|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

**Факультет «Информатика и системы управления»**

**Кафедра ИУ5 «Системы обработки информации и управления»**

Дисциплина «Разработка интернет-приложений»

Расчетно-пояснительная записка

Тема: «Проверка кодов студентов»

Студент: Цуприков Д.С.

Группа ИУ5-53Б

Преподаватель: Канев А.И.

2023 г.

# **ВВЕДЕНИЕ**

В настоящее время учебные заведения сталкиваются с рядом вызовов, связанных с проверкой и оценкой работы студентов. Одним из таких вызовов является необходимость эффективного и точного контроля за тем, как студенты пишут свои программы и коды. В свете этого введение системы по проверке кодов студентов становится актуальной и важной задачей для университета. Такая система может значительно упростить процесс проверки работ, увеличить объективность оценки и сэкономить время преподавателей из-за уменьшения времени на очный прием студентов, что позволит им более полноценно уделить внимание обучению и поддержке своих студентов.

Целью работы является реализация системы для проверки кода студентов, включающей веб-сервис, веб-приложение, мобильное приложение и выделенный сервис автотестирования кода.

Система предназначена для студентов и модераторов. На платформе предусмотрен доступ к списку языков программирования с информацией о выданных заданиях. Пользователи могут создавать формы, выбирая языки программирования и указывая необходимые комментарии. Система автоматизирует процесс создания, учета и отслеживания форм, обеспечивая эффективное управление логистическими операциями. Модераторы могут не только принимать и отклонять формы, но и редактировать существующие языки программирования или добавлять новые.

Нефункциональные требования к разрабатываемой системе:

* 1. Должна поддерживаться кроссплатформенность.
  2. Интерфейс системы и текст ошибок должны быть русифицируемы.

В ходе работы необходимо выполнить следующие задачи:

* 1. Разработать дизайн приложения.
  2. Создать базу данных в PostgreSQL.
  3. Создать веб-сервис на языке GoLang 1.20.
  4. Реализовать интерфейс гостя на технологии React.
  5. Развернуть веб-приложение React на GitHub Pages.
  6. Добавить авторизацию и аутентификацию в веб-сервис.
  7. Реализовать интерфейс пользователя в React.
  8. Реализовать интерфейс модератора в React.
  9. Создать мобильное приложение на Swift.
  10. Создать выделенный сервис для проведения автотестирования кодов на Python.
  11. Подготовить набор документации, включающий РПЗ, ТЗ и набор диаграмм.

# **БИЗНЕС-ПРОЦЕСС**

В университете по разным предметам может преподаваться большое число языков программирования. Для того чтобы сделать процесс отправки кодов с выполненными заданиями по этим языкам более организованным, студенты заранее формируют формы на проверку кодов по определённым языкам программирования. Каждая форма вначале создаётся как черновик, в который можно добавлять и из которого можно удалять языки программирования. В одной форме может быть сразу несколько языков программирования, и тогда они будут доставляться все вместе.

Когда студент определился с составом формы, он ее формирует, и она отправляется на модерирование. После этого её уже нельзя модерировать. Можно также посмотреть историю прошлых проверок. Сформированная форма проверяется модератором, он принимает решение об одобрении или отклонении формы. Формы требуют автоматической проверки, приведённых в них кодов. За это отвечает внешний сервис автотестирования, он оповещает о его результате.

Получить более детальную информацию можно из фотографии, названия языка программирования и предмета, на котором преподается данный язык. Также указываются тип задания и его описание.

Учебный план может обновляться, то есть в нем могут соответственно появляться новые дисциплины, поэтому модератор может вносить информацию о новых языках программирования. Если же в данных есть ошибка он имеет возможность редактировать уже имеющиеся языки программирования, а также удалять, если учебный план поменялся и языки оказались исключены из него. Функции пользователей с разными ролями описаны на диаграммах прецедентов (рис. 1).

Изображение выглядит как текст, диаграмма, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 1 - Диаграмма прецедентов

Гостям доступен просмотр языков программирования. Зарегистрированные гости – студенты. Они могут добавлять языки программирования в форму, просматривать список своих форм и сформировывать текущую форму. Формы обрабатываются модераторами. В результате обработки формы её либо одобряют, либо отклоняют. Модератору также доступны функции для работы с языками программирования, а именно: просмотр в табличном виде, редактирование, создание и удаление. Процесс оформления формы отражен на диаграмме деятельности (рис. 2).

