Федеральное государственное автономное образовательное учреждение  
высшего образования   
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ   
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет информатики, математики и компьютерных наук   
НИУ ВШЭ-Нижний Новгород  
Образовательная программа «Программная инженерия»

**Курсовая работа**

на тему «Разработка информационной системы для регистрации на мероприятия НИУ ВШЭ – Нижний Новгород»

по направлению подготовки бакалавров 09.03.04 «Программная инженерия»

Выполнила:   
студентка группы 20ПИ-1

Титова Надежда Дмитриевна

Руководитель:

доцент каф. ИСиТ

Асеева Наталья Владимировна

Нижний Новгород 2023

**Аннотация.** Данная работа выполнена в рамках группового курсового прикладного проекта, ставящего своей целью разработку регистрационной системы для мероприятий факультета информатики, математики и компьютерных наук нижегородского кампуса НИУ ВШЭ. В рамках данной пояснительной записки подробно рассматривается разработка требований к системе, разработка и реализация дизайна продукта, включая постановку частного технического задания, обоснованный выбор средств разработки, описание разработанных дизайна и прототипа и, собственно, frontend-части регистрационной системы, а также тестирование полученной системы.

**Ключевые слова**: регистрационная система.

Оглавление

[Введение 4](#_gjdgxs)

[1. Проектирование системы 7](#_30j0zll)

[1.1 Разработка требований 7](#_1fob9te)

[1.2 Средства и технологии разработки 14](#_3znysh7)

[2. Описание программного продукта 15](#_2et92p0)

[2.1 Дизайн и прототип 15](#_tyjcwt)

[2.2 Реализация страниц системы 19](#_3dy6vkm)

[3. Тестирование программного продукта 22](#_1t3h5sf)

[Заключение 28](#_4d34og8)

[Источники 29](#_2s8eyo1)

[Приложения 31](#_17dp8vu)

# Введение

В настоящее время информатизация настолько глубоко проникла в повседневную жизнь общества, что даже люди, не связанные непосредственно со сферой информационных технологий, очень во многих аспектах жизни полагаются на компьютеризированные решения. Наличие всевозможных систем получения, хранения и обмена информацией делает мгновенно доступным то, на что вживую пришлось бы тратить заметно больше времени, и таким образом значительно упрощает и рационализирует жизненные процессы – как с точки зрения человека-клиента, желающего получить необходимую ему услугу без привязки к времени и месту, так и для предоставляющей эту услугу стороны, которой не приходится тратить ресурсы на обработку потоков информации вручную [17].

Одной из многочисленных задач, требующих подобного автоматизированного решения, является регистрация на мероприятия. Многие учреждения и сегодня, устраивая мероприятия, поручают сотрудникам работу по подсчету участников, сбору и систематизации их данных, которая вполне может производиться автоматически. Такое решение сделало бы участие приятнее и для гостей, ведь намного удобнее в текущем ритме жизни иметь возможность регистрироваться на мероприятия через Интернет, сразу получая информацию о доступности мест и прочих организационных моментах. Необходимость в подобном решении возникла и у нижегородского кампуса Высшей школы экономики, где мы обучаемся. Университет регулярно проводит открытые мероприятия как для студентов, так и для широкой аудитории – заинтересованных в поступлении школьников, а также просто всех желающих; при этом единой системы регистрации у кампуса не существует, и по необходимости каждое подразделение ищет собственные способы сбора информации с привлечением сторонних инструментов, не всегда понятных и удобных. Именно в необходимости рационализировать и автоматизировать этот процесс, чтобы упростить организационную работу факультетов и улучшить впечатления посетителей, и заключается актуальность нашей работы.

Таким образом, целью её стала разработка системы, которая могла бы использоваться как для регистрации на проводимые мероприятия, так и для управления ими со стороны преподавателей и менеджеров факультета. Для обеих сторон взаимодействия основными критериями, которые планировалось достичь, были универсальность и гибкость, а также простота и наглядность в использовании.

Основными этапами, пройденными по мере выполнения нами работы, стали:

* сбор требований от руководства и их формализация в виде технического задания для удовлетворения существующей необходимости в продукте;
* обзор существующих на рынке систем аналогичного назначения и их сравнение с нашим предполагаемым продуктом;
* планирование и распределение ролей в проекте, принимающее во внимание предыдущий опыт работы и дальнейшее желание развиваться;
* выбор инструментов и технологий разработки в соответствии с поставленными целями;
* выстраивание архитектуры и пошаговая разработка обоих аспектов продукта (серверной и клиентской части);
* интеграция проделанной работы, тестирование и финализация прототипа системы;
* ретроспектива и анализ соответствия проделанной работы изначальным планам;
* подготовка к будущему внедрению продукта в реально используемые сервисы НИУ ВШЭ.

Так как данная работа выполнялась в группе, то описанные выше задачи были разделены между участниками. Выполненная мною часть включает разработку и реализацию дизайна системы (фронтенд, frontend). В ней будет конкретизировано обрисованное ранее техническое задание, обоснован выбор технологий и инструментов для разработки и выбор дизайна, и описана основная структура кода и программных компонентов связанной с ними части продукта.

# 1. Проектирование системы

## 1.1 Разработка требований

В настоящее время существуют разнообразные подходы к разработке требований и управлению ими, однако большая часть подходов совпадает в одном: процесс составления и анализа требований включает следующие этапы:

* выявление требований
* анализ требований
* спецификация требований
* проверка требований

Для выявления требований выделяют [20] следующие подходы: интервьюирование, анкетирование, наблюдение за производственной деятельностью, создание прототипов системы, анализ существующей нормативной документации к системе, анализ предыдущих версий системы. В результате анализа данных подходов и применимости их к нашей задаче, было решено выбрать для выявления требований интервьюирование. Наш выбор обусловлен тем, что документации к регистрационным системам факультета, как и их предыдущих версий, не существует - до текущего момента для этой задачи использовались Google Forms; для наблюдения за производственной деятельности необходимо было бы проследить процесс организации мероприятий факультета, их планирования и регистрации на них, что в целом реально, но затратно по времени и не слишком удобно; интервьюирование же снабдило бы нас всей необходимой информацией, поскольку заказчик в лице руководства факультета был готов подробно прокомментировать все ожидания от работы системы и ответить на все возникающие у нас вопросы. Также в дальнейшем для уточнения требований использовалось прототипирование системы, о чем подробнее рассказывается в пункте 2.1.

После проведения интервью и анализа полученных ответов мы приступили к созданию спецификаций.

Поскольку в разработке требований к программному обеспечению выделяется несколько уровней требований: бизнес-требования, пользовательские требования и функциональные требования, было решено сформулировать все эти виды требований на начальном этапе работы.

В первую очередь были выделены бизнес-требования как требования первой степени важности в разработке любого продукта. Была сформулирована следующая цель: создание единой системы регистрации, которая могла бы использоваться для регистрации школьников на мероприятия факультета ИМиКН, доступной в использовании людям без специфических знаний в программировании.

Далее мы провели работу со следующим уровнем требований - пользовательскими требованиями. Нами было выделено два сегмента наших потенциальных пользователей: организаторы и участники мероприятий.

Для пользователей-организаторов мы выделили следующие пользовательские требования к системе:

1. Система должна быть удобна в использовании пользователями, не обладающими специфическими знаниями в IT, например, менеджерам учебных офисов.
2. Система должна либо иметь отдельный интерфейс для организаторов событий, либо уметь обрабатывать данные в виде xlsx-файлов с четко заданным форматом заполнения.
3. Организатор события должен иметь возможность создавать события, заполнять необходимую информацию о них, создавать и настраивать расписание мероприятий, редактировать события, просматривать и редактировать список участников события и мероприятий, и удалять события.

Также нами были выделены следующие пользовательские требования, касающиеся непосредственно участников событий:

1. При использовании системы для регистрации на мероприятие у среднестатистического пользователя старшего школьного возраста не должно возникать проблем с пониманием работы системы и необходимой для регистрации последовательности действий.
2. Участник события должен иметь возможность заполнить личную информацию о себе, выбрать событие для посещения и выбрать мероприятия события для посещения.
3. При завершении регистрации участник должен иметь возможность скачать файл со списком выбранных им мероприятий и их расписанием.

После разработки пользовательских требований, спустившись еще на один уровень требований ниже, мы перешли к написанию функциональных требований к системе. Ниже приведен полученный список функциональных требований.

1. По ссылке, размещенной на стороннем ресурсе (например, на сайте ВШЭ или в соцсетях), открывается веб-страница событий системы.
   1. Страница содержит форму регистрации, которую пользователю предлагается заполнить. Список полей следующий:
      1. ФИО
      2. Место обучения
      3. Контактный телефон
      4. Адрес электронной почты для связи
   2. На странице также представлен список событий, предлагаемых пользователю.
      1. Для каждого из событий указан следующий список свойств:
         1. Название
         2. Аннотация
         3. Даты проведения (доступен выбор дней посещения мероприятия)
         4. Место проведения
      2. Каждое событие также имеет кнопку «Зарегистрироваться».
         1. Кнопки становятся активными, когда форма регистрации заполнена корректно и выбран для посещения хотя бы один день события (при наличии свободных мест на событие).
2. При нажатии любой из кнопок «Зарегистрироваться» открывается страница выбора мероприятий данного события.
   1. На странице представлен список мероприятий, доступных для посещения.
   2. Мероприятия сгруппированы по дню и времени проведения.
   3. Для каждого из мероприятий указан следующий список свойств:
      1. Название
      2. Аннотация
      3. Место проведения
      4. Ведущий
      5. Количество доступных мест
      6. Если данное мероприятие дублируется в разное время в рамках данного события, указывается также уведомление об этом.
   4. Каждое мероприятие также имеет отметку о возможности регистрации (radio button).
      1. Мероприятия без доступных мест недоступны для регистрации (в состоянии disabled), все остальные - доступны.
      2. При выборе мероприятия также становятся недоступными для выбора те, что совпадают с ним по времени.
   5. Страница содержит кнопку “Завершить регистрацию”.
      1. При нажатии кнопки введенные пользователем анкетные данные и выбранные мероприятия сохраняются в системе.
      2. При нажатии кнопки также формируется документ с информацией о выбранных мероприятиях.
3. После обработки анкеты пользователь переходит на страницу с сообщением об успешности или неуспешности регистрации.
   1. Страница содержит сообщение о том, успешно ли прошла регистрация.
   2. На странице представлен список мероприятий, которые пользователь выбрал для посещения (расписание).
   3. Страница содержит кнопку “Скачать расписание в формате .csv”.
      1. При нажатии на кнопку загружается файл с расписанием, сформированный ранее (2.5.2).
   4. Страница содержит кнопку “Вернуться к началу”.
      1. При нажатии на кнопку пользователь возвращается на страницу выбора события.
4. Информация о заявках хранится и обрабатывается в базе данных системы.
   1. Отправленные пользователями заявки обрабатываются системой и учитываются следующим образом:
      1. Подсчитывается количество занятых мест.
      2. Формируются списки имён участников для оформления необходимых документов (списков на вход, сертификатов об участии и т.д.).
      3. Формируются списки контактов для рассылки информации о мероприятии.
5. Администраторы системы имеют возможность управлять списками имеющихся событий и мероприятий.
   1. Управление элементами этих двух списков происходит через панель администрирования, где администратор может их просматривать, редактировать и удалять.

После разработки бизнес-требований, а также пользовательских и функциональных требований, остался вид требований, еще нами не рассмотренный - нефункциональные требования. В рамках работы с нефункциональными требованиями мы определили общий список целей, которых хотели бы добиться реализованным продуктом, с точки зрения восприятия пользователями и полезности для них:

* наглядность – расположение и цветовое решение элементов интерфейса должны делать использование удобным и не требующим чрезмерных умственных затрат от пользователей, включая тех, кому не приходилось ранее работать с ним;
* информативность – предоставляемая системой информация должна как можно полнее удовлетворять запросам пользователей;
* рациональность – система должна быть достаточно эффективной и корректно работающей, чтобы воспользоваться ей было быстрее, корректнее и комфортнее, чем достичь той же цели каким-либо методом из использовавшихся ранее;
* гибкость – система должна иметь возможность подстраиваться под различные нужды организации мероприятий;
* портативность – продукт должно быть легко интегрировать в уже используемые факультетом сервисы и вообще в другие окружения, потенциально способные пригодиться;

Для разработки дизайна системы, которая является частью данной работы, были выделены дополнительные требования, образовавшие следующий список.

1. Система должна быть реализована в соответствии с действующим брендбуком НИУ ВШЭ [1], с использованием шрифтов HSE Sans и HSE Slab [3] и цветовых палитр НИУ ВШЭ [4].
2. Для всех элементов, использующихся на страницах системы (текстовые поля, поля ввода, кнопки, чекбоксы и прочее), должен быть разработан собственный дизайн.
3. Все предполагаемые действия пользователя в рамках использования системы должны сопровождаться подсказками и комментариями на странице.

