

## ОБ АСПЕКТАХ РАЗРАБОТКИ ПРОГРАММНОГО МОДУЛЯ ИНТЕГРАЦИИ СТРАХОВАНИЯ И ОБРАБОТКИ ДАННЫХ ДЛЯ СИСТЕМЫ ОНЛАЙН КРЕДИТОВАНИЯ

*Прокопчук В.И.*

*Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь*

*Сычева Ю.С. – кандидат педагогических наук*

В работе рассматриваются основные аспекты разработки программного модуля интеграции страхования и обработки данных для системы онлайн кредитования. Излагается концепция построения архитектуры модуля. Выявлены основные требования и определены инструменты, используемые для разработки.

Кредитование населения банками имеет большое социальное значение, так как способствует удовлетворению жизненно важных потребностей населения в различных товарах и услугах; выполняет экономические задачи, позволяя рационально использовать временно свободные денежные средства вкладчиков. За счет кредитования банки получают большую часть прибыли.

Кредитные операции являются одним из наиболее динамично растущих направлений банковской деятельности, приносящих существенную часть их прибыли. В то же время кредитование физических лиц способствует повышению благосостояния населения и является крайне важным элементом национальной экономики, способствующим при устойчивом развитии процессам расширенного воспроизводства. Рыночные отношения способствуют развитию кредитных отношений, поскольку кредитование и стимулирование покупательной способности населения является одним из факторов ускорения экономической активности в жилищном строительстве, производстве, розничной торговле и сфере услуг.

Наличие непоплаченных долгов является всегда риском для улучшения финансового состояния граждан, несмотря на временные преимущества, предоставляемые кредитными организациями для удовлетворения потребностей населения. В условиях появления новых кредитных продуктов, снижаются требования к заемщику, все более очевидным становится риск невозврата заемных средств банку и необходимость его страхования.

Так как банк является одним из ключевых звеньев кредитования, именно он будет нести убытки в случае возникновения кредитных рисков.

Банки должны быть застрахованы от рисков. Перед тем как предоставить кредит, банк должен проверить финансовое состояние клиента. В этой ситуации приобретает задача качественной и всесторонней, но в тоже время оперативной, проверке заемщиков.

Конкурентные преимущества будут иметь высокотехнологичные банки, которые способны при относительно низких издержках обработать колоссальный объем документов и материалов, обеспечить оперативный и индивидуальный подход к клиентам.

Программный модуль интеграции страхования и обработки данных для системы онлайн кредитования предназначен для использования в автоматизированных системах кредитования.

Разработка модуля предполагает следующие этапы:

1. Обзор и анализ ресурсов схожей тематики, в рамках которого выявляются достоинства и недостатки существующих аналогов, определяются преимущества разрабатываемого модуля и основные этапы его проектирования.

2. Разработка архитектуры модуля. Серверная часть модуля представляет собой фасад для передачи информации между приложением и узлами страховой компании по протоколу SOAP.

При работе с удаленным интерфейсом, каждый его вызов обходится дорого. В результате нужно уменьшить количество вызовов, а это значит, что нужно передавать больше данных с каждым вызовом. Один из способов сделать это – использовать множество параметров. Тем не менее, это часто неудобно для программирования – на самом деле, это часто невозможно с такими языками, как Java, которые возвращают только одно значение.

Решение заключается в создании Data Transfer Object, который может содержать все данные для вызова. Основная причина использования Data Transfer Object заключается в объединении нескольких удаленных вызовов в один. Стоит отметить, что еще одним преимуществом является инкапсуляция механизма сериализации для передачи данных.

Для передачи сообщения, необходимо создать Data Transfer Object и передать его в фасад.

Фасад – это класс, который предоставляет простой интерфейс для сложной подсистемы и может обеспечить ограниченную функциональность по сравнению с работой с подсистемой напрямую. Однако он включает только те функции, которые действительно важны для клиентов. Фасад удобен, когда нужно интегрировать приложение со сложной библиотекой или удаленным интерфейсом, которые имеют десятки функций, но вам нужна часть функциональности.

