|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

КАФЕДРА \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**К КУРСОВОЙ РАБОТЕ**

**НА ТЕМУ:**

**Разработка сервиса автоматизации регистрации участников на мероприятия.**

Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

(Группа) (Подпись, дата) (И.О.Фамилия)

Руководитель курсовой работы **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

(Подпись, дата) (И.О.Фамилия)

Консультант **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

(Подпись, дата) (И.О.Фамилия)

Москва, 2024 г.

Содержание

[**Введение** 3](#_Toc176510277)

[**1 Обзор предметной области** 4](#_Toc176510278)

[1.1 Обзор классических способов автоматизации регистрации на мероприятия 4](#_Toc176510279)

[1.2 Актуальность автоматизации через telegram бот 5](#_Toc176510280)

[1.3 Практическая реализация 6](#_Toc176510281)

[1.3.1 Предметная область и требования 6](#_Toc176510282)

[1.3.2 Модель «Сущность-связь» 6](#_Toc176510283)

[**2 Проектирование приложения** 10](#_Toc176510284)

[2.1 Предметная область и требования 10](#_Toc176510285)

[2.2 Реляционная модель 10](#_Toc176510286)

[2.3 Проектирование архитектуры приложения 10](#_Toc176510287)

[**3 Разработка приложения** 11](#_Toc176510288)

[**4 Тестирование** 12](#_Toc176510289)

[**5 Заключение** 13](#_Toc176510290)

[**6 Список использованных источников** 14](#_Toc176510291)

# **Введение**

В условиях активной общественной жизни и большого числа мероприятий, проводимых в высших учебных заведениях, автоматизация процесса регистрации участников становится важной частью эффективного управления событиями. В частности, в МГТУ имени Баумана ежегодно проводится не менее пятисот различных мероприятий. Важно обеспечить оперативную и удобную регистрацию участников, чтобы каждое событие проходило на высоком организационном уровне.

Быстрая и эффективная регистрация участников на мероприятия позволяет существенно сократить временные затраты и повысить удобство для студентов, преподавателей и гостей университета. Возможность зарегистрироваться на мероприятие онлайн ускоряет процесс для участников, что особенно актуально в условиях большого количества проводимых мероприятий.

Для организаторов, автоматизация процесса регистрации открывает дополнительные возможности для управления массовыми мероприятиями. При проведении более пятисот мероприятий в год крайне важно иметь инструменты, которые позволяют эффективно обрабатывать и систематизировать данные о зарегистрированных участниках.

Таким образом, в условиях высокой активности и большого количества проводимых мероприятий, как в МГТУ имени Баумана, автоматизация регистрации является необходимым инструментом для обеспечения успешной организации и проведения событий, а также для повышения уровня удовлетворенности участников и организаторов.

# **1 Обзор предметной области**

## Обзор классических способов автоматизации регистрации на мероприятия

Автоматизировать регистрацию участников на мероприятия можно тремя основными способами:

- Google или Yandex формы;

- Веб-сайт;

- Telegram бот;

В оценке качества различных сервисов регистрации на мероприятия метриками являются удобство для организаторов, трудоёмкость организации рассылок, время на создание регистрации, простота внесения изменений и удобство для конечного пользователя.

Бот в Telegram проявляет себя как удобное решение для организаторов, интегрируясь с системой мессенджера и предоставляя возможности адаптации для различных пользователей. Организации рассылок легко интегрируется на этапе разработки. Создание регистрации требует значительного времени, поскольку связано с разработкой программного продукта. Внесение изменений также трудоёмко, из-за необходимости внесения изменений в исходный код программного продукта.

Google или Yandex Форма обеспечивает удобное взаимодействие и быструю настройку регистрации, требующую лишь списка вопросов. Форма не является адаптивной, что ограничивает ее в использовании разными группами участников. Организация рассылок представляет собой ручное написание сообщений каждому участнику. Внесение изменений в форму представляет собой не трудоёмкий процесс.

Веб-сайт предоставляет удобный интерфейс для регистрации и обладает возможностями адаптивности к разным группам участников, если это предусмотрено при разработке. Организация рассылок не представляет собой трудоёмкий процесс, если данный функционал интегрирован при разработке. Создание регистрации занимает много времени из-за трудоёмкого процесса разработки сайта. Внесение изменений также является сложным и требует обращения к разработчикам, что может потребовать дополнительных временных ресурсов.

