**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук  
Департамент программной инженерии

|  |  |
| --- | --- |
| СОГЛАСОВАНО  Приглашенный преподаватель  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А. Н. Мешков «\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2024 г. | УТВЕРЖДАЮ  Академический руководитель образовательной программы «Программная инженерия» профессор департамента программной инженерии, канд. техн. наук  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Н. А. Павлочев «\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2024 г. |



**CЕРВИС ITEM COMPOSITION**

**Техническое задание**

**ЛИСТ УТВЕРЖДЕНИЯ**

**RU.17701729.02.06-01 ТЗ 01-1-ЛУ**

**Исполнители:**

студент группы БПИ226  
\_\_\_\_\_\_\_\_ / Альберштейн Г. Р. /  
«\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2024 г

студент группы БПИ226

\_\_\_\_\_\_\_\_/ Беликов Г.К. /  
«\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2024 г

**Москва 2024**

УТВЕРЖДЕН

RU.17701729.02.06-01 ТЗ 01-1-ЛУ

**CЕРВИС ITEM COMPOSITION**

**Техническое задание**

**RU.17701729.02.06-01 ТЗ 01-1**

**Листов 27**

**Москва 2024**

# АННОТАЦИЯ

Техническое задание – это основной документ, оговаривающий набор требований и порядок создания программного продукта, в соответствии с которым производится разработка программы, ее тестирование и приемка.

Настоящее Техническое задание на разработку «Сервис Item composition» содержит следующие разделы: «Введение», «Основание для разработки», «Назначение разработки», «Требования к программе», «Требования к программным документам», «Технико-экономические показатели», «Стадии и этапы разработки», «Порядок контроля и приемки» и приложения [7].

В разделе «Введение» указано наименование и краткая характеристика области применения программы.

В разделе «Основания для разработки» указан документ, на основании которого ведется разработка и наименование темы разработки.

В разделе «Назначение разработки» указано функциональное и эксплуатационное назначение программного продукта.

Раздел «Требования к программе» содержит основные требования к функциональным характеристикам, к надежности, к условиям эксплуатации, к составу и параметрам технических средств, к информационной и программной совместимости, к маркировке и упаковке, к транспортировке и хранению, а также специальные требования.

Раздел «Требования к программным документам» содержит предварительный состав программной документации и специальные требования к ней.

Раздел «Технико-экономические показатели» содержит ориентировочную экономическую эффективность, предполагаемую годовую потребность, экономические преимущества разработки программы.

Раздел «Стадии и этапы разработки» содержит стадии разработки, этапы и содержание работ.

В разделе «Порядок контроля и приемки» указаны общие требования к приемке работы.

Настоящий документ разработан в соответствии с требованиями:

1. ГОСТ 19.101-77 Виды программ и программных документов [1];
2. ГОСТ 19.102-77 Стадии разработки [2];
3. ГОСТ 19.103-77 Обозначения программ и программных документов [3];
4. ГОСТ 19.104-78 Основные надписи [4];
5. ГОСТ 19.105-78 Общие требования к программным документам [5];
6. ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом [6];
7. ГОСТ 19.201-78 Техническое задание. Требования к содержанию и оформлению [7].

Изменения к данному Техническому заданию оформляются согласно   
ГОСТ 19.603-78 [8], ГОСТ 19.604-78 [9].

