

МИНИСТЕРСТВО ПО РАЗВИТИЮ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ
И КОММУНИКАЦИЙ РЕСПУБЛИКИ УЗБЕКИСТАН

САМАРКАНДСКИЙ ФИЛИАЛ
ТАШКЕНТСКОГО УНИВЕРСИТЕТА ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ ИМЕНИ АЛ-ХОРАЗМИЙ

ФАКУЛЬТЕТ «КОМПЬЮТЕРНЫЙ ИНЖИНИРИНГ»
КАФЕДРА «ПРОГРАММНЫЙ ИНЖИНИРИНГ»

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

ТЕМА: РАЗРАБОТКА ЭЛЕКТРОННО-ИНФОРМАЦИОННОЙ СИСТЕМЫ
ФИРМЫ ПО ОКАЗАНИЮ ПЕРЕВОДЧЕСКИХ УСЛУГ.

Рассмотрена на заседании кафедры
протокол № ____ от “ ____ ” _____ 2018г.

и допущена к защите.

Исполнитель:
студент группы 411

_____ Коржовов С.

Заведующий кафедрой
_____ доц. Каршиев А.Б.
“ ____ ” _____ 2018

Научный руководитель:
_____ Шамсутдинова Н.

САМАРКАНД 2018

Оглавление

ВВЕДЕНИЕ	3
ГЛАВА 1. ТЕОРЕТИЧЕСКИЕ ПРЕДПОСЫЛКИ И АНАЛИЗ СОЗДАНИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ ДЕЯТЕЛЬНОСТИ ФИРМ ПО ОКАЗАНИЮ ПЕРЕВОДЧЕСКИХ УСЛУГ	5
1.1. Предпосылки создания системы	5
1.1.1. Анализ существующих систем управление проектами переводов	6
1.2. Анализ методик разработки web-приложений	12
1.2.1. Обзор инструментальных средств разработки Web-сайтов	14
1.2.2. Анализ программно-аппаратной платформы реализации Web- приложения	15
1.3. Требования к пользовательскому интерфейсу	18
1.4. Обзор веб фреймворков	20
ГЛАВА 2. ТЕХНОЛОГИЯ ПРОЕКТИРОВАНИЯ И СОЗДАНИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ ДЕЯТЕЛЬНОСТИ ФИРМ ПО ОКАЗАНИЮ ПЕРЕВОДЧЕСКИХ УСЛУГ	24
2.1. Требования к информационной системе фирмы переводов	24
2.2. Проектирование базы данных	24
2.2.1 Обзор метода проектировки базы данных MySQL	25
2.2. Разработка базы данных	26
2.3. Требования к web-приложению	30
2.3.1. Проектирование интерфейса приложения	30
2.3.2. Парадигма MVC	31
ГЛАВА 3. ТЕХНОЛОГИЯ СОЗДАНИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ ДЕЯТЕЛЬНОСТИ ФИРМ ПО ОКАЗАНИЮ ПЕРЕВОДЧЕСКИХ УСЛУГ	34
3.1. Разработка web-приложения	35
3.2. Описание программы	35
3.3. Технические решения безопасности труда	41
ЗАКЛЮЧЕНИЕ.....	46
СПИСОК ЛИТЕРАТУРЫ.....	47
ПРИЛОЖЕНИЕ	48

ВВЕДЕНИЕ

В Узбекистане государство обладая наибольшим административным ресурсом, играет должную роль флагмана развития ИКТ. Но мировой опыт свидетельствует, что ИКТ – это прежде всего прерогатива бизнеса, и не только в вопросах использования технологий, а именно по части их производства и продажи. В этой связи в ближайшей перспективе перенимание у государства вектора инициирования и разработки конкурентоспособных коммуникационных технологий бизнесом с целью их использования, в т. ч. в коммерческих целях.

19 февраля был принят Указ Президента Республики Узбекистан «О мерах по дальнейшему совершенствованию сферы информационных технологий и коммуникаций». Потребовано обеспечение стимулирования роста цифровой экономики, в том числе за счет внедрения комплексных автоматизированных систем управления на производстве, создания благоприятных условий для развития электронной коммерции.

С 1 июня 2018 года субъекты предпринимательства будут вправе осуществлять экспорт товаров, работ (услуг), за исключением отдельных товаров по перечню, определяемому Кабинетом Министров Республики Узбекистан, посредством электронных торговых площадок в сети Интернет без заключения экспортного контракта и изъятия банковской комиссии за экспорт товара, работ (услуг). При этом оборот товаров, реализуемых предпринимателями путем использования электронной коммерции, считается розничной торговлей вне зависимости от объема. [1]

Актуальность темы

В настоящее время очень развит механизм дистанционной работы. Переводческая компания нуждается в четкой процедуре управления проектами: контроль поступающих заказов, наблюдение за ходом работ, управление хранением исходных документов, переведенных файлов и баз данных компании.

Исходя из этого можно сказать что разработка этой системы управления переводческими проектами будет актуальной на рынке.

Цели работы: Целью данной работы является разработка web-приложение, назначением которого будет ведение тщательного контроля поступающих заказов, наблюдением за ходом работ, выполняемых штатными переводчиками или субподрядчиками, управление хранением исходных документов, переведенных файлов и баз данных компании или единичных заказчиков, устранение технических ошибок, поддержка результативной работы переводчиков и их обучение.

Для достижения цели требуется решить **следующие задачи:** изучить архитектуру, инструменты и особенности разработки web-приложений; разработать базу данных в СУБД MySQL; разработать web-приложение, работающее с данной базой данных; разработать web-интерфейс пользователя; провести тестирование разработанного приложения.

Структура и объем работы

Работа состоит из введения, двух глав, заключения и библиографии. Объем работы составляет 47 страниц, объем библиографии – 13 источников.

Первая глава, «теоретические предпосылки и анализ web разработок», содержит основную информацию об переводческих системах и разработке web приложениях.

Вторая и третья глава содержит «Разработка кроссплатформенного web-приложения», содержит подробную информацию об этапах разработки информационной системы, включая описание инструментов разработки приложения, разработку базы данных, реализацию интерфейса веб-сайта, а также информацию о результатах тестирования. В заключении приводятся основные результаты работы. Приложение содержит тестовые данные и интерфейс системы.

ГЛАВА 1. ТЕОРЕТИЧЕСКИЕ ПРЕДПОСЫЛКИ И АНАЛИЗ СОЗДАНИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ ДЕЯТЕЛЬНОСТИ ФИРМ ПО ОКАЗАНИЮ ПЕРЕВОДЧЕСКИХ УСЛУГ

1.1. Предпосылки создания системы

Любая переводческая компания, состоит ли она из одного человека или имеет большой штат постоянных сотрудников, нуждается в четкой и ясной методике управления, позволяющей определить точный ход работ для каждого этапа проекта – от момента получения исходного документа до момента отправки переведенного файла клиенту.

Переводческая компания нуждается в четкой процедуре управления проектами: требуется тщательный контроль поступающих заказов, наблюдение за ходом работ, выполняемых штатными переводчиками или субподрядчиками, управление хранением исходных документов, переведенных файлов и баз данных компании, устранение технических ошибок, поддержка результативной работы переводчиков и их обучение. Если все эти задачи будут решены и тщательно проконтролированы, качество перевода повысится, а его стоимость, соответственно, уменьшится.

Контроль хода работ одинаково важен как для фрилансера, так и для переводческой компании, когда они получают работу более чем от одного клиента, и, особенно, если им нужен субподряд. Например, когда компания получает от пяти разных клиентов по 300 страниц в неделю, и распределяет их среди 15 переводчиков-субподрядчиков, контроль хода работ становится жизненно важным. Каждая работа должна быть внесена в сводную ведомость управления проектами. Если компания не станет делать это регулярно, ей придется столкнуться с серьезными проблемами в конце месяца, когда потребуется выставить счет клиентам и заплатить переводчикам. При выставлении счета клиентам, отслеживании состояния работы и оплате работы переводчиков, необходимо использовать сводную ведомость управления проектами как основу.

После получения работы, в систему управления проектами необходимо внести следующие сведения: название проекта или имя клиента, имя файла,

дату получения, количество слов или страниц (если это необходимо), приблизительное количество переводческих/редакторских человеко-часов, и срок сдачи проекта. Упорядоченное внесение этих данных поможет правильно распланировать ход работ.

На более позднем этапе, когда работа проанализирована и распределена между переводчиками, в систему управления проектами необходимо внести следующие дополнительные данные: имена переводчиков, количество слов или страниц, выданных каждому переводчику, приблизительное количество человеко-часов для каждого переводчика, и дату и время, когда переведенный материал должен быть передан в компанию.

Когда компания получает переведенный материал, информация о работе, выполненной переводчиком, должна быть внесена в сводную ведомость для дальнейшего осуществления оплаты.

1.1.1. Анализ существующих систем управление проектами переводов

Деятельность фирм переводов, как и любой другой компании, состоит из массы рутинных процессов, без надлежащей организации которых невозможна согласованная работа сотрудников и развитие его бизнеса. Нужно вести базу клиентов и исполнителей, координировать работу переводчиков, следить за рентабельностью проектов и правильностью начисления зарплат, а главное — видеть картину в целом и планировать будущее компании.

Проведенное исследование с целью выяснить, какие программы и системы управления используются в различных бюро переводов.

На момент написания работы ответы на вопросы предоставили 110 переводческих компаний из разных стран мира. Это не так уж много. Но она позволяет понять общую картину.

Термин TMS (Translation Management System — система управления переводами) многих вводит в заблуждение, т. к. под ним подразумевают совершенно разные системы. Одни думают, что TMS-системы служат для

распределения работ между разными группами переводчиков, другие — для интеграции на сайты и перевода их содержимого, третьи — для ведения заказов и учета финансов. [6]

В исследовании рассматривались так называемые TBMS (Translation Business Management System — система управления переводческим бизнесом). TMS-системы этого «подвида» не обладают функционалом CAT-инструментов и не предоставляют средств автоматизированного перевода: они предназначены для управления процессом перевода, а не для его выполнения. Часто они интегрированы с другими системами.

При вопросе о востребованности систем управления переводами Две трети компаний на вопрос о том, используют ли они какую-либо систему ответили утвердительно:

Пользуетесь ли вы какой-либо системой управления переводческим бизнесом?

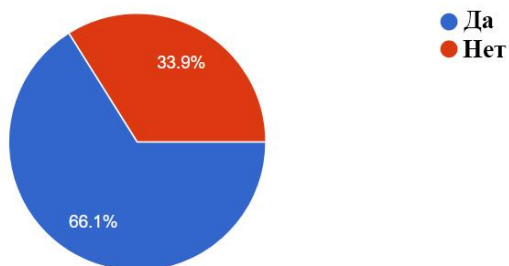


Рис.1.1. Диаграмма ответов на вопрос “Пользуетесь ли вы какой-либо системой управление переводческим бизнесом”

Однако после анализа ответов выяснилось, что многие респонденты просто не поняли вопроса из-за размытости термина TMS и имели в виду такие программы, как *Trados*, *SmartCAT*, *MemSource*, *MemoQ*, *Grammarly*, и даже «перевод вручную». Если исключить такие ответы, то картина получается следующей:

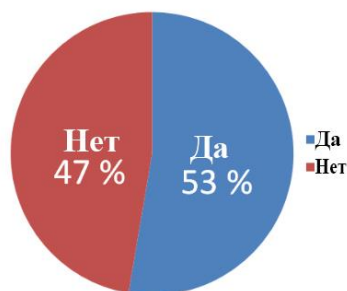


Рис.1.2. Диаграмма ответов на вопрос “Пользуетесь ли вы какой-либо системой управление переводческим бизнесом”

То есть какими-либо системами управления бизнесом пользуется только половина бюро, принявших участие в опросе. Полагается, что среди бюро, не принявших участие в исследовании, процент еще ниже.

Чем крупнее компания, тем больше вероятность того, что она использует TBMS-систему (это закономерно):

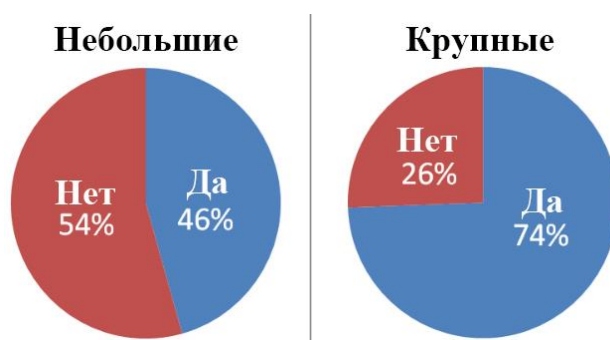


Рис.1.3. Диаграмма ответов на вопрос “Пользуетесь ли вы какой-либо системой управление переводческим бизнесом” между крупными и небольшими компаниями

Таким образом, если небольшие компании (с командой до пяти менеджеров) больше чем в половине случаев обходятся без специального инструмента для координации своей деятельности, то в более крупных этот процент резко падает (до 26 %). Это объяснимо: чем крупнее компания, тем сложнее управлять ею «врукопашную», удерживая все нюансы в голове или в *Excel*. Кроме того, компаниям, не пользующимся современными системами управления бизнесом, сложнее расти и развиваться: по мере развития рутина

отнимает все больше времени у руководителей, и они «не успевают» заниматься развитием бизнеса, из-за чего темпы роста часто замедляются.

Почему не используют? Рейтинг причин выглядит следующим образом:

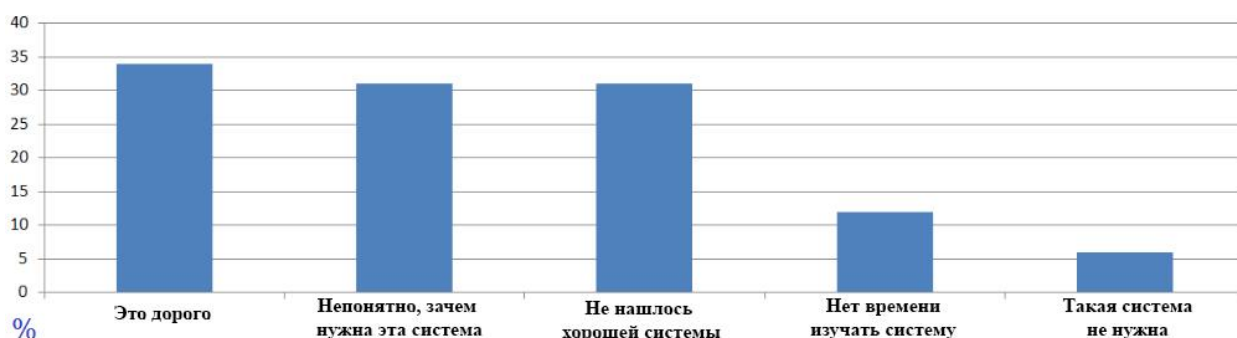


Рис.1.4 Диаграмма ответов на вопрос “Почему не используют?”

Три главные причины — это цена, непонимание функционала и сложности в поиске. Каждая третья компания озвучила как минимум одну из этих проблем.

Значительно меньше компаний либо не находят времени на изучения возможностей таких систем (12 %), либо считают, что такая система не нужна им в принципе (6 %). Характерно, что все эти компании небольшого размера. То есть они либо еще «не доросли» до тех размеров, когда становится сложно управлять разросшейся командой, либо, наоборот, уже выросли настолько, что у их руководителей не осталось времени на улучшения процессов в бюро: все их время уходит на рутину.

Половина компаний, не пользующихся системами управления переводческим бизнесом, осознают их преимущества и рассматривают возможность их внедрения в будущем.

Что-же используют в компаниях, утвердительно ответивших на вопрос об использовании системы управления бизнесом, предпочтения распределились следующим образом:

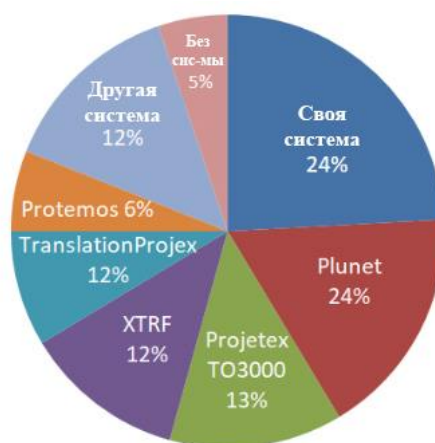


Рис.1.5 Диаграмма ответов на вопрос “Что-же используют?”

Почти каждое четвертое бюро переводов пользуется системой собственной разработки. Основная причина этого в том, что руководители таких компаний в свое время просто не нашли на рынке систему, полностью удовлетворяющую их потребности, и поэтому вынуждены были создать собственную.

Главное преимущество такого подхода — возможность полностью адаптировать систему под свои нужды. А главный недостаток — стоимость: оплата труда программистов порой превышает стоимость готового решения в разы, причем чаще всего реальные расходы оказываются выше ожидаемых. Кроме того, уже разработанная и введенная в эксплуатацию система требует постоянной поддержки и обновления, иначе по прошествии нескольких лет она морально устаревает, — на это тоже уходят время и деньги.

Подобные решения доступны только компаниям, доходы которых позволяют держать в команде программистов, постоянно работающих над системой. Небольшие компании способны разработать «с нуля» разве что простые программы с базовым функционалом. Поэтому 3/4 компаний предпочитают более экономный вариант — покупку готовой системы управления проектами.

Наибольшее количество компаний пользуются такими системами, как *Plunet*, *Projex* и *XTRF*. Эти три решения существуют более 10 лет и

успели собрать внушительную базу пользователей. Относительно недавно появились системы *TranslationProjex* и *Protomos*, которые набирают популярность. Рынок динамичен, и через несколько лет соотношение долей систем на рынке может кардинально измениться.

12 % респондентов используют другие системы управления бюро переводов, но доля каждой из них незначительна (по крайней мере, среди принявших участие в этом опросе). Еще небольшая группа (около 5 %) приспособили для себя программы, изначально не предназначенные для переводческого бизнеса.

На вопрос удовлетворенности этими системами получили результат что из-за того что в каждой компании формируется своя уникальная корпоративная культура, методика выполнения рабочих процессов и политика в отношении доступа сотрудников к информации, удовлетворяющего абсолютно всех, существовать не может: обязательно кого-нибудь что-нибудь не устраивает.

Опрос респондентов оценить степень удовлетворенностью текущей системой по шкале от 1 (очень плохо) до 5 (отлично) составил среднюю оценку 3,9 балла.

Примечательно, что удовлетворенность системой собственной разработки составила в среднем 4,1 балла, а приобретенными — 3,8 балла. Возможно, впрочем, на оценку влияет психологическая привязанность к собственной разработке.

Удовлетворенность различными системами отличается, но у каждой есть как высокие, так и низкие оценки. Кроме того, примерно одинакова удовлетворенность старыми и новыми системами. Вывод прост: у разных бюро разные потребности, и определяющим фактором является соответствие системы конкретным требованиям.

Примечательно, что два основных замечания к системам управления переводческим бизнесом противоречат друг другу: пользователи хотят большей функциональности, но при этом считают, что в системах слишком много лишнего. Эта дилемма встает перед разработчиками постоянно: как

сделать систему функциональной, не перегрузив ее ненужными функциями.

Чаще всего разработчик выбирает какой-либо один из путей и занимает нишу, подходящую конкретной группе переводческих компаний.

Стоимость как недостаток практически не упоминалась — то есть функционал имеет большее значение, чем цена.

Сделав выбор в пользу какой-либо из систем, бюро с крайней неохотой переходит на другую, даже более современную и удобную. Во-первых, компания пытается окупить сделанные ранее инвестиции и не готова тратить дополнительные средства на конкурирующее решение. Во-вторых, смена системы чаще всего технически сложна: возникают сложности с переносом данных и приходится переобучать сотрудников. Поэтому даже если удовлетворенность текущей системой недостаточна, переход на другую труден психологически. Об этом свидетельствуют «тройки», поставленные некоторыми респондентами.

В итоге можно сказать что использование системы управления переводческими проектами и бизнесом позволяет компаниям увеличить свою эффективность и ускорить рост. Но так или иначе перед руководителем любого бюро встает вопрос выбора:

- 1) Разрабатывать систему самостоятельно или приобрести готовое решение?
- 2) Если покупать систему, то какую? Здесь важны как функционал, так и стоимость.

1.2. Анализ методик разработки web-приложений

На сегодняшний день существует огромное количество различных методов создания сайтов, различающихся в зависимости от назначения и типа сайта, умений разработчика или финансовых возможностей заказчика. При этом классифицировать методы можно по разным признакам: используемому программному обеспечению, архитектуре сервера, технологическим признакам либо даже маркетинговым. Однако перечисленные особенности касаются, прежде всего, разработчика, провайдера либо регистратора.

Разнообразных же методов разработки, отличающихся друг от друга, десятки. Несмотря на множество технологических различий, принципиально отличающихся подходов к разработке можно выделить лишь несколько. Если группировать методы создания сайтов, исходя из того, что признак цена-время является самым востребованным, то можно выделить три основных метода.

1 Разработка с нуля.

Данный метод наиболее длительный и трудный. При этом он требует знаний хотя бы одного языка веб-программирования (PHP, HTML, CSS), а также опыт работы с базами данных (MySQL). Можно также воспользоваться наиболее популярными программами для создания сайтов вручную, к которым относятся Dreamweaver, FrontPage, WebSite X5, а также Photoshop (для работы с изображениями).

Однако всю рутинную работу разработчику придется выполнить самому. Результаты при этом могут получиться более чем скромными, характерными для новичков и любителей в сфере SEO. Главным минусом данного метода является необходимость осуществлять все самостоятельно, затрачивая значительное количество времени и сил.

2 Использование конструктора.

Самый простой метод разработки, поскольку не требует знаний профессионального веб-программиста. Однако на качественную работу также рассчитывать не стоит. Существует множество инструментов для разработки по тем либо иным шаблонам и макетам. К самым популярным и хорошо изученным конструкторам относятся «narod.ru», «setup.ru», «umi.ru», «ucoz.ru» и подобные им.

Зачастую, воспользовавшись конструктором на начальном этапе, разработчик может впоследствии пожалеть о потерянном на изучение работы конструктора времени, поскольку данный метод таит в себе пару серьезных недостатков. Один заключается в ограниченности возможностей такого рода конструктора, невозможности реализовать более, нежели предусмотрено его

функциями, а другой - в наличии специальных фрагментов кода и сложности раскрутки созданного ресурса.

3 Использование системы управления сайтом.

Самый быстрый, а также широко используемый метод. На сегодняшний день широко используются такие системы («движки»), как WordPress, Joomla, Django, Yii и прочие, которые идеально подойдут для сайтов либо блогов. Для форумов более подходящими станут: phpbb, ExBB, PunBB, SMF.

Использование движков (CMS) предоставляет доступ к множеству современных шаблонов и схем построения, а также значительно ускоряет и упрощает работу разработчика.

1.2.1. Обзор инструментальных средств разработки Web-сайтов

Для верстки веб-страниц придумали немало различных инструментов. Все редакторы делятся на текстовые и визуальные. Визуальные редакторы выгодно отличаются тем, что не требуют от пользователя знаний html и css.

Мы можем создавать сайт как на листе бумаги - что нарисовали, то и получили, но есть и свои минусы - все визуальные редакторы имеют огрехи и ограниченные возможности.

Поэтому настоящие профессионалы предпочитают не отказываться от использования текстовых редакторов. В любом случае лучше иметь в своем распоряжении два редактора: визуальный и текстовой.

Рассмотрим подробнее существующие визуальные редакторы:

1 Macromedia Dreamweaver. Высокотехнологичный инструмент специально для создания web-страниц. По статистике это одна из лучших программ на сегодняшний день. Разработчики заявляют, что Macromedia Dreamweaver служит для проектирования сложных профессиональных web-сайтов. Dreamweaver - это не только визуальный редактор, это сложный и мощный инструмент, на освоение которого придется потратить время;

2 Adobe GoLive - прекрасное решение для тех, кто предпочитает работать с программами от Adobe. Знакомая среда позволяет легко и быстро

разобраться, что здесь к чему. GoLive всего лишь визуальный редактор, который поддерживает Html, CSS и XML. Зато GoLive имеет чудесный встроенный редактор кода;

3 Microsoft FrontPage По словам разработчиков FrontPage создана для проектирования сложных веб-узлов, а также позволяет управлять разработанным сайтом. FrontPage прекрасно дружит как с Html, Css, так и с Dhtml и Javascript. Программа дает широкие возможности для управления изображениями и flash-технологиями;

4 Hometown - это наиболее популярный и мощный текстовый редактор. В нем довольно просто и удобно работать с Html-кодом, а также с CSS-редактором;

5 HTML Pad это самый серьезный конкурент программы Hometown, так как наряду с ней пользуется любовью и признанием пользователей. Помимо стандартного набора функций HTML Pad дружит с JavaScript, VBScript, Perl, SSI и ASP. А помимо все прочего программа содержит много полезных материалов по CSS и Html;

6 Notepad или обычный Блокнот эта самая простая программа, с которой многие начинали свои первые шаги в веб-программировании. Блокнот есть у всех без исключения в стандартных программах.

1.2.2. Анализ программно-аппаратной платформы реализации Web-приложения

Сегодняшний рынок программного обеспечения предъявляет большие требования к создаваемым проектам. Так, для современных программных средств важными требованиями являются переносимость, мультиплатформенность и масштабируемость.

Под переносимостью подразумевается возможность использовать программное средство на разных программно-аппаратных платформах без существенной переработки кода.

Масштабируемость означает возможность добавления новых функций и свойств программного средства с минимальным изменением всего кода в целом. Идеальным является вариант, который позволяет наращивать мощность ПП без изменения основного кода, лишь добавляя новые модули.

Поэтому при разработке учитывались оба этих требования. Естественно, создать достаточно сложное ПО, которое работало бы на всех известных платформах, практически невозможно, но следует стремиться обеспечить его функциональность на самых распространенных платформах [2].

Исходя из того, что платформа IBM PC является наиболее распространенной в Узбекистане, было принято решение разрабатывать веб-сайт именно под эту аппаратную платформу. Проанализировав системное программное обеспечение IBM PC-совместимой компьютерной техники, были получены следующие результаты: 73 % - OS семейства Windows, 16 % - Linux, 11 % - Free BSD, Open BSD, SCO, Mac OS X, Novell NetWare. Исходя из этих результатов, а так же из соображения, что программное обеспечение должно функционировать на как можно большем количестве платформ, было принято решение разрабатывать ПС с таким расчетом, чтобы обеспечить функционирование, как его отдельных компонентов, так и всего комплекса в целом на двух основных программных платформах: Windows и Linux. Причем это требования распространяется на ту часть разрабатываемого программного обеспечения, которое должно быть размещено на веб-сервере.

Веб-сервер - это сервер, принимающий HTTP запросы от клиентов. Обычно в роли клиентов выступают веб-браузеры, и выдающий им HTTP ответы, вместе с которыми передаются HTML страницы, изображения, файлы, медиа-потoki или другие данные. Веб-серверы являются основой Всемирной паутины.

Фактически веб-сервером называют как программное обеспечение обеспечивающее все вышеперечисленные действия, так и компьютер, на котором это программное обеспечение работает. Клиенты получают доступ к

веб-серверу по средствам единого указателя ресурсов - URL адресу нужной им веб-страницы или другого ресурса располагаемого на сервере.

Apache HTTP-сервер (назван именем группы племён североамериканских индейцев апачей; кроме того, является сокращением от англ. а patchy server; среди русских пользователей общепринято искажённое апач) - свободный веб- сервер [2].

Apache является кроссплатформенным ПО, поддерживает операционные системы Linux, BSD, Mac OS, Microsoft Windows, Novell NetWare, BeOS.

Основными достоинствами Apache считаются надёжность и гибкость конфигурации. Он позволяет подключать внешние модули для предоставления данных, использовать СУБД для аутентификации пользователей, модифицировать сообщения об ошибках и т. д. Поддерживает IPv6.

Хампп

ХАМПП – это именно *локальный сервер*, предназначенный для установки на домашнем компьютере для имитации в нем среды интернета.

Каждая буква в наименовании - ХАМПП – имеет определенное значение:

Х – означает, что вебсервер ХАМПП способен работает с любой (х – произвольной) операционной системой. В настоящее время ХАМПП работает в четырех операционных программах: Windows и Linux поддерживаются полностью (для Linux существует специализированный вариант – LAMP, а для Windows есть облегченные версии Lite), для Mac OS X и Solaris - версий не столь много. [13]

Следующие буквы в слове ХАМПП имеют происхождение от первых букв в наименовании основных компонентов: Apache, MySQL, PHP, Perl.

Веб-сервер ХАМПП – это не одна программа, а сборка, в которую разработчики включили: Apache, MySQL, PHP+PEAR, Perl, mod_php, mod_perl, mod_ssl, OpenSSL, phpMyAdmin, Webalizer, FileZilla Server, Mercury

Transport System for Win32 and NetWare Systems v3.32, JpGraph, mcrypt, eAccelerator, SQLite, WEB-DAV+mod_auth_msql, Tomcat.

Apache – это *серверная программа*, которая как раз обеспечивает среду интернета – то есть позволяет серверным программам взаимодействовать с программой браузер, приходящей на сайт. Программа **сервер Apache** является ядром веб сервера **XAMPP**.

MySQL – *база данных*, которая состоит из таблиц, в которых описано месторасположение файла сайта и его характеристики, нужные для вывода на экран. Считается одной из самых быстрых баз.

PHP+PEAR и **Perl** – это *компиляторы* распространенных языков, на которых пишутся сайты. Эти компоненты позволяют браузеру правильно прочитать код, на котором написаны скрипты сайта.

Mercury Transport System for Win32 and NetWare Systems v3.32 – это почтовая программа, обеспечивающая пересылку почтовых отправок в почтовые ящики.

Остальные программы являются вспомогательными – облегчают работу пользователя, как например, **FileZilla Server** – это ftp-клиент, при помощи которого можно закачивать файлы на сервер.

1.3. Требования к пользовательскому интерфейсу

Современные специалисты, и тем более специалисты, специализирующиеся в области информационных технологий и связи, должны в достаточной мере владеть Интернет-технологиями, и в частности веб-технологиями, включая создание веб-сайтов средствами HTML и CSS. Овладение такими технологиями, помимо возможности размещения собственной информации в Интернете, поможет на более высоком уровне использовать бесчисленные информационные ресурсы, расположенные во Всемирной паутине.

Приведем ряд общих требований, предъявляемых к веб-сайтам:

1. легкость освоения;

2. запоминаемость;
3. эффективность использования;
4. надежность использования;
5. удовлетворение пользователя.

При создании веб-сайтов необходимо руководствоваться следующим:

1. не существует типичных людей, всегда пытайтесь представить в качестве посетителя вашего сайта реально существующего человека;
2. избегайте использования текста, графики и фона одного оттенка;
3. поддерживайте высокий контраст. Избегайте использования текста, фона и графики одинаковой яркости;
4. избегайте фонов, перегруженных графическими элементами;
5. не заставляйте пользователей запоминать информацию, так как узнавание намного проще, чем воспроизведение из памяти;
6. ссылки, уже посещенные пользователем, должны отличаться цветом и/или стилем от тех, по которым еще не был осуществлен переход;
7. так как запоминать зрительные образы намного проще, делайте страницу так, чтобы она оставалась в памяти у пользователя и отличалась от остальных;
8. ограничивайте размер однородных групп (например ссылок), из которых необходимо сделать выбор 5 - 9 элементами. Это связано с «магическим» числом миллера - $7 / 2$, указывающего, сколько объектов одновременно человек может удерживать во внимании;
9. надо помнить, что время, которое пользователь согласен потратить на ожидание, пропорционально ожидаемому результату;
10. если время загрузки страницы составляет более 30 с., необходимо обеспечить пользователю уверенность в том, что процесс действительно происходит. Это можно реализовать, например, при помощи индикатора выполнения процесса;
11. затраченное время значит для пользователя намного больше, чем закаченные байты;

12.сводите к минимуму расстояние, преодолеваемое указателем мыши между двумя последовательными вариантами выбора;

13.постарайтесь свести к минимуму расстояние между наиболее часто посещаемыми областями страницы (особенно это касается главной страницы) и кнопкой «назад» браузера.

Ниже приведены некоторые наиболее распространенные традиции в веб- дизайне:

- логотип сайта в верхнем левом углу одновременно является ссылкой на главную страницу;

- ссылки часто продублированы в нижней части страницы. На длинных страницах применяются ссылки, возвращающие пользователя в верхнюю часть страницы;

- то, на чем возможен щелчок кнопкой мыши, имеет специальное выделение. Например, текст ссылок обычно имеет подчеркивание или какое-либо иное выделение. Кроме того, обычно применяется динамическое выделение, когда объект изменяет свой внешний вид в момент прохода над ним указателя мыши.

1.4. Обзор веб фреймворков

Фреймворк (англ. framework - каркас, структура) - структура программной системы; программное обеспечение, облегчающее разработку и объединение разных компонентов большого программного проекта. Употребляется также слово «каркас», а некоторые авторы используют его в качестве основного, в том числе, не базируясь вообще на англоязычном аналоге [2]. Можно также говорить о каркасном подходе как о подходе к построению программ, где любая конфигурация программы строится из двух частей: первая, постоянная часть - каркас, не меняющийся от конфигурации к конфигурации и несущий в себе гнезда, в которых размещается вторая, переменная часть - сменные модули (или точки расширения).

Yii - это высокоэффективный основанный на компонентной структуре PHP-фреймворк для разработки масштабных веб-приложений. Он позволяет

максимально применить концепцию повторного использования кода и может существенно ускорить процесс веб-разработки. Название Yii (произносится как Yee или [ji:]) означает простой (easy), эффективный (efficient) и расширяемый (extensible).

Подобно большинству других PHP-фреймворков, Yii - это MVC-фреймворк. Используемая модель MVC (model-view-controller) дает большие возможности в построении модульного приложения. Также модульные приложения дают возможность работать над приложением целой группы людей, где каждый отвечает за свой кусок или модуль приложения. Еще одним преимуществом Yii является статичность его кода, что позволяет легко переносить большие нагрузки.

Превосходство Yii над другими фреймворками заключается в эффективности, широких возможностях и качественной документации. Yii изначально спроектирован очень тщательно для соответствия всем требованиям при разработке серьёзных веб-приложений. Yii не является ни побочным продуктом какого-либо проекта, ни сборкой сторонних решений. Он является результатом большого опыта авторов в разработке веб-приложений, а также их исследований наиболее популярных веб-фреймворков и приложений [10].

ASP.NET MVC Framework - фреймворк для создания веб-приложений, который реализует шаблон Model-view-controller. Данный фреймворк добавлен Microsoft в ASP.NET.

В апреле 2009 года исходный код ASP.NET MVC был опубликован под лицензией Microsoft Public License (MS-PL). 27 марта 2012 года лицензия была изменена на Apache License 2.0 [6].

Шаблон архитектуры Model-View-Controller (MVC) разделяет приложение на три основных компонента: модель, представление и контроллер.

Платформа ASP.NET MVC представляет собой альтернативу схеме веб-форм ASP.NET при создании веб-приложений. Платформа ASP.NET MVC является легковесной платформой отображения с широкими возможностями

тестирования и, подобно приложениям на основе веб-форм, интегрирована с существующими функциями ASP.NET, например с главными страницами и проверкой подлинности на основе членства. Платформа MVC определяется в сборке System.Web.Mvc.

Платформа ASP.NET MVC предоставляет следующие возможности:

- Разделение задач приложения (логика ввода, бизнес-логика и логика пользовательского интерфейса), широкое возможности тестирования и разработки на основе тестирования. Все основные контракты платформы MVC основаны на интерфейсе и подлежат тестированию с помощью макетов объекта, которые имитируют поведение реальных объектов приложения. Приложение можно подвергать модульному тестированию без запуска контроллеров в процессе ASP.NET, что ускоряет тестирование и делает его более гибким. Для тестирования возможно использование любой платформы модульного тестирования, совместимой с .NET Framework.

- Расширяемая и дополняемая платформа. Компоненты платформы ASP.NET MVC можно легко заменить или настроить. Разработчик может подключать собственный механизм представлений, политику маршрутизации URL-адресов, сериализацию параметров методов действий и другие компоненты. Платформа ASP.NET MVC также поддерживает использование моделей контейнера внедрения зависимости (DI) и инверсии элемента управления (IOC). Модель внедрения зависимости позволяет внедрять объекты в класс, а не ожидать создания объекта самим классом. Модель инверсии элемента управления указывает на то, что если один объект требует другой объект, то первые объекты должны получить второй объект из внешнего источника (например, из файла конфигурации). Это облегчает тестирование.

- Расширенная поддержка маршрутизации ASP.NET. Этот мощный компонент сопоставления URL-адресов позволяет создавать приложения с понятными URL-адресами, которые можно использовать в поиске. URL-адреса не должны содержать расширения имен файлов и предназначены для

поддержки шаблонов именования URL-адресов, обеспечивающих адресацию, оптимизированную для поисковых систем (SEO) и для передачи репрезентативного состояния (REST).

- Поддержка использования разметки в существующих файлах страниц ASP.NET (ASPX), элементов управления (ASCX) и главных страниц (MASTER) как шаблонов представлений. Вместе с платформой ASP.NET MVC можно использовать существующие функции ASP.NET, например вложенные главные страницы, встроенные выражения (<%= %>), декларативные серверные элементы управления, шаблоны, привязку данных, локализацию и т. д.

- Поддержка существующих функций ASP.NET. ASP.NET MVC позволяет использовать такие функции, как проверка подлинности с помощью форм и Windows, проверка подлинности по URL-адресу, членство и роли, кэширование вывода и данных, управление состоянием сеанса и профиля, наблюдение за работоспособностью, система конфигурации и архитектура поставщика [2].

Zend Framework - это свободный каркас на PHP для разработки веб-приложений и веб-сервисов.

Zend старается следовать духу PHP, предоставляет простые интерфейсы и мощную функциональность для разработки приложений. Он предоставляет расширения для построения современных, быстрых и безопасных сайтов.

Основывается на идеях MVC. Разрабатывается компанией Zend. В Казахстане продуктами компании Zend среди фреймворков занимает высшие ступени. Не мало высокозагруженных сайтов, обслуживаемых казахстанскими хостинг компаниями. Как и во всем мире фреймворки довольно прилично уступают CMS (Content Management system).

Помимо MVC-компонентов Zend Framework содержит множество библиотек, полезных для построения приложения. [12]

ГЛАВА 2. ТЕХНОЛОГИЯ ПРОЕКТИРОВАНИЯ И СОЗДАНИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ ДЕЯТЕЛЬНОСТИ ФИРМ ПО ОКАЗАНИЮ ПЕРЕВОДЧЕСКИХ УСЛУГ

2.1. Требования к информационной системе фирмы переводов

Web- приложение должно выполнять следующие функции:

- Регистрация заказов
- Регистрация переводчиков ;
- создание таблицы отправки заказов (тип заказа, кто переводит, кто получает, когда отправляется переводчику, когда отправил переведенный тезис переводчик, когда получил заказчик перевод);
- просмотр и редактирование таблицы расписания переводов;
- добавление заказов в расписание;
- Не подлежащая редактированию база данных об истории заказов переводов;
- создание ведомостей выполненных переводов за неделю или месяц;
- просмотр и изменение базы файлов заказов и переводов;
- просмотр информации о заказчиках, о переводчиках и компаниях.

2.2. Проектирование базы данных

Для выполнений требований поставленных к приложению понадобится база данных которая будет хранить данные:

1. Об зарегистрированных заказах и их заказчиках;
2. Об переводчиках;
3. Об заказах которые отправлены переводчикам;
4. Об отправленных переводчиками обработанных (переведенных) документах;
5. Об языках перевода;
6. И данные которые будут хранит резервную копии сведений об всех заказах.

2.2.1 Обзор метода проектировки базы данных MySQL

MySQL (произносится май-эс-кю-эл) - очень быстрая, надежная система управления реляционными базами данных (СУРБД). Данные в ее базах хранятся в виде логически связанных между собой таблиц, доступ к которым осуществляется с помощью языка запросов SQL. Mysql - свободно распространяемая система. База данных позволяет эффективно хранить, искать, сортировать и получать данные. Сервер MySQL управляет доступом к данным, позволяя работать с ними одновременно нескольким пользователям, обеспечивает быстрый доступ к данным и гарантирует предоставление доступа только имеющим на это право пользователям. Следовательно, MySQL является многопользовательским, многопоточковым сервером. Он применяет SQL (Structured Query Language - язык структурированных запросов), используемый по всему миру стандартный язык запросов в базы данных.

В настоящее время пакет MySQL доступен как программное обеспечение с открытым исходным кодом, но в случае необходимости можно получить и коммерческие лицензии. [6-11]

Приступая к созданию высокозагруженного сайта, можно использовать множество различных продуктов.

Потребуется выбрать аппаратное обеспечение для Web-сервера, операционную систему, программное обеспечение Web-сервера, систему управления базами данных и язык программирования или создания сценариев.

Выбор некоторых из этих компонентов будет зависеть от уже произведенных выборов. Например, не все операционные системы могут работать на любом оборудовании, не все языки создания сценариев могут обеспечить подключение ко всем базам данных и т.д.

Система MySQL обладает такой же гибкостью.

2.2. Разработка базы данных

На рисунке Рис.2.1. представлена структура базы данных нашего web-приложения Treanslator данных состоит из шести таблиц: «clients», «user», «adminorder», «answer_order», «language», «translate».

База данных была реализована на основе встроенной СУБД MySql .

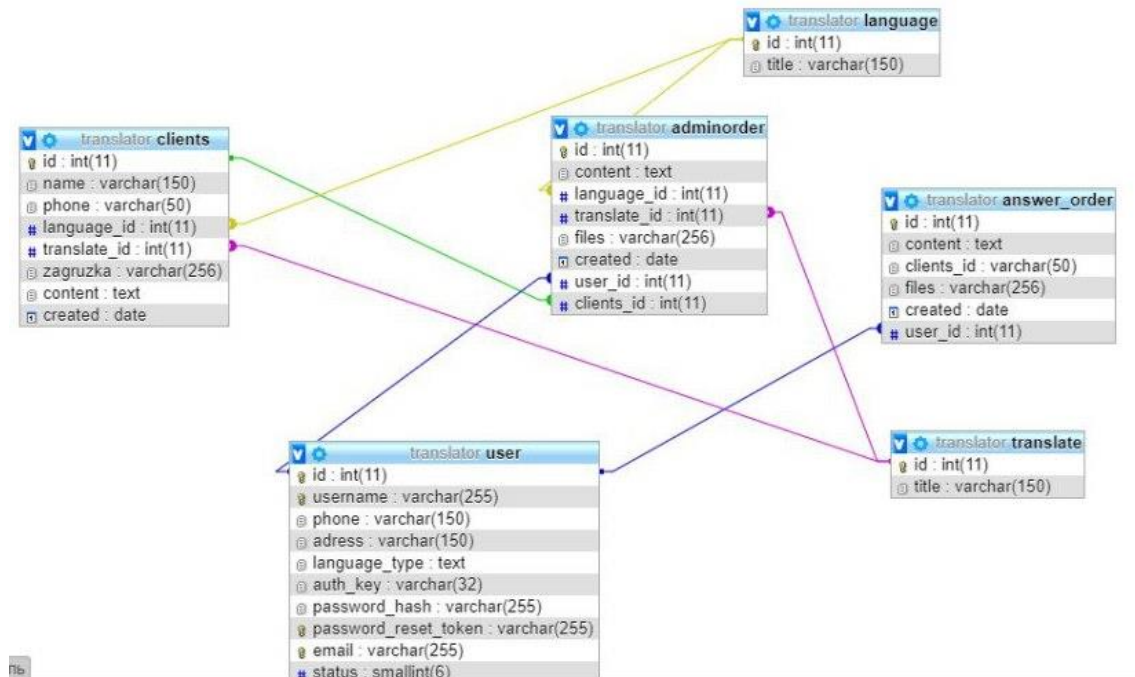


Рис.2.1. Связь таблиц в базе данных

Таблица «clients» будет хранить данные об заказах (Рис.2.2.). Для этого понадобятся следующие сведения

- ФИО заказчика;
- номер телефона чтобы связаться для подтверждения заказа;
- с какого языка на какой перевести текст;
- файл документа;
- комментарий, предпочтение заказчика;
- дата создание заказа;

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
1	id	int(11)			Нет	Нет		AUTO_INCREMENT	Изменить Удалить Первичный Ещё
2	name	varchar(150)	utf8_general_ci		Нет	Нет			Изменить Удалить Первичный Ещё
3	phone	varchar(50)	utf8_general_ci		Нет	Нет			Изменить Удалить Первичный Ещё
4	language_id	int(11)			Нет	Нет			Изменить Удалить Первичный Ещё
5	translate_id	int(11)			Нет	Нет			Изменить Удалить Первичный Ещё
6	zagruzka	varchar(256)	utf8_general_ci		Нет	Нет			Изменить Удалить Первичный Ещё
7	content	text	utf8_general_ci		Нет	Нет			Изменить Удалить Первичный Ещё
8	created	date			Нет	Нет			Изменить Удалить Первичный Ещё

Рис.2.2. Таблица «clients»

Таблица «user» будет хранить данные об переводчиках (Рис.2.3.). Она будут хранить следующие данные:

- ФИО переводчика;
- номер телефона чтобы связаться для подтверждение регистрации;
- с какого языка на какой может переводить тексты;
- пароль к доступу на личную страничку заказов а администратра;
- электронный адрес переводчика;
- статус переводчика (при неактивном статусе переводчик не будет иметь доступ к своей акаунта);

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
1	id	int(11)			Нет	Нет		AUTO_INCREMENT	Изменить Удалить Ещё
2	username	varchar(255)	utf8_unicode_ci		Нет	Нет			Изменить Удалить Ещё
3	phone	varchar(150)	utf8_unicode_ci		Нет	Нет			Изменить Удалить Ещё
4	adress	varchar(150)	utf8_unicode_ci		Нет	Нет			Изменить Удалить Ещё
5	language_type	text	utf8_unicode_ci		Нет	Нет			Изменить Удалить Ещё
6	auth_key	varchar(32)	utf8_unicode_ci		Нет	Нет			Изменить Удалить Ещё
7	password_hash	varchar(255)	utf8_unicode_ci		Нет	Нет			Изменить Удалить Ещё
8	password_reset_token	varchar(255)	utf8_unicode_ci		Да	NULL			Изменить Удалить Ещё
9	email	varchar(255)	utf8_unicode_ci		Нет	Нет			Изменить Удалить Ещё
10	status	smallint(6)			Нет	10			Изменить Удалить Ещё
11	created_at	int(11)			Нет	Нет			Изменить Удалить Ещё
12	updated_at	int(11)			Нет	Нет			Изменить Удалить Ещё

Рис.2.3. Таблица «user»

Таблица «adminorder» будет хранить данные об заказах которые переданы переводчикам (Рис.2.4.), а точнее:

- файл отправленный заказчиком;
- комментарий отправляемый переводчику (в комментариях будет храниться сообщение от администрации для переводчика);
- с какого языка на какой перевести текст;
- сам файл который надо перевести;
- датаправки файла для перевода переводчеку ;
- идентификационный номер переводчика которому отправляется заказ;
- идентификационный номер заказа сформированный заказчиком;

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
1	id	int(11)			Нет	Нет		AUTO_INCREMENT	Изменить Удалить Первичный Ещё
2	content	text	utf8_general_ci		Нет	Нет			Изменить Удалить Первичный Ещё
3	language_id	int(11)			Нет	Нет			Изменить Удалить Первичный Ещё
4	translate_id	int(11)			Нет	Нет			Изменить Удалить Первичный Ещё
5	files	varchar(256)	utf8_general_ci		Нет	Нет			Изменить Удалить Первичный Ещё
6	created	date			Нет	Нет			Изменить Удалить Первичный Ещё
7	user_id	int(11)			Нет	Нет			Изменить Удалить Первичный Ещё
8	clients_id	int(11)			Нет	Нет			Изменить Удалить Первичный Ещё

Рис.2.4. Таблица «adminorder»

Таблица «answer_order» будет хранить данные об переведенных заказах которые переданы переводчиками администрации (Рис.2.5.), а точнее:

- файл отправленный переводчиками;
- комментарий отправляемый администрации (в комментариях будет храниться сообщение от переводчика для администрации);
- переведенных файл;
- дата отправки перевода;
- идентификационный номер переводчика который отправляет переведенный документ;
- идентификационный номер заказа сформированный заказчиком;

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
1	id	int(11)			Нет	Нет		AUTO_INCREMENT	Изменить Удалить Первичный Ещё
2	content	text	utf8_general_ci		Нет	Нет			Изменить Удалить Первичный Ещё
3	clients_id	varchar(50)	utf8_general_ci		Нет	Нет			Изменить Удалить Первичный Ещё
4	files	varchar(256)	utf8_general_ci		Нет	Нет			Изменить Удалить Первичный Ещё
5	created	date			Нет	Нет			Изменить Удалить Первичный Ещё
6	user_id	int(11)			Нет	Нет			Изменить Удалить Первичный Ещё

Рис.2.5. Таблица «answer_order»

Также создали 2 дополнительные таблицы. В первой таблице будут храниться информации об языке на котором документ. Эта таблица называется “language”. Она хранит в себе только 2 поля :

- идентификационный номер;
- и сама информации об языка виде, например UZB, RUS, ENG или другие языки в этом формате.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
1	id	int(11)			Нет	Нет		AUTO_INCREMENT	Изменить Удалить Первичный Ещё
2	title	varchar(150)	utf8_general_ci		Нет	Нет			Изменить Удалить Первичный Ещё

Рис.2.6. Таблица “language”

Вторая таблица «translate» информации об языке на который должен он переводится заказ. Он также имеет 2 поля как и таблица «language» :

- идентификационный номер;
- и сама информации об языка виде, например UZB, RUS, ENG или другие языки в этом формате.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
1	id	int(11)			Нет	Нет		AUTO_INCREMENT	Изменить Удалить Первичный Ещё
2	title	varchar(150)	utf8_general_ci		Нет	Нет			Изменить Удалить Первичный Ещё

Рис.2.7. таблица «translate»

2.3. Требования к web-приложению

Web- приложение должно выполнять следующие функции:

- Регистрация заказов;
- Регистрация переводчиков ;
- создание таблицы отправки заказов (тип заказа, кто переводит, кто получает, когда отправляется переводчику, когда отправил переведенный тезис переводчик, когда получил заказчик перевод);
- просмотр и редактирование таблицы расписания переводов;
- добавление заказов в расписание;
- Не подлежащая редактированию база данных об истории заказов переводов;
- создание ведомостей выполненных переводов за неделю или месяц;
- просмотр и изменение базы файлов заказов и переводов;
- просмотр информации о заказчиках, о переводчиках и компаниях.

2.3.1. Проектирование интерфейса приложения

Веб-сайт должен обладать оригинальным графическим дизайном, обеспечивающим надлежащую функциональность. Для веб-сайта должен быть разработан набор уникальных элементов фирменного стиля и визуальной идентификации: знак, логотип, цветовое решение, условные значки для элементов интерфейса.

Дизайн должен обеспечивать соответствие следующим основным требованиям:

- графические элементы должны быть выполнены с учетом специфики представления графической информации на веб-страницах;
- доступ к основной информации должен быть максимально быстрым и внимание пользователя должно концентрироваться на наиболее важных навигационных элементах;

- основная информация, меню и другие элементы навигации должны быть доступны без горизонтального прокручивания полностью открытого окна браузера;

- навигация по разделам должна быть предельно интуитивно-понятной;

- следует уделить должное внимание заголовкам и титулам. Очень важно учесть систему отступов и выравниваний. Необходимо иметь определенные принципы в верстке заголовков, выделений, абзацев, чтобы текст правильно и легко воспринимался.

При разработке дизайна веб-сайта необходимо учитывать следующие принципы:

- системный подход к работ над дизайном, обобщение передовых тенденций;

- концентрация на посетителе и на информации (user centered design); - легкость в навигации по ресурсу (easy navigation);

- легкость в восприятии информации посетителями веб-сайта;

- минимальный "вес" страниц (в килобайтном отношении);

- соблюдение повторяемости элементов внутри системы (modularity);

- качественная и детальная проработка всей графики;

- при использовании на главной и других страницах иллюстративного образа должно быть соответствие теме раздела и не высокий объем файлов картинки. Обязательно стилевое единство иллюстраций и всего оформления;

- изобразительные элементы и даже просто цвет и тон фона не должны мешать читаемости текстовых объектов;

- изобразительные объекты не должны вытеснять основную информацию за пределы первого окна [16].

2.3.2. Парадигма MVC

Расшифровывается это понятие как Model View Controller. То есть для описания жизненного цикла программы используется три "типа данных", скажем так.

Модель - в ней описываются данные, которые хранятся в системе.

Представление - отображает данные в нужном виде.

Контроллер - принимает запрос от пользователя, получает данные из модели и передаёт их в представление.



Рис.2.8. Механизм работы MVC

Если при запросе не нужно обращаться к данным, хранящимся на сайте - то можно передавать данные для ответа сразу в представление.

Если ответом является json строка, или изображение, или просто какой-то набор данных не требующий дизайна(например вы разрабатываете API) - то ответ можно передавать прямо из контроллера.

Во фреймворке **Yii**(и не только в нём) всё это разнесено по папкам **models**, **views** и **controllers**

Контроллер

- Имя контроллера должно заканчиваться на *Controller*, например *PostController*, *UserController*, *ImageController* и т.п. Имя файла - точно такое же, с добавлением *.php*. В проекте они

AnswerController , *FreelanserController* , *OrdersController*,
SendController, *SiteController* (Рис.2.9.)

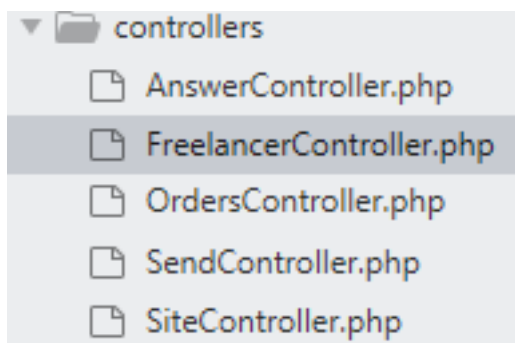


Рис.2.9. Список контроллеров в программе

- По-умолчанию классы контроллеров наследуются от класса **Controller**.
- Все методы класса контроллера, имена которых начинаются на *action* будут урлами. То есть если есть класс *GolosController* и в нем есть метод *actionBest* - то на сайте будет урл */golos/best* , который будет выводить на экран то, что вернёт метод *actionBest*.

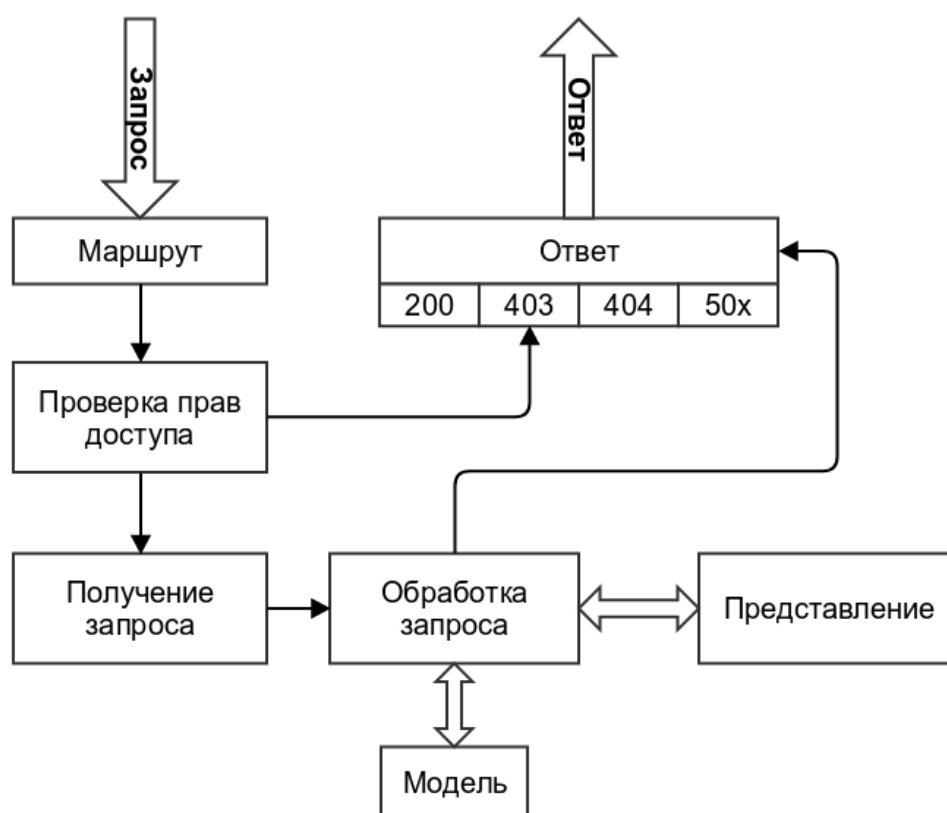


Рис.2.10. Механизм работы Модели

Модель

- Модели могут называться как угодно.
- По-умолчанию классы моделей наследуются от класса **Model**.
- Если экземпляр модели это запись в таблице БД - то класс модели наследуется от класса `\yii\db\ActiveRecord`.
- У большинства моделей есть "атрибуты" - это публичные или приватные свойства класса.

Представление

Представляет собой шаблон(обычный php файл), в котором будут доступны переменные, которые в него передаёт контроллер (Рис.2.11). В проекте мы созданы следующие файлы: *about.php*, *contact.php*, *error.php*, *index.php*, *languages.php*, *login.php*, *signup.php*, *resetPasword.php*.

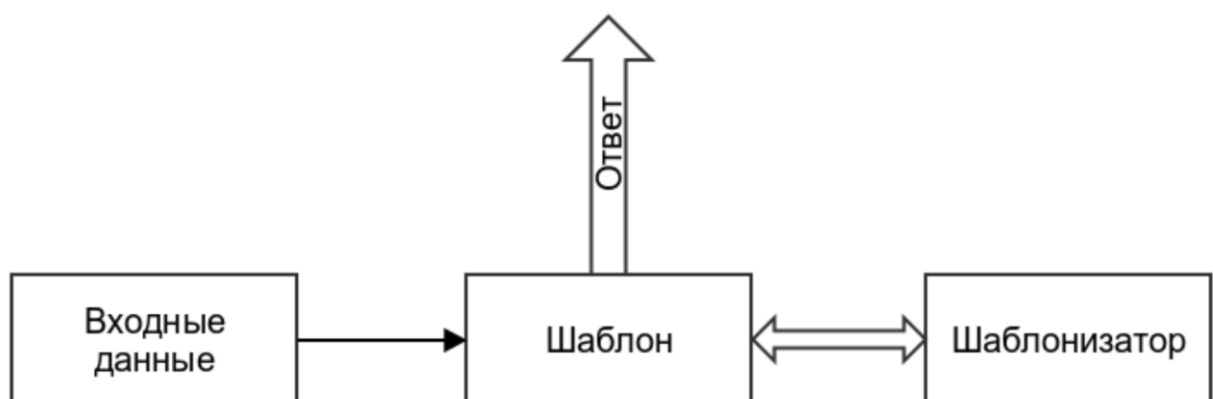


Рис.2.11. Механизм работы Представление

ГЛАВА 3. ТЕХНОЛОГИЯ СОЗДАНИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ ДЕЯТЕЛЬНОСТИ ФИРМ ПО ОКАЗАНИЮ ПЕРЕВОДЧЕСКИХ УСЛУГ

3.1. Разработка web-приложения

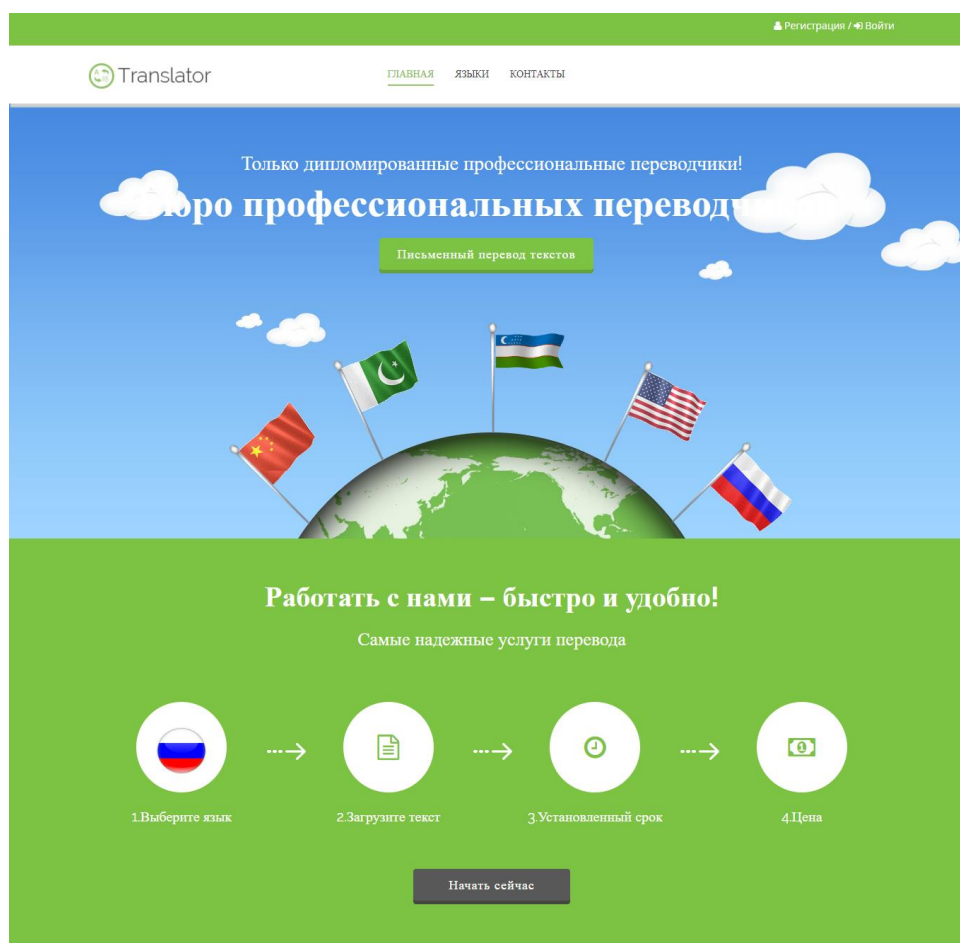
Приложение написано на объектно-ориентированном языке программирования PHP с применением технологии фреймворков Yii 2, Ajax, JQuery. Также использовался XML для передачи структурированных данных и приема данных из различных веб-форм, осуществленных на сайте. Для соединения и манипуляции с базой данных сайт использует особый класс Yii фреймворка «db». В качестве базы данных используется СУБД MySQL. Хранение паролей пользователей осуществляется не в чистом виде, а в виде хэш-функции. Хэш-функции генерируются алгоритмом MD5.

Ядро сайта спроектировано на основополагающих принципах стандарта MVC. MVC (Model-View-Controller) предполагает раздельное использование модели данных, интерфейса и взаимодействия с пользователем. Этот подход дает возможности проектированию веб-приложения каркасным, модульным методом. Достоинством модульного метода проектирования веб-приложения является то, что при изменении части приложения он не оказывает влияние на остальную часть программы.

3.2. Описание программы

Теперь, что касается внешнего вида главной страницы сайта, показанного на Рис 3.1. В топе сайта с права расположены ссылки для регистрации переводчиков и вход в личную страничку переводчика. После родним слева расположился логотип сайта, который ведет на главную страницу сайта. Далее меню навигации по сайту из 3 ссылок на главную, на страницу про языки которые переводит это фирма и их. контакты. Ниже слоган компании или предприятия, который характеризует компанию и приветственно встречает каждого нового посетителя сайта.

Далее по всей ширине сайта расположен горизонтальное меню навигации по сайту. При необходимости можно создать выпадающее меню для экономии места и группировании однотипных категорий.



Доступные языки

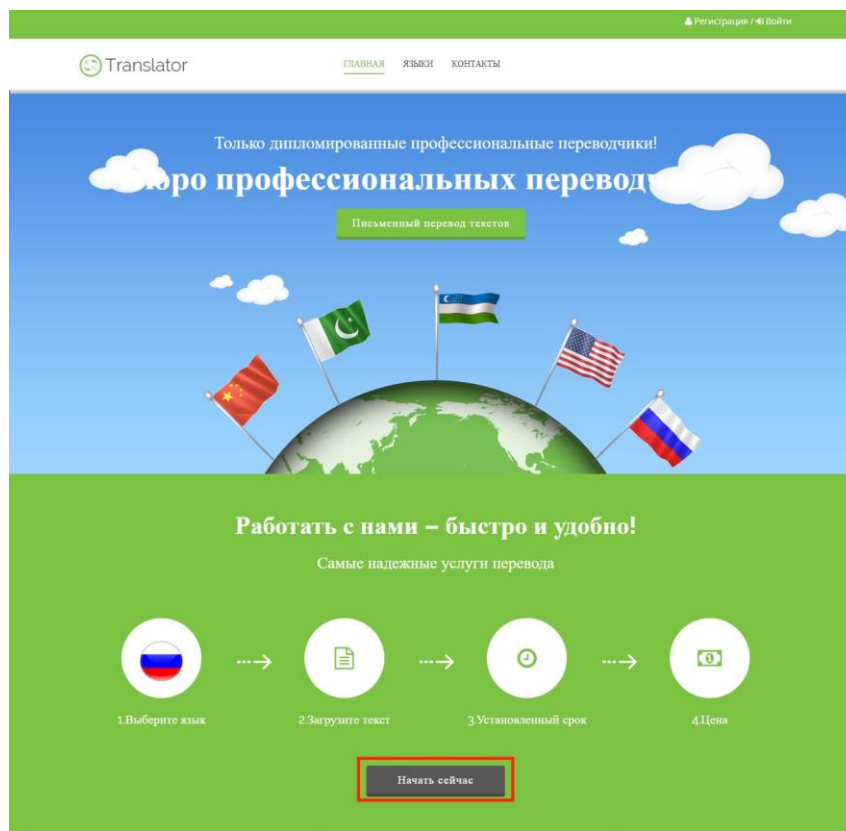
Рис 3.1. Главная страница

Пользователями являются штатные работники переводческих фирм, ихние субподрядчики-фрилансеры и конечно же заказчики.

Как. Выше уже было отмечено с web-приложением будут положатся 3 типа ползователей:

- 1 тип пользователей это “Заказчики”;
- 2 тип пользователей это “Преводчики”;
- 3 тип пользователей “Администрация”.

Пользователи 1-ого типа для оформление заказа выбирают кнопку “Начать сейчас” на главной странице сайта Рис 3.2.:



Доступные языки

Рис 3.2. Главная страница и кнопка “Начать сейчас”

После нажатие этой кнопки на странице появляется форма для осуществление заказа Рис 3.3.. Данные сохраняются в базу данных заказов:

Рис 3.3. Форма регистрации заказа

После отправлении заказа виде файла придет уведомление о принятии заказа.

Пользователи 2-го типа переводчики. Так как эти пользователи будут получать заказы для переводов они должны войти в личный кабинет. Если переводчик не зарегистрирован то он должен зарегистрироваться. Для этого на главной странице на самом верху по правой стороне есть кнопка “Регистрация” (Рис.3.4).

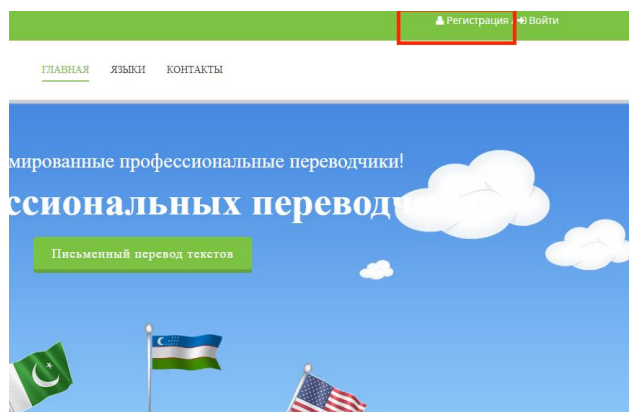


Рис3.4 кнопка “Регистрация”

Как только переводчик выберет этот элемент он перепет на страницу регистрации “Пользователя - переводчика” Рис.3.5.: Пользователь заполняет свои данные и нажимает кнопку “Регистрация”. И его кандидатура переводится на рассмотрение, компания связывается с ним и водит его в базу данных переводчиков компании.

The image shows a registration form on a website. The form is titled 'Translator' and has a green header bar with 'Регистрация / Войти' (Registration / Login) on the right. The form fields are: Username (Sowb), Phone (+998901234567), Address (Samarkand Shahar), Email (sowb@gmail.com), Password (*****), and Language Type (Rus Uzbek). There is a 'Регистрация' (Registration) button at the bottom of the form.

Рис.3.5. Форма регистрации переводчиков

Для в хода в свою личный кабинет переводчик должен нажать кнопку “Вход” на главной странице на самом верху по правой стороне. (Рис.3.6)

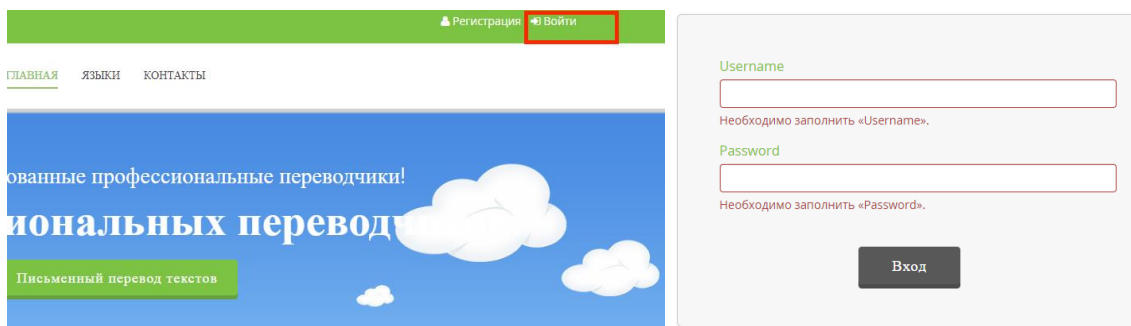


Рис. 3.6. Кнопка “Вход” и форма “Входа в личный кабинет переводчика”

После этого пользователь переходит на свою личную страницу. Личная страничка состоит и верхней части сайта на правой части имя пользователя - переводчика (Рис.3.7.-4) в виде выпадающего меню, который при нажатие выводит 2 под меню “Просмотр заказов” и “Выход” (Рис.3.7.-3). Тела сайта 2 таблицы, которые, первая все заказы от администрации (Рис.3.7.-1) и вторая история выполненных заказов (Рис.3.7.-2). После выполнение заказа от администрации Переводчик должен отправить переведенных документ. Для этого он выбирает кнопку “Отправить заказ” (Рис.3.7) и открывается форма отправки перевода.

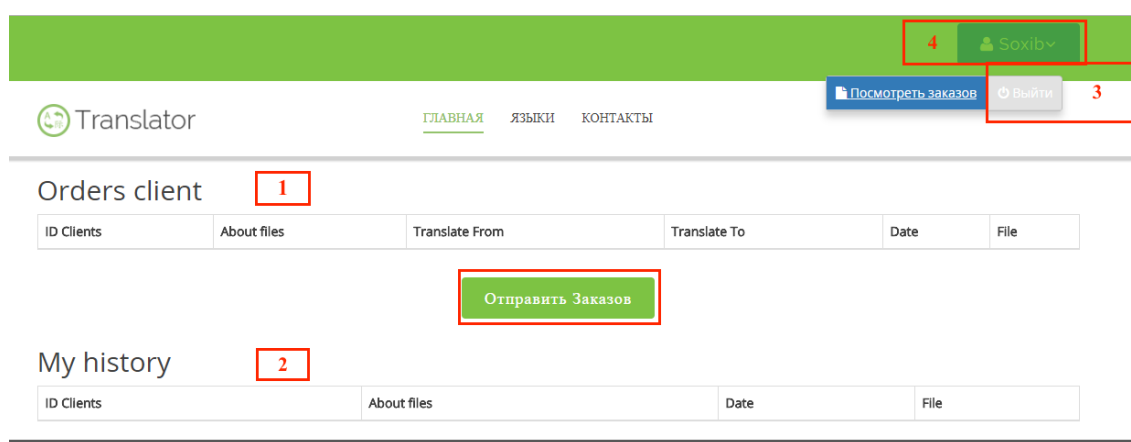


Рис.3.7. Личная страница переводчика

Важный момент при заполнение формы в том что переводчик отправляет заказ по айди (ID) клиента. Потом пишет какие-то указание если они у него возникли при переводе и выбирает файл который отправляет. При нажатие кнопки отправить файл отправляется администрации.

Modal title ×

Clients ID

5

Content

I am translate this from Uzbek to Russian

File

Выберите файл

Документ ...ord.docx

Отправить

Рис.3.8. Форма отправки заказа

3 тип пользователей администратия в этом документ обороте выполняет роль посредника и поручитель что перевод буден выполнен высоко-квалификационно. Поэтому все файлы проходит через их проверку.

Страничка администратии имеет отдельный доступ через бек-енд (back-end) сайта. Администратии входит со соим логин и паролем и них открывается следующая страница:

Моя компания

Главная

SendOrder

AnswerOrder

OrderClients

свободный художник

Авторизоваться

Главная / Заказ клиентов

Заказ клиентов

Создать заказ клиентов

Показаны 1-5 из 5 предметов.

#	Я БЫ	Язык оригинала	Язык перевода	Дата	файлы	
1	5	UZ	анг	2018-03-29	6 марта 2018 г. состоялся Ученый Совет СамМИ.docx	
2	6	6116516	3121321	2018-03-29	6 марта 2018 г. состоялся Ученый Совет СамМИ.docx	
3	7	UZ	анг	2018-03-30	339664560-2018-01-01.csv	
4	8	узбек	английский	2018-04-20	IMG-10.jpg	
5	9	cscsc	ОКК	2018-05-18	Parviza.docx	

© Моя компания 2018

Работает на Yii Framework

Рис.3.9. Главная страница администратора

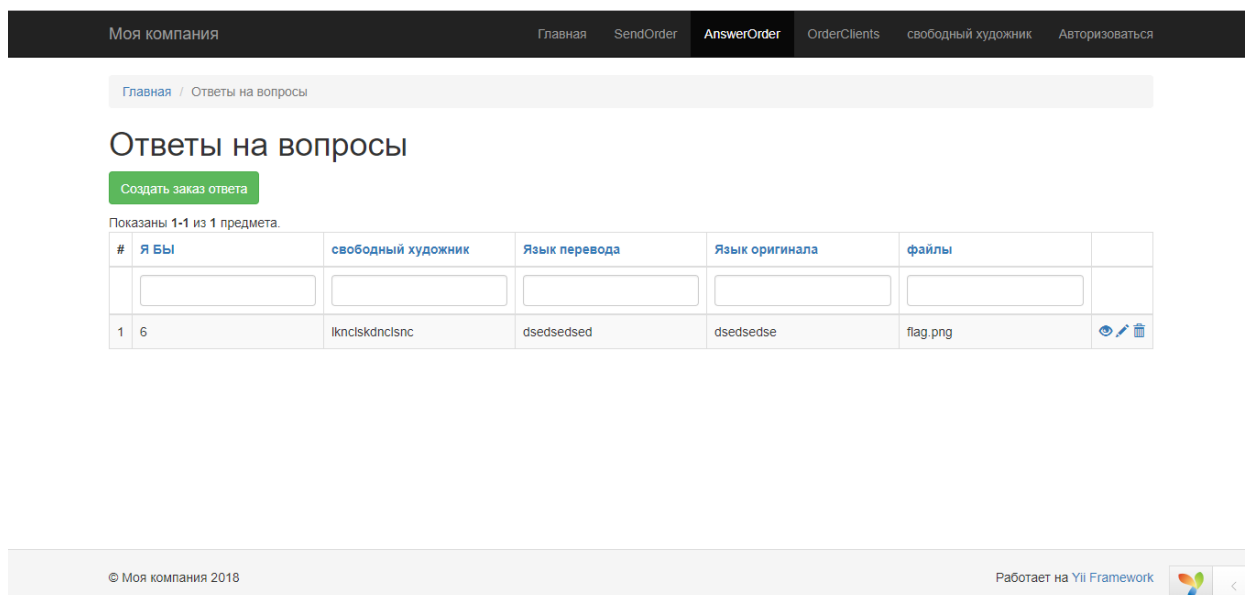


Рис.3.10. Страница полученных переводов

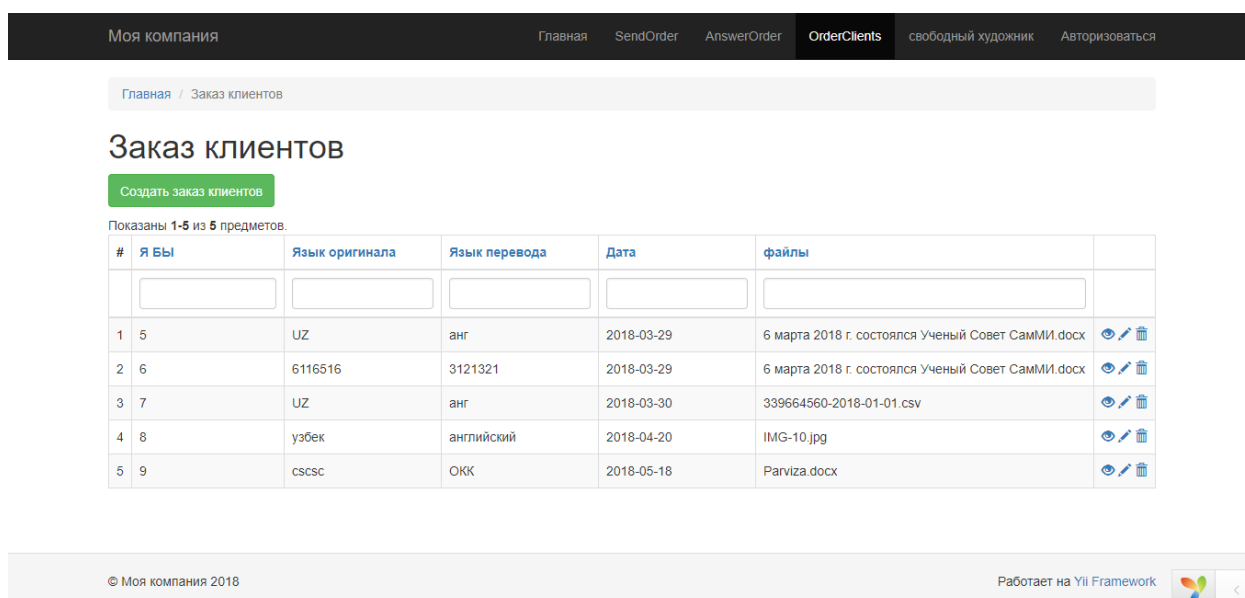


Рис.3.11. Страница полученных заказов

3.3. Технические решения безопасности труда

Анализ потенциально опасных и вредных производственных факторов

Опасные и вредные производственные факторы по природе возникновения делятся на следующие группы:

- физические;
- химические;
- психофизиологические;
- биологические.

В помещении лаборатории на программиста могут негативно действовать следующие физические факторы:

- повышенная и пониженная температура воздуха;
- чрезмерная запыленность и загазованность воздуха;
- повышенная и пониженная влажность воздуха;
- недостаточная освещенность рабочего места;
- превышающий допустимые нормы шум;
- повышенный уровень ионизирующего излучения;
- повышенный уровень электромагнитных полей;
- повышенный уровень статического электричества;
- опасность поражения электрическим током;
- блеклость экрана дисплея.

К химически опасным факторам, постоянно действующим на программиста относятся следующие: возникновение, в результате ионизации воздуха при работе компьютера, активных частиц.

Биологические вредные производственные факторы в данном помещении отсутствуют.

К психологически вредным факторам, воздействующим на оператора в течение его рабочей смены можно отнести следующие:

- нервно-эмоциональные перегрузки;
- умственное напряжение;
- перенапряжение зрительного анализатора.

Далее более подробно рассмотрены опасные и вредные факторы, воздействующие на программиста, возникшие в связи с разработкой данной системы.

Микроклимат рабочей зоны программиста

Микроклимат производственных помещений - это климат внутренней среды этих помещений, который определяется действующими на организм человека сочетаниями температуры, влажности и скорости движения воздуха.

Для создания и автоматического поддержания в лаборатории независимо от наружных условий оптимальных значений температуры, влажности, чистоты и скорости движения воздуха, в холодное время года используется водяное отопление, в теплое время года применяется кондиционирование воздуха. Кондиционер представляет собой вентиляционную установку, которая с помощью приборов автоматического регулирования поддерживает в помещении заданные параметры воздушной среды.

Воздействие шума на программиста. Защита от шума

В помещениях с низким уровнем общего шума, каким является лаборатория где работает программист, источниками шумовых помех могут стать вентиляционные установки, кондиционеры или периферийное оборудование для ЭВМ (плоттеры, принтеры и др.). Длительное воздействие этих шумов отрицательно сказывается на эмоциональном состоянии персонала.

Согласно УзДСт 12.0.002-2005 [9] эквивалентный уровень звука не должен превышать 50 дБА. Для того, чтобы добиться этого уровня шума рекомендуется применять звукопоглощающее покрытие стен.

В качестве мер по снижению шума можно предложить следующее:

- облицовка потолка и стен звукопоглощающим материалом (снижает шум на 6 - 8 дб);
- экранирование рабочего места (постановкой перегородок, диафрагм);
- установка в компьютерных помещениях оборудования, производящего минимальный шум;
- рациональная планировка помещения.

Опасность повышенного уровня напряженности электромагнитного поля

Электромагнитные поля, характеризующиеся напряженностями электрических и магнитных полей, наиболее вредны для организма человек. Основным источником этих проблем, связанных с охраной здоровья людей,

использующих в своей работе автоматизированные информационные системы на основе персональных компьютеров, являются дисплеи (мониторы), особенно дисплеи с электронно-лучевыми трубками. Они представляют собой источники наиболее вредных излучений, неблагоприятно влияющих на здоровье программиста.

ПЭВМ являются источниками таких излучений как:

- ультрафиолетового 200 - 400 нм;
- видимого 400 - 700 нм,
- ближнего инфракрасного 700 - 1050 нм;
- радиочастотного 3 кГц - 30 МГц;
- электростатических полей.

Может возникнуть опасность по уровням напряженности электромагнитного поля. На расстоянии 5 - 10 см от экрана и корпуса монитора уровни напряженности могут достигать 140 В/м по электрической составляющей, что значительно превышает допустимые значения [9]. Предельно допустимые значения характеристик ЭМП указана в таблице 5.1.

Т а б л и ц а 5.1 - Предельно допустимые значения характеристик ЭМП

Наименование параметров	Допустимое Значение
Напряженность электромагнитного поля по электрической составляющей на расстоянии 50 см от поверхности видеомонитора	10 В/м
Напряженность электромагнитного поля по магнитной составляющей на расстоянии 50 см от поверхности видеомонитора	0,3 А/м
Напряженность электростатического поля не должно превышать: для взрослых пользователей	20 кВ/м
Напряженность электромагнитного поля на расстоянии 50 см вокруг ВДТ по электрической составляющей должна быть не более:	
в диапазоне частот 5 Гц - 2 кГц;	25 В/м
в диапазоне частот 2 - 400 кГц	2,5 В/м
Плотность магнитного потока должна быть не более:	
в диапазоне частот 5 Гц - 2 кГц;	250 нТл
в диапазоне частот 2 - 400 кГц	25 нТл
Поверхностный электростатический потенциал не должен превышать	500 В

Ультрафиолетовое излучение полезно в небольших количествах, но в больших дозах приводит к дерматиту кожи, головной боли, рези в глазах. Инфракрасное излучение приводит к перегреву тканей человека (особенно хрусталика глаза), повышению температуры тела. Уровни напряженности электростатических полей должны составлять не более 20 кВ/м. Поверхностный электростатический потенциал не должен превышать 500 В.

При повышенном уровне напряженности полей следует сократить время работы за компьютером, делать пятнадцатиминутные перерывы в течение полутора часов работы и, конечно же, применять защитные экраны.

Защитный экран, изготавливаемый из мелкой сетки или стекла, собирает на себе электростатический заряд. Для снятия заряда экран монитора заземляют.

Для предупреждения внедрения опасной техники все дисплеи должны проходить испытания на соответствие требованиям безопасности.

Так как работа программиста по виду трудовой деятельности относится к группе В - творческая работа в режиме диалога с ЭВМ, а по напряженности работы ко II категории тяжести, я предлагаю сократить время работы за компьютером, делать перерывы, суммарное время которых должно составлять 50 минут при 8-ми часовой смене.

ЗАКЛЮЧЕНИЕ

В данной выпускной квалификационной работе мною было спроектировано и реализовано web-приложение, в котором удобный web-интерфейс сочетается с функционалом приложения. В ходе выполнения работы были проделаны следующие результаты:

1. Разработан web - приложение для переводческой фирмы.
2. Разработана база данных в приложении MyRphAdmin.
3. Изучены методы работы с фреймворком Yii2
4. Проведено тестирование разработанного web - приложения .

В разработке интерфейса были использованы новейшие технологии twitter Bootstrap, использующий каноны CSS 3, и javascript. Самая последняя разработка язык гипертекстовый разметки HTML 5 также был использован в данном проекте. Это дает гарантию, что сайт будет одинаково правильно открываться на любом цифровом носителе - как на планшете, так и на компьютере.

В последнее время задача информационной поддержки деятельности организации становится действительно актуальной. Для переводческих фирм которые работают с переводчиками с разных точек мира эти системы становятся одних из важнейших инструментов в их делопроизводстве.

Так как система была разработана в сотрудничестве с сотрудниками фирмы по оказания переводческих услуг, можно предположить, что этот программный продукт может быть с успехом применен в информационной сфере Узбекистана. Проект реализован по современным технологиям, с обеспечением высокой степенью защиты данных, соответствует всем требованиям, предъявляемых к такому типу сервисам.

СПИСОК ЛИТЕРАТУРЫ

1. <http://president.uz/ru/lists/view/1414> -Президент провёл совещание по развитию сферы ИКТ: 09.01.2018
2. Роб П., Коронел К. Системы баз данных: проектирование, реализация и управление. - 5-е изд. - СПб.: БХВ-Петербург, 2004. - 1040 с.
3. Горбунов-Посадов М.М. Расширяемые программы. - М.: Полиптих, 1999. - 336 с.
4. MySQL разработка Web-приложений. – 4-е изд., перераб. И доп. — МПб.: БХВ-Петербург, 2013. — 560 с.
5. Scott Guthrie. ASP.NET MVC 1.0 650 с.
6. Л. Томсон, Л. Веллинг. Разработка Web-приложений на PHP и MySQL. - М.: Diasoft, 2003. - 672 с.
7. Common Sense Advisory. "Translation management systems and subcategories". *Multilingual* (March 2007): 83–86.
8. Разработка и управление требованиями. Практическое руководство пользователя/Элизабет Халл, Кен Джексон, Джемир Дик. — СПб.: БХВ-Петербург, 2012. — 229 с.
9. УзДСт 12.0.002-2005. Система стандартов научной организации труда, промышленной санитарии гигиены, безопасности. - М.: Агенство Узстандарт - 10с.

Интернет ресурслары:

10. <http://www.yiiframework.ru/>
11. <https://framework.zend.com>
12. <http://mysql.com/>
13. <https://www.apachefriends.org/index.html>

ПРИЛОЖЕНИЕ

Листинга главного контроллера приложения

```
<?php
namespace frontend\controllers;

use Yii;
use yii\base\InvalidParamException;
use yii\web\BadRequestHttpException;
use yii\web\Controller;
use yii\filters\VerbFilter;
use yii\filters\AccessControl;
use common\models\LoginForm;
use frontend\models>PasswordResetRequestForm;
use frontend\models\ResetPasswordForm;
use frontend\models\SignupForm;
use frontend\models\ContactForm;
use yii\web\UploadedFile;
use yii\helpers\Url;
use common\models\Clients;
use common\models\Adminorder;
use common\models\AnswerOrder;

/**
 * Site controller
 */
class SiteController extends Controller
{
    /**
     * @inheritdoc
     */
    public function behaviors()
    {
        return [
            'access' => [
                'class' => AccessControl::className(),
                'only' => ['logout', 'signup'],
            ],
        ];
    }
}
```



```

        'rules' => [
            [
                'actions' => ['signup'],
                'allow' => true,
                'roles' => ['?'],
            ],
            [
                'actions' => ['logout'],
                'allow' => true,
                'roles' => ['@'],
            ],
        ],
    ],
    'verbs' => [
        'class' => VerbFilter::className(),
        'actions' => [
            'logout' => ['post'],
        ],
    ],
];
}

/**
 * @inheritdoc
 */
public function actions()
{
    return [
        'error' => [
            'class' => 'yii\web\ErrorAction',
        ],
        'captcha' => [
            'class' => 'yii\captcha\CaptchaAction',
            'fixedVerifyCode' => YII_ENV_TEST ? 'testme' : null,
        ],
    ],

```

```

];
}

/**
 * Displays homepage.
 *
 * @return mixed
 */
public function actionIndex()
{
    $model = new Clients();

    $model->created = date('Y-m-d H:i:s');
    if ($model->load(Yii::$app->request->post())) {
        $model->file = UploadedFile::getInstance($model,'file');

        $model->file->saveAs(
            Url::to('.../frontend/web/upload/') . $model->file->baseName.'.'.$model->file-
>extension
        );

        $model->zagruzka = $model->file->baseName.'.'.$model->file->extension ;
        $model->save(false);

        Yii::$app->session->setFlash('success', 'Спасибо, что обратились к нам. Мы ответим
вам как можно скорее.');
```

```

        return $this->redirect(['index']);
    } else {
        return $this->render('index', [
            'model' => $model,
        ]);
    }
}

```

```

}

/**
 * Logs in a user.
 *
 * @return mixed
 */
public function actionLogin()
{
    if (!Yii::$app->user->isGuest) {
        return $this->goHome();
    }

    $model = new LoginForm();
    if ($model->load(Yii::$app->request->post()) && $model->login()) {
        return $this->goBack();
    } else {
        return $this->render('login', [
            'model' => $model,
        ]);
    }
}

/**
 * Logs out the current user.
 *
 * @return mixed
 */
public function actionLogout()
{
    Yii::$app->user->logout();

    return $this->goHome();
}

```

```

/**
 * Displays contact page.
 *
 * @return mixed
 */
public function actionContact()
{
    $model = new ContactForm();
    if ($model->load(Yii::$app->request->post()) && $model->validate()) {
        if ($model->sendEmail(Yii::$app->params['adminEmail'])) {
            Yii::$app->session->setFlash('success', 'Thank you for contacting us. We will respond
to you as soon as possible.');
```

} else {
 Yii::\$app->session->setFlash('error', 'There was an error sending your message.');

}

return \$this->refresh();
} else {
 return \$this->render('contact', [
 'model' => \$model,
]);
}
}

/**
 * Displays about page.
 *
 * @return mixed
 */

/**
 * Signs user up.
 *

```

* @return mixed
*/
public function actionSignup()
{
    $model = new SignupForm();
    if ($model->load(Yii::$app->request->post())) {
        if ($user = $model->signup()) {
            if (Yii::$app->getUser()->login($user)) {
                return $this->goHome();
            }
        }
    }

    return $this->render('signup', [
        'model' => $model,
    ]);
}

/**
 * Requests password reset.
 *
 * @return mixed
 */
public function actionRequestPasswordReset()
{
    $model = new PasswordResetRequestForm();
    if ($model->load(Yii::$app->request->post()) && $model->validate()) {
        if ($model->sendEmail()) {
            Yii::$app->session->setFlash('success', 'Check your email for further instructions.');

            return $this->goHome();
        } else {
            Yii::$app->session->setFlash('error', 'Sorry, we are unable to reset password for the
provided email address.');
        }
    }
}

```

```

    }

    return $this->render('requestPasswordResetToken', [
        'model' => $model,
    ]);
}

/**
 * Resets password.
 *
 * @param string $token
 * @return mixed
 * @throws BadRequestHttpException
 */
public function actionResetPassword($token)
{
    try {
        $model = new ResetPasswordForm($token);
    } catch (InvalidParamException $e) {
        throw new BadRequestHttpException($e->getMessage());
    }

    if ($model->load(Yii::$app->request->post()) && $model->validate() && $model->resetPassword()) {
        Yii::$app->session->setFlash('success', 'New password saved.');
```

```

        return $this->goHome();
    }

    return $this->render('resetPassword', [
        'model' => $model,
    ]);
}

public function actionLanguages(){
    return $this->render('languages');
```

```

    }

    public function actionFreelancer(){
        $model = new AnswerOrder;
        $userId = Yii::$app->user->id;
        $model->user_id = $userId;
        $model->created = date('Y-m-d H:i:s');

        if ($model->load(Yii::$app->request->post())) {
            $model->file = UploadedFile::getInstance($model,'file');

            $model->file->saveAs(
                Url::to(' ../frontend/web/upload/answerorders/') . $model->file->baseName.'.'.$model->file->extension
            );
            $model->files = $model->file->baseName.'.'.$model->file->extension ;
            if ($model->save()){

                return $this->redirect( [ 'freelancer', 'id' => $model->id ] );

            }
            } else {

                return $this->render('freelancer', [
                    'model' => $model,
                ]);
            }
        }
    }
}

```