

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Некоммерческое акционерное общество  
АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ

кафедра Компьютерных технологий

«Допущен к защите»  
Заведующий кафедрой \_\_\_\_\_

(Ф.И.О., ученая степень, звание)

« \_\_\_\_\_ » \_\_\_\_\_ 20\_\_ г.  
(подпись)

ДИПЛОМНЫЙ ПРОЕКТ

На тему: Разработка поисково-информационной системы "Талада"

Специальность Вычислительная техника и программное обеспечение

Выполнил (а) Ахунбаев К. С. ВТ-12-3  
(Фамилия и инициалы) группа

Научный руководитель Тургандиев Э. С. к. ф. м. н.  
(Фамилия и инициалы, ученая степень, звание)

Консультанты:

по экономической части:

Велицкая А. Ч. к. э. н., доцент  
(Фамилия и инициалы, ученая степень, звание)  
« 02 » 06 2016 г.  
(подпись)

по безопасности жизнедеятельности:

Брихобабко Н. Г. д. х. н., профессор  
(Фамилия и инициалы, ученая степень, звание)  
« 23 » 05 2016 г.  
(подпись)

по применению вычислительной техники:

Тургандиев Э. С. к. ф. м. н.  
(Фамилия и инициалы, ученая степень, звание)  
« 02 » 06 2016 г.  
(подпись)

Нормоконтролер: Тургандиев Э. С. к. ф. м. н.  
(Фамилия и инициалы, ученая степень, звание)  
« 02 » 06 2016 г.  
(подпись)

Рецензент: Тургандиев Э. С.  
(Фамилия и инициалы, ученая степень, звание)  
« 02 » 06 2016 г.  
(подпись)

Алматы 2016 г.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Некоммерческое акционерное общество  
АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ

Факультет Аэрокосмических и Информационных технологий  
Специальность Вычислительная техника и программное обеспечение  
Кафедра Компьютерных технологий

ЗАДАНИЕ

на выполнение дипломного проекта

Студент Алишбаев Жанжан Серикабайұлы  
(фамилия, имя, отчество)

Тема проекта Разработка лансково-информационной системы  
"Тамара"

утверждена приказом ректора № 148 от «19» октября 2015 г.

Срок сдачи законченной работы «\_\_» \_\_\_\_\_ 20\_\_ г.

Исходные данные к проекту требуемые параметры результатов проектирования (исследования) и исходные данные объекта

проектирование и разработка системы с возможностью  
организации взаимодействия между потребителем и  
поставщиком услуг

Перечень подлежащих разработке дипломного проекта вопросов или краткое содержание дипломного проекта:

Использование новейших технологий уменьшающих  
нагрузку на сервер

Перечень графического материала (с точным указанием обязательных чертежей)

Blank lines for listing graphical material.

Рекомендуемая основная литература

1. Htmlbook.ru
2. Yiiframework.ru

Blank lines for additional recommended literature.

Консультанты по проекту с указанием относящихся к ним разделов

Раздел	Консультант	Сроки	Подпись
БД/СД	Будимов И.Г.	18.05 - 23.05.16	<i>[Signature]</i>
Эксперт. часть	Беленкова А.Ч.	24.04 - 02.06.16	<i>[Signature]</i>
Нормоконтроль	Тургалиев Э.С.		<i>[Signature]</i>



## **Аңдатпа**

Осы дипломдық жобаның мақсаты іс-шараларды ұйымдастыру бойынша қызметтер туралы ақпарат тарату және шығарып алу үшін бірыңғай тұғырнама жасайды «Tamada.kz», іздеу және ақпараттық жүйесін (веб-портал) құру болып табылады.

Іс-шара ұйымдастырушылары мен қызмет көрсетушілер жұмысын жеңілдету мақсатында әзірленген интерфейс және деректер базасын «Асаба».

Интерфейс, ол веб интерфейсін барлық заманауи талаптарға сай, жаңа технологияларды пайдалана отырып, әзірленген.

Тек техникалық-экономикалық негіздеменің және есептелген жоба әзірлеу құны мен саналады денсаулық және қауіпсіздік мәселелерімен соңғы тарауда мәмілелерге.

## **Аннотация**

Целью данного дипломного проекта является создание поисково-информационной системы (веб-портала) «Tamada.kz», которые создадут единую платформу для размещения и поиска информации по услугам в сфере организации мероприятий.

Разработан интерфейс и база данных «Тамада» с целью упрощения работы организаторов мероприятий и поставщиков услуг.

Интерфейс разработан с использованием новейших технологий, отвечает всем современным требованиям к веб-интерфейсу.

Так же в последних главах рассматривается технико-экономическое обоснование и рассчитывается цена разработки проекта и рассматриваются вопросы безопасности жизнедеятельности.

## **Annotation**

The aim of this diploma project is the creation of search and information system (web portal) «Tamada.kz», which will create a single platform for the distribution and retrieval of information on services in the organization of events.

Developed interface and database "Master of ceremonies" in order to facilitate the work of event organizers and service providers.

The interface is designed using the latest technology, it meets all modern requirements to the web interface.

Just in the last chapter deals with the feasibility study and the calculated project development cost and considered health and safety issues.

## Содержание

Введение.....	12
1 Теоретическая часть.....	13
1.1 Основные требования к программным решениям, программным средствам и аппаратному обеспечению.....	13
1.1.1 Общие требования.....	13
1.1.2 Требования к Backend части .....	13
1.1.3 Требования к серверу.....	14
1.2 Структура веб-портала и функциональные требования к нему.....	14
1.2.1 Анализ аналогичных приложений.....	14
1.2.2 Структура .....	14
1.2.3 UML диаграммы.....	16
1.3 Проектирование базы данных.....	18
1.3.1 Определение атрибутов всех сущностей .....	18
1.4 Используемые технологии программирования.....	39
1.4.1 СУБД MySql.....	39
1.4.2 PHP.....	40
1.4.2.1 Традиционность.....	40
1.4.2.2 Простота.....	41
1.4.2.3 Безопасность .....	41
1.4.2.4 Гибкость .....	42
1.4.2.5 Бесплатное распространение .....	43
1.4.3 Yii.....	43
1.4.3.1 Требования.....	43
1.4.3.2 Сравнение.....	44
1.4.4 SphinxSearch.....	44
2 Расчетная часть.....	45
2.1 Расчет места для хранения базы данных .....	45
2.2 Меры по обеспечению безопасности .....	56
2.2.1 Предотвращение межсайтового скриптинга .....	56
2.2.2 Предотвращение подделки межсайтовых запросов .....	56
2.2.3 Предотвращение атак через cookie.....	57
3 Разработка поисково-информационной системы .....	57
3.1 Этапы разработки.....	57

3.2 Разработка интерфейса «Tamada.kz» .....	59
4 Техничко-экономическая часть .....	71
4.1 Цель проекта .....	71
4.2 Трудовые ресурсы, используемы в работе .....	71
4.3 Оборудование, использованное в работе.....	72
4.4 Расчет затрат и стоимости работ по реализации проекта .....	72
4.5.1 Расчет фонда оплаты труда .....	73
4.5.2 Расчет затрат по социальному налогу.....	75
4.5.3 Расчет амортизационных отчислений.....	76
4.5.4 Расчет затрат на электроэнергию .....	76
4.5.5 Расчет накладных расходов .....	77
4.5.6 Суммарные затраты на реализацию проекта.....	77
4.5.7 Цена реализации проекта .....	78
5 Безопасность жизнедеятельности.....	79
5.1 Анализ условий труда.....	79
5.2 Метеорологические условия, вентиляция и кондиционирование воздуха.....	81
5.3 Расчет уровня шума .....	81
5.4 Расчет вентиляции офиса .....	82
Заключение .....	85
Список используемой литературы .....	86
Приложение А .....	87

## **Введение**

В нашей жизни так часто бывают важные события – день рождения, свадьба, юбилей, детский праздник, корпоративное мероприятие. Всем хочется, чтобы торжество было незабываемым и особенным.

Например, свадьба. Это важное событие в жизни, как жениха с невестой, так и их родителей. Очень много сил, нервов и конечно, средств, тратится на организацию праздника. Свадьба бывает не каждый день, и, если пытаться организовать торжество самостоятельно, ошибки будут неизбежны.

Для того, чтобы облегчить приготовления к столь важному событию в вашей жизни, разработан веб-портал «Tamada.kz». Система поможет найти и выбрать необходимые услуги или товары, в одном месте. Присутствует возможность ознакомиться со всеми объявлениями, и четко определив параметры, найти и выбрать, именно то, что нужно.

В связи с большим количеством запросов, одновременно, к системе, интегрирована система полнотекстового поиска Sphinx. Данное решение, позволило значительно уменьшить время обработки и выполнения запросов, также стабилизировало работу сервера базы данных при увеличивающейся нагрузке.



## **1 Теоретическая часть**

### **1.1 Основные требования к программным решениям, программным средствам и аппаратному обеспечению**

#### **1.1.1 Общие требования**

Программное обеспечение (ПО) должно разрабатываться по модульному принципу и представлять собой набор функционально законченных подпрограмм.

Настоятельно рекомендуется использовать только свободное программное обеспечение.

При настоятельной потребности использовать платные программные продукты необходимо согласовать с заказчиком их применение. Не допускается использовать в составе ПО нелегальные копии или нелицензированные модификации программных продуктов.

Особое внимание следует уделять вопросам патентной чистоты и соблюдения требований Copyright.

#### **1.1.2 Требования к Backend части**

Все установочные настройки допускается осуществлять с помощью конфигурационных файлов.

Управление параметрами системы, аккаунтами пользователей, объектами и прочими характеристиками будет осуществляться через визуализированную административную панель.

Для реализации данной системы с обеспечением дальнейшего ее развития, расширения функциональных возможностей и необходимым запасом производительности необходимо использовать хорошо структурированную и апробированную базу данных (СУБД). Данным требованиям соответствует свободная реляционная СУБД MySQL с открытым исходным кодом, распространяющаяся под лицензиями GNU GPL.

Технология и архитектура построения ПО должны обеспечивать высокую производительность и длительный срок службы, гарантирующий эффективное использование накапливаемых данных. ПО будет реализовано по клиент-серверной технологии, с помощью языков программирования PHP, HTML, JavaScript. Требования к программным средствам и аппаратному обеспечению.

### **1.1.3 Требования к серверу**

Программное средство должно работать под управлением операционной системы семейства Linux, предпочтительной ОС является Debian.

Минимально необходимыми для работы программного средства являются следующие параметры оборудования:

2-х ядерный процессор типа Intel® Core 2 Duo™ или AMD® Phenom™ с тактовой частотой не меньше 2ГГц и 2ГБ ОЗУ.

Рекомендуемыми для работы программного средства являются следующие параметры оборудования:

4-х ядерный процессор типа Intel® Xeon™ или 8-и ядерный AMD® Opteron™ с тактовой частотой не меньше 2ГГц и 4ГБ ОЗУ. Рекомендуемая дисковая подсистема – 4 7200rpm Enterprise Sata Drive, объемом от 300 ГБ, RAID5 или RAID10.

## **1.2 Структура веб-портала и функциональные требования к нему**

### **1.2.1 Анализ аналогичных приложений**

В процессе разработки веб-портала был проведен сравнительный анализ по нескольким сайтам. Анализ показал, отсутствие поиска по категориям, фильтров по объявлениям и регионам, также онлан-планировщика.

### **1.2.2 Структура**

В системе будут пользователи 4 типов:

- Администратор;
- Модератор;
- Рекламодатель;
- Пользователь.

Возможности данных типов пользователей:

- Администратор системы обладает полным доступом ко всем модулям системы.
- Модератор системы имеет доступ к объявлениям, новостям на сайте и модулю баннерной сети, также проводит работу по учёту рекламодателей и платежей;
- Рекламодатель имеет право размещать, удалять и редактировать объявления в каталоге услуг.
- Пользователь может просматривать объявления в каталоге услуг, читать и комментировать статьи в разделе «Полезная информация»,

оставлять отзывы о объявлениях, пользоваться онлайн калькулятором расходов и клиентским мобильным приложением.

- В структуру веб-портала войдут следующие разделы:
- Каталог объявлений — перечень услуг и товаров, разделенных по категориям;
- Реклама — система баннерной сети, подсветки объявлений и «горячих» объявлений;
- Отзывы — система рейтинга рекламодателей с возможностью оценки качества по нескольким параметрам;
- Полезная информация — информационные статьи в различных рубриках с возможностью комментирования;
- Пользователи — система регистрации и авторизации пользователей через социальные сети;
- Платежи — система проведения оплат через банковские карты и сервис «Киви»;
- Личный кабинет рекламодателя — возможность создания анкеты и портфолио;
- Личный кабинет пользователя — кабинет пользователя с сохранением рабочих данных приложения, истории просмотров и избранным.
- Чат-система поддержки - брендированный чат для поддержки пользователей.
- В структуру клиентского мобильного приложения войдут следующие разделы:
- Каталог объявлений — перечень услуг и товаров, разделенных по категориям;
- Реклама — уведомления об акциях и «горячих» объявлениях;
- Онлайн калькулятор бюджета — инструмент для расчета затрат для организации мероприятия, в том числе: выбор места проведения мероприятия, выбор слуг и товаров из каталога, список гостей, список рассадки гостей, блокнот пользователя;

Услуги будут различаться по следующим категориям:

- «Свадьбы»
- «Юбилеи»
- «Корпоративные вечера»
- «Детские праздники»
- «Другие торжества»
- Полный список категорий должен быть предоставлен Заказчиком.

- Каталог услуг:
- Тамада, ведущие
- Фотография
- Видеосъемка
- Агентства
- Тимбилдинг
- Шоу-программа (танцоры, артисты разных жанров)
- Музыкальные коллективы
- Аниматоры для детей
- Оформление зала
- Банкетные залы
- Свадебные салоны
- Салоны красоты
- Стилисты, визажисты, парикмахеры
- Кондитерские
- Ди-джей
- Подарки
- Транспорт (свадебный картеж, лимузины и т.п.)

Все услуги будут дифференцированы по регионам страны. Регион выбирается во время подачи объявления, если пользователь не выбрал его раньше. На главной странице будет размещен фильтр по регионам.

### **1.2.3 UML диаграммы**

Для проектирования базы данных «Тамада» необходимо основные диаграммы UML:

- Диаграмма классов;
- Диаграмма прецедентов;

#### **Диаграмма классов**

Диаграмма классов, Static Structure diagram диаграмма, демонстрирующая классы системы, их атрибуты, методы и взаимосвязи между ними. Входит в UML. [1].

В данном случае демонстрируются основные связи между всеми классами поисково-информационной системы. Диаграмма классов на рисунке 1.2



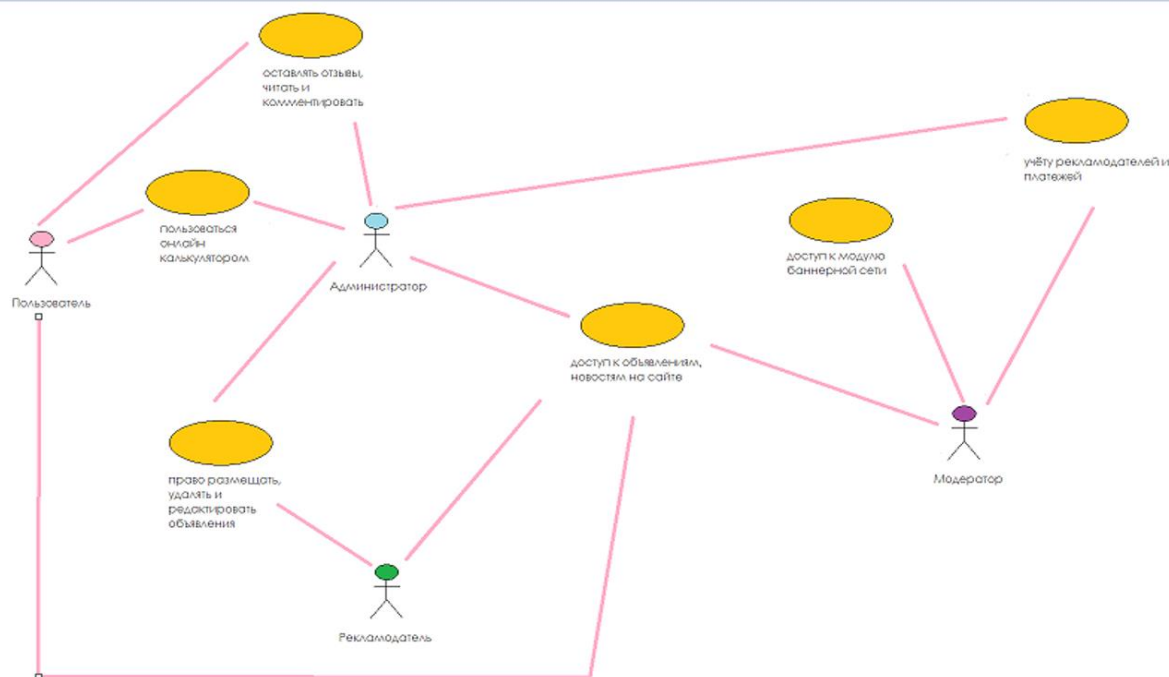


Рисунок 1.4 – Диаграмма прецедентов.

### 1.3 Проектирование базы данных

#### 1.3.1 Определение атрибутов всех сущностей

После проведения анализа, была разработана архитектура базы данных поисково-информационной системы. В таблицах 1.1 – 1.30 приведены ее сущности и их абстрактное представление.

Таблица 1.1 – Таблица статей

blog_article		
id		int(11)
title		varchar(300)
short_description		TEXT
full_text		TEXT
create_date		DATETIME
modified_date		DATETIME
published		BIT(1)
blog_article_cat_assoc статьи		
article_id		int (11)
category_id		int(11)

```

class Article extends CActiveRecord {

    /**
     * @return string the associated database table name
     */
    public function tableName() {
        return 'blog_article';
    }

    /**
     * @return array validation rules for model attributes.
     */
    public function rules() {
        // NOTE: you should only define rules for those attributes that
        // will receive user inputs.
        return array(
            array('title, full_text', 'required'),
            array('title', 'length', 'max' => 300),
            array('short_description, full_text, title, published', 'safe'),
            // The following rule is used by search().
            // @todo Please remove those attributes that should not be searched.
            array('id, title, short_description, full_text, create_date, modified_date', 'safe', 'on' => 'search'),
        );
    }
}

```

Рисунок 1.5 – Класс статей

Таблица 1.2 – Ассоциативная таблица статей и категорий

blog_article_cat_assoc		
article_id		int (11)
category_id		int(11)

```

public function relations() {
    // NOTE: you may need to adjust the relation name and the related
    // class name for the relations automatically generated below.
    return array(
        'articles' => array(self::MANY_MANY, 'Article', 'blog_article_cat_assoc(category_id, article_id)'),
    );
}

/**
 * @return array customized attribute labels (name=>label)
 */
public function attributeLabels() {
    return array(
        'id' => 'ID',
        'name' => 'Название',
    );
}

/**
 * Retrieves a list of models based on the current search/filter conditions.
 *
 * Typical usecase:
 * - Initialize the model fields with values from filter form.
 */

```

Рисунок 1.6 – Участок кода, связывающий категории и статьи

Используемая связь многие ко многим. Это позволяет статье принадлежать к нескольким категориям.

Таблица 1.3 – Таблица категорий

blog_category		
Id		INT (11)
Name		Varchar(255)

```

class Category extends CActiveRecord {

    /**
     * @return string the associated database table name
     */
    public function tableName() {
        return 'blog_category';
    }

    /**
     * @return array validation rules for model attributes.
     */
    public function rules() {
        // NOTE: you should only define rules for those attributes that
        // will receive user inputs.
        return array(
            array('name', 'length', 'max' => 255),
            // The following rule is used by search().
            // @todo Please remove those attributes that should not be searched.
            array('id, name', 'safe', 'on' => 'search'),
        );
    }
}

```

Рисунок 1.7 – Класс категорий

Таблица 1.4 – Таблица комментариев

blog_comment		
user_id		INT(11)
article_id		INT(11)
text		TEXT

```

class Comment extends CActiveRecord
{
    /**
     * @return string the associated database table name
     */
    public function tableName()
    {
        return 'blog_comment';
    }

    /**
     * @return array validation rules for model attributes.
     */
    public function rules()
    {
        // NOTE: you should only define rules for those attributes that
        // will receive user inputs.
        return array(
            array('text', 'required'),
            array('user_id, article_id', 'numerical', 'integerOnly'=>true),
            // The following rule is used by search().
            // @todo Please remove those attributes that should not be searched.
            array('id, user_id, article_id, text', 'safe', 'on'=>'search'),
        );
    }

    /**
     * @return array relational rules.
     */
}

```

Рисунок 1.8 – Класс комментариев



Таблица 1.5 – Таблица объявлений

catalog_advert		
Id		INT (11)
Title		Varchar(255)
text		TEXT
Company_id	company-advert-fk	Int(11)
Region_id	Region-advert-fk	Int(11)
Service_id	service-advert-fk	Int(11)
created_date		Date_time
edited_date		Date_time
Published		Int(1)
Is_disabled		Int(1)
price_tg		Decimal(8.2)
Price_usd		Decimal(8.2)
Price_eur		Decimal(8.2)
default_currency		TINYINT(4)
minServiceAmount		INT(11)
maxServiceAmount		INT(11)
coloredDate		DATETIME
toTopDate		DATETIME

```

/**
 * This is the model class for table "catalog_advert".
 *
 * The followings are the available columns in table 'catalog_advert':
 * @property integer $id
 * @property string $title
 * @property string $text
 * @property integer $company_id
 * @property integer $region_id
 * @property datetime $create_date
 * @property datetime $edited_date
 * @property integer $price
 * @property decimal $price_tg
 * @property decimal $price_usd
 * @property decimal $price_eur
 * @property int $default_currency
 */
class Advert extends CActiveRecord {

    public $minPrice;
    public $maxPrice;
    public $companyRating;

    protected $_price = null;

    const YES = 1;
    const NO = 0;

```

Рисунок 1.9 – Класс объявлений

Таблица 1.6 - Таблица ОТЗЫВОВ ОБЪЯВЛЕНИЙ

catalog_advert_feedback		
Id		INT(11)
Advert_id	advert-feedback-fk	Int(11)
user_id	user-feedcom-fk	Int(11)
Rate		DOUBLE(2.1)
Approved		BIT(1)
Text		TEXT
created_on		DATETIME
modified_on		DATETIME

```

class AdvertFeedback extends ActiveRecord {

    /**
     * @return string the associated database table name
     */
    public function tableName() {
        return 'catalog_advert_feedback';
    }

    /**
     * @return array validation rules for model attributes.
     */
    public function rules() {
        // NOTE: you should only define rules for those attributes that
        // will receive user inputs.
        return array(
            array('advert_id, user_id, approved', 'numerical', 'integerOnly' => true),
            array('rate', 'numerical'),
            array('text', 'safe'),
            // The following rule is used by search().
            // @todo Please remove those attributes that should not be searched.
            array('id, advert_id, user_id, rate, approved', 'safe', 'on' => 'search'),
        );
    }

    /**
     * @return array relational rules.
     */
}

```

Рисунок 1.10 – Класс отзывов объявлений

Таблица 1.7 – Таблица ответов на отзывы

catalog_advert_feedback_comment		
Id		INT(11)
feedback_id	feedback-comment-fk	Int(11)
Text		text
Created_on		DATETIME
Modified_on		DATETIME

```

class AdvertFeedbackComment extends CActiveRecord
{
    /**
     * @return string the associated database table name
     */
    public function tableName()
    {
        return 'catalog_advert_feedback_comment';
    }

    /**
     * @return array validation rules for model attributes.
     */
    public function rules()
    {
        // NOTE: you should only define rules for those attributes that
        // will receive user inputs.
        return array(
            array('text', 'required'),
            array('feedback_id', 'numerical', 'integerOnly'=>true),
            // The following rule is used by search().
            // @todo Please remove those attributes that should not be searched.
            array('id, feedback_id, text', 'safe', 'on'=>'search'),
        );
    }
}

```

Рисунок 1.11 – Класс ответов на отзывы

Таблица 1.8 – Таблица действий для продвижения объявлений

catalog_advert_promotions		
Id		INT(11)
Name		VARCHAR(255)
Description		TEXT
Label		VARCHAR(255)
Price		Int(11)
timeHours		Int(11)
Is_disabled		Int(1)
Test		Varchar(50)

```

class AdvertPromotion extends CActiveRecord
{
    /**
     * @return string the associated database table name
     */
    public function tableName()
    {
        return 'catalog_advert_promotions';
    }

    /**
     * @return array validation rules for model attributes.
     */
    public function rules()
    {
        // NOTE: you should only define rules for those attributes that
        // will receive user inputs.
        return array(
            array('name, label', 'required'),
            array('price, timeHours, is_disabled', 'numerical', 'integerOnly'=>true),
            array('name, label', 'length', 'max'=>255),
            array('description', 'safe'),
            // The following rule is used by search().
            // @todo Please remove those attributes that should not be searched.
            array('id, name, description, label, price, timeHours, is_disabled', 'safe', 'on'=>'search'),
        );
    }
}

