Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Факультет комп’ютерних наук

Кафедра програмного забезпечення ЕОМ

КУРСОВИЙ ПРОЕКТ

ПОЯСНЮВАЛЬНА ЗАПИСКА

WEB сервіс пошуку терміналів банку

(Тема курсового проекту )

Студент гр. ПЗАСс-10-2 Яворський С.В

Керівник проекту асистент Новіков Ю. С

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Кафедра: Програмного забезпечення ЕОМ

Дисципліна: Комплексний курсовий проект

Спеціальність: Програмне забезпечення автоматизованих систем

Курс 5. Група ПЗАСс-10-2. Семестр ІХ.

ЗАВДАННЯ

на курсовий проект студента

Яворського Сергія Володимировича

1. Тема проекту: Автоматизована система пошуку терміналів банку
2. Термін здачі студентом закінченого проекту: 15 – січня – 2011 р.
3. Вихідні дані до проекту: Дані банків, координати розташування - терміналів, відділень банку
4. Зміст розрахунково-пояснювальної записки: Введення, аналіз предметної області банківської системи, огляд існуючих систем банківського напрямку, управління системою пошуку банків та їх терміналів, електронні гроші, постановка задачі, обґрунтування вибору технології розроблення, програмного середовища та мови програмування, мова програмування, технологія розроблення ASP.NET, середовище розроблення Visual Studio 2008, розроблення структури системи менеджменту процессу знаходження банківських терміналів: модулі моніторингу появи нових терміналів та відділень банків, проектування програмної системи, визначення функціональної моделі системи, визначення структури бази даних, описання програмної реалізації системи, висновок, перелік посилань
5. Перелік графічного матеріалу (зміст слайдів на презентацію): Актуальність, мета розробки, класи користувачів, декомпозиція функції навчання, схема бази даних, інтерфейс користувачів, висновок
6. КАЛЕНДАРНИЙ ПЛАН

|  |  |  |
| --- | --- | --- |
| № | Назва етапу | Термін виконання |
| 1 | Видача теми ККП, узгодження і затвердження теми  | 20-10-2010 – 22-10-2010 р.  |
| 2 | Огляд існуючих систем, постановка задачі, вибір середовища проектування. | 22-10-2010 – 06-10-2010 р.  |
| 3 | Формування функціональних вимог щодо системи, яка проектується. | 06-11-2010 – 14-11-2010 р.  |
| 4 | Об'єктна декомпозиція системи, яка проектується, її моделювання. | 14-11-2010 – 26-03-2010 р.  |
| 5 | Розробка бази даних. | 26-11-2010 – 06-11-2010 р.  |
| 6 | Кодування програмної системи, яка проектується. | 06-12-2010 – 28-12-2010 р.  |
| 7 | Тестування і доопрацювання розробленої програмної системи. Розробка програмної документації. | 28-12-2010 – 03-01-2011 р.  |
| 8 | Оформлення пояснювальної записки, додатків, графічного матеріалу. | 03-01-2011 – 05-01-2011 р.  |
| 10 | Перевірка виконаного проекту керівником, допуск до захисту  | 05-01-2011 – 10-01-2011 р.  |
| 11 | Захист проекту  | 10-01-2011 – 15-01-2011 р.  |

Студент гр. ПЗАСс-10-2 Яворський С.В.

Керівник асистент Новіков Ю.С.

(прізвище, ім'я, по-батькові)

«15» січня 2011 р.

РЕФЕРАТ / ABSTRACT

Пояснювальна записка до комплексного курсового проекту : 39 ст., 4 рис., 1 табл., 14 джерел.

Об’єктом дослідження є процес взаємодії сервісу пошуку з клієнтом.

Методом розробки обрано технологію систем автоматизованого пошуку терміналів: надання широкого спектру інформації про існуючі банки, надання інформації про місце їх знаходження. Середовище для розробки системи взаємодії обрано Visual Studio 2008, мова програмування – C#.

Метою роботи є розробка системи для моніторингу та пошуку терміналів, відділень банків. Дана система дозволить створити загальний робочий простір для всіх учасників процесу пошуку, яке інформуватиме про стан процесу знаходження. Також система створює ресурс для учбових матеріалів в Інтернеті. Робота ґрунтується на досвіді міжнародної співпраці кафедри ПОЕВМ Харківського національного університету радіоелектроніки по підготовці фахівців в області розробки програмного забезпечення. Основною метою даної роботи є спрощення пошуку відделень банків та банкоматів на території нашої країни.

ПОШУК, ВІДДІЛЕННЯ, БАНК, МОНІТОРИНГ, МЕНЕДЖМЕНТ, , ASP.NET, СХЕМА БАЗИ ДАНИХ.

Explanatory note to the complex course of the project: 53 sec., 35 fig., 1 tab., 15 sources.

The object of research is a process of interaction with client service search.

The method chosen technology development of automated search terminals: pektru provide wide information about existing banks, provide information on where to find them. A framework for developing a system of interaction selected Visual Studio 2008 programming language - C #.

The aim is to develop a system for management, monitoring and retrieval terminals and bank branches. This system will create a common workspace for all participants of the search, which inform the process of finding the state. Also, the system creates a resource for educational materials on the Internet. The work is based on the experience of international cooperation department POEVM Kharkiv National University of Radio Electronics in training specialists in the field of software development. The main purpose of this paper is to simplify the search bank branches and atms in the country.

SEARCH, BRANCH, BANK, MONITORING, MANAGEMENT, ASP.NET, DATABASESHEMA, WCF.

