МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Институт **математики, физики и информационных технологий**

Кафедра **«Прикладная математика и информатика»**

09.03.03 ПРИКЛАДНАЯ ИНФОРМАТИКА

ПРИКЛАДНАЯ ИНФОРМАТИКА В СОЦИАЛЬНОЙ СФЕРЕ

БАКАЛАВРСКАЯ РАБОТА

на тему **Разработка мобильного мессенджера с шифрованием данных**

Давыдов А.Е. _____

Султанов Т.Г. _____

Допустить к защите
Заведующий кафедрой к.тех.н, доцент, А.В. Очеповский _____

«_____»_____20___г.

Тольятти 2017

АННОТАЦИЯ

Целью бакалаврской работы является разработка и реализация приложения обмена сообщениями с криптографической защитой информации, адаптированного под сотовые телефоны и персональный компьютер средствами языка программирования PHP и реляционной СУБД MySQL.

Объект исследования: сервис по обмену сообщений между преподавателями и студентами с шифрованием данных.

Предмет исследования: технология шифрования данных.

Методы исследования: реинжиниринг бизнес-процессов обмена сообщениями**,** методы структурного и объектно-ориентированного анализа и проектирования.

В аналитической части произведен анализ предметной области «КАК ЕСТЬ», на основе структурного подхода разработана концептуальная модель «КАК ДОЛЖНО БЫТЬ» бизнес-процесса тестирования системы по обмену сообщениями. Центра. Выработаны требования к внедряемой ИС. В качестве средств разработки выбрана язык программирования php, СУБД - MSSQL.

На стадии логического проектирования на основе объектно-ориентированного подхода разработана логическая модель ИС. С помощью методологии IDEF1X разработана логическая модель данных ИС.

Реализована ИС и даны рекомендации по ее аппаратно-программному обеспечению.

Тестирование ИС подтвердило соответствие ее функциональности установленным требованиям.

Работа содержит 55 страниц, 35 рисунков, 10 таблиц, 67 источников.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

Безопасность личных данных всегда была актуальной проблемой. Одной из важнейших вещей, нуждающихся в надежной защите, является переписка. Множество людей по всему миру используют мобильные приложения как для личного, так и для делового общения. Последние годы конкуренция на рынке мессенджеров очень высока. Доступный Интернет у каждого в смартфоне позволил мессенджерам стать самыми часто используемыми приложениями. Проанализировав самые популярные из мобильных мессенджеров на устойчивость к различным угрозам, можно прийти к выводу, что многие из них не обеспечиваю надежной защиты. В большинстве из них вся переписка хранится на удаленных серверах в незашифрованном виде и в любой момент может быть прочитана или передана. Конечно, некоторые приложения осуществляют шифрование данных клиент-клиент, но существует еще множество угроз, например, таких как чтение данных сервис-провайдерами, перехват текущего ключа шифрования, неподлинная личность собеседника и т.д. Особенно актуальна это проблема среди возможно корпоративных организации, где есть свои особенности и правила. В Тольяттинском государственном университете (ТГУ) есть внутренняя почта для общения преподавателей и студентов, но это не всегда удобно, так как, любой человек в среднем проверяет почту один раз в день, но бывают ситуации, когда нужно срочно связаться с преподавателем.

На отечественном компьютерном рынке существуют достаточно большое количество программных продуктов позволяющих связаться с человеком, но к сожалению, не все программы отвечают требованиям Федерального закона № 152-ФЗ от 27.07.2006 г. "О персональных данных". При всем многообразии программного обеспечения на рынке программных продуктов отсутствуют программы, которые можно было бы применить непосредственно в условиях ТГУ. Поэтому создание информационной системы данного предприятия носило узкий прикладной характер и, в связи с этим, потребовало учитывать определенные особенности, обеспечивающие использование нестандартных

свойств подсистемы.

Цель создания системы можно разделить на следующие группы:

1. Возможность общения средствами Интернет между преподавателями и студентами без необходимости прямого контакта;
2. Шифрование личных данных и сообщений

Целью бакалаврской работы является разработка и реализация приложения обмена сообщениями с криптографической защитой информации адаптированного по сотовые телефоны и персональный компьютер средствами языка программирования PHP и реляционной СУБД MySQL.

Для достижения поставленной цели необходимо решить следующие задачи:

- провести анализ теоретической и научно-методической литературы по проблеме криптографической защитой;
- рассмотреть алгоритмы и протоколы для организации обеспечения безопасности личной переписки;
- разработать концепцию системы:
- изучить возможные средства реализации и выбрать наиболее эффективное;
- спроектировать концептуальную модель приложения;
- построить логическую и физическую модели базы данных;
- проанализировать некоторые существующие мессенджеры;
- разработать и спроектировать элементы системы.

Объект исследования: сервис по обмену сообщений между преподавателями и студентами с шифрованием данных.

Предмет исследования: Предметом исследования является технология шифрования данных.

Теоретическая и практическая значимость бакалаврской работы. Теоретическая значимость бакалаврской работы заключается в выработке на основании проведенного анализа предметной области идей по реализации новой системе обмена сообщениями.

Практическая значимость: состоит в автоматизации системы, которая позволит сократить временные затраты на общение преподавателей и студентов с помощью шифрования данных.

Цель, сформулированные задачи, предмет и объект исследования определили структуру и содержание бакалаврской работы.

Структура бакалаврской работы. Бакалаврская работа состоит из введения, трех глав, заключения и библиографического списка.

Во **введении** обоснована актуальность темы бакалаврской работы, определена ее цель, на основании которой сформулированы объект, предмет и задачи исследования, изложена практическая значимость полученных результатов.

В **главе 1** «Аналитическая часть» проводится обследование предметной области, описывается существующее состояние деятельности компании, анализируются бизнес-процессы обмена сообщений проводится анализ уровня автоматизации.

В **главе 2** «Проектная часть» проектируется структура разработки элементов системы обмена сообщениями, предъявляются требования к разрабатываемой системе.

В **главе 3** где предъявляются требования к архитектуре разрабатываемой системы, происходит описание основных процессов в виде блок-схем и описание экранных форм. Производится экономическая эффективность системы и этап тестирования системы.

В **заключении** излагаются основные выводы и результаты проведенной работы, определяются направления дальнейшего совершенствования информационной системы.

Глава 1 Функциональное моделирование предметной области

1.1. Технико-экономическая характеристика предметной области

Основой современного общества является информационный обмен. Информация является одной из самых фундаментальных сущностей окружающего нас мира. И с каждым днем обмен существующей информацией только растет. Растет и скорость обмена информацией. Развивается множество видов представления данных, а так же форма их передачи и представления пользователям. С такой же быстротой информация устаревает. Для ее обновления используются различные системы связи с различной степенью оперативности. Именно задача обмена информацией стала причиной появления различного рода сетей, технических информационных систем, в том числе компьютерных сетей.

Сначала сети объединяли отдельные компьютеры в пределах комнаты или здания. Далее появились сети, объединявшие компьютеры разных городов и стран. И сегодня мы можем наблюдать глобальную сеть, которая на самом деле является объединением множества различных систем более низкого уровня. Благодаря сети Интернет человечество получило в свои руки уникальную систему, все возможности которой, до конца не исследованы до сих пор. Говоря об обмене информацией, или в конечном итоге данными, важнейшим фактором является не только скорость обмена, но и возможность взаимодействия с источником информации, возможность задавать вопрос и получать ответ в реальном масштабе времени, получать свежие новости и сообщать их другим. Все это становиться особенно значимо, когда мы общаемся не с безликим сервером, а с вполне реальным человеком, который находится, может быть на другом конце земного шара, а может быть в соседней с вами комнате и вне зависимости от этого, вы можете в мгновение ока передать ему необходимую информацию и столь же быстро получить ответ. Именно такой обмен информацией называется интерактивным. Слово интерактивный происходит от английского «interact» и обычно переводиться как «двустороннее взаимодействие

в реальном времени» или как глагол со значением влиять друг на друга. Сегодня одной из наиболее перспективных современных технологий интерактивного общения становиться Instant Messanging (англ. «мгновенное сообщение»). Instant Messanging это сетевой сервис обмена сообщениями в «реальном» времени, т.е. с фактически мгновенной доставкой, когда задержка не ощутима.

Наиболее актуален вопрос обмена сообщениями, является для корпоративных компании, где наиболее остро стоит вопрос по охране персональных данных. Проблема информационной безопасности в корпоративных сетях сегодня очень остро стоит перед компаниями любого уровня. Эта задача стоит и перед ТГУ, так как большая часть обучения студентов происходит самостоятельно.

Таким образом, разрабатываемая система должна выполнять предъявленные функциональные требования и тем самым удовлетворять потребности заказчика.

1.2 Обоснование необходимости автоматизированного варианта решения и формирование требований к новой технологии

Работа системы, направленной на поддержку обменом сообщений выглядит следующим образом. Основным лицом является пользователь (пользователь системы), для которого определены следующие функциональные требования деятельности, изображенные на рис. 1.1.

Рисунок 1.1 - Функциональные требования деятельности пользователя

Каждое действие, представленное на схеме, суть множество вложенных процессов.

Опишем функциональные требования, выполняемые администратором системы. Схема изображена на рис. 1.2.

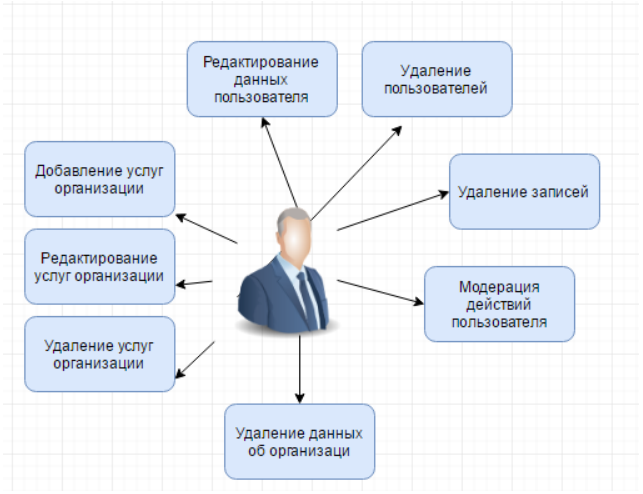

Рисунок 1.2 - Функциональные требования деятельности администратора

Основной информацией в системе являются:

1) личные данные пользователей;
2) информация об организации и ее услугах;
3) информация о связи между контактами в системе.

1.3 Анализ существующих разработок на предмет соответствия сформулированным требованиям

На данный момент программы для сетевого общения представлены в широком ассортименте, разнообразны по функциональным возможностям и целям, для которых применяются. При этом на сегодняшний день на рынке программного обеспечения не представлено качественных и удобных программ для сетевого общения, которые бы удовлетворяли требованиям большинства пользователей. В тоже время многие из представленных программных продуктов, не дают качественных возможностей шифрования личной переписки. Рассмотрим наиболее распространенные, популярные продукты, которые предоставляют возможность коллективного общения, в том числе возможность не только текстового, но и визуального (графического) обмена информацией.

Сегодня на рынке мессенджеров серьезная конкуренция (рис. 1). Почти у каждой социальной сети есть свой мессенджер: Facebook Messenger, Вконтакте, Hangouts (в недавнем прошлом GoogleTalk). Так же свои «родные» мессенджеры есть и у мобильных ОС: iMessage для iOS, Messenger для WindowsPhone, BlackBerryMessenger для BlackBerry. Многие слышали про Viber, WhatsApp, Skype и прочие приложения, о которых вы наверняка уже наслышаны [1]. Но есть мессенджеры, которые не успели нашуметь на российском рынке, но в свою очередь способные произвести впечатление своими возможностями.

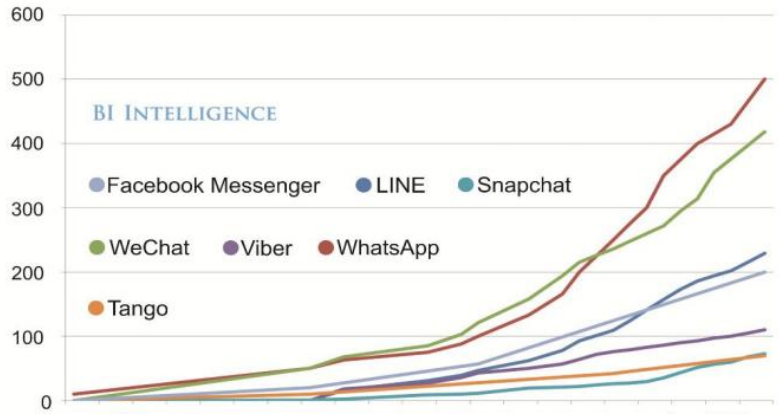

Рисунок 1.3- Количество активных пользователей за месяц, млн.
(Исследование BI Intelligence)

Мессенджеры – один из самых перспективных средств коммуникации в условиях доступного мобильного интернета, и в будущем ожидается только увеличение пользователей

Рассмотрим таблицу 1.1 где праведен краткий анализ популярных мессенджеров.

Таблица 1.1 – Анализ мессенджеров по шифрованию

Мессенджер	**End-to-end шифрование**	**Шифро-ванные груп чаты**	**Desktop клиент с шифро-ванием**	**Шифро-ванные голосовые звонки**	**Нужен сервер**	**Основной индификатор**
Telegram	Да, нужно отдельно создавать защищенный	До сервера	Cutegram	Нет	Да	Тел. номер, Nickname

	чат					
Signal	Да (mobile only)	Да (mobile only)	нет	Да	Да	Тел. номер
WhatsApp	Да (Android only)	Нет	HTTPS	?	Да	Тел. номер
Skype	Нет	Нет	До сервера	Нет	Да	Логин
Viber	Нет	Нет	До сервера	?	Да	Тел. номер

Таблица 1.2- Анализ мессенджеров по функциональности

Название	Кросс платформеность	Достоинства	Недостатки
Telegram	+	• Защита от несанкционированного чтения осуществляется благодаря применению протокола связи MTProto; • Наличие опции переписки в «секретном чате» — Secret Chat. Благодаря ней вы можете общаться при помощи зашифрованных сообщений; • Наличие таймера для автоуничтожения сообщений спустя заданное время (например, 1 час, а не через максимум всего 10 секунд, как в Viber); • Возможность передачи файлов большого размера (до 1 Гб); • Высочайшая скорость работы, в т.ч. доставки ваших файлов адресату. Нет задержек, характерных другим мессенджерам; • Синхронизация между пользователями.	• Не поддерживает в меню интерфейса русский язык. • отсутствие возможности голосовых звонков
WhatsApp	+	• Низкий расход батареи. • Простой интерфейс. • Синхронизация с контактами из адресной книги. • Поддержка групповых чатов. • Работы на таких платформах, как iOS, Android, Windows Phone, BlackBerry, Nokia Symbian и Nokia S40, у WhatsApp есть компьютерная	• Программа не сильна в вопросах шифрования • В Whats App нет стикеров, таких как в Telegram или Viber.

