**Техническое описание проекта по курсу ООАД**

**Система автоматического учета посещаемости студентов НГУ**

**Студент ФИТ НГУ Зулин Даниил Константинович**

**Студент ФИТ НГУ Усова Дарья Сергеевна**

**группа 17206**

**Версия 1.0.0**

Содержание

[1. Введение](#_heading=h.30j0zll)

[1.1 Цель](#_heading=h.1fob9te)

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

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

[1.4 Ссылки](#_heading=h.tyjcwt)

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

[2. Предметная область проекта](#_heading=h.2s8eyo1)

[2.1 Существующие проблемы](#_heading=h.17dp8vu)

[2.2 Предполагаемое решение](#_heading=h.3rdcrjn)

[3. Требования к программному решению](#_heading=h.1ksv4uv)

[3.1 Роли](#_heading=h.44sinio)

[3.2 Функциональные требования для роли Роль1](#_heading=h.1y810tw)

[3.2.1 <Use Case Name 1>](#_heading=h.4i7ojhp)

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

[3.3 Функциональные требования для роли Роль2](#_heading=h.1ci93xb)

[3.3.1 <Use Case Name 1>](#_heading=h.3whwml4)

[3.3.2 <Use Case Name 2>](#_heading=h.2bn6wsx)

[3.4 Нефункциональные требования](#_heading=h.3as4poj)

[4. Обзор архитектуры](#_heading=h.49x2ik5)

[4.1.1 Компонентная модель системы](#_heading=h.147n2zr)

[4.1.1.1 Компонент 1](#_heading=h.3o7alnk)

[4.1.1.2 Компонент 2](#_heading=h.23ckvvd)

[4.1.2 Компоненты сторонних производителей](#_heading=h.ihv636)

[4.1.3 Схема развертывания приложения](#_heading=h.32hioqz)

[5. Допущения и ограничения](#_heading=h.2grqrue)

[6. Известные проблемы](#_heading=h.28h4qwu)

[6.1 Невысокая производительность приложения](#_heading=h.nmf14n)

Техническое описание проекта по курсу ООАД

# Введение

## Цель

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

## Область действия

Документ разработан в рамках проекта *"Контроль посещаемости"* на основе стандартного шаблона *Inteks SEP* и предназначен для использования студентами ФИТ и преподавателями курса 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.*

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

**Таблица 1: Определения и сокращения**

|  |  |
| --- | --- |
| **Термин** | **Описание** |
| Пара | Пара - два академических часа |
| 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

Предметной областью проекта будет являться система обучения, а точнее учет посещаемости студентов. Сбор этих данных необходим для разрешения некоторых вопросов, касающихся успеваемости ученика, так как отсутствие на занятиях может стать причиной выговора. Для учета посещаемости также необходимо наличие актуального расписания, а также информации о принадлежности студента к группе

## Существующие проблемы

1. **Отсутствие надлежащего контроля посещаемости лекций**. В данный момент некоторые преподаватели выбирают систему бального оценивания, в которую также желают включить посещаемость лекций. Но вследствие большого количества человек в аудитории перекличка и ручной контроль посещаемости занимают существенное время, что делает этот метод неудобным и, в некоторых случаях, ненадежным. Применение же автоматизированной системы могло бы предоставить надежную, а главное, не требующую человеческих усилий, информацию о посещаемости студентов

**2. Человеческий фактор на семинарских занятий и достоверность данных о посещаемости.** Зачастую учет посещаемости на семинарах и лекциях ведется путем переклички, либо же учащиеся сами записываются в список посещения. В первом случае есть вероятность, что преподаватель забудет об отметке посещаемости учащихся, также возможны ошибки в фамилиях студентов, дате проведения занятия и так далее. Во втором случае студент может появиться в списке посещения, даже если он сам на занятии не присутствовал. Автоматическая система учета посещаемости могла бы решить данные проблемы, облегчив работу преподавателя, а также предоставить достоверные данные о посещаемости каждого отдельного студента.

**3. Дополнительная мотивация посещать занятия.** Многие лекции не посещаются студентами из-за того, что на них нет учета посещаемости, автоматический учет может стать дополнительной причиной пойти на занятие.

## Предполагаемое решение

Данные проблемы могут быть решены с помощью внедрения нашей компьютерной системы, основной целью которой будет являться сбор и хранение данных о посещаемости студентов. С её помощью, лекторам не придется полагаться на старост или тратить более 15 минут от занятия на перекличку. Данная система может быть внедрена на семинарские занятия, что, в свою очередь, исключит возможность появления некорректных данных и предотвратит возможные споры между студентами и преподавателем

Также, мы планируем создать API, на основе которого могут быть созданы приложения для студентов, где они могли бы получать краткую сводку о своей успеваемости и получать дополнительную мотивацию к учебе.

# Требования к программному решению

Данный раздел описывает требования к программной системе, разрабатываемой в рамках проекта *"Контроль посещаемости".*

## 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. Преподаватель – сотрудник университета, ответственный за обучение студентов
2. Разработчик – человек, имеющий право получать доступ к информации через API сервиса

## Функциональные требования

* + 1. У сервиса должна быть возможность получить данные о посещении учеником занятия, в момент, когда он пришел на него.
    2. Возможность хранить данные о посещаемости по каждому отдельному студенту.
    3. Возможность получить данные о каждом ученике/группе учеников из базы данных.
    4. Преподаватель должен иметь возможность увидеть сводку об успеваемости по каждому конкретному ученику в группе.
    5. Информация о посещаемости по каждому предмету должна храниться, и должна быть возможность получить её.
    6. Информация о посещаемости студентами предметов каждого преподавателя должна храниться, и должна быть возможность получить её.

## Нефункциональные требования

* + 1. Учет посещаемости должен осуществляться автоматически. (распознавание изображений с камер/QR код/NFC)
    2. Для каждого преподавателя должно быть расписание, в соответствии с которым отмечаются посещения учеников.
    3. Обновление расписания каждого преподавателя должно происходить автоматически ежедневно в 00:00
    4. Обновление расписания должно занимать не более 5 минут
    5. Получение данных должно быть представлено в виде генерации отчета в формате pdf

# 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.]*