Изображение выглядит как текст, диаграмма, линия, План

Автоматически созданное описание

Рисунок 2 - Диаграмма деятельности

Студент выбирает языки программирования, затем формирует на основе выбранных языков форму. Эту форму затем обрабатывает асинхронный сервис, а потом и модератор. Возможные состояния заявки отражены на диаграмме состояний (рис. 3).

Изображение выглядит как текст, диаграмма, линия, снимок экрана

Автоматически созданное описание

Рисунок 3 - Диаграмма состояний заявки

При выборе первого языка программирования формируется черновик. Последующие выбранные языки программирования добавляются в этот черновик. Студент затем формирует форму, удаляет её или выходит из приложения. Сформированную форму обрабатывает модератор, он может либо одобрить, либо отклонить её.

# **АРХИТЕКТУРА**

Архитектура системы отображена на диаграмме развертывания (рис. 4).

Бэкенд разворачивается на самом устройстве, а базы данных и s3-хранилище изображений в докер контейнерах. Сервис выдачи результата автотестирования и хранилище изображений разворачиваются отдельно и необязательно на том же компьютере, что и основной сервис.

Изображение выглядит как текст, диаграмма, План, Параллельный

Автоматически созданное описание

Рисунок 4 - Диаграмма развертывания

Мобильное и браузерное приложения обращаются к веб-сервису на базе языке Golang 1.20 через REST Web-API. Использование Golang обосновано его проверенной эффективностью, признанием в индустрии и современностью языка.

Данные хранятся в СУБД PostgreSQL, их структура отражена на ER диаграмме (рис. 5). СУБД PostgreSQL является одним из стандартов индустрии, поэтому было решено использовать её. Структура данных довольна проста. Помимо базовых полей, форма также имеет поле autotest, отражающее результат автоматической проверки кодов, которые она в себя включает. Модель языков программирования представляет собой набор полей, необходимых исключительно для бизнес-логики. Для хранения в одной форме нескольких языков программирования используется промежуточная таблица «codes», реализующая связь М-М и включающая в себя поле github, в котором в качестве дополнительной информации может храниться ссылка на репозиторий, где расположен код на определённом языке программирования, отсылаемый на проверку.

Изображение выглядит как текст, Шрифт, диаграмма, чек

Автоматически созданное описание

Рисунок 5 - ER диаграмма

Устройство бэкенда приложения изображено на диаграмме классов бэкенда (рис 6.). Модели имеют связи с таблицами в базе данных. Также некоторые модели имеют связи с внешними сервисами. В частности, языки программирования имеют связь с сервером статических файлов, т. к. в языках программирования хранится ссылка на их изображение, хранимое на сервере статических файлов.

Изображение выглядит как текст, диаграмма, Параллельный, План

Автоматически созданное описание

Рисунок 6 *-* Диаграмма классов бэкенда

Связь фронтенда и бэкенда отражена на диаграмме классов фронтенда (рис. 7). Ключевые страницы имеют связь с API аутентификации и авторизации, т. к. доступ к ним осуществляется только для авторизованных пользователей с определенными правами (ролями).

Изображение выглядит как текст, диаграмма, чек, линия

Автоматически созданное описание

Рисунок 7 *-* Диаграмма классов фронтенда

# **АЛГОРИТМЫ**

Алгоритм работы системы отображен на диаграмме последовательности (рис. 8). В основе системы лежит веб-сервис, реализующий внутри себя всю бизнес-логику. Он предоставляет доступ к методам из следующих доменов: языки программирования, формы и аутентификация. Методы следуют правилам REST API.

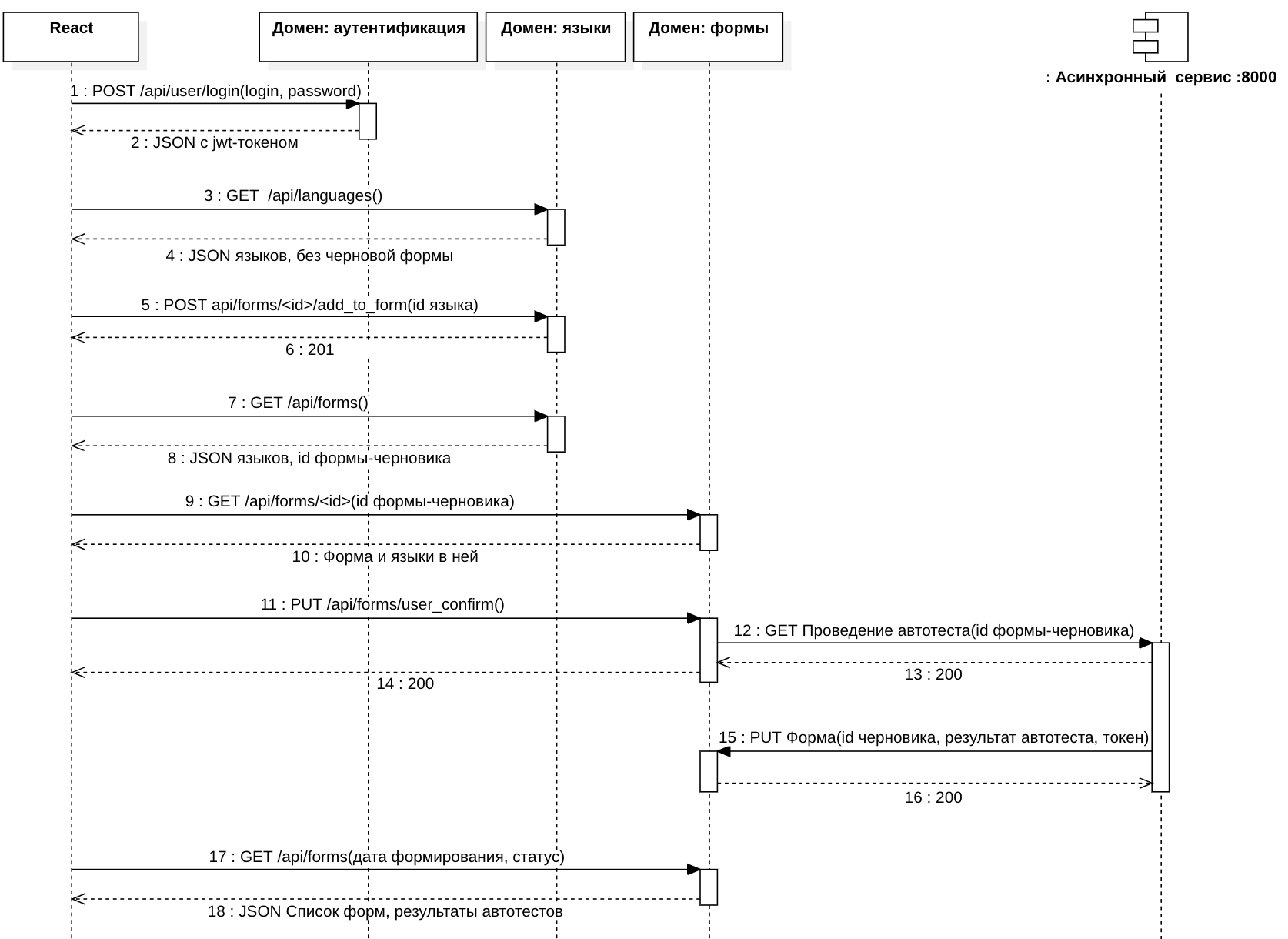


Рисунок 8 - Диаграмма последовательности

В начале бизнес-процесса происходит аутентификация пользователя. Для этого он отправляет через графический интерфейс запрос, передавая в нем логин и пароль. Если аккаунт с такими данными существует, то пользователь получает JWT в ответном запросе. Если же такого аккаунта не существует, или пароль введен неверно, пользователь получит ошибку. В таком случае ему надо либо пройти регистрацию, либо ввести пароль верно. Затем графический интерфейс пользователя запрашивает у веб-сервиса список языков программирования, которые возвращаются в JSON-формате. Пользователь выбирает язык программирования, который хочет получить, и, нажимая на кнопку «добавить» в графическом интерфейсе, отправляет запрос на добавление языка программирования в свою черновую заявку. Этот процесс может продолжаться несколько раз.

Когда пользователь определится с выбором, он нажимает на кнопку «сформировать» в графическом интерфейсе. После этого приложение запрашивает id черновой формы пользователя и затем отправляет запрос на формирование этой формы. В этот момент основной веб-сервис выполняет асинхронный запрос к сервису автотестирования кода, чтобы он сообщил, результаты проверки, как только автотестирование закончится. Через некоторое время пользователь увидит, что результат автотестирования у него появился.

Процесс рассмотрения форм происходит также через графический интерфейс. Модераторы могут просматривать списки всех форм и, нажимая на соответствующие кнопки, отправлять запросы на одобрение или отклонение форм в основной веб-сервис. В эти запросы также можно включить фильтры по имени создателя формы, текущему статусу и диапазону дат, в котором должны были быть созданы формы. Также через графический интерфейс модераторы могут управлять непосредственно языками программирования. Им доступны такие функции, как создание и редактирование языков программирования, просмотр списка языков программирования и удаление их. Для каждой из этих функция присутствует свой метод, отправляемый на основной веб-сервис.

# **ОПИСАНИЕ ИНТЕРФЕЙСА**

Главное меню приложения включает пункты, которые доступны в зависимости от роли пользователя (рис. 9, 10).

**Изображение выглядит как Мультимедийное программное обеспечение, программное обеспечение, снимок экрана, текст

Автоматически созданное описание**

Рисунок 9 - Главное меню (неавторизованный пользователь)

Изображение выглядит как текст, Мультимедийное программное обеспечение, программное обеспечение, мультимедиа

Автоматически созданное описание

Рисунок 10 - Главное меню (модератор)

Первоначальная страница для всех пользователей и гостей. В зависимости от типа пользователя её содержимое меняется. Гости, например, могут посмотреть список языков программирования и подробную информацию о них, а также нажать на кнопку «Войти», либо «Регистрация». У студента уже есть кнопки «Языки программирования», «Формы», «Корзина» и «Выход», у модератора к ним добавляется «Таблица языков программирования».

На странице с формой авторизации (рис. 11) отображается форма, через которую гость входит в свой аккаунт. При успешном вводе логина и пароля гость получает JWT, который сохраняется в local storage и используется при отправлении запросов.

**Изображение выглядит как текст, снимок экрана, дизайн

Автоматически созданное описание**

Рисунок 11 - Страница авторизации

На странице с формой регистрации (рис. 12) гости могут завести аккаунт. Для этого нужно указать логин, и пароль. Если введенный логин уже занят, система попросит пользователя сменить его.

**Изображение выглядит как текст, снимок экрана, дизайн

Автоматически созданное описание**

Рисунок 12 - Страница регистрации

На странице со списком языков программирования (рис. 13) отображается список языков программирования в виде карточек. У каждой карточки есть кнопка «Подробнее», переносящая пользователя на страницу с подробной информацией об языке программирования, и кнопка «Добавить в корзину» для добавления языка программирования в черновую форму. Сверху находится фильтр языков программирования, там можно ввести название языка программирования для отображения нужных данных пользователю.

**Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание**

Рисунок 13 - Страница со списком языков программирования

На странице с подробным описанием языка программирования (рис. 14) отображается подробная информация об языке программирования: название, предмет, задание и его описание.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 14 - Страница с подробным описанием языка программирования

На странице со списком форм (рис. 15, 16) отображается список форм. В зависимости от типа пользователя этот список будет функционально отличаться. Так, для студентов отображается список созданных ими форм: статус, дата создания, дата формирования, дата завершения, комментарии и кнопка «Подробнее» для открытия формы. Также можно отфильтровать формы по статусу, либо по дате формирования формы, выбрав либо одну дату, либо промежуток.

Изображение выглядит как текст, программное обеспечение, Мультимедийное программное обеспечение, снимок экрана

Автоматически созданное описание

Рисунок 15 - Страница со списком форм (студент)

Для модератора функция этой страницы шире. Для них отображается список всех форм всех пользователей с более подробной информацией: создатель формы, статус, автотест, дата создания, дата формирования, дата завершения, комментарии, кнопка «Подробнее» для открытия конкретной формы, и две кнопки «Подтвердить» и «Отменить» для подтверждения или отмены формы. Также присутствует фильтр по создателю формы.

Изображение выглядит как текст, снимок экрана, программное обеспечение, число

Автоматически созданное описание

Рисунок 16 - Страница со списком форм (модератор)

На странице с подробным описанием формы (рис. 17) отображается подробная информация о форме. Список выбранных языков программирования виде карточек, а также все поля формы.

Изображение выглядит как снимок экрана, текст, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 17 - Страница с подробным описанием формы

На странице с таблицей языков программирования (рис. 18) модератор может в компактном и удобном формате просмотреть список всех языков программирования, существующих в системе. Отображаются следующие поля: изображение языка программирования, название, предмет, задание и его описание. Также можно нажать на кнопки «Редактировать» или «Удалить» для изменения (рис. 19), либо удаления языка программирования. Можно отфильтровать языки программирования по названию, либо добавить новый язык программирования (рис. 20).

Изображение выглядит как снимок экрана, текст, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 18 - Страница с таблицей языков программирования

Изображение выглядит как текст, снимок экрана, программное обеспечение, мультимедиа

Автоматически созданное описание

Рисунок 19 - Страница редактирования языка программирования

Изображение выглядит как программное обеспечение, Мультимедийное программное обеспечение, текст, Значок на компьютере

Автоматически созданное описание

Рисунок 20 - Страница создания языка программирования

На странице редактирования языка программирования (рис. 19) модератор может отредактировать существующий язык программирования. Для редактирования доступны все поля: название, предмет, задание, описание задания, и изображение языка программирования.

На странице создания языка программирования (рис. 20) модератор может создать новый язык программирования. Для заполнения доступны все поля: название, предмет, задание, описание задания и изображение языка программирования.

# **ЗАКЛЮЧЕНИЕ**

В ходе работы были достигнуты следующие результаты:

1. Был разработан дизайн приложения с помощью набора стилей CSS и HTML тегов.
2. Была создана база данных в PostgreSQL.
3. Был создан веб-сервис на Golang, с использованием веб-фреймворка Gin.
4. Разработан интерфейс гостя с использованием технологии React Framework и подключен к веб-сервису.
5. Приложение интерфейса было развернуто на сервисе GitHub Pages по ссылке: https://mlechny.github.io/DevelopmentOfInternetApplications\_frontend/.
6. В веб-сервис добавлена авторизация через JWT, а методы задокументированы через Swagger.
7. Реализован интерфейс студента. Доступ к нему имеют только авторизированные пользователи.
8. Выделенный сервис был разработан и развернут в отдельном виртуальном окружении Python.
9. Реализован интерфейс модератора для подтверждения новых форм и редактирования языков программирования.
10. Было реализовано мобильное приложение на Swift, повторяющее интерфейс веб-приложения на React.
11. Подготовлен набор документации, включающий РПЗ, ТЗ и набор UML диаграмм.
12. Исходный код проекта доступен в GitHub https://github.com/Mlechny/DevelopmentOfInternetApplications.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

* 1. Система автоматизации процедур оценки образовательных результатов, сбора и обработки контекстных данных для решения задач управления в образовании [Электронный ресурс] // Science-education. URL: https://science-education.ru/ru/article/view?id=26576 (дата обращения: 13.11.2023).
  2. Информатика и системы управления [Электронный ресурс] // bmstu. URL: https://bmstu.ru/faculty/iu (дата обращения: 28.10.2023).
  3. Руководство по Gin Web Framework [Электронный ресурс] // GoLang. URL: https://github.com/gin-gonic/gin (дата обращения: 11.09.2023).
  4. Руководство по Gorm [Электронный ресурс] // Gorm. URL: https://gorm.io/docs/index.html (дата обращения: 14.10.2023)
  5. Полное практическое руководство по Docker [Электронный ресурс] // Habr. URL: https://habr.com/ru/articles/310460/ (дата обращения: 13.10.2023).
  6. Руководство по React [Электронный ресурс] // Metanit. URL: https://metanit.com/web/react/ (дата обращения: 11.11.2023).
  7. Quick Start – React [Электронный ресурс] // React. URL: https://react.dev/learn (дата обращения: 11.11.2023).
  8. Руководство по React Bootstrap [Электронный ресурс] // React Bootstrap. URL: https://react-bootstrap.netlify.app/docs/getting-started/introduction (дата обращения: 05.11.2023).
  9. Руководство по PostgreSQL [Электронный ресурс] // Metanit. URL: https://metanit.com/sql/postgresql/ (дата обращения: 05.10.2023).
  10. Руководство Apple-разработчика [Электронный ресурс] // Swift. URL https://developer.apple.com/documentation/swift/ (дата обращения: 08.10.2023).
  11. Документация Django [Электронный ресурс] // Django. URL: https://django.fun/docs/django/5.0/ (дата обращения: 22.12.2023).

**Приложение. Техническое задание**

|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

**Факультет «Информатика и системы управления»**

**Кафедра ИУ5 «Системы обработки информации и управления»**

Дисциплина «Разработка интернет-приложений»

Техническое задание

Тема: «Проверка кода студентов»

Студент: Цуприков Д.С.

Группа ИУ5-53Б

Преподаватель: Канев А.И.

2023 г.

1. **Цель работы**

Реализовать систему для проверки кода студентов, включающую в себя веб-сервис, веб-приложение, мобильное нативное приложение и выделенный сервис автотестирования отправляемого кода.

1. **Назначение**

Система предназначена для студентов и модераторов. Здесь предусмотрен доступ к выбору различных языков программирования для отправки. Система предоставляет автоматизированный способ создания, учета и отслеживания форм. Также она позволяет модераторам подтверждать или отклонять формы. Модераторы имеют возможность редактировать существующие языки программирования и создавать новые.

1. **Задачи**:
   1. Разработать дизайн приложения.
   2. Создать базу данных в PostgreSQL.
   3. Создать веб-сервис на golang.
   4. Реализовать интерфейс гостя на технологии React.
   5. Развернуть веб-приложение React на GitHub Pages.
   6. Добавить авторизацию и аутентификацию в веб-сервис.
   7. Реализовать интерфейса студента в React.
   8. Реализовать интерфейс модератора в React.
   9. Создать мобильное приложение на SwiftUI.
   10. Создать выделенный сервис для проведения автотестирования кода на Python.
   11. Подготовить набор документации, включающий РПЗ, ТЗ и набор диаграмм.
2. **Методы веб-сервиса:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№** | **Метод** | **Описание** | **URL** | **Входные данные** | **Выходные**  **данные** |
| * 1. Методы пользователей | | | | | |
| **4.1.1.** | POST | Регистрация | api/user/sign\_up | login: string  password: string |  |
| **4.1.2.** | POST | Вход в аккаунт | api/user/login | login: string  password: string | jwt-token |
| **4.1.3.** | GET | Выход из аккаунта. Доступно только авторизованным пользователям. | api/user/logout | jwt-token |  |
| * 1. Методы языков программирования | | | | | |
| **4.2.1.** | GET | Возвращает список языков программирования, удовлетворяющих переданным критериям и черновую форму пользователя. | api/languages | name: string | {  languages:  {  uuid: string,  name: string,  subject: string,  task: string,  image\_url: string,  description: string,  }[],  draft\_form: string?  } |
| **4.2.2.** | GET | Возвращает изображение из файлового хранилища сервиса. image\_url - путь до файла. | api/images/image\_url | image\_url: string | Файл |
| **4.2.3.** | POST | Добавление нового языка программирования.  Доступно только модератору. | api/languages | jwt-token  {  name: string,  subject: string,  image: file?,  task: string,  description: string  } | language\_id: string |
| **4.2.4.** | GET | Возвращает язык программирования по уникальному id | api/languages/{id} | id: string | {  uuid: string,  name: string,  subject: string, image\_url: string,  task: string,  description: string  } |
| **4.2.5.** | PUT | Обновляет информацию об языке программирования.  Доступно только модератору. | api/languages/{id} | jwt-token  id: string  {  name: string?,  subject: string?,  image: file?,  task: string?,  description: string?  } |  |
| **4.2.6.** | DELETE | Помечает язык программирования как удаленный. Доступно только модератору. | api/languages/{id} | jwt-token  id: string |  |
| **4.2.7.** | POST | Добавляет в форму пользователя язык программирования. Доступно только авторизованным пользователям. | api/languages/{id}/add\_to\_form | jwt-token  id: string |  |
| * 1. Методы форм | | | | | |
| **4.3.1.** | GET | Возвращает список форм пользователя, отфильтрованный по дате формирования и статусу. Доступно только авторизованным пользователям. | api/forms | jwt-token  {  status: string,  formation\_date\_start: datetime,  formation\_date\_end: datetime  } | {  uuid: string,  status: string, creation\_date: timestamp, formation\_date: timestamp,  completion\_date: timestamp?,  moderator\_id: string?,  student\_id: string, comments: string?,  autotest: string  }[] |
| **4.3.2.** | GET | Возвращает информацию о форме. Доступно авторизованным пользователям. | api/forms/{id} | jwt-token  id: string | {  form:  {  uuid: string,  status: string,  creation\_date: datetime,  formation\_date: datetime?,  completion\_date: datetime?,  moderator\_id: string?, comments: string?,  autotest: string?  }  languages:  {  uuid: string,  name: string,  subject: string,  image\_url: string,  task: string,  description: string  }[]  } |
| **4.3.3.** | PUT | Добавляет в форму пользователя комментарий. Доступно авторизованным пользователям. | api/forms/ | jwt-token  comments: string |  |
| **4.3.4.** | DELETE | Устанавливает статус форме «удалена». Только если её текущий статус «черновик». Доступно авторизованным пользователям. | api/forms/ | jwt-token |  |
| **4.3.5.** | DELETE | Удаление языка программирования из формы пользователя. Доступно авторизованным пользователям. | api/forms/delete\_language/{id} | jwt-token  id: string |  |
| **4.3.6.** | PUT | Устанавливает форме статус «отклонена» или «завершена». Только если её текущий статус «сформирована». Доступно только модераторам. | api/forms/{id}/moderator\_confirm | jwt-token  id: string  confirm: bool |  |
| **4.3.7** | PUT | Устанавливает форме статус «сформирована». Только если её текущий статус «черновик». Доступно авторизованным пользователям. | api/forms/user\_confirm | jwt-токен |  |
| **4.3.8** | PUT | Добавляет ссылку на гитхаб для таблицы м-м.  Доступно авторизованному пользователю - создателю этой формы. | api/forms/{id}/change\_github | jwt-token  id: string  github: string | {  language\_id: string,  form\_id: string,  github: string  } |
| **4.3.9** | PUT | Устанавливает результат автотеста кодов из формы. | api/forms/{id}/testing | id: string  testing\_result: bool,  token: string |  |
| * 1. Метод внешнего сервиса | | | | | |
| **4.4.1.** | POST | Автоматически проверяет коды из соответствующей формы. Метод внешнего сервиса. | api/testing | form\_id: string |  |