ПЕРЕЧЕНЬ УСЛОВНЫХ СОКРАЩЕНИЙ

ADO – ActiveX Data Objects

ASP – Active Server Pages

ECMA – European Carton Makers Association

IDEF – Integrated DEFinition

IIS – Internet Information Service

ISO – International Organization for Standardization

UI – User Interface

URL – Universal Resource Locator

ВВЕДЕНИЕ

С развитием научно-технического прогресса и электронных технологий современный человек оказался буквально окружен электронными интерактивными сервисами. Банкоматы, терминалы, автозаправочные колонки, персональные компьютеры, сотовые телефоны, электронные органайзеры (КПК и смартфоны) – без этого сложно представить жителя ХХІ века. Они облегчают нашу жизнь, делают ее мобильнее, удобнее, комфортнее. Поэтому очень важно, чтобы нахождение того или иного банкомата или банковского отделения было быстрым и безболезненным. Именно поэтому данная разработка является актуальной.

На данный момент такова проблема стоит очень остро, так как в нашей стране официально зарегистрировано 170 банков, половина из которых имеет большую сеть терминалов во всех населенных пунктах Украины. Количество банков так велико, что зачастую предприятия и фирмы на которых работает население, используют не общий для всех банк, а разные банки. Из-за этого возникает проблема – люди пользуются услугами разных банков.

Зачастую, если человек хочет снять наличные, он хочет это сделать как можно скорее, то есть речь может идти не о прямом посещении клиентом отделения банка, а о снятии наличных в терминалах банка. Терминалы банков в основном разбросаны по городу, и найти их не так уж легко, они установлены в супермаркетах, магазинах, аптеках. То есть, там, где человек бывает как можно чаще и тем самым может всегда воспользоваться услугами банкоматов.

Зачастую люди получают зарплату “на карточки”, такая же ситуация и со стипендиями и пенсией. Но большинство людей предпочитают сразу же снимать деньги со счёта, доверяя наличным.

Банковские карты, несомненно, имеют ряд достоинств и недостатков. Среди достоинств стоит отметить такие как маленький размер, нет необходимости носить с собой большую сумму наличных денег, при поездки за рубеж нет проблемы с обменом валюты, а также простота и удобство использования.

К сожалению, наша страна не Канада, в которой всего 4 банка на страну, сеть терминалов огромна, и принявшись за поиски банкомата не факт, что за углом стоит именно тот терминал, который необходим Вам.

Именно с целью поиска терминала нужного банка и разрабатывается наш проект. Система позволит найти необходимый терминал банка в считанные секунды, выведет на карту ближайшие банкоматы в зависимости от местонахождения пользователя.

1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ АВТОМАТИЗИРОВАННОГО ПОИСКА

* 1. Обзор существующих систем автоматизированного поиска

Автоматизация банковских технологий в нашей стране прошла несколько этапов своего развития. Первоначально это были достаточно простые программные продукты, которые автоматизировали отдельные аспекты банковской деятельности на базе традиционных СУБД.

Процесс автоматизации банковских технологий перешел на новый этап в конце 80-х начале 90-х годов. Это напрямую связано с банковской реформой 1989 года, когда на рынке банковских услуг появились коммерческие банки (КБ).

С развитием финансового и фондового рынков сфера деятельности КБ расширялась, возрос и объем перерабатываемой информации.

В новых условиях стал неизбежным переход к комплексной автоматизации банковской деятельности. В силу различия банков по размерам, структуре, используемой методологии, т.е. всех тех параметров, которые, в первую очередь, характеризуют банк как объект приложения информационных технологий, расширился круг используемых ими АБС.

На рынке программных средств банковских технологий появились организации поставщики, такие как, «Программ банк», «Инверсия», «Асофт», «Rstyle», «Diasoft» и другие, которые начали активно удовлетворять имеющийся спрос.

Выбор банками тех или иных систем автоматизации связан, как правило, с соотношением цена – надежность – производительность. Многим банкам, имеющим разнородный компьютерный парк, широкую сеть филиалов и отделений, приходится решать проблему не только собственной сетевой интеграции, но переходить на планирование всей системной инфраструктуры информационной технологии.

В инфраструктуре следует выделить пять составляющих:

-информационное обеспечение;

-техническое оснащение;

-программные средства;

-системы связи и коммуникации (внутренние и внешние );

-системы безопасности, защиты и надежности.

Состав информационного обеспечения, его организация определяются составом поставленных перед банком задач. К традиционным для любого банка задачам относится операционная (расчетно-кассовая) деятельность. Автоматизация этого участка работы может решить многие проблемы большинства малых и средних банков на сегодняшний день. При таком подходе банковская технология строится на программном продукте «Операционный день банка» (ОДБ). Он включает в себя такие программы, как «Ведение банковских договоров», «Платежные поручения», «Касса», «Ведение неторговых операций», «Ведение переводных операций» и другие.

Для обеспечения комплексности автоматизации банковской деятельности требуется ряд важных программных средств, позволяющих оценить состояние банка на любой момент времени, вести скоростной обмен информацией со своими филиалами и отделениями, а также с другими банками, осуществлять разноску сумм по корреспондентским счетам, их обработку и другие функции. Сюда можно отнести так называемую систему «Клиент‑банк», дающую возможность клиенту банка осуществлять платежи и проводить другие операции, минуя операциониста и не выходя из своего офиса.

Важным традиционным направлением банковской деятельности является кредитование, приносящее, как правило, до 75% дохода банка. Автоматизация этой сферы деятельности позволяет не только вести автоматизированный контроль за прохождением платежей, но и, что наиболее важно, прогнозировать состояние банка как с точки зрения получения денег, так и по предстоящем выплатам по привлеченным средствам.

Кроме традиционных направлений в комплексную систему организации деятельности органично должно входить решение таких задач, как автоматизация работы с ценными бумагами, дилинг, биржевые операции, организация межбанковского обмена электронными копиями документов, аналитическая оценка деятельности банка и его клиентов и многие другие.

Решение комплексных задач автоматизации возможно лишь с привлечением современных программно‑аппаратных средств. Поэтому сейчас наметилась тенденция приобретения банками мощных компьютеров и развитого ПО. Наряду с этим, банки активно разрабатывают собственное ПО. Расширяется использование банками сетевых технологий.

* 1. Мир электронных денег

Электронные деньги — это денежные обязательства эмитента в электронном виде, которые находятся на электронном носителе в распоряжении пользователя. Такие денежные обязательства соответствуют следующим трем критериям:

* Фиксируются и хранятся на электронном носителе.
* Выпускаются эмитентом при получении от иных лиц денежных средств в объёме не меньшем, чем эмитированная денежная стоимость.
* Принимаются, как средство платежа другими (помимо эмитента) организациями.

Термин «электронные деньги» является относительно новым и часто применяется к широкому спектру платежных инструментов, которые основаны на инновационных технических решениях. Электронным деньгам свойственно внутреннее противоречие — с одной стороны они являются средством платежа, с другой — обязательством эмитента, которое должно быть выполнено в традиционных неэлектронных деньгах. Такой парадокс можно пояснить с помощью исторической аналогии: в свое время банкноты тоже рассматривались, как обязательство, которое подлежит оплате монетами или драгоценными металлами. Очевидно, что с течением времени, электронные деньги будут являться одной из разновидностей формы денег (монеты, банкноты, безналичные деньги и электронные деньги). Так же очевидно, что в будущем центробанки будут производить эмиссию электронных денег, так же как сейчас чеканят монету и печатают банкноты.

Электронные деньги, являясь неперсонифицированным платежным продуктом, могут иметь отдельное обращение, отличное от банковского обращения денег, однако могут и обращаться в т.ч и в государственных или банковских платежных системах.

Как правило, обращение электронных денег происходит при помощи компьютерных сетей, Интернета, платёжных карт, электронных кошельков и устройств, работающих с платежными картами (банкоматы, POS-терминалы, платежные киоски и т. д.). Также, используются и другие платежные инструменты различной формы: браслеты, брелоки, блоки мобильных телефонов и т. д., в которых есть специальный платежный чип. Электронные деньги являются составной частью электронной экономики.

Электронные деньги обычно разделяют на два типа: на базе смарт-карт (англ. card-based) и на базе сетей (англ. network-based). И первая, и вторая группа подразделяются на анонимные (неперсонифицированные) системы, в которых разрешается проводить операции без идентификации пользователя и не анонимные (персонифицированные) системы, требующие обязательной идентификации пользователя.

Следует также различать электронные фиатные деньги и электронные нефиатные деньги. Электронные фиатные деньги, обязательно выражены в одной из государственных валют и являются разновидностью денежных единиц платежной системы одного из государств. Государство законами обязывает всех граждан принимать к оплате фиатные деньги. Соответственно, эмиссия, обращение и погашение электронных фиатных денег происходит по правилам национальных законодательств, центробанков или других государственных регуляторов. Электронные нефиатные деньги — являются электронными единицами стоимости негосударственных платежных систем. Соответственно, эмиссия, обращение и погашение (обмен на фиатные деньги) электронных нефиатных денег, происходят по правилам негосударственных платежных систем. Степень контроля и регулирования государственными органами таких платежных систем в разных странах сильно отличаются. Часто, негосударственные платежные системы привязывают свои электронные нефиатные деньги к курсам мировых валют, однако государства никак не обеспечивают надежность и реальную ценность таких стоимостных единиц. Электронные нефиатные деньги являются разновидностью кредитных денег.

Одной из распространенных ошибок есть отнесение к электронным деньгам современных средств доступа к банковскому счету, а именно, традиционных банковских платежных карт (как микропроцессорных, так и с магнитной полосой), а также интернет-банкинга. В системах, которые осуществляют расчеты электронными деньгами, банковские счета используются только при вводе и выводе денег из системы. При этом используется консолидированный банковский счет эмитента электронных денег, а не карточные или текущие счета пользователей. При эмиссии электронных денег, традиционные деньги зачисляются на консолидированный банковский счет эмитента. При предъявлении электронных денег для погашения, традиционные деньги списываются с консолидированного банковского счета эмитента.

Ещё одной типичной ошибкой является отнесение к электронным деньгам предоплаченных одноцелевых карт, таких как: подарочная карта, топливная карта, телефонная карта и т. д. Использование такого платёжного инструмента не означает осуществления нового платежа. Реальный платёж осуществляется в момент покупки или пополнения такой карты. Её использование не порождает новых денежных потоков и является простым обменом информации о потребленных товарах или услугах.

В 1996 году руководители центробанков стран G10, заявили про намерение осуществлять мониторинг электронных денег в странах мира. С этого времени, «Банк международных расчётов» при поддержке мировых центробанков регулярно анализирует развитие электронных денег и соответствующих систем. Сначала данные были конфиденциальными и были доступными только центробанкам, а с мая 2000 года стали общедоступными. В исследовании 2004 года приняли участие центробанки 95 стран и выяснилось, что электронные деньги функционируют в 37 странах мира[3].

Электронные деньги могут быть анонимными и персонифицированными. По своей природе, электронные деньги ближе к анонимным наличным деньгам, чем к персонифицированным безналичным. Наличие или отсутствие анонимности обеспечивается правилами и механизмами обращения электронных денег в определенной платежной системе.

Большинство государственных регуляторов, а также государственные и негосударственные платежные системы, различными способами пытаются стимулировать персонификацию пользователей электронных денег и операции с ними (см. знай своего клиента). Например, для электронных денег на базе сетей, платежные системы ограничивают размер электронного кошелька для анонимного пользователя, увеличивая лимиты персонифицированным пользователям системы. Для электронных денег на базе карт, ограничивают максимальную сумму в кошельке и вводят персонализированные механизмы пополнения.

В настоящее время электронные деньги рассматриваются, как потенциальный заменитель наличности для микро-платежей. Однако по своим качествам, электронные деньги способны частично заменить или полностью вытеснить при расчетах наличные деньги. Искусственное ограничение суммы, которая может сберегаться в электронном кошельке, вызвано неуверенностью регуляторов в надежности и безопасности использования такого платежного инструмента. Очевидно, что при отсутствии негативных примеров, этот лимит будет увеличиваться или совсем отменен.

Преимущества электронных денег заключаются в том, что они особенно полезны и удобны при осуществлении массовых платежей небольших сумм. Например, при платежах в транспорте, кинотеатрах, клубах, оплате коммунальных услуг, оплате различных штрафов, расчетах в интернете и т. д. Процесс платежа электронными деньгами осуществляется быстро, не возникает очередей, не надо выдавать сдачу, деньги переходят от плательщика к получателю быстро.

Электронные деньги корректнее всего сравнивать с наличными деньгами, так как обращение безналичных денег, обязательно персонифицировано и известны реквизиты обеих сторон. В случае расчётов электронными деньгами, достаточно знать реквизиты получателя денег.

Электронные деньги имеют следующие преимущества перед наличными деньгами:

* превосходная делимость и объединяемость — при проведении платежа не возникает необходимость в сдаче;
* высокая портативность — величина суммы не связана с габаритными или весовыми размерами денег, как в случае с наличными деньгами;
* очень низкая стоимость эмиссии электронных денег — не надо чеканить монеты и печатать банкноты, использовать металлы, бумагу, краски и т. д.;
* не нужно физически пересчитывать деньги, эта функция переносится на инструмент хранения или платежный инструмент;
* проще, чем в случае с наличными деньгами, организовать физическую охрану электронных денег;
* момент платежа фиксируется электронными системами, воздействие человеческого фактора снижается;
* при платеже через фискализированное эквайринговое устройство торговцу невозможно укрыть средства от налогообложения;
* электронные деньги не нужно пересчитывать, упаковывать, перевозить и организовывать специальные хранилища;
* идеальная сохраняемость — электронные деньги не теряют своих качеств с течением времени;
* идеальная качественная однородность — отдельные экземпляры электронных денег не обладают уникальными свойствами (как, например, царапины на монетах);
* безопасность — защищенность от хищения, подделки, изменения номинала и т. п., обеспечивается криптографическими и электронными средствами.

Недостатки электронных денег

* отсутствие устоявшегося правового регулирования, — многие государства ещё не определились в своем однозначном отношении к электронным деньгам;
* несмотря на отличную портативность, электронные деньги нуждаются в специальных инструментах хранения и обращения;
* как и в случае наличных денег, при физическом уничтожении носителя электронных денег, восстановить денежную стоимость владельцу невозможно;
* отсутствует узнаваемость — без специальных электронных устройств нельзя легко и быстро определить, что это за предмет, сумму и т. д.;
* средства криптографической защиты, которыми защищаются системы электронных денег ещё не имеют длительной истории успешной эксплуатации;
* теоретически, заинтересованные лица могут пытаться отслеживать персональные данные плательщиков и обращение электронных денег вне банковской системы;
* безопасность (защищенность от хищения, подделки, изменения номинала и т. п.) — не подтверждена широким обращением и беспроблемной историей;
* теоретически возможны хищения электронных денег, посредством инновационных методов, используя недостаточную зрелость технологий защиты.

Ниже приведена схема платежа с помощью цифровых денег.

Покупатель заранее обменивает реальные деньги на электронные. Хранение наличности у клиента может осуществляться двумя способами, что определяется используемой системой:

-на жестком диске компьютера.

-на смарт-картах.

Разные системы предлагают разные схемы обмена. Некоторые открывают специальные счета, на которые перечисляются средства со счета покупателя в обмен на электронные купюры. Некоторые банки могут сами эмитировать электронную наличность. При этом она эмитируется только по запросу клиента с последующим ее перечислением на компьютер или карту этого клиента и снятием денежного эквивалента с его счета. При реализации же слепой подписи покупатель сам создает электронные купюры, пересылает их в банк, где при поступлении реальных денег на счет они заверяются печатью и отправляются обратно клиенту.

Наряду с удобствами такого хранения, у него имеются и недостатки. Порча диска или смарт-карты оборачивается невозвратимой потерей электронных денег. Покупатель перечисляет на сервер продавца электронные деньги за покупку. Деньги предъявляются эмитенту, который проверяет их подлинность.

В случае подлинности электронных купюр счет продавца увеличивается на сумму покупки, а покупателю отгружается товар или оказывается услуга.

Одной из важных отличительных черт электронных денег является возможность осуществлять микроплатежи. Это связано с тем, что номинал купюр может не соответствовать реальным монетам (например, 37 копеек). Эмитировать электронные наличные могут как банки, так и небанковские организации. Однако до сих пор не выработана единая система конвертирования разных видов электронных денег. Поэтому только сами эмитенты могут гасить выпущенную ими электронную наличность. Кроме того, использование подобных денег от нефинансовых структур не обеспечено гарантиями со стороны государства. Однако, малая стоимость транзакции делает электронную наличность привлекательным инструментом платежей в Интернет.

1.3 Постановка задачи

Неизбежность соответствия украинской банковской системы к европейским стандартам требует интеграции текущей системы в общеевропейскую систему. На первом этапе необходимо упорядочить огромное количество отделений, банкоматов различных банков. Чтобы упростить данный процесс, необходимо создать единое «рабочее пространство» для пользователей банковскими услугами, в котором бы аккумулировалась и обрабатывалась вся необходимая информация о банках.

Основываясь на опыте международного сотрудничества кафедры ПОЭВМ Харьковского национального университета радиоэлектроники по подготовке специалистов в области разработки программного обеспечения, ставится задача создать систему автоматизированного поиска. Основная трудность возникает в управлении процессом поиска из-за территориальной удаленности банковских отделений и терминалов и отсутствия единого информационного пространства для эффективного обмена информацией в процессе автоматизированного поиска.

Целью разработки системы является автоматизация процесса менеджмента и мониторинга поиска с целью эффективного взаимодействия между банками и их пользователям.

Задачами системы являются:

* формализация и накопление информации о банках, их отделениях, терминалах;
* модернизация и ускорение поиска;
* менеджмент процесса автоматизированного поиска;
* мониторинг процесса поиска терминалов и отделений банков.

В данной работе ставится задача реализовать подсистему, которая отвечает за менеджмент и мониторинг банковских отделений и банкоматов.

2 ОБОСНОВАНИЕ ВЫБОРА ТЕХНОЛОГИИ РАЗРАБОТКИ, ПРОГРАММНОЙ СРЕДЫ И ЯЗЫКА ПРОГРАММИРОВАНИЯ

2.1 Язык программирования C#

Этот мощный и одновременно гибкий язык вызывает у программистов чувство гармонии при написании программ. Язык C# заслужил большое уважение и популярность среди разработчиков самых разных программных продуктов. Последнюю пару лет C# играл важную роль в производстве устойчивых к сбоям продуктов – от настольных приложений до Web-сервисов, от высокоуровневых решений в автоматизации бизнес-процессов до программ системного уровня и от однопользовательских продуктов до корпоративных решений в сетевых распределенных средах.

Язык C# сертифицирован Международной организацией по стандартизации (ISO). Таким образом, эта разработка Microsoft имеет статус международных стандартов. Также, язык C# признан в качестве стандарта Европейской ассоциацией производителей компьютеров (ECMA).

Язык программирования С# имеет ряд преимуществ:

* + подлинная объектная ориентированность (всякая языковая сущность претендует на то, чтобы быть объектом);
	+ компонентно-ориентированное программирование;
	+ безопасный (по сравнению с языками C и C++) код;
	+ унифицированная система типизации;
	+ поддержка событийно-ориентированного программирования;
	+ «родной» язык для создания приложений в среде .NET;
	+ объединение лучших идей современных языков программирования: Java, C++, Visual Basic и др.

Исходя с достоинств и с выше перечисленных преимуществ языка C#, в качестве языка для разработки системы был выбран именно этот объектно-ориентированный язык.

2.2 Технология разработки ASP NET

На текущий момент для построения web-приложений используются различные технологии и подходы. Создавая новый сайт, разрабатывая информационную систему, команда разработчиков часто сталкивается с проблемой – какую технологию выбрать для решения поставленной задачи. Предпочтения обычно отдаются знакомым технологиям, или же диктуются рамками поставленной задачи. Но независимо от выбранной технологии, каждый раз при создании нового web-сайта, зачастую приходится проходить множество одних и тех же этапов разработки.

Подходящей и удобной платформой для создания порталов можно использовать среду разработки Visual Studio .NET и технологию ASP.NET. Эта технология позволяет использовать готовые компонентные архитектуры, например, элементы управления ASP.NET, не теряя при этом гибкость разрабатываемой системы [7].

Преимущество технологии ASP.NET перед остальными в высокой степени абстракции, построенной над стандартным HTML кодом: использование объектно-ориентированной парадигмы, поддержку нескольких языков программирования, наличие универсального основания, содержащего тысячи уже готовых для использования в проектах решений – Microsoft .NET Framework. По сравнению с другими web-технологиями имеет такие достоинства: поддержка визуальных компонентов, инкапсулирующих не только вывод HTML, но и Javascript и даже AJAX. Кроме встроенных, существуют компоненты сторонних производителей; возможность наследовать свой компонент от чужого и улучшать его, адаптируя под свои потребности; встроенные визуальные компоненты для редактирования и отображения данных, навигации, авторизации и т.д; встроенные средства кэширования, мониторинга, позволяющие повысить производительность и масштабируемость приложений; шаблоны дизайна, пользовательские элементы управления, модель codebehind – позволяют не только отделить программный код от представления, но и использовать многократно один и тот же код; встроенная авторизация и аутентификация; усовершенствованная архитектура доступа к данным ADO.NET и др [8].

ADO.NET относится к промежуточному программному обеспечению, которое внутри каждого приложения ответственно за доступ к базе данных и основано на .NET. Следовательно, использование ADO.NET играет решающую роль в создании эффективных, высокопроизводительных приложений для базы данных. Технология ADO.NET – это технология доступа к данным, включающая набор высокоуровневых интерфейсов, которые позволяют разработчикам обращаться к данным на любом .NET-совместимом языке программирования [9].

Для создания web-приложения был выбран современный подход – использование технологий ASP.NET 2.0. Для решения поставленной задачи была разработана, описанная далее, гибкая трехуровневая система на основе технологии ASP.NET.

2.3 Среда разработки Visual Studio 2008

При разработке программного обеспечения корпорация Microsoft уделяет большое внимание тому, чтобы его могли в полной мере использовать разные категории пользователей. Расширение линейки продуктов призвано ответить на уникальные запросы всех заинтересованных групп – любителей и студентов, профессиональных разработчиков и IT-профессионалов, архитекторов IT-систем предприятий. Продукты Visual Studio 2008 и .NET Framework 2.0 позволяют разработчикам и командам разработчиков создавать высокопроизводительные, защищенные и надежные решения, используя готовую платформу разработки. Возможности Visual Studio 2008 включают в себя такие наборы инструментов, как Visual Studio Team System – расширяемую платформу разработки приложений на протяжении всего жизненного цикла приложения, которая позволяет предоставлять сервисно-ориентированные решения для обеспечения совместной работы между командами разработчиков [10].

В качестве среды разработки был выбран мощнейший инструмент – Visual Studio 2008, оболочка .NetFramework 2.0.

Разработчики часто сталкиваются с многочисленными проблемами по обработке информации; например, с потребностью в более быстрых и основанных на управлении данными решениях, с потребностью увеличить производительность и мобильность штата разработчиков, а также с настойчивыми требованиями уменьшить общий бюджет отдела информационных технологий, увеличивая инфраструктуру в соответствии со строго возрастающими требованиями. Microsoft SQL Server 2008 призван помочь справиться с этими проблемами. MS SQL Server 2008 представляет собой интегрированное решение по управлению и анализу данных, которое помогает:

* строить, развертывать и управлять промышленными приложениями, которые являются более безопасными, масштабируемыми и надежными.
* увеличивать продуктивность информационных технологий, уменьшая сложность построения, развертывания и управления приложениями.
* разделять данные между платформами, приложениями и устройствами для облегчения соединения внутренних и внешних систем.
* контролировать стоимость, не жертвуя качеством выполнения, доступностью, масштабируемостью и безопасностью.

Исходя с выше перечисленных достоинств MS SQL Server 2008, в качестве сервера базы данных был выбран именно этот продукт.

Таким образом, выдвигаются такие требования к программному обеспечению для сервера:

* Microsoft Windows Server 2003 SP2 или Windows XP SP2;
* IIS 6.0 или выше;
* SQL Server 2008;
* Microsoft .NET Framework 2.0.

3 РАЗРАБОТКА СТРУКТУРЫ СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ПОИСКА ОТДЕЛЕНИЙ И ТЕРМИНАЛОВ БАНКОВ

3.1 Проектирование программной системы

Современный мир, в котором движется все очень быстро, потеря времени может стоить очень дорого. Люди дорожат своим временем, средствами, учитывают тайм менеджмент и не хотят их тратить на поиски впустую.

Итак, перечислим сервисы которые будет предоставлять система:

* Поиск банкоматов с отображением их на карте с учётом вашего текущего местоположения. Ваше текущее местоположение с помощью HTML5 Geolocation. Если Ваш браузер этого не поддерживает или Ваше местоположение не удалось точно определить, достаточно кликнуть на карте, что бы указать, где Вы находитесь и начать поиск;
* Поиск магазинов, где Вы можете расплатиться с помощью банковских карт;
* Поиск кафе, баров, ресторанов, где вы можете провести время, расплатившись без использования наличных средств;
* Информация и статьи о банковских картах, их особенностях и способах оплаты.

После установления системы в первых пунктах поисковиков и наплыва большого количества пользователей, в систему будет внедрен одни из видов монетизации – мультимидийная (баннеры, контекстная реклама).

Для разграничения прав в системе было выделено классы пользователей представленные в таблице 1.1.

Таблица 1.1 – Классы пользователей

|  |  |
| --- | --- |
| Класс | Описание |
| Администратор | * Бан юзера (забанить, разбанить)
* Удалять пользователей, изменять данные
* Создавать резервные копии БД
* Редактировать информационные блоки
* Восстанавливать БД (при необходимости)
* Создавать опрос, голосование
* Добавлять, удалять баннеры
 |
| Пользователь | - Регистрироваться- Просматривать информацию (новости, данные)- Голосовать, участвовать в опросах- Авторизоваться- Воспользоваться системой поиска |
| Гость | - Регистрироваться - Просматривать информацию (новости, данные) |

3.2 Определение функциональной модели системы

На рисунке 3.1 представлена диаграмма вариантов использования разработанной системы. Модель вариантов использования представляет собой модель того, как разные классы пользователи взаимодействуют с системой для решения своих проблем или задач [11]. Модель вариантов использования описывает цели пользователей, взаимодействие между пользователями и системой и требуемое поведение системы для удовлетворения этих целей.

Рисунок 1.1 – Use Case диаграмма системы

Для каждого класса пользователей разработана Activity диаграмма отображающая логику и последовательность переходов от одной деятельности к другой. Результат деятельности может привести к изменению состояния системы или возвращению некоторого значения. Activity диаграмма показана для каждого класса пользователей и представлена на рисунках 3.2a, 3.2b, 3.2c.

Рисунок 2.2a – Activity диаграмма (Администратор)

На рисунке 2.2a показана активность администратора, в процессе работы с программной системой.

Рисунок 2.2b – Activity диаграмма (Гость)

Рисунок 2.2c – Activity диаграмма (Пользователь)

Все перечисленные диограммы предназначены для визуального отображения основной архитектурыограммы. Модуль результатов обучения и статистики предназначен так же и для студентов.

Функциональная модель (IDEF) системы показывает, какие этапы жизненного цикла процесса поиска происходят в системе и как они зависят друг от друга [12]. Входными данными в систему являются: информация о лоакии банков, локация отделений банков. Выходные данные: местонахождение зап, обновленнашиваеммых данных относительно местанахождения пользователя.

4 ЗАЩИТА ИНФОРМАЦИИ В СИСТЕМЕ

Web-приложения – наиболее распространенные программные сервисы, доступные через интернет, поэтому они являются лакомым куском для всяких злоумышленников, желающих получить доступ к сети и похитить ценную информацию, испортить данные или как-то иначе скомпрометировать систему. Обеспечение безопасности Web-приложения – весьма серьезная задача, которой нужно уделять должное внимание на всех этапах – при проектировании, разработке, развертывании и эксплуатации.

Проектирование описываемой системы, было основано на нескольких ключевых принципах: все данные, вводимые в систему, считаются злонамеренными, область системы, доступная для атак, должна быть минимизирована. Эти универсальные принципы, начиная с этапа проектирования, гарантируют, что приложение будет защищено настолько, насколько это возможно.

Данное web-приложение было разработано и развернуто с помощью средств защиты, предоставляемых ASP.NET 2.0 и IIS 6.0.

IIS и ASP.NET служат основой для построения механизмов управления доступом, а ASP.NET 2.0 расширяет их возможности, предоставляя готовые прикладные блоки, которые можно использовать для быстрого развертывания таких механизмов.

На выходе механизм IIS-аутентификации всегда дает идентификацию для Windows, представляющую пользователя, от которого исходил запрос. В IIS доступны следующие встроенные типы аутентификации: Anonymous (анонимная), Integrated Windows (средствами Windows), Basic (базовая), Digest (по хэшу), Certificate Mapping (с сертификатами) и Microsoft Passport. Аутентификация подразумевает проверку подлинности клиентов по их учетным записям в домене. ASP.NET поддерживает три типа аутентификации: Windows (средствами Windows), Forms (на основе web-форм) и Passport.

В данной системе используется Forms-аутентификация, которая применяет схему аутентификации прикладного уровня, основанную на билетах. Такая схема предназначена для ASP.NET-приложений, которые не связывают со своими пользователями учетные записи Windows. Аутентификацию Forms применяется совместно с анонимной IIS-аутентификацией. Схема аутентификации системы для приложения задается в конфигурационном файле. На рисунке рисунок 4.1 представлена схема аутентификации данного приложения.

Рисунок 3.1 – Схема аутентификации

При аутентификации на основе форм, когда доступ к ресурсу, требующему аутентификации, отклоняется, пользователь переадресовывается на заранее заданный URL, указывающий на страницу входа. Это специальная ASP.NET-страница, в которой пользователь может ввести удостоверения (логин, пароль). Затем страница входа сверяет удостоверения с базой данных и с помощью класса System.Web.FormsAuthentication формирует зашифрованный билет аутентификации для клиента. Этим билетом может быть cookie или специальный большой двоичный объект, добавляемый к URL. При каждом последующем запросе клиент передает билет, и механизм аутентификации на основе форм автоматически обрабатывает его, генерируя идентификацию пользователя для данного запроса.

ASP.NET 2.0 предоставляет два компонента для работы с аутентификацией: сервис Membership и семейство элементов управления Login.

Сервис Membership, доступный через класс System.Web.Security.Membership, позволяет определить различные виды членства на сайте. Информацию о членах можно хранить в различных местах – в базах данных, текстовых файлах или даже в учетных записях Windows. Данная ситема использует в качестве хранилища базу данных. Конфигурировать членство можно индивидуально для каждого пользователя или на основе ролей с помощью сервиса Role Manager. Роли облегчают конфигурирование, так как можно создавать роли и потом добавлять пользователей к готовым ролям. На рисунке 4.2 представлена схема провайдера данной системы.

После определения идентификация пользователя, выдавшего запрос, принимаются решения, связанные с управлением доступом. В ASP.NET два встроенных механизма авторизации, управляющих доступом на уровне URL: File и URL.

В данной работе используется URL авторизация. При авторизации URL действуют правила управления доступом, задаваемые в конфигурационном файле, которые разрешают или запрещают доступ в зависимости от имени пользователя или ролей.

Рисунок 4.2 – Схема Membership провайдера

Правила управления доступом для авторизации URL задаются на уровне URL в конфигурационном разделе (рисунко 4.3).

Рисунок 4.3 – Схема авторизации

Важным моментом в обеспечении защиты web-приложения является отсутствие файлов, которые не планируется передавать клиентам. В описываемой системе такие файлы были удалены из физической структуры каталогов, начиная с самого верхнего каталога, помеченного в конфигурации IIS как Web-приложение (виртуальный каталог). Так как если файл не принадлежит Web-пространству имен, он не будет доступен при запросах к этому пространству. Так же отслеживаются файлы загружаемые в систему.

Следующая настройка (рисунок 4.4) файла Web.config запрещает загружать в систему клиентам данного приложения файлы с размером больше, чем указано.

Рисунок 4.4 – Элемент файла Web.config

ВЫВОДЫ

Анализ представленных на рынке систем автоматизированного поиска позволяет отметить общую тенденцию в подходе к разработке таких систем: в рамках разрабатываемых средств они стремятся охватить все элементы традиционного поиска.

Основными недостатками существующих систем являются недостаточная их локализация. То есть они являются общими системами и не учитывают потребности конкретных процессов.

Эти проблемы может решить разработанная система, которая автоматизирует процесс поиска банкоматов и отделений банков с целью эффективного взаимодействия между пользователями банковских услуг и банками. Внедрение системы в автоматизированный поиск банкоматов отделений позволит:

* Поиск банкоматов с отображением их на карте с учётом вашего текущего местоположения. Ваше текущее местоположение с помощью HTML5 Geolocation. Если Ваш браузер этого не поддерживает или Ваше местоположение не удалось точно определить, достаточно кликнуть на карте, что бы указать, где Вы находитесь и начать поиск;
* Поиск магазинов, где Вы можете расплатиться с помощью банковских карт;
* Поиск кафе, баров, ресторанов, где вы можете провести время, расплатившись без использования наличных средств;
* Информация и статьи о банковских картах, их особенностях и способах оплаты.

Возможность удобного поиска с затратой минимальных усилий, быстрое нахождение банкоматов и отделений банков в любом уголке Украины,– все это делает систему автоматизированного поиска банкоматов не опциональным дополнением, а продиктованной современными условиями необходимостью.

ПЕРЕЧЕНЬ ССЫЛОК

* + - * 1. Zoho – Online Office, Word Processor, Spreadsheet, Presentation, CRM Электронный ресурс – Режим доступа: http://zoho.com/ – 16.02.2008 г. – Загл. с экрана.
				2. Moodle – A Free, Open Source Course Management System for Online Learning Электронный ресурс – Режим доступа: http://moodle.org/ – 10.03.2008 г. – Загл. с экрана.
				3. WebCT Communities Электронный ресурс – Режим доступа: http://www.webct.com/webct/ – 10.02.2008 г. – Загл. с экрана.
				4. Группы Google Электронный ресурс – Режим доступа: http://groups.google.com.ua/ – 26.04.2008 г. – Загл. с экрана.
				5. IBM Lotus Software Электронный ресурс – Режим доступа: http://www-306.ibm.com/software/lotus/ – 26.02.2008 г. – Загл. с экрана.
				6. Рихтер Дж. Программирование на платформе Microsoft .NET Framework. Мастер–класс Дж. Рихтер: пер. с англ. – СПб.: Питер, 2005. – 512 с.
				7. ASP.NET (.NET Framework) Электронный ресурс – Режим доступа: http://msdn.microsoft.com/en-us/library/bb400852.aspx – 10.05.2008 г. – Загл. с экрана.
				8. Троелсен Э. Язык программирования C# 2005 и платформа .NET 2.0. HTML Э. Троелсен: – М.; ООО «И.Д. Вильямс», 2007. – 1168 с.
				9. Сеппа Д. Программирование на Microsoft ADO.NET 2.0. Мастер класс Д. Сеппа: пер. с англ. – М.: Издательский дом «Русская редакция»; СПб.: Питер, 2007. – 784 с.
				10. Мак-Дональд М. Microsoft ASP.NET с примерами на C# 2005 для профессионалов М. Мак-Дональд: пер. с англ. – М.; ООО «И.Д. Вильямс», 2006. – 1408 с.
				11. Буч Г., Якобсон А. UML Г.Буч. - СПб.:Питер, 2005.-736 с.
				12. Integration Definition Metodology – Режим доступа: http://www.idef.ru/ – 12.12.2007 г. – Загл. с экрана.
				13. Дейт К. Введение в системы базы данных, 8–е издание К. Дейт: пер. с англ. – М.: Издательский дом «Вильямс», 2005. – 1328 с.
				14. Хэндерсон К. Профессиональное руководство по SQL Server: хранимые процедуры, XML, HTML К. Хэндерсон. – СПб.:Питер, 2005. – 620 с.

Приложение А

Программный код приложения

/// <summary>

/// Represent course information.

/// </summary>

public class Course

{

...

}

/// <summary>

/// The class CourseManagement represents methods that makes possible do

/// actions with course information.

/// </summary>

public class CourseManagement

{

/// <summary>

/// Gets all courses that are available for study.

/// </summary>

/// <returns>The collection of courses.</returns>

[DataObjectMethod(DataObjectMethodType.Select)]

public IList<Course> SelectCourses()

{

List<Course> courseList = new List<Course>();

SqlConnection connection = new SqlConnection(\_connectionString);

using (SqlCommand command = new SqlCommand(Procedure.SelectAllCourse, connection))

{

command.CommandType = CommandType.StoredProcedure;

connection.Open();

SqlDataReader sqlDataReader = command.ExecuteReader();

while (sqlDataReader.Read())

{

Course course = new Course(sqlDataReader.GetGuid(0), HttpUtility.HtmlDecode(sqlDataReader.GetString(1)), HttpUtility.HtmlDecode(sqlDataReader.GetString(7)), sqlDataReader.GetGuid(8), sqlDataReader.GetDateTime(3), sqlDataReader.GetDateTime(4), sqlDataReader.GetBoolean(5), sqlDataReader.GetInt32(6), sqlDataReader.GetInt32(11), HttpUtility.HtmlDecode(sqlDataReader.GetString(2)), HttpUtility.HtmlDecode(sqlDataReader.GetString(9)), HttpUtility.HtmlDecode(sqlDataReader.GetString(10)));

courseList.Add(course);

}

connection.Close();

}

return courseList;

}

/// <summary>

/// Gets subjects of the specified course.

/// </summary>

/// <param name="courseId">Course identifier</param>

/// <returns>The collection of courses.</returns>

public List<string> GetCourseSubjects(Guid courseId)

{

List<string> subjects = new List<string>();

using (SqlConnection connection = new SqlConnection(\_connectionString))

{

using (SqlCommand cmd = new SqlCommand(Procedure.SelectCurriculumCourseContent, connection))

{

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.Add(new SqlParameter(ParameterName.CourseId, SqlDbType.UniqueIdentifier));

cmd.Parameters[ParameterName.CourseId].Value = courseId;

connection.Open();

SqlDataReader reader = cmd.ExecuteReader();

while (reader.Read())

{

subjects.Add(reader.GetString(2));

}

connection.Close();

}

}

return subjects;

}

/// <summary>

/// Insert course to database.

/// </summary>

/// <param name="course">Course data.</param>

/// <returns>Inserted course identifier.</returns>

[DataObjectMethod(DataObjectMethodType.Insert)]

public Guid InsertCourse(Course course)

{

SqlConnection connection = new SqlConnection(\_connectionString);

Guid courseId = new Guid();

using (SqlCommand command = new SqlCommand(Procedure.InsertCourse, connection))

{

command.CommandType = CommandType.StoredProcedure;

command.Parameters.Add(TitleParameter, SqlDbType.NVarChar).Value = course.Title;

command.Parameters.Add(CourseKeyParameter, SqlDbType.NVarChar).Value = course.CourseKey;

command.Parameters.Add(DescriptionParameter, SqlDbType.NVarChar).Value = course.Description;

command.Parameters.Add(StartDateParameter, SqlDbType.DateTime).Value = course.StartDate;

command.Parameters.AddWithValue(EndDateParameter, SqlDbType.DateTime).Value = course.EndDate;

command.Parameters.Add(IsMasterParameter, SqlDbType.Bit).Value = course.IsMaster;

command.Parameters.Add(AmountCreditsParameter, SqlDbType.Float).Value = course.AmountCredits;

command.Parameters.Add(MainTeacherIdParameter, SqlDbType.UniqueIdentifier).Value = course.MainTeacherId;

command.Parameters.Add(ReadingParameter, SqlDbType.NVarChar).Value = course.Reading;

command.Parameters.Add(AssessmentMethodParameter, SqlDbType.NVarChar).Value = course.AssessmentMethod;

command.Parameters.Add(CountStudentInTeamParameter, SqlDbType.Int).Value = course.CountStudentInTeam;

connection.Open();

courseId = new Guid(command.ExecuteScalar().ToString());

connection.Close();

}

return courseId;

}