

Новосибирский Государственный Университет  
Факультет Информационных Технологий

# **Техническое описание проекта по курсу ООАД**

**NSUMedia бот**

**Студенты ФИТ НГУ  
Петров Сергей  
Авдеев Виталий  
группа 21211**

**Версия 1.0.0**

## Содержание

1. Введение	2
1.1 Цель	2
1.2 Область действия	3
1.3 Определения и сокращения	3
1.4 Ссылки	3
1.5 Краткое описание	3
2. Предметная область проекта	4
2.1 Существующие проблемы	4
о Предполагаемое решение	4
3. Требования к программному решению	4
3.1 Роли	4
а. Функциональные требования для роли Роль1	4
3.1.1 <Use Case Name 1>	4
3.1.2 <Use Case Name 2>	4
3.2 Функциональные требования для роли Роль2	4
3.2.1 <Use Case Name 1>	4
3.2.2 <Use Case Name 2>	4
3.3 Нефункциональные требования	4
- Обзор архитектуры	5
3.3.1 Компонентная модель системы	5
3.3.1.1 Компонент 1	5
3.3.1.2 Компонент 2	5
3.3.2 Компоненты сторонних производителей	5
3.3.3 Схема развертывания приложения	5
4. Допущения и ограничения	5
5. Известные проблемы	6
5.1 Невысокая производительность приложения	6

# Техническое описание проекта по курсу ООАД

## 1. Введение

### 1.1 Цель

Данный документ представляет собой техническое описание проекта "NSUMedia бот" и содержит основные требования к разрабатываемой в рамках проекта программной системе и описание архитектуры программного решения.

### 1.2 Область действия

Документ разработан в рамках проекта "NSUMedia бот" на основе стандартного шаблона и предназначен для использования студентами ФИТ и преподавателями дисциплины ООАД.

### 1.3 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

### 1.4 Ссылки

В тексте содержатся ссылки на следующие документы:

[1] diagrams.asta - UML диаграммы

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

### 1.5 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 – описание архитектуры выбранного решения.

## 2. Предметная область проекта

Обучение и учебный процесс студентов требуют доступа к обширным объемам учебного материала, не всегда имеющего в библиотеках или общедоступных сайтах. Студенты всегда ищут удобные способы доступа к этой информации, чтобы успешно учиться и готовиться к экзаменам. Однако поиск, организация и сортировка учебных материалов может быть вызовом, а существующие платформы и ресурсы не всегда удовлетворяют потребности студентов.

### 2.1 Существующие проблемы

*Студенты часто сталкиваются с неудобствами при поиске и доступе к необходимым учебным материалам. Ресурсы, предоставляемые учебными заведениями, могут быть неструктурированными, и важные материалы могут быть разбросаны по разным источникам. Это приводит к потере времени и снижению эффективности обучения. Кроме того, не всегда существует простой способ обновления или дополнения материалов, а также обмена ими между студентами.*

### 2.2 Предлагаемое решение

Мы предлагаем создать Telegram-бота, который будет решать эти проблемы, предоставляя студентам удобный доступ к учебным материалам, организованным по курсам, семестрам и дисциплинам. Студенты смогут легко загружать, обновлять и обмениваться учебными материалами через этого бота. Наш бот станет надежным партнером студентов в учебном процессе, помогая им легко находить учебные ресурсы и управлять ими.

## 3. Требования к программному решению

Данный раздел описывает требования к программной системе, разрабатываемой в рамках проекта "NSUMedia бот".

### 3.1 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. <Роль1> – <краткое описание роли>
2. <Роль2> – <краткое описание роли>

#### а. Функциональные требования для роли Роль1

*ссссс сделать описание требований к системе в соответствии с Use-Case моделью. Для каждой роли необходимо ввести отдельный пункт 2-го уровня, такой как а)*

##### 3.1.1 <Use Case Name 1>

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

##### 3.1.2 <Use Case Name 2>

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

#### 3.2 Функциональные требования для роли Роль2

##### 3.2.1 <Use Case Name 1>

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

##### 3.2.2 <Use Case Name 2>

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

### 3.3 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.

### 3.3.1 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]*

#### 3.3.1.1 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]*

#### 3.3.1.2 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]*

### 3.3.2 Компоненты сторонних производителей

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

### 3.3.3 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]*

## 4. 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.

## 5. Известные проблемы

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

### 5.1 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.]*