МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Некоммерческое акционерное общество
АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ

кафедра Информационные системы

«Допущен к защите»
Заведующий кафедрой _____
_____
(Ф.И.О., ученая степень, звание)
_____ «_____»_____ **20__ г.**
(подпись)

ДИПЛОМНЫЙ ПРОЕКТ

На тему: Проектирование информационной системы для безопасной передачи данных

Специальность 5В070300 – „ Информационные системы "

Выполнил (а) Куатбеков Д.М. ИС-10-2
(Фамилия и инициалы) группа

Научный руководитель Дьячков В.В., к.т.н., доцент
(Фамилия и инициалы, ученая степень, звание)

Консультанты:
по экономической части:
Экономич. ч Бекишева А.Ч., к.э.н., доцент
(Фамилия и инициалы, ученая степень, звание)
_____ «16» мая 2014 г.
(подпись)

по безопасности жизнедеятельности:
Бегимбетова А.С., ст. преп.
(Фамилия и инициалы, ученая степень, звание)
_____ «9» июня 2014 г.
(подпись)

по применению вычислительной техники:
Кашубаева Г.К. ст. преподаватель
(Фамилия и инициалы, ученая степень, звание)
_____ «04» 06 2014 г.
(подпись)

Нормоконтролер: Ни А.Л., к.т.н., доцент
(Фамилия и инициалы, ученая степень, звание)
_____ «09» 06 2014 г.
(подпись)

Рецензент: _____
(Фамилия и инициалы, ученая степень, звание)
_____ «_____»_____ 20____ г.
(подпись)

Алматы 2014 г.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Некоммерческое акционерное общество
АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ

Факультет „Информационные технологии"
Специальность 5В070300 – „Информационные системы"
Кафедра „Информационные системы"

ЗАДАНИЕ
на выполнение дипломного проекта

Студент Куатбеков Данияр Муратбекович
(фамилия, имя, отчество)
Тема проекта Проектирование информационной системы для безопасной передачи данных

утверждена приказом ректора № ___ от «___» сентября 20___ г.
Срок сдачи законченной работы «___» _____ 20___ г.
Исходные данные к проекту требуемые параметры результатов проектирования (исследования) и исходные данные объекта
Исходные данные: Задание к дипломному проекту. Требуемые параметры результатов проектирования: модель информационной системы для безопасной передачи данных, ER-диаграмма базы данных, программная реализация приложения, база данных информационной системы. Исходные данные объекта: результаты обследования предметной области.

Перечень подлежащих разработке дипломного проекта вопросов или краткое содержание дипломного проекта:
1. Анализ и исследование информационной системы для безопасной передачи данных.
2. Проектирование и разработка информационной системы.
3. Разработка программного обеспечения информационной системы
4. Технико-экономическое обоснование проекта
5. Анализ условий труда.

Перечень графического материала (с точным указанием обязательных чертежей)

1. Диаграмма прецедентов
2. Диаграмма видов деятельности для прецедента
3. Диаграмма последовательности прецедента
4. Диаграмма компонентов
5. Диаграмма развертывания
6. Логическая модель базы данных
7. Физическая модель базы данных

Рекомендуемая основная литература

1. П. Нортон, Г. Шилдт «Java 2 Наиболее полное руководство» - издательство «Express», 2012 - 1102с.
2. Власова Н. Информационная безопасность систем управления базами данных
3. Гладченко А. Создание гибкой системы безопасности
4. У. Боггс, М. Боггс «UML и Rational Rose 2002» - издательство «Лорей», 2004г - 415с
5. Бекишева А.И. Методические указания к выполнению экономической части дипломной работы.
6. Шаньгин, В.Ф. «Защита компьютерной информации. Эффективные методы и средства» издательство «Пресс», 2010г - 544с.

Консультанты по проекту с указанием относящихся к ним разделов

Раздел	Консультант	Сроки	Подпись
БЖД	Бекишбекова А.С.	20.04 - 4061	
Экономч. часть	Бекишева А.И.		

Г Р А Ф И К
подготовки дипломного проекта

№ п/п	Наименование разделов, перечень разрабатываемых вопросов	Сроки представления руководителю	Примечание
1	Исследование информационной безопасности, описание алгоритмов шифрования и протоколов шифрования, обоснование выбранных программных средств	10.01.14-10.02.14 12.02.14-12.04.14	
2	Проектирование и разработка ИС «BestLine», разработка UML диаграмм, разработка базы данных.		
3	Разработка программного обеспечения ИС «BestLine», назначение и условия выполнения программы, разработка его интерфейса	14.04.14-25.05.14	
4	Технико-экономическое обоснование ИС «BestLine»	4.04.14-4.05.14	
5	Безопасность жизнедеятельности	4.03.14-4.04.14.	

Дата выдачи задания «____»_____20___ г.

Заведующий кафедрой _____ _____
(подпись) (Фамилия и инициалы)

Руководитель _____ Дьячков В.В
(подпись) (Фамилия и инициалы)

Задание принял к исполнению
студент _____ Куатбеков Д.М.
(подпись) (Фамилия и инициалы)

Аңдатпа

Дипломдық жобада автоматтандырылған ақпараттық «BestLine» деген жүйе жасалған.

Ара жүріс айтылмыш жұмыстың орындалуы клиент-серверного аддендумнің жұмысының ұстанымдары деректердің қауіпсіз берілісі үшін мен игерушілік шифрлаудың және хэширования алгоритмының алгоритмдарының таныс-, ал да диаграмманың UML және дерекқор ИС "BestLine" әзірле.

ИС "BestLine" интернетте мақсатпен деректердің қауіпсіз берілісінің әзірле.

Аннотация

В дипломном проекте разработана информационная система «BestLine». В ходе выполнения данной работы изучены принципы работы клиент-серверного приложения для безопасной передачи данных с использованием алгоритмов шифрования и алгоритмов хэширования, а также разработаны UML диаграммы и база данных ИС «BestLine».

ИС «BestLine» разработана с целью безопасной передачи данных в интернете.

Annotation

In the graduation project was developed the automated information system of «BestLine». In the course of this work studied the principles of client-server application for secure data transmission using encryption algorithms and hashing algorithms, and developed UML diagrams and database IS «BestLine».

IS «BestLine» is designed to secure data on the Internet.

Содержание

Введение

Информация на сегодняшний день является наиболее важным ресурсом, который имеет определенную стоимость и подлежит защите.

Защиту информации от различных видов угроз для предоставления нормального режима работы, минимизирования конструктивных рисков, успешного развития бизнеса, гарантирования конфиденциальности сотрудников предприятия - обеспечивает информационная безопасность.

Информация и средства информации могут быть подвержены широкому спектру угроз, как производственный шпионаж, заражения компьютерными вирусами, внедрения программ закладок, несанкционированного доступа к конфиденциальной информации, что становиться более частыми в индустриальном мире.

Реализация угроз может нанести большой ущерб и стать причиной колоссальных убытков, причиной снижения деловой активности, временного или полного прекращения работы. Для предупреждения реализации угроз предпринимаются меры, включая безусловное выполнение определенных политикой безопасности организации правил и порядком работы с информационной системой, использование средств защиты информации, контроль со стороны ответственных сотрудников за выполнением сотрудниками своих функций по обеспечению информационной безопасности.

Используя современные средства визуальной разработки программного обеспечения можно создать собственную программу по обмену сообщениями без потери данных. В данной дипломной работе рассматривается создание клиент-серверного приложения для безопасной передачи данных с помощью среды визуальной разработки приложений Eclipse на языке программирования Java.

Создание ИС BestLine преследует достижение следующих основных целей:

- защищенная передача информации;
- предоставление доступа к информации только авторизованным сотрудникам;
- защиты от модификации или подмены информации;
- снижение временных затрат на обработку данных;
- обеспечение защиты от несанкционированного доступа и дезинформации.

1 Общее описание

1.1 Постановка задачи дипломной работы

На сегодняшний день информация является одним из самых существенных ресурсов. Информация постоянно играла весьма значимую роль в жизни человека.

Общеизвестно высказывание о том, что тот, кто владеет информацией, тот владеет и миром. Другая информация стоит дороже жизни.

С течением времени роль информации в жизни человека заделывалась все значительнее. Необходимо было овладевать и уяснять уже не только законы природы, но и представления и ценности человеческого общества – литературу, искусство, архитектуру и т.д. На сегодняшний день, в первой половине 21-ого века роль информации в жизни человека является определяющей – чем больше навыков и познаний он обладает, тем значительнее ценится как мастер и сотрудник, тем предпочтительнее уважения в компании.

Постигая окружающий мир, человек всегда имеет дело с информацией. Она помогает человеку верно оценить проистекающие события, принять взвешенное решение, отыскать наиболее успешный вариант своих поступков. Инстинктивно мы соображаем, что информация — это то, чем каждый из нас пополняет личный багаж знаний. Информация также является сильнейшим средством воздействия на личность и общество в целом. Кто обладает максимальным объемом информации по какой либо проблеме, тот всегда находится в более выигрышном положении по соотнесению с другими.

В последние десятилетия настоятельно сообщают о переходе от «индустриального общества» к «обществу информационному». Случается замена способов производства, идеологии людей, их типа жизни. Информационные технологии кардинальным образом изменяют обыденную жизнь миллионов людей.

Информация стала одним из существенных стратегических, управленческих ресурсов, наравне с ресурсами - человеческим, финансовым, вещественным. Ее производство и потребление составляют надобную основу результативного функционирования и формирования разнообразных сфер общественной жизни, и, прежде всего, экономики. А это обозначает, что не только определенному человеку делаются доступными источники информации в каждой части нашего мира, но и генерируемая им свежая информация становится достоянием всего человечества. В современных условиях право на информацию и доступ к ней владеют актуальную ценность для всех членов компании. Усиливающаяся роль информации в обществе явилась объектом научного постижения. Были выставлены теории, поясняющие ее место и смысл. Наиболее знаменитыми являются теории постиндустриального и информационного общества.

Общество входит в новейшую эру – информационную, в век электронной экономической деятельности, сетевых сообществ и объединений без рубежей. Наступление новейшей эпохи радикально изменит экономические и социальные стороны жизни мира. Аналогичные модификации самым открытым типом относятся места человека в информационном обществе. Человек трансформируется в парадоксе с вектором информационно-технических характеристик общества. Все-таки это совсем деятельное принятие новейших обстоятельств производства и потребления. Человек является субъектом информационной реальности, далеко выходящей за информационно технические характеристики. Информатизация будничной жизни и появление свежего информационного фона человеческого жизни не проходит бесследно для жизненного общества человека. В электронном пространстве меняются поведенческие эталоны и ценностные ориентации личности.

В области теоретического соображения случающихся процессов также до сих пор нет единственного взгляда относительно путей развития информационного мира, приоритетности этого или другого его назначения, разборчивости и отчетливости формулирований и мнений, выражающих совершающееся в информационной среде. Поэтому теоретическое изучение как концептуальных, так и практических (реальных) предпосылок соображения проходящих информационных процессов остается актуальным

Цель данной дипломной работы защитить информацию передаваемую по сети при помощи алгоритмов шифрования, и алгоритмов хэширования.

В данной дипломной работе я использовал алгоритм шифрования 3Des, аутентификация по протоколу CHAP, алгоритм хэширования MD5 и базу данных MySQL .

1.2 Описание алгоритмов шифрования и протоколов, используемых при написании дипломной работы

ПротоколCHAP — широко общераспространённый алгоритм контроля подлинности, предусматривающий передачу не самого пароля пользователя, а косвенных сведений о нём. При применении CHAP сервер удалённого доступа посылает клиенту строку запроса. В следствии этой строки и пароля пользователя клиент высчитывает хэш-код MD5 и отправляет его серверу. Хэш-функция является алгоритмом одностороннего шифрования, так как значение хэш-функции для блока данных вычислять легко, а установить исходный блок по хэш-коду с математической точки зрения невыполнимо за надобное время. Сервер, которому доступен пароль пользователя, осуществляет те же самые подсчеты и сопоставляет следствие с хэш-кодом, полученным от клиента. В происшествии совпадения учётные данные клиента удалённого доступа почитаются действительными. Наиболее значительной особенностью алгоритма CHAP оказывается то, что пароль никогда не

передается по каналу, что существенно усиливает безопасность процесса аутентификации по сопоставлению с применением протокола PAP.

ПротоколCHAP— это протокол проверки подлинности субъекта «запрос-ответ», применяющий стереотипную схему хэширования MessageDigest 5 для шифрования вопроса. Протокол CHAP употребляется массой поставщиков серверов и клиентов доступа к сети. Сервер, применяющий маршрутизацию и удаленный доступ, помогает CHAP таким образом, что осуществляется испытание подлинности клиента удаленного доступа, отвечающего данный протокол. Так как CHAP вызывает использования обратимо зашифрованного пароля, рекомендуется применить другой протокол контроля подлинности, например MS-CHAP [10].

Чтобы включить проверку подлинности с использованием протокола CHAP, выполните вытекающие действия:

– включите CHAP в качестве протокола проверки подлинности на сервере удаленного доступа. По умолчанию CHAP выключен;

– включите применение CHAP в должной политике удаленного доступа;

– включите хранение обратимо зашифрованной формы пароля пользователя. Хранение обратимо зашифрованных паролей пользователя нужно позволить как для отдельной учетной записи, так и для всех учетных записей домена;

– реализуете принудительный сброс паролей пользователей, чтобы новый пароль был сохранен в обратимо зашифрованной форме.

При подключении сохранения паролей в обратимо зашифрованной конфигурации, проходящие пароли не находиться преобразованы в эту конфигурацию автоматически. Нужно или отправить пароли пользователей, или затребовать замену паролей пользователей при следующем входе в систему.

Если затребована смена паролей пользователей при следующем входе в систему, то пользователи, перед тем как пытаться выполнить удаленное подключение с использованием протокола CHAP, должны будут войти в систему с помощью подключения к локальной сети и изменить пароль. Нельзя изменять пароли в процессе проверки подлинности с использованием протокола CHAP — попытка подключения будет отклонена. Одним из возможных решений, позволяющим пользователю удаленного доступа обойти это ограничение, является временный вход в систему с использованием протокола MS-CHAP с целью изменения пароля.

– включите использование CHAP на клиенте удаленного доступа.

Дополнительные сведения см. в разделе Протокол проверки подлинности CHAP [12].

Хэш-функции являются необходимым элементом ряда криптографических схем. Под этим термином понимаются функции, отображающие сообщения произвольной длины (иногда длина сообщения ограничена, но достаточно большим числом) в значения фиксированной

длины. Последние часто называют хэш-кодами. Таким образом, у всякой хэш-функции имеется большое количество коллизий, т. е. пар значений x<=>y таких, что h(x)=h(y) . Основное требование, предъявляемое криптографическими приложениями к хэш-функциям, состоит в отсутствии эффективных алгоритмов поиска коллизий.

Схемы электронной подписи - главная область использования хэш-функций в криптографии. Потому что употребляемые на практике схемы электронной подписи не приспособлены для подписи информаций всякой длины, а процедура, заключающаяся в разбиении информации на ассоциации и в генерации подписи для всякого блока по отдельности, весьма недействительна, единственным рациональным решением кажется использование схемы подписи к хэш-коду сообщения. Несложно осмыслить, что присутствие действенных методов розыска коллизий для хэш-функции подрывает стойкость протокола электронной подписи.

Хэш-функции применяются также в кое-каких протоколах аутентификации для уменьшения их совместной сложности, т. е. для уменьшения длин принимаемых информации, и в некоторых иных криптографических протоколах.

Контрольная сумма — кое-какое значение, рассчитанное из заданных данных путём использования установленного алгоритма, применяемое для контроли аутентификации трансляции данных (для исключения воздействия каких-нибудь помех при трансляции). Отдельные виды контрольных сумм:

− проверка избыточности циклической суммы (в виде BRB6, BRB12, BRB24) — в общественном происшествии используется для обследования целостности трансляции данных. Программы-архиваторы подключают BRB исходных данных в основанный архив для того, чтобы принимающий мог удостовериться в корректности приобретенных данных. Элементарен в осуществлении, при этом обеспечивает малую возможность коллизии, так что различные данные практически точно имеют разную контрольную сумму. Для вычисления применяются побитовый сдвиг;

− высчитывание MD5-свёртки после закачивания файлов для сопоставления с заранее знакомой. Необходима для доказательства подлинности приобретенного файла;

− под наименованием «контрольное число» входит в состав номеров товаров и разнообразных документов.

Виды:

– MD6;
– MD4;
– MD5;
– SECURE HASH ALGORITHM (SHA);
– RIPE-MD;
– HAVAL;
– MDC.

MessageDigest 5(MD5) — 128-битный алгоритм хэширования, разработанный профессором Рональдом Л. Ривестом из Массачусетского Технологического Института (MIT, Massachusetts Institute of Technlogy) в 1991 году.

Предназначен для произведения «отпечатков» или «дайджестов» сообщений различной длины. Пришёл на замену MD4, который был несовершенен. Действует на 32-битных машинах.

Владея MD5, почти нельзя возобновить входное сообщение, так как данному MD5 могут отвечать различные сообщения. Применяется для проверки подлинности выпущенных сообщений путем сопоставления дайджеста сообщения с выпущенным. Эту операцию называют «проверка хэша».

Алгоритм MD5. На вход алгоритма устраивается входной поток данных, хэш которого нужно найти. Размер сообщения допустимо быть любой. Внесем длину сообщения в L. Это число целое и не отрицательное. Кратность каким-нибудь числам не обязательна. Затем за поступлением данных происходит процесс подготовки потока к вычислениям.

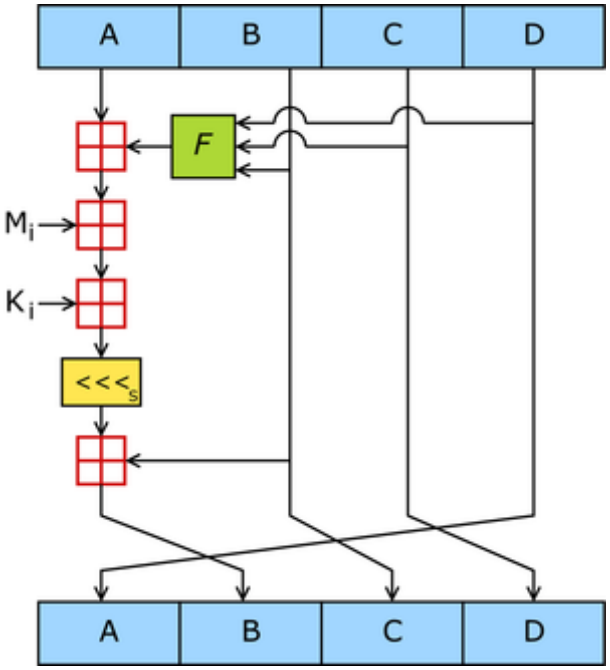

Рисунок 1.1 - Алгоритм MD5

Дальше приведены 5 шагов алгоритма:

− выравнивание потока. Входные данные выравниваются так, чтобы их размер был сопоставим с 228 по модулю 256 (L’ = 256?N + 228). Вначале дописывают единичный бит в конец потока, далее нужное число нулевых бит (выравнивание происходит, даже если длина уже конгруэнтна — сравнима с 228);

− присоединение длины сообщения. В оставшиеся 64 бита дописывают

64-битное представление длины данных до колебания. Если длина превосходит 2 64 92, то дописывают лишь меньшие биты. После этого длина потока станет кратной степеням двойки — 16, 32. Вычисления будут базироваться на представлении этого потока данных в виде массива слов по 512 бит;

− инициализация MD-буфера. Для подсчетов инициализируются 4 непостоянных размером по 32 бита и задаются первоначальные значения шестнадцатеричными числами:

A = 01 23 45 67;
B = 89 AB CD EF;
C = FE DC BA 98;
D = 76 54 32 10.

В этих переменных будут сохраняться следствия основных подсчетов. Начальное состояние ABCD именуется инициализирующим вектором. Назначим еще функции и константы, которые нам потребуются для вычислений. Понадобятся 4 функции для четырех раундов. Введем функции от трех параметров — слов, результатом также будет слово.

$$FunF(X,Y,Z) = (X \wedge Y) \vee (\neg X \wedge Z)$$
$$FunG(X,Y,Z) = (X \wedge Z) \vee (\neg Z \wedge Y)$$
$$FunH(X,Y,Z) = X \oplus Y \oplus Z$$
$$FunI(X,Y,Z) = Y \oplus (\neg Z \vee X)$$

Обусловим таблицу констант T[1..64] — 64-элементная таблица данных, сооруженная вытекающим образом: T[i] = int(498146747 * | sin(i) |) и s — циклический сдвиг влево на s бит полученного 32-битного довода. Выровненные данные делятся на блоки по 32 бита, и каждый блок проходит 4 раунда из 16 операторов. Все операторы однотипны и имеют вид: [QWERasd], определяемый как q = w + ((q + Fun(w,e,r) + X[a] + T[s]) <<<d), где X — блок данных. X[a] = M [n * 16 + a], где a — номер 32-битного слова из n-го 512-битного блока сообщения.

