

УДК 004.65

АВТОМАТИЗАЦИЯ ДЕЯТЕЛЬНОСТИ ОТДЕЛА БАНКА ПО ВЫДАЧЕ ИПОТЕЧНЫХ КРЕДИТОВ

Аббакумов Андрей Александрович, Егунова Алла Ивановна
ФГБОУ ВПО «Мордовский государственный университет им. Н. П. Огарева»,
Российская Федерация, г. Саранск,
E-mail: abbakumov_aa@mail.ru, тел. +7(927) 6427452,
430005, г.Саранск, ул. Богдана Хмельницкого, д. 39

Аннотация. В статье рассматриваются существующие автоматизированные системы выдачи ипотечных кредитов, а также процесс создания собственной программной системы, учитывающей достоинства и недостатки действующих АИС в области ипотеки.

Ключевые слова: ипотека, заявка, кредит, фреймворк, Ruby on Rails.

Тема ипотечного кредитования в настоящее время очень актуальна. Количество потенциальных заемщиков растет изо дня в день. Но существует ряд проблем в этой сфере, решение которых заключается в автоматизации процесса выдачи кредитов. К таким проблемам относятся:

- проблема документооборота. Клиент вынужден тратить драгоценное время, бегая с документами, простаивая в очередях. Сотрудники банка, работники кредитного комитета и страховой компании обмениваются документами в бумажной форме;
- недостаточно оперативный обмен информацией между банком, клиентом и страховой компанией;
- сложность получения вышестоящими руководителями банка информации о выданных кредитах;
- долгая процедура андеррайтинга (сопоставление рисков выдаваемым кредитам) и реструктуризации (изменение условий кредитования);
- сложность контроля за соблюдением своевременных выплат по кредиту.

Очевидно, что существует потребность в информационной системе, способной решить эти проблемы.

Существует несколько действующих автоматизированных систем в области ипотеки. В частности, Федеральное агентство по ипотечному жилищному кредитованию с 2005 года использует систему, разработанную компаниями EPAM и VDI. Они построили свою систему на базе собственного ПО с использованием технологий ActiveX, VB Script, а в качестве СУБД использовали MSSQL 2000. Национальная ипотечная компания закупила АИС фирмы ОАО "ЦМД-софт", построенную на базе Microsoft Dynamics CRM. Крупные российские банки используют свои автоматизированные системы, в основе которых, как правило, лежит промышленная СУБД. Но все вышеперечисленные автоматизированные системы имеют ряд недостатков, среди которых можно выделить высокую себестоимость, и ориентацию систем исключительно на сотрудников банка, то есть отсутствие клиентского интерфейса.

Таким образом, было принято решение построить разрабатываемую систему как веб-приложение, для предоставления доступа к системе не только сотрудникам банка, но и клиентам. Веб-приложение — клиент-серверное приложение, в котором клиентом выступает браузер, а сервером — веб-сервер. Логика веб-приложения распределена между сервером и клиентом, хранение данных осуществляется, преимущественно, на сервере, обмен информацией происходит по сети. Одним из преимуществ такого подхода является тот факт, что клиенты не зависят от конкретной операционной системы пользователя, поэтому веб-приложения являются межплатформенными сервисами. Вместо того чтобы писать

различные версии для различных операционных систем, приложение создается один раз для произвольно выбранной платформы и на ней разворачивается. Для создания веб-приложений на стороне сервера используются разнообразные технологии и языки программирования.

Таким образом, все программное обеспечение, необходимое для создания системы в виде веб-приложения, можно разделить на следующие компоненты:

- Среда программирования
- СУБД
- Веб-сервер

Наиболее оптимальным выбором среды программирования является Фреймворк Ruby on Rails. Это – программный каркас, написанный на языке программирования Ruby. Ruby on Rails предоставляет архитектурный образец MVC (Model-View-Controller — Модель-Представление-Контроллер). Такая архитектура подразумевает, что модель данных приложения, пользовательский интерфейс и управляющая логика разделены на три отдельных компонента, так, что модификация одного из компонентов оказывает минимальное воздействие на другие компоненты. Модель отвечает за поддержку состояния приложения. Иногда это состояние является кратковременным, продолжающимся только на время нескольких взаимодействий с пользователем. А иногда состояние является постоянным и сохраняется вне приложения, чаще всего в базе данных. В модели прописаны все бизнес-правила, применяемые к данным. Помещая реализацию бизнес-правил в модель, мы гарантируем, что ничто иное в приложении не может их испортить. Модель работает и сторожем, и хранителем данных. Представление отвечает за формирование пользовательского интерфейса, который обычно основан на данных модели. Хотя представление может преподносить пользователю различные способы ввода данных, оно никогда не занимается их непосредственной обработкой. Работа представления завершается, как только данные будут отображены. Допускается существование множества представлений, имеющих доступ к одним и тем же данным модели, часто с различными целями. Контроллеры организуют работу приложения. Они воспринимают события внешнего мира, взаимодействуют с моделью и отображают соответствующее представление для пользователя.

Таким образом, модели, представления и контроллеры разрабатываются как различающиеся по функциональным возможностям фрагменты, а среда Rails связывает их вместе при выполнении программы. Одно из положительных свойств Rails заключается в том, что связывание процессов основано на использовании разумных установок по умолчанию, поэтому для того, чтобы все заработало, обычно не приходится создавать какие-либо внешние конфигурационные метаданные.

Важной составляющей разрабатываемой информационной системы является СУБД. Rails способна взаимодействовать со следующими серверами БД: MySQL, Firebird, PostgreSQL, IBM DB2, Oracle и Microsoft SQL Server. Выбор пал на реляционную СУБД MySQL, поддерживающую SQL (структурированный язык запросов) и может применяться в качестве SQL-сервера. Связь Rails с СУБД имеет свои особенности, в виду использования объектно-реляционного отображения. Для связи кода приложения с базой данных понадобятся два уровня программного обеспечения. Первый уровень — это драйвер базы данных, представляющий собой Ruby-библиотеку, подключающую низкоуровневый API базы данных к высокоуровневому миру программирования на языке Ruby. Второй уровень программного обеспечения представляет собой Rail-адаптер базы данных. Ему отводится место между библиотекой Ruby и приложением. Каждая библиотека базы данных будет иметь свой собственный API, зависящий от конкретной базы. Rails-адаптеры баз данных скрывают эти различия, поэтому Rails-приложения не нуждаются в сведениях о том, с какой именно базой данных они работают. Возможно использование одного и того же программного кода для работы с разными базами данных. В случае смены базы данных, нужно только поменять драйвер базы данных.

Ruby on Rails может работать с любым веб-сервером, поддерживающим FastCGI. В язык Ruby уже встроены 2 веб-сервера: Mongrel и Webrick. Для построения системы выбран веб-сервер Mongrel.

На основании предложенной архитектуры, была построена программная система, автоматизирующая деятельность отдела банка по выдаче ипотечных кредитов, предоставляющая интерфейсы доступа как для сотрудников банка, так и для клиентов. В системе реализовано разграничение прав доступа между всеми пользователями. Клиент банка имеет возможность подать заявку на предоставление ему кредита и на изменение условий кредитования в случае возникновения проблем с погашением кредита. Система предоставляет оператору возможность обработки заявок клиентов и передачи отработанных заявок в кредитный комитет, способна анализировать заявки и генерировать рекомендации по поводу целесообразности выдачи кредита тому или иному клиенту. В основе анализа лежит кредитный скоринг – это система оценки кредитоспособности (кредитных рисков) лица, основанная на численных статистических методах. Он заключается в присвоении баллов по заполнению анкеты, разработанной оценщиками кредитных рисков – андеррайтерами. По результатам набранных баллов инспектор кредитного комитета принимает решение об одобрении или отказе в выдаче кредита. Считается, что существует корреляция между некоторыми социальными данными (наличие детей, отношение к браку, наличие высшего образования) и добросовестностью заемщика. Кредитный скоринг является упрощенной системой анализа заемщика, что позволяет снизить требования к квалификации кредитного инспектора, занятого рассмотрением заявок на кредит, и увеличить скорость их рассмотрения. Также система фиксирует просрочки по выплатам и генерирует отчеты для вышестоящих руководителей банка (общая сумма выданных ипотечных кредитов, выплаты по кредитам в заданный промежуток времени, выплаты по кредитам для заданного должника, выплаты по кредитам для изменивших условия кредитного договора, просрочки по выплате кредитов и др.).

В итоге проведенной работы, была разработана автоматизированная информационная система, удовлетворяющая выдвинутым к ней требованиям, которая решает проблемы документооборота, обмена информацией между клиентами и сотрудниками банка, упрощает процедуру андеррайтинга за счет применения кредитного скоринга, фиксирует просрочки по выплате и генерирует отчеты по выданным кредитам. Результатом ее применения является повышение эффективности процесса выдачи ипотечных кредитов. Помимо этого, система является универсальной, ее можно интегрировать с другими банковскими системами и использовать для любого другого вида кредитования.

Литература

- 1 Томас Д., Хэнссон Д. Х. Гибкая разработка веб-приложений в среде Rails. – СПб.: Питер, 2008. – 716 с.
- 2 Фернандес О. Путь Rails. Подробное руководство по созданию приложений в среде Ruby on Rails. – Пер. с англ. – СПб: Символ-Плюс, 2009. – 768 с.
- 3 Фултон Х. Программирование на языке Ruby. – М.: ДМК Пресс, 2007. – 688 с.
- 4 Аббакумов А.А., Егунова А.И., Панкратов М.В. Использование Ruby on Rails для автоматизации учета ипотечных кредитов // Электроника и информационные технологии. – 2011 выпуск 1 (10). – 2011. - http://fetmag.mrsu.ru/2011-1/pdf/Use_Ruby_On_Rails.pdf. 0421100067/0017.

AUTOMATION OF THE ACCOUNT OF HYPOTHECARY CREDITS

Abbakumov A.A., Egunova A.I.

N. P. Ogarev's Mordovian State University

Russian Federation, Saransk city,

E-mail: abbakumov_aa@mail.ru, tel.+7(927) 6427452,

39, Bogdana Hmel'nitskogo str, room 502, 430005, Saransk, Mordovia, Russia

Annotation. This article considers the existing automated mortgage providing systems, as well as chooses the programming environment for creating as particular programming system, which takes into account both advantages and disadvantages of the present AIS in the hypothecary area.

Keywords: mortgage; application; credit; framework; Ruby on Rails.

Аббакумов Андрей Александрович

Россия, РМ, г. Саранск

Мордовский государственный университет им. Н. П. Огарёва

E-mail: abbakumov_aa@mail.ru.

Тел. +7(927) 6427452

Закончил Мордовский государственный университет им. Н. П. Огарёва в 2001 г. по специальности автоматизированные системы обработки информации и управления. В 2006 г. защитил кандидатскую диссертацию и работает на должности доцента кафедры автоматизированные системы обработки информации и управления в Мордовском государственном университете им. Н. П. Огарёва.

Егунова Алла Ивановна

Россия, РМ, г. Саранск

Мордовский государственный университет им. Н. П. Огарёва

E-mail: vystasalla@gmail.com.

Тел. +7(834)2290602

Закончила филиал «Восход», Московского авиационного института им. С. Орджоникидзе в 1991 г. по специальности «Электронные вычислительные системы, комплексы и сети». В 1999 году защитила кандидатскую диссертацию и работает на должности доцента кафедры автоматизированные системы обработки информации и управления в Мордовском государственном университете им. Н. П. Огарёва.