		версия клиента, а также web-версия, т.е. можно общаться в обычном браузере • Бесплатные аудио- и видеозвонки, которые можно делать посредством Интернет-соеденинения (3G или Wi-Fi) • отправка фотографии и видео, можно делиться PDF-файлами (книгами, журналами), слайд-шоу и другими документами. Единственное ограничение — размер файла не должен превышать 100 Мб. • Фото, аудио и видео-материалы вначале отправляются на •	

Продолжение таблицы №1.2

		• специальный HTTP-сервер, а потом уже передаются в уменьшенном варианте конечному получателю. Это позволяет экономить интернет-трафик. • Программа с 2016-го года полностью бесплатна. В данный момент у приложения нет никаких встроенных покупок.	
Skype	+	• Возможность общаться посредством видеозвонка. • Нет привязки к какому-то определенному устройству. • Программа может устанавливаться везде: от смартфона до «умного» телевизора. • Возможность совершать звонки на стационарные телефоны.	• Большой объем файла • Необходим хороший сигнал интернета
Viber	+	• Бесплатные звонки и сообщения - Отсутствие рекламы - Стабильная связь даже на 2G - Работает в фоне и “ест” очень мало оперативной памяти - Обилие смайликов, в том числе анимированные - Голосовые сообщения Недостатки Viber:	• Отсутствие видеозвонков • На Symbian доступен лишь обмен текстовыми сообщениями • Без интернета Viber бесполезен

Лидером по итоговой оценке сегодня стал Viber. Это приложение является наиболее универсальным представителем своего сегмента – чаты, аудио- и видеосвязь, конференции. Тут есть даже звонки на обычные телефоны. Чуть позади расположился WhatsApp. Он более популярен, но не позволяет делать вызовы на мобильные номера.

Третье место делят Skype и Telegram. Они также весьма популярны, но имеют определенные недостатки Skype больше ориентирован на стационарные ПК, а в Telegram нет голосовой связи.

Если подробно рассматривать таблицу по функциональным особенностям, то можно убедиться, что практически все мессенджеры имеют версии для нескольких платформ. Аудиосвязь есть во всех приложениях кроме Telegram, видеовызовы отсутствуют еще и в WhatsApp.

Звонки на обычные телефоны поддерживают лишь 2 приложения – Viber, Skype. Основным идентификатором пользователя практически везде служит номер телефона, хотя в Hangouts, Skype и Facebook Messenger применяются Google Аккаунт, e-mail или учетная запись Facebook соответственно. Для обычных вызовов и переписок хорошо подойдут WhatsApp или Viber, для звонков на стационарные номера – Skype, если нужны только чаты, то можно остановиться на Telegram.

1.4 Постановка задачи на разработку проекта

Проектируемая информационная система должна обеспечивать пользователям возможность обмена сообщениями с шифрованием данных в пределах корпорации. Организация данной сети как Web-портал, в значительной мере, обеспечит популярность и адаптированность ресурса с к различным устройствам (рис.1.4)

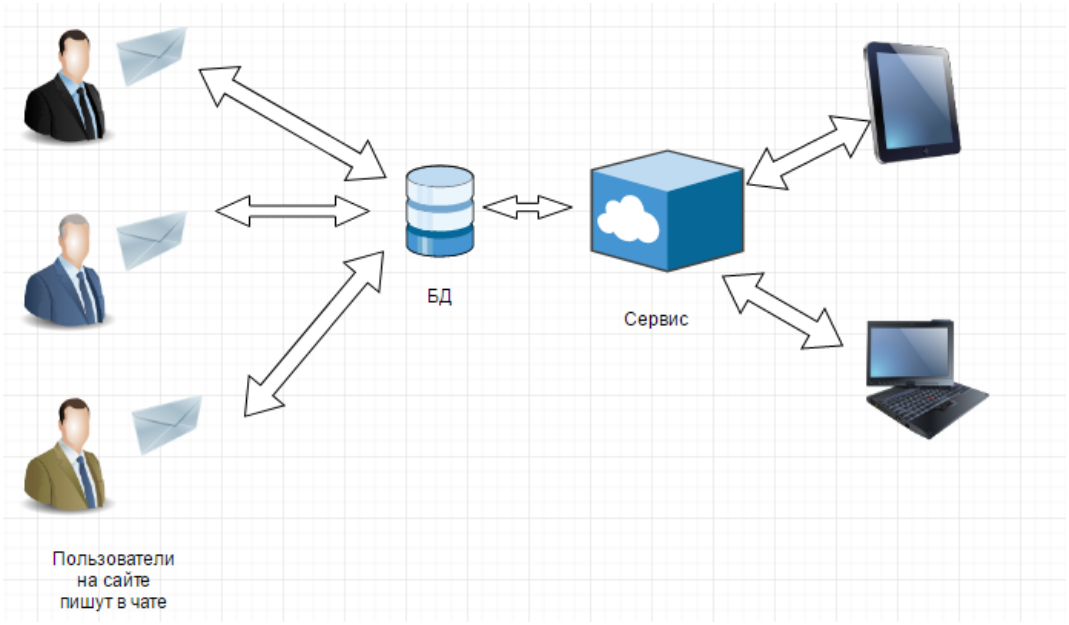

Рисунок 1.4 – Обмен сообщением с помощью web-сервера

Рассмотрим общие задачи, возлагаемые на систему:

1. Организация обмена сообщениями с шифрованием на Web-портала.
2. Контролирование информационных потоков в системе администратором портала.
3. Обеспечение различных методов поиска информации.

Основными объектами системы являются:

1. Пользователь.
2. Администратор.
3. Организация.

Объект «Пользователь системы»

Пользователь системы является главным объектом в Web-портале. Наличие пользователей и связей между ними, формируют взаимодействие с помощью обмена сообщениями, образуя некую «социальную сеть».

Объект «Организация»

Объект «Организация» предназначен для обеспечения и поддержки обменна сообщениями. Рассмотрим возможности и механизмы, обеспечивающие реализацию данной задачи:

1. Просмотр всех записей организации.
1. Удаление данных организации.
2. Просмотр всех услуг организации.
3. Просмотр информации о конкретной услуги организации.
4. Добавление информации о новой услуги организации.
5. Редактирование данных услуги организации.
6. Удаление данных услуги организации.

Перейдем к рассмотрению общей характеристики организации решения задачи на ЭВМ.

1.5 Общая характеристика организации решения задачи на ЭВМ

В предыдущем пункте бакалаврской работы, были рассмотрены основные задачи проектируемой системы. Теперь разработаем схему потоков данных, описывающую процесс преобразования информации от ее ввода в систему до выдачи потребителю.

Контекстная диаграмма организации системы по обмену сообщений, направленной на поддержку обмена сообщениями представлена на рис. 1.4.

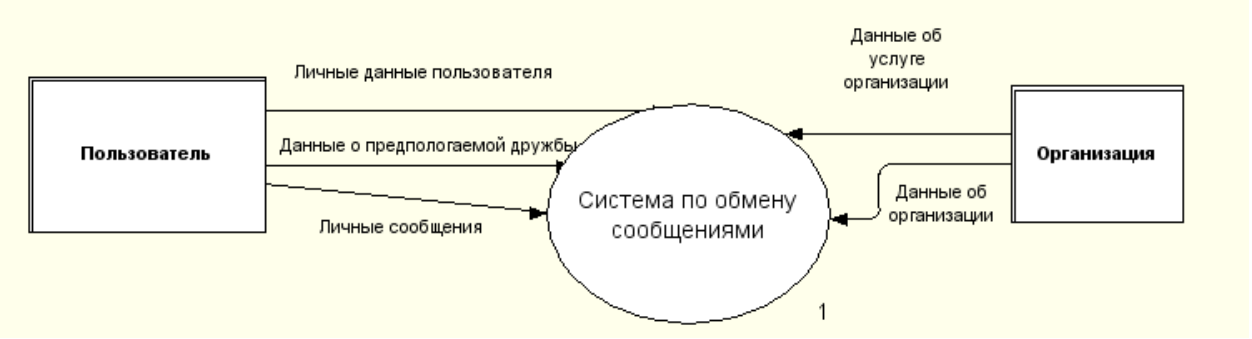

Рисунок 1.5- Контекстная диаграмма организации обмена сообщений

Внешними сущностями в данной диаграмме являются:

1. Пользователь – пользователь системы.
2. Организация – компания, представляющая и поддерживающая обмен сообщениями.

Стрелками изображены потоки данных, передаваемые от сущностей к единой системе, которая в дальнейшем будет представлены в виде ряда подсистем. Декомпозиция первого уровня процесса «Организация обмена сообщениями» спроектирована и представлена на рис. 1.5.

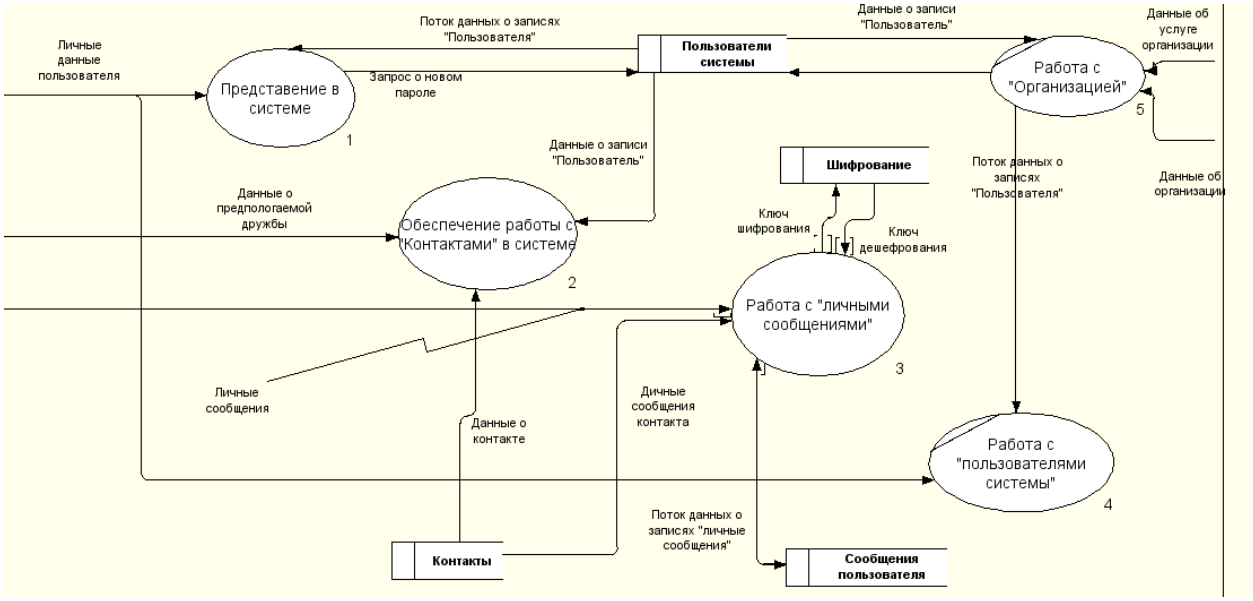

Рисунок 1.6 - Декомпозиция первого уровня системы

На контекстной диаграмме уровне представлены основные процессы обмена сообщениями. Рассмотрим декомпозицию второго уровня процесса «Представление в системе», изображенную на рис. 1.7.

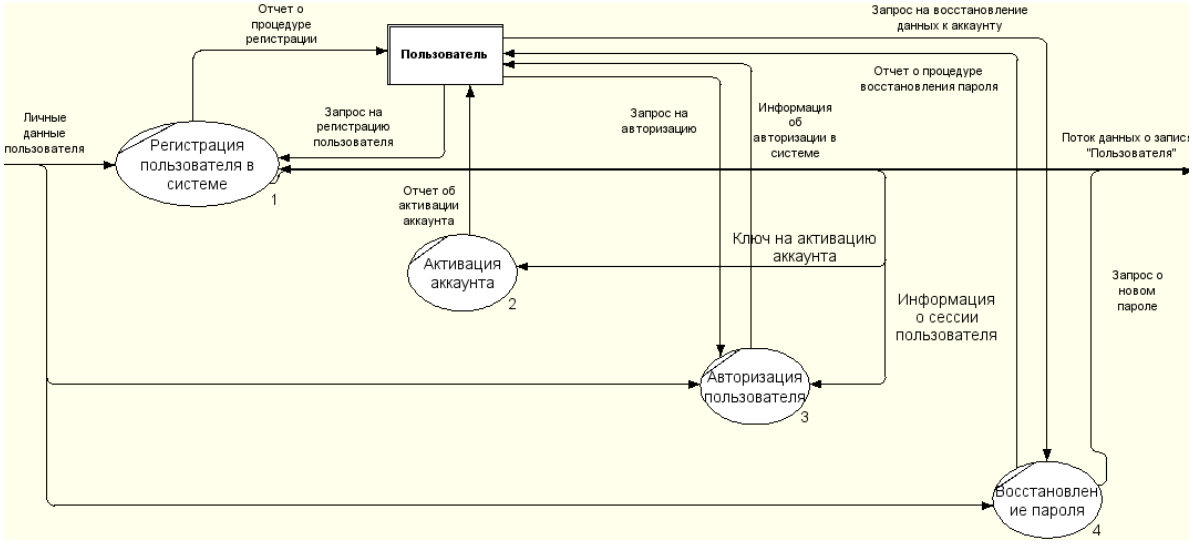

Рисунок 1.7 -Декомпозиция второго уровня процесса «Представление в системе»

Рассмотрим процесс «Обеспечение работы с «Контактами» в системе». Декомпозиция процесса изображена на рис. 1.8.

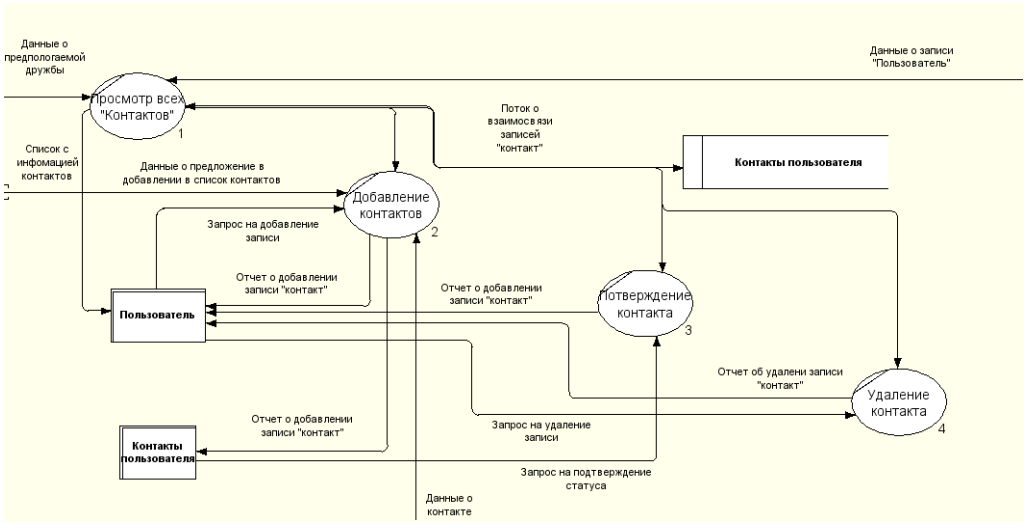

Рисунок 1.8 - Декомпозиция второго уровня процесса «Обеспечение работы с «Контактами» в системе»

Процессы подсистемы «Работа с личными сообщениями» представлены на рис. 1.9.

