Студенты ФИТ НГУ

группа

Версия 1.0.1

Содержание

[1. Введение 3](#_Toc50299501)

[1.1 Цель 3](#_Toc50299502)

[1.2 Область действия 3](#_Toc50299503)

[1.3 Определения и сокращения 3](#_Toc50299504)

[1.4 Ссылки 3](#_Toc50299505)

[1.5 Краткое описание 3](#_Toc50299506)

[2. Предметная область проекта 4](#_Toc50299507)

[2.1 Существующие проблемы 4](#_Toc50299508)

[2.2 Предполагаемое решение 4](#_Toc50299509)

[3. Требования к программному решению 4](#_Toc50299510)

[3.1 Роли 4](#_Toc50299511)

[3.2 Функциональные требования для роли Роль1 4](#_Toc50299512)

[3.2.1 <Use Case Name 1> 4](#_Toc50299513)

[3.2.2 <Use Case Name 2> 4](#_Toc50299514)

[3.3 Функциональные требования для роли Роль2 4](#_Toc50299515)

[3.3.1 <Use Case Name 1> 4](#_Toc50299516)

[3.3.2 <Use Case Name 2> 4](#_Toc50299517)

[3.4 Нефункциональные требования 4](#_Toc50299518)

[4. Обзор архитектуры 5](#_Toc50299519)

[4.1.1 Компонентная модель системы 5](#_Toc50299520)

[4.1.1.1 Компонент 1 5](#_Toc50299521)

[4.1.1.2 Компонент 2 5](#_Toc50299522)

[4.1.2 Компоненты сторонних производителей 5](#_Toc50299523)

[4.1.3 Схема развертывания приложения 5](#_Toc50299524)

[5. Допущения и ограничения 5](#_Toc50299525)

[6. Известные проблемы 6](#_Toc50299526)

[6.1 Невысокая производительность приложения 6](#_Toc50299527)

# Введение

## Цель

Данный документ представляет собой техническое описание проекта *Депозитный и Кредитный Калькулятор* и содержит основные требования к разрабатываемой в рамках проекта программной системе и описание архитектуры программного решения.

## Область действия

Документ разработан в рамках проекта Депозитный и Кредитный Калькулятор на основе стандартного шаблона *Inteks SEP* и предназначен для использования студентами ФИТ и преподавателями курса ООАД.

## Определения и сокращения

Таблица : Определения и сокращения

|  |  |
| --- | --- |
| Термин | Описание |
| Кредит | Денежная сумма, предоставляемая кредитором на определённый срок и на определённых условиях. |
| Депозит | Сумма денег, переданная лицом [кредитному учреждению](https://ru.wikipedia.org/wiki/%D0%9A%D1%80%D0%B5%D0%B4%D0%B8%D1%82%D0%BD%D0%BE%D0%B5_%D1%83%D1%87%D1%80%D0%B5%D0%B6%D0%B4%D0%B5%D0%BD%D0%B8%D0%B5) с целью получить доход в виде [процентов](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%86%D0%B5%D0%BD%D1%82%D0%BD%D1%8B%D0%B9_%D0%B4%D0%BE%D1%85%D0%BE%D0%B4), образующихся в ходе финансовых операций с вкладом. |
| Аннуитетная схема погашения кредита | Схема ежемесячных выплат, при которой сумма (тело) *кредита* остается неизменной на протяжении всего срока погашения долга, но меняется ее структура. |
| Дифференцированная схема погашения кредита | Схема ежемесячных выплат, при которой размер ежемесячного платежа по погашению кредита постепенно уменьшается к концу периода кредитования. |
| Капитализация процентов | Причисление процентов к сумме вклада, позволяет в дальнейшем осуществлять начисление процентов на проценты. |

## Ссылки

В тексте содержатся ссылки на следующие документы:

<Имя файла документа>, v<версия> - <описание документа>

Ссылки приводятся в виде [N], где N – номер документа в вышеприведенном списке.

## Краткое описание

Содержание данного документа построено таким образом, чтобы дать ответ на следующие вопросы:

* Какие проблемы предметной области должен решать будущий программный продукт
* Посредством какой функциональности системы будут достигнуто решение проблем предметной области
* Какова архитектура программного решения

Описание предметной области и проблем, для решения которых предназначен будущий программный продукт, приведены в разделе 2.

Раздел 3 содержит описание требований к программному решению, раздел 4 – описание архитектуры выбранного решения.

# Предметная область проекта

Цель нашего проекта – облегчить трудоемкие расчеты платежей как *кредитов*, так и *депозитов*. Согласитесь, это в жизни каждого человека довольно актуально.

## Существующие проблемы

Данный проект решает следующие проблемы:

1. Высокая сложность расчетов.

* Большое количество нюансов предметной области, в которые обычный человек может быть не посвящён.
* Предложить пользователям простое приложение, за которым скрывается сложная реализация расчетов.

2. Недооценка переплат по *кредитам*.