## 1.2 Средства и технологии разработки

Для разработки интерфейса и прототипа системы был выбран сервис Figma. Этот выбор продиктован удобством и популярностью использования сервиса в разработке подобных продуктов [18], наглядностью и интуитивно понятным интерфейсом, возможностью как разработки элементов дизайна, так и создания прототипа, наличием многопользовательского режима редактирования (что полезно при работе над групповым проектом, которым является данная курсовая работа), возможностью экспортировать разработанные объекты в разных форматах (например, в формате css), а также предшествующим опытом работы с данным сервисом.

Для реализации разработанного дизайна были выбраны HTML и CSS благодаря их широкой популярности в подобных проектах, доступности в освоении и существованию множества подробных руководств и примеров использования.

# 2. Описание программного продукта

## 2.1 Дизайн и прототип

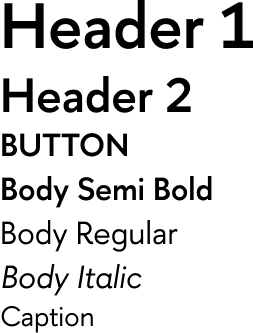
Итак, для разработки интерфейса и прототипа системы был выбран сервис Figma.

В процессе работы с Figma с оглядкой на обозначенные выше функциональные и нефункциональные требования были выполнены [5] следующие задачи:

1. Создание текстовых и цветовых стилей для проекта.

После загрузки и рассмотрения шрифтов НИУ ВШЭ, для дальнейшей работы был выбран шрифт HSE Sans [3] как более подходящий для использования на электронных носителей (легкий для восприятия, без засечек). На основе данного шрифта после анализа различных его начертаний были разработаны [7] следующие текстовые стили (рис.1):

* Body Regular (начертание Regular, размер 16px, для использования в текстах, которые не требуют акцентов)
* Body Italic (начертание Italic, размер 16px, курсив)
* Body Semi Bold (начертание Semi Bold, размер 16px, для расставления акцентов в текстовых блоках и подзаголовков)
* Caption (начертание Regular, размер 14px, для подсказок пользователю и текстов, на которых не должно сосредотачиваться внимание)
* Header 1 (начертание Semi Bold, размер 32px, для главных заголовков)
* Header 2 (начертание Semi Bold, размер 24px, для заголовков меньшего размера и значимости, чем при использовании Header 1)
* Button (начертание Semi Bold, размер 16px, Uppercase, для кнопок)

  
рис. 1. Разработанные текстовые стили

Также была выбрана цветовая палитра системы (рис.2), за основу взяты рекомендованные брендбуком НИУ ВШЭ цветовые палитры [4] - основная палитра университета, и палитра для подразделений, связанных с точными науками, к которым относится Факультет информатики, математики и компьютерных наук. Из не упомянутых в палитрах цветов также использовались черный и белый цвета. Также в качестве акцентного красного цвета для сообщений об ошибках был взят красный цвет из палитры для другого подразделения, поскольку, как и все цвета, представленные в брендбуке, он хорошо сочетается с базовой цветовой палитрой.

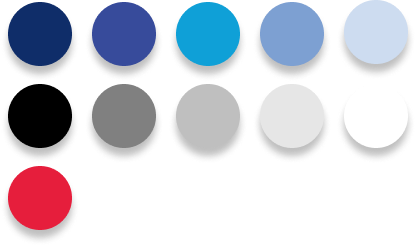


рис. 2. Разработанная цветовая палитра

1. Разработка компонентов

При разработке компонентов интерфейса системы были реализованы [8] следующие компоненты:

* Поля ввода текста (содержат заголовок с названием поля, и само поле ввода, существующее в нескольких состояниях - активное, неактивное, заполненное, пустое (внутри placeholder), ошибочное)
* Кнопки (двух типов - primary и secondary - для ярких и акцентных кнопок, являющихся главными на странице, и второстепенных кнопок, которые не должны акцентировать на себе внимание; каждая кнопка существует в двух состояниях - enabled и disabled)
* Чекбоксы (в нескольких состояниях - невыбранный, выбранный, недоступный для выбора)
* Макеты для вывода мероприятий и элементов расписания мероприятия (представляют собой фреймы с совокупностью вышеперечисленных компонентов, созданные для уточнения формата вывода)

1. Отрисовка экранов

Для разработки прототипа системы регистрации необходимым было разработать макеты экранов с примером содержимого для каждой из трех страниц в различных состояниях (в зависимости от заполнения различных блоков). В числе прочего, было реализовано 14 экранов для первой страницы и 7 для второй. В качестве примера содержимого для макета было взято расписание Зимней школы математики и компьютерных наук “Интеллектуал-2022”, проводившейся на факультете в декабре 2022 года, как подробно демонстрирующее особенности расписаний мероприятий такого рода (например, наличие нескольких потоков, повторяемость мероприятий и распределение участников на небольшие подгруппы для посещения обедов). Полученные экраны можно увидеть на странице “Экраны” прикрепленного документа в Figma [5] (рис. 3).