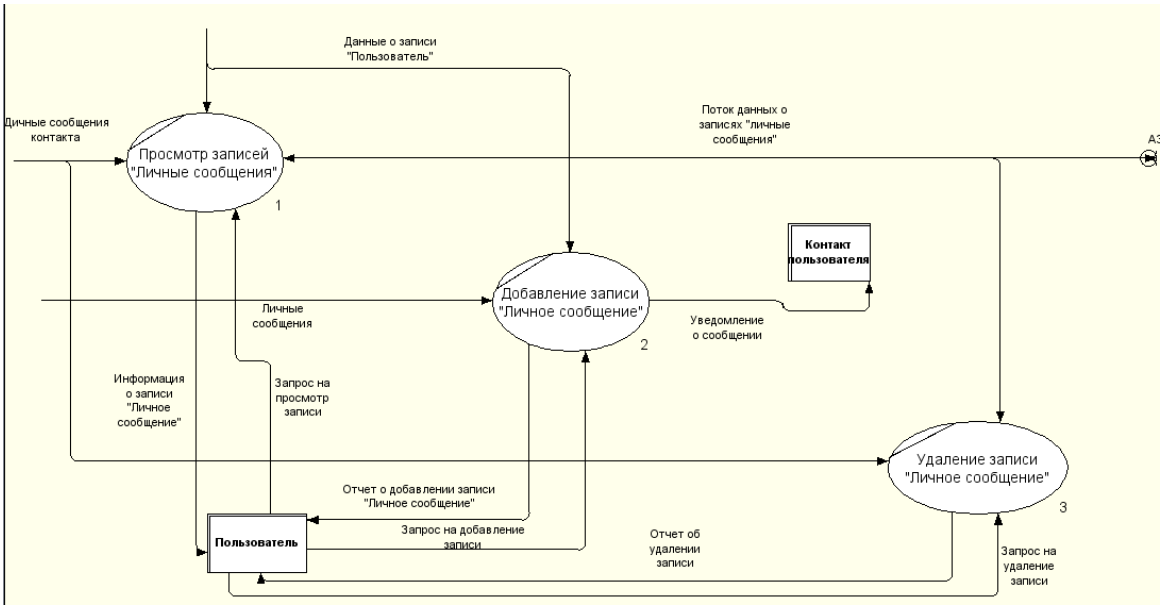

Рисунок 1.9- Декомпозиция второго уровня «Работа с «Личными сообщениями»»

Диаграмма поясняет как происходит происходит процесс обмена личными сообщениями между сущностями «Пользователь» и «Контакт пользователя».

Рассмотрим декомпозицию процесса «Работа с пользователями в системе», представленную на рис. 1.10.

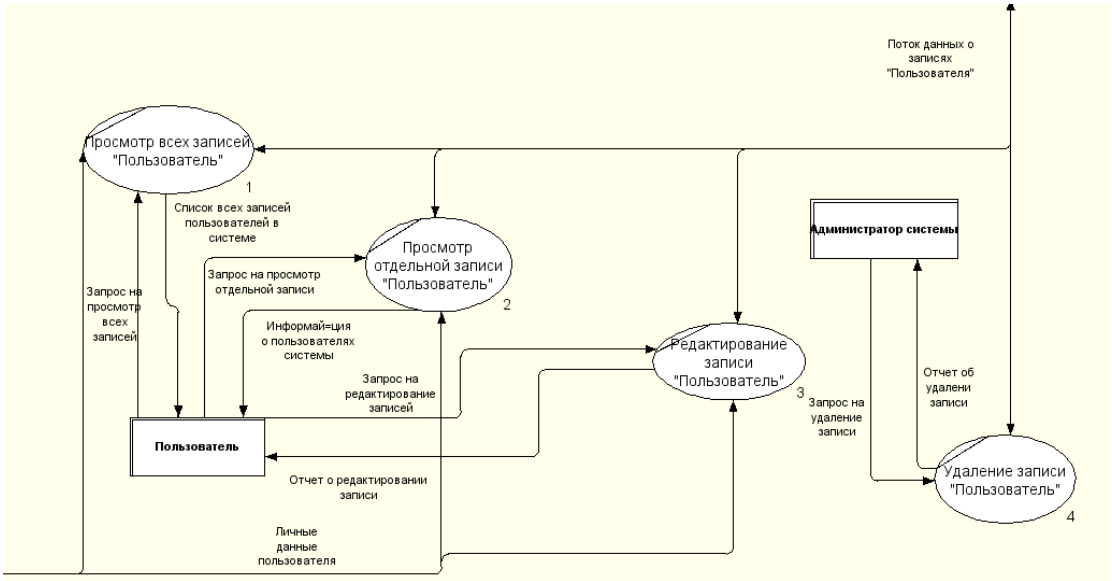

Рисунок 1.10- Декомпозиция второго уровня процесса «Работа с пользователями в системе»

Пользователь может совершать все действия, за исключением удаления аккаунта, что доступно только администратору.

Выводы по главе 1

В первой главе работы проведен анализ предметной области, спроектированы модели программного продукта и описаны предъявляемые требования. В результате обзора аналогов были выявлены их достоинства и недостатки. В конце главы были описаны и программные средства разработки.

Глава 2 Логическое проектирование системы по обмену сообщения

2.1 Выбор технологии логического моделирования

База данных – это модель предметной области. Процесс проектирования базы данных включает разработку концептуальной медли, залее логическтой и, наконец, физической модели БД.

Схема инфологической модели данных представлена на рис. 2.1.

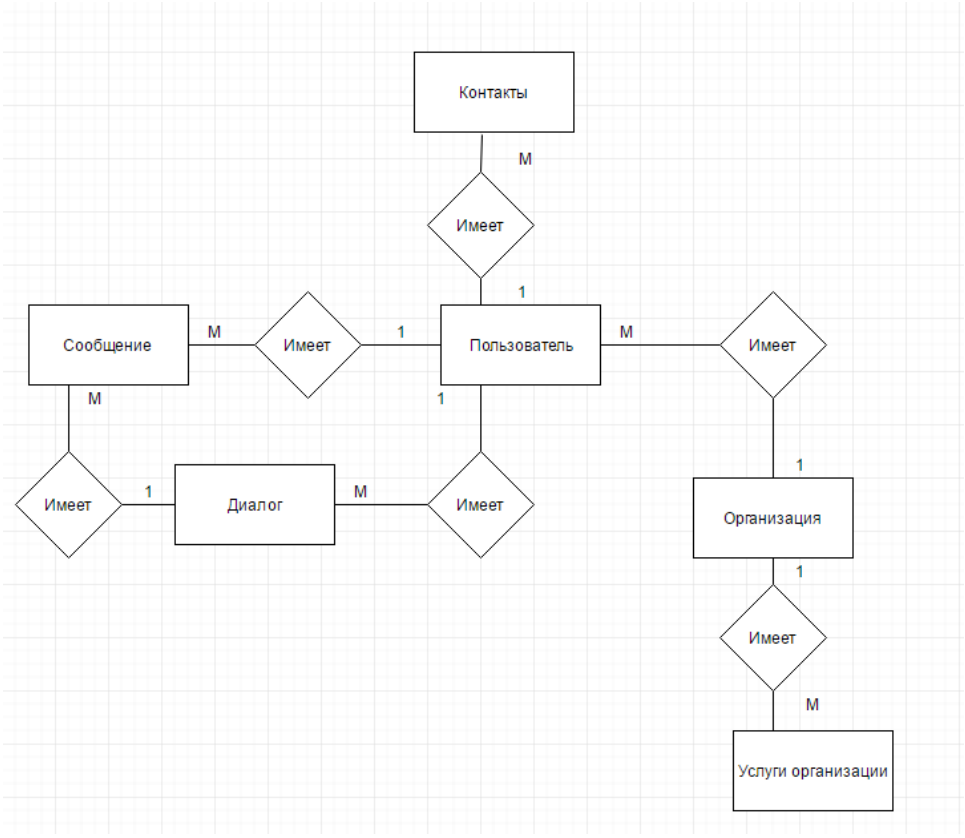

Рисунок 2.1- Инфологическая модель данных

2.2 Логическая модель данных

Этап логического проектирования включает в себя создание схемы базы данных на основе конкретной модели данных.

Реляционная логическая модель получается в результате нормализации отношений, выделенных в концептуальной модели.

Наиболее широко используемым средством разработки логических моделей баз данных являются диаграммы «сущность-связь» – Entity-Relationship (ER-

диаграммы). Диаграмма даталогической модели данных, созданная при помощи свободно распространяемого программного обеспечения «DBDesigner 4», представлена на рис. 2.2.

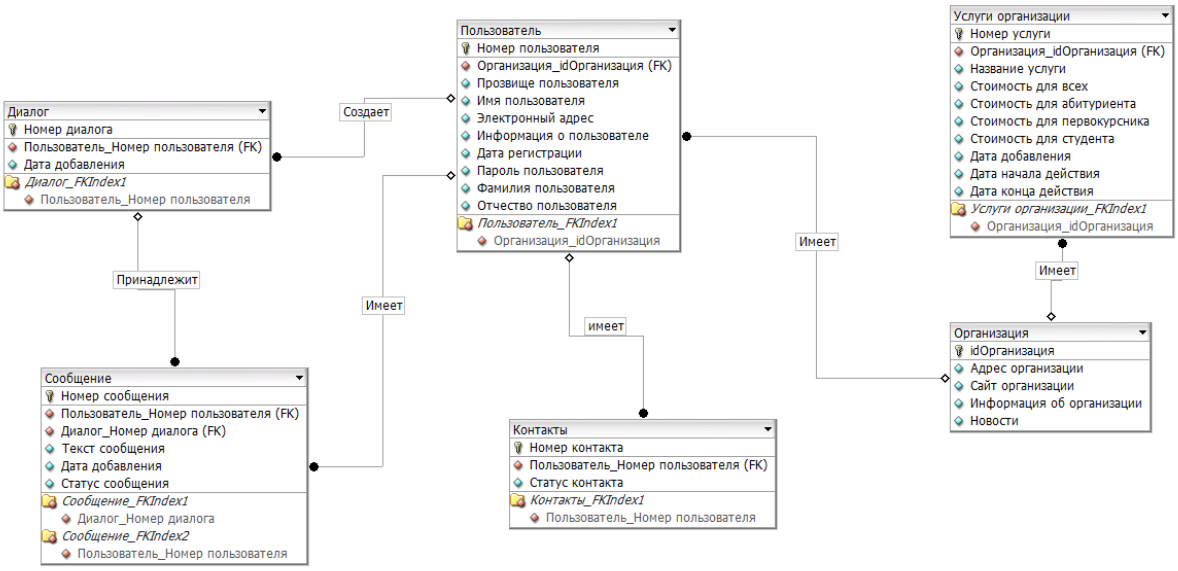

Рисунок 2.2 - Даталогическая модель данных

Очередным этапом проектирования БД является выбор конкретной СУБД.

В виду большой популярности среди разработчиков Web приложений и большого числа успешно завершенных проектов в качестве СУБД в работе будет использовать MySQL.

Физическая модель данных представлена на рис. 2.3. При разработке схемы, была использована программа «DBDesigner 4.

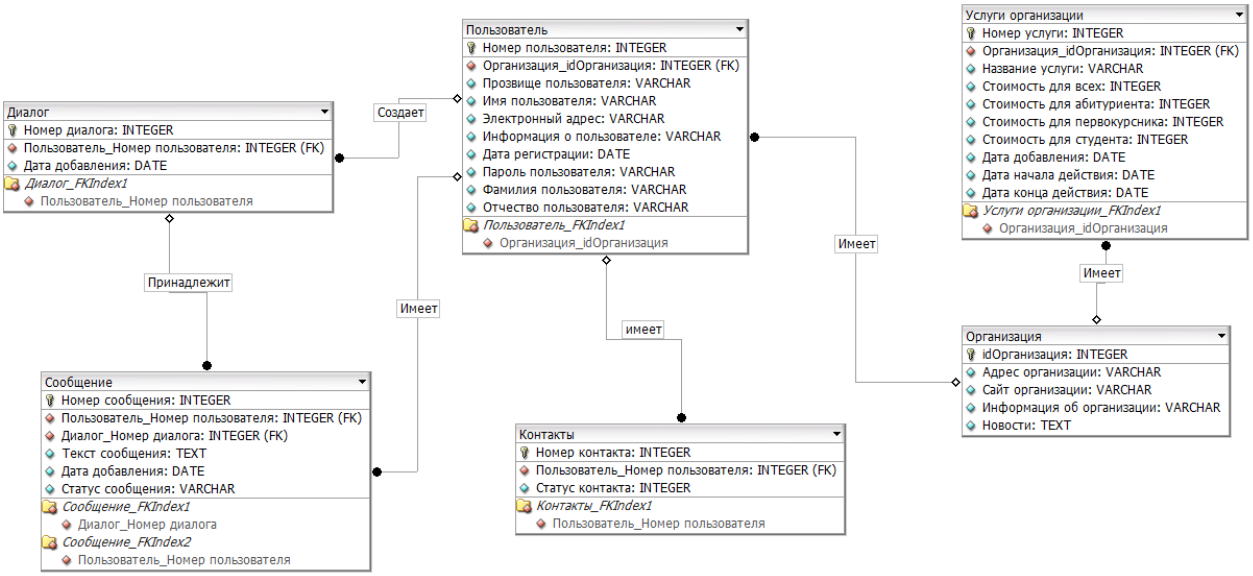

Рисунок 2.3- Физическая модель данных

2.3 Входы и выходы системы

К основным источникам входной информации можно отнести.

1. Данные пользователей системы.
2. Данные о контактах пользователя.
3. Личные сообщения пользователей.
4. Информация об организациях и их услугах.

Ввод информации в базу данных осуществляется посредством заполнения полей форм.

На рис. 2.4 представлена форма ввода данных об услуге организации.

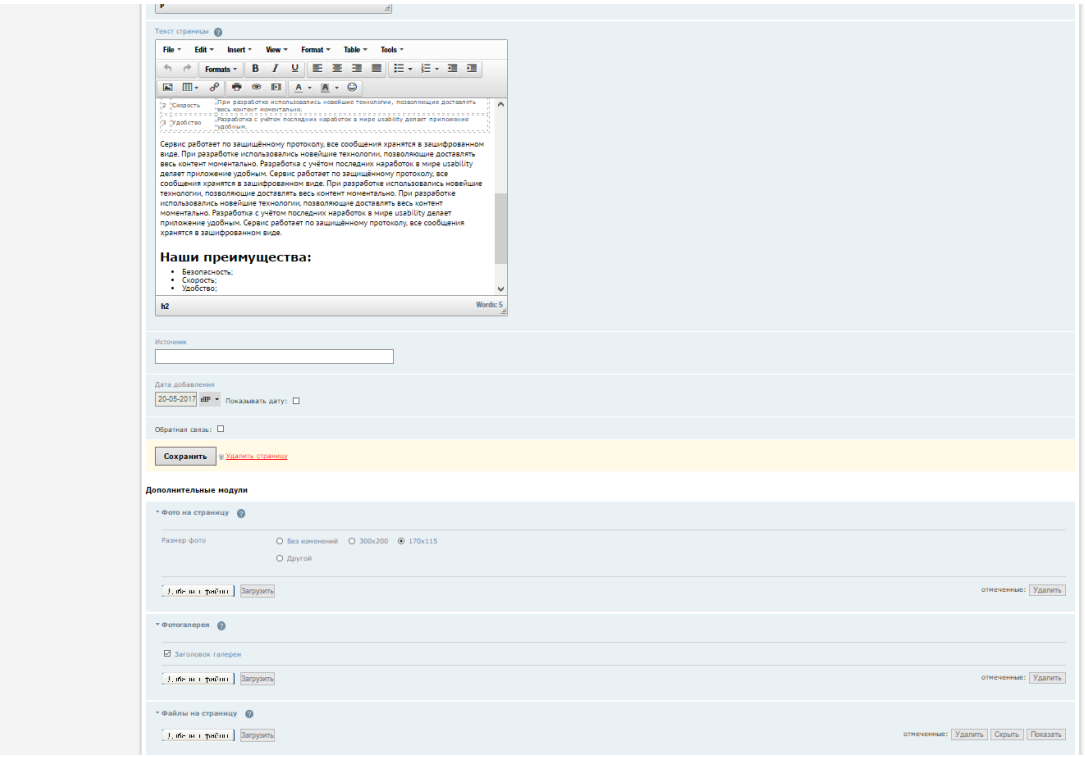

Рисунок 2.4 - Добавление информации в базу данных на примере формы добавления информации

Выходными данными системы являются:

1. Код в формате HTML.
2. Данные в XML – используются в RSS ленте.
3. Мультимедийные данные.

Опишем вывод информации, используя в качестве примера добавленную пользователем услугу (рис. 2.5).

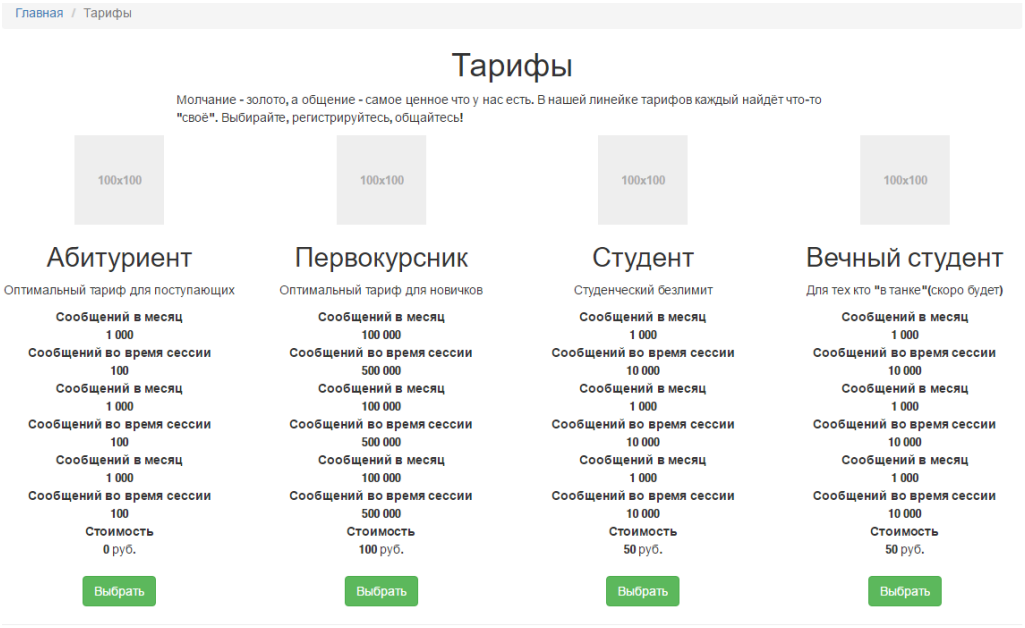

Рисунок 2.5 - Формирование данных для вывода информации

Более подробную информацию о формировании выходных данных можно узнать в соответствующем разделе разработки модулей системы.

Выводы по главе 2

В главе была спроектирована концептуальная модель, на которой были показаны сущности системы и взаимодействие между ними. На основе концептуальной модели были спроектированы логическая и физическая модели. В этих моделях показано описание объектов предметной области, их атрибутов и взаимосвязей между ними в том объеме, в котором они подлежат непосредственному хранению в базе данных системы.

Глава 3 Физическое проектирование системы обмена сообщениями с шифрованием

3.1 Выбор архитектуры и технологии разработки программного обеспечения

Для эффективного решения поставленной задачи необходимо определится с комплексом технических и аппаратных средств.

Разрабатываемая информационная система будет иметь трехуровневую архитектуру «клиент-сервер».

В сети Интернет информационные системы строятся в виде трехуровневой системы, представленной на рис. 3.1. В качестве клиента в большинстве случаев выступает Web-браузер.

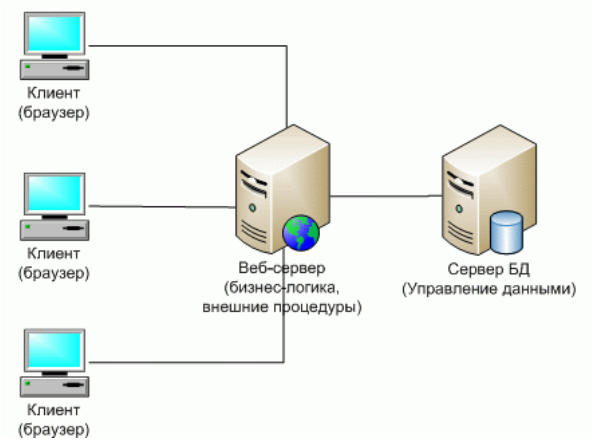

Рисунок 3.1 - Трехзвенная архитектура информационной системы

В соответствии с выбранной архитектурой необходимо определиться с программными средствами для серверной и для клиентской части Web-портала.

В таблице 3.1 представлены программные продукты для «хостинга», на котором будет расположен Web-портал.

Таблица 3.1 -Программное обеспечение серверной части

Категория	Название программного продукта
Операционная система	Семейство Unix
Web-сервер	Apache 2.0
СУБД	MySQL 5
Интерпретатор	PHP 5.2.6
FTP-сервер	VSFTPD
Управление «хостингом»	CPanel

В качестве операционной системы выбрана UNIX-подобная операционная система.

В качестве Web-сервера был выбран Apache

В качестве языка программирования был выбран язык «PHP» - популярный и свободно распространяемый язык Web-программирования.

В таблице 3.2 представлено программное обеспечение для клиентской части.

Таблица 3.2-Программное обеспечение клиентской части

Категория	**Название программного продукта**
Операционная система	Windows, Linux, MacOS
Браузер	Firefox, Opera, Internet Explorer, Google Chrome
Проигрыватель	Adobe Flash Player

Для того, чтобы клиент смог посетить Web-портал ему необходимо наличие операционной системы, браузера. В качестве операционной системы можно выбрать либо свободно распространяемую ОС, либо коммерческую. По своим техническим характеристикам, современные операционные системы на базе Linux с поддержкой графической оболочки мало чем уступают коммерческим. Их применение будет достаточным для просмотра интересных мест в сети Интернет.

Подводя итог, можно сказать, что для разработки Web-портала, можно использовать не только коммерческое, но и свободно распространяемое программное обеспечение. Эти требования могут относиться к серверной и клиентской частям.

3.2 Разработка программного обеспечения

3.2.1 Защита системы и личных данных пользователей от злоумышленников

При разработке рассматриваемого программного обеспечения необходимо обеспечить определенную степень защиты системы. С этой целью необходимо предусмотреть выполнение следующих условий:

1. Разграничить уровни доступа к информации.
2. Защитить базу данных от SQL-инъекций.
3. Обеспечить проверку всех входных и противоречивых данных.
4. Защитить файловую структуру.
5. Выполнить шифрование данных.
6. Разработать систему защиты от «спам-роботов».
7. Обеспечить административный мониторинг сервера

Рассмотрим выполнение указанных условий более подробно.

3.2.2 Проектирование уровня доступа к информации

При проектировании системы были определены основные роли пользователей: Гость, Пользователь, Администратор. Приведем перечень действий, совершаемых указанными пользователями.

Гость может:

- просматривать страницы;
- осуществлять регистрацию в системе;
- просматривать блоги;
- просматривать RSS-ленты.
- производить авторизацию в системе.

Пользователь системы может делать все операции Гостя, а также:

- изменять личные данные;
- просматривать список контактов;
- отправлять личные сообщения;
- читать личные сообщения;
- искать контакты;
- удалять переписку;
- удалять контакты.

Администратор может выполнять все действия Пользователя, а также:

- добавлять, изменять и удалять категории;
- изменять данные других пользователей;
- удалять пользователей;
- редактировать услуги;
- проводить мониторинг действий в системе.

3.2.3 Защита от SQL-инъекций

Чтобы защитить систему от от SQL-инъекций можно добавить кавычки в тело запроса. Для это разработана функция, код которой представлен на рис. 3.2.

```
#Функция добавляет кавычки
function q($value)
{
        if (!is_numeric($value))
                $value = "'".mysql_real_escape_string($value)."'";
        return($value);
}
#Функция снимает экранирование
if (function_exists('get_magic_quotes_gpc') && get_magic_quotes_gpc())
{
function stripslashes_deep($value)
{
        if(is_array($value))
        {
                $value = array_map('stripslashes_deep', $value);
        }
        elseif (!empty($value) && is_string($value))
        {
                $value = stripslashes(trim($value));
return($value);
}
$_POST = stripslashes_deep($_POST);
$_GET = stripslashes_deep($_GET);
$_COOKIE = stripslashes_deep($_COOKIE);
}
#Запрос на удаление услуг организации
$query = "SELECT service_id FROM service WHERE company_id = ".q($company_id);
```

Рисунок 3.2 – Код функции защиты от SQL-инъекций

3.2.4 Алгоритмы проверки входной информации

Корректность вводимой информации достигается использованием ряда техник: ввод масок, использование регулярных выражений. На стороне сервера желательно использовать регулярные выражения. Например, регулярное выражение для проверки адреса электронной посты представлено на рис. 3.3.

```
#Проверка Email адреса
If (!empty($company_email) &&
   !preg_match("/^(?:[a-z0-9]+(?:[-_\.]?[a-z0-9]+)?
    @[a-z0-9]+(?:\.?[a-z0-9]+)?\.[a-z]{2,5})$/i", $company_email))
$msg[] = "Неправильно введен E-mail!";
```

Рисунок 3.3 – Регулярное выражение проверки адреса электронной почты

3.2.5 Шифрование данных

Для шифрования личных данных, включая личные пароли, иожно применять различные схемы защиты. Одним из алгоритмов, хорошо зарекомендовавших себя на практике является алгоритм MD5. Данный алгоритм использует 128-битный алгоритм хеширования данных. Например, обработка пароля производится с помощью следующего фрагмента:

$password = md5($password);

3.3. Взаимосвязь модулей программного средства

Как описывалось выше, разрабатываемая система состоит из набора взаимосвязанных модулей. Взаимосвязь модулей представлена на рис. 3.4.

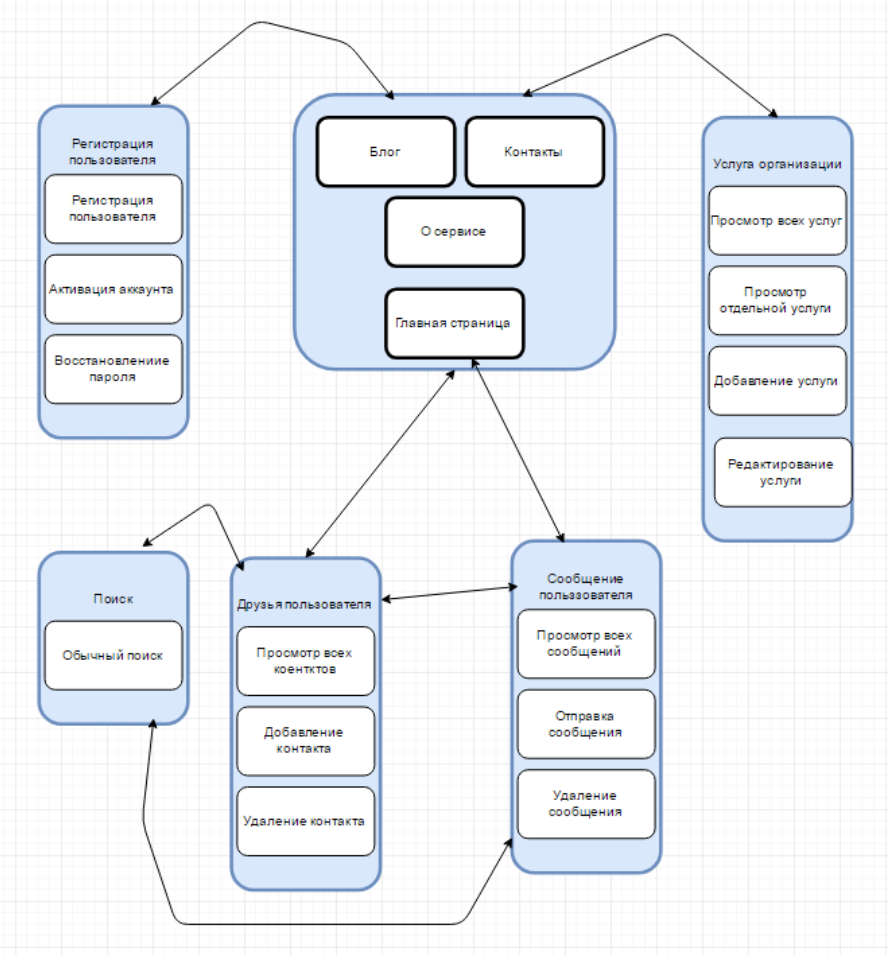

Рисунок 3.4 - Функциональная схема модулей системы

На схеме представлены модули системы и выполняемые ими функции.

