# Реферат

Цель выпускной квалификационной работы – разработать информационную систему, предоставляющую информацию о фотооборудовании, позволяющую добавлять своё оборудование в БД, продавать и обсуждать его.

В ходе выполнения выпускной квалификационной работы был произведен анализ предметной области, включивший в себя сравнение с аналогами разрабатываемой системы, а также формирование функциональных требований к системе, требований пользователей к системе. На основании поставленных требований была разработана структура системы, построена диаграмма вариантов использования. Были также составлены инфологическая и даталогическая модели.

Система была реализована практически полностью, за исключением отображений некоторых разделов.

Данная ВКР также включила в себя расчет трудоёмкости, себестоимости разрабатываемой системы и обзор мер по обеспечению безопасности при эксплуатации системы.

**Введение**

Всё больше людей, перед тем как пойти в магазин для покупки фотооборудования, заходят в интернет и ищут всевозможную информацию об интересующей их продукции, такую как обзоры, цены, фотографии, отзывы и т.д. Так же многие фотографы покупают и продают как новую, так и б. у. фототехнику через интернет. При этом достаточно сложно найти самое выгодное предложение, так как в сети Интернет существуют десятки ресурсов по продаже той или иной фототехники, на которых цена может различаться в разы. При выборе оборудования люди ищут статьи, обзоры и отзывы об интересующей их технике, но и это весьма затруднительно, так как качественная информация в большинстве своём опубликована на сайтах, посвященных конкретному производителю.

На данный момент существует достаточно много сайтов посвященных фотографии и фотооборудованию. Но ни один из них в полной мере не реализует все описанные выше потребности людей, увлекающихся фотографией.

Целью данного дипломного проекта является создание такой системы, которая максимально упростит процесс выбора фотооборудования, поиск минимальной цены и магазина. При этом у пользователя должна быть возможность просматривать статьи, отзывы и обзоры про интересующее его оборудование.

# 1. Конструкторский раздел

##

## 1.1 Анализ предметной области и постановка задачи

###

### 1.1.1 Назначение системы

Проектируемая система должна упростить выбор фотооборудования, максимально отвечающего требованиям пользователя, и поиск самого выгодного предложения по покупке этого оборудования.

### 1.1.2 Цели создания системы

К основным целям можно отнести:

* Создание единого информационного пространства по тематикам фототехника и фотография;
* Обеспечение оперативного информационного взаимодействия потенциальных пользователей посредством публикаций, интерфейсов обмена данными и прочих сервисов системы;
* Предоставление пользователю исчерпывающей информации по данной тематике;

Наряду с общими целями создание системы должно преследовать следующие специфические цели:

* Привлечение большего числа пользователей, что необходимо для расширения системы и достижения целей;
* Привлечение партнёров, а именно магазинов и(или) людей, основной деятельностью которых является купля-продажа фототехники и сопутствующих товаров, с целью взаимовыгодного сотрудничества.

### 1.1.3 Обзор аналогов

**«Яндекс Маркет»**

«Яндекс Маркет» – наиболее схожий по функционалу сервис с проектируемым. Даёт возможность посмотреть характеристики оборудования и его фотографии, прочитать отзывы о товаре и магазинах, обсудить оборудование и посмотреть фотографии, сделанные с его помощью. Имеет достаточно удобный расширенный поиск. У магазинов есть возможность автоматически синхронизировать свой каталог с каталогом сервиса (см. рисунок 1). К недостаткам с точки зрения пользователя можно отнести: отсутствие обзоров и статей, пользователи не могут продать своё оборудование, представители магазинов сами добавляются в каталог, у пользователя нет возможности добавить магазин с возможно более низкими ценами, нет сравнения с зарубежными ценами.

К недостаткам с точки зрения магазинов можно отнести: магазины платят за клики по ссылке, более справедливо было бы взымать с магазина процент с продажи товара, так как пользователь может и нажал на ссылку, а купил вряд ли. Нет возможности оформить заказ, не переходя на сайт магазина, следовательно, магазину обязательно надо иметь свой сайт, что затруднительно для мелких предпринимателей.

Рисунок 1 – Яндекс Маркет

**Молоток.ру**

Молоток.ру – самый крупный в интернет-аукцион в русскоязычной части интернета, позволяет пользователям продавать б. у. и новые вещи, технику и многое другое (см. рисунок 2). Достаточно большой выбор б.у. фототехники, основным недостатком является то, что нет единой базы оборудования в самом сервисе, поэтому описание товаров приходится искать на других сайтах, что весьма не удобно. Товары размещают в основном только физические лица, нет информации о ценах в магазинах. Очень затруднено общение пользователей.

Рисунок 2 – Молоток.ру

**Различные фото форумы**

Форумы предназначены в первую очередь для обсуждения пользователями конкретной темы, хотя многие публикуют там обзоры и продают через них оборудование, но это крайне неудобно.

Пример:

* http://olympus.ourlife.ru/forum/
* http://www.oly43club.ru/
* http://www.club-nikon.ru/forum
* http://www.photoforum.ru/forum/

###

### 1.1.4 Классы пользователей

Предполагается, что данная система должна быть проста в использовании и доступна широкому кругу пользователей. В связи с этим, можно выделить два основных класса пользователей:

* любой пользователь Интернета, желающий купить фотооборудование и(или) интересующийся фотооборудованием и(или) желающий посмотреть / разместить фотографии в интернете;
* представители магазинов и(или) люди, основной деятельностью которых является купля-продажа фототехники и сопутствующих товаров;
* модераторы, которые будут контролировать работу сайта, добавлять новый материал и следить, чтобы остальные пользователи не нарушали правила.

### 1.1.5 Требования пользователей к системе

Основная цель создания любой программной системы – создание такого программного продукта, который помогает пользователю выполнять необходимые задачи. Сначала необходимо определить требования, которым должна удовлетворять система.

Для того чтобы более точно понять, как должна работать система, используется описание функциональности системы через варианты использования (Use Case или прецеденты). Варианты использования это – описание последовательности действий, которые может осуществлять система в ответ на внешние воздействия пользователей или других программных систем. Варианты использования отражают функциональность системы с точки зрения получения значимого результата для пользователя, поэтому они точнее позволяют ранжировать функции по значимости получаемого результата.

В данной системе можно выделить три основные роли:

* незарегистрированный пользователь – любой пользователь сети;
* зарегистрированный пользователь;
* модератор системы – пользователь, которому предоставляется доступ к редактированию и обновлению базы данных системы и сайта в целом.

Ниже приведены диаграммы вариантов использования системы для каждой роли. На рисунке 3 отражены варианты использования, специфические для модератора. Стоит отметить, что также для администратора доступны все действия, доступные зарегистрированному пользователю (см. рисунок 5).

Рисунок 3 – Диаграмма вариантов использования модератора

Рисунок 4 – Диаграмма вариантов использования для незарегистрированного пользователя

###

### 1.1.6 Требования к функциональности системы

1. Регистрация пользователя.
2. Простой поиск по названию.
3. Расширенный поиск по категориям, характеристикам, ценам, рейтингу оборудования и т.д.
4. Вывод на экран списков оборудования, соответствующего заданным характеристикам.
5. Вывод на экран подробной информации об оборудовании, при этом выводится краткая информация из других разделов, связанная с ним.
6. Добавление и редактирование оборудования (информации о нем).
7. Создание, просмотр и редактирование статей.
8. Добавление, просмотр и редактирование информации о магазине.
9. Загрузка фотографий, создание, просмотр, редактирование альбомов.
10. Разделение оборудования, статей и альбомов на категории.
11. Возможность связывать оборудование, статьи, альбомы и магазины с тегами. При этом пользователи могут повлиять на уровень принадлежности тега к данному оборудованию (статье и т.д.).
12. Возможность оценивать оборудование, статьи, фотографии и магазины.
13. Возможность комментировать статьи и фотографии. При этом пользователи могут повлиять на степень соответствия комментария к тематике данной статьи или фотографии.
14. Возможность оставлять рекомендации об оборудовании и магазинах. При этом пользователи могут указать, полезна ли им была эта рекомендация или нет.
15. Возможность отправлять сообщения другим пользователям.
16. У зарегистрированного пользователя должна быть возможность добавления оборудования в личные списки «моё оборудование», «продам», «куплю».
17. У партнёров должна быть возможность добавления (актуализации) своего каталога товаров как в ручном режиме, так и в автоматическом.