Классические методы автоматизации регистрации на мероприятия (Google/Yandex формы, веб-сайты и Telegram-боты) обладают различными характеристиками, которые необходимо учитывать при выборе подходящего решения. Google и Yandex формы отличаются простотой настройки и минимальными временными затратами, однако ограничены в функциональности и адаптивности, что делает их менее универсальными для использования с разными группами участников. Веб-сайты и Telegram-боты, напротив, предоставляют широкие возможности для кастомизации и интеграции, однако их создание требует значительных ресурсов, как временных, так и технических, а внесение изменений в такие системы связано с высокой трудоёмкостью.

## Актуальность автоматизации через telegram бот

Автоматизация регистрации с использованием Telegram-бота обладает рядом преимуществ, что делает данный подход предпочтительным для организаторов мероприятий. Прежде всего, Telegram-боты обеспечивают интеграцию с популярной платформой обмена сообщениями, что значительно упрощает взаимодействие пользователей с системой. Это исключает необходимость установки дополнительного программного обеспечения или использования внешних веб-ресурсов, что повышает доступность и удобство регистрации для участников.

Telegram-боты обладают высокой степенью адаптивности, предоставляя организаторам возможность настроить сбор данных в соответствии с требованиями конкретного мероприятия. Инструмент позволяет автоматизировать процесс рассылок, отправку уведомлений и напоминаний, что снижает трудозатраты на коммуникации с участниками. Важным преимуществом является возможность интеграции бота с внешними системами, такими как базы данных или сторонних API, что облегчает хранение и обработку информации.

Хотя разработка Telegram-бота требует первоначальных затрат ресурсов, этот метод позволяет значительно снизить затраты времени на последующих этапах регистрации и обработки данных. Бот также может быть масштабируемым, обеспечивая регистрацию для мероприятий различного масштаба и адаптацию под особенности различных категорий пользователей

Таким образом, при решении проблем, связанных с разработкой и модификацией telegram-бота, будет получен универсальный сервис для автоматизации регистрации участников на мероприятия, сочетающий в себе гибкость, удобство и высокую степень автоматизации регистрации.

## Практическая реализация

### 1.3.1 Предметная область и требования

### 1.3.2 Модель «Сущность-связь»

В ER модель (рисунок 1) включено семь сущностей:

1) bot\_owner - сущность, характеризующая пользователя, который создаёт бота, т. е. представителя студенческой организации, проводящей мероприятие. Идентификатор сущности «login», атрибуты сущности:

- password\_hash – пароль в хэшированном представлении;

- email – электронная почта;

- telegram\_link – ссылка на telegram;

- chatId – идентификатор пользователя в telegram;

2) bot - сущность, характеризующая генерируемого telegram бота с идентификатором «telegram\_token» и атрибутами:

- name – название бота;

- team\_limit – ограничение на максимальное количество участников в команде;

- start – указатель на первый вопрос регистрации;

3) block - сущность, представляющая вопрос регистрации. Атрибуты сущности:

- type – атрибут, определяющий тип элемента регистрации - элемент перечисления из трёх элементов: “Вопрос”, “Сообщение”, “Выбор”;

- default – указатель на следующий элемент регистрации, не пустой в случае непустого списка кнопок у элемента регистрации;

- title – название элемента регистрации;

- text - текст сообщения, выводимого конечному пользователю в Telegram;

- state – порядковый номер блока;

4) option - сущность, представляющая собой кнопку для элемента регистрации, с атрибутами:

- value - значение, которое будет считаться ответом конечного пользователя после нажатия на кнопку;

- next - указатель на элемент регистрации, который следует после нажатия на данную кнопку;

5) answer - сущность, представляющая собой ответ конечного пользователя на вопрос регистрации с единственным атрибутом:

- value - текст ответа на вопрос;

6) participant – сущность, представляющая собой конечного участника мероприятия с идентификатором chatId - представляющий собой уникальный идентификатор, предоставляемый telegram API и атрибутами:

- current\_state – номер вопроса, на который отвечает пользователь в данный момент;­­

8) mailing - сущность представляющая собой рассылку внутри telegram бота с атрибутом:

- text - текст рассылки.

Между описанными сущностями были построены связи, согласующиеся с особенностями предметной области описанных в пункте 1.3.1 .

Организатор мероприятия, представляемый сущностью “bot\_owner” может создавать произвольное количество ботов для регистрации, следовательно между сущностями “bot\_onwer” и “bot” образуется связь “один ко многим”, а минимальные кардинальные числа равны 1 и 0 соответственно.

В свою очередь каждый бот имеет произвольное количество участников, а участники также могут участвовать в произвольном количестве мероприятий, поэтому между сущностями “bot” и “participant” образуется связь “многие ко многим” с минимальными кардинальными числами 1 и 0 соответственно.

Количество рассылок внутри каждого бота также произвольно, поэтому между сущностями “bots” и “mailings” образуется связь “Один ко многим” с минимальными кардинальными числами 1 и 0 соответственно.

Каждый бот однозначно можно представить в виде не пустого набора элементов регистрации, представляемого сущностью “blocks”, поэтому между сущностями “bots” и “blocks” образуется связь “Один ко многим” с минимальными кардинальными числами 1 и 1 соответственно.

В каждом боте содержится произвольное количество ответов участников на вопросы регистрации, поэтому между сущностями “bots” и “answers” образуется связь “Один ко многим” с минимальными кардинальными числами 1 и 0 соответственно.

Каждый элемент регистрации может ссылаться на другие элементы регистрации посредством атрибута “default”, поэтому между сущностями “blocks” и “blocks” образуется рекурсивная связь типа “Один к одному” с минимальными кардинальными числами 1 и 0, потому что элементы регистрации могут ссылаться друг на друга и посредством сущности “options”.

Каждый блок регистрации типа “Выбор” может иметь конечный набор кнопок, который представляется сущностью “option”, поэтому между сущностями “blocks” и “option” образуется связь вида “Один ко многим” с минимальными кардинальными числами 1 и 0 соответственно. В свою очередь каждой кнопке соответствует ровно один следующий элемент регистрации, поэтому между сущностями “blocks” и “options” образуется связь “Один к одному” с минимальными кардинальными числами 1 и 1.

Каждый ответ должен ссылаться на один из существующих блоков регистрации для дальнейшей идентификации, поэтому между сущностями “blocks” и “answers”  образуется связь  “Один ко многим” c минимальными кардинальными числами 1 и 0 соответственно .

Помимо этого, каждый ответ должен соотноситься с участником мероприятия, который ответил на него, поэтому между сущностями “participants” и “answers” образуется связь “Один ко многим” с минимальными кардинальными числами 1 и 0 соответственно.

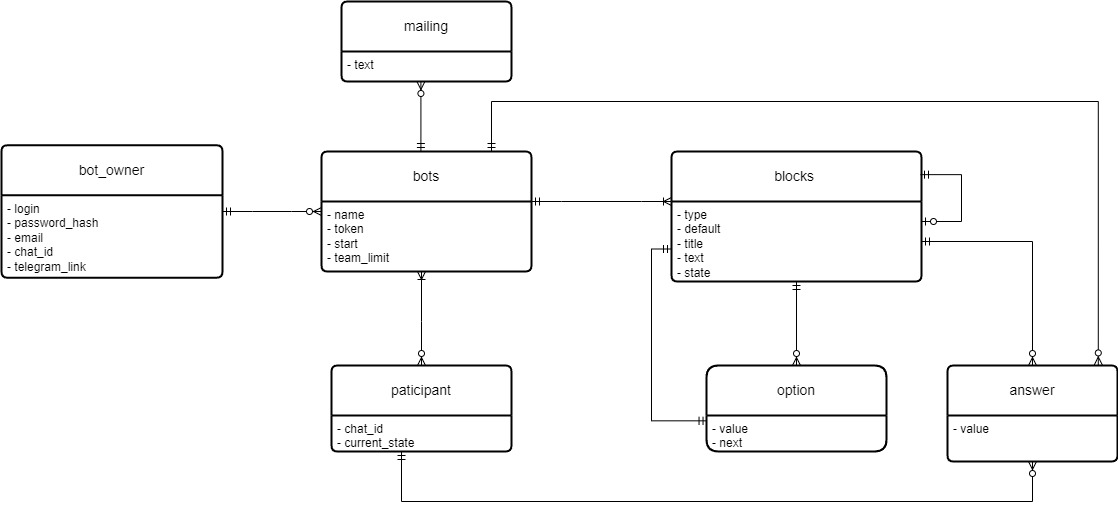


Рисунок 1. Модель "Сущность-связь"

# **2 Проектирование приложения**

## 2.1 Предметная область и требования

## 2.2 Реляционная модель

В соответствии с правилами преобразования, из созданной ранее модели «сущность-связь», была получена реляционная модель, представленная на рисунке номер 2.

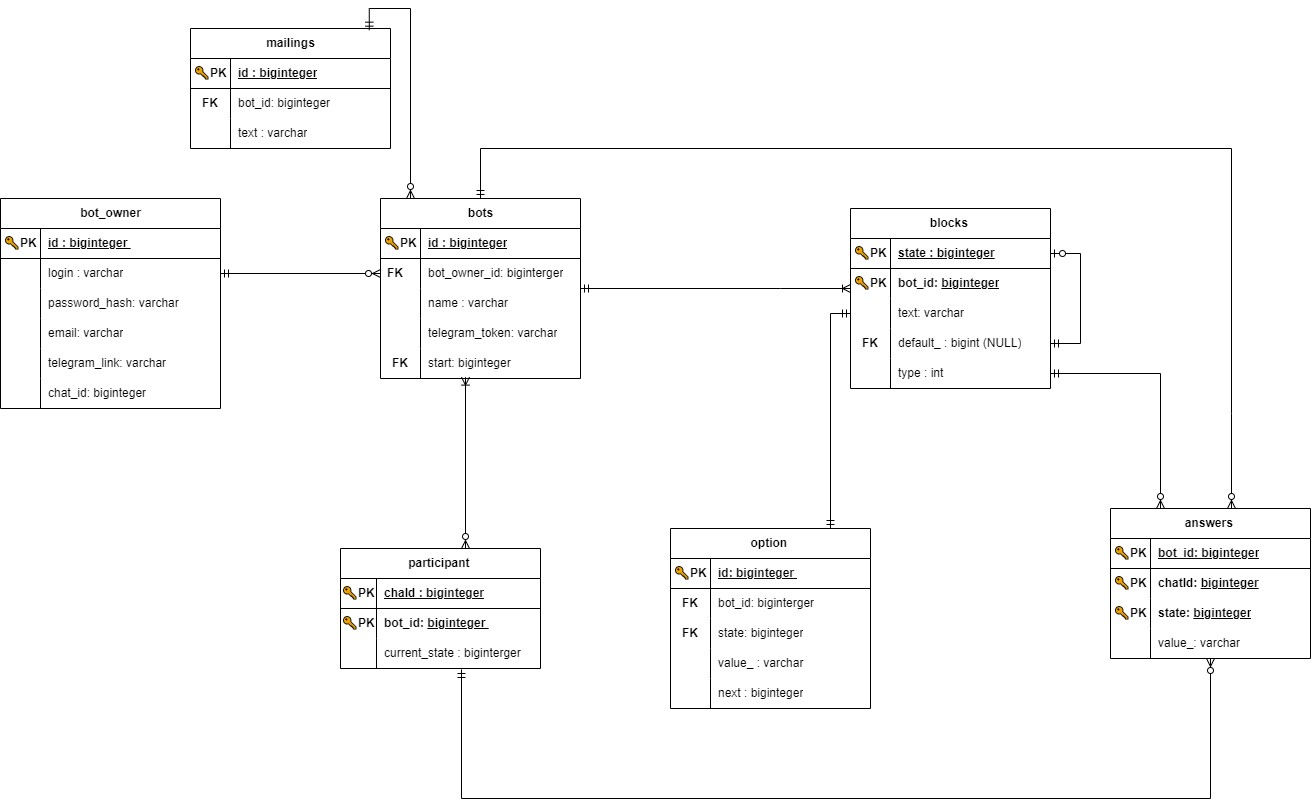


Рисунок 2 - Реляционная модель

# **3** **Разработка приложения**

## 3.1 Разработка архитектуры приложения

В разрабатываемом сервисе можно выделить три основные блока, представленных на рисунке 3:

- Сервис авторизации;

- Сервис работы с ботами;

- Пользовательский интерфейс.

Сервис авторизации имеет свою базу данных с единственной таблицей “bot\_owner”, описанной в разделе 1.3.2, и производит поиск по существующим организаторам или осуществляет регистрацию нового организатора мероприятия. После успешной авторизации, организатору предоставляется JWT (JSON web token) токен, который предоставляет доступ к взаимодействию с сервисом работы с ботами.

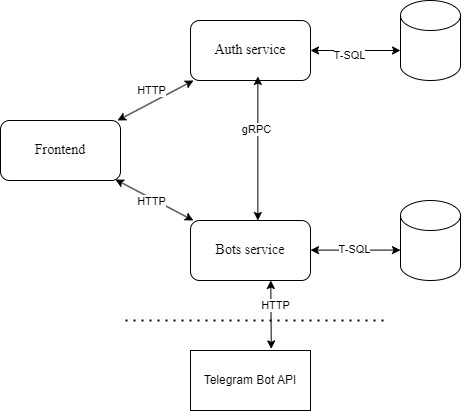


Рисунок 3 - Архитектура приложения.

## 3.2 Ключевые типы данных и функции

# **4 Тестирование**

# **5 Заключение**

# **6 Список использованных источников**