# ТЕРМИНЫ И СОКРАЩЕНИЯ

1. Boilerplate код - шаблонный код, нетворческий программный код, который программисту приходится писать вследствие требований языка программирования, операционной системы, библиотеки подпрограмм, манеры программирования и прочего.
2. Батч-запрос - это объединение нескольких команд в один запрос для ускорения процессов, которые посылают много лёгких команд и основную часть времени находятся в ожидании ответа.
3. Runtime - время выполнения программы. В этом значении runtime обозначает период, когда программа находится в стадии исполнения. Термин употребляется для разграничения со стадией компиляции, когда исходный код преобразуется в машинный или промежуточный код.
4. RPS (от англ. request per second) - количество запросов в секунду времени.
5. GRPC (Remote Procedure Calls) — это система удалённого вызова процедур (RPC) с открытым исходным кодом, разработанная в Google в 2015 году. Основное назначение — реализация взаимодействия между микросервисными архитектурами при разработке приложений.
6. Endpoint (в переводе с англ. — конечная точка) — это шлюз, который соединяет серверные процессы приложения с внешним интерфейсом. Простыми словами, это адрес, на который отправляются сообщения.
7. Observability (в переводе с англ. наблюдаемость) — это процесс регулярной, всесторонней диагностики внутреннего состояния ИТ-системы на основе таких данных как метрики, события, логи, трассировки транзакций.
8. Backend — это внутренняя часть цифрового продукта, которая находится на сервере и скрыта от глаз пользователя.
9. Frontend — это разработка пользовательских функций и интерфейса. К ним относится всё, что пользователи видят на сайте или в приложении, и с чем можно взаимодействовать: картинки, выпадающие списки, меню, анимация, карточки товаров, кнопки, чекбоксы, интерактивные элементы.

# 

# СОДЕРЖАНИЕ

[**АННОТАЦИЯ 2**](#_heading=h.1fob9te)

[**ТЕРМИНЫ И СОКРАЩЕНИЯ 4**](#_heading=h.rfkk783hkwwj)

[**СОДЕРЖАНИЕ 5**](#_heading=h.cnlh4in18va3)

[**1. ВВЕДЕНИЕ 7**](#_heading=h.3znysh7)

[1.1. Наименование программы 7](#_heading=h.2et92p0)

[1.2. Краткая характеристика и область назначения 7](#_heading=h.4lp3uw1z0scz)

[**2. ОСНОВАНИЯ ДЛЯ РАЗРАБОТКИ 8**](#_heading=h.7hp5v05pz9lm)

[2.1. Документы, на основании которых ведется разработка 8](#_heading=h.1t3h5sf)

[2.2. Наименование темы разработки 8](#_heading=h.4d34og8)

[**3. НАЗНАЧЕНИЕ РАЗРАБОТКИ 9**](#_heading=h.2s8eyo1)

[3.1. Функциональное назначение 9](#_heading=h.17dp8vu)

[3.2. Эксплуатационное назначение 9](#_heading=h.3rdcrjn)

[**4. ТРЕБОВАНИЯ К ПРОГРАММЕ 10**](#_heading=h.26in1rg)

[4.1. Требования к функциональным характеристикам 10](#_heading=h.lnxbz9)

[4.2. Организация входных данных 10](#_heading=h.2jxsxqh)

[4.3. Организация выходных данных 10](#_heading=h.z337ya)

[4.4. Требования к временным характеристикам 11](#_heading=h.3j2qqm3)

[4.5. Требования к интерфейсу 11](#_heading=h.1y810tw)

[4.6. Условия эксплуатации 11](#_heading=h.2xcytpi)

[4.7. Требования к составу и параметрам технических средств 11](#_heading=h.1ci93xb)

[4.8. Требования к информационной и программной совместимости 11](#_heading=h.3whwml4)

[4.9. Требования к маркировке и упаковке 12](#_heading=h.2bn6wsx)

[4.10. Требования к транспортировке и хранению 12](#_heading=h.qsh70q)

[**5. ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ 13**](#_heading=h.3as4poj)

[5.1. Состав программной документации 13](#_heading=h.1pxezwc)

[5.2. Специальные требования к программной документации 13](#_heading=h.49x2ik5)

[**6. ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ 14**](#_heading=h.2p2csry)

[6.1. Ориентировочная экономическая эффективность 14](#_heading=h.147n2zr)

[6.2. Предполагаемая потребность 14](#_heading=h.3o7alnk)

[6.3. Экономические преимущества разработки по сравнению с отечественными и зарубежными аналогами 14](#_heading=h.23ckvvd)

[**7. СТАДИИ И ЭТАПЫ РАЗРАБОТКИ 15**](#_heading=h.ihv636)

[7.1. Стадии разработки, этапы и содержание работ 15](#_heading=h.zcc7zddsqa7i)

[7.2. Сроки разработки и исполнители 17](#_heading=h.iwseq6sejwg9)

[**8. ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ 18**](#_heading=h.32hioqz)

[**ПРИЛОЖЕНИЕ 1 19**](#_heading=h.sxmqvqntczix)

[**СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ 19**](#_heading=h.1hmsyys)

# 

# ВВЕДЕНИЕ

## Наименование программы

Наименование программы – «Сервис Item composition»

Наименование программы на английском языке – «Item composition service».

## Краткая характеристика и область назначения

«Сервис Item composition» - это серверное связующее программное обеспечение, связывающее другие системные компоненты для реализации динамического форматирования и представления данных по требуемым шаблонам.

Сервис может быть применим в любой системе, где необходимо обрабатывать данные в разных форматах представления (посты, короткие видео, карточки товаров, пуш-уведомления и тд).

# ОСНОВАНИЯ ДЛЯ РАЗРАБОТКИ

## Документы, на основании которых ведется разработка

Основанием для разработки является учебный план подготовки бакалавров по направлению 09.03.04 «Программная инженерия» и утвержденная академическим руководителем программы тема курсового проекта.

## Наименование темы разработки

Наименование темы разработки – «Сервис Item composition».

Программа выполнена в рамках темы курсовой работы «Сервис Item composition» в соответствии с учебным планом подготовки бакалавров (НИУ ВШЭ, факультет компьютерных наук) по направлению «Программная инженерия».

# НАЗНАЧЕНИЕ РАЗРАБОТКИ

## Функциональное назначение

Функциональное назначение – централизованное управление конфигурациями представления данных и эффективное взаимодействие между системными компонентами, включая обновление и применение конфигураций в реальном времени без остановки сервиса. Сервис обеспечивает гибкость в работе с HTTP и GRPC клиентами, минимизирует сетевую нагрузку за счет оптимизации запросов и объединения их в батчи, а также предоставляет интерфейсы для динамической настройки и конфигурирования внешних и внутренних систем.

## Эксплуатационное назначение

«Сервис Item composition» - Сервис будет использоваться для ленты новостей в среде Т-банк.

Конечными пользователями сервиса будут пользователи социальных сетей и страниц с новостями структур Т-банка .

# ТРЕБОВАНИЯ К ПРОГРАММЕ

## Требования к функциональным характеристикам

* + 1. Реализация API сервиса, а именно GRPC endpoint’а согласно контракту service.proto, через который происходит всё взаимодействие в сервисе.
    2. Интерфейсы для конфигурирования, представляющие собой .yaml файлы с описанием представления данных или описанием клиентов сервиса.
    3. Поддержка HTTP и GRPC клиентов для поставки необходимых данных из сторонних сервисов.
    4. При получении нового конфига сервис должен его применять, не требуя перезагрузки. То есть вся внутренняя кодогенерация, в том числе кодогенерация клиентского boilerplate кода должна выполняться в runtime.
    5. Поддержка зависимостей клиентов для построения сложных многокомпонентных запросов.
    6. Максимально уменьшение сетевого взаимодействия, посредством батч-запросов.

## Организация входных данных

На вход серверной части поступает запрос на применение конфигурации к какому-то набору данных. Данные предоставляются в формате JSON, используя протокол GRPC в соответствии с контрактом service.proto.

На вход клиентской части может поступать запрос на выполнение определенного действия, включая запросы к внешним HTTP или GRPC клиентам для динамического формирования данных в запросе сервиса.

## Организация выходных данных

В серверной части на выход подается GRPC ответ в формате JSON. Содержимое данного ответа должно содержать информацию о статусе обработки запроса, результатах применения конфигураций, динамически сформированных запросах или ошибок. Ответ также может включать данные для последующего взаимодействия с клиентами.

## Требования к временным характеристикам

Ответ на запрос пользователя должен получаться не позднее, чем через 1 секунды после отправки при нормальной нагрузке на сервис.

## Требования к интерфейсу

Интерфейс взаимодействия с серверной частью представляет из себя статические .yaml файлы для конфигурации модели представления данных, а также конфигурации HTTP и GRPC клиентов.

В конфигурации модели представления данных указываются view модели для входных данных и шаблоны форматирования выходных данных. Интерфейс поддерживает как статическое указание данных в шаблоне, так и указание запроса в сервис, из которого динамически должны браться информация для форматирования. Также интерфейс позволяет использовать условное ветвление.

В конфигурации клиентов указывается служебная информация о транспорте подключения, протоколе взаимодействия, а также описание API клиента.

Все изменения или добавления новых шаблонов/клиентов должны применяться без перезапуска сервиса, то есть в runtime.

## Условия эксплуатации

Требуемая классификация – пользователь, обладающий доступом к компьютеру, а также Интернетом.

## Требования к составу и параметрам технических средств

Для надежной работы программы требуется следующий состав технических средств:

1. Любое устройство с выходом в Интернет и возможностью развернуть Docker контейнер;
2. Свободный доступ в Интернет и стабильное соединение;

Требования к аппаратным характеристикам системы могут варьироваться от необходимой пропускной способности сервиса и входящего RPS.

## Требования к информационной и программной совместимости

Серверная часть должна быть реализована на языке программирования Go с использованием фреймворка для GRPC, такого как grpc-go. Для HTTP интерфейсов можно использовать фреймворк gin-gonic или аналогичный. Данные конфигураций и состояния системы должны храниться в базе данных MongoDB.

Клиентская часть должна быть реализована с использованием библиотеки grpc-go для взаимодействия с серверной частью через GRPC. При необходимости HTTP запросов может быть использована библиотека net/http, встроенная в стандартную библиотеку Go.

Формат данных: Обмен данными между клиентской и серверной частями должен производиться в формате JSON, соответствующем описанным в service.proto структурам.

Протокол взаимодействия: Основной протокол взаимодействия между компонентами системы — GRPC. При необходимости поддерживать HTTP взаимодействие для интеграции с другими системами.

Совместимость с внешними системами: Сервис должен поддерживать интеграцию с внешними API и клиентами, принимая и передавая данные в формате JSON по протоколам HTTP или GRPC.

Серверная часть должна обеспечивать масштабируемость и поддержку асинхронных запросов, включая выполнение батчевых операций, а также обеспечивать достаточную observability (логирование и трейсинг).

## Требования к маркировке и упаковке

Программа распространяется в виде электронного пакета, содержащего программную документацию и приложение (исполняемые файлы и прочие необходимые для работы файлы, в том числе файлы с исходным кодом), а также Docker файл, который предназначен для быстрого развертывания клиентской и серверной части.

## Требования к транспортировке и хранению

Транспортировка продукта должна осуществляться без нарушения полноты комплекта, предоставленного разработчиком изначально. Хранение продукта осуществляется с помощью репозитория в режиме публичного доступа на “Github”.

# ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ

## Состав программной документации

* «Сервис Item composition». Техническое задание (ГОСТ 19.201-78);
* «Сервис Item composition». Программа и методика испытаний (ГОСТ 19.30178);
* «Сервис Item composition». Текст программы (ГОСТ 19.40178);
* «Сервис Item composition». Пояснительная записка (ГОСТ 19.40479);
* «Сервис Item composition». Руководство оператора (ГОСТ 19.50579);

## Специальные требования к программной документации

* Документы к программе должны быть выполнены в соответствии с ГОСТ 19.106-78 и ГОСТами к каждому виду документа (см. п. 5.1.);
* Документация сдается в электронном виде в формате .pdf или .docx., сам программный продукт сдается в виде архива формата .zip или .rar;

# ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ

## Ориентировочная экономическая эффективность

Сервис может использоваться в любой микросервисной backend архитектуре для централизации форматирования и представления различных данных и передачи на frontend. Данный сервис уменьшает количество boilerplate кода для разработчиков программного обеспечения и предоставляет удобное API взаимодействия. Данный сервис поможет разработчикам значительно упростить процесс разработки и масштабирования своих сервисов.

## Предполагаемая потребность

Данная информация является закрытой и не подлежит распространению

## Экономические преимущества разработки по сравнению с отечественными и зарубежными аналогами

Так как данный сервис предназначается для интеграции с внутренними компонентами и

сервисами и предоставляет специфические и уникальные функции у него не может быть конкурентов в opensource. Однако есть инструмент от Microsoft, который позволяет валидировать и трансформировать JSON из одной схемы в другую, а также динамически выполнять C# скрипты над ним. Но данный инструмент не может полноценно заменить сервис Item Composition из-за невозможности интеграции с другими компонентами, что является одной из ключевых функций реализуемого продукта.

# СТАДИИ И ЭТАПЫ РАЗРАБОТКИ

## Стадии разработки, этапы и содержание работ

Стадии и этапы разработки были выявлены с учетом ГОСТ 19.102-77 [2]:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Стадии разработки** | **Этапы работ** | **Содержание работ** | **Исполнители работ** | **Сроки выполнения** |
| 1. Техническое задание | Подготовительные работы | * Постановка задачи. * Обоснование возможности решения поставленной задачи. * Определение структуры входных и выходных данных. * Предварительный выбор методов решения задач. | * Беликов Георгий Константинович * Альберштейн Герман Русланович | 21.11.2024 |
| Разработка и утверждение технического задания | * Определение требований к программе. * Определение требований к техническим средствам. * Определение стадий. Этапов и сроков разработки программы и документации на нее. * Согласование и утверждение технического задания. | * Беликов Георгий Константинович * Альберштейн Герман Русланович | 04.12.2024,  четверг |
| 2. Прототип продукта | Разработка программы | Разработка и тестирование программы. | * Беликов Георгий Константинович * Альберштейн Герман Русланович | XX.XX.2024,  XX |
| Разработка программной документации | Разработка программных документов в соответствии с требованиями ГОСТ 19.101-77 |
| Испытания программы | * Разработка, согласование и утверждение порядка и методики испытаний. * Отладка программы и программной документации по результатам испытаний. |
| 3. Внедрение | Подготовка и передача программы | * Подготовка и передача программы и программной документации для сопровождения | * Беликов Георгий Константинович * Альберштейн Герман Русланович | XX.XX.2024,  XX |

## Сроки разработки и исполнители

Программный продукт должен быть завершен не позднее 1.04.2025 - утвержденного срока сдачи работы.

Исполнители - Беликов Георгий Константинович, Альберштейн Герман Русланович.

# ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ

Проверка программного продукта, в том числе и на соответствие техническому заданию, описанию из протокола встречи (см список литературы п. 10), осуществляется заказчиком согласно “Программе и методике испытаний”, а также пункту 5.2 настоящего технического задания, производится на основе следующих элементов контроля:

1. Функциональное тестирование программы.
2. Проверка корректного заполнения данных в базе данных после создания игры.
3. Проверка на корректность обновления данных в базе данных после произошедшего события.
4. Проверка правильности работы функционала различных карт, в соответствии с правилами игры.
5. Проверка производительности продукта на устройствах с низкими характеристиками.
6. Проверка соответствия требованиям надежности в п. 4.4 настоящего ТЗ.

Каждый элемент контроля осуществляется в соответствии с документом “«Сервис Item composition». Программа и методика испытаний (ГОСТ 19.30178)”.

# ПРИЛОЖЕНИЕ 1

# СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. ГОСТ 19.101-77 Виды программ и программных документов. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
2. ГОСТ 19.102-77 Стадии разработки. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
3. ГОСТ 19.103-77 Обозначения программ и программных документов. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
4. ГОСТ 19.104-78 Основные надписи. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
5. ГОСТ 19.105-78 Общие требования к программным документам. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
6. ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
7. ГОСТ 19.201-78 Техническое задание. Требования к содержанию и оформлению. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
8. ГОСТ 19.603-78 Общие правила внесения изменений. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
9. ГОСТ 19.604-78 Правила внесения изменений в программные документы, выполненные печатным способом. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2

**ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ**

| Лист регистрации изменений | | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Номера листов (страниц) | | | | | Всего листов (страниц в докум.) | № документа | Входящий № сопроводительного докум. и дата | Подп. | Дата |
| Изм. | Измененных | Замененных | Новых | Аннулированных |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |