Содержание

Введение 3

1 ОБЩАЯ ЧАСТЬ 4

* 1. Анализ предметной области 4
  2. Анализ конкурентов 5
  3. Исследование целевой аудитории 7
  4. Выбор средств и сред разработки 8

2 СПЕЦИАЛЬНАЯ ЧАСТЬ 19

2.1 Описание разрабатываемых функций 19

2.2 Требования к информационной системе 26

2.2.1 Описание требований к информационной системе 26

2.3 Диаграмма вариантов использования 28

2.4 Диаграмма состояний 30

2.5 Схема данных 35

2.6 Пользовательские сценарии 37

2.7 Прототипы основных интерфейсов 39

3 ЭКОНОМИЧЕСКАЯ ЧАСТЬ 40

3.1 Расчет затрат на разработку и эксплуатацию 41

3.2 Расчет эк. эффекта и определение срока окупаемости 50

Заключение 53

Перечень используемых источников 54

Приложение А – Листинг программы

Приложение Б – Скриншоты сайта

Введение

Преддипломная практика является необходимым этапом обучения, с целью формирования у обучающихся требуемых компетенций. Ее ценность заключается в приобретении новых и закреплении уже полученных знаний в условиях реального производства. В период ее прохождения обучающиеся должны закрепить полученные знания в теории и приобрести практические навыки.

В современном мире все больше и больше компаний внедряют в свое производство IT технологии для автоматизации и модернизации некоторых процессов. Так и компания «Умный мир» решает разработать модуль личного кабинета «Заказчик» цифровой платформы.

Целью производственной практики является:

Разработка модуля личного кабинета «Заказчик» цифровой платформы для компании «Умный Мир (SmartWorld)»

В платформе заказчик сможет:

Просматривать все технические задания, посмотреть информацию об одном заказе, а также создавать / редактировать / удалять пункты в задании.

Просматривать все заявки, посмотреть информацию об одной заявке создавать / редактировать / удалять заявки.

Добавлять / редактировать срочность заявки.

Добавлять / редактировать статус завершения заявки.

А также просматривать личный кабинет

1. Общая часть

1.1 Анализ предметной области

Анализ предметной области – начальный этап разработки любого проекта

Одна из первых задач, с решением которых сталкивается разработчик программной системы - это изучение, осмысление и анализ предметной области. Дело в том, что предметная область сильно влияет на все аспекты проекта: требования к системе, взаимодействие с пользователем, модель хранения данных, реализацию

Анализ предметной области данной работы будет включать в себя анализ конкурентов будущего проекта, исследование целевой аудитории и изучение необходимого функционала на сайте.

Анализ конкурентов подразумевает оценку состояния сайтов в первую очередь, имеющих отношение к выбранной в разработке проекта предметной области. Прежде чем приступить к созданию проекта должны быть проанализированы сайты – конкуренты.

1.2 Анализ конкурентов

Сайт конкурент – www.remontnik.ru

Плюсы:

Понятный и простой интерфейс для пользователей.

Возможность быстро создать заявку на работы.

Система обратных отзывов прозрачна и есть отрицательные отзывы о работах.

Минусы:

Отсутствие гарантий - общение заказчика и исполнителя происходит напрямую и нет гарантов что при закупке строительных материалов вас не обманут.

Отсутствует система контроля работы – Заказчик не может контролировать процесс работы через сайт.

Недостаточно информации на сайте - открывая контакт случайно, заказчик не проинформирован о том, что с просматриваемого подрядчика сразу списывается сумма денег.

Сайт конкурент - www.stroitel.club

Плюсы:

Присутствует обратная связь и поддержка.

Разделения услуг по компаниям и мастерам.

Минусы:

Сайт на протоколе HTTP.

Отсутствует система отзывов.

Интерфейс старомодный и не понятный.

Полное отсутствие seo оптимизации

Отсутствует alt почти у всех изображений

Отсутствует система гарантий

Связь заказчика и исполнителя не через сайт

Сайт конкурент – www.russia.superjob.ru/

Плюсы:

Понятный и простой интерфейс

Огромная база работ по разным услугам

Сортировка по опыту работы

Общение происходит через сайт путем отправления резюме

Минусы:

Отсутствует система отзывов.

1.3 Исследование целевой аудитории

Целевая аудитория со стороны заказчика:

В основном это дееспособные люди возрастом 18 – 40 лет.

Возраст заказчиков плавающий в основном это люди 25 – 60 лет.

Географическое положение - Российская федерация.

Сфера деятельности: строительные работы.

Почему заказчикам удобно будет пользоваться модулем?

Модуль:

1. поможет им при помощи посредника составлять грамотное и подробное ТЗ, также понятное и исполнителю работ;
2. обеспечит контроль заявок через сайт;
3. Заказчик сможет не волноваться об времени исполнении работ так как ответственность за их решения ложиться на систему.

1.4 Выбор средств разработки

Язык разработки – PHP.

Фреймворк разработки – Laravel.

Технология – REST API

Система управления базами данных – MySQL.

Интерфейс для работы с СУБД MySQL – PhpMyAdmin.

А так же предоставляю описание технологий и средств разработки и привожу плюсы и минусы каждой для обоснования своего выбора.

Что такое PHP?

PHP - это серверный скриптовый язык программирования, предназначенный для создания сайтов и веб-приложений.

Следующим пунктом, нам нужно перевести эти три буквы - PHP - естественно, что они никак не переводятся - потому, что это аббревиатура. Поэтому, сперва нам нужно разобрать аббревиатуру PHP на слова:

Первоначально аббревиатура PHP означала "Personal Home Pages", что переводится как Персональная домашняя страница. Первая версия PHP была создана Расмусом Лердорфом в 1994 г. и представляла собой набор инструментов для отслеживания посетителей Web-страницы.

Но! после того, как php стал развиваться и уже не соответствовал первоначальной расшифровке - пришлось изменить его значение на:

Hypertext Preprocessor (Препроцессор Гипертекста) – это программная генерация HTML разметки по содержимому базы данных или по любой другой структурированной информации, размещенной на сервере.

Мы не можем просто открыть PHP файл в браузере, как привыкли делать это с HTML файлами. Для работы PHP скриптов необходима специальная программа - веб-сервер.

Сайты на PHP интерактивны. Более удобны и гибки в управлении PHP скрипты предназначены для единовременного выполнения набора команд. Скрипт запустился, выполнил свою задачу и "умер". И так при каждом запуске.

Зачем нужен PHP?

Язык PHP нужен для написания программ, которые работают на сервере и выполняют различные серверные задачи, такие как отправка почты, взаимодействие с базой данных и т.д.

Разберём подробно на примерах.

Практически на любом сайте можно встретить форму.

Формы имеют самое разное назначение: регистрация и авторизация, оформление заказа, оплата чего-либо, комментарии на форуме и множество других функций.

Так вот. С помощью HTML и CSS мы можем определить внешний вид формы. С помощью Javascript мы можем добавить красивые анимационные эффекты.

Но мы не можем обработать данные, которые содержатся в этой форме. Нельзя сохранить информацию о клиенте в единой базе данных, авторизовать пользователя, отправить письмо ему на E-mail и т.д.

Для решения подобных задач нам необходим специальный язык программирования, который может работать на сервере. Одним из таких языков и является PHP.

PHP также используется для автоматической генерации HTML страниц.

Крупные интернет-магазины могут иметь тысячи наименований товаров. Конечно, могли бы создать для каждого товара отдельную html-страницу, но представьте, сколько времени уйдет на создание этих страниц.

Благодаря связке PHP и любой современной СУБД (системы управления базами данных) мы можем создать всего один html-шаблон, а PHP уже будет подставлять в него товары из базы данных.

Язык PHP идеально подходит для создания сайтов и веб-приложений любой

сложности. Блоги, интернет-магазины, лендинги (одностраничные сайты), API и прочие веб-сервисы - всё это можно сделать на PHP.

Также PHP подходит для написания консольных скриптов. Запускать их можно как разово, так и с определённой периодичностью.

Что касается десктопных приложений, здесь у PHP достаточно скудный

инструментарий, поэтому на практике PHP не подходит для написания настольных и мобильных приложений.

Почему именно PHP?

Плюсы:

1. Невысокие расходы на разработку

PHP имеет открытый исходный код и распространяется по GPL лицензии. Он позволяет довольно быстро создавать прототипы и готовые сайты. Существует большое сообщество программистов, которое работает с PHP и делится своим опытом: оказывает поддержку разработчикам и предоставляет большое количество полезной документации, размещенной в свободном доступе, инструкций и руководств по PHP программированию, шаблонов и примеров, а также скриптов и программ на PHP.

1. Быстрое создание Agile веб-приложений

Применение PHP фреймворков позволяет создавать веб-приложения в кратчайшие сроки. Он также делает приложения масштабируемыми и легко обслуживаемыми, позволяя им развиваться с течением времени. Вот список лучших PHP фреймворков, которые мы используем для этих целей: Laravel, CodeIgniter, CakePHP, Symphony, Zend Framework 2, Yii, etc.

1. Кросс-платформенность

PHP совместим с большинством операционных систем, таких как Windows, Linux, Solaris, Mac OSX, etc. Он также поддерживает все серверы, лидирующие по количеству инсталляций (Apache, IIS, iPlanet, etc.).

1. Простое расширение функциональности веб-сайта

Фунциональность вашего сайта можно легко расширить благодаря обилию расширений и библиотек, представленных в интернете в свободном доступе. Добавить кастомные расширения, созданные PHP разработчиками, также не составит труда.

1. Поддержка сайтов для бизеса любого типа

Бизнес-приложение или сайт на базе PHP, большой или маленький, может быть с легкостью разработан в условиях ограниченного времени и бюджета. Более продвинутое PHP программирование подразумевает создание CRM решений, интернет-магазинов, веб-сайтов с базами данных, и др.

PHP широко используется для разработки удобных веб-сайтов и привлечения качественной аудитории на ваш сайт, что в итоге ведет к повышению рентабельности инвестиций (ROI).

1. Документация. Подробная, с примерами, на русском языке.

Для этого языка присутствует документация на русском, и английских языках.

Минусы:

1. Отсутствие лидера — многие технологии и языки имеют лидера, архитектора, который определяет облик технологии, задает вектор развития, принимает решение о том, что должно быть обязательно, а чего не будет никогда (Linux, Python, Ruby и т. п). В РНР лидера нет, многие решения и конструкции — это компромисс заинтересованных групп и исторически сложившихся реалий.
2. Непоследовательный синтаксис — при изучении языка РНР, особенно старой части, основанной на функциях, можно заметить, что часть функций имеет префиксы array\_, str\_, часть не имеет. Параметры функций могут быть расположены не совсем логично и не так, как в другой функции этой же группы.

Laravel - это бесплатный PHP фреймворк с открытым исходным кодом, созданный Тейлором Отвеллом для разработки веб-приложений по архитектурному шаблону MVC.

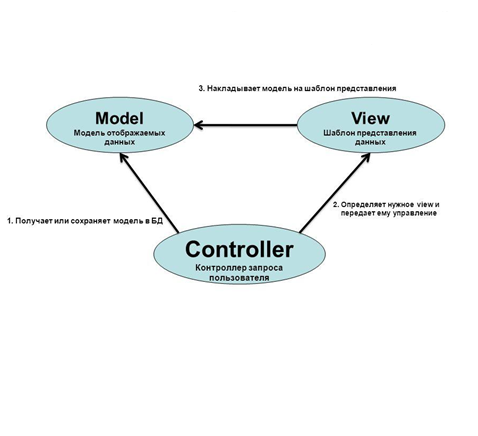


Рисунок 1 – Изображение MVC парадигмы

Почему Laravel?

Плюсы

1. Гибкая система маршрутизации

Одним из самых очевидных плюсов Laravel, является гибкая система маршрутизации, позволяющая составить самые разные проверки маршрута веб-приложения. Вы можете выделить маршруты в специальные группы, использовать пространство имен, указать параметры маршрута, использовать регулярные выражения, настроить поддоменную маршрутизацию и многое другое.

1. Понятный и простой синтаксис

Содержит много синтаксические возможностей, применение которых не влияет на поведение программы, но делает язык понятнее. Синтаксис API фреймворка достаточно простой и понятный. Здесь нет длинных и сложных конструкций, а только краткие и продуманные названия функций.

1. Высокая безопасность

Фреймворк имеет встроенную защиту от sql иньекций и xss атак.

1. Аутентификация

Фреймворк включает в себя встроенные механизмы аутентификации и авторизации пользователей, которую можно перенастроить под свои потребности.

1. Ошибки и исключения

Содержит удобный механизм обработки ошибок и исключений.

1. Интеграция с mail-сервисом

Предоставляет чистый и простой API для работы с почтой чтобы сделать отправку почты через локальную или облачную службу по выбору. В том числе есть механизм для построения очередей отправки почты.

1. Понятная документация
2. Eloquent - cистема объектно-реляционного отображения (ORM) для работы с базами данных

Минусы

1. Нарушение обратной совместимости между различными версиями
2. Требование к знанию английского языка развивается (почти вся документация на английском языке)

Что такое MySQL?

MySQL - свободная реляционная система управления базами данных.

MySQL — это самая популярная из всех крупных серверных БД. Разобраться в ней очень просто, да и в сети о ней можно найти большое количество информации. Хотя MySQL и не пытается полностью реализовать SQL-стандарты, она предлагает широкий функционал. Приложения общаются с базой данных через процесс-демон.

Преимущества

Гибкость и простота в использовании

Вы можете изменить исходный код, чтобы он соответствовал вашим собственным ожиданиям, и вам не нужно ничего платить за этот уровень свободы, включая варианты обновления до расширенной коммерческой версии. Процесс установки относительно прост и не должен занимать более 30 минут.

Высокая производительность

Широкий спектр кластерных серверов поддерживает MySQL. Независимо от того, храните ли вы большие объёмы данных электронной коммерции или выполняете тяжелую бизнес-аналитику, MySQL может помочь вам с оптимальной скоростью.

Промышленный стандарт

Отрасли используют MySQL в течение многих лет, а это означает, что для опытных разработчиков имеются обильные ресурсы. Пользователи MySQL могут рассчитывать на быструю разработку программного обеспечения, а эксперты-фрилансеры, желающие работать в меньшей степени, будут нуждаться в них.

Безопасность

Безопасность ваших данных должна быть вашей главной задачей при выборе правильного программного обеспечения СУРБД. С помощью системы доступа и управления учётными записями MySQL устанавливает высокий уровень безопасности. Доступна проверка на основе хоста и шифрование пароля.

Много функций

MySQL поддерживает большую часть функционала SQL.

Недостатки

Известные ограничения

По определению, MySQL не может сделать всё, что угодно, и в ней присутствуют определённые ограничения функциональности.

Вопросы надёжности

Некоторые операции реализованы менее надёжно, чем в других РСУБД.

Застой в разработке

Хотя MySQL и является open-source продуктом, работа над ней сильно заторможена.

phpMyAdmin - Бесплатный инструмент, создан для непосредственного администрирования. Позволяет выполнять не сложные запросы без их написания. Удобно использовать когда удаленный доступ к базе запрещен например хостером, ведь он запускается на сервере.

Плюсы при администрировании:

Возможность запускать непосредственно на сервере, что удобно при хостинге и когда хостер запрещает удаленный доступ к базе

Почему REST API?

REST - Стиль архитектуры программного обеспечения для распределенных систем.

Плюсы:

1. Независимость от языка/фреймворка/платформы

Клиенты на Java, JS, .Net и т.д.

1. Легкость разработки
2. Легковесность

Недостатки

1. Привязан к протоколу HTTP
2. Специальная часть

2.1 Описание разрабатываемых функций

Для модуля личного кабинета «Заказчик» цифровой платформы для компании «Умный Мир (SmartWorld)» будут использоваться следующие функции:

Функция «Создать ТЗ»

Функция «Изменить ТЗ»

Функция «Удалить ТЗ»

Функция «Вывод ТЗ»

Функция «Просмотр ТЗ»

Функция «Создать заявку»

Функция «Изменить заявку»

Функция «Изменить статус заявки»

Функция «Изменить срочность заявки»

Функция «Удалить заявку»

Функция «Просмотр конкретной заявки»

Функция «Вывод заявок»

Функция «Авторизация пользователя»

Функция «Регистрация пользователя»

Функция «Просмотр личного кабинета пользователя»

Функция «Создать заявку»

Данная функция реализует добавление новых записей в таблицу «Заявки» для пользователя.

Входные данные: введенные данные с клавиатуры в поля: «Название», «Содержимое заявки», «Срочность заявки».

Выходные данные: новая запись в таблице.

Описание работы подпрограммы:

1. Считывает данные введенные с клавиатуры;
2. Если имеются не заполненные, или некорректно заполненные поля, выводится предупреждающее сообщение;
3. Производится запись введённых значений в поля таблицы;
4. Изменения сохраняются в базе данных;
5. Конец работы подпрограммы.

Функция «Изменить заявку»

Данная функция реализует редактирование существующих записей в таблицу «Заявки» для пользователя.

Входные данные: «id» и введенные данные с клавиатуры в поля: «Название», «Содержимое заявки»

Выходные данные: редактированная запись в таблице.

Описание работы подпрограммы:

1. Редактируем данные заявки;
2. Если какое-то поле оказалось пустым, выводится ошибка;
3. Конец работы подпрограммы.

Функция «Изменить статус заявки»

Данная функция реализует редактирование существующих записей в таблицу «Заявки» для пользователя.

Входные данные: «id» и введенные данные с клавиатуры в поля: «Статус заявки».

Выходные данные: редактированная запись в таблице.

Описание работы подпрограммы:

1. Конец работы подпрограммы;
2. Если какое-то поле оказалось пустым, выводится ошибка;
3. Конец работы подпрограммы.

Функция «Изменить срочность заявки»

Данная функция реализует редактирование существующих записей в таблицу «Заявки» для пользователя.

Входные данные: «id» и введенные данные с клавиатуры в поля: «Срочность заявки».

Выходные данные: редактированная запись в таблице.

Описание работы подпрограммы:

1. Редактируем срочность заявки;
2. Если какое-то поле оказалось пустым, выводится ошибка;
3. Конец работы подпрограммы.

Функция «Удалить заявку»

Данная функция реализует удаление существующих записей в таблице «Заявки» для пользователя.

Входные данные: id заявки.

Выходные данные: удаление записи в таблице.

Описание работы подпрограммы:

1. Удаление заявки из списка заявок;
2. Конец работы подпрограммы.

Функция «Просмотр конкретной заявки»

Данная функция реализует вывод существующей записи в таблице «Заявки» для пользователя.

Входные данные: id пользователя, id заявки

Выходные данные: содержимое таблицы в одном экземпляре.

Описание работы подпрограммы:

1. Вывод информации об одной заявке;
2. Конец работы подпрограммы.

Функция «Вывод заявок»

Данная функция реализует вывод существующих записей в таблице «Заявки» для пользователя.

Входные данные: id пользователя

Выходные данные: содержимое таблицы.

Описание работы подпрограммы:

1. Вывод информации из таблицы Заявки;
2. Конец работы подпрограммы.

Функция «Авторизация пользователя»

Данная функция реализует авторизацию существующего пользователя.

Входные данные: введенные данные с клавиатуры в поля: «Логин»,

«Пароль»

Выходные данные: токен.

Описание работы подпрограммы:

1. Проверка введенных данных с клавиатуры;
2. Если поля не заполнены или такого пользователя нет выводится ошибка;
3. Конец работы подпрограммы.

Функция «Регистрация пользователя»

Данная функция реализует создание нового пользователя в таблице «Пользователи».

Входные данные: введенные данные с клавиатуры в поля: «Логин», «Пароль», «Email», «Повтор пароля».

Выходные данные: -.

Описание работы подпрограммы:

1. Считывает данные введенные с клавиатуры;
2. Если имеются не заполненные, или некорректно заполненные поля, выводится предупреждающее сообщение;
3. Производится запись введённых значений в поля таблицы;
4. Изменения сохраняются в базе данных.

Функция «Личный кабинет пользователя»

Данная функция реализует вывод существующих данных для пользователя в таблице «Пользователи».

Входные данные: id пользователя

Выходные данные: содержимое таблицы

Описание работы подпрограммы:

1. Проверка на авторизован ли пользователь;
2. Если пользователь не авторизован его переадресует на страницу авторизации;
3. Вывод личных данных текущего пользователя;
4. Конец работы подпрограммы.

Требования к информационной системе

2.2.1 Описание требований к информационной системе

Браузер, который необходим:

1. Google Chrome (версия 61 и выше);
2. Opera (версия 51 и выше);
3. Mozila Firefox (версия 59 и выше);
4. Safari (версия 13 и выше);
5. Microsoft Edge (версия 18 и выше).

Операционные системы:

*Windows*

1. Windows 10;
2. Windows 8;
3. Windows 7.

*MacOS*

OS X Yosemite 10.10 или более поздней версии.

*Linux*

1. Ubuntu 18.03 (64-разрядная версия) или старше;
2. Debian 8 или старше;
3. openSUSE 13.3 или старше;
4. Fedora Linux 24 или старше.

Минимальные характеристики к компьютеру:

Таблица 1 – Минимальные характеристики к компьютеру.

|  |  |
| --- | --- |
| OC | Минимальные характеристики |
| Windows | Операционная система: Windows 7 (32 или 64 bit);  Процессор: двухъядерный Intel Core i3-i5 от 2,5-3,2GHz;  Оперативная память: 6 Gb и выше;  Жесткий диск: SSD от 128GB (при установке используется около 100 Мбайт) |
| Mac OS | Операционная система: Mac 10.6–10.11;  Macbook Pro 2011, Macbook Air 2012 или более новые версии |

2.3 Диаграмма вариантов использования



Рисунок 2.3.1 – Диаграмма вариантов использования гость



Рисунок 2.3.2 – Диаграмма вариантов использования заказчик

2.4 Диаграмма состояний



Рисунок 2.4.1 – Диаграмма состояний общий вид



Рисунок 2.4.2 – Диаграмма состояний авторизация. Рисунок 2.4.3. – Регистрация



Рисунок 2.4.4 – Диаграмма состояний создание заявки



Рисунок 2.4.5 – Диаграмма состояний редактирование заявки



Рисунок 2.4.6 – Диаграмма состояний удаление заявки  


Рисунок 2.4.7 – Диаграмма состояний создание ТЗ



Рисунок 2.4.8 – Диаграмма состояний редактирование ТЗ



Рисунок 2.4.9 – Диаграмма состояний удаление ТЗ



Рисунок 2.4.10 – Диаграмма состояний личный кабинет пользователя



Рисунок 2.4.11 – Диаграмма состояний редактирование статуса заявки

2.5 Схема данных

База данных будет состоять из нескольких таблиц. Взаимосвязь таблиц представлена на рисунке.

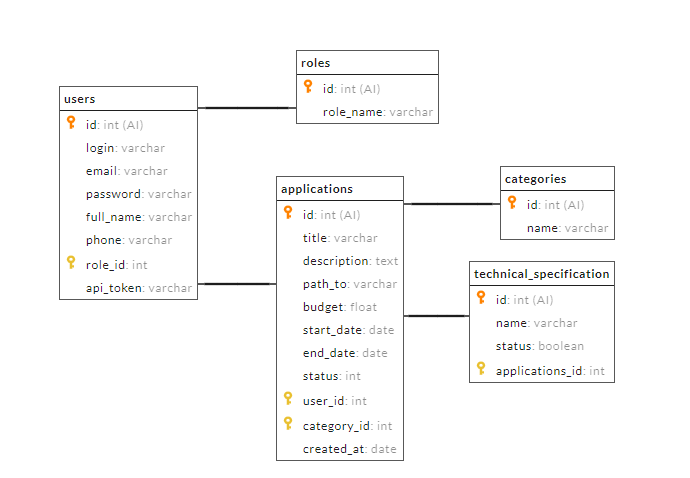


Рисунок 2.5.1 – Схема связи в базе данных

Таблица 2 – «users» пользователи

|  |  |  |  |
| --- | --- | --- | --- |
| Название поля | Тип данных | Описание | Ключ |
| id | int(10) | Код пользователя | Первичный ключ |
| login | varchar(50) | Логин |  |
| password | varchar(50) | Пароль |  |
| email | varchar(100) | Почта |  |
| full\_name | varchar(50) | Имя |  |
| phone | varchar(50) | Телефон |  |
| role\_id | int(11) | Номер роли |  |
| api\_token | varchar(50) | Токен |  |

Таблица 3 – «technical\_specification» техническое задание

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название поля | Тип данных | Описание | Ключ | Связь |
| id | int(10) | Код технического задания | Первичный |  |
| name | varchar(50) | Название |  |  |
| status | boolean | Статус |  |  |
| applications\_id | int | Код пользователя | Вторичный | applications.id |

Таблица 4 – «applications» заявки

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название поля | Тип данных | Описание | Ключ | Связь |
| id | int(10) | Код заявки | Первичный |  |
| title | varchar(50) | Название заявки |  |  |
| description | varchar(255) | Содержимое заявки |  |  |
| user\_id | Int(10) | Код пользователя | Вторичный | user.id |
| path\_to | varchar(255) | Путь до тз |  |  |
| urgensy | int | Срочность |  |  |
| status | boolean | Статус |  |  |
| start\_date | date | Дата начала |  |  |
| end\_date | date | Дата |  |  |
| category\_id | int(11) | Код категории |  | category.id |
| created\_at | date | Дата создания |  |  |

Таблица 5 – «roles» роли

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название поля | Тип данных | Описание | Ключ | Связь |
| id | int(11) | Код роли |  |  |
| role\_name | varchar(20) | Название роли |  |  |

Таблица 6 – «categories» категории

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название поля | Тип данных | Описание | Ключ | Связь |
| id | int(11) | Код категории |  |  |
| name | varchar(20) | Название категории |  |  |

2.6 Пользовательские сценарии

Вариант использования функции «Личный кабинет». Суть использования данной функции будет заключаться в том, что пользователь может получить полный список своих данных.

Вариант использования функции «Создать техническое задание». Суть данной функции в том, что заказчик может создать техническое задание.

Вариант использования функции «Редактировать техническое задание». Суть данной функции в том, что заказчик может выбрать из списка заявок нужное техническое задание для редактирования и изменить его.

Вариант использования функции «Удалить техническое задание». Суть данной функции в том, что заказчик может выбрать из списка заданий нужное задание для удаления и удалить его.

Вариант использования функции «Создать заявку». Суть данной функции в том, что заказчик может создать заявку.

Вариант использования функции «Редактирование заявки». Суть данной функции в том, что заказчик может выбрать из списка заявок нужную заявку для редактирования и изменить её.

Вариант использования функции «Удаление заявки». Суть данной функции в том, что заказчик может выбрать из списка заявок нужную заявку для удаления и удалить её.

Вариант использования функции «Добавление статуса заявки». Суть данной функции в том, что заказчик может выбрать из списка заявок нужную заявку и добавить ей статус.

Вариант использования функции «Добавление срочности заявки». Суть данной функции в том, что заказчик может выбрать из списка заявок нужную заявку и добавить ей срочность.

Вариант использования функции «Изменение статуса заявки». Суть данной функции в том, что заказчик может выбрать из списка заявок нужную заявку и изменить её статус.

Вариант использования функции «Вывод всех заявок». Суть использования данной функции будет заключаться в том, что заказчик может получить полный список всех заявок.

Вариант использования функции «Вывод всех технических заданий». Суть использования данной функции будет заключаться в том, что заказчик может получить полный список всех заданий.

2.7 Прототипы основных интерфейсов

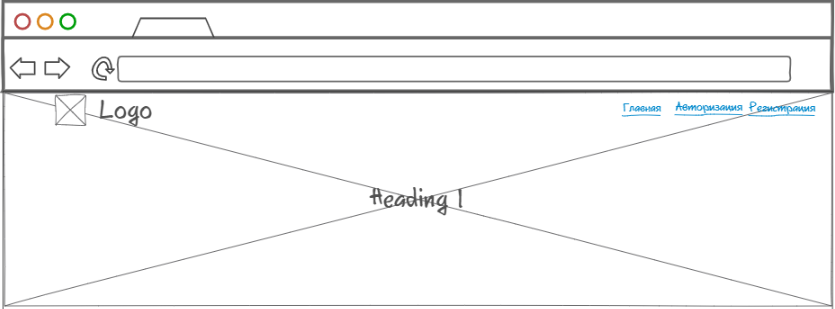


Рисунок 2.7.1 – Прототип главной страницы, пользователь не авторизован

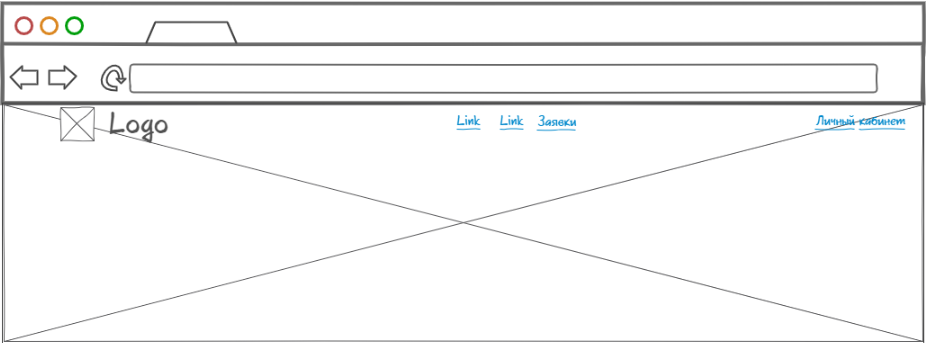


Рисунок 2.7.2 – Прототип главной страницы, пользователь авторизован

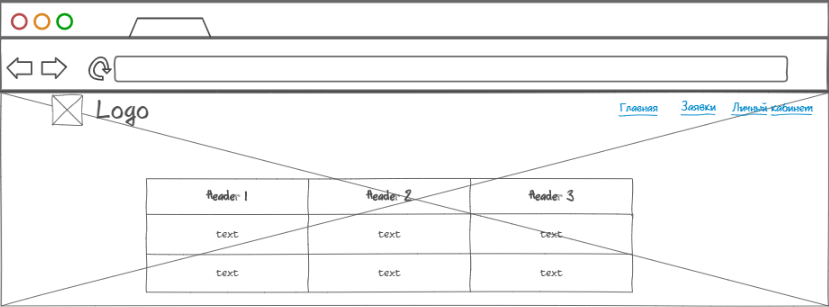


Рисунок 2.7.3 – Прототип страницы заявки

3 Экономическая часть

Экономическая часть диплома — это такая часть, в которой изложено технико-экономическое обоснование проекта ( ТЭО )

Задачи экономической части диплома

* проанализировать главные критерии и показатели, которые определяют себестоимость изделия или плановой калькуляции;
* произвести расчет показателей плановой калькуляции или сметы затрат в рамках выбранного алгоритма действий.

ТЭО - документ, в котором представлена информация, из которой выводится целесообразность (или нецелесообразность) создания продукта или услуги.

Экономическая часть включает 2 раздела:

1) расчет затрат на разработку и эксплуатацию.

2) расчет экономического эффекта и определение срока окупаемости.

3.1 Расчет затрат на разработку и эксплуатацию

3.1.1 Расчет затрат на разработку

Основными компонентами затрат на разработку программ и решение задачи на ЭВМ являются затраты, связанные с оплатой труда специалистов на разработку программы, обслуживание и эксплуатацию ЭВМ в период отладки программы и решения задачи, то есть рассчитываются прямые и косвенные затраты.

При определении полной себестоимости программы учтены все материальные расходы, расходы по заработной плате, отчисления в социальные статьи и составлена калькуляция затрат в следующей последовательности:

- основная заработная плата персонала;

- дополнительная заработная плата персонала;

- отчисления в социальные статьи;

- стоимость работ на ЭВМ;

- расчет стоимости материалов;

- расчет косвенных затрат на разработку программы.

При расчете всех экономических показателей была составлена таблица 7, в которой указаны все этапы работы по разработке программы и решению задачи, исполнитель каждого этапа, трудоемкость и стоимость исполнения.

Стоимость каждого этапа определена, исходя из оклада исполнителей и времени выполнения этапа.

Количество рабочих часов в месяце равно 168 часов, то есть 21 рабочий день в месяце по 8 часов.

Стоимость часа работы программиста определяется по следующей формуле 1.

Ст.ч.п=Оклад.п / К.р.ч (1)

25000 / 168 = 148 руб./час

где Ст.ч.п – стоимость часа работы программиста, руб.;

Оклад.п – оклад программиста;

К.р.ч – количество рабочих часов в месяце, час.

Таблица 7 – Этапы разработки

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименование этапов работ | Исполнитель | Трудоемкость, час | Плата за час, руб./час | Стоимость исполнения, руб. |
| Постановка задачи | Программист | 2 | 148 | 296 |
| Изучение литературы | Программист  ЭВМ | 15 | 148 | 2220 |
| Технический проект | Программист  ЭВМ | 15 | 148 | 2220 |
| Эскизный проект | Программист  ЭВМ | 20 | 148 | 2960 |
| Кодирование | Программист  ЭВМ | 30 | 148 | 4440 |
| Отладка программы  Тестирование | Программист  ЭВМ | 20 | 148 | 2960 |
| Оптимизация программы | Программист  ЭВМ | 10 | 148 | 1480 |
| Оформление документации | Программист  ЭВМ | 5 | 148 | 740 |
| Итого | Программист  ЭВМ | 117 115 | 1184 | 17316 |

Начисляемая заработная плата для каждого исполнителя по формуле 2:

ЗП = Сти \* РК (2)  
22510,8 = 17316\* 1.3

Где ЗП – начисленная заработная плата, руб.;

Сти- стоимость исполнения из таблицы 1 для каждого исполнителя, руб;

РК - районный коэффициент (в Томске РК=1,3).

На основе рассчитанной заработной платы определяем значение отчислений на социальные нужды.

Дополнительная заработная плата персонала рассчитывается по формуле 3

ЗПдоп = ЗП× 0,10, (3)

где ЗПдоп  – дополнительная заработная плата персонала;

ЗП – основная заработная плата персонала, руб.

Дополнительная заработная плата программиста

ЗПдоп = 22510,8 × 0,1 = 2251,08 руб.

Отчисления на социальные нужды рассчитаем для каждого исполнителя по следующей формуле 4:

Осн=(ЗП+ЗПдоп)\*0,302 (4)

7478,1= (22510,8 + 2251,08) \* 0.302

где Осн - отчисления на социальные нужды (30,2%, в т.ч.:

22% - в пенсионный фонд РФ (ПФРФ);

2,9% - в федеральный фонд социального страхования (ФФСС);

5,1% - в федеральный фонд обязательного медицинского страхования (ФФОМС). ЗП - начисленная заработная плата, руб.

Далее определим стоимость работы ЭВМ по следующей формуле 5:

Срм=Смч×Тм (5)

Срм = 18,9 + 2063 = 2081,9 руб

Смч = Срм/ Тм

Смч = 2081,9 / 90 = 23,1 руб

где Срм - стоимость работы ЭВМ,

Смч - стоимость машинного часа в рублях (определяется исходя из стоимости потребленной электроэнергии (кВт/ч) и амортизации компьютера);

Тм - общее время работы ЭВМ (данные из таблицы 1).

Расходы денежных средств, связанный с энергопотреблением технических средств можно найти по формуле 6:

Рэ = Кдн \* Враб \* Мс \* Цэн (6)

Рэ = 90\*0,06\*3,5 = 18,9 руб

где Кдн - период написания программы, дн,

Враб - длительность рабочей смены, ч,

Мс - мощность, потребляемая техническими средствами, кВтч;

Цэн - стоимость электроэнергии по действующим тарифам, р/кВтч.

Годовые амортизационные отчисления рассчитывается по формуле 7:

A =W \* Na (7)

А = 28314\*(1/5) = 5662,8 руб.

где А - годовые амортизационные отчисления,

W - первоначальная стоимость оборудования (программного обеспечения),

Na - норма амортизации,

Na = l/T, Т - срок полезного использования.

Амортизационные отчисления за период создания программного продукта, продукта (8), руб.:

Aп = А\*Кдн / Крг (8)

Ап = 5662,8 \* 90/247 = 2063 руб

где Кдн - количество отработанных дней;

Крг - количество рабочих дней в году.

Следующий элемент сметы расходов - прочие (накладные) расходы. В прочие расходы входит оплата услуг связи и интернета, командировочные расходы, коммерческие (транспортировка, хранение) и др. Для расчета накладных расходов можно воспользоваться следующими методами учета расходов: традиционным, котловым и методом «директ-костинг».

Так как основные расходы на реализацию проекта связаны с оплатой труда, то прочие расходы будем рассчитывать традиционным методом по формуле 9.

Кр = ЗП \* Кнр (9)

Кр = 22510,8 × 0,05 = 1125,54 руб

где Кр - прочие расходы,

ЗП -заработная плата,

Кнр - коэффициент накладных расходов (5-10%).

Таблица 8 – Расчёт себестоимости проекта

|  |  |
| --- | --- |
| Наименование статей расходов | Стоимость работ, руб. |
| Заработная плата (основная, дополнительная) | 24761,88 |
| Отчисление на соц. Нужды | 7478,1 |
| Прочие расходы | 1125,54 |
| Стоимость работы ЭВМ | 2081,9 |
| Накладные расходы | 1400 |
| Итого | 36847,42 |

3.1.2 Расчет годовых затрат на эксплуатацию

Годовые затраты на эксплуатацию программы рассчитываются по формуле 10.

Ср.м.год = N \* Cр.м + EH \* C (10)  
10659,2= 230 \* 14,3031+ 0,2 \* 36847,42

где Ср.м.год - годовые затраты на эксплуатацию программы;

N - поток заявок (количество обращений к программе);

Ср.м - стоимость одного непосредственного решения (формула 11);

Ен - нормативный коэффициент сложности (0,2-0,8);

С - себестоимость разработки программы (итог таблицы 8).

Стоимость одного решения рассчитывается по формуле 11.

Ср.м = См.ч \* Тр + Стч(исп) \* Qр \* Кр \* (1+Кк.р) (11)

14,3031=23,1 \* 0,007+148\*0,07\*1,3\*1,05

где Ср.м - стоимость одного непосредственного решения;

Тр - время решения задачи на компьютере, в ч.;

Стчсп - стоимость часа работы специалиста;

Qp - трудоемкость исполнителя в часах,

Стч(исп) - стоимость часа работы исполнителя;

Кр - районный коэффициент;

Кк.р - коэффициент косвенных расходов.

3.2 Расчет экономического эффекта и определение срока окупаемости

Экономический эффект достигается при эксплуатации и характеризуется экономией живого и овеществленного труда в общественном производстве, выраженной в денежной форме (прибыль предприятия), а также снижением затрат.

Социальный эффект заключается в обеспечении комфортных условий жизни населения и развития экономики страны.

Для того чтобы определить экономическую эффективность проекта необходимо рассчитать затраты на эксплуатацию ранее употреблявшимся образом

Расчёт эффективности основных показателей проекта.

Стоимость выполнения работ ранее употреблявшимся способом рассчитывается по формуле 12.

Ср.сп = Стчсп\*Тсп\*Кр\*(1+Кк.р) (12)

144,3= 148\*1.5\*1.3\*0.5

где Ср.сп - затраты на выполнение работ ранее употреблявшимся способом;

Стчсп - стоимость часа работы специалиста;

Тсп - временные затраты специалиста на одно решение, в ч.;

Кр - районный коэффициент;

Кк.р - коэффициент косвенных расходов.

Годовая стоимость выполнения работ ранее употреблявшимся способом рассчитывается по формуле 13.

Ср.сп.год = N\*Cр.ст (13)

Ср.сп.год = 600 \* 144,3= 86580

Где Ср.сп.год годовые затраты на выполнение работ ранее употреблявшимся способом;

N -поток заявок.

Экономическая эффективность рассчитывается по формуле 14.

Эгод = Ср.сп.год - Ср.м.год (14)

Эгод = 86580 - 10659,2 = 75920,8

Где Ср.сп.год – Годовые затраты на выполнение работ ранее употреблявшимся способом;

Ср.м.год - годовые затраты на эксплуатацию программы.

Определение коэффициента экономической эффективности программы.

Коэффициента экономической эффективности показывает сколько на 1 руб. вложенных затрат в разработку и эксплуатацию, получаем экономии. Чем больше данное значение, тем эффективнее проект.

Данный коэффициент рассчитывается по формуле 15.

Ер = Эгод/(С+Ср.м.год) (15)

Ер = 75920,8/(36847,42+10659,2) = 1,59

Определение срока окупаемости.

Срок окупаемости показывает через какой период времени возместятся затраты на разработку и эксплуатацию.

Срок окупаемости возможно рассчитать одной из формул 16 или 17.

Ток = (С + Ср.м.год) / Эгод (16)

Ток = 1/Ер (17)

Ток = (36847,42+10659,2)/ 75920,8= 0,6 года

Также необходимо составить график выполнения работ и определить наиболее затратные этапы.

Таблица 9 – График выполнения этапов

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование этапов | 1мес. | 2 мес. | 3 мес. |
| Постановка задачи |  |  |  |
| Технический проект |  |  |  |
| Эскизный  проект |  |  |  |
| Разработка алгоритма |  |  |  |
| Отладка/ тестирование |  |  |  |
| Оптимизация |  |  |  |
| Оформление сопроводительной документации |  |  |  |

Заключение

В заключении хотелось бы упомянуть важность затронутой проблемы.

И был разработан модуль «Заказчик» для цифровой платформы Smart Alliance Компании ООО «Умный мир».

И в нем были разработаны все планируемые функции.

Список использованных источников

1. The PHP Group. «PHP is a popular general-purpose scripting language that is especially suited to web development. Fast, flexible and pragmatic, PHP powers everything from your blog to the most popular websites in the world».

URL: «https://www.php.net/».

Статья. [Дата обращения апрель 2020г.]

1. Станислав Ильичев. «MVC: что это такое и какое отношение имеет к пользовательскому интерфейсу».

URL: «https://tproger.ru/articles/mvc/»

Статья. [Дата обращения апрель 2020г.]

1. Krishna Srinivasan . «REST API Best Practices».

URL: «https://javabeat.net/rest-api-best-practices/

Статья. [Дата обращения апрель 2020г.]

1. Официальная документация Фреймворка Laravel

URL : https://laravel.com/docs/7.x/mail

Статья. [Дата обращения апрель 2020г.]

1. Русскоязычное сообщество Laravel Framework

Документация по laravel на русском

URL: https://laravel.su/docs/5.4/eloquent

Статья. [Дата обращения апрель 2020г.]