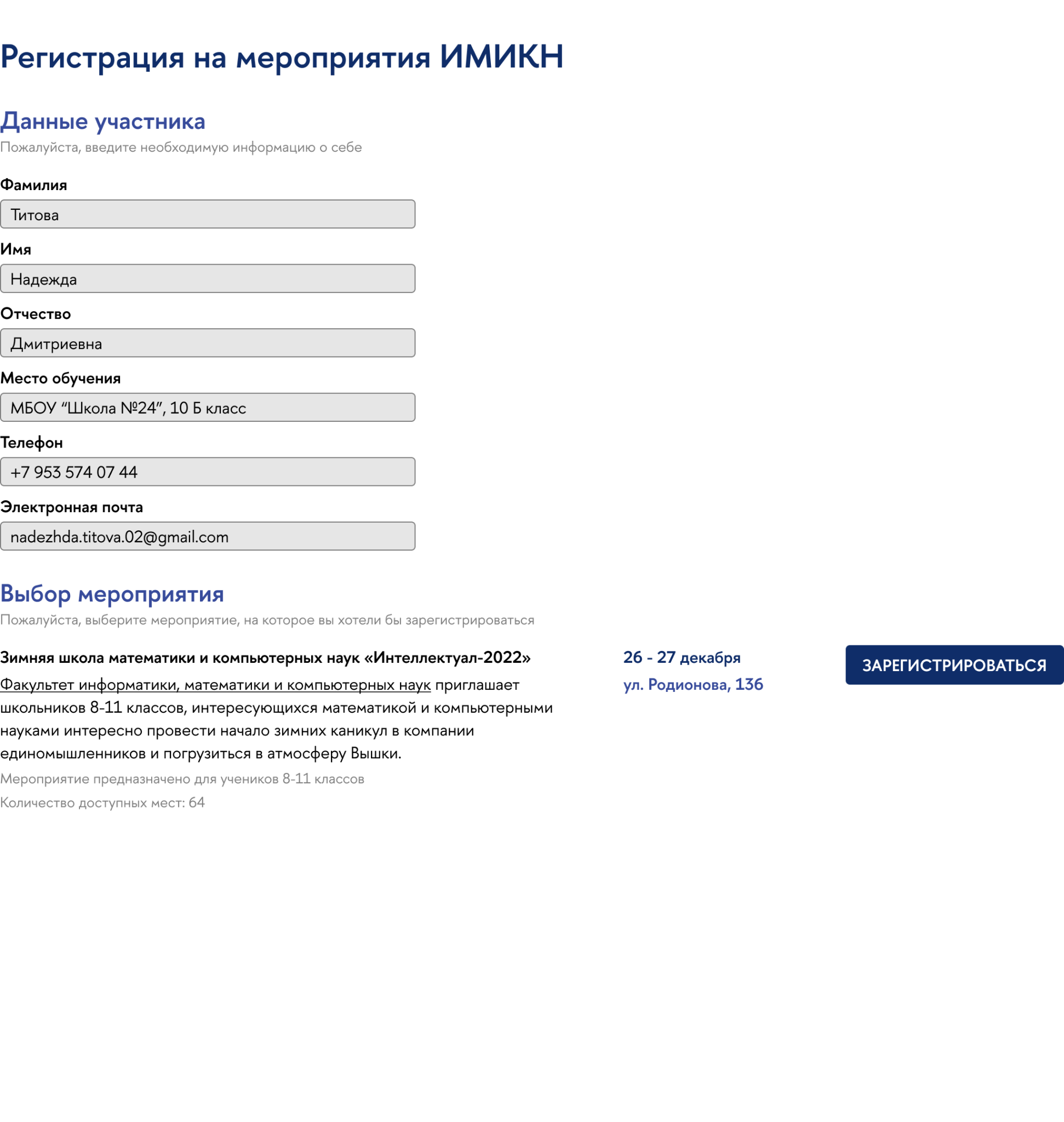


рис. 3. Пример реализованного в Figma макета экрана

1. Создание прототипа

При использовании Руководства по созданию прототипов в Figma [9] был реализован прототип продукта [6]. Данный прототип позволяет интерактивно взаимодействовать с разработанным интерфейсом и получить представление о работе и дизайне в дальнейшем реализованной регистрационной системы.

## 2.2 Реализация страниц системы

После детальной проработки дизайна и создания прототипа стало возможным перейти к реализации интерфейса в виде HTML кода с использованием CSS.

Непосредственно касаются frontend-части системы регистрации следующие файлы, с которыми и велась работа (все файлы доступны на странице github с исходным кодом [10]):

* register.html - страница ввода личных данных пользователя и выбора мероприятия для посещения
* timetable.html - страница настройки расписания посещения
* completed.html - страница подтверждения регистрации
* style.css - файл с реализацией стилей проекта, который используют все html-файлы

HTML-файлы собраны в директории regsys/templates/regsys/, CSS-файл хранится в regsys/static/regsys/.

Также задействована директория regsys/static/regsys/fonts/, в которой хранятся различные начертания шрифта HSE Sans в форматах woff2, woff, otf и eof. Такое многообразие форматов продиктовано необходимостью работы конечного продукта в различных браузерах [11]. Так, сначала система проверяет, установлен ли шрифт локально, если нет - пытается подгрузить шрифт в различных форматах в вышеуказанном порядке, пока не найдет подходящий браузеру, в котором система запущена (рис. 4).



рис. 4. Реализация подключения одного из шрифтов

Почти все ключевые моменты дизайна интерфейса отражены в css-файле. С помощью правила font-face подключены все необходимые шрифты, далее с их помощью воспроизведены приведенные выше текстовые стили регистрационной системы. При использовании определенного перечня списка источников [12][13][14] стилизованы поля ввода, кнопки и чекбоксы (рис. 5). Также была предпринята попытка стилизовать системные radio button, но из-за проблем в реализации от этой идеи пришлось отказаться.

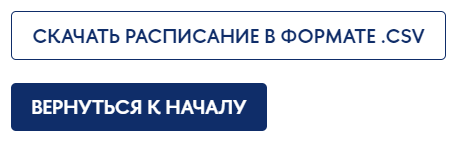
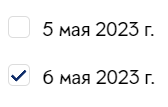


рис. 5. Примеры стилизованных элементов

HTML-файлы проекта были созданы на основе шаблонов django [15], после добавления в них всей необходимой информации страницы и связывания с прочими файлами проекта все компоненты страницы были стилизованы, шрифты текстов изменены на соответствующие текстовым стилям, а также был изменен формат вывода - информация на странице отцентрирована [16], отступы приведены к соответствующему виду.

# 3. Тестирование программного продукта

В рамках работы над курсовым проектом также было проведено тестирование полученного продукта. В тестировании нашего продукта, как и в тестировании большинства продуктов такого рода, можно выделить следующие этапы: модульное тестирование, интеграционное тестирование и системное тестирование. Поскольку пока мы имеем дело с прототипом будущей регистрационной системы, о приемочном тестировании и тестировании производительности пока говорить не приходится, они будут проведены на других этапах работы над проектом. Если кратко сформулировать сущность различных этапов тестирования, модульное тестирование подразумевает проверку отдельных частей базы кода, интеграционное тестирование - проверку различных сегментов кода сайта в виде независимых функций или модулей, системное тестирование - тестирование сайта на уровне пользовательского интерфейса и функционала, проверку корректности совместной работы различных элементов сайта [19].

В целом, в рамках работы над проектом тестированию и его автоматизации уделялось не так много внимания, поэтому внушительная часть работы по тестированию реализована вручную.

Далее будут подробно разобраны этапы тестирования нашей регистрационной системы.

1. Модульное тестирование.