```

Рисунок 1.12 – Класс действий для продвижения объявлений

Таблица 1.9 – Ассоциативная таблица пользователей и объявлений

catalog_advert_user		
user_id	advert-favourite-fk	Int(11)
advert_id	user-favourite-fk	Int(11)

```

public function relations() {
    // NOTE: you may need to adjust the relation name and the related
    // class name for the relations automatically generated below.
    return array(
        'feedback' => array(self::HAS_MANY, 'AdvertFeedback', 'advert_id'),
        'approvedFeedback' => array(self::HAS_MANY, 'AdvertFeedback', 'advert_id', 'on' => 'approved=1'),
        'service' => array(self::BELONGS_TO, 'Service', 'service_id'),
        'region' => array(self::BELONGS_TO, 'Region', 'region_id'),
        'company' => array(self::BELONGS_TO, 'Company', 'company_id'),
        'user' => array(self::BELONGS_TO, 'User', array('user_id' => 'id'), 'through' => 'company'),
        'userLiked' => array(self::MANY_MANY, 'User', 'catalog_advert_user(advert_id, user_id)')
    );
}

public function disable()
{
    if(!$this->is_disabled) {
        $this->is_disabled = 1;

        $this->save();
    }
}

public function enable()
{
    if($this->is_disabled) {
        $this->is_disabled = 0;
    }
}

```

Рисунок 1.13 – Участок кода реализующий связь между пользователем и объявлениями

Таблица 1.10 – Таблица компаний

catalog_company		
Id		Int(11)
name		VARCHAR(255)
User_id	user-company-fk	Int(11)
description		Text
created_date		DATETIME
edited_date		DATETIME
phone		VARCHAR(255)
billing		VARCHAR(255)
address		VARCHAR(255)
map		VARCHAR(255)
cash		Int(1)
cashless		Int(1)
cards		Int(1)
approved		Int(1)
is_disabled		Int(1)
Portfolio		TEXT
site_link		VARCHAR(255)
email		VARCHAR(255)

```

class Company extends ActiveRecord {

    public $rated;

    /**
     * @return string the associated database table name
     */
    public function tableName() {
        return 'catalog_company';
    }

    /**
     * @return array validation rules for model attributes.
     */
    public function rules() {
        // NOTE: you should only define rules for those attributes that
        // will receive user inputs.
        return array(
            array('name', 'required'),
            array('name', 'unique'),
            array('user_id', 'numerical', 'integerOnly' => true),
            array('name,email, site_link, description, portfolio, phone, address, map', 'safe'),
            array('name', 'length', 'max' => 255),
            // The following rule is used by search().
            // @todo Please remove those attributes that should not be searched.
            array('id, name, user_id', 'safe', 'on' => 'search'),
            array('cash, cashless, cards', 'boolean'),
        );
    }
}

```

Рисунок 1.14 – Класс компаний

Таблица 1.11 – Таблица мероприятий

catalog_event		
id		Int(11)
Name		Varchar(255)

```

class Event extends ActiveRecord {

  /**
   * @return string the associated database table name
   */
  public function tableName() {
    return 'catalog_event';
  }

  /**
   * @return array validation rules for model attributes.
   */
  public function rules() {
    // NOTE: you should only define rules for those attributes that
    // will receive user inputs.
    return array(
      array('name', 'length', 'max' => 255),
      array('name', 'unique'),
      // The following rule is used by search().
      // @todo Please remove those attributes that should not be searched.
      array('id, name', 'safe', 'on' => 'search'),
    );
  }

  public function behaviors() {
    return array('CAAdvancedArBehaviorPriority' => array(
      'class' => 'application.components.CAdvancedArBehaviorPriority'));
  }
}

```

Рисунок 1.15 – Класс мероприятий

Таблица 1.12 – Ассоциативная таблица мероприятий и типов услуг

catalog_event_service_assoc		
event_id		Int(11)
service_id		Int(11)
priority		SMALLINT(6)

```

class EventServiceAssoc extends ActiveRecord
{
  /**
   * @return string the associated database table name
   */
  public function tableName()
  {
    return 'catalog_event_service_assoc';
  }

  /**
   * @return array validation rules for model attributes.
   */
  public function rules()
  {
    // NOTE: you should only define rules for those attributes that
    // will receive user inputs.
    return array(
      array('event_id, service_id, priority', 'numerical', 'integerOnly'=>true),
      // The following rule is used by search().
      // @todo Please remove those attributes that should not be searched.
      array('event_id, service_id, priority', 'safe', 'on'=>'search'),
    );
  }

  /**
   * @return array relational rules.
   */
  public function relations()

```

Рисунок 1.16 – Класс реализующий связь между мероприятиями и типами услуг

Таблица 1.13 – Таблица типов услуг

catalog_service		
id		Int(11)
Name		Varchar(255)
disabled		Int(1)

```

class Service extends CActiveRecord {
    /**
     * @return string the associated database table name
     */
    public function tableName() {
        return 'catalog_service';
    }

    /**
     * @return array validation rules for model attributes.
     */
    public function rules() {
        // NOTE: you should only define rules for those attributes that
        // will receive user inputs.
        return array(
            array('name', 'length', 'max' => 255),
            array('name', 'unique'),
            array('name, disabled, imageUrl', 'safe'),
            // The following rule is used by search().
            // @todo Please remove those attributes that should not be searched.
            array('id, name, imageUrl', 'safe', 'on' => 'search'),
        );
    }

    public function behaviors() {
        return array('CAdvancedArBehaviorPriority' => array(
            'class' => 'application.components.CAdvancedArBehaviorPriority'),
            'avaBehavior' => array(
                'class' => 'ImageAttachmentBehavior',
            ),
        );
    }
}

```

Рисунок 1.17 – Класс типов услуг

Таблица 1.14 – Таблица стоимости основных валют

Currency		
Id		INT(11)
Name		VARCHAR(255)
Value		DECIMAL(8,2)
Updated_date		DATETIME

```

class Currency extends ActiveRecord
{
    /**
     * @return string the associated database table name
     */
    public function tableName()
    {
        return 'currency';
    }

    /**
     * @return array validation rules for model attributes.
     */
    public function rules()
    {
        // NOTE: you should only define rules for those attributes that
        // will receive user inputs.
        return array(
            array('name', 'length', 'max'=>255),
            array('value', 'length', 'max'=>8),
            array('update_date', 'safe'),
            // The following rule is used by search().
            // @todo Please remove those attributes that should not be searched.
            array('id, name, value, update_date', 'safe', 'on'=>'search'),
        );
    }

    /**
     * @return array relational rules.
     */
}

```

Рисунок 1.18 – Класс валют

Таблиц 1.15 – Таблица часто задаваемых вопросов

faq		
id		Int(11)
name		VARCHAR(255)
text		TEXT
category_id		INT(11)

```

class Faq extends ActiveRecord
{
    /**
     * @return string the associated database table name
     */
    public function tableName()
    {
        return 'faq';
    }

    /**
     * @return array validation rules for model attributes.
     */
    public function rules()
    {
        // NOTE: you should only define rules for those attributes that
        // will receive user inputs.
        return array(
            array('name, text, category_id', 'required'),
            array('category_id', 'numerical', 'integerOnly'=>true),
            array('name', 'length', 'max'=>50),
            // The following rule is used by search().
            // @todo Please remove those attributes that should not be searched.
            array('id, name, text, category_id', 'safe', 'on'=>'search'),
        );
    }

    /**

```

Рисунок 1.19 – Класс часто задаваемых вопросов



Таблица 1.16 – Таблица категорий часто задаваемых вопросов

faq_category		
id		Int(11)
Name		Varchar(255)

```

class FaqCategory extends ActiveRecord
{
    /**
     * @return string the associated database table name
     */
    public function tableName()
    {
        return 'faq_category';
    }

    /**
     * @return array validation rules for model attributes.
     */
    public function rules()
    {
        // NOTE: you should only define rules for those attributes that
        // will receive user inputs.
        return array(
            array('name', 'required'),
            array('name', 'length', 'max'=>50),
            ['name', 'unique'],
            // The following rule is used by search().
            // @todo Please remove those attributes that should not be searched.
            array('id, name', 'safe', 'on'=>'search'),
        );
    }

    /**
     * @return array relational rules.
     */
}

```

Рисунок 1.20 – Класс категорий часто задаваемых вопросов

Таблица 1.17 – Таблица файлов

File		
Id		Int(11)
title		Varchar(255)
link		Varchar(255)
upload_date		DATETIME
type	file_type_idx	TINYINT(1)
owner_id		INT(11)
mime	file_mime	Varchar(255)

```

class File extends CActiveRecord {

    private $_file = null;

    const TYPE_DEFAULT = 0;
    const TYPE_COMPANY = 1;
    const TYPE_USER = 2;
    const TYPE_AVATAR = 3;

    /**
     * @return string the associated database table name
     */
    public function tableName() {
        return 'file';
    }

    /**
     * @return array validation rules for model attributes.
     */
    public function rules() {
        // NOTE: you should only define rules for those attributes that
        // will receive user inputs.
        return array(
            array('type, owner_id', 'numerical', 'integerOnly' => true),
            array('title, link, mime', 'length', 'max' => 255),
            array('upload_date', 'safe'),
            // The following rule is used by search().
            // @todo Please remove those attributes that should not be searched.
            array('id, title, link, upload_date, type, owner_id, mime', 'safe', 'on' => 'search'),
        );
    }

    /**
     * @return array relational rules.
     */
    public function relations()
}

```

Рисунок 1.21 – Класс файлов

Таблица 1.18 – Таблица типовых страниц

info_pages страницы		
id		INT(11)
title		Varchar(255)
text		MEDIUMTEXT
alias		VARCHAR(50)

```

class InfoPage extends CActiveRecord
{
    /**
     * @return string the associated database table name
     */
    public function tableName()
    {
        return 'info_pages';
    }

    /**
     * @return array validation rules for model attributes.
     */
    public function rules()
    {
        // NOTE: you should only define rules for those attributes that
        // will receive user inputs.
        return array(
            array('title', 'length', 'max'=>255),
            array('title, alias, text', 'safe'),
            // The following rule is used by search().
            // @todo Please remove those attributes that should not be searched.
            array('id, title, text', 'safe', 'on'=>'search'),
        );
    }

    /**
     * @return array relational rules.
     */
    public function relations()
}

```

Рисунок 1.22 – Класс типовых страниц

Таблица 1.19 – Таблица авторизационных данных пользователя

Login		
id		Int(11)
identity		VARCHAR(255)
password		VARCHAR(255)
created_on		DATETIME
is_active		TINYINT(1)
is_disabled		TINYINT(1)
network		Varchar(255)
user_id		Int(11)
updated_on		DATETIME

```

class Login extends CActiveRecord
{
    /**
     * @return string the associated database table name
     */
    public function tableName()
    {
        return 'login';
    }

    /**
     * @return array validation rules for model attributes.
     */
    public function rules()
    {
        // NOTE: you should only define rules for those attributes that
        // will receive user inputs.
        return array(
            array('identity, is_active, is_disabled', 'filter', 'filter' => 'trim'),
            array('identity, password', 'required'),
            array('identity', 'unique'),
            array('is_active, is_disabled, user_id', 'numerical', 'integerOnly'=>true),
            array('identity, password, network', 'length', 'max'=>255),
            array('created_on', 'safe'),
            // The following rule is used by search().
            // @todo Please remove those attributes that should not be searched.
            array('id, identity, password, created_on, is_active, is_disabled, network, user_id', 'safe', 'on'=>'sea
        );
    }
}

```

Рисунок 1.23 – Класс авторизационных данных пользователя

Таблица 1.20 – Таблица истории платежей

payment_qiwi_history		
txn_id		Varchar(28)
user_id		INT(11)
dateTime		DATETIME
txn_date		DATETIME
Amount		Decimal(8,2)
Prv_txn		Varchar(20)

```

class PaymentQwiHistory extends CActiveRecord
{
    /**
     * @return string the associated database table name
     */
    public function tableName()
    {
        return 'payment_qwi_history';
    }

    /**
     * @return array validation rules for model attributes.
     */
    public function rules()
    {
        // NOTE: you should only define rules for those attributes that
        // will receive user inputs.
        return array(
            array('txn_id, user_id, dateTime, txn_date, amount, prv_txn', 'required'),
            array('txn_id, user_id', 'numerical', 'integerOnly'=>true),
            array('amount', 'length', 'max'=>8),
            // The following rule is used by search().
            // @todo Please remove those attributes that should not be searched.
            array('txn_id, user_id, dateTime, txn_date, amount', 'safe', 'on'=>'search'),
        );
    }

    /**
     * @return array relational rules.
     */
}

```

Рисунок 1.24 – Класс истории платежей

Таблица 1.21 – Таблица регионов

region	
id	Int(11)
name	Varchar(255)

```

class Region extends CActiveRecord
{
    /**
     * @return string the associated database table name
     */
    public function tableName()
    {
        return 'region';
    }

    /**
     * @return array validation rules for model attributes.
     */
    public function rules()
    {
        // NOTE: you should only define rules for those attributes that
        // will receive user inputs.
        return array(
            array('name', 'length', 'max'=>255),
            // The following rule is used by search().
            // @todo Please remove those attributes that should not be searched.
            array('id, name', 'safe', 'on'=>'search'),
        );
    }

    /**
     * @return array relational rules.
     */
    public function relations()
    {
    }
}

```

Рисунок 1.25 – Класс регионов

Таблица 1.22 – Таблица баннеров

reklama		
id		Int(11)
region_id		Int(11)
text		TEXT
zone		Int(11)
link		VARCHAR(255)
title		VARCHAR(255)
image		VARCHAR(255)
start_time		DATETIME
end_time		DATETIME
views_count		Int(11)
client_views		Int(11)
is_approved		Int(1)

```

class Reklama extends CActiveRecord
{
    /**
     * @return string the associated database table name
     */
    public function tableName()
    {
        return 'reklama';
    }

    /**
     * @return array validation rules for model attributes.
     */
    public function rules()
    {
        // NOTE: you should only define rules for those attributes that
        // will receive user inputs.
        return array(
            array('company_id, link, title, views_count, zone, client_views', 'required'),
            array('company_id, zone, client_views, region_id, views_count, is_approved', 'numerical', 'integerOnly'=
            array('link, title, image', 'length', 'max'=>255),
            array('image', 'file', 'types'=>'jpg, gif, png', 'allowEmpty'=>true),
            array('image', 'required', 'on'=>array('create')),
            // The following rule is used by search().
            // @todo Please remove those attributes that should not be searched.
            array('id, company_id, link, title, image, zone, client_vies, start_time, end_time, views_count, is_appr
        );
    }
}

```

Рисунок 1.26 – Класс баннеров

Таблица 1.23 – Таблица ценовых пакетов для показа рекламы

reklama_price		
id		Int(11)
price		Int(11)
sale		Int(11)
region_id		Int(11)

```

class ReklamaPrice extends CActiveRecord
{
    /**
     * @return string the associated database table name
     */
    public function tableName()
    {
        return 'reklama_price';
    }

    /**
     * @return array validation rules for model attributes.
     */
    public function rules()
    {
        // NOTE: you should only define rules for those attributes that
        // will receive user inputs.
        return array(
            array('price, sale, region_id', 'required'),
            array('price, sale', 'numerical', 'integerOnly'=>true),
            // The following rule is used by search().
            // @todo Please remove those attributes that should not be searched.
            array('id, price, sale', 'safe', 'on'=>'search'),
        );
    }

    /**
     * @return array relational rules.
     */
    public function relations()

```

Рисунок 1.27 – Класс ценовых пакетов для показа рекламы

Таблица 1.24 – Таблица учета миграций

tbl_migration	
version	VARCHAR(255)
apply_time	TIMESTAMP

```

abstract class CDbMigration extends CComponent
{
    private $_db;

    /**
     * This method contains the logic to be executed when applying this migration.
     * Child classes may implement this method to provide actual migration logic.
     * @return boolean Returning false means, the migration will not be applied.
     */
    public function up()
    {
        $transaction=$this->getDbConnection()->beginTransaction();
        try
        {
            if($this->safeUp()===false)
            {
                $transaction->rollback();
                return false;
            }
            $transaction->commit();
        }
        catch(Exception $e)
        {
            echo "Exception: ".$e->getMessage().' ('.$e->getFile().':'.$e->getLine().")\n";
            echo $e->getTraceAsString()."\n";
            $transaction->rollback();
            return false;
        }
    }

```

Рисунок 1.28 – Базовый класс миграций

Таблица 1.25 – Таблица пользователя

user	
id	Int(11)
firstname	VARCHAR(255)
lastname	VARCHAR(255)
email	VARCHAR(255)
created_on	DATETIME
updated_on	DATETIME
last_visit_on	DATETIME
email_verified	Int(1)
is_disabled	Int(1)
role_id	Int(1)
activation_key	VARCHAR(255)
region_id	Int(1)
phone	VARCHAR(255)
mobile_token	VARCHAR(255)
is_first	Int(1)

```

class User extends CActiveRecord {
    public function tableName() {
        return 'user';
    }

    public function rules() {
        // password is unsafe on purpose, assign it manually after hashing only if not empty
        return array(
            array('email, firstname, lastname, is_disabled', 'filter', 'filter' => 'trim'),
            array('activation_key, created_on, updated_on, last_visit_on, email_verified', 'filter', 'filter' => 'trim', 'on' => 'create'),
            array('email, firstname, lastname, is_disabled, region_id', 'default', 'setOnEmpty' => true, 'value' => null),
            array('activation_key, created_on, updated_on, last_visit_on, email_verified', 'default', 'setOnEmpty' => true, 'value' => null),
            array('email, is_disabled, email_verified', 'required', 'except' => 'search'),
            array('created_on, updated_on, last_visit_on', 'date', 'format' => array('yyyy-MM-dd', 'yyyy-MM-dd HH:mm', 'yyyy-MM-dd HH:mm:ss')),
            array('activation_key', 'length', 'max' => 128, 'on' => 'create'),
            array('is_disabled, email_verified', 'boolean'),
            array('email', 'unique', 'except' => 'search'),
            array('phone', 'match', 'pattern' => '/^[0-9 ]+$/'),
        );
    }

    public function relations() {
        return array(
            'role' => array(self::BELONGS_TO, 'UserRole', 'role_id'),
            'companies' => array(self::HAS_MANY, 'Company', 'user_id'),
            'adverts' => array(self::HAS_MANY, 'Advert', array('id' => 'company_id'), 'through' => 'companies'),
            'region' => array(self::BELONGS_TO, 'Region', 'region_id'),
            'logins' => array(self::HAS_MANY, 'Login', 'user_id'),
        );
    }
}

```

Рисунок 1.29 – Класс пользователя

Таблица 1.26 – Таблица пользовательских уведомлений

user_notification		
id		Int(11)
type		Int(11)
title		VARCHAR(255)
text		text
from		DATETIME
to		DATETIME
withEmail		Int(1)
isRead		Int(1)
date		DATETIME

```

class UserNotification extends CActiveRecord
{
    public static $types = array(
        0 => array(
            'name' => 'Сайт'
        ),
        1 => array(
            'name' => 'Пользовательский'
        ),
    );
    /**
     * @return string the associated database table name
     */
    public function tableName()
    {
        return 'user_notification';
    }
    /**
     * @return array validation rules for model attributes.
     */
    public function rules()
    {
        // NOTE: you should only define rules for those attributes that
        // will receive user inputs.
        return array(
            array('type, from, to, withEmail', 'numerical', 'integerOnly'=>true),
            array('title', 'length', 'max'=>255),
            array('text', 'safe'),
            // The following rule is used by search().

```

Рисунок 1.30 – Класс пользовательских уведомлений

Таблица 1.27 – Таблица ролей пользователей

user_role		
id		Int(11)
name		VARCHAR(255)



```

class UserRole extends CActiveRecord
{
    /**
     * @return string the associated database table name
     */
    public function tableName()
    {
        return 'user_role';
    }

    /**
     * @return array validation rules for model attributes.
     */
    public function rules()
    {
        // NOTE: you should only define rules for those attributes that
        // will receive user inputs.
        return array(
            array('name', 'length', 'max'=>255),
            // The following rule is used by search().
            // @todo Please remove those attributes that should not be searched.
            array('id, name', 'safe', 'on'=>'search'),
        );
    }

    /**
     * @return array relational rules.
     */
    public function relations()
    {

```

Рисунок 1.31 – Класс ролей

Таблица 1.28 – Таблица кошелька пользователя

user_wallet		
id		Int(11)
user_id		Int(11)
amount		Int(11)

```

class UserWallet extends CActiveRecord
{
    public static $operationType = array(
        1 => array('label' => 'Пополнение'),
        2 => array('label' => 'Списание')
    );

    public static $incomeType = array(
        1 => array('label' => 'Администрация'),
        2 => array('label' => 'Система Qiwi')
    );

    public static $outcomeType = array(
        0 => array('name' => 'site', 'label' => 'Администрация'),
        1 => array('name' => 'toTop', 'label' => 'Продвижение объявления "Поднять наверх"'),
        2 => array('name' => 'color', 'label' => 'Продвижение объявления "Выделить цветом"'),
        3 => array('name' => 'banner', 'label' => 'Оплата рекламы в баннерной системе')
    );

    /**
     * @return string the associated database table name
     */
    public function tableName()
    {
        return 'user_wallet';
    }

    /**
     * @return array validation rules for model attributes.
     */
    public function rules()

```

Рисунок 1.32 – Класс кошелька пользователя

Таблица 1.29 – Таблица истории кошелька пользователя

user_wallet_history		
id		Int(11)
user_id		Int(11)
type		VARCHAR(255)
amount		Int(11)
incomeType		Int(11)
outcomeType		Int(11)
detail		VARCHAR(255)
date		VARCHAR(255)

```

class WalletHistory extends CActiveRecord
{
    /**
     * @return string the associated database table name
     */
    public function tableName()
    {
        return 'user_wallet_history';
    }

    /**
     * @return array validation rules for model attributes.
     */
    public function rules()
    {
        // NOTE: you should only define rules for those attributes that
        // will receive user inputs.
        return array(
            array('user_id, type', 'required'),
            array('user_id, type, incomeType, outcomeType', 'numerical', 'integerOnly'=>true),
            array('amount', 'length', 'max'=>8),
            array('detail, date', 'safe'),
            // The following rule is used by search().
            // @todo Please remove those attributes that should not be searched.
            array('id, user_id, type, amount, incomeType, outcomeType, detail, date', 'safe', 'on'=>'search'),
        );
    }

    /**
     * @return array relational rules.

```

Рисунок 1.33 – Класс истории кошелька пользователя

Таблица 1.30 – Таблица видео

video		
id		Int(11)
source		VARCHAR(255)
description		TEXT

```
class Video extends CActiveRecord
{
    /**
     * @return string the associated database table name
     */
    public function tableName()
    {
        return 'video';
    }

    /**
     * @return array validation rules for model attributes.
     */
    public function rules()
    {
        // NOTE: you should only define rules for those attributes that
        // will receive user inputs.
        return array(
            array('source', 'required'),
            array('source, description', 'length', 'max'=>255),
            // The following rule is used by search().
            // @todo Please remove those attributes that should not be searched.
            array('id, source, description', 'safe', 'on'=>'search'),
        );
    }

    /**
     * @return array relational rules.
     */
    public function relations()

```

Рисунок 1.34 – Класс видео

## 1.4 Используемые технологии программирования

Для реализации основной идеи данного проекта можно использовать различные языки и подходы программирования. Такие как языки для серверной части (PHP, Java, Ruby, Python C# и т.п.) или же языки для клиентской части (Html, JavaScript и т.п.) и так же разные виды СУБД (Sybase, MySQL, MSSQL, Oracle и т.п.). Далее я объясню почему я использовал именно основную связку Yii (php framework)+MySql+SphinxSearch.

### 1.4.1 СУБД MySql

MySQL — свободная реляционная система управления базами данных. Разработку и поддержку MySQL осуществляет корпорация Oracle, получившая права на торговую марку вместе с поглощённой Sun Microsystems, которая ранее приобрела шведскую компанию MySQL AB.

Продукт распространяется как под GNU General Public License, так и под собственной коммерческой лицензией. Помимо этого, разработчики создают функциональность по заказу лицензионных пользователей. Именно благодаря такому заказу почти в самых ранних версиях появился механизм репликации.

MySQL является решением для малых и средних приложений. Входит в состав серверов WAMP, AppServ, LAMP и в портативные сборки серверов Денвер, XAMPP, VertrigoServ. Обычно MySQL используется в качестве сервера, к которому обращаются локальные или удалённые клиенты,

однако в дистрибутив входит библиотека внутреннего сервера, позволяющая включать MySQL в автономные программы.

Гибкость СУБД MySQL обеспечивается поддержкой большого количества типов таблиц: пользователи могут выбрать как таблицы типа MyISAM, поддерживающие полнотекстовый поиск, так и таблицы InnoDB, поддерживающие транзакции на уровне отдельных записей. Более того, СУБД MySQL поставляется со специальным типом таблиц EXAMPLE, демонстрирующим принципы создания новых типов таблиц. Благодаря открытой архитектуре и GPL-лицензированию, в СУБД MySQL постоянно появляются новые типы таблиц.

## **1.4.2 PHP**

Причина почему был использован yii framework, это высокая производительность в сравнении с другими framework'ами, также защищенность и быстрота реализации поставленной задачи. Для начала следует рассказать, про сам php.

Практический характер PHP из-за пяти основных характеристик:

- традиционализм;
- простота;
- коэффициент полезного действия;
- безопасность;
- гибкость.

Существует еще одна "особенность", что делает PHP особенно привлекательным: это бесплатно! Кроме того, с открытым исходным кодом (Open Source).

### **1.4.2.1 Традиционность**

PHP язык будет казаться знакомым программистам, работающим в различных областях. Многие языковые конструкции, заимствованные из C, Perl.

PHP код очень похож на тот, который происходит в обычной программе в C или Pascal. Это значительно сокращает первоначальные усилия в изучении PHP. PHP - это язык, который сочетает в себе преимущества Perl и C и специально направлена на работу Интернета, с универсальным языком (хотя и с некоторыми оговорками) и ясным синтаксисом.

Хотя PHP является относительно молодой язык, он приобрел такую популярность среди веб-программистов, которые на данный момент является чуть ли не самым популярным языком для создания веб-приложений (скриптов).

### 1.4.2.2 Простота

PHP скрипт может состоять из 10 000 строк или одной линии - все это зависит от специфики вашего приложения. Вам не нужно загружать библиотеку, указать специальные параметры компиляции или что-нибудь подобное. PHP простой механизм начинает выполнять код после первой управляющей последовательности (<?) И продолжается до того момента, когда он встретил последовательность парень экранирования (?>). Если код имеет правильный синтаксис, он выполняется в точности, как указано программистом.

PHP - это язык, который может быть встроен непосредственно в страницы HTML-код, который, в свою очередь, будут правильно обработаны PHP -interpretatorom. Мы можем использовать PHP для написания CGI-скриптов и избавиться от многих операторов неуклюжим вывода текста. Мы можем принести PHP для создания HTML-документов, избавившись от ряда внешних вызовов сценария.

Большое разнообразие функций PHP избавит вас от написания многопользовательский определенные функции в С или Pascal.  
коэффициент полезного действия

Эффективность является чрезвычайно важным фактором при программировании для многопользовательских сред, среди которых и веб связаны между собой.

Очень важным преимуществом PHP является его "движок". "Двигатель» PHP не является ни компилятор или интерпретатор. Он является толкователем передающего. Такое устройство "двигателя» PHP скрипты могут обрабатывать достаточно высокую скорость.

По некоторым оценкам, большинство PHP-скриптов (особенно не очень большой) они обрабатываются быстрее, чем аналогичные программы, написанные на языке Perl. Тем не менее, это не делает разработчиков PHP, скомпилированные бинарные файлы будут значительно работать быстрее - в десятки, а иногда и в сотни раз. Но производительность PHP достаточно, чтобы создать достаточно серьезный веб-приложение. Сведения о структуре и характеристиках PHP «движка» можно найти здесь.

### 1.4.2.3 Безопасность

PHP предоставляет разработчикам и администраторам гибкие и эффективные средства безопасности, которые разделены на две категории: средства системного уровня и ресурсов на уровне приложений.

В PHP реализованы механизмы безопасности, которые находятся под контролем администраторов; при правильной настройке PHP это обеспечивает максимальную свободу действий и безопасность. PHP может работать в так называемый безопасный режим (Safe Mode), что ограничивает возможность

использования пользователей PHP ряд важных показателей. Например, вы можете ограничить максимальное время выполнения и использование памяти (неконтролируемый расход памяти отрицательно сказывается на производительности сервера). По аналогии с CGI-BIN администратор может также устанавливать ограничения на каталоги, в которых можно просматривать пользователь и выполнять PHP-скрипты и использовать PHP-скрипты для просмотра конфиденциальной информации на сервере (например, файл PASSWORD).

Стандартный набор функций PHP включает в себя ряд надежных механизмов шифрования. PHP также совместим со многими сторонними приложениями фирм, которые могут быть легко интегрированы с защищенными технологиями электронной коммерции (электронной коммерции). Еще одним преимуществом является то, что исходный код PHP скриптов не может рассматриваться в браузере, поскольку сценарий компилируется до его отправки по запросу пользователя. Реализация PHP на стороне сервера, чтобы предотвратить хищение нетривиальных пользовательских сценариев, знание которых имеет по крайней мере, для выполнения команды View Source.

#### **1.4.2.4 Гибкость**

Поскольку PHP встроен (встроенный) язык, он является очень гибким по отношению к потребностям разработчика. Хотя PHP, как правило, рекомендуется для использования в сочетании с HTML, это может также быть интегрирован в JavaScript, WML, XML и другие языки. Кроме того, хорошо структурированные приложения PHP легко расширяться по мере необходимости (впрочем, это относится ко всем основным языкам программирования).

Нет проблем с в зависимости от браузера в качестве клиента перед отправкой PHP скрипты не составляются полностью на стороне сервера. На самом деле, PHP скрипты могут быть переданы в любое устройство с браузером, в том числе мобильные телефоны, электронные записные книжки, пейджеры и портативные компьютеры, не говоря уже о традиционных ПК. Программисты занимается поддержкой утилиты можно запускать PHP из командной строки.

Поскольку PHP не содержит код, ориентированный на конкретный веб-сервер, пользователи не ограничены конкретными серверами (возможно, неизвестных им). Apache, Microsoft IIS, Netscape Enterprise Server, Stronghold и Зевс - PHP работает на всех перечисленных серверах. Поскольку эти серверы работают на разных платформах, PHP в целом является независимым от платформы языка, и есть на платформах, таких как UNIX, Solaris, FreeBSD и Windows 95/98 / NT / 2000 / XP / 2003.

Наконец, PHP инструменты позволяют программисту работать с внешними компонентами, такими как Enterprise Java Beans или объекты Win32 COM. С этими новыми особенностями PHP занимает свое достойное место среди современных технологий и обеспечивает расширение масштабов проектов, необходимых пределов.

#### **1.4.2.5 Бесплатное распространение**

Стратегия Open Source и распространять исходный код программ среди масс, были, несомненно, благотворно сказывается на многих проектах, в первую очередь - Linux, хотя успех проекта Apache сильно усилило позиции сторонников Open Source. То же самое относится и к истории PHP, поскольку поддержка клиентов со всего мира оказалась очень важным фактором в развитии PHP проекта.

Принятие стратегии Open Source и бесплатное распространение исходного кода PHP был неоценимую услугу для пользователей. Кроме того, отзывчивый сообщества пользователей PHP является своего рода "коллективной поддержки клиентов", а также популярные электронные конференции могут найти ответы даже на самые сложные вопросы.

#### **1.4.3 Yii**

Yii — это высокоэффективный основанный на компонентной структуре PHP-фреймворк для разработки масштабных веб-приложений. Он позволяет максимально применить концепцию повторного использования кода и может существенно ускорить процесс веб-разработки. Название Yii (произносится как Yee или [ji:]) означает простой (easy), эффективный (efficient) и расширяемый (extensible).

##### **1.4.3.1 Требования**

Для запуска веб-приложений, построенных на Yii, вам понадобится веб-сервер с поддержкой PHP версии 5.1.0.

Для разработчиков, желающих использовать Yii, крайне полезным будет понимание концепции объектно-ориентированного программирования (ООП), так как Yii — это строго объектно-ориентированный фреймворк.

Для чего Yii будет лучшим выбором?

Yii — это фреймворк для веб-программирования общего назначения, который может быть использован для разработки практически любых веб-приложений. Благодаря своей легковесности и наличию продвинутых средств кэширования, Yii особенно подходит для разработки приложений с большим потоком трафика, таких как порталы, форумы, системы управления контентом (CMS), системы электронной коммерции и др.

### 1.4.3.2 Сравнение

Подобно большинству других PHP-фреймворков, Yii — это MVC-фреймворк.

Превосходство Yii над другими фреймворками заключается в эффективности, широких возможностях и качественной документации. Yii изначально спроектирован очень тщательно для соответствия всем требованиям при разработке серьёзных веб-приложений. Yii не является ни побочным продуктом какого-либо проекта, ни сборкой сторонних решений. Он является результатом большого опыта авторов в разработке веб-приложений, а также их исследований наиболее популярных веб-фреймворков и приложений.

### 1.4.4 SphinxSearch

**Sphinx** (англ. *SQL Phrase Index*) — система полнотекстового поиска, разработанная Андреем Аксеновым и распространяемая по лицензии GNU GPL. Отличительной особенностью является высокая скорость индексации и поиска, а также интеграция с существующими СУБД (MySQL, PostgreSQL) и API для распространённых языков веб-программирования (официально поддерживаются PHP, Python, Java; существуют реализованные сообществом API для Perl, Ruby, .NET и C++).

Преимущества:

- Высокая скорость индексации (до 10-15 МБ/сек на каждое процессорное ядро);
- Высокая скорость поиска (до 150—250 запросов в секунду на каждое процессорное ядро с 1 000 000 документов);
- Высокая масштабируемость (крупнейший известный кластер индексирует до 3 000 000 000 документов и поддерживает более 50 миллионов запросов в день);
- Поддержка распределенного поиска;
- Поддержка нескольких полей полнотекстового поиска в документе (до 32 по умолчанию);
- Поддержка нескольких дополнительных атрибутов для каждого документа (то есть группы, временные метки и т. д.);
- Поддержка стоп-слов;
- Поддержка однобайтовых кодировок и UTF-8;
- Поддержка морфологического поиска — имеются встроенные модули для английского, русского и чешского языков; доступны модули для французского, испанского, португальского, итальянского,



румынского, немецкого, голландского, шведского, норвежского, датского, финского, венгерского языков;

- Нативная поддержка MySQL (всех типов таблиц, в том числе MyISAM, InnoDB, NDB, архив и т. д.);
- Нативная поддержка PostgreSQL;
- Поддержка ODBC совместимых баз данных (MS SQL, Oracle и т. д.).

Иными словами, SphinxSearch был применен в данном проекте с целью, снятия нагрузки с сервера базы данных. Это осуществляется посредством, индексации необходимых таблиц раз в сутки.

## 2 Расчетная часть

### 2.1 Расчет места для хранения базы данных

В данном шаге, нужно знать какой объем памяти станет занимать создаваемая БД. Объем наружной памяти, нужный для функционирования системы, формируется из двух составляющих: память, занимаемая модулями СУБД и память, отводимая под данные ( $M_d$ ). Более существенным обычно является  $M_d$ . Размер памяти, занимаемый программными модулями юзера, традиционно слишком невелик по сравнению с объемом самих данных, поэтому имеет возможность не учитываться. Так как в БД есть похожие таблицы посчитаем усреднено их значение. В работе рассчитывается ориентировочный наибольший объем памяти занимаемой Базы Данных. Расчет физической памяти таблиц указана в таблицах 2.1-2.30

Таблица 2.1- Подсчет физической памяти для blog\_article

blog_article статьи		
id	4	int(11)
title	301	varchar(300)
short_description	4	TEXT
full_text	4	TEXT
create_date	8	DATETIME
modified_date	8	DATETIME
published	1	BIT(1)

Общий объем поля: 330 байт

Количество: ~ 15 000

Требуемая память: 4.95 Мбайт

Таблица 2.2 - Подсчет физической памяти для blog\_article\_cat\_assoc

blog_article_cat_assoc статьи		
article_id	4	int (11)
category_id	4	int(11)

Общий объем поля: 406 байт

Количество: ~ 50

Требуемая память: 20 300 байт

Таблица 2.3 - Подсчет физической памяти для blog\_category

blog_category статьи		
Id	4	INT (11)
Name	256	Varchar(255)

Общий объем поля: 300 байт

Количество: ~ 15

Требуемая память: 4500 байт

Таблица 2.4 - Подсчет физической памяти для blog\_comment

blog_comment статьи		
user_id	4	INT(11)
article_id	4	INT(11)
text	4	TEXT

Общий объем поля: 12 байт

Количество: ~ 50 000

Требуемая память: 600 кбайт

Таблица 2.5 - Подсчет физической памяти для catalog\_advert

catalog_advert объявления		
Id	4	INT (11)
Title	256	Varchar(255)
text	4	TEXT
Company_id	4	Int(11)
Region_id	4	Int(11)
Service_id	4	Int(11)
created_date	8	Date_time
edited_date	8	Date_time
Published	4	Int(1)
Is_disabled	4	Int(1)
price_tg	9	Decimal(8.2)
Price_usd	9	Decimal(8.2)
Price_eur	9	Decimal(8.2)
default_currency	1	TINYINT(4)
minServiceAmount	4	INT(11)
maxServiceAmount	4	INT(11)
coloredDate	8	DATETIME
toTopDate	8	DATETIME

Общий объем поля: 392 байт

Количество: ~ 150 000

Требуемая память: 58 800 000 байт

Таблица 2.6 - Подсчет физической памяти для catalog\_advert\_feedback

catalog_advert_feedback объявления		
Id	4	INT(11)
Advert_id	4	Int(11)
user_id	4	Int(11)
Rate	8	DOUBLE(2.1)
Approved	1	BIT(1)
Text	4	TEXT
created_on	8	DATETIME
modified_on	8	DATETIME

Общий объем поля: 41 байт

Количество: ~ 300 000

Требуемая память: 12 300 000 байт

Таблица 2.7 - Подсчет физической памяти для catalog\_advert\_feedback\_comment

catalog_advert_feedback_comment объявления		
Id	4	INT(11)
feedback_id	4	Int(11)
Text	4	text
Created_on	8	DATETIME
Modified_on	8	DATETIME

Общий объем поля: 28 байт

Количество: ~ 500 000

Требуемая память: 14 000 000 байт

Таблица 2.8 - Подсчет физической памяти для catalog\_advert\_promotions

catalog_advert_promotions объявления		
Id	4	INT(11)
Name	256	VARCHAR(255)
Description	4	TEXT
Label	256	VARCHAR(255)
Price	4	Int(11)
timeHours	4	Int(11)
Is_disabled	4	Int(1)
Test	51	Varchar(50)

Общий объем поля: 583 байт

Количество: ~ 2

Требуемая память: 1166 байт

Таблица 2.9 - Подсчет физической памяти для catalog\_advert\_user

catalog_advert_user объявления		
user_id	4	Int(11)
advert_id	4	Int(11)

Общий объем поля: 8 байт

Количество: ~ 15 000

Требуемая память: 120 000 байт

Таблица 2.10 - Подсчет физической памяти для catalog\_company

catalog_company объявления		
Id	4	Int(11)
name	256	VARCHAR(255)
User_id	4	Int(11)
description	4	Text
created_date	8	DATETIME
edited_date	8	DATETIME
phone	256	VARCHAR(255)
billing	256	VARCHAR(255)
address	256	VARCHAR(255)
map	256	VARCHAR(255)
cash	4	Int(1)
cashless	4	Int(1)
cards	4	Int(1)
approved	4	Int(1)
is_disabled	4	Int(1)
Portfolio	4	TEXT
site_link	256	VARCHAR(255)
email	256	VARCHAR(255)

Общий объем поля: 1617 байт

Количество: ~ 80 000

Требуемая память: 129 360 000 байт

Таблица 2.12 - Подсчет физической памяти для catalog\_event

catalog_event объявления		
id	4	Int(11)
Name	256	Varchar(255)

Общий объем поля: 260 байт

Количество: ~ 10

Требуемая память: 2 600 байт

Таблица 2.12 – Ассоциативная таблица мероприятий и типов услуг

catalog_event_service_assoc		
event_id	4	Int(11)
service_id	4	Int(11)
priority	4	SMALLINT(6)

Общий объем поля: 12 байт

Количество: ~ 10

Требуемая память: 1 200 байт

Таблица 2.13 – Таблица типов услуг

catalog_service		
id	4	Int(11)
Name	256	Varchar(255)
disabled	4	Int(1)

Общий объем поля: 264 байт

Количество: ~ 10

Требуемая память: 2 640 байт

Таблица 2.14 – Таблица стоимости основных валют

Currency		
Id	4	INT(11)
Name	256	VARCHAR(255)
Value	4	DECIMAL(8,2)
Updated_date	4	DATETIME

Общий объем поля: 268 байт

Количество: ~ 10

Требуемая память: 2 680 байт

Таблиц 2.15 – Таблица часто задаваемых вопросов

faq		
id	4	Int(11)
name	256	VARCHAR(255)
text	4	TEXT
category_id	4	INT(11)

Общий объем поля: 268 байт

Количество: ~ 10

Требуемая память: 2 680 байт

Таблица 2.16 – Таблица категорий часто задаваемых вопросов

faq_category		
id	4	Int(11)
Name	256	Varchar(255)

Общий объем поля: 260 байт

Количество: ~ 5

Требуемая память: 1 300 байт

Таблица 2.17 – Таблица файлов

File		
Id	4	Int(11)
title	256	Varchar(255)
link	256	Varchar(255)
upload_date	8	DATETIME
type	1	TINYINT(1)
owner_id	4	INT(11)
mime	256	Varchar(255)

Общий объем поля: 785 байт

Количество: ~ 100000

Требуемая память: 78 500 000 байт

Таблица 2.18 – Таблица типовых страниц

info_pages страницы		
id	4	INT(11)
title	256	Varchar(255)
text	4	MEDIUMTEXT
alias	51	VARCHAR(50)

Общий объем поля: 315 байт

Количество: ~ 10

Требуемая память: 3 150 байт

Таблица 2.19 – Таблица авторизационных данных пользователя

Login		
id	4	Int(11)
identity	256	VARCHAR(255)
password	256	VARCHAR(255)
created_on	8	DATETIME
is_active	1	TINYINT(1)
is_disabled	1	TINYINT(1)
network	256	Varchar(255)
user_id	4	Int(11)
updated_on	8	DATETIME

Общий объем поля: 794 байт

Количество: ~ 300 000

Требуемая память: 158 800 кбайт

Таблица 2.20 – Таблица истории платежей

payment_qiwi_history		
txn_id	29	Varchar(28)
user_id	4	INT(11)
dateTime	8	DATETIME
txn_date	8	DATETIME
Amount	9	Decimal(8,2)
Prv_txn	21	Varchar(20)

Общий объем поля: 79 байт

Количество: ~ 100 000

Требуемая память: 2 600 байт

Таблица 2.21 – Таблица регионов

region		
id	4	Int(11)
name	256	Varchar(255)

Общий объем поля: 260 байт

Количество: ~ 12

Требуемая память: 3 120 байт



Таблица 2.22 – Таблица баннеров

reklama		
id	4	Int(11)
region_id	4	Int(11)
text	4	TEXT
zone	4	Int(11)
link	256	VARCHAR(255)
title	256	VARCHAR(255)
image	256	VARCHAR(255)
start_time	8	DATETIME
end_time	8	DATETIME
views_count	4	Int(11)
client_views	4	Int(11)
is_approved	4	Int(1)

Общий объем поля: 812 байт

Количество: ~ 10

Требуемая память: 8 120 кбайт

Таблица 2.23 – Таблица ценовых пакетов для показа рекламы

reklama_price		
id	4	Int(11)
price	4	Int(11)
sale	4	Int(11)
region_id	4	Int(11)

Общий объем поля: 16 байт

Количество: ~ 10

Требуемая память: 160 байт

Таблица 2.24 – Таблица учета миграций

tbl_migration		
version	256	VARCHAR(255)
apply_time	8	TIMESTAMP

Общий объем поля: 264 байт

Количество: ~ 100

Требуемая память: 26 400 байт

Таблица 2.25 – Таблица пользователя

user		
id	4	Int(11)
firstname	256	VARCHAR(255)
lastname	256	VARCHAR(255)
email	256	VARCHAR(255)
created_on	8	DATETIME
updated_on	8	DATETIME
last_visit_on	8	DATETIME
email_verified	4	Int(1)
is_disabled	4	Int(1)
role_id	4	Int(1)
activation_key	256	VARCHAR(255)
region_id	4	Int(1)
phone	256	VARCHAR(255)
mobile_token	256	VARCHAR(255)
is_first	4	Int(1)

Общий объем поля: 1584 байт

Количество: ~ 100 000

Требуемая память: 158 400 000 байт

Таблица 2.26 – Таблица пользовательских уведомлений

user_notification		
id	4	Int(11)
type	4	Int(11)
title	256	VARCHAR(255)
text	4	text
from	8	DATETIME
to	8	DATETIME
withEmail	4	Int(1)
isRead	4	Int(1)
date	8	DATETIME

Общий объем поля: 300 байт

Количество: ~ 500 000

Требуемая память: 150 000 000 байт

Таблица 2.27 – Таблица ролей пользователей

user_role		
id	4	Int(11)
name	256	VARCHAR(255)

Общий объем поля: 260 байт

Количество: ~ 10

Требуемая память: 2 600 байт

Таблица 2.28 – Таблица кошелька пользователя

user_wallet		
id	4	Int(11)
user_id	4	Int(11)
amount	4	Int(11)

Общий объем поля: 260 байт

Количество: ~ 100 000

Требуемая память: 26 000 000 байт

Таблица 2.29 – Таблица истории кошелька пользователя

user_wallet_history		
id	4	Int(11)
user_id	4	Int(11)
type	256	VARCHAR(255)
amount	4	Int(11)
incomeType	4	Int(11)
outcomeType	4	Int(11)
detail	256	VARCHAR(255)
date	256	VARCHAR(255)

Общий объем поля: 788 байт

Количество: ~ 1 000 000

Требуемая память: 788 Мбайт

Таблица 2.30 – Таблица видео

video		
id	4	Int(11)
source	256	VARCHAR(255)
description	4	TEXT

Общий объем поля: 264 байт

Количество: ~ 10

Требуемая память: 2 640 байт

Итого:

$M_c = 4950000 + 20300 + 4500 + 600000 + 58800000 + 12300000 + 14000000 + 1166 + 120000 + 129360000 + 600 + 1200 + 2640 + 2680 + 1300 + 78500000 + 3150 + 158000 + 2600 + 120 + 8120 + 160 + 26400 + 158400000 + 150000000 + 2600 + 26000000 + 788000000 + 2640 = 1421268176$  байт  $\approx 1.4$  Гб.

Объем таблиц колоссальны. Для этого предусмотрен сервер, указанный в технических требованиях к системе.

## **2.2 Меры по обеспечению безопасности**

### **2.2.1 Предотвращение межсайтового скриптинга**

Межсайтовый скриптинг (также известный как XSS) — злонамеренный сбор информации пользователя через страницы веб-приложения. Чаще всего, производящий атаку, используя уязвимости приложения, включает в текст страницы JavaScript, VBScript, ActiveX, HTML или Flash. Делается это для получения информации других пользователей приложения и последующего её использования в нехороших целях. К примеру, плохо написанный форум может отображать сообщения пользователей без какой-либо проверки. Атакующий может вставить JavaScript-код в сообщение. Все, кто прочитает это сообщение, выполнит код на своём компьютере.

Чтобы не допустить XSS-атак, нужно всегда проверять то, что ввёл пользователь, прежде чем это отображать. Конечно, чтобы не допустить ввода скриптов, можно кодировать все HTML-сущности. В некоторых ситуациях такое поведение нежелательно, так как ввод HTML становится недоступен.

Yii включает в себя библиотеку HTMLPurifier и предоставляет разработчику полезный компонент CHtmlPurifier, который может отфильтровать весь вредоносный код при помощи тщательно проверенного белого листа. Также компонент делает код совместимым со стандартами.

### **2.2.2 Предотвращение подделки межсайтовых запросов**

Подделка межсайтового запроса (CSRF) — атака, при которой сайт атакующего заставляет браузер пользователя выполнить какое-либо действие на другом сайте.

К примеру, на сайте атакующего есть страница, содержащая тэг `img` с атрибутом `src`, указывающим на сайт банка:

`http://bank.example/перевод?сумма=10000&кому=кулхацкеру`. Если в браузере пользователя установлен `cookie`, позволяющий запомнить его на сайте, посещение такой страницы вызовет перевод 10000 тугриков нехорошему кулхацкеру. В CSRF, в отличие от межсайтового скриптинга, основанного на

доверии пользователя к некоторому сайту, используется доверие сайта определённому пользователю.

Для того, чтобы не допустить CSRF, важно придерживаться простого правила: GET — только для получения данных. Ничего менять при GET-запросах нельзя. Для POST необходимо использовать случайное значение, которое можно проверить на сервере и убедиться, что запрос идёт оттуда, откуда нужно.

В Yii реализована защита от CSRF-атаки, проводимой через POST. Защита основана на хранении случайного значения в cookie и сравнения его со значением в POST.

### 2.2.3 Предотвращение атак через cookie

Защита cookie очень важна, так как именно в них чаще всего хранится ID сессии. Если злоумышленник получит ID сессии, он получит и всю информацию, которая в ней хранится.

Есть несколько способов предотвращения атак через cookie:

- Использовать SSL для создания защищённого соединения и передавать cookie только через него. Атакующий не сможет расшифровать содержимое передаваемых cookie.
- Вовремя объявлять сессию устаревшей, включая все cookie и маркеры сессии, для того, чтобы снизить возможность атаки.
- Предотвратить XSS, тем самым исключив захват cookie.
- Проверять данные cookie и определять, изменены ли они.
- В Yii реализована проверка на изменения через подсчёт хэша HMAC от значений cookie.

## 3 Разработка поисково-информационной системы

### 3.1 Этапы разработки

Таблица 3.1 – Календарный план работ

Период	Задачи
1-ая неделя	Составление и внесение изменений в ТЗ
2-4 неделя	Разработка интерфейсов и дизайна (разработчик интерфейсов, дизайнер)
2-12 неделя – 2-3 недели – 4-5 недели – 6-8 недели	Разработка веб-портала: – Модули объявлений, регистрации, пользователей (backend и frontend)

<ul style="list-style-type: none"> <li>- 9-10 недели</li> <li>- 11-12 недели</li> </ul>	<ul style="list-style-type: none"> <li>разработчики – 3 человека)</li> <li>- Модуль отзывов и рейтингов (backend и frontend разработчики – 3 человека)</li> <li>- Модуль платежей и платных услуг (backend и frontend разработчики – 3 человека)</li> <li>- Модуль полезной информации и портфолио рекламодателей (backend и frontend разработчики – 2 человека)</li> <li>- Тестирование веб-портала и наполнение информацией (тестер и разработчики)</li> </ul>
<p>9-13 недели</p> <ul style="list-style-type: none"> <li>- 9 неделя</li> <li>- 10-11 недели</li> <li>- 12 неделя</li> <li>- 13 неделя</li> </ul>	<p>Разработка веб-приложения «Онлайн калькулятор»</p> <ul style="list-style-type: none"> <li>- Авторизация, личный кабинет, просмотр каталога услуг (backend и frontend разработчики – 2 человека)</li> <li>- Калькулятор торжеств, Список гостей (backend и frontend разработчики – 2 человека)</li> <li>- Блокнот пользователя, выгрузка в PDF (frontend разработчик)</li> <li>- Тестирование и наполнение информацией (тестер и разработчики)</li> </ul>
<p>13-17 недели</p> <ul style="list-style-type: none"> <li>- 13-15 недели</li> <li>- 13 неделя</li> <li>- 14-15 недели</li> <li>- 15-16 недели</li> <li>- 17 неделя</li> </ul>	<p>Наращивание функционала веб-портала</p> <ul style="list-style-type: none"> <li>- Баннерная сеть (backend и frontend разработчики – 3 человека)</li> <li>- Чат-система (backend и frontend разработчики – 2 человека))</li> <li>- Создание рабочих мест модераторов (backend и frontend разработчики – 2 человека)</li> <li>- Нарашивание функционала личного кабинета (backend и frontend разработчики – 2</li> </ul>

	<p>человека)</p> <ul style="list-style-type: none"> <li>– Тестирование и наполнение информацией (тестер и разработчики)</li> </ul>
<p>15-21 недели</p> <ul style="list-style-type: none"> <li>– 15-16 неделя</li> <li>– 16-18 недели</li> <li>– 18 неделя</li> <li>– 19 неделя</li> <li>– 20-21 недели</li> </ul>	<p>Разработка мобильного приложения под iOS и Android</p> <ul style="list-style-type: none"> <li>– Регистрация, личный кабинет, просмотр каталога услуг (backend и frontend разработчики – 3 человека)</li> <li>– Калькулятор торжеств, Список гостей (backend и frontend разработчики – 2 человека)</li> <li>– Блокнот пользователя, выгрузка в PDF (frontend разработчики – 2 человека)</li> <li>– Система уведомлений (бэкэнд и фронтэнд разработчики)</li> <li>– Тестирование под разные платформы и наполнение информацией (тестер)</li> </ul>
22-25 недели	<p>Комплексное тестирование и доработка системы (тестер и разработчики)</p>
26 неделя	<p>Завершение разработки, подписание актов приемки-передачи</p>

### 3.2 Разработка интерфейса «Tamada.kz»

Интерфейс поисково-информационной системы (веб-портала) «Tamada.kz» разработан с учетом интуитивно понятного flatfull дизайна.

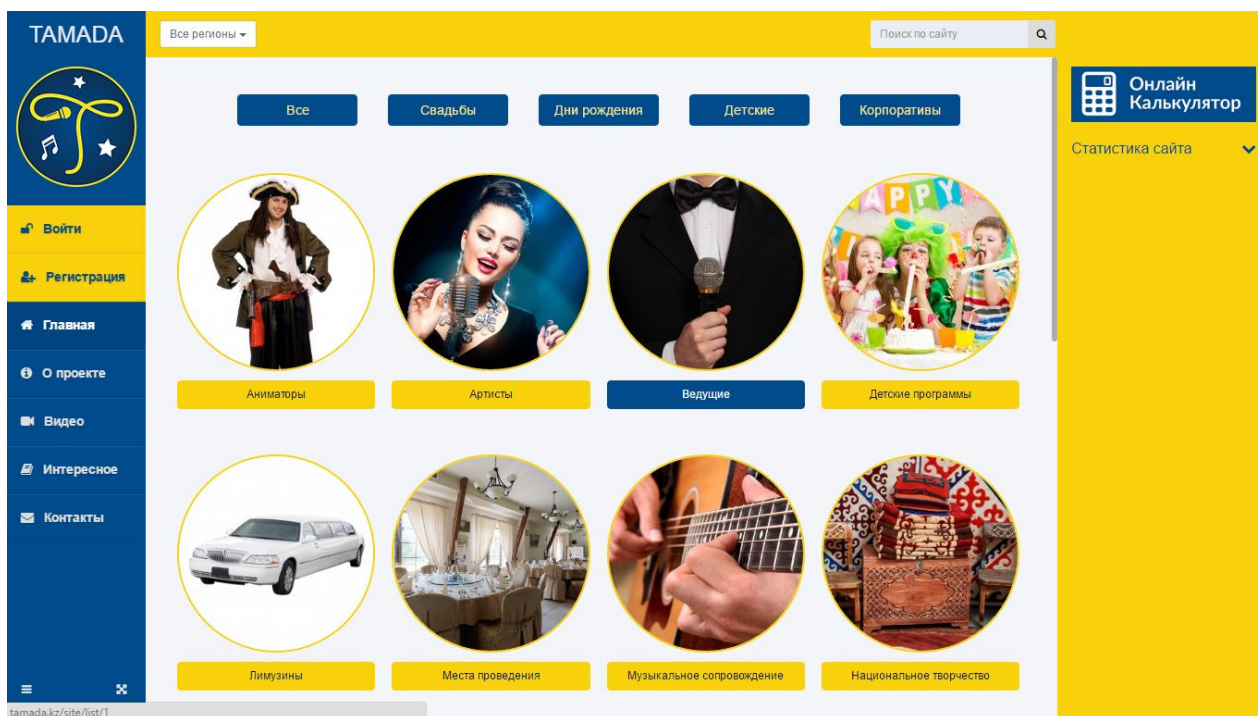


Рисунок 3.1 – Главная страница системы

В связи с популяризацией социальных сетей, в портале предусмотрена возможность регистрации и авторизации через них.

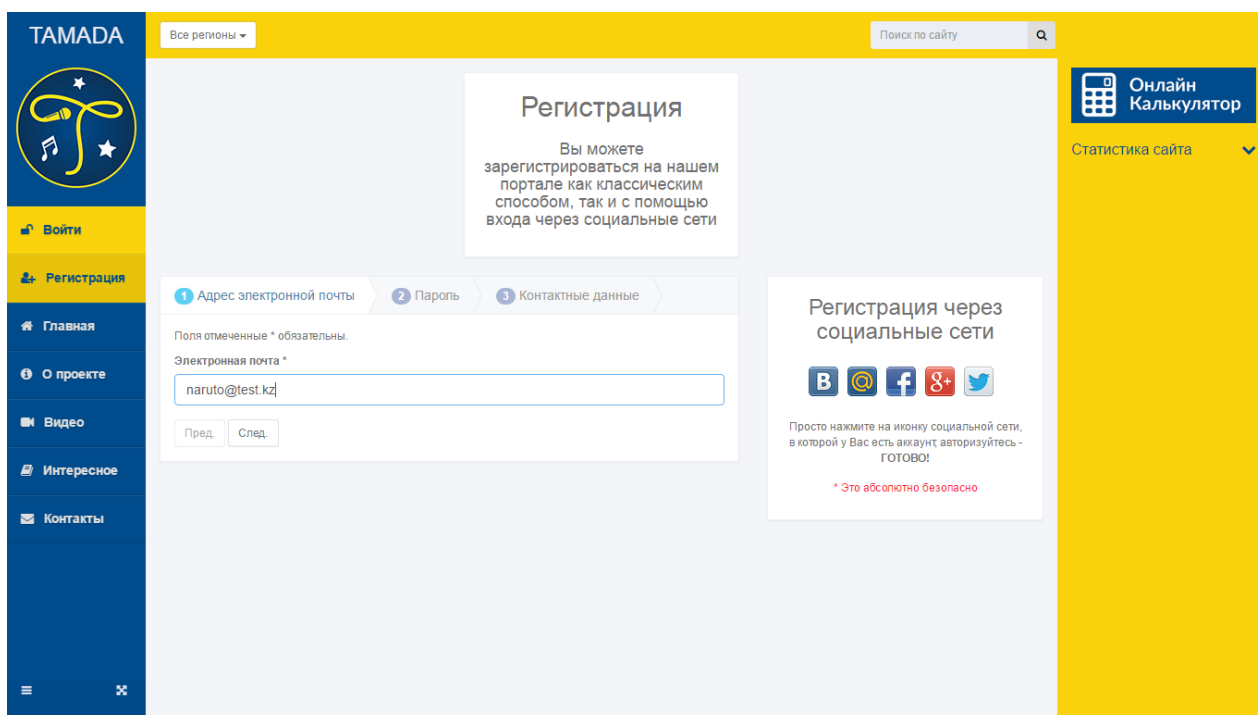


Рисунок 3.2 – Регистрация



После регистрации пользователю становится доступен основной функционал системы. Также, в любой удобный момент можно изменить основную информацию аккаунта.

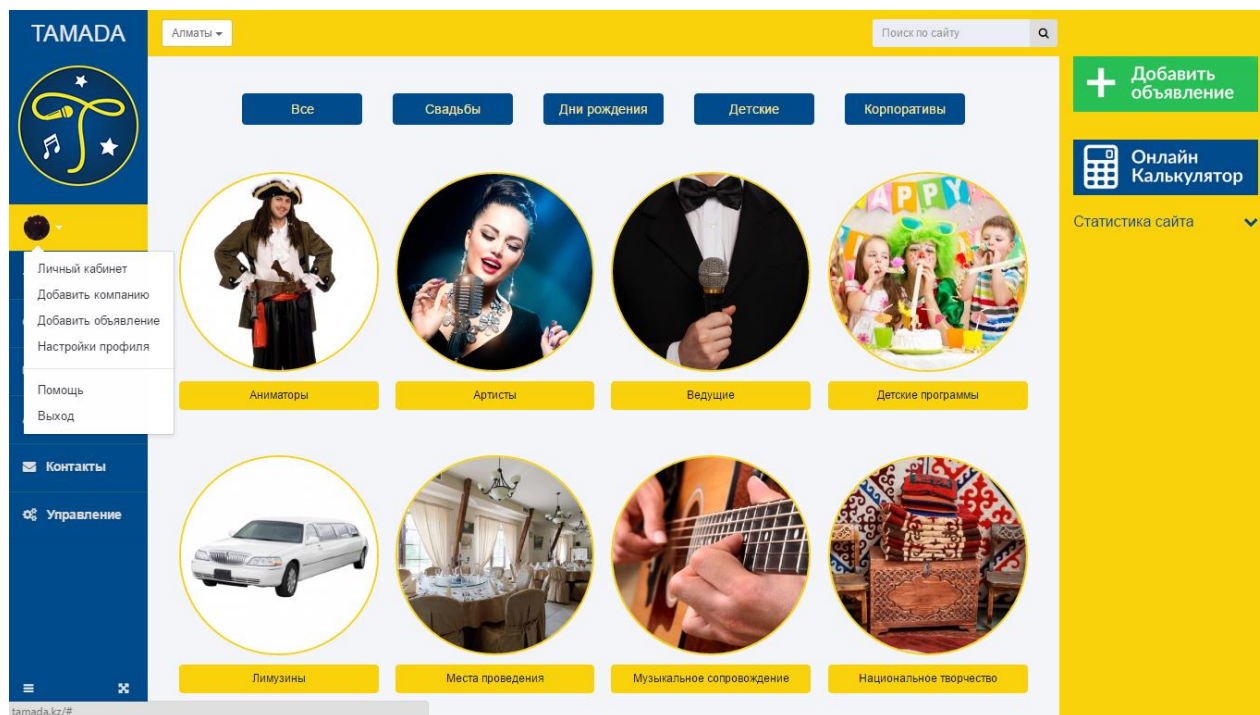


Рисунок 3.3 – Главная страница после регистрации

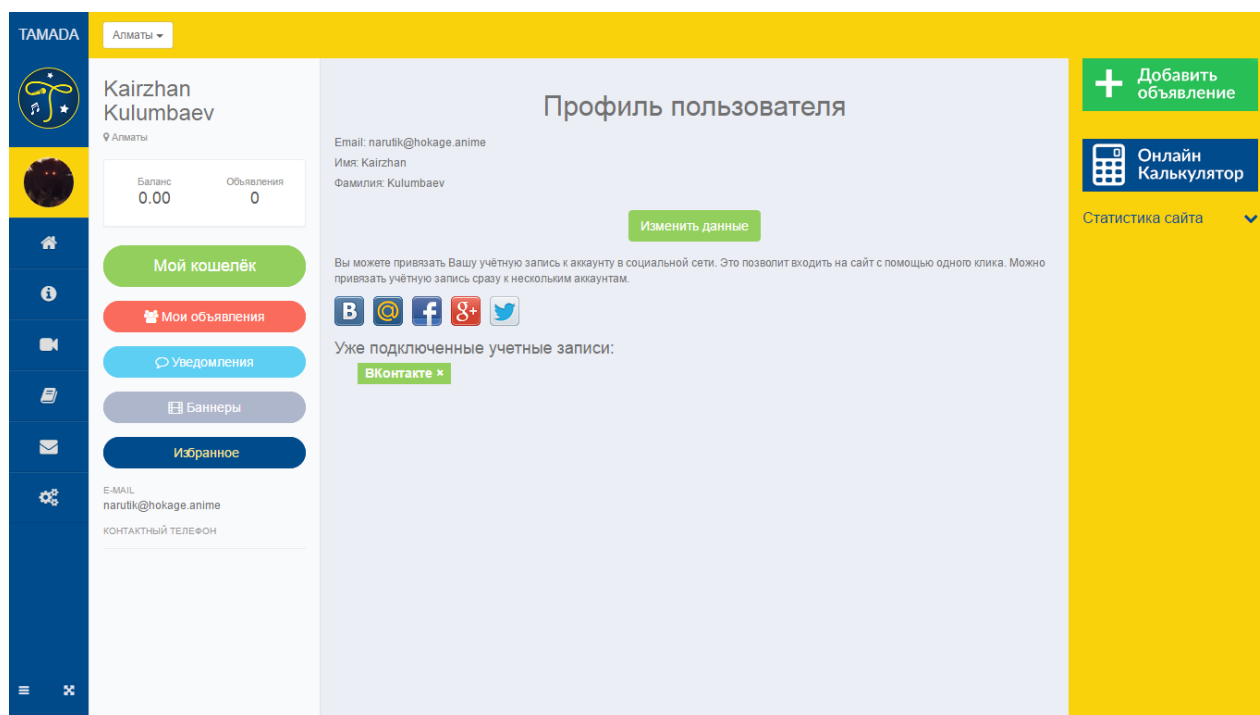


Рисунок 3.4 – Личный кабинет пользователя

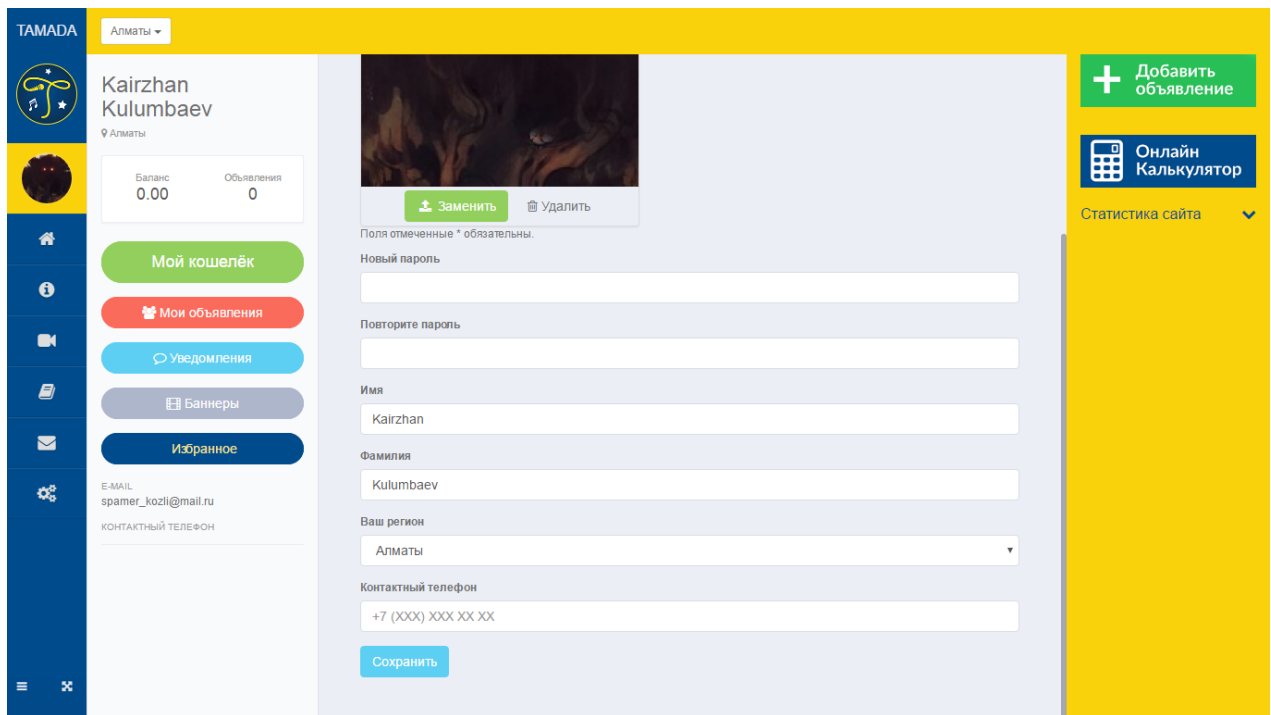


Рисунок 3.5 – изменение личной информации пользователя

Также пользователь, может подать объявление, в случае, если у пользователя нет компаний, от лица которых будут подаваться объявления, система переадресует, его на создание компании.

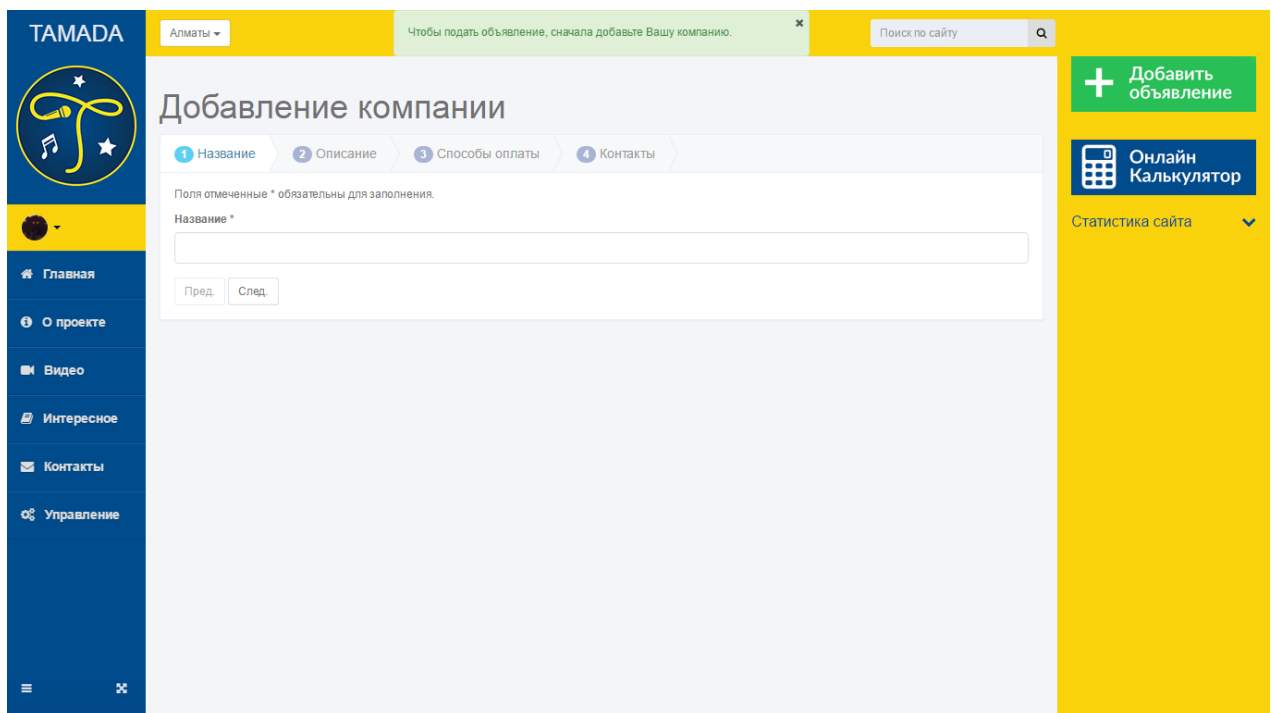


Рисунок 3.6 – Добавление компании

Для понятности, также был разработан интерактивный помощник, по инструментарию доступному в описании компании.

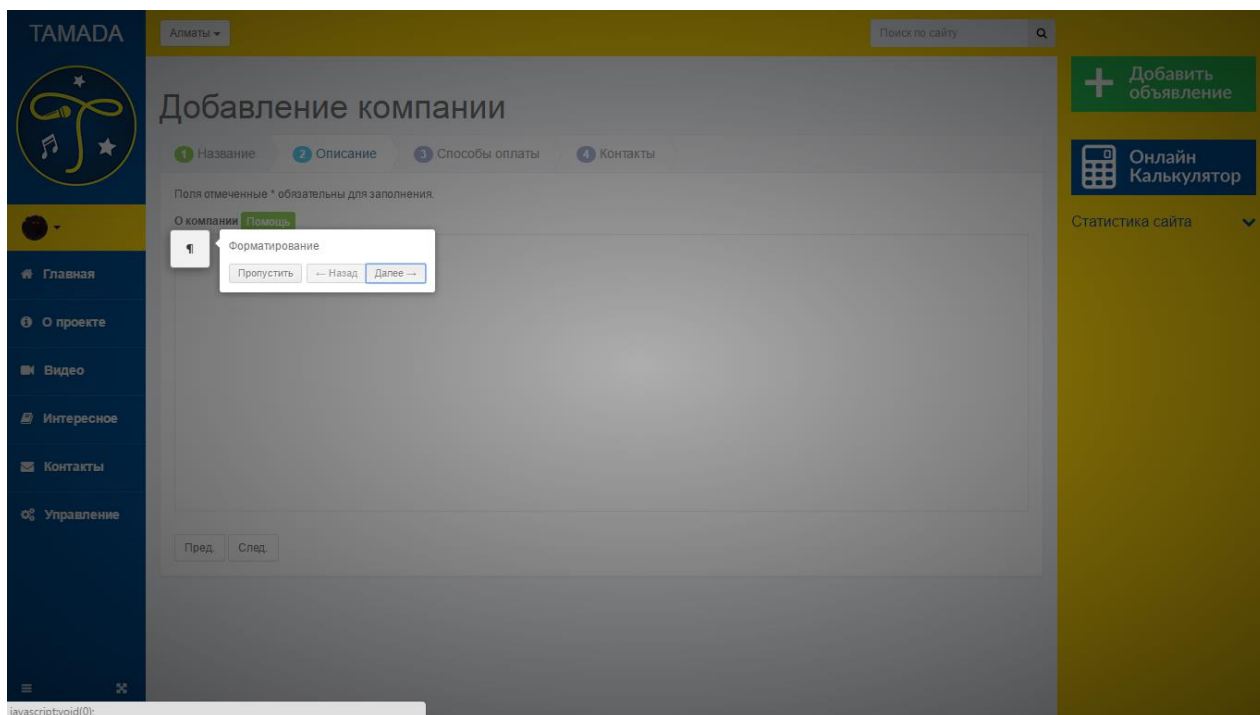


Рисунок 3.7 – Демонстрация интерактивного помощника

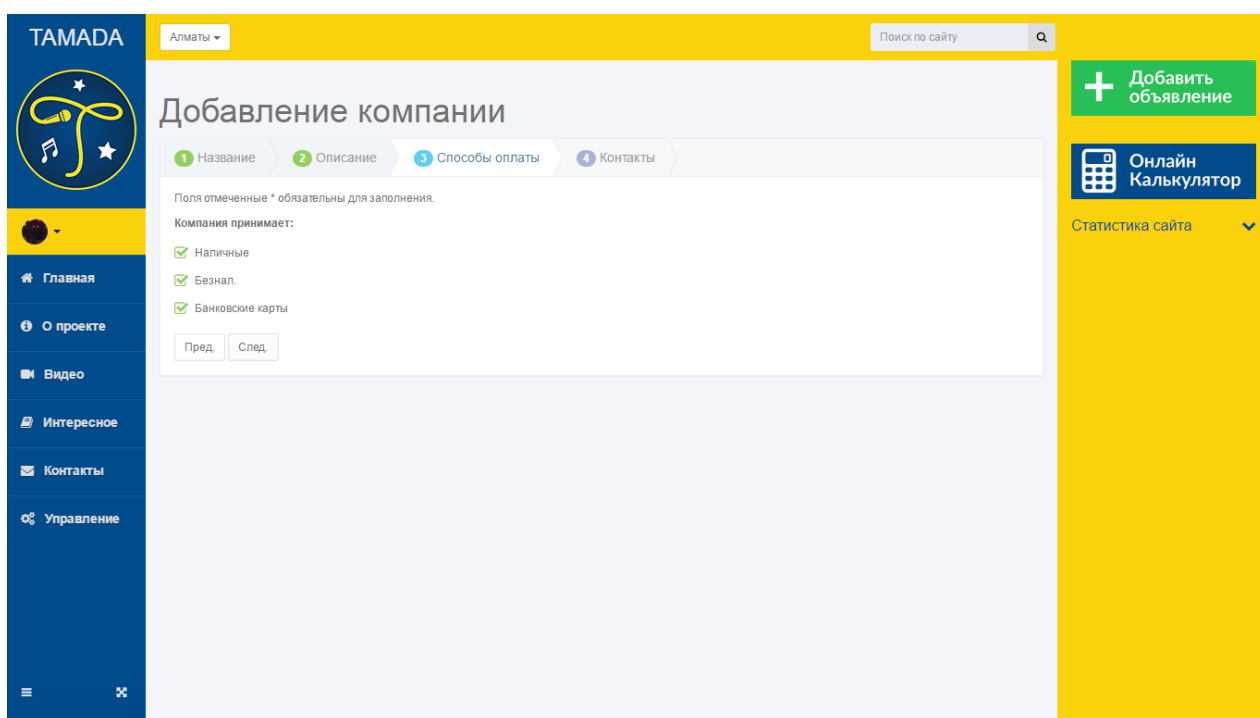


Рисунок 3.8 – Указание способов оплаты услуг, предоставляемых компанией

Также разработан помощник при указании адреса компании, автоматически ставящий маркер на карте.

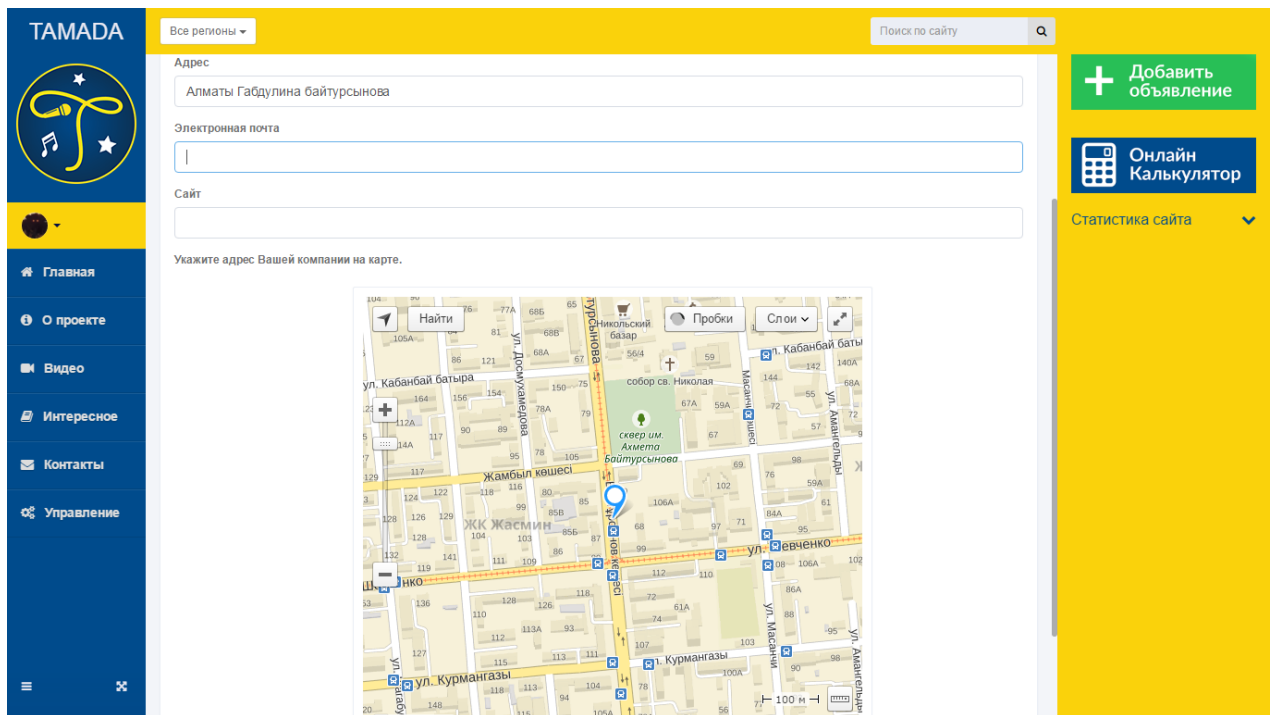


Рисунок 3.9 – Демонстрация геокодирования

После добавления компании, нас встречает модальное окно, предлагающее добавить объявление или продолжить работу на портале.

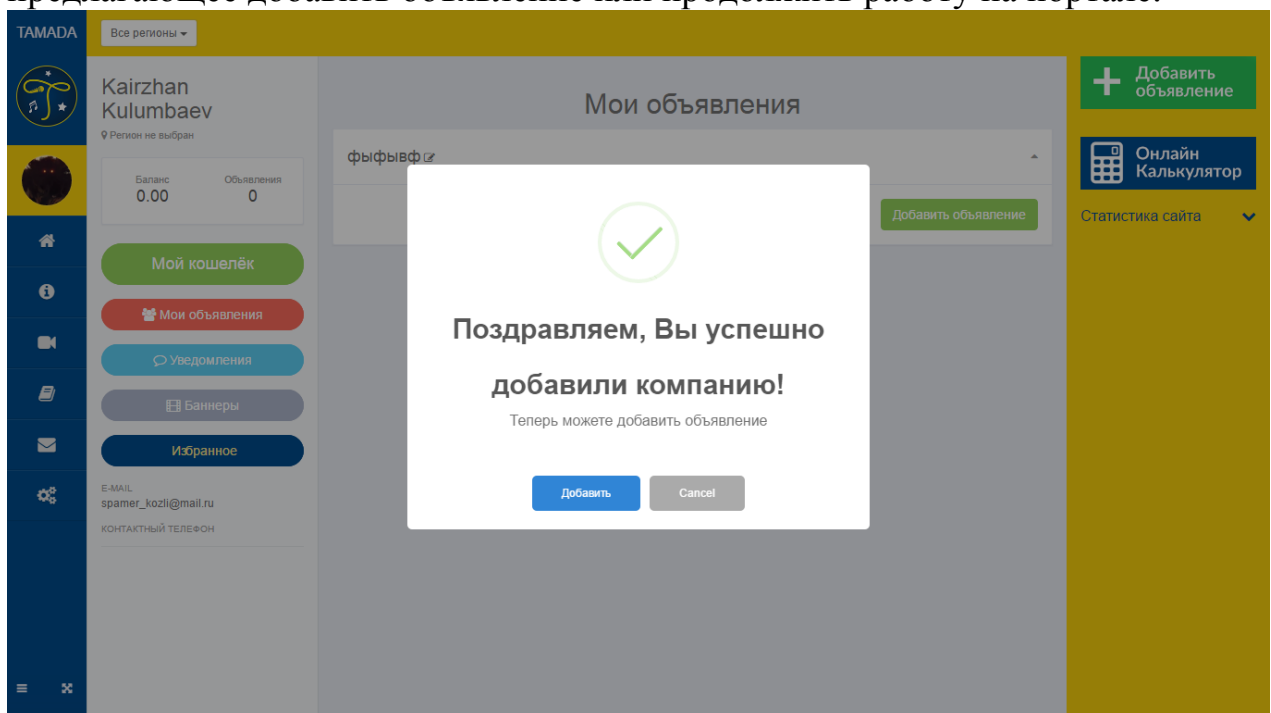


Рисунок 3.10 – Уведомление о создании компании

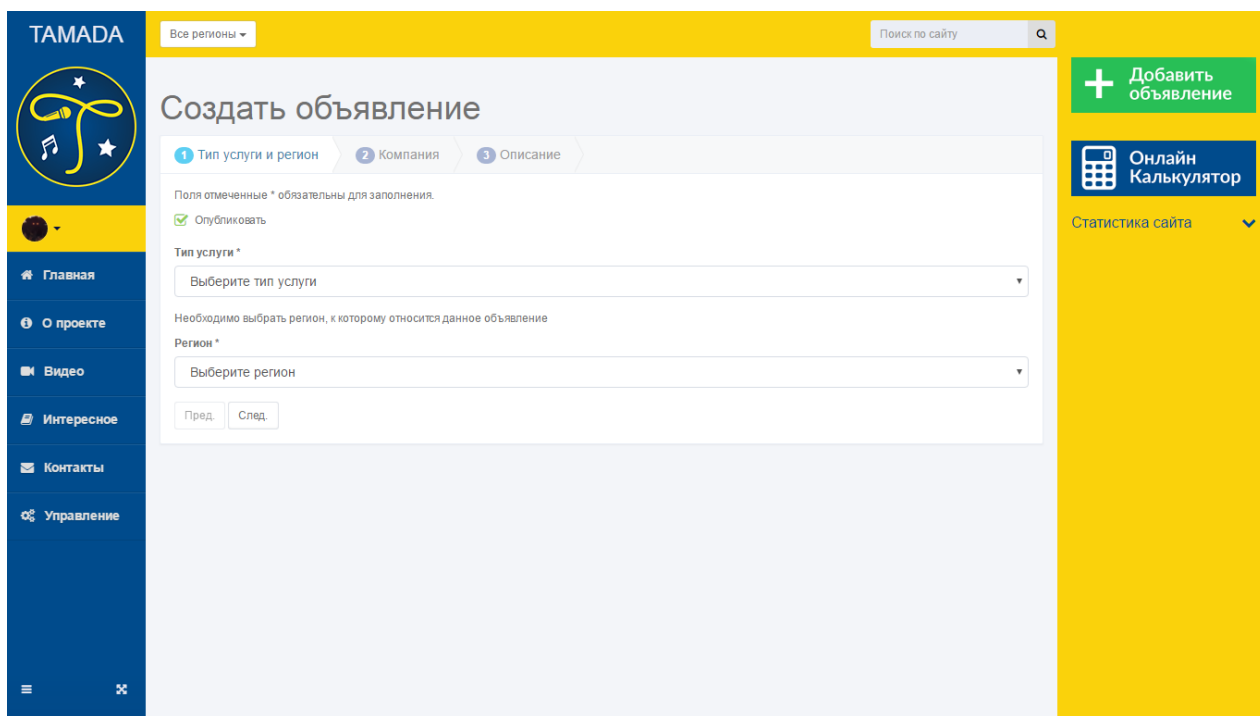


Рисунок 3.11 – Создание объявления

После создания объявления необходимо сделать еще одну операцию, это загрузка картинки и проверка данных объявления.

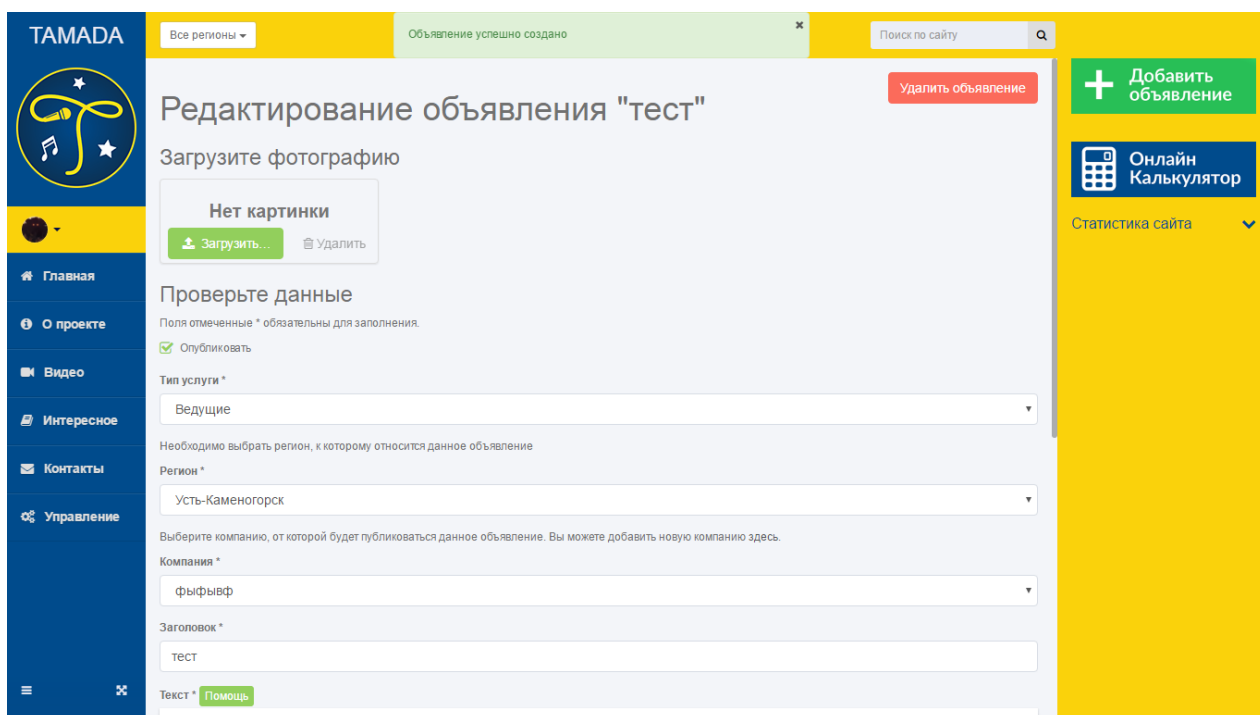


Рисунок 3.12 – Проверка данных объявления

Также реализована функция стор для изображений.

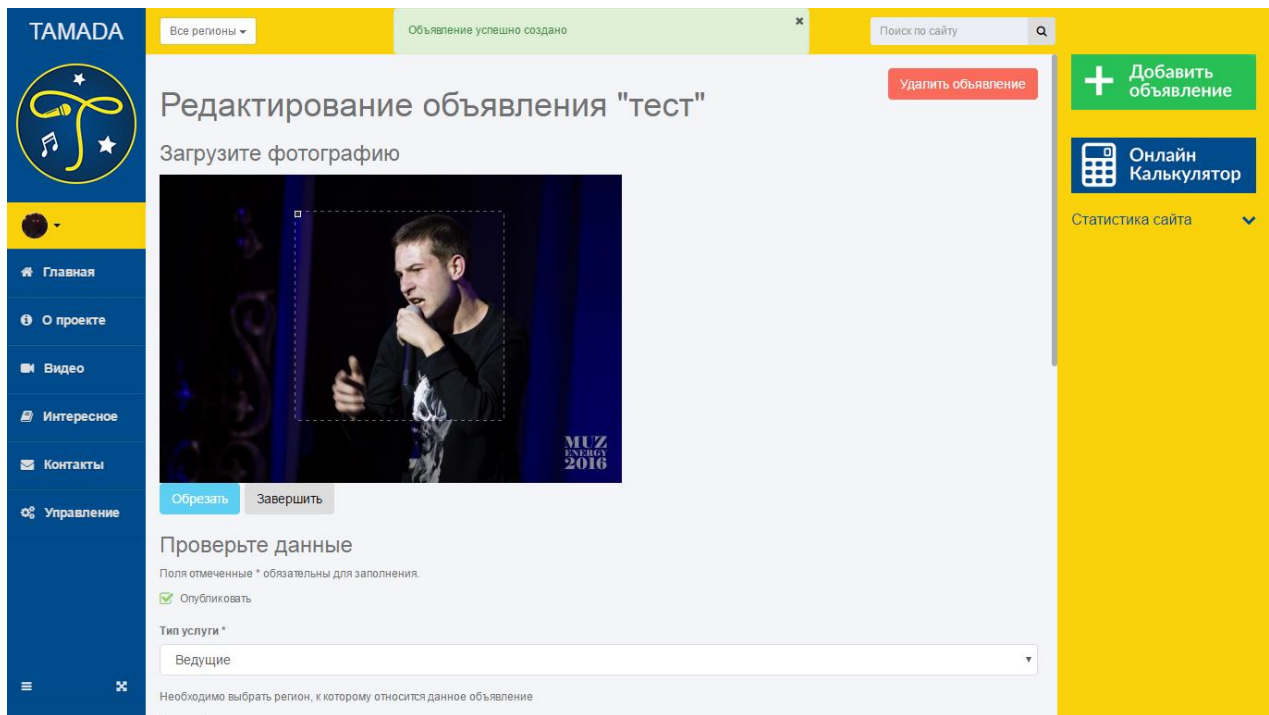


Рисунок 3.13 – Демонстрация функции стор

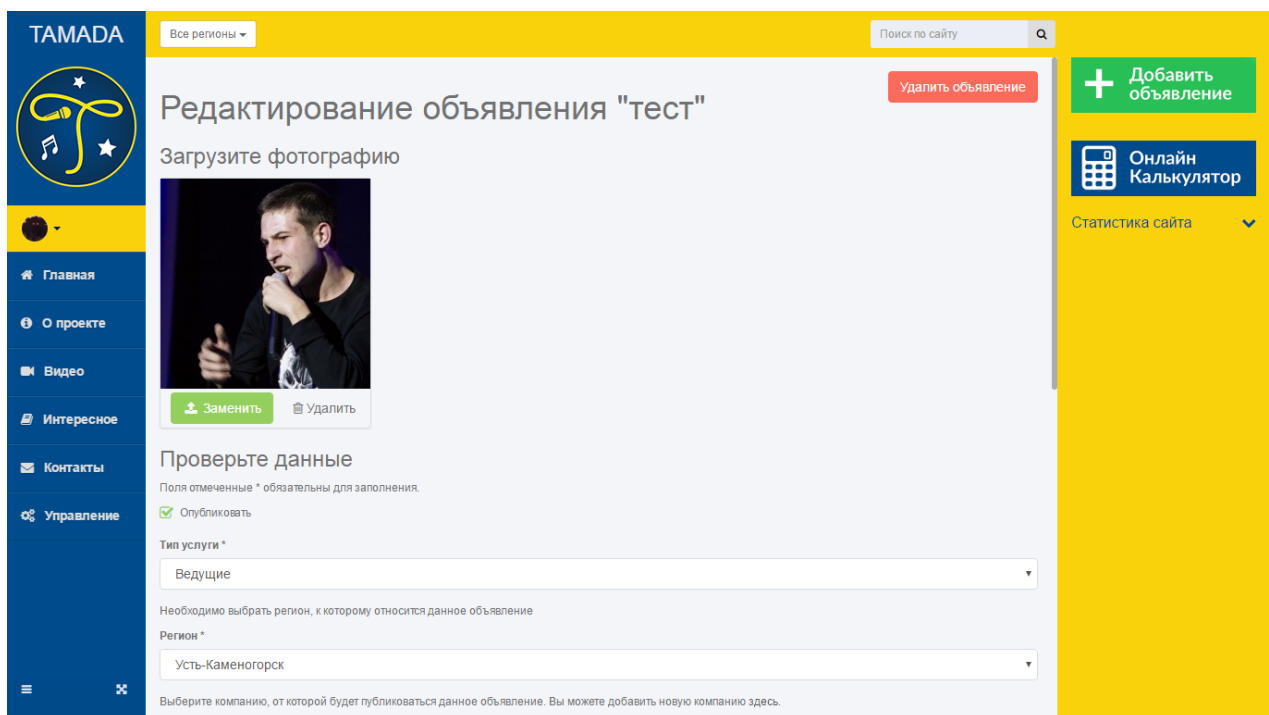


Рисунок 3.14 – Демонстрация функции стор

Теперь наше объявление сохранено и его могут увидеть все гости портала. Также можно выделить его цветом или поднять в топ (при поиске). Но это стоит денег.

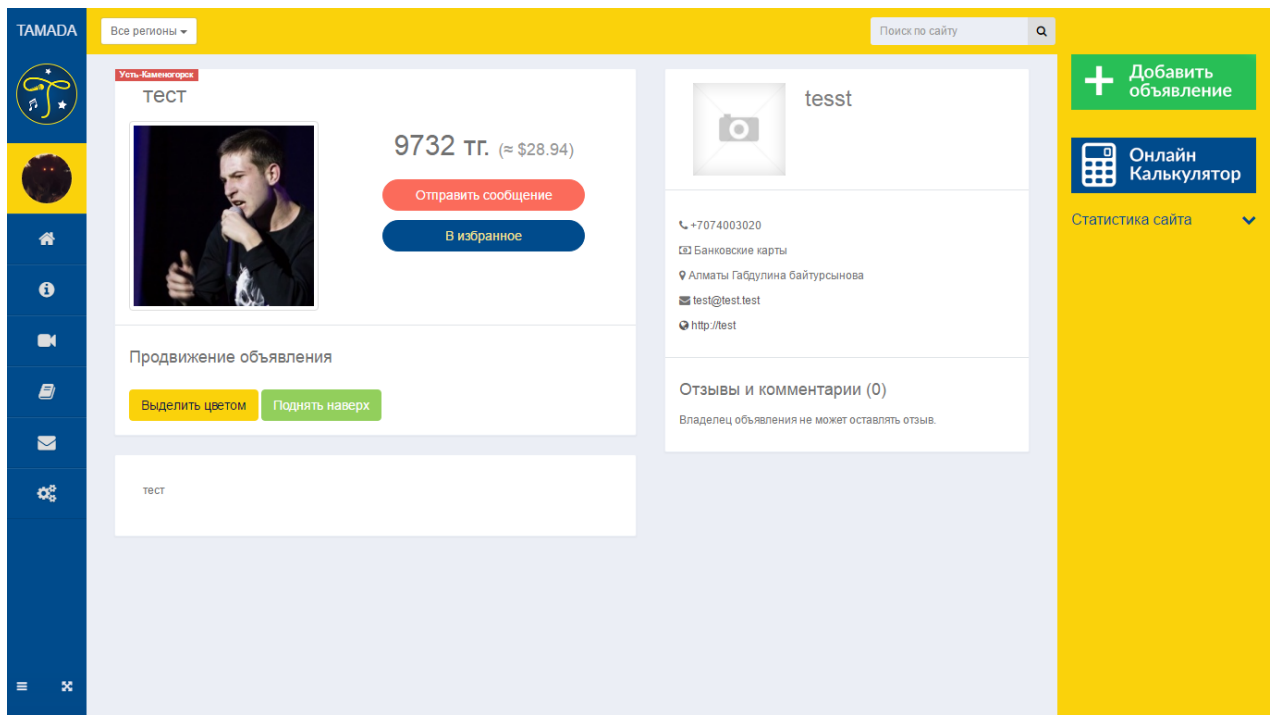


Рисунок 3.15 – Просмотр тестового объявления

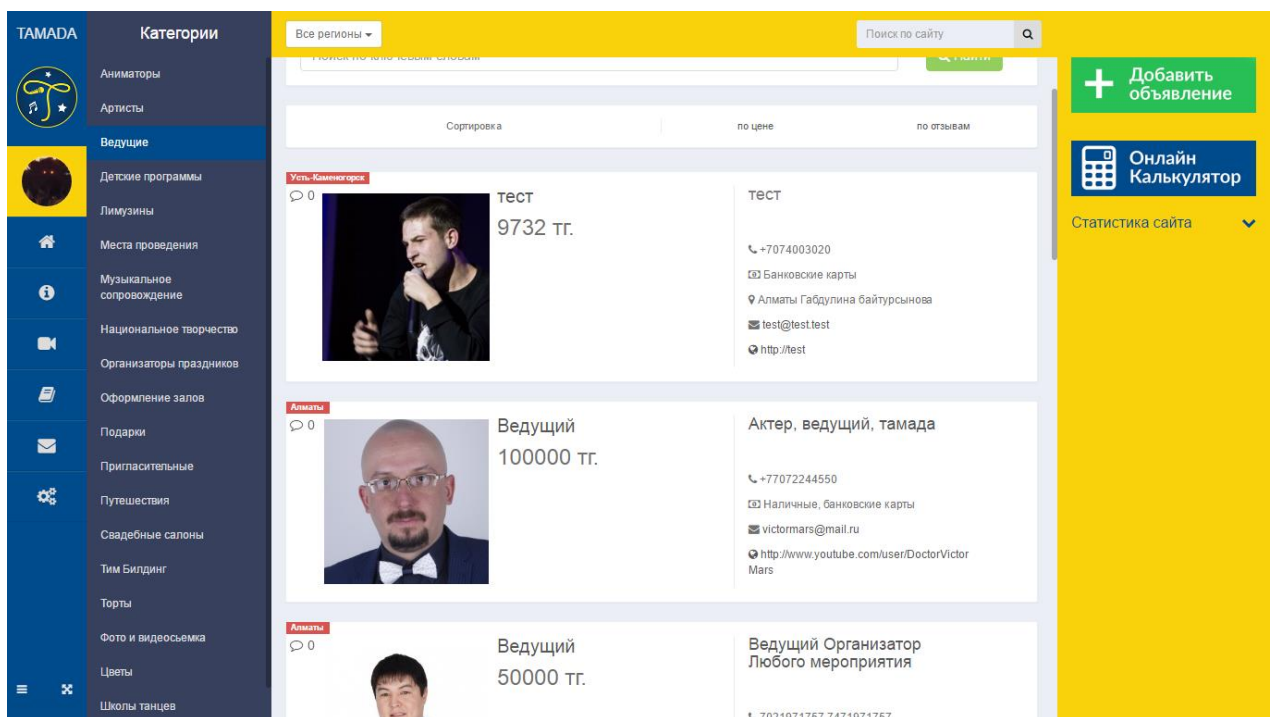


Рисунок 3.16 – Общий список объявлений в категории «Ведущие»

Также присутствует возможность фильтра по регионам, по ценам и по рейтингам. Также присутствует возможность сортировки и поиска по ключевым словам.

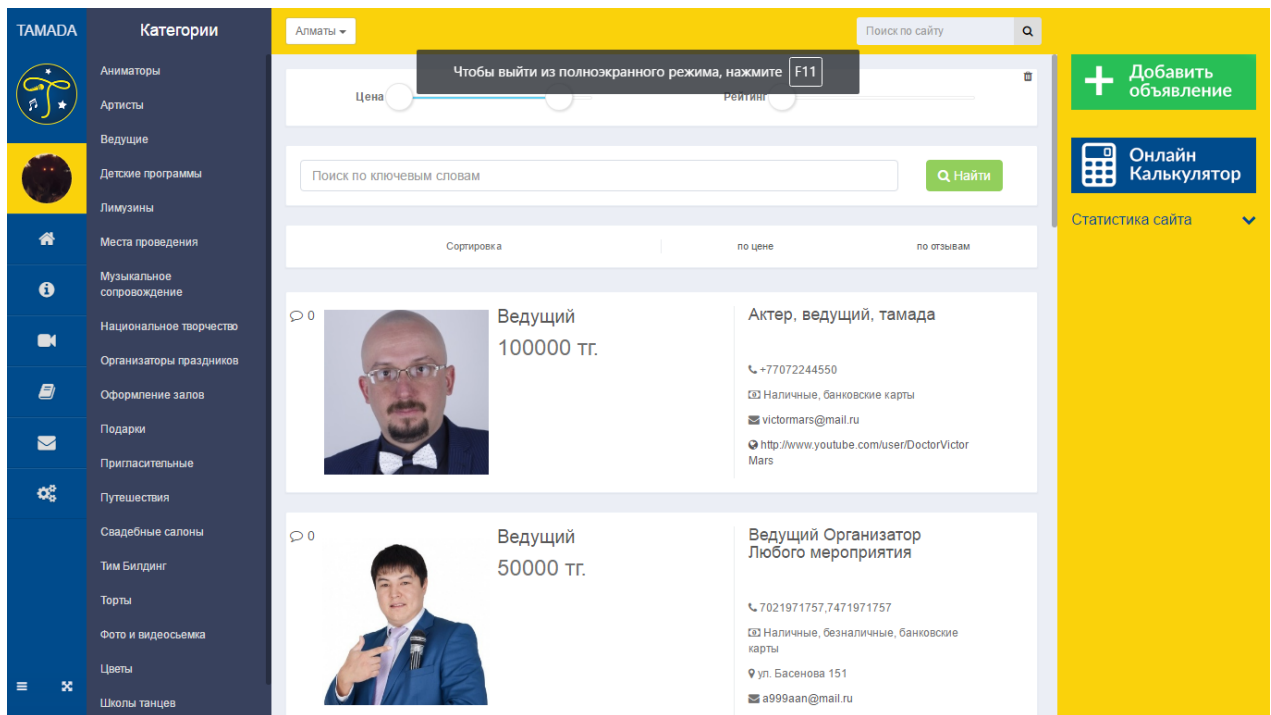


Рисунок 3.17 – Демонстрация фильтра по регионам

Поиск, по ключевым словам, проигнорирует фильтр по регионам.

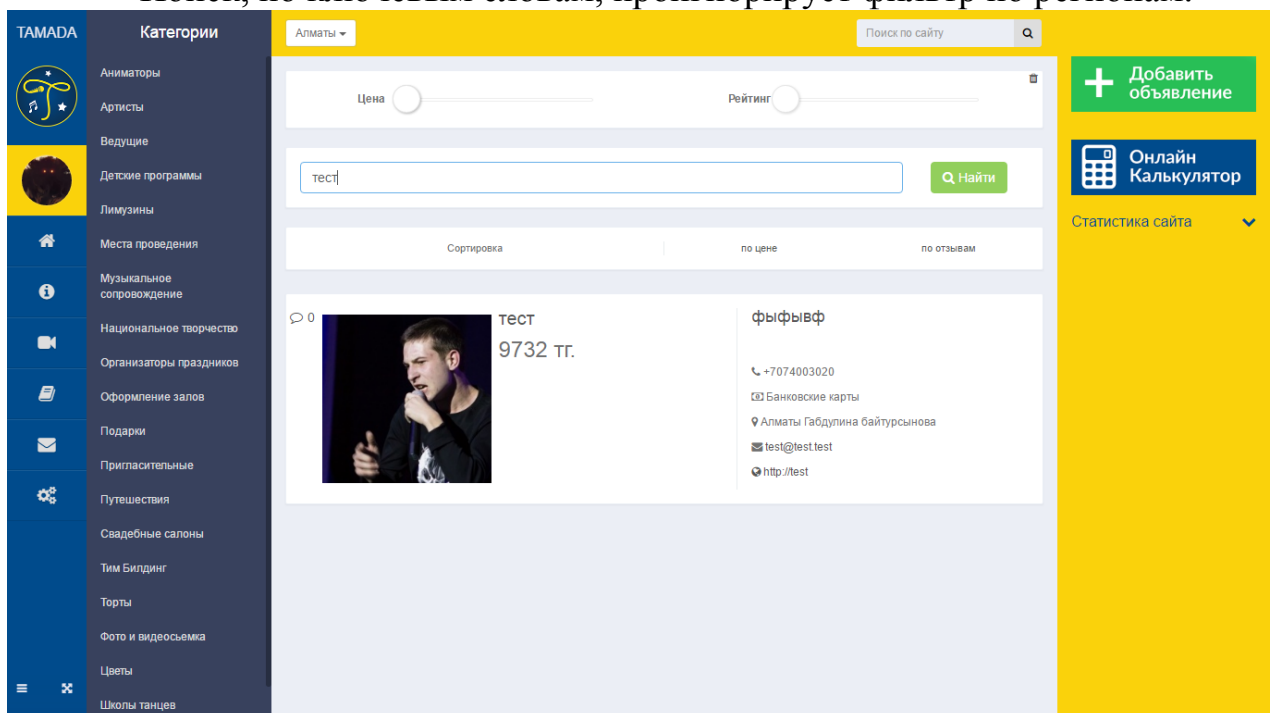


Рисунок 3.18 – Поиск по ключевым словам

Fulltext поиск не игнорирует фильтр по регионам.



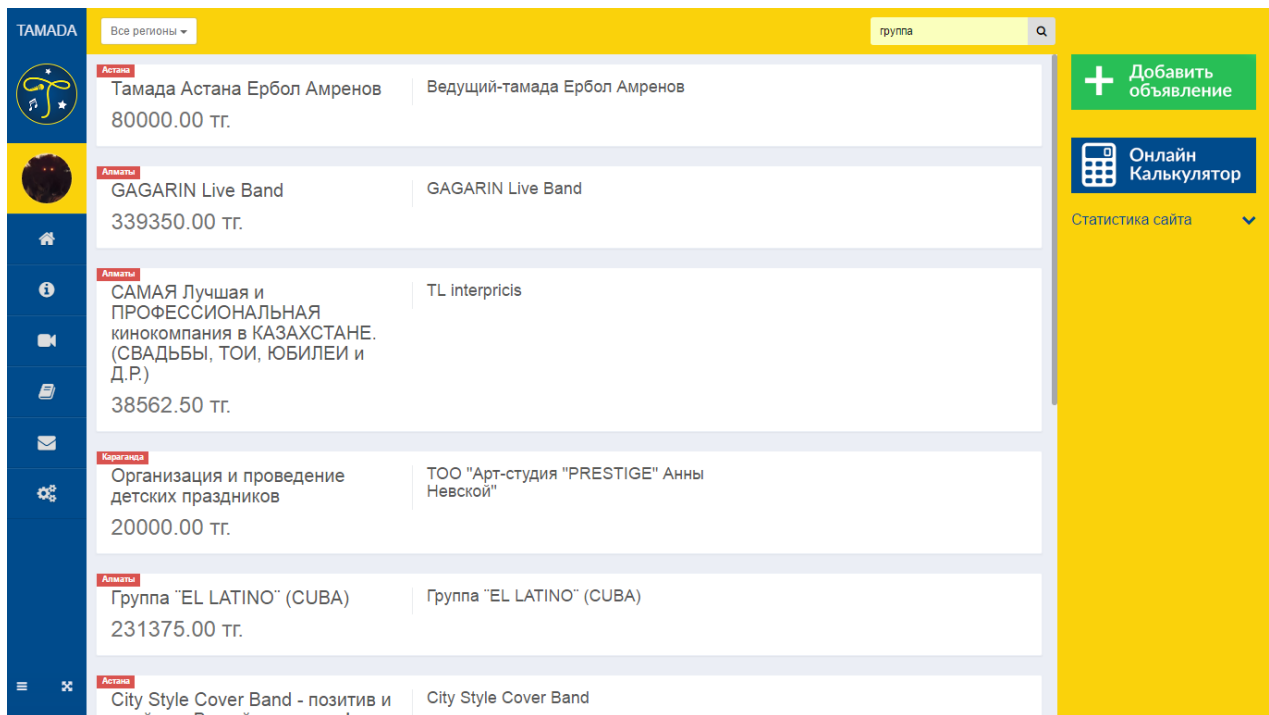


Рисунок 3.19 – Демонстрация fulltext поиска через data файлы

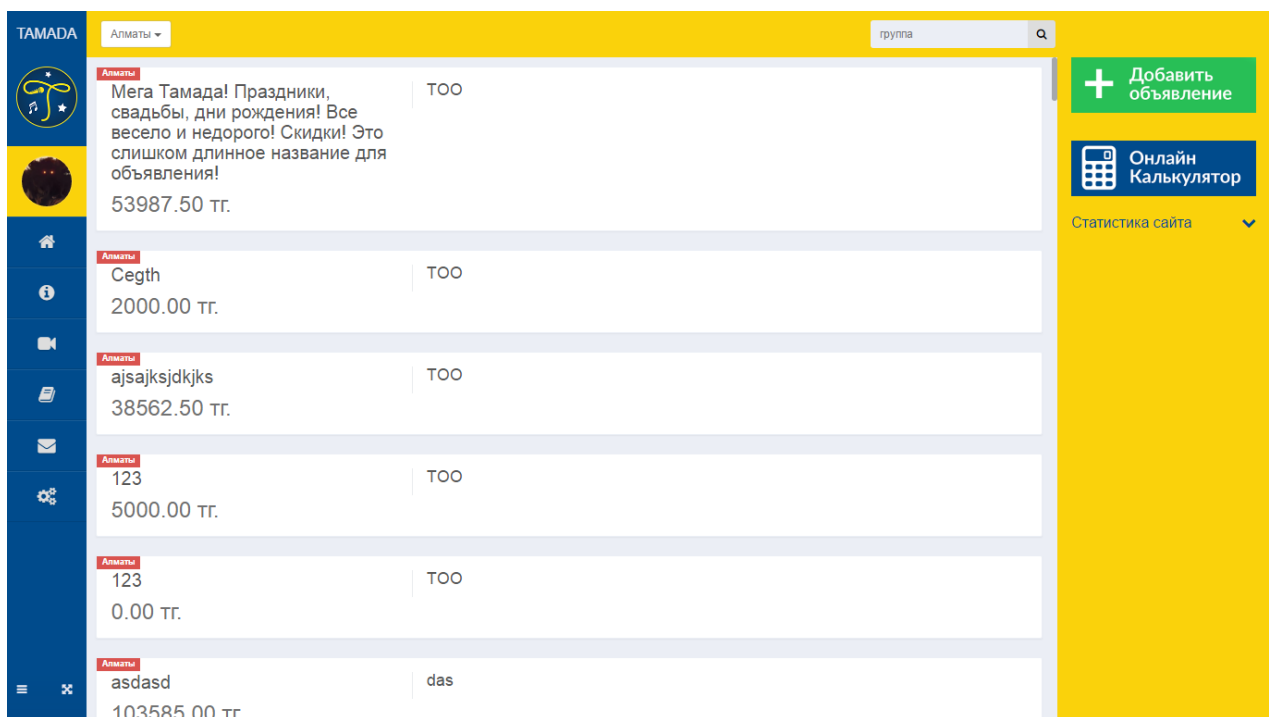


Рисунок 3.20 - Демонстрация fulltext поиска через data файлы с учетом фильтра по регионам

Показ административной панели будет минимальным, т.к. она запрещена к демонстрации.

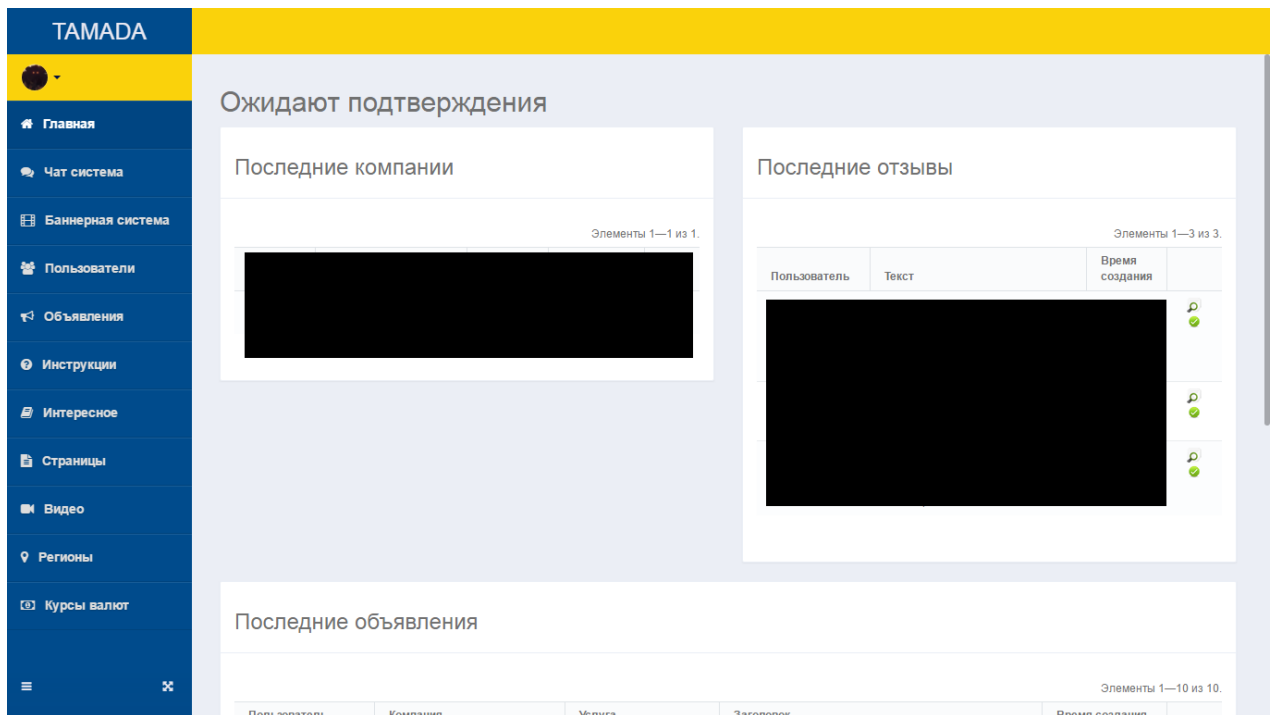


Рисунок 3.21 – Демонстрация административной панели

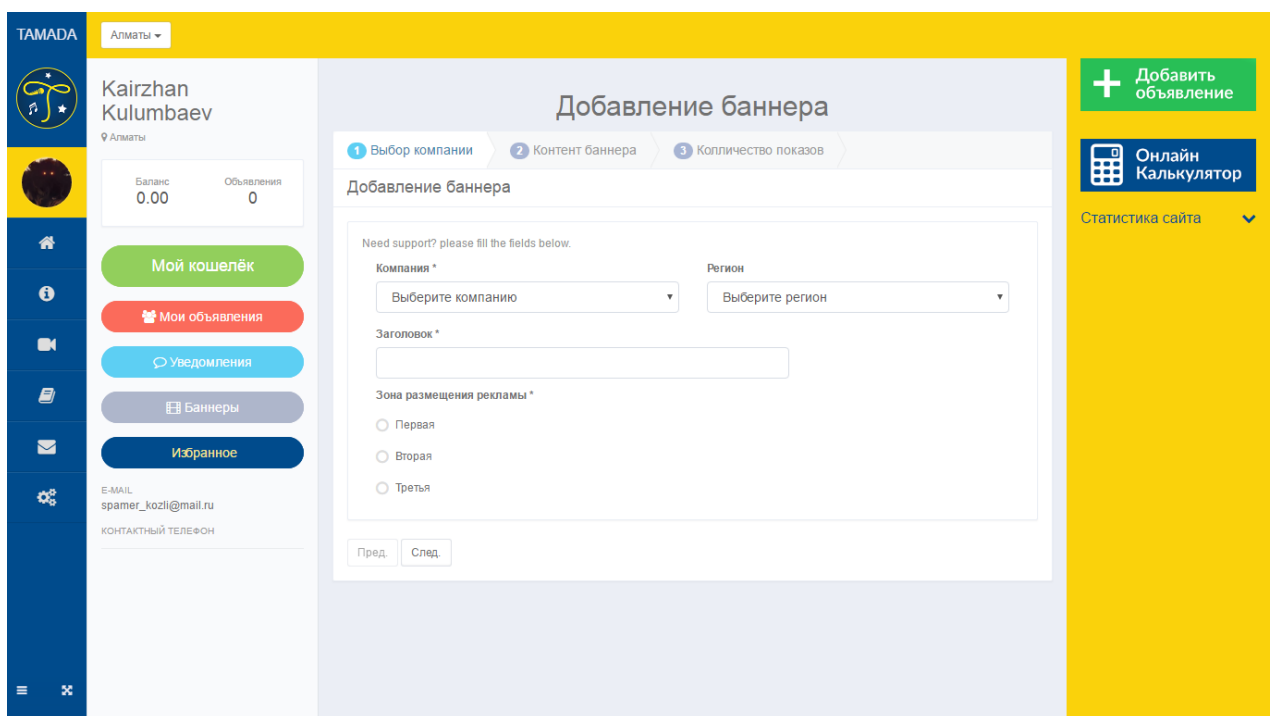


Рисунок 3.22 – Добавление баннера

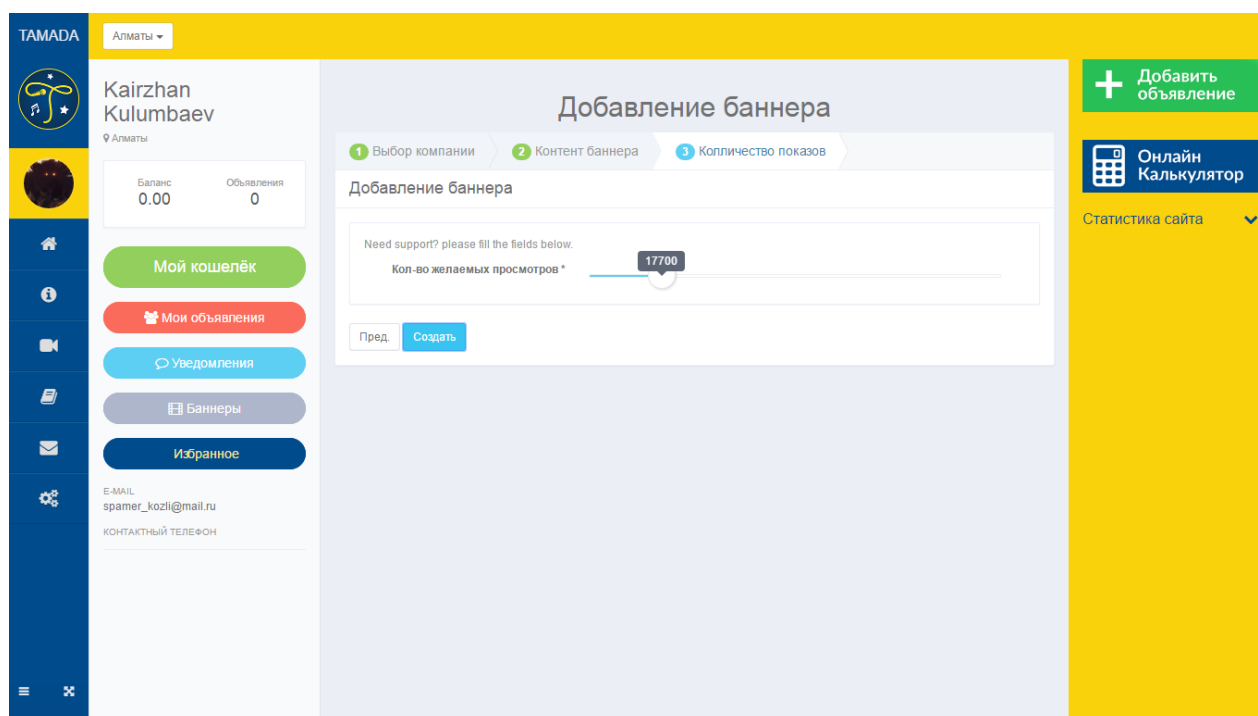


Рисунок 3.33 – Установка количества показа банера

После добавления банера, необходимо произвести оплату. Пополнение кошелька производится через систему терминалов – qiwі.

## 4 Техничко-экономическая часть

### 4.1 Цель проекта

Целью данного дипломного проекта является создание поисково-информационной системы (веб-портала) «Tamada.kz», которые создадут единую платформу для размещения и поиска информации по услугам в сфере организации мероприятий. В этом разделе дипломного проекта рассматривается экономическая часть, которая отражает трудовые, временные и финансовые затраты на данный проект.

### 4.2 Трудовые ресурсы, используемы в работе

В работе над данным проектом приняли участие:  
программист – программирование и разработка алгоритмов.

Количество сотрудников, задействованных в проекте, и их месячная заработная плата представлены в таблице 4.1.

Таблица 4.1 – Данные о сотрудниках

Должность	Количество	Зарботная плата в месяц
Программист	1	120 000
Итого	1	120 000

### 4.3 Оборудование, использованное в работе

Характеристики оборудования, используемого в работе, а также его стоимость приведены в таблице 4.2.

Таблица 4.2 – Перечень оборудования

Название оборудования	Характеристики	Количество	Стоимость за единицу без учета НДС, тенге
Ноутбук Acer v5-552g-10578g50akk	Intel Core i5-4200U, 8 Gb DDR3, 1000 Gb HDD, GeForce GT 740M	1	114400
Программное обеспечение	NetBeans IDE	1	Бесплатно
Итого		2	114400

### 4.4 Расчет затрат и стоимости работ по реализации проекта

Разработка веб-приложения требует большого количества интеллектуальных затрат сотрудников, выполняющих работу, а также необходимых технических средств для ее реализации. Все это требует финансовых вложений, на основе которых высчитывается конечная стоимость проекта.

Затраты на разработку данного приложения вычисляются по формуле:

$$C = \text{ФОТ} + C_n + A + P_a + \text{Э} + C_{\text{спр}} + H \quad (4.1)$$

где ФОТ – фонд оплаты труда;

$C_n$  – социальный налог;

$A$  – амортизационные отчисления;

$P_a$  – расходы на аренду ;

$\mathcal{E}$  – затраты на электроэнергию;

$C_{пр}$ – прочие расходы (использование интернета );

$H$  – накладные расходы.

#### 4.5.1 Расчет фонда оплаты труда

ФОТ складывается из основной и дополнительной заработной платы сотрудников и рассчитывается по формуле:

$$\text{ФОТ} = Z_{\text{осн}} + Z_{\text{доп}} \quad (4.2)$$

где  $Z_{\text{осн}}$  – основная заработная плата;

$Z_{\text{доп}}$  – дополнительная заработная плата.

Для расчета затрат на основную заработную плату используются данные о среднедневном заработке и фактическом времени работы каждого сотрудника.

Средний дневной заработок каждого работника рассчитывается по формуле:

$$D = \frac{Z_{\text{ПМ}}}{D_r} \quad (4.3)$$

где  $Z_{\text{ПМ}}$  – ежемесячный размер заработной платы;

$D_r$  – количество рабочих дней в месяце (21 день).

Программиста:

$$D = \frac{120000}{21} = 5714 \text{ тенге/день}$$

Заработная плата за один час работы сотрудника рассчитывается по формуле:

$$H = D / \text{Ч}_p \quad (4.4)$$

где  $D$  – средний дневной заработок работника;

$\text{Ч}_p$  – количество часов рабочего дня (8 часов).

Программист

$$H = 5714 / 8 = 714 \text{ тенге/час}$$

Длительность цикла в днях по каждому виду работ определяется по формуле:

$$t_n = \frac{T}{q_n * z * K} \quad (4.5)$$

где T – трудоемкость этапа, норма-час;

$q_n$  – количество исполнителей по этапу;

z – продолжительность рабочего дня, z = 8 часов;

K – коэффициент выполнения норм времени, K = 1,1.

Полученную величину  $t_n$  округляем в большую сторону до целых дней.

$$t_1 = \frac{16}{1*8*1,1} \approx 2 \text{ дня – Программист, постановка задачи;}$$

$$t_2 = \frac{24}{1*8*1,1} \approx 3 \text{ дня – Программист, изучение литературы;}$$

$$t_3 = \frac{16}{1*8*1,1} \approx 2 \text{ дня – Программист, выбор среды разработки;}$$

$$t_4 = \frac{112}{1*8*1,1} \approx 14 \text{ дня – Программист, разработка форм приложения;}$$

$$t_5 = \frac{16}{1*8*1,1} \approx 3 \text{ дня – Программист, тестирование приложения;}$$

$$t_6 = \frac{16}{1*8*1,1} \approx 2 \text{ дня – Программист, отладка приложения;}$$

$$t_7 = \frac{16}{1*8*1,1} \approx 1 \text{ дня – Программист, оформление и сдача отчета.}$$

В таблице 4.3 приведены сводные результаты расчета затрат на основную заработную плату сотрудников.

Таблица 4.3 – Сводные результаты расчета затрат на основную заработную плату.

Наименование этапов работ	Исполнитель	Трудоемкость		Длительность цикла, дни	З-плата за часы работ, тенге	Сумма зар. платы, тенге
		Нормы-часы	% от общей трудоемкости			
Постановка задачи	Программист	16	10	2	714	11424
Изучение литературы	Программист	24	10	3	714	17136

Выбор среды разработки	Программист	16	5	2	714	11424
Разработка форм приложения	Программист	112	50	14	714	79968
Тестирование приложения	Программист	24	10	3	714	17136
Отладка приложения	Программист	16	10	2	714	11424
Оформление и сдача отчета	Программист	8	5	1	714	5714
Итого		216	100	27		154226

Таким образом, согласно произведенным расчётам основная заработная плата составляет 154226 тенге.

Дополнительная заработная плата составляет 10% от основной заработной платы и рассчитывается по формуле:

$$Z_{\text{доп}} = Z_{\text{осн}} * 0,1 \quad (4.6)$$

$$Z_{\text{доп}} = 154226 * 0,1 = 15422 \text{ тенге}$$

В результате расчетов, согласно формуле 4.2, суммарный фонд оплаты труда составит:

$$\text{ФОТ} = 154226 + 15422 = 169648 \text{ тенге}$$

#### 4.5.2 Расчет затрат по социальному налогу

Социальный налог составляет 11% от дохода сотрудника и рассчитывается по формуле:

$$C_{\text{н}} = (\text{ФОТ} - \text{ПО}) * 11\% \quad (4.7)$$

где ПО – пенсионные отчисления, который составляют 10% от ФОТ и не облагаются социальным налогом, рассчитываются по формуле

$$ПО = ФОТ * 10\% \quad (4.8)$$

$$ПО = 169648 * 0,1 = 16965 \text{тенге}$$

Таким образом социальный налог составит

$$Сн = (169648 - 16965) * 0,11 = 16795 \text{ тенге}$$

#### 4.5.3 Расчет амортизационных отчислений

Амортизационные отчисления рассчитываются по формуле:

$$Ai = \frac{N_A * C_{пер} * N}{100 * 12 * n} \quad (4.9)$$

где  $N_A$  – норма амортизации;

$C_{пер}$  – первоначальная стоимость оборудования;

$N$  – количество дней на выполнение работ;

$n$  – количество рабочих дней в месяце.

Следовательно, амортизационные отчисления по используемому оборудованию и ПО, в соответствии с формулой 9 составят

На оборудование:

$$A = \frac{40 * 114400 * 27}{100 * 12 * 21} = 4903 \text{ тенге}$$

#### 4.5.4 Расчет затрат на электроэнергию

Так как в процессе реализации проекта используется техническое оборудование, необходимо рассчитать затраты на электроэнергию, потребляемую данным оборудованием.

Для расчета затрат на электроэнергию используется формула 10.

$$\mathcal{E} = \mathcal{E}_{эл.эн.об} + \mathcal{E}_{доп} \quad (4.10)$$

где  $\mathcal{E}_{эл.эн.об}$  – затраты на электроэнергию для оборудования;

$\mathcal{E}_{доп}$  – затраты электроэнергии на дополнительные нужды.

Расходы электроэнергии на оборудование рассчитываются по формуле

4.11

$$\mathcal{E}_{эл.эн.об} = W * T * S * K_{исп} \quad (4.11)$$

где  $W$  – потребляемая оборудованием мощность, кВт;

$T$  – время работы, часы;

$S$  – тариф (1кВт/час = 16,9 тенге);

$K_{исп}$  – коэффициент использования ( $K_{исп} = 0,9$ ).

Потребляемая мощность Acer TravelMate 7750G-32314G50Mnss составляет 90 Вт.



Время высчитывается на основе количества рабочих дней и рабочих часов в день.

Таким образом общая сумма затрат на электроэнергию для оборудования:

$$\text{Зэл. эн. об} = 0,09 * 27 * 8 * 16,9 * 0,9 = 295,6 \text{ тенге}$$

Затраты на дополнительные нужды берутся в размере 5% от затрат на оборудование и рассчитываются по формуле:

$$\text{Здоп} = \text{Зэл. эн. об} * 5\% \quad (4.12)$$

И составляют

$$\text{Здоп} = 295,6 * 0,05 = 14,7 \text{ тенге}$$

Суммарные затраты на электроэнергию составляют

$$\text{Э} = 295,6 + 14,7 = 310,3 \text{ тенге}$$

#### 4.5.5 Расчет накладных расходов

Накладные расходы рассчитываются в размере 50% от всех затрат.

$$\text{Н} = (\text{ФОТ} + \text{Ос} + \text{А} + \text{Э}) * 50\% \quad (4.13)$$

$$\text{Н} = (169648 + 16795 + 4903 + 310,3) * 0,5 = 95828 \text{ тенге}$$

#### 4.5.6 Суммарные затраты на реализацию проекта

Таким образом себестоимость разработки данного приложения, согласно формуле 1 составляет (где  $P_a = 50\,000$  тг,  $C_{пр} = 2000$  тенге):

$$\begin{aligned} C &= 169648 + 16795 + 4903 + 50000 + 310,3 + 2000 + 95828 \\ &= 339484,3 \text{ тенге} \end{aligned}$$

Сводные результаты расчета стоимости разработки приложения и их структура представлены на рисунке 1 и в таблице 4:

Таблица 4.4 – Затраты на разработку приложения

Наименование затрат	Сумма, тенге
ФОТ	169648
Социальный налог	16795
Амортизационные отчисления	4903
Аренда помещения	50000
Затраты на электроэнергию	310,3
Прочие расходы	2000
Накладные расходы	95828
Итого	339484,3



Рисунок 4.1 – Затраты на разработку приложения

#### 4.5.7 Цена реализации проекта

Цена проекта складывается из себестоимости и желаемого чистого дохода.

$$Цп = С + П \quad (4.14)$$

Где С – стоимость приложения;

П – чистый доход.

Для определения начальной цены используется желаемый уровень рентабельности. Для данной отрасли он составляет 25%.

$$Цп = С * \left(1 + \frac{P}{100}\right) \quad (4.15)$$

где P – рентабельность.

$$Цп = 339484,3 * \left(1 + \frac{25}{100}\right) = 509226,5 \text{ тенге}$$

Цена реализации проекта складывается из цены проекта и НДС, рассчитывается по формуле

$$Цр = Цп + НДС \quad (4.16)$$

где НДС – налог на добавленную стоимость.

Согласно Налоговому Кодексу Республики Казахстан НДС составляет 12%, то есть в данном случае равен

$$НДС = 61107 \text{ тенге.}$$

В итоге получаем цену реализации проекта равной

$$Цр = 509226,5 + 61107 = 570333,5 \text{ тенге}$$

## **5 Безопасность жизнедеятельности**

### **5.1 Анализ условий труда**

Работа с компьютером она характеризуется значительным умственным напряжением и операторов стресса нервно-эмоционального, высокой интенсивности зрительной работы и достаточно большой нагрузкой на мышцы рук при работе с клавиатурой компьютера. Большое значение имеет рациональное дизайн и оформление рабочего места, что важно для поддержания оптимального рабочего положения человека-оператора.

В процессе работы с компьютером необходимо соблюдать правильный режим труда и отдыха. В соответствии закону РК о «Безопасности и охране труда» от 28 февраля 2004 г. 528-ІІ ЗРК, закона о труде в РК от 10.12.99 г. 493-І, закона о промышленной безопасности на опасных производственных объектах от 3.4.02 г. 3/ч-ІІ ЗРК. Согласно закону, в компании должна быть предусмотрены меры по обеспечению выполнения закона. В противном случае, персонал отмечено значительное напряжение зрительного аппарата с появлением жалоб на неудовлетворенность, головные боли, раздражительность, нарушение сна, усталость и боль в глазах, в нижней части спины, шеи и рук.

Эргономические аспекты проектирования видеотерминальных рабочих мест, в частности, являются: высота рабочей поверхности, размеры пространства требования ног пространство документов на рабочем месте (наличие и размер подставки документов, возможность различного размещения документов, расстояние от глаз пользователя до экрана, документа, клавиатуры и т.д.), характеристики рабочего кресла, требования к поверхности рабочего стола, регулируемость элементов рабочего места.

Основные элементы рабочего пространства программиста стол и стул. Основным рабочим положением является положение сидя.

Рабочая поза сидя вызывает минимальное утомление программиста. Рациональное расположение рабочего места предусматривает четкий порядок и размещение постоянства объекта, ручных инструментов и документации. То, что требуется для выполнения работ чаще, расположено в зоне легкой досягаемости рабочего пространства.

Существенное значение для продуктивной и качественной работы на компьютере имеют размеры знаков, плотность их размещения, коэффициент контрастности и яркости символов и фона экрана. Если расстояние от глаз до экрана дисплея оператора составляет 60 ... 80 см, высота знака должна составлять не менее 3 мм, оптимальное соотношение ширины и высоты знака составляет 3: 4, и расстояние между метками - 15 ... 20% от их высоты. Соотношение яркости фона экрана и символов - от 1: 2 до 1:15.

Во время использования компьютера, врачи советуют устанавливать монитор на расстоянии 50-60 см от глаз. Эксперты также считают, что верхняя часть видеодисплея должна быть на уровне глаз или чуть ниже. Когда человек смотрит прямо перед собой, его глаза открыты шире, чем когда он смотрит вниз. В этой области просмотр значительно возрастает, вызывая обезвоживание глаз. Кроме того, если экран установлен высоко, а глаза широко открыты, нарушается функция моргания. Это означает, что глаз не закрывается полностью, не получают достаточного количества влаги, что приводит к усталости.

Создание благоприятных условий труда и правильное эстетическое проектирования рабочих мест в производстве имеет большое значение как для облегчения труда и повышения его привлекательности, положительно влияет на производительность.

Кондиционирование воздуха должно обеспечивать автоматическое поддержание параметров микроклимата в необходимых пределах в течение всех сезонов года, очистку воздуха от пыли и от вредных частиц, создание небольшого избыточного давления в чистых помещениях для исключения поступления неочищенного воздуха. Предусматривается, также, возможность индивидуальной регулировки раздачи воздуха в отдельных помещениях. Температура подаваемого в помещение воздуха должна быть не ниже 18<sup>0</sup>С [СНиП 11-33-75 «Отопление, вентиляция и кондиционирование воздуха»].

Таблица 1 – нормы температуры, относительной влажности, скорости движения воздуха в рабочей зоне помещения [Санитарное законодательство РК (Санитарные нормы и правила по гигиене труда в промышленности)].

Таблица 5.1 – Таблица отражающая микроклимат

Температура наружного воздуха, С°	Параметры воздушной среды на постоянном рабочем месте					
	Оптимальные			Допустимые		
	Температура, С°	Относительная влажность, %	Скорость движения не более, м/с	температура, С°	Относительная влажность, %	Скорость движения не более, м/с
Ниже +10	20-22	40-60	0.2	18-22	не более 70	0.2

## 5.2 Метеорологические условия, вентиляция и кондиционирование воздуха.

Метеорологические условия - сочетание температуры, относительной влажности и скорости движения воздуха. Установлены нормы производственного микроклимата:

- в холодное время года температура воздуха 22-24<sup>0</sup>С, относительная влажность 60-40%, скорость движения воздуха 0,1м/с.
- в теплое время года температура воздуха 23-25<sup>0</sup>С, относительная влажность 60-40%, скорость движения воздуха 0,1м/с-0,2м/с.

Воздух в помещениях должен быть очищен от вредных веществ, пыли и микроорганизмов. Патогенная флора должна быть исключена. Общее количество колоний на 1 м<sup>3</sup> не должна превышать 1000. Воздух рабочей зоны должен соответствовать установленным требованиям с незначительным избытком тепла от видеотерминалов и устройств отображения информации.

## 5.3 Расчет уровня шума

Одним из неблагоприятных факторов окружающей среды является высокий уровень шума, создаваемого печатающих устройств, оборудования для кондиционирования воздуха, вентиляторы охлаждения в самих компьютерных системах.

Для того, чтобы рассмотреть вопрос о необходимости и целесообразности снижения шума необходимо знать уровни шума на рабочем месте оператора.

Уровень шума, возникающий от нескольких некогерентных источников, работающих одновременно, подсчитывается на основании принципа энергетического суммирования излучений отдельных источников

$$L_{\Sigma} = 10 \lg \sum_{i=1}^{i=n} 10^{0,1L_i} \quad (5.1)$$

где  $L_i$  - уровень звукового давления  $i$ -го источника шума;

$n$  - количество источников шума.

Полученные результаты расчета сравниваются с допустимым значением шума для данного рабочего места. Если результаты расчета выше допустимых уровней шума, необходимость принятия специальных мер для снижения уровня шума. К ним относятся: облицовка стен и потолка зала звукопоглощающими материалами, уменьшение шума в источнике, правильная планировка оборудования и рациональная организация рабочего места оператора.

Уровни звукового давления источников шума, действующих на оператора на его рабочем месте представлены в табл. 2

Как правило, станция оператора оснащено следующим оборудованием: жесткий диск в системном блоке, вентилятор (ы) охлаждения ПК, монитор, клавиатуру, и получил тер сканера.

Подставляя значения уровня звукового давления для каждого вида оборудования в формулу, получим:

$$L_{\Sigma} = 10 * \lg(104 + 104.5 + 101.7 + 101.7 + 101 + 104.5 + 104.2) + 10 \lg 6 = 52.28 \text{ дБ} \quad (5.2)$$

Полученное значение не превышает допустимый уровень шума для рабочего места оператора, равный 65 дБ (ГОСТ 12.1.003-83). И если мы считаем, что вряд ли такие периферийные устройства как сканер и принтер будут использоваться одновременно, эта цифра будет еще ниже. Кроме того, если принтер находится непосредственно следствии оператор-присутствие обязательно, так как принтер оснащен механизмом автоматической листов.

#### 5.4 Расчет вентиляции офиса

Системы отопления и системы кондиционирования следует устанавливать так, чтобы ни теплый, ни холодный воздух не направлялся на людей. На производстве рекомендуется создавать динамический климат с определенными перепадами показателей. Температура воздуха у поверхности пола и на уровне головы не должна отличаться более, чем на 5 градусов. В производственных помещениях помимо естественной вентиляции предусматривают приточно-вытяжную вентиляцию. Основным параметром, определяющим характеристики вентиляционной системы, является кратность обмена, т.е. сколько раз в час сменится воздух в помещении.

Расчет для помещения

$V_{\text{вент}}$  - объем воздуха, необходимый для обмена;

$V_{\text{пом}}$  - объем рабочего помещения.

Для расчета примем следующие размеры рабочего помещения:

длина  $B = 7.35$  м;

ширина  $A = 4.9$  м;

высота  $H = 4.2$  м.

Соответственно объем помещения равен:

$$V_{\text{помещения}} = A * B * H = 151,263 \text{ м}^3 \quad (5.3)$$

Необходимый для обмена объем воздуха  $V_{\text{вент}}$  определим исходя из уравнения теплового баланса:

$$V_{\text{вент}} * C (t_{\text{уход}} - t_{\text{приход}}) * \gamma = 3600 * Q_{\text{избыт}} \quad (5.4)$$

$Q_{\text{избыт}}$  - избыточная теплота (Вт)

$C = 1000$  - удельная теплопроводность воздуха (Дж/кгК)

$\gamma = 1.2$  - плотность воздуха (мг/см).

Температура уходящего воздуха определяется по формуле:

$$t_{\text{уход}} = t_{\text{р.м.}} + (H - 2)t, \text{ где} \quad (5.5)$$

$t = 1-5$  °С - превышение  $t$  на 1м высоты помещения;

$t_{\text{р.м.}} = 25$  °С - температура на рабочем месте;

$H = 4.2$  м - высота помещения;

$t_{\text{приход}} = 18$  °С.

$t_{\text{уход}} = 25 + (4.2 - 2) \cdot 2 = 29.4$

$$Q_{\text{избыт}} = Q_{\text{изб.1}} + Q_{\text{изб.2}} + Q_{\text{изб.3}}, \text{ где} \quad (5.6)$$

$Q_{\text{изб.1}}$  - избыток тепла от электрооборудования и освещения.

$$Q_{\text{изб.1}} = E \cdot p, \text{ где} \quad (5.7)$$

$E$  - коэффициент потерь электроэнергии на теплоотвод (  $E=0.55$  для освещения);

$p$  - мощность,  $p = 40 \text{ Вт} \cdot 15 = 600 \text{ Вт}$ .

$Q_{\text{изб.1}} = 0.55 \cdot 600 = 330 \text{ Вт}$

$Q_{\text{изб.2}}$  - теплоступление от солнечной радиации,

$$Q_{\text{изб.2}} = m \cdot S \cdot k \cdot Q_c, \text{ где} \quad (5.8)$$

$m$  - число окон, примем  $m = 4$ ;

$S$  - площадь окна,  $S = 2.3 \cdot 2 = 4.6 \text{ м}^2$ ;

$k$  - коэффициент, учитывающий остекление. Для двойного остекления

$k = 0.6$ ;

$Q_c = 127 \text{ Вт/м}$  - теплоступление от окон.

$Q_{\text{изб.2}} = 4.6 \cdot 4 \cdot 0.6 \cdot 127 = 1402 \text{ Вт}$

$Q_{\text{изб.3}}$  - тепловыделения людей

$$Q_{\text{изб.3}} = n \cdot q, \text{ где} \quad (5.9)$$

$q = 80 \text{ Вт/чел.}$ ,  $n$  - число людей, например,  $n = 6$

$Q_{\text{изб.3}} = 6 \cdot 80 = 480 \text{ Вт}$

$Q_{\text{избыт}} = 330 + 1402 + 480 = 2212 \text{ Вт}$

Из уравнения теплового баланса следует:

$$V_{\text{вент}} = \frac{3600 \cdot 2932}{1000(29.4 - 18)} = 926 \text{ м}^3$$



Рисунок 5.1 – Вентиляционная система в офисе



## **Заключение**

В результате выполнения дипломной работы была спроектирована и разработана база данных для поисково-информационной системы «Тамада».

Был проведен сравнительный анализ существующих аналогов с точки зрения логики взаимодействия между поставщиком и потребителем.

В данном дипломном проекте был разработан веб-портал для размещения и просмотра объявлений, который позволяет пользователям быстро и удобно получить информацию, а также узнать полную информацию о заинтересовавшем вас объявлении и компании, которая подала данное объявление.

В технико-экономической части работы была рассчитана общая стоимость поисково-информационной системы и количество финансовых затрат на ее разработку. В результате расчетов выяснилось, что разработка системы является экономически эффективным и целесообразным решением.

В разделе безопасности жизнедеятельности был проведен анализ трудовых условий и оценка рабочего помещения, а так же предоставляется расчет систем пожаротушения и кондиционирования.

В процессе выполнения дипломной работы была достигнута цель разработки и выполнены все поставленные задачи.

## Список используемой литературы

1. Эрик Фримен – Паттерны программирования 2016.
2. Обзорная статья: Язык программирования Php
3. Yii. Русскоязычное общество программистов yiiframework.ru
4. К. Дж. Дейт Учебник - Введение в системы баз данных 2013
5. Томас Коннолли, Каролин Бегг - Базы данных. Проектирование, реализация и сопровождение. Теория и практика 2013
6. Дженнифер Нидерст Роббинс - Web-дизайн. Справочник 2014
7. Мейер - CSS — каскадные таблицы стилей. Подробное руководство 2013
8. Кристофер Шмитт - CSS рецепты программирования 2012
9. Энди Бадд - Мастерская CSS. Профессиональное применение web-стандартов 2014
10. Муссиано, Кеннеди - HTML и XHTML. Подробное руководство.
11. Заенцев И. В. Нейронные сети: основные модели И. В. Заенцев. — Воронеж: Изд-во Воронежского госуд. ун-та, 1999. — 76 с.
12. Хорошевский В Базы знаний интеллектуальных систем. Учебник. — СПб.: Питер, 2000, - 225 с.
13. Джонс М. Т. Программирование искусственного интеллекта в приложениях М. Тим Джонс; Пер. с англ. Осипов А. И. — М.: ДМК Пресс, 2006. — 312 с.
14. Сатимова Е.Г. Проектирование баз данных. Методические указания к выполнению лабораторных работ (для студентов всех специальностей) – Алматы: АИЭС, 2009. - 37с.
15. Методические указания к выполнению экономической части дипломных работ для студентов специальности 5В070400 – Вычислительная техника и программное обеспечение. З.Д. Еркешева, Г.Ш. Боканова Алматы: АУЭС, 2013 – 40 с.
16. ГОСТ 12.1.005-88 ОБЩИЕ САНИТАРНО-ГИГИЕНИЧЕСКИЕ ТРЕБОВАНИЯ К ВОЗДУХУ РАБОЧЕЙ ЗОНЫ
17. ГОСТ 30494-96 ПАРАМЕТРЫ МИКРОКЛИМАТА В ПОМЕЩЕНИЯХ
18. yiiframework.ru
19. htmlbook.com

## Листинг программной части

```

<?php

class SiteController extends ACLController
{

    public $actionACL = array(
        'index' => array('guest'),
        'error' => array('guest'),
        'captcha' => array('guest'),
        'contact' => array('guest'),
        'video' => array('guest'),
        'news' => array('guest'),
        'page' => array('guest'),
        'list' => array('guest'),
        'search' => array('guest'),
    );

    public $layout = '//layouts/main';
    /**
     * Declares class-based actions.
     */
    public function actions()
    {
        return array(
            // captcha action renders the CAPTCHA image displayed on the contact
            'captcha'=>array(
                'class'=>'CCaptchaAction',
                'testLimit' => 1,
                'backColor'=>0xFFFFFFFF,
            ),
            // page action renders "static" pages stored under
            'protected/views/site/pages'
            // They can be accessed via: index.php?r=site/page&view=FileName
            'page'=>array(
                'class'=>'CViewAction',
            ),
        );
    }

    public function actionPage($id)
    {

```

```

// $alias = $_GET['alias'];
$model = InfoPage::model()->findByAttributes(array('alias' => $id));

if(!$model || $model->alias == 'contact' || $model->alias == 'interesting') {
    throw new CHttpException(404, 'Страница не найдена');
}
$this->render('page', array('model' => $model));
}

/**
 * This is the default 'index' action that is invoked
 * when an action is not explicitly requested by users.
 */
public function actionIndex()
{
    $criteria = new CDbCriteria();
    $criteria->condition = 'disabled=0';
    $criteria->order = 'name';

    $services = Service::model()->findAll($criteria);

    $view = Yii::app()->mobileDetect->isMobile() ? 'index_mobile' : 'index';
    $this->render($view, array('services' => $services));
}

/**
 * This is the action to handle external exceptions.
 */
public function actionError()
{
    if($error=Yii::app()->errorHandler->error)
    {
        if(Yii::app()->request->isAjaxRequest)
            echo $error['message'];
        else
            $this->render('error', $error);
    }
}

public function actionVideo(){
    $this->render('video');
}

/**
 * Displays the contact page
 */
public function actionContact()

```

```

{
    $contact = InfoPage::model()->findByAttributes(array('alias'=>'contact'));
    $model=new ContactForm;
    if(isset($_POST['ContactForm']))
    {
        $model->attributes=$_POST['ContactForm'];
        if($model->validate()
        {
            $name='=?UTF-8?B?'.base64_encode($model->name).'?=';
            $subject='=?UTF-8?B?'.base64_encode($model->subject).'?=';
            $headers="From: $name <{$model->email}>\r\n".
                "Reply-To: {$model->email}\r\n".
                "MIME-Version: 1.0\r\n".
                "Content-Type: text/plain; charset=UTF-8";

            mail(Yii::app()->params['adminEmail'],$subject,$model->body,$headers);
            Yii::app()->user->setFlash('contact','Thank you for contacting us.
We will respond to you as soon as possible. ');
            $this->refresh();
        }
    }
    $this->render('contact',array('model'=>$model, 'contact'=> $contact));
}

protected function performAjaxValidation($model) {
if (isset($_POST['ajax']) && $_POST['ajax'] === 'user-form') {
    $model->setScenario("ajax");
    echo CActiveForm::validate($model);
    Yii::app()->end();
}
}

public function actionList($id) {

    $this->layout = 'application.views.layouts.adverts';
    $criteria = new CDbCriteria();

    $criteria->select = array('t.*', new CDbExpression('IFNULL(AVG(f.rate),0) as
companyRating'));
    $criteria->join = 'LEFT JOIN catalog_advert_feedback as f ON f.advert_id = t.id
AND f.approved = 1';
    $criteria->condition = 't.service_id =:service AND t.is_disabled = 0 AND t.published
= 1';
    $criteria->group = 't.id';
    // $criteria->order = 'IF( UNIX_TIMESTAMP(toTopDate) >=
UNIX_TIMESTAMP(),toTopDate, t.created_date) DESC';

```

```

$criteria->params = array();
$criteria->params[':service'] = intval($id);

if(isset($_GET['rt'])) {
    $criteria->mergeWith(array('having' => 'companyRating >= :rating'));
    $criteria->params[':rating'] = intval($_GET['rt']);
    $adFilter['rt'] = intval($_GET['rt']);
}

if (Yii::app()->user->region != 0) {
    $criteria->mergeWith(array(
        'condition' => 'region_id=:region',
    ));
    $criteria->params[':region'] = Yii::app()->user->region;
}

if(isset($_GET['kw'])) {
    $criteria->compare('t.title', $_GET['kw'], true);
    $criteria->compare('t.text', $_GET['kw'], true, 'OR');
}

if(isset($_GET['r1'])&&isset($_GET['r2'])) {
    $criteria->mergeWith(array('condition' => 't.price_tg >= :r1'));
    $criteria->mergeWith(array('condition' => 't.price_tg <= :r2'));
    $criteria->params[':r1'] = floatval($_GET['r1']);
    $criteria->params[':r2'] = floatval($_GET['r2']);
}

$criteria->with = array('company', 'region');
$criteria->mergeWith(array('condition' => 'company.is_disabled=0'));

$sort = new AdvertSort();
$sort->defaultOrder = "IF( UNIX_TIMESTAMP(toTopDate) >=
UNIX_TIMESTAMP(),toTopDate, t.created_date) DESC";
$sort->attributes = array(
    'rating'=>array(
        'asc'=>'companyRating ASC',
        'desc'=>'companyRating DESC',
    ),
    'price_tg', // add all of the other columns as sortable
);

$criteriaProvider = new CActiveDataProvider('Advert', array('criteria' =>
$criteria, 'sort'=>$sort,
    'pagination'=> array(
        'pageSize'=>10,

```

```

        ),
    ));

    //Get prices in current service
    $scr = new CDbCriteria();
    $scr->select = 'MIN(price_tg) as minPrice, MAX(price_tg) as maxPrice';
    $scr->condition = 'service_id=:service AND is_disabled=0';
    $scr->params[':service'] = intval($id);

    $priceFilter = Advert::model()->find($scr);

    //Get total count of advert in selected srvice category
    $currentService = Service::model()->findByPk(intval($id));
    $advertCount = $currentService->advertCount;

    //Get selected item prices
    $price = array_map(function($n) { return $n->price_tg; }, $dataProvider->getData());

    $adFilter['minPrice'] = empty($price) ? $priceFilter->minPrice : min($price);
    $adFilter['maxPrice'] = empty($price) ? $priceFilter->maxPrice : max($price);

    //Hack for slider bug when selected range is [0,0]
    if($adFilter['maxPrice'] == 0)
        $adFilter['maxPrice'] = 1;

    if($advertCount == 0)
    {
        $this->render('adverts/empty');
        Yii::app()->end();
    }
    $this->render('adverts/list', array('dataProvider' => $dataProvider,
        'adFilter' => $adFilter,
        'priceFilter' => $priceFilter,
        'advertCount' => $advertCount));
}

public function actionSearch($q){
    $this->layout = 'application.views.layouts.ad';
    //$this->render('search/test');
    if(isset($_GET) && $_GET['q'] != null){
        $q = Yii::app()->request->getParam('q');
        $$sql = 'SELECT * FROM catalog_advert WHERE MATCH(' . Yii::app()->sphinx->quoteValue($_GET['q']).')';
        $result = Yii::app()->sphinx->createCommand($$sql)->queryAll();
        //die(var_dump($result));
        //die($$sql);
        $adv = [];
    }
}

```

```

        $i = 0;
        foreach($result as $r){
            //die(var_dump($r));
            $adv[] = $r['id'];
            //$i+=1;
        }
        //die(var_dump($adv));
        $criteria = new CDbCriteria;
        $criteria->addInCondition('id', $adv);
        if(Yii::app()->user->region != 0)
            $criteria->condition = 'region_id='.Yii::app()->user->region;
        $models = Advert::model()->findAll($criteria);
        //die(var_dump($models));
        if($result){
            $this->render('search/search', ['models' => $models,'query' => $q]);
        }
        else{
            $this->render('search/empty');
        }
    }
    else{
        $this->render('search/error');
    }
}

}

<?php

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

/**
 * Description of ApiController
 *
 * @author LaptopUser
 */
class ApiController extends ACLController {

    public $actionACL = array(
        'saveimageattachment' => array('user'),
        'regionSelect' => array('guest'),
        'ajaxcrop' => array('user'),
        'redactorImageUpload' => array('user'),

```



```

'ajaxFeedbackComment' => array('user'),
'ajaxFavourite' => array('user'),
'ajaxMessage' => array('guest'),
'showPromotionModal' => array('user'),
'advertPromote' => array('user')
);

public function actions() {
    return array(
        'saveimageattachment' => 'ImageAttachmentAction',
    );
}

public function actionRegionSelect() {
    if (isset($_POST['regionSelect'])) {
        Yii::app()->user->setState('region', $_POST['regionSelect']);
        if (!Yii::app()->user->isGuest) {
            Yii::app()->user->setRegion($_POST['regionSelect']);
        }
        $cookie = new CHttpCookie('region', $_POST['regionSelect']);
        $cookie->expire = time() + (60 * 60 * 24 * 365); // (1 year)
        Yii::app()->request->cookies['region'] = $cookie;
        //return 'Region set';
    }

    Yii::app()->end();
}

public function actionAjaxcrop()
{
    if(Yii::app()->request->isAjaxRequest) {
        Yii::import('application.extensions.jcrop.EJCropper');
        $jcropper = new EJCropper();
        if($_POST['jCropper_w'] == 0 || $_POST['jCropper_h'] == 0){
            echo 'fail';
            Yii::app()->end();
        }
        $jcropper->targ_w = $_POST['jCropper_w'];
        $jcropper->targ_h = $_POST['jCropper_h'];
        $ds = DIRECTORY_SEPARATOR;
        $dir = strtolower($_POST['modelName']);
        if( $dir == 'advert') $dir = 'obiav';
        $filePath = Yii::app()->basePath.$ds.'..' . $ds.'upload' . $ds.$dir.$ds.'preview';
        $jcropper->thumbPath = $filePath;

        // some settings ...
        $jcropper->jpeg_quality = 95;
    }
}

```

```

$jcropper->png_compression = 8;

// get the image cropping coordinates (or implement your own method)
$coords = $jcropper->getCoordsFromPost('jCropper');

// returns the path of the cropped image, source must be an absolute path.
$thumbnail = $jcropper->crop($filePath.$ds.$_POST['id'].'.jpg', $coords);

if(is_file($thumbnail)) echo 'OK';
}
Yii::app()->end();
}

public function actionRedactorImageUpload($type)
{
    $file = CUploadedFile::getInstanceByName('file');

    $allowedTypes = array('image/png', 'image/jpeg', 'image/jpg');

    if( $file->size <= 5*1024*1024)
    {

        $owner_id = Yii::app()->user->id;

        $ds = DIRECTORY_SEPARATOR;

        $targetPath = '..' . $ds . 'upload' . $ds . 'editor' . $ds . $owner_id . $ds . $type .
$ds;

        $fileName = date('Ydm') . '_' . microtime();

        $filePath = $targetPath . $fileName;

        $savePath = Yii::app()->basePath . $ds . $filePath; //Ссылка для сохранения
файла

        if (!is_dir(Yii::app()->basePath . $ds . $targetPath))
            mkdir(Yii::app()->basePath . $ds . $targetPath, 0777, true);

        if($file->saveAs($savePath)) {
            $mime = CFileHelper::getMimeType($savePath);
            if(in_array($mime, $allowedTypes)) {

                $img = Image::factory($savePath);
                $img->resize(700, null);
                $img->save();
            }
        }
    }
}

```

```

        $link = "/upload/editor/" . $owner_id . "/" . $type . "/" . $fileName;

        $array = array(
            'filelink' => $link
        );

        } else {
            unlink($savePath);
            $array = array(
                "error" => true,
            );
        }
    } else {
        $array = array(
            "error" => true,
        );
    }
}
else
{
    $array = array(
        "error" => true,
    );
}

echo stripslashes(CJSON::encode($array));
}

public function actionAjaxFeedbackComment() {
    if(Yii::app()->request->isAjaxRequest && isset($_POST)) {
        $form = new FeedbackCommentForm();
        $form->attributes = $_POST;
        if($form->validate()) {
            if($form->save()) {
                Yii::app()->user->setFlash('success','Сообщение успешно добавлено.');
            }
        }
        else
            Yii::app()->user->setFlash('danger','Произошла ошибка. ' . $form->getError('text') . ' Попробуйте еще раз.');
```

```

    }
    Yii::app()->end();
}

public function actionAjaxFavourite() {
```

```

if(Yii::app()->request->isAjaxRequest && isset($_POST)) {

    $user = User::model()->with('favourites')->findByPk(Yii::app()->user->id);
    $advert = Advert::model()->findByPk($_POST['advert_id']);

    if($user && $advert) {

        $current = array_keys(CHtml::listData($user->favourites, 'id', 'title'));

        if(($key = array_search($advert->id, $current)) !== false) {
            unset($current[$key]);
            $result = 'removed';
        }
        else
        {
            $current[] = $advert->id;
            $result = 'added';
        }

        $user->favourites = $current;

        if($user->save())
            echo $result;
        else
            echo "error";
    }
}

public function actionAjaxMessage()
{
    if(Yii::app()->request->isAjaxRequest && isset($_POST)) {
        $pr = new CHtmlPurifier();
        $data = array();
        $data['email'] = Yii::app()->user->email;
        $data['name'] = $_POST['name'];
        $data['text'] = $_POST['text'];

        $advert = Advert::model()->findByPk(intval($_POST['advert']));
        if(!$advert)
            throw new CHttpException(404, 'Advert not found!');

        $model = new UserNotification();
        $model->title = "У Вас новое сообщение к объявлению ".$advert->title;
        $model->text = $this->renderPartial('/notifications/adMessage', $data, true);
        $model->to = $advert->user->id;
        $model->withEmail = 1;
    }
}

```

```

if($model->save()) {
    Yii::app()->user->setFlash('success', 'Сообщение отправлено. ');
    $model->refresh();
    if($model->withEmail) {
        $data = array(
            'view' => 'newMessage',
            'subject' => 'Новое сообщение на портале Tamada.kz',
            'params' => array(
                'title' => $model->title
            )
        );
        $user = User::model()->findByPk($model->to);
        if($user) {
            $user->sendNotification($data);
        }
    }
}
else {
    Yii::app()->user->setFlash('danger', 'К сожалению, произошла ошибка,
попробуйте еще раз. ');
}
}

Yii::app()->end();
}

public function actionShowPromotionModal() {
    if(Yii::app()->request->isAjaxRequest && isset($_POST)) {

        $model = AdvertPromotion::model()->findByAttributes(array('name' =>
$_POST['type'], 'is_disabled=0');
        if (null != $model) {
            echo CJSON::encode($model->getAttributes());
        } else
            throw new CHttpException(500, 'Promotion not found. ');
    }

    Yii::app()->end();
}

public function actionAdvertPromote() {
    if(Yii::app()->request->isAjaxRequest && isset($_POST)) {
        $advert_id = intval($_POST['advert_id']);
        $advert = Advert::model()->findByPk($advert_id);
        if(!$advert)

```

```

        throw new CHttpException(500, 'Advert not found');

        $promo_id = intval($_POST['type_id']);
        $promo = AdvertPromotion::model()->findByPk($promo_id);
        if(!$promo)
            throw new CHttpException(500, 'Promotion not found');

        if($promo->promote($advert)) {
            echo "success";
        } else {
            echo "error";
        }
    }
    Yii::app()->end();
}
}

```

<?php

```

class AdvertController extends ACLController {

    public $actionACL = array(
        'create' => array('user'),
        'delete' => array('user'),
        'update' => array('user'),
        'view' => array('guest'),
    );

    public $layout = 'application.views.layouts.main';
    /**
     * Displays a particular model.
     * @param integer $id the ID of the model to be displayed
     */
    public function actionView($id) {

        $this->layout = '';

        $model = Advert::model()->with('approvedFeedback',
            'company')->findByPk(intval($id));

        if ($model === null)
            throw new CHttpException(404, 'Страница не найдена.');
```

```

        if($model->company->is_disabled || $model->is_disabled)
            throw new CHttpException(403, 'Доступ ограничен.');
```

```

        $feedbackForm = new FeedbackForm();

```

```

if (isset($_POST['FeedbackForm'])) {
    $feedbackForm->attributes = $_POST['FeedbackForm'];
    $feedbackForm->advert_id = $id;
    $feedbackForm->user_id = Yii::app()->user->id;

    if ($feedbackForm->validate()) {
        if ($feedbackForm->save()) {
            Yii::app()->user->setFlash('success', 'Спасибо! Ваш отзыв будет опубликован
после проверки модератором.');
```

```
        } else {
            Yii::app()->user->setFlash('danger', 'К сожалению, произошла ошибка.
попробуйте еще раз или обратитесь к администратору.');
```

```
        }
    }
}

$this->pageTitle = $model->title;

$this->render('view', array(
    'model' => $model,
    'feedbackForm' => $feedbackForm,
));
}

/**
 * Creates a new model.
 * If creation is successful, the browser will be redirected to the 'view' page.
 */
public function actionCreate($id = null) {
    if(Yii::app()->user->companiesCount == 0) {
        Yii::app()->user->setFlash('success', 'Чтобы подать объявление, сначала добавьте
Вашу компанию.');
```

```
        $this->redirect('/catalog/company/create');
    }
    $model = new Advert;

    // Uncomment the following line if AJAX validation is needed
    // $this->performAjaxValidation($model);
    if (isset($id))
        $model->company_id = intval($id);

    if (isset($_POST['Advert'])) {
        $model->attributes = $_POST['Advert'];

        if(isset($_POST['currency']))
            $model->default_currency = $_POST['currency'];

        if(isset($_POST['serviceAmount'])) {
            if(isset($_POST['minServiceAmount']) && isset($_POST['maxServiceAmount']))
{
```

```

        $model->minServiceAmount = intval($_POST['minServiceAmount']);
        $model->maxServiceAmount = intval($_POST['maxServiceAmount']);
    }
}

$model->purify();

if ($model->save())
{
    Yii::app()->user->setFlash('success', 'Объявление успешно создано');
    $this->redirect(array('update', 'id' => $model->id));
    Yii::app()->end();
}

$this->render('create', array(
    'model' => $model,
));
}

/**
 * Updates a particular model.
 * If update is successful, the browser will be redirected to the 'view' page.
 * @param integer $id the ID of the model to be updated
 */
public function actionUpdate($id) {

    $model = $this->loadModel($id);

    $this->isOwnAdvert($model);
    // Uncomment the following line if AJAX validation is needed
    // $this->performAjaxValidation($model);

    if (isset($_POST['Advert'])) {
        $model->attributes = $_POST['Advert'];

        $model->price = $_POST['Advert']['price'];
        $model->published = isset($_POST['Advert']['published']) ?
$_POST['Advert']['published'] : 0 ;

        if(isset($_POST['serviceAmount'])) {
            if(isset($_POST['minServiceAmount']) && isset($_POST['maxServiceAmount']))
            {
                $model->minServiceAmount = intval($_POST['minServiceAmount']);
                $model->maxServiceAmount = intval($_POST['maxServiceAmount']);
            }
        } else {
            $model->minServiceAmount = null;
            $model->maxServiceAmount = null;
        }
    }
}

```



```

        if(isset($_POST['currency']))
            $model->default_currency = $_POST['currency'];

        $model->purify();

        if ($model->save())
            $this->redirect(array('view', 'id' => $model->id));
    }

    $this->render('update', array(
        'model' => $model,
    ));
}

/**
 * Deletes a particular model.
 * If deletion is successful, the browser will be redirected to the 'admin' page.
 * @param integer $id the ID of the model to be deleted
 */
public function actionDelete($id) {

    $model = Advert::model()->with('userLiked')->findByPk($id);

    if ($this->isOwnAdvert($model))
    {
        $transaction=$model->dbConnection->beginTransaction();

        try
        {
            if($model->delete()) {
                $transaction->commit();
                Yii::app()->user->setFlash('success', 'Объявление удалено.');
```

```

* Returns the data model based on the primary key given in the GET variable.
* If the data model is not found, an HTTP exception will be raised.
* @param integer $id the ID of the model to be loaded
* @return Advert the loaded model
* @throws CHttpException
*/
public function loadModel($id) {
    $model = Advert::model()->findByPk($id);
    if ($model === null)
        throw new CHttpException(404, 'The requested page does not exist.');
```

```

    return $model;
}

/**
* Performs the AJAX validation.
* @param Advert $model the model to be validated
*/
protected function performAjaxValidation($model) {
    if (isset($_POST['ajax']) && $_POST['ajax'] === 'advert-form') {
        echo CActiveForm::validate($model);
        Yii::app()->end();
    }
}

protected function isOwnAdvert($model) {
    if (!$this->z('ownAdvert', array('advert' => $model)))
        throw new CHttpException(403, 'Access denied');

    return true;
}
}

<?php $this->pageTitle = 'Главная'; ?>
<div class="m-b-sm nav-events text-center">
    <div class="events-menu">
        <button class="btn btn-primary lt sort-options" data-filter="all" data-
sort="default">Все</button>
        <button class="btn btn-primary lt sort-options" data-filter=".category-1" data-
sort="sort1:asc">Свадьбы</button>
        <button class="btn btn-primary lt sort-options" data-filter=".category-2" data-
sort="sort2:asc">Дни рождения</button>
        <button class="btn btn-primary lt sort-options" data-filter=".category-3" data-
sort="sort3:asc">Детские</button>
        <button class="btn btn-primary lt sort-options" data-filter=".category-4" data-
sort="sort4:asc">Корпоративы</button>
    </div>
</div>
<div class="shuffle" style="">
    <?php foreach($services as $i => $service) : ?>
        <?php

```

```

        $classes = "mix item";
        $sort = "";

        foreach($service->priorities as $id=>$priority)
        {
            $classes.=' category-'. $id;
            $sort.=" data-sort". $id.'='.$priority.'";
        }

        echo '<div class="'. $classes.'" ' . $sort;
    ?>

    <?php echo '>'?>
        <a href="/site/list/<?php echo $service->id ?>"></a>
        <div class="text-center related-news" style="position: relative">
            <div class="clear" style="height: 35px;"></div>
            <a class="service-button btn btn-s-md btn-warning" href="/site/list/<?php echo
$service->id ?>"><?php echo $service->name ?></a>
        </div>
    </div>
    <?php endforeach; ?>
        <div class="gap"></div>
        <div class="gap"></div>
    </div>
    <?php
        Yii::app()->clientScript->registerScriptFile('/js/imagesloaded.pkgd.min.js',
        CClientScript::POS_END); ?>
    <script src="http://cdn.jsdelivr.net/jquery.mixitup/latest/jquery.mixitup.min.js"></script>

    <script>
    $(document).ready(function(){
        var grid = $('<div class="shuffle">');

        grid.mixItUp();

        $(".sort-options").click(function() {
            var filter = $(this).data("filter");
            var sort = $(this).data("sort");

            console.log(filter)
            console.log(sort)
            grid.mixItUp('multiMix', {
                filter: filter,
                sort: sort,
            });
        });
    });
    </script>

```