###

### 1.1.7 Требования к дизайну системы

1. Дизайн должен содержать минимум изображений;
2. Дизайн ИСПФ должен быть консервативным – использовать неброские, неяркие цвета и тона;
3. Элементы дизайна не должны сильно влиять на скорость загрузки страниц.


### Рисунок 5 – Диаграмма вариантов использования для зарегистрированного пользователя

### 1.1.8 Требования к шрифтам и кодировке

1. В системе должны использоваться шрифты, входящие в стандартную поставку ОС Windows (старше 98) и ОС Linux;
2. Размер (кегль) шрифтов должен обеспечивать удобство восприятия текста при минимально допустимом размере экрана.
3. Кодировка страниц системы: UTF-8.

###

### 1.1.9 Требования к средствам просмотра системы

Сайт должен обеспечивать корректное отображение данных в следующих браузерах:

1. Internet Explorer (версия 5.5 и выше);
2. Opera (версия 7.0 и выше);
3. Mozilla Firefox (версия 1.0 и выше).

###

### 1.1.10 Требования к компоновке страниц системы

Ширина информативной области страниц сайта должна быть статична и не превышать 1024px.

###

### 1.1.11 Требование к содержанию системы

Страницы Сайта должны быть оптимизированы под поисковые системы Google и Яндекс.

##

## 1.2 Проектирование системы

###

### 1.2.1 Структура сайта

Основываясь на требованиях пользователей к системе, была составлена общая структура сайта системы. Она изображена на рисунке 6.

Приведем описание общей структуры сайта системы с кратким описанием его компонентов:

1. PhotoHQ – главная страница, на которой будет описано предназначение сайта, что на нём можно найти. Также на этой странице будет ссылка на раздел «Экскурсия по сайту», в котором будут описаны преимущества регистрации и кратки инструкции пользователя;

Рисунок 6 – Общая структура сайта

1. Раздел «Регистрация» – Раздел предназначен для регистрации нового пользователя;
2. Раздел «Поиск» – Форма поиска по разделам;
3. Раздел «Оборудование»:
* Списки оборудования – выборки оборудования по критериям, таким как: оценка, теги, тип, количество пользователей у которых оно есть и т.д.;
* Информация об оборудовании – страница, на которой расположена информация по конкретному оборудованию и краткая информация из других разделов, связанная с ним;
* Цены на оборудование – список цен конкретного оборудования, с возможностью перейти на сайт магазина для заказа или связаться с продавцом;
* Сообщить об ошибке – форма для отправки модераторам информации о неточности в описании;
1. Раздел «Статьи»:
* Списки статей – выборки статей по критериям, таким как: оценка, теги и время публикации. Так же они могут быть результатом расширенного поиска;
* Просмотр статьи – страница на которой расположена сама статья, краткая информация из других разделов, связанная с ней, и комментарии. Статья может располагаться на удалённом сайте, тогда происходит редирект на внешний источник;
* Добавление статьи – форма для добавления статьи;
1. Раздел «Магазины»:
* Списки магазинов – выборки магазинов по критериям, таким как: оценка, местоположение и тип.
* Информация о магазине – страница, на которой расположена информация по конкретному магазину и краткая информация из других разделов, связанная с ним.
* Добавление магазина – форма добавления магазина.
1. Раздел «Фотографии»:
* Списки фотографии – выборки фотографии по критериям, таким как: оценка, теги, альбом и т.д.
* Список альбомов – Страница, на которой расположен список альбомов конкретного пользователя.
* Просмотр фото – Страница, на которой будет расположена фотография, информация о ней и комментарии.
* Добавление фотографии – форма загрузки фотографии и добавления информации о ней.
1. Раздел «Пользователи»:
* Информация о пользователе – Страница с информацией о пользователе;
* Моя страница – страница пользователя;
* Редактирование \ добавление информации о себе – Форма Редактирования \ добавления информации о себе;
* Список оборудования пользователя – список оборудования, которое есть у пользователя;
* Куплю – список оборудования, которое пользователь хотел бы купить;
* Сообщения – личные сообщения пользователя.

###

### 1.2.2 Структура базы данных

Проанализировав функциональные требования к системе, а также требования пользователей системы была составлена инфологическая и даталогическая модель базы данных. Ниже будут приведены даталогическая и инфологическая модели базы данных.

**Инфологическая модель базы данных**

Цель инфологического моделирования – обеспечение наиболее естественных для человека способов сбора и представления той информации, которую предполагается хранить в создаваемой базе данных. На рисунке 7 изображена инфологическая схема базы данных.

**Даталогическая модель базы данных**

На рисунке 8 представлены основные таблицы базы данных системы. Таблицы, автоматически генерируемые фреймворком Django, были скрыты, поскольку не имеют отношения к данному дипломному проекту.

Рисунок 7 – Инфологическая схема базы данных

Рисунок 8 – Структура базы данных

Таблица **users** хранит основную информацию о пользователях

Таблица 1.1 – Users

|  |  |
| --- | --- |
| Название поля | Тип поля |
| id | Целочисленный |
| name | Строковый |
| first\_name | Строковый |
| last\_name | Строковый |
| email | Строковый |
| password | Строковый |
| is\_stuff | Логический |
| is\_active | Логический |
| is\_superuser | Логический |
| last\_login | Целочисленный |
| date\_joined | Целочисленный |

* + id – уникальное поле таблицы численного типа, содержащее номер пользователя;
	+ name – отображаемое имя пользователя;
	+ first\_name – имя пользователя;
	+ last\_name – фамилия пользователя;
	+ email – электронная почта пользователя (логин пользователя для входа в систему);
	+ password – пароль пользователя, для входа в систему;
	+ is\_stuff – флаг активности пользователя;
	+ is\_active – флаг активации профиля (1 – профиль активирован, 0 – профиль не активирован);
	+ is\_superuser – флаг суперюзера(модератора);
	+ last\_login – дата и время последней авторизации;
	+ date\_joined – дата и время регистрации;
* Таблица **geo\_countries** хранит список стран

Таблица 1.2 – Geo\_countries

|  |  |
| --- | --- |
| Название поля | Тип поля |
| id | Целочисленный |
| name | Строковый |

* + id – уникальное поле таблицы численного типа, содержащее номер страны;
	+ name – название страны;
* Таблица **geo\_cities** хранит список городов

Таблица 1.2 – Geo\_cities

|  |  |
| --- | --- |
| Название поля | Тип поля |
| id | Целочисленный |
| name | Строковый |
| country\_id | Целочисленный |

* + id – уникальное поле таблицы численного типа, содержащее номер города;
	+ name – название города;
	+ country\_id – номер страны;
* Таблица **us\_info** хранит дополнительную информацию о пользователе

Таблица 1.3

|  |  |
| --- | --- |
| Название поля | Тип поля |
| user\_id | Целочисленный |
| city\_id | Целочисленный |
| icq | Целочисленный |
| avatar\_path | Строковый |

* + user\_id – уникальное поле таблицы численного типа, содержащее номер пользователя;
	+ city\_id – номер города;
	+ icq – ICQ UIN пользователя;
	+ avatar\_path – путь к фотографии пользователя;
* Таблица **us\_groups** хранит названия групп пользователей

Таблица 1.4 – Us\_groups

|  |  |
| --- | --- |
| Название поля | Тип поля |
| id | Целочисленный |
| name | Строковый |

* + Id – уникальное поле таблицы численного типа, содержащее номер группы;
	+ name – название группы;
* Таблица **us\_permissions** хранит список прав доступа

Таблица 1.5 – Us\_permissions

|  |  |
| --- | --- |
| Название поля | Тип поля |
| id | Целочисленный |
| name | Строковый |
| content\_type\_id | Целочисленный |
| codename | Строковый |

* + id – уникальное поле таблицы численного типа, содержащее номер прав доступа;
	+ name – название прав доступа;
	+ content\_type\_id – номер тира содержимого;
	+ codename – перечень прав;
* Таблица **us\_group\_permissions** хранит связи групп пользователей и прав доступа, соответствующих им

интернет магазин фотооборудование цена

Таблица 1.6 – Us\_group\_permissions

|  |  |
| --- | --- |
| Название поля | Тип поля |
| id | Целочисленный |
| group\_id | Целочисленный |
| permission\_id | Целочисленный |

* + id – уникальное поле таблицы численного типа, содержащее номер связи;
	+ group\_id – номер группы пользователей;
	+ permission\_id – номер прав доступа;
* Таблица **us\_user\_groups** хранит список принадлежностей пользователей к группам

Таблица 1.7 – Us\_user\_groups

|  |  |
| --- | --- |
| Название поля | Тип поля |
| id | Целочисленный |
| user\_id | Целочисленный |
| group\_id | Целочисленный |

* + Id – уникальное поле таблицы численного типа, содержащее номер принадлежности;
	+ user\_id – номер пользователя;
	+ group\_id – номер группы;
* Таблица **messages** хранит сообщения пользователей

Таблица 1.8 – Messages

|  |  |
| --- | --- |
| Название поля | Тип поля |
| id | Целочисленный |
| subject | Строковый |
| body | Текст |
| sender\_id | Целочисленный |
| recipient\_id | Целочисленный |
| parent\_msg\_id | Целочисленный |
| send\_at | Целочисленный |
| read\_at | Целочисленный |
| replied\_at | Целочисленный |
| sender\_deleted\_at | Целочисленный |
| recipient\_deleted\_at | Целочисленный |

* + Id – уникальное поле таблицы численного типа, содержащее номер сообщения;
	+ subject – тема сообщения;
	+ body – текст сообщения;
	+ sender\_id – номер профиля отправителя;
	+ recipient\_id – номер профили получателя;
	+ parent\_msg\_id – номер сообщения, в ответ на которое написано данное;
	+ send\_at – дата и время отправления;
	+ read\_at – дата и время прочтения;
	+ replied\_at – дата и время ответа;
	+ sender\_deleted\_at – дата и время удаления отправителем;
	+ recipient\_deleted\_at – дата и время удаления получателем;
* Таблица it\_categories хранит категории содержимого (древовидная структура, вложенные множества)

Таблица 1.9 – It\_categories

|  |  |
| --- | --- |
| Название поля | Тип поля |
| id | Целочисленный |
| name | Строковый |
| parent\_id | Целочисленный |
| order | Целочисленный |
| slug | Строковый |
| alternate\_title | Строковый |
| description | Текст |
| lft | Целочисленный |
| rght | Целочисленный |
| tree\_id | Целочисленный |
| level | Целочисленный |

* + id – уникальное поле таблицы численного типа, содержащее номер категории;
	+ name – название категории;
	+ parent\_id – номер родительской категории;
	+ order – порядок категории;
	+ slug – название категории на латинице;
	+ alternate\_title – альтернативное название;
	+ description – описание категории;
	+ lft – левый индекс;
	+ rght – правый индекс;
	+ tree\_id – номер дерева категорий;
	+ level – уровень вложенности категории;
* Таблица **it\_brands** хранит список фирм производителей

Таблица 1.10 – It\_brands

|  |  |
| --- | --- |
| Название поля | Тип поля |
| id | Целочисленный |
| name | Строковый |
| description | Текст |
| logo\_path | Строковый |
| country\_id | Целочисленный |
| user\_id | Целочисленный |
| status | Целочисленный |
| created | Целочисленный |
| slug | Строковый |

* + id – уникальное поле таблицы численного типа, содержащее номер фирма;
	+ name – название фирмы;
	+ description – описание фирмы;
	+ logo\_path – путь к логотипу;
	+ country\_id – номер страны;
	+ user\_id – номер профиля представителя (0, если нет);
	+ status – статус публикации (опубликовано, требует модерации, отказано в публикации, требует доработки и т.д.);
	+ created – дата и время добавления в базу;
	+ slug – название фирмы на латинице;
* Таблица **items** хранит список оборудования и его основные характеристики

Таблица 1.11 – Items

|  |  |
| --- | --- |
| Название поля | Тип поля |
| id | Целочисленный |
| name | Строковый |
| brand\_id | Целочисленный |
| description | Текст |
| category\_id | Целочисленный |
| user\_id | Целочисленный |
| slug | Строковый |
| status | Целочисленный |
| created | Целочисленный |
| max\_price | Вещественный |
| min\_price | Вещественный |
| av\_price | Вещественный |

* + id – уникальное поле таблицы численного типа, содержащее номер оборудования;
	+ name – название оборудования;
	+ brand\_id – номер фирмы производителя;
	+ description – описание оборудования;
	+ category\_id – номер категории, к которой принадлежит оборудование;
	+ user\_id – номер пользователя, который добавил оборудование;
	+ slug – название оборудования на латинице;
	+ status – статус публикации;
	+ created – дата и время добавления оборудования;
	+ max\_price – максимальная цена на оборудование;
	+ min\_price – минимальная цена на оборудование;
	+ av\_price – средняя цена на оборудование;
* Таблица **us\_want\_it** связывает пользователей с оборудованием, которое они хотят купить

Таблица 1.12 – Us\_want\_it

|  |  |
| --- | --- |
| Название поля | Тип поля |
| id | Целочисленный |
| item\_id | Целочисленный |
| user\_id | Целочисленный |

* + id – уникальное поле таблицы численного типа, содержащее номер связи;
	+ item\_id – номер оборудования;
	+ user\_id – номер пользователя;
* Таблица **us\_have\_it** связывает пользователей с оборудованием, которое у них есть

Таблица 1.13 – us\_have\_it

|  |  |
| --- | --- |
| Название поля | Тип поля |
| id | Целочисленный |
| item\_id | Целочисленный |
| user\_id | Целочисленный |

* + id – уникальное поле таблицы численного типа, содержащее номер связи;
	+ item\_id – номер оборудования;
	+ user\_id – номер пользователя;
* Таблица **it\_pics** хранит изображения оборудования

Таблица 1.14 – it\_pics

|  |  |
| --- | --- |
| Название поля | Тип поля |
| id | Целочисленный |
| it\_id | Целочисленный |
| path | Строковый |
| rating | Вещественный |
| status | Целочисленный |

* + id – уникальное поле таблицы численного типа, содержащее номер изображения;
	+ it\_id – номер оборудования;
	+ path – путь к файлу изображения;
	+ rating – количество просмотров изображения;
	+ status – статус публикации;
* Таблица **it\_synonyms** хранит синонимы названий оборудования

Таблица 1.15 – it\_synonyms

|  |  |
| --- | --- |
| Название поля | Тип поля |
| id | Целочисленный |
| it\_id | Целочисленный |
| title | Строковый |

* + id – уникальное поле таблицы численного типа, содержащее номер синонима;
	+ it\_id – номер оборудования;
	+ title – синоним;
* Таблица **it\_prop\_names** хранит названия параметров оборудования