В рамках работы над проектом было реализовано исключительно ручное модульное тестирование. Это обусловлено тем, что масштабы нашего сервиса достаточно малы для того, чтобы автоматизированное тестирование здесь казалось целесообразным. Модульное тестирование проводилось на этапе разработки отдельных элементов системы. Тестирование проводилось путем ручной проверки работоспособности и корректности отдельных модулей в зависимости от их функционала. Из-за многочисленности файлов в структуре проекта приводить далее набор тестов для каждого из них не кажется рациональным, однако хочется отметить, что, если при работе с файлами backend-части проекта речь скорее шла сразу об интеграционном тестировании - трудно представить работу отдельного элемента без сопутствующих ему, то при тестировании непосредственно веб-страниц, элементов html и css тестирование сводилось к проверке корректности работы полей ввода и вывода, и корректности отображения компонентов после их стилизации. Ниже в качестве примера представлен список требований, на соблюдение которых проверялся файл completed.html при работе с ним в процессе тестирования:

* Страница завершения регистрации доступна успешно открывается.
* Элементы страницы не накладываются друг на друга.
* Элементы страницы соответствуют заявленным цветовым и текстовым стилям.
* На странице отображается сообщение об успешной регистрации на событие.
* На странице отображается имя, введенное пользователем при регистрации.
* На странице представлен выбранный пользователем список мероприятий для посещения с указанием времени и места проведения, а также имени ведущего мероприятия.
* На странице есть кнопки скачивания расписания типа secondary и возврата к началу типа primary.
* При нажатии кнопки “Скачать расписание в формате csv” скачивается csv-файл, содержимое которого соответствует настроенному пользователем расписанию.
* При нажатии кнопки “Вернуться к началу” происходит переход на страницу выбора события.

Разумеется, не все тесты, выполнявшиеся на этапе модульного тестирования, выполнялись успешно, что помогало найти недочеты в коде и понять природу ошибок, однако хочется заметить, что все файлы финальной версии проекта успешно прошли модульное тестирование.

1. Интеграционное тестирование

Интеграционное тестирование также проводилось пока исключительно в ручном формате. Проверялась работоспособность системы после обновления отдельных модулей, например, при изменении любого файла серверной части производился запуск сервера, и проверялся следующий функционал системы:

* Работают страницы администрирования системы - доступна авторизация администратора, просмотр, создание, редактирование и удаление им отдельных полей.
* Прогружаются и работают все три страницы, доступные участникам мероприятий: страница ввода личных данных и выбора события для регистрации, страница настройки расписания и страница завершения регистрации.
* Доступны для заполнения все поля ввода в системе, введенная информация успешно сохраняется на сервере.
* Корректно работает навигация между страницами системы.

В дальнейшей работе над проектом (поскольку пока мы реализовали исключительно прототип системы, дальнейшая работа над проектом подразумевается) планируется использование Github Actions, как очень удобной платформы для интеграционного тестирования с учетом того, что весь процесс разработки в рамках данной курсовой работы ведется с использованием Github.

1. Системное тестирование

Во время системного тестирования для автоматизации монотонной работы ввода данных и проверки корректности их обработки, а также корректности работы сервиса в целом было принято решение использовать Selenium IDE (в виде расширения для Google Chrome). Этот выбор обусловлен интуитивно-понятным интерфейсом системы, возможностью записи действий пользователя и широкой популярностью сервиса в тестировании функциональности и корректности веб-сервисов. В результате работы с Selenium IDE путем записи действий пользователя было создано несколько тестов следующих типов:

* Тесты с корректным заполнением полей ввода (рис. 6, 7).

В данных тестах проверяется работа системы при самом ожидаемом, последовательном и корректном наборе действий пользователя: пользователь открывает страницу ввода данных и выбора события (страница регистрации); заполняет поля ФИО, места обучения и контактных данных корректными значениями; выбирает одну или больше дат для посещения одного из доступных для регистрации мероприятий; нажимает кнопку “Зарегистрироваться”; переходит на страницу настройки расписания; выбирает мероприятия для посещения; нажимает на кнопку завершения регистрации; переходит на страницу завершения регистрации; скачивает свое расписание; возвращается к началу.

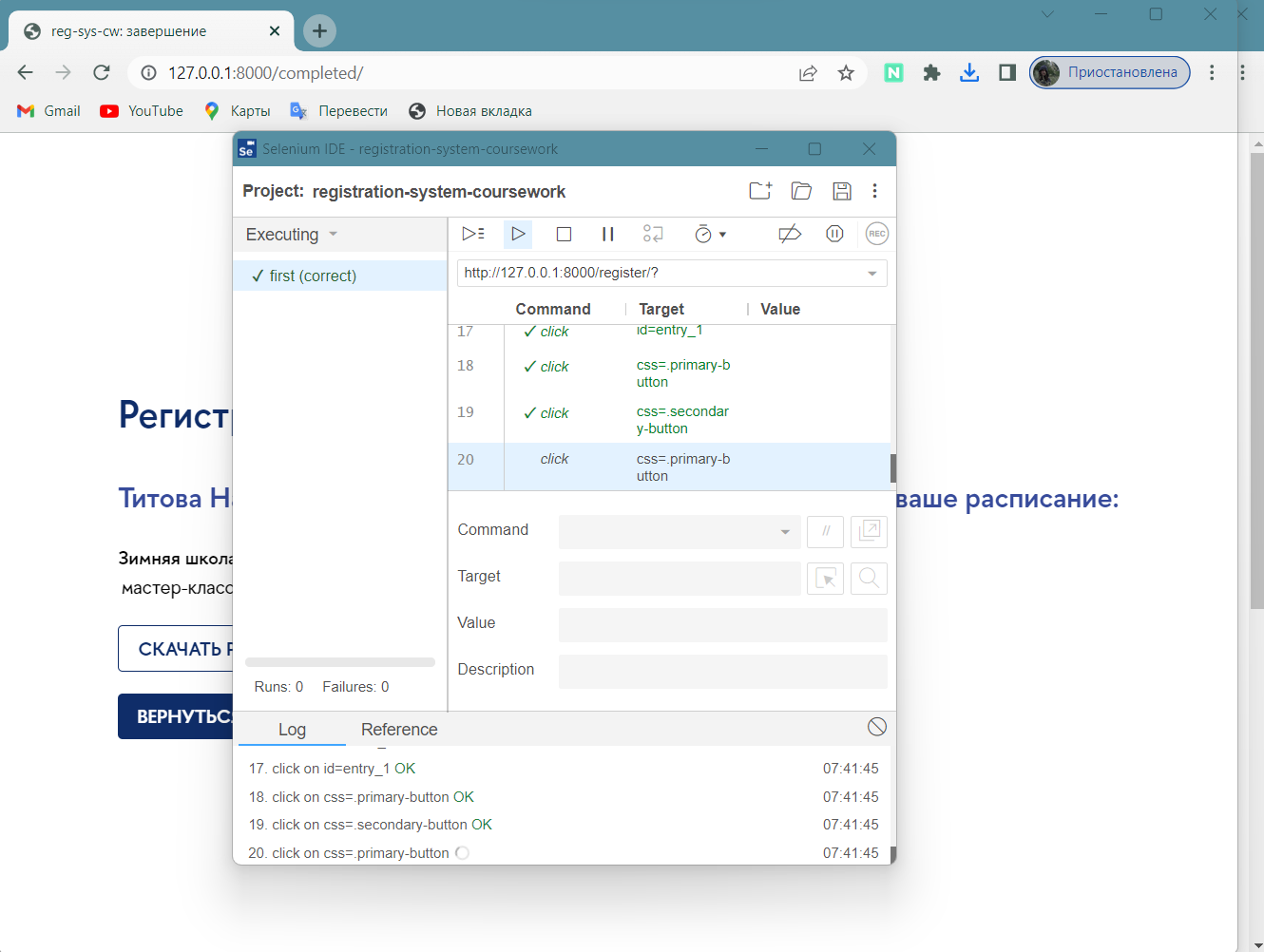


рис. 6. Использование Selenium IDE в системном тестировании

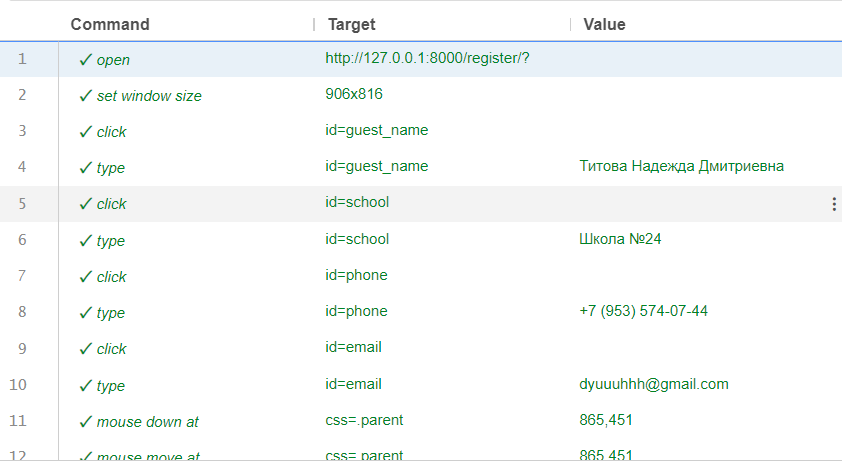


рис. 7. Использование Selenium IDE в системном тестировании

* Тесты с некорректным заполнением полей ввода.

В целом, тесты данной группы работали аналогично тестам, описанным в предыдущем пункте, и включали примерно те же шаги. Однако выполняющий их абстрактный пользователь совершал различные ошибки при работе с системой - заполнял поля невалидными значениями, оставлял поля пустыми, пытался зарегистрироваться на мероприятия, регистрация на которые закрыта, а также перейти на страницы настройки расписания и завершения регистрации, минуя страницу ввода личных данных.

Тесты последней группы позволили нам оценить не только корректность взаимодействия элементов системы и работоспособность системы в целом, но и корректность обработки исключений и ввода ошибочных данных, а также поведение системы в нестандартных ситуациях. Так, с помощью тестов этой группы были также осуществлены пограничное тестирование и стресс-тестирование.

Данные тесты регулярно запускались на этапе разработки интерфейса и дизайна - для уточнения корректности работы и отображения всех элементов после незначительных изменений в коде.

Также в рамках системного тестирования были проверены время отклика системы (в пределах нормального) и корректность работы системы в разных версиях разных браузеров, таких как Google Chrome, Yandex Browser, Microsoft Edge и Mozilla Firefox.

Разработанная система прошла все написанные нами тесты, и по результатам тестирования нашей системы мы можем судить о том, что она работает корректно. В процессе тестирования готового продукта мы сверились со списком требований, установленных нами в начале работы над проектом, и убедились, что регистрационная система удовлетворяет всем установленным требованиям, что позволяет судить об успешности ее реализации.

# Заключение

Подводя итог деятельности в рамках данной курсовой работы, хочется отметить, что полученный продукт удовлетворяет всем поставленным функциональным требованиям и техническому заданию. Все цели, ставившиеся в начале работы над проектом, успешно достигнуты.

Реализован прототип системы регистрации на мероприятия факультета, который в дальнейшем (например, после интеграции на сайт университета) может быть использован при организации различного рода мероприятий.

При работе над курсовым проектом был получен новый опыт разработки и реализации дизайна и интерфейса системы, а также опыт разработки требований и тестирования реального продукта. Хочется отметить, что это также первый для авторов проект, который получит какое-либо реальное применение, что влияет на специфику реализации.

В дальнейшем планируется продолжение работы над проектом, более обширное тестирование системы и развитие ее из прототипа в реально использующийся продукт. Таким образом, мы можем судить о том, что у данного проекта есть потенциал для дальнейшего развития.

Весь исходный код проекта, а также вся документация по проекту представлены в github-репозитории [10].

# Источники

[1] Обновленный стиль Вышки (актуальный брендбук НИУ ВШЭ). 2022. <https://www.hse.ru/info/brandbook>

[2] Руководство по использованию фирменного стиля НИУ ВШЭ на русском языке. 2022. <https://www.hse.ru/info/brandbook#guide>

[3] Шрифты НИУ ВШЭ. <https://www.hse.ru/info/brandbook#font>

[4] Цветовые палитры НИУ ВШЭ. [https://www.hse.ru/info/brandbook#palette](https://www.hse.ru/info/brandbook#font)

[5] Дизайн системы в Figma. 2023. <https://www.figma.com/file/Kt2FDXipEGxZJmpCemKc7s/Registration-system?t=3hfefkVQvIIGJvIm-6>

[6] Прототип системы в Figma. 2023. <https://www.figma.com/proto/Kt2FDXipEGxZJmpCemKc7s/Registration-system?page-id=1%3A2&node-id=6-95&viewport=120%2C127%2C0.14&scaling=scale-down&starting-point-node-id=6%3A95&show-proto-sidebar=1>

[7] Styles in Figma. 2023. <https://help.figma.com/hc/en-us/articles/360039238753-Styles-in-Figma>

[8] Guide to components in Figma. 2023. <https://help.figma.com/hc/en-us/articles/360038662654-Guide-to-components-in-Figma>

[9] Guide to prototyping in Figma. 2023. <https://help.figma.com/hc/en-us/articles/360040314193-Guide-to-prototyping-in-Figma>

[10] Репозиторий с исходным кодом проекта. 2023. <https://github.com/yualapshina/registration-system-coursework>

[11] Подключение шрифтов в CSS. 2020. <https://snipp.ru/html-css/font-face>

[12] Стилизация старого доброго элемента button. 2020. <https://habr.com/ru/companies/ruvds/articles/489820/>

[13] Стилизация текстовых полей формы. 2021. <https://itchief.ru/html-and-css/styling-text-input>

[14] Стилизация флажков и переключателей с использованием CSS3. 2020. <https://habr.com/ru/articles/489476/>

[15] The Django template language. 2023. <https://docs.djangoproject.com/en/4.1/ref/templates/language/>

[16] Выравниваем блок по центру страницы. 2014. <https://habr.com/ru/articles/238449/>

[17] Горбунова О.Н. Информатизация общества и формирование трудового ресурса: проблемы, пути решения. Социально-экономические явления и процессы. 2012.