1. **Функциональные требования:**
   1. Главное меню. Здесь находятся кнопки для регистрации и авторизации или информация о текущем пользователе, и меню со списком доступных пунктов.
      1. Доступна всем пользователям.
      2. Действия
         1. Регистрация – переход на страницу 5.3. Только для гостей.
         2. Войти – переход на страницу 5.2. Только для гостей.
         3. Открыть список языков программирования – переход на страницу 5.4. Для всех пользователей.
         4. Открыть список форм – переход на страницу 5.6. Только для авторизированных пользователей.
         5. Выйти – производит запрос (метод 4.1.3).
         6. Управление языками программирования – переход на страницу 5.9. Только для модераторов.
   2. Страница с формой авторизации. На этой странице находится форма авторизации.
      1. Доступна гостям.
      2. Действия
         1. Войти – производит запрос (метод 4.1.2).
         2. Перейти к окну регистрации – перенаправляет на страницу 5.3.
   3. Страница с формой регистрации. На этой странице находится форма регистрации.
      1. Доступна гостям.
      2. Действия
         1. Зарегистрироваться – производит запрос (метод 4.1.1), в котором передаются введенные данные формы.
         2. Перейти к окну авторизации - перенаправляет пользователя на страницу 5.2.
   4. Страница со списком языков программирования. На этой странице располагается список всех языков программирования и панель фильтрации.
      1. Доступна всем пользователям.
      2. Выводится информация об языках программирования в виде карточек (метод 4.2.2).
         1. Изображение языка программирования.
         2. Название языка программирования.
         3. Предмет.
      3. Действия
         1. Отфильтровать языки программирования по указанному названию (метод 4.2.1).
         2. «Корзина» - перенаправляет на страницу формы 5.7.
         3. Добавить язык программирования в форму (метод 4.2.7.). Только для авторизированных пользователей.
         4. «Подробнее» (открыть данный язык программирования) – перенаправляет на страницу 5.5.
   5. Страница с подробным описанием языка программирования.
      1. Доступна всем пользователям.
      2. Выводится информация об языке программирования в построчном формате (метод 4.2.4).
         1. Изображение языка программирования.
         2. Название языка программирования.
         3. Предмет.
         4. Тип задания.
         5. Описание задания.
   6. Страница со списком форм.
      1. Доступна только авторизированным пользователям.
      2. Выводится информация о формах в табличном формате (методы 4.3.1).
         1. Статус формы.
         2. Дата создания.
         3. Дата формирования.
         4. Дата завершения.
         5. Кто создал. Только для модераторов.
         6. Результат автотестирования.
         7. Комментарии.
      3. Действия
         1. Открыть подробную информацию о форме – перенаправляет на страницу 5.7.
         2. Установить форме статус «отклонена» или «завершена» (метод 4.3.6.). Только для модераторов.
         3. Отфильтровать список форм по диапазону дат формирования и/или статусу (метод 4.3.1.) и/или создателю (только для модератора).
   7. Страница с подробным описанием формы. Пользователь может изменить набор языков программирования в форме. Модератор может просматривать заявки всех пользователей.
      1. Доступна только авторизированным пользователям.
      2. Выводится информация о форме в карточной форме. Список языков программирования, в виде карточек (метод 4.3.2.).
         1. Изображение языка программирования.
         2. Название языка программирования.
         3. Предмет.
      3. Действия
         1. Удалить язык программирования из формы (метод 4.3.5.). Если форма находится в статусе черновика. Только для создателя формы.
         2. Сформировать форму (метод 4.3.7.). Если форма находится в статусе черновика. Только для создателя формы.
         3. Удалить форму (метод 4.3.4.). Если форма находится в статусе черновика. Только для создателя формы.
         4. Сохранить форму (метод 4.3.3). Доступно в режиме редактирования.
         5. Отменить изменения (метод 4.3.3). Доступно в режиме редактирования.
         6. Подробнее (открыть язык программирования) – перенаправляет на страницу 5.5.
   8. Страница редактирования/создания языка программирования. Модератор может изменить существующий или создать новый язык программирования.
      1. Доступна только модераторам.
      2. Выводится информация об языке программирования в построчном формате.
         1. Изображение языка программирования.
         2. Название языка программирования.
         3. Предмет.
         4. Тип задания.
         5. Описание задания.
      3. Действия
         1. Изменение полей языка программирования. Можно изменить все вышеперечисленные поля языка программирования: указать для новой услуги/ изменить для существующей (методы 4.2.3 и 4.2.5).
         2. Сохранить язык программирования (метод 4.2.5). Доступно в режиме редактирования.
         3. Отменить изменения (метод 4.2.4). Доступно в режиме редактирования.
   9. Страница с таблицей языков программирования. Предоставляет модератору удобный способ отображения всех языков программирования.
      1. Доступна только модераторам.
      2. Выводит информацию об языках программирования в табличном виде (метод 4.2.1).
         1. Изображение языка программирования.
         2. Название языка программирования.
         3. Предмет.
      3. Действия
         1. Удалить язык программирования (метод 4.2.6).
         2. Редактировать язык программирования – перенаправляет на страницу 5.8.
         3. Добавить язык программирования – перенаправляет на страницу 5.8.
2. **Требования к программному обеспечению:**
   1. Серверная часть
      1. ОС: Linux/Windows/MacOS
      2. Docker 18.09.0 и выше
      3. СУБД: PostgreSQL 15.0 и выше
      4. GO 1.20.0 и выше
      5. nginx 1.19.2-alpine
      6. minio RELEASE.2022-10-15T19-57-03Z
      7. redis 7.2.2
      8. Node.js. 20.10.10 и выше
      9. Django 5.0
      10. Swift 5.7
   2. Клиентская часть
      1. ОС: Windows/MacOS/Linux
      2. Веб-браузер: Safari 11.1+/Chrome 40+/Opera 27+/Firefox 44+
3. **Требования к аппаратному обеспечению:**
   1. Серверная часть
      1. Процессор минимум 2-ядерный с частотой от 2 ГГц.
      2. Оперативная память от 8 Гб.
      3. Место на жестком диске от 2 Гб.
   2. Клиентская часть
      1. Процессор с частотой от 1ГГц.
      2. Оперативная память от 2 Гб.