Таблица 1.16 – it\_prop\_names

|  |  |
| --- | --- |
| Название поля | Тип поля |
| id | Целочисленный |
| name | Строковый |
| measurement | Строковый |
| decor | Целочисленный |
| category\_id | Целочисленный |
| obligatory | Логический |
| status | Целочисленный |
| created | Целочисленный |

* + id – уникальное поле таблицы численного типа, содержащее номер параметра;
	+ name – название параметра;
	+ measurement – измерение;
	+ decor – номер режима отображения;
	+ category\_id – номер категории;
	+ obligatory – обязательность параметра;
	+ status – статус публикации;
	+ created – дата и время создания параметра;
* Таблица **it\_prop\_int** хранит значения целочисленных параметров

Таблица 1.17 – it\_prop\_int

|  |  |
| --- | --- |
| Название поля | Тип поля |
| id | Целочисленный |
| it\_id | Целочисленный |
| name\_id | Целочисленный |
| value | Целочисленный |

* + id – уникальное поле таблицы численного типа, содержащее номер значения;
	+ it\_id – номер оборудования;
	+ name\_id – номер параметра;
	+ value – значение параметра;
* Таблица **it\_prop\_bool** хранит значения логических параметров

Таблица 1.18 – it\_prop\_bool

|  |  |
| --- | --- |
| Название поля | Тип поля |
| id | Целочисленный |
| it\_id | Целочисленный |
| name\_id | Целочисленный |
| value | Логический |

* + id – уникальное поле таблицы численного типа, содержащее номер значения;
	+ it\_id – номер оборудования;
	+ name\_id – номер параметра;
	+ value – значение параметра;
* Таблица **it\_prop\_flo** хранит значения параметров

Таблица 1.19 – it\_prop\_flo

|  |  |
| --- | --- |
| Название поля | Тип поля |
| id | Целочисленный |
| it\_id | Целочисленный |
| name\_id | Целочисленный |
| value | Вещественный |

* + id – уникальное поле таблицы численного типа, содержащее номер значения;
	+ it\_id – номер оборудования;
	+ name\_id – номер параметра;
	+ value – значение параметра;
* Таблица **it\_prop\_str** хранит значения строковых параметров

Таблица 1.20 – it\_prop\_str

|  |  |
| --- | --- |
| Название поля | Тип поля |
| id | Целочисленный |
| it\_id | Целочисленный |
| name\_id | Целочисленный |
| value | Строковый |

* + id – уникальное поле таблицы численного типа, содержащее номер значения;
	+ it\_id – номер оборудования;
	+ name\_id – номер параметра;
	+ value – значение параметра;
* Таблица **it\_prop\_list\_name** хранит значения параметров с типом «список»

Таблица 1.21 – it\_prop\_list\_name

|  |  |
| --- | --- |
| Название поля | Тип поля |
| id | Целочисленный |
| name | Строковый |

* + id – уникальное поле таблицы численного типа, содержащее номер значения списка;
	+ name – значение списка;
* Таблица **it\_prop\_list\_name\_prop** хранит связь параметра со списком его значений

Таблица 1.22 – it\_prop\_list\_name\_prop

|  |  |
| --- | --- |
| Название поля | Тип поля |
| id | Целочисленный |
| it\_prop\_list\_name\_id | Целочисленный |
| it\_prop\_name\_id | Целочисленный |

* + it\_prop\_list\_name\_id – номер значения списка;
	+ it\_prop\_name\_id – номер параметра;
* Таблица **shops** хранит информацию о магазине

Таблица 1.23 – shops

|  |  |
| --- | --- |
| Название поля | Тип поля |
| id | Целочисленный |
| user\_id | Целочисленный |
| city\_id | Целочисленный |
| name | Строковый |
| logo\_path | Строковый |
| description | Текст |
| slug | Строковый |
| status | Целочисленный |
| created | Целочисленный |

* + id – уникальное поле таблицы численного типа, содержащее номер магазина;
	+ user\_id – номер пользователя, который представляет данный магазин (0 если нет);
	+ city\_id – номер города в котором расположен магазин;
	+ name – название магазина;
	+ logo\_path – путь к файлу с логотипом магазина;
	+ description – описание магазина;
	+ slug – название магазина латиницой;
	+ status – статус публикации;
	+ created – дата и время добавления магазина;
* Таблица **sh**\_**goods** хранит список товаров пользователей или магазинов

Таблица 1.24 – sh\_goods

|  |  |
| --- | --- |
| Название поля | Тип поля |
| id | Целочисленный |
| it\_id | Целочисленный |
| user\_id | Целочисленный |
| shop\_id | Целочисленный |
| used | Логический |
| price | Вещественный |
| comment | Текст |
| created | Целочисленный |

* + id – уникальное поле таблицы численного типа, содержащее номер товара;
	+ it\_id – номер оборудования;
	+ user\_id – номер пользователя;
	+ shop\_id – номер магазина (0 если нет);
	+ used – б. у. Товар или нет;
	+ price – цена товара;
	+ comment – комментарий к товару;
	+ created – дата добавления товара;
* Таблица **galleries** хранит список галерей

Таблица 1.25 – galleries

|  |  |
| --- | --- |
| Название поля | Тип поля |
| Id | Целочисленный |
| date\_added | Целочисленный |
| title | Строковый |
| title\_slug | Строковый |
| description | Текст |
| is\_public | Логический |

* + id – уникальное поле таблицы численного типа, содержащее номер галереи;
	+ date\_added – дата создания галереи;
	+ title – название галереи;
	+ title\_slug – название галереи латиницей;
	+ description – описание галереи;
	+ is\_public – публичная ли галерея;
* Таблица **gal**\_**photos** хранит список фотографий

Таблица 1.26 – gal\_photos

|  |  |
| --- | --- |
| Название поля | Тип поля |
| id | Целочисленный |
| photo\_path | Строковый |
| user\_id | Целочисленный |
| date\_tacken | Целочисленный |
| view\_count | Целочисленный |
| title | Строковый |
| title\_slug | Строковый |
| caption | Текст |
| date\_added | Целочисленный |
| is\_public | Логический |

* + id – уникальное поле таблицы численного типа, содержащее номер фотографии;
	+ photo\_path – путь до файла с фотографией;
	+ user\_id – номер пользователя, добавившего фотографию;
	+ date\_tacken – дата и время публикации фотографии;
	+ view\_count – количество просмотров фотографии;
	+ title – заголовок фотографии;
	+ title\_slug – заголовок фотографии латиницей;
	+ caption – описание фотографии;
	+ date\_added – дата и время добавления фотографии;
	+ is\_public – публична ли фотография;
* Таблица **gal\_photo\_gal** хранит принадлежность фотографий к альбомам

Таблица 1.27 – gal\_photo\_gal

|  |  |
| --- | --- |
| Название поля | Тип поля |
| id | Целочисленный |
| gallery\_id | Целочисленный |
| photo\_id | Целочисленный |

* + id – уникальное поле таблицы численного типа, содержащее номер принадлежности;
	+ gallery\_id – номер галереи;
	+ photo\_id – номер фотографии;
* Таблица **dj\_content\_type** хранит типы содержимого сайта

Таблица 1.28 – dj\_content\_type

|  |  |
| --- | --- |
| Название поля | Тип поля |
| id | Целочисленный |
| name | Строковый |
| app\_label | Строковый |
| model | Строковый |

* + id – уникальное поле таблицы численного типа, содержащее номер типа содержимого;
	+ name – название типа содержимого;
	+ app\_label – название модуля (префикс таблицы);
	+ model – модель (название таблицы);
* Таблица **tags** хранит список тегов

Таблица 1.29 – tags

|  |  |
| --- | --- |
| Название поля | Тип поля |
| id | Целочисленный |
| name | Строковый |

* + Id – уникальное поле таблицы численного типа, содержащее номер тега;
	+ name – название тега;
* Таблица **tag**\_**taggeditem** хранит связи тегов и содержимого сайта

Таблица 1.30 – tag\_taggeditem

|  |  |
| --- | --- |
| Название поля | Тип поля |
| id | Целочисленный |
| tag\_id | Целочисленный |
| content\_type\_id | Целочисленный |
| object\_id | Целочисленный |

* + id – уникальное поле таблицы численного типа, содержащее номер связи;
	+ tag\_id – номер тега;
	+ content\_type\_id – номер типа содержимого;
	+ object\_id – номер содержимого;
* Таблица **comments** хранит комментарии пользователей

Таблица 1.31 – comments

|  |  |
| --- | --- |
| Название поля | Тип поля |
| id | Целочисленный |
| content\_type\_id | Целочисленный |
| object\_pk | Целочисленный |
| user\_id | Целочисленный |
| user\_name | Строковый |
| user\_email | Строковый |
| user\_url | Строковый |
| comment | Текст |
| submit\_date | Целочисленный |
| ip\_address | Целочисленный |
| is\_public | Логический |
| is\_removed | Логический |

* + id – уникальное поле таблицы численного типа, содержащее номер комментария;
	+ content\_type\_id – номер типа содержимого;
	+ object\_pk – номер содержимого;
	+ user\_id – номер пользователя, оставившего комментарий;
	+ user\_name – имя пользователя;
	+ user\_email – адрес электронной почты пользователя;
	+ user\_url – ссылка на сайт пользователя;
	+ comment – текст комментария;
	+ submit\_date – дата и время добавления комментария;
	+ ip\_address – IP адрес пользователя, оставившего комментарий;
	+ is\_public – публичный ли комментарий;
	+ is\_removed – дата и время удаления комментария;
* Таблица **cm\_flags** хранит историю изменений комментария

Таблица 1.32 – cm\_flags

|  |  |
| --- | --- |
| Название поля | Тип поля |
| id | Целочисленный |
| user\_id | Целочисленный |
| comment\_id | Целочисленный |
| flag | Строковый |
| flag\_date | Целочисленный |

* + id – уникальное поле таблицы численного типа, содержащее номер записи;
	+ user\_id – номер пользователя, изменившего комментарий;
	+ comment\_id – номер комментария;
	+ flag – действие, которое произвёл пользователь;
	+ flag\_date – дата и время изменения;
* Таблица **recomindations** хранит рекомендации пользователей

Таблица 1.33 – recomindations

|  |  |
| --- | --- |
| Название поля | Тип поля |
| id | Целочисленный |
| content\_type\_id | Целочисленный |
| object\_id | Целочисленный |
| user\_id | Целочисленный |
| plus | Текст |
| minus | Текст |
| general | Текст |
| positive | Целочисленный |
| negative | Целочисленный |
| status | Целочисленный |
| created | Целочисленный |

* + id – уникальное поле таблицы численного типа, содержащее номер рекомендации;
	+ content\_type\_id – номер типа содержимого;
	+ object\_id – номер содержимого;
	+ user\_id – номер пользователя, оставившего рекомендацию;
	+ plus – плюсы;
	+ minus – минусы;
	+ general – прочие;
	+ positive – количество пользователей, посчитавших рекомендацию полезной;
	+ negative – количество пользователей, посчитавших рекомендацию безполезной;
	+ status – статус публикации;
	+ created – дата и время создания рекомендации;
* Таблица **articles** хранит статьи пользователей

Таблица 1.34 – articles

|  |  |
| --- | --- |
| Название поля | Тип поля |
| id | Целочисленный |
| title | Строковый |
| text | Текст |
| short\_text | Текст |
| is\_remote | Логический |
| url | Строковый |
| user\_id | Целочисленный |
| slug | Строковый |
| status | Целочисленный |
| created | Целочисленный |

* + id – уникальное поле таблицы численного типа, содержащее номер статьи;
	+ title – заголовок статьи;
	+ text – текст статьи;
	+ short\_text – сокращённый текст статьи;
	+ is\_remote – статья расположена на удалённом ресурсе;
	+ url – адрес ресурса на котором расположена статья;
	+ user\_id – номер пользователя, опубликовавшего статью;
	+ slug – заголовок статьи латиницой;
	+ status – статус публикации;
	+ created – дата и время добавления статьи;
* Таблица **rating\_scores** хранит рейтинг содержимого сайта

Таблица 1.35 – rating\_scores

|  |  |
| --- | --- |
| Название поля | Тип поля |
| id | Целочисленный |
| content\_type\_id | Целочисленный |
| object\_id | Целочисленный |
| score | Вещественный |
| votes | Целочисленный |

* + id – уникальное поле таблицы численного типа, содержащее номер записи;
	+ content\_type\_id – номер типа содержимого;
	+ object\_id – номер содержимого;
	+ score – средняя оценка;
	+ votes – количество проголосовавших пользователей;
* Таблица **rating\_votes** хранит оценки содержимого сайта пользователями

Таблица 1.36 – rating\_votes

|  |  |
| --- | --- |
| Название поля | Тип поля |
| id | Целочисленный |
| content\_type\_id | Целочисленный |
| object\_id | Целочисленный |
| score | Вещественный |
| user\_id | Целочисленный |
| ip\_address | Целочисленный |

* + id – уникальное поле таблицы численного типа, содержащее номер оценки;
	+ content\_type\_id – номер типа содержимого;
	+ object\_id – номер содержимого;
	+ score – оценка;
	+ user\_id – номер пользователя;
	+ ip\_address – IP адрес пользователя;
* Таблица **rating\_weight** хранит вес содержимого сайта

Таблица 1.37 – rating\_weight

|  |  |
| --- | --- |
| Название поля | Тип поля |
| id | Целочисленный |
| content\_type\_id | Целочисленный |
| object\_id | Целочисленный |
| weight | Целочисленный |
| votes | Целочисленный |

* + id – уникальное поле таблицы численного типа, содержащее номер записи;
	+ content\_type\_id – номер типа содержимого;
	+ object\_id – номер содержимого;
	+ weight – вес содержимого;
	+ votes – количество пользователей, оценивщих вес содержимого;
* Таблица **rating\_weight\_vote** хранит оценки веса содержимого пользователями

### 1.2.3 Диаграмма классов

На диаграмме классов основного модуля (см. рисунок 9) классы, которые будут реализовывать взаимодействие с базой данных посредством ORM Django. Все классы будут наследоваться от стандартного класса Model, который включён в состав фреймворка. В классе Model уже реализованы необходимые функции для работы с базой данных.


## Рисунок 9 – Диаграмма классов

## 1.3 Реализация системы

В связи с большим объёмом работ окончательная реализация системы ещё не завершена. На данный момент реализованы все классы системы, методы взаимодействия классов между собой и методы их взаимодействия с базой данных. С точки зрения архитектуры программного обеспечения MVC полностью реализованы уровень модели (Model) и уровень контроллера(Controller), частично реализован уровень отображения (View).

### 1.3.1 План контроля изменений

В ходе реализации программного средства производился контроль промежуточных версий. Для контроля версий использовалась программа Git GUI – клиент для системы контроля версий Git, выполненный как расширение оболочки Linux.

### 1.3.2 Интерфейс системы

В соответствии с требованиями, выдвинутыми к дизайну системы, был разработан интерфейс системы.

Основная структура сайта состоит из пяти блоков:

1. «Шапка» сайта – в нём располагается логотип сайта, форма регистрации или краткая информация об авторизовавшемся пользователе.
2. Основное меню – горизонтальное меню навигации по основным разделам сайта.
3. Дополнительное меню – вертикальное меню содержит пункты которого зависят от раздела, который открыт в данный момент.
4. Содержимое сайта – основной блок в котором выводится информация, запрашиваемая пользователем.
5. «Подвал» – блок, в котором находятся ссылки на дополнительную информацию о сайте.

