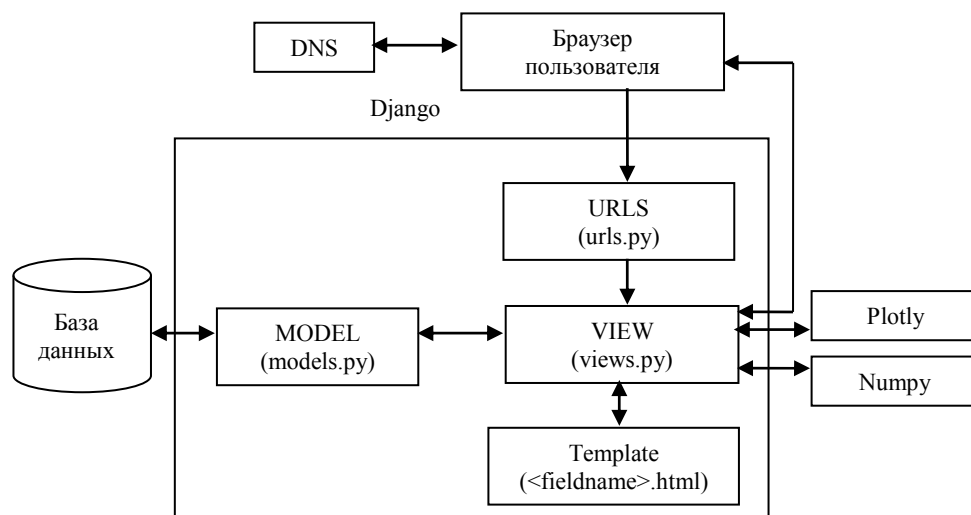


Рис. 2. Архитектура приложения  
Fig. 2. Application architecture

– база данных – содержит все данные приложения;

– plotly – библиотека для визуализации статистических данных;

– numpy – библиотека для обработки статистических данных.

– DNS-сервер (Domain name server) – сервер, хранящий IP-адреса сайтов, задача которого заключается в выдаче браузерам пользователей соответствующего сайта по имени домена.

Общий вид архитектуры приложения представлен на рис. 2.

#### Особенности программной реализации.

В этой части взаимодействие с БД осуществляется с помощью механизма объектно-ориентированного отображения (Object-Relation Mapper) [13]. Система ORM автоматизирует взаимодействие с базой данных и применяет объектно-ориентированный подход вместо инструкций SQL.

Создание отношений в базе данных осуществляется посредством моделей. Каждая модель представляет собой Python-класс, унаследованный от класса Model из модуля django.db.models. При этом структура каждого класса соответствует структуре соответствующих таблиц в базе данных.

Для обработки запросов в разработанном приложении использовался подход, также основанный на использовании Python-классов (Classed-based-views). Использование представлений-классов при разработке приложения позволило сократить размер кода, а также использовать другие преимущества объектно-ориентированного программирования (наследование, множественное наследование, примеси и др.).

Фреймворк Django предоставляет разработчику большое количество классов, которые могут быть использованы для ускорения и облегчения процесса разработки. Однако возможности этих классов не всегда соответствуют решаемым задачам. Так, например, Django обладает встроенной системой аутентификации пользователя, представленной классом User, от которой в данном случае пришлось отказаться в связи с необходимостью хранения большего количества данных о пользователе, чем предусмотрено этим классом. Поэтому для реализации нужной функциональности был создан собственный класс AdvUser, унаследованный от того же абстрактного класса, что и User – AbstractUser. При этом состав атрибутов класса AdvUser был определен в соответствии с разработанной схемой БД.

Для реализации некоторых функций приложения использовалась технология AJAX (Asynchronous JavaScript and XML) [15], [16]. Главное преимущество этой технологии перед стандартным подходом заключается в быстром отклике на действия пользователя.

В частности, с помощью технологии AJAX в разработанном приложении был реализован алгоритм обновления диаграмм во встроенной аналитической панели. Результат работы алгоритма заключается в автоматической перерисовке диаграмм при изменении исходных данных.

Пример такой трансформации диаграмм на странице «Мероприятия» приложения представлен на рис. 3. На диаграммах отображается количество мероприятий разного типа за выбранный

интервал времени. В качестве варьируемых исходных данных для построения диаграмм в данном случае выступает диапазон времени, который пользователь задает посредством ввода начальной и конечной дат в соответствующие поля ввода, расположенные в нижней части диаграммы.

На рис. 3 видно, что при изменении анализируемого периода времени на диаграммах происходит изменение не только численных значений, соответствующих количеству проведенных мероприятий каждого типа, но и самого набора типов мероприятий, поскольку в разные периоды времени проводилось разное количество мероприятий разного типа.

Во всех шаблонах проекта используется механизм наследования от базового шаблона, который отображает верхнее меню, шапку и подвал сайта.

Также в шаблонах используется модуль для работы с фреймворком bootstrap4 [18], который необходимо подключить в начале HTML-документа.

**Обсуждение результатов.** Разработанное приложение представляет собой многостраничное веб-приложение. Доступ к той или иной его странице определяется статусом пользователя. Далее приводятся некоторые примеры работы приложения для пользователей со статусом организатора и администратора.

Например, страница «Бюджет» (рис. 4) предназначена для формирования бюджета конкретного мероприятия. На этой странице пользователю предоставляется возможность выбора мероприятия, для которого необходимо сформировать бюджет.

После выбора мероприятия пользователю становится доступно меню для формирования бюджета выбранного мероприятия (рис. 5).



Рис. 3. Трансформация внешнего вида диаграммы при изменении исходных данных  
 Fig. 3. Transforming the appearance of the chart when changing the source data

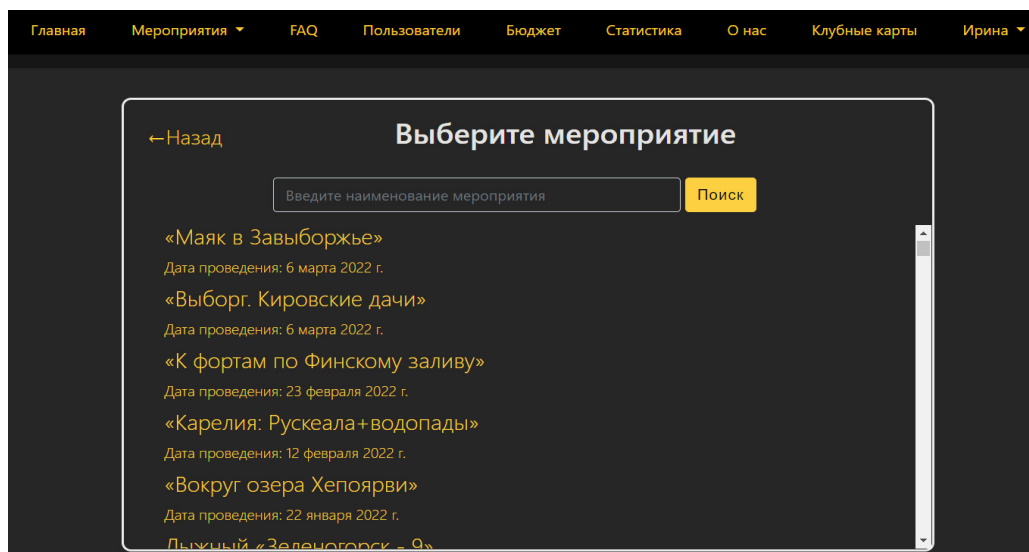


Рис. 4. Страница «Бюджет»  
 Fig. 4. Budget Page

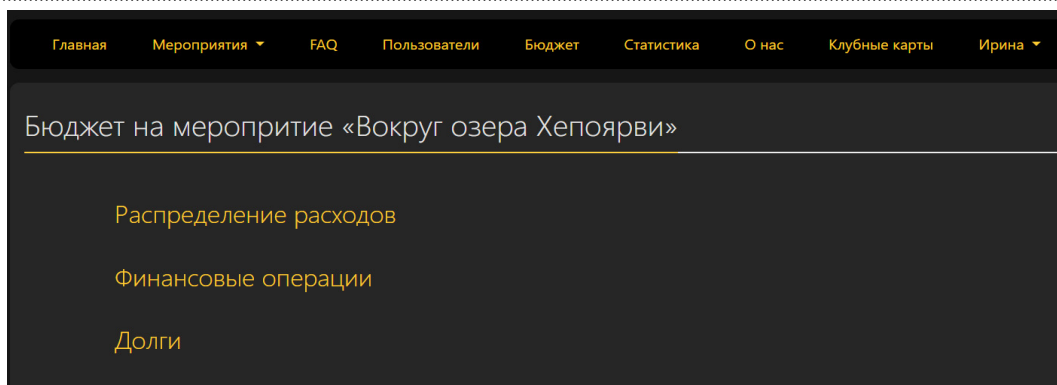


Рис. 5. Страница «Бюджет на мероприятие»  
Fig. 5. Event Budget Page

Перейдя в пункт меню «Распределение расходов», пользователь может увидеть категории расходов мероприятия. При помощи специальной формы пользователь имеет право добавлять, редактировать категории расходов мероприятия и удалять их. С помощью другой специальной формы пользователь может распределить расходы различных категорий среди участников и организаторов данного мероприятия. Форма для распределения расходов представлена на рис. 6.

ФИО	Организатор	Стоимость	Удалить
Павлов Владислав Игоревич	<input checked="" type="checkbox"/>	500,00	Удалить
Герасимов Денис	<input type="checkbox"/>	500,00	Удалить
Михайлов Михаил Валерьевич	<input type="checkbox"/>	500,00	Удалить

Рис. 6. Форма для распределения расходов  
Fig. 6. Cost Distribution Form

До нажатия кнопки «Сохранить и закончить» пользователь может вносить любые изменения в распределение расходов. При нажатии данной кнопки статус распределения категории расходов меняется на завершенный.

После окончания формирования расходов пользователю становятся доступными пункты меню «Финансовые операции» и «Долги».

Перейдя в «Финансовые операции», пользователь сможет увидеть список доступных ему финансовых операций: прием оплаты от участника, совершение оплаты контрагенту и отображение списка операций.

При приеме оплаты от участника организатор в первую очередь должен выставить участнику счет по мероприятию. Счет содержит в себе информацию о затратах на пользователя, входящих в стоимость участия в мероприятии, дополнительных расходах и стоимости участия, зависящей от наличия у участника клубной карты и выполнения репоста записи о мероприятии в социальной сети.

После выставления счета участнику организатор имеет возможность с помощью специальной формы ввести в базу данных сумму, которую заплатил участник мероприятия.

При совершении оплаты контрагенту организатор должен из выпадающего списка категорий расходов выбрать категорию, по которой он хочет совершить оплату, и затем ввести сумму оплаты.

После ввода всей информации о выбранном мероприятии пользователь может просмотреть список всех совершенных в рамках этого мероприятия финансовых операций с возможностью фильтрации операций по плательщикам и получателям. Возможный вариант вывода такого списка представлен на рис. 7.

При переходе в пункт меню «Долги» пользователю предоставляется возможность просмотра задолженностей клуба или участников. При просмотре задолженности участников мероприятия он может отфильтровать задолженности по участникам.

Возможный вариант просмотра задолженностей участников представлен на рис. 8.



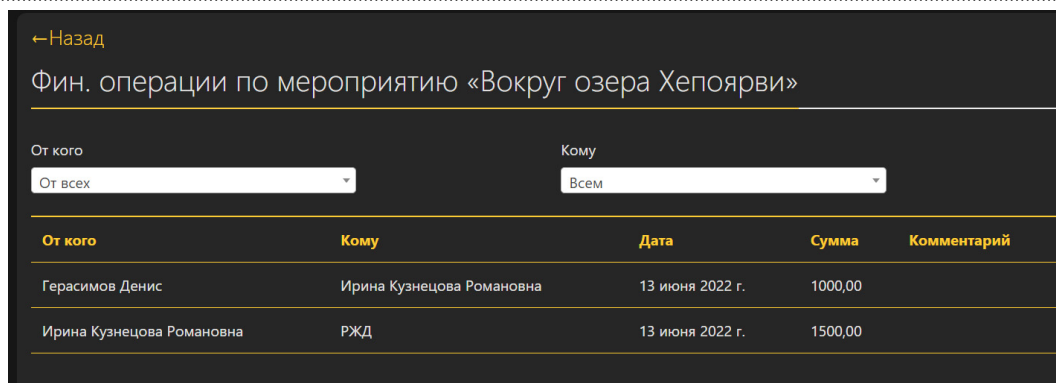


Рис. 7. Просмотр финансовых операций по мероприятию  
 Fig. 7. View event financial transactions

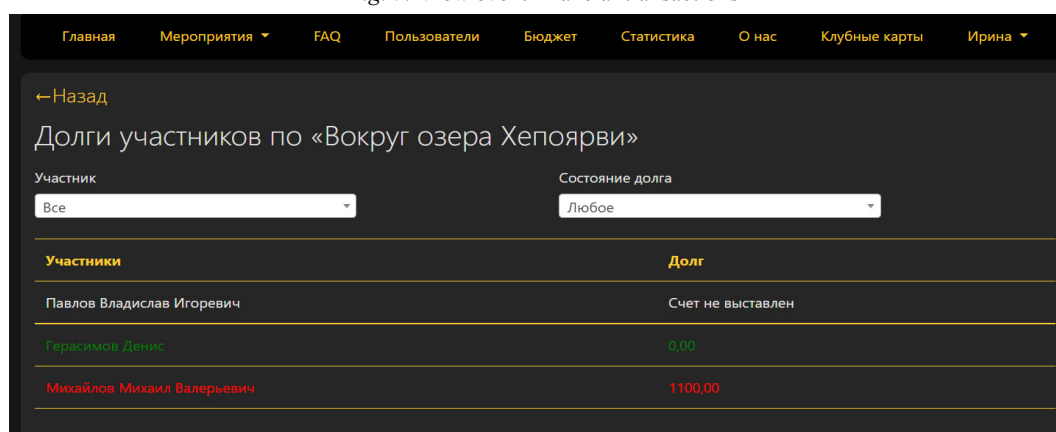


Рис. 8. Просмотр задолженностей участников мероприятия  
 Fig. 8. View debts of event participants



Рис. 9. Статистика мероприятий  
 Fig. 9. Events statistics

На странице «Статистика» пользователь с правами организатора или администратора может просмотреть и проанализировать статистическую информацию по деятельности клуба, представленную в виде различных графиков и диаграмм. Страница «Статистика» имеет две вкладки: «Мероприятия» и «Бюджет».

На вкладке «Мероприятия» реализован вывод в графическом виде различной информации о

проведенных мероприятиях, необходимой для ее последующего анализа руководством клуба. На этой вкладке, в частности, для заданного периода времени можно вывести: столбчатую диаграмму распределения мероприятий по типам, круговую диаграмму количества мероприятий по организаторам, линейный график пройденной дистанции для мероприятий различного типа, столбчатую диаграмму распределения мероприятий по

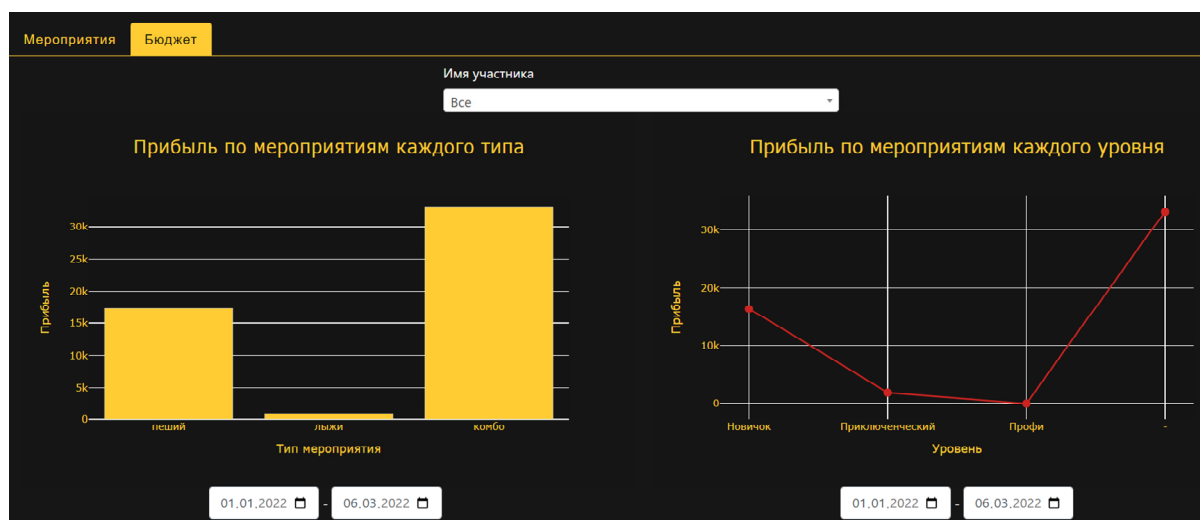


Рис. 10. Статистика по бюджету  
Fig. 10. Budget statistics

направлениям, круговую диаграмму количества участников по типам мероприятий и линейный график количества мероприятий различной сложности. При этом, как было показано ранее, у пользователя есть возможность изменения временного диапазона данных, на основании которых строятся диаграммы.

Вариант визуализации статистики мероприятий представлен на рис. 9.

Вкладка «Бюджет» предназначена для вывода информации, касающейся финансовой составляющей деятельности клуба. В частности, эта вкладка содержит: столбчатую диаграмму распределения прибыли по типам мероприятий; линейный график распределения прибыли по уровням сложности мероприятия; круговую диаграмму прибыли по организаторам мероприятий; столбчатую диаграмму распределения прибыли по различным направлениям мероприятия.

При построении диаграмм и графиков пользователь может фильтровать данные по участникам мероприятия и менять временной диапазон.

Вариант визуализации статистики по бюджету представлен на рис. 10.

**Выводы и заключение.** С помощью библиотеки Django и языков программирования Python и JavaScript разработано веб-приложение, относящееся к классу бизнес-приложений и предназначенное для оптимизации деятельности клуба активных путешественников. Отличительной особенностью приложения является наличие встроенной аналитической панели, позволяющей проводить предварительный анализ данных о деятельности клуба. Разработанное приложение позволило руководству клуба отказаться от использования распространенного подхода в деятельности малого и среднего бизнеса, заключающегося в решении отдельных задач с помощью различных программных средств.

#### Список литературы

1. Кузнецова И. Р., Букунов А. С., Букунов С. В. Автоматизированная система организации деятельности клуба активных путешественников // Изв. СПбГЭТУ «ЛЭТИ». 2021. № 1. С. 36–45.
2. Bobriakov I. A Comparative Analysis of Top 6 BI and Data Visualization Tools in 2018. URL: <https://www.kdnuggets.com/2018/02/comparative-analysis-top-6-bi-data-visualization-tools-2018.html> (дата обращения 04.04.2022).
3. Феррари А., Руссо. М. Анализ данных при помощи Microsoft Power BI и Power Pivot для Excel. М.: ДМК Пресс, 2020. 324 с.
4. QlikView. URL: <https://www.qlik.com/us/products/qlikview> (дата обращения 10.04.2022).
5. Klipfolio. URL: <https://www.klipfolio.com> (дата обращения 12.04.2022).
6. Microsoft Power BI. URL: <https://powerbi.microsoft.com> (дата обращения 15.04.2022).
7. Tableau – система интерактивной бизнес-аналитики (BI). URL: <https://biconsult.ru/products/tableau> (дата обращения 20.04.2022).
8. Plotly Open Source Graphing Library for Python. URL: <https://plotly.com/python> (дата обращения 16.11.2021).
9. Гарсия-Молина Г., Ульман Дж., Уилом Дж. Системы баз данных. Полный курс. М.: Вильямс, 2017. 415 с.
10. Тарасов С. В. СУБД для программиста. Базы данных изнутри. Омск: Соломон, 2015. 320 с.

- 
11. Златопольский Д. Основы программирования на языке Python. М.: ДМК-Пресс, 2018. 396 с.
  12. Дронов В. А. Django 2.1. Практика создания веб-сайтов на Python. СПб.: БХВ-Петербург, 2019. 672 с.
  13. Меле А. Django 2 в примерах. М.: ДМК-Пресс, 2019. 408 с.
  14. Дакетт Дж. HTML и CSS. Разработка и дизайн веб-сайтов. М.: Эксмо, 2020. 480 с.
  15. Никольский А. П. JavaScript на примерах. Практика, практика и только практика. СПб.: Наука и техника, 2018. 115 с.
  16. Браун Э. Изучаем JavaScript. Руководство по созданию современных веб-сайтов. М.: Альфа-книга, 2017. 245 с.
  17. Шварц Б., Ткаченко В., Зайцев П. MySQL по максимуму. Оптимизация, репликация, резервное копирование. СПб.: Питер, 2018. 336 с.
  18. Моретто С. Bootstrap в примерах. М.: ДМК-Пресс, 2017. 314 с.
- 

#### Информация об авторах

**Кузнецова Ирина Романовна** – студентка СПбГАСУ.

E-mail: irinak191998@mail.ru

**Букунов Александр Сергеевич** – ассистент СПбГАСУ.

E-mail: sasbukunov@yandex.ru

**Букунов Сергей Витальевич** – канд. техн. наук, доцент СПбГАСУ.

E-mail: sergeybukunov@yandex.ru

<https://orcid.org/0000-0001-9032-5084>

#### References

1. Kuznetsova I. R., Bukunov A. S., Bukunov S. V. Avtomatizirovannaya sistema organizacii deyatelnosti kluba aktivnyh puteshestvij // Izv. SPbGETU «LETI». 2021. № 1. S. 36–45. (In Russ.).
  2. Bobriakov I. A Comparative Analysis of Top 6 BI and Data Visualization Tools in 2018. URL: <https://www.kdnuggets.com/2018/02/comparative-analysis-top-6-bi-data-visualization-tools-2018.html> (data obrashcheniya 04.04.2022).
  3. Ferrari A., Russo. M. Analiz dannyh pri pomoshchi Microsoft Power BI i Power Pivot dlya Excel. М.: ДМК Press, 2020. 324 s. (In Russ.).
  4. QlikView. URL: <https://www.qlik.com/us/products/qlikview> (data obrashcheniya 10.04.2022).
  5. Klipfolio. URL: <https://www.klipfolio.com> (data obrashcheniya 12.04.2022).
  6. Microsoft Power BI. URL: <https://powerbi.microsoft.com> (data obrashcheniya 15.04.2022).
  7. Tableau – sistema interaktivnoj biznes analitiki (BI). URL: <https://biconsult.ru/products/tableau> data obrashcheniya 20.04.2022). (In Russ.).
  8. Plotly Open Source Graphing Library for Python. URL: <https://plotly.com/python> (data obrashcheniya 16.11.2021).
  9. Garsia-Molina G., Ul'man Dzh., Uilom Dzh. Sistemy baz dannyh. Polnyj kurs. М.: Vil'yams, 2017. 415 s. (In Russ.).
  10. Tarasov S. V. SUBD dlya programmista. Bazy dannyh iznutri. Omsk: Solomon, 2015. 320 s. (In Russ.).
  11. Zlatopol'skij D. Osnovy programmirovaniya na yazyke Python. М.: ДМК-Пресс, 2018. 396 s. (In Russ.).
  12. Dronov V. A. Django 2.1. Praktika sozdaniya veb-sajtov na Python. SPb.: BHV-Peterburg, 2019. 672 s. (In Russ.).
  13. Mele A. Django 2 v primerah. М.: ДМК-Пресс, 2019. 408 s. (In Russ.).
  14. Dakett Dzh. HTML i CSS. Razrabotka i dizajn veb-sajtov. М.: Eksmo, 2020. 480 s. (In Russ.).
  15. Nikol'skij A. P. JavaScript na primerah. Praktika, praktika i tol'ko praktika. SPb.: Nauka i tekhnika, 2018. 115 s. (In Russ.).
  16. Braun E. Izuchaem JavaScript. Rukovodstvo po sozdaniyu sovremennyh veb-sajtov. М.: Al'fa-kniga, 2017. 245 s. (In Russ.).
  17. SHvarc B., Tkachenko V., Zajtsev P. MySQL po maksimumu. Optimizaciya, replikaciya, rezervnoe kopirovanie. SPb.: Piter, 2018. 336 s. (In Russ.).
  18. Moretto S. Bootstrap v primerah. М.: ДМК-Пресс, 2017. 314 s.
- 

#### Information about the authors

**Irina R. Kuznetsova** – student of SPbGASU.

E-mail: irinak191998@mail.ru

**Alexandr S. Bukunov** – assistant of SPbGASU.

E-mail: sasbukunov@yandex.ru

**Sergey V. Bukunov** – Cand. Sci. (Eng.), Assistant Professor of SPbGASU.

E-mail: [sergeybukunov@yandex.ru](mailto:sergeybukunov@yandex.ru)

<https://orcid.org/0000-0001-9032-5084>

Статья поступила в редакцию 22.08.2022; принята к публикации после рецензирования 25.08.2022; опубликована онлайн 21.10.2022.

Submitted 22.08.2022; accepted 25.08.2022; published online 21.10.2022.

---