3.4 Описание модулей приложения

3.4.1 Модуль «Главная страница»

Центральным модулем программного продукта является главная страница.. Главная страница состоит из следующих частей:

1. Верхняя часть портала.

1.1. Навигация.

1.2. Поля авторизации.

2. Основная часть портала.

2.1. Вверхняя часть.

2.1.1. Переход на оптсание о сервисе.

2.2. Центральная часть.

2.2.1. Информационная часть.

2.2.2. Актуальные услуги.

2.2.3. Последние добавленные комментарии.

3. Нижняя часть портала.

3.1. Счетчик посещение.

3.2. Дополнительная навигация.

3.3. Копирайт.

Изображение главной страницы представлено на рис. 3.5.

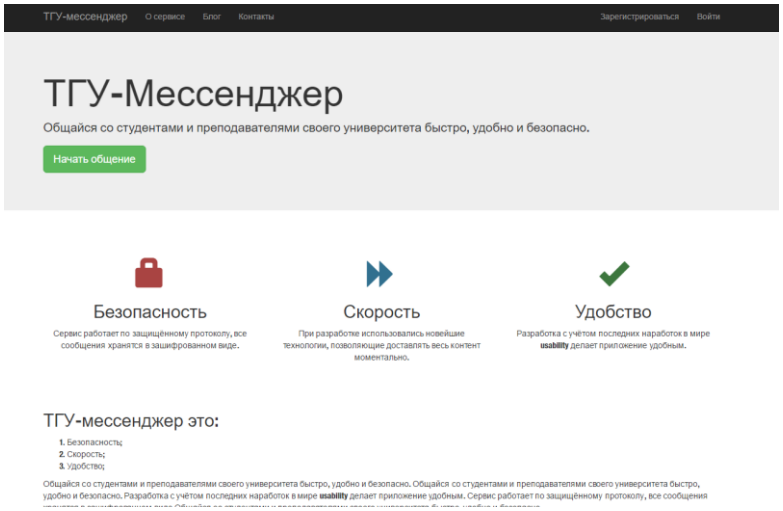

Рисунок 3.5 - Изображение главной страницы системы

Все элементы главной страницы основаны на общем шаблоне, который специально разработан для лучшего виртуального общения.

3.5.2. Модуль «Регистрация»

Для того чтобы пользователь смог общаться в сети, просматривать информацию, ему необходимо зарегистрироваться в системе. Для этого он должен перейти по ссылке «Регистрация» и заполнить единственное поле – электронный адрес (рис.3.6). После этого система автоматически генерирует пароль, которому можно зайти в систему.

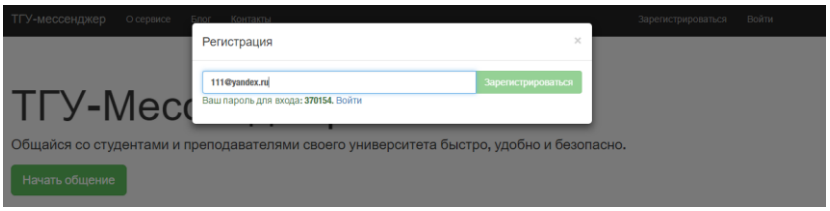

Рисунок 3.6- Регистрация пользователя

После авторизации в системе пользователь может внести изменение о пароле, имени, фамилии, отчестве, Нику и установить шифрование (рис.3.7)

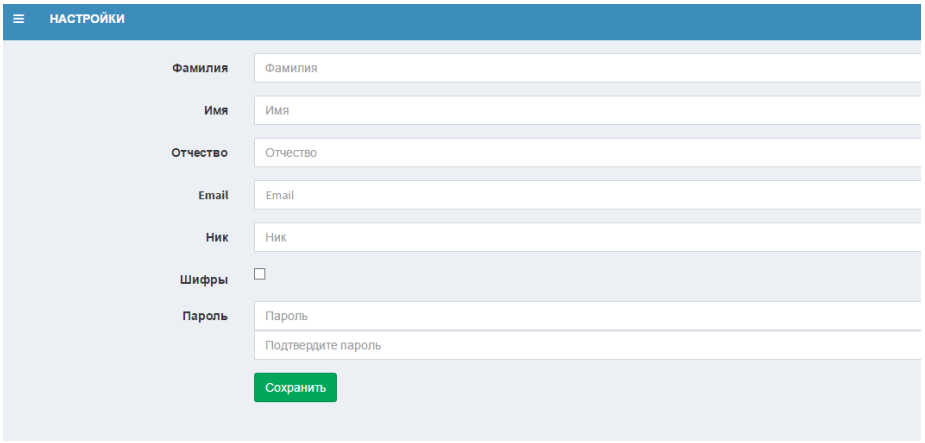

Рисунок 3.7- Регистрация пользователя

Блок-схема процесса регистрации пользователя представлена на рисуноке 3.8.

С целью предотвращения некорректного заполнения полей формы приводятся примеры заполнения. Для проверки подлинности электронного ящика реализован следующий механизм: после регистрации пользователю высылается электронное письмо, содержащее ссылку на страницу активации. После нажатия на ссылку пользователь активирует свой аккаунт и может пользоваться системой.

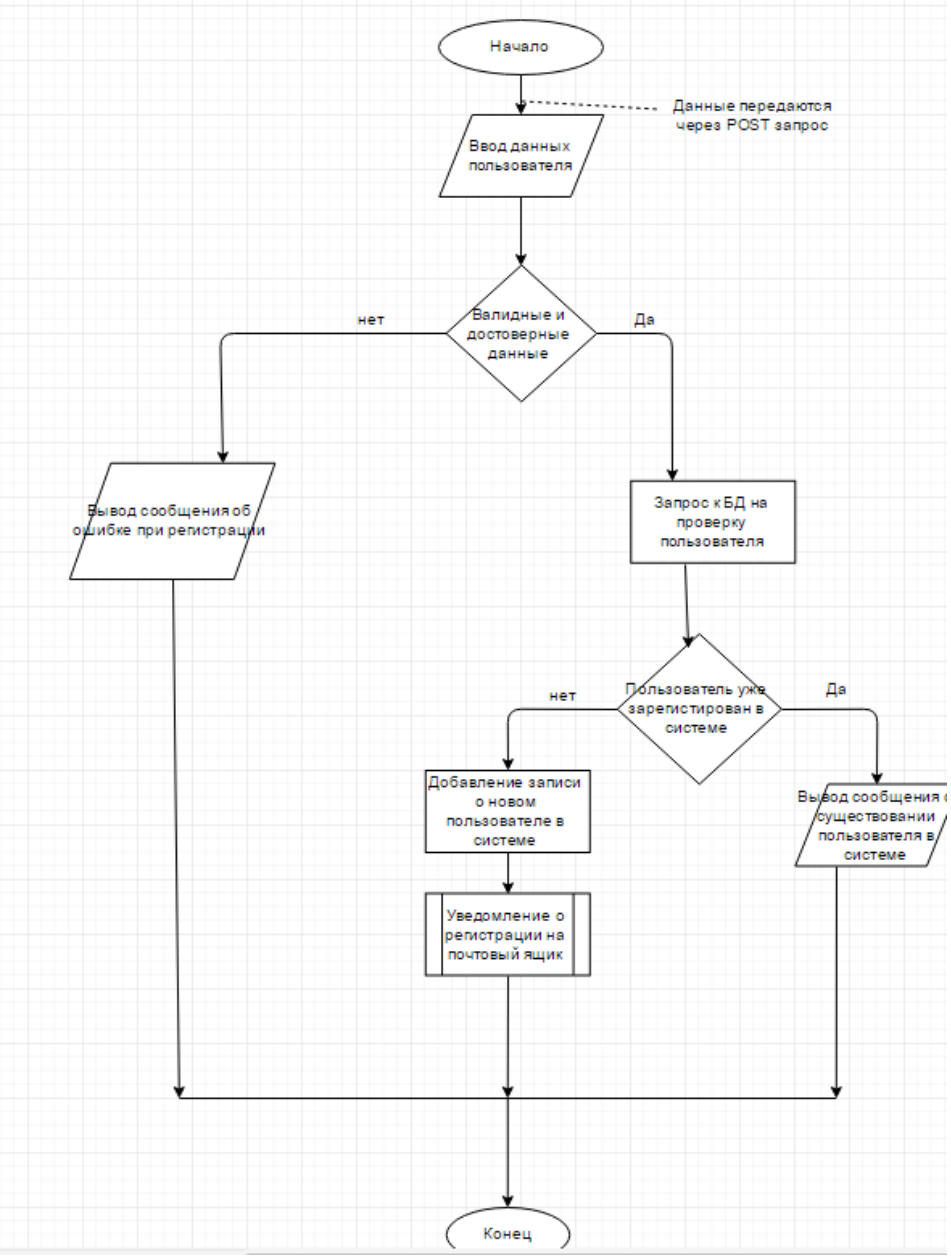

Рисунок 3.8 - Блок-схема процесса «Регистрация пользователя»

3.5.3. Модуль «Пользователь системы»

Зарегистрировавшись в системе, пользователь должен необходимо иметь возможность просматривать личную страницу и редактировать соответсвующие данные.

На модуль «Пользователь» возложено выполнение следующих функций:

1. Просмотр записей.
2. Просмотр личной страницы.

3. Редактирование данных.

4. Удаление пользователя.

Просмотреть пользователей можно после авторизации на портале. Рядом с каждым автором присутствует его логин в системе и выборочные личные данные.

После прохождения процедуры авторизации, пользователь может зайти на личную страницу. На личной странице находятся следующие элементы:

1. Личные данные.

2. Навигационное меню.

2.1. «Мои контакты».

2.2. «Мои сообщения».

2.3. «Настройки».

Изображение личной страницы администратора портала изображено на рис. 3.9.

На представленном изображении видно, что сначала отображается логин пользователя в системе. Если пользователь добавлял компании, то они отображаются в отсортированном виде в соответствующем блоке.

Каждому пользователю предоставляется возможность редактирования личных данных. Изменять можно следующие поля: «Имя», «E-mail», «Пароль»,. Необязательными полями для изменения являются: «Пароль». Если поле «Пароль» оставить пустым, то данные останутся прежними.

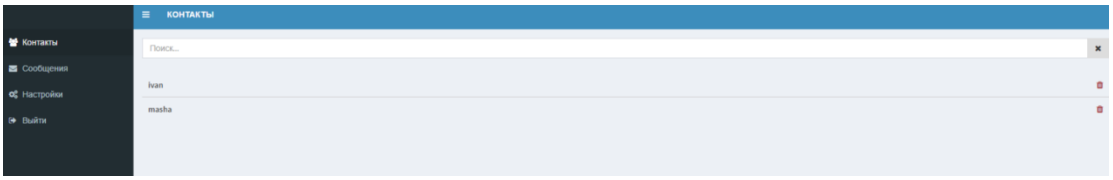

Рисунок 3.9 - Изображение личной страницы пользователя

Изображение формы редактирования данных пользователя изображено на рис. 3.10.

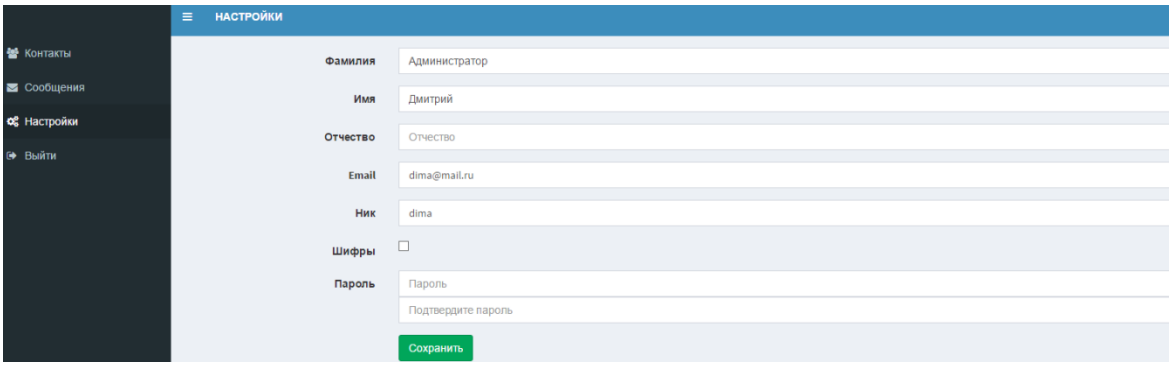

Рисунок 3.10 - Форма редактирования данных пользователя

Администратору Web-портала представляется возможность редактирования и удаления любых пользователей системы. При удалении автора учитывается связь с другими объектами, которые в дальнейшем будут уничтожены.

Рассмотрев основные функции модуля «Пользователь системы», перейдем к рассмотрению модуля «Сообщения пользователей», который позволяет обмениваться личными сообщениями внутри портала.

3.5.4. Модуль «Сообщение пользователя»

Каждый пользователь системы должен иметь возможность обмениваться личными сообщениями с другими путешественниками. Так как система ориентирована на построение «социальной сети», то данная функция является необходимым инструментом коммуникации.

Рассмотрим основные функции модуля:

1. Просмотр всех сообщений пользователя.
2. Отправка сообщения.
3. Удаление сообщения.

У каждого пользователя в личной странице есть ссылка «Сообщения». Перейдя по этой ссылке, получим список всех сообщений пользователя. Так как длина сообщения имеет фиксированный размер, то все сообщения представлены в развернутом виде. Рядом с каждым сообщением расположена ссылка на удаление.

Удалять сообщения может только сам пользователь, которому оно было

направлено. Администратор портала не обеспечивает контроль сообщений, так как каждый человек имеет право на личную переписку.

Изображение со списком сообщений представлено на рис. 3.11.

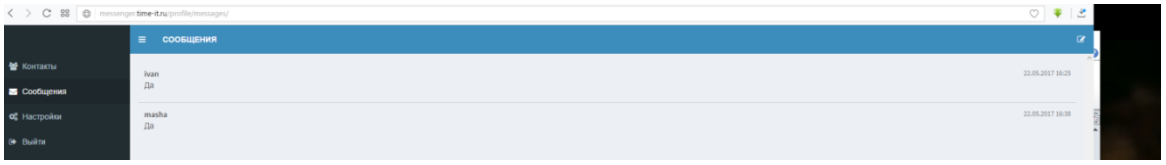

Рисунок 3.11 - Список сообщений пользователя

На «скриншоте» видно, что рядом с каждым сообщением расположено имя пользователя. Правее каждого сообщения находится ссылка для удаления сообщения.

Чтобы отправить сообщение другому пользователю, необходимо посетить его личную страницу и нажать на ссылку «Отправить сообщение». После этого, поверх всех элементов, откроется форма, в которой можно будет написать текстовое сообщение. На рис 3.12 изображена форма отправки личного сообщения.

