Министерство образования Республики Беларусь

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Факультет прикладной математики и информатики

Кафедра компьютерных технологий и систем

**СИСТЕМА ДОПОЛНЕННОЙ РЕАЛЬНОСТИ ДЛЯ АНАЛИЗА ПОСЛЕДОВАТЕЛЬНОСТИ СНИМКОВ**

Курсовой проект

Захарчени Вадима Валерьевича

студента 4 курса 4 группы,

специальность

«информатика»

Научный руководитель:

доцент, кандидат технических наук

Недзьведь Александр Михайлович

Минск, 2017

**Аннотация**

Захарченя В.В. Исследование возможностей дополненной реальности для анализа последовательности снимков: Курсовой проект / Минск: БГУ, 2017. – 27 с.

Исследуются задачи программного средства, его структура, способы проверки заданий, организация обмена данными. Рассматривается разграничение доступа пользователей, а также технологии для реализации.

**Анатацыя**

Захарчэня В.В. Даследаванне магчымасцяў дапоўненай рэальнасці для аналізу паслядоўнасці здымкаў: Курсавы праэкт / Мінск: БДУ, 2017. - 27 с.

Даследуюцца задачы праграмнай сродкі, яе структура, спосабы праверкі заданняў, арганізацыя абмену дадзенымі. Разглядаецца размежаванне доступу карыстальнікаў, а таксама тэхналогіі для рэалізацыі.

**Annotation**

Zakharchenya V.V. Creating a software means of organization of issue and delivery of individual homework for check: Course project / Minsk: BSU, 2017. - 27 p.

The problem of software tools, its structure, tasks, test methods, the organization of data exchange are studied. Concurrent access users, as well as for the implementation of technology are considered.

РЕФЕРАТ

Курсовая работа, 27 с., 15 рис., 5 источников.

Ключевые слова: ДОМАШНИЕ ЗАДАНИЯ, ВИДЫ ДОМАШНИХ ЗАДАНИЙ, РАЗГРАНИЧЕНИЕ ДОСТУПА, JAVA, ВЕБ-ПРИЛОЖЕНИЕ

Объект исследования – системы взаимодействия преподавателя и студента (ученика). Предмет исследования – веб-приложение для организации взаимодействия преподавателя и студента

Цель работы – изучение системы взаимодействия преподавателя и студента (ученика) в рамках выдачи/сдачи домашних заданий. Изучение технологий для реализации соответствующего программного средства.

Методы исследования – анализ существующих систем, программирование на Java, Java EE.

Результатом является Web-приложение, организующее взаимодействие преподавателя и студента.

Областью применения является организация образовательного процесса.

РЭФЕРАТ

Курсавы праект, 27 с., 15 м., 5 крыніц.

Ключавыя словы: ДАМАШНІЯ ЗАДАННІ, ВІДЫ ХАТНЯЕ ЗАДАННЕ, РАЗМЕЖАВАННЕ ДОСТУПУ, JAVA, ВЭБ-ДАДАТАК

Аб'ект даследавання - сістэмы ўзаемадзеяння выкладчыка і студэнта (вучня). Прадмет даследавання - вэб-дадатак для арганізацыі ўзаемадзеяння выкладчыка і студэнта

Мэта работы - вывучэнне сістэмы ўзаемадзеяння выкладчыка і студэнта (вучня) у рамках выдачы / здачы хатніх заданняў. Вывучэнне тэхналогій для рэалізацыі адпаведнай праграмнай сродкі.

 Метады даследавання - аналіз існуючых сістэм, праграмаванне на Java, Java EE.

Вынікам з'яўляецца Web-дадатак, які арганізуе ўзаемадзеянне выкладчыка і студэнта.

Вобласцю ўжывання з'яўляецца арганізацыя адукацыйнага працэсу.

ESSAY

Annual project, 27 p., 15 fig., 5 sources.

Keywords: homework, homework TYPES, access control, JAVA, Web applications

Object of study – the system of interaction of the teacher and student (learner). Subject of research - the web application for the organization of interaction of the teacher and student

Purpose – to study of the interaction system of teacher and student (student) in the framework of the issuance / delivery of homework. Learning technologies for the implementation of appropriate software tools.

Study methods – analysis of existing systems, programming in Java, Java EE.

***The result*** is a Web-based application that organizes the interaction of the teacher and the student.

The field of application is the organization of educational process.

**Оглавление**

[ВВЕДЕНИЕ 6](#_Toc500927917)

[1 Существующие интерфейсы и системы, а также методы анализа изображений. 7](#_Toc500927918)

[1.1 Виртуальная реальность 7](#_Toc500927919)

[1.2 Дополненная реальность 10](#_Toc500927920)

[1.3 Аффинное преобразование 11](#_Toc500927921)

[1.4 Калибровка камер 13](#_Toc500927922)

[1.5 Устройство дополненной реальности как средство достижения поставленной задачи 15](#_Toc500927923)

[2 Постановка задачи и рассмотрение объектов исследования 18](#_Toc500927924)

[2.1 Постановка задачи 18](#_Toc500927925)

[2.2 Описание различий функционалов разных типов аккаунтов 20](#_Toc500927926)

[2.3 Нюансы процесса взаимодействия преподавателя и студента 20](#_Toc500927927)

[3 Описание используемых технологий 21](#_Toc500927928)

[3.1 Описание языка программирования 21](#_Toc500927929)

[3.2 Описание технологии создания Web-приложений 22](#_Toc500927930)

[4 Практическая реализация 26](#_Toc500927931)

[4.1 Программная реализация прототипа приложения 26](#_Toc500927932)

[ЗАКЛЮЧЕНИЕ 33](#_Toc500927933)

[Список использованных источников 34](#_Toc500927934)

ВВЕДЕНИЕ

В настоящее время мы можем наблюдать как стремительное развитие технологий в различных областях науки, так и объём информации, который необходимо обрабатывать.

С давних времён для воздействия на окружающие предметы, управления ими, человек использовал по большей части свои руки. После появления компьютера – одного из величайших изобретений человечества, принцип управления кардинально не изменился.

Но с ростом производительности компьютеров появляется необходимость увеличения потоков информации между человеком и компьютером. Как следствие, возникает потребность в новых интерфейсах взаимодействия с компьютером, и, соответственно, новых методах анализа входного потока информации и вывода результатов этого анализа в приемлемой удобной форме.

Целью данной работы является исследование возможностей систем дополненной реальности для анализа и обработки входного потока информации в виде видеоряда, последовательности изображений, а также способов вывода результатов работы.

1. Существующие интерфейсы и системы, а также методы анализа изображений.
   1. Виртуальная реальность

Виртуальная реальность — созданный техническими средствами мир (объекты и субъекты), передаваемый человеку через его ощущения: зрение, слух, обоняние, осязание и другие. Виртуальная реальность имитирует как воздействие, так и реакции на воздействие. Для создания убедительного комплекса ощущений реальности компьютерный синтез свойств и реакций виртуальной реальности производится в реальном времени.

Объекты виртуальной реальности обычно ведут себя близко к поведению аналогичных объектов материальной реальности. Пользователь может воздействовать на эти объекты в согласии с реальными законами физики (гравитация, свойства воды, столкновение с предметами, отражение и т. п.). Однако часто в развлекательных целях пользователям виртуальных миров позволяется больше, чем возможно в реальной жизни (например: летать, создавать любые предметы и т. п.).

Не следует путать виртуальную реальность с дополненной. Их коренное различие в том, что виртуальная конструирует новый искусственный мир, а дополненная реальность лишь вносит отдельные искусственные элементы в восприятие мира реального.

Системами «виртуальной реальности» называются устройства, которые более полно по сравнению с обычными компьютерными системами имитируют взаимодействие с виртуальной средой, путём воздействия на все пять имеющихся у человека органов чувств.

В настоящее время существует несколько основных типов систем, обеспечивающих формирование и вывод изображения в системах виртуальной реальности:

**Шлем / очки виртуальной реальности (HMD - display).**

Современные шлемы виртуальной реальности представляют собой скорее очки, нежели шлем, и содержат один или несколько дисплеев, на которые выводятся изображения для левого и правого глаза, систему линз для корректировки геометрии изображения, а также систему трекинга, отслеживающую ориентацию устройства в пространстве. Как правило, системы трекинга для шлемов виртуальной реальности разрабатываются на основе гироскопов, акселерометров и магнитометров. Для систем этого типа важен широкий угол обзора, точность работы системы трекинга при отслеживании наклонов и поворотов головы пользователя, а также минимальная задержка между детектированием изменения положения головы в пространстве и выводом на дисплеи соответствующего изображения (Рисунок 1.1).



Рисунок 1.1 – Пример игры, основанной на технологии виртуальной реальности

**MotionParallax3D дисплеи**

К устройствам этого типа относится множество различных устройств: от некоторых смартфонов до комнат виртуальной реальности (CAVE). Системы данного типа формируют у пользователя иллюзию объёмного объекта за счёт вывода на один или несколько дисплеев специально сформированных проекций виртуальных объектов, сгенерированных исходя из информации о положении глаз пользователя. При изменении положения глаз пользователя относительно дисплеев, изображение на них соответствующим образом меняется. Все системы данного типа задействуют зрительный механизм восприятия объёмного изображения параллакс движения (Motion Parallax). Также, в большинстве своём, они обеспечивают вывод стереоизображения с помощью стереодисплеев, задействуя стереоскопическое зрение. Системы трекинга для MotionParallax3D дисплеев отслеживают координаты глаз пользователей в пространстве. Для этого используются различные технологии: оптическая (определение координат глаз пользователя на изображении с камеры, отслеживание активных или пассивных маркеров), существенно реже - ультразвуковая. Зачастую системы трекинга могут включать в себя дополнительные устройства: гироскопы, акселерометры и магнитометры. Для систем данного типа важна точность отслеживания положения пользователя в пространстве, а также минимальная задержка между детектированием изменения положения головы в пространстве и выводом на дисплеи соответствующего изображения. Системы данного класса могут выполняться в различных форм-факторах: от виртуальных комнат с полным погружением до экранов виртуальной реальности размером от трёх дюймов.

**Виртуальный ретинальный монитор**

Устройства данного типа формируют изображение непосредственно на сетчатке глаза. В результате пользователь видит изображение, «висящее» в воздухе перед ним. Устройства данного типа ближе к системам дополненной реальности, поскольку изображения виртуальных объектов, которые видит пользователь, накладываются на изображения объектов реального мира. Тем не менее, при определённых условиях (тёмная комната, достаточно широкое покрытие сетчатки изображением, а также в сочетании с системой трекинга), устройства данного типа могут использоваться для погружения пользователя в виртуальную реальность.

Также существуют различные гибридные варианты: например, система CastAR, в которой получение корректной проекции изображения на плоскости достигается за счёт расположения проекторов непосредственно на очках, а стереоскопическое разделение - за счёт использования световозвращающего покрытия поверхности, на которую ведётся проецирование. Но пока такие устройства широко не распространены и существуют лишь в виде прототипов.

На данный момент самыми совершенными системами виртуальной реальности являются проекционные системы, выполненные в компоновке комнаты виртуальной реальности (CAVE). Такая система представляет собой комнату, на все стены которой проецируется 3D-стереоизображение. Положение пользователя, повороты его головы отслеживаются трекинговыми системами, что позволяет добиться максимального эффекта погружения. Данные системы активно используются в маркетинговых, военных, научных и других целях.

* 1. Дополненная реальность

Дополненная реальность ­– результат введения в поле восприятия любых сенсорных данных с целью дополнения сведений об окружении и улучшения восприятия информации (Рисунок 1.2).

Дополненная реальность – воспринимаемая смешанная реальность (англ. mixed reality), создаваемая с использованием «дополненных» с помощью компьютера элементов воспринимаемой реальности (когда реальные объекты монтируются в поле восприятия).

Среди наиболее распространенных примеров дополнения воспринимаемой реальности – параллельная лицевой цветная линия, показывающая нахождение ближайшего полевого игрока к воротам при телевизионном показе футбольных матчей, стрелки с указанием расстояния от места штрафного удара до ворот, «нарисованная» траектория полета шайбы во время хоккейного матча, смешение реальных и вымышленных объектов в кинофильмах и компьютерных или гаджетных играх и т. п.

Существует несколько определений дополненной реальности: исследователь Рональд Азума (англ. Ronald Azuma) в 1997 году определил её как систему, которая:

1. совмещает виртуальное и реальное;
2. взаимодействует в реальном времени;
3. работает в 3D.

В 1994 году Пол Милгром и Фумио Кисино описали континуум «виртуальность-реальность» (англ. Milgram's Reality-Virtuality Continuum) – пространство между реальностью и виртуальностью, между которыми расположены дополненная реальность (ближе к реальности) и дополненная виртуальность (ближе к виртуальности). Дополненная реальность — результат добавления к воспринимаемым как элементы реального мира мнимых объектов (обычно в качестве вспомогательной информации).

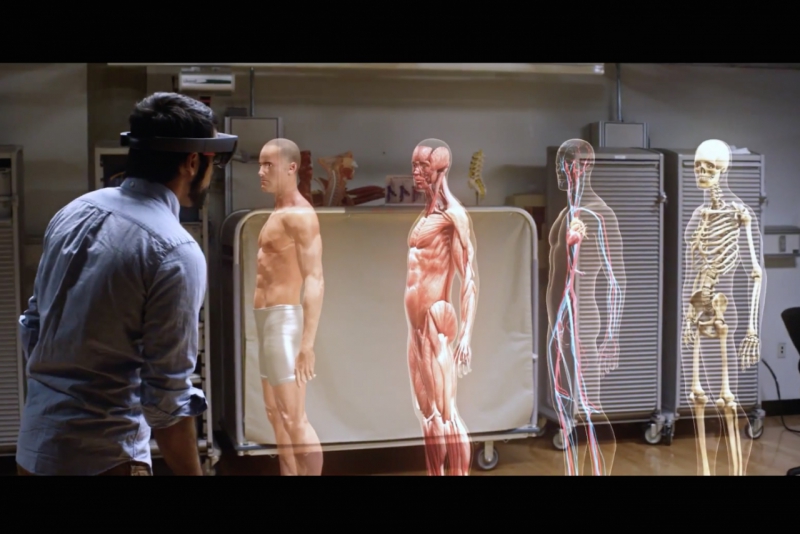


Рисунок 1.2 – Пример реализации дополненной реальности

Иногда в качестве синонимов используют термины «расширенная реальность», «улучшенная реальность», «обогащённая реальность», «увеличенная реальность». Правда, такое использование названных терминов в общем случае неправильно, — термины «расширенная реальность», «увеличенная реальность», «обогащённая реальность» применимы лишь для обозначения определённых форм и и аспектов практического применения дополненной реальности, тогда как применимость термина «улучшенная реальность» вовсе сомнительна.

* 1. Аффинное преобразование

Аффинное преобразование ­– отображение плоскости или пространства в себя, при котором параллельные прямые переходят в параллельные прямые, пересекающиеся в пересекающиеся, скрещивающиеся в скрещивающиеся.

Аффинное преобразование евклидова пространства – взаимно однозначное точечное отображение плоскости или пространства на себя, при к-ром трем точкам, лежащим на одной прямой, соответствуют три точки, также лежащие на одной прямой. Таким образом, при аффинном преобразовании прямые переходят в прямые. Аффинное преобразование плоскости переводит пересекающиеся прямые в пересекающиеся, параллельные – в параллельные. При аффинном преобразовании пространства каждая плоскость аффинно отображается на некоторую плоскость; при этом пересекающиеся плоскости переходят в пересекающиеся, параллельные – в параллельные. Кроме того, сохраняется взаимное расположение двух прямых: пересекающиеся прямые переходят в пересекающиеся, параллельные - в параллельные, скрещивающиеся – в скрещивающиеся.

Примерами аффинных преобразований являются:

* преобразование движения
* преобразование растяжения
* преобразования подобия

Движение — преобразование метрического пространства, сохраняющее расстояние между соответствующими точками. Иначе говоря, движение — это изометрия пространства в себя.

Несмотря на то, что движение определяется на всех метрических пространствах, этот термин более распространён в евклидовой геометрии и смежных областях. В метрической геометрии (в частности, в римановой геометрии) чаще говорят: изометрия пространства в себя. В общем случае метрического пространства (например, для неплоского риманова многообразия) движения могут существовать далеко не всегда.

В евклидовом (или псевдоевклидовом) пространстве движение автоматически сохраняет также углы, так что сохраняются все скалярные произведения (Рисунок 1.3).

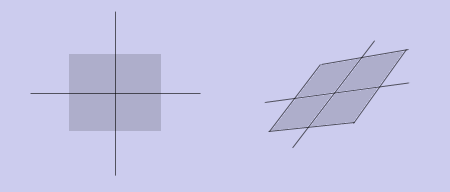


Рисунок 1.3 – Скос – объединение нескольких аффинных преобразований

Растяжение плоскости относительно оси с коэффициентом — преобразование плоскости, при котором каждая точка переходит в такую точку , что расстояние от прямой , до в раз больше, чем до точки , и проекции точек и на прямую совпадают.

Подобие — преобразование евклидова пространства, при котором для любых двух точек , и их образов , имеет место соотношение где — коэффициент подобия.

* 1. Калибровка камер

Современные видеокамеры, имеющие многолинзовые объективы (Рисунок 1.4), вносят геометрические искажения в получаемое изображение. Для задач компьютерного зрения наличие искажений нежелательно или недопустимо, поэтому требуется их устранение программным способом. Это достигается путем использования математической модели оптической системы видеокамеры и определенных экспериментально в процессе калибровки ее внутренних и внешних параметров.

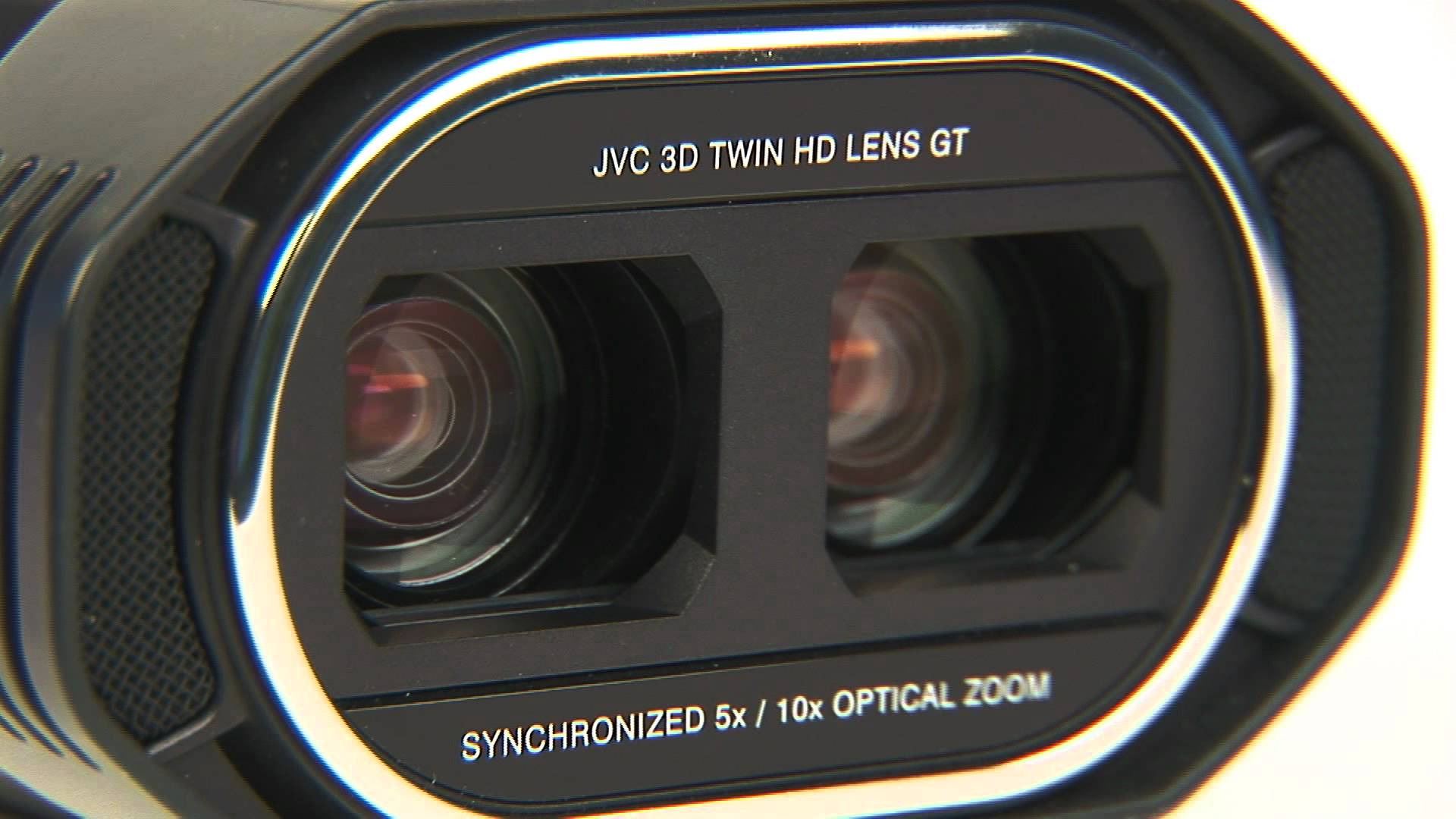


Рисунок 1.4 – Современная камера с двумя линзами

Калибровка камеры — это задача получения внутренних и внешних параметров камеры по имеющимся фотографиям или видео, отснятыми ею. Для калибровки видеокамер предлагается использовать реализованные в программной библиотеке OpenCV алгоритмы, учитывающие радиальное и тангенциальное искажение.

Радиальное искажение описывается следующей формулой:

Наличие радиального искажения проявляется в виде «бочки» или эффекта «рыбий глаз».

Тангенциальное искажение происходит потому, что плоскость линзы не идеально параллельно плоскости изображения. Это может быть исправлено с помощью формул:

Позиция каждой точки пикселя с координатами в исходном изображении будет скорректирована на новую точку с координатами

Таким образом, у нас есть пять параметров искажения, которые в OpenCV представлены в виде одной строки матрицы с 5 столбцами:

Теперь для блока преобразования мы используем следующую формулу:

При этом присутствие объясняется использованием гомографии системы (и координат ). Неизвестные параметры и (фокусные расстояния) и (, ), которые являются оптическими центрами. Если для обеих осей общее фокусное расстояние используется с заданным соотношением сторон (обычно ), а затем и в верхней формуле мы будем иметь одно фокусное расстояние . Матрица, содержащая эти четыре параметра, называется матрицей камеры.

Процесс определения этих двух матриц называется калибровкой. Расчет этих параметров осуществляется с помощью основных геометрических уравнений. Уравнения, используемые в зависимости от выбранных объектов калибрующие.

Проведенные учёными натурные эксперименты с широкоугольной камерой для задач стабилизации видео и целеуказания показали эффективность указанных подходов к устранению оптических искажений.

* 1. Устройство дополненной реальности как средство достижения поставленной задачи

**Microsoft HoloLens** – очки дополненной реальности, разработанные компанией Microsoft. Используют 32-разрядную операционную систему Windows Holographic (Рисунок 1.5).

HoloLens представляет собой надеваемый на голову обруч с расположенными перед глазами тонированными линзами с волнообразной призматической структурой, которые преломляют и отправляют в глаза пользователя изображения с расположенных по бокам микродисплеев. Для использования HoloLens должно быть откалибровано межзрачковое расстояние. Размер устройства может быть приспособлен под размер головы пользователя с помощью специального колёсика.



Рисунок 1.5 – Очки дополненной реальности Microsoft HoloLens

В верхней части расположены 2 пары кнопок – для управления яркостью экрана (над левым ухом) и громкостью звука (над правым). Соседние кнопки имеют разную форму (одна выпуклая, другая вогнутая) с той целью, чтобы их можно было различать на ощупь. Динамики расположены у нижнего края устройства; они позволяют слышать как звуки виртуальной реальности, так и звуки, исходящие извне.

В отличие от большинства других устройств виртуальной, дополненной или смешанной реальности, HoloLens автономны и не требуют подключения к ПК, смартфону или игровой консоли.

HoloLens использует 64-разрядный 4-ядерный процессор Intel Atom x5-Z8100 с частотой 1,04 ГГц. В дополнение к центральному и графическому процессорам HoloLens имеет голографический процессор (англ. holographic processing unit), разработанный Microsoft специально для HoloLens. Голографический процессор, размещённый в корпусе 12 × 12 мм, использует 28 цифровых сигнальных процессоров производства Tensilica для обработки и интеграции данных, поступающих со всех сенсоров, а также пространственного сканирования (англ. spatial mapping) помещения, распознавания жестов, голоса и речи. По утверждению разработчиков, голографический процессор обрабатывает «терабайты информации». SoC и голографический процессор имеют 8 Мб встраиваемой памяти SRAM и по 1 Гб LPDDR3.

Встроенное хранилище данных имеет объём 64 Гб, из них около 10 Гб занимает операционная система, так что пользователю остаются доступны 54,09 Гб.

Объём оперативной памяти — 2 Гб.

Приложения для HoloLens не могут использовать больше 900 Мб памяти, в случае превышения этого лимита работа приложения прерывается.

HoloLens обладают 4 камерами (по 2 с каждой стороны) для сканирования окружения и ориентации в пространстве, 4 микрофонами, гиростабилизатором, датчиком глубины, 2MP видеокамерой, сенсором окружающего освещения.

Оптика HoloLens устроена очень сложно (если сравнивать, например, с устройствами виртуальной реальности), что обусловлено необходимостью не просто выводить изображение на экран, но ещё и правильно совмещать его с объектами реального мира. Жидкокристаллические проекторы с разрешением сторон 16:9, которые Microsoft назвала «световыми движками» (англ. light engines), создают изображение, которое затем проходит через визуализационную оптику (англ. imaging optics), волновод, combiner (устройство, совмещающее проекцию и изображение реального мира) и дифракционные решётки. Линзы имеют 3 слоя — для синего, зелёного и красного цветов — каждый со своими дифракционными свойствами.

Теми, кто пользовался HoloLens, отмечается, что они обладают небольшим полем зрения (при этом субъективные оценки размера поля зрения существенно разнятся), особенно по сравнению со шлемами виртуальной реальности, что часто характеризуется как их основной недостаток; при этом на Electronic Entertainment Expo 2015 Кудо Цунода, вице-президент Microsoft, сообщил, что поле зрения вряд ли значительно изменится к выходу окончательной версии устройства.

HoloLens может генерировать бинауральный звук, что позволяет имитировать направление звука, создавая иллюзию того, что он исходит от виртуального источника.

HoloLens содержат электрический аккумулятор объёмом 16 500 мА·ч, которого должно хватать на 2-3 часа активного использования или 2 недели в спящем режиме. HoloLens можно использовать во время зарядки аккумулятора.

1. Постановка задачи и рассмотрение объектов исследования
   1. Постановка задачи

Итак, задача состоит в том, чтобы проанализировать входные данные и предоставить результаты анализа в форме, удобной для восприятия любым человеком.

Необходимо учитывать целевую аудиторию, т. е. стремиться к тому, чтобы отсутствовала необходимость в определённых знаниях и навыках для корректного считывания информации (Рисунок 2.1).



Рисунок 2.1 – Пример сложного интерфейса дополненной реальности

Проанализировав особенности и возможности виртуальной и дополненной реальностей, логичным шагом будет выбор в пользу дополненной реальности как наиболее подходящей для данной задачи.

Следовательно, инструментом, с которым необходимо будет работать, является вышеописанные очки дополненной реальности HoloLens от компании Microsoft.