В качестве примера рассмотрим интерфейс страницы просмотра информации об оборудовании. На этой странице оказался самый сложный с точки зрения реализации интерфейс.

Основной проблемой при создании интерфейса было большое количество информации об оборудовании, которую надо было разместить на одной странице. Было принято решение разделить всю информацию блоки. При этом видимостью большинства из них может управлять пользователь.

Рассмотрим некоторые блоки более подробно:

1. Обязательный – Блок в котором отображаются обязательный характеристики оборудования, диапазон цен на него, средняя цена, средняя оценка, изображение, форма оценки и ссылки на добавление в личные списки пользователя. Пользователь не может управлять его видимостью.
2. Текстовое описание – Краткое описание оборудования. Пользователь может управлять видимостью блока. Нажав на заголовок блока его можно скрыть или раскрыть. В тексте выделяются ключевые слова в виде ссылок, при нажатии на которые пользователь переходит на другие страницы сайта.
3. Облако тегов – перечень тегов описывающих данное оборудование. Каждый из тегов имеет вес, который указывает на степень соответствия оборудования данному тегу. Пользователь может управлять видимостью блока. Так же пользователь может влиять на вес тега, нажимая на кнопки («+» и «–») справа от названия тега. Изменяя вес тега пользователь либо соглашается, что оборудование соответствует тегу («+»), либо не соглашается («–»).
4. Информация из других разделов и цены – Пользователь не может управлять видимостью блока. Блок состоит из четырёх вкладок:
* Рекомендации, которые оставляют пользователи. Отображаются две самые полезные рекомендации. Пользователь может влиять на уровень полезности, нажимая на кнопки «Да» или «Нет». Снизу находятся две ссылки, при помощи которых можно добавить рекомендацию или посмотреть все рекомендации.

2. Технологический раздел

##

## 2.1 Выбор средств разработки

###

### 2.1.1 Выбор языка программирования

Для реализации данной системы были выбраны следующие языки и библиотеки:

1. HTML (от англ. HyperText Markup Language – «язык разметки гипертекста») – стандартный язык разметки документов в Internet. Язык HTML интерпретируется браузером и отображается в виде документа, в удобной для человека форме.
2. Python – высокоуровневый язык программирования общего назначения с акцентом на производительность разработчика и читаемость кода. Синтаксис ядра Python минималистичен. В то же время стандартная библиотека включает большой объём полезных функций. Python поддерживает несколько парадигм программирования, в том числе структурное, объектно-ориентированное, функциональное, императивное и аспектно-ориентированное. Основные архитектурные черты – динамическая типизация, автоматическое управление памятью, полная интроспекция, механизм обработки исключений, поддержка многопоточных вычислений и удобные высокоуровневые структуры данных. Код в Питоне организовывается в функции и классы, которые могут объединяться в модули (которые в свою очередь могут быть объединены в пакеты).
3. Django (Джанго) – свободный фреймворк для веб-приложений на языке Python. Django облдает MVC структурой. Работа с базами данных осуществляется с помощью ORM.
4. JavaScript – объектно-ориентированный скриптовый язык программирования. Необходим для сценариев, которые обеспечивают интерактивность веб-страницам, и для реализации технологии Ajax.
5. jQuery – библиотека JavaScript, фокусирующаяся на взаимодействии JavaScript и HTML. Библиотека jQuery помогает легко получать доступ к любому элементу DOM, обращаться к атрибутам и содержимому элементов DOM, манипулировать ими. Также библиотека jQuery предоставляет удобный API по работе с Ajax.
6. CSS (от англ. Cascading Style Sheets – каскадные таблицы стилей) – технология описания внешнего вида документа, написанного языком разметки.
7. SQL (от англ. Structured Query Language – «язык структурированных запросов») – универсальный компьютерный язык, применяемый для создания, модификации и управления данными в реляционных базах данных.

### 2.1.2 Выбор ОС

Спроектированная система может быть реализована в большинстве современных операционных системах, таких как: Windows XP, Windows 7, системах семейства Linux, поскольку всё необходимое программное обеспечение данными системами поддерживается. Под необходимым программным обеспечением понимается сервер баз данных MySQL, Web-сервер Apache, интерпретатор языка Python.

В качестве операционной системы для разработки была выбрана ОС Ubuntu 10.04.

### 2.1.3 Выбор среды разработки и системы управления версиями

В качестве основной среды разработки была выбрана Aptana Studio.

Aptana Studio – кроссплатформенная, свободная, c открытым исходным кодом интегрированная среда разработки приложений (IDE) для создания динамических веб-приложений. Включает в себя такие функции как автодополнение набираемых конструкций на лету для кода JavaScript, HTML, и CSS, выделение цветом JavaScript, HTML и CSS кода, предупреждения об ошибках и возможность для настройки и расширения пользовательского интерфейса.

С помощью дополнительных плагинов Aptana Studio также поддерживает разработку для следующих языков и платформ: Ruby on Rails, PHP, Python, Adobe Air и Apple iPhone.

Помимо самостоятельных версий для Windows NT, Mac OS и GNU/Linux, Aptana доступна в качестве плагина для платформы Eclipse.

Основные особенности Aptana Studio:

* Подсветка текста и возможность сворачивания блоков, согласно синтаксису языка программирования
* Настраиваемый пользователем режим подсветки синтаксиса
* Авто-завершение набираемого слова
* Автоматическое определение состояния файла
* Поддержка большого количества языков
* Закладки
* Интеграция с фреймфорком Django
* Интеграция с системой управления версиями Git

В качестве системы контроля версий была использована Git.

Git (произн. «гит») – распределённая система управления версиями файлов. Удалённый доступ к репозиториям Git обеспечивается git-daemon, SSH- или HTTP-сервером. TCP-сервис git-daemon входит в дистрибутив Git и является наряду с SSH наиболее распространённым и надёжным методом доступа. Метод доступа по HTTP, несмотря на ряд ограничений, очень популярен в контролируемых сетях, потому что позволяет использование существующих конфигураций сетевых фильтров.

### 2.1.4 Требования к аппаратному обеспечению

Для работы с системой необходимо любое устройство, имеющее доступ к сети интернет, и с установленным WEB-браузером, поддерживающим JavaScript. Это может быть персональный компьютер, ноутбук, мобильное устройство и т.д.

## 2.2 Разработка эксплуатационной документации

###

### Руководство администратора

Установка системы будет заключаться в установке и настройке программного обеспечения сервера. Настройки клиентов не требуется, так как единственным требованием для работы программы, является наличие программы браузера. Программа браузер входит в начальный комплект любой современной операционной системы и дополнительных настроек не требует.

Для обеспечения работы на сервере должно быть установлено следующее программное обеспечение:

* WEB-сервер Apache;
* mod Python для Apache;
* СУБД MySQL;

В данном руководстве будет описан процесс установки системы на сервер, управляемый ОС Linux.

Для установки необходимого ПО выплоним в терминале следующую команду:

sudo apt-get install apache2 libapache2-mod-python mysql-server mysql-client python-mysqldb

Во время установки следуйте несложным инструкциям установщика.

