**Техническое описание проекта по курсу ООАД**

**<Название проекта>**

**Студент ФИТ НГУ <ФИО>**

**группа 20207**

**Версия 1.0.0**

Содержание

[1.](#_heading=h.30j0zll) Введение 3

[1.1](#_heading=h.1fob9te) Цель 3

[1.2](#_heading=h.3znysh7) Область действия 3

[1.3](#_heading=h.2et92p0) Определения и сокращения 3

[1.4](#_heading=h.tyjcwt) Ссылки 3

[1.5](#_heading=h.3dy6vkm) Краткое описание 3

[2.](#_heading=h.1t3h5sf) Предметная область проекта 4

[2.1](#_heading=h.4d34og8) Существующие проблемы 4

[2.2](#_heading=h.2s8eyo1) Предполагаемое решение 4

[3.](#_heading=h.17dp8vu) Требования к программному решению 4

[3.1](#_heading=h.3rdcrjn) Роли 4

[3.2](#_heading=h.26in1rg) Функциональные требования для роли Роль1 4

[3.2.1](#_heading=h.lnxbz9) <Use Case Name 1> 4

[3.2.2](#_heading=h.35nkun2) <Use Case Name 2> 4

[3.3](#_heading=h.1ksv4uv) Функциональные требования для роли Роль2 4

[3.3.1](#_heading=h.44sinio) <Use Case Name 1> 4

[3.3.2](#_heading=h.2jxsxqh) <Use Case Name 2> 4

[3.4](#_heading=h.z337ya) Нефункциональные требования 4

[4.](#_heading=h.3j2qqm3) Обзор архитектуры 5

[4.1.1](#_heading=h.4i7ojhp) Компонентная модель системы 5

[4.1.1.1](#_heading=h.2xcytpi) Компонент 1 5

[4.1.1.2](#_heading=h.1ci93xb) Компонент 2 5

[4.1.2](#_heading=h.3whwml4) Компоненты сторонних производителей 5

[4.1.3](#_heading=h.2bn6wsx) Схема развертывания приложения 5

[5.](#_heading=h.qsh70q) Допущения и ограничения 5

[6.](#_heading=h.1pxezwc) Известные проблемы 6

[6.1](#_heading=h.49x2ik5) Невысокая производительность приложения 6

Техническое описание проекта по курсу ООАД

# Введение

## Цель

*[Здесь и далее текст в угловых скобках, такой как <Название проекта>, является полем документа. Вместо того, чтобы редактировать каждое место, где используется поле, нужно изменить его значение в свойствах документа, затем выделить весь текст и выполнить команду ‘Обновить поле’ по правой кнопке мыши.*

***Этот и прочие комментарии, выделенные синим, в финальной версии документа нужно удалить.****]*

Данный документ представляет собой техническое описание проекта *<Название проекта>* и содержит основные требования к разрабатываемой в рамках проекта программной системе и описание архитектуры программного решения.

## Область действия

Документ разработан в рамках проекта *<Название проекта>* на основе стандартного шаблона и предназначен для использования студентами ФИТ и преподавателями дисциплины ООАД.

## Определения и сокращения

*[В этой таблице нужно перечислить все термины предметной области, используемые далее в документе. В тексте документа термины имеет смысл выделять курсивом. Текст, выделенный зеленым, является ПРИМЕРОМ, в вашем проекте он* ***может и должен*** *быть другим.]*

**Таблица 1: Определения и сокращения**

| **Термин** | **Описание** |
| --- | --- |
| ATM | Automated Teller Machine - банкомат |
| VISA | Система пластиковых карт VISA |

## Ссылки

В тексте содержатся ссылки на следующие документы:

1. <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

Приложение, которое помогает студентам факультета информационных технологий Новосибирского Государственного Университета иметь быстрый доступ к учебным материалам, которые точно помогут им в изучении любого предмета. Так как все пособия, выложенные в этом приложении, рекомендованы студентами, которые уже окончили ВУЗ. Материалы разделены по курсам и отсортированы по предметам, а также по частоте использования учебного пособия для удобства пользования.

## Существующие проблемы

1. Огромное количество ненужного материала на просторах Интернета.
2. Наличие недостоверной информации в Интернете

Каждый пользователь сети способен написать статью или пособие, которое затрагивает ту или иную научную область. И далеко не факт, что данный человек компетентен в этой сфере.

## Предполагаемое решение

Возможно несколько решений данных проблем:

1. Ограничить возможность выкладывать информацию непроверенным лицам

2. Создание приложения, которое хранит в себе только проверенные материалы.

3. Забить и жить дальше

В наших силах разработать мобильное приложение, которое облегчит жизнь студентам факультета информационных технологий и решит обе этих проблемы.

# Требования к программному решению

Данный раздел описывает требования к программной системе, разрабатываемой в рамках проекта *<Название проекта>.*

## Роли

*[Как не сложно заметить, роль – это actor в терминах UML]*

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. Посетитель – пользователь, который просматривает ресурсы приложения

## Функциональные требования для роли Роль1

### <Use Case Name 1>

Скачивает материал

Действующие лица: Пользователь, система.

User-story: пользователь может скачать материалы, чтобы потом пользоваться ими на своём ПК

Предусловия: пользователь открыл нужный материал

Постусловия: материал становится доступным вне приложения

Триггер: пользователь нажал левой кнопкой мыши на кнопку скачать

Основной поток: система передает управление для скачивание

Альтернативные потоки: -

Исключения:

1. Пользователь потерял соединение с Интернетом

Дополнительные требования: -

### <Use Case Name 2>

Оставить отзыв или предложение

Действующие лица: Пользователь, система.

User-story: пользователь может оставить отзыв или предложение, чтобы создатели приложения могли узнать мнение пользователей и внести правки

Предусловия: пользователь перешёл на страницу отзывов

Постусловия: все пользователи приложения могут увидеть отзыв

Триггер: пользователь нажал на кнопку “Send”

Основной поток:

1. Система предлагает окно для ввода отзыва
2. Система отправляет данные о пользователе на сервер.
3. Сервер сохраняет данные о пользователе в базе данных

Альтернативные потоки: -

Исключения:

1. Пользователь нажал на кнопку “Send” но не заполнил поле отзыва
2. Пользователь потерял соединение с Интернетом

Дополнительные требования: -

### <Use Case Name 2>

Переход по ссылкам

Действующие лица: Пользователь, система.

User-story: пользователь может перемещаться по приложению, просматривая страницы

Предусловия: пользователь загрузил страницу платформы

Постусловия: пользователь видит новую страницу

Триггер: пользователь нажал на ссылку

Основной поток: система предоставляет новую страницу

Альтернативные потоки: -

Исключения:

1. Пользователь потерял соединение с Интернетом

Дополнительные требования: -

## 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** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

*[В качестве описания версии можно указывать какие изменения/дополнения были сделаны в этой версии по отношению к предыдущей.]*

Лист регистрации проверок

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

*[Здесь описываются результаты проверки документа. Для каждой проверки указывается число, версия документа, описание результатов проверки и имя человека, который делал проверку.]*