

Новосибирский Государственный Университет
Факультет Информационных Технологий

Техническое описание проекта по курсу ООАД

Виджет-расписание

**Студенты ФИТ НГУ
Королев Дмитрий Алексеевич
Косарев Никита Сергеевич
группа 19204**

Версия 1.0.0

Содержание

1.	Введение	3	
1.1	Цель	3	
1.2	Область действия	3	
1.3	Краткое описание	3	
2.	Предметная область проекта	4	
2.1	Существующие проблемы	4	
2.2	Предполагаемое решение	4	
3.	Требования к программному решению	4	
3.1	Роли	4	
3.2	Функциональные требования для роли Пользователь	4	
3.2.1	Use-case		4
3.3	Нефункциональные требования	4	
4.	Обзор архитектуры	5	
4.1	Компонентная модель системы		5
4.1.1	Компонент 1		5
4.1.2	Компонент 2		5
4.2	Компоненты сторонних производителей		5
4.3	Схема развертывания приложения		5
5.	Допущения и ограничения	5	
6.	Известные проблемы	6	
6.1	Малый список обработанных контент-провайдеров	6	

Техническое описание проекта по курсу ООАД

1. Введение

1.1 Цель

Упростить доступ к расписанию вуза для студентов по вводу номера их группы.

1.2 Область действия

Документ разработан в рамках проекта “Виджет расписания НГУ для рабочего стола” на основе стандартного шаблона *Inteks SEP* и предназначен для использования студентами ФИТ и преподавателями курса ООАД.

1.3 Краткое описание

В рамках нашего проекта рассматривается проблема труднодоступного расписания с сайта НГУ.

Результат: готовое приложение с "вменяемым" фронтендом и максимально простым, но легко модулируемым бэкэндом, позволяющее по номеру группы получить фронтное видение расписания.

В теории возможно расширение под получение расписания для преподавателей.

Реализация: для реализации нами будут использованы следующие технологии

Фронтенд будет написан на языке JavaScript, с использованием фреймворков.

Бэкэнд будет написан на языке Java.

Связующим звеном между фронтендом и бэкэндом и реализующей API компонентой будет выступать Spring Framework.

2. Предметная область проекта

2.1 Существующие проблемы

- Суть проблемы заключается в труднодоступности расписания НГУ
- Нужно пройти через несколько вкладок сайта с тяжеловесным дизайном.
- Решение проблемы заключается в парсинге html-кода страницы группы напрямую по URL-адресу.

2.2 Предполагаемое решение

Создать виджет для рабочего стола компьютера, который отображает расписание на текущий день с возможностью пролистывания

3. Требования к программному решению

3.1 Роли

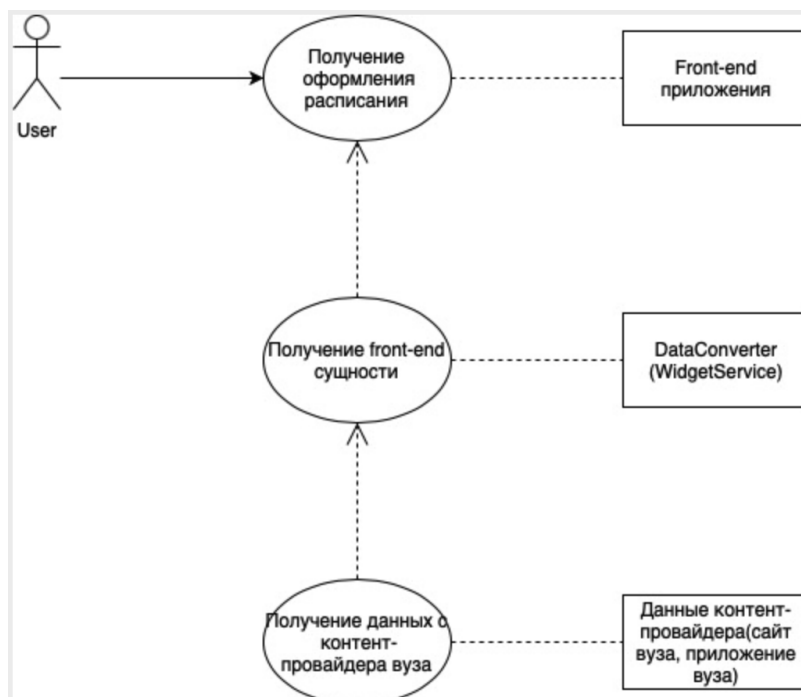
В предлагаемой к разработке системе идентифицированы следующие роли:

Пользователь – получатель информации о расписании

3.2 Функциональные требования для роли Пользователь.

Функциональных требований к роли нет.

3.2.1 Use Case



3.2.2 Получение данных с контент-провайдера вуза - получение данных в виде сопоставления день недели - расписание пар в этот день

3.2.3 Получение front-end сущности - получение данных готовых к обработке графической компонентой приложения

3.2.4 Получения оформления расписания - представление расписания в графическом виде для пользователя

3.2.5 DataConverter - конвертирует полученные данные из компактно-упакованного вида в формат, готовый для обработки графической компонентой.

3.3 Нефункциональные требования

- Единственным требованием является доступ к сети Интернет.

4. Обзор архитектуры

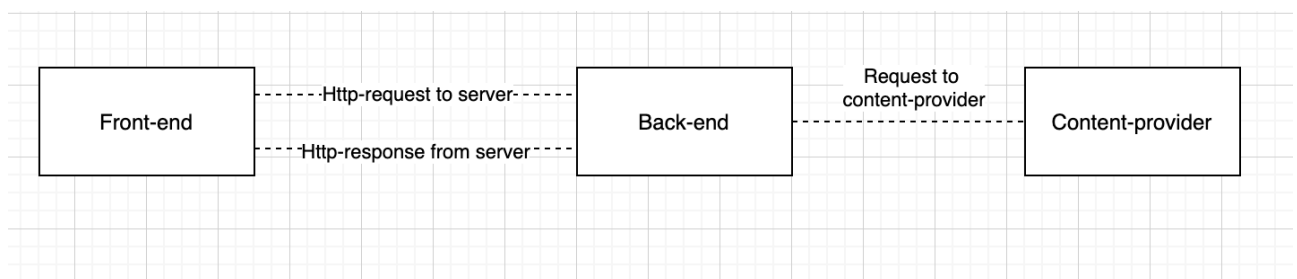
4.1 Компонентная модель системы

Компоненты взаимодействуют между собой посредством глобального http-запроса к серверу.

Front-end - визуальная компонента, предназначена для финальной конвертации и валидного визуального отображения данных.

Back-end - логическая компонента, предназначена для изъятия данных и предварительной конвертации перед их обработкой визуальной компонентой.

Content-provider - сайт, приложение вуза. Источник данных.



Front-end и Back-end компоненты взаимодействуют посредством GET http-запроса, с телом запроса, регулирующим искомую группу данных. Формат возвращаемых Back-end'ом данных - json.

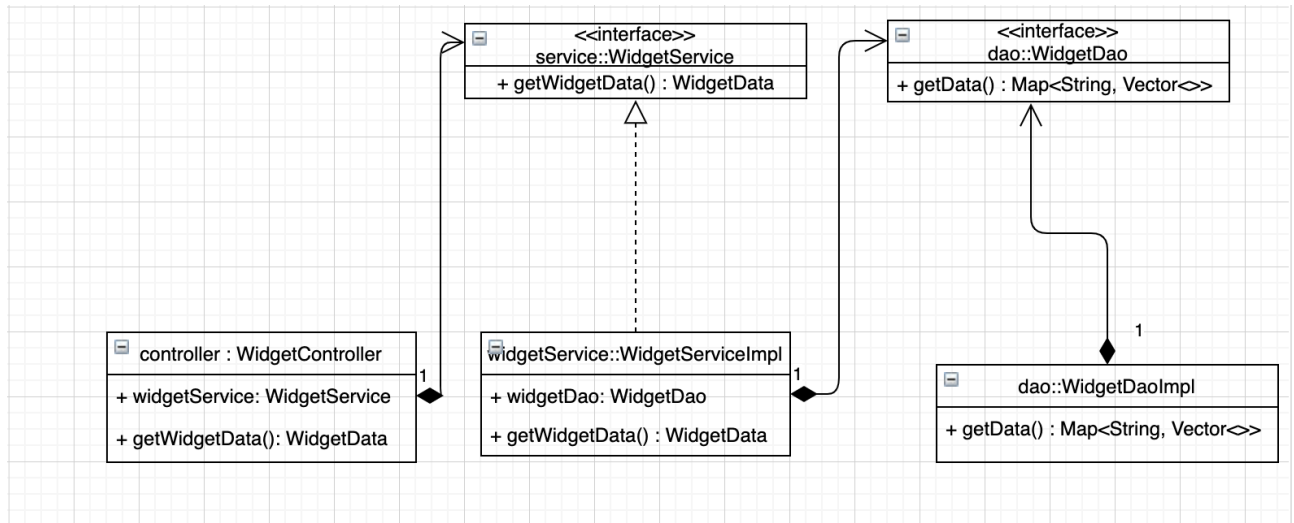
Back-end и content-provider(приложение или сайт вуза) взаимодействуют посредством GET http-запроса, с изъятием данных с источника данных. Формат возвращаемых content-provider'ом данных - html-код страницы расписания.

4.1.1 Front-end

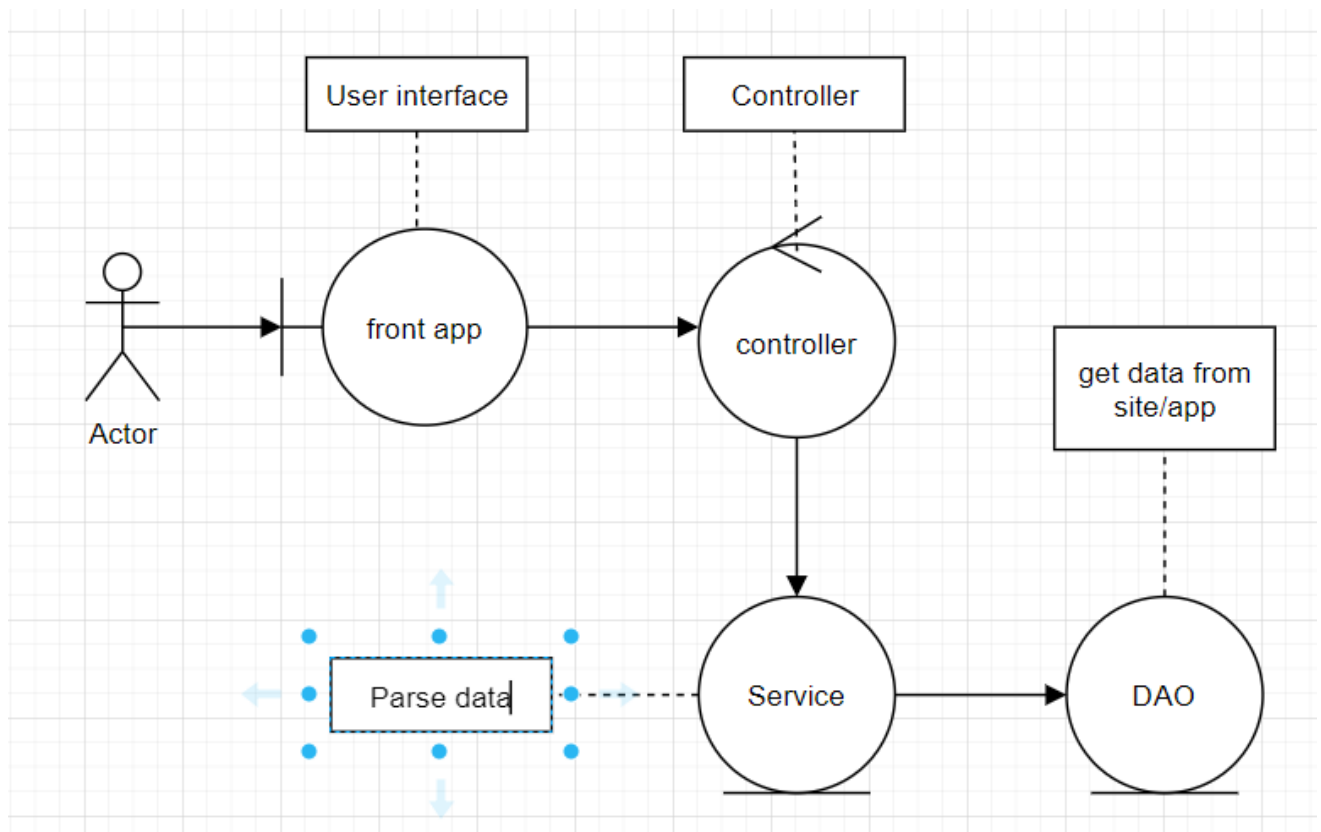
Данный компонент использован для визуального отображения информации.

Состоит из одного Java-класса FrontApp.

4.1.2 Back-end



Каждый вышележащий слой композитует в себе инъекцию имплементации экземпляра нижележащего.
 WidgetController композитует в себе инъекцию WidgetService.
 WidgetService композитует в себе инъекцию WidgetDao.
 WidgetServiceImpl выполняет интерфейс WidgetService.
 WidgetDaoImpl выполняет интерфейс WidgetDao.



4.2 Компоненты сторонних производителей

1. Spring Framework - используется для связи front-end и back-end компонент приложения.
2. Jsoup - для изъятия данных с контент-провайдера вуза.

4.3 Схема развертывания приложения

Front-end - развертывается на машине пользователя, back-end - развертка на удаленном сервере.

- 1) Пользователю поставляется только фронтовая компонента и Java SDK, бэкэндовая компонента разворачивается на удаленном сервере.
- 2) Для запуска необходимо установить Java SDK и запустить jar-файл.

5. Допущения и ограничения

При разработке проекта принято допущение, что все пользователи приложения - студенты НГУ, которым нужен доступ к данным лишь одного контент-провайдера.

6. Известные проблемы

6.1 Малый список обработанных контент-провайдеров

Проблема	Функционал завязан только на сайте НГУ.
Ранг	4(ниже среднего)
Влияние на проект	Нет
Пути решения	Модульность бэкэнда позволяет создать отдельные DAO-классы под отдельных контент-провайдеров.

Лист регистрации изменений

Дата	Версия	Описание	Автор
14.09.2021	0.0.2	Создание заготовки backend'a	Косарев
16.09.2021	0.0.3	Добавлен парсер сайта и полностью завершён back-end	Королев, Косарев
22.09.2021	0.0.8	Написан front-end	Королев
23.09.2021	1.0.0	Проект готов к запуску в release	Королев, Косарев

[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.]