* За процентами люди часто не видят реальных сумм, которые они должны будут кредитору.
* Реализация подсчетов переплат и вывод этой информации в удобном формате

3. Переоценка выгоды *депозитов*.

* Часто люди идут в банк, думая, что это лучший способ вложить деньги, но подсчет реальной выгоды, может направить их в другую сферу инвестиций.
* Реализация подсчетов полученных процентов и вывод этой информации в удобном формате

## Предполагаемое решение

Создание простого и удобного приложения, с помощью которого каждый человек сможет детальную информацию о том или ином *кредите* или *депозите*. Для удобства пользователя вся информация будет представлена в виде таблицы в приложении с возможностью ее сохранения в файл формата CSV.

# Требования к программному решению

Данный раздел описывает требования к программной системе, разрабатываемой в рамках проекта Депозитный и Кредитный Калькулятор.

## Роли

Роль - это что-то (например: другая система) или кто-то (например: человек) вне системы, которые взаимодействуют с ней. В предлагаемой к разработке системе идентифицированы следующие роли:

1. <Роль1> – <краткое описание роли>
2. <Роль2> – <краткое описание роли>

## Функциональные требования для роли Роль1

[В этом пункте необходимо сделать описание требований к системе в соответствии с Use-Case моделью. Для каждой роли необходимо ввести отдельный пункт 2-го уровня, такой как 3.2]

### <Use Case Name 1>

[В этом пункте необходимо сделать описание данного Use-Case.]

### <Use Case Name 2>

[В этом пункте необходимо сделать описание данного Use-Case.]

## Функциональные требования для роли Роль2

### <Use Case Name 1>

[В этом пункте необходимо сделать описание данного Use-Case.]

### <Use Case Name 2>

[В этом пункте необходимо сделать описание данного Use-Case.]

## Нефункциональные требования

[В этом пункте необходимо описать нефункциональные требования, такие как:

* Производительность
* Масштабируемость
* Ограничения по используемым компонентам
* Необходимость миграции данных из legacy систем
* И т.д.]

# Обзор архитектуры

Этот раздел описывает архитектуру системы.

### Компонентная модель системы

[Здесь приводится Component diagram - диаграмма компонентов системы, со связями между компонентами и интерфейсами между ними, а также описание их взаимодействия. Для каждого компонента дается краткое описание его места и предназначения в системе]

#### Компонент 1

[Здесь приводится более подробное описание предназначения компонента и Package diagram – диаграмма пакетов, из которых состоит данный компонент. Обязательно выделение на диаграмме интерфейсов пакета, служащих для связи с другими пакетами (фасад пакета), а также ключевых классов, используемых другими пакетами в use-case реализациях]

#### Компонент 2

[Здесь приводится более подробное описание предназначения компонента и Package diagram – диаграмма пакетов, из которых состоит данный компонент. Обязательно выделение на диаграмме интерфейсов пакета, служащих для связи с другими пакетами (фасад пакета), а также ключевых классов, используемых другими пакетами в use-case реализациях]

### Компоненты сторонних производителей

[Здесь приводится список использованных компонент сторонних производителей, использованных при разработке системы, с указанием их предназначения в системе]

### Схема развертывания приложения

[Здесь приводится Deployment diagram - диаграмма развертывания системы, со связями между узлами и указанием способа связи (протокола). На диаграмме обязательно указать, какие компоненты находятся на том или ином узле]

# Допущения и ограничения

[Краткое описание допущений, которые подразумевает данный проект, и любых ограничений (например, по бюджету, участникам, требуемому оборудованию, срокам и т.п.), накладываемых на его выполнение.]

Пример: При разработке проекта принято допущение, что число транзакций в единицу времени значительно (более чем в 10 раз) снижается в ночное время, что позволяет в период с 01:00 до 6:00 производить автоматическое обновление программного обеспечения системы, требующее полной перезагрузки и остановки сервиса на период до 5 минут.

# Известные проблемы

Ниже приводятся известные на данный момент проблемы и недоработки выработанного программного решения, а также возможные пути их устранения в последующих итерациях проекта.

## Невысокая производительность приложения

|  |  |
| --- | --- |
| Проблема | Производительность приложения экспоненциально деградирует при общем числе пользователей выше 10000 и числе одновременных сессий выше 100. |
| Ранг | 10 (высокий) |
| Влияние на проект | Невозможность использования системы при числе пользователей более 10000. |
| Пути решения | Кластеризация веб-сервера и сервера базы данных, а также применение load balancer в точке маршрутизации запроса к веб-серверу. |

Лист регистрации изменений

|  |  |  |  |
| --- | --- | --- | --- |
| Дата | Версия | Описание | Автор |
| 06.09.2020 | 1.0.1 | Добавлен пункт 2 (Project Vision), внесение 5 терминов в пункт 1.3 | Гайдамака А.В, Ланчуковская К.C. |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Лист регистрации проверок

|  |  |  |  |
| --- | --- | --- | --- |
| Дата | Версия | Описание | Автор |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |