

Санкт-Петербургский государственный университет

Бакова Елена Зауровна

Выпускная квалификационная работа

Проектирование и реализация сетевого
тренажера для обучения компьютерным
сетям на базе веб-сервиса Mimetnet

Уровень образования: бакалавриат

Направление *02.03.03 «Математическое обеспечение и администрирование
информационных систем»*

Основная образовательная программа *СВ.5162.2020 «Технологии программирования»*

Научный руководитель:
доцент кафедры системного программирования, к.ф.-м.н., Д. В. Луцев

Консультант:
старший преподаватель, И. В. Зеленчук

Рецензент:
старший инженер по тестированию, ООО «Ядро центр технологий мобильной связи»
К. С. Трофимов

Санкт-Петербург
2024

Saint Petersburg State University

Elena Bakova

Bachelor's Thesis

Design and implementation of a network
simulator for computer networking education
based on Mimir web service

Education level: bachelor

Speciality *02.03.03 "Software and Administration of Information Systems"*

Programme *CB.5162.2020 "Technology Programming"*

Scientific supervisor:
C.Sc., docent D. V. Luciv

Consultant:
Senior lecturer I. V. Zelenchuk

Reviewer:
Senior Testing Engineer at Yadro LLC K. S. Trofimov

Saint Petersburg
2024

Оглавление

Введение	4
1. Постановка задачи	6
2. Обзор	7
2.1. Существующие решения	7
2.2. Вывод	10
3. Требования	11
3.1. Проверка ответов	11
3.2. Пояснения к ответам	11
3.3. Разделы в тестах	11
3.4. Роли пользователей	12
3.5. Время на прохождение	12
3.6. Типы вопроса	12
4. Архитектура	13
4.1. Типовой сценарий работы тренажера	13
4.2. Компоненты системы	14
4.3. Используемые технологии	15
5. Особенности реализации	17
5.1. Проектирование	17
5.2. Реализация	20
6. Апробация	26
Заключение	27
Список литературы	28

Введение

В вузах на технических специальностях существует множество предметов, затрагивающих тему сетевых технологий. В частности, в СПбГУ на математико-механическом факультете проводится курс по компьютерным сетям, также в рамках нескольких других учебных дисциплин есть темы, связанные с этой областью. Однако эти курсы предполагают изучение только теоретических материалов. Для обеспечения более эффективного обучения компьютерным сетям необходима практика, так как просто лекционного материала может быть недостаточно. Демонстрация работы компьютерных сетей может помочь в усвоении курса — обучающийся лучше понимает, как всё работает и где применяется на практике.

С появлением средств эмуляции и визуализации процесс практического изучения компьютерных сетей сильно упрощается, потому как не требует наличия физического оборудования. Для того чтобы обучающиеся могли самостоятельно увидеть, как работает сеть, как отправляются и принимаются пакеты, в Санкт-Петербургском государственном университете был создан веб-эмулятор компьютерной сети Miminet [1], реализованный на базе эмулятора Mininet [2].

Однако обучающийся в силу неопытности не всегда может придумать, какую сеть ему нужно построить и на какие аспекты работы компьютерных сетей ему следует обратить внимание. Намного эффективнее было бы, если бы у обучающегося была возможность настроить или исправить ту или иную сеть с уже заданной топологией. Это избавит его от необходимости самостоятельно придумывать примеры и позволит глубже погрузиться в тему сетевых технологий.

Кроме того, самостоятельная практическая работа играет важную роль в закреплении материала. Поэтому разработка и внедрение тренажеров с практическими заданиями по компьютерным сетям помогут сделать образовательный процесс более эффективным.

Также в дополнение к практическим заданиям было решено добавить и теоретические вопросы, которые помогут как при подготовке к экзаме-

нам или техническим собеседованиям, так и при проверке собственных знаний.

1. Постановка задачи

Целью данной работы является создание тренажера по компьютерным сетям для веб-эмулятора Miminet. Для ее достижения были поставлены следующие задачи.

1. Провести обзор существующих решений.
2. Спроектировать тренажер.
3. Реализовать приложение.
4. Провести апробацию.

2. Обзор

2.1. Существующие решения

Обзор проводился с целью изучения уже имеющихся способов создания онлайн-тестов, а также для рассмотрения преимуществ и недостатков таких систем, которые можно учесть при создании продукта.

В системах рассматривались следующие особенности:

- процесс создания теста,
- возможность разбить тест по разделам,
- возможность создавать вопросы с разным типом ответов (множественный выбор, сортировка),
- возможность ограничивать время ответа.

Инструмента, совмещающего в себе и теоретические, и практические задачи найдено не было. Поэтому рассматривались отдельно популярные платформы для тестирования и для заданий по компьютерным сетям.

2.1.1. Платформы для тестирования

Stepik [3] — образовательная платформа, а также конструктор онлайн-тестов. Каждый курс разделён на модули, в модуле содержатся отдельные шаги разного типа (пример, рисунок 1). Типами тестового шага являются вопрос с вариантами ответа, задача на сопоставление или на сортировку и табличная задача. Платформа позволяет добавлять комментарии к ответам (как верным, так и неверным), а также ограничивать число попыток. Ограничивать время ответа можно только в экзаменационном модуле, преподаватель задаёт время на прохождение всего экзамена, пройти его можно всего один раз.

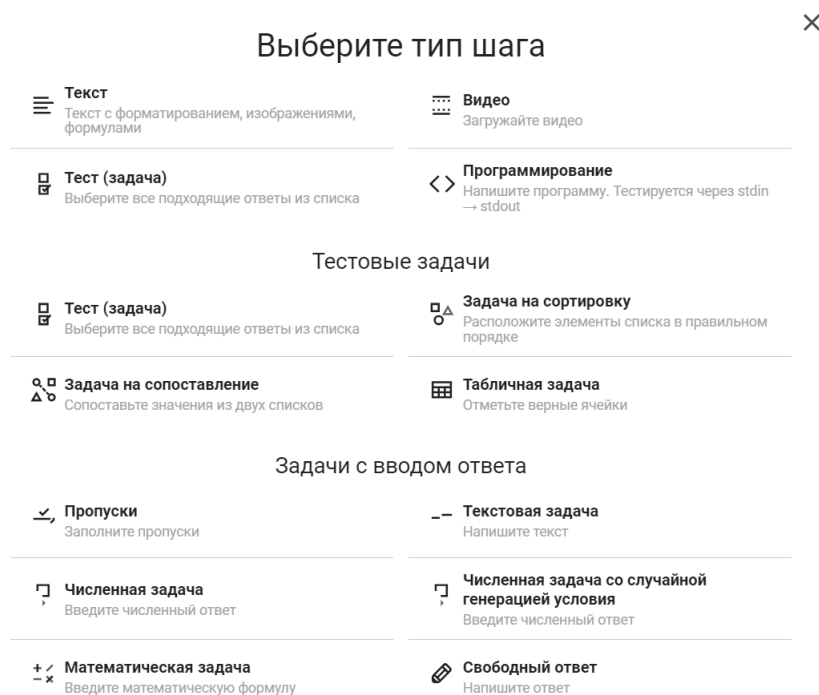


Рис. 1: Меню выбора типа шага при создании курса на платформе Stepik

Открытое образование [4] — образовательная платформа, предлагающая как платные, так и бесплатные курсы от ведущих российских вузов. Курсы разделены на модули, оканчивающиеся контрольными заданиями. Задания предполагают множественный или единственный выбор либо ответ в виде текста или числа. Количество попыток ограничивается либо для всего теста, либо для одного конкретного вопроса (например, рисунок 2). Также на представленном рисунке видно, что платформа позволяет задавать пояснения к ответу. Время ограничивается только для экзаменационного модуля.

Вопрос 1

1/1 point (graded)

Существование различных определений философии является следствием:

- ☐ наличия разных переводов древнегреческого слова «философия» на современные европейские языки (включая русский)
- ☒ многообразие человеческого опыта
- ☐ невыразимости философской мысли в языке
- ☐ принципиальной интуитивности философии



Ответ

Верно:

Если имеется множество ответов на вопрос: «Что такое философия?», то его решение не может быть сведено ни к выбору наиболее правильного ответа, ни к поиску абстрактного тождества зачастую несовместимых друг с другом философских учений. Вот почему при исследовании этого вопроса необходимо обращаться к той многообразной действительности, на которую и направлено философское осмысление. Вот в чем на самом деле и заключается многообразие и мнимое несоответствие, несочетаемость друг с другом ответов на этот вопрос. И тогда окажется необходимым не столько сопоставление уже имеющихся ответов, сколько исследование отношения философского сознания к повседневному и историческому опыту людей.

[Сохранить](#) | [Show answer](#)

Отправить

Вы использовали 1 из 3 попыток

✓ Верно (1/1 балл)

Рис. 2: Пример вопроса на платформе Открытое образование

Системы управления обучением

Была рассмотрена одна из наиболее популярных LMS-систем — Moodle [5] — система организации дистанционного обучения. Moodle имеет широкий спектр функциональностей для создания теоретических опросов (например, вопросы с множественным выбором, открытым ответом или расстановкой приоритетов). Платформа позволяет управлять временем прохождения тестов, настраивать контроль доступа и имеет систему автоматической проверки ответов. Тем не менее внедрение такой системы может быть сложным процессом ввиду необходимости освоения её функционала, а также опытного администрирования для оптимальной работы.

Сервисы онлайн-опросов — инструменты, позволяющие создавать анкеты и формы сбора информации. Например, Google Формы [6] — одна из популярных платформ, позволяет создать опросы, содержащие

несколько разделов, можно выбрать тип вопроса (некоторые требуют ручной проверки), а также добавить пояснение к ответу, ограничить время на ответ невозможно.

2.1.2. Платформы для заданий по сетям

Cisco Packet Tracer [7] — симулятор сетей, предназначенный для получения практических навыков работы с сетями. На платформе преподаватель может создавать задания для обучающихся или брать готовые задачи из шаблонов. Также симулятор позволяет наглядно изучить работу сети, отследить пути перемещения пакетов и рассмотреть другие параметры. Приложение требует установки на компьютер, исходный код закрыт, что не позволяет осуществить персональную настройку программы под личные требования. Ранее Cisco Packet Tracer был доступен бесплатно для студентов академии Cisco.

Graphical Network Simulator (GNS3) [8] — эмулятор, часто используемый для обучения сетевым технологиям, подготовки к сертификационным экзаменам, а также для разработки и тестирования сетевых конфигураций. Программа требует установки на компьютер, но является бесплатной и имеет открытый исходный код. В GNS3 используются механизмы виртуализации, что усложняет работу с продуктом и делает саму программу ресурсоёмкой.

Boson NetSim [9] — платный программный продукт, используемый для подготовки к сертификационным экзаменам. Предоставляет доступ к виртуальным лабораториям, а также симуляцию сетевых устройств.

2.2. Вывод

Таким образом, ни одна из рассмотренных платформ полностью не может удовлетворить всем критериям, поэтому ни одно из решений не было использовано. Тем не менее все преимущества и недостатки систем были учтены при составлении требований и проектировании тренажера.

3. Требования

Для лучшего понимания, какой должна быть система, был сформулирован следующий список требований к тренажеру.

1. Автоматическая проверка ответов.
2. Наличие пояснения к ответу.
3. Наличие разделов в тестах.
4. Наличие ролей у пользователей.
5. Ограничение времени на прохождение.
6. Поддержка различных типов вопросов.

Рассмотрим их более детально.

3.1. Проверка ответов

Проверку всех ответов необходимо проводить автоматически, без необходимости участия преподавателя. Это нужно для того, чтобы обучающийся мог сразу увидеть свой результат без ожидания.

3.2. Пояснения к ответам

Пояснения к правильным и неправильным ответам в тестах помогут понять, в чем ошибка, или лучше разобраться в теме при правильном ответе.

3.3. Разделы в тестах

Тесты могут содержать довольно много вопросов (например, экзамен на 80–100 вопросов), чтобы не проходить все темы за один раз, и для более простой навигации необходимо в тестах сделать разделы.

3.4. Роли пользователей

Не все пользователи должны иметь возможность изменять тесты. Необходимо добавить роли пользователей, чтобы ограничить доступ к редактированию, удалению и изменению вопросов.

3.5. Время на прохождение

Для обеспечения эффективности и реалистичности прохождения тестов система должна предусматривать ограничение времени на их прохождение, что поможет пользователям управлять своим временем и подготовиться к условиям реального тестирования.

3.6. Типы вопроса

В системе должна быть возможность использовать различные типы вопросов, чтобы обеспечить разнообразие и гибкость в создании тестов.

4. Архитектура

В данном разделе будут описаны следующие важные аспекты тренажера: типовой сценарий работы, схема компонентов и технологии реализации.

4.1. Типовой сценарий работы тренажера

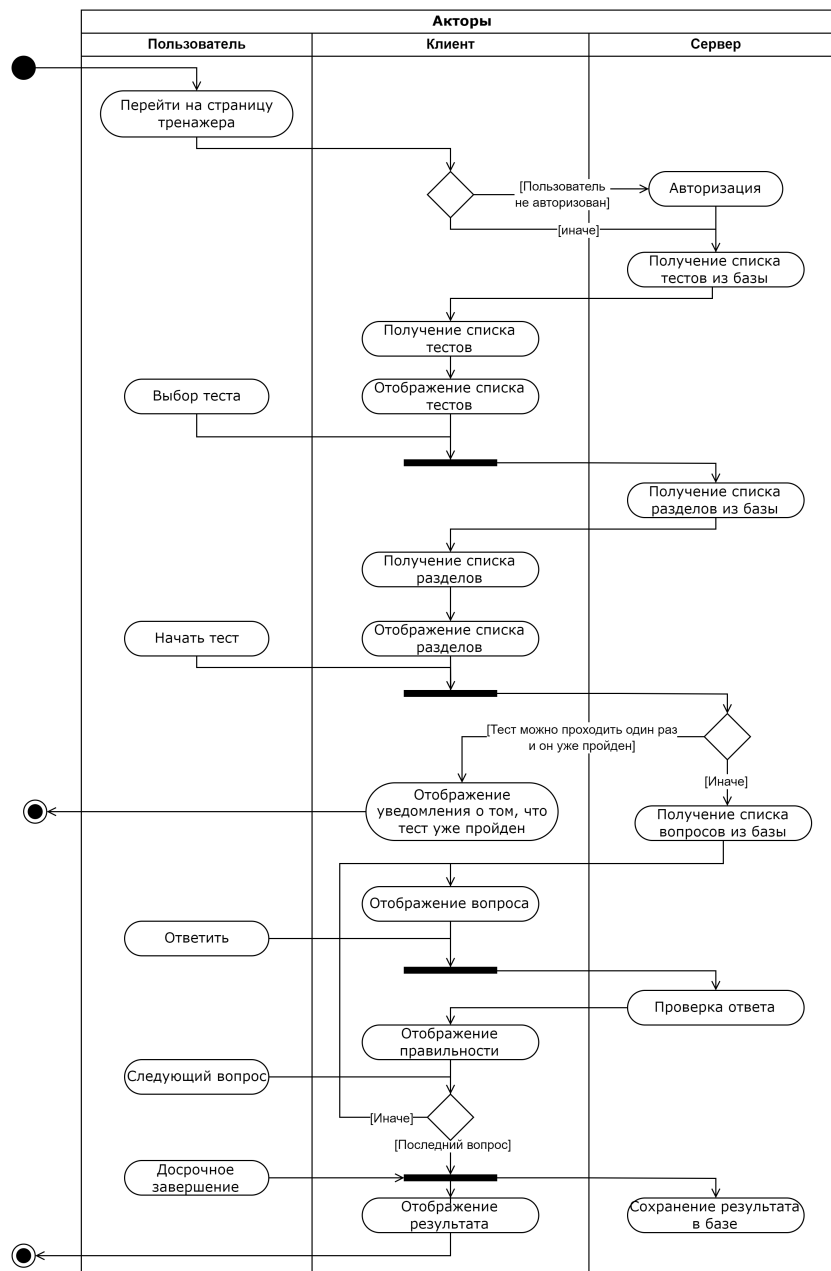


Рис. 3: Диаграмма активностей тренажера

На рисунке 3 представлена диаграмма активностей, описывающая типовой сценарий взаимодействия пользователя с тренажером.

1. Пользователь запускает тренажер. Если пользователь не авторизован, происходит редирект на страницу входа в систему. Если он авторизован, то выполняется загрузка списка тестов из базы и переход на страницу с доступными тестами.
2. Пользователь выбирает конкретный тест и переходит на страницу с разделами этого теста.
3. Пользователь выбирает начать тест. Если пользователь проходил этот тест ранее, и его нельзя проходить повторно, то появляется уведомление о том, что тест уже был пройден. Иначе тест запускается.
4. Пользователь начинает проходить тест — отображается первый вопрос. После выбора ответа на сервере производится проверка и выдаются пояснения к ответу. Затем пользователь переходит к следующему вопросу, и цикл действий повторяется.
5. Если пользователь ответил на последний вопрос, то отображается страница с результатами всего теста. Если пользователь досрочно завершает тест, также отображается результат.

4.2. Компоненты системы

На рисунке 4 представлена диаграмма компонентов Miminet, цветом выделены компоненты, которые были модифицированы в ходе данной работы.

Пользователь взаимодействует с приложением через веб-браузер, который также отвечает за отображение и функциональность всех страниц тренажера. Все HTTP-запросы отправляются сначала на веб-сервер Nginx и далее перенаправляются на веб-сервер uWSGI (Web Server Gateway Interface), который в свою очередь отправляет его Python-приложению. Приложение взаимодействует с базой данных с помощью

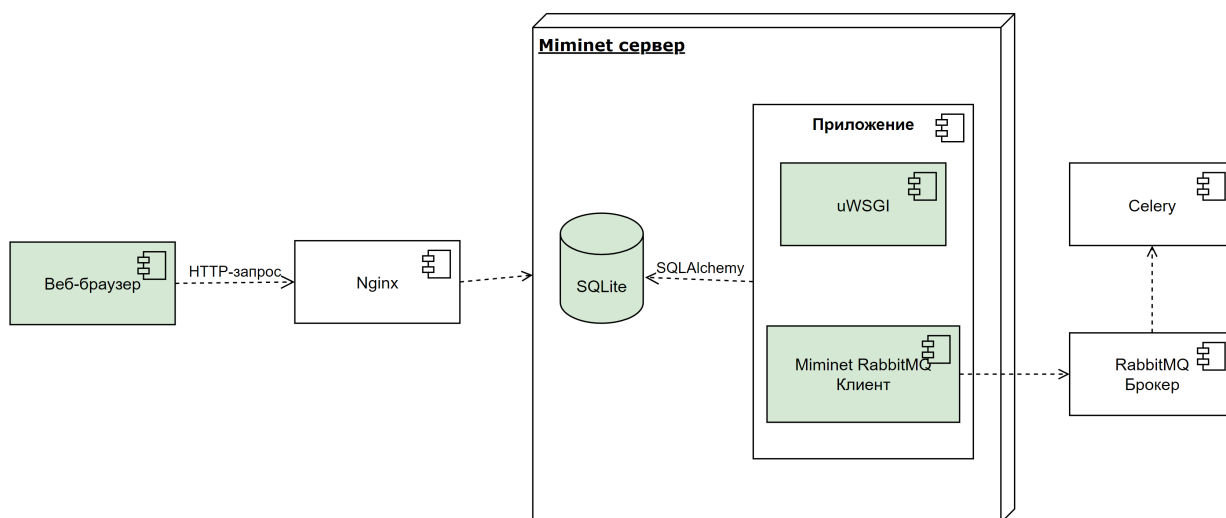


Рис. 4: Диаграмма компонентов Miminet

библиотеки SQLAlchemy. RabbitMQ — брокер сообщений, управляющий запросами на эмуляцию сетей. Модуль Celery обрабатывает запросы из очереди — с помощью библиотеки IPMininet, запущенной на виртуальной машине, происходит эмуляция сетей.

4.3. Используемые технологии

Miminet разрабатывается командой больше года, поэтому все используемые технологии были выбраны до начала данной работы.

- **Flask** [10] — легковесный и гибкий фреймворк для создания веб-приложений и API на языке **Python**. Основными преимуществами фреймворка являются простой синтаксис и наличие шаблонов, что позволяет быстро создавать веб-приложения. В контексте создания тренажера для Miminet использовался для создания конечных точек обработки HTTP-запросов, а также для взаимодействия с базой данных.
- **Jinja2** [11] — шаблонизатор для языка **Python**, интегрированный с Flask. Шаблоны Jinja интегрируются в HTML и позволяют встраивать переменные и логику прямо в код страниц, упрощая их создание и обслуживание.

- **JavaScript** [12] — используется для улучшения интерактивности веб-страниц. В частности, он используется вместе с **AJAX**¹ для динамической загрузки контента без перезагрузки страницы, обеспечивая более отзывчивый пользовательский опыт.
- **jQuery** [13] — это библиотека JavaScript, используемая в Miminet для упрощения работы с DOM (Document Object Model) [14] элементами, управления событиями и обеспечения кроссбраузерной совместимости. Благодаря jQuery управление клиентской стороной веб-приложения становится более эффективным и удобным.
- **Bootstrap** [15] — это библиотека CSS и JavaScript, используемая для создания веб-интерфейсов. Инструментарий предоставляет готовые компоненты и стили для быстрой разработки приложений, обеспечивая единообразный и современный дизайн.
- **SQLite** [16] — лёгкая встраиваемая СУБД, используемая для хранения и управления данными.
- **SQLAlchemy** [17] — библиотека для работы с базами данных на языке Python, предоставляющая ORM². Используется для упрощения взаимодействия с реляционной СУБД.

¹Asynchronous Javascript and XML — метод создания интерактивных веб-интерфейсов, позволяющий обмениваться данными между браузером и сервером без полной перезагрузки веб-страницы, что увеличивает скорость и удобство веб-приложений.

²Object-Relational Mapping

5. Особенности реализации

Работа над созданием тренажера велась в паре с другим студентом, его часть работы описана в тексте «Разработка тренажера для Miminet».

В данном разделе будут описаны этапы проектирования и реализации тренажера.

5.1. Проектирование

При проектировании тренажёра сначала были определены типы вопросов и практических задач. Затем было решено, как именно преподаватель будет формулировать практические задачи, а также был создан прототип интерфейса тренажёра.

5.1.1. Типы заданий

На основании обзора были определены следующие типы теоретических заданий:

- с вариантами ответов,
- сортировка,
- сопоставление.

Для практических заданий круг возможных условий ограничивается командами эмулятора. На данный момент в Miminet доступны следующие команды:

- ping (1 пакет),
- ping (с опциями),
- traceroute (с опциями),
- отправить данные (TCP/UDP),
- добавить маршрут,

- добавить запись в ARP-cache.

Примеры возможных практических заданий:

- ping (1 пакет) — необходимо настроить сеть с заранее заданной конфигурацией (возможно, пустой) так, чтобы пакет с одного хоста отправлялся на другой.
- ping (с кольцом) — отличается от предыдущего задания тем, что помимо работы ping проверяется маршрут движения пакетов (в одну и другую сторону должны быть различны).

Для каждой команды возможно установить различные исходные конфигурации сети, что обеспечивает формирование обширного набора задач. Кроме того, список доступных типов команд будет дополняться.

Для первой версии тренажера была реализована поддержка задачи с условием ping (1 пакет), которая проверяет путь ICMP-сообщений от одного хоста к другому.

5.1.2. Практические задачи

Было необходимо определить наилучший способ задания практических задач. Рассматривались два варианта:

1. Декларативный
2. Императивный

В первом подходе пользователь выбирает тип условия из списка и заполняет все параметры, необходимые для этого условия. Такой подход удобен и прост для пользователя, а также лёгок в реализации, однако он ограничивает варианты возможных задач и требует постоянной поддержки.

С другой стороны, в императивном подходе пользователь напрямую пишет код на конкретном языке программирования, описывая необходимые условия. Этот код выполняется после эмуляции сети и возвращает результат проверки. Такой подход даёт неограниченные возможности в

создании задач, однако может быть небезопасен и сложен как в реализации, так и для пользователя.

Преимущество императивного подхода в его больших возможностях нивелируется ограниченностью числа доступных команд платформы Miminet. Поэтому, учитывая ещё и простоту использования, был выбран декларативный подход.

5.1.3. Макет интерфейса

Для более полного понимания, каким должен быть интерфейс приложения, с помощью сервиса Figma [18] был разработан макет (рисунок 5). Он постранично визуализирует структуру и дизайн приложения, включая расположение элементов управления, внешний вид различных компонентов и их взаимодействие. Создание макета позволило получить четкое представление о том, как приложение будет выглядеть и функционировать, а также оценить удобство его использования.

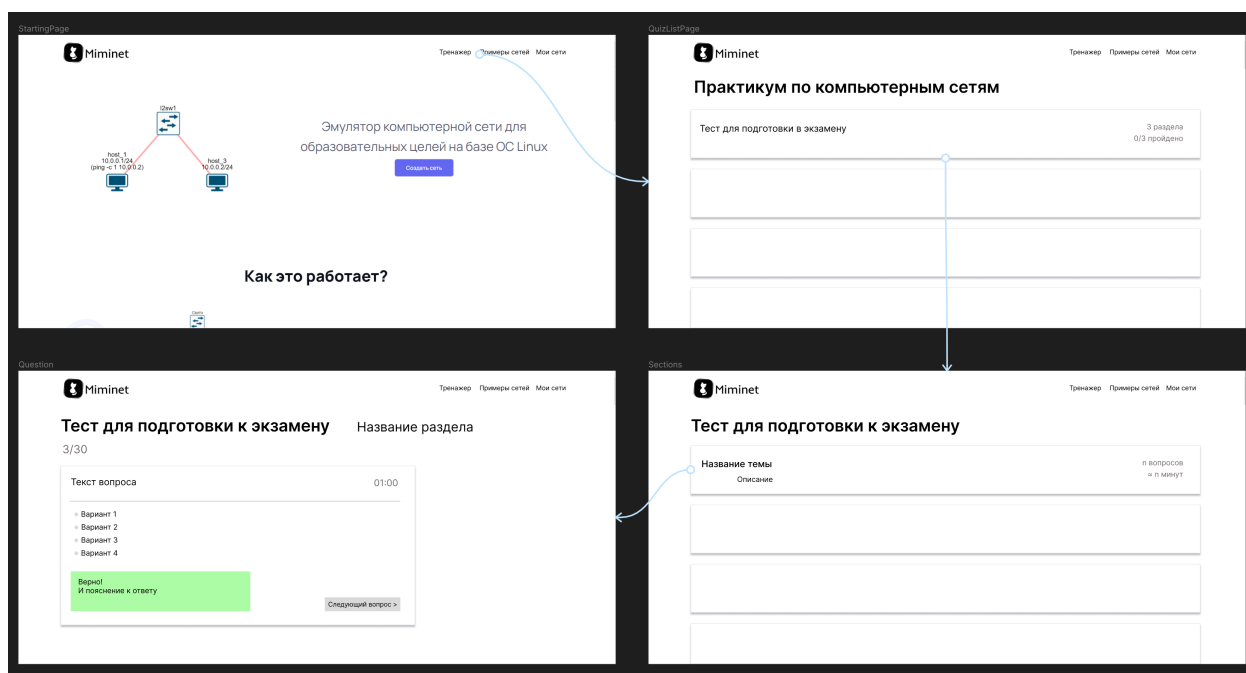


Рис. 5: Прототип интерфейса созданный в Figma

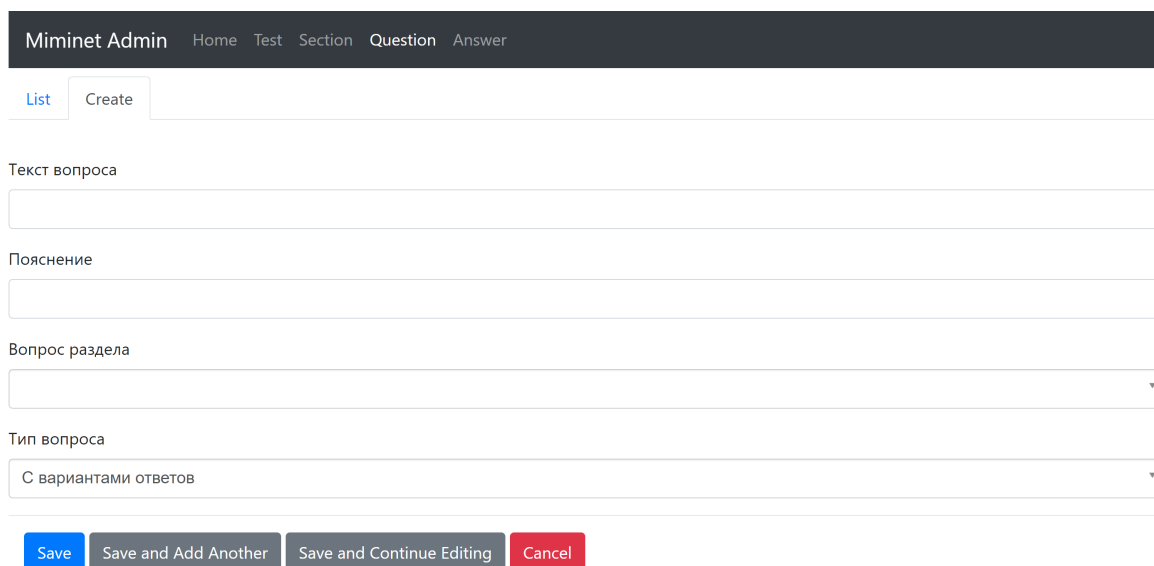
5.2. Реализация

5.2.1. Процесс создания теста преподавателем

Для облегчения управления данными приложения была создана административная панель с помощью Flask Admin [19]. Фреймворк обеспечивает быструю разработку и гибкую конфигурацию интерфейса управления данными веб-приложения.

Для создания теста пользователь на вкладке «Тест» указывает название, описание и можно ли перепроходить тест. Затем для теста создаются разделы в нём, для этого уже на вкладке «Раздел» аналогично заполняются название, описание и время на прохождение, а также к какому тесту принадлежит раздел. Теоретические вопросы тоже формируются путем заполнения формы (рисунок 6). Затем формируются ответы в зависимости от типа вопроса:

- вариант ответа, его корректность,
- вариант ответа и его позиция,
- пары вариантов ответов.



The screenshot shows the 'Miminet Admin' web interface. At the top is a dark navigation bar with links: Home, Test, Section, Question, and Answer. Below this, there are two buttons: 'List' (in blue) and 'Create' (in white). The main form area contains several input fields: 'Текст вопроса' (Question text) with a text area, 'Пояснение' (Explanation) with a text area, 'Вопрос раздела' (Section question) with a dropdown menu, and 'Тип вопроса' (Question type) with a dropdown menu showing 'С вариантами ответов' (With answer options). At the bottom of the form are four buttons: 'Save' (blue), 'Save and Add Another' (grey), 'Save and Continue Editing' (grey), and 'Cancel' (red).

Рис. 6: Форма для создания вопроса

Практическое задание представляет собой текст задания, начальную конфигурацию сети, список команд, которые необходимо выполнить, и количество устройств каждого типа, доступных для использования в решении. На данный момент практическое задание нельзя создать через административную панель, сделать это можно выполнив POST-запрос на сервер, указав все необходимые данные.

5.2.2. Процесс проверки задания

Сеть задаётся тремя массивами: *nodes*, *edges*, *packets*. *Nodes* содержит имя устройства, идентификатор, позицию отображения и соединения с другими устройствами. *Edges* хранит данные о соединениях устройств: начало, конец и идентификатор ребра. *Packets* содержит тип пакета и его путь.

Когда пользователь нажимает кнопку «Ответить», на клиенте проверяется, проэмулирована ли сеть. Если нет, то вызывается эмуляция сети. После завершения процесса эмуляции отправляется POST-запрос с массивами *packets*, *nodes*, *edges* на сервер, где происходит проверка. Программа проходит по массиву *packets* и в зависимости от типа задания проверяет соответствующие пакеты: их наличие, отправитель и получатель, сам путь пакета.

5.2.3. Страницы тренажера

В ходе работы неоднократно вносились изменения и доработки в тренажёр и его интерфейс, основанные на замечаниях пользователей и команды. Первая версия тренажера была запущена в марте 2024 (рисунки 7, 8).

Основы компьютерных сетей

Общие сведения, физический и канальный уровень

Самопроверка знаний по общим сведениям компьютерных сетей, а также по физическому и канальному уровню.

21 вопрос

15 минут

[Начать прохождение](#)

TCP/IP

Набор вопросов и практических заданий на проверку знаний по стеку TCP/IP

9 вопросов

30 минут

[Начать прохождение](#)

Рис. 7: Страница теста в первой версии тренажера

Основы компьютерных сетей

Общие сведения, физический и канальный уровень

1/4 00:29:31

×

Кто стоял у основ технологии коммутации пакетов?

- ☐ Стив Джобс и (независимо) Билл Гейтс
- ☐ Ларри Робертс и Боб Тейлор
- ☐ Александр Грэхем Белл и Томас Алва Эдисон
- ☒ Пол Барен и Леонард Клейнрок

Верно!

Ученый Леонард Клейнрок в 1960-х создал математическую теорию коммутации пакетов, которая лежит в основе технологии интернета.

[Следующий вопрос](#)

Рис. 8: Страница с вопросом в первой версии тренажера

Далее дизайн интерфейса был изменен. На странице теста убрана кнопка «Начать прохождение», теперь для начала тестирования необходимо нажать на название раздела (рисунок 9). Добавлено отображение результата последнего прохождения теста: кнопка со ссылкой на последний результат. Для старых сессий, которые были созданы до добавления страницы с результатами, предусмотрено текстовое отображение результата.

Основы компьютерных сетей

Общие сведения, физический и канальный уровень

Темы: маршрутизаторы и т.д.

1/4 верно

30 минут

Практические задания

0/1 верно

30 минут

Рис. 9: Обновленная страница теста

Для тестов, которые нельзя перепройти, было добавлено модальное окно с предупреждением. Перед началом теста пользователю сообщается, что пройти его можно будет только один раз (рисунок 10).

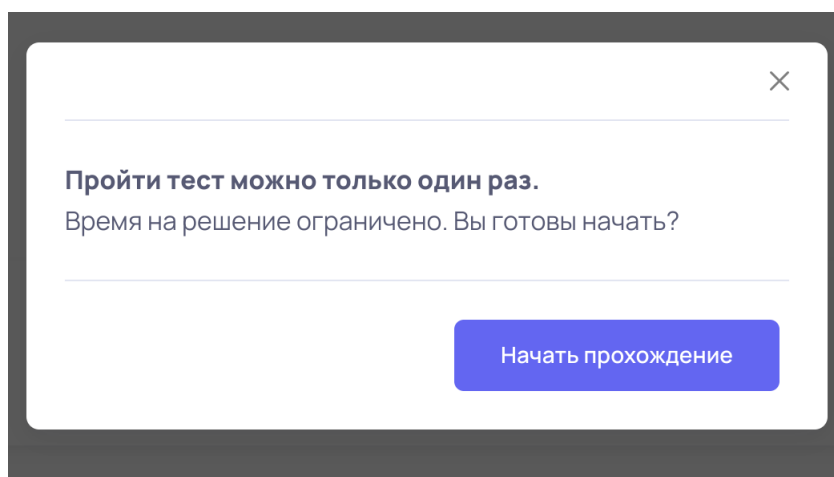


Рис. 10: Окно с предупреждением о том, что тест нельзя перепройти

На странице тестирования уменьшена карточка вопроса, кнопка «Ответить» перенесена влево для более удобного использования, также изменен дизайн отображения пояснения (рисунок 11).

1/4 00:29:55



Кто стоял у основ технологии коммутации пакетов?

- ☐ Стив Джобс и (независимо) Билл Гейтс
- ☐ Ларри Робертс и Боб Тейлор
- ☒ Пол Бэрен и Леонард Клейнрок
- ☐ Александр Грэхем Белл и Томас Алва Эдисон

Верно! Следующий вопрос

Ученый Леонард Клейнрок в 1960-х создал математическую теорию коммутации пакетов, которая лежит в основе технологии интернета.

Рис. 11: Обновленная страница с вопросом

5.2.4. Поддержка языка разметки

В процессе использования тренажера появилась потребность в более гибком отображении заданий. Для этого было решено добавить поддержку языка разметки Markdown (пример, рисунок 12) с использованием библиотеки Showdown, а также подсветка кода с помощью библиотеки highlight.js.

7/20 00:19:05



Какие числа будут напечатаны?

```
for i in range(3, 7):  
    if not i % 2:  
        continue  
    print(i)
```

- ☐ 4, 5, 6
- ☐ 3, 5, 7
- ☐ 4, 6
- ☐ 3, 5

Ответить

Рис. 12: Пример вопроса с кодом

5.2.5. Результаты теста

Если тренажёр будет использоваться как платформа для тестирования, то преподавателю также необходимо видеть результаты прохождения теста обучающимися.

Доступ к результатам осуществляется по ссылке, уникальной для каждой сессии. Для входа не требуется авторизация.

На странице (пример, рисунок 13) отображаются название теста и раздела, количество правильных ответов, время, затраченное на прохождение теста, а также для каждого вопроса указывается, был ли дан правильный ответ.

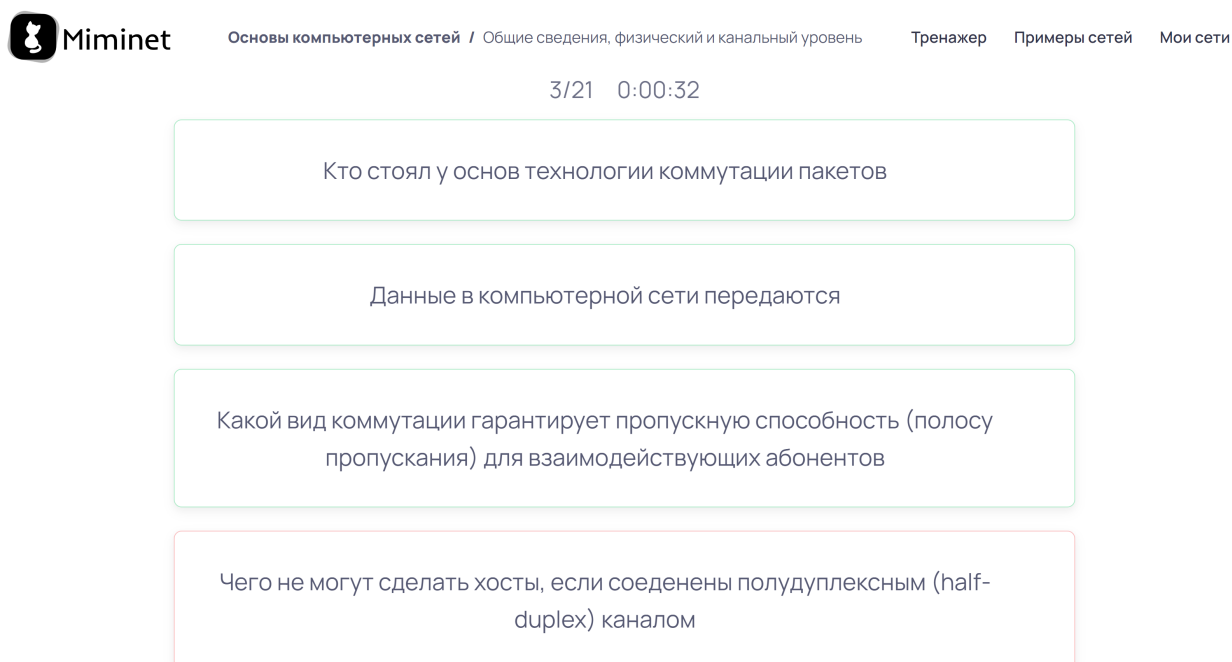


Рис. 13: Пример вопроса с кодом

6. Апробация

Для оценки удобства использования тренажера была проведена апробация по методике System Usability Scale [20]. Обучающимся было предложено пройти тесты в тренажере Miminet, а затем каждый участник заполнил анкету из 10 вопросов. В опросе участвовало 8 учащихся, общая оценка составила 90,6. Из этого можно сделать вывод о том, что интерфейс тренажера прост в использовании и понятен для нового пользователя.

С марта 2024 года у тренажера было 114 уникальных пользователей, которые совершили в общей сложности 470 сессий.

Также тренажер был использован командой KORNFELD при отборе кандидатов на стажировку YADRO Импульс³.

³Страница стажировки. Ссылка: <https://careers.yadro.com/impulse/> (дата обращения 12.05.2024)

Заключение

В ходе данной работы были достигнуты следующие результаты.

- Рассмотрены образовательные платформы, имеющие курсы с заданиями по сетям: платформы Stepik, Открытое образование, Moodle; изучены инструменты для создания практических заданий по сетям Cisco Packet Tracer, GNS3 и Boson NetSim. В результате были выявлены требования к тренажеру.
- Создана архитектура тренажера, включая типы заданий и определение задания, прототип интерфейса. Выбраны технологии для реализации: фреймворк Flask (для создания API), инструментарий Bootstrap (для быстрой разработки веб-интерфейсов с единообразным дизайном.), библиотеки jQuery (для управления DOM-элементами и событиями на клиентской стороне) и SQLAlchemy (для взаимодействия с СУБД через ORM).
- Выполнена реализация системы:
 - реализован интерфейс тренажера, создана функциональность страниц и обращения к API сервера;
 - реализована логика проверки практических задач;
 - создана административная панель, поддерживающая пользовательские роли.
- Тренажер был апробирован и интегрирован в Miminet и доступен на сайте⁴.
- Результаты работы были представлены на конференции «Современные технологии в теории и практике программирования» и опубликованы в сборнике её материалов [21].

Код доступен в GitHub репозитории⁵, имя пользователя: [ElenaBakova](#).

⁴Сайт Miminet. Ссылка: <https://miminet.ru> (дата обращения 10.05.2024)

⁵Репозиторий Miminet. Ссылка: <https://github.com/mimi-net/miminet> (дата обращения 10.05.2024)

Список литературы

- [1] СПбГУ. Веб-эмулятор Mimirnet. — 2023. — URL: <https://mimirnet.ru/> (дата обращения: 22.10.2023).
- [2] Using mininet for emulation and prototyping software-defined networks / Rogério Leão Santos De Oliveira, Christiane Marie Schweitzer, Ailton Akira Shinoda, Ligia Rodrigues Prete // 2014 IEEE Colombian conference on communications and computing (COLCOM) / Ieee. — 2014. — P. 1–6.
- [3] Макаров Игорь Николаевич, Маркина Полина Николаевна, Савинова Анна Джумберовна. Опыт использования образовательной платформы Stepik при создании онлайн-курса // Актуальные проблемы совершенствования высшего образования. — 2020. — P. 195–197.
- [4] Открытое образование главная страница. — 2024. — URL: <https://openedu.ru/> (дата обращения: 24.02.2024).
- [5] Милованов Максим Михайлович. Разработка системы управления образовательным процессом для системы дистанционного обучения Moodle // Программные системы и вычислительные методы. — 2015. — no. 2. — P. 145.
- [6] Vasantha Raju NNSH, Harinarayana NS. Online survey tools: A case study of Google Forms // National conference on scientific, computational & information research trends in engineering, GSSS-IETW, Mysore. — 2016.
- [7] Javid Sheikh Raashid. Role of packet tracer in learning computer networks // International Journal of Advanced Research in Computer and Communication Engineering. — 2014. — Vol. 3, no. 5. — P. 6508–6511.
- [8] Dayanand Lal N, Ghorbani Behnam, Vaghri Solmaz. A survey on the use of GNS3 for virtualizing computer networks. — 2016.

- [9] Киселева Елизавета Александровна. Обзор программных средств для обучения специалистов в области сетевых технологий // Постулат. — 2020. — по. 8 август.
- [10] Фреймворк Flask. — 2023. — URL: <https://flask.palletsprojects.com/en/3.0.x/> (дата обращения: 18.12.2023).
- [11] Шаблонизатор Jinja2. — 2023. — URL: <https://jinja.palletsprojects.com/en/2.10.x/> (дата обращения: 18.12.2023).
- [12] Документация по JavaScript. — 2024. — URL: <https://developer.mozilla.org/en-US/docs/Web/JavaScript> (дата обращения: 24.02.2024).
- [13] Библиотека jQuery. — 2023. — URL: <https://jquery.com/> (дата обращения: 20.12.2023).
- [14] Document Object Model. — 2023. — URL: <https://dom.spec.whatwg.org/> (дата обращения: 20.12.2023).
- [15] Официальный сайт инструментария Bootstrap. — 2024. — URL: <https://getbootstrap.com/> (дата обращения: 24.02.2024).
- [16] Официальная страница SQLite. — 2024. — URL: <https://www.sqlite.org/> (дата обращения: 24.02.2024).
- [17] Официальная страница SQLAlchemy. — 2024. — URL: <https://www.sqlalchemy.org/> (дата обращения: 24.02.2024).
- [18] Хвостенко Татьяна Михайловна, Великсар Дмитрий Сергеевич. Figma-перспективный инструмент современного веб-дизайнера // Вестник образовательного консорциума Среднерусский университет. Информационные технологии. — 2019. — по. 2. — Р. 7–10.
- [19] Flask Admin. — 2024. — URL: <https://flask-admin.readthedocs.io/en/latest/> (дата обращения: 18.02.2024).

- [20] Klug Brandy. An overview of the system usability scale in library website and system usability testing // Weave: Journal of Library User Experience. — 2017. — Vol. 1, no. 6.
- [21] Е. З. Бакова Д. В. Луцев. Проектирование и разработка тренажёра на базе веб-сервиса Miminet // Современные технологии в теории и практике программирования: Сборник материалов научно-практической конференции студентов, аспирантов и молодых ученых. — 2024. — Р. 10–11.