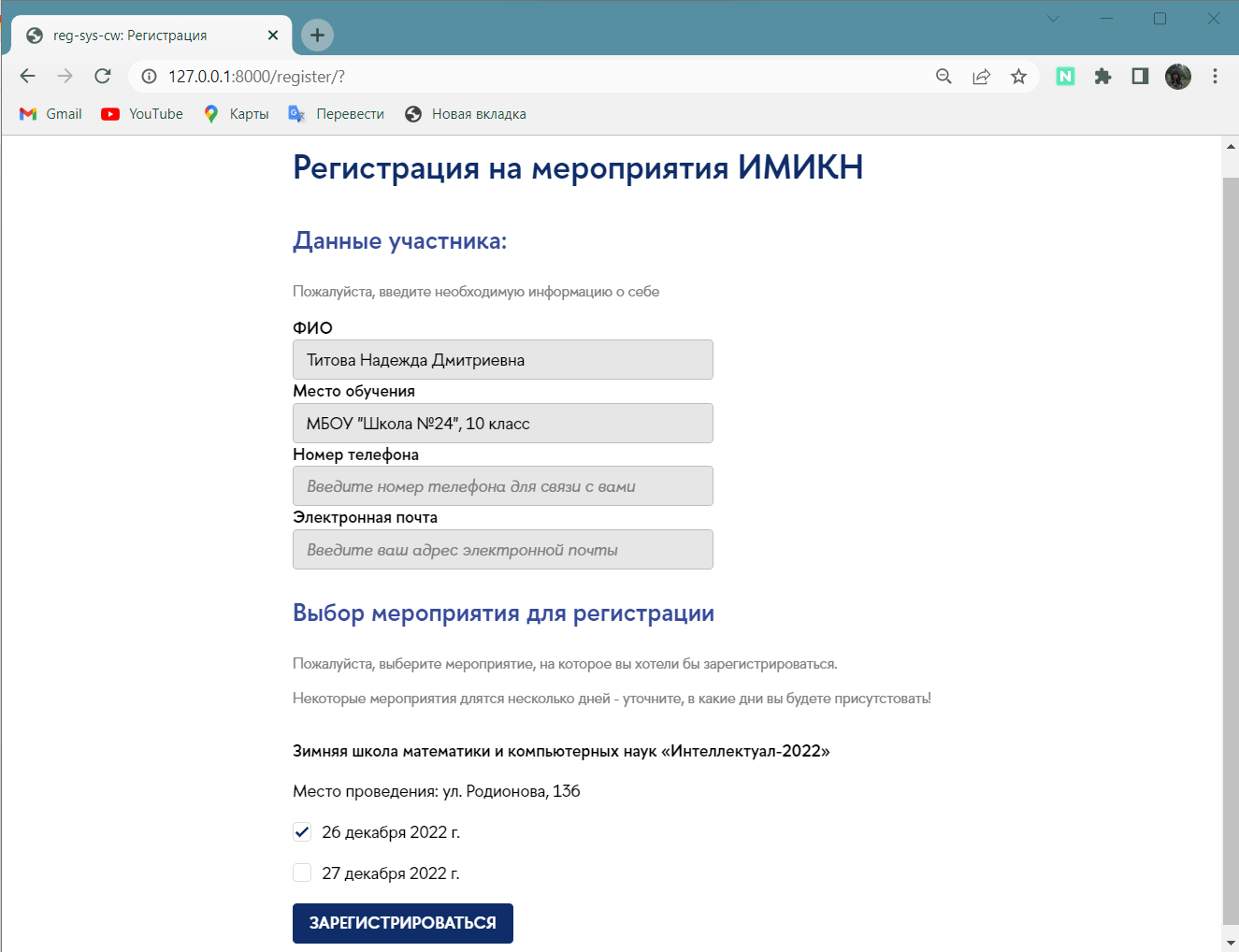
[18] Разбор UI/UX на примере прототипа в Figma и основные принципы. Инструменты для создания динамических прототипов. Плюсы и минусы Figma. 2020. <https://habr.com/ru/articles/508028/>

[19] Основы тестирования веб-приложений. 2022. <https://habr.com/ru/companies/ruvds/articles/676752/>

