**Введение**

В современном понимании история кредитования ведет свое начало со времен промышленной революции. Именно тогда на смену ростовщикам пришли полноценные коммерческие банки с развитой филиальной сетью. Такое перераспределение было связано с бурным развитием производства и торговли.

А потребительское кредитование явилось следствием появления такого понятия, как массовое производство. Т. е. фактически банки вышли на рынок розничных частных кредитов только после Второй мировой войны.

В наше время кредитование является очень важной частью в жизни большей части населения. Теперь, с появлением Личных кабинетов пользователя, которые сейчас имеются у каждого крупного банка, количество времени, необходимое для получения кредита сократилось, так как клиенту теперь не обязательно идти напрямую в банк, чтобы подать заявку на получение кредита. Воспользовавшись сайтом или приложение для мобильного устройства, пользователь может подать заявку в течении нескольких минут, но время ожидания ответа от банка все равно остается слишком большим, при этом, на текущий момент, клиенты не могут знать точную сумму, которую банк одобрит по заявке по потребительскому или жилищному кредиту или с каким лимитом могут выпустить кредитную карту. Так же на текущий момент Клиенты не всегда знают, что они могут повлиять на величину суммы кредита, и не знают, какие конкретно действия им необходимо выполнить для увеличения суммы кредита и для снижения их текущей долговой нагрузки. Не каждого клиента это устраивают такие условия, поэтому возникает необходимость в разработке такого инструмента, с помощью которого пользователь будет знать ту сумму, кредит на которую банк сможет одобрить с наибольшей вероятностью.

Чтобы решить представленные выше проблемы, с которыми сталкиваются клиенты, в ПАО «Сбербанк-Технологии» было принято решении о внедрении проекта «Кредитный потенциал».

Актуальность выпускной квалификационной работы заключается в том, что внедрение данной системы позволит любому клиенту «Сбербанк Онлайн», после предоставления необходимых данных и согласий на запрос информации во внешних источниках данных, онлайн получить ответы на вопросы, касающиеся возможности банка предоставить пользователю кредит, какую потенциальную сумму, на какой срок и на каких условиях банк сможет предоставить, а также запросить информацию о том, каким способом клиент сможет увеличить сумму кредита.

Целью выпускной квалификационной работы является разработка подсистемы конвертации результатов расчета кредитного потенциала пользователя для системы «Сбербанк-Онлайн».

Объектом исследования является ПАО «Сбербанк-Технологии». К предмету исследования относится проект «Кредитный потенциал».

Для достижения цели выпускной квалификационной работы, необходимо выполнить следующие задачи:

1. **Сравнительный анализ аналогов проектируемого объекта**

В сравнительном анализе проектируемого объекта выполняется обзор существующих на данный момент аналогов разрабатываемой системы.

Сервис Freeformatter позволяет выполнять конвертацию файлов самых распространенных форматов, таких как XML, JSON, HTML, SQL.

Преимущества:

1. возможность конвертации файлов большинства возможных форматов;
2. большое количество настраиваемых параметров непосредственно перед самой конвертацией;
3. возможность работы в браузере;
4. выбор исходного файла возможен как из списка локальных файлов, так и ссылкой на интернет-ресурс;
5. конвертер является бесплатным.

Недостатки:

1. данный сервис не имеет русской локализации;
2. при конвертации объемного файла, возможно нарушение исходной структуры документа;
3. отсутствует возможность предварительного просмотра итогового документа прямо в браузере.

Сервис Code Beautify является интернет-сервисом, позволяющим выполнять конвертацию XML в JSON.

Преимущества:

1. минималистичный и понятный интерфейс;
2. высокая точность конвертируемого файла;
3. возможность работы в браузере;
4. возможность предварительного просмотра итогового файла;
5. конвертер является бесплатным.

Недостатки:

1. данный сервис не имеет русской локализации;
2. отсутствие возможности выбора локального файла;
3. конвертация файла занимает сравнительно большое количество времени.

Описанные выше продукты не подходят для использования заказчиком, т.к. от разрабатываемой системы требуется синхронизация с базой данных, хранящей информацию о существующих продуктах и предложениях.

1. **Выбор технологии, среды и языка программирования**

Согласно техническому заданию, для разработки программы будет использоваться язык высокого уровня Java.

Разработка будет выполняться в среде программирования IntelliJ IDEA, т.к. данная среда является самой популярной и имеет весь функционал, необходимый для разработки проектируемого веб-приложения. (добавить описание функций)

## **2.1 Выбор жизненного цикла**

Жизненным циклом программного обеспечения является некий период времени, начинающийся с момента принятия решения о необходимости создания ПО и заканчивается в момент его полного изъятия из эксплуатации.

При рассмотрении различных вариантов жизненного цикла для подсистемы расчета кредитного потенциала, была выбрана модель Scrum.

Scrum – это современная методология, позволяющая управлять разработкой и модернизацией программного обеспечения. Она предназначена для небольших групп разработчиков, разбивающих разработку на этапы, которые выполняются в течении ограниченного времени (спринтов). В среднем, один спринт длится около двух недель. При окончании которого отслеживается прогресс и планируется следующий спринт, если он необходим.

Основные роли в Scrum:

Product Owner

Владелец продукта ориентирован на понимание требований бизнеса и рынка, а затем определяет приоритет той или иной работы, выполняемой командой разработчиков.

Scrum Master

Scrum Master является лидером команды разработчиков. Именно он является координатором всех выполняемых командой задач.

Он понимает работу, выполняемую командой, и может помочь в оптимизации выполняемых процессов. Будучи главным, он планирует и распределяет необходимые ресурсы (как людские, так и логистические) для выполнения одного или нескольких спринтов.

Scrum Master также разрешает возникающие проблемы и отвлекающие команду факторы.

Команда разработчиков

Команда разработчиков является небольшой командой, состоящей из 5-7 человек. Члены команды имеют разные наборы навыков которые пересекаются между собой, поэтому каждый член команды может предоставить другому помощь, а также все части проекта выполняются совместно. Все члены команды помогают друг другу, чтобы обеспечить успешное завершение спринта.

Команда разработчиков управляет планом для каждого спринта. Они прогнозируют то количество работы, которое, по их мнению, они могут завершить за время прохождения самого спринта, руководствуясь для этого сведениями о прошлых разработках.

Достоинства:

1. Возможность быстрого запуска проекта с наиболее приоритетными функциями и минимально возможным бюджетом;
2. Ежедневный контроль над ходом работ, и более гибкий контроль над бюджетом проекта;
3. Частые демонстрации проекта. Применение данной методологии предполагает регулярную демонстрацию разработок заказчику (заказать эффективный сайт можно, скажем, тут, - https://2atom.ru/ , - заодно сможете подсчитать стоимость в калькуляторе), что позволяет в будущем избежать полного провала работы команды и разочарований клиента;
4. Возможность вносить коррективы в техническое задание по ходу реализации проекта, что является несомненным преимуществом для заказчика.

Недостатки:

1. Сложности при заключении договоров. Scrum в принципе не подразумевает наличие фиксированного бюджета и фиксированного технического задания, что затрудняет юридическое оформление такого рода договоренностей;
2. Большое количество исключений. Специалисты в этой области считают данную методологию неприменимой для работы с государственными заказами, а также совершенно нерабочей при низкой квалификации команды, заниженных сроках работ или бюджете, некомпетентном менеджере проекта. В то время как другие методологии позволяют завершить проект при подобных условиях, хотя и на низком уровне;
3. Узкая специализация методов. Так, например, если использовать Scrum при разработке сайтов, этапы дизайна и контента уже будут выходить за рамки методологии и требовать совершенно иного подхода.

## **2.2 Выбор подхода к разработке**

Существует два основных подхода к разработке и проектированию информационных систем:

1. Структурный;
2. Объектно-ориентированный.

Для проектирования данной информационной системы, будет использоваться объектно-ориентированный подход.

## **Выбор инструментальных средств**

В соответствии с техническим заданием, для выполнения разработки будет использоваться язык программирования Java.

Java - язык программирования высокого уровня (ЯВУ), который изначально предназначался для программирования бытовых электронных устройств (например, телефонов).

Затем данный язык программирования начали применять для программирования браузеров, а также для написания полноценных приложений. Графические элементы этих приложений стали оформлять в виде компонентов, которые стали популярны среди распределенных систем и промежуточного программного обеспечения.

Вскоре Java стали использовать для программирования серверов, взаимодействующих с распределенными базами данных. После этого, некоторые системы управления базами данных (СУБД) внедрили описываемый язык программирования в свое ядро.

IntelliJ IDEA - молодое Java IDE (интегрированная среда разработки). Данная среда разработки, в отличии от большинства других, в большинстве своем ориентирована на ручное написание кода, в отличии от визуальных систем. В данной среде имеется встроенная поддержка рефакторинга и гибкие возможности по запуску и отладке разработанных приложений.