|  |  |
| --- | --- |
| Институт (факультет) | Информационных технологий |
| Направление подготовки (специальность) | 09.03.02 Информационные системы и технологии |
| Выпускающая кафедра | Математическое и программное обеспечение ЭВМ |

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

|  |  |
| --- | --- |
| Название работы | Разработка подсистемы кредитного потенциала пользователя |
| мобильного банка | |
|  | |
|  | |
|  | |

|  |  |
| --- | --- |
| Студента | Смирнова Дмитрия Сергеевича |
|  | Ф.И.О. |
|  | |

ДОПУСТИТЬ К ЗАЩИТЕ

|  |  |  |
| --- | --- | --- |
| Директор института (декан факультета) |  | (Е.В. Ершов) |
| Заведующий выпускающей кафедрой |  | (Е.В. Ершов) |
| Руководитель выпускной квалификационной работы |  | (О.В. Юдина) |
| Консультант по технико-экономическому обоснованию |  | (Л.Н. Виноградова) |
| Нормоконтролер |  | (Л.Н. Виноградова) |
| Выпускник |  | (Д.С. Смирнов) |

Череповец

2018 г.

Оглавление

[Введение 5](#_Toc516822177)

[Часть 1. Основная часть 7](#_Toc516822178)

[1.1 Сравнительный анализ аналогов проектируемого объекта 7](#_Toc516822179)

[1.2 Выбор технологии, среды и языка программирования 9](#_Toc516822180)

[1.2.1 Выбор жизненного цикла 9](#_Toc516822181)

[1.2.2 Выбор подхода к разработке 11](#_Toc516822182)

[1.2.3 Выбор инструментальных средств 12](#_Toc516822183)

[1.3 Анализ и уточнение требований к программному продукту 13](#_Toc516822184)

[1.4 Разработка спецификаций проектируемой системы 14](#_Toc516822185)

[1.4.1 Построение диаграмм вариантов использования 14](#_Toc516822186)

[1.4.1.1 Вариант использования Просмотр витрины кредитных продуктов 14](#_Toc516822187)

[1.4.1.2 Вариант использования Просмотр матрицы продукта 16](#_Toc516822188)

[1.4.1.3 Вариант использования Просмотр шкалы кредитного потенциала 17](#_Toc516822189)

[1.4.2 Построение контекстных диаграмм классов 18](#_Toc516822190)

[1.4.3 Построение диаграмм последовательностей системы 19](#_Toc516822191)

[1.4.4 Построение диаграмм деятельностей сценариев вариантов использования 31](#_Toc516822192)

[1.4.4.1 Построение диаграммы деятельностей для сценария варианта использования Просмотр витрины кредитных продуктов 32](#_Toc516822193)

[1.4.4.2 Построение диаграммы деятельностей для сценария варианта использования Просмотр Матрицы продукта 34](#_Toc516822194)

[1.4.4.3 Построение диаграммы деятельностей для сценария варианта использования Просмотр шкалы кредитного потенциала 35](#_Toc516822195)

[1.4.5 Построение структур данных и построение диаграмм отношений компонентов данных 37](#_Toc516822196)

[1.5 Проектирование системы 37](#_Toc516822197)

[1.5.1 Проектирование структуры системы и построение диаграммы пакетов 37](#_Toc516822198)

[1.5.2 Проектирование классов в пакетах 38](#_Toc516822199)

[1.5.2.1 Проектирование пакета Конвертация 38](#_Toc516822200)

[1.5.2.1.1 Проектирование классов пакета Конвертация 38](#_Toc516822201)

[1.5.2.1.2 Построение диаграмм последовательностей действий 39](#_Toc516822202)

[1.5.2.1.3 Построение диаграммы коопераций 42](#_Toc516822203)

[1.5.2.1.4 Построение уточненной диаграммы классов пакета Конвертация 43](#_Toc516822204)

[1.5.2.1.5 Построение детальной диаграммы классов пакета Конвертация 43](#_Toc516822205)

[1.5.2.2 Проектирование пакета База данных 47](#_Toc516822206)

[1.5.3 Построение диаграмм компонентов 53](#_Toc516822207)

[1.5.4 Построение диаграмм размещения 55](#_Toc516822208)

[1.5.5 Спецификация модулей 56](#_Toc516822209)

[1.6 Проектирование интерфейса пользователя 58](#_Toc516822210)

[1.6.1 Построение графа диалога 58](#_Toc516822211)

[1.6.2 Разработка форм ввода-вывода информации 63](#_Toc516822212)

[1.7 Выбор стратегии тестирования, разработка тестов, программа и методика испытаний 65](#_Toc516822213)

[1.7.1 Стандарты качества программного обеспечения 65](#_Toc516822214)

[1.7.2 Методы тестирования 66](#_Toc516822215)

[1.7.4 Объект испытаний 67](#_Toc516822216)

[1.7.5 Требования к информационному, аппаратно-программному обеспечению и документации 67](#_Toc516822217)

[1.7.5.1 Требования к функциональным характеристикам серверного приложения 68](#_Toc516822218)

[1.7.5.2 Требования к функциональным характеристикам базы данных 68](#_Toc516822219)

[1.7.5.3 Требования к надежности системы 68](#_Toc516822220)

[1.7.6 Состав и порядок испытаний 69](#_Toc516822221)

[1.7.7 Результаты проведения испытаний 69](#_Toc516822222)

[Часть 2. Технико-экономическое обоснование 70](#_Toc516822223)

[2.1 Расчёт трудоёмкости разработки 70](#_Toc516822224)

[2.2 Расчет себестоимости разработки 72](#_Toc516822225)

[2.3 Расчет цены разработки 76](#_Toc516822226)

[2.4 Расчёт экономической эффективности 77](#_Toc516822227)

[Заключение 80](#_Toc516822228)

[Список литературы 82](#_Toc516822229)

[Приложение 1. Техническое задание 84](#_Toc516822230)

# 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.

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 [11].

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 течение 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, как они могут повлиять на величину суммы кредита, и не знают, какие конкретно действия им необходимо совершить для увеличения суммы кредита или для снижения их текущей долговой нагрузки. Не каждого клиента 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 АО «Сбербанк-Технологии» было принято решении о реализации 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.

Объектом исследования является функциональная подсистема Единой Фронтальной Системы Сбербанка – «Кредиты физических лиц. Кредитный потенциал». Предметом исследования являются компоненты обработки и конвертации кредитного потенциала.

Целью выпускной квалификационно работы является разработка подсистемы конвертации для упрощения взаимодействия системы кредитного потенциала с существующим сервисом «Сбербанк Онлайн».

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

1. Анализ предметной области;
2. Моделирование подсистемы;
3. Разработка подсистемы;
4. Внедрение подсистемы.

Часть 1. Основная часть

## 1.1 Сравнительный анализ аналогов проектируемого объекта

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

Сервис Freeformatter (рис.1) позволяет выполнять конвертацию файлов самых распространенных форматов, таких как XML, JSON, HTML, SQL.

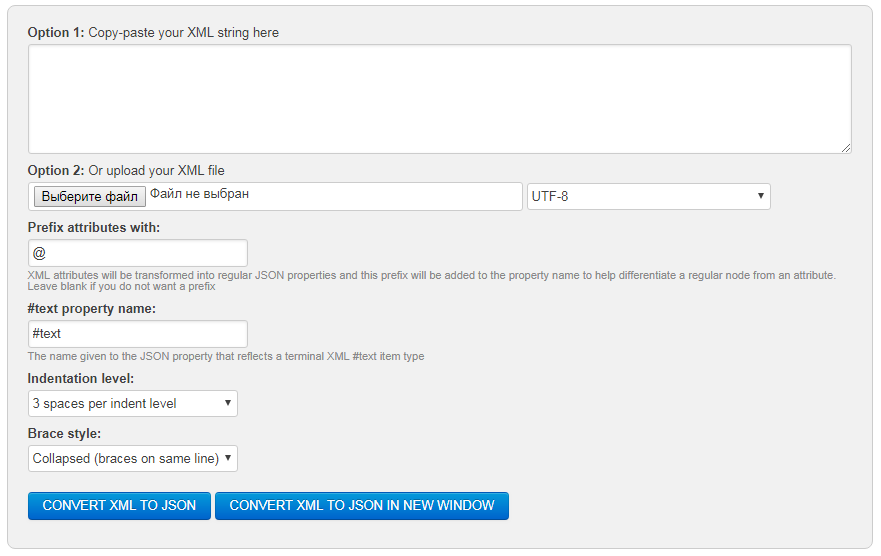


Рис. 1. Сервис Freeformatter

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 (рис. 2) является интернет-сервисом, позволяющим выполнять конвертацию XML в JSON.

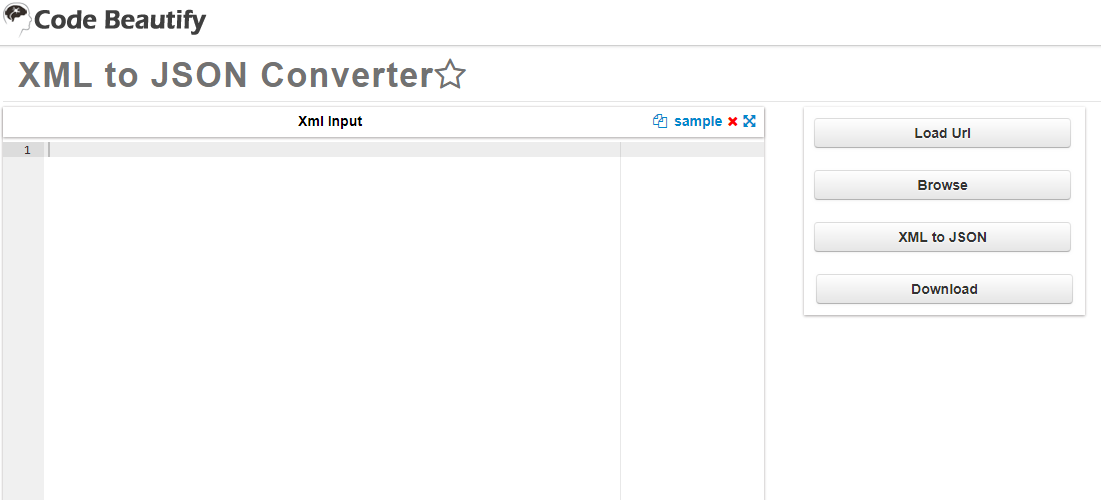


Рис. 2. Сервис Code Beautify

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.

## 1.2 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.

### 1.2.1 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:

1. Product Owner

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

2. 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.

3. Develop Team

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 [15].

Достоинства Scrum [9]:

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, 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.

Схема работы во время спринта при использовании Scrum изображена на рис. 3.

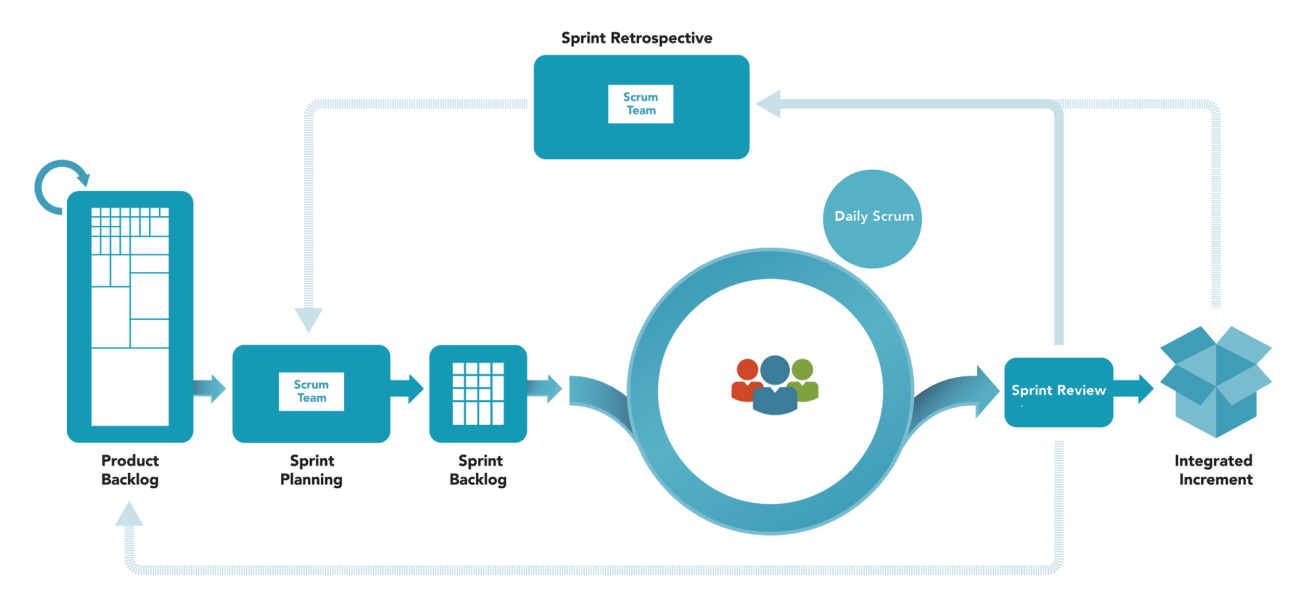


Рис. 3. Принцип работы со Scrum

### 1.2.2 Выбор подхода к разработке

Существует два основных подхода к разработке и проектированию информационных систем [4]:

1. Структурный;
2. Объектно-ориентированный.

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

Проектирование информационной системы с использованием объектно-ориентированного подхода выполняется архитектором системы и включает в себя [7]:

1. Идентификацию архитектурных решений и механизмов, необходимых для проектирования системы;
2. Анализ взаимодействий между классами, выявление подсистем и интерфейсов;
3. Формирование архитектурных уровней;
4. Проектирование структуры потоков управления;
5. Проектирование конфигурации системы.

### 1.2.3 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) [10].

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 [13, 14].

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 [5].

IntelliJ IDEA - Java IDE (интегрированная среда разработки). Данная среда разработки, в отличие 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.

## 1.3 Анализ и уточнение требований к программному продукту

Под Кредитным потенциалом понимается максимальный объем платежеспособности клиента, исходя из свободной части которого, Банк готов ему предварительно одобрить потенциальную сумму кредита/кредитного лимита по карте, основываясь на имеющейся информации, включая данные о доходах и расходах на жизнь

При получении запроса на конвертацию файла, системе предписано выполнять следующие действия:

1. Прочитать файл;
2. Выбрать из файла необходимые для вывода записи;
3. В соответствии с указанными в полученном файле идентификационными номерами, извлечь из базы данных необходимые значения и подставить их в документ;
4. Выполнить конвертацию файла;
5. Отправить файл обратно на сервер.

Разрабатываемая подсистема позволит упростить взаимодействие между существующими на данный момент сервисами мобильного банка «Сбербанк Онлайн» и системой кредитного потенциала пользователя.

По причине того, что все данные, получаемые подсистемой конвертации, будут закодированы, разработка также получит связь с базой данных, где будет храниться вся необходимая информация о кредитных продуктах и кредитном потенциале.

## 1.4 Разработка спецификаций проектируемой системы

### 1.4.1 Построение диаграмм вариантов использования

Диаграммы вариантов использования позволяют (Use-Case диаграммы) описывать возникающие между действующими лицами, участвующих в некотором процессе, зависимости и взаимоотношения [16].

Данные диаграммы описывают только внешнее устройство системы, т.е. не предназначены для отображения устройства самого проекта. Use-Case диаграммы необходимы для простоты взаимодействия между пользователями разрабатываемой системы, а также для определения необходимых ей характеристик [1].

Каждый вариант использования, изображенный на диаграмме, описывает группу действий в системе, приводящих к некоторому результату.

Варианты использования описывают взаимодействия пользователя с системой.

Вариант использования описывает одно или несколько действий приводящих к определенному результату.

Варианты использования описывают взаимодействия между пользователем (пользователями), и системой.

Действующее лицо является источником взаимодействия непосредственно с самой системой. В роли действующего лица могут выступать люди либо другие системы либо внешние события [6].

Действующие лица являются ролями, от которых зависит то, какими способами будет происходить взаимодействия с самой системой.

Диаграмма вариантов использования изображена на рис. 4.

### 1.4.1.1 Вариант использования Просмотр витрины кредитных продуктов

Краткое описание: данный вариант использования описывает просмотр клиентом банка списка актуальных кредитных продуктов.

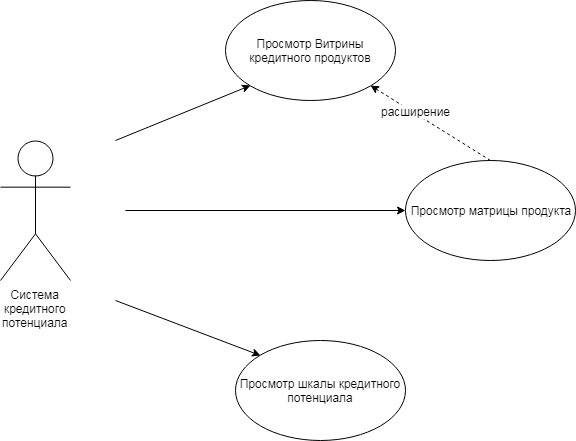


Рис. 4. Диаграмма вариантов использования

Ход событий при выполнении данного варианта использования изображен в табл. 1.

Таблица 1

Типичный ход событий

|  |  |
| --- | --- |
| Действие пользователя | Отклик системы |
| 1. Система кредитного потенциала выполняет запуск подсистемы конвертации | 1. Подсистема запускается |
| 1. Система отправляет запрос на просмотр витрины кредитных продуктов | 1. Подсистема получает от сервиса ППРБ файл с информацией обо всех доступных пользователю кредитных предложениях, и выполняет его конвертацию, затем отправляет сконвертированный файл на презентационный слой приложения (сайт или приложение) |

### 1.4.1.2 Вариант использования Просмотр матрицы продукта

Краткое описание: данный вариант использования описывает детальный просмотр клиентом банка одного из актуальных кредитных продуктов

Ход событий при выполнении данного варианта использования изображен в табл. 2 – табл. 3.

Таблица 2

Типичный ход событий

|  |  |
| --- | --- |
| Действие пользователя | Отклик системы |
| 1. Система кредитного потенциала выполняет запуск подсистемы конвертации | 1. Подсистема запускается |
| 1. Система отправляет запрос на просмотр выбранного внешним пользователем кредитного продукта из списка на главной странице | 1. Система получает от сервиса ППРБ файл с информацией о выбранном продукте, и выполняет его конвертацию, затем отправляет сконвертированный файл на презентационный слой приложения (сайт или приложение) |

Таблица 3

Альтернатива «Выбор продукта из Витрины»

|  |  |
| --- | --- |
| Действие пользователя | Отклик системы |
| 1 | 2 |
| 1. Система кредитного потенциала выполняет запуск подсистемы конвертации | 1. Подсистема запускается |

Продолжение табл. 3

|  |  |
| --- | --- |
| 1 | 2 |
| 1. Система отправляет запрос на просмотр витрины кредитных продуктов | 1. Подсистема получает от сервера файл с информацией обо всех доступных пользователю кредитных предложениях, и выполняет его конвертацию, затем отправляет готовый файл обратно на сервер |
| 1. Система отправляет запрос на просмотр выбранного внешним пользователем кредитного продукта | 1. Система получает от сервиса ППРБ файл с информацией о выбранном продукте, и выполняет его конвертацию, затем отправляет сконвертированный файл на презентационный слой приложения (сайт или приложение) |

### 1.4.1.3 Вариант использования Просмотр шкалы кредитного потенциала

Краткое описание: данный вариант использования описывает запрос клиентом банка на вывод кредитного потенциала

Ход событий при выполнении данного варианта использования изображен в табл. 4.

Таблица 4

Типичный ход событий

|  |  |
| --- | --- |
| Действие пользователя | Отклик системы |
| 1 | 2 |
| 1. Система кредитного потенциала выполняет запуск подсистемы конвертации | 1. Подсистема запускается |

Продолжение табл. 4

|  |  |
| --- | --- |
| 1 | 2 |
| 1. Система кредитного потенциала отправляет запрос на просмотр шкалы кредитного потенциала | 1. Подсистема получает от сервиса ППРБ файл с информацией о ранее рассчитанном кредитном потенциале пользователи, а также о сумме возможного кредитования, и выполняет его конвертацию, затем отправляет сконвертированный файл на презентационный слой приложения (сайт или мобильное приложение) |

### 1.4.2 Построение контекстных диаграмм классов

Диаграмма классов - диаграмма, отображающая классы, интерфейсы, кооперации, а также связь между ними [18].

Диаграммы классов содержат следующие объекты:

1. Классы;
2. Интерфейсы;
3. Кооперации;
4. Отношения.

Диаграммы классов могут содержать в себе некие пакеты или подсистемы, которые используются системой для группирования элементов модели.

Все классы в системе чаще всего делятся на группы. Принцип деления на группы жестко не регламентирован и может разниться от системы к системе, но, опираясь на опыт разработчиков, было выделено несколько более часто встречающихся групп [2]:

1. Граничные - объекты, реализующие связь интерфейсов с пользователями и внешней средой;
2. Сущностные - объекты-блоки, необходимый для хранения информации и организации баз данных и файловых систем;
3. Управляющие – активные объекты, реализующие управление процессами;
4. Классы прикладной логики – объекты, реализующие, логику решения задач системы.

Диаграммы классов используются при разработке и моделировании вида системы (с точки зрения проектирования) для описания всех функциональных требований, которые предъявляются к разрабатываемой системе.

Данный вид диаграмм используется при:

1. Моделировании словаря системы - для принятия решения о том, какие абстракции являются частью системы, а какие – нет;
2. Моделировании коопераций – объединений тех элементов, которые способны обеспечивать некоторое кооперативного поведения частей системы;
3. Моделировании логической схемы БД – схемы проекты базы данных. Чаще всего это осуществляется с помощью диаграмм классов.

Концептуальная модель предметной области представлена на рис. 5.

### 1.4.3 Построение диаграмм последовательностей системы

Диаграмма последовательностей отображает все взаимодействия между элементами системы, которые представлены как некоторая последовательность сообщений между экземплярами классов, компонентами системы, подсистемами и ее субъектами. Чтение диаграммы происходит сверху вниз, т.е. время увеличивается вниз по диаграмме [17].

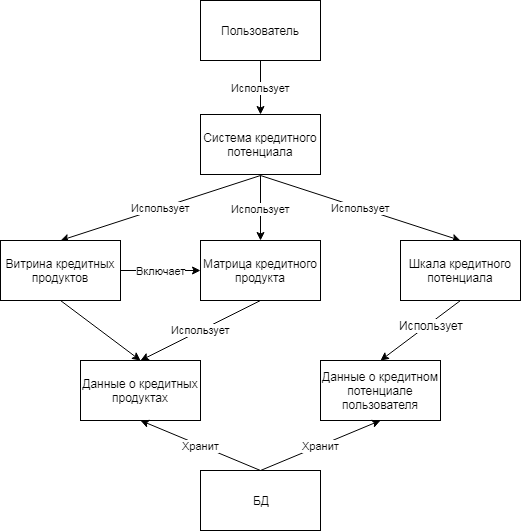


Рис. 5. Концептуальная модель предметной области

Целью создания диаграммы последовательностей системы является визуализация экземпляров и событий [3].

На диаграмме последовательностей допустимо использование стандартных стереотипов класса:

1. Actor – участник процесса;
2. Boundary – разграничитель, используется для классов, отделяющих внутреннюю структуру системы от внешней среды;
3. Control – контроллер, активный элемент, использующийся для выполнения операций над объектами;
4. Entity – сущность, используется для обозначения классов, которые хранящую информацию об объектах.

Диаграмма последовательностей системы для варианта использования Просмотр витрины кредитных продуктов

Диаграмма последовательностей представлена на рис. 6.

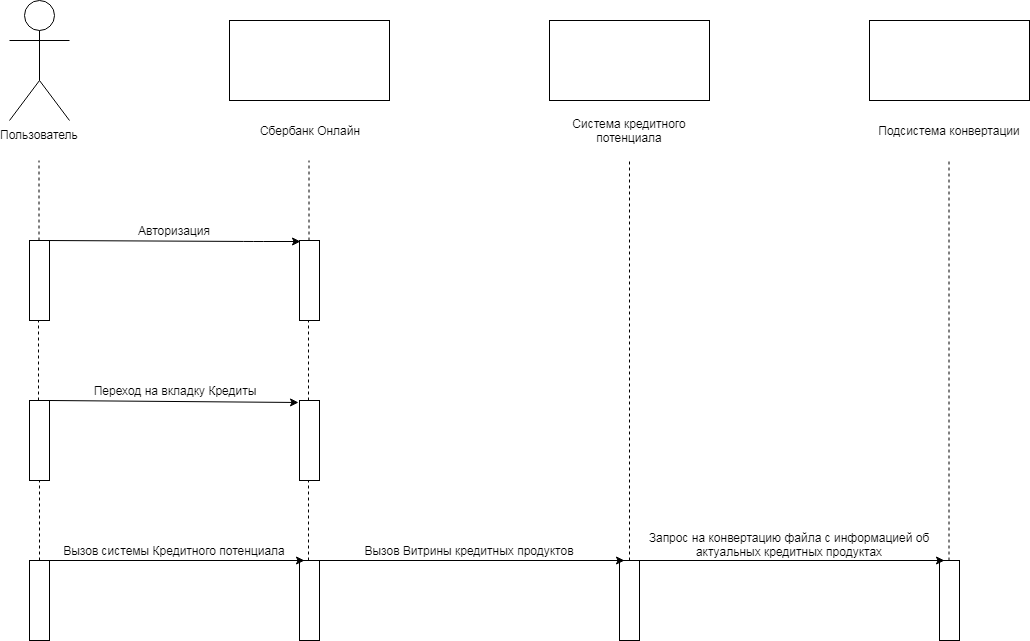


Рис. 6. Диаграмма последовательностей системы для варианта использования Просмотр витрины кредитных продуктов

Описание операции «Просмотр витрины кредитных продуктов» представлено в табл. 5 – табл. 9.

Таблица 5

Описание операции «Авторизация»

|  |  |
| --- | --- |
| Раздел | Описание |
| 1 | 2 |
| Имя | Авторизация |

Продолжение табл. 5

|  |  |
| --- | --- |
| 1 | 2 |
| Обязанности | Авторизация и вход в личный кабинет пользователя сервиса «Сбербанк Онлайн» |
| Тип | Системная |
| Ссылки | Вариант использования Просмотр витрины кредитных продуктов |
| Примечания |  |
| Исключения |  |
| Вывод | Вход в личный кабинет «Сбербанк Онлайн» выполнен |
| Предусловия |  |
| Постусловие |  |

Таблица 6

Описание операции «Переход на вкладку Кредиты»

|  |  |
| --- | --- |
| Раздел | Описание |
| Имя | Переход на вкладку Кредиты |
| Обязанности | Переход пользователя из Личного кабинета в раздел «Кредиты» |
| Тип | Системная |
| Ссылки | Вариант использования Просмотр витрины кредитных продуктов |
| Примечания |  |
| Исключения |  |
| Вывод | «Сбербанк Онлайн» открывает вкладку Кредиты |
| Предусловия |  |

Таблица 7

Описание операции «Вызов системы кредитного потенциала»

|  |  |
| --- | --- |
| Раздел | Описание |
| Имя | Вызов системы кредитного потенциала |
| Обязанности | Вход пользователя в систему кредитного потенциала |
| Тип | Системная |
| Ссылки | Вариант использования Просмотр витрины кредитных продуктов |
| Примечания |  |
| Исключения |  |
| Вывод | Система кредитного потенциала запускается |
| Предусловия |  |
| Постусловие |  |

Таблица 8

Описание операции «Вызов Витрины кредитных продуктов»

|  |  |
| --- | --- |
| Раздел | Описание |
| Имя | Вызов Витрины кредитных продуктов |
| Обязанности | Осуществление перехода пользователя в систему кредитного потенциала |
| Тип | Системная |
| Ссылки | Вариант использования Просмотр витрины кредитных продуктов |
| Примечания |  |
| Исключения |  |
| Вывод | Система кредитного потенциала была запущена |
| Предусловия |  |
| Постусловие |  |

Таблица 9

Описание операции «Запрос на конвертацию файла с информацией об актуальных кредитных продуктах»

|  |  |
| --- | --- |
| Раздел | Описание |
| Имя | Запрос на конвертацию файла с информацией об актуальных кредитных продуктах |
| Обязанности | Конвертация полученного от системы кредитного потенциала файла |
| Тип | Системная |
| Ссылки | Вариант использования Просмотр витрины кредитных продуктов |
| Примечания |  |
| Исключения |  |
| Вывод | Подсистема выполнила конвертацию полученного файла |
| Предусловия |  |
| Постусловие | Синхронизация с БД |

Диаграмма последовательностей системы для варианта использования Просмотр матрицы продукта

Диаграмма последовательностей представлена на рис. 7.

Описание операции «Просмотр матрицы продукта» представлено в табл. 10 – табл. 14.

Таблица 10

Описание операции «Авторизация»

|  |  |
| --- | --- |
| Раздел | Описание |
| 1 | 2 |
| Имя | Авторизация |

Продолжение табл. 10

|  |  |
| --- | --- |
| 1 | 2 |
| Обязанности | Авторизация и вход в личный кабинет пользователя сервиса «Сбербанк Онлайн» |
| Тип | Системная |
| Ссылки | Вариант использования Просмотр матрицы продукта |
| Примечания |  |
| Исключения |  |
| Вывод | Вход в личный кабинет «Сбербанк Онлайн» выполнен |
| Предусловия |  |
| Постусловие |  |

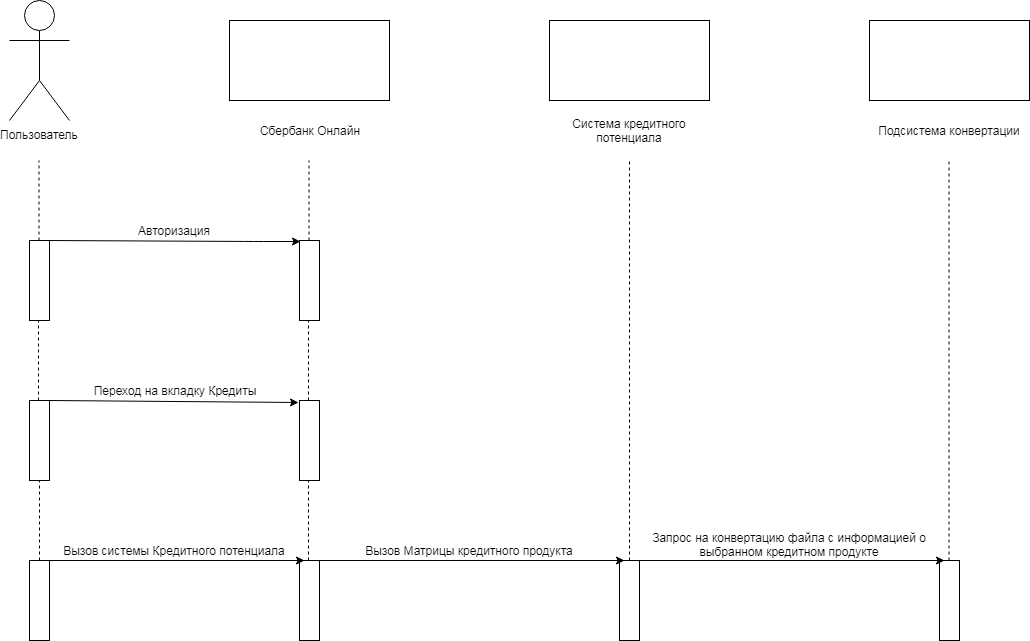


Рис. 7. Диаграмма последовательностей системы для варианта использования Просмотр матрицы продукта

Таблица 11

Описание операции «Переход на вкладку Кредиты»

|  |  |
| --- | --- |
| Раздел | Описание |
| Имя | Переход на вкладку Кредиты |
| Обязанности | Переход пользователя из Личного кабинета в раздел «Кредиты» |
| Тип | Системная |
| Ссылки | Вариант использования Просмотр матрицы продукта |
| Примечания |  |
| Исключения |  |
| Вывод | «Сбербанк Онлайн» открывает вкладку Кредиты |
| Предусловия |  |

Таблица 12

Описание операции «Вызов системы кредитного потенциала»

|  |  |
| --- | --- |
| Раздел | Описание |
| Имя | Вызов системы кредитного потенциала |
| Обязанности | Вход пользователя в систему кредитного потенциала |
| Тип | Системная |
| Ссылки | Вариант использования Просмотр матрицы продукта |
| Примечания |  |
| Исключения |  |
| Вывод | Система кредитного потенциала запускается |
| Предусловия |  |
| Постусловие |  |

Таблица 13

Описание операции «Вызов Витрины кредитных продуктов»

|  |  |
| --- | --- |
| Раздел | Описание |
| Имя | Вызов Витрины кредитных продуктов |
| Обязанности | Осуществление перехода пользователя в систему кредитного потенциала |
| Тип | Системная |
| Ссылки | Вариант использования Просмотр матрицы продукта |
| Примечания |  |
| Исключения |  |
| Вывод | Система кредитного потенциала была запущена |
| Предусловия |  |
| Постусловие |  |

Таблица 14

Описание операции «Запрос на конвертацию файла с информацией о выбранном кредитном продукте»

|  |  |
| --- | --- |
| Раздел | Описание |
| Имя | Запрос на конвертацию файла с информацией о выбранном пользователем кредитном продукте |
| Обязанности | Конвертация полученного от системы кредитного потенциала файла |
| Тип | Системная |
| Ссылки | Вариант использования Просмотр матрицы продукта |
| Примечания |  |
| Исключения |  |
| Вывод | Подсистема выполнила конвертацию полученного файла |
| Предусловия |  |
| Постусловие | Синхронизация с БД |

Диаграмма последовательностей системы для варианта использования Просмотр шкалы кредитного потенциала

Диаграмма последовательностей представлена на рис. 8.

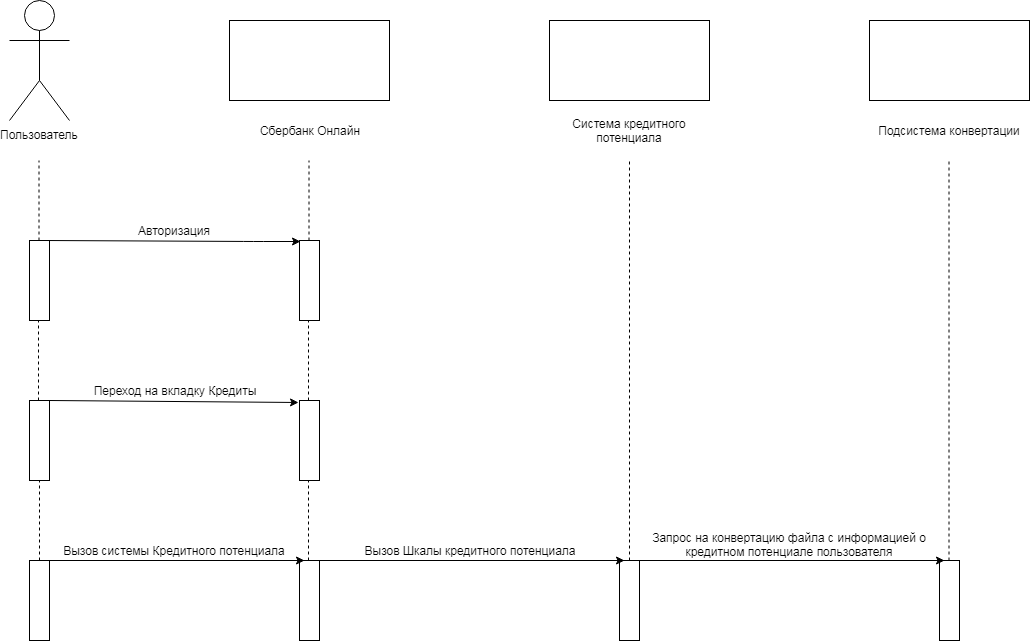


Рис. 8. Диаграмма последовательностей системы для варианта использования Просмотр шкалы кредитного потенциала

Описание операции «Просмотр шкалы кредитного потенциала» представлено в табл. 15 – табл. 19.

Таблица 15

Описание операции «Авторизация»

|  |  |
| --- | --- |
| Раздел | Описание |
| 1 | 2 |
| Имя | Авторизация |
| Обязанности | Авторизация и вход в личный кабинет пользователя сервиса «Сбербанк Онлайн» |

Продолжение табл. 15

|  |  |
| --- | --- |
| 1 | 2 |
| Тип | Системная |
| Ссылки | Вариант использования Просмотр шкалы кредитного потенциала |
| Примечания |  |
| Исключения |  |
| Вывод | Вход в личный кабинет «Сбербанк Онлайн» выполнен |
| Предусловия |  |
| Постусловие |  |

Таблица 16

Описание операции «Переход на вкладку Кредиты»

|  |  |
| --- | --- |
| Раздел | Описание |
| Имя | Переход на вкладку Кредиты |
| Обязанности | Переход пользователя из Личного кабинета в раздел «Кредиты» |
| Тип | Системная |
| Ссылки | Вариант использования Просмотр шкалы кредитного потенциала |
| Примечания |  |
| Исключения |  |
| Вывод | «Сбербанк Онлайн» открывает вкладку Кредиты |
| Предусловия |  |

Таблица 17

Описание операции «Вызов системы кредитного потенциала»

|  |  |
| --- | --- |
| Раздел | Описание |
| 1 | 2 |

Продолжение табл. 17

|  |  |
| --- | --- |
| 1 | 2 |
| Имя | Вызов системы кредитного потенциала |
| Обязанности | Вход пользователя в систему кредитного потенциала |
| Тип | Системная |
| Ссылки | Вариант использования Просмотр шкалы кредитного потенциала |
| Примечания |  |
| Исключения |  |
| Вывод | Система кредитного потенциала запускается |
| Предусловия |  |
| Постусловие |  |

Таблица 18

Описание операции «Вызов Витрины кредитных продуктов»

|  |  |
| --- | --- |
| Раздел | Описание |
| Имя | Вызов Витрины кредитных продуктов |
| Обязанности | Осуществление перехода пользователя в систему кредитного потенциала |
| Тип | Системная |
| Ссылки | Вариант использования Просмотр шкалы кредитного потенциала |
| Примечания |  |
| Исключения |  |
| Вывод | Система кредитного потенциала была запущена |
| Предусловия |  |
| Постусловие |  |

Таблица 19

Описание операции «Запрос на конвертацию файла с информацией о кредитном потенциале пользователя»

|  |  |
| --- | --- |
| Раздел | Описание |
| Имя | Запрос на конвертацию файла с информацией о ранее рассчитанном кредитном потенциале пользователя |
| Обязанности | Конвертация полученного от системы кредитного потенциала файла |
| Тип | Системная |
| Ссылки | Вариант использования Просмотр шкалы кредитного потенциала |
| Примечания |  |
| Исключения |  |
| Вывод | Подсистема выполнила конвертацию полученного файла |
| Предусловия |  |
| Постусловие | Синхронизация с БД |

### 1.4.4 Построение диаграмм деятельностей сценариев вариантов использования

Применение диаграмм деятельностей возможно для описания событий в вариантах использования. Весь поток событий возможно объяснить с помощью словесного описания, но это применимо только к небольшим потокам, не имеющим множества альтернативных решений.

Основным элементом диаграмм деятельностей является сама деятельность, изображенная на диаграмме в виде закругленного прямоугольника с текстовым описанием.

Любая диаграмма деятельности имеет точку входа, которая является началом событий. При этом, конечная точка не всегда является обязательной. Также точка, показывающая окончание событий, может быть не одна.

На диаграмме могут присутствовать объекты и их потоки. Отображение объектов и состояний объектов дает разработчику понятие о том, как и в какой момент происходит смена состояний объекта. Связь объектов с деятельностями происходит с использованием потоков объектов.

### 1.4.4.1 Построение диаграммы деятельностей для сценария варианта использования Просмотр витрины кредитных продуктов

Учитывая описание предметной области в виде контекстной диаграммы классов, анализируем описание сценария варианта использования Вывод витрины кредитных продуктов. Разбиваем процесс на отдельные операции. Полученные операции показаны на диаграмме деятельностей (рис. 9). Описание деятельностей представлено в табл. 20.

Таблица 20

Описание деятельностей для сценария варианта использования Просмотр витрины кредитных продуктов

|  |  |  |
| --- | --- | --- |
| № | Деятельность | Описание |
| 1 | 2 | 3 |
| 1 | Авторизация | Пользователь выполняет вход в сервис «Сбербанк Онлайн» |
| 2 | Переход на главную страницу сервиса «Сбербанк Онлайн» | Выполняется перенаправление на домашнюю страницу сервиса «Сбербанк Онлайн» |
| 3 | Переход на вкладку «Кредиты» | Отображение экранной формы «Кредиты» по запросу пользователя |
| 4 | Переход в Систему кредитного потенциала | Из вкладки «Кредиты» пользователь выполняет переход к Системе кредитного потенциала |
| 5 | Вызов Витрины кредитных продуктов | Пользователь выбирает опцию просмотра Витрины кредитных продуктов |

Продолжение табл. 20

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| 6 | Системный запрос на конвертацию файла с информацией об актуальных кредитных продуктах | Система кредитного потенциала отправляет исходный файл с данными об актуальных кредитных продуктах подсистеме конвертации |

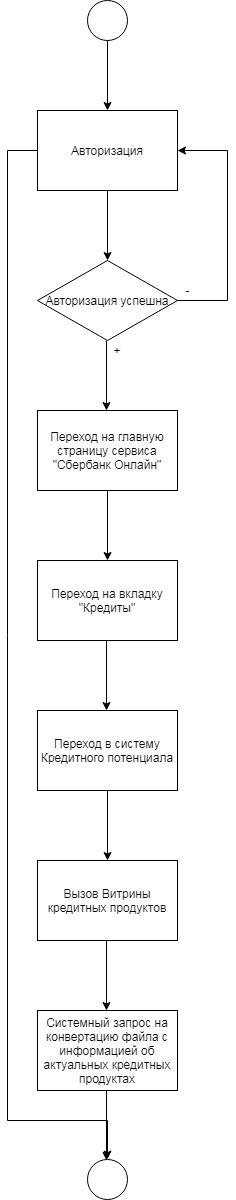


Рис. 9. Диаграмма деятельностей для сценария варианта использования Просмотр витрины кредитных продуктов

### 1.4.4.2 Построение диаграммы деятельностей для сценария варианта использования Просмотр Матрицы продукта

Полученные операции показаны на диаграмме деятельностей (рис. 10). Описание деятельностей представлено в табл. 21.

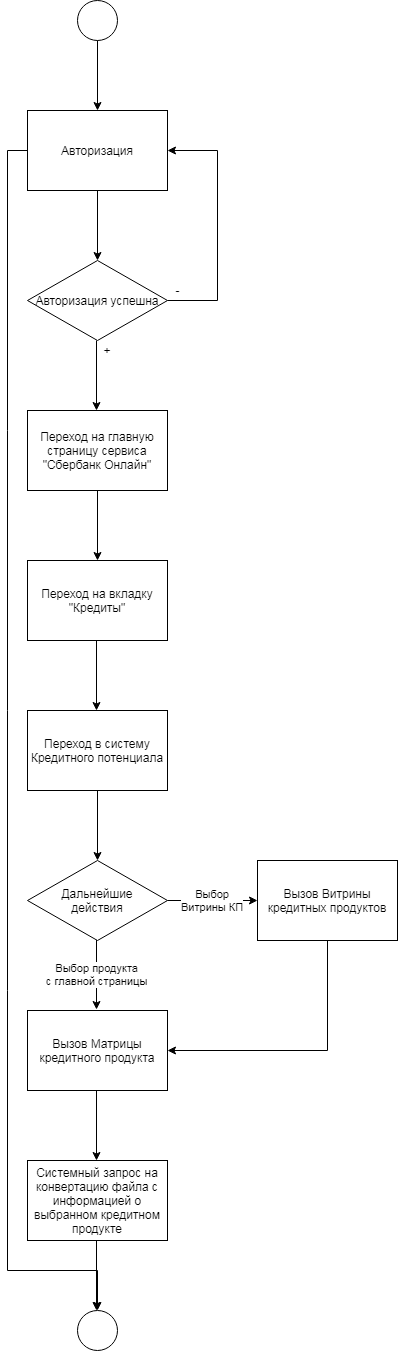


Рис. 10. Диаграмма деятельностей для сценария варианта использования Просмотр Матрицы продукта

Таблица 21

Описание деятельностей для сценария варианта использования Просмотр Матрицы продукта

|  |  |  |
| --- | --- | --- |
| № | Деятельность | Описание |
| 1 | Авторизация | Пользователь выполняет вход в сервис «Сбербанк Онлайн» |
| 2 | Переход на главную страницу сервиса «Сбербанк Онлайн» | Выполняется перенаправление на домашнюю страницу сервиса «Сбербанк Онлайн» |
| 3 | Переход на вкладку «Кредиты» | Отображение экранной формы «Кредиты» по запросу пользователя |
| 4 | Переход в Систему кредитного потенциала | Из вкладки «Кредиты» пользователь выполняет переход к Системе кредитного потенциала |
| 5 | Вызов Витрины кредитных продуктов | Пользователь выбирает опцию просмотра Витрины кредитных продуктов |
| 6 | Вызов Матрицы кредитного продукта | Пользователь выбирает опцию просмотра Матрицы отдельного кредитного продукта |
| 7 | Системный запрос на конвертацию файла с информацией об актуальных кредитных продуктах | Система кредитного потенциала отправляет исходный файл с данными о выбранном пользователем кредитном продукте |

### 1.4.4.3 Построение диаграммы деятельностей для сценария варианта использования Просмотр шкалы кредитного потенциала

Полученные операции показаны на диаграмме деятельностей (рис. 11). Описание деятельностей представлено в табл. 22.

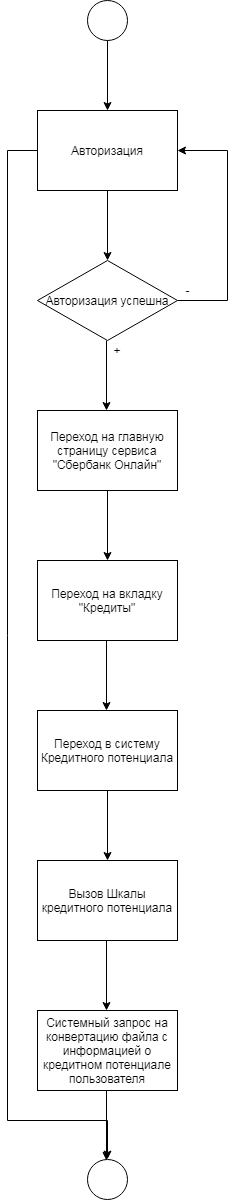


Рис. 11. Диаграмма деятельностей для сценария варианта использования Просмотр шкалы кредитного потенциала

### 1.4.5 Построение структур данных и построение диаграмм отношений компонентов данных

Для нормальной и полноценной работы почти любой информационной системы, требуется наличие баз данных (БД). В зависимости от выбранной предметной области, происходит формирование первой версии таблицы базы данных, на основе которой происходит разработка итоговых таблиц путем нормализации.

Нормализация данных – процесс изменения базы данных, при котором избыточность данных сводится к минимуму, но целостность этих данных сохраняется.

Нормализованной считается такая база данных, таблицы которой не нарушают правил нормальных форм.

Нормальная форма – правила, отвечающие за корректность организации данных.

В современных условиях, для комфортного использования, достаточно следовать первым трем нормальных формам.

Процесс нормализации представлен в прил. 2.

* 1. Проектирование системы

### 1.5.1 Проектирование структуры системы и построение диаграммы пакетов

При выполнении анализа диаграммы вариантов использования (рис. 4), были выделены следующие пакеты:

1. Продукты – классы, реализующие анализ и обработку информации, связанной с кредитными продуктами
2. Пользователи - классы, реализующие анализ и обработку информации, связанной с кредитным потенциалом пользователей системы
3. База данных – классы, описывающие взаимодействия разрабатываемой системы с базами данных
4. Обработка исключительных ситуаций – классы, предназначенные для обработки ошибок

Диаграмма пакетов представлена на рис. 12.

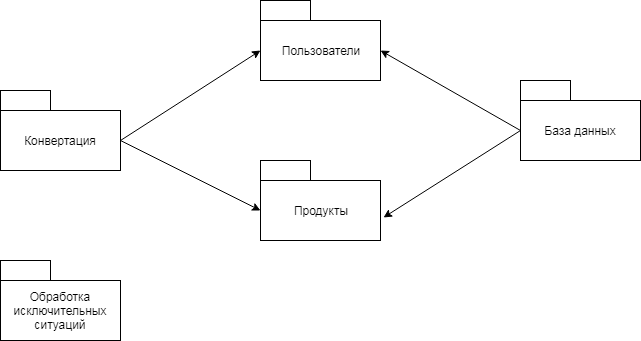


Рис. 12. Диаграмма пакетов

1.5.2 Проектирование классов в пакетах

В данном пункте рассматривается проектирование пакета Конвертация.

### 1.5.2.1 Проектирование пакета Конвертация

В данном пункте рассматривается проектирование пакета Конвертация.

1.5.2.1.1 Проектирование классов пакета Конвертация

Описание классов рассматриваемого пакета представлено на рис. 13.

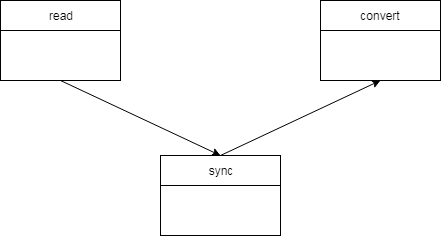


Рис. 13. Классы пакета Конвертация

Таблица 23

Описание классов пакета Конвертация

|  |  |
| --- | --- |
| Класс | Описание |
| read | Класс содержит методы для считывания данных из исходного файла |
| sync | Класс содержит методы для синхронизации и объединения информации о пользователях и продуктах из базы данных |
| convert | Класс содержит методы для выполнения конвертации готового файла |

1.5.2.1.2 Построение диаграмм последовательностей действий

Диаграммы последовательностей изображают взаимодействие между группами объектов и их поведения.

На данных диаграммах изображаются внутренние объекты программы и последовательность сообщений, необходимых для связи объектов в процессе реализации вариантов использования.

В ходе выполнения анализа вариантов использования было выявлено ситуаций:

1. Нормальный ход событий;
2. Ошибка чтения файла;
3. Ошибка синхронизации с базой данных;
4. Ошибка конвертации файла.

На рис. 14 – рис. 17 изображены диаграммы последовательностей взаимодействия объектов классов для пакета Конвертация. Диаграммы отображают взаимодействие классов в выше описанных ситуациях.

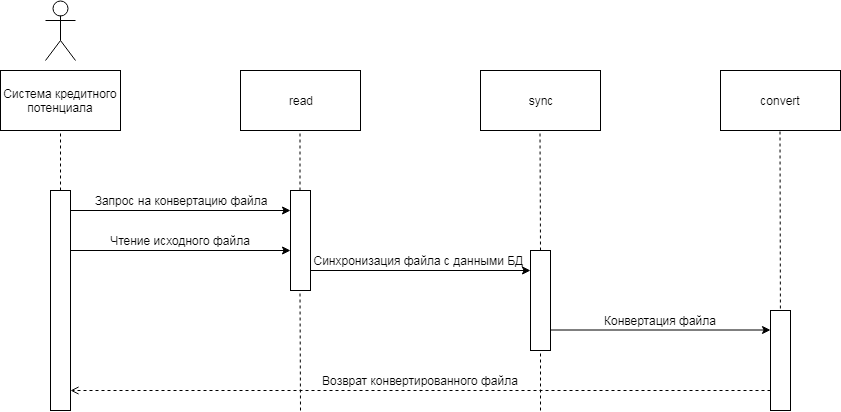


Рис. 14. Нормальный ход событий

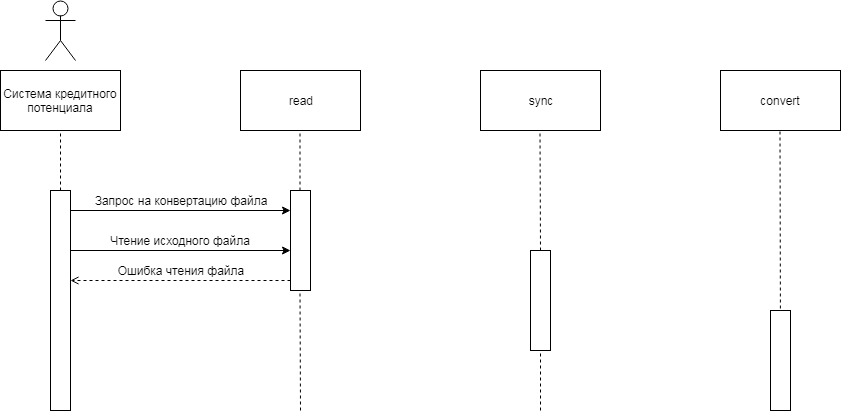


Рис. 15. Ошибка чтения файла

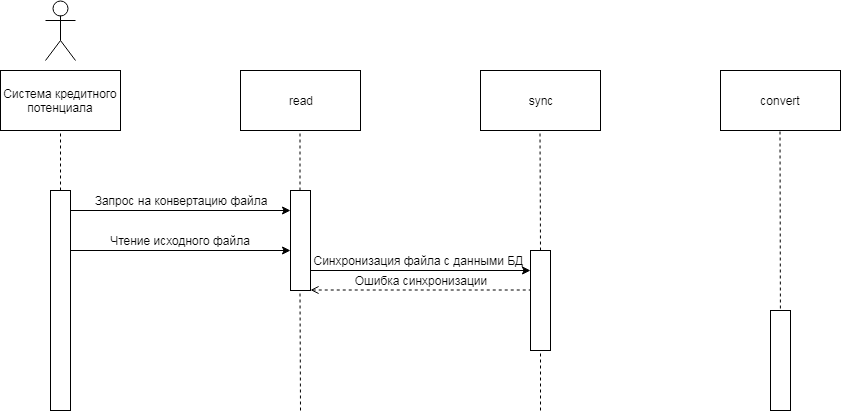


Рис. 16. Ошибка синхронизации с базой данных

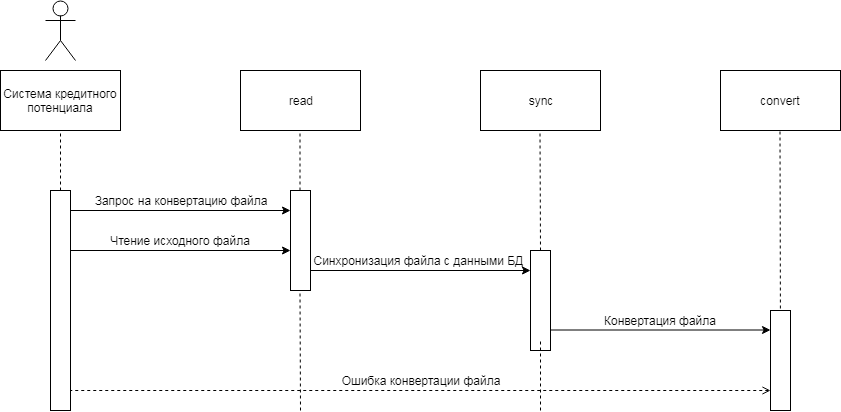


Рис. 17. Ошибка конвертации файла

1.5.2.1.3 Построение диаграммы коопераций

Кооперации - объединение множества объектов классов, которые взаимодействуют с целью реализации вариантов использования системы.

Диаграмма кооперации отображает участвующие во взаимодействии объекты и ассоциации между ними. При необходимости, на диаграмме указываются также имена объектов и динамические связи.

Диаграмма коопераций пакета Конвертация представлена на рис. 18.

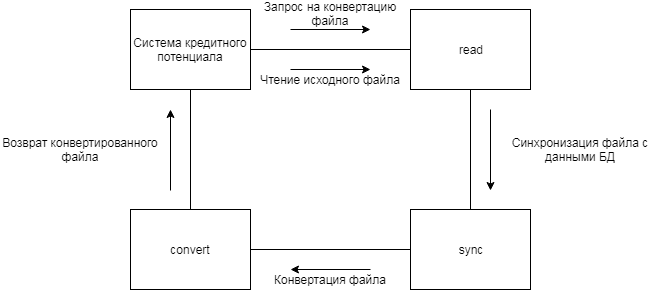


Рис. 18. Диаграмма кооперации

1.5.2.1.4 Построение уточненной диаграммы классов пакета Конвертация

Уточненная диаграмма классов рассматриваемого пакета представлена на рис. 19.

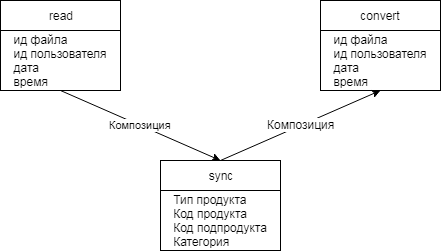


Рис. 19. Уточненная диаграмма классов

Связь между всеми классами пакета – композиция. Это означает, что, при уничтожении какого-либо объекта в одном из классов, будут уничтожены объекты и в других связанных классах.

1.5.2.1.5 Построение детальной диаграммы классов пакета Конвертация

Детальная диаграмма классов рассматриваемого пакета представлена на рис. 20. В табл. 24 – табл. 29 представлено описание полей и методов классов.

Для построения детальной диаграммы классов требуется проектирование всей структуры класса и поведения его объектов.

Структура объекта - совокупность атрибутов и операций данного класса.

Атрибут класса - поле или совокупность полей, содержащихся в объекте класса [4].



Рис. 20. Детальная диаграмма классов

Таблица 24

Описание полей класса «read»

|  |  |
| --- | --- |
| Название поля | Назначение |
| ид файла | Идентификационный номер полученного документа |
| ид пользователя | Идентификационный номер пользователя |
| дата | Дата получения файла |
| время | Время получения файла |

Таблица 25

Описание методов класса «read»

|  |  |
| --- | --- |
| Название метода | Назначение |
| Конструктор() | Конструктор по умолчанию, для создания пустого объекта класса read |
| Конструктор(ид файла, ид пользователя, дата, время) | Конструктор по умолчанию, для создания объекта класса read с информацией о полученном файле |
| Принять (ид файла) | Функция для получения файла |
| Считать (ид файла) | Получить всю информацию из файла |
| Парсинг (ид файла) | Создание объектов класса. Каждая строка файла – отдельный класс |
| Деструктор() | Деструктор для удаления объекта класса read |

Таблица 26

Описание полей класса «sync»

|  |  |
| --- | --- |
| Название поля | Назначение |
| 1 | 2 |
| тип продукта | Тип кредитного продукта |
| код продукта | Код кредитного продукта |
| код подпродукта | Код кредитного подпродукта |
| категория | Наименование категории продукта |

Таблица 27

Описание методов класса «sync»

|  |  |
| --- | --- |
| Название метода | Назначение |
| 1 | 2 |
| Конструктор() | Конструктор по умолчанию, для создания пустого объекта класса sync |

Продолжение табл. 27

|  |  |
| --- | --- |
| 1 | 2 |
| Конструктор(тип продукта, код продукта, код подпродукта, категория) | Конструктор по умолчанию, для создания объекта класса sync с информацией о полях |
| ПодключениеБД() | Функция для подключения подсистемы к базе данных |
| ПолучитьСтроку() | Функция, выполняющая поиск ключевых строк в файле |
| ЗаменаЗначения() | Функция, выполняющая замену идентификационных номеров в документе на соответствующие им значения |
| Деструктор() | Деструктор для удаления объекта класса sync |

Таблица 28

Описание полей класса «convert»

|  |  |
| --- | --- |
| Название поля | Назначение |
| 1 | 2 |
| ид файла | Идентификационный номер документа |
| ид пользователя | Идентификационный номер пользователя |
| дата | Дата получения файла |
| время | Время получения файла |

Таблица 29

Описание методов класса «convert»

|  |  |
| --- | --- |
| Название метода | Назначение |
| 1 | 2 |

Продолжение табл. 29

|  |  |
| --- | --- |
| 1 | 2 |
| Конструктор() | Конструктор по умолчанию, для создания пустого объекта класса convert |
| Конструктор(ид файла, ид пользователя, дата, время) | Конструктор по умолчанию, для создания объекта класса convert с информацией о полученном файле |
| Считать() | Функция для считывания строк файла |
| Конвертировать() | Функция, выполняющая конвертацию файла |
| Отправить() | Функция, выполняющая отправку готового файла сервису ППРБ |
| Деструктор() | Деструктор для удаления объекта класса convert |

### 1.5.2.2 Проектирование пакета База данных

В данном пункте рассматривается проектирование пакета База данных.

На основе анализа предметной области, проведенного ранее, был сформирован первый вариант таблиц с данными.

Логический уровень проектирования базы данных представлен в разделе 1.4.5

Физическая модель представлена на рис. 21.

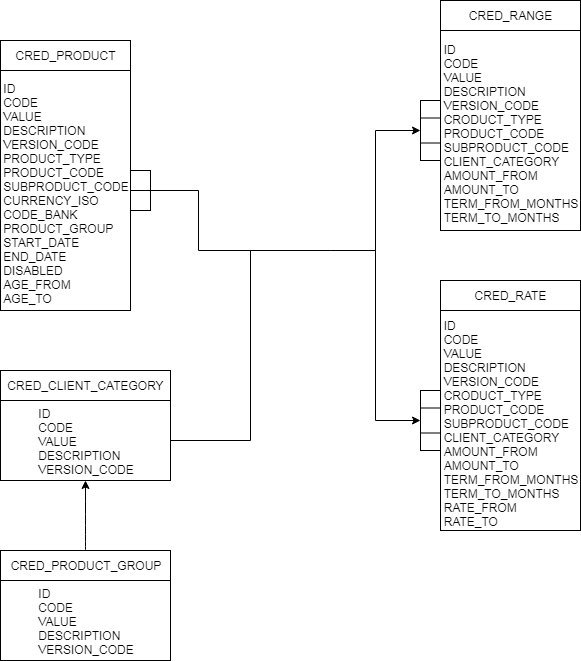


Рис. 21. Физическая модель базы данных

После создания физической модели данных была спроектирована БД.

Таблица 30

Идентификаторы таблицы «CRED\_PRODUCT»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибут | Идентификатор | Тип | Размерность | Описание |
| 1 | 2 | 3 | 4 | 5 |
| ИД продукта | ID | char | 36 | Идентифи-кационный номер продукта |

Продолжение табл. 30

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |
| Код | CODE | varchar2 | 64 | Код продукта |
| Значение | VALUE | varchar2 | 100 | Значение продукта |
| Описание | DESCRIPTION | varchar2 | 64 | Описание продукта |
| Код версии | VERSION\_CODE | varchar2 | 64 | Код версии продукта |
| Тип продукта | PRODUCT\_TYPE | varchar2 | 30 | Тип продукта |
| Код продукта | PRODUCT\_CODE | varchar2 | 30 | Код продукта |
| Код подпродукта | SUBPRODUCT\_CODE | varchar2 | 30 | Код подпродук-та |
| Код валюты | CURRENCY\_ISO | varchar2 | 3 | Код представле-ния валюты |
| Код банка | CODE\_BANK | number | 2,0 | Код банка |
| Группа продукта | PRODUCT\_GROUP | varchar2 | 64 | Группа продукта |
| Начальная дата | START\_DATE | date | - | Дата начала |
| Конечная дата | END\_DATE | date | - | Дата окончания |
| Неактивно | DISABLED | number | 1,0 | Доступ-ность продукта |
| Возраст от | AGE\_FROM | number | 2,0 | Возраст от |
| Возраст до | AGE\_TO | number | 2,0 | Возраст до |

Таблица 31

Идентификаторы таблицы «CLIENT\_CATEGORY»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибут | Идентификатор | Тип | Размерность | Описание |
| ИД категории | ID | char | 36 | Идентифи-кационный номер категории |
| Код | CODE | varchar2 | 64 | Код категории |
| Значение | VALUE | varchar2 | 64 | Значение категории |
| Описание | DESCRIPTION | varchar2 | 64 | Описание категории |
| Версия кода | VERSION\_CODE | varchar2 | 64 | Версия кода категории |

Таблица 32

Идентификаторы таблицы «PRODUCT\_GROUP»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибут | Идентификатор | Тип | Размерность | Описание |
| 1 | 2 | 3 | 4 | 5 |
| ИД категории | ID | char | 36 | Идентифи-кационный номер группы продукта |
| Код | CODE | varchar2 | 64 | Код группы продукта |
| Значение | VALUE | varchar2 | 64 | Значение группы продукта |

Продолжение табл. 32

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |
| Описание | DESCRIPTION | varchar2 | 64 | Описание группы продукта |
| Версия кода | VERSION\_CODE | varchar2 | 64 | Версия кода группы продукта |

Таблица 33

Идентификаторы таблицы «CRED\_RANGE»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибут | Идентификатор | Тип | Размерность | Описание |
| 1 | 2 | 3 | 4 | 5 |
| ИД шкалы | ID | char | 36 | Идентифи-кационный номер шкалы |
| Код | CODE | varchar2 | 64 | Код шкалы |
| Значение | VALUE | varchar2 | 64 | Значение шкалы |
| Описание | DESCRIPTION | varchar2 | 64 | Описание шкалы |
| Версия кода | VERSION\_CODE | varchar2 | 64 | Код версии шкалы |
| Тип продукта | PRODUCT\_TYPE | varchar2 | 30 | Тип продукта |
| Код продукта | PRODUCT\_CODE | varchar2 | 30 | Код продукта |
| Код подпродукта | SUBPRODUCT\_CODE | varchar2 | 30 | Код подпродук-та |

Продолжение табл. 33

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |
| Категория | CLIENT\_CATEGORY | varchar2 | 30 | Категория клиента |
| Сумма от | AMOUNT\_FROM | number | 12,2 | Сумма от |
| Сумма до | AMOUNT\_TO | number | 12,2 | Сумма до |
| В месяц от | TERM\_FROM\_MONTHS | number | 4,0 | Выплата в месяц от |
| В месяц до | TERM\_TO\_MONTHS | number | 4,0 | Выплата в месяц до |

Таблица 34

Идентификаторы таблицы «CRED\_RATE»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибут | Идентификатор | Тип | Размерность | Описание |
| 1 | 2 | 3 | 4 | 5 |
| ИД рейтинга | ID | char | 36 | Идентифи-кационный номер рейтинга |
| Код | CODE | varchar2 | 64 | Код рейтинга |
| Значение | VALUE | varchar2 | 64 | Значение рейтинга |
| Описание | DESCRIPTION | varchar2 | 64 | Описание рейтинга |
| Версия кода | VERSION\_CODE | varchar2 | 64 | Код версии рейтинга |
| Тип продукта | PRODUCT\_TYPE | varchar2 | 30 | Тип продукта |
| Код продукта | PRODUCT\_CODE | varchar2 | 30 | Код продукта |

Продолжение табл. 34

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |
| Код подпродукта | SUBPRODUCT\_CODE | varchar2 | 30 | Код подпродук-та |
| Категория | CLIENT\_CATEGORY | varchar2 | 30 | Категория клиентов |
| Сумма от | AMOUNT\_FROM | number | 12,2 | Сумма от |
| Сумма до | AMOUNT\_TO | number | 12,2 | Сумма до |
| В месяц от | TERM\_FROM\_MONTHS | number | 4,0 | Выплата в месяц от |
| В месяц до | TERM\_TO\_MONTHS | number | 4,0 | Выплата в месяц до |
| Рейтинг от | RATE\_FROM | number | 5,2 | Рейтинг от |
| Рейтинг до | RATE\_TO | number | 5,2 | Рейтинг до |

### 1.5.3 Построение диаграмм компонентов

Диаграммы компонентов отображают то, как выглядит модель системы на физическом уровне. На данных диаграммах изображаются все компоненты программного обеспечения и связи между ними.

Каждый класс модели преобразуется в компонент исходного кода. После создания, они присоединяются к диаграмме компонентов. Между отдельными компонентами программы изображают зависимости, которые соответствуют зависимостям на этапе компиляции или выполнения программы.

Диаграмма компонентов представлена на рис.22, описание компонентов предоставлено в табл. 35.

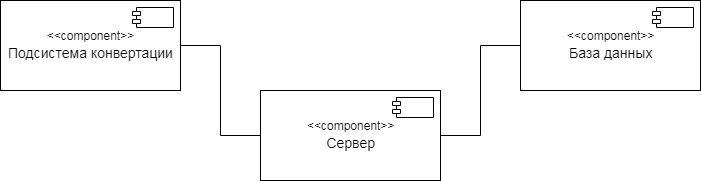


Рис. 22. Диаграмма компонентов

Таблица 35

Описание компонентов

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование | Назначение | Входные данные | Выходные данные |
| Подсистема конвертации | Предназначена для считывания и обработки исходного XML-файла и последующей его конвертации | Исходный файл в формате XML | Конвертированный файл в формате JSON |
| Сервер | Предназначен для приема и отправки файлов, а также для связи приложения «Сбербанк Онлайн» с сервером | Запрос на конвертацию файла | Результат запроса |
| База данных | Предназначена для хранения данных о кредитных продуктах и кредитном потенциале пользователей | Данные, добавляемые уполномоченными сотрудниками АО «Сбербанк Технологии» | БД, содержащая данные о кредитных продуктах и кредитном потенциале пользователей |

### 1.5.4 Построение диаграмм размещения

При физическом проектировании распределенных программных систем необходимо определить наиболее оптимальный вариант размещения программных компонентов на реальном оборудовании в локальной или глобальной сетях. Для этого используют специальную модель UML - диаграмму размещения (рис. 22).

На диаграмме размещения отображается топология системы, распределение ее компонентов по узлам и маршруты, необходимые для передачи информации между узлами.

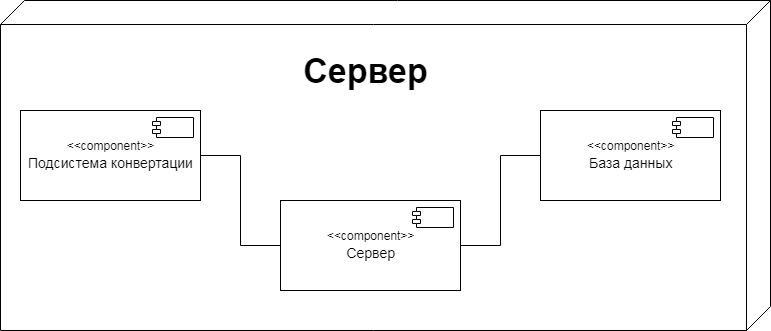


Рис. 23. Диаграмма размещения

Компоненты программного обеспечения связаны между собой. Приложение Подсистема конвертации, а также база данных, используемая данным приложением, расположены на сервере. При отправке конечным пользователем (клиентом банка) запроса на вывод Витрины кредитных продуктов, Матрицы продукта или Шкалы кредитного потенциала, приложение Подсистема конвертации начинает работу.

### 1.5.5 Спецификация модулей

Описание модулей подсистемы конвертации представлено в табл.36-39.

Таблица 36

Компоненты подсистемы конвертации

|  |  |
| --- | --- |
| Название компонента | Описание |
| 1 | 2 |
| body.java | Класс, отвечающий за формирование структуры документа |
| capacity\_unit.java | Класс, отвечающий за формирование заголовка файла |
| config.java | Класс, содержащий методы для работы и передачи номера документа |
| dashboard.java | Класс, отвечающий за работу с информацией о доступных и недоступных пользователю кредитных продуктах |
| GetClientRiskMetricsRs.java | Класс, содержащий методы для работы со структурой документа |
| lendingCapacity.java | Класс, содержащий методы для преобразования и корректного отображения дат и чисел (сумм) |
| message.java | Класс, отвечающий за корректное отображение текста в конечном файле |
| option.java | Класс, возвращающий значения идентификационных номеров в документе |
| product.java | Класс, содержащий методы для работы с информацией о кредитных продуктах матрицы |

Продолжение табл. 36

|  |  |
| --- | --- |
| 1 | 2 |
| subProduct.java | Класс, содержащий методы для работы со шкалой кредитного потенциала и подпродуктами |
| xml2json\_v01.java | Класс, отвечающий за формирование конечного файла |

Таблица 37

Спецификация компонента xml2json\_v01.java

|  |  |
| --- | --- |
| Функция или свойство | Описание |
| main(String[] args) | Запуск функций конвертера, разложение файла на классы |
| generateJSON(GetClientRiskMetricsRs obj) | Функция генерации JSON-файла |

Таблица 38

Спецификация компонента dashboard.java

|  |  |
| --- | --- |
| Функция или свойство | Описание |
| 1 | 2 |
| dashboard(List<String> \_messages, List<product> \_availableProducts, List<product> \_unavailableProducts, List<Double> \_loans) | Отбор доступных и недоступных продуктов из исходного документа |
| setAvailableProducts (List<product> availableProducts) | Формирование списка доступных клиенту продуктов |

Продолжение табл. 38

|  |  |
| --- | --- |
| 1 | 2 |
| etUnavailableProducts (List<product> unavailableProducts) | Формирование списка недоступных клиенту продуктов |

Таблица 39

Спецификация компонента lendingCapacity.java

|  |  |
| --- | --- |
| Функция или свойство | Описание |
| lendingCapacity(Boolean \_calculated, String \_calculationDate, Boolean \_expired, String \_expirationDate, List<capacity\_unit> \_capacity) | Поиск в исходном документе числовых значений |
| setCalculationDate(String calculationDate) | Функция считывания даты расчета |
| setExpirationDate (String expirationDate) | Функция считывания даты окончания |

## 1.6 Проектирование интерфейса пользователя

### 1.6.1 Построение графа диалога

Интерфейс пользователя – совокупность программных и аппаратных средств, которые способны обеспечивать комфортное взаимодействие пользователя с программным обеспечением, основой которого являются диалоги.

Диалог – процесс обмена некоторыми данными между пользователем и программой. Данный процесс выполняется в реальном времени и направлен на решение конкретной задачи. Любой диалог состоит из процессов ввода и вывода, физически обеспечивающих связь пользователя и программы.

Граф диалога – взвешенный граф, на каждой вершине которого находится картинка на экране или определенное состояние диалога, имеющее набор неких действий, доступных пользователю программного обеспечения. Дуги на графе отображают возможные изменения состояний при совершении пользователем указанных действий. В качестве весов дуг указывают условия переходов из состояния в состояние и операции, выполняемые во время перехода [20].

Разрабатываемая подсистема не требует разработки отдельного интерфейса, так как она будет внедряться в уже существующую систему кредитного потенциала, интерфейс которой был разработан сотрудниками АО «Сбербанк Технологии».

Граф диалога изображен на рис. 24. Описание вершин графа представлено в табл. 36. Описание дуг графа представлено в табл. 37.

Таблица 36

Описание вершин графа

|  |  |
| --- | --- |
| Вершина графа | Описание |
| 1 | 2 |
| Главное окно программы | Начальная экранная форма приложения «Сбербанк Онлайн» |
| Вкладка «Кредиты» | Экранная форма с набором элементов, позволяющих управлять кредитами |
| Окно «Кредитный потенциал» | Экранная форма с набором элементов, позволяющих пользователю вести работу со своим кредитным потенциалом |
| Заявка на кредит | Экранная форма с набором элементов, позволяющих создавать и редактировать текущие кредиты |

Продолжение табл. 36

|  |  |
| --- | --- |
| 1 | 2 |
| Окно «Информация» | Экранная форма, содержащая подробную информацию о системе кредитного потенциала |
| Заявка на кредит | Экранная форма с набором элементов, позволяющих заполнить анкету на получение кредита |
| Экран успеха | Экранная форма – уведомление об успешной подаче заявки на кредитование |
| Условия кредитования | Экранная форма, содержащая подробную информацию об актуальных условиях кредитования |
| Редактирование кредита | Экранная форма с набором элементов, позволяющих редактировать выбранный кредит |
| Заполнение анкетных данных | Экранная форма с набором элементов, позволяющих заполнить анкету, необходимую для расчета кредитного потенциала |
| Окно добавления кредита | Экранная форма с набором элементов, позволяющих добавить в систему информацию об уже существующем кредите |

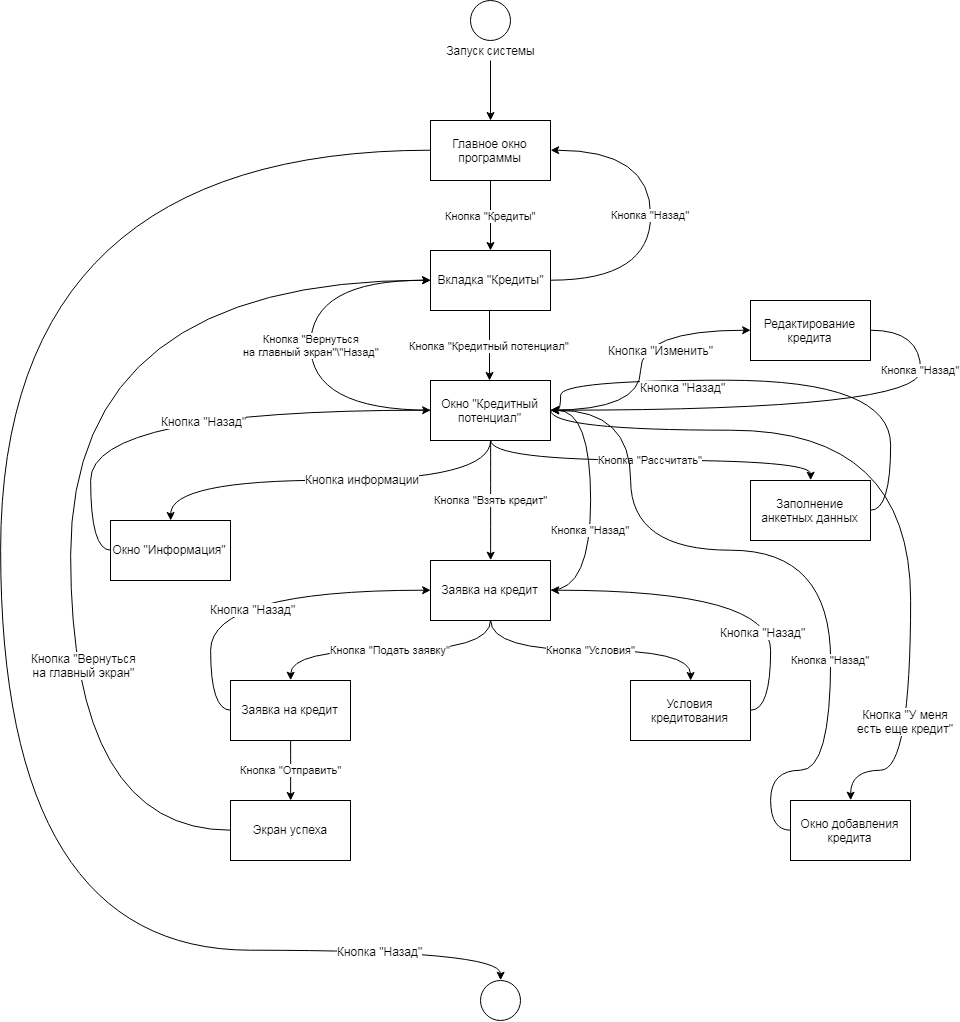


Рис. 24. Граф диалога

Таблица 37

Описание дуг графа

|  |  |
| --- | --- |
| Дуга графа | Описание |
| 1 | 2 |
| Запуск системы | При запуске сервиса «Сбербанк Онлайн» открывается Главное окно программы |

Продолжение табл. 37

|  |  |
| --- | --- |
| 1 | 2 |
| Кнопка «Кредиты» | При нажатии на кнопку «Кредиты» выполняется переход к экранной форме Вкладка «Кредиты» |
| Кнопка «Кредитный потенциал» | При нажатии на кнопку «Кредитный потенциал» выполняется переход к экранной форме Окно «Кредитный потенциал» |
| Кнопка «Назад» | При нажатии на кнопку «Назад» выполняется переход к предыдущей экранной форме либо выход из системы, если кнопка была нажата на экранной форме Главное окно программы |
| Кнопка «Взять кредит» | При нажатии на кнопку «Взять кредит» выполняется переход к экранной форме Заявка на кредит |
| Кнопка информации | При нажатии на кнопку информации выполняется переход к экранной форме Окно «Информация» |
| Кнопка «Вернуться на главный экран» | При нажатии на кнопку «Вернуться на главный экран» выполняется переход к экранной форме Вкладка «Кредиты» |
| Кнопка «Подать заявку» | При нажатии на кнопку «Подать заявку» выполняется переход к экранной форме Заявка на кредит |

Продолжение табл. 37

|  |  |
| --- | --- |
| 1 | 2 |
| Кнопка «Отправить» | При нажатии на кнопку «Отправить» выполняется переход к экранной форме Экран успеха |
| Кнопка «Условия» | При нажатии на кнопку «Условия» выполняется переход к экранной форме Условия кредитования |
| Кнопка «Изменить» | При нажатии на кнопку «Изменить» выполняется переход к экранной форме Редактирование кредита |
| Кнопка «Рассчитать» | При нажатии на кнопку «Рассчитать» выполняется переход к экранной форме Заполнение анкетных данных |
| Кнопка «У меня уже есть кредит» | При нажатии на кнопку «У меня уже есть кредит» выполняется переход к экранной форме Окно добавления кредита |

### 1.6.2 Разработка форм ввода-вывода информации

Разрабатываемая подсистема не требует разработки отдельного интерфейса, так как она будет внедряться в уже существующую систему кредитного потенциала, интерфейс которой был разработан сотрудниками АО «Сбербанк Технологии» [19]. Данный интерфейс является интерфейсом со свободной навигацией.

Интерфейс программы представлен на рис. 24-25.

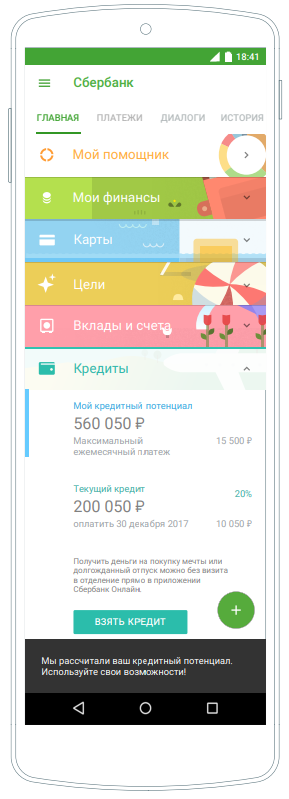


Рис. 24. Главное окно программы

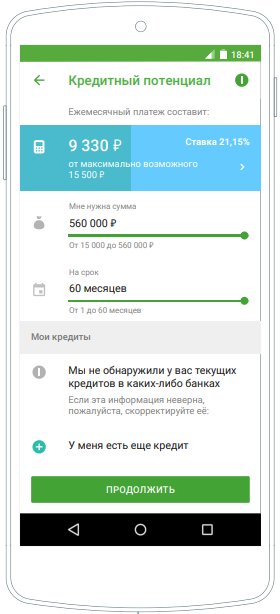


Рис. 25. Окно системы кредитного потенциала

## Выбор стратегии тестирования, разработка тестов, программа и методика испытаний

Тестирование программного обеспечения – процесс анализа программного обеспечения, позволяющий выявить ошибки и дефекты. Тестирование является плановой деятельностью, т.к. оно должно выполняться упорядоченно.

Цель тестирования – демонстрация корректной работы разработанного программного обеспечения и соответствие спецификациям.

### 1.7.1 Стандарты качества программного обеспечения

На данный момент, основным стандартом качества является стандарт ISO 9126 (ГОСТ Р ИСО / МЭК 9126-93).

ISO 9126 – международный стандарт, который определяет качественные характеристики программного обеспечения. Данный стандарт делится на 4 части, которые описывают следующие вопросы:

Модель качества;

Внешние метрики качества;

Внутренние метрики качества;

Метрики качества в использовании.

Первая часть стандарта (ISO 9126-1) отвечает за классификацию качества программного обеспечения по шести наборам характеристик:

1. Функциональность – пригодность для применения, корректность, способность к взаимодействию, защищенность;
2. Надежность – уровень завершенности, устойчивость к дефектам, восстанавливаемость, доступность, готовность;
3. Практичность – понятность, простота использования, изучаемость, привлекательность;
4. Эффективность – временная эффективность, используемость ресурсов;
5. Сопровождаемость – удобство для анализа, изменяемость, стабильность, тестируемость
6. Мобильность – адаптируемость, простота установки, сосуществование, замещаемость.

Вторая и третья части стандарта (ISO 9126-2 и ISO 9126-3) регламентируют формализацию внешних и внутренних метрик характеристик качества.

Четвертая часть стандарта (ISO 9126-4) отвечает за концепцию трех видов метрик и рекомендуемые виды измерений характеристик программных средств.

### 1.7.2 Методы тестирования

Существует несколько методов тестирования:

1. Тестирование методом «чёрного ящика»;
2. Тестирование методом «белого ящика»;
3. Тестирование методом «серого ящика»;
4. Тестирования не функциональных аспектов.

Тестирование методом «черного ящика» - тип тестирования, при котором выполняется проверка только функциональных возможностей программного обеспечения. Другие элементы, такие как дизайн или алгоритм, при данном типе тестирования, не рассматриваются. Тесты построены на основе требований и спецификаций приложения. Тестирование по стратегии «чёрного ящика» заключается в том, что выполняется выбор данных и соответствующим функциональным возможностям, а также тестирования функциональных возможностей в соответствии с функциональными спецификациями.

При тестировании методом «белого ящика» рассматривается внутренняя логика и структура кода программного обеспечения. Тесты, основанные на методе «белого ящика» включают в себя полную проверку исходного кода программы, а также внутренней логики кода. При реализации тестов, построенных на основе данного метода, тестолог имеет дело непосредственно с самим кодом программы.

Тестирование методом «серого ящика» включает в себя признаки методов «черного ящика» и «белого ящика» одновременно. При использовании данного метода тестирования, тестолог обладает лишь общими знаниями о программном обеспечении, необходимыми для проведения теста. Тесты, основанные на методе «серого ящика», осуществляется с помощью имеющейся технической документации.

Для тестирования не функциональных аспектов программного обеспечения, применяются следующие тесты:

1. Тестирование производительности;
2. Тестирование «Юзабилити»;
3. Тестирование безопасности;
4. Тестирование качества интернационализации.

### 1.7.4 Объект испытаний

Объектом испытаний является сконфигурированная подсистема конвертации.

### 1.7.5 Требования к информационному, аппаратно-программному обеспечению и документации

Подсистема кредитного потенциала пользователя мобильного банка должна состоять из:

1. Серверное приложение, содержащее весь необходимый функционал для получения, чтения и конвертации исходного документа;
2. База данных, хранящая информацию о кредитных продуктах и шкалах кредитного потенциала

1.7.5.1 Требования к функциональным характеристикам серверного приложения

Работа с серверным приложением будет осуществляться Единой Фронтальной Системой в тот момент, когда она будет получать запрос от пользователя мобильного банка.

Серверное приложение должно обладать следующим функционалом:

1. Возможность встраивания в уже существующую систему;
2. Возможность получения исходного файла извне;
3. Возможность сохранения готового файла на сервере.

1.7.5.2 Требования к функциональным характеристикам базы данных

Работа с базой данных будет осуществляться подсистемой конвертации и уполномоченными сотрудниками АО «Сбербанк Технологии».

Подсистеме конвертации будет доступна возможность чтения информации из базы данных, при необходимости сопоставления идентификационных номеров, указанных в получаемом документе и соответствующих им значений в базе данных.

Сотрудники будут иметь как возможность просмотра базы данных, так и возможность занесения новой информации в базу.

1.7.5.3 Требования к надежности системы

Для корректного функционирования программного обеспечения необходимо выполнить следующие требования:

1. Сервер должен быть подключён к энергосети через источник бесперебойного питания для предотвращения сбоев в работе программы;
2. В программном обеспечении должны быть предусмотрены возможные ошибки, связанные с некорректным типом или структурой получаемых файлов (документов).

### 1.7.6 Состав и порядок испытаний

Состав испытаний:

1. Тестирование корректности считывания данных из исходного файла;
2. Тестирование записи данных из файла в дерево объектов;
3. Тестирование конвертации объектов в новый файл;
4. Тестирование формирования структуры конечного файла;
5. Тестирование сохранения нового файла.

### 1.7.7 Результаты проведения испытаний

Результаты проведенных испытаний программного обеспечения представлены в табл. 38 – табл. 39.

Таблица 38

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Дата  тестирования | Тестируемый модуль или  функция | Кто  проводил  тестирование | Описание теста | Результаты  тестирования |
| 12.05.2018 | main(String[] args) | Разработчик | Корректное считывание данных из файла | Успех |
| 14.05.2018 | main(String[] args) | Разработчик | Запись данных из файла в | Успех |
| 15.05.2018 | generateJSON(GetClientRiskMetricsRs obj) | Разработчик | Конвертация объектов | Ошибка: нарушение структуры |
| 15.05.2018 | generateJSON(GetClientRiskMetricsRs obj) | Разработчик | Конвертация объектов | Успех |
| 15.05.2018 | generateJSON(GetClientRiskMetricsRs obj) | Разработчик | Формирование структуры | Успех |
| 18.05.2018 | generateJSON(GetClientRiskMetricsRs obj) | Разработчик | Сохранение файла | Успех |

Все тесты были проведены разработчиком. Выявленные в ходе тестирования ошибки исправлены.

# Часть 2. Технико-экономическое обоснование

## 2.1 Расчёт трудоёмкости разработки

* Предполагаемое число операторов q = 1200;
* коэффициент сложности разработки с = 1,2;
* коэффициент коррекции программы в ходе ее разработки р = 0,1;
* коэффициент квалификации разработчиков k = 0,5;
* коэффициент увеличения затрат труда вследствие недостатков описания задачи B = 1.

Трудоемкость разработки программного обеспечения решения задачи можно рассчитать по формуле:

|  |  |
| --- | --- |
|  | (1) |

где – затраты труда на изучение описания задачи; – затраты труда на разработку алгоритма решения задачи; – затраты труда на программирование по готовой блок-схеме; – затраты труда на отладку программы на ЭВМ; – затраты труда на подготовку документации.

Результат:

чел-час

Условное число операторов Q в программе определяется по формуле:

|  |  |
| --- | --- |
|  | (2) |

Затраты труда на изучение описания задачи (с учетом квалификации программиста):

|  |  |
| --- | --- |
|  | (3) |

чел-час

Затраты труда на разработку алгоритма решения задачи :

|  |  |
| --- | --- |
|  | (4) |

чел-час

Затраты труда на программирование по готовой блок-схеме:

|  |  |
| --- | --- |
|  | (5) |

чел-час

Затраты труда на отладку программы на ЭВМ при автономной отладке одной задачи:

|  |  |
| --- | --- |
|  | (6) |

чел-час

при комплексной отладке задачи:

|  |  |
| --- | --- |
|  | (7) |

чел-час

Затраты труда на подготовку документации по задаче:

|  |  |
| --- | --- |
|  | (8) |

чел-час

Затраты труда на подготовку материалов в рукописи:

|  |  |
| --- | --- |
|  | (9) |

чел-час

Затраты труда на редактирование, печать и оформление документации:

|  |  |
| --- | --- |
|  | (10) |

чел-час

### 2.2 Расчет себестоимости разработки

В себестоимость разработки входят: основная заработная плата разработчиков, дополнительная заработная плата, единый социальный налог, затраты на использование машинного времени, затраты на носители информации, затраты на текущий и профилактический ремонт вычислительной техники, прочие эксплуатационные расходы.

Себестоимость разработки рассчитывается по формуле:

|  |  |
| --- | --- |
| , | (11) |

где – себестоимость программного продукта, руб.; – затраты на оплату труда, руб.; – затраты на использование машинного времени, руб.; – затраты на носители информации, руб.; – затраты на текущий и профилактический ремонт вычислительной техники, руб.; – прочие эксплуатационные расходы, руб.

Результат:

руб

Состав команда проекта: программист-дипломник, руководитель ВКР.

Затраты на оплату труда при разработке программного продукта вычисляются по формуле:

|  |  |
| --- | --- |
| , | (12) |

где – общая зарплата работника за час; – отчисления с зарплаты, %; – время написания программы. Время написания программы совпадает с временем работы компьютера.

Результат:

руб.

руб.

Заработная плата программиста за час определяется по следующей формуле:

|  |  |
| --- | --- |
| , | (13) |

где – ставка программиста; – фонд рабочего времени в месяц, ч.

Результат, с учётом :

руб.

руб.

Заработная плата дополнительная определяется по следующей формуле:

|  |  |
| --- | --- |
| , | (14) |

где – заработная плата; – норма отчислений на дополнительную зарплату (10 %).

Результат:

*руб.*

*руб.*

Зарплата общая вычисляется по следующей формуле:

|  |  |
| --- | --- |
|  | (15) |

Результат:

руб.

руб.

Отчисления на соцстрах, фонд занятости и пенсионный фонд вычисляются по следующей формуле:

|  |  |
| --- | --- |
| , | (15) |

где – отчисления на соцстрах (0,5 % от Зобщ); – отчисления в фонд занятости (0,5 % от Зобщ); – отчисления в Пенсионныйфонд (2 % от ).

Результат:

руб.

руб.

Затраты на использование машинного времени вычисляются по формуле:

|  |  |
| --- | --- |
| , | (16) |

где – затраты на использование машинного времени, руб.; – стоимость одного часа машинного времени, руб./ч; – время использования вычислительной техники, ч.

Результат:

руб.

Таблица 40

Данные по заработной плате

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Должность  разработчика | Разряд | Время работы, мес. | Стпр,  руб. | Зпр,  руб. | Здоп,  руб. | Зобщ,  руб. | Отч,  руб. | Зтр, руб. |
| Программист-дипломник | 1 | 96 | 30000 | 312,5 | 97,7 | 410,2 | 49,2 |  |
| Научный руководитель ВКР | 5 | 96 | 19400 | 202,1 | 40,8 | 242,9 | 29,2 |  |

Стоимость одного часа машинного времени рассчитывается по формуле:

|  |  |
| --- | --- |
| , | (17) |

где – стоимость одного часа машинного времени, руб./ч; – покупная цена компьютера, руб.; – срок службы компьютера, год; – количество рабочих дней в году; – время работы компьютера в течение суток, ч; – стоимость одного кВт · ч электроэнергии, руб.; – мощность вычислительной системы, кВт.

Результат:

руб/ч

Время использования вычислительной техники () рассчитывается по следующей формуле:

|  |  |
| --- | --- |
| , | (18) |

где – количество дней разработки ПО; – время работы компьютера в течение суток, ч.

Результат:

ч.

Затраты на носители информации принимаются в размере 2 % от цены вычислительной техники.

Затраты на текущий и профилактический ремонт принимаются в размере 4 % от цены вычислительной техники. Прочие эксплуатационные расходы включают в себя затраты на освещение, отопление, охрану, уборку и текущий ремонт помещений. Они принимаются в размере 10 % от стоимости помещения (или его аренды), где происходит разработка программного продукта.

### 2.3 Расчет цены разработки

Определение цены является одной из труднейших задач, стоящих перед любым предприятием. И именно цена предопределяет успехи предприятия – объемы продаж, доходы, получаемую прибыль.

Установление определенной цены на программный продукт служит для последующей его продажи и получения прибыли. Очень важно назначить цену таким образом, чтобы она не оказалась слишком высокой или слишком низкой.

Для определения минимальной цены, ниже которой разработчику будет невыгодно продавать программный продукт, используется следующая формула:

|  |  |
| --- | --- |
| , | (19) |

где – цена программного продукта, руб.; – себестоимость программного продукта, руб.; – норматив прибыли (20 %, в формуле = 0,2).

Результат:

руб.

### 2.4 Расчёт экономической эффективности

В расчет экономической эффективности входит определение эксплуатационных расходов и капитальных затрат потребителя, годовой экономии эксплуатационных расходов у одного потребителя, срока окупаемости программного продукта, годового экономического эффекта.

Расходы потребителя, связанные с эксплуатацией программы, определяются по следующей формуле:

|  |  |
| --- | --- |
| , | (20) |

где – эксплуатационные расходы потребителя, руб.; – объем машинного времени в течение года, необходимый для решения данной задачи с использованием программы, ч; – стоимость одного часа машинного времени, руб./ч; – цена программного продукта, руб.; – срок службы программного продукта, год. Обычно составляет 1 – 2 года, затем выпускается новая версия программного продукта.

час.

Результат:

руб.

Если на момент внедрения программного продукта у потребителя все работы выполнялись вручную, тогда капитальные затраты рассчитываются по формуле:

|  |  |
| --- | --- |
| , | (21) |

где – капитальные расходы потребителя, руб.; – объем машинного времени в течение года, необходимый для решения данной задачи с использованием программы, ч; – полезный годовой фонд времени работы вычислительной техники, принимается условно 2000 ч в год; – капитальные затраты на вычислительную технику, для которой предназначена программа, руб.

Результат:

руб.

Капитальные затраты на вычислительную технику рассчитываются по формуле:

|  |  |
| --- | --- |
| , | (22) |

где – цена вычислительной техники, руб.; – прочие расходы потребителя, связанные с помещением (отопление, освещение, уборка и т.д.), принимаются в размере 10 % от стоимости помещения потребителя (или его аренды), руб.

Результат:

руб.

Для расчета годовой экономии эксплуатационных расходов потребителя вычисляются эксплуатационные затраты потребителя при решении задачи вручную:

|  |  |
| --- | --- |
| , | (23) |

где – эксплуатационные расходы потребителя при решении задачи вручную, руб.; – фонд заработной платы персонала, обслуживающего решение задачи вручную, руб.; 12 – количество месяцев в году; 1,21 – поправочный коэффициент.

Результат:

руб.

Тогда годовая экономия эксплуатационных расходов у одного потребителя рассчитывается по формуле:

|  |  |
| --- | --- |
|  | (24) |

Результат:

руб.

Срок окупаемости программного продукта рассчитывается по формуле:

|  |  |
| --- | --- |
|  | (25) |

Результат:

года

Годовой экономический эффект, получаемый одним пот­ре­бителем, рассчитывается по формуле:

|  |  |
| --- | --- |
| , | (25) |

где – нормативный коэффициент эффективности дополнительных капитальных вложений, равный 0,15.

руб/год

# Заключение

В результате выпускной квалификационной работы были выполнены следующие задачи:

1. Изучен процесс расчет кредитного потенциала;
2. Изучены особенности работы приложений и библиотек, написанных на языке Java.
3. Сконфигурирована оптимальная система для автоматизации процесса конвертации;
4. Разработана удалённая база данных.

В результате была сконфигурирована подсистема конвертации. С помощью данной информационной системы можно осуществить интеграцию новых модулей в уже существующую систему.

Кроме того в ходе выполнения выпускной квалификационной работы были освоены следующие компетенции: Способность проводить предпроектное обследование объекта проектирования, системный анализ предметной области, их взаимосвязей (ПК-1); Способность проводить техническое проектирование (ПК-2); Способность проводить рабочее проектирование (ПК-3); Способность проводить выбор исходных данных для проектирования (ПК-4); Способность проводить моделирование процессов и систем (ПК-5); Способность оценивать надежность и качество функционирования объекта проектирования (ПК-6); Способность проводить расчет экономической эффективности (ПК-9); Способность разрабатывать, согласовывать и выпускать все виды проектной документации (ПК-10); Способность к проектированию базовых и прикладных информационных технологий (ПК-11); Способность разрабатывать средства реализации информационных технологий (методические, информационные, математические, алгоритмические, технические и программные) (ПК-12); Готовность участвовать в постановке и проведении экспериментальных исследований (ПК-23); Способность обосновывать правильность выбранной модели, сопоставляя результаты экспериментальных данных и полученных решений (ПК-24); Способность оформлять полученные рабочие результаты в виде презентаций, научно-технических отчетов, статей и докладов на научно-технических конференциях (ПК-26).

Список литературы

1. Буч Г. Язык UML. Руководство пользователя – М.: ДМК-Пресс, 2007. – 496 с.;
2. Вендров А.М. CASE-технологии. Современные методы и средства проектирования информационных систем – М.: Финансы и статистика, 2009 г. – 758 с.;
3. Вендров, А.М. Проектирование программного обеспечения экономических информационных систем – М.: Финансы и статистика, 2006. - 544 с.;
4. Грекул В.И. Проектирование информационных систем – СПб.: Бином, 2008. – 304 с.;
5. Давыдов С.В. IntelliJ IDEA. Профессиональное программирование на Java. Наиболее полное руководство – СПб.: БХВ-Петербург, 2005. – 800 с.;
6. Лешек А. Анализ и проектирование информационных систем с помощью UML – М.: Вильямс, 2008. - 816 с.;
7. Мейер Б. Объектно-ориентированное конструирование программных систем – СПб.: Русская Редакция, 2005. – 1204 с.;
8. Пирогов В.Ю. Информационные системы и базы данных. Организация и проектирование – СПб.: БХВ-Петербург, 2009. – 528 с.;
9. Пихлер Р. Управление продуктом в Scrum. Agile-методы для вашего бизнеса – М.: Манн, Иванов и Фербер, 2017. – 240 с.;
10. Сиерра К. Изучаем Java – М.: Эксмо, 2017. – 720 с.;
11. Тавасиев А.М. Банковское кредитование – М.: ИНФРА-М, 2010. - 656 с.;
12. Туманов В.Е. Основы проектирования реляционных баз данных – М.; Лаборатория знаний, 2014. – 424 с.;
13. Хабибуллин И. Ш. Самоучитель Java (3-е изд.) – СПб.: БХВ-Петербург, 2008. – 758 с.;
14. Шилдт Г. Java 8. Руководство для начинающих - М.: Вильямс, 2015. - 720 c.;
15. Scrum [Электронный ресурс] – Электрон. текстовые дан. – Режим доступа: <https://ru.atlassian.com/agile/scrum>, свободный;
16. Диаграмма вариантов использования [Электронный ресурс] – Электрон. текстовые дан. – Режим доступа: https://habr.com/post/47940/, свободный;
17. Диаграмма последовательностей [Электронный ресурс] – Электрон. текстовые дан. – Режим доступа: https://msdn.microsoft.com/ru-ru/library/dd409377, свободный;
18. Контекстная диаграмма классов [Электронный ресурс] – Электрон. текстовые дан. – Режим доступа: http://deryabych.narod.ru/5/5, свободный;
19. Сбербанк Технологии [Электронный ресурс] – Электрон. текстовые дан. – Режим доступа: https://vk.com/sbertech, свободный;
20. Граф диалога [Электронный ресурс] – Электрон. текстовые дан. – Режим доступа: https://studfiles.net/preview/4166884/page:12/, свободный.

# Приложение 1

МИНОБРНАУКИ РОССИИ

федеральное государственное бюджетное образовательное учреждение

высшего образования

«ЧЕРЕПОВЕЦКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Институт информационных технологий

Кафедра математического и программного обеспечения ЭВМ

|  |
| --- |
| УТВЕРЖДАЮ |
| Зав. кафедрой МПО ЭВМ |
| д.т.н., профессор \_\_\_\_\_\_\_\_ Ершов Е.В. |
| «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2018 г. |

РАЗРАБОТКА ПОДСИСТЕМЫ КРЕДИТНОГО ПОТЕНЦИАЛА

ПОЛЬЗОВАТЕЛЯ МОБИЛЬНОГО БАНКА

Техническое задание на выпускную квалификационную работу

Листов \_\_\_

|  |  |  |
| --- | --- | --- |
|  | Руководитель: | доцент Юдина О.В. |
|  | Исполнитель: | студент гр. 1ИСб-00-41оп |
|  |  | Смирнов Д.С.. |

2018 г.

Введение

Программное обеспечение – подсистема кредитного потенциала пользователя мобильного банка, должна выполнять конвертацию исходных документов формата XML, содержащих структурированную информацию о кредитных продуктах и шкалах кредитного потенциала, в формат JSON.

1. Основания разработки и область применения

Основанием для разработки служит задание на выпускную квалификационную работу, выданное в обособленном подразделении АО «Сбербанк-Технологии» в г. Череповце.

Дата утверждения: 12 февраля 2018 года.

Наименование темы разработки: разработка подсистемы кредитного потенциала пользователя мобильного банка.

1. Назначение разработки

Программа должна по запросу осуществлять конвертацию получаемого исходного файла в формате XML в файл JSON, после чего выполнять автоматическое сохранение конечного файла.

Для большей точности конвертации и сохранения начальной структуры документа, программой осуществляется считывание исходного файла построчно и последующая запись считанных строк в объекты.

Данная разработка предназначена для наиболее удобной и корректной интеграции системы кредитного потенциала в уже существующую Единую Фронтальную Систему (ЕФС).

Также разрабатываемое программное обеспечение предоставляет возможность интеграции сторонних разработок в будущем.

1. Требования к программе

Подсистема кредитного потенциала пользователя мобильного банка должна состоять из:

1. Серверное приложение, содержащее весь необходимый функционал для получения, чтения и конвертации исходного документа;
2. База данных, хранящая информацию о кредитных продуктах и шкалах кредитного потенциала

3.1 Требования к функциональным характеристикам серверного приложения

Работа с серверным приложением будет осуществляться Единой Фронтальной Системой в тот момент, когда она будет получать запрос от пользователя мобильного банка.

Серверное приложение должно обладать следующим функционалом:

1. Возможность встраивания в уже существующую систему;
2. Возможность получения исходного файла извне;
3. Возможность сохранения готового файла на сервере.

3.2 Требования к функциональным характеристикам базы данных

Работа с базой данных будет осуществляться подсистемой конвертации и уполномоченными сотрудниками АО «Сбербанк Технологии».

Подсистеме конвертации будет доступна возможность чтения информации из базы данных, при необходимости сопоставления идентификационных номеров, указанных в получаемом документе и соответствующих им значений в базе данных.

Сотрудники будут иметь как возможность просмотра базы данных, так и возможность занесения новой информации в базу.

3.3 Требования к надежности системы

Для корректного функционирования программного обеспечения необходимо выполнить следующие требования:

1. Сервер должен быть подключён к энергосети через источник бесперебойного питания для предотвращения сбоев в работе программы;
2. В программном обеспечении должны быть предусмотрены возможные ошибки, связанные с некорректным типом или структурой получаемых файлов (документов);

3.4 Условия эксплуатации

Климатические условия эксплуатации, при которых должны обеспечиваться заданные характеристики, должны удовлетворять требованиям, предъявляемым к техническим средствам в части условий их эксплуатации.

3.5 Требования к составу и параметрам технических средств

Для эксплуатации серверного ПО необходимо иметь сервер со следующими минимальными характеристиками:

* 1. Операционная система Windows Server 2016;
  2. Процессор Intel Xenon E3 1200;
  3. Свободное место на жёстком диске 100 Мбайт;
  4. Объём оперативной памяти 4 Гбайта;

База данных также будет располагаться на сервере

3.6 Требования к информационной и программной совместимости

3.6.1 Требования к исходным кодам и языкам программирования

Серверное приложение должно быть разработано с учётом следующих рекомендаций:

1. Код программы реализован на языке высоко уровня JAVA с применением объектно-ориентированного подхода;
2. Интерфейс ПО выполнен в консольном виде.

3.6.2 Требования к базе данных

Разрабатываемая база данных должна быть полностью совместима с системой управления базами данных (СУБД) Oracle 12c.

3.6.3 Требования к упаковке и маркировке

Диск с программным обеспечением должен быть упакован в бумажный конверт, прикреплённый к папке с документацией. Диск и его упаковка должны содержать информацию о названии программы и ФИО разработчика программы.

3.6.4 Требования к хранению

Курсовой проект необходимо хранить в сухом помещении. Избегать попадания прямых солнечных лучше, а также контакта с источниками воды и огня.

4. Требования к программной документации

4.1 Структурные требования к расчётно-пояснительной записке

Программная документация содержит расчетно-пояснительную записку, руководство пользователя, наборы тестовых данных и результатов тестирования, спецификации, текст программы.

4.2 Требования к оформлению программной документации

При оформлении текстовых и графических материалов, следует придерживаться общих требований ЕСПД.

Текстовые документы оформляются на белых листах формата А4 (210х297 мм). В соответствии с общими требованиями поля листа определяются следующим образом: левое – 30 мм, правое – 10 мм, верхнее – 20 мм, нижнее – 20 мм. Формат текста: Word for Windows, через один интервала, шрифт – Times New Roman Cyr, размер шрифта – 14, отступ первой строки абзаца – 0,75 см, выравнивание – по ширине. Количество знаков в строке, считая пробелы, не должно превышать 60 символов. Текст программы может быть расположен в две колонки, шрифт – Times New Roman Cyr, размер шрифта – 8.

Номер страницы проставляется в середине верхнего поля страницы арабской цифрой. Первая страница – титульный лист, вторая страница – аннотация, с третьей страницы начинается оглавление. Номер страницы на титульном листе не проставляется.

Наименование разделов, подразделов, пунктов должно быть кратким и соответствовать содержанию. Каждая новая глава печатается с новой страницы. Наименование разделов (основных частей) пишется прописными буквами по центру строки. Расстояние между заголовками и текстом, а также между заголовками разделов и подразделов должно быть равно двум интервалам.

Наименования подразделов и пунктов размещаются с абзацного отступа (0,75 см) и печатаются с прописной буквы, без подчеркивания и без точки в конце.

При использовании ссылок на пункты, разделы и подразделы указывается порядковый номер раздела или пункта (например: «в разд. 2», «в п. 2.3.1»).

Рисунки, таблицы и формулы нумеруются арабскими цифрами последовательно, так называемая сквозная нумерация, или в пределах раздела (относительная нумерация). В приложении нумеруются в пределах приложения. Каждый рисунок имеет порядковый номер и название, помещаемые под рисунком по центру.

Таблицы в основном применяются для оформления цифрового материала. Шрифт – Times New Roman Cyr, размер шрифта – 14. Иногда возможен 12-й размер шрифта. Обозначения единиц физических величин необходимо применять в системе СИ. Номер таблицы размещается в правом верхнем углу над заголовком.

5. Стадии и этапы разработки

Стадии и этапы разработки представлены в табл. П1.1

Таблица П1.1

Стадии и этапы разработки

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Номер этапа разработки | Наименование этапа  разработки ПО | Сроки  Разработки | Результат  выполнения | Отметка о  выполнении |
| 1 | 2 | 3 | 4 | 5 |
| 1 | Разработка технического задания | 14.02.2018 - 120.02.2018 | Документ «Техническое задание» |  |
| 2 | Проведение сравнительного анализа отечественных и зарубежных аналогов проектируемой системы | 21.02.2018 – 10.03.2018 | Сравнительный анализ отечественных и зарубежных аналогов проектируемой системы в РПЗ |  |

Продолжение табл. П1.1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |
| 3 | Выбор технологии, среды и языка программирования | 11.03.2018 – 15.03.2018 | Выбор технологии, среды и языка программирования в РПЗ |  |
| 4 | Анализ процесса обработки информации, выбор структур данных для ее хранения, выбор методов и алгоритмов решения задачи | 16.03.2018 – 25.03.2018 | Анализ процесса обработки информации, выбор структур данных для ее хранения, выбор методов и алгоритмов решения задачи в РПЗ |  |
| 5 | Разработка спецификаций проектируемой системы | 26.03.2018 – 10.04.2018 | Разработка спецификаций проектируемой системы в РПЗ |  |
| 6 | Проектирование системы | 11.04.2018 – 24.04.2018 | Проектирование системы в РПЗ |  |
| 7 | Проектирование интерфейса пользователя для серверного приложения | 25.04.2018 – 30.04.2018 | Проектирование интерфейса пользователя в РПЗ |  |
| 8 | Разработка серверного приложения | 04.05.2018 – 21.05.2018 | Работоспособное серверное приложение |  |
| 9 | Выбор стратегии тестирования, разработка тестов, прохождение тестов | 05.06.2018 | Выбор технологии, среды и языка программирования в РПЗ |  |
| 10 | Написание руководство пользователя | 15.06.2018 | Приложение 4. Руководство пользователя |  |

6. Порядок контроля и приемки

Порядок контроля и приёма работ представлен в табл. П1.2.

Таблица П1.2

Порядок контроля и приемки

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование  контрольного этапа  выполнения  курсовой работы | Сроки  контроля | Результат выполнения | Отметка о приемке  результата  контрольного этапа |
| 1 | 2 | 3 | 4 |
| Выдача задания | 12.02.18 | Получено задание на выполнение курсового проекта |  |
| Техническое задание | 14.03.18 | Документ «Техническое задание» |  |
| Расчётно-пояснительная записка | 08.06.18 | Документ «Расчётно-пояснительная записка» |  |

Продолжение табл. П1.2

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| Приложения к расчётно-пояснительной записке | 15.06.2018 | Документы «Приложение 2», «Приложение 3», «Приложение 4», «Приложение 5», «Приложение 6» |  |
| Защита курсовой работы | 21.06.18 | Защита курсовой работы |  |