Далее необходимо создать пользователя в СУБД MySQL саму базу данных:

mysql – user=root mysql

mysql> CREATE USER 'x5'@'localhost' IDENTIFIED BY '2307881310';

mysql> CREATE DATABASE `x5\_photohq` DEFAULT CHARSET UTF8;

mysql>exit

Теперь необходимо записать на диск саму систему, это мы сделаем при помощи системы контроля версий Git:

sudo apt-get install git

cd /var/www

git clone ssh://x5@93.157.127.130/home/x5/photohq.ru

x5@93.157.127.130 password:2307881310

Теперь в папке /var/www/photohq.ru находятся исходные тексты системы. Для инициализации системы необходимо выполнить следующие команды:

cd /var/www/pahotohq.ru/django\_projects/photohq

manage.py syncdb

После выполнения данных команды будут созданы все необходимые таблицы в базе данных x5\_photohq и будет создана учётная запись админиcтратора. Далее необходимо настроить web-сервер, для этого надо созадть виртуальный хост со следующими параметрами:

<location «/">

SetHandler python-program

PythonHandler django.core.handlers.modpython

SetEnv DJANGO\_SETTINGS\_MODULE photohq.settings

PythonPath «['/var/www/photohq.ru/django\_projects'] + sys.path»

</location>

<location «/media»>

SetHandler None

</location>

<location «/admin\_media»>

SetHandler None

</location>

<location «/phpmyadmin»>

SetHandler None

</location >

<locationmatch». (jpg|gif|png)$»>

SetHandler None

</locationmatch>

Так же надо добавить модули в папку интерпретатора Python:

ln – s `pwd` /var/www/photohq.ru/django\_src/django /usr/lib/python2.6/dist-packages/django

ln – s `pwd` /var/www/photohq.ru/django\_projects/photohq /usr/lib/python2.6/dist-packages/photohq

На этом установка закончена. По адресу, который вы указали при создании виртуального хоста будет доступна система.

# 3. Технико-экономический раздел

Для проведения успешного проекта важно понять объем предстоящих работ, требуемые ресурсы, предстоящие задачи, необходимые усилия. При планировании программного проекта нужно оценить людские ресурсы, продолжительность, стоимость.

## 3.1 Определение основных модулей

Проектируемая система предполагает наличие следующих модулей:

Interface – модули, реализующие интерфейс сайта.

Items – модули, реализующие добавление, редактирование, удаление, поиск и фильтрацию оборудования.

Users – модули, реализующие регистрацию, авторизацию пользователя в системе, страницу пользователя и списки оборудования пользователя.

Shops – модуль, реализующий добавление и редактирование магазинов и товаров. Так же модуль обеспечивает просмотр информации о магазинах и товарах.

Gallery – модуль, реализующий добавление альбомов и фотографий.

Articles – модуль, реализующий добавление, редактирование и удаление статей.

Comments – модуль, реализующий добавление, редактирование и удаление комментариев.

Rating – модуль, реализующий оценку рейтингов и весов содержимого. Так же этот модуль реализует добавление, редактирование и удаление рекомендаций пользователей.

Taging – модуль, реализующий добавление, редактирование и удаление тегов, связывание тегов и содержимого.

## 3.2 Расчет трудозатрат и продолжительности разработки программного проекта

В ходе предварительного анализа были определены 9 модулей. Определим приблизительно их размеры в строках кода.

Таблица 3.1 Оценка объема в строках кода

|  |  |
| --- | --- |
| Имя модуля | Оценка объема в строках исходного кода |
| Пессимистическая | Наиболее вероятностная | Оптимистическая |
| Interface | 300 | 200 | 150 |
| Items | 800 | 600 | 500 |
| Users | 300 | 200 | 150 |
| Shops | 300 | 200 | 150 |
| Gallery | 200 | 150 | 100 |
| Articles | 200 | 150 | 100 |
| Comments | 200 | 150 | 100 |
| Rating | 150 | 100 | 80 |
| Taging | 100 | 70 | 50 |
| Итого | 2550 | 1820 | 1380 |

Рассматриваемый проект может быть отнесен к распространенному типу программных систем вследствие своих малых размеров. Значение коэффициентов уровней базовой модели КОМОСТ для данного программного проекта: a=2.4, b=1.05, c=2.5, d=0.38.

Уравнения для расчета трудозатрат и продолжительности разработки проектируемой системы:

ТР=a\*(РП)b (3.1)

В этих уравнениях трудоемкость ТР выражается в человеко-месяцах, размер программного продукта РП в тысячах строк исходного кода, а длительность ДР разработки в месяцах.

Рассчитаем трудоемкость этапа анализа и проектирования программного проекта:

= 6.41 (чел.-мес) = 192 (чел.-дней)

 = 4,5 (чел.-мес) = 135 (чел.-дней)

 = 3,37 (чел.-мес) = 101 (чел.-дней)

Рассчитаем трудоемкость программирования функциональных блоков:

Interface

 = 0,68 (чел.-мес) = 20 (чел.-дней)

 = 0,44 (чел.-мес) = 13 (чел.-дней)

 = 0,32 (чел.-мес) = 10 (чел.-дней)

2) Items

 = 1,9 (чел.-мес) = 57 (чел.-дней)

 = 1,4 (чел.-мес) = 42 (чел.-дней)

 = 1,16 (чел.-мес) = 35 (чел.-дней)

3) Users

 = 0,68 (чел.-мес) = 20 (чел.-дней)

 = 0,44 (чел.-мес) = 13 (чел.-дней)

 = 0,32 (чел.-мес) = 10 (чел.-дней)

4) Shops

 = 0,68 (чел.-мес) = 20 (чел.-дней)

 = 0,44 (чел.-мес) = 13 (чел.-дней)

 = 0,32 (чел.-мес) = 10 (чел.-дней)

5) Gallery

 = 0,44 (чел.-мес) = 13 (чел.-дней)

 = 0,32 (чел.-мес) = 10 (чел.-дней)

 = 0,21 (чел.-мес) = 6 (чел.-дней)

6) Articles

 = 0,44 (чел.-мес) = 13 (чел.-дней)

 = 0,32 (чел.-мес) = 10 (чел.-дней)

 = 0,21 (чел.-мес) = 6 (чел.-дней)

7) Comments

 = 0,44 (чел.-мес) = 13 (чел.-дней)

 = 0,32 (чел.-мес) = 10 (чел.-дней)

 = 0,21 (чел.-мес) = 6 (чел.-дней)

8) Rating

 = 0,32 (чел.-мес) = 10 (чел.-дней)

 = 0,21 (чел.-мес) = 6 (чел.-дней)

 = 0,16 (чел.-мес) = 5 (чел.-дней)

9) Taging

 = 0,20 (чел.-мес) = 6 (чел.-дней)

 = 0,15 (чел.-мес) = 5 (чел.-дней)

 = 0,10 (чел.-мес) = 3 (чел.-дней)

Суммирование результатов по всем процедурам дает следующие результаты:

 = 15.14 (чел.-мес) = 454 (чел.-дней)

 = 10,4 (чел.-мес) = 312 (чел.-дней)

 = 7,63 (чел.-мес) = 229 (чел.-дней)

Для малых проектов на основе опытных данных модели КОМОСТ было установлено, что этап испытаний занимает 25 процентов от времени, затраченного на программирование.

Трудоемкость испытаний:

 = 114 (чел.-дней)

 = 78 (чел.-дней)

 = 57 (чел.-дней)

Оценка общей трудоемкости разработки получаются в результате суммированием оценок трудоемкости анализа, проектирования, программирования и испытаний:

 192 + 454 + 114 = 760 (чел.-дней) = 25,33 (чел.-мес)

 135 + 312 + 78 = 525 (чел.-дней) = 17,5 (чел.-мес)

 101 + 229 + 57 = 387 (чел.-дней) = 12,9 (чел.-мес)


## 3.3 Расчет себестоимости разработки программного обеспечения

Себестоимость разработки программного обеспечения включает в себя следующие виды материальных затрат:

1. расходные материалы и комплектующие изделия;
2. заработная плата разработчика ПО;
3. ЕСН – единый социальный налог;
4. накладные расходы;
5. содержание и эксплуатация вычислительных средств.

Расчет затрат на материалы и комплектующие изделия

При разработке ПО в расходные материалы включаются: картриджи, диски, бумага и т.п. Затраты на материалы являются суммой затрат на материалы отдельных видов.