Данные очки дополненной реальности имеют различные датчики и сенсоры, из которых наиболее весомыми являются 4 камеры, которые предоставляют видеоряд, который как раз и является последовательностью снимков.

При анализе последовательности снимков подразумевается, что данные снимки приблизительно одних и тех же объектов, но снятых под разными углами, на разных расстояниях и с различных ракурсов.

При всей динамике содержимого изображений для корректного анализа необходимо использовать определённые маркеры, которые должны соответствовать требованиям для успешного позиционирования объекта (Рисунок 2.2).



Рисунок 2.2 – Пример использования маркеров на лице для позиционирования его в пространстве

* 1. Описание различий функционалов разных типов аккаунтов

Сразу можно условно выделить 4 вида предполагаемых пользователей:

1. Родитель
2. Ученик
3. Преподаватель
4. Администратор

На следующей схеме описаны функциональные возможности для каждого типа пользователей:

Рисунок 2.1 – Схема функциональных возможностей

* 1. Нюансы процесса взаимодействия преподавателя и студента

Очень важно чётко определить все особенности процесса взаимодействия преподавателя со студентами.

Так, например, стоит ограничить программно возможности преподавателя, и тем самым защитить обучающихся, например, от таких ситуаций, когда время на выполнение задания слишком мало. Это позволит помочь преподавателю и предотвратит случайные ошибки.

Также стоит учесть интересы студентов. Например, автоматически не принимать и не отсылать преподавателю пустые решения, либо решения, которые не соответствуют заранее установленным требованиям преподавателя.

Это освободит преподавателя от лишней работы, а также предотвратит случайные ошибки студентов, например, студента, который случайно отправил не тот файл, либо файл оказался повреждённым, или не того формата.

1. Описание используемых технологий
   1. Описание языка программирования

Реализация программы, системы сдачи выдачи домашнего задания проходит на языке программирования Java. Java — объектно-ориентированный язык программирования, разрабатываемый компанией Sun Microsystems с 1991 года и официально выпущенный 23 мая 1995 года. Изначально новый язык программирования назывался Oak (James Gosling) и разрабатывался для бытовой электроники, но впоследствии был переименован в Java и стал использоваться для написания апплетов, приложений и серверного программного обеспечения.

Программы на Java могут быть транслированы в байт-код, выполняемый на виртуальной java-машине (JVM) — программе, обрабатывающей байт-код и передающей инструкции оборудованию, как интерпретатор, но с тем отличием, что байт-код, в отличие от текста, обрабатывается значительно быстрее.

Язык Java зародился как часть проекта создания передового программного обеспечения для различных бытовых приборов. Реализация проекта была начата на языке C++, но вскоре возник ряд проблем, наилучшим средством борьбы с которыми было изменение самого инструмента — языка программирования. Стало очевидным, что необходим платформо-независимый язык программирования, позволяющий создавать программы, которые не приходилось бы компилировать отдельно для каждой архитектуры и можно было бы использовать на различных процессорах под различными операционными системами.

Язык Java потребовался для создания интерактивных продуктов для сети Internet. Фактически, большинство архитектурных решений, принятых при создании Java, было продиктовано желанием предоставить синтаксис, сходный с C и C++. В Java используются практически идентичные соглашения для объявления переменных, передачи параметров, операторов и для управления потоком выполнением кода. В Java добавлены все хорошие черты C++.

**Три ключевых элемента объединились в технологии языка Java:**

* Java предоставляет для широкого использования свои апплеты (applets) — небольшие, надежные, динамичные, не зависящие от платформы активные сетевые приложения, встраиваемые в страницы Web. Апплеты Java могут настраиваться и распространяться потребителям с такой же легкостью, как любые документы HTML
* Java высвобождает мощь объектно-ориентированной разработки приложений, сочетая простой и знакомый синтаксис с надежной и удобной в работе средой разработки. Это позволяет широкому кругу программистов быстро создавать новые программы и новые апплеты
* Java предоставляет программисту богатый набор классов объектов для ясного абстрагирования многих системных функций, используемых при работе с окнами, сетью и для ввода-вывода. Ключевая черта этих классов заключается в том, что они обеспечивают создание независимых от используемой платформы абстракций для широкого спектра системных интерфейсов
  1. Описание технологии создания Web-приложений

Веб-приложение Java создает интерактивные веб-страницы, содержащие различные типы языков разметки (HTML, XML и т.д.), а также динамическое содержимое. Это содержимое обычно формируется веб-компонентами, например страницами JavaServer (JSP), сервлетами и компонентами JavaBean, которые позволяют изменять данные и осуществлять их временное хранение, взаимодействовать с базами данных и веб-службами, а также отображать содержимое в ответ на запросы клиентов.

Так как многие задачи разработки веб-приложения могут повторяться или требовать наличия избыточного шаблонного кода, то для уменьшения количества общих действий следует применять веб-платформы. Многие платформы (например, JavaServer Faces) предоставляют библиотеки для создания шаблонов страниц и управления сеансами, а также часто обеспечивают повторное использование кода.

Естественным выбором реализации программного средства является выбор в пользу Web-приложения. Соответственно, использована технология Java EE(Рисунок 3.1).



Рисунок 3.1 – Java EE

Java EE (Enterprise Edition) представляет собой широко используемую платформу, содержащую набор взаимосвязанных технологий, которые существенно сокращают стоимость и сложность разработки, развертывания многоуровневых серверных приложений, а также управления ими. Платформа Java EE основана на платформе Java SE и предоставляет набор интерфейсов API (интерфейсов разработки приложений) для разработки и запуска портируемых, надежных, масштабируемых и безопасных серверных приложений.

**Java EE в числе прочих содержит следующие компоненты:**

* Enterprise JavaBeans (EJB): управляемая серверная архитектура компонентов, используемая для инкапсуляции бизнес-логики приложения. Технология EJB позволяет осуществлять быструю и упрощенную разработку распределенных, транзакционных, безопасных и переносимых приложений, основанных на технологии Java.
* Интерфейс API сохранения состояния Java (Java Persistence API, JPA): инфраструктура, позволяющая разработчикам управлять данными с помощью объектно-реляционного сопоставления (ORM) в приложениях, созданных на платформе Java.

JavaScript представляет собой объектно-ориентированный язык сценариев и в основном используется в клиентских интерфейсах веб-приложений. Ajax (асинхронный JavaScript и XML) является технологией Web 2.0, позволяющей выполнять изменения веб-страницы без ее обновления. Для реализации компонентов и функций Ajax на веб-страницах можно использовать наборы инструментов JavaScript.

Для более полного представления и понимания архитектуры стоит обраться к технологии ASP.NET MVC.

Платформа ASP.NET MVC представляет собой фреймворк для создания сайтов и веб-приложений с помощью реализации паттерна MVC.

Концепция паттерна (шаблона) MVC (model - view - controller) предполагает разделение приложения на три компонента:

Контроллер (controller) представляет класс, обеспечивающий связь между пользователем и системой, представлением и хранилищем данных. Он получает вводимые пользователем данные и обрабатывает их. И в зависимости от результатов обработки отправляет пользователю определенный вывод, например, в виде представления.

Представление (view) - это собственно визуальная часть или пользовательский интерфейс приложения. Как правило, html-страница, которую пользователь видит, зайдя на сайт.

Модель (model) представляет класс, описывающий логику используемых данных.

Общую схему взаимодействия этих компонентов можно представить следующим образом (Рисунок 3.2).

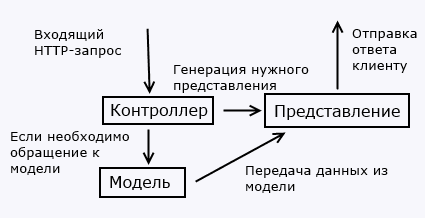


Рисунок 3.2 – Схема взаимодействия

В этой схеме модель является независимым компонентом - любые изменения контроллера или представления не затрагивают модель. Контроллер и представление являются относительно независимыми компонентами, и нередко их можно изменять независимо друг от друга.

Благодаря этому реализуется концепция разделение ответственности, в связи с чем легче построить работу над отдельными компонентами. Кроме того, вследствие этого приложение обладает лучшей тестируемостью. И если нам, допустим, важная визуальная часть или фронтэнд, то мы можем тестировать представление независимо от контроллера. Либо мы можем сосредоточиться на бэкэнде и тестировать контроллер.

Конкретные реализации и определения данного паттерна могут отличаться, но в силу своей гибкости и простоты он стал очень популярным в последнее время, особенно в сфере веб-разработки.

Свою реализацию паттерна представляет платформа ASP.NET MVC. 2013 год ознаменовался выходом новой версии ASP.NET MVC - MVC 5, а также релизом Visual Studio 2013, которая предоставляет инструментарий для работы с MVC5.

Хотя во многих аспектах MVC 5 не слишком сильно будет отличаться от MVC 4, многое из одной версии вполне применимо к другой, но в то же время есть и существенные отличия:

В MVC 5 изменилась концепция аутентификации и авторизации. Вместо SimpleMembershipProvider была внедрена система ASP.NET Identity, которая использует компоненты OWIN и Katana.

Для создания адаптивного и расширяемого интерфейса в MVC 5 используется css-фреймворк Bootstrap

Добавлены фильтры аутентификации, а также появилась функциональность переопределения фильтров

В MVC 5 также добавлены атрибуты маршрутизации

Это наиболее важные нововведения в MVC 5