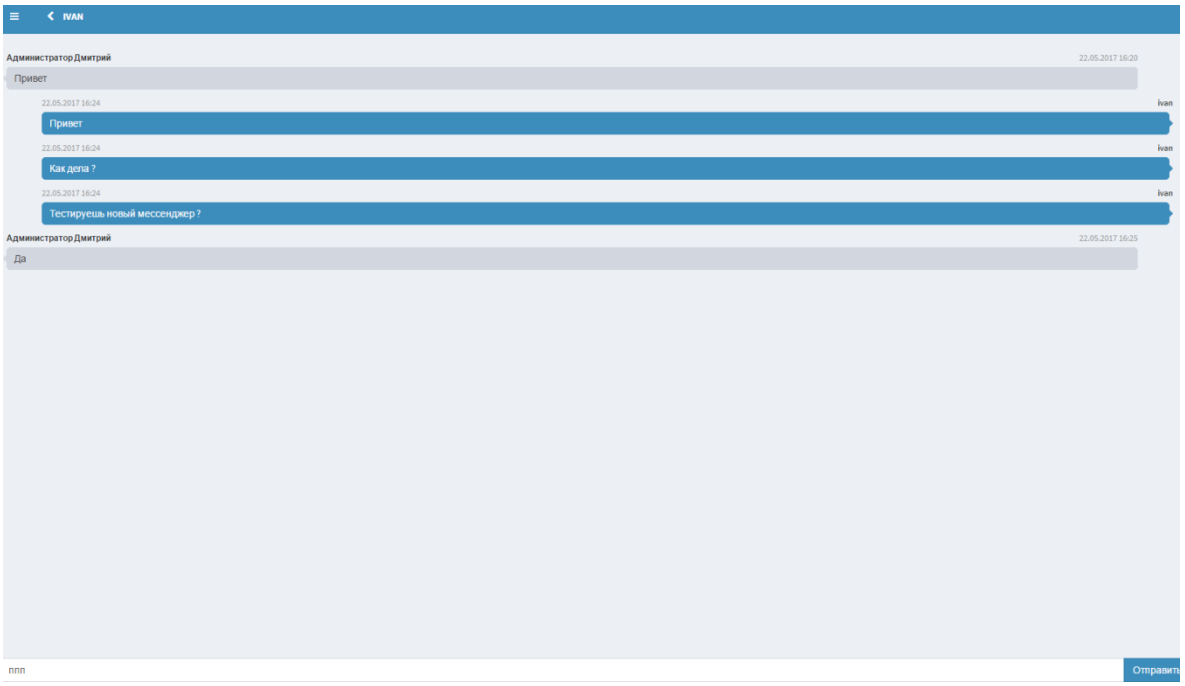

Рисунок 3.12 - Форма отправки личного сообщения

Плавное появление формы осуществляется за счет подключения библиотеки «Light Box 2», которая позволяет быстро и эффективно работать с подобными фреймами. Библиотека разработана при помощи скриптового языка программирования «JavaScript».

Все сообщения передаваемые от пользователей шифруются по алгоритму, представленному на рис.3.13.

Сообщения шифруются до отправки на сервер, хранятся на сервере в зашифрованном виде, приходят к клиенту в зашифрованном виде, перед показом пользователю расшифровываются (Приложение Б).

Чтобы быстро найти пользователя в системе для переписки, необходимо создать «френд-лист», в котором будут находиться контакты пользователя. Для реализации этих возможностей реализован модуль «Контакты».

```
msgEncode = function(message,userid){
alphax = getAlphaX()
result_message = ''
userid = userid.toString()
key = 0
for(i=0;i<userid.length;i++) key = key + parseInt(userid[i])
for(i=0;i<message.length;i++) {
pos = alphax.indexOf(message[i])
if(pos == -1){
result_message = result_message + message[i]
} else {
pos = (pos+key)%alphax.length
result_message = result_message + alphax[pos]
}
}
return result_message
}

msgDecode = function(message,userid){
alphax = getAlphaX()
result_message = ''
userid = userid.toString()
key = 0
for(i=0;i<userid.length;i++) key = key + parseInt(userid[i])
for(i=0;i<message.length;i++) {
pos = alphax.indexOf(message[i])
if(pos == -1){
result_message = result_message + message[i]
} else {
pos = pos-key
if(pos<0) pos = alphax.length+pos
result_message = result_message + alphax[pos]
}
}
return result_message
}
```

Рисунок 3.13 – Алгоритм шифрования

Рисунок 3.14 - Алгоритм шифрования

3.4.5 Модуль «Контакты»

Часто общаясь в системе с одними и теми же пользователями, появляется необходимость создать свой список друзей, в котором можно будет быстро и легко найти требуемого собеседника. Для осуществления данной возможности предназначен модуль «Контакты», который выполняет следующие задачи:

1. Добавление контакта.
2. Подтверждение контакта.
3. Удаление контакта.
4. Просмотр списка контакта.

Просмотреть весь список своих контактов можно в личном кабинета, кликнув по ссылке «Контакты». Объекты располагаются в виде ячеек таблицы. В каждой строке находится не более трех друзей. На рис. 3.15 можно увидеть список друзей администратора портала.

Рисунок 3.15 - Изображение страницы с контактами

Друзья отображаются в виде блоков, в которых находятся имя контакта. Рядом с каждым объектом располагается ссылка на «Удаления друга».

Чтобы пользователь смог добавить друга в свой «френд-лист», ему необходимо найти его по поиску. После этого будет осуществлена проверка на существование пользователя в системе. Если результат будет положительным, то пользователь увидит информационное сообщение о том, что он уже дружит с данным пользователем. В противном случае, система выдаст ошибку, что пользователя не существует.

Когда пользователь системы хочет расформировать свой «френд-лист», то ему необходимо удалить выбранного друга. Для этого требуется нажать на ссылку «Удалить друга» и искомая запись в таблице «friend» будет удалена, а пользователю отобразится информационное сообщение о завершении процедуры.

3.4.6 Модуль «Поиск»

Трудно представить себе систему в котором нет механизма для поиска данных. Модуль «Поиск» позволяет искать данные по следующим алгоритмам:

1. Поиск по контактам.
2. Поиск по сообщениям.

Используя поиск по контактам, можно найти информация сразу по всем пользователям. Поиск по сообщениям позволяет искать информацию по всем перепискам.

Поисковая фраза проходит специальную обработку, при которой, в каждом слове отбрасывается приставка и суффикс, чтобы поиск был наиболее успешным.

Форма поиска и вывода данных изображена на рис. 3.16.

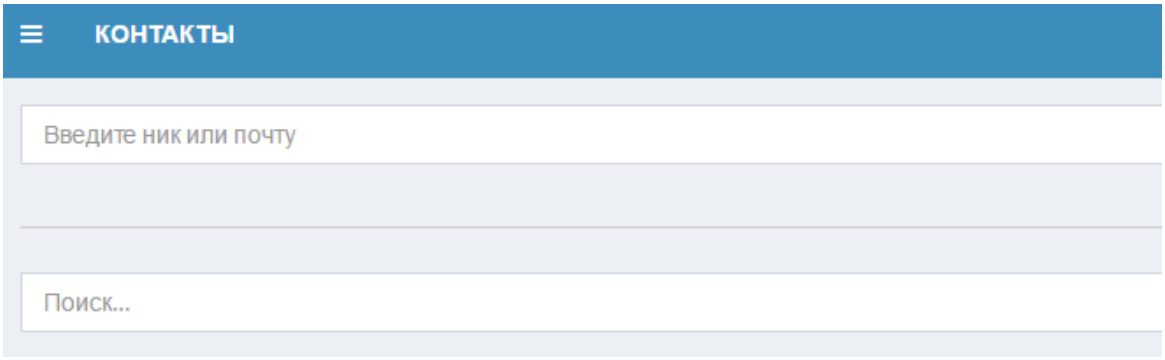

Рисунок 3.16 - Страница поиска информации

3.5 Тестирование программного проекта

Для обеспечения надежности и работоспособности системы были проделаны ряд испытаний, как отдельных функций, так и системы в целом. Для этого были определены следующие механизмы тестирования:

1. Модульное тестирование – тестируется минимально возможный для тестирования компонент, например, отдельный класс или функция.
2. Альфа-тестирование – имитация реальной работы с системой штатным разработчиком.
3. Бета-тестирование – тестирование реальными пользователями системы.

Каждый модуль системы проходил тщательное тестирование. Основной упор был на проверку шифрование данных. Был создан тестовый файл «test.php», в котором производились различные испытания.

Альфа-тестирование производилось администратором портала на Unix-сервере. Первые ошибки выявились при регистрации пользователей, так как их фотографии хранятся на сервере и для директории должны быть указаны полные права на изменение. Учитывая эти требования, ошибки были ликвидированы.

После серии испытаний и доработок, система перешла в режим бета-тестирования. Пользователи регистрировались на портале и добавляли сообщения. Если возникали ошибки, то на сервере появлялись «.log» файлы, в которых было указано место и тип ошибки.

Выводы по главе 3

На основании спроектированных моделей были разработаны алгоритмы работы модулей системы, это позволило составить список основных компонентов системы. Также была рассмотрена реализация компонентов системы. Разработана база данных и определены уровни защиты системы и личных данных пользователей системы. Спроектирована структурная схема информационной системы, а также описание основных функциональных модулей системы. Для наиболее сложных и интересных алгоритмов были приведены блок-схемы.

ЗАКЛЮЧЕНИЕ

В бакалаврской работе было разработано программное обеспечение Мобильный мессенджер ТГУ, реализующее следующие функции:

1. Организация обмена сообщениями.
2. Возможность просмотра истории сообщении.
3. Контролирование информационных потоков администратором портала.
3. Обеспечение различных методов поиска информации.
6. Разделение прав доступа пользователей к информации.
7. Возможность вывода информации в виде отчета.

Разработанный сервис в полной мере соответствует требованиям, предъявленным на этапе постановки задачи. В ходе выполнения бакалаврской работы был проведен детальный анализ предметной области, изучены документы и акты, а также выполнены все этапы проектирования и разработки ИС. Сервис по обмену сообщений реализован современными методами программирования с использованием системного подхода решения задач. Бакалаврская работа является самостоятельным проектом и будет в дальнейшем развиваться.

В настоящее время пользователи системы в сети Интернет можно по следующим адресам: http://messenger.time-it.ru. Регулярная регистрация пользователей в системе, а также добавление различного рода материалов, позволят провести исследования, чтобы рассмотреть динамику социальных взаимодействий.

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

Нормативно-правовые акты

1. ГОСТ 7.32-2001. Отчет о научно-исследовательской работе. Структура и правила оформления.

2. ГОСТ 7.1-2003. Библиографическая запись. Библиографическое описание документа.

3. ГОСТ 7.82-2001. Библиографическая запись. Библиографическое описание электронных ресурсов.

4. ГОСТ 19.701 – 90. Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения (ИСО 5807–85) [Текст]. Введен

5. 1992–01–01. – М.: Изд-во стандартов, 1992. – 14 с. – (Единая система программной документации).

6. ГОСТ Р 34.12-2015. Информационная технология. Криптографическая защита информации. Блочные шифры [Электронный ресурс] // Технический комитет по стандартизации «Криптографическая защита информации» (ТК 26) [Электронный ресурс]. URL: https://tc26.ru/standard/gost/GOST_R_3412-2015.pdf

7. ГОСТ Р 34.13-2015. Информационная технология. Криптографическая защита информации. Режимы работы блочных шифров [Электронный ресурс] // Технический комитет по стандартизации «Криптографическая защита информации» (ТК 26) [Электронный ресурс]. URL: https://www.tc26.ru/standard/gost/GOST_R_3413-2015.pdf

8. ГОСТ 2.105 – 95. Общие требования к текстовым документам [Текст].М.: Изд-во стандартов, 1996. – 29 с. – (Единая система конструкторской документации).

9. Федеральный закон «Об электронной подписи» от 06.04.2011 № 63-ФЗ [Электронный ресурс] // КонсультантПлюс [Электронный ресурс]. URL: http://www.consultant.ru/document/cons_doc_LAW_112701/

10. ГОСТ Р 34.11-2012. Информационная технология. Криптографическая

защита информации. Функция хэширования [Электронный ресурс] // ГОСТ Эксперт [Электронный ресурс]. URL: http://gostexpert.ru/data/files/34.11-2012/70020.pdf

11. Закон Российской Федерации от 15.06.96г. № 72-ФЗ " О товариществах собственников жилья"

12. Закон Российской Федерации от 24.12.92 г. N 4218-1 "Об основах Федеральной жилищной политики" (в редакции Федерального закона №9-ФЗ)

13. Гражданский кодекс РФ Ч.2.Офиц. изд. – М.: ИНФРА-М, 1996

Научная и методическая литература

14. Балдин К. В. Информационные системы в экономике [Электронный ресурс] : учебник / К. В. Балдин, В. Б. Уткин. - 7-е изд. - Москва : Дашков и К°, 2012. - 395 с. – ISBN 978-5-394-01449-9.

15. Буренин С. Н. Web-программирование и базы данных [Электронный ресурс] : учеб. практикум / С. Н. Буренин. - Москва : Моск. гуманит. ун-т, 2014.

- 120 с. - ISBN 978-5-906768-17-9.

16. Вдовин В. М. Предметно-ориентированные экономические информационные системы [Электронный ресурс] : учебное пособие / В. М. Вдовин, Л. Е. Суркова, А. А. Шурупов. - 3-е изд. - Москва : Дашков и К°, 2013.

- 388 с. : ил. - ISBN 978-5-394-02262-3.

17. Золотов С. Ю. Проектирование информационных систем [Электронный ресурс] : учеб. пособие / С. Ю. Золотов ; Томский гос. ун-т систем управления и радиоэлектроники. - Томск : Эль Учебное пособие Контент, 2013. - 86 с. - ISBN 978-5-4332-0083-8.

18. Реинжиниринг бизнес-процессов [Электронный ресурс] : учеб. пособие / А. О. Блинов [и др.] ; под ред. А. О. Блинова. - Москва : ЮНИТИ- ДАНА, 2012. - 341 с. - ISBN 978-5-238-01823-2.

19. Шелухин О. И. Моделирование информационных систем [Электронный ресурс] : учеб. пособие. 004 / О. И. Шелухин. - 2-е изд., перераб. и доп. - Москва : Горячая линия - Телеком, 2012. - 516 с. : ил. - ISBN 978-5- 9912-

0193-3.

20. Антонов, В.Ф. Методы и средства проектирования информационных систем [Электронный ресурс].: учебное пособие / В.Ф. Антонов, А.А. Москвитин ; Федеральное государственное автономное образовательное учреждение высшего профессионального образования «Северо-Кавказский федеральный университет», Министерство образования и науки Российской Федерации. - Ставрополь : СКФУ, 2016. - 342 с. : ил.

21. Медведкова, И.Е. Базы данных [Электронный ресурс]./ И.Е. Медведкова, Ю.В. Бугаев, С.В. Чикунов ; Министерство образования и науки РФ, ФГБОУ ВПО «Воронежский государственный университет инженерных технологий» ; науч. ред. Г.В. Абрамов. - Воронеж : Воронежский государственный университет инженерных технологий, 2014. - 105 с. : ил.

22. Гущин, А.Н. Базы данных [Электронный ресурс].: учебник / А.Н. Гущин. - М. : Директ-Медиа, 2014. - 266 с. : ил.,табл., схем. - ISBN 978-5-4458-5147-9

23. Проектирование информационных систем. Проектный практикум [Электронный ресурс] : учебное пособие / А.В. Платёнкин, И.П. Рак, А.В. Терехов, В.Н. Чернышов ; Министерство образования и науки Российской Федерации, Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Тамбовский государственный технический университет». - Тамбов : Издательство ФГБОУ ВПО «ТГТУ», 2015. - 81 с. : ил., схем. - Библ. в кн. - ISBN 978-5-8265-1409-2

24. Алексунин, В.А. Электронная коммерция и маркетинг в Интернете: Учебное пособие. – 3-е изд. – М.: Издательско-торговая корпорация «Дашков и К », 2013. – 214с.

25. Сорокин, А.А. Объектно-ориентированное программирование [Электронный ресурс]: учебное пособие (курс лекций) / А.А. Сорокин ; Федеральное государственное автономное образовательное учреждение высшего профессионального образования «Северо-Кавказский федеральный университет»,

Министерство образования и науки Российской Федерации. - Ставрополь : СКФУ, 2014. - 174 с.

26. Ясенев, В.Н. Информационные системы и технологии в экономике [Электронный ресурс]: учебное пособие / В.Н. Ясенев. - 3-е изд., перераб. и доп. - М. : Юнити-Дана, 2015. - 560 с. : табл., граф., ил., схемы - Библиогр.: с. 490-497. - ISBN 978-5-238-01410-4

27. Савельева, Н.В. Язык программирования PHP [Электронный ресурс]/ Н.В. Савельева. - 2-е изд., испр. - М. : Национальный Открытый Университет «ИНТУИТ», 2016. - 330 с. : схем., ил.

28. Строганов, А.С. Ваш первый сайт с использованием PHP-скриптов [Электронный ресурс]: учебное пособие / А.С. Строганов. - 3-е изд.. испр. и доп. - М. : Диалог-МИФИ, 2015. - 288 с. : ил. - ISBN 978-5-86404-226-7

29. Баженова, И.Ю. SQL и процедурно-ориентированные языки [Электронный ресурс]/ И.Ю. Баженова. - 2-е изд., испр. - М. : Национальный Открытый Университет «ИНТУИТ», 2016. - 167 с. : ил. - Библиогр. в кн. - ISBN 5-94774-539-9

30. Маркин, А.В. Построение запросов и программирование на SQL [Электронный ресурс]: учебное пособие / А.В. Маркин. - 3-е изд., перераб. и доп. - М. : Диалог-МИФИ, 2014. - 384 с. : ил. - Библиогр.: с. 364-366. - ISBN 978-5-86404-227-4

31. Карпова, Т.С. Базы данных: модели, разработка, реализация [Электронный ресурс]: учебное пособие / Т.С. Карпова. - 2-е изд., исправ. - М. : Национальный Открытый Университет «ИНТУИТ», 2016. - 241 с. : ил.

32. Бедердинова, О.И. Информационные технологии общего назначения [Электронный ресурс]: учебное пособие / О.И. Бедердинова, Ю.А. Водовозова ; Федеральное государственное автономное образовательное учреждение высшего профессионального образования «Северный (Арктический) федеральный университет имени М.В. Ломоносова», Министерство образования и науки Российской Федерации. - Архангельск : САФУ, 2015. - 84 с. : ил. - Библиогр. в кн.

- ISBN 978-5-261-01077-7

33. Балдин, К.В. Информационные системы в экономике [Электронный ресурс]: учебник / К.В. Балдин, В.Б. Уткин. - 7-е изд. - М. : Издательско-торговая корпорация «Дашков и К°», 2017. - 395 с. : ил. - Библиогр. в кн. - ISBN 978-5-394-01449-9

34. Прохорова, О.В. Информатика [Электронный ресурс]: учебник / О.В. Прохорова; Министерство образования и науки РФ, Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Самарский государственный архитектурно-строительный университет», Кафедра прикладной математики и вычислительной техники. - Самара : Самарский государственный архитектурно-строительный университет, 2013. - 106 с. : ил. - Библиогр. в кн. - ISBN 978-5-9585-0539-5

35. Управление данными [Электронный ресурс]: учебник / Ю.Ю. Громов, О.Г. Иванова, А.В. Яковлев, В.Г. Однолько ; Министерство образования и науки Российской Федерации, Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Тамбовский государственный технический университет». - Тамбов : Издательство ФГБОУ ВПО «ТГТУ», 2015. - 192 с. : ил., табл., схем. - Библиогр. в кн.. - ISBN 978-5-8265-1385-9

36. Управление данными [Электронный ресурс]: учебное пособие / Ю.Ю. Громов, О.Г. Иванова, А.В. Яковлев, В.Г. Однолько ; Министерство образования и науки Российской Федерации, Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Тамбовский государственный технический университет». - Тамбов : Издательство ФГБОУ ВПО «ТГТУ», 2014. - 192 с. : ил - Библиогр. в кн. - ISBN 978-5-8265-1374-3

37. Мухсинова, Л. Исследование систем управления [Электронный ресурс]: учебное пособие / Л. Мухсинова ; Министерство образования и науки Российской Федерации, Федеральное государственное бюджетное

образовательное учреждение высшего профессионального образования «Оренбургский государственный университет». - Оренбург : ОГУ, 2013. - 459 с

38. Абрамова, Л.В. Инструментальные средства информационных систем [Электронный ресурс]: учебное пособие / Л.В. Абрамова ; Министерство образования и науки Российской Федерации, Северный (Арктический) федеральный университет имени М.В. Ломоносова. - Архангельск : САФУ, 2013. - 118 с. : ил. - Библиогр. в кн. - ISBN 978-5-261-00851-4

39. Москвитин, А.А. Решение задач на компьютерах [Электронный ресурс]: учебное пособие / А.А. Москвитин. - М. ; Берлин : Директ-Медиа, 2015. - Ч. II. Разработка программных средств. - 427 с. : ил., схем., табл. - Библиогр. в кн. - ISBN 978-5-4475-3646-6

40. Антонов, В.Ф. Методы и средства проектирования информационных систем [Электронный ресурс]: учебное пособие / В.Ф. Антонов, А.А. Москвитин ; Федеральное государственное автономное образовательное учреждение высшего профессионального образования «Северо-Кавказский федеральный университет», Министерство образования и науки Российской Федерации. - Ставрополь : СКФУ, 2016. - 342 с. : ил.

41. Информационные технологии в производстве и бизнесе [Электронный ресурс]: учебник / А.Г. Схиртладзе, В.Б. Моисеев, А.В. Чеканин, В.А. Чеканин ; Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Пензенский государственный технологический университет», Минобрнауки России. - Пенза : ПензГТУ, 2015. - 548 с. : табл., схем., ил.

42. Салий, В. Н. Криптографические методы и средства защиты информации : учеб. пособие [Электронный ресурс] / В. Н. Салий. Саратов : 2012. 41 с. Загл. с экрана.

43. Основы криптографии [Электронный ресурс] / А. П. Алферов, А. Ю. Зубов, А. С. Кузьмин, А. В. Черемушкин. М. : Гелиос АРВ, 2013. 480 с. Загл. с экрана.

44. Баричев, С. Г. Основы современной криптографии [Электронный ресурс] / С. Г. Баричев, В. В. Гончаров, Р. Е. Серов. М. : Горячая линия- Телеком, 2011. 175 с. Загл. с экрана. Яз. рус. 6 Шнайер, Б. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си / Б. Шнайер. М. : Издательство ТРИУМФ, 2003. 816 с.

45. Шаханова, М. В. Современные технологии информационной безопасности : учебно-методический комплекс [Электронный ресурс] / М. В. Шаханова. М. : Проспект, 2015. 216 с.

46. Асимметричные криптосистемы шифрования [Электронный ресурс] // Your Private Network [Электронный ресурс]. URL: http://ypn.ru/197/asymmetricencryption-system/2/

47. Анисимов, В. В. Протоколы аутентификации (идентификации) [Электронный ресурс] / В. В. Анисимов // Anisimovkhv [Электронный ресурс]. URL: https://sites.google.com/site/anisimovkhv/learning/kripto/lecture/tema11

48. Идентификация и аутентификация [Электронный ресурс] // Научная библиотека [Электронный ресурс]. URL: http://sernam.ru/ss_23.php

49. Технологии идентификации – CHAP [Электронный ресурс] // Cisco [Электронный ресурс]. URL: http://www.cisco.com/russian_win/warp/ public/3/ru/solutions/sec/mer_tech_ident-chap.html/

50. Молдовян, Н. А. Введение в криптосистемы с открытым ключом [Электронный ресурс] / Н. А. Молдовян, А. А. Молдовян. М. : БХВ-Петербург, 2005. 285 с.

51. Семенов, Ю. А. Протокол аутентификации Нидхэма-Шредера в случаях симметричной и асимметричной системы шифрования [Электронный ресурс] / Ю. А. Семенов // CIT Forum [Электронный ресурс]. URL: http://citforum.ru/nets/semenov/6/n_s_p_k.shtml

Электронные ресурсы

52. Мобильный протокол MTProto [Электронный ресурс] // Документация Telegram [Электронный ресурс]. URL: https://tlgrm.ru/docs/mtproto

53. Viber [Электронный ресурс]. URL: http://www.viber.com/ru/ (дата обращения: 28.11.2016).

54. WhatsApp [Электронный ресурс]. URL: https://www.whatsapp.com/

55. ICQ [Электронный ресурс]. URL: https://icq.com/android/ru

56. Google Hangouts [Электронный ресурс]. URL: https://hangouts.google.com/

57. Skype [Электронный ресурс]. URL: https://www.skype.com/ru/

58. Telegram Messenger [Электронный ресурс]. URL: https://telegram.org/

Литература на иностранном языке

59. Understanding Cryptography: A Textbook for Students and Practitioners. Год издания: 2010 Авторы: Christof Paar, Jan Pelzl Жанр: Учебное пособие Издательство: Springer

60. Mastering Bitcoin: Unlocking Digital Cryptocurrencies. Год издания: 2014 Автор: Andreas M. Antonopoulos Жанр: Учебное пособие Издательство: O'Reilly Media

61. Cryptography Engineering: Design Principles and Practical Applications. Год издания: 2010 Авторы: Niels Ferguson, Bruce Schneier, Tadayoshi Kohno Жанр: Учебное пособие Издательство: Wiley

62. P. B Burtyka, O. B Makarevich. Symmetric fully homomorphic encryption using decidable matrix equations // Proceedings of the 7nd international conference on security of information and networks ACM, 2014, p. 67–70.

63. D. Boneh, C. Gentry, S. Halevi, F. Wang, D. J Wu. Private database queries using somewhat homomorphic encryption // Applied cryptography and network security Springer, 2013, p. 102–118.

64. J. Wiens, J. Guttag, E. Horvitz. Learning evolving patient risk processes for c. diff colonization // Icml workshop on machine learning from clinical data, 2012.

65. A. Singh, J. V Guttag. A comparison of non-symmetric entropy-based classification trees and support vector machine for cardiovascular risk stratification // Engineering in medicine and biology society, embc, 2011 annual international

conference of the ieee IEEE, 2011, p. 79–82.

66. A. Singh, G. Nadkarni, J. Guttag, E. Bottinger. Leveraging hierarchy in medical codes for predictive modeling // Proceedings of the 5th acm conference on bioinformatics, computational biology, and health informatics ACM, 2014, p. 96–103.

67. M. A Pathak, B. Raj, S. Rane, P. Smaragdis. Privacy preserving speech processing, http://www.cs.illinois.edu/~paris/pubs/pathak-spm2013.pdf.

ПРИЛОЖЕНИЕ А

Алгоритм шифрования

alpha - исходный алфавит
abcdefghijklmnopqrstuvwqyzABCDEFGHIJKLMNOPQRSTUVWXYZабвгдеёжзийк
лмнопрстуфхцчшщъыьэюяАБВГДЕЁЖЗИЙКЛМНОПРщТУФХЦЧШЩЪЫЬЭЮ
Я -.,!?:()[]<>=@#%^&*+/~

alphax - alpha смешивается случайным образом (разово) и хранится на сервере в виде ключа, отдается клиенту php-скриптом при загрузке диалога/диалогов
[!щжRjБXbTДqЛWт#Ы-
йzЯMEiCyMow~ЧП.qQcxH>*ЩSkOFrn)=юшбvцзнDaoZЬJNьgUЭЁcЪHBdV%]Cm
Ю+фhXpЗ:tОъпРВФ@,^GeУuШу&lIмд/аЦЖлYГfИА<гЕвАЙяиёч(РТыкКеэрKL?
s

key - генерируется путём суммирования цифр id-пользователя этого сообщения

message - нешифрованное сообщение пользователя

messagex - шифрованное сообщение получается путем цикличного сдвига вправо символов из message на key в алфавите alphaX.

ПРИЛОЖЕНИЕ Б

Листинг программы

Просмотр сообщений в диалогах

```
<?php
class profile extends application_controller {

	public function __construct() {}

	public function getById($id){
		return $this->db->get_row('SELECT * FROM users WHERE id = '.(int)$id);
	}

	public function getByEmail($email){
		return $this->db->get_row('SELECT * FROM users WHERE email = "'.$email.'"');
	}

	public function getByNickname($nickname){
		return $this->db->get_row('SELECT * FROM users WHERE nickname =
"'.$nickname.'"');
	}

	public function reg($email){
		if ($this->getByEmail($email)) return false;
		else {
			$user = array(
				'email' => $email,
				'password' => rand(100000,999999),
				'nickname' => $email,
				'lastactivity' => time()
			);
			$user['id'] = $this->db->insert('users', $user);
			return $user;
		}
	}

	public function auth($email,$password){
		$user = $this->getByEmail($email);
		if (empty($user)) return false;
		else {
			if($user['password'] == $password) return $user['id'];
			else return false;
		}
	}

	public function find($search){
		$user = $this->getByNickname($search);
		if(empty($user)) $user = $this->getByEmail($search);
		if(empty($user)) return false;
		else return $user;
	}
```

```
	public function addContact($user,$contact){
		if($this->isContact($user,$contact)) return false;
		return $this->db->insert('contact_list',array(
			'id_user' => (int)$user,
			'id_user_contact' => (int)$contact,
			'date' => time()
		));
	}

	public function delContact($user,$contact){
		return $this->db->delete('contact_list',array(
			'id_user' => (int)$user,
			'id_user_contact' => (int)$contact,
		));
	}

	public function isContact($user, $contact){
		return $this->db->get_one('SELECT id from contact_list WHERE id_user =
'.(int)$user.' AND id_user_contact = '.(int)$contact);
	}

	public function getContactList($user){
		return $this->db->get_all('SELECT u.id, u.email, u.name, u.lname, u.sname,
u.nickname, u.lastactivity, cl.date, cl.status FROM contact_list cl LEFT JOIN users u ON u.id =
cl.id_user_contact WHERE cl.id_user = '.(int)$user);
	}

	public function findDialog($users){
		// $users = asort($users);
		$did = $this->db->get_one('SELECT id_dialog FROM dialogs_members WHERE
id_user = '.(int)$users[0].' AND id_dialog IN (SELECT id_dialog FROM dialogs_members WHERE
id_user = '.(int)$users[1].')');
		if(empty($did)){
			// Диалога между этими пользователями нет, создаём новый.
			$did = $this->db->insert('dialogs',array(
				'date' => time(),
				'id_user_create' => $users[0]
			));
			foreach ($users as $item) {
				$this->db->insert('dialogs_members', array(
					'id_dialog' => $did,
					'id_user' => $item,
					'last_view_time' => ($item == $users[0]) ? time() : 0
				));
			}
		}
		return $did;
	}

	public function sendMessage($user,$recipient,$message){
		return $this->db->insert('dialogs_messages', array(
```

```
                        'id_sender'             => (int)$user,
                        'id_dialog'     => $this->findDialog(array((int)$user,(int)$recipient)),
                        'date'                  => time(),
                        'text'                  => trim($_POST['message'])
                ));
        }

        public function getDialogList($user){
                $sql = 'SELECT * FROM dialogs_members WHERE id_user = '.(int)$user;
                return $this->db->get_all($sql);
        }

        public function getDialogMembers($did, $self = 0){
                $sql = 'SELECT id_user FROM dialogs_members WHERE id_dialog = '.(int)$did.'
AND id_user != '.(int)$self;
                return $this->db->get_one($sql);
        }

        public function getLastMessage($did){
                return $this->db->get_row('SELECT * FROM dialogs_messages WHERE id_dialog =
'.(int)$did.' ORDER BY date DESC LIMIT 0,1');
        }

        public function getMessageList($did,$id = 0){
                return $this->db->get_all('SELECT * FROM dialogs_messages WHERE id_dialog =
'.(int)$did.' AND id > '.(int)$id.' ORDER BY date ASC');
        }

}
?>
```

ПРИЛОЖЕНИЕ В

Листинг программы

Регистрация пользователя

```
<?php
	class profile_controller extends application_controller {

		public function index() {
			$uid = $this->getUserId();
			if(!$uid) $this->logout();
			$this->updateActivity();
			header('Location:/profile/contacts/');
		}

		public function messages(){
			$uid = $this->getUserId();
			if(!$uid) $this->logout();
			$this->updateActivity();
			$user = $this->profile->getById($uid);

			$arResult['dialogs_list'] = $this->profile->getDialogList($uid);
			if(!empty($arResult['dialogs_list'])){
				foreach ($arResult['dialogs_list'] as &$item) {
					$duid = $this->profile-
>getDialogMembers($item['id_dialog'],$uid);
					$duser = $this->profile->getById($duid);
					$info = (!empty($duser['lname']) && !empty($duser['name'])) ?
$duser['lname'].' '.$duser['name'] : $duser['nickname'];
					$item['info'] = !empty($info) ? $info : $duser['email'];
					$item['duid'] = $duid;

					$message = $this->profile-
>getLastMessage($item['id_dialog']);
					$item['text'] = $message['text'];
					$item['date'] = date('d.m.Y H:i',$message['date']);
					$item['id_sender'] = $message['id_sender'];
					if($message['date'] > $item['last_view_time']) $item['unread'] =
true;
				}
			}

			$this->layout = 'profile';
			$this->html->tpl_vars['menu_messages'] = true;
			$this->html->tpl_vars['encodemsg'] = $user['encodemsg'];
			$this->html->tpl_vars['userid'] = $uid;
```

```
                $this->html->tpl_vars['alphax'] = $this->config->get('alphax','site');
                $this->html->render('profile/header_messages.html',array(),'profile_header');
                $this->html->render('profile/messages.html', $arResult,'content');

        }

        public function contacts(){
                $uid = $this->getUserId();
                if(!$uid) $this->logout();
                $this->updateActivity();

                $arResult['contact_list'] = $this->profile->getContactList($uid);
                if(!empty($arResult['contact_list'])){
                        foreach ($arResult['contact_list'] as &$item) {
                                $item['lastactivity'] = date('d.m.Y H:i', $item['lastactivity']);
                                $item['date'] = date('d.m.Y H:i', $item['date']);

                                $name = (!empty($item['lname']) && !empty($item['name'])) ?
$item['lname'].' '.$item['name'] : $item['nickname'];
                                $item['info'] = !empty($name) ? $name : $item['email'];
                        }
                }

                $this->layout = 'profile';
                $this->html->tpl_vars['menu_contacts'] = true;
                $this->html->render('profile/header_contacts.html',array(),'profile_header');
                $this->html->render('profile/contacts.html', $arResult,'content');

        }

        public function dialogs(){
                $uid = $this->getUserId();
                if(!$uid) $this->logout();
                $this->updateActivity();
                $user = $this->profile->getById($uid);

                $duid = (int)$_GET['id_user']; // id собеседника
                $arResult = array();
                if($this->profile->isContact($uid, $duid)){
                        $duser = $this->profile->getById($_GET['id_user']); // собеседник
                        if(!empty($duser)) {
                                $dialog_name = (!empty($duser['lname']) &&
!empty($duser['name'])) ? $duser['lname'].' '.$duser['name'] : $duser['nickname'];
                                $this->html->tpl_vars['dialog_name'] = !empty($dialog_name)
```

```
? $dialog_name : $duser['email'];
                                }

                                $arResult['duid'] = $duid;
                                $arResult['did'] = $this->profile->findDialog(array($uid,$duid));

                                $this->updateDialogsActivity($arResult['did']);
                                $arResult['message_list'] = $this->profile->getMessageList($arResult['did']);
                                $muser = array();
                                if(!empty($arResult['message_list'])){
                                        foreach ($arResult['message_list'] as &$item) {
                                                $item['date'] = date('d.m.Y H:i', $item['date']);
                                                if(empty($muser[$item['id_sender']]))
$muser[$item['id_sender']] = $this->profile->getById($item['id_sender']);

                                                $name = (!empty($muser[$item['id_sender']]['lname'])
&& !empty($muser[$item['id_sender']]['name'])) ? $muser[$item['id_sender']]['lname'].'
'.$muser[$item['id_sender']]['name'] : $muser[$item['id_sender']]['nickname'];

                                                $item['info'] = !empty($name) ? $name :
$muser[$item['id_sender']]['email'];
                                                if($item['id_sender'] == $uid) $item['self'] = true;
                                        }
                                }

                        } else {
                                $this->html->tpl_vars['dialog_name'] = 'Ошибка';
                                $this->html->tpl_vars['dialog_error'] = 'Пользователь отсутствует в
Вашем списке контактов!';
                                }


                        $this->layout = 'profile';
                        $this->html->tpl_vars['menu_dialogs'] = true;
                        $this->html->tpl_vars['encodemsg'] = $user['encodemsg'];
                        $this->html->tpl_vars['userid'] = $uid;
                        $this->html->tpl_vars['alphax'] = $this->config->get('alphax','site');
                        $this->html->render('profile/header_dialogs.html',array(),'profile_header');
                        $this->html->render('profile/dialogs.html', $arResult,'content');

                }

                public function settings(){
                        $uid = $this->getUserId();
```

```
		if(!$uid) $this->logout();
		$this->updateActivity();

		if(!empty($_POST)){
			$user = array(
				'email'  => trim($_POST['email']),
				'name'		=> trim($_POST['name']),
				'lname'		=> trim($_POST['lname']),
				'sname'		=> trim($_POST['sname']),
				'nickname'	=> trim($_POST['nickname']),
				'encodemsg'	=> !empty($_POST['encodemsg']) ? true : false,
			);
			if(!empty(trim($_POST['password'])) &&
!empty(trim($_POST['password-repeat'])) && (trim($_POST['password']) ==
trim($_POST['password-repeat']))) $user['password'] = trim($_POST['password']);
			$this->db->update('users',$user,$uid);
		} else $user = $this->profile->getById($uid);
		$this->layout = 'profile';
		$this->html->tpl_vars['menu_settings'] = true;
		$this->html->render('profile/header_settings.html',array(),'profile_header');
		$this->html->render('profile/settings.html', $user,'content');
	}

	public function logout(){
		if(!empty($this->getUserId())) $this->session->set('userid',false);
		header('Location:/');
	}

	public function getUserId(){
		return $this->session->get('userid');
	}

	public function updateDialogsActivity($did){
		$this->db->update('dialogs_members',
			array('last_view_time'=>time()),
			array(
				'id_user' => $this->getUserId(),
				'id_dialog' => $did
			)
		);
	}

	public function updateActivity(){
		$this->db->update('users',array('lastactivity'=>time()),$this->getUserId());
```

```
}

public function reg(){
	if(empty($_POST['email'])){
		echo json_encode(array(
			'status' => 'error',
			'error' => 'Email не может быть пустым',
			'data' => false,
		));
		die();
	} else {
		$user = $this->profile->reg($_POST['email']);

	}

	if(!empty($user)){
		unset($user['nickname']);
		unset($user['lastactivity']);
		$this->session->set('userid',$user['id']);
		echo json_encode(array(
			'status' => 'ok',
			'data' => $user
		));
	} else {
		echo json_encode(array(
			'status' => 'error',
			'error' => 'Пользователь с таким email существует!'
		));
	}
	die();
}

public function auth(){
	if(empty($_POST['email']) || empty($_POST['password'])){
		echo json_encode(array(
			'status' => 'error',
			'error' => 'Поля email и пароль не могут быть пустыми'
		));
		die();
	} else {
		$user = $this->profile->auth($_POST['email'],$_POST['password']);

	}
```

```
        if(!empty($user)){
                $this->session->set('userid',$user);
                echo json_encode(array(
                        'status' => 'ok'
                ));
        } else {
                echo json_encode(array(
                        'status' => 'error',
                        'error' => 'Пользователь с такими email/пароль существует!'
                ));
        }
        die();
}

public function find(){
        $uid = $this->getUserId();
        if(!$uid) $this->logout();

        if(empty($_POST['search'])){
                echo json_encode(array(
                        'status' => 'error',
                        'error' => 'Укажите почту или ник пользователя',
                        'data' => false,
                ));
                die();
        } else {
                $user = $this->profile->find($_POST['search']);

                if($user['id'] == $uid){
                        echo json_encode(array(
                                'status' => 'error',
                                'error' => 'Нельзя добавить себя в список контактов',
                                'data' => false,
                        ));
                        die();
                }
        }

        if(!empty($user)){
                if($this->profile->addContact($this->getUserId(),$user['id'])){
                        unset($user['password']);
                        unset($user['id']);
                        unset($user['lastactivity']);
                        echo json_encode(array(
```

```
                                        'status' => 'ok',
                                        'data' => $user
                                ));
                        } else {
                                echo json_encode(array(
                                        'status' => 'error',
                                        'error' => 'Пользователь был добавлен раньше',
                                        'data' => false,
                                ));
                        }
                } else {
                        echo json_encode(array(
                                'status' => 'error',
                                'error' => 'Пользователь с таким ником/email не
существует!'
                        ));
                }
                die();
        }

        public function uncontact(){
                $uid = $this->getUserId();
                if(!$uid) $this->logout();

                if(empty($_POST['id'])){
                        echo json_encode(array(
                                'status' => 'error',
                                'error' => 'Произошла ошибка',
                                'data' => false,
                        ));
                        die();
                } else {
                        $this->profile->delContact($uid, $_POST['id']);
                        echo json_encode(array(
                                'status' => 'ok',
                                'data' => false
                        ));
                        die();
                }
        }

        public function send(){
                $uid = $this->getUserId();
                if(!$uid) $this->logout();
```

```
                if(empty($_POST['duid']) || empty($_POST['message'])){
                        echo json_encode(array(
                                'status' => 'error',
                                'error' => 'Произошла ошибка',
                                'data' => false,
                        ));
                        die();
                } else {
                        if($this->profile->isContact($uid,$_POST['duid'])){
                                $message = $this->profile->sendMessage($uid,$_POST['duid'],$_POST['message']);
                                if($message){
                                        echo json_encode(array(
                                                'status' => 'ok',
                                                'data' => false
                                        ));
                                        die();
                                } else {
                                        echo json_encode(array(
                                                'status' => 'error',
                                                'error' => 'Сообщение не отправлено',
                                                'data' => false,
                                        ));
                                        die();
                                }
                        } else {
                                echo json_encode(array(
                                        'status' => 'error',
                                        'error' => 'Пользователь отсутствует в Вашем листе контактов',
                                        'data' => false,
                                ));
                                die();
                        }
                }
        }

        public function update(){
                $uid = $this->getUserId();
                if(!$uid) $this->logout();

                if(empty($_POST['duid']) || empty($_POST['id'])){
                        echo json_encode(array(
```

```
                    'status' => 'error',
                    'error' => 'Произошла ошибка',
                    'data' => false,
                ));
                die();
            } else {
                if($this->profile->isContact($uid,$_POST['duid'])){
                    $arResult['duid'] = $_POST['duid'];
                    $arResult['did'] = $this->profile->findDialog(array($uid,$arResult['duid']));
                    $this->updateDialogsActivity($arResult['did']);
                    $arResult['message_list'] = $this->profile->getMessageList($arResult['did'],$_POST['id']);
                    $muser = array();
                    if(!empty($arResult['message_list'])){
                        foreach ($arResult['message_list'] as &$item) {
                            $item['date'] = date('d.m.Y H:i', $item['date']);
                            if(empty($muser[$item['id_sender']])) $muser[$item['id_sender']] = $this->profile->getById($item['id_sender']);

                            $name = (!empty($muser[$item['id_sender']]['lname']) && !empty($muser[$item['id_sender']]['name'])) ? $muser[$item['id_sender']]['lname'].' '.$muser[$item['id_sender']]['name'] : $muser[$item['id_sender']]['nickname'];
                            $item['info'] = !empty($name) ? $name : $muser[$item['id_sender']]['email'];
                            if($item['id_sender'] == $uid) $item['self'] = true;
                        }
                    }
                    $text_message = $this->html->render('profile/dialogs_messages.html', $arResult);
                    echo json_encode(array(
                        'status' => 'ok',
                        'data' => $text_message
                    ));
                    die();
                } else {
                    echo json_encode(array(
                        'status' => 'error',
                        'error' => 'Пользователь отсутствует в Вашем листе контактов',
                        'data' => false,
                    ));
                    die();
```

```
                    }
                }
            }
        }
    ?>
```