МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Ижевский государственный технический университет

имени М.Т. Калашникова»

Кафедра «Программное обеспечение»

Отчет

по лабораторной работе №2

по дисциплине “Проектирование и конструирование программного обеспечения”

на тему “Разработка базового описания”

Выполнил

студент группы Б21-191-1: Я.Е.Синявин

Принял

доцент: М.О. Еланцев

Ижевск

2024

1. Разработка прототипов экранных форм

Все формы можно найти по ссылке: https://www.figma.com/design/2wp5Pp4m5veZFSlGMh76Wh/GuessCode-Design?node-id=0-1&t=hOgvPORjzZVTzkln-1

Описание:

1. MainPage (Unauthorized)

Домашняя страница сайта, которую видит незарегистрированный пользователь. Отсюда он может зарегистрировать или войти в свой аккаунт.

1. MainPage (Authorized)

Домашняя страница сайта, которую видит зарегистрированный пользователь. Отсюда он может выйти из аккаунта, посмотреть свой профиль, а также приступить к решению задач.

1. RegisterPage

Форма для регистрации пользователя, также предусмотрена авторизация через GitHub пользователя.

1. LoginPage

Форма для входа в аккаунт, отсюда также можно попасть на страницу регистрации или войти с помощью GitHub.

1. FAQPage

Форма с ответами на часто задаваемые вопросы.

1. LeaderboardPage

Форма с актуальным положением рейтинговой таблицы.

1. ProfilePage

Форма, которая содержит основную информацию пользователя.

1. KataSelectPage

Форма для выбора задачи на решение.

1. Диаграмма сущностей (ER)
2. Разработка api системы (описание каждой сущности можно найти ниже)
3. POST “api/auth/register”

Регистрация пользователя

Входная информация:

body: RegisterDto

Выходная информация:

200 OK

1. POST “api/auth/login”

Вход в аккаунт пользователя

Входная информация:

body: LoginDto

Выходная информация:

TokenDto

1. POST “api/auth/login/github”

Вход в аккаунт пользователя через профиль GitHub

Входная информация:

body: code (string)

Выходная информация:

TokenDto

1. POST “api/kata-administration”

Создает кату

Входная информация:

body: KataDto

Выходная информация:

200 OK

1. PUT “api/kata-administration”

Обновляет значения каты

Входная информация:

body: KataDto

Выходная информация:

200 OK

1. DELETE “api/kata-administration”

Удаляет кату

Входная информация:

query: kataId (number)

Выходная информация:

200 OK

1. GET “api/kata-search/{kataId}”

Возвращает кату по ее идентифекатору

Входная информация:

query: kataId (number)

Выходная информация:

KataDto

1. GET “api/kata-search”

Возвращает набор кат, подходящих под переданный фильтр

Входная информация:

query: kataLanguage (number), kataType (number), kataDifficulty (number)

Выходная информация:

KataDto[]

1. PUT “api/kata-solve”

Проверяет решение каты

Входная информация:

body: KataAnswerDto

Выходная информация:

KataSolveResultDto

1. GET “api/profile-info/{userId}”

Возвращает информацию по профилю пользователя по его идентификатору

Входная информация:

query: userId (number)

Выходная информация:

UserProfileDto

1. GET “api/user”

Возвращает информацию по пользователю по его идентификатору

Входная информация:

query: userId (number)

Выходная информация:

UserDto

KataAnswerDto:

{

"kataId": {

"type": "integer",

"format": "int64"

},

"optionId": {

"type": "integer",

"format": "int32"

}

}

KataDto:

{

"title": {

"type": "string",

"nullable": true

},

"kataLanguage": {

"type": "integer",

"nullable": true

},

"kataDifficulty": {

"type": "integer",

"nullable": true

},

"kataType": {

"type": "integer",

"nullable": true

},

"kataRawJsonContent": {

"type": "string",

"nullable": true

},

"authorId": {

"type": "integer",

"format": "int64"

}

}

KataSolveResultDto:

{

"isAnswerCorrect": {

"type": "boolean"

},

"error": {

"type": "string",

"nullable": true

}

}

LoginDto:

{

"username": {

"type": "string",

"nullable": true

},

"password": {

"type": "string",

"nullable": true

}

}

RegisterDto:

{

"email": {

"type": "string",

"nullable": true

},

"username": {

"type": "string",

"nullable": true

},

"password": {

"type": "string",

"nullable": true

}

}

TokenDto:

{

"accessToken": {

"type": "string",

"nullable": true

},

"userId": {

"type": "integer",

"format": "int64",

"nullable": true

}

}

UserDto:

{

"username": {

"type": "string",

"nullable": true

},

"password": {

"type": "string",

"nullable": true

},

"email": {

"type": "string",

"nullable": true

},

"registrationDate": {

"type": "string",

"format": "date-time"

},

"rank": {

"type": "integer"

},

"gitHubProfileId": {

"type": "integer",

"format": "int64",

"nullable": true

},

"userProfileId": {

"type": "integer",

"format": "int64",

"nullable": true

},

"roleId": {

"type": "integer",

"format": "int64",

"nullable": true

}

}

UserProfileDto:

{

"avatarUrl": {

"type": "string",

"nullable": true

},

"description": {

"type": "string",

"nullable": true

},

"activityStatus": {

"$ref": "#/components/schemas/ActivityStatus"

},

"userId": {

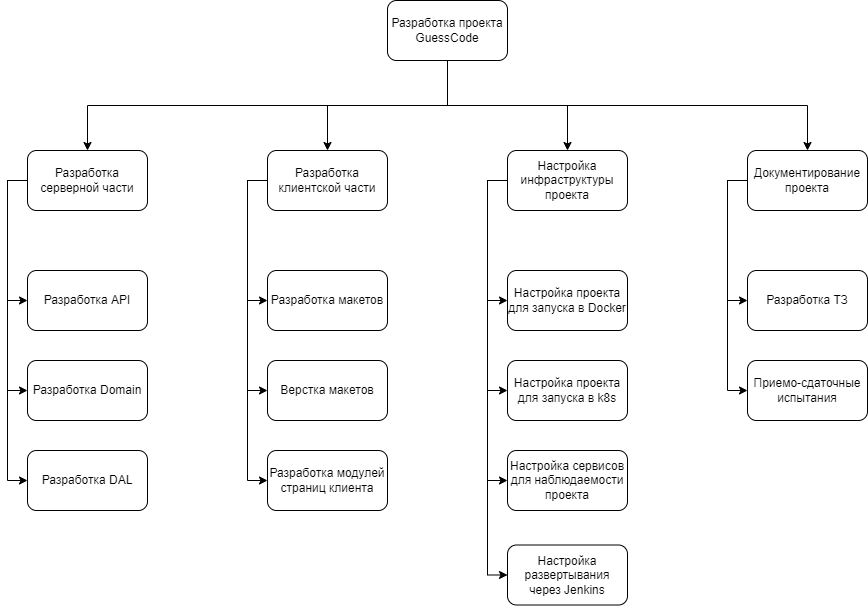
"type": "integer",

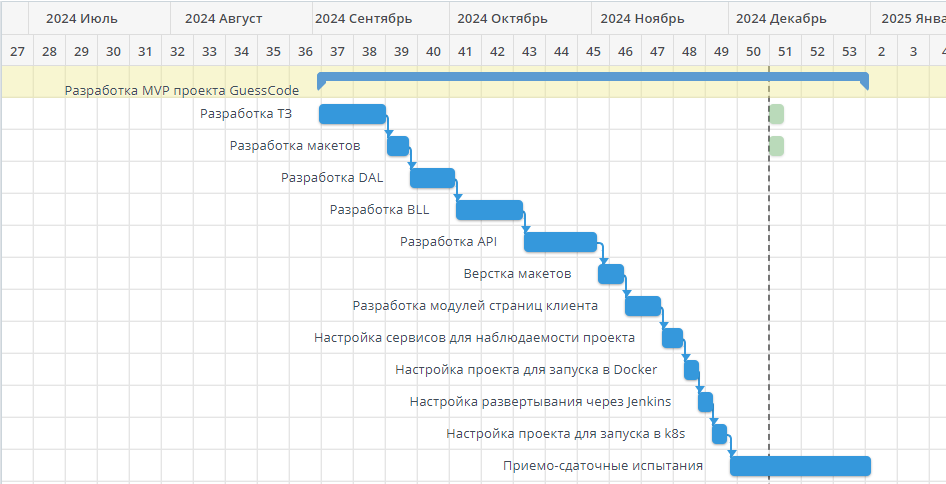
"format": "int64"

}

}

1. Иерархическая структура работ



1. Оценка времени выполнения проекта по методу PERT
2. Базовое расписание в виде диаграммы Ганта