Архитектурный документ

Банковская информационная система для поддержки процесса выдачи и обслуживания кредита

Команда 24

# Раздел регистрации изменений

|  |  |  |  |
| --- | --- | --- | --- |
| **Версия документа** | **Дата** | **Описание изменения** | **Автор** |
| v1.0 | 01.06.2020 | Первоначальная версия | Гринкевич Т.А. |
| v1.1 | 02.06.2020 | Изменен раздел “Представления архитектуры” | Рябичев А.М. |

# Введение

## Название проекта

Банковская информационная система для поддержки процесса выдачи и обслуживания кредита.

## Рамки проекта (Scope)

Проект включает в себя разработку информационной системы банка для поддержки процесса выдачи и обслуживания кредита.

# Общее описание архитектуры, задействованные архитектурные представления

Документ включает в себя:

* представление прецедентов - стр.7-8
* логическое представление архитектуры - стр.9-10
* представление архитектуры процессов - стр.11-12
* представление развертывания - стр.13
* представление архитектуры данных - стр.13
* представление разработки - стр.14
* представление производительности - стр.14

В документ не было включено физическое представление архитектуры и представление безопасности из-за отсутствия соответствующих требований и предположений.

# Архитектурные факторы (цели и ограничения)

|  |  |  |
| --- | --- | --- |
| **Роль** | **Представляет интересы** | **Обязанности в рамках проекта** |
| Программист | Исполнитель | Разработка информационной системы |
| Тестировщик | Исполнитель | Тестирование и откладка |
| Менеджер проекта | Исполнитель | Контроль процесса разработки продукта на всех этапах |
| Представитель заказчика | Обычный пользователь (клиент и сотрудник), системный администратор | Разработка требований |

# Технические описания архитектурных решений

## Техническое описание №1

### Проблема

Как должен быть обеспечен доступ к списку встреч в системе?

### Идея решения

Следует использовать стандартные средства JDBC

### Факторы

Требования ТЗ позволяют использовать стандартные средства, так как не требуется особенной производительности

Информация не будет утеряна при корректной реализации

### Решение

Вся информация о встречах, расписании встреч и временных интервалах будет размещена в таблицах в базе данных.

### Мотивировка

Такое решение является самым удобным и простым в реализации.

### Неразрешенные вопросы

Отсутствуют.

## Техническое описание №2

### Проблема

Как должен быть обеспечена передача данных в системы для скоринга и систему нотификаций?

### Идея решения

Использовать библиотеки Apache HttpClient.

### Факторы

Требования ТЗ позволяют использовать стандартные средства, так как не требуется особенной производительности.

Информация не будет утеряна при корректной реализации.

### Решение

Пример: Подробное описание решения, как оно будет реализовано. Возможно, ссылки на другие части архитектурного документа

### Мотивировка

* Реализация всех HTTP методов (GET, POST, PUT, DELETE, HEAD, OPTIONS, and TRACE).
* Поддерживает шифрование HTTPS (HTTP over SSL) протокола.
* Тунелированное HTTPS соединение через HTTP прокси, при помощи метод CONNECT.
* Соединение через SOCKS прокси (версий 4 & 5) используя встроенную поддержку Java socket.
* Аутентификация через Basic, Digest и методы шифрования NTLM (NT Lan Manager).
* Встроенный механизм кастомной аутентификации
* Pluggable secure sockets implementations, making it easier to use third party solutions
* Оптимизирована под мульти-тред приложения. Поддерживает настройку максимального количества соединений. Находит и закрывает устаревшие соединения.
* Прямой доступ к коду ответа и заголовкам, отправленным сервером.
* Возможность установить время ожидания соединения.
* HttpMethods определяют шаблон команд для обеспечения параллельных запросов и эффективного повторного использования соединений.

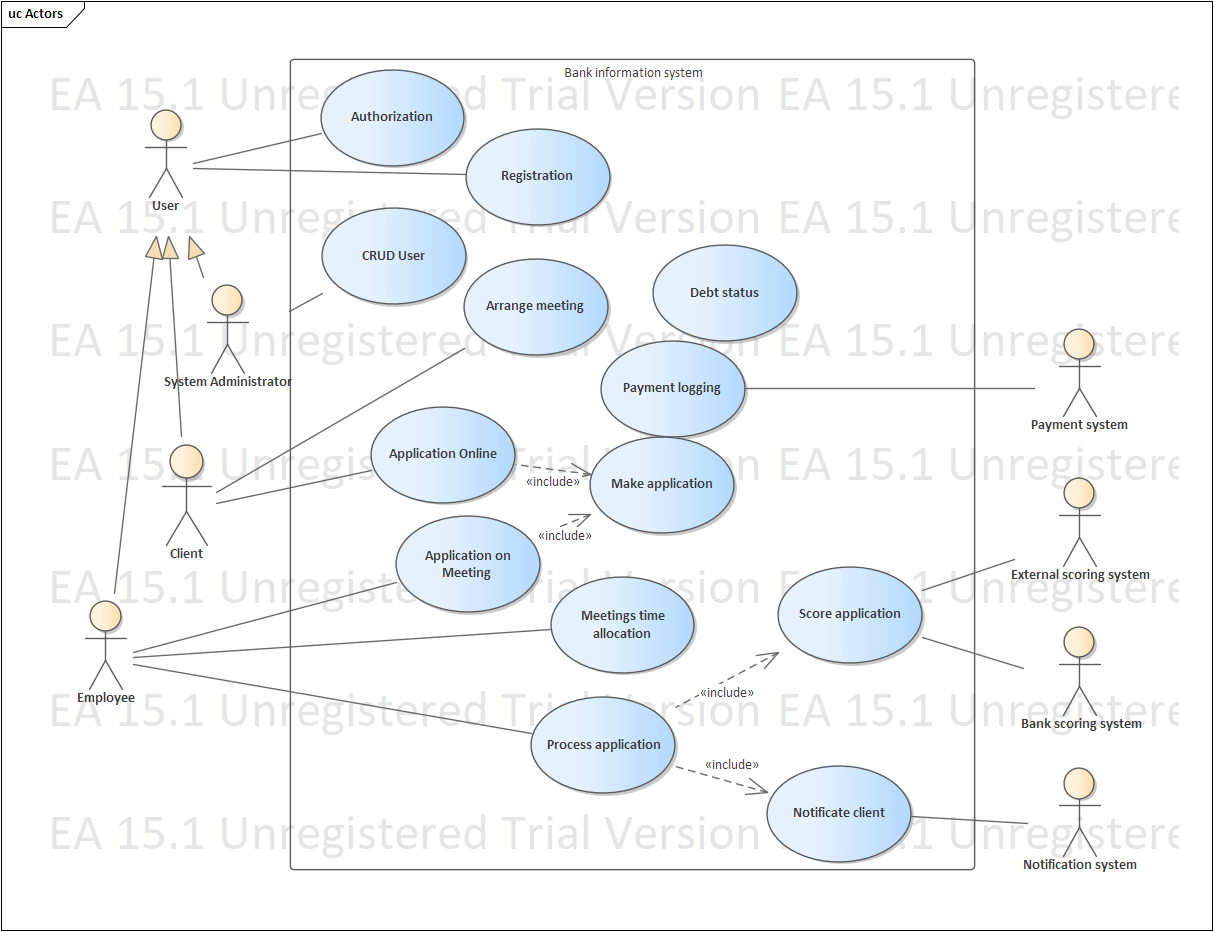
### Неразрешенные вопросы

Отсутствуют.

# Представления архитектуры

В данном разделе подробно описывается каждое из используемых в проекте архитектурных представлений.

## Представление прецедентов (сценариев использования)



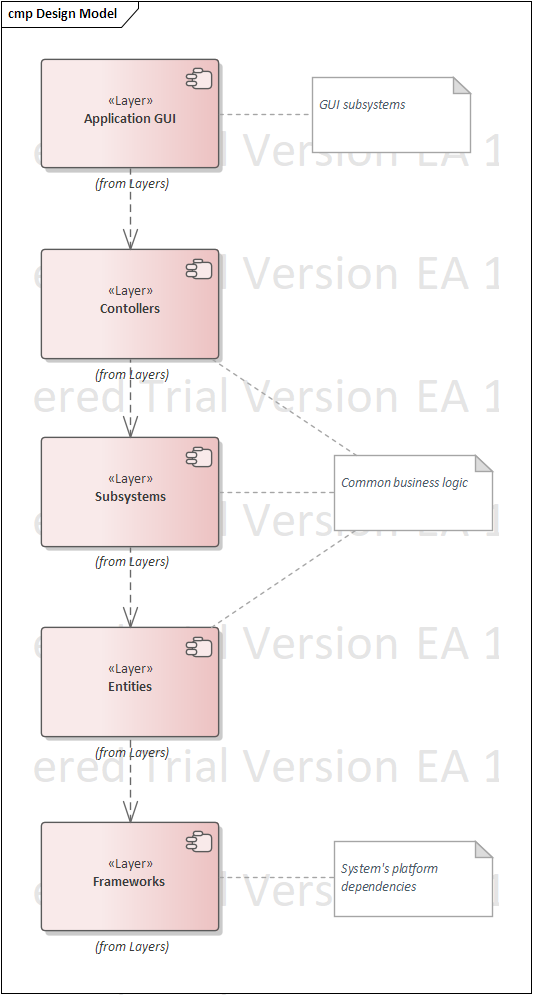
**Основные actors:**

* “Bank scoring system”
* “Client”
* “Employee”
* “External scoring system”
* “Notification system”
* “Payment system”
* System Administrator”
* “User”

**Описание прецедентов:**

* “Authorization” - вход зарегистрированного актора “User” в систему.
* “Registration” - создание актором “User” профиля в системе.
* “CRUD User” - действие актора “System Administrator” по изменению хранящихся в системе профилей.
* “Arrange meeting” - выбор актором “Client” времени для личной встречи для подачи заявления на кредит.
* “Debt status” - функционал просмотра состояния задолженности.
* “Payment logging” - получение от вторичного актора “Payment System” информации о поступлении платежей по выданным кредитам.
* “Application Online” - действие актора “Client” по составлению заявки на получение кредита через сайт банка.
* “Make application” - функционал системы по составлению заявки на получение кредита из имеющихся данных.
* “Application on Meeting” - действие актора “Employee” по составлению заявки на получение кредита на личной встречи сотрудника с клиентом.
* “Meeting time allocation” - действие актора “Employee” по распределению своего рабочего времени на личные встречи с клиентами.
* “Score application” - получение PD по от “External scoring system” и “Bank scoring system”.
* “Process application” - действие актора “Employee” по подтверждению или отклонению заявки на получение кредита.
* “Notificate client” - уведомление клиента с помощью “Notification system” о принятом решение о кредите всеми доступными способами.

## Логическое представление архитектуры



**вот эти слои я хз это какие-то определения из учебника мне кажется**

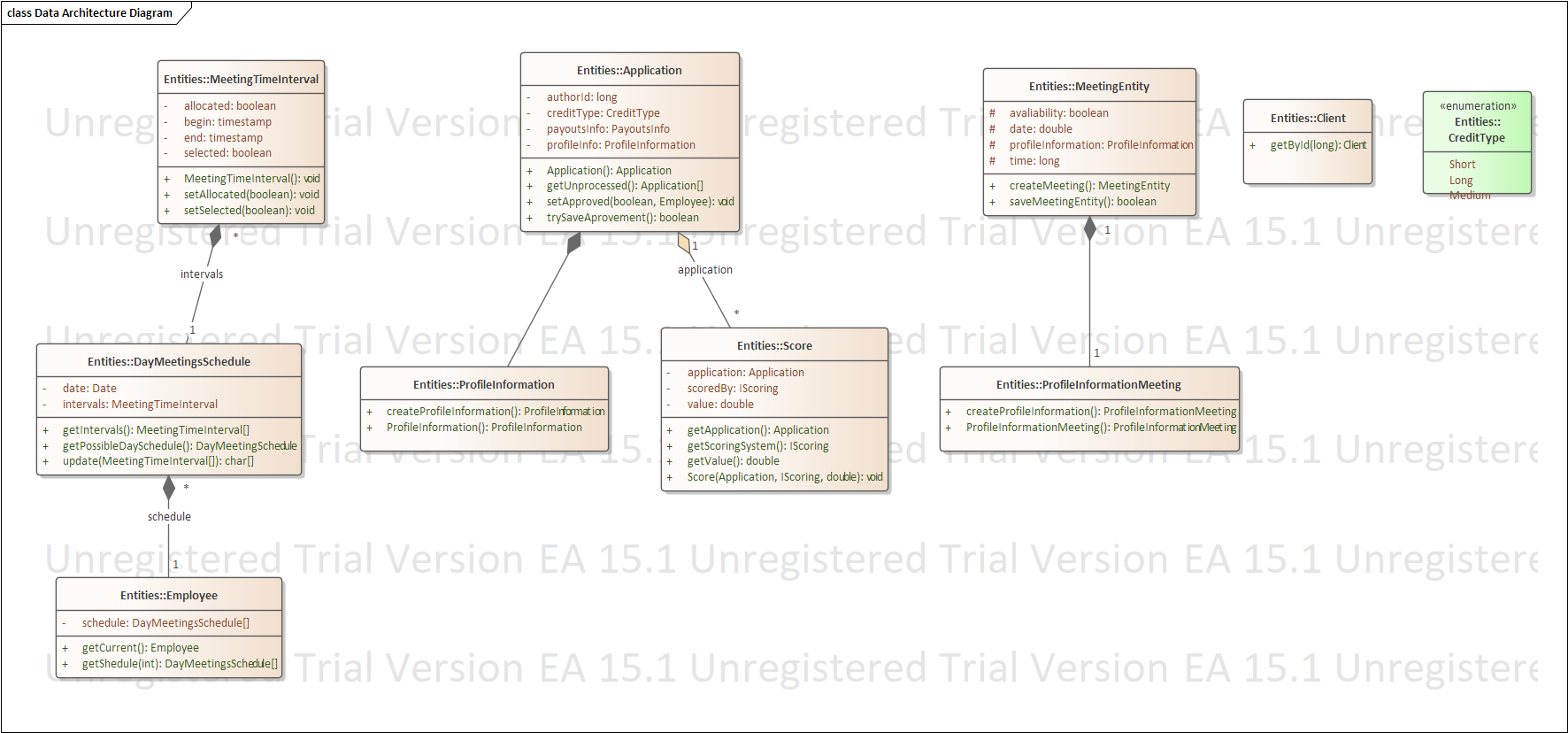
**Описание слоев:**

* “Application GUI” - слой содержит подсистемы, отвечающие за взаимодействие с пользователями через графический интерфейс.
* “Controllers” - слой содержит подсистемы, управляющие взаимодействием и обработкой данных; инкапсулирует бизнес-логику.
* “Subsystems” - слой содержит подсистемы, отвечающие за взаимодействие с внешними системами.
* “Entities” - слой содержит пакеты со специфическими для системы типами данных.
* “Frameworks” - слой содержит использованные в системе сторонние библиотеки.

## Представление архитектуры процессов

## Представление развертывания

## Представление архитектуры данных



## Представление разработки

Проект с исходным кодом разделен на слои, подсистемы и папки в соответствии с документом. Для разработки используется среда разработки IntelliJ IDEA. Для контроля версий используется Git. Тестирование производится с помощью JUnit тестов.

## Представление производительности

Для обеспечения лучшей производительности в проекте используются алгоритмы сортировки и поиска. Для хранения и обработки данных используются SQL запросы.

# Приложения

## Словарь терминов

|  |  |
| --- | --- |
| **Термин** | **Значение** |
| ПО | Программное обеспечение |
| Скоринговая система | Система, осуществляющая кредитный скоринг |
| PD­­­­ | Вероятность неосуществления клиентом своих долговых обязательств |
| Вердикт по кредиту | Решение по одобрению или отказу запрошенного клиентом кредитного продукта |
| Статус кредита | Кредит считается закрытым, если клиент выполнил свои долговые обязательства по кредиту, и открытым в ином случае. |