Затраты на материалы каждого вида рассчитываются по формуле (3.9):

 руб., (3.2)

где – стоимость данного вида материалов, руб.;

 – цена за принятую единицу, руб.;

 – норма расходов материалов, шт. (кг);

Затраты на материалы приведены в таблице 3.6:

Таблица 3.2 – Смета затрат на материалы для разработки ПО

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование материала и единица измерения | Цена за единицу материалов, руб. | Количество материалов, шт. | Стоимость, руб. |
| CD-R диск 700 Мб, TDK, шт. | 21 | 1 | 21 |
| Бумага офисная (80 г./см2, 500 листов) Canon Office, уп. | 140 | 1 | 140 |
| Картридж HP51640A для принтера HP 1200 DJ, шт. | 909 | 1 | 909 |
| Итого: | 1070 |

Общие затраты на материалы составили 1070 рублей.

**Расчет заработной платы на создание программного обеспечения**

Заработная плата на создание ПО включает в себя основную и дополнительную заработную плату и определяется по формуле (3.3):

 (3.3)

где – основная заработная плата, руб.;

– дополнительная заработная плата, руб.

Заработная плата основная включает в себя заработную плату по договору, плюс премию и рассчитывается по формуле (3.4):

 (3.4)

где – заработная плата по договору, руб.;

– премия в размере 10% от заработной платы по договору.

 (3.5)

где – ставка программиста по договору, р.;

– общая трудоемкость создания ПО.

Часовая тарифная ставка программиста по договору подряда составляет 35000 р.\мес., исходя из 8-ми часового рабочего дня (среднее число рабочих дней в месяце 22). Подставив в формулу (3.5) часовую ставку по договору и трудоемкость создания ПО (=17,5 мес) получим:

руб.,

Таким образом, основная заработная плата (с учетом премии в размере 30% от заработной платы по тарифу) составит:

 руб.

Дополнительная заработная плата устанавливается в размере 10% от основной. В выпускной квалификационной работе принимать Зд в размере от основной заработной платы и рассчитывается по формуле (3.6):

 руб. (3.6)

Таким образом, общие затраты на заработную плату составляют:

 руб.

Расчет единого социального налога

Единый социальный налог рассчитывается по формуле:

 руб., (3.7)

где - ставка единого социального налога равная 26,2%.

**Расчет накладных расходов**

Накладные расходы – это расходы на управление и хозяйственное обслуживание. Величина накладных расходов устанавливается в процентах от основной заработной платы и рассчитывается по формуле (3.8).

Процент накладных расходов у ИП Горлов П.С. принимается равным 10% от основной заработной платы:

 руб. (3.8)

**Расчет затрат на содержание и эксплуатацию вычислительных средств:**

Годовые затраты на эксплуатацию вычислительных средств определяются по формуле:

 руб., (3.9)

где – стоимость электроэнергии;

– затраты на техническое обслуживание и ремонт;

– амортизационные отчисления.

Стоимость электроэнергии, потребляемой вычислительной аппаратурой, определяется по формуле (3.10):

 руб., (3.10)

где – номинальная потребляемая мощность компьютера (=0.4 кВт);

– коэффициент загрузки (=0,8);

– общее время использования ЭВМ (=3120 часа);

– стоимость одного кВт-часа электроэнергии (=0,88 руб.).

Затраты на техническое обслуживание и ремонт вычислительной аппаратуры принимаются равными 5% от ее балансовой стоимости и определяются по формуле (3.11):

 руб., (3.11)

где – балансовая стоимость вычислительной аппаратуры,

– годовой эффективный фонд времени =1878 часов (см. формулу (3.13)).

Балансовая стоимость вычислительной аппаратуры определяется по формуле (3.12):

 руб., (3.12)

где – цена вычислительной аппаратуры (=40000 руб.),

– ставка затрат на доставку и монтаж (=0,1).

Годовой эффективный фонд времени определяется по формуле (3.13):

 часов, (3.13)

где – число календарных дней,

– число выходных дней в году,

– число праздничных дней в году,

– продолжительность рабочего дня, (=8 час.),

– планируемый процент потерь времени на ремонт вычислительной аппаратуры (=3%).

Амортизационные отчисления определяются по формуле:

 руб., (3.14)

где – норма амортизации (=30%).

Примечание: Для электронно-вычислительной техники в соответствии с Постановлением Правительства РФ 01.01.02 срок полезного использования составляет 3–5 лет. Следовательно, норма амортизации может быть принята =(20ч30)%.

Результаты расчета себестоимости разработки программного обеспечения сведены в таблицу 3.7.

Таблица 3.3 – Сметная калькуляция на разработку программных средств

|  |  |
| --- | --- |
| Вид затрат | Сумма рублей |
| Материалы и комплектующие изделия | 1070 |
| Основная заработная плата | 673200 |
| Дополнительная заработная плата | 67320 |
| Единый социальный налог | 194016 |
| Эксплуатационные затраты | Затраты на электроэнергию | 878 |
| Амортизация | 22000 |
| Затраты на ремонт и техническое обслуживание | 3654 |
| Накладные расходы | 67320 |
| Итого: | 1029458 |

#

# Заключение

В ходе выполнения выпускной квалификационной работы был произведен анализ предметной области, включающий подробный обзор аналогов системы, выявлены их недостатки и преимущества.

Основываясь на обзоре аналогов системы, были разработаны требования к функциональности системы, к дизайну системы, к шрифтам и средствам просмотра, а также выявлены требования пользователей к системе.

На основании составленных требований пользователей к системе была разработана общая структура сайта.

Разработаны так же инфологическая и даталогическая модель базы данных системы, разработана диаграмма классов системы.

Система была практически полностью реализована, за исключением отображений некоторых разделов.

Также была рассчитана трудоёмкость, себестоимость разработанной системы и составлен перечень мер по обеспечению безопасности при эксплуатации системы.

# Список использованных источников

1. Марк Лутц. Изучаем Python. Издательство: Символ-Плюс, 2009 г. Мягкая обложка, 848 стр.
2. Джефф Форсье, Пол Биссекс, Уэсли Чан. Django. Разработка веб-приложений на Python. Издательство: Символ-Плюс, 2009 г. Мягкая обложка, 456 стр.
3. Вигерс К. Разработка требований к программному обеспечению [Текст]. М.: Издательско-торговый дом «Русская редакция», 2004. 576 с.
4. Фаулер М., Скотт Д. UML. Основы. Второе издание [Текст]. Пер. с англ. СПб.: Символ-Плюс, 2002. 192 с.
5. Карпова Т.С. Базы данных: модели, разработка, реализация [Текст]. − СПб.: Питер, 2002. − 304 с.
6. ГОСТ 30403–96. Конструкции строительные. Метод определения пожарной опасности [Электронный ресурс]. – Введ. 1996–07–01. – Адрес в Интернет: http://www.gosthelp.ru.
7. Орлов С.А. Технологии разработки программного обеспечения [Текст]. − СПб.: Питер, 2002. − 464 с.
8. Исаков Л.С., Корчагина Р.Л., Ткалин А.И. Оценка экономической эффективности программных средств. Учебное пособие по выполнению экономического раздела выпускной квалификационной работы [Текст]. – СПб.: БГТУ, 2005. – 22 с
9. СНиП 21–01–97. Пожарная безопасность зданий и сооружений [Электронный ресурс]. введ. 1998–01–01. – Адрес в Интернет: http://www.mbty.ru.
10. ГОСТ 12.1.004–91. Пожарная техника для защиты объектов. Основные виды. Размещение и обслуживание [Электронный ресурс]. – Введ. 1985–01–01. Адрес в Интернет: http://www.mbty.ru.
11. ГОСТ 12.0.003–74. Опасные и вредные производственные факторы [Электронный ресурс]. Введ. 1976–01–01. – Адрес в Интернет: http://www.docload.ru/