Поскольку в SOAP сервисах информация передается в XML формате, необходимо преобразовать ответ от сервиса в удобный для работы формат. Для преобразования из одного формата в другой удобно использовать адаптер.

Адаптер – это специальный объект, который преобразует интерфейс одного объекта, чтобы другой объект мог его понять. Адаптер оборачивает один из объектов, чтобы скрыть сложность преобразования. Адаптер может не только преобразовывать данные в различные форматы, но и помогать объектам с разными интерфейсами взаимодействовать.

Внутри модуля Data Transfer Object будет преобразован в необходимый для передачи формат с помощью адаптера и передан на соответствующий узел. Ответ от узла будет преобразован в Data Transfer Object и возвращен из адаптера.

3. Создание макета пользовательского интерфейса. Под макетом интерфейса понимается визуальное представление интерфейса пользователя. На этом этапе мы распределяем все важные элементы управления на веб-странице и формируем их поведение на пользовательское взаимодействие.

4. Создание логической модели базы данных. С помощью логической модели будут получено графическое представление исследуемой предметной области. Логическая модель дает возможность выделить и графически представить сущности изучаемой предметной области, а также взаимоотношения между ними.

5. Проектирование физической модели базы данных. Итогом этапа является база данных, которая будет использоваться при разработке модуля.

6. Разработка и интеграция серверной части модуля. Завершив этот этап, у нас будут готовые обработчики пользовательских запросов и формирования ответов; обработчики для анализа, передачи и конвертирования пользовательских данных; обработчики по формированию страховых соглашений и документов в PDF формате.

7. Разработка и интеграция клиентской части модуля. Финальный этап основан на создании полноценного пользовательского интерфейса с помощью спроектированного макета интерфейса.

Предлагаемая система представляет из себя модуль для серверной части и клиентской части. Для использования клиентской части необходим только браузер. Таким образом обмен информацией между клиентом и сервером происходит по сети, что является плюсом в вопросе адаптации приложения, так как этот вид приложений является межплатформенным и не зависит от операционной системы пользователя.

При реализации было решено разделить функциональность. Это позволит разделить ответственность между компонентами системы.

В качестве серверной платформы разработки приложения была выбрана платформа ASP.NET и язык программирования C#. Для создания полноценной клиентской части приложения были использованы: язык гипертекстовой разметки HTML5, каскадные таблицы стилей CSS3, язык программирования TypeScript, расширяющий возможности JavaScript и фреймворк Vue.JS.

Выбранной СУБД для хранения информации является Microsoft SQL Server 2017. Для работы с базой данных был использован ORM-фреймворк Entity Framework на базе .NET. Концепцией данного фреймворка является понятие сущности, которая представляет собой набор данных, ассоциированных с определенным объектом.

Таким образом в результате проделанной работы был разработан модуль, который позволяет:

- интегрировать процесс страхования заемщиков в процесс оформления заявления на получение кредита;

- автоматизировать процесс обработки рисков и персональной информации;

- автоматизировать процесс изменения процентной политики.

Ключевой целью поэтапной разработки становится постепенное выявление не обнаруженных ранее ошибок и недочетов кода. В рамках такого процесса разработки программного обеспечения и заказчик, и исполнитель могут столкнуться с рядом достаточно узкого спектра ошибок, связанных с частичной рассогласованностью данных, а также срывов выполнения программных процедур в связи с применением многопользовательского доступа.

В рамках процесса поэтапной разработки образуется окончательная картина взаимодействия пользователя, а также определяется степень лояльности к разработанному интерфейсу.

Среди основных преимуществ модуля можно выделить:

- функциональные компоненты могут быть изменены, переписаны или заменены без изменений в остальных частях программного средства;

- отдельный модуль легче сопровождать и модифицировать;

- разработанный модуль можно интегрировать в любую систему, разработанную на языке платформы .NET.