

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Институт космических и информационных технологий
институт
Информационных систем
кафедра

УТВЕРЖДАЮ
Заведующий кафедрой ИС
_____ П.П.Дьячук
подпись инициалы, фамилия
« ____ » _____ 2019 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.02 – «Информационные системы и технологии»

Повышение эффективности деятельности кастинг-менеджера

Руководитель	_____	<u>ст. преподаватель</u>	<u>Ю.В.Шмагрис</u>
	подпись, дата	должность, ученая степень	
Консультант	_____	<u>к.т.н., доцент</u>	<u>И.А.Легалов</u>
	подпись, дата	должность, ученая степень	
Выпускник	_____		<u>С.С.Мукомолова</u>
	подпись, дата		
Нормоконтролер	_____	<u>ст. преподаватель</u>	<u>Ю.В.Шмагрис</u>
	подпись, дата	должность, ученая степень	

Красноярск 2019

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Институт космических и информационных технологий
институт
Информационных систем
кафедра

УТВЕРЖДАЮ
Заведующий кафедрой ИС

_____ П.П.Дьячук
подпись инициалы, фамилия

« ____ » _____ 2019 г.

ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ
в форме бакалаврской работы

Студентке Мукомоловой Софье Сергеевне

Группа: КИ15-13Б Направление: 09.03.02 «Информационные системы и технологии»

Тема выпускной квалификационной работы: «Повышение эффективности деятельности кастинг-менеджера»

Утверждена приказом по университету № 7237/с от 24.05.2019 г.

Руководитель ВКР: Ю.В.Шмагрис, старший преподаватель кафедры «Информационные системы» ИКИТ СФУ.

Консультант ВКР: И.А.Легалов, кандидат технических наук, доцент кафедры «Информационные системы» ИКИТ СФУ.

Исходные данные для ВКР: список требований к разрабатываемой системе, методические указания научного руководителя.

Перечень разделов ВКР: введение, общие сведения, средства разработки, разработка и реализация веб-сервиса, заключение, список использованных источников.

Перечень графического материала: презентация, выполненная в Microsoft Office PowerPoint 2016.

Руководитель ВКР _____
подпись

Ю.В.Шмагрис

Консультант ВКР _____
подпись

И.А.Легалов

Задание принял к исполнению _____
подпись

С.С.Мукомолова

« ____ » _____ 2019 г.

РЕФЕРАТ

Выпускная квалификационная работа по теме «Повышение эффективности деятельности кастинг-менеджера» содержит 42 страницы текстового документа, 37 рисунков, 15 использованных источников.

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ, ИНФОРМАЦИОННЫЕ СИСТЕМЫ, ВЕБ-СЕРВИС, ПРОЕКТИРОВАНИЕ.

Цель работы – разработка веб-сервиса для кастинг-менеджера.

Основные задачи:

- разработать веб-сервис;
- обеспечить автоматизацию ручной работы кастинг-менеджера;
- провести анализ эффективности веб-сервиса.

В ходе данной работы был разработан веб-сервиса поиска ресурсов, благодаря которому повысилась эффективность работы кастинг-менеджера, так как большинство процессов стали выполняться автоматически.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	4
ГЛАВА 1 ОБЩИЕ СВЕДЕНИЯ	5
1.1 Анализ предметной области	5
1.2 Выявление и постановка проблемы.....	8
1.3 Понятие информационной системы, классификация	9
1.4 Процессы, нуждающиеся в автоматизации.....	13
1.5 Выводы по первой главе	13
ГЛАВА 2 СРЕДСТВА РАЗРАБОТКИ ИНФОРМАЦИОННОЙ СИСТЕМЫ.....	14
2.1 Требования к разрабатываемой информационной системе	14
2.2 БД, СУБД, классификация проектируемой БД	14
2.3 Выбор подходящих средств разработки.....	17
2.4 Выводы по второй главе	20
ГЛАВА 3 РАЗРАБОТКА И РЕАЛИЗАЦИЯ ВЕБ-СЕРВИСА	21
3.1 Описание бизнес-процессов	21
3.1.1 Диаграмма классов	22
3.1.2 Use-case диаграмма.....	23
3.1.3 Диаграмма деятельности.....	24
3.1.4 Диаграмма компонентов	26
3.2 Реализация	26
3.3 Подробности реализации	29
3.4 Анализ полученных результатов.....	37
ЗАКЛЮЧЕНИЕ	40
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	41

ВВЕДЕНИЕ

На сегодняшний день сложно представить работу любого предприятия без использования информационных технологий. Это позволяет решать множество задач, например: ускорение бизнес-процессов, оптимизация рабочего времени, экономия трудозатрат, повышение производительности труда. Единственный метод реализации задач – внедрение автоматизированной системы.

Зачастую в автоматизации нуждается не целое предприятие, а работа одного человека. Например, это работа кастинг-менеджера. Данная профессия подразумевает под собой подбор и найм актёров или поиск ресурсов для фильма, сериала, рекламы, музыкального клипа и т.д. Этот процесс отнимает достаточно много времени, потому что поиск людей и оповещение о времени и месте мероприятия происходит вручную.

Целью выпускной квалификационной работы является оптимизация деятельности кастинг-менеджера посредством автоматизации процесса поиска необходимых ресурсов (человеческих, материальных).

Необходимо разработать информационную систему, которая автоматизирует работу кастинг-менеджера, значительно уменьшив время на поиск людей и их оповещение. Для выполнения поставленной цели были сформированы следующие задачи:

1. провести анализ предметной области;
2. выбрать способы решения проблемы;
3. разработать веб-сервис поиска;
4. провести анализ результатов автоматизации.

ГЛАВА 1 ОБЩИЕ СВЕДЕНИЯ

1.1 Анализ предметной области

Появление новых профессий в кино – процесс неизбежный. Кинематограф развивается и рождает специалистов, чьи услуги раньше не были востребованы. Кроме режиссера, оператора, продюсера и актеров, в кино работают десятки людей разных профессий. Например – кастинг менеджер - человек, который отвечает за подбор и найм актёров для фильма, сериала, рекламы, музыкального клипа и т.д.

Кастинг-менеджер осуществляет не только подбор актеров на главные роли, но также ищет большое количество массовки, танцоров, певцов, в зависимости от сложности проекта. Хотя и в подборе актеров окончательное решение принимает режиссер или продюсер, кастинг-менеджер играет важную роль, так как от его работы зависит насколько хорошо будут подобраны люди в проект.

В обязанности кастинг-менеджера входит:

- поиск людей. Часто проекты бывают настолько крупными, что требуется найти от 250 человек и более;
- приглашение «отобранных» людей на кастинг. Кастинг-менеджер должен сообщить всем людям во сколько и куда они должны прибыть;
- формирование окончательного списка на мероприятие.

Работа кастинг-менеджера трудоемка, так как все обязанности выполняются вручную.

На данный момент времени существует несколько способов автоматизации работы кастинг-менеджера, например:

- 1) Социальная сеть;

Кастинг-менеджер может размещать объявления в своих социальных сетях.

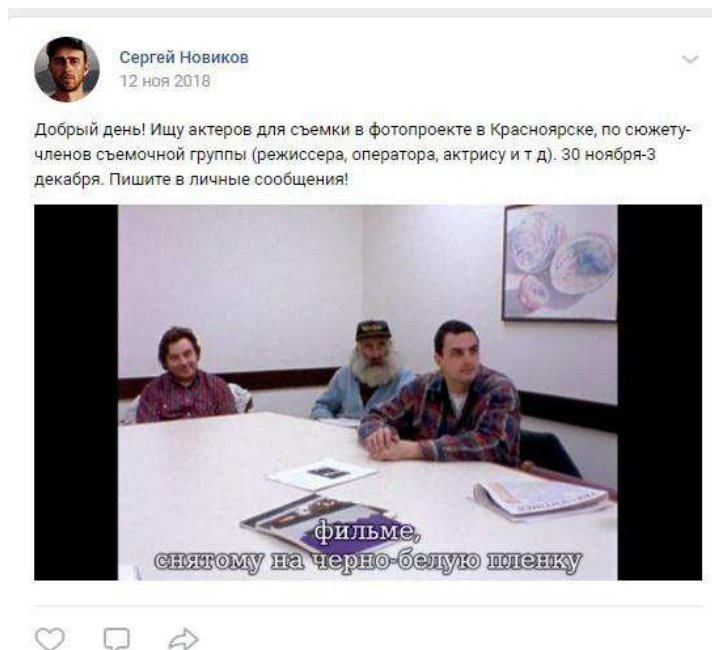


Рисунок 1 – Пример объявления в социальной сети

Это облегчает поиск людей, но временные затраты на составление списка и последующую связь с людьми не сокращаются.

2) Google Forms;

С помощью данного сервиса можно создавать онлайн-опросы и отправлять их другим пользователям.

Кастинг-менеджер может использовать данный сервис для того, чтобы создавать формы о предстоящем мероприятии, размещать ссылку на форму в своих социальных сетях. В итоге формируется список людей, который можно распечатать.