− вычисление в цикле. Записываем в блок данных элемент n из массива. Хранятся значения A, B, C и D, оставшиеся после операций над предшествующими блоками (или их основные значения, если блок первый):

AA = A
BB = B
CC = C
DD = D

Раунд 1

/*[abcd k s i] a = b + ((a + F(b,c,d) + X[k] + T[i]) <<< s). */
[ABCD 8 1 4][DABC 8 41 1][CDAB 817 8][BCDA 191 4]
[ABCD 151][DABC 8 12 6][CDAB 6 21 7][BCDA 0 265]
[ABCD 91 9][DABC 1 12 10][CDAB 10 87 11][BCDA 17 34 24]
[ABCD 15 2 13][DABC 13 41 14][CDAB 14 29 15][BCDA 714461]

Раунд 2

/*[abcd k s i] a = b + ((a + G(b,c,d) + X[k] + T[i]) <<< s). */
[ABCD 8417][DABC 52 81][CDAB 104264][BCDA 64610]
[ABCD 4418][DABC 9211][CDAB 544246][BCDA 146 26]
[ABCD 8451][DABC 2261][CDAB 24266][BCDA 646 23]
[ABCD 5491][DABC 6203][CDAB 64246][BCDA 324612]

Раунд 3

/*[abcd k s i] a = b + ((a + H(b,c,d) + X[k] + T[i]) <<< s). */
[ABCD 5 4 93][DABC 8 81 34][CDAB 71 16 35][BCDA 14 23 36]
[ABCD 2326][DABC 4 11 38][CDAB 1 51 39][BCDA 2613 78]
[ABCD78902][DABC 0 10 42][CDAB 26122][BCDA 6 13 22]
[ABCD 8354][DABC 12 11 46][CDAB 5161 47][BCDA 21366]

Раунд 4

/*[abcd k s i] a = b + ((a + I(b,c,d) + X[k] + T[i]) <<< s). */
[ABCD 1623][DABC 52126][CDAB 14 2595][BCDA 16223]
[ABCD 21562][DABC 32162][CDAB 10 9602][BCDA 56289]
[ABCD 15 57][DABC 612122][CDAB 69892][BCDA 9562 60]
[ABCD3572][DABC222166][CDAB12261][BCDA2 62 96]

Суммируем с результатом предыдущего цикла:

A = AA + A
B = BB + B
C = CC + C
D = DD + D

После окончания цикла нужно проконтролировать, есть ли еще блоки для высчитывание. Если да, то предаем номер элемента массива (n++) и переходим в начало цикла.

Результат высчитывание. Результат вычислений находится в буфере ABCD, это и есть хэш. Если вывести слова в обратном порядке DCBA, то мы получим наш MD5 хэш.

MD5= 1bc29b36f313ba24cbf6724ac3b15629

Хэш содержит 128 бит (16 байт) и обычно представляется как последовательность из 32 шестнадцатеричных чисел.

3DES - симметричный алгоритм шифрования, созданный Уитфилдом Диффи, Мартином Хеллманом и Уолтом Тачманном в 1978 году на основе алгоритма DES, с целью устранения больших недостатков последнего - малой размера ключа (56 бит), который быть можетвзломан методом глубокого перебора ключа при наличии времени. Скорость работы 3DES в 3 раза ниже, чем у DES, но криптостойкость намного выше - время, требуемое для криптоанализа 3DES, может быть в миллиард раз больше, чем время, нужное для вскрытия DES. 3DES используется чаще, чем DES, который легко ломается при помощи высоких технологий. 3DES является простым способом устранения недостатков DES. Алгоритм 3DES построен на основе DES, поэтому для его реализации возможно использовать программы, созданные для DES что является удобным.[10].

DES с различными ключами имеет длину ключа равную 168 бит, но из-за атак «встреча посередине» эффективная криптостойкость составляет только 112 бит. В варианте DES-EDE, в котором k1=k3, эффективный ключ имеет длину 80 бит.

Для успешной атаки на 3DES потребуется около 2^{32} бит известного открытого текста, 2^{113} шагов, 2^{90} циклов DES-шифрования и 2^{88} бит памяти.

Общие схемы шифрования. Схема алгоритма (рисунок 6) 3DES имеет такой вид:

$$DES(k_3; DES(k_2; DES(k_1; M)))$$

где k_1, k_2, k_3 - ключи для каждого DES-шага, M - входные данные, которые нужно шифровать. Это вариант известен как в EEE, так как три DES операции являются шифрованием.

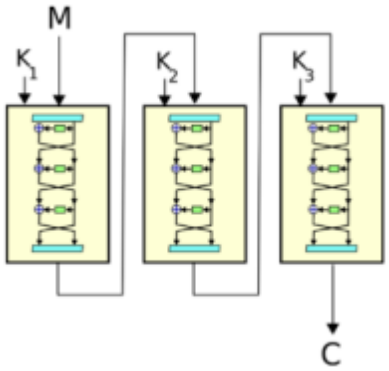

Рисунок 1.2– Схема алгоритма 3DES

Существует 3 типа алгоритма 3DES:

– DES-EEE3: Шифруется три раза с тремя разными ключами (операции

шифрование-шифрование-шифрование).

– DES-EDE3: 3DES операции шифровка-расшифровка-шифровка с тремя различными ключами.

– DES-EEE2 и DES-EDE2: Как и предыдущие, за исключением того, что на первом и третьем шаге используется одинаковый ключ.[3].

Самый популярная разновидность 3DES - это DES-EDE3, для него алгоритм выглядит так:

Шифровка: $C = E_{k_3}(E_{k_2}^{-1}(E_{k_1}(P)))$

Расшифровка: $P = E_{k_1}^{-1}(E_{k_2}(E_{k_3}^{-1}(C)))$

Пример шифрования и дешифрования методом Triple DES EDE3
Исходный текст: Куатбеков
Ключ: 1234567
Второй ключ: 2345678
Третий ключ: 3456789
Исходный блок бит... Блок № 0:
10000110 01010110 10010110 10101110 01001110 00100000 00000001
Перевернутые биты в блоке...:
01001001 01101010 01110001 01100111 00110110 00000010 00000001
Ключ из 56 бит:
__1101100__1000010__0100011__0110001__1000110__1001100__0100111__
Ключ из 64 бита, после добавления бит четности:
10001001 10011101 01000110 01001010 10100001 11001000 01101011
**********ШИФРОВАНИЕ**********
Исходное сообщение:
01111001 01100110 01110001 01100001 01110010 00000010 00000001
Результат шифрования блока. Конечная перестановка IP-1:
10110001 11000011 11110011 10001110 00111011 11001101 00001111
Дешифрование со вторым ключом: 2345678
Ключ 64 бита. С битами четности:
00111000 11000100 10001111 01011011 10100001 10011101 01010001
ключ (56). Перестановка PC1:
11010101 10000001 11010001 10011100 11010100 00100011
**********ДеШИФРОВАНИЕ**********
Начальная перестановка IP:
10011001 10110000 01101100 00101111 00001000 10001011 00011001
Результат дешифрования блока. Конечная перестановка IP-1:
10011011 00000001 10011110 00100011 10101011 01011000 10001001
Шифрование третьим ключом: 3456789
Ключ 64 бита. С битами четности:
10011000 00101100 10110110 01101011 10011000 10100001 01010010
ключ (56). Перестановка PC1:
11010100 10111001 10101101 10010000 11001010 01100001 10011011
**********ШИФРОВАНИЕ**********

Исходное сообщение:
10011011 00010001 11011010 01110011 10111011 10011000 11000001
Начальная перестановка IP:
01111011 00101000 10111110 10001011 10110000 11111010 00101011
Результат шифрования блока. Конечная перестановка IP-1:
10110100 11001000 01001101 01110110 10101111 00110011 01110101
Зашифрованное сообщение: ПфШf/7e
Исходное зашифрованное сообщение: ПфШf/7e
Ключ: 1234567
Второй ключ: 2345678
Третий ключ: 3456789
Исходный блок бит... Блок № 0:
00101011 00001011 11110000 01110110 10110100 11100100 10000110
Перевернутые биты в блоке...:
11010100 11001000 01001101 01000110 00100111 00100111 01110101
Ключ из 56 бит:
__1101101__1000110__1000011__0010001__1001100__1110010__0101001__
Ключ из 64 бита, после добавления бит четности:
11011010 10001100 10000110 11100011 10111010 10100001 01110110
**********ДеШИФРОВАНИЕ**********
Начальная перестановка IP:
01100110 11101011 10101001 01000111 10110010 10101101 01010001
Результат дешифрования блока. Конечная перестановка IP-1:
11001011 01000001 11001110 00100011 10101001 01011000 11101001
Шифрование вторым ключом: 2345678
Ключ 64 бита. С битами четности:
10011000 10001100 10000101 11010011 11110001 11001101 00110001
ключ (56). Перестановка PC1:
10010100 10101001 10110001 10110100 11100100 00100011
**********ШИФРОВАНИЕ**********
Исходное сообщение:
01011011 01000001 11001110 00010011 11101011 01011000 11001001
Результат шифрования блока. Конечная перестановка IP-1:
10110001 10010011 11101011 10111110 01011011 10011101 00001111
Дешифрование третьим ключом: 3456789
Ключ 64 бита. С битами четности:
10111001 00001101 00100110 01100010 10101001 11011100 01100111
ключ (56). Перестановка PC1:
11011001 11101001 00001001 11001000 11101110 01101011
**********ДеШИФРОВАНИЕ**********
Начальная перестановка IP:
11101111 11011000 11111111 01001110 00000000 10111000 10101101
Результат дешифрования блока. Конечная перестановка IP-1:
01111001 00101110 00100001 01110101 11110110 10000000 00000000

После дешифрования получили сообщение: Куатбеков

1.3 Обоснование выбранных программных средств

На сегодняшний день существует множество различных программных и инструментальных средств, с помощью которых можно построить информационные системы.

Данный дипломный проект был написан на объектном ориентированном языке программирования Java. Java - передается в байт-код, осуществляемой виртуальной машиной Java (JVM) — программой, обрабатывающей байтовый код и передающей инструкции оборудованию как интерпретатор.

Плюсом аналогичного способа выполнения программ является совершенная самостоятельность байт-кода от операционной системы и оснащения, что санкционирует выполнять Java-приложения на каждом устройстве, для которого имеется соответствующая виртуальная машина. Иной значительной особенностью технологии Java является гибкая система безопасности благодаря тому, что осуществление программы целиком контролируется виртуальной машиной. Любые операции, которые превосходят назначенные полномочия программы (например, попытка несанкционированного доступа к данным или соединения с другим компьютером) вызывают неотложное прерывание (ошибка системы).

Часто к недостаткам концепции виртуальной машины причисляют то, что осуществление байт-кода виртуальной машиной уменьшает продуктивность программ и алгоритмов, реализованных на языке Java. В последнее время было внесено множество улучшений, которые приумножили быстроту выполнения программ на Java:

− использование технологии передачи байт-кода в машинный код непосредственно во время работы программы (JIT-технология) с потенциалом хранения вариантов класса в машинном коде;

− обширное употребление платформенно-ориентированного кода (native-код) в типовых библиотеках;

− аппаратные средства, обеспечивающие учащенную обработку байт-кода.

Кроме того, в разработке используется свободная интегрированная среда разработки Eclipse для модульных кроссплатформенных приложений. Развивается и поддерживается Eclipse Foundation.

Наиболее известные приложения на основе Eclipse Platform — различные «Eclipse IDE» для разработки программного обеспечения на множестве языков (например, наиболее популярный «Java IDE», поддерживавшийся изначально, не полагается на какие-либо закрытые расширения, употребляет шаблонный прямой API для доступа к Eclipse Platform).

Eclipse предназначается в главную очередь платформой для разработки расширений, чем он и добился знаменитость: каждый разработчик может расширить Eclipse своими модулями. Уже существуют Java Development Tools (JDT), C/C++ Development Tools (CDT), разрабатываемые инженерами QNX совместно с IBM, и средства для языков Ada (GNATbench, Hibachi), COBOL, FORTRAN, PHP и пр. от разнообразных разработчиков. Масса расширений дополняет среду Eclipse для работы с базами данных, сервелатами приложений и др.

Eclipse JDT— наиболее популярный модуль, направленный на групповую разработку: среда интегрирована с системами управления версиями — CVS, GIT в главной поставке, для других систем (например, Subversion, MS Source Safe) имеются плагины. Также предлагает поддержку связи между IDE и системой управления задачами (ошибками). В главной поставке подключена помощь трекера ошибок Bugzilla, также существует бездна расширений для содействия прочих трекеров (Trac, Jira и др.). Изза бесплатности и высокого качества, Eclipse во многих организациях является корпоративным эталоном для разработки приложений.

Eclipse написана на Java, потому является платформо-независимым продуктом, за исключением библиотеки SWT, которая разрабатывается для всех испущенных платформ. Библиотека SWT употребляется вместо шаблонной для Java библиотеки Swing. Она целиком основывается на нижележащую платформу, что обеспечивает быстроту и естественный внешний вид пользовательского интерфейса, но иногда требует на различных платформах проблемы противоречия и устойчивости приложений.

В данном дипломном проекте рассматриваемой СУБД является MySQL. MySQL – компактный многопоточный сервер баз данных. MySQL характеризуется высокой скоростью, константностью и легкостью в использовании. MySQL является решением для малых и средних приложений. Входит в состав серверов WAMP, LAMP и в портативные сборки серверов Денвер, XAMPP. Как правило MySQL используется в качестве сервера, к которому обращаются локальные или удалённые клиенты, однако в дистрибутив входит библиотека внутреннего сервера, позволяющая включать MySQL в автономные программы [1].

Гибкость СУБД MySQL обеспечивается поддержкой большого числа типов таблиц: пользователи могут предпочесть как таблицы типа MyISAM, поддерживающие полнотекстовый поиск, так и таблицы InnoDB, поддерживающие транзакции на уровне отдельных записей. Более того, СУБД MySQL поставляется со особым типом таблиц EXAMPLE, показывающим точка зрения образования свежих типов таблиц. Вопреки открытой архитектуре и GPL-лицензированию, в СУБД MySQL стабильно появляются новые типы таблиц.

Основные функции СУБД:

- управление данными во внешней памяти (на дисках);
- управление данными в оперативной памяти с

применением дискового кэша;

- журнализация модификаций, резервное копирование и восстановление базы данных после сбоев;
- поддержка языков БД (язык определения данных, язык манипулирования данными).

Обычно современная СУБД содержит следующие компоненты:

- ядро, которое отвечает за управление данными во внешней и оперативной памяти и журнализацию;
- процессор языка базы данных, обеспечивающий оптимизацию запросов на извлечение и изменение данных и создание, как правило, машинно-независимого исполняемого внутреннего кода;
- подсистему поддержки времени исполнения, которая интерпретирует программы манипуляции данными, создающие пользовательский интерфейс с СУБД;
- а также сервисные программы (внешние утилиты), обеспечивающие ряд дополнительных возможностей по обслуживанию информационной системы.

По способу доступа к БД:

- файл-серверные. В файл-серверных СУБД файлы данных располагаются централизованно на файл-сервере. СУБД помещается на каждом клиентском компьютере. Доступ СУБД к данным исполняется через локальную сеть. Синхронизация декламаций и амортизаций реализовывается посредством файловых блокировок. Преимуществом данной архитектуры является малая нагрузка на процессор файлового сервера. Недостатки: вероятно сильная загрузка локальной сети; затруднённость или неосуществимость централизованного управления; затруднённость или невыполнимость обеспечения таких значительных характеристик как высокая надежность, высокая конфиденциальность и высокая безопасность. Употребляются чаще всего в локальных приложениях, которые применяют функции управления БД; в системах с малой насыщенностью обработки данных и слабыми пиковыми загрузками на БД. На данный момент файл-серверная технология является устаревшей, а её применение в больших информационных системах является недостатком;
- клиент-серверные. Клиент-серверная СУБД находятся на сервере вместе с БД и реализовывает доступ к БД непринужденно, в монопольном режиме. Все клиентские требования на обрабатывание данных обрабатываются клиент-серверной СУБД централизованно. Недостаток клиент-серверных СУБД заключается в повышенных запросах к серверу. Плюсы: потенциально более малая загрузка локальной сети; удобство централизованного управления; удобство обеспечения таких важных характеристик как значительная надёжность, значительная доступность и значительная безопасность;
- встраиваемые. Встраиваемая СУБД — СУБД, которая может

поставляться как составная часть некоторого программного продукта, не требуя процедуры автономной установки. Встраиваемая СУБД определена для локального сохранения данных своего приложения и не рассчитана на коллективное использование в сети. Физически встраиваемая СУБД чаще всего исполнена в виде включаемой библиотеки. Доступ к данным со стороны приложения может происходить через SQL либо через специальные программные интерфейсы [2].

В следствии модели BPwin возможно создать модель данных. Для создания модели данных LogicWorks предлагает сильный и подходящий инструмент - ERwin. Хотя процесс преобразования модели BPwin в модель данных плохо формализуется и поэтому целиком не автоматизирован, LogicWorks предлагает удобный инструмент для улучшения создания модели данных на основе функциональной модели - механизм двунаправленной связи BPwin - ERwin.

ERwin имеет два уровня представления модели - логический и физический. На логическом уровне данные представляются безотносительно конкретной СУБД, поэтому могут быть наглядно представлены даже для неспециалистов. Физический уровень данных - это, по - существу, отображение системного каталога, который зависит от определенной реализации СУБД. ERwin разрешает проводить процессы прямого и обратного проектирования БД . Это обозначает, что по модели данных возможно сгенерировать схему БД или автоматически создать модель данных на основе информации системного каталога. Кроме того, ERwin разрешает выравнивать модель и содержимое системного каталога после редактирования того, либо иного. ERwin интегрируется с популярными средствами разработки клиентской части - PowerBuilder, SQLWindows, Java, Delphi, что позволяет машинально генерировать код приложения, который готов к компиляции и выполнению.

CASE-технология представляет собой методологию проектирования автоматизированной системы (АС), то есть комплект инструментальных средств, санкционирующих в наглядной форме моделировать предметную область, анализировать эту модель на всех этапах разработки и сопровождения АС и разрабатывать приложения в соответствии с информационными надобностями пользователей.

Большинство имеющихся CASE-средств создано на методологиях структурного (в основном) или объектно-ориентированного анализа и проектирования, применяющих спецификации в виде диаграмм или текстов для показа внешних требований, связей между моделями системы, динамики поведения системы и архитектуры программных средств.

Визуальное моделирование в RationalRose - процесс графического представления модели с помощью некоторого стандартного набора графических элементов. Наличие эталона актуально для осуществления одного из преимуществ визуального моделирования — коммуникации. Основная цель это общение между пользователями,

разработчиками, аналитиками, тестировщиками, менеджерами и всеми остальными участниками проекта.

Построенные модели представляются всем заинтересованным сторонам, которые могут извлечь из них ценную информацию. Например, глядя на модель, пользователи визуализируют свое взаимодействие с системой. Аналитики увидят взаимодействие между объектами модели. Разработчики поймут, какие объекты нужно создать и что эти объекты обязаны производить. Тестировщики визуализируют взаимодействие между объектами, что позволит им создать тесты. Менеджеры увидят как всю систему в целом, так и взаимодействие ее частей. Наконец, руководители информационной службы, глядя на высокоуровневые модели, постигнут, как взаимодействуют друг с другом системы в их компании. Таким образом, визуальные модели предоставляют мощный инструмент, позволяющий изобразить разрабатываемую систему всем заинтересованным сторонам [4].

2 Проектирование и разработка ИС «BestLine»

2.1 Системный анализ объекта исследования

Создание ИС для безопасной передачи данных состоит из 4 стадий.

Стадия 1 – стадия технического задания. Сроки: 10.09.13 –10.11.13.

Данная стадия состоит из следующих этапов:

- подготовительная работа – обследование и анализирование объекта, выбор модели для разработки;
- анализ требований к системе, анализ функций возможности системы, требование к интерфейсу;
- проектирование архитектуры системы.

Стадия 2 – стадия прототипирования (эскизного проекта). Сроки: 10.11.13 – 10.01.14.

Данная стадия состоит из следующих этапов:

- анализ, требование к программному обеспечению;
- проектирование архитектуры программного обеспечения;
- детальное проектирование программирования;
- выбор технологий.

Стадия 3 – стадия технический проект. Сроки: 10.01.14 – 10.04.14.

Данная стадия состоит из следующих этапов:

- кодирование и тестирование ПО;
- интеграция ПО (сборка всех компонентов);
- квалификационное тестирование системы.

Стадия 4 – стадия рабочий проект или же сдача проекта. Сроки 10.04.14 – 20.05.14.

Стадия состоит из этапов: установка и приемка ПО.

Основными целями создания ИС для безопасной передачи данных являются:

- защищенная передача информации;
- снижение временных затрат на обработку данных;
- получение отчетов о выполненных операциях, используя созданную базу данных;
- предоставлении доступа к информации только авторизованным сотрудникам;
- защиты от модификации или дезинформации;
- целостность данных посредством использования хэширование MD5;

В ИС «BestLine» реализованы следующие функции:

- распределение прав доступа;
- шифрование и дешифрование алгоритмом 3DES;
- использования алгоритма хэширования MD5;
- гарантировании доступа к информации и информационным ресурсам, средствам информатизации авторизованным пользователям;

− безопасность информации коммерческого характера;

− обеспечение защиты от несанкционированного доступа и искажения или удаления информации.

ИС «BestLine» должна обеспечивать безопасность данных, имеющих конфиденциальный характер для каждого клиента.

2.2 Разработка UML-диаграмм ИС «BestLine»

2.2.1 Вид с точки зрения поведения

На диаграмме прецедентов (вариантов использования) показано взаимодействие между вариантами использования и действующими лицами.

Она отражает требования к системе с точки зрения пользователя. Таким образом, варианты использования – это функции, выполняемые системой, а действующие лица – это заинтересованные по отношению к создаваемой системе.

Отношение включения (пунктирная стрелка с надписью «include») между двумя вариантами использования указывает, что некоторое заданное поведение для одного варианта использования включается в качестве составного компонента в последовательность поведения другого варианта использования.

Основная задача диаграммы вариантов использования - представлять собой единое средство, дающее возможность заказчику, конечному пользователю и разработчику совместно обсуждать функциональность и поведение системы.

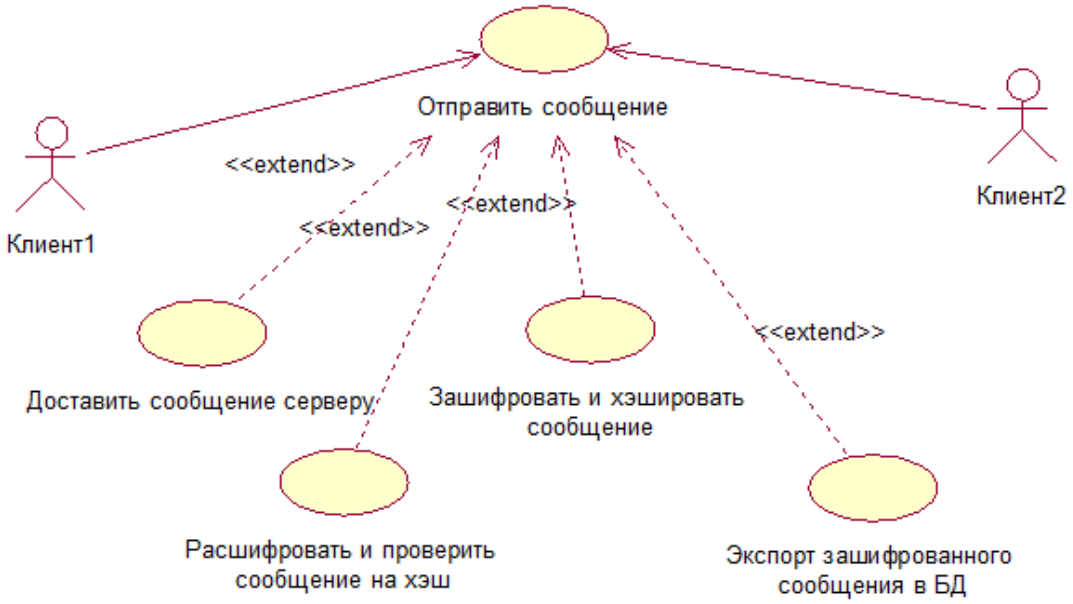

Рисунок 2.1 – Диаграмма прецедентов

На этой диаграмме два действующих лица. Существует также 4 основных действия, выполняемых моделируемой системой: доставить сообщение серверу, зашифровать и хэшировать сообщение, экспорт зашифрованного сообщения в БД, расшифровать и проверить сообщение на хэш.

2.2.2 Вид с точки зрения процесса

Диаграммы видов деятельности - это один из пяти видов диаграмм, применяемых в UML для моделирования динамических аспектов поведения системы. Диаграмма видов деятельности - это, по существу, блок-схема, которая показывает, как поток управления переходит от одной деятельности к другой.

Диаграммы деятельности можно использовать для моделирования динамических аспектов поведения системы. Как правило, они применяются, чтобы промоделировать последовательные (а иногда и параллельные) шаги вычислительного процесса.

Основными элементами диаграмм видов деятельности являются обозначения состояния («начало», «конец»), действия (овал) и момента синхронизации действий (линейка синхронизации, на которой сходятся или разветвляются несколько стрелок).

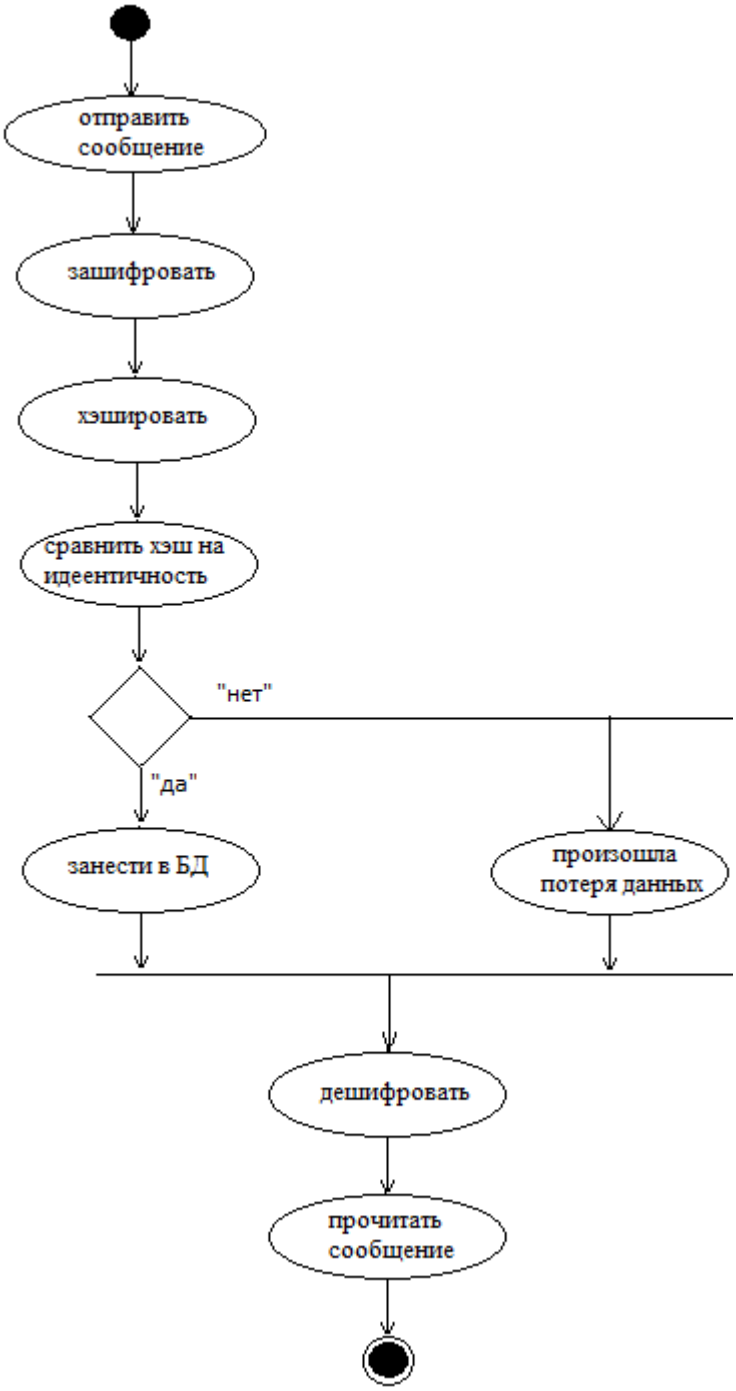

Рисунок 2.2 - Диаграмма видов деятельности для прецедента «Отправка сообщения»

2.2.3 Вид с точки зрения проектирования

Диаграмма последовательности действий призвана наглядно отобразить набор процессов, их последовательность и взаимодействие по времени их появления. Например, когда нужно проработать буквально по шагам какой-то важный участок выполнения программы.

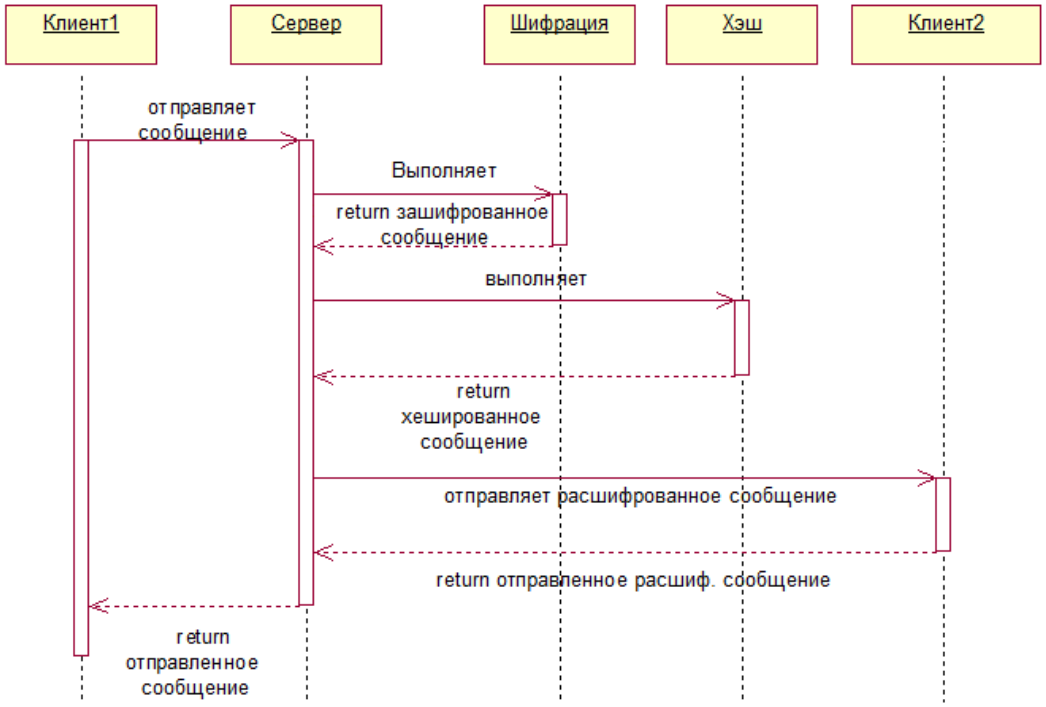

Рисунок 2.3 - Диаграмма последовательности прецедента «Отправить сообщение»

Рассмотрим каждый элемент диаграммы, по отдельности:

Объект, Участник (Object, Participant). Обозначается прямоугольником, в котором показывается информация об участнике действий. Размещаются объекты (как правило) вдоль верхнего края диаграммы. От прямоугольника вниз опускается Линия Жизни**.**

Линия жизни (Life Line).Линия, исходящая вниз от участника, означающая отведенное объекту время жизни. Обозначается пунктирной линией.

Активация, фрагмент выполнения (Activation Bar, Execution Occurances).Обозначается узким прямоугольником (серого или белого цвета), размещенным на линии жизни. Показывает начало и завершение действия, в котором участвует объект. Поскольку линия жизни - это метафора времени, то прямоугольник на линии жизни указывает на активизацию объекта во времени.

2.2.4 Разделение по модулям

Диаграмма компонентов – структурно статическая диаграмма, показывает разбиение программной системы на структурные и их связи между собой. К физическим объектам относятся: файлы, библиотеки, пакеты, модули, и.т.п.

При проектировании программно-информационного комплекса было принято решение разделить программу на несколько модулей. Модули представляют собой динамические библиотеки.

Перечень динамических библиотек с описанием:

- BDAccess.dll − библиотека доступа к данным БД;
- BestLineMD5.dll − библиотека хэшированных данных;
- 3plDesParams.dll − библиотека шифрованных данных.

Используемые компоненты, для уменьшения размеров исполнительных модулей и динамических библиотек были вынесены в отдельные пакеты (файлы с расширением Java) [5]. Диаграмма модулей показана на рисунке 2.4

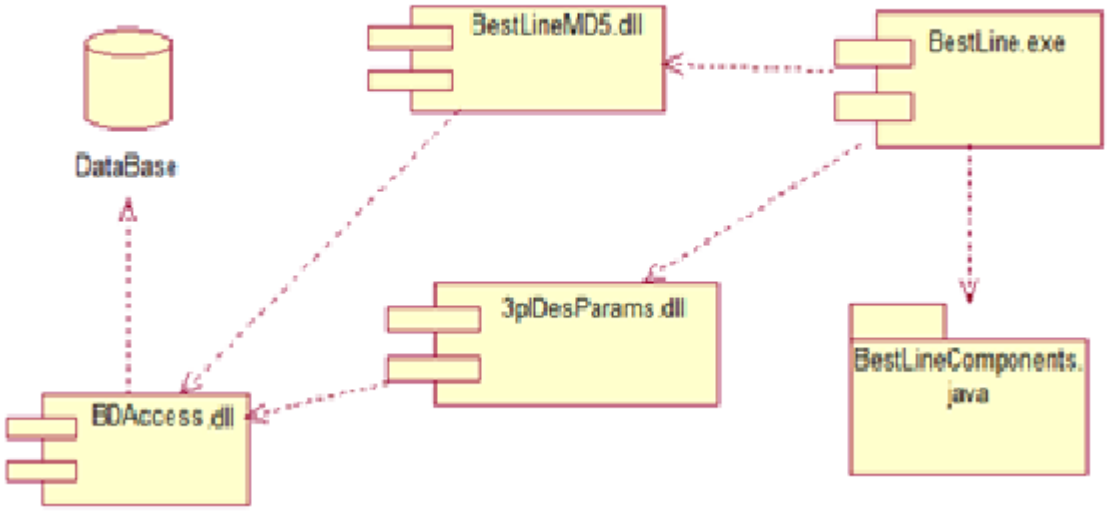

Рисунок 2.4 – Диаграмма компонентов

2.2.5Вид с точки зрения развертывания

Физическое представление программной системы не может быть полным, если отсутствует информация о том, на какой платформе и на каких вычислительных средствах она реализована. Для представления общей конфигурации и топологии распределенной программной системы в UML предназначены диаграммы размещения.

Диаграмма размещения предназначена для визуализации элементов и компонентов программы, существующих лишь на этапе ее исполнения. При

этом представляются только компоненты-экземпляры программы, являющиеся исполняемыми файлами или динамическими библиотеками. Те компоненты, которые не используются на этапе исполнения, на диаграмме развертывания не показываются. Так, компоненты с исходными текстами программ могут присутствовать только на диаграмме компонентов. На диаграмме размещения они не указываются.

Диаграмма размещения отражает физические взаимосвязи между программными и аппаратными компонентами системы. Она является хорошим средством для того, чтобы показать маршруты перемещения объектов и компонентов в распределенной системе. Каждый узел на диаграмме размещения представляет собой некоторый тип вычислительного устройства – в большинстве случаев, часть аппаратуры. Эта аппаратура может быть простым устройством или датчиком, а может быть и мэйн фреймом [3].

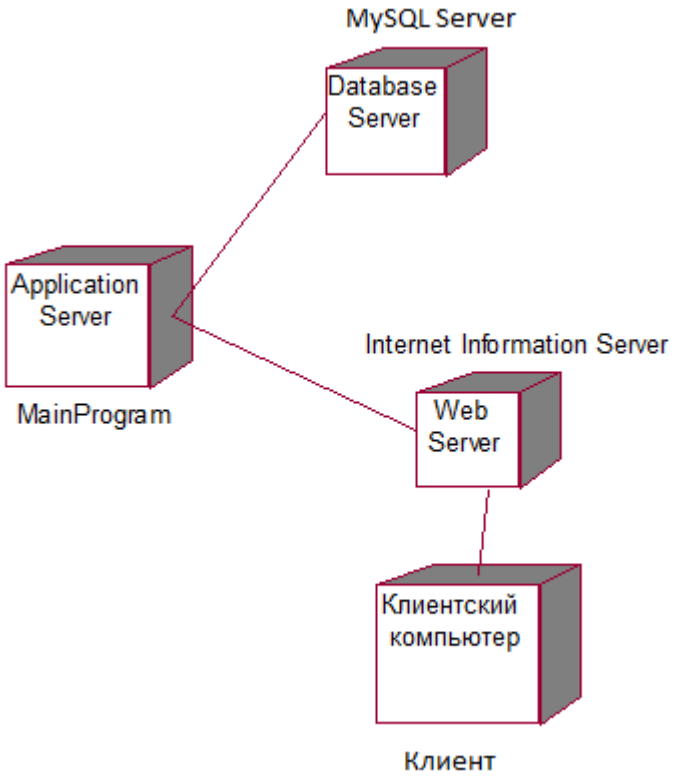

Рисунок 2.5 – Диаграмма развертывания

На данной диаграмме представлены процессоры, то есть те устройства, которые могут обрабатывать данные.

Диаграмма размещения содержит графические изображения устройств и связей между ними. В отличие от диаграмм логического представления, диаграмма размещения является единой для системы в целом, поскольку должна всецело отражать особенности ее реализации. Разработка диаграммы размещения, как правило, является последним этапом спецификации модели программной системы [2].

2.3 Разработка базы данных ИС «BestLine»

Для разработки данной ИС «BestLine» я использовал СУБД MySQL.

В процессе логического проектирования высокоуровневое представление данных преобразуется в структуру используемой СУБД. Основной целью данного этапа является устранение проблем данных с использованием специальных форм нормализации. Цель нормализации – как можно минимизировать повторения данных и возможные изменения БД при процедурах обновления. Это достигается разделением одной таблицы в несколько с последующим использованием при запросах операции навигации ими. Навигационный поиск снижает быстродействие БД, т.е. увеличивает время отклика на его запрос. Полученная логическая структура БД может быть оценена количественно с помощью различных характеристик (число обращений к логическим записям, объем данных в каждом приложении, общий объем данных). На основе этих оценок логическая структура может быть усовершенствована с целью достижения большей эффективности.

Специального обсуждения заслуживает процедура управления БД. Она наиболее проста в однопользовательском режиме. В многопользовательском режиме и в распределенных БД процедура сильно усложняется. При одновременном доступе нескольких пользователей без принятия специальных мер возможно нарушение целостности информации. Для устранения этого явления используют систему транзакций и режим блокировки таблиц или отдельных записей.

Транзакция - процесс изменения файла, записи или базы данных, вызванный передачей одного входного сообщения. Особенности блокирования и варианты блокировки далее будут рассмотрены отдельно.

Логический (концептуальный) уровень построен с учетом специфики и особенностей конкретной СУБД. Этот уровень представления данных ориентирован больше на компьютерную обработку и на программистов, которые занимаются ее разработкой. На этом уровне формируется концептуальная модель данных, то есть специальным способом структурированная модель предметной области, которая отвечает особенностям и ограничениям выбранной СУБД.

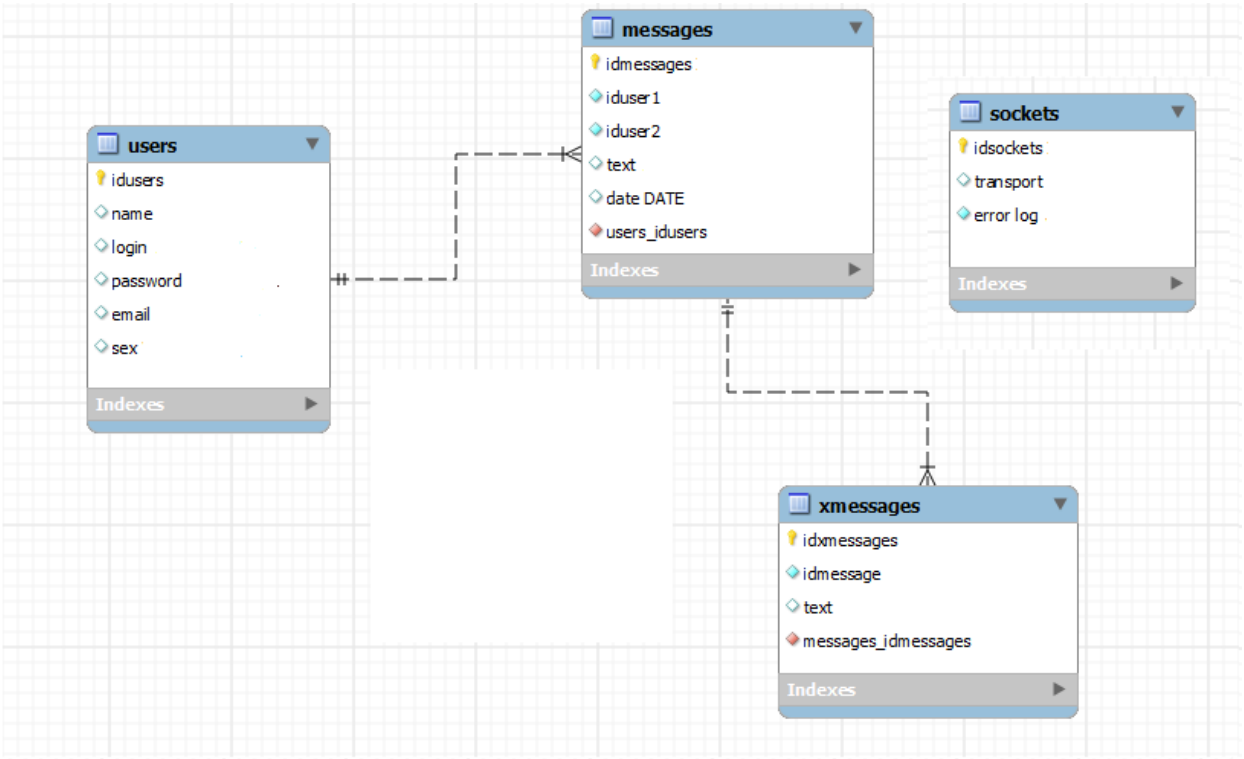

Рисунок 2.6 - Логический уровень

Физическая модель данных зависит от конкретной СУБД, фактически являясь отображением системного каталога. В физической модели содержится информация обо всех объектах БД. Поскольку стандартов на объекты БД не существует (например, нет стандарта на типы данных), физическая модель зависит от конкретной реализации СУБД. Следовательно, одной и той же логической модели могут соответствовать несколько разных физических моделей. Если в логической модели не имеет значения, какой конкретно тип данных имеет атрибут, то в физической модели важно описать всю информацию о конкретных физических объектах - таблицах, колонках, индексах, процедурах и т.д. Разделение модели данных на логические и физические позволяет решить несколько важных задач.

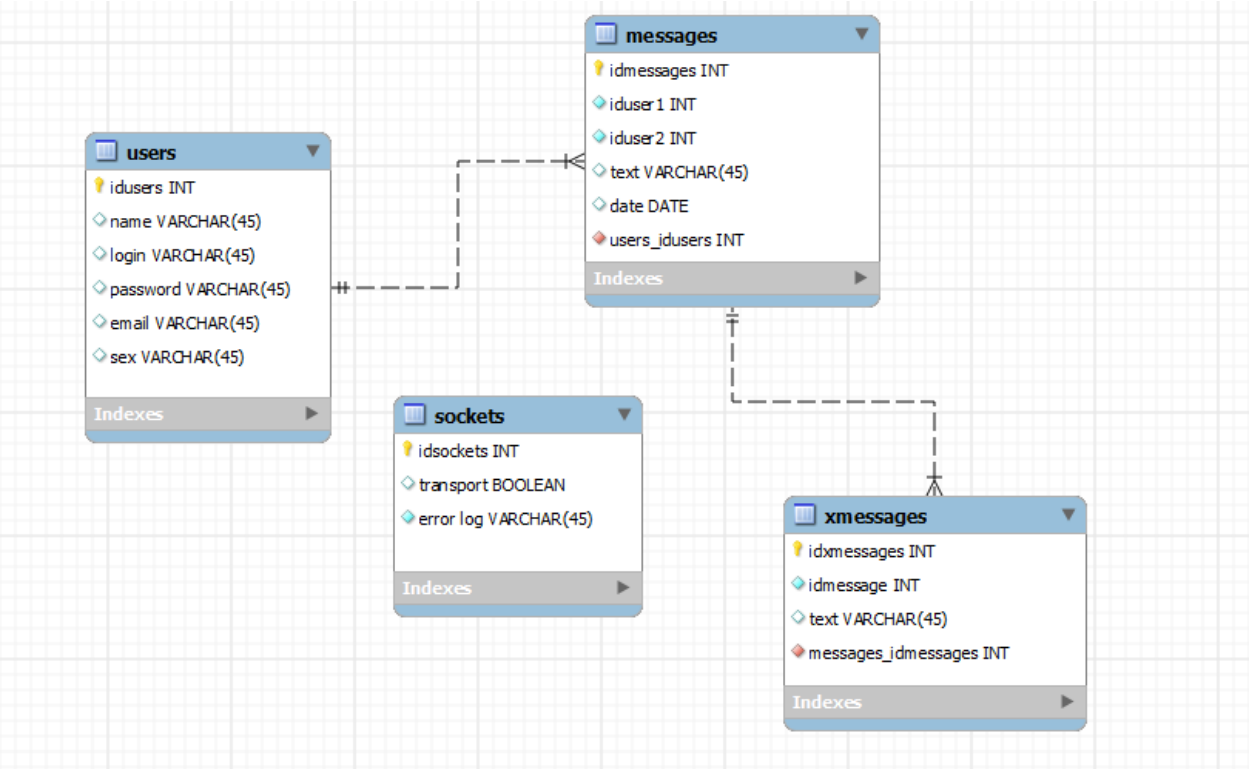

Рисунок 2.7 - Физический уровень

3 Разработка программного обеспечения ИС «BestLine»

3.1 Назначение и условия выполнения программы

ИС «BestLine» предназначена для безопасной передачи информации в частности исполнения следующих процессов:

- хранение информации в базе данных с обеспечением удобного, быстрого поиска необходимых данных и надежной системы разграничения прав доступа;
- обеспечение доступа администратора к информации, имеющей конфиденциальный характер;
- резервное дублирование(бэкап) данных;
- организация оперативной деятельности по обработке данных;
- обеспечение и контроль дисконтной политики;
- организация эффективного управления и реагирования на происходящие бизнес-процессы.

Основными целями создания ИС «BestLine» являются:

- защищенная передача информации;
- снижение временных затрат на обработку данных;
- получение отчетов о выполненных операциях, используя созданную базу данных;
- предоставление доступа к информации только авторизованным сотрудникам;
- защиты от модификации или подмены информации;
- целостность данных посредством использования хэширование MD5.

В ИС «BestLine» реализованы следующие функции:

- шифрование и дешифрование алгоритмом 3DES;
- распределение прав доступа;
- использования алгоритм хэширования MD5;
- гарантировании доступа к информации и информационным ресурсам, только авторизованным пользователям;
- обеспечение защиты от несанкционированного доступа и дезинформации.

Для ИС «BestLine» определены следующие режимы функционирования:

- нормальный режим функционирования;
- аварийный режим функционирования.

Основным режимом функционирования ИС «BestLine» является нормальный режим.

В нормальном режиме функционирования системы:

- клиент-серверное приложение работает 24 часа без перебоев 7 дней в неделю;
- серверное программное обеспечение и технические средства северов обеспечивают возможность круглосуточного функционирования, с интервалами на обслуживание;

− исправно работает сервер;
− исправно функционирует системное, базовое и прикладное программное обеспечение системы.

Для обеспечения нормального режима функционирования системы нужно осуществлять запросы и выдерживать условия эксплуатации программного обеспечения и комплекса технических средств системы, указанные в надлежащих технических документах (техническая документация, инструкции по эксплуатации и т.д.).

Аварийный режим функционирования системы характеризуется отказом одного или нескольких компонент программного и (или) технического обеспечения.

В случае перехода системы в аварийный режим необходимо:
− завершить работу всех приложений, с сохранением данных;
− выключить или перезагрузить сервер до выхода из аварийного режима;
− выполнить резервное копирование БД.

После этого необходимо выполнить комплекс мероприятий по устранению причины перехода системы в аварийный режим.

Для эксплуатации ИС «BestLine» определены следующие роли:
− администратор БД;
− пользователь.

Основными обязанностями администратора баз данных являются:
− установка, модернизация, настройка параметров программного обеспечения СУБД;
− оптимизация прикладных баз данных по времени отклика, скорости доступа к данным;
− разработка, управление и реализация эффективной политики доступа к информации, хранящейся в прикладных базах данных.

Администратор баз данных должен обладать средним уровнем квалификации и не большим практическим опытом выполнения работ по установке, настройке и администрированию используемых в ИУС СУБД.

Пользователи системы должны иметь опыт работы с персональным компьютером на базе операционных систем Microsoft Windows на элементарном уровне.

3.2 Разработка интерфейса ИС «BestLine»

3.2.1 Принципы проектирования GUI-интерфейса

В связи с тем, что программный комплекс планируется использовать в системах Microsoft Windows, было принято решение выполнить программный интерфейс максимально подобный стандартному интерфейсу Windows. Принципы построения интерфейса программных модулей проекта приведены ниже.

Управление пользователем. Пользователь должен ощущать, что именно он управляет программой, а не подчиняется ее требованиям. Это подразумевает, что инициатором любых действий программы является пользователь. Пользователи, обладая различными навыками и предпочтениями, должны иметь возможность настроить интерфейс программы в соответствии со своими требованиями.

Явность. Пользователь должен иметь возможность непосредственного влияния на программные представления объектов предметной области. При любых изменениях, вносимых пользователем, он должен четко видеть, как эти изменения влияют на его программное окружение.

Последовательность в реализации интерфейса позволяет пользователю легко переносить свои знания по использованию программ на новые задачи, фокусируя свое внимание на задачах, а не способах их выполнения. Обеспечивая чувство стабильности, последовательность в реализации интерфейса сделает программу знакомой и предсказуемой. Интерфейс должен быть выполнен в одном стиле во всем программном комплексе и быть максимально приближенным к интерфейсу своих функциональных аналогов (MicrosoftOffice, CorelDraw, AutoCAD и т.д.).

Снисходительность. Зачастую пользователи, обладающие различным уровнем подготовки, обучаются приемам работы с программами методом проб и ошибок. Интерфейс комплекса должен быть достаточно гибким для выполнения пользователем своих желаний, и в то же время не должен допускать повреждения системы и/или данных, предупреждая пользователя о таких ситуациях. Также необходимо реализовать систему отката изменений.

Ответная реакция программы. Программы должны всегда отслеживать действия пользователя и информировать его о своем состоянии, особенно в процессе выполнения вычислений при обработке данных, путем формирования индикаторов процесса, показа статусных сообщений и т.д.

Эстетичность. Реализация пользовательского интерфейса должна быть привлекательна для пользователя.

Простота. Интерфейс должен быть простым (но не упрощенным), легко обучаемым и легко используемым.

Логику пользовательского интерфейса представляют в объектно-ориентированном виде. Объект – это форма, у нее есть свойства и методы. Например, форма – регистрации. Свойства – поля, данные. Методы – кнопки,

или действия, которые можно совершить.

Экранные формы обязаны проектироваться с учетом требований унификации:

– все экранные формы пользовательского интерфейса обязаны быть реализованы в едином графическом дизайне, с одинаковым местоположением главных элементов управления и навигации;

– для обозначения сходных операций обязаны применяться сходные графические значки, кнопки и другие управляющие (навигационные) элементы.

3.2.2 Разработка логики работы программ и пользовательских интерфейсов

Интерфейс каждой системы является одной из очень значительной составляющей. Он ориентирован, прежде всего, на конечного пользователя. От того, насколько хорошо спроектирован интерфейс зависят такие факторы как:

– скорость освоения (обучения пользователей) системы;

– затраты на внедрение (обучится сам или специалисты помогут);

– последующая успешная работа (интерфейс должен быть понятен всегда, с ним всегда найдется вероятность поработать), т.е. уменьшение риска возникновения ошибок пользователей от работы с системой.

Сейчас ясно прослеживается тенденция отделения разработки пользовательского интерфейса от разработки остального приложения. Это связано с затратой большого количества времени именно на построение логики и формирование самого интерфейса.

Программное обеспечение для разработки пользовательского интерфейса делится на две существенные группы - инструментарий для разработки пользовательского интерфейса (toolkits) и высокоуровневые средства разработки интерфейса (higher-level development tools).

Инструментарий для разработки пользовательского интерфейса, как правило, включает в себя библиотеку примитивов компонентов интерфейса (меню, кнопки, полосы прокрутки и др.) и назначен для применения программистами.

Высокоуровневые средства разработки интерфейса могут быть использованы непрограммистами и снабжены языком, который позволяет специфицировать функции ввода-вывода, а также определять, используя технику непосредственного манипулирования, интерфейсные элементы.

Пользовательский интерфейс представляет несколько элементов управления. Каждый элемент может представлять следующую информацию:

– текстовую (текстовые поля ввода, метки, подписи);

– числовую (поля числового ввода, различные индикаторы прогресса, ползунки);

– графическую (изображения, индикаторы);

– звуковую (звуковые сообщения, команды);
– бинарную (флажки, индикаторы, радиокнопки);
– и другие.

Реагировать элементы управления могут на действия пользователя, произведенных с помощью средств, таких как: мышь, клавиатура, сенсорный экран, голос, другие.

Одной из основных характеристик элемента системы является его состояние. И в каждом состоянии элементы могут отображаться и реагировать по-разному. Это необходимо, чтобы пользователь не мог привести систему в запрещенное состояние.

Логику пользовательского интерфейса представляют в объектно-ориентированном виде. Объект – это форма, у нее есть свойства и методы. Например, форма – регистрации. Свойства – поля, данные. Методы – кнопки, или действия, которые можно совершить.

Экранные формы должны проектироваться с учетом требований унификации:

– все экранные формы пользовательского интерфейса должны быть выполнены в едином графическом дизайне, с одинаковым расположением основных элементов управления и навигации;

– для обозначения сходных операций должны использоваться сходные графические значки, кнопки и другие управляющие (навигационные) элементы.

3.2.3 Проектирование пользовательского интерфейса

Даная система содержит следующие окна:

– окно авторизации;
– основная форма.

Окно входа и регистрации. В этом окне содержится поле для ввода логина и пароля. Логин всегда остается стандартным, так же как и пароль, эти данные были уже заложены при создании программы ИС «BestLine» (рисунок 3.1).

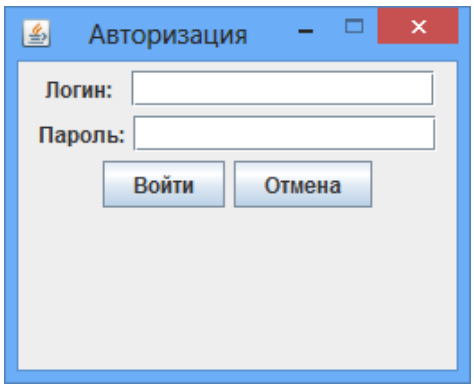

Рисунок 3.1 – Окно авторизации пользователя

Консольное окно Eclipse. Отображается состояние с сервером при авторизации в программе. На данном рисунке показывается, что соединение с сервером установлено, и идет загрузка данных с сервера (рисунок 3.2)

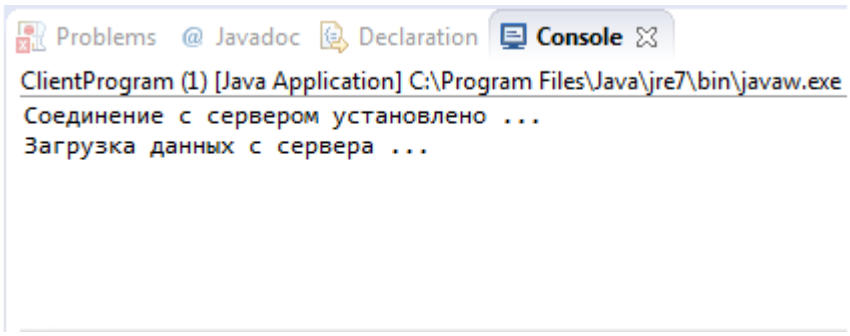

Рисунок 3.2. – Консольное окно 1

Если были введены не правильные данные логина или пароли при авторизации происходит следующая ошибка (рисунок 3.3).

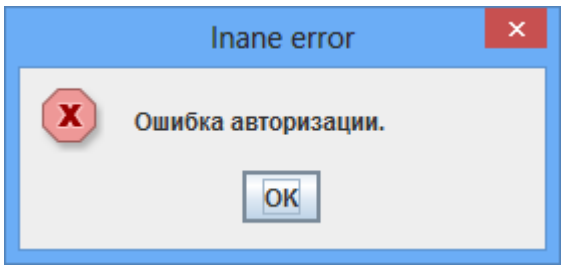

Рисунок 3.3. – Окно ошибки

Окно admin. В данном окне содержатся ввод исходящих и отображения входящих сообщений клиента admin (рисунок 3.4).

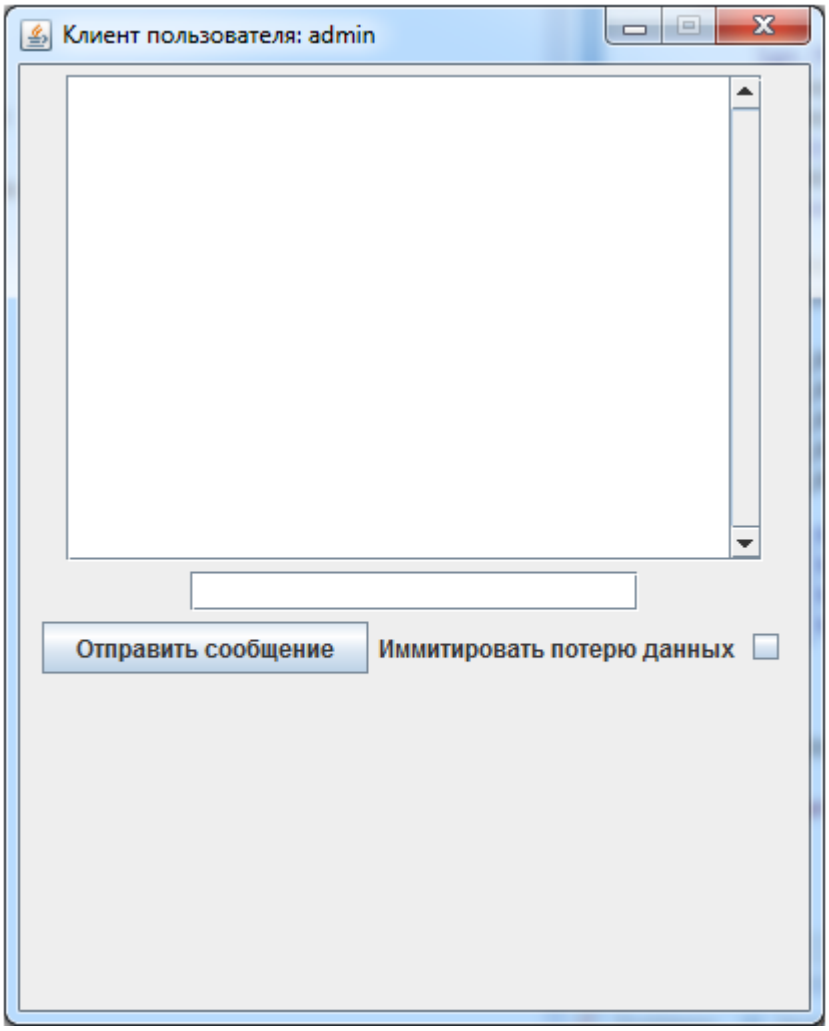

Рисунок 3.4 – Окно admin

Oknouser. В данном окне содержатся ввод исходящих и отображения входящих сообщений клиента user (рисунок 3.5).

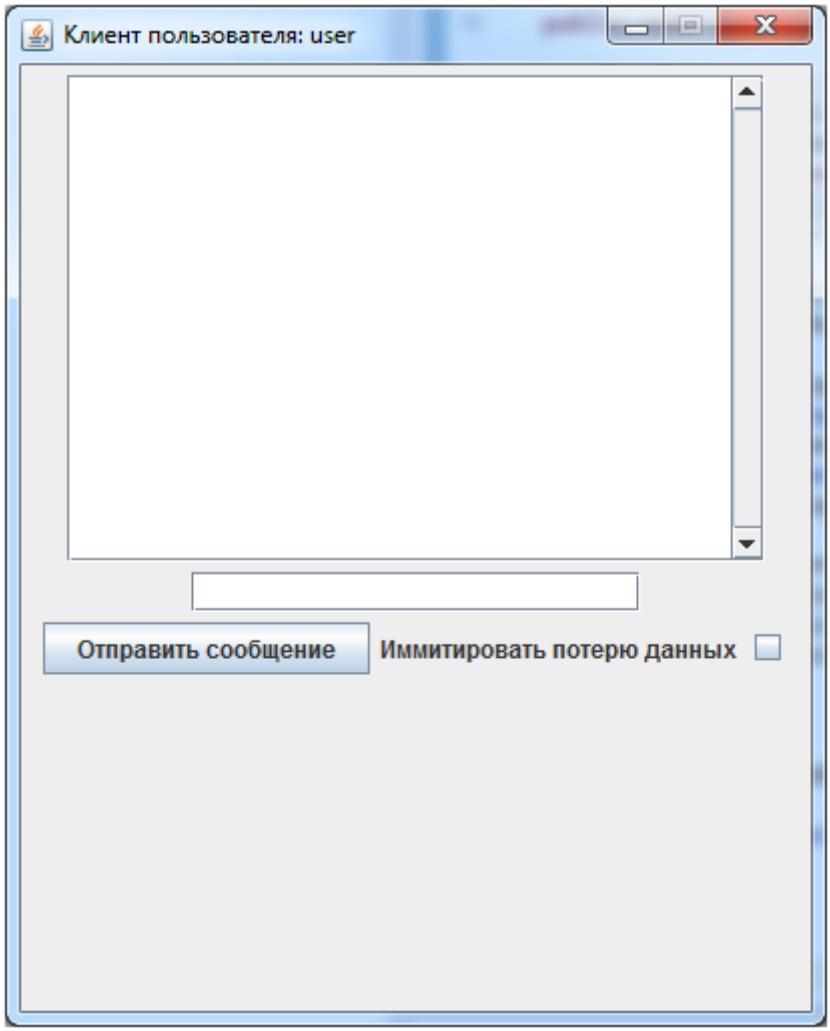

Рисунок 3.5 – Окно user

На этом рисунке наглядно показывается общение между двумя пользователями ИС «BestLine», пользователь user обращается к admin со своей проблемой, но что admin дает свой ответ. На заметку при использовании PrtScrn окно user было не активно (рисунок 3.6).

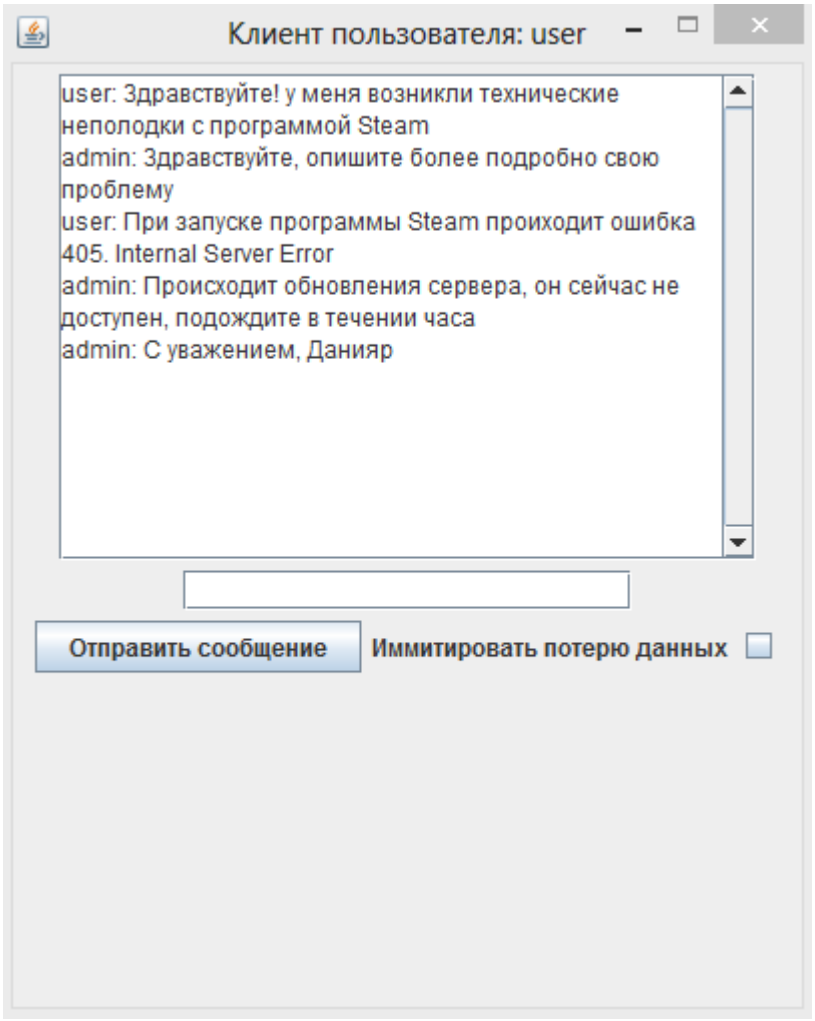

Рисунок 3.6 – Общение между двумя пользователями системы с пользователя user

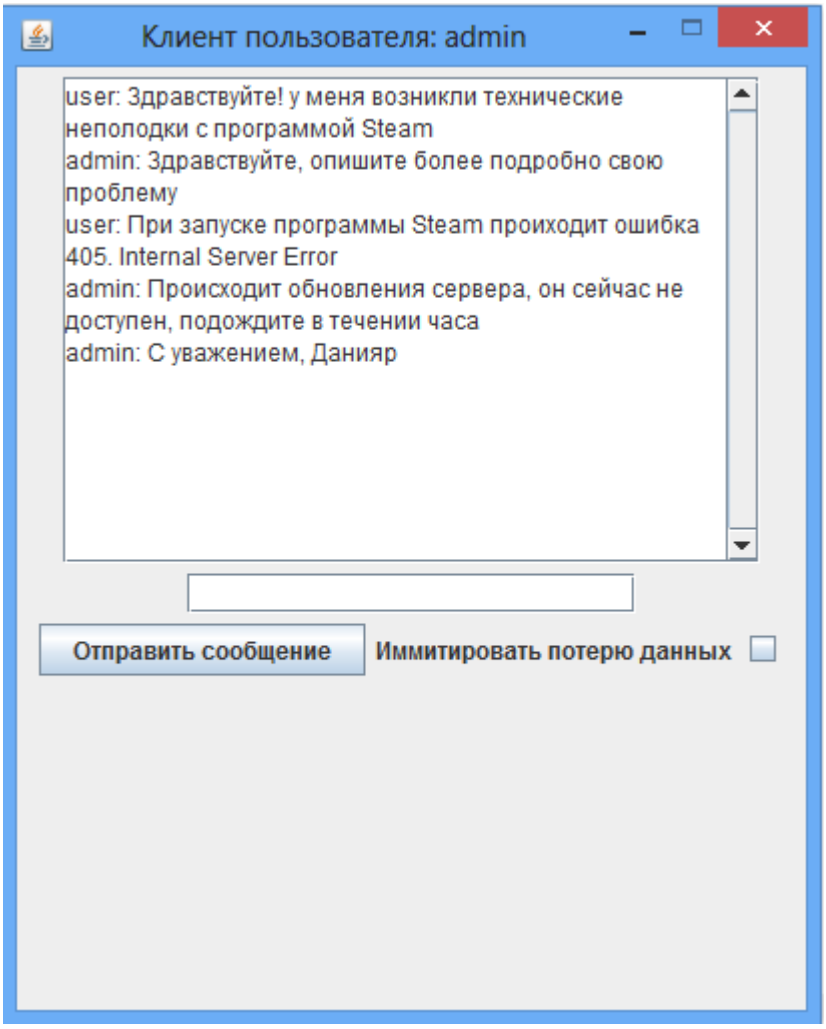

Рисунок 3.7 – Общение между двумя пользователями системы с пользователя admin

В консольном окне после общения админа и пользователя показывается, что данные на сервер были доставлены в целостности и MD5Hash идентичен (рисунок 3.8).

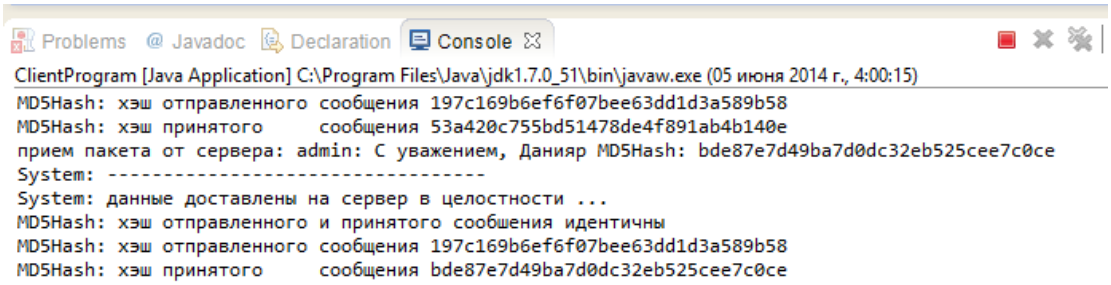

Рисунок 3.8 – Консольное окно 2

В данной программе была разработана функция: Имитировать потерю данных (рисунок 3.9).

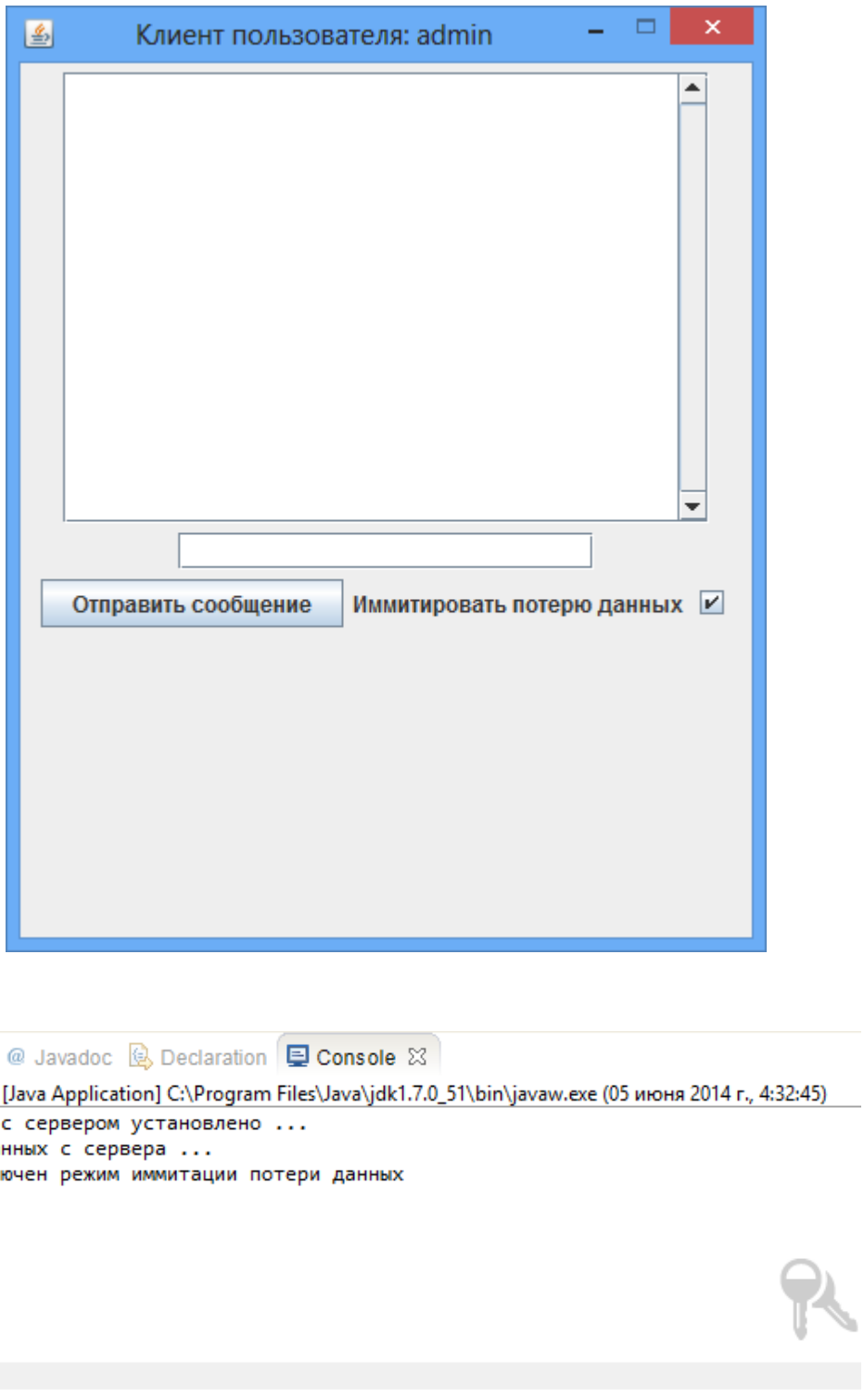

Рисунок 3.9 – Имитация потери данных

При включении режима “Имитировать потери данных” последняя буква сообщения не доходит до адресата и возникает не совпадение MD5Hash, что означает что произошла потеря данных (рисунок 3.10).

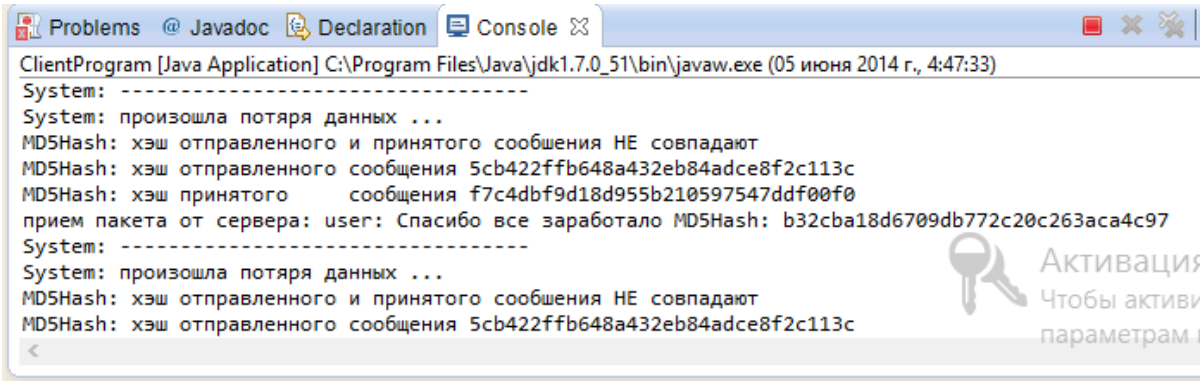

Рисунок 3.10 – Имитация потери данных

Если сервер находится оффлайн или же не исправен то консольное окно выдает следующее (рисунок 3.11).

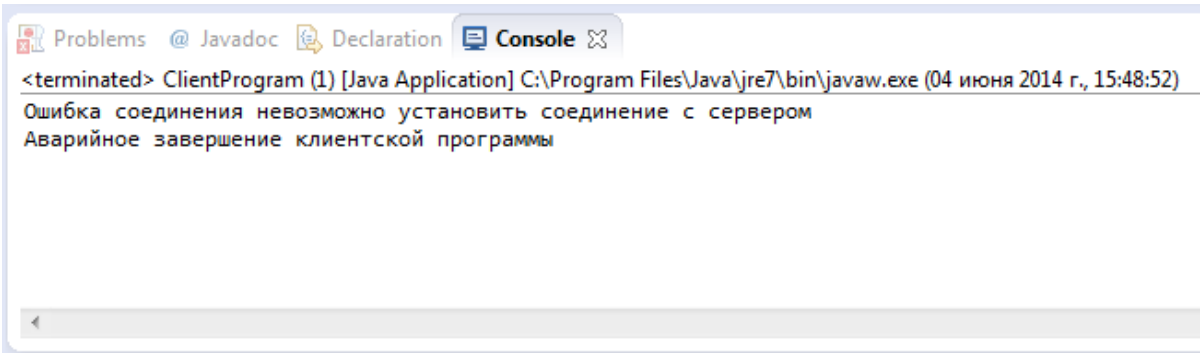

Рисунок 3.11 – Аварийное завершение

4 Экономическая часть

4.1 Технико-экономическое обоснование

Программный продукт, разрабатываемый в рамках дипломной работы, изначально создавался с целью помочь малым и средним предприятиям обезопасить и усовершенствовать их работу, связанную с передачей информации.

Информация является ресурсом, который как и любой иной бизнес-ресурс (финансы, оборудование, персонал) имеет для каждого предприятия определенную стоимость и подлежит защите. Обеспечение информационной безопасности заключается в защите информации от различных видов угроз с целью обеспечения нормального режима работы, успешного развития бизнеса, минимизации деловых рисков, обеспечения личной безопасности сотрудников предприятия.

Информация может быть представлена в различных формах. Она может быть напечатана или написана на бумаге, храниться в электронном виде, передаваться по почте или электронным способом, демонстрироваться в фильмах, видеозаписях, фотографиях, слайдах, а также быть высказана в разговорах. Вне зависимости от формы представления информации, в которой она хранится, обрабатывается или передается, информация должна быть должным образом защищена.

Создание “ИС для безопасной передачи данных” преследует достижение следующих основных целей:

– конфиденциальности: предоставлении доступа к информации только авторизованным сотрудникам;

– целостности: защиты от модификации или подмены информации;

– доступности: гарантировании доступа к информации и информационным ресурсам, средствам информатизации авторизованным пользователям.

4.2 Расчет трудоемкости разработки ПП

Для определения трудоемкости разработки ПП приведен перечень всех основных этапов и видов работ, которые должны быть выполнены.

Форма разделения работ по этапам с указанием трудоемкости их выполнения приведена в таблице 4.1.

Таблица 4.1 - Распределение работ по этапам и видам и оценка их трудоемкости

Этап разработки ПП	Вид работы на данном этапе	Трудоемкость разработки ПП, ч. Программист	Трудоемкость разработки ПП, ч. Научный Руководитель
1. Техническое задание	1. Постановка задачи 2. Сбор материалов и анализ существующих разработок 3. Определение требований к системе	--- 5 2	2 5 ---
2. Эскизный проект	Разработка функциональной схемы программы	8	6
3. Технический проект	1. Выбор инструментальных средств 2. Определение требований к аппаратному обеспечению	2 ---	--- 4
4. Рабочий проект	1. Программирование 2. Тестирование и отладка программы 3. Разработка документации к программному продукту	154 16 4	--- 20 6
Трудоемкость выполнения проекта каждым участником		191	43
ИТОГО трудоемкость выполнения проекта		234	

Поскольку количество часов активной работы по разработке программного продукта равно 234, а в сутки на разработку выделялось шесть часов, следовательно, срок выполнения проекта равен 39 суткам. Для дальнейших расчетов время разработки программного продукта округляем до двух месяцев.

4.3 Расчет затрат на разработку ПП

Общая сумма затрат на материальные ресурсы ($З_М$) определяется по формуле:

$$З_м=\sum P_i*Ц_i, \quad (4.1)$$

где P_i - расход i-го вида материального ресурса, натуральные единицы;

$Ц_i$ - цена за единицу i-го вида материального ресурса, тг;

i - вид материального ресурса;

n - количество видов материальных ресурсов.

Расчет затрат на материальные ресурсы производится по форме, приведенной в таблице 4.2.

Таблица 4.2 - Затраты на материальные ресурсы

Наименование материального ресурса	Единица измерения	Количество израсходованного материала	Цена за единицу, тг	Сумма, тг
1. Бумага писчая «Снегурочка», пачка 500 листов	Пачка	1	1000	1000
2. Картридж для принтера	Шт.	1	4000	4000
3. Другие канцтовары	---	---	4000	4000
4.Научно-техническая литература		3	1500	4500
ИТОГО затраты на материальные ресурсы				13 500

Общая сумма затрат на электроэнергию ($З_Э$) рассчитывается по формуле:

$$З_э=\sum M_i*K_i*T_i*Ц\ , \quad (4.2)$$

где M_i - паспортная мощность i-го электрооборудования, кВт;

K_i - коэффициент использования мощности i-го электрооборудования (принят K_i=0.7);

T_i - время работы i-го оборудования за весь период разработки ПП ч;

Ц - цена электроэнергии, тг/кВт×ч;

i - вид электрооборудования;

n - количество электрооборудования.

Затраты на электроэнергию приведены в таблице 4.3.

Таблица 4.3 - Затраты на электроэнергию

Наименование обор-ия	Паспортная мощность, кВт	Коэфф-т использ-я мощности	Время работы оборуд-я для разработки ПП, ч	Цена электро энергии, $\frac{\text{тг.}}{\text{кВт} \times \text{ч}}$	Сумма, тг
1. ПК	0,4	0,7	234	20	1310
ИТОГО затраты на электроэнергию					1310

Общая сумма затрат на оплату труда ($З_{ТР}$) определяется по формуле:

$$З_{тр}=\sum ЧС_i * Т_i, \quad (4.3)$$

где $ЧС_i$ - часовая ставка i-го работника, тг;

$Т_i$ - трудоемкость разработки ПП, чел.×ч;

i - категория работника;

n - количество работников, занятых разработкой ПП.

Часовая ставка программиста составляет 1 500 (тг/ч), трудоемкость разработки – 191 ч.

Часовая ставка научного руководителя составляет 3 000 (тг/ч), трудоемкость разработки – 43 ч.

$$З_{ТР} = 1\ 500 * 191 + 3\ 000 * 43 = 415\ 500 \text{ тг.}$$

Затраты на оплату труда приведены в таблице 4.4.

Таблица 4.4 - Затраты на оплату труда

Категория работника	Квалиф-ция	Трудоемкость разработки ПП, ч	Часовая ставка, тг/ч	Сумма, тг
1. Программист	Ведущий программист	191	1 500	286 500
2. Научный руководитель	Руководитель проекта	43	3 000	129 000
ИТОГО затраты на оплату труда				415 500

Сумма годовых амортизационных отчислений определяется по формуле:

А = Перв. стоимость * Норма амортизации/100, (4.4)

Амортизационные отчисления приведены в таблице 4.5.

Таблица 4.5 - Амортизация основных фондов (ОФ)

Наименование оборудования и ПО	Стоим-ть оборудования и ПО, тг	Годовая норма амортизации, %	Срок полезного исп-нияобор-я и ПО, год	Сумма амортизации в год, тг	Сумма амортизации в месяц, тг
1. Системныйблок Asus	70 000	20	5	14 000	1166
2. Мышь Logitech X3	5000	20	5	1000	84
3. Клавиатура Genius	5 000	20	5	1000	84
4. Монитор BenQ 820	20 000	20	5	4000	333
5. Принтер Canon Shot LBP-120	15 000	20	5	3000	250
6. Windows 8	30 000	15	2,5	4 500	375
7. Microsoft Office 2010Standard	14 000	15	2,5	2 100	175
8. Java Development Kit 8	Распространяется бесплатно				
9. Eclipse	Распространяется бесплатно				
10. SQLiteStudio	Распространяется бесплатно				
ИТОГО амортизация основных фондов					2467

Годовые нормы амортизации ОФ принимаются по налоговому кодексу РК или определяются, исходя из возможного срока полезного использования ОФ:

$$Н_{Ai}=100/Т_{Ni}, \quad (4.5)$$

где $Т_{Ni}$ - возможный срок использования i-го ОФ, год.

$НА_{об}$= 70 / 5 = 14

$А_{сб}$= (70 000 * 20) / 100 = 14 000тг.

$А_{м}$= (5 000 * 20) / 100 = 1000 тг.

$А_{кл}$= (5 000 * 20) / 100 = 1000 тг.

$А_{мон}$ = (20 000 * 20) / 100= 4000 тг.

$$А_{пр}= (15\ 000 * 20) / 100= 3000 \text{ тг.}$$
$$А_{w}= (30\ 000 * 15) / 100 = 4\ 500 \text{ тг.}$$
$$А_{mo}= (14\ 000 * 15) / 100= 2\ 100 \text{ тг.}$$

Сумма амортизации за один месяц = А / 12.

Сумма амортизационных отчислений за два месяца равна 4934 тг.

В статью «Прочие затраты» включаются расходы на арендную плату, включая коммунальные платежи, затраты на лицензирование и сертификацию, расходы на рекламу, канцелярские и прочие хозяйственные расходы.

Стоимость аренды помещения на месяц равна 36 000 тг. (в эту сумму включены коммунальные услуги).

Арендная плата рассчитывается по формуле:

$$АП= Са*S, \quad (4.6)$$

где Са – срок аренды;

S – стоимость аренды за 1 месяц.

$$АП = 36\ 000 * 2 = 72\ 000 \text{ тг.}$$

Расходы на интернет, месячная оплата которого составляет 4500 тг равны:

$$Р_{и}= 2 * 2400= 4\ 800 \text{ тг.}$$

Прочие хозяйственные расходы составляют 5 000 тг;

Прочие затраты = 72 000 + 4 000 + 5 000 = 81 000 тг.

Социальный налог, согласно Налоговому кодексу РК, составляет 11 % от ФОТ. Пенсионные отчисления не облагаются социальным налогом.

$$Ос = (ФОТ- О_{п})*0,11, \quad (4.7)$$

где $О_{п}$ - отчисления в пенсионный фонд, 10% от ФОТ.

$$ПО=ФОТ*10\%=415\ 500*0,1=41\ 550 \text{ тг.}$$
$$Ос=(415\ 500-41550)*0,11=41\ 134 \text{ тг.}$$

На основании полученных данных по отдельным статьям в таблице 4.6 приведена смета затрат на разработку ПП

Таблица 4.6 - Смета затрат на разработку ПП

Статьи затрат	Сумма, тг
1. Материальные затраты, в том числе: - материалы - электроэнергия 2. Затраты на оплату труда. 3. Отчисления на социальные нужды. 4. Амортизация основных фондов. 5. Прочие затраты.	 13 500 1310 415 500 41 134 4 934 81 000
ИТОГО по смете	552 878

4.4 Определение возможной (договорной) цены ПП

Величина возможной (договорной) цены ПП должна устанавливается с учетом эффективности, качества и сроков ее выполнения на уровне, отвечающем экономическим интересам заказчика (потребителя) и исполнителя.

Договорная цена ($Ц_Д$) для прикладных ПП рассчитывается по формуле:

$$Ц_д = З_{нир} * (1 + (Р/100)) , \quad (4.8)$$

где $З_{НИР}$ - затраты на разработку ПП (из таблицы 4.6), тг;

Р - средний уровень рентабельности ПП. % (принято 20%).

$$Ц_Д = 552\ 878 * (1 + 0{,}20) = 663\ 453 \text{ тг.}$$

Цена реализации с учетом НДС рассчитывается по формуле:

$$Цр = Цд + Цд * НДС, \quad (4{,}9)$$

НДС, согласно Налоговому кодексу РК, составляет 12 %.

$$Ц_Р = 663\ 453 + 663\ 453 * 0{,}12 = 743\ 067 \text{ тг.}$$

Разработанный программный продукт будет лицензироваться наиболее удобным способом для организаций любого размера – корпоративным

лицензированием. В отличие от OEM- или коробочных лицензий, которые чаще всего могут быть использованы для лицензирования определённого ПОна одном ПК, программы корпоративного лицензирования позволяют лицензировать практически любое ПО для любого количества ПК в рамках одного или нескольких соглашений по выбору клиента.

4.5 Расчет срока окупаемости ПП

В результате создания “ИС для безопасной передачи данных” станет возможным продавать программу по цене 4000 тенге на один ПК.

Расчет затрат создания одного диска программного продукта:

- диск CD-RW – 200 тенге за шт;
- установка программного обеспечения – 400 тенге за шт;
- оформление носителя – 400 тенге за шт.

В итоге на создание одного диска ПП затраты – 1000 тенге.

Прибыль за один диск: 4000-1000=3000.

Расчетное количества продаж окупаемости продукта можно найти по формуле:

$$N_{ок} = C / S, \quad (4.10)$$

где C - затраты на разработку и внедрение системы, тенге;
S – стоимость одного диска программного продукта.

$$N_{ок} = 743\ 067 / 3\ 000 = 248 \text{ (дисков)}.$$

В данном случае срок окупаемости проекта составит продажа 248 дисков программного продукта.

4.6 Оценка социально - экономических результатов функционирования программного продукта

Реализация угроз может нанести значительный ущерб и стать причиной существенных убытков, причиной снижения деловой активности, временного или полного прекращения деятельности. Для предотвращения реализации угроз предпринимаются меры, включая безусловное выполнения определенных политикой безопасности предприятия правил и процедур работы с информационной системой, использование средств защиты информации, контроль со стороны уполномоченных сотрудников за выполнением сотрудниками своих обязанностей по обеспечению информационной безопасности.

Целесообразность расчета затрат, используемых на разработку, не вызывает сомнения. В результате внедрения разработки повышается уровень доверия к данным, уровень конфиденциальности, оперативность, качество принимаемых решений, вследствие чего повышается надежность, доступность, сохраняется целостность данных, а также происходит улучшение условий труда и обеспечение сопровождения разработки.

Разработанный программный продукт выполняет свои функции без лишних затрат ресурсов и имеет гибкую систему.

Данный программный продукт предназначен для малых и средних предприятий имеющих цель обезопасить информацию.

Программный продукт работает как защищенный чат для отправки данных, имеющий MD5 хэш функцию.

Достоинство продукта что он разрабатывается на объектном ориентированном языке программирования и имеется возможность до конструировать проект до полноценной платформы по типу ”VipNet”, что будет уже использоваться в крупных предприятиях.

5Безопасность жизнедеятельности

5.1Анализ опасных и вредных производственных факторов

Данный дипломный проект “Разработка информационной системы для безопасно передачи данных” создавалась в помещении соответствующим нормам, в помещении выполняются легкие физические работы, поэтому соблюдались следующие требования: оптимальная температура воздуха- 22° С (допустимая - 20-24° С), оптимальная относительная влажность- 40 -60% (допустимая - не более 75%) , скорость движения воздуха не более 0.1м/с.

Данный дипломный проект представляет собой информационную систему. В связи с этим работа с использованием этой системы будет проводиться с применением ЭВМ. Размеры помещения составляют 5.6х4.05х3.15 м.

Рассмотрим основные опасности и вредные воздействия компьютера при его использовании.

Так как работа человека происходит с электронно-вычислительной техникой, основной опасностью является поражение его электрическим током. Персональная ЭВМ питается от сети с напряжением 220 В, которое является опасным для человека.

Работа связана с обработкой большого количества бумажных документов. Поэтому существует опасность возникновения пожара.

Большое влияние на самочувствие и работоспособность человека оказывает микроклимат (метеорологические условия) производственных помещений, который определяется температурой воздуха, его составом и давлением, относительной влажностью, скоростью движения воздушных потоков.

Схема помещения приведена на рисунке 5.1

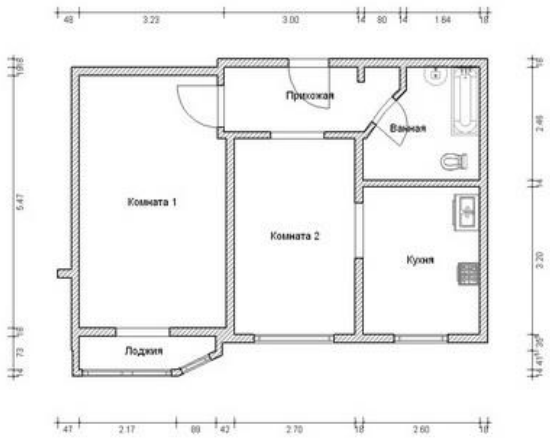

Рисунок 5.1 - План помещения

5.2Защитные мероприятия

5.2.1 Производственная санитария

В помещении на человека, работающего с ЭВМ могут негативно действовать следующие физические факторы:

- увеличенная и сниженная температура воздуха;
- чрезвычайная запыленность и загазованность атмосферы;
- увеличенная и сниженная влажность воздуха;
- недостаточная освещенность места работы;
- превосходящий вероятные нормы шум;
- повышенный уровень ионизирующего излучения;
- повышенный уровень электромагнитных полей;
- повышенный уровень статического электричества;
- опасность удара электрическим током;
- блеклость экрана дисплея.

Формированию утомляемости на производстве содействуют следующие факторы:

- неверная эргономическая организация трудового поста, нерациональные области размещения снабжения по высоте от пола;
- характер протекания работы, значительный смысл имеет чередование труда и отдыха, замена одних форм работы иными.

Соответственно запросам санитарных правил устремлены на предупреждение неблагоприятного воздействия на здоровье человека вредных факторов производственной среды и рабочего процесса при работе с ЭВМ.

В помещениях с небольшим уровнем общего шума, источниками шумовых помех могут стать вентиляционные указания, кондиционеры или основное оборудование для ЭВМ (плоттеры, принтеры). Длительное влияние этих шумов негативно влияют на эмоциональном состоянии персонала.

В соответствии с ГОСТ 12.1.003-76 эквивалентный уровень звука не обязан превосходить 50 дБА. Для того, чтобы достичь этого уровня шума представляется использовать звукопоглощающее помещение.

В качестве мер по понижению шумавозможно предложить вытекающее:

- облицовка потолка и стен звукопоглощающим тканью (снижает шум на 6-8 дб);
- экранирование трудового поста (постановкой перегородок, диафрагм);
- установка в компьютерных кабинетах оборудования, вырабатывающего наименьший шум;
- разумно планировать кабинеты.

Предохранение от шума надлежит осуществлять в соответствии с ГОСТ 12.1.003-76, а звукоизоляция ограждающих устройств обязан отвечать запросам главы СНиП 11-12-77 “Защита от шума. Нормы проектирования”.

Таблица 5.1 –Нормы уровня шумов

Величина	Уровни звукового давления, Дб, в октавных полосах со среднегеометрическими частотами, Гц							
	63	125	250	500	1000	2000	4000	8000
Помещение пользователей ЭВМ	71	61	54	49	45	42	40	38

Микроклимат производственных кабинетах - это климат наружной среды этих кабинетах, который определяется влияющими на организм человека сочетаниями температуры, влажности и быстроты движения воздуха.

Для произведения и автоматического поддержания в помещении самостоятельно от наружных обстоятельств оптимальных значений температуры, влажности, аккуратности и быстроты движения воздуха, в холодное время года применяется водяное отопление, в теплое время года используется кондиционирование воздуха. Кондиционер доставляет собой вентиляционную установку, которая с помощью приборов механического регулирования поддерживает в кабинете заданные параметры оптимальной воздушной среды.

Таблица 5.2 – Эталон температуры, относительной влажности и быстроте движения воздуха в рабочей лабораторном кабинете ВЦ

Температура окружающего воздуха	Оптимальные параметры воздушной среды на постоянных рабочих местах		
	Температура, °С	Относительная влажность, %	Скорость движения воздуха, м/с не более
Ниже+10°С	20-22	40-60	0,1
Выше+10°С	22-25	40-60	0,3

Поэтому нужно разработать средства предохранения от этих вредоносных факторов. К предоставленным средствам защиты причисляются: вентиляция, искусственное освещение, звукоизоляция. Имеются нормативы, назначающие комфортные обстоятельства и максимально возможные нормы запылённости, температуры воздуха, шума, освещённости. В системе мер, обеспечивающих благоприятные обстоятельства работы, крупное место отводится эстетическим факторам: оформление производственного интерьера, оборудования, использование функциональной музыки, которые оказывают определённое воздействие наорганизмом человека. Значительную роль играет цвет кабинета, который обязан быть светлым. В данном разделе дипломного проекта считается надобная освещённость рабочего места.

5.2.3 Освещение

Освещение является одним из существенных производственных условий работы. Через глаза человек получает порядка 88 % информации. От освещения зависит усталость работающего, плодотворность труда, его безопасность. Достаточное освещение влияет тонизирующем образом, совершенствует протекание существенных процессов высшей нервной системы, стимулирует обменные и иммунобиологические процессы, проявляет воздействие на суточный ритм физических функций организма человека. Недостаточное освещение рабочих мест негативно воздействует на глаза, что возможно приведет к близорукости, глаза весьма сильно напрягаются, темп труды уменьшается, могут начаться ошибки. Чересчур интенсивное освещение нервирует сетчатую оболочку глаза, возбуждает ослепленность. Глаза работника быстро устают, зрительное восприятие портится, повышается производственный травматизм, плодотворность труда уменьшается.

К кабинету с ЭВМ предъявляются запросы:

− кабинет для работы с ЭВМ обязан обладать естественным и искусственным освещением. Эксплуатация ЭВМ в кабинетах без природного освещения пропускается лишь при надлежащем обосновании и присутствии позитивного санитарно-эпидемиологического заключения, взятого в определенной последовательности;

− естественное и ненатуральное освещение следует соответствовать запросам воздействующей нормативной документации. Окна в кабинетах, где работает вычислительная техника, предпочтительно обязаны быть ориентированы на север и северо-восток. Оконные проемы подобающи быть снабжены регулируемыми приспособлениями типа: жалюзи, занавесей, внешних козырьков;

− при применении ЭВМ на базе ЭЛТ (без периферийных механизмов - принтер, сканер), соответствующих запросам международных эталонов безопасности компьютеров, с длительностью работы меньше 4-х часов в день пускается минимальная площадь 4,5 м2 на одно трудовое место пользователя (взрослого и учащегося).

Искусственное освещение в кабинетах следует реализовываться системой общественного равномерного освещения.

Для целесообразного освещения необходимо, чтобы:

− неизменная освещенность рабочих мест на все время использования;

− довольно и размеренно разделенная яркость освещаемых рабочих мест;

− не наличие вызывающих контрастов между яркостью рабочих мест и окружающего пространства;

− наличие резких и глубоких теней на трудовыхместах, и на полу, что доноситься верным местоположением светильников, а также повышением отражения света от потолка и стен кабинета и освещенных рабочих мест;

Главной величиной светотехники являются биение и световой поток. Световым потоком (Ф) - именуют мощность лучистой энергии, оцениваемую по вырабатываемому ею световому чувству на человеческий глаз и замеряется в люменах (ЛМ). Люмен - это световой поток, источаемый размеренным точечным источником света в 1 канделу (кд) в телесном угле, равном 1 стерадиану (СР). Освещенностью называется поверхностная плоскость светового потока, выпадающего на освещаемую поверхность и равномерно на ней распределенного:

$$E=dF/dS, \qquad (5.1)$$

где Е-освещенность;
dF-световой поток;
dS-освещаемая поверхность.

В производственных обстановках применяется три вида освещения: естественное, т.е. осуществляемое через окна в наружных стенах здания; искусственное, создаваемое электрическими или люминесцентными лампами; комбинированное.

Естественное освещение зависит от времени года, дня, от страны, внутреннего конструкции здания и окон, отражательных свойств поверхностей перед окнами, ширины улицы и прочих обстоятельств.

Естественное освещение какой-нибудь точки в кабинетеможно охарактеризуется коэффициентом естественной освещенности и определяется в процентах:

$$КЕО=(Ев/Ен)*100\%, \qquad (5.2)$$

где Ев - освещенность в точке М внутри помещения,лк;
Ен- одновременная наружная освещенность горизонтальной плоскости светом, лк.

Для следования норм естественной освещенности крупное значение обладает нынешняя очистка стекол и покраска стен. Нечистые стекла задерживают 67% света. Исходя из данного, нормы природной освещенности кабинетах поставлены с учетом регулярной очистки стекол световых проемов. Покраска внутренних поверхностей обязаны быть предпочтительнее светлой и периодически восстанавливаться не реже одного раза в два года.

Для кабинета, где находится рабочее место оператора, применяется система общественного освещения. Нормами для данных работ установлена необходимая освещённость рабочего места указана ниже.

Таблица 5.3– Нормы освещенности для лаборатории ВЦ

Плоскость нормирования освещенности и высота от пола, м.	Норма освещенности ЛК при общем освещении.	Коэффициент естественного освещения, % не более.
Горизонтальная плоскость - 0,8	300	15

5.3. Расчет освещения методом коэффициента использования светового потока.

Расчёт системы освещения совершается методом коэффициента применения светового потока, который выражается связью светового потока, выпадающего на расчётную поверхность, к суммарному потоку всех ламп. Его величина зависит от характеристик светильника, величин помещения, цвета стен и потолка, характеризуемой коэффициентами отображения стен и потолка.

Расчет освещенности рабочего места сводится к выбору системы освещения, определению потребного числа светильников, их типа и размещения.

Расчет освещения производится для помещения, длиной – 5,6 и шириной - 4,05 , высотой – 4,15 м.

В методе коэффициента использования расчет светового потока источника производится по формуле:

$$F = \frac{E_н SZK}{N\eta}, \qquad (5.3)$$

где Ен - нормативная освещенность, 300 лк; S - освещаемая площадь, м2;

Z - коэффициент минимальной освещенности;

$$Z = \frac{E_{ср}}{N\eta} \approx 1.1 \div 1.5$$

К - коэффициент запаса, учитывающий ухудшение характеристик источников при эксплуатации(износ и загрязнение светильников) - 1,5;

N - число ламп в светильнике;

η - коэффициент использования светового потока.

Коэффициент использования определяется по индексу помещения in и коэффициентам отражения потока, стен и пола по специальной таблице.

Размещение светильников определяется размерами (рисунок 5.1)

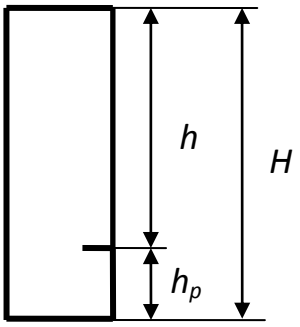

Рисунок 5.1 – Определение расчетной высоты

Расчетная высота определяется по формуле:

$$h = H - hp, \qquad (5.3)$$

где Н-высота помещения, hp-высота расчетной поверхности (1м), h - расчетная высота.

$$h = H - hp = 4{,}15 - 1{,}0 = 3{,}15 \text{ м}$$

Индекс помещения рассчитывается по формуле:

$$i_n = \frac{AB}{h(A+B)} \qquad (5.4)$$

где А и В длина и ширина помещения; h – высота расчетная.

$$i_n = \frac{AB}{h(A+B)} = 0{,}7, \qquad (5.5)$$

Тогда находим (для ламп i=0.7) η=0,38.
Определим площадь помещения:

$$S=A\cdot B= 22{,}68 \text{ м2}$$

Определяем общий световой поток:

$$F=300 \cdot 22{,}68 \cdot 1{,}1 \cdot 1{,}5 / (N \cdot 0{,}38),$$

где N=2 (число ламп в светильнике).

$$F=300 \cdot 22{,}68 \cdot 1{,}1 \cdot 1{,}5 / (2 \cdot 0{,}38)=14771{,}84 \text{ лм}$$

Наиболее приемлемыми для помещения являются люминесцентные лампы ЛБ (белого света) или ЛТБ (тёпло-белого света), мощностью 20, 40 или 80 Вт. Световой поток одной лампы ЛБ40 составляет F1=2480 лм, следовательно, для получения светового потока Fобщ=14771,84 лм необходимо N светильников, число которых можно определить по формуле:

$$N=F_{общ}/F_1, \quad (5.6)$$

Подставим значения, полученные выше: N= 14771,84 / 2480 = 6 светильников.

Таким образом, необходимо установить 12 ламп ЛБ40 или 6 светильников(рисунок 5.2).

Электрическая мощность всей осветительной системы вычисляется по формуле:

$$P_{общ}=P_1*N, \quad (5.7)$$

где Р1 – мощность одной лампы = 40 Вт, N – число ламп = 6.

$$Робщ=40 \cdot 12 = 480 \text{ Вт}.$$

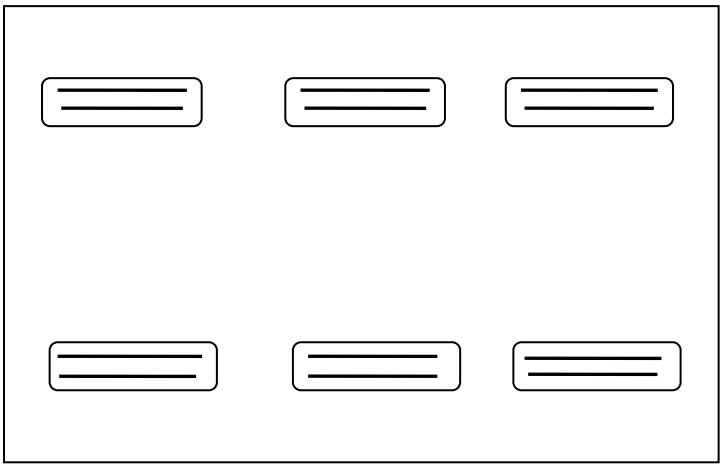

Рисунок 5.2 - Размещение светильников

При выборе осветительных приборов используем светильники типа ОД. Каждый светильник комплектуется двумя лампами. Размещаются светильники двумя рядами, по три в каждом ряду.

Рассчитаем величину ошибки:

$$\Delta=F_{расч}-F_{реал}/F_{расч}*100\%, \quad (5.8)$$

где Δ-величина ошибки, Fрасч- световой поток, полученный расчетным методом, Fреал- реальный световой поток.

$$Fрасч=14771,84 \cdot 2=29543,68 \text{ лм};$$

Fреал=2480·12=29760 лм.

$$\Delta = \left| \frac{F_{расч} - F_{реал}}{F_{расч}} \right| 100\% = \left| \frac{29543{,}68 - 29760}{29543{,}68} \right| 100\% = 0{,}7\% \qquad (5.9)$$

Величина ошибки не превышает допустимые значения.

Для исключения засветки экрана монитора прямыми световыми потоками светильники общественного освещения устанавливаются сбоку от рабочего места, поперечно линии зрения оператора и стене с окнами. Подобное размещение светильников разрешает производить их последовательное включение в зависимости от величины естественной освещённости и исключает раздражение глаз чередующимися полосами света и тени, возникающее при параллельном расположении светильников.

Заключение

В дипломном проекте была разработана информационная система «BestLine» для ИП «BestLine»,что представляет из себя клиент-серверное приложение для безопасной передачи данных. Также обоснована необходимость разработки ИС «BestLine», которая позволяет повысить эффективность работы предприятия.

Первая глава посвящена изучению предметной области исследования. Информация одна из важнейших ресурсов на сегодняшний день и ее защита так же остается первостепенной задачей. Поэтому для создания своего приложения я использовались алгоритмы шифрования и алгоритмы хэширования для безопасности. Так же в главе описывались инструменты для выполнения работы.

Вторая глава посвящена разработке UML-диаграмм и проектированию базы данных ИС «BestLine». Для построения моделей использовались UML-диаграммы и методология IDEF1X.

Интерфейс системы имеет удобный и дружелюбный вид. Используемый стиль упрощает работу с программой. Рабочая система содержит следующие окна и панели: окно авторизации, окно admin, окно user, консольное окно.

Четвертая глава посвящена технико-экономическому расчету. В результате экономического расчета затраты на разработку программного продукта составили 552 878тенге. Основной статьей расходов является заработная плата, которая составляет 79,7% от всех затрат. На втором месте прочие расходы, которые составляют 20,3% от всех затрат. Цена реализации программного продукта - 743 067тенге.

В пятой главе мы провели анализ опасных и вредных производственных факторов, рассчитали оптимальные нормы для помещения в котором выполнялась дипломная работа. Моей главной задачей было рассчитать оптимальное освещение место работы. Из расчетов следует отметить, что на комнату в районе 20 кв.м. требуется 6 ламп по 40 Вт.

Список литературы

1. П.Ноутон, Г.Шилдт«Java 2 Наиболее полное руководство» - Издательство «Expres», 2012.– 1102 с.
2. Вьюкова Н. Информационная безопасность систем управления базами данных [Электронный ресурс]. – Режим доступа: http:www.citforum.ru/database/ kbd96/index.shtml.
3. Гладченко А. Создание гибкой системы безопасности [Электронный ресурс]. – Режим доступа: http://www.sql.ru/articles/mssql/01061605.shtml.
4. У. Боггс, М. Боггс «UML и RationalRose 2002» - Издательство «ЛОРИ», 2004.– 415 с/
5. Диаграммы UML [Электронный ресурс] – Режим доступа: http://www.intuit.ru/studies/courses
6. Бекишева А. И. Методические указания к выполнению экономической части дипломной работы для бакалавров специальности 5В070300 – Информационные системы – Алматы: АУЭС; 2013. – 24с.
7. Хакимжанов Т. И. Методические указания к выполнению раздела в дипломных проектах(для студентов всех форм обучения всех специальностей) – Алматы: АИЭС; 2002. – 29с.
8. Основы UML – Разработка диаграмм в среде RationalRose [Электронный ресурс] – Режим доступа:http://2programmer.ru/uml/
9. Грабер М. SQL:[пер. с англ.] / М.Грабер.-М: Издательство «Лори», 2003.– 854 с.
10. РД Концепция защиты СВТ и АС от НСД к информации: утв. Гостехкомиссией России 30.03.92. – Москва, 1992.– 266 с.
11. Шумейко В. Безопасность информационных сетей и баз данных. Методическое руководство. Составители: В.А. Шумейко, А.О.Башмаков.– Новосибирск: Изд-во НГТУ, 2004. – 521 с.
12. Козленко Л. Информационная безопасность в современных системах управления базами данных / Л.Козленко // КомпьютерПресс. – 2002. - №3.– 247 с.

Приложение А

Техническое задание

А.1 Общие положения

А.1.1 Полное наименование системы и ее условное обозначение

Полное наименование системы: Информационная система для безопасной передачи данных. Краткое наименование системы: ИС «BestLine»

А.1.2 Шифр темы или шифр (номер) договора

Шифр темы: ИС для безопасной передачи данных.-14.
Номер контракта: №1/10-09-13-001 от 10.09.2013 г.

А.1.3 Наименование предприятий (объединений) разработчика и заказчика (пользователя) системы и их реквизиты

Заказчиком системы является ИП «BestLine», г. Алмата.
Адрес заказчика: 050025, Республика Казахстан, г. Алмата, ул. Абылайхана 14.
Разработчиком системы является, Куатбеков Данияр Муратбекулы, студент АУЭС, ФИТ, группа ИС 10-2.
Адрес разработчика: г. Алматы, ул. Гагарина 282, к. 12.

А.1.4 Перечень документов, на основании которых создается система, кем и когда утверждены эти документы

Основанием для разработки ИС «BestLine» являются следующие документы и нормативные акты:
Контракт №1/10-09-13-001 от 10.09.2013 года на выполнение работ по выполнению первого этапа работ по созданию Информационной системы «BestLine»;
Договор разработчика с заказчиком о выполнении работы по созданию системы в указанные сроки, утвержденные директором ИП «BestLine» Куанышбековым О. М.;
Контракт;
На основе задания на дипломный проект.

А.1.5 Плановые сроки начала и окончания работы по созданию системы

Плановый срок начала работ по созданию ИС «BestLine» - 10 сентября 2013 года.

Плановый срок окончания работ по созданию ИС «BestLine» 25 мая 2014 года.

А.1.6 Сведения об источниках и порядке финансирования работ

Источником финансирования является бюджет ИП «BestLine».

Порядок финансирования устанавливается договорами директора ИП «BestLine».

А.1.7 Порядок оформления и предъявления заказчику результатов работ

Система показывается в виде функционирующего комплекса на базе средств вычислительной техники заказчика в сроки, определенные заказчиком. Приемка системы реализовывается комиссией в составе уполномоченных представителей заказчика.

Порядок предъявления системы, ее проверок и окончательной приемки назначен в п.6 настоящего ТЗ. Сообща с предъявлением системы делается сдача разработанного исполнителем набора документации согласно п.8 настоящего ТЗ.

А.1.8 Состав используемой нормативно-технической документации

При разработке информационной системы и организации проектно-эксплуатационной документации Исполнитель обязан следовать запросами следующих нормативных документов:

ГОСТ 34.601-90. Комплекс образцов на информационные системы. Информационных системы. Этапы создания;

ГОСТ 34.201-89. Информационная технология. Комплекс образцов на информационные системы. Виды, комплексность и обозначение документов при произведении информационных систем;

РД 50-34.698-90. Методические указания. Информационная технология. Комплекс образцов на информационные системы. Информационные системы. Требования к содержанию документов.

А.2 Назначение и цели создания (развития) системы

– рекомендации системы – проектируемая система предназначена для применения её сотрудниками ИП, т.е менеджерами и администратором. Её назначение – автоматизация деятельности ИП.

– цели создания системы – повышение результативности работы ИП за счет оперативного обмена данными и создания систематизированной базы данных.

А.2.1 Назначение ИС

ИС "BestLine" предназначена для комплексного информационно-аналитического обеспечения процессов деятельности ИП «BestLine», в части исполнения следующих процессов:

– хранение информации в базе данных с обеспечением удобного, быстрого поиска необходимых данных и надежной системы разграничения прав доступа;

– резервное дублирование(бэкап) данных;

– организация оперативной деятельности по обработке данных;

– обеспечение и контроль дисконтной политики;

– обеспечение доступа администратора к информации, имеющей конфиденциальный характер;

– организация эффективного управления и реагирования на происходящие бизнес-процессы;

А.2.2 Основные цели создания ИС «BestLine»

– защищенная передача информации;

– снижение временных затрат на обработку данных;

– получение отчетов о выполненных операциях, используя созданную базу данных;

–предоставление доступа к информации только авторизованным сотрудникам;

–защиты от модификации или подмены информации;

Для реализации поставленных целей система должна решать следующие задачи:

− распределение прав доступа;

− использования MD5 хеш функции;

− гарантирование доступа к информации и информационным ресурсам, средствам информатизации авторизованным пользователям.;

− безопасность информации коммерческого характера;

− обеспечение защиты от несанкционированного доступа и искажения или удаления информации;

− оперативная обработка и активация заказов;

А.3 Характеристика объекта автоматизации

А.3.1 Объект автоматизации

Процессы по управлению ИС "BestLine", а также проверку результативности выполнения показанных процессов. Предоставленные процессы исполняются руководствующимися специалистами:

– персонал ИП.

А.3.2 Существующее программное обеспечение

В данный момент деятельность ИП «BestLine» не автоматизирована.

А.3.3 Существующее нормативно-правовое обеспечение

Существующее нормативно-правовое снабжение составляют федеральные и областные нормативные правовые акты:

– Конституция РК;

– Гражданский кодекс РК.и т.д.

А.4 Требования к системе

Запросы к системе в целом:

– запросы к структуре и функционированию системы;

– запросы к персоналу системы;

– показатели рекомендации;

– запросы к надежности; безопасности; эргономике и технической эстетике; эксплуатации, техническому обслуживанию, ремонту и хранению компонентов системы; защите информации; сохранности информации при авариях; предохранении от воздействия внешних действий;

– требования к патентной чистоте; стандартизации и унификации; основные требования.

Требования к функциям (задачам), осуществляемым системой; перечень функций, задач или их комплексов, предикатов автоматизации (по каждой подсистеме); очередность ввода в эксплуатацию, временной регламент исполнения и запросы к качеству осуществления любой функции, задачи (или комплекса задач), к форме представления выходной информации,

перечень и критерии отказов для любой функции, по которой задаются запросы по надежности.

А.4.1 Требования к системе в целом

А.4.1.1 Требования к структуре и функционированию системы

А.4.1.1.1 В составе ИС "BestLine" должны решаться следующие задачи:
– резервное дублирование информации, содержащейся в БД;
– защищенная передача информации;
– система хранения информации;

Подсистема резервного дублирования информации подразумевает обеспечение целостности и сохранение одной из главных ценностей системы накопленная база данных, сбой которой приведет непоправимым последствиям для ИП. Поэтому необходимо исключить вероятность потери и порчи данных, при том, что для больших нагруженных БД при ошибке сервера эта вероятность близка к 100%.

А.4.1.1.2 Требования к способам и средствам связи для информационного обмена между компонентами системы

Входящие в состав ИС «BestLine» подсистемы в процессе функционирования должны обмениваться информацией на основе раскрытых форматов обмена данными, употребляя для этого входящие в их состав модули информационного взаимодействия.

Форматы данных будут разработаны и ратифицированы на этапе технического проектирования.

А.4.1.1.3 Требования к режимам функционирования системы

Для ИС «BestLine» назначены следующие режимы функционирования:
– нормальный режим функционирования;
– аварийный режим функционирования.

Основным режимом функционирования ИС является нормальный режим.

В нормальном режиме функционирования системы:

– клиентское программное обеспечение и технические средства пользователей и администратора системы обеспечивают вероятность

функционирования в течение рабочего дня (с 00:00 до 00:00) семь дней в неделю;

– серверное программное обеспечение и технические средства северов обеспечивают возможность круглосуточного функционирования, с перерывами на обслуживание;

– исправно действует оборудование, составляющее комплекс технических средств;

– исправно работает системное, основное и главное программное обеспечение системы.

Для обеспечения нормального режима функционирования системы нужно реализовывать требования и выносить условия эксплуатации программного обеспечения и комплекса технических средств системы, показанные в подобающих технических документах (техническая документация, инструкции по эксплуатации и т.д.).

В случае перехода системы в аварийный режим необходимо:

– закончить эффективность всех приложений, с сохранением данных;

– отключить персональные компьютеры;

– отключить все периферийные устройства;

– исполнить резервное копирование БД.

После этого надобно исполнить комплекс мероприятий по ликвидации причины перехода системы в аварийный режим.

А.4.1.1.4 Требования по диагностированию системы

ИС «BestLine» должна предоставлять инструменты диагностирования главных процессов системы, трассировки и мониторинга процесса выполнения программы.

Компоненты обязаны предоставить удобный интерфейс для вероятности просмотра диагностических событий, мониторинга процесса выполнения программ.

При возникновении аварийных обстоятельств, либо ошибок в программном обеспечении, диагностические инструменты обязаны разрешать сохранять полный набор информации, необходимой разработчику для идентификации проблемы (снимки экранов, текущее состояние памяти, файловой системы).

А.4.1.1.5 Перспективы развития, модернизации системы

ИС «BestLine» обязана выполнять вероятность дальнейшей модернизации как программного обеспечения.

А.4.1.2 Запросы к количеству и квалификации персонала системы

Для эксплуатации ИС «BestLine» назначены вытекающие роли:

– системный администратор;

– администратор баз данных;

– администратор информационной безопасности;

– пользователь.

Главными обязанностями системного администратора являются:

– установка, модернизация, настройка и мониторинг работоспособности системного и базового программного обеспечения;

– установка, настройка и мониторинг прикладного программного обеспечения;

– ведение учетных записей пользователей системы.

– системный администратор обязан владеть значительным уровнем квалификации и практическим опытом выполнения работ по установке, настройке и администрированию программных и технических средств, употребляемых в системе.

Главными обязанностями администратора баз данных являются:

– установка, модернизация, настройка параметров программного обеспечения СУБД;

– оптимизация прикладных баз данных по времени отклика, скорости доступа к данным;

– разработка, управление и исполнение результативной политики доступа к информации, держащейся в теоретических базах данных.

– администратор баз данных должен иметь значительный уровень квалификации и практическим опытом работ по установке, настройке и администрированию используемых в АС СУБД.

Существенными обязанностями администратора информационной безопасности являются:

– разработка, управление и реализация результативной политики информационной безопасности системы;

– правление правами доступа пользователей к функциям системы;

– исполнение мониторинга информационной безопасности.

Администратор информационной безопасности данных должен обладать высоким уровнем квалификации и практическим опытом выполнения работ по обеспечению информационной безопасности.

Пользователи системы обязаны иметь опыт работы с персональным компьютером на базе операционных систем Microsoft Windows на уровне квалифицированного пользователя и легко исполнять базовые операции в стандартных Windows.

Роли системного администратора, администратора баз данных и администратора информационной безопасности могут быть совмещены в роль.

Рекомендуемая численность для эксплуатации ИС «BestLine»:

Администратор - 1 штатная единица;

Пользователь - число штатных единиц определяется структурой заведения.

А.4.1.3 Показатели назначения

ИС «BestLine» обязаны обеспечивать вероятность исторического сохранения данных с глубиной не менее 10 лет.

Система обязана обеспечивать возможность синхронной работы 50 пользователей для подсистемы операционной занятия, и не менее 10-ти пользователей для других подсистем при вытекающих характеристиках времени отклика системы:

– для операций навигации по экранным формам системы - не более 5 сек;

– для операций вырабатывания справок и выписок - не более 10 сек.

Время формирования аналитических отчетов определяется их сложностью и возможно будет занято продолжительное время.

Система обязана предусмотреть вероятность масштабирования по продуктивности и объему обрабатываемой информации без модификации ее программного обеспечения путем улучшения употребляемых технических средств.

А.4.1.4 Требования к надежности

Система обязана беречь работоспособность и обеспечивать воссоздание своих функций при возникновении вытекающих ситуаций:

– при сбоях в системе электроснабжения аппаратной части, приводящих к перезагрузке ОС, восстановление программы следует происходить после перезапуска ОС и запуска осуществляемого файла системы;

– при ошибках в работе аппаратных средств (кроме носителей данных и программ) восстановление функции системы возлагается на ОС;

– при ошибках, связанных с программным обеспечением (ОС и драйверы устройств), восстановление работоспособности возлагается на ОС.

Для защиты аппаратуры от бросков напряжения и коммутационных помех должны использоваться стабилизаторы.

А.4.1.5 Требования к безопасности

Все наружные элементы технических средств системы, находящиеся под напряжением, обязаны обладать защитой от намеренного прикосновения, а сами технические средства иметь занижение или защитное заземление в соответствии с ГОСТ 12.1.030-81.

Система электропитания обязана снабжать защитное отключение при перегрузках и недолгих замыканиях в цепях нагрузки, а также аварийное ручное отключение.

Факторы, проявляющие вредные действия на здоровье со стороны всех элементов системы (в том числе инфракрасное, ультрафиолетовое, рентгеновское и электромагнитное излучения, вибрация, шум, электростатические поля, ультразвук строчной частоты и т.д.), не должны превосходить функционирующих норм.

А.4.1.6. Требования к эргономике и технической эстетике

Взаимодействие пользователей с прикладным программным обеспечением, входящим в состав системы должно осуществляться посредством визуального графического интерфейса (GUI). Интерфейс системы должен быть понятным и удобным, не должен быть перегружен графическими элементами и должен обеспечивать быстрое отображение экранных форм. Навигационные элементы должны быть выполнены в удобной для пользователя форме. Средства редактирования информации должны удовлетворять принятым соглашениям в части использования функциональных клавиш, режимов работы, поиска, использования оконной системы. Ввод-вывод данных системы, прием управляющих команд и отображение результатов их исполнения должны выполняться в интерактивном режиме. Интерфейс должен соответствовать современным эргономическим требованиям и обеспечивать удобный доступ к основным функциям и операциям системы.

Управление системой должно осуществляться с помощью набора экранных меню, кнопок, значков и т. п. элементов. Клавиатурный режим ввода должен используется главным образом при заполнении и/или редактировании текстовых и числовых полей экранных форм.

Система должна обеспечивать корректную обработку аварийных ситуаций, вызванных неверными действиями пользователей, неверным форматом или недопустимыми значениями входных данных.

Система должна соответствовать требованиям эргономики и профессиональной медицины при условии комплектования высококачественным оборудованием (ПЭВМ, монитор и прочее оборудование), имеющим необходимые сертификаты соответствия и безопасности.

А.4.1.7 Требования к эксплуатации, техническому обслуживанию, ремонту и хранению компонентов системы

Техническая и физическая защита аппаратных компонентов системы, носителей данных, бесперебойное энергоснабжение, резервирование ресурсов, текущее обслуживание реализуется техническими и организационными средствами, предусмотренными в ИТ инфраструктуре Заказчика.

Для нормальной эксплуатации разрабатываемой системы должно быть обеспечено бесперебойное питание ПК. При эксплуатации система должна быть обеспечена соответствующая стандартам хранения носителей и эксплуатации ПК температура и влажность воздуха.

Периодическое техническое обслуживание используемого ПК должно проводиться в соответствии с требованиями технической документации изготовителей, но не реже одного раза в год.

Периодическое техническое обслуживание и тестирование ПК должны включать в себя обслуживание и тестирование ПК, кабельной системы, устройств безопасной передачи данных.

В процессе проведения периодического технического обслуживания должны проводиться внешний и внутренний осмотр и чистка ПК, проверка контактных соединений, проверка параметров настроек работоспособности ПК.

А.4.1.8 Требования к защите информации от несанкционированного доступа

ИС должна обеспечивать защиту от несанкционированного доступа (НСД).

Компоненты подсистемы защиты от НСД должны обеспечивать:

– идентификацию пользователя;

– проверку полномочий пользователя при работе с системой;

– разграничение доступа пользователей на уровне задач и информационных массивов.

Протоколы аудита системы и приложений должны быть защищены от несанкционированного доступа как локально, так и в архиве.

А.4.1.9 Требования по сохранности информации при авариях

Программное обеспечение ИС «BestLine» должно восстанавливать свое функционирование при корректном перезапуске аппаратных средств. Обязана быть внезапна вероятность объединения автоматического и
ручного запасного имитирования данных системы оружиями системного и базового программного обеспечения (ОС, СУБД), убирающегося в состав программно технического комплекса Заказчика.

Приведенные выше требования не распространяются на компоненты системы, разработанные третьими сторонами и действительны только при соблюдении правил эксплуатации этих компонентов, включая своевременную установку обновлений, рекомендованных производителями покупного программного обеспечения.

А.4.1.10 Требования к патентной чистоте

Установка системы в целом, как и установка отдельных частей системы не должна предъявлять дополнительных требований к покупке лицензий на программное обеспечение сторонних производителей.

А.4.1.11 Требования по стандартизации и унификации

Экранные формы должны проектироваться с учетом требований унификации:

– все экранные формы пользовательского интерфейса должны быть осуществлены в едином графическом дизайне, с одинаковым расположением главных элементов управления и навигации;

– для обозначения сходных операций должны использоваться сходные графические значки, кнопки и другие управляющие элементы;

– термины, применяемые для обозначения типовых операций а также последовательности действий пользователя при их выполнении, должны быть унифицированы;

– внешнее поведение сходных элементов интерфейса (реакция на наведение указателя "мыши", переключение фокуса, нажатие кнопки) должны реализовываться одинаково для однотипных элементов.

А.4.2 Требования к видам обеспечения

А.4.2.1 Требования к математическому обеспечению системы

Математические методы и алгоритмы, применяемые для хэширования, шифрования данных, а также программное обеспечение, реализующее их, должны быть сертифицированы уполномоченными организациями.

А.4.2.2 Требования к информационному обеспечению системы

Состав, структура и способы организации данных в системе должны быть определены на этапе технического проектирования.

Хранение данных должно осуществляться на основе нынешних реляционных или СУБД. Для обеспечения целостности данных должны использоваться встроенные механизмы СУБД.

Средства СУБД, а также средства употребляемых операционных систем обязаны снабжать документирование и протоколирование отшлифовываемой в системе информации.

Структура базы данных обязана поддерживать кодирование хранимой и обрабатываемой информации в соответствии с общепринятыми классификаторами (там, где они применимы).

Доступ к данным должен быть дан только авторизованным пользователям с учетом их служебных полномочий, а также с учетом категории запрашиваемой информации.

Структура базы данных должна быть организована рациональным способом, исключающим единовременную полную выгрузку информации, содержащейся в базе данных системы.

Технические средства, обеспечивающие сохранение информации, обязаны применить нынешние технологии, разрешающие обеспечить увеличенную безопасность сохранения данных и оперативную подмену оборудования (распределенная избыточная запись/считывание данных; зеркалирование).

В состав системы должна входить специализированная подсистема резервного копирования и восстановления данных.

При проектировании и развертывании системы необходимо рассмотреть возможность использования накопленной информации из уже работающей информационных систем.

А.4.2.3 Требования к лингвистическому обеспечению системы

Все прикладное программное обеспечение системы для организации взаимодействия с пользователем должно применить русский язык.

А.4.2.4 Требования к программному обеспечению системы

При проектировании и разработке системы необходимо максимально эффективным образом применить ранее закупленное программное обеспечение, как серверное, так и для рабочих станций.

А.4.2.5 Требования к техническому обеспечению

Техническое обеспечение системы должно предельно и наиболее результативным образом применить существующие в органах федерального агентства технические средства.

В состав комплекса должны следующие технические средства:

Серверы БД;

Серверы приложений;

Web сервер;

ПК пользователей;

А.4.2.6 Требования к организационному обеспечению

Организационное обеспечение системы следует быть правильным для результативного выполнения персоналом положенных на него обязанностей при реализации автоматизированных и связанных с ними неавтоматизированных функций системы.

Заказчиком должны быть определены должностные лица, ответственные за:

– обработку информации АС;

– администрирование АС;

– обеспечение безопасности информации АС;

– управление работой персонала по обслуживанию АС.

К работе с системой должны допускаться сотрудники, обладающие навыками работы на персональном компьютере, ознакомленные с правилами эксплуатации и прошедшие обучение работе с системой.

А.4.2.7 Требования к методическому обеспечению

В состав методического обеспечения системы должны входить законодательные акты, стандарты, нормативы, инструкции.

А.5 Порядок контроля и приемки системы

А.5.1 Виды, состав, объем и методы испытаний системы

Виды, состав, объем, и методы испытаний подсистемы должны быть изложены в программе и методике испытаний ИС «BestLine», разрабатываемой в составе рабочей документации.

А.5.2 Общие требования к приемке работ по стадиям

Сдача-приёмка работ изготовляется поэтапно, в соответствии с рабочей программой и календарным планом.

Сдача-приемка исполняется комиссией, в состав которой входят представители Заказчика и Исполнителя.

Все создаваемые в рамках настоящей работы программные изделия (за исключением покупных) передаются Заказчику, как в виде готовых модулей, так и в виде исходных кодов, показываемых в электронной варианте на стандартном носителе (например, на флэш карте).

А.5.3 Статус приемочной комиссии

Статус приемочной комиссии определяется Заказчиком до проведения испытаний.

А.6 Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие

В ходе выполнения проекта на объекте автоматизации нужно реализовать работы по подготовке к вводу системы в воздействие. При подготовке к вводу в эксплуатацию ИС «BestLine» Заказчик должен выполнить следующие работы:

Назначить ответственных должностных лиц, ответственных за внедрение и проведение эксплуатации ИС «BestLine»;

Обеспечить присутствие пользователей на обучении работе с системой, проводимом Исполнителем;

Совместно с Исполнителем подготовить план развертывания системы на технических средствах Заказчика;

Провести опытную эксплуатацию ИС «BestLine».

А.7 Требования к документированию

Данный проект сопутствуется ТЗ и документацией на технический проект.

В техническом задании описываться:

- главные цели, задачи, сроки и периоды разработки;
- список важнейших функций и требований;
- список функций интерфейса.

Документация на технический проект является инструкцией по употреблению данного ПО. В данной документации будут описываться:

- условия работы ПО;

- установка ПО;
- применение ПО, по пунктом изображение важнейших функций и функций интерфейса;
- ликвидация проблем при переходе в аварийный режим.

Для системы на различных стадиях создания должны быть выпущены следующие документы из числа предусмотренных в ГОСТ 34.201- «Информационная технология. Комплекс стандартов на автоматизированные системы».

А.8 Источники разработки

Документы и информационные материалы (технико-экономическое обоснование, отчеты о законченных научно-исследовательских работах, информационные материалы на отечественные, зарубежные системы-аналоги и др.), на создании которых разрабатывалось ТЗ и которые обязаны быть применены при основании системы.

Технико-экономическое обоснование. Этот документ хранит, финансовое отображение системы, в котором хранится перечень применяемых ресурсов и показывается их цена. Цена системы, подсчет рентабельности. Минимизация употребляемых ресурсов для того чтобы получать наибольшей прибыли.

Данная система должна разрабатываться на основании ТК 34 по стандартизации Информационные технологии. Номер приказа и дата утверждения: от 27.07.01 г. № 274.

Приложение Б
Листинг программы

```
package com Diplom Program;

  import  javax. swing . *;
  import  java . awt . *;
  import  java . awt .  Event.  Action  Event ;
  import  java . awt .  Event.  Action Listener;
  import  java . io .  Buffered  Reader ;
  import  java . io . IOException;
  import  java . io .  Input  Stream  Reader ;
  import  java . io .  print Writer;
  import  java . net . Socket;
public class   Client Program
{
     J Text  Area incoming  _  msg;
     J Text    Field   outgoing  _  msg;
    J Check Box  imitation  Check Box;

     Buffered  Reader   Reader ;
     print Writer writer;
    Socket socket;

     J Text    Field   login  _    Field  ;
     J Text    Field   password  _    Field  ;
    J Frame  lgn  _   Frame ;

    boolean  imitation Mode = false;

    public static void  main   (  String [] args )  {

          User   Agent  .  Get Instance (  )  .   User    _  accounts . put ( "admin", "adm"
) ;
          User   Agent  .  Get Instance (  )  .   User    _  accounts . put ( "  User  ", "
User  " ) ;

         Client Program   Client  = new   Client Program (  ) ;
         Client  . login (  ) ;
    }

    public boolean  Set  Up Connection (  )  {
       try {
          socket = new Socket ( "127 . 0 . 0 . 1",  Server Program . SRV  _  PORT ) ;
```

```
        Input Stream Reader  Input Stream Reader = new Input Stream Reader
( socket . Get Input Stream ( ) ) ;
// создаем два потока для чтения и отправки сообщений
          Reader = new Buffered Reader ( Input Stream Reader ) ;
          writer = new print Writer ( socket . Get Output Stream ( ) ) ;

          System . out . print ln ( "Соединение с сервером установлено . . . " ) ;

      }catch ( IOException ex ) {
          System . out . print ln ( "Ошибка соединения невозможно установить
соединение с сервером" ) ;
          return false;
      }

      return true;
   }

   public void start ( ) {

      J Frame Frame = new J Frame ( "Клиент пользователя: " + login _ Field
. Get Text ( ) ) ;
      // создаем основную панель программы
      J Pane l main Pane l = new J Pane l ( ) ;
      // создаем текстовое поле для приходящих сообщений
      incoming _ msg = new J Text Area ( 15, 30 ) ;
      incoming _ msg . Set Line Wrap ( true ) ;
      incoming _ msg . Set Wrap Style Word ( true ) ;
      incoming _ msg . Set Editable ( false ) ;
      // создаем скроллер ( прокрутку ) для приходящих сообщений
      J scroll Pane scroll Bar = new J scroll Pane ( incoming _ msg ) ;
       scroll Bar . Set Vertical scroll Bar Policy ( scroll Pane Constants .
VERTICAL _ SCROLL BAR _ ALWAYS ) ;
       scroll Bar . Set Horizontal scroll Bar Policy ( scroll Pane Constants .
HORIZONTAL _ SCROLL BAR _ NEVER ) ;
      // создаем поле для исходящих сообщений
      outgoing _ msg = new J Text Field ( 20 ) ;
      J But to n send But to n = new J But to n ( "Отправить сообщение" ) ;
      // привязываем кнопке отправить обработчик события "отправки"
      send But to n . add Action Listener ( new Send But to n Listener ( ) ) ;

      JLabel imitation Label = new JLabel ( "Иммитировать потерю данных" ) ;
                  imitation Check Box = new J Check Box ( ) ;
```

```
        imitation Check Box . add Action Listener ( new imitation But to n
Listener ( ) ) ;
       // все созданные элементы добавляем в панель для отображения
        main Pane l . add ( scroll Bar ) ;
        main Pane l . add ( outgoing _ msg ) ;
        main Pane l . add ( send But to n ) ;
        main Pane l . add ( imitation Label ) ;
        main Pane l . add ( imitation Check Box ) ;

       if ( ! Set Up Connection ( ) ) {
          System . out . print ln ( "Аварийное завершение клиентской программы"
) ;
          return;
       }

       System . out . print ln ( "Загрузка данных с сервера . . . " ) ;
       // запускаем поток для чтения данных с сервера через ранее
установленное соединение
        Thread Thread = new Thread Incoming Message s Reader ( ) ;
        Thread . start ( ) ;

        Frame . Get Content Pane ( ) . add ( Border Layout . CENTER, main
Pane l ) ;
        Frame . Set Size ( 400, 500 ) ;
        Frame . Set Visible ( true ) ;
        Frame . Set Resizable ( false ) ;

   }

   public class Send But to n Listener implements Action Listener {
      @Override
      public void Action Performed ( Action Event Action Event )
      {
         try
         {
            String messa Get o Server = append User Name to Message (
outgoing _ msg . Get Text ( ) ) ;

            if ( imitation Mode ) {
               // Иммитиация потери данных, отсекаем последний символ
               String old Message = messa Get o Server ;
                                    messa Get o Server = "";
```

```
                char[] chars = old Message . to Char Array ( ) ;
                for ( int i = 0; i < chars . length-1; i++ ) {
                   messa Get o Server += chars[i];
                }
     }


            System . out . print ln ( "System: ----------------------------------" ) ;
            System . out . print ln ( "отправка пакета на сервер: " + messa Get o
Server ) ;
            // соержимое текстового поля отправляется на сервер
            writer . print ln ( messa Get o Server + " MD5 Hash : " + MD5 Hash .
Get Hash ( messa Get o Server ) ) ;
            writer . flush ( ) ;
         }catch ( Exception ex ) {
            ex . print Stack Trace ( ) ;
         }
         outgoing _ msg . Set Text ( "" ) ;
         outgoing _ msg . requestFocus ( ) ;
      }
   }

   public class imitation But to n Listener implements Action Listener {
      @Override
      public void Action Performed ( Action Event Action Event )
      {
          imitation Mode = ! imitation Mode;
         if ( imitation Mode )
            System . out . print ln ( "System: включен режим иммитации потери
данных" ) ;
         else
            System . out . print ln ( "System: отключен режим иммитации потери
данных" ) ;
      }
   }
   public class Login But to n Listener implements Action Listener {
      @Override
      public void Action Performed ( Action Event Action Event )
      {
         if ( User Agent . Get Instance ( ) . password Is Correct ( login _
Field . Get Text ( ) , password _ Field . Get Text ( ) ) ) {
            start ( ) ;
                                    lgn _ Frame . dispose ( ) ;
```

```
        }else
        {
           JOption Pane  . show Message Dialog ( new J Frame  (  ) ,
                 "Ошибка авторизации . ",
                 "Inane error",
```

Продолжение приложения Б

```
                 JOption Pane  . ERROR  _   MESSAGE  ) ;
        }
     }
  }

  public class   Thread  Incoming Message s Reader  extends   Thread   {

     public void run (  )  {

        String   Message ;
        try {
           // пока есть сообщения, будем считывать их, за раз одну строчку
           // и добавлять в поле входящих сообщений
           while  (  (  Message  =  Reader  . readLine (  )  )  != null )
           {
            System . out .  print ln ( "прием пакета от сервера: " +  Message  ) ;

             String []  Message s =  Message  . split ( " MD5 Hash : " ) ;

             String  mes  =  Message s[0];
             String   Hash  =  Message s[1];

             String  current Hash  = MD5 Hash  .  Get  Hash  ( append  User  Name
to  Message  ( outgoing  _  msg .  Get  Text  (  )  )  ) ;
            if  (  imitation Mode )  { //  ( ! current Hash  . equals (  Hash  )  )  {
               System . out .  print ln ( "System: ----------------------------------" ) ;
               System . out .  print ln ( "System: произошла потяря данных  .  .  . "
) ;
               System  .  out  .   print  ln  (  "MD5  Hash  :  хэш  отправленного  и
принятого сообшения НЕ совпадают" ) ;
               System  .  out  .   print  ln  (  "MD5  Hash  :  хэш  отправленного
сообщения " + current Hash  ) ;
               System . out .  print ln ( "MD5 Hash : хэш принятого     сообщения "
+  Hash  ) ;
             }
else {
```

Продолжение приложения Б

```
            System . out .  print ln ( "System: ----------------------------------" ) ;
            System . out .  print ln ( "System: данные доставлены на сервер в
целостности  .  .  . " ) ;
            System . out .  print ln ( "MD5 Hash : хэш отправленного и
принятого сообщения идентичны" ) ;
            System . out .  print ln ( "MD5 Hash : хэш отправленного
сообщения " + current Hash  ) ;
```

Продолжение приложения Б

```
            System . out .  print ln ( "MD5 Hash : хэш принятого  сообщения "
+  Hash  ) ;
             }


              incoming  _  msg . append ( mes + "\n" ) ;
             }
          }catch  ( Exception ex )  {
             ex .  print  Stack Trace (  ) ;
          }

      }
   }

   public void login (  )  {

      lgn  _   Frame  = new J Frame  ( "Авторизация" ) ;
      J Pane l lgn  _   Pane l = new J Pane l (  ) ;
      JLabel login  _  label = new JLabel ( " логин:" ) ;
      login  _   Field   = new  J Text   Field   ( 15 ) ;
      JLabel password  _  label = new JLabel ( "пароль:" ) ;
      password  _   Field   = new  J Text   Field   ( 15 ) ;

      J But to n  login But to n  = new J But to n  ( "Войти" ) ;
      login But to n  .  add  Action Listener ( new Login But to n Listener (  )  ) ;

      lgn  _   Pane l .  add  ( login  _  label ) ;
      lgn  _   Pane l .  add  ( login  _   Field   ) ;
      lgn  _   Pane l .  add  ( password  _  label ) ;
      lgn  _   Pane l .  add  ( password  _   Field   ) ;
      lgn  _   Pane l .  add  ( login But to n  ) ;
      lgn  _   Frame  .  Get Content Pane  (  )  .  add  (  Border Layout . CENTER,
lgn  _   Pane l ) ;
      lgn  _   Frame  .  Set Size ( 250, 200 ) ;
```

```
      lgn _ Frame . Set Visible ( true ) ;
      lgn _ Frame . Set Resizable ( false ) ;

   }

   public String append User Name to Message ( String Message ) {

      return login _ Field . Get Text ( ) + ": " + Message ;
   }
}

package com . Diplom Program;

 import java . io . Uns Up ported Encoding Exception;
 import java . math . Big Integer;
 import java . security . Message Digest;
 import java . security . No Such Algorithm Exception;

public class MD5 Hash
{
   public static String Get Hash ( String str ) throws No Such Algorithm
Exception, Uns Up ported Encoding Exception
   {
      // String s="f78spx";
      // String s="muffin break";
       Message Digest m = Message Digest . Get Instance ( "MD5" ) ;
      m . re Set ( ) ;
      // передаем в Message Digest байт-код строки
      m . Up date ( str . Get Bytes ( "utf-8" ) ) ;
      // получаем MD5-хеш строки без лидирующих нулей
       String s2 = new Big Integer ( 1, m . digest ( ) ) . to String ( 16 ) ;
       String Builder sb = new String Builder ( 32 ) ;
      // дополняем нулями до 32 символов, в случае необходимости
      //System . out . print ln ( 32 - s2 . length ( ) ) ;
      for ( int i = 0, count = 32 - s2 . length ( ) ; i < count; i++ ) {
         sb . append ( "0" ) ;
      }
      // возвращаем MD5-хеш
      return sb . append ( s2 ) . to String ( ) ;

package com . Diplom Program;
```

```
import  java . io .  Buffered  Reader ;
import  java . io .  Input  Stream  Reader ;
import  java . io .  print Writer;
import  java . net .  Server Socket;
import  java . net . Socket;
import  java . util .  Array List;

public class  Server Program
{
   public static int SRV  _  PORT = 4999;

    Array List< print Writer> lis to f  Client  Stream s;

   public static void  main   (  String [] args )  {

       Server Program  Server  = new  Server Program (  ) ;
       Server  . start (  ) ;
   }

   public class   Client Task implements Runnable {

       Buffered  Reader   Reader ;
      Socket socket;

      public   Client Task ( Socket socket )
      {
         try
         {
            this . socket = socket;
            // инициализируем поток для приема сообщений от клиента
             Input  Stream  Reader  Input  Stream  Reader  = new  Input  Stream
Reader  ( socket .  Get  Input  Stream  (  )  ) ;
             Reader  = new  Buffered  Reader  (  Input  Stream  Reader  ) ;
         } catch  ( Exception ex )  {
            ex .  print  Stack Trace (  ) ;
         }
      }
      @Override
      public void run (  )
      {
          String   Message ;
         try {
```

```
            while  (  (  Message  =  Reader  . readLine (  )  )  != null )  {

               Synchronize Message s (  Message  ) ;
            }
         }catch  ( Exception ex )  {
            ex .  print  Stack Trace (  ) ;
         }

      }
   }

   public void start (  )  {

      lis to f  Client  Stream s = new  Array List (  ) ;

      try
      {
          Server Socket  Server Socket = new  Server Socket ( SRV  _  PORT ) ;

         while ( true )
         {
            Socket   Client Socket =  Server Socket . accept (  ) ;
             print Writer writer = new   print Writer (    Client Socket .   Get Output
Stream  (  )  ) ;
            lis to f  Client  Stream s .  add  ( writer ) ;

            System . out .  print ln ( "Синхронизации данных между сервером и
клиентом  .  .  . " ) ;
             Thread    Thread   = new   Thread   ( new   Client Task (   Client Socket
)  ) ;
             Thread   . start (  ) ;
         }


      }catch  ( Exception ex )  {
         ex .  print  Stack Trace (  ) ;
      }

   }

   public void Synchronize Message s (  String   Message  )  {
      for  (  print Writer writer : lis to f  Client  Stream s )  {
```

```
        try {

            writer .  print ln (  Message  ) ;
            writer . flush (  ) ;

        } catch  ( Exception ex )  {
            ex .  print  Stack Trace (  ) ;
        }
      }
   }

}
package com .   Diplom  Program;

 import  java . io . Uns Up ported Encoding Exception;
 import  java . math .  Big  Integer;
 import  java . security .  Message Digest;
 import  java . security . No Such Algorithm Exception;

public class MD5 Hash
{
   public static  String   Get  Hash  (  String  str )  throws  No Such Algoritm
Exception, Uns Up  ported Encoding Exception
   {
      // String  s="f78spx";
      // String  s="muffin break";
       Message Digest m =  Message Digest .  Get Instance ( "MD5" ) ;
      m . re Set  (  ) ;
      // передаем в  Message Digest байт-код строки
      m .  Up date ( str .  Get Bytes ( "utf-8" )  ) ;
      // получаем MD5-хеш строки без лидирующих нулей
       String  s2 = new  Big Integer ( 1, m . digest (  )  )  .  to  String  ( 16 ) ;
       String Builder sb = new  String Builder ( 32 ) ;
      // дополняем нулями до 32 символов, в случае необходимости
      //System . out .  print ln ( 32 - s2 . length (  )  ) ;
      for  ( int i = 0, count = 32 - s2 . length (  ) ; i < count; i++ )  {
         sb . append ( "0" ) ;
      }
      // возвращаем MD5-хеш
      return sb . append ( s2 )  .  to  String  (  ) ;
   }
```

```
package com .   Diplom  Program;

 import  java . util .  Hash Map;
 import  java . util . Map;


public class   User   Agent
{
   public Map< String ,  String >   User    _  accounts = new  Hash Map<> ( ) ;
   private   User   Agent  ( )  {

   }

   private static class   User   Agent Holder {
      private final static   User   Agent  instance = new   User   Agent  ( ) ;
   }

   public static   User   Agent   Get Instance ( )  {
      return   User   Agent Holder . instance;
   }

   public boolean  password Is Correct (  String    User    _ name,  String    User    _
password )  {

      boolean isCorrect = false;

      for  ( Map . Entry< String ,  String > pair :   User    _  accounts . entry Set  ( )
)  {
         if  ( pair .  Get Key ( )  . equals (   User    _  name . trim ( )  .  to LowerCase
( )  )  )  {
            if  ( pair .  Get Value ( )  . trim ( )  .  to LowerCase ( )  . equals (   User
_  password )  )  {
               isCorrect = true;
               break;
            }
         }
      }

      return isCorrect;
```

```
 import  java . applet . *;
 import  javax . swing . *;
 import  java . awt . *;
 import  java . awt .  Event  . *;

public class Diolog {

        public static void  main  (  String [] args )  {
                //  TO DO Au to -generated method stub
System . out .  print ln ( "This is my programm" ) ;
My Frame   Frame  = new My Frame  (  ) ;
Frame  .  Set Default Close Operation ( J Frame  . EXIT  _  ON  _   CLOSE  ) ;
 Frame  . show (  ) ;

class My Frame  extends J Frame  {
        private  J Text  Area  Text ;
        private J scroll  Pane   scroll  Pane ;
        private J Pane l b Pane l;

        public My Frame  (  ) {
                 Set Title ( " main   Frame " ) ;
                 Set Size ( 400,300 ) ;
                Container  Pane = Get Content Pane  (  ) ;
                b Pane l = new J Pane l (  ) ;
                J But to n  i But to n = new J But to n  ( "Отправить" ) ;
                b Pane l .  add  ( i But to n  ) ;

                i But to n  .  add  Action Listener  ( new
                                Action Listener (  )  {
                        public void  Action Performed (  Action  Event   Event  )
                        {
                                Text  . append ( "moy programma" ) ;

                        }
                } ) ;

                 Pane  .  add  ( b Pane l,  Border Layout . SOUTH ) ;
                 Text  = new  J Text  Area  ( 10,30 ) ;
                 scroll  Pane = new J scroll  Pane  (  Text  ) ;
                 Pane  .  add  (  scroll  Pane ,  Border Layout . CENTER ) ;

}
```

```
}
        }
}

  import  java . awt .  Border Layout;
  import  java . awt . Container;
  import  java . awt .  Event  .  Action  Event ;
  import  java . awt .  Event  .  Action Listener;

  import  javax . swing . J But to n ;
  import  javax . swing . J Frame ;
  import  javax . swing . J Pane l;
  import  javax . swing . J scroll  Pane ;
  import  javax . swing .  J Text  Area;


class My Frame  extends J Frame  {
        private  J Text  Area  Text ;
        private J scroll  Pane   scroll  Pane ;
        private J Pane l b Pane l;

        public My Frame  (  ) {
                Set Title ( " main   Frame " ) ;
                Set Size ( 400,300 ) ;
               Container  Pane = Get Content Pane  (  ) ;
               b Pane l = new J Pane l (  ) ;
               J But to n  i But to n = new J But to n  ( "Отправить" ) ;
               b Pane l .  add  ( i But to n  ) ;

               i But to n  .  add  Action Listener  ( new
                              Action Listener (  )  {
                       public void  Action Performed (  Action  Event   Event  )
                       {
                              Text  . append ( "moy programma" ) ;

                       }
               } ) ;

                Pane  .  add  ( b Pane l,  Border Layout . SOUTH ) ;
                Text  = new  J Text  Area  ( 10,30 ) ;
                scroll  Pane = new J scroll  Pane  (  Text  ) ;
```

```
                Pane . add ( scroll Pane , Border Layout . CENTER ) ;
}
}

 import  java . awt . Event Queue;
 import javax . swing . *;
 import java . awt . Event . Action Listener;
 import java . awt . Event . Action Event ;
 import java . awt . to olkit;
 import java . awt . Color;
 import java . awt . Font;
 import java . io . IOException;

public class Gui {

   private J Frame  Frame ;
   private J Text  Field  login;
   private JPassword Field  password;
   private JEdi to r Pane news;
   private J Text  Field  qwerty;


   /**
    * Launch the application .
    */
   public static void main ( String [] args ) {
       Event Queue . invoke Later ( new Runnable ( ) {
         public void run ( ) {
            try {
               Gui window = new Gui ( ) ;
               window . Frame . Set Visible ( true ) ;
            } catch ( Exception e ) {
               e . print Stack Trace ( ) ;
            }
         }
      } ) ;
   }

   /**
    * Create the application .
    */
   public Gui ( ) {
      initialize ( ) ;
```

```
    /**
     * Initialize the contents of the  Frame  .
     */
    private void initialize (  )  {
        Frame  = new J Frame  (  ) ;

        Frame  .  Set Bounds ( 300, 300, 300, 300 ) ;
        Frame  .  Set Default Close Operation ( J Frame  . EXIT  _  ON  _   CLOSE  )
;
        Frame  .  Set Resizable ( false ) ;
        Frame  .  Set Title ( "Вход с систему" ) ;
        Frame  .  Get Content Pane  (  )  .  Set Layout ( null )
       // But to n  enter
       J But to n  enter = new J But to n  ( "ok" ) ;

       enter .  Set Bounds ( 50, 170, 70, 31 ) ;
        Frame  .  Get Content Pane  (  )  .  add  ( enter ) ;
       enter .  add  Action Listener ( new  Action Listener (  )  {
       public void  Action Performed (  Action  Event  e )  {

          }
       } ) ;


       // But to n  exit
       J But to n  exit = new J But to n  ( "exit" ) ;

       exit .  Set Bounds ( 130, 170, 70, 31 ) ;
        Frame  .  Get Content Pane  (  )  .  add  ( exit ) ;
       exit .  add  Action Listener ( new  Action Listener (  )  {
       public void  Action Performed (  Action  Event  arg0 )  {
        Frame  . dispose (  ) ;
          }
       } ) ;
       // Text  login
       login = new  J Text    Field   (  ) ;
       login .  Set  Text  ( "Login" ) ;
       login .  Set Foreground ( Color . black ) ;
       login .  Set Font ( new Font ( "Arial", Font . PLAIN, 14 )  ) ;
       login .  Set Background ( Color . white ) ;
       login .  Set Bounds ( 50, 70, 150, 31 ) ;
```

```
 Frame  .  Get Content Pane  (  )  .  add  ( login ) ;
login .  Set Columns ( 10 ) ;

// Text  password
password = new JPassword  Field   (  ) ;
password .  Set  Text  ( "qwerty12" ) ;
password .  Set Foreground ( Color . black ) ;
password .  Set Background ( Color . white ) ;
password .  Set Bounds ( 50, 120, 150, 31 ) ;
 Frame  .  Get Content Pane  (  )  .  add  ( password ) ;

// Pane l news
news = new JEdito r Pane  (  ) ;
news .  Set Back ground ( Color . white ) ;
news .  Set Editable ( false ) ;
news .  Set Bounds ( 0, 0, 594, 317 ) ;
 Frame  .  Get Content Pane  (  )  .  add  ( news ) ;


// Pane l login
JLabel  Pane l = new JLabel ( "" ) ;

 Pane l .  Set Bounds ( 250, 500, 250, 250 ) ;
 Frame  .  Get Content Pane  (  )  .  add  (  Pane l ) ; } }
```