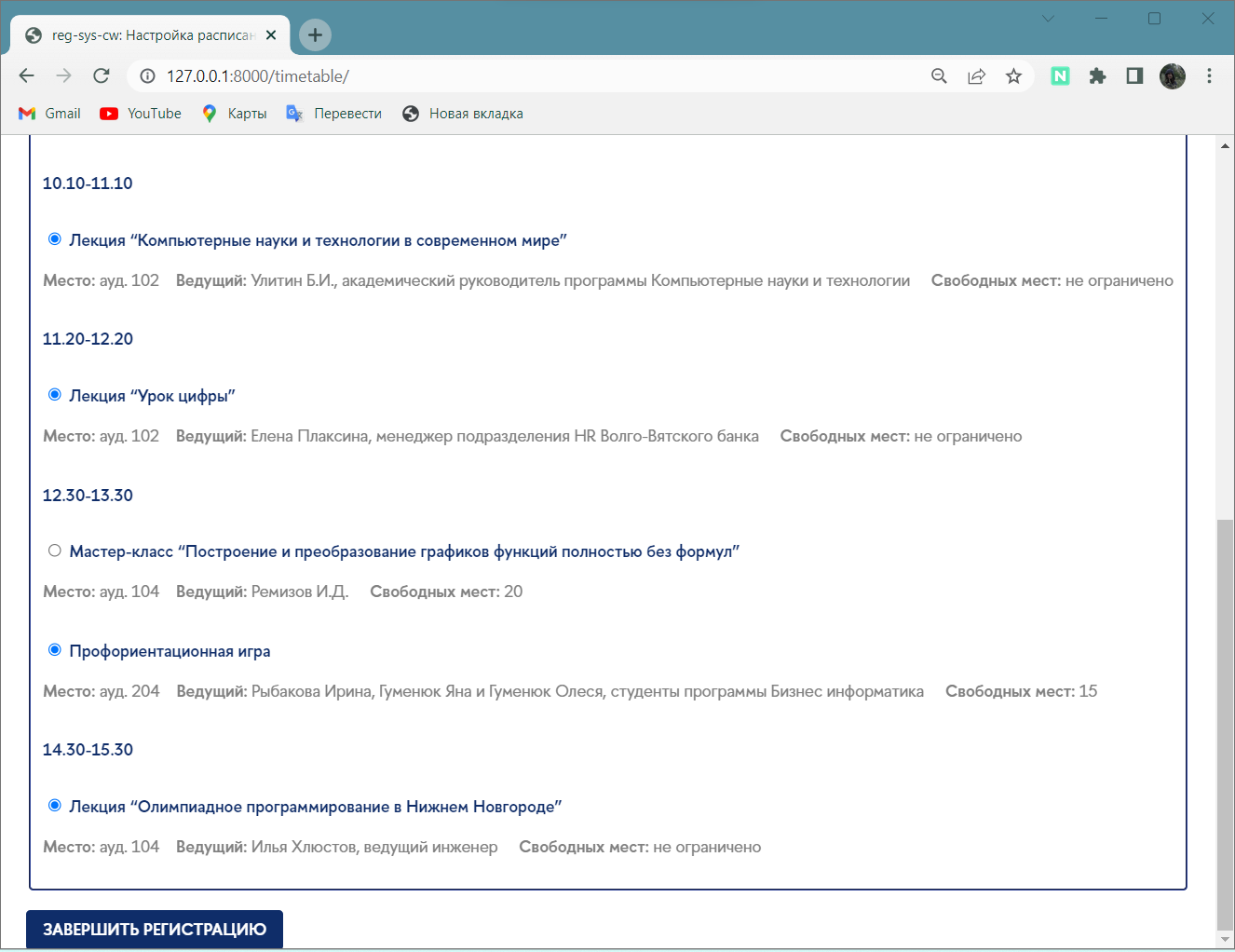
[20] Григораш Виталий. Управление качеством требований. Методы выявления требований. <https://studfile.net/preview/4532891/page:8/>

[21] Лапшина Юлия. Разработка информационной системы для регистрации на мероприятия НИУ ВШЭ – Нижний Новгород. 2023. <https://lms.hse.ru/ap_service.php?getwork&guid=97ECA030-62FD-4A6A-9687-5CD30F6BDD3C>

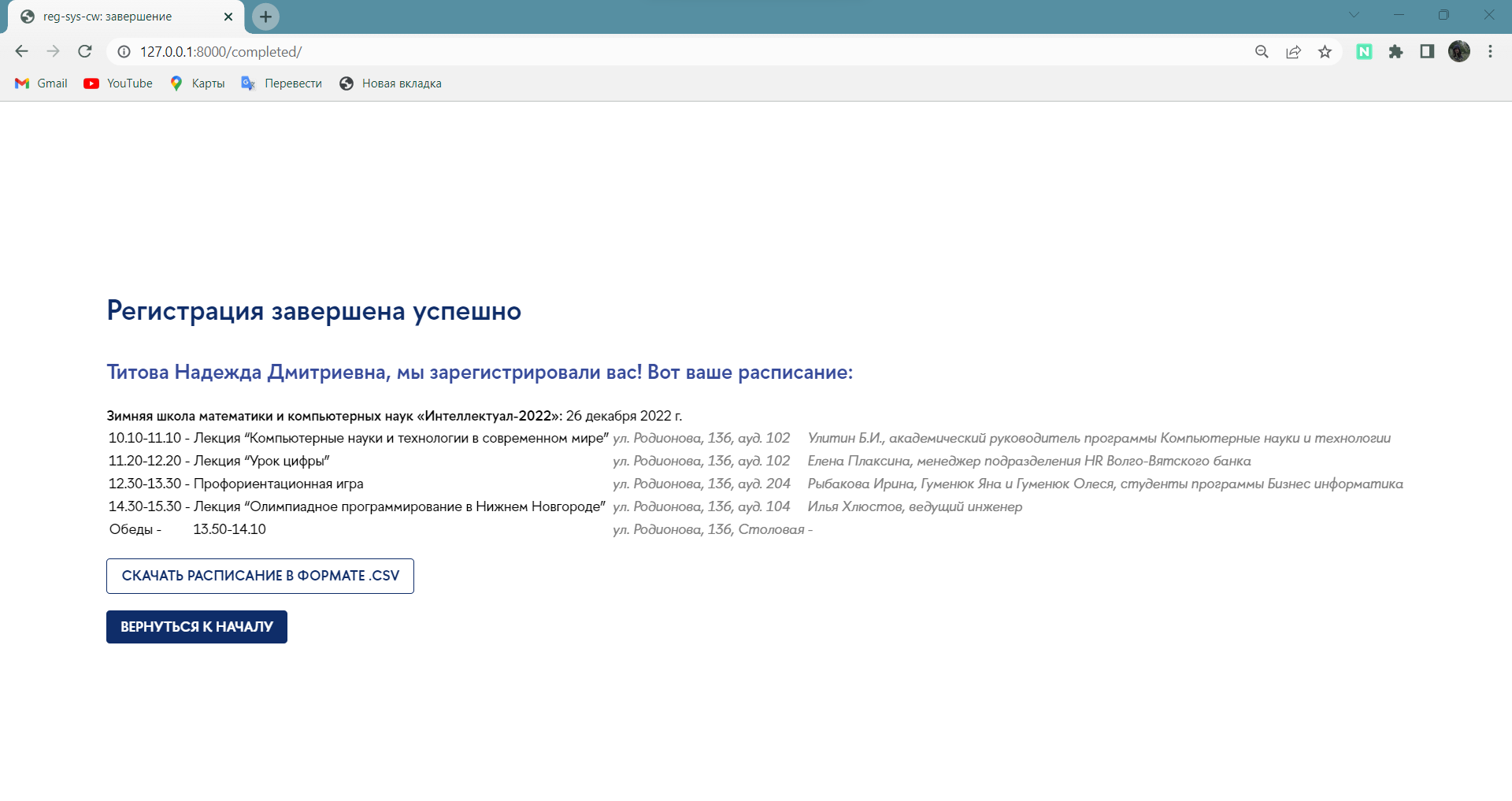
# Приложения



Приложение 1. Страница ввода личных данных и выбора мероприятия для регистрации.



Приложение 2. Часть страницы настройки расписания.



Приложение 3. Страница завершения регистрации.