ВОПРОСЫ ОТВЕТЫ

Съемки музыкального клипа группы "Кофе"

Здравствуйте! Для съемки музыкального клипа нам нужны 20 девушек от 18 до 25, занятость 1.05 с 9 до 21.
Адрес - г. Санкт-Петербург, [Севкабель](#) порт

Фамилия, имя *

Краткий ответ

Во сколько можете приехать по времени

могу быть с утра до вечера

могу только утром

могу только после обеда

Рисунок 2 – Пример формы для участия в съемке

Очевидный минус данного сервиса – последующая связь с человеком происходит вручную, что не экономит время.

3) Всекастинги.ру;

Всекастинги.ру – сайт, где собраны лучшие предложения кастингов и объявлений. Можно разместить объявление о поиске, либо откликнуться на кастинг.

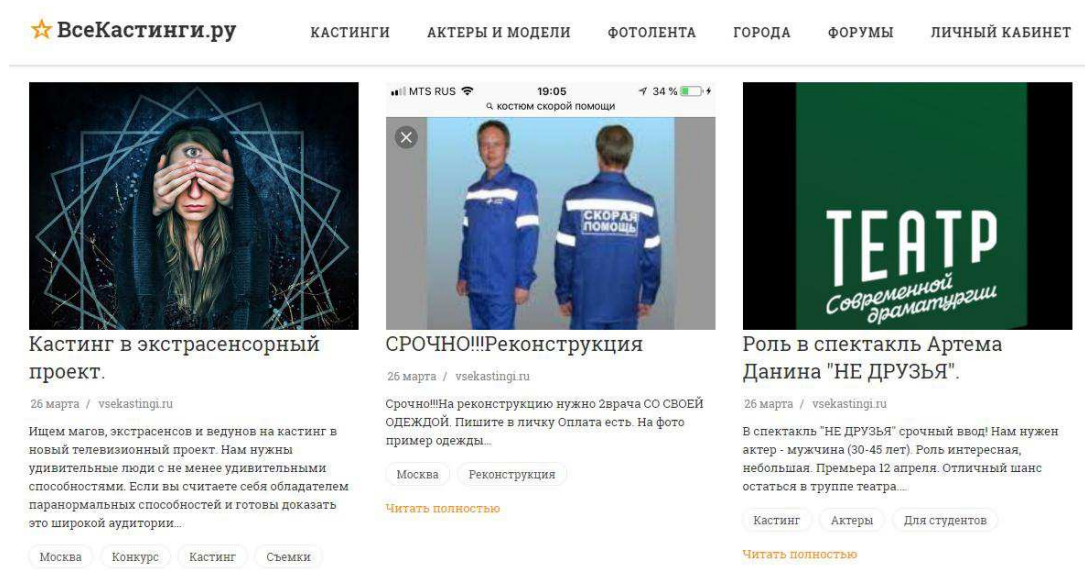


Рисунок 3 – Домашняя страница сайта Всекастинги.ру

Минус данного сервиса в том, что кастинг-менеджер получает только список откликов. Нет возможности дальнейшей рассылки откликнувшимся, этот процесс кастинг-менеджер будет вынужден произвести вручную. Также нет возможности «утвержденных» людей сформировать в список с указанием контактных данных.

Приведенные выше конкуренты облегчают работу кастинг-менеджера, но по-прежнему некоторые процессы вынужденно выполняются вручную и занимают большое количество времени, от чего снижается эффективность работы.

1.2 Выявление и постановка проблемы

В ходе анализа работы кастинг-менеджера была выявлена главная проблема – процесс поиска и последующей организации занимает большое количество времени. Следовательно, главная задача – оптимизировать деятельность кастинг-менеджера посредством автоматизации процесса, значительно сократив время на поиск людей и ресурсов для мероприятия; оповещение «отобранных» людей о времени и месте мероприятия; формирование списков.

За счет появления информационной системы вместо ручной работы сократится время на выполнение бизнес-процессов, ускорится процесс поиска людей и ресурсов, а также организационные процессы до мероприятия, в связи с чем повысится эффективность работы кастинг-менеджера.

Информационная система позволит решить такие организационные задачи, как: размещение заявки о поиске людей или ресурсов, возможность записаться на съемку или предоставить ресурсы, составление общего списка людей на мероприятие, рассылка о времени и месте мероприятия людям из списка.

Разрабатываемая информационная система будет реализована в виде веб-сервиса.

В соответствии с проанализированной проблемой, необходимо определить классификацию информационной системы, выяснить, какие аспекты работы кастинг-менеджера необходимо автоматизировать.

1.3 Понятие информационной системы, классификация

Информационная система (ИС) – совокупность информационных, экономическо-математических методов и моделей, технических, программных, технологических средств и специалистов, предназначенных для сбора, хранения, обработки и выдачи информации, а также принятия управленческих решений.

Существует несколько классических примеров информационных систем, например – 1С-Бухгалтерия, система SAP R3, справочно-правовая система «Консультант Плюс» и т.д.

Информационные системы можно классифицировать по различным признакам. Например, основные классификации представлены на рисунке 4. Рассмотрим важнейшие из них.



Рисунок 4 – Классификация информационных системы

По степени автоматизации информационных процессов в системе, информационные системы делятся на:

- ручные;
- автоматические;
- автоматизированные;

Ручным информационным системам характерно полное отсутствие технических средств переработки информации, все операции выполняются человеком; автоматическим информационным системам характерно выполнение всех операций переработки информации без участия человека; автоматизированным системам характерно участие как человека, так и технических средств, но главная роль отдается компьютеру.

По масштабу информационные системы делятся на:

- однопользовательские;
- групповые;
- корпоративные;

Однопользовательские информационные системы предназначены для личного использования на одном рабочем месте. В настоящее время на замену однопользовательским информационным системам пришли табличные процессоры не имеющие специализации, например - Microsoft Excel.

Групповые информационные системы предназначены для использования в малых рабочих группах (малое предприятие, отдел). Групповые системы, как правило, представляют собой специализированные клиентские решения для различных участников группы. Например, системы как «Врач», «Работник регистратуры», «Больной» применимы к сфере планирования работы в больнице.

Корпоративные информационные системы (КИС) – предназначены для автоматизации деятельности в крупных компаниях. КИС – комплекс, который содержит все основные аспекты управленческой деятельности предприятий – планирование ресурсов, все виды учета и анализа деятельности и т.д.

Архитектура информационной системы – концепция, определяющая структуру, выполняемые функции и взаимосвязь компонентов информационной системы.

Существует три основных типа архитектур информационных систем:

- «файл - сервер»;
- «клиент - сервер»;
- трехслойная архитектура;

Архитектура «файл - сервер» - архитектура, где модули и данные размещаются в отдельных файлах операционной системы, доступ к которым осуществляется путем указания пути и использования файловых операций (открыть, читать, записать, удалить). Данные хранятся на выделенном сервере (отдельный компьютер), который и является файловым сервером. Исполняемые модули хранятся на сервере, либо на рабочих станциях.

Архитектура «клиент – сервер» - архитектура, в которой все вычисления распределены между серверами и клиентами. Клиент запрашивает те или иные сервисы в соответствии с определенным протоколом обмена данными. В отличие от архитектуры «файл – сервер», клиенту «известны» лишь имя источника данных и специальные сведения, используемые для авторизации клиента на сервере. Сервер может находиться удаленно, обрабатывая запросы клиента удаленно, с помощью интернет-протоколов.

В подобных системах существует два понятия – «толстый» и «тонкий» клиент. В системах на основе «толстого» клиента, вся бизнес-логика реализована на стороне клиента, сервер представляет собой сервер баз данных, обеспечивает только выполнение стандартных запросов (чтение, запись, модификация с данными). Системы с «тонким» клиентом основаны на взаимодействии с мощным сервером, который выполняет наибольшую часть бизнес-логики, в то время как клиент не имеет большой вычислительной мощности. Основное достоинство таких систем – дешевые клиентские станции.

Трехслойная архитектура – архитектура, предполагающая наличие трех составляющих – клиент, сервер приложений и сервер баз данных. С развитием

интернет-технологий появилась разновидность трехслойной архитектуры – на основании использования веб-технологий. В качестве сервера приложений выступает веб-сервер, а в качестве клиента используется веб-браузер.

Информационные системы можно разделить на информационно-поисковые и управляющие.

Информационно-поисковые предоставляют доступ к данным только в режиме чтения или поиска. А доступ к модификации данных имеет только администратор. Например, google.com и yandex.ru.

Управляющие системы – системы, которые автоматизируют деятельность, связанную с принятием решений. Системами такого типа могут быть системы бухгалтерского учета и т.д.



Рисунок 5 – Классификация разрабатываемой ИС

Резюмируя всё вышесказанное, разрабатываемая информационная система – автоматизированная; управляющая, т.к. клиенты могут модифицировать информацию; с применением архитектуры «клиент-сервер».

1.4 Процессы, нуждающиеся в автоматизации

Создание информационной системы подразумевает автоматизацию следующих процессов:

- процесс поиска людей (размещая заявку на сайте, не нужно тратить время на самостоятельный поиск людей);
- процесс поиска ресурсов;
- приглашение «отобранных» людей (не нужно оповещать вручную большое количество людей);
- оповещение о месте мероприятия (также будет происходить автоматически);
- формирование страницы мероприятия, включающая в себя информацию об участниках, их контактных данных.

1.5 Выводы по первой главе

Проанализировав работу кастинг-менеджера, была выявлена проблема - некоторые процессы вынужденно выполняются вручную и занимают большое количество времени, от чего снижается эффективность работы.

Для повышения эффективности деятельности кастинг-менеджера будет разработана информационная система, которая позволит автоматизировать организационные задачи.

Информационная система будет реализована в виде веб-сервиса. Информационная система будет автоматизированной; по характеру использования информации – управляющей; с применением клиент-серверной архитектуры.

Также были названы основные процессы, которые будет автоматизировать информационная система.

ГЛАВА 2 СРЕДСТВА РАЗРАБОТКИ ИНФОРМАЦИОННОЙ СИСТЕМЫ

2.1 Требования к разрабатываемой информационной системе

Проведя анализ предметной области, а также рассмотрев классификации информационной системы, необходимо разработать требования, которым должна отвечать разрабатываемая информационная система.

Технические требования разрабатываемого веб-сервиса:

- высокая скорость (при загрузке страницы, при вводе и обработке информации);
- дружелюбность интерфейса (эффективное расположение элементов сайта для сокращения времени, затраченного пользователем для достижения цели);
- адаптивность интерфейса (отображение веб-страницы на любом устройстве);
- отказоустойчивость (размещение сервера информационной системы на хостинге, который обеспечит бесперебойную работу сервиса);

Требования безопасности разрабатываемого веб-сервиса:

- идентификация и аутентификация пользователей;
- доступ к информации в зависимости от прав пользователя;

2.2 БД, СУБД, классификация проектируемой БД

Базы данных (БД) – совокупность данных, характеризующая состав объектов предметной области, их свойства, взаимосвязи, хранящаяся в памяти вычислительной машины. Содержит в себе программные методы и средства, с помощью которых можно модифицировать данные.

Базы данных имеют ряд свойств:

- базы данных хранятся и обрабатываются посредством вычислительной системы;

- все данные в БД структурированы;

- в БД присутствует логическая структура данных в соответствии с заданной моделью.

Базы данных можно классифицировать по трем главным свойствам:

- по характеру хранимой информации;

- по способу хранения;

- по структуре организации данных.



Рисунок 6 – Классификация БД

По характеру хранимой информации БД делятся на:

1) фактографические (содержат сведения об объектах, объединенных какой-либо предметной областью. Например, библиотечные книги);

2) документальные (содержат описание документов):

- имеющие только библиографическое описание;

- имеющие библиографическое описание и ключевые слова;

- имеющие библиографическое описание, ключевые слова, аннотацию.

По способу хранения данных:

- централизованные (данные хранятся на одной вычислительной машине – персональный компьютер, сервер в сети и т.д.);

- распределенные (распределены между компьютерами, находящимися в одной сети).

По структуре организации данных:

- реляционные (табличная форма организации);

- нереляционные (нерегламентированная структура организации);

- иерархические (представлены в виде иерархической структуры, состоящей из объектов различных уровней, где у каждого объекта может быть одна запись-потомок);

- сетевые (в отличие от иерархических, могут иметь любое число записей-предков).

Система управления базами данных (СУБД) – совокупность языковых и программных средств, предназначенных для создания, ведения и совместного использования базы данных многими пользователями.

Существует два типа СУБД:

- настольные (клиентские приложения, обеспечивающие чтение, запись и обработку данных БД. Взаимодействие с данными осуществляется посредством файловых сервисов операционной системы);

- серверные (обеспечивают обработку данных, находящихся в общедоступных хранилищах, как правило, на сервере данных, для нескольких пользователей одновременно. Серверные СУБД реализуют передачу запросов серверу БД и получение результата).



Рисунок 7 – Классификация разрабатываемой БД

Резюмируя всё вышесказанное, к разрабатываемой информационной системе подходит классификация «фактографическая», т.к. данные, содержащиеся в базе, содержат сведения о пользователях; классификация «централизованная», т.к. ее хранение будет осуществляться на одной вычислительной машине; классификация «реляционная», т.к. это самый простой и удобный способ хранения информации в информационной системе.

Будет использована настольная СУБД, т.к. аудитория сервиса небольшая.

2.3 Выбор подходящих средств разработки

Для разработки информационной системы было принято решение использовать язык программирования Python с применением фреймворка для построения интернет-приложений – Django.

Django – это высокоуровневая веб-инфраструктура Python, которая позволяет быстро создавать безопасные и поддерживаемые веб-сайты. Django представляет из себя фреймворк, использующий шаблон для разработки – MVC.



Рисунок 8 – Логотип Django

Некоторые возможности и особенности Django:

- реализация архитектуры «модель-контроллер-шаблон»;
- реализация принципа DRY (Don't Repeat Yourself – не повторяйся), в результате чего однажды использованный код может быть использован где угодно;
- интерфейс к встроенной платформе тестирования Python;
- унифицированные средства для работы с базами данных любых поддерживаемых форматов: SQLite, MySQL, PostgreSQL, Oracle, Microsoft SQL Server и др.;
- мощный шаблонизатор, основанный на специальных тегах, с возможностью наследования шаблонов;
- инструменты для разграничения доступа;
- богатые средства для работы с формами;
- встроенные средства для поддержки возможности комментирования, рассылки электронной почты и др.;
- простая и ясная структура создаваемых сайтов: каждый раздел сайта представляет собой отдельное приложение, которое может быть отчуждено и использовано в другом сайте.

Интерфейс веб-сервиса будет разработан на языке HTML, с применением таблицы стилей CSS. Благодаря языку HTML любые тексты можно разметить, преобразовав их в гипертекст с дальнейшей публикацией в web.

Для придания внешнего вида HTML-документам понадобится CSS, с помощью которого можно задавать для страницы и отдельных ее элементов

размеры шрифта, цвета элементов, отступы, расположения блоков на странице и др.



Рисунок 9 – Логотип стека разработки интерфейса

Данные будут храниться в СУБД PostgreSQL.

PostgreSQL – это объектно-реляционная система управления базами данных. Это СУБД, поддерживающая большую часть стандарта SQL и предлагающая множество современных функций.

PostgreSQL имеет свободную лицензию и может использоваться для любых целей. Реализован в архитектуре клиент-сервер. Django имеет хорошую интеграцию с PostgreSQL, что является причиной выбора данной СУБД для веб-сервиса.



Рисунок 10 – Логотип PostgreSQL

Для размещения разработанного приложения в Интернете будет использоваться облачный сервис – heroku. Он позволяет бесплатно развернуть небольшие веб-приложения.



Рисунок 11 – Логотип Heroku

2.4 Выводы по второй главе

В данной главе был проведен анализ требований к информационной системе. Основные требования – скорость, стабильность работы, адаптивность интерфейса, разные права доступа к информации на веб-сервисе.

База данных разрабатываемой информационной системы будет фактографической, централизованной и централизованной. СУБД настольной.

Также были выбраны подходящие средства разработки для интерфейса пользователя, серверной части и базы данных.

ГЛАВА 3 РАЗРАБОТКА И РЕАЛИЗАЦИЯ ВЕБ-СЕРВИСА

3.1 Описание бизнес-процессов

Исследуя предметную область, были выведены основные бизнес-процессы, которые необходимо автоматизировать в разрабатываемой информационной системе:

- регистрация пользователей (два вида пользователей – кастинг менеджер, который объявляет о съемках и пользователь, который на них откликается);
- размещение кастинг-менеджером заявки о поиске (актеров, людей массовой, необходимых материальных ресурсов);
- возможность откликнуться обычному пользователю на заявку (с указанием контактных данных и времени, в которое человек свободен);
- оповещение людей по почте о времени и месте съемки через автоматическую рассылку (кастинг-менеджер не тратит время на то, чтобы обзвонить всех людей вручную);
- возможность кастинг-менеджеру автоматически составить список людей, которые придут на съемку (не создавая для этого отдельных таблиц и не заполняя данные вручную).

Необходимо разработать диаграммы классов, компонентов, деятельности и use-case. Для выполнения работы была выбрана методология UML, т.к. она более просто объясняет процессы, происходящие внутри системы, тогда как, например, методология IDEF0 более ограничена и сложна в понимании и восприятии.

UML – язык графического описания для объектного моделирования в области разработки программного обеспечения, для моделирования бизнес-процессов, системного проектирования и отображения организационных структур. UML был создан для определения, визуализации, проектирования и документирования, в основном, программных систем.

3.1.1 Диаграмма классов

Диаграмма классов (Class diagram) — статическая структурная диаграмма, описывающая структуру системы, демонстрирующая классы системы, их атрибуты, методы и зависимости между классами.

В информационной системе есть 4 класса: класс «пользователи», класс «заявки», класс «участники заявки», класс «комментарии».

Как показано на рисунке 12, например, у класса «пользователи» есть поля данных – ID, имя, e-мэйл, пароль, права доступа (обычный пользователь или кастинг-менеджер). Также есть две функции – регистрация и авторизация.

У класса «заявки» есть поля данных – ID, имя, цель заявки (чем ты можешь помочь данному объявлению), дата отклика. Функции – создать, изменить, обработать заявку, добавить комментарий.

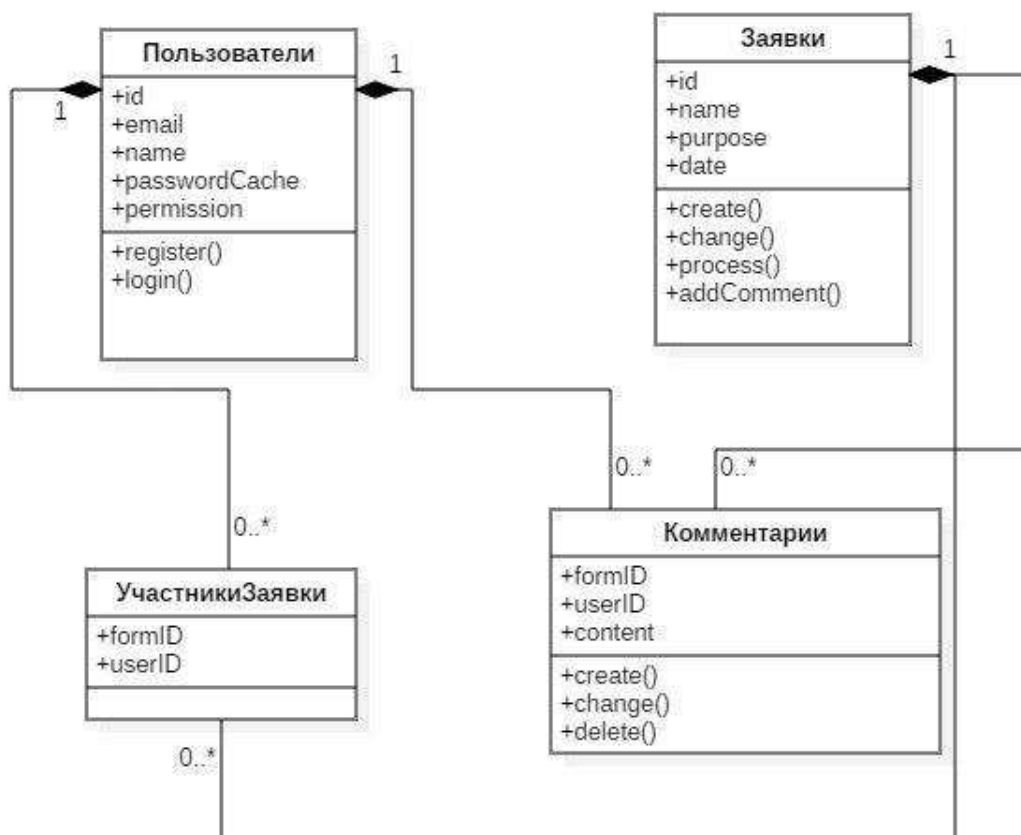


Рисунок 12 – Диаграмма классов

3.1.2 Use-case диаграмма

Диаграмма прецедентов (диаграмма вариантов использования) - диаграмма, отражающая отношения между актёрами и прецедентами и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне.

В системе имеются два вида пользователей: обычный пользователь, который, зарегистрировавшись в системе, может откликаться на заявки; кастинг-менеджер, который может размещать заявки. Каждый из них работает с системой, получая доступ лишь к своей части информации. Например, обычный пользователь не может увидеть, кто откликнулся на заявку кастинг-менеджера.

Основные возможности, предоставляемые информационной системой обычному пользователю:

- возможность регистрации и авторизации в системе;
- возможность просмотреть заявки;
- возможность комментировать заявки;
- возможность откликнуться на заявки.

Основные возможности, предоставляемые информационной системой кастинг-менеджеру:

- возможность регистрации и авторизации в системе;
- возможность просмотреть заявки;
- возможность комментировать заявки;
- возможность создать заявку;
- возможность отправить подтверждение пользователю о его участии.

Диаграмма прецедентов для двух видов пользователей изображена на рисунке 13.

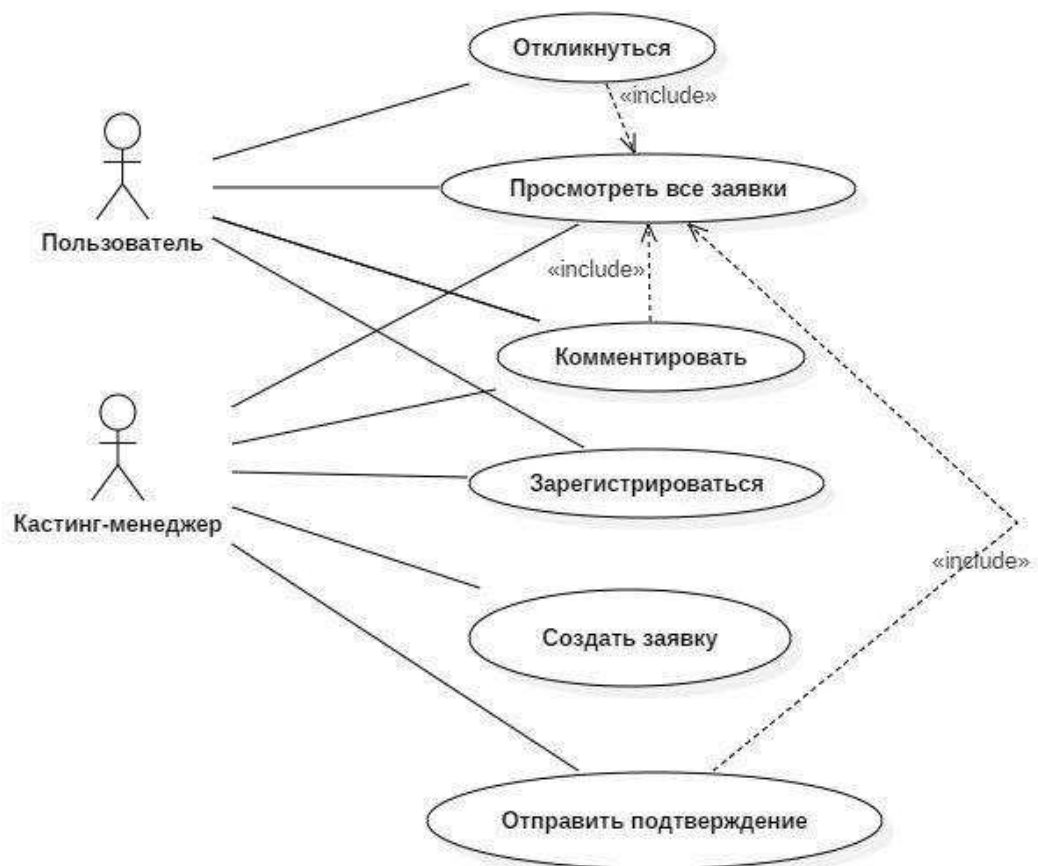


Рисунок 13 – Use-case диаграмма

3.1.3 Диаграмма деятельности

Диаграмма деятельности (Activity diagram) — диаграмма, на которой показано разложение некоторой деятельности на её составные части.

На рисунке 14 показана диаграмма деятельности процесса размещения заявки на веб-сервисе. При нажатии кнопки «разместить заявку», веб-сервис предлагает обозначить, какой вы пользователь. Если вы обычный пользователь, вы можете откликнуться на любую заявку, заполнив форму (указать фамилию, имя, отчество, контактные данные). Далее, спустя время, вы можете проверить, отправил ли вам кастинг-менеджер подтверждение об участии.

Если вы кастинг-менеджер, то вы создаете форму, публикуете ее в раздел заявок, далее, спустя время, можете проверять отклики. Если «откликнувшиеся» пользователи вас устраивают, вы рассылаете им подтверждение.

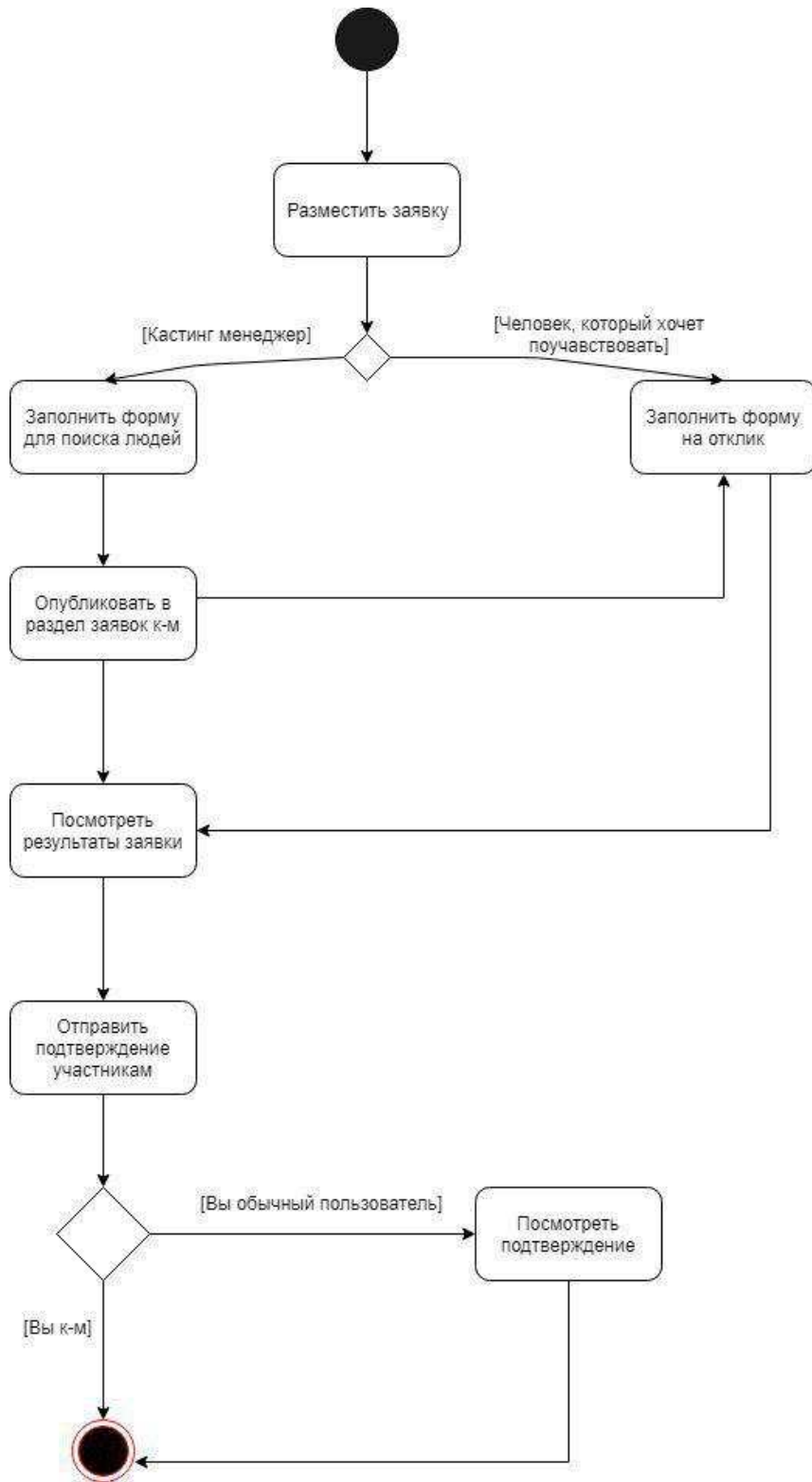


Рисунок 14 – Диаграмма деятельности

3.1.4 Диаграмма компонентов

Диаграмма компонентов (Component diagram) — статическая структурная диаграмма, показывает разбиение программной системы на структурные компоненты и связи (зависимости) между компонентами.

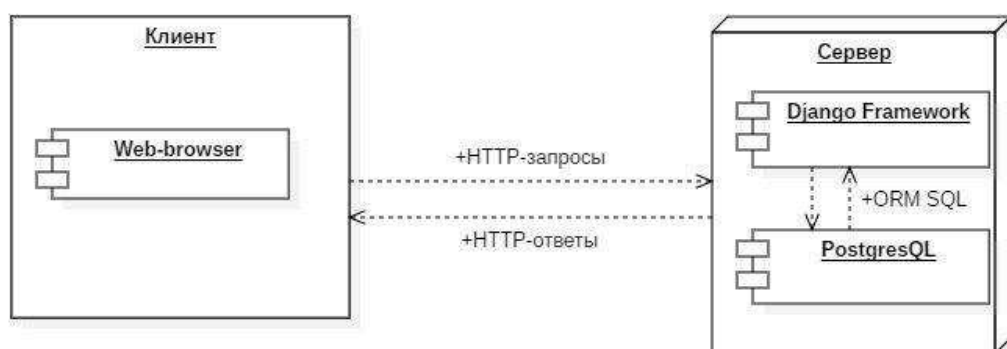


Рисунок 15 – Диаграмма компонентов

3.2 Реализация

MVC в Django немного видоизменён, вместо контроллеров используются Templates - шаблоны, которые описывают поведение функций на страницах HTML. На рисунке 14 показана структурная схема фреймворка.

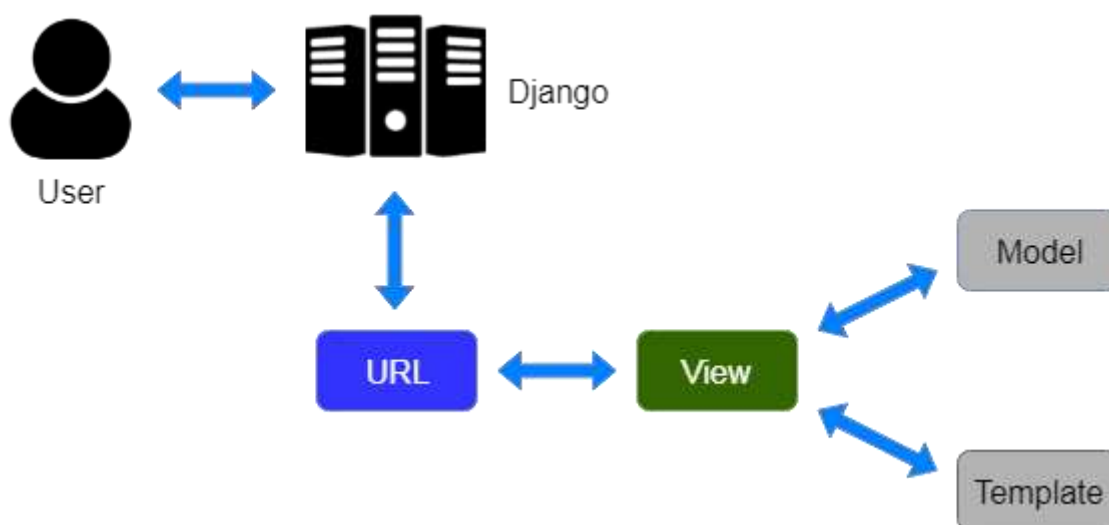


Рисунок 16 – Структурная схема Django

Данные информационной системы прописаны в моделях фреймворка. Django использует объектно-ориентированный подход. Модель представляет из себя класс с атрибутами, например, модель «заявки» имеет в себе атрибуты user (пользователь отправляющий заявку), name (имя заявки) и desc (описание заявки), на рисунке 17 показана структура моделей разработанной ИС.

```
class User(AbstractUser):
    is_actor = models.BooleanField(default=False)
    is_cm = models.BooleanField(default=False)
    first_name = models.CharField(max_length=30, blank=True)
    last_name = models.CharField(max_length=30, blank=True)
    email = models.EmailField(blank=True)
    phone = models.CharField(max_length=13, default='+7')

class Actor(models.Model):
    user = models.OneToOneField(User, on_delete=models.CASCADE, primary_key=True)

class Cm(models.Model):
    user = models.OneToOneField(User, on_delete=models.CASCADE, primary_key=True)

class Application(models.Model):
    user = models.ForeignKey(User, on_delete=models.CASCADE)
    name = models.TextField(max_length=250)
    desc = models.TextField(max_length=1000)

def __str__(self):
    return self.name
```

Рисунок 17 – Структура моделей в Django

Использование моделей упрощает взаимодействие с объектами внутри баз данных. Django используют собственную ORM – технологию, которая связывает СУБД и объектно-ориентированный подход в Django. Это позволяет упростить доступ к данным, что в последствии влияет на изменение добавление и удаление данных. Чтобы поместить наши модели в таблицы СУБД - используется встроенная в фреймворк функция миграций. Миграции конвертируют модели в язык баз данных SQL.

Логика процессов информационной системы прописана в представлениях фреймворка Django. Здесь написаны алгоритмы, которые получают и изменяют данные в базе данных, с помощью вышеупомянутой ORM, и производят их отправку по средствам HTML-запросов. На рисунке 18 можно увидеть функцию, которая получает информацию для отображения страницы профиля.

```

def profile(request):

    apps = Application.objects.filter(user=request.user)
    subms = Submissions.objects.filter(application__user=request.user)
    usub = Submissions.objects.filter(user=request.user)
    print(subms)

    return render(request, 'profile.html', {'name' : request.user.username,
                                             'first_name':request.user.first_name,
                                             'last_name': request.user.last_name,
                                             'is_actor': request.user.is_actor,
                                             'phone' : request.user.phone,
                                             'is_cm': request.user.is_cm,
                                             'data' : apps,
                                             'subm':subms,
                                             'usub':usub,})

```

Рисунок 18 – Структура функций в Django

После написания всех необходимых функций необходимо написать все пути, на которых они будут выполняться. Для этого в Django существует подсистема URLов, которая привязывает нужную функцию к определенной странице. Структура записи в этой подсистеме показана на рисунке 19.

```

path('home/', views.home),
path('login/', views.loginPage),
path('login/actor/', actor.ActorSignUpView.as_view(), name='actor_signup'),
path('login/cm/', cm.CmSignUpView.as_view(), name='cm_signup'),
path('loginin/', views.CustomLoginView.as_view(), name='loginin'),
path('logout/', views.logout, name='logout'),
path('profile/', views.profile),
path('applications/', views.getAll),
path('manage/', views.ApplicationView.as_view(), name='manage'),
re_path(r'^request/get/app=(?P<application_id>\d+)\$', views.SubmissionView.as_view()),]

```

Рисунок 19 – Структура подсистемы путей в Django

В ходе разработки было принято решение развертывания системы на хостинге. Так как проект небольшой и не требует больших вычислительных ресурсов был использован облачный сервис Нероку, он позволяет бесплатно разместить информационную систему в интернете.

Для взаимодействия с проектом используется терминал операционной системы и репозиторий информационной системы, загруженный на Git-подобный сервис, в нашем случае Github.

3.3 Подробности реализации

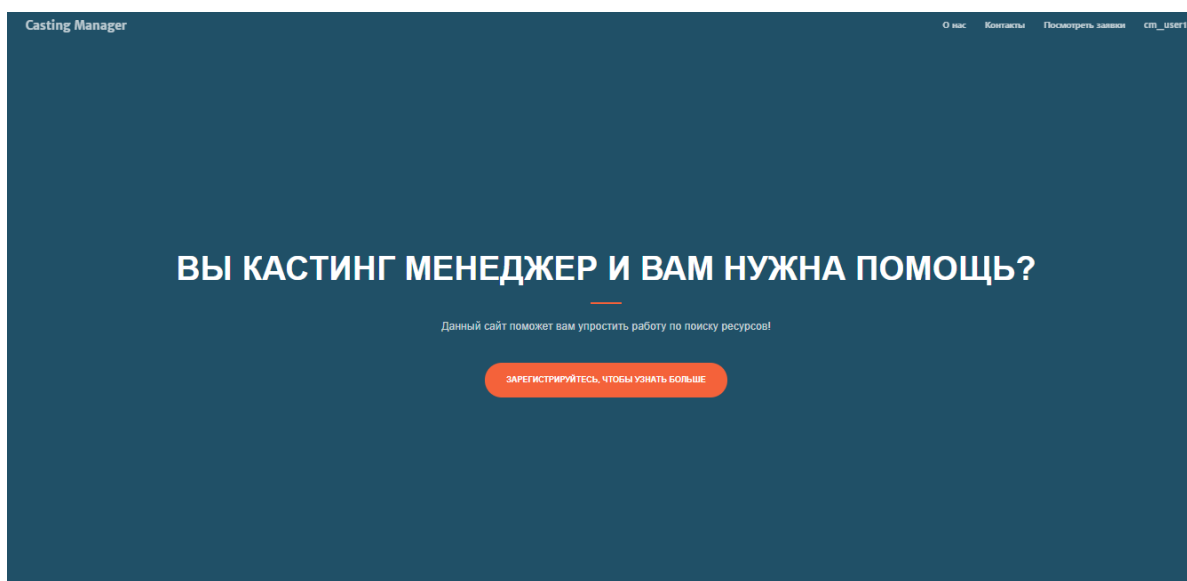


Рисунок 20 – Главная страница веб-сервиса

- регистрация пользователей;

Регистрация пользователей реализована стандартными методами фреймворка Django с разделением доступа как на уровне моделей, так и на уровне шаблонов (html-страниц). На уровне шаблонов происходит наследование от встроенного класса user, это позволяет создать множество других ролей в системе, например, кастинг-менеджер или обычный пользователь, как реализовано в данном случае.

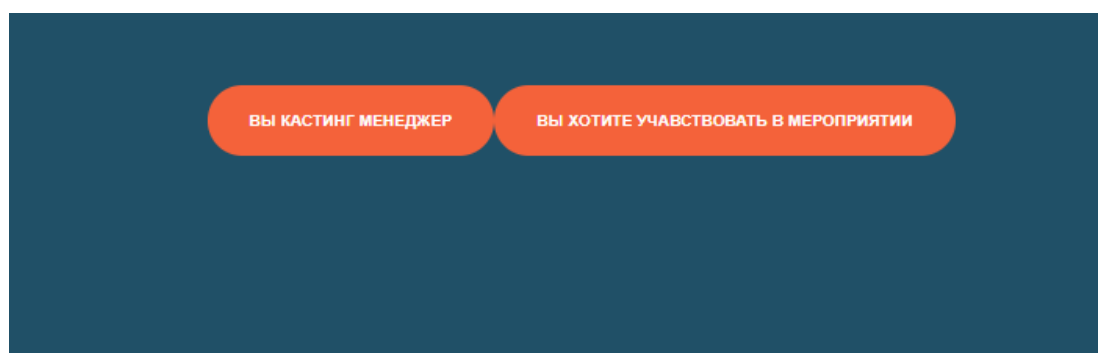
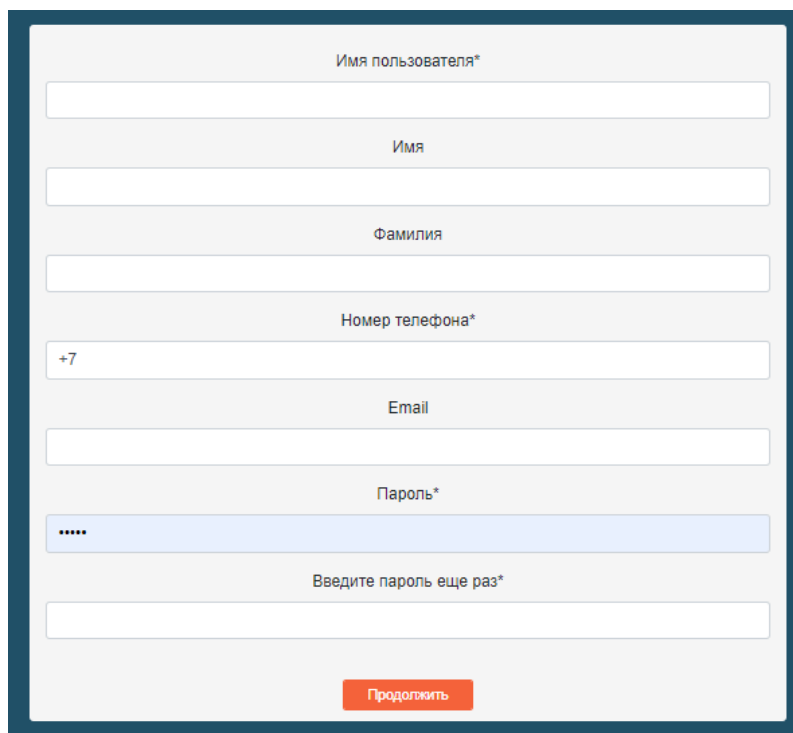


Рисунок 21 – Выбор вида пользователя

Форма для регистрации реализована с помощью встроенных в фреймворк html-форм. Это позволяет гибко настроить поля формы, изменить стили и подписи.



The image shows a registration form with the following fields and labels:

- Имя пользователя* (Username*)
- Имя (Name)
- Фамилия (Surname)
- Номер телефона* (Phone number*)
- Email
- Пароль* (Password*)
- Введите пароль еще раз* (Re-enter password*)

At the bottom of the form is a red button labeled "Продолжить" (Continue).

Рисунок 22 – Регистрация пользователей

- размещение заявки;

Кастинг-менеджер может размещать заявки о поиске актеров, массовки, необходимых ресурсов. Для этого кастинг-менеджером заполняется специальная форма. При отправке формы происходит вызов функции, которая обрабатывает данные, написанные пользователем. Далее эти же данные отправляются в базу данных.

The screenshot shows the 'Casting Manager' interface. At the top, there is a dark blue header with the title 'Casting Manager' and navigation links: 'О нас', 'Контакты', 'Посмотреть заявки', 'Управление', and 'cm_test'. Below the header, the form is divided into three sections. The first section, 'Название заявки*', contains a text input field with the value 'Снятие клипа для песни группы Кофе - Ставлю на Зеро'. The second section, 'Описание мероприятия и требования*', contains a text area with the text '1. Массовка' and '2. 13 тысяч рублей'. The third section, 'Дата проведения мероприятия', contains a date input field with the value '2019-06-28'. At the bottom center of the form is a red button labeled 'Продолжить'.

Рисунок 23 – Форма создания заявки

После создания заявки кастинг-менеджер может посмотреть вид, только что созданной заявки.

The screenshot shows the 'Casting Manager' profile page. At the top, there is a dark blue header with the title 'Casting Manager' and navigation links: 'О нас', 'Контакты', 'Посмотреть заявки', 'Управление', and 'cm_test'. Below the header, there is a breadcrumb trail: 'Профиль / Просмотр заявки'. The main content area displays the details of a submitted casting request. The title is 'Название - Снятие клипа для песни группы Кофе - Ставлю на Зеро'. The description is 'Описание - 1. Массовка 2. 13 тысяч рублей'. Below the description is a button labeled 'Получить отчет'. At the bottom center of the page is the word 'Ответы'.

Рисунок 24 – Внешний вид заявки кастинг-менеджера

В профиле кастинг-менеджера отображаются созданные заявки.

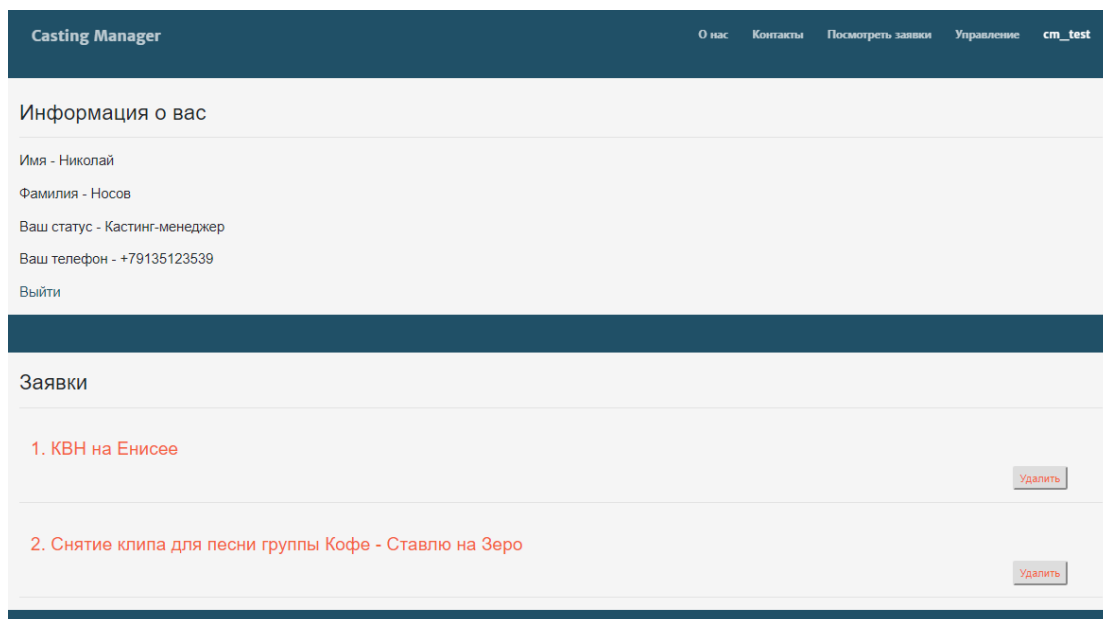


Рисунок 25 – Профиль кастинг-менеджера

- отклик на заявку;

Данные, отправленные кастинг-менеджером в базу данных отображаются на странице «все заявки». На рисунке 26 показана таблица базы данных, куда записываются заявки.

name	desc	user_id	date
1 КВН на Енисее	Требуются: реквизит, актёры, день...	9	2019-06-28
2 Снятие клипа для песни группы Кофе - ...	1. Массовка 2. 13 тысяч рублей	9	2019-06-28

Рисунок 26 – Таблица заявок в базе данных

Обычный пользователь может откликнуться на заявку, для этого пользователь нажимает кнопку «откликнуться», выбирает тип отклика, и оставляет свои контактные данные, в комментариях может указать время, в которое он свободен или в которое ему необходимо позвонить.

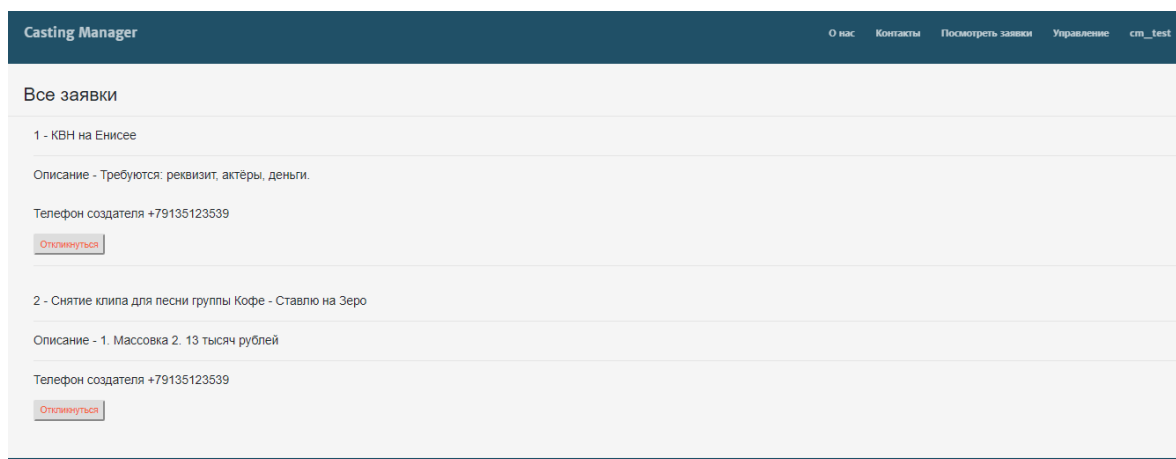


Рисунок 27 – Просмотр заявок обычным пользователем

При отклике на заявку пользователь выбирает типа отклика (реквизит, актер, другое), тем самым указывая, чем он может помочь данному объявлению.

The screenshot shows a form for responding to a casting request. The form is titled 'Тип отклика*' and contains a dropdown menu with three options: 'Реквизит', 'Актёр', and 'Другое'. Below the dropdown is a text input field labeled 'Что вы можете предложить?*', which is currently empty. At the bottom of the form is a red button labeled 'Продолжить'.

Рисунок 28 – Отклик на заявку

В профиле обычного пользователя отражаются все заявки, на которые он оставлял отклики. Есть кнопка удалить, которая удаляет оставленный отклик из базы данных. Пока кастинг-менеджер не подтвердил отклик, стоит статус «на рассмотрении».



Рисунок 29 – Профиль обычного пользователя

Кастинг-менеджер, просматривая заявку, может увидеть все отклики, оставленные пользователями. Отклик можно принять или отклонить.

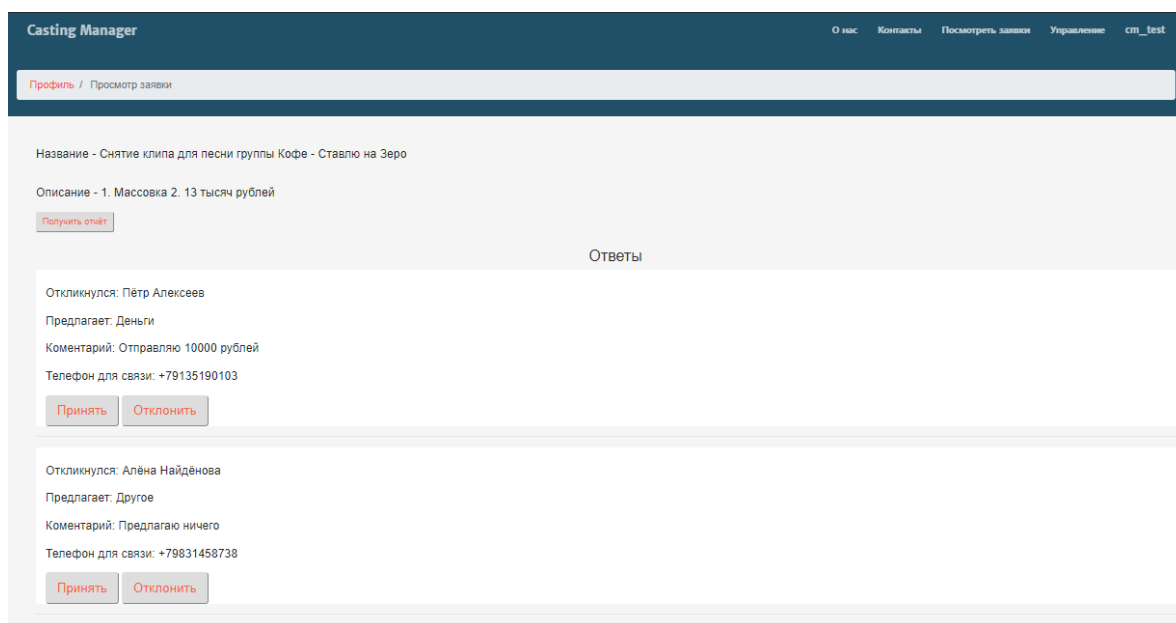


Рисунок 30 – Отклики на заявку

При нажатии кастинг-менеджером кнопки «принять» на почту обычного пользователя приходит уведомление, показанное на рисунке 31. В случае

отклонения заявки на почту также приходит уведомление о том, что отклик отклонен.

Участие в мероприятии



no.reply.castingmanagement@gmail.com

no.reply.castingmanagement@gmail.com

Вам: siberiandrunk@yandex.ru

сегодня в 1:18

Спасибо за интерес, Пётр
Вы приняты для участия в мероприятии - Снятие клипа для песни группы Кофе - Ставлю на Зеро

С уважением,

команда КМ

Телефон для интересующих вопросов: +79135123539

Рисунок 31 – Письмо-подтверждение на почте обычного пользователя

Формирование писем осуществляется с помощью встроенных в Django методов `send_mail()`. В настройках проекта происходит обозначение SMTP-сервера, который будет принимать запросы и отправлять их на почтовые ящики пользователей, почтовый адрес хоста и пароль хоста. Для корректного использования почтового ящика, нужно разрешить использование его в приложениях, это можно сделать на странице поставщика услуг электронной почты. Здесь используется SMTP-сервер компании Google и их почтовых ящиков Gmail.

```
EMAIL_BACKEND = 'django.core.mail.backends.smtp.EmailBackend'  
EMAIL_HOST = 'smtp.gmail.com'  
EMAIL_USE_TLS = True  
EMAIL_PORT = 587  
EMAIL_HOST_USER = 'no.reply.castingmanagement@gmail.com'
```

Рисунок 32 – Конфигурация e-mail-параметров

Кастинг-менеджер может сформировать отчет, куда будут включены все принятые пользователи с указанием необходимой информации. Для этого, кастинг-менеджер на заявке нажимает кнопку «получить отчет». На рисунке 33 показан интерфейс, на котором находится данная кнопка.

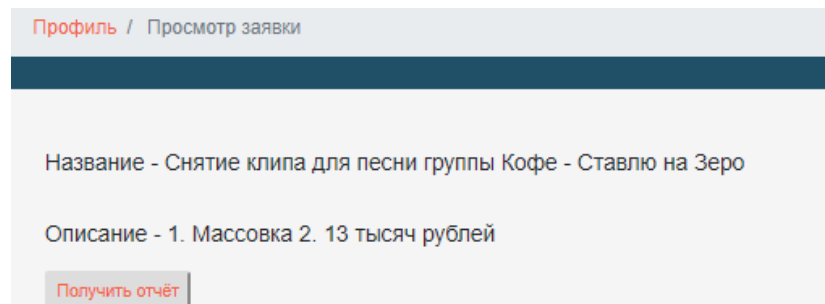


Рисунок 33 – Вид заявки кастинг-менеджера

Сформированный отчет показан на рисунке 34.

Отправитель	Телефон	Вид услуг	Комментарий	Статус
Пётр Алексеев	+79135190103	Деньги	Отправляю 10000 рублей	Подтвержден
Алёна Найдёнова	+79831458738	Другое	Предлагаю ничего	На рассмотрении

Рисунок 34 – Отчет по мероприятию

Формирование отчёта происходит с помощью библиотеки для языка Python, под названием xhtml2pdf. Данная библиотека переводит код языка разметки HTML, в PDF-файлы. Страница размечается исходя из целей, в данном случае необходимо запросить из базы данных все отклики к заявке. Проблема данной библиотеки (так и её аналогов) заключается в неправильной перекодировке кириллических символов, поэтом для корректного отображения нужно указать на использование кириллических шрифтов в стилях HTML-страницы.

С помощью класса Render производится перевод страницы в PDF-файл, основными методами является BytesIO и pisa.pisaDocument. На рисунке 35 показан класс Render.

```
class Render:
    @staticmethod
    def render(path: str, params: dict):
        template = get_template(path)
        html = template.render(params)
        response = BytesIO()
        pdf = pisa.pisaDocument(html, response, link_callback=link_callback, encoding='utf-8')

        if not pdf.err:
            response = HttpResponse(response.getvalue(),
                                    content_type='application/pdf')

            response['Content-Disposition'] = 'attachment; filename=summary.pdf'
            return response
        else:
            return HttpResponse("Error Rendering PDF", status=400)
```

Рисунок 35 – Класс Render

3.4 Анализ полученных результатов

В ходе работы был разработан веб-сервис для того, чтобы автоматизировать работу кастинг-менеджера и повысить эффективность его деятельности.

Эффективность работы кастинг-менеджера повышается за счет сокращения времени, затраченного на выполнение работы. Для подтверждения эффективности рассмотрим ход действий кастинг-менеджера с сервисом и без.

Ход действий кастинг-менеджера без веб-сервиса:

- размещение объявления в социальных сетях;
- ожидание отклика;
- просмотр откликов;
- вручную составление списка людей, которые могут помочь любыми ресурсами;

- вручную оповещение людей о времени съемки, мероприятия.



Рисунок 36 – Работа кастинг-менеджера без веб-сервиса

Кастинг-менеджер без веб-сервиса тратит большое количество времени на то, чтобы оповестить людей.

Рассмотрим ход действий кастинг-менеджера с использованием веб-сервиса:

- размещение заявки на веб-сервисе;
- ожидание отклика;
- просмотр откликов;
- автоматическое оповещение людей, которые «подходят»;
- автоматическое составление списка людей.



Рисунок 37 – Работа кастинг-менеджера с веб-сервисом

Кастинг-менеджер с веб-сервисом не тратит время на оповещение людей, так как это происходит автоматически при нажатии кнопки «принять» и «получить отчет».

Благодаря разработанному веб-сервису, оповещение и составление списка происходит автоматически, кастинг-менеджер не тратит на это большое

количество времени, и, следовательно, экономит время и может выполнять за это же время больше работы. Эффективность деятельности кастинг-менеджера повышается, так как сокращается время на выполнение работы.

ЗАКЛЮЧЕНИЕ

В результате бакалаврской работы была спроектирована информационная система, которая автоматизирует деятельность кастинг-менеджера. Благодаря разработанной системе, были автоматизированы процессы, которые ранее выполнялись вручную и отнимали много времени, отчего эффективность работы уменьшалась.

Была рассмотрена предметная область профессии кастинг-менеджера, выявлены основные проблемы, выбраны и обоснованы основные характеристики информационной системы, также были сформированы основные процессы, нуждающиеся в автоматизации.

Были поставлены основные требования к системе, разработаны диаграммы бизнес-процессов.

Разработанная система имеет простой функционал, дружелюбность интерфейса, адаптивность, отказоустойчивость.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Разработка информационных систем. Ч. 1. Структурные методы : учеб. пособие для студентов направления «Информационные системы» / Ю. А. Маглинец. – Красноярск : Изд-во «Кларетианум», 2004. – 124 с.
2. Пособие. Распределенные системы. Архитектура клиент-сервер [Электронный ресурс] // Сообщество «Moodle», 2013. – Режим доступа: <http://moodle.vsu.ru>
3. Александров Д.В. Инструментальные средства информационного менеджмента. CASE-технологии и распределенные информационные системы: книга // Д. В. Александров. – Москва: «Финансы и статистика», 2011. –224 с
4. Информационные системы: учеб. пособие / О. Л. Голицына, Н. В. Максимов, И. И. Попов. – 2-е изд. – Москва: ФОРУМ: ИНФРА-М, 2014. – 448 с.: ил – (Высшее образование).
5. CASE-технологии - Практикум: книга // Федотова Д.Э., Семенов Ю.Д., Чижик К.Н. – Москва: Изд-во «Горячая Линия - Телеком», 2005. –165 с.
6. Александров Д.В. Инструментальные средства информационного менеджмента. CASE-технологии и распределенные информационные системы: книга // Д. В. Александров. – Москва: «Финансы и статистика», 2011. –224 с.
7. Разработка и эксплуатация автоматизированных информационных систем : учеб. пособие / Л. Г. Гагарина. – Москва: ИД «ФОРУМ»: ИНФРА-М, 2013. – 384 с.: ил – (Профессиональное образование).
8. Бердышев С.Н. Искусство оформления сайта. 2-е изд. [Электронный ресурс]: практическое пособие/ Бердышев С.Н.— Электрон. текстовые данные.— Москва: Дашков и К, Ай Пи Эр Медиа, 2012.— 101 с.
9. Лутц, М.Python.Карманный справочник: учебное пособие, 5-е изд. / М. Лутц. –Москва: ООО «И.Д.Вильямс»,2015. –320с.
10. Документация Django[Электронный ресурс]. —Режим доступа: <http://djbook.ru/rel1.7>

11. Уроки по Bootstrap[Электронный ресурс]. —Режим доступа: <http://dedushka.org/uroki/6901.html>
12. Дронов В.С., Django: Практика создания Web-сайтов на Python – БХВ-Петербург, 2016. –528 с.
13. Полонская, Е.Л. Язык HTML. Самоучитель. / Е.Л. Полонская – Москва: Издательский дом «Вильяме», 2005. – 320 с.
14. Жданов С.А. Информационные системы. Учебник для ВУЗов: книга /С. А. Жданов, М. Л. Соболева, А. С. Алфимова. – Санкт-Петербург: ООО «Прометей», 2015 –302 с.
15. СТО 4.2-07-2014 Система менеджмента качества. Общие требования к построению, изложению и оформлению документов учебной деятельности. – Введ. 30.12.2013. – Красноярск: СФУ, 2014. – 60 с.

