

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий  
Кафедра «Прикладная математика и информатика»

09.03.03 ПРИКЛАДНАЯ ИНФОРМАТИКА

ПРИКЛАДНАЯ ИНФОРМАТИКА В СОЦИАЛЬНОЙ СФЕРЕ

### БАКАЛАВРСКАЯ РАБОТА

на тему **Веб-приложение для управления мероприятиями (на примере ООО  
«ЭПАМ Систэмз»)**

Студент	<u>А.В. Шандуренко</u>	_____
Руководитель	<u>О.П. Михеева</u>	_____
Консультант по аннотации	<u>Н.В. Яценко</u>	_____

**Допустить к защите**

Заведующий кафедрой к.тех.н., доцент А.В. Очеповский \_\_\_\_\_

« \_\_\_\_\_ » \_\_\_\_\_ 20\_\_ г.

Тольятти 2017

# АННОТАЦИЯ

к выпускной квалификационной работе

Тема работы: Веб-приложение для управления мероприятиями (на примере ООО «ЭПАМ Систэмз»). Выполнена студентом группы ПИБ-1301 Шандуренко А.В.

Целью бакалаврской работы является разработка веб-приложения для управления мероприятиями для ООО «ЭПАМ Систэмз».

Объект исследования – процесс управления мероприятиями. Предмет исследования – автоматизация процессов по управлению мероприятиями.

Первая глава представляет собой описание функционального моделирования предметной области. В нее входит выбор технологии концептуального моделирования, разработка и анализ модели бизнес-процесса «КАК ЕСТЬ», анализ и сравнение существующих решений для управления мероприятиями, описание требований к проектируемой информационной системе, разработка модели бизнес-процесса «КАК ДОЖНО БЫТЬ».

Вторая глава представляет собой описание логического проектирования информационной системы. В нее входит выбор технологии логического моделирования, разработка логической модели информационной системы, описание требований к аппаратно-программному обеспечению информационной системы.

Третья глава представляет собой описание физического проектирования информационной системы. В нее входит выбор архитектуры информационной системы, технологии разработки программного обеспечения, СУБД, рассмотрение основного функционала разработанной информационной системы и ее тестирование.

Результатом выполнения ВКР является разработанная информационная система управления мероприятиями для ООО «ЭПАМ Систэмз».

В работе использовано 6 таблиц, 25 рисунков, список использованной литературы содержит 79 источников. Объем выполненной бакалаврской работы

## **ABSTRACT**

The title of the graduation work is «Web-application for Event Management (on the Example of OOO «EPAM Systems»»).

The aim is to develop the web-application of event management for OOO «EPAM Systems».

The object is event management process.

The subject is automation of event management processes.

The first chapter describes the functional modeling of the subject area, choice of conceptual modeling technology, development and analysis of the business process model «AS IS», comparative analysis of existing information systems for event management and their comparison, the requirements for the projected information system, development of a business process model «HOW TO BE».

The second chapter depicts the logical design of the information system, the choice of logical modeling technology, development of a logical model of the information system, includes a diagram of use cases and a class diagram, development of a conceptual and logical model of the database, as well as the hardware and software requirements for the information system.

The third chapter addresses to the physical design of the information system, the choice of information system architecture, software development technology, the choice of DBMS, development of a physical data model for the selected DBMS, consideration of main functional of the developed information system, testing of the developed information system.

The result of the graduation work is developed web-application of event management for OOO «EPAM Systems».

The work presents 6 tables, 25 figures, the list of references contains 79 sources. The total amount of the graduation qualification work is 61 pages.

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	4
ГЛАВА 1 ФУНКЦИОНАЛЬНОЕ МОДЕЛИРОВАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ .....	6
1.1 Техничко-экономическая характеристика предметной области.....	6
1.2 Концептуальное моделирование предметной области .....	7
1.2.1 Выбор технологии концептуального моделирования предметной области .....	7
1.2.2 Разработка и анализ модели бизнес-процесса «КАК ЕСТЬ» .....	9
1.2.3 Обоснование необходимости автоматизированного варианта решения и формирование требований к проектируемой информационной системе .....	13
1.3 Анализ существующих разработок на предмет соответствия сформулированным требованиям .....	14
1.4 Постановка задачи на разработку проекта создания автоматизированной информационной системы .....	18
1.5 Разработка модели бизнес-процесса «КАК ДОЛЖНО БЫТЬ» .....	19
Выводы по главе 1.....	21
ГЛАВА 2 ЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ АВТОМАТИЗИРОВАННОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ .....	22
2.1 Выбор технологии логического моделирования автоматизированной информационной системы .....	22
2.2 Разработка логической модели автоматизированной информационной системы .....	23
2.3 Разработка концептуальной и логической моделей данных автоматизированной информационной системы.....	26
2.4 Требования к аппаратно-программному обеспечению автоматизированной информационной системы .....	28
Выводы по главе 2.....	29
ГЛАВА 3 ФИЗИЧЕСКОЕ ПРОЕКТИРОВАНИЕ АВТОМАТИЗИРОВАННОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ .....	30

3.1 Выбор архитектуры автоматизированной информационной системы .....	30
3.2 Выбор технологии разработки программного обеспечения автоматизированной информационной системы.....	31
3.3 Выбор системы управления базами данных автоматизированной информационной системы .....	33
3.4 Разработка физической модели данных автоматизированной информационной системы .....	34
3.5 Описание функциональности автоматизированной информационной системы и ее тестирование.....	35
Выводы по главе 3.....	41
ЗАКЛЮЧЕНИЕ .....	42
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ .....	43
ПРИЛОЖЕНИЕ А .....	51
ПРИЛОЖЕНИЕ Б.....	54
ПРИЛОЖЕНИЕ В .....	57
ПРИЛОЖЕНИЕ Г.....	58
ПРИЛОЖЕНИЕ Д.....	59

## ВВЕДЕНИЕ

Управление мероприятиями представляет собой процесс разработки и принятия плана его проведения. Управление мероприятиями является важным инструментом для достижения определенных целей.

Независимо от причины создания мероприятия, зачастую нет достаточного запаса времени для тщательного планирования и организации мероприятия, но в тоже время есть желание провести его на высоком уровне, профессионально, с минимальными денежными и временными затратами, при этом добиться максимальной эффективности.

Планирование мероприятий является сложным, но в тоже время эффективным инструментом для организации любого вида мероприятий на высоком уровне.

Процесс управления мероприятием подразумевает наличие стадии планирования, когда определяются дата и время, место проведения мероприятия, обсуждение других организационных вопросов, связанных с его проведением. Данное действие зачастую проводится очно (для этого необходимо личное присутствие организаторов планируемого мероприятия), но также может проводиться и дистанционно. Для удобного планирования мероприятий необходима специальная информационная система, способная предоставить организатором площадку для управления мероприятиями.

**Целью** бакалаврской работы является разработка веб-приложения для управления мероприятиями для ООО "ЭПАМ Систэмз".

**Объектом исследования** в данной работе является процесс управления мероприятиями.

**Предмет исследования** – автоматизация процессов по управлению мероприятиями.

Для достижения поставленной цели в работе решаются следующие **задачи**:

- изучение предметной области;
- анализ проблем в существующем бизнес-процессе;

- анализ существующих решений на рынке;
- обоснование выбора архитектуры информационной системы;
- обоснование выбора технологий разработки;
- обоснование выбора СУБД;
- тестирование разработанной информационной системы.

При написании работы использовались методы исследования: анализ, системный подход, методы моделирования бизнес-процессов, методы моделирования информационной системы и ее баз данных при помощи CASE-средств. Также при написании работы использовались учебные пособия, стандарты по моделированию, проектированию и разработке информационных систем.

Первая глава представляет собой описание функционального моделирования предметной области. В нее входит выбор технологии концептуального моделирования, разработка и анализ модели бизнес-процесса «КАК ЕСТЬ», анализ и сравнение существующих решений для управления мероприятиями, описание требований к проектируемой информационной системе, разработка модели бизнес-процесса «КАК ДОЖНО БЫТЬ».

Вторая глава представляет собой описание логического проектирования информационной системы. В нее входит выбор технологии логического моделирования, разработка логической модели информационной системы, описание требований к аппаратно-программному обеспечению информационной системы.

Третья глава представляет собой описание физического проектирования информационной системы. В нее входит выбор архитектуры информационной системы, технологии разработки программного обеспечения, СУБД, рассмотрение основного функционала разработанной информационной системы и ее тестирование.

Результат проделанной работы планируется для внедрения в ООО «ЭПАМ Систэмз».

# ГЛАВА 1 ФУНКЦИОНАЛЬНОЕ МОДЕЛИРОВАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

## 1.1 Технико-экономическая характеристика предметной области

EPAM Systems – американская ИТ-компания, начала свою деятельность в 1993 году. Ведущий разработчик программного обеспечения в Центральной и Восточной Европе.

Основные направления деятельности компании:

- разработка программного обеспечения;
- тестирование программного обеспечения;
- интеграция приложений;
- портирование и миграция приложений;
- ИТ-консалтинг;
- разработка цифровых стратегий.

Филиал компании в Тольятти начал свою деятельность в 2008 году. На сегодняшний день в штате тольяттинского офиса более 60 технических специалистов различных направлений, таких как: PHP, .NET, Java, Oracle, тестирование и QA.

Организационная структура представлена на рисунке 1.1.

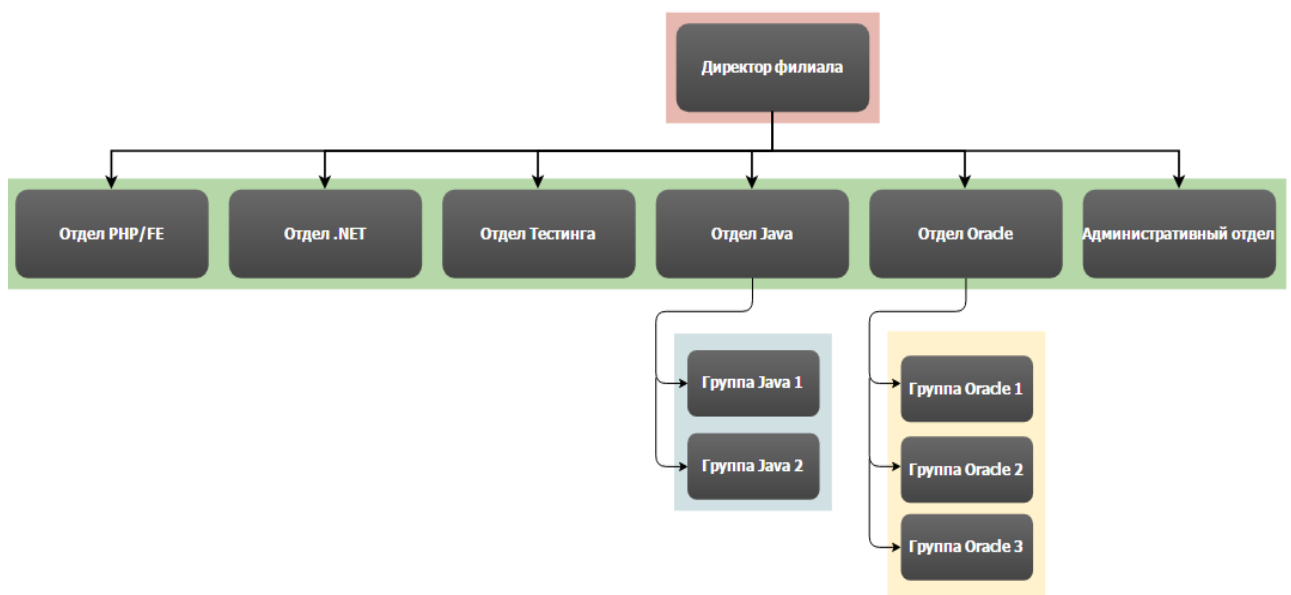


Рисунок 1.1 – Схема организации ООО «ЭПАМ Систэмз»



В качестве предметной области выступает отдельный вид деятельности объединения. Объединение представляет собой всех сотрудников данной организации, заинтересованных в проведение послерабочего досуга. В качестве отдельного вида деятельности выступает процесс организации послерабочего досуга, который включает в себя планирование и проведение мероприятия. Рассмотрим подробнее процесс планирования мероприятия. На данной стадии ответственное лицо, которое выступает в роли организатора мероприятия, составляет список потенциальных участников планируемого мероприятия, собирает их в одном месте и проводит опрос среди присутствующих с целью определения места, даты и времени проведения мероприятия или же, пишет сообщение на электронную почту, также возможен вариант прохождения опроса через телефонную связь.

После того как была описана технико-экономическая характеристика деятельности, перейдем к процессу концептуального моделирования, а именно, сделаем выбор технологии концептуального моделирования предметной области.

## **1.2 Концептуальное моделирование предметной области**

### **1.2.1 Выбор технологии концептуального моделирования предметной области**

Моделирование предметной области является одним из самых важных этапов на стадии проектирования информационной системы. Необходимо провести сравнительный анализ наиболее известных нотаций концептуального моделирования информационных систем: UML, IDEF0 и ARIS.

UML (Unified Modeling Language) – представляет собой язык графического описания предназначенный для объектно-ориентированного моделирования бизнес-процессов в основном для программных систем. Используется для создания абстрактной модели системы, которая называется UML моделью [20].

IDEF0 (Integration Definition for Function Modeling) – методология функционального моделирования бизнес-процессов. IDEF0 отображает структуру и функции системы, а также потоки информации которые связывают эти функции [21].

ARIS (Architecture of Integrated Information Systems) – методология для моделирования бизнес-процессов организаций и программный продукт немецкой компании Software AG [24].

Результат сравнительного анализа методологий представлен в таблице 1.1.

Таблица 1.1 – Сравнительный анализ методологий

<b>Критерий сравнения</b>	<b>Нотация IDEF0</b>	<b>Нотация UML</b>	<b>Нотация ARIS</b>
Легкость изучения	Легко	Средне	Сложно
Удобство по созданию моделей	Легко	Сложно	Сложно
Подход к проектированию	Функциональный	Объектно-ориентированный	Процессный
Система хранения данных модели	Модели хранятся в файлах	Модели хранятся в файлах	Объектная база данных
Возможность декомпозиции	Неограниченная	Неограниченная	Неограниченная

Исходя из анализа таблицы 1.1, делаем выводы о том, что IDEF0 является наиболее подходящей нотацией, так как она легка в изучении и удобна для создания моделей, по сравнению с UML и ARIS, которые в свою очередь сложны в изучении, имеют определенные сложности при создании моделей, также модели в UML и ARIS часто трудны для восприятия и изучения.

Основа методологии IDEF0 заключается в графическом описании взаимосвязанных протекающих процессов. Модель IDEF0 представляет собой блок, который выполняет некоторую бизнес-функцию. Четыре стороны этого блока имеют различное предназначение.

После того как был осуществлён выбор технологии концептуального моделирования, перейдем к разработке модели бизнес-процесса «КАК ЕСТЬ».

## 1.2.2 Разработка и анализ модели бизнес-процесса «КАК ЕСТЬ»

Бизнес-процесс – это совокупность взаимосвязанных действий, направленных на создание определенного продукта. Для наглядности бизнес-процессы зачастую визуализируются в виде блок-схем бизнес-процессов.

Модель бизнес-процесса «КАК ЕСТЬ» представляет собой модель существующего состояния организации. Данная модель систематизирует происходящие в данный момент процессы, а также используемые информационные объекты. На основе анализа данной модели выявляются узкие места в организации, узкие места в взаимодействии бизнес-процессов, после этого определяется необходимость внесения тех или иных изменений в существующей структуре. Данную модель еще называют функциональной и делают при помощи различных CASE-средств с использованием различных графических нотаций. На этапе создания модели бизнес-процесса «КАК ЕСТЬ» важно создать максимально похожую модель, которая основана на реальных потоках процессов.

Ramus – это инструмент для разработки моделей бизнес-процессов.

Контекстная модель «КАК ЕСТЬ» процесса «Проведение мероприятия» в нотации IDEF0, выполненная с использованием CASE-средства Ramus, представлена на рисунке 1.2.

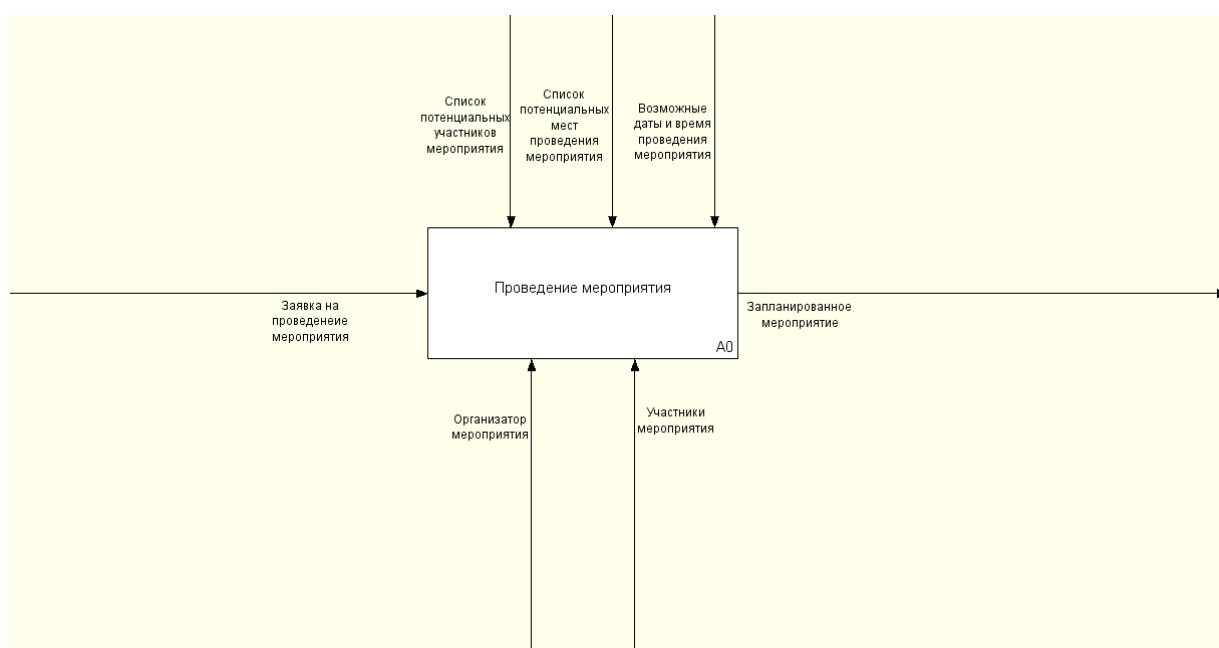


Рисунок 1.2 – Контекстная модель «КАК ЕСТЬ»

Рассмотрим подробнее процесс А0 «Проведение мероприятия», на вход процесса поступает заявка на проведение какого-либо мероприятия, на выходе из процесса получаем запланированное мероприятие.

Для лучшего понимания особенностей главного процесса выполним его декомпозицию, то есть разделение целого процесса на части. Модель проведенной декомпозиции приведена на рисунке 1.3.

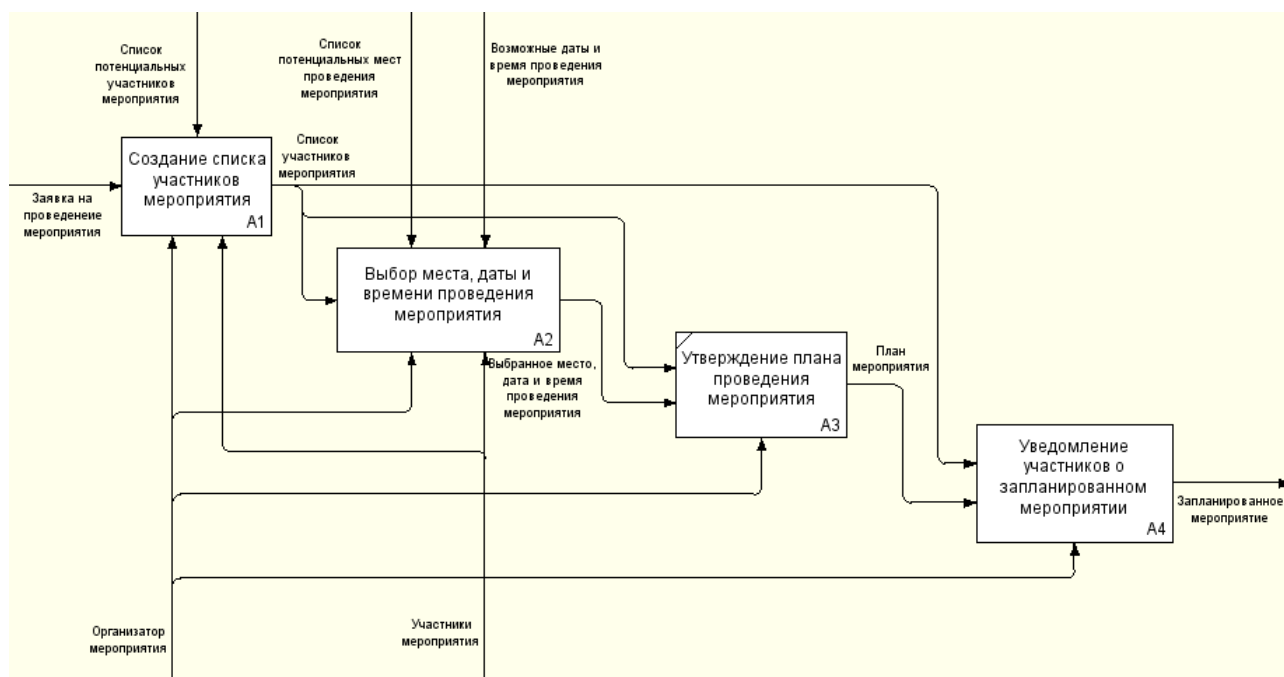


Рисунок 1.3 – Декомпозиция контекстной модели «КАК ЕСТЬ»

После проведения декомпозиции контекстной модели «КАК ЕСТЬ», рассмотрим более подробно каждый из процессов:

1. Процесс А1 «Создание списка участников мероприятия» подразумевает, что организатор получает заявку на проведение мероприятия от потенциальных участников, или, организатор сам изъявляет желание провести какое-либо мероприятие, после этого организатор проводит опрос потенциальных участников и формирует список участников.

2. Процесс А2 «Выбор места, даты и времени проведения мероприятия» подразумевает, что организатор проводит опрос участников мероприятия с целью выбора места, даты и времени проведения мероприятия.

3. Процесс А3 «Утверждение плана проведения мероприятия» представляет собой, утверждение плана проведения мероприятия, а точнее, утверждение списка участников мероприятия, а также, места, даты и времени проведения мероприятия.

4. Процесс А4 «Уведомление участников о запланированном мероприятии» является заключительным и подразумевает, что организатор уведомляет всех участников мероприятия о запланированном мероприятии, сообщает место, дату и время проведения мероприятия.

По итогам проведенного анализа, делаем вывод что процессы А1, А2 и А4 могут быть автоматизированы с целью предоставления организаторам и участникам возможности дистанционного прохождения этапов планирования мероприятия, а также, уменьшения затрат ресурсов (материальных и трудовых) как со стороны организаторов, так и со стороны участников.

Проведем декомпозицию процессов, подлежащих автоматизации. Это необходимо для лучшего понимания особенностей их протекания.

Модель проведенной декомпозиции процесса А1 «Создание списка участников мероприятия» представлена на рисунке 1.4.

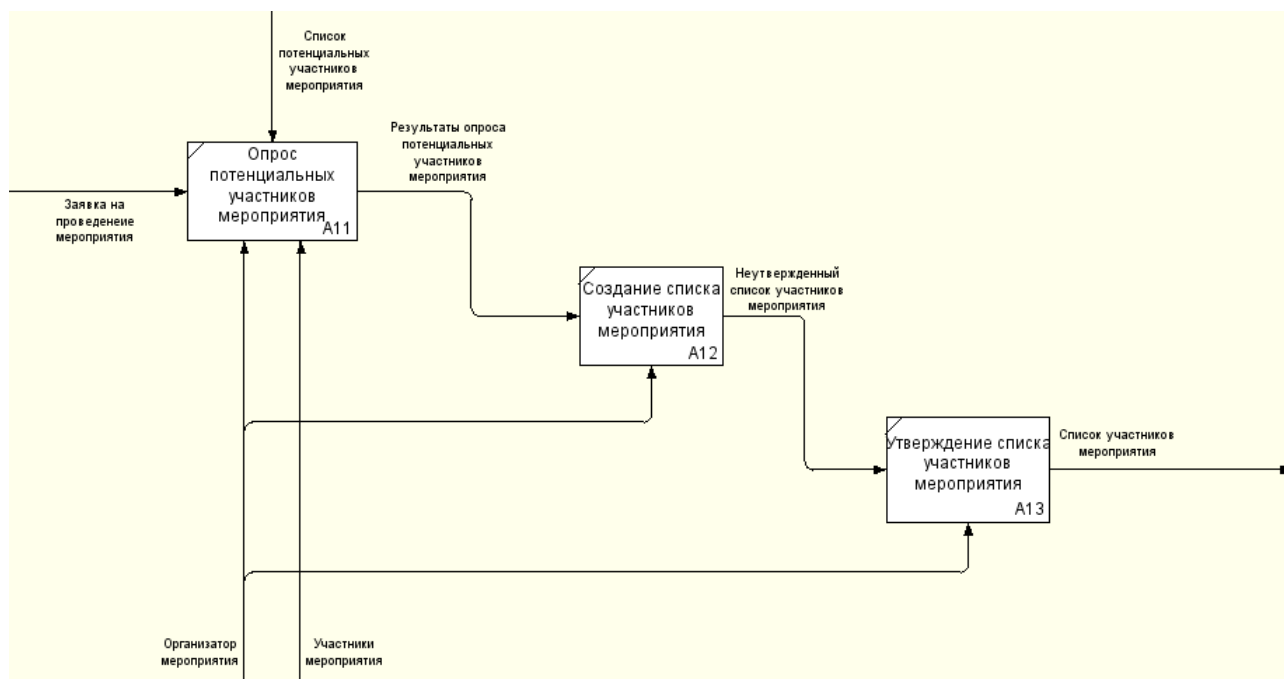


Рисунок 1.4 – Декомпозиция процесса А1 «Создание списка участников мероприятия»

Далее проведем декомпозицию процесса А2 «Выбор места, даты и времени проведения мероприятия». Результат представлен на рисунке 1.5.

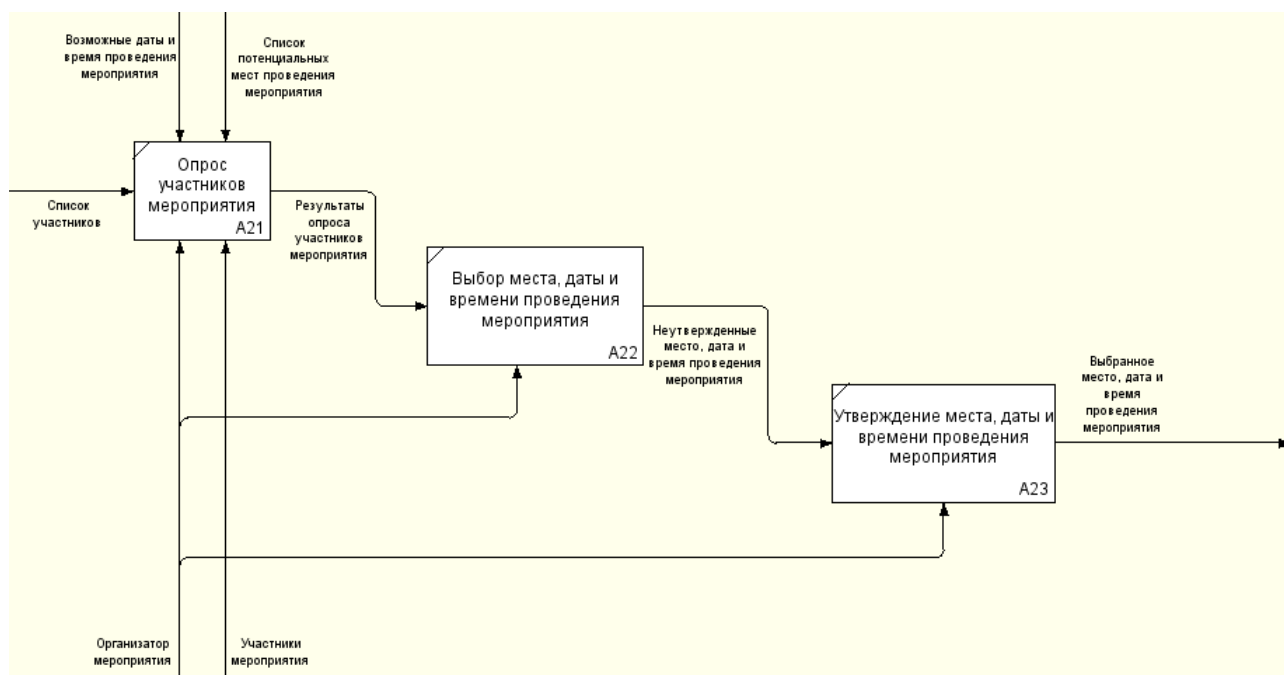


Рисунок 1.5 – Декомпозиция процесса А2 «Выбор места, даты и времени проведения мероприятия»

И наконец, представим результат декомпозиции последнего процесса А4 «Уведомление участников о запланированном мероприятии», модель декомпозиции представлена на рисунке 1.6.

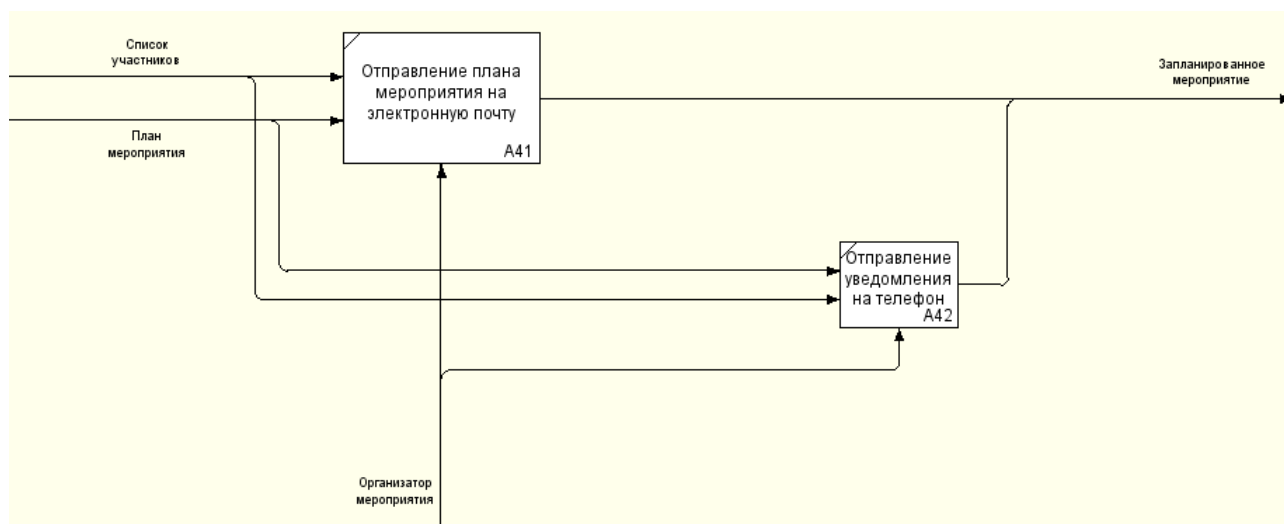


Рисунок 1.6 – Декомпозиция процесса А4 «Уведомление участников о запланированном мероприятии»

После того, как была разработана модель бизнес-процесса «КАК ЕСТЬ», был проведен ее анализ с последующей декомпозицией на подпроцессы, перейдем к обоснованию необходимости автоматизированного варианта решения и формированию требований к проектируемой информационной системе.

### **1.2.3 Обоснование необходимости автоматизированного варианта решения и формирование требований к проектируемой информационной системе**

После проведенного анализа процессов А1 «Создание списка участников мероприятия», А2 «Выбор места, даты и времени проведения мероприятия» и А4 «Уведомление участников о запланированном мероприятии», входящих в А0 «Проведение мероприятия» были выявлены следующие недостатки:

1. Необходимость очного присутствия организаторов и, при необходимости, участников на стадии планирования мероприятия, либо, организатору приходится звонить или писать письма на почту потенциальным участникам, а так как график у всех разных, то данный процесс трудоемкий как по времени, так и со стороны ресурсов, затрачиваемых на данный процесс (оплата связи, интернета).

2. Неэффективный документооборот, выражающийся в необходимости вести списки. С учетом того, что данные постоянно меняются, приходится постоянно корректировать списки, а так как для планирования мероприятия важно иметь актуальную информацию, процесс получается затратным.

Согласно проведенному анализу делаем вывод о том, что необходимость разработки автоматизированного варианта решения является целесообразной, поскольку оно позволит решить существующие на данный момент проблемы.

После того как были выявлены все недостатки существующего бизнес-процесса, необходимо описать требования к проектируемой информационной системе. Перейдем к описанию требований, которым должна будет соответствовать будущая информационная система.

Для определения требований к проектируемой информационной системе используем модель FURPS+. Акроним FURPS расшифровывается следующим образом:

- Functionality (функциональность);
- Usability (удобство использования);
- Reliability (надежность);
- Performance (производительность);
- Supportability (сопровождаемость).

FURPS+ – это расширенная версия классификации требований к программным системам. Символ «+» расширяет модель FURPS, добавляя ей ограничения разделенные на следующие группы:

- Implementation (ограничения разработки);
- Interface (ограничения на интерфейсы);
- Design (ограничения проектирования);
- Physical (физические ограничения).

Таблица с требованиями к проектируемой информационной системе представлена в приложении А.

После того как были сформулированы и описаны требования к проектируемой информационной системы, перейдем к анализу существующих решений в данной предметной области.

### **1.3 Анализ существующих разработок на предмет соответствия сформулированным требованиям**

Проведем сравнительный анализ наиболее известных информационных систем для управления мероприятиями. Это необходимо для того, чтобы более точно определить дальнейшие направления проектирования будущей информационной системы, а также для того чтобы не повторять чужих ошибок при разработке.



Далее рассмотрим существующие аналоги информационных систем для управления мероприятиями.

1. TimePad, пользовательский интерфейс представлен на рисунке 1.7.

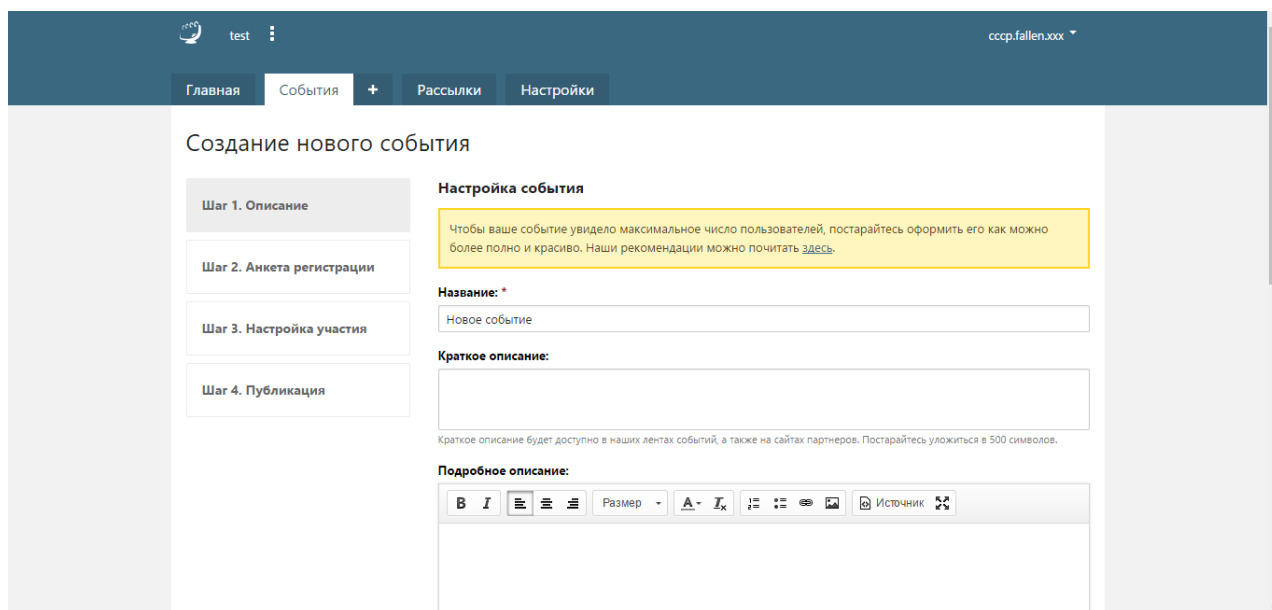


Рисунок 1.7 – Создание нового мероприятия в Timepad

Располагается по адресу <https://www.welcome.timepad.ru>, представляет собой сервис для организации и продвижения мероприятий. Данный сервис обладает рядом преимуществ:

- возможность привлечения дополнительной аудитории для мероприятия за счет рекламы на сайте;
- возможность создания мероприятий с входом по билетам, продажа билетов через сервис;
- доступ к статистике продаж и регистраций;
- расширенный поиск по мероприятиям;
- почтовые рассылки по аудитории.

К сожалению, данный сервис также обладает и рядом недостатков, таких как:

- частично платный функционал;
- необходимость платить отчисления за продажи билетов.

2. Eventboost, пользовательский интерфейс представлен на рисунке 1.8.

The screenshot shows the 'Event Details' page in the Eventboost application. At the top, there is a navigation bar with 'Event Details' selected. Below it is a green banner with a message about the trial period. The main form contains several sections: 'Event name' with a text input and a 'Hashtag' input; 'Event type' with a dropdown menu; 'Event language' with a dropdown and a note 'Active language(s): EN'; and 'Event options' with two checkboxes, 'Registration' (checked) and 'Enable guest to add partners'. A 'Save' button is located at the top right of the form area. The top of the page features the Eventboost logo, 'New Event' and 'My Events' buttons, a notification bell, a 'BUY' button, and a user profile with the email 'cccc.fallen.xxx@gmail.com' and a language dropdown set to 'EN'.

Рисунок 1.8 – Создание нового мероприятия в Eventboost

Располагается по адресу <https://www.eventboost.com>, представляет собой сервис для организации мероприятий. Из преимуществ данного сервиса можно выделить:

- широкий функционал для создания страницы мероприятия;
- расширенный поиск по мероприятиям.

Данный сервис также обладает рядом недостатков:

- сервис платный;
- необходимо время, чтобы разобраться как пользоваться данным сервисом;
- отсутствие русскоязычного интерфейса;
- низкая скорость работы.

3. Вконтакте, пользовательский интерфейс представлен на рисунке 1.9.

Располагается по адресу <https://www.vk.com>, представляет собой социальную сеть, в функционале которой есть возможность создания страниц мероприятий.

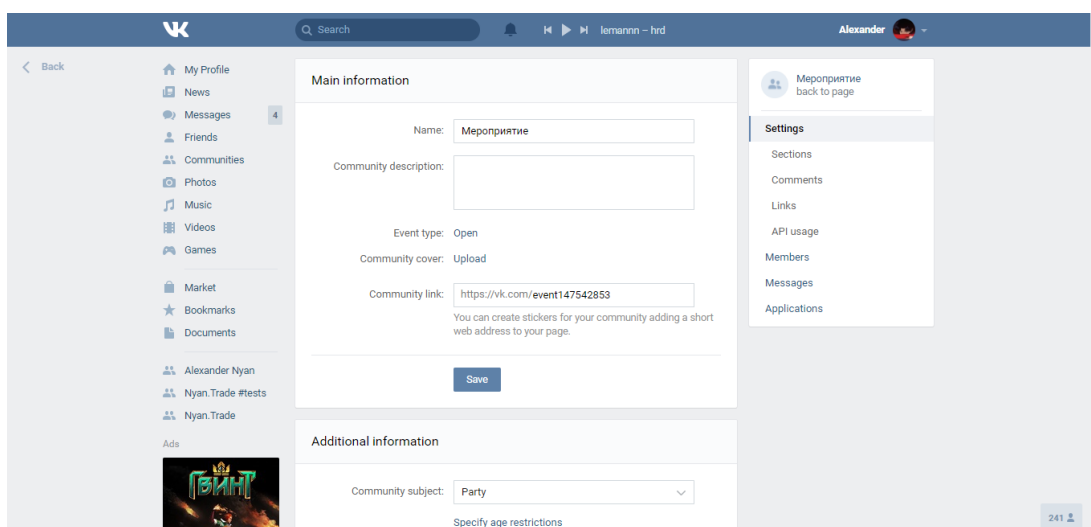


Рисунок 1.9 – Создание нового мероприятия в Вконтакте

Из преимуществ можно выделить:

- удобный и интуитивно-понятный интерфейс;
- возможность привлечения дополнительной аудитории для мероприятия за счет рекламы на сайте;
- возможность добавления на страницу мероприятия медиа-информации, такой как фотографии, видеозаписи и аудиозаписи;
- популярность данной социальной сети, в следствие чего, наличие большой и активной аудитории.

Но также существуют и недостатки в виде:

- довольно скудный функционал для планирования и создания мероприятий;
- данная социальная сеть не предназначена для такого типа задач.

Результат анализа информационных систем для управления мероприятиями представлен в таблице 1.3.

Таблица 1.3 – Сравнение существующих аналогов информационных систем

№	Требование	TimePad	Eventboost	ВКонтакте
1	Просмотр списка мероприятий доступных для посещения	+	-	+
2	Поиск мероприятий по определенным параметрам	+	+	-

Продолжение таблицы 1.3

№	Требование	TimePad	Eventboost	ВКонтакте
3	Планирование мероприятия	-	-	-
4	Создание мероприятия	+	+	+
5	Редактирования созданного мероприятия	+	+	+
6	Приглашение пользователей поучаствовать в мероприятии	+	+	+
7	Возможность поделиться информацией о мероприятии в социальных сетях	-	-	+
8	Возможность принимать заявки на участие в мероприятии	-	-	-
9	Добавление пользователей себе в контакт-лист	-	-	+
10	Сообщать участникам о одобрении заявки	-	-	-
Итого:		5/10	4/10	6/10

Исходя из анализа таблицы 1.3, делаем вывод, что каждая из существующих информационных систем для управления мероприятиями имеет определенные недостатки.

После того, как был проведен сравнительный анализ существующих информационных систем для управления мероприятиями, необходимо сделать вывод о необходимости проектирования и разработки новой информационной системы, а также описать цель ее создания и основные требования к будущей информационной системе.

#### **1.4 Постановка задачи на разработку проекта создания автоматизированной информационной системы**

Исходя из анализа, проведенного в пункте 1.3 делаем вывод, что ни одна из рассмотренных уже существующих информационных систем в данной предметной области не удовлетворяет всем необходимым требованиям. Таким образом, еще раз подтверждается необходимость проектирования и разработки новой информационной системы для управления мероприятиями.

Сформулируем цель и основные требования к будущей информационной системе.

Целью создания информационной системы является предоставление удобного инструмента для управления мероприятиями, автоматизация и

упрощение рутинных процессов, возникающих в процессе планирования и организации мероприятий.

Для достижения поставленной цели были сформированы основные требования к будущей информационной системе:

1. Интуитивно-понятный интерфейс.
2. Поиск мероприятий по определенным параметрам.
3. Возможность поделиться информацией по мероприятии в социальных сетях.
4. Создание и редактирование страниц мероприятий.
5. Возможность добавлять пользователей к себе в контакт-лист.

Организаторы мероприятий тратят много времени и сил на выполнение необходимых задач. Именно поэтому необходима автоматизированная информационная система управления мероприятиями, которая позволит сократить время, затрачиваемое на процессы управления мероприятиями, а также обеспечит более качественное и надёжное выполнение процессов.

После того как была выполнена постановка задачи на разработку, описана цель ее создания и основные требования к будущей информационной системе, перейдем к разработке модели бизнес-процесса «КАК ДОЛЖНО БЫТЬ».

### **1.5 Разработка модели бизнес-процесса «КАК ДОЛЖНО БЫТЬ»**

Была построена контекстная модель «КАК ДОЛЖНО БЫТЬ», данная модель отображает будущее предполагаемое состояние предметной области, она создается на основе контекстной модели «КАК ЕСТЬ» с устранением недостатков в существующей организации бизнес-процессов, а также, с их усовершенствованием и оптимизацией. Это достигается за счет устранения выявленных во время анализа контекстной модели «КАК ЕСТЬ» узких мест. В традиционном реинжиниринге рекомендуется проводить автоматизацию бизнес-процессов именно на основании модели «КАК ДОЛЖНО БЫТЬ», также, как и проектировать информационную систему. Это позволяет снизить риск проявления автоматизации как исключительно источника затрат из-за

автоматизации несовершенных процессов. Контекстная модель «КАК ДОЛЖНО БЫТЬ» представлена на рисунке 1.10. По сравнению с моделью «КАК ЕСТЬ», добавился новый механизм – «Информационная система», который представляет собой информационную систему для управления мероприятиями отвечает за автоматизацию бизнес-процессов управления мероприятиями.

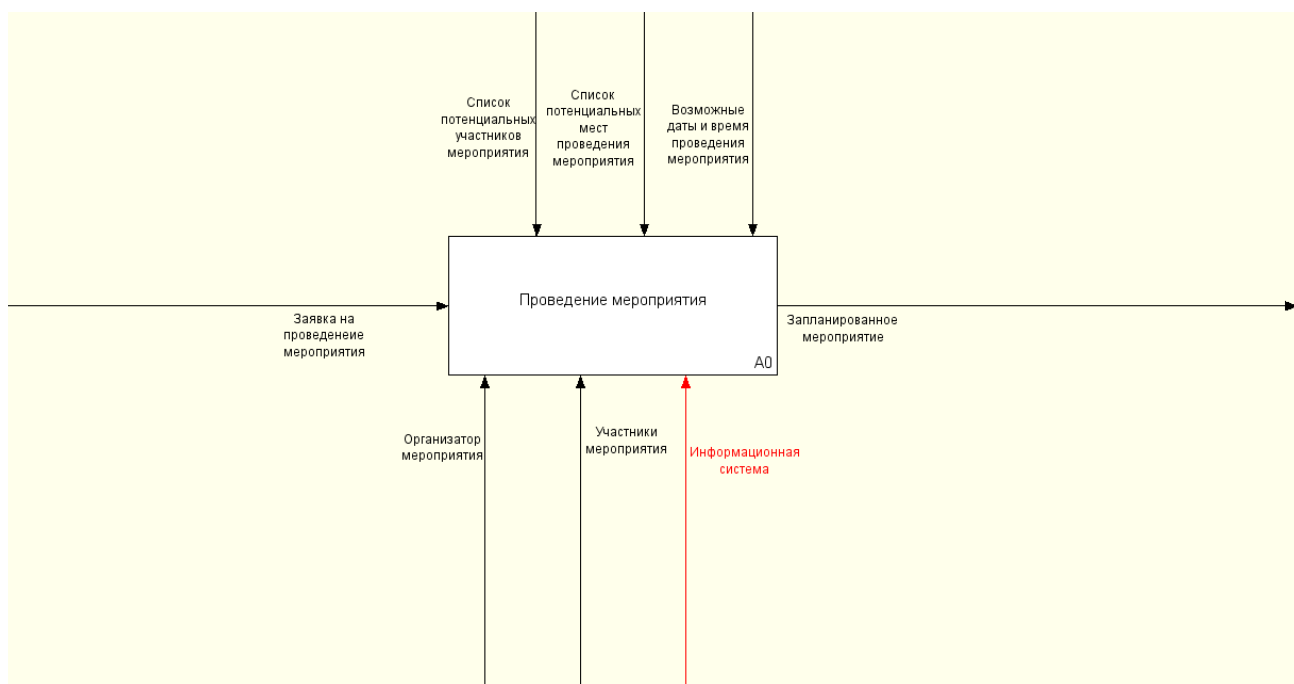


Рисунок 1.10 – Контекстная модель «КАК ДОЛЖНО БЫТЬ»

С помощью информационной системы организаторы смогут создавать страницу мероприятия и приглашать других пользователей принять участие в нем. Пользователи смогут просматривать список доступных для посещения мероприятий и отправлять заявки на участие в них.

Работать с информационной системой будут организаторы мероприятий и пользователи, которые хотят принять участие в планируемых мероприятиях. Для работы с информационной системой необходимо будет пройти процедуру авторизации, в случае если пользователь не зарегистрирован в системе, пользователю необходимо будет пройти процедуру регистрации.

Также проведем декомпозицию контекстной модели «КАК ДОЛЖНО БЫТЬ» результат декомпозиции представлен на рисунке 1.11.

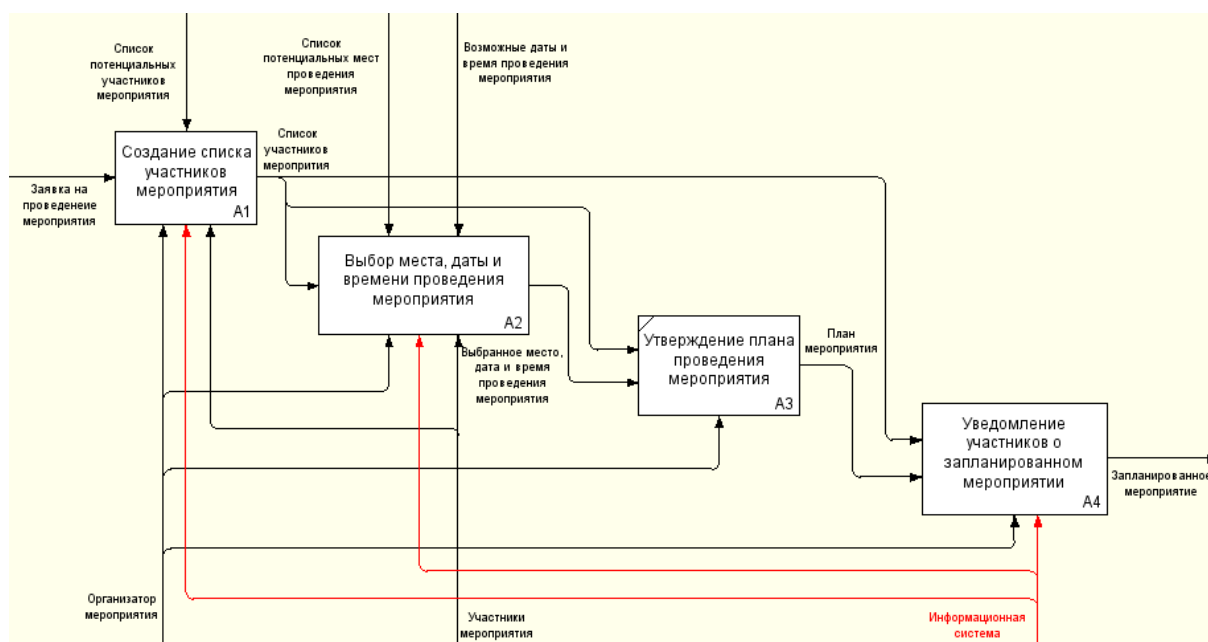


Рисунок 1.11 – Декомпозиция контекстной модель «КАК ДОЛЖНО БЫТЬ»

После того как была разработана контекстная модель бизнес-процесса «КАК ДОЛЖНО БЫТЬ», перейдем к поведению итогов второй главы.

### Выводы по главе 1

Первая глава представляет собой описание функционального моделирования предметной области. Был осуществлен выбор технологии концептуального моделирования, также была осуществлена разработка и анализ модели бизнес-процесса «КАК ЕСТЬ», после чего был сделан вывод что часть процессов нуждается в автоматизации. Также был проведен сравнительный анализ существующих информационных систем для управления мероприятиями и их сравнение, согласно проведенному анализу ни одна из рассмотренных информационных систем не удовлетворяет всем необходимым требованиям, исходя из этого делаем вывод что разработка новой информационной системы действительно необходима. Были сформированы и описаны требования к проектируемой информационной системе, после этого была выполнена постановка задачи на разработку новой информационной системы и разработка модели бизнес-процесса «КАК ДОЛЖНО БЫТЬ».

## **ГЛАВА 2 ЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ АВТОМАТИЗИРОВАННОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ**

### **2.1 Выбор технологии логического моделирования автоматизированной информационной системы**

Логическое моделирование предметной области используется для описания процесса и его развития, создаются сценарии, которые направлены на увязку логической последовательности происходящих событий. Логическое моделирование представляет собой анализ логики развития прогнозируемого объекта и создание на этой основе моделей-образов.

Логическое моделирование подразумевает создание логической модели информационной системы и модели ее базы данных на основе концептуальной модели «КАК ДОЛЖНО БЫТЬ», которая была описана в первой главе, пункта 1.5. Логическая модель информационной системы будет описана при помощи диаграммы вариантов использования и диаграммы классов. Проектирование базы данных подразумевает создание концептуальной и логической модели базы данных информационной системы. Создание диаграммы вариантов использования и диаграммы классов, а также создание концептуальной и логической модели баз данных информационной системы будет осуществлено при помощи CASE-средства StarUML с использованием языка моделирования UML. Использование CASE-средств позволяет значительно повысить эффективность работы, сокращается время разработки и уменьшается число программных ошибок.

StarUML – это средство UML-моделирования, данный инструмент моделирования поддерживает UML 2.0 (унифицированный язык моделирования) и MDA (модельно-управляемая архитектура), предоставляет возможности для разработки одиннадцати различных видов диаграмм. Функционал данного продукта расширяется за счет установки сторонних плагинов.



После того как была выбрана технология логического моделирования перейдем к разработке логической модели информационной системы.

## 2.2 Разработка логической модели автоматизированной информационной системы

Процесс создания логической модели информационной системы подразумевает переход от контекстной модели «КАК ДОЛЖНО БЫТЬ» к диаграмме вариантов использования. Данная диаграмма позволяет выявить основные процессы, протекающие в системе, выявить их взаимосвязь, а также способствует выделению функциональной структуры информационной системы. Также необходимо разработать диаграмму классов, которая отображает структуру информационной системы в контексте классов объектно-ориентированного программирования, их взаимосвязи между ними, атрибуты и методы.

Разработанная диаграмма вариантов использования представлена на рисунке 2.1.

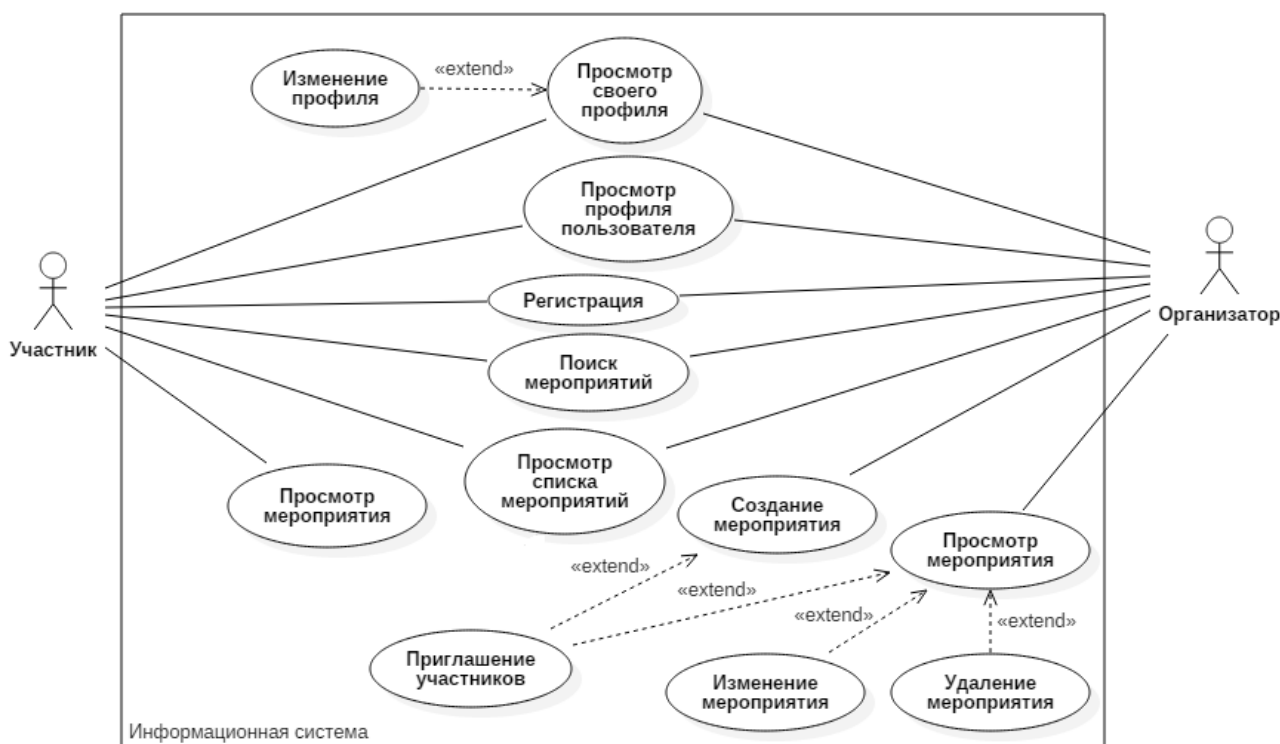


Рисунок 2.1 – Диаграмма вариантов использования

На диаграмме, изображенной на рисунке 2.1 представлены следующие действующие лица (актеры):

- организатор – пользователь веб-приложения, который создает страницы планируемых к проведению мероприятий, заполняет необходимую организационную информацию, рассылает другим пользователям приглашения принять участие в них, в общем – занимается процессом планирования, организации и проведения мероприятий;
- участник – пользователь веб-приложения, который просматривает список доступных для посещения мероприятий, оставляет заявки на их посещение, добавляет других пользователей себе в контакт-лист, например, для дальнейшего общения, также участник может принимать участие в обсуждении различных деталей мероприятий в котором пользователь планирует участвовать.

В таблице 2.1 отражена краткая характеристика прецедентов (т.е. самих вариантов использования) диаграммы.

Таблица 2.1 – Краткая характеристика прецедентов

<b>Прецедент</b>	<b>Характеристика</b>
Регистрация	Регистрация пользователя в системе с последующим созданием его профиля.
Создание мероприятия	Организатор создает страницу мероприятия, заполняет необходимую информацию.
Изменение мероприятия	Организатор вносит изменения в информацию на странице мероприятия.
Удаление мероприятия	Организатор удаляет страницу запланированного мероприятия.
Просмотр мероприятия	Пользователь просматривает страницу запланированного мероприятия.
Просмотр списка мероприятий	Пользователь просматривает список возможных для посещения мероприятий.
Поиск мероприятий	Пользователь выполняет поиск мероприятий по определенным параметрам.
Приглашение участников	Пользователь приглашает других пользователей системы принять участие в запланированном мероприятии.
Просмотр профиля пользователя	Пользователь просматривает профиль другого пользователя системы.
Просмотр своего профиля	Пользователь просматривает свой профиль.
Изменение профиля	Пользователь вносит изменения в свой профиль.

Далее необходимо рассмотреть разработанную диаграмму классов, данная диаграмма представлена на рисунке 2.2.

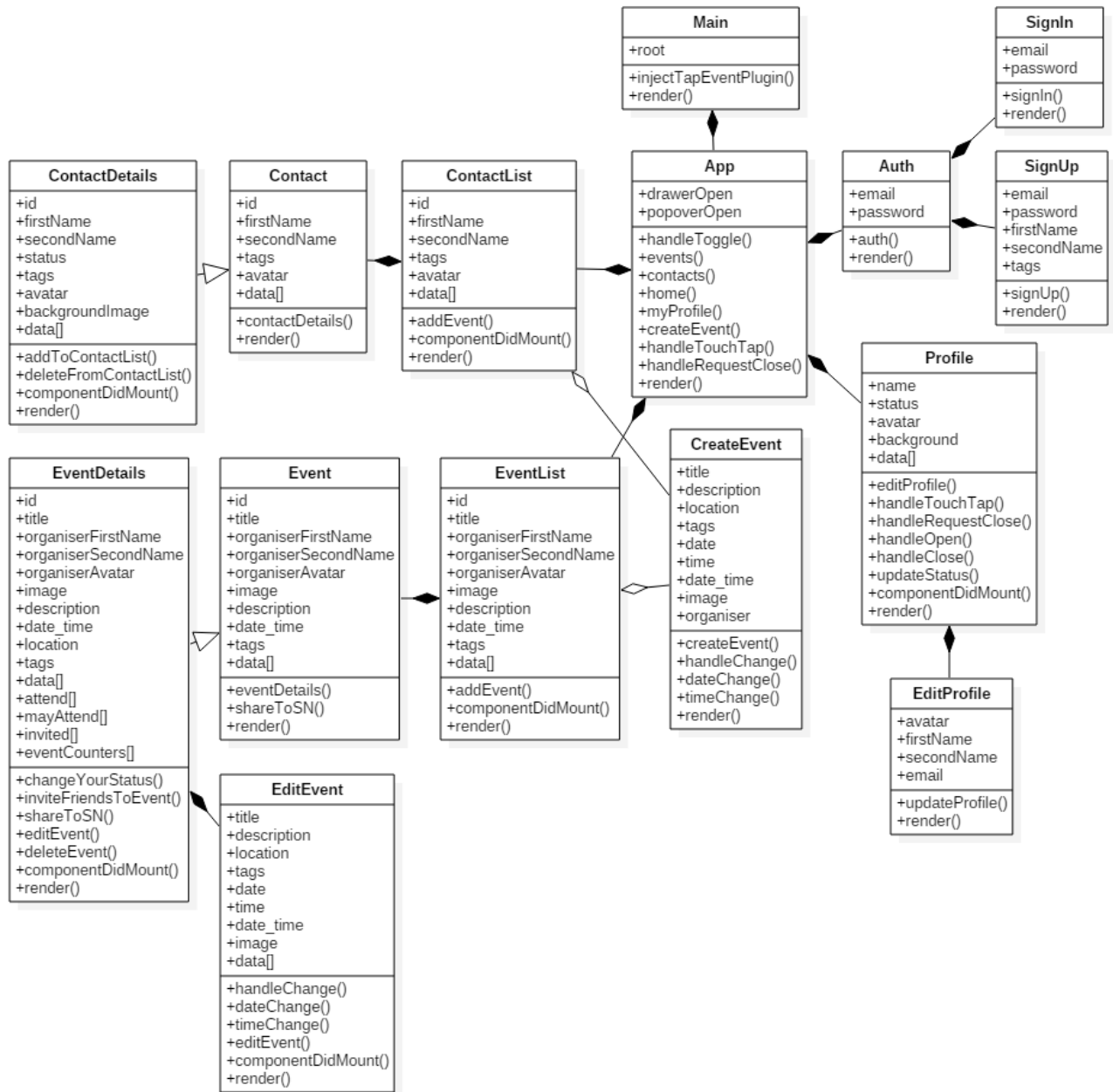


Рисунок 2.2 – Диаграмма классов

После того как была построена логическая модель информационной системы, перейдем к разработке концептуальной и логической моделей базы данных информационной системы на основании разработанных ранее диаграммы вариантов использования и диаграммы классов.

## 2.3 Разработка концептуальной и логической моделей данных автоматизированной информационной системы

Концептуальная модель данных представляет собой отражение предметной области, для которой разрабатывается база данных, описание главных сущностей, а также их отношения.

Разработанная концептуальная модель данных представлена на рисунке 2.3.

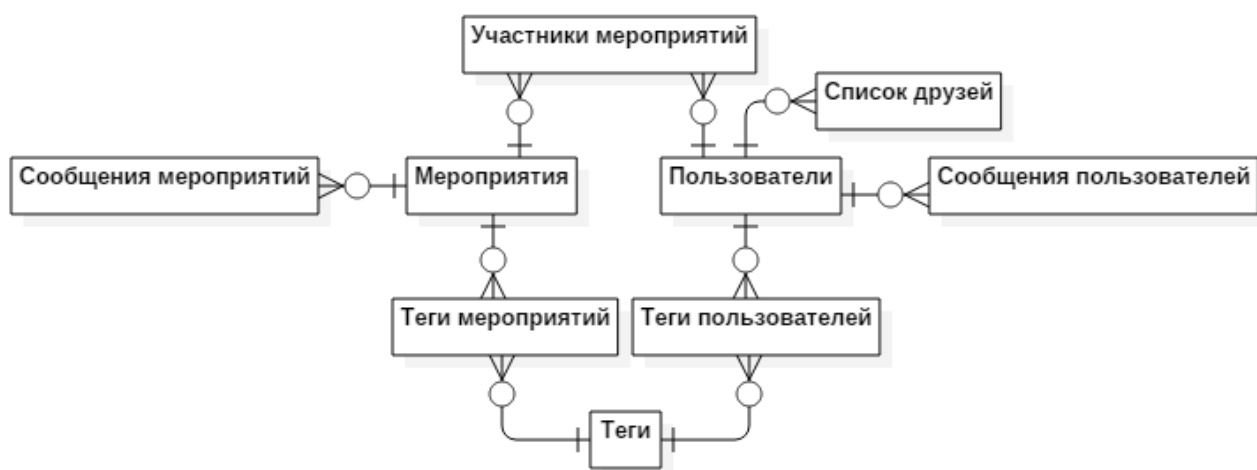


Рисунок 2.3 – Концептуальная модель данных

Концептуальная модель данных изображенная на рисунке 2.3 содержит в себе 9 сущностей: «Мероприятия», «Сообщения мероприятий», «Участники мероприятий», «Теги мероприятий», «Пользователи», «Сообщения пользователей», «Теги пользователей», «Список друзей», «Теги».

Все сущности связаны отношением «один-ко-многим».

Далее необходимо рассмотреть разработанную логическую модель данных, данная модель расширяет концептуальную модель данных.

Логическая модель данных представляет собой структуру базы данных, но без привязки к какой-либо СУБД. Данная модель расширяет концептуальную модель данных путем определения атрибутов, уточнением состава сущностей и их взаимосвязей между собой.

Разработанная логическая модель данных представлена на рисунке 2.4.

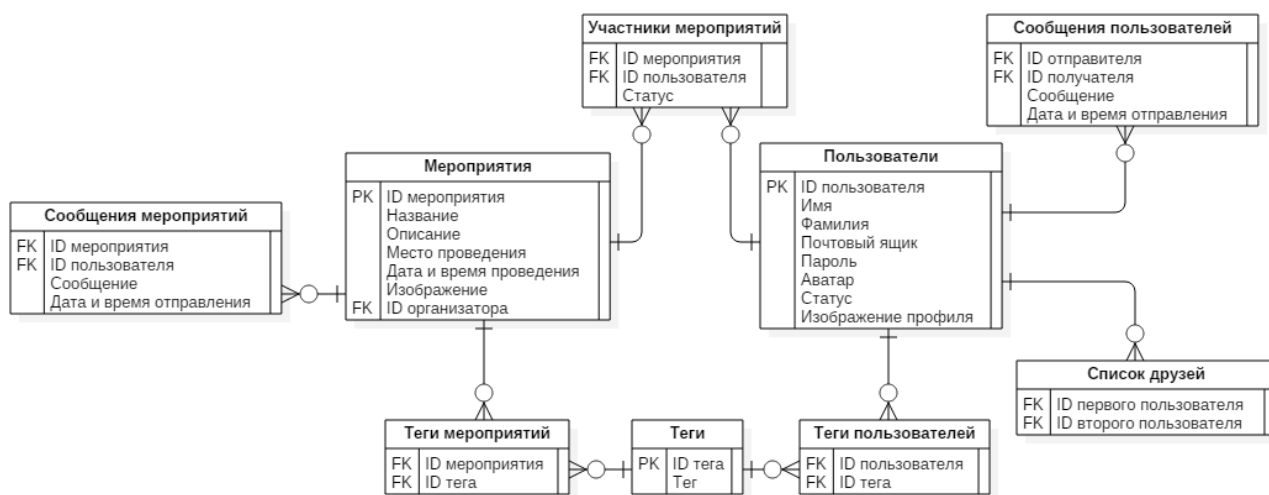


Рисунок 2.4 – Логическая модель данных

Логическая модель данных изображенная на рисунке 2.4 содержит в себе 9 сущностей:

- сущность «Мероприятия», содержит данные о мероприятии, название мероприятия, описание, место проведение, дата и времени проведения, а также, содержит информацию об организаторе мероприятия;
- сущность «Сообщения мероприятий», хранит в себе данные о сообщениях, оставленных пользователями на страницах мероприятий, в данной сущности содержится содержание сообщения который определенный пользователь оставил на определённой странице мероприятия с меткою о времени;
- сущность «Теги», содержит в себе теги, которые пользователи используют у себя в профилях или же которые организаторы указывают на их страницах мероприятий;
- сущность «Теги мероприятий», описывает какие теги к какой странице мероприятия относятся;
- сущность «Пользователи», хранит в себе данные о пользователе, такие как имя пользователя, его фамилия, почтовый ящик и пароль;
- сущность «Теги пользователей», описывает какие теги к каким профилям пользователей относятся;

- сущность «Участники мероприятий», описывает какие пользователи и с каким статусом относятся к мероприятиям;
- сущность «Сообщения пользователей», описывает какое сообщение пользователь написал другому пользователю;
- сущность «Список друзей», описывает какой пользователь находится в контакт-листе другого пользователя.

Все сущности связаны отношением «один-ко-многим».

После того как была построена концептуальная и логическая модель данных информационной системы, перейдем к формированию и описанию требований к аппаратно-программному обеспечению информационной системы.

## **2.4 Требования к аппаратно-программному обеспечению автоматизированной информационной системы**

Техническое обеспечение системы должно максимально и наиболее эффективно использовать существующие на предприятии технические средства.

В состав технического комплекса должны входить следующие технические средства:

- сервер СУБД и АИС;
- ПК пользователей;
- мобильные устройства пользователей.

Минимальные требования к серверу СУБД и АИС:

- процессор – частота не менее 2.4 ГГц;
- оперативная память – не менее 4 Гб;
- свободное место на жестком диске - не менее 25 Гб;
- подключение к сети Интернет;
- операционная система - Windows.

Минимальные требования к ПК пользователей:

- процессор – частота не менее 2 ГГц;
- оперативная память – не менее 2 Гб;
- подключение к сети Интернет;
- ПК под управлением любой популярной ОС (Windows, MacOS, Linux и др.);
- любой из популярных браузеров (Google Chrome, Mozilla Firefox, Opera и др.).

Минимальные требования к мобильным устройствам пользователей:

- мобильное устройство под управлением любой популярной мобильной ОС (Android, iOS и др.);
- любой из популярных мобильных браузеров (Google Chrome, Mozilla Firefox и др.);
- оперативная память – не менее 2 Гб;
- подключение к сети Интернет.

После того как были сформулированы и описаны требования к аппаратно-программному обеспечению информационной системы, перейдем к подведению итогов второй главы.

## **Выводы по главе 2**

Вторая глава представляет собой описание логического проектирования информационной системы. Был сделан выбор технологии логического моделирования, после чего была построена логическая модель информационной системы, включающая в себя диаграмму вариантов использования и диаграмму классов, а также, было выполнено построение концептуальной и логической модели базы данных. После этого были сформулированы и описаны все требования к аппаратно-программному обеспечению информационной системы.

# ГЛАВА 3 ФИЗИЧЕСКОЕ ПРОЕКТИРОВАНИЕ АВТОМАТИЗИРОВАННОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ

## 3.1 Выбор архитектуры автоматизированной информационной системы

Архитектура информационной системы представляет собой распределение функций по ее подсистемам и компонентам, определяет, как границы подсистем, так и их взаимодействие между собой. Основой архитектуры информационной системы является трехзвенная клиент-серверная архитектура. Данная архитектура изображена на рисунке 3.1.

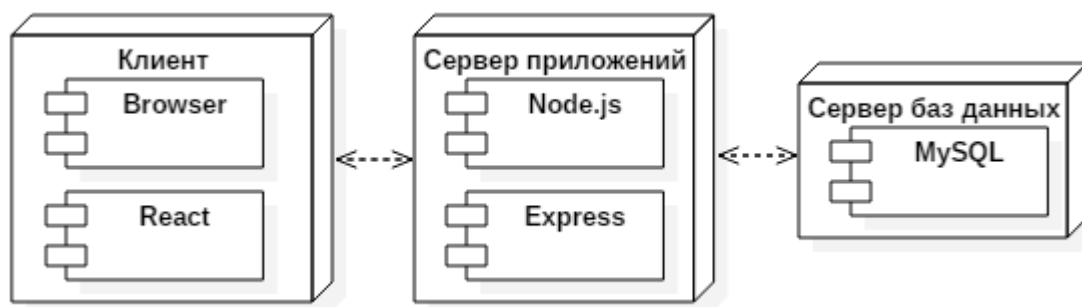


Рисунок 3.1 – Трехзвенная клиент-серверная архитектура

Архитектура информационной системы, изображенная на рисунке 3.1 содержит в себе три звена: клиент – сервер приложений – сервер баз данных:

- первое звено – клиент (слой клиента), представляет собой любой популярный веб-браузер (Google Chrome, Mozilla Firefox, Internet Explorer и др.) с использованием библиотеки React от компании Facebook для создания UI компонент (компонент пользовательского интерфейса);
- второе звено – сервер приложений (средний связующий слой, слой логики), представляет собой веб-сервер на платформе Node.js с использованием веб-фреймворка Express для создания API (программного интерфейса приложения);
- третье звено – сервер баз данных (слой данных), представляет собой сервер баз данных под управлением СУБД MySQL.



Клиент взаимодействует с сервером приложений при помощи API.

Сервер приложений взаимодействует с сервером баз данных при помощи SQL-запросов.

Данная архитектура обладает хорошей масштабируемостью, гибкостью, конфигурируемостью, а также обладает высокой надежностью и безопасностью. Все функции распределены между сервером приложений и сервером баз данных, что гарантирует высокую скорость работы. Благодаря этому данная архитектура имеет низкие требования к скорости сети между клиентом и сервером приложений, низкие требования к техническим характеристикам и производительности клиента. Именно поэтому трехзвенная клиент-серверная архитектура лучше всего подойдет для информационной системы.

После того как была выбрана архитектура информационной системы, перейдем к выбору технологии разработки программного обеспечения информационной системы.

### **3.2 Выбор технологии разработки программного обеспечения автоматизированной информационной системы**

Для выбора технологии разработки программного обеспечения информационной системы проведем сравнительный анализ таких инструментов разработки как: интегрированные среды разработки, WebStorm и Visual Studio, и текстовые редакторы, Atom и Notepad++. По итогам сравнительного анализа сделаем выбор наиболее подходящего инструмента для разработки программного обеспечения информационной системы.

Интегрированная среда разработки (IDE) – представляет собой комплекс инструментов для разработки программного обеспечения.

WebStorm – это интегрированная среда разработки на HTML+CSS+JavaScript от компании JetBrains, данная IDE разработана на базе платформы IntelliJ IDEA [53].

Visual Studio – это интегрированная среда разработки от компании Microsoft, позволяет разрабатывать как консольные, так и приложения с графическим интерфейсом, также данная IDE позволяет разрабатывать веб-приложения для всех платформ поддерживаемых Windows [54].

Atom – это современный текстовый редактор, функционал которого расширяется при помощи множества плагинов написанных на Node.js и встраиваемых под управлением Git Control, является кроссплатформенным решением [56].

Notepad++ – это текстовый редактор для Windows, базовый функционал которого может быть расширен при помощи подключения сторонних плагинов [57].

Результат сравнительного анализа технологий разработки программного обеспечения приведен в таблице 3.1.

Таблица 3.1 – Сравнительный анализ технологий разработки

Критерии оценки	WebStorm	Visual Studio	Atom	Notepad++
Кроссплатформенность	+	-	+	-
Бесплатность использования	-	-	+	+
Быстродействие	-	-	+	+
Подсветка синтаксиса	+	+	+	+
Автодополнение слов	+	+	+	-
Удобная навигация по коду	+	+	+	-
Подсветка синтаксических ошибок	+	+	+	-
Итого:	5/7	2/7	7/7	3/7

Исходя из анализа таблицы 3.1, делаем вывод что текстовый редактор Atom является наилучшим инструментом для разработки программного обеспечения информационной системы. Данный инструмент получил наибольшее количество баллов и отвечает всем критериям оценки.

После того как была выбрана технология разработки программного обеспечения информационной системы, перейдем к выбору СУБД для информационной системы.

### **3.3 Выбор системы управления базами данных автоматизированной информационной системы**

Для выбора СУБД необходимо рассмотреть самые популярные СУБД и провести их сравнительный анализ. Для анализа были выбраны следующие СУБД: MySQL, PostgreSQL, Microsoft SQL Server, Oracle Database и Firebird. По итогам сравнительного анализа сделаем выбор наиболее подходящей СУБД.

MySQL – на данный момент это одна из самых распространённых и популярных СУБД. Данная СУБД обладает широким функционалом и высокой гибкостью. В Интернете можно найти огромное количество документации по работе с данной СУБД [61].

PostgreSQL – это одна из наиболее развитых СУБД которая является реальной альтернативой коммерческим СУБД. Данная СУБД ориентирована на полное соответствие стандартам SQL [62].

Microsoft SQL Server – это СУБД разработанная компанией Microsoft. Основным используемым языком запросов является Transact-SQL. Transact-SQL является процедурным расширением языка SQL, созданное Microsoft [65].

Oracle Database – это СУБД разработанная компанией Oracle, которая поддерживает некоторые технологии реализующие объектно-ориентированный подход к организации баз данных [66].

Firebird – независимый проект некоторых разработчиков СУБД [68].

В ходе обсуждения выбора СУБД вместе с Заказчиком были сформулированы и определены основные критерии выбора СУБД:

- кроссплатформенность;
- бесплатность использования;
- быстроедействие;
- удобство и простота использования;
- наличие достаточного объема документаций по работе с СУБД;
- распространённость;
- опыт работы с данной СУБД.

Результат сравнительного анализа СУБД приведен в таблице 3.2.

Таблица 3.2 – сравнительный анализ СУБД

Критерии оценки	MySQL	PostgreSQL	Microsoft SQL Server	Oracle Database	Firebird
Кроссплатформенность	+	+	-	+	+
Бесплатность использования	+	+	-	-	+
Быстродействие	+	+	+	+	-
Удобство и простота использования	+	+	+	-	+
Наличие достаточного объема документаций по работе с СУБД	+	+	-	-	+
Распространённость	+	+	-	+	-
Опыт работы с данной СУБД	+	-	+	-	-
Сумма баллов:	7	6	3	3	4

Исходя из анализа таблицы 3.2, делаем вывод что СУБД MySQL является наилучшим выбором, поэтому выбираем ее в качестве СУБД. Данная СУБД получила наибольшее количество баллов и отвечает всем критериям оценки.

После того как была выбрана СУБД для информационной системы, перейдем к разработке физической модели данных информационной системы.

### **3.4 Разработка физической модели данных автоматизированной информационной системы**

В отличии от логической модели данных, физическая модель данных отображает структуру базы данных на примере конкретной СУБД, в пункте 3.3 был сделан выбор СУБД для информационной системы. Разработка физической модели данных будет осуществлено при помощи CASE-средства StarUML.

Для создания базы данных информационной системы будет использован инструмент MySQL Workbench.

MySQL Workbench – представляет собой инструмент для визуального проектирования баз данных, который включает в себя проектирование, моделирование, создание и эксплуатацию базы данных для СУБД MySQL [69].

Определившись с инструментами для разработки физической модели данных и создания базы данных, перейдем к разработке физической модели данных.

Разработанная физическая модель данных представлена на рисунке 2.3.

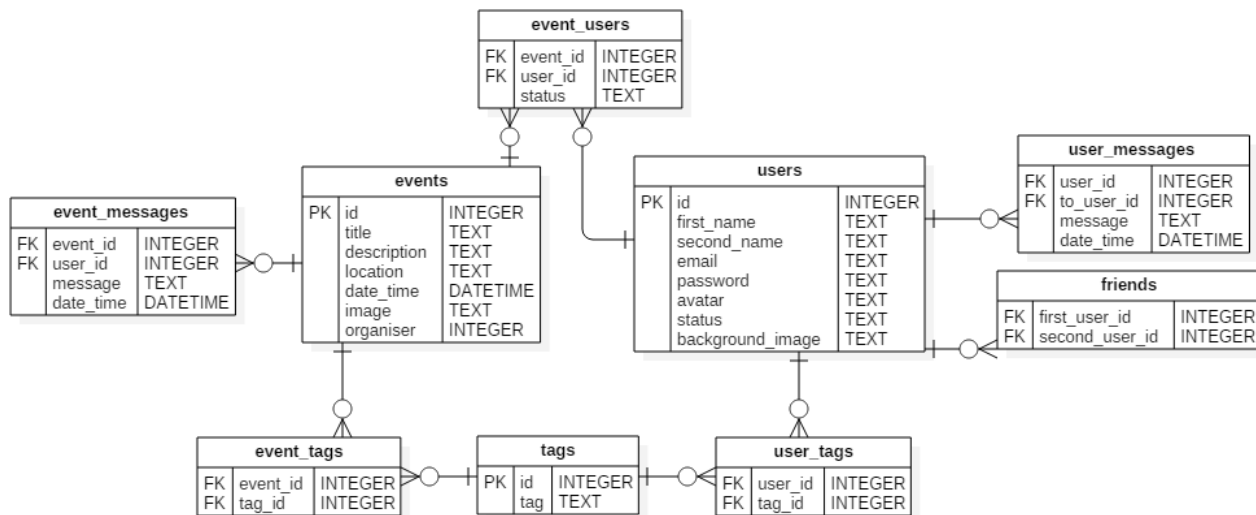


Рисунок 3.2 – Физическая модель данных

Физическая модель данных изображенная на рисунке 3.2 содержит в себе 9 сущностей: «events», «event\_messages», «event\_users», «event\_tags», «users», «user\_messages», «user\_tags», «friends», «tags».

Все сущности связаны отношением «один-ко-многим».

После того как была построена физическая модель данных, перейдем к описанию функциональности информационной системы.

### 3.5 Описание функциональности автоматизированной информационной системы и ее тестирование

В данном пункте будут описаны ключевые функции разработанной информационной системы. Работать с данной системой будут организаторы мероприятий, а также участники. Данные роли были рассмотрены и описаны во второй главе, пункта 2.2.

После прохождения авторизации, пользователь попадает на главную страницу, со списком мероприятий доступных для посещения. Данная страница изображена на рисунке 3.3.

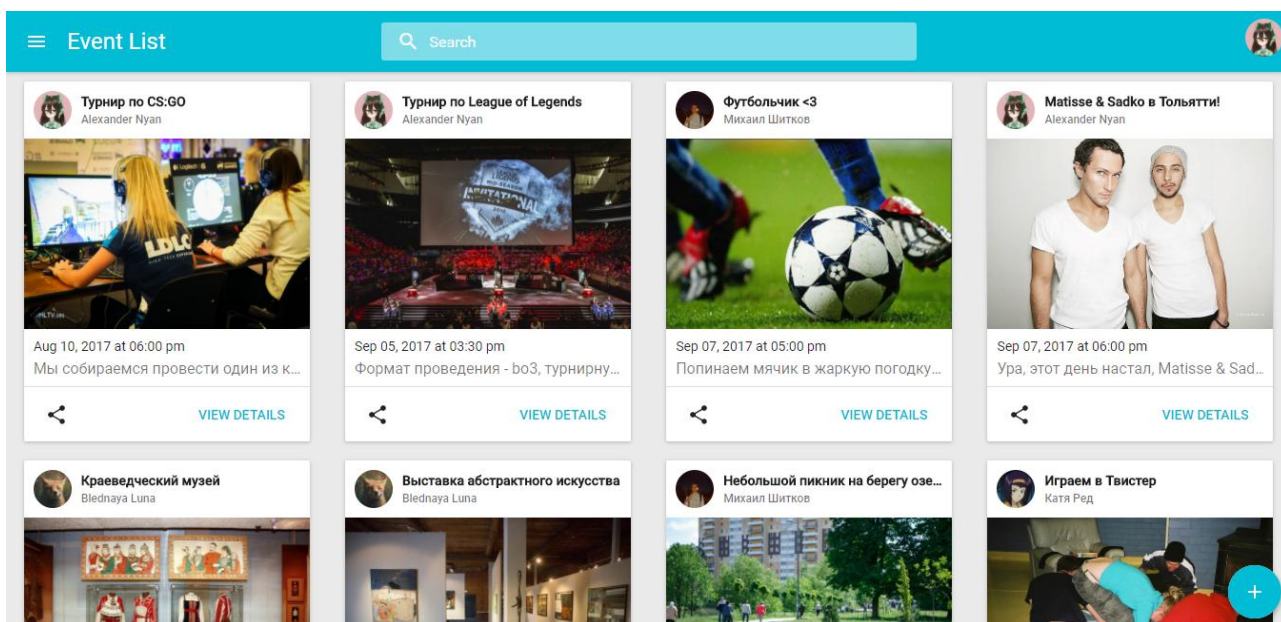


Рисунок 3.3 – Главная страница

При нажатии кнопки «VIEW DETAILS», пользователь попадает на страницу мероприятия на которой содержится вся информация. Данная страница представлена на рисунке 3.4. Фрагмент кода представлен в приложении В.

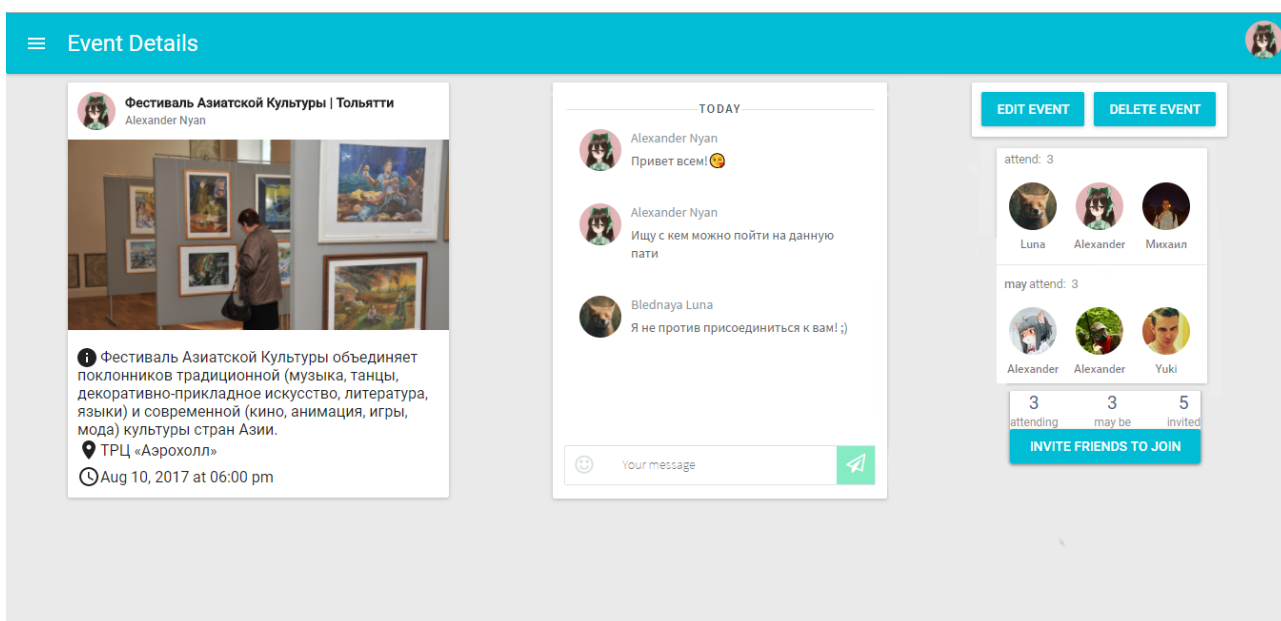


Рисунок 3.4 – Страница запланированного мероприятия

На странице, изображенной на рисунке 3.4 пользователь может ознакомиться с подробной информацией выбранного мероприятия.

Также на данной странице пользователь может принять участие в обсуждении деталей организации и проведения мероприятия или пообщаться с другими участниками. Пользователь может пригласить принять участие в данном мероприятии других пользователей из его контакт-листа если нажмет на кнопку «INVITE FRIENDS TO JOIN». Если зашедший на эту страницу пользователь является организатором данного мероприятия, то ее функционал расширяется, организатор может изменить информацию о мероприятии путем нажатия кнопки «EDIT EVENT» или удалить страницу данного мероприятия путем нажатия кнопки «DELETE EVENT». Также пользователь может ознакомиться со списками пользователей, которые собираются принять участие, пользователей, которые еще не уверены в своем решении посетить данное мероприятие и пользователей, которые были приглашены принять участие в данном мероприятии, но еще не дали свой ответ на приглашение.

Далее рассмотрим страницу изменения информации о мероприятии его организатором. Страница изменения мероприятия приведена на рисунке 3.5.

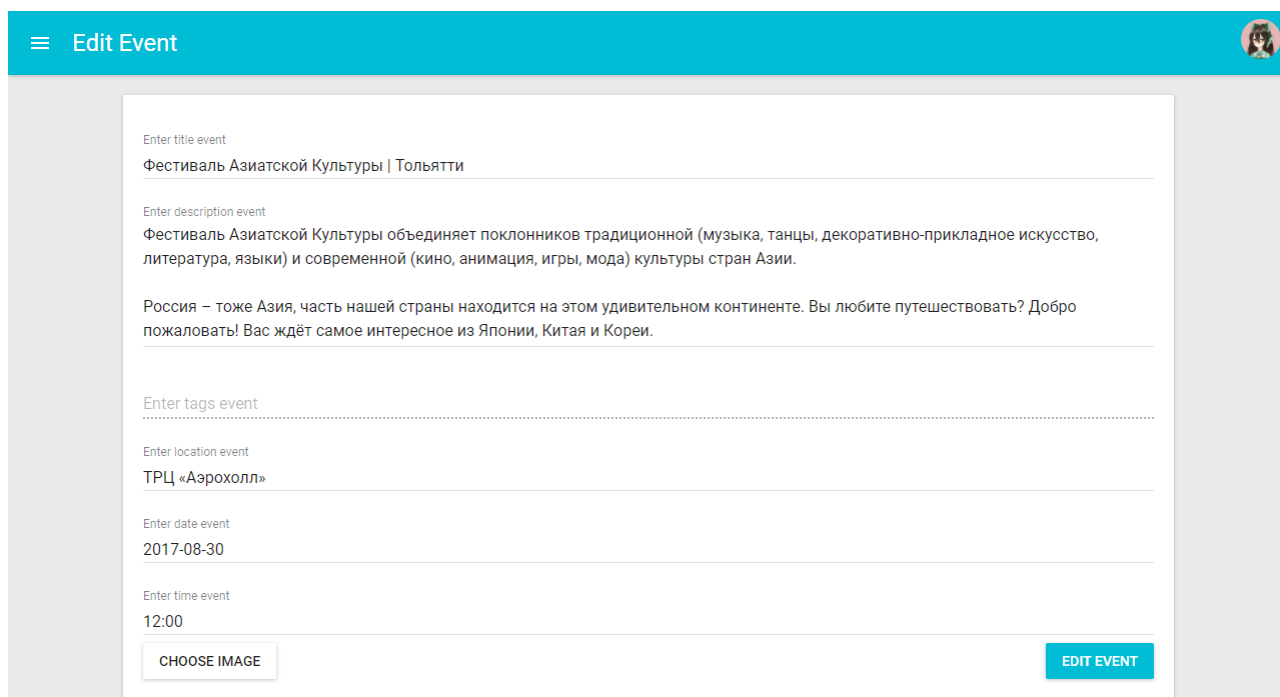


Рисунок 3.5 – Страница изменения мероприятия

На странице, изображенной на рисунке 3.5, организатор мероприятия вносит необходимые изменения после чего сохраняет внесенные изменения путем нажатия кнопки «EDIT EVENT».

Вернемся к рассмотрению главной страницы. Если пользователь нажмет на кнопку с знаком «+», то он попадает на страницу создания мероприятия. Данная страница изображена на рисунке 3.6.

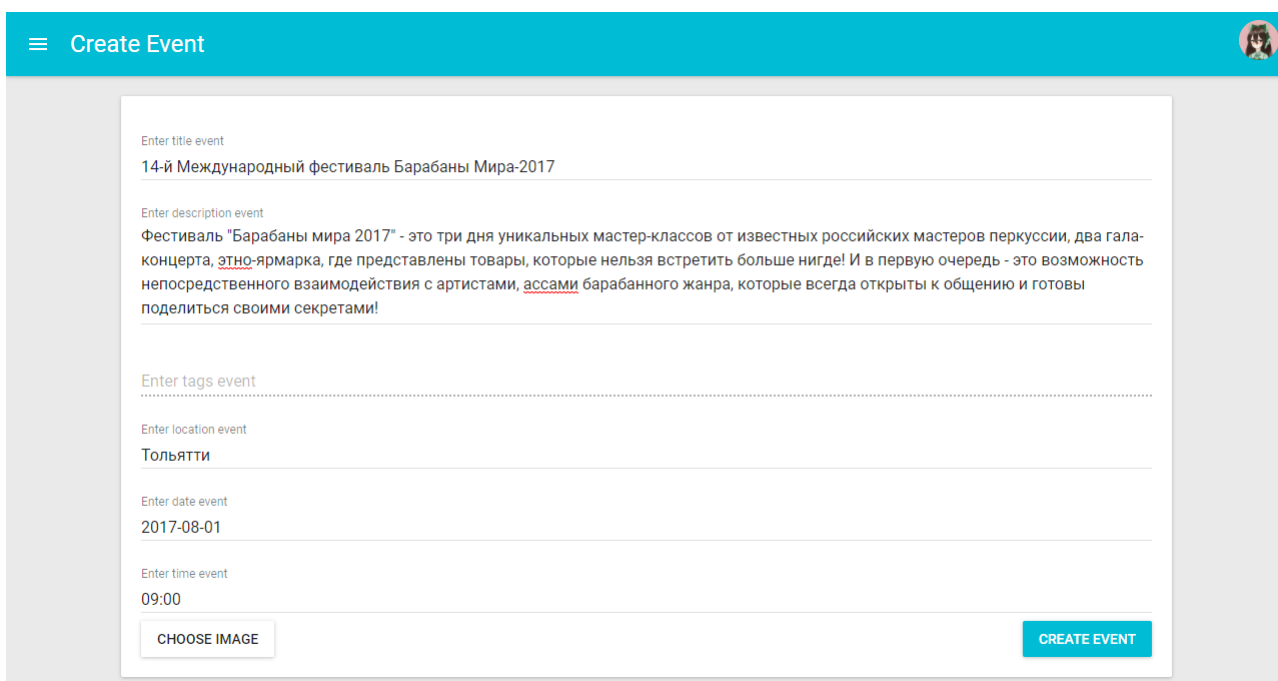


Рисунок 3.6 – Страница создания мероприятия

На странице, изображенной на рисунке 3.6 организатор вносит данные о планируемом мероприятии, нажимает кнопку «CREATE EVENT», после чего создается страница мероприятия. Если пользователь нажмет на иконку профиля в верхнем правом углу, появится меню управления аккаунтом.

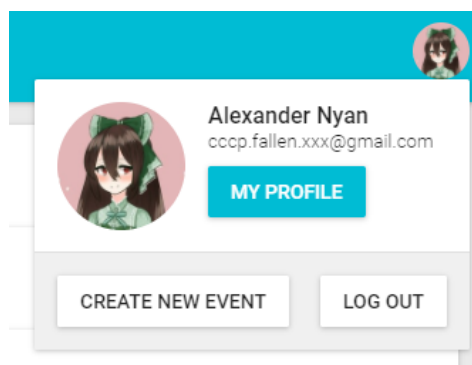


Рисунок 3.7 – Страница с контакт-листом пользователя



На странице, изображенной на рисунке 3.7 пользователь может перейти в свой профиль путем нажатия кнопки «MY PROFILE», создать страницу мероприятия если нажмет кнопку «CREATE NEW EVENT» и выйти из своего аккаунта нажатием кнопки «LOG OUT».

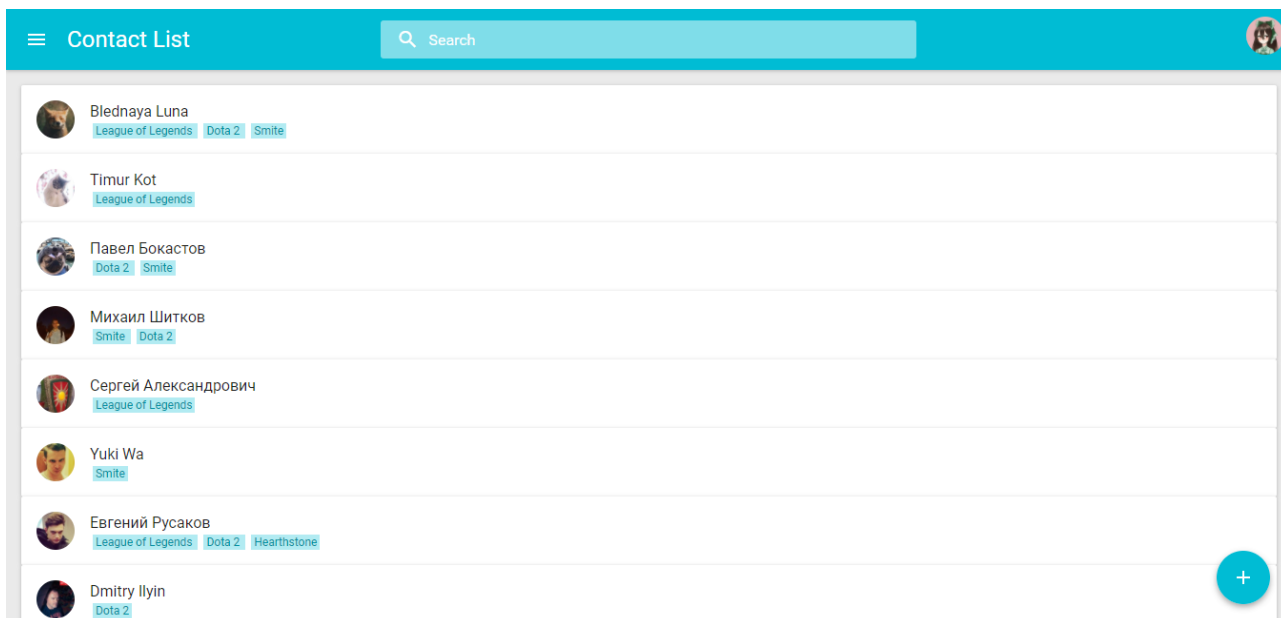


Рисунок 3.8 – Страница с контакт-листом пользователя

Контакт-лист пользователя представляет собой страницу со списком сохранённых контактов других пользователей. Данная страница представлена на рисунке 3.8. Фрагмент кода представлен в приложениях Г и Д.

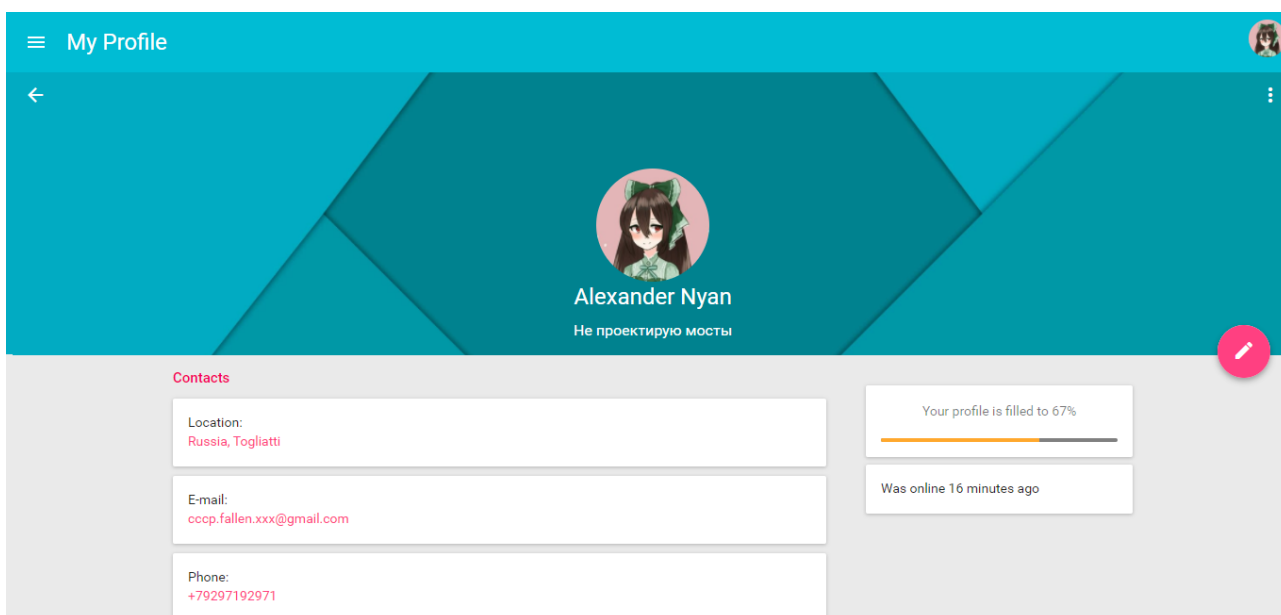


Рисунок 3.9 – Страница профиля пользователя

На странице, изображенной на рисунке 3.9, пользователь может просмотреть подробную информацию о другом пользователе так же выглядит и профиль пользователя, данная страница содержит его контактные данные и список посещенных им мероприятий.

Владелец аккаунта может изменить информацию своего профиля путем нажатия кнопки с карандашом, после нажатия данной кнопки пользователь попадает на страницу редактирования профиля, данная страница представлена на рисунке 3.10.

The image shows a web interface for editing a user profile. At the top, there is a teal navigation bar with a hamburger menu icon, the text 'Edit Profile', and a small circular profile picture of a user. Below this bar, on the right side, is a red button labeled 'SAVE'. The main content area is a white form with a circular profile picture of a character with dark hair and a green bow. The form contains several input fields: 'First name' with the value 'Alexander', 'Second name' with 'Nyan', 'Middle name (only one word)' which is empty, 'Summary' with the text 'Не проектирую мосты', 'City' with a dropdown menu showing 'Togliatti', 'Activity' with 'Student', and 'Website' with 'http://nyan-nyan.ru'.

Рисунок 3.10 – Страница редактирования профиля пользователя

После того, как был рассмотрен основной функционал разработанной информационной системы перейдем к ее тестированию.

Тестирование информационной системы подразумевает проверку соответствия между реальным и ожидаемым поведением информационной системы на наборе тестов.

Для разработанной информационной системы был выбран метод тестирования такой как тестирование «черного ящика».

Тестирование «черного ящика» – это метод тестирования, который базируется только на тестировании функционального поведения с точки зрения внешнего мира, без доступа к внутренней структуре информационной системы

и без доступа к базе данных. Стратегия поведенческого теста исходит из технических требований и их спецификаций.

Таблица с описанием тест-кейсов представлена в приложении Б.

В результате тестирования разработанной информационной системы методом «черного ящика» ошибок в функционировании выявлено не было.

После того, как было произведено тестирование разработанной информационной системы перейдем к подведению итогов третьей главы.

### **Выводы по главе 3**

Третья глава представляет собой описание физического проектирования информационной системы. В главе был сделан выбор архитектуры информационной системы, выбрана технология разработки программного обеспечения, сделан выбор СУБД и осуществлена разработка физической модели данных под выбранную СУБД, рассмотрен основной функционал разработанной информационной системы. На заключительном этапе было проведено тестирование разработанной информационной системы.

## ЗАКЛЮЧЕНИЕ

Тема данной бакалаврской работы поднимает актуальную проблему разработки информационной системы управления мероприятиями для ООО «ЭПАМ Систэмз».

Результатом работы является информационная система, представляющая собой удобный инструмент для управления мероприятиями, позволяющий сократить время и ресурсы затрачиваемые на процессы управления мероприятиями, обеспечивающий качественное и надежное выполнение этих процессов.

В ходе работы была достигнута поставленная цель и выполнены основные задачи:

- проведен анализ предметной области;
- проведен анализ проблем в существующем бизнес-процессе;
- проведен анализ существующих решений на рынке;
- проведено логическое проектирование информационной системы;
- с помощью технологий Node.js, Express, React и MySQL реализована информационная система для управления мероприятиями;
- проведено тестирование разработанной информационной системы.

Информационная система находится на стадии апробации, в дальнейшем предполагается развитие проекта и усовершенствование разработанной информационной системы, расширение ее функционала, добавление новых локализаций и портирование веб-приложения на мобильные платформы.

Автором работы был приобретен ценный опыт проектирования и разработки информационных систем который будет использован в дальнейшей профессиональной деятельности.

## СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

### *Нормативно-правовые акты*

1. ГОСТ 7.32-2001. Отчет о научно-исследовательской работе. Структура и правила оформления.
2. ГОСТ 7.1-2003. Библиографическая запись. Библиографическое описание документа.
3. ГОСТ 7.82-2001. Библиографическая запись. Библиографическое описание электронных ресурсов.
4. ГОСТ 19.701 – 90. Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения (ИСО 5807–85) [Текст]. Введен 70 1992–01–01. – М.: Изд-во стандартов, 1992. – 14 с. – (Единая система программной документации).
5. ГОСТ 2.105 – 95. Общие требования к текстовым документам [Текст]. – М.: Изд-во стандартов, 1996. – 29 с. – (Единая система конструкторской документации).

### *Научная и методическая литература*

6. Балдин К. В. Информационные системы в экономике : учебник / К. В. Балдин, В. Б. Уткин. - 7-е изд. - Москва : Дашков и К°, 2012. - 395 с. – ISBN 978-5-394-01449-9.
7. Буренин С. Н. Web-программирование и базы данных : учеб. практикум / С. Н. Буренин. - Москва : Моск. гуманит. ун-т, 2014. - 120 с. - ISBN 978-5-906768-17-9.
8. Вдовин В. М. Предметно-ориентированные экономические информационные системы : учебное пособие / В. М. Вдовин, Л. Е. Суркова, А. А. Шурупов. - 3-е изд. - Москва : Дашков и К°, 2013. - 388 с. : ил. - ISBN 978-5-394-02262-3.
9. Золотов С. Ю. Проектирование информационных систем : учеб. пособие / С. Ю. Золотов ; Томский гос. ун-т систем управления и радиоэлектроники. - Томск : Эль Учебное пособие Контент, 2013. - 86 с. - ISBN 978-5-4332-0083-8.

10. Карпова И. П. Базы данных : курс лекций и материалы для практ. занятий : учеб. пособие для студентов техн. фак. / И. П. Карпова. - Санкт-Петербург : Питер, 2013. - 240 с. : ил. - (Учебное пособие). - Библиогр.: с. 233-234. - Прил.: с. 211-232. - Алф. указ.: с. 235-240. - ISBN 978-5-496-00546-3 : 418-60.

11. Реинжиниринг бизнес-процессов : учеб. пособие / А. О. Блинов [и др.] ; под ред. А. О. Блинова. - Москва : ЮНИТИ- ДАНА, 2012. - 341 с. - ISBN 978-5-238-01823-2.

12. Шелухин О. И. Моделирование информационных систем : учеб. пособие. 004 / О. И. Шелухин. - 2-е изд., перераб. и доп. - Москва : Горячая линия - Телеком, 2012. - 516 с. : ил. - ISBN 978-5- 9912-0193-3.

13. Грекул В. И. Проектирование информационных систем : учеб. пособие / В. И. Грекул, Г. Н. Денищенко, Н. Л. Коровкина. - 2-е изд., испр. - Москва : Интернет-Ун-т Информ. Технологий : БИНОМ. Лаб. знаний, 2008. - 299 с.

14. Рудинский И. Д. Технология проектирования автоматизированных систем обработки информации и управления : учеб. пособие / И. Д. Рудинский. - Москва : Горячая линия - Телеком, 2011. - 304 с. - ISBN 978-5-9912-0148-3.

#### *Электронные ресурсы*

15. IDEF0: функциональное моделирование деловых процессов / [Электронный ресурс]: [http://www.info-system.ru/designing/methodology/sadt/sadt\\_for\\_bp.html](http://www.info-system.ru/designing/methodology/sadt/sadt_for_bp.html), (дата обращения: 18.05.2017).

16. Функциональное моделирование бизнес-процессов с использованием ППП Design/IDEF / [Электронный ресурс]: <http://market-pages.ru/biznesproc1/6.html>, (дата обращения: 21.05.2017).

17. Концептуальная модель / [Электронный ресурс]: [https://ru.wikipedia.org/wiki/Концептуальная\\_модель](https://ru.wikipedia.org/wiki/Концептуальная_модель), (дата обращения: 21.05.2017).

18. Лекция 6: Методологии моделирования предметной области / [Электронный ресурс]: <http://www.intuit.ru/studies/courses/2195/55/lecture/1628>, (дата обращения: 22.05.2017).
19. Особенности концептуального моделирования предметной области / [Электронный ресурс]: <https://habrahabr.ru/post/256823/>, (дата обращения: 11.05.2017).
20. UML / [Электронный ресурс]: <https://ru.wikipedia.org/wiki/UML>, (дата обращения: 17.05.2017).
21. IDEF0 / [Электронный ресурс]: <https://ru.wikipedia.org/wiki/IDEF0>, (дата обращения: 10.05.2017).
22. Основные методологии обследования организаций. Стандарт IDEF0. / [Электронный ресурс]: <http://www.cfin.ru/vernikov/idef/idef0.shtml>, (дата обращения: 13.05.2017).
23. Знакомство с нотацией IDEF0 и пример использования / [Электронный ресурс]: <https://habrahabr.ru/post/322832/>, (дата обращения: 10.05.2017).
24. ARIS / [Электронный ресурс]: <https://ru.wikipedia.org/wiki/ARIS>, (дата обращения: 12.05.2017).
25. ARIS - основы теории / [Электронный ресурс]: [http://citforum.ru/seminars/cis99/vest\\_03.shtml](http://citforum.ru/seminars/cis99/vest_03.shtml), (дата обращения: 27.05.2017).
26. Бизнес-процесс / [Электронный ресурс]: <https://ru.wikipedia.org/wiki/Бизнес-процесс>, (дата обращения: 18.05.2017).
27. AS-IS модель / [Электронный ресурс]: <http://piter-soft.ru/automation/more/glossary/process/as-is-model/>, (дата обращения: 13.05.2017).
28. FURPS / [Электронный ресурс]: <https://ru.wikipedia.org/wiki/FURPS>, (дата обращения: 12.05.2017).
29. Требования к системе: классификация FURPS+ / [Электронный ресурс]: <https://sysana.wordpress.com/2010/09/16/furps/>, (дата обращения: 16.05.2017).

30. Лекция 2: Понятие требования. Классификации требований / [Электронный ресурс]: <http://www.intuit.ru/studies/courses/2188/174/lecture/4714?page=2>, (дата обращения: 18.05.2017).

31. TO-BE модель / [Электронный ресурс]: <http://piter-soft.ru/automation/more/glossary/process/to-be-model/>, (дата обращения: 22.05.2017).

32. Проектирование информационных систем / [Электронный ресурс]: [http://www.business-process.ru/is/theory/theory\\_design\\_is.html](http://www.business-process.ru/is/theory/theory_design_is.html), (дата обращения: 23.05.2017).

33. Лекция 8: Разработка и внедрение информационной системы / [Электронный ресурс]: <http://www.intuit.ru/studies/courses/1055/271/lecture/6880?page=5>, (дата обращения: 18.05.2017).

34. Лекция 3: Логическое моделирование / [Электронный ресурс]: <http://www.intuit.ru/studies/courses/3440/682/lecture/14036>, (дата обращения: 25.05.2017).

35. Лекция 8: Обзор CASE-средств для построения диаграмм UML / [Электронный ресурс]: <http://www.intuit.ru/studies/courses/1007/229/lecture/5963?page=6>, (дата обращения: 26.05.2017).

36. Создание логической модели / [Электронный ресурс]: <http://itteach.ru/bpwin/sozdanie-logicheskoy-modeli>, (дата обращения: 15.05.2017).

37. Логическая модель предметной области / [Электронный ресурс]: <http://analyst.by/diagrams/logicheskaya-model-predmetnoy-oblasti>, (дата обращения: 17.05.2017).

38. Диаграмма прецедентов / [Электронный ресурс]: [https://ru.wikipedia.org/wiki/Диаграмма\\_прецедентов](https://ru.wikipedia.org/wiki/Диаграмма_прецедентов), (дата обращения: 23.05.2017).



39. Лекция 3: Элементы графической нотации диаграммы вариантов использования / [Электронный ресурс]: <http://www.intuit.ru/studies/courses/32/32/lecture/1004?page=2>, (дата обращения: 26.05.2017).
40. Диаграмма классов / [Электронный ресурс]: [https://ru.wikipedia.org/wiki/Диаграмма\\_классов](https://ru.wikipedia.org/wiki/Диаграмма_классов), (дата обращения: 12.05.2017).
41. Отношения классов — от UML к коду / [Электронный ресурс]: <https://habrahabr.ru/post/150041/>, (дата обращения: 17.05.2017).
42. Лекция 3: Виды диаграмм UML / [Электронный ресурс]: <http://www.intuit.ru/studies/courses/1007/229/lecture/5954?page=2>, (дата обращения: 22.05.2017).
43. Концептуальная модель / [Электронный ресурс]: [https://ru.wikipedia.org/wiki/Концептуальная\\_модель](https://ru.wikipedia.org/wiki/Концептуальная_модель), (дата обращения: 28.05.2017).
44. Типы моделей данных корпоративного хранилища данных / [Электронный ресурс]: [http://www.prj-exp.ru/dwh/dwh\\_model\\_types.php](http://www.prj-exp.ru/dwh/dwh_model_types.php), (дата обращения: 29.05.2017).
45. 1.2.2. Логическая модель / [Электронный ресурс]: [http://sernam.ru/book\\_cbd.php?id=5](http://sernam.ru/book_cbd.php?id=5), (дата обращения: 18.05.2017).
46. Архитектура системы / [Электронный ресурс]: [https://ru.wikipedia.org/wiki/Архитектура\\_системы](https://ru.wikipedia.org/wiki/Архитектура_системы), (дата обращения: 16.05.2017).
47. Лекция 4: Архитектурные особенности проектирования и разработки Веб-приложений / [Электронный ресурс]: <http://www.intuit.ru/studies/courses/956/467/lecture/28784>, (дата обращения: 15.05.2017).
48. ReactJS для глупых людей / [Электронный ресурс]: <https://habrahabr.ru/post/249107/>, (дата обращения: 27.05.2017).
49. Изучение React — для чего, откуда, как? / [Электронный ресурс]: <https://habrahabr.ru/post/269303/>, (дата обращения: 27.05.2017).

50. Node.js / [Электронный ресурс]: <https://ru.wikipedia.org/wiki/Node.js>, (дата обращения: 29.05.2017).
51. API / [Электронный ресурс]: <https://ru.wikipedia.org/wiki/API>, (дата обращения: 25.05.2017).
52. Технология разработки программного обеспечения / [Электронный ресурс]: <http://vscode.ru/articles/tehnologiya-razrabotki-po.html>, (дата обращения: 13.05.2017).
53. WebStorm / [Электронный ресурс]: <https://ru.wikipedia.org/wiki/WebStorm>, (дата обращения: 18.05.2017).
54. Microsoft Visual Studio / [Электронный ресурс]: [https://ru.wikipedia.org/wiki/Microsoft\\_Visual\\_Studio](https://ru.wikipedia.org/wiki/Microsoft_Visual_Studio), (дата обращения: 22.05.2017).
55. Atom: новый редактор кода от Гитхаба / [Электронный ресурс]: <https://habrahabr.ru/post/214079/>, (дата обращения: 12.05.2017).
56. Atom / [Электронный ресурс]: <https://ru.wikipedia.org/wiki/Atom>, (дата обращения: 15.05.2017).
57. Notepad++ / [Электронный ресурс]: <https://ru.wikipedia.org/wiki/Notepad++>, (дата обращения: 15.05.2017).
58. Редакторы для кода / [Электронный ресурс]: <https://learn.javascript.ru/editor>, (дата обращения: 16.05.2017).
59. Редактор или IDE? Очередная попытка анализа / [Электронный ресурс]: <https://habrahabr.ru/post/265197/>, (дата обращения: 27.05.2017).
60. Система управления базами данных / [Электронный ресурс]: [https://ru.wikipedia.org/wiki/Система\\_управления\\_базами\\_данных](https://ru.wikipedia.org/wiki/Система_управления_базами_данных), (дата обращения: 13.05.2017).
61. MySQL / [Электронный ресурс]: <https://ru.wikipedia.org/wiki/MySQL>, (дата обращения: 27.05.2017).
62. PostgreSQL / [Электронный ресурс]: <https://ru.wikipedia.org/wiki/PostgreSQL>, (дата обращения: 29.05.2017).

63. Что такое PostgreSQL? / [Электронный ресурс]: [http://www.sai.msu.su/~megera/postgres/talks/what\\_is\\_postgresql.html](http://www.sai.msu.su/~megera/postgres/talks/what_is_postgresql.html), (дата обращения: 17.05.2017).

64. Чем PostgreSQL лучше других SQL баз данных с открытым исходным кодом. Часть 1 / [Электронный ресурс]: <https://habrahabr.ru/post/282764/>, (дата обращения: 15.05.2017).

65. Microsoft SQL Server / [Электронный ресурс]: [https://ru.wikipedia.org/wiki/Microsoft\\_SQL\\_Server](https://ru.wikipedia.org/wiki/Microsoft_SQL_Server), (дата обращения: 16.05.2017).

66. Oracle Database / [Электронный ресурс]: [https://ru.wikipedia.org/wiki/Oracle\\_Database](https://ru.wikipedia.org/wiki/Oracle_Database), (дата обращения: 18.05.2017).

67. Основные характеристики СУБД Oracle / [Электронный ресурс]: <http://bourabai.ru/dbt/servers/oracle2.htm>, (дата обращения: 22.05.2017).

68. Firebird / [Электронный ресурс]: <https://ru.wikipedia.org/wiki/Firebird>, (дата обращения: 25.05.2017).

69. MySQL Workbench / [Электронный ресурс]: [https://ru.wikipedia.org/wiki/MySQL\\_Workbench](https://ru.wikipedia.org/wiki/MySQL_Workbench), (дата обращения: 12.05.2017).

70. Тестирование информационных систем / [Электронный ресурс]: <http://aplana.ru/services/testing/testirovanie-sistem>, (дата обращения: 12.05.2017).

71. Тестирование по стратегии чёрного ящика / [Электронный ресурс]: [https://ru.wikipedia.org/wiki/Тестирование\\_по\\_стратегии\\_чёрного\\_ящика](https://ru.wikipedia.org/wiki/Тестирование_по_стратегии_чёрного_ящика), (дата обращения: 15.05.2017).

#### *Литература на иностранном языке*

72. Getting Started with MongoDB, Node.js and Restify / [Электронный ресурс]: <https://www.mongodb.com/blog/post/getting-started-with-mongodb-nodejs-and-restify>, (дата обращения: 27.05.2017).

73. Learning from the top 10 mistakes developers make / [Электронный ресурс]: <https://dev.to/charanrajgolla/learning-from-top-10-mistakes-done-by-developers>, (дата обращения: 13.05.2017).

74. Node.js Post-Mortem Diagnostics & Debugging / [Электронный ресурс]: <https://blog.risingstack.com/post-mortem-diagnostics-debugging-node-js-at-scale/>, (дата обращения: 15.05.2017).
75. A Comprehensive Guide To HTTP/2 Server Push / [Электронный ресурс]: <https://www.smashingmagazine.com/2017/04/guide-http2-server-push/>, (дата обращения: 17.05.2017).
76. 10 Best Practices for Writing Node.js REST APIs / [Электронный ресурс]: <https://blog.risingstack.com/10-best-practices-for-writing-node-js-rest-apis/>, (дата обращения: 16.05.2017).
77. A Guide to npm: The Node.js Package Manager / [Электронный ресурс]: <https://www.toptal.com/javascript/a-guide-to-npm-the-node-package-manager>, (дата обращения: 22.05.2017).
78. 10 Tips to Become a Better Node Developer in 2017 / [Электронный ресурс]: <https://www.sitepoint.com/10-tips-to-become-a-better-node-developer/>, (дата обращения: 13.05.2017).
79. WHY LOOPBACK IS BETTER THAN EXPRESS.JS? / [Электронный ресурс]: <https://da-14.com/blog/why-loopback-better-expressjs>, (дата обращения: 14.05.2017).

## ПРИЛОЖЕНИЕ А

### Требования к информационной системе

№	Требование	Статус	Полезность	Риск	Стабильность
<b>Требования к функциональности</b>					
1	Регистрация пользователей	Одобренные	Критичное	Средний	Низкая
2	Авторизация пользователей	Одобренные	Критичное	Средний	Низкая
3	Просмотр списка мероприятий для посещения	Одобренные	Критичное	Средний	Низкая
4	Поиск мероприятий по определенным параметрам	Одобренные	Критичное	Средний	Низкая
5	Планирование мероприятия	Предложенные	Критичное	Средний	Низкая
6	Создание мероприятия	Одобренные	Критичное	Средний	Низкая
7	Редактирования созданного мероприятия	Одобренные	Критичное	Средний	Низкая
8	Приглашение пользователей поучаствовать в мероприятии	Одобренные	Критичное	Средний	Низкая
9	Возможность поделиться информацией о мероприятии в социальных сетях	Предложенные	Критичное	Средний	Низкая
10	Возможность принимать заявки на участие в мероприятии	Предложенные	Критичное	Средний	Низкая
11	Добавление пользователей себе в контакт-лист	Одобренные	Критичное	Средний	Низкая
12	Сообщать участникам о одобрении заявки	Предложенные	Критичное	Средний	Низкая
<b>Требования к удобству использования</b>					
13	Интуитивно-понятный интерфейс	Одобренные	Критичное	Низкий	Низкая
14	Адаптивный дизайн (ПК и мобильные версии)	Одобренные	Критичное	Низкий	Низкая
15	Дизайн интерфейса в едином стиле и цветовой гамме	Одобренные	Критичное	Низкий	Низкая

Продолжение таблицы «требования к информационной системе»

№	Требование	Статус	Полезность	Риск	Стабильность
16	Защита от человеческого фактора (проверка корректности введенных данных, подтверждение действий пользователя)	Предложенные	Критичное	Низкий	Низкая
17	Наличие справочной информации по работе с системой	Предложенные	Критичное	Низкий	Низкая
<b>Требования к надежности</b>					
18	Время доступности системы – 24 часа в сутки 7 дней в неделю	Одобрены	Критичное	Низкий	Низкая
19	Частота сбоев – не более чем раз в неделю	Одобрены	Критичное	Низкий	Средняя
20	Время восстановления системы после сбоя – не более 1 суток	Одобрены	Критичное	Низкий	Низкая
21	Возможность резервного копирования данных	Предложенные	Критичное	Средний	Средняя
<b>Требования к производительности</b>					
22	Время отклика системы – не более чем 5 секунд	Одобрены	Критичное	Средний	Низкая
23	Система должна иметь возможность обеспечить одновременную работу как минимум 50 пользователям	Одобрены	Критичное	Средний	Средняя
<b>Требования к сопровождаемости</b>					
24	Время устранения обнаруженных в системе проблем – не более чем 24 часа	Одобрены	Критичное	Низкий	Средняя
25	Частота резервного копирования – раз в неделю	Предложенные	Критичное	Средний	Средняя
26	Локализация – добавление новых локализаций интерфейса	Предложенные	Критичное	Средний	Низкая

Продолжение таблицы «требования к информационной системе»

№	Требование	Статус	Полезность	Риск	Стабильность
27	Возможность расширения – наращивания дополнительного функционала	Предложенные	Критичное	Средний	Средняя
<b>Ограничения проектирования</b>					
28	Хранение данных необходимо реализовать с помощью реляционной БД	Одобренные	Критичное	Низкий	Низкая
<b>Ограничения реализации</b>					
29	В качестве СУБД должна быть использована MySQL	Одобренные	Критичное	Низкий	Низкая
30	Вся бизнес-логика должна быть реализована на языке JavaScript	Одобренные	Критичное	Низкий	Низкая

## ПРИЛОЖЕНИЕ Б

### Тест-кейсы

№	Название	Инструкции	Ожидаемый результат
1	Создание страницы мероприятия	Находясь на главной странице нажать на кнопку с знаком «+», ввести всю необходимую информацию в поля ввода, после чего создать страницу мероприятия путем нажатия кнопки «CREATE EVENT»	Созданная страница мероприятия
2	Изменение страницы мероприятия	Находясь на странице ранее созданного мероприятия, нажать кнопку «EDIT EVENT», внести необходимые изменения информации в поля ввода, после чего сохранить изменения путем нажатия кнопки «EDIT EVENT»	Произошло изменение информации на странице мероприятия
3	Удаление страницы мероприятия	Находясь на странице ранее созданного мероприятия, нажать кнопку «DELETE EVENT», в всплывающем диалоговом окне подтвердить решение удалить страницу мероприятия путем нажатия кнопки «YES»	Страница мероприятия была удалена
4	Просмотр страницы мероприятия	Находясь на главной странице нажать на кнопку «VIEW DETAILS» любого мероприятия	Отображена страница мероприятия с подробной информацией
5	Просмотр списка приглашенных к участию в мероприятии пользователей	Находясь на странице ранее созданного мероприятия, нажать на счетчик «invited»	Появилось окно с списком пользователей приглашенных к участию в мероприятии
6	Просмотр списка тех пользователей кто собирается посетить мероприятие	Находясь на странице ранее созданного мероприятия, нажать на счетчик «attending»	Появилось окно с списком пользователей кто собирается посетить мероприятие
7	Просмотр списка тех пользователей кто еще не уверен в решении принять участие в мероприятии	Находясь на странице ранее созданного мероприятия, нажать на счетчик «may be»	Появилось окно с списком пользователей кто еще не уверен в решении принять участие в мероприятии



Продолжение таблицы «тест-кейсы»

№	Название	Инструкции	Ожидаемый результат
8	Просмотр списка тех пользователей кто собирается посетить мероприятие	Находясь на странице ранее созданного мероприятия, нажать на область «Members»	Появилось окно с списком пользователей кто собирается посетить мероприятие
9	Просмотр списка тех пользователей кто еще не уверен в решении принять участие в мероприятии	Находясь на странице ранее созданного мероприятия, нажать на область «May attend»	Появилось окно с списком пользователей кто еще не уверен в решении принять участие в мероприятии
10	Просмотр списка друзей, которые могут посетить мероприятие	Находясь на странице ранее созданного мероприятия, нажать на область «...friends will attend»	Появилось окно с списком друзей которые могут посетить мероприятие
11	Возможность писать сообщения в чате на странице мероприятия	Находясь на странице ранее созданного мероприятия, ввести сообщение в поля вода, после чего отправить сообщение нажав кнопку «SEND»	Сообщение отправилось в чат
12	Просмотр списка контактов	Находясь на любой странице, нажать кнопку меню после чего выбрать в выпадающем меню пункт меню «Contacts»	Отображена страница с списком контактов пользователя
13	Просмотр профиля контакта	Находясь на странице списка контактов нажать на любой контакт из списка	Отображена страница с профилем выбранного пользователя
14	Добавление пользователя в свой контакт-лист	Находясь на странице другого пользователя нажать на кнопку с иконкой добавления в друзья	Пользователь добавлен в список контактов
15	Удаление пользователя из своего контакт-листа	Находясь на странице другого пользователя, который находится у вас в друзьях, нажать на кнопку с иконкой удаления из друзей	Пользователь удален из списка контактов
16	Просмотр своего профиля пользователя	Находясь на любой странице, нажать на иконку своего профиля, после чего нажать кнопку «VIEW PROFILE»	Отображена страница своего профиля пользователя
17	Выход из своего профиля пользователя	Находясь на любой странице, нажать на иконку своего профиля, после чего нажать кнопку «SIGN OUT»	Отображена страница входа/регистрации

Продолжение таблицы «тест-кейсы»

№	Название	Инструкции	Ожидаемый результат
18	Вход в свой профиль пользователя	Находясь на странице входа\регистрации, нажать на вкладку «SIGN IN», ввести учетные данные, после чего нажать кнопку «SIGN IN»	Отображена главная страница
19	Создание профиля пользователя	Находясь на странице входа\регистрации, нажать на вкладку «SIGN UP», ввести необходимые для регистрации данные, после чего нажать кнопку «SIGN UP»	Отображена страница профиля пользователя

## ПРИЛОЖЕНИЕ В

### Фрагмент кода класса «App.js»

```
app.get('/eventCounters/:id', function(req, res) {
  connection.query(`select
    (select count(*)
     from izeee.user2event
     join izeee.users on izeee.users.id=izeee.user2event.user_id
     where izeee.user2event.event_id=${req.params.id} and
izeee.user2event.status=3)
    as "attend",
    (select count(*)
     from izeee.user2event
     join izeee.users on izeee.users.id=izeee.user2event.user_id
     where izeee.user2event.event_id=${req.params.id} and
izeee.user2event.status=2)
    as "may_attend",
    (select count(*)
     from izeee.user2event join izeee.users on
izeee.users.id=izeee.user2event.user_id
     where izeee.user2event.event_id=${req.params.id} and
izeee.user2event.status=0)
    as "invited"`,
  function(err, data) {
    if (err)
      throw err;
    res.send(data[0]);});});
```

## ПРИЛОЖЕНИЕ Г

### Фрагмент кода класса «ContactList.jsx»

```
constructor() {
  super();
  this.state = { data: [] }

  static contextTypes = { router: React.PropTypes.object.isRequired }

  addEvent = () => { this.context.router.push('/addcontact'); }

  componentDidMount() {
    fetch('http://localhost:3001/contacts').then((res) => res.json()).then((data) =>
    { this.setState({ data }); });
  }

  render() {
    return (
      <div style={styles.event}>
        <List>
          { this.state.data.map((e) => (<Contact key={e.id} id={e.id}
name={e.name} avatar={e.avatar} tag={e.tag}/>)) }
        </List>
        <FloatingActionButton style={styles.button}
onTouchTap={this.addEvent}>
          <ContentAdd/>
        </FloatingActionButton>
      </div>);
  }
}
```

## ПРИЛОЖЕНИЕ Д

### Фрагмент кода класса «Contact.jsx»

```
static contextTypes = { router: PropTypes.object.isRequired }

contactDetails = (e) => {
  this.context.router.push(`/contact/${this.props.id}`);
}

render() {
  const {
    name,
    tag,
    avatar,
  } = this.props;
  const image = <Avatar src={avatar}/>;
  return (
    <div>
      <Paper>
        <ListItem
          primaryText={name}
          leftAvatar={image}
          secondaryText={tag.map((e, i) => (<span style={styles.tags}
key={i} onTouchTap={this.test}>{e} </span>))}
          onTouchTap={this.contactDetails}
        />
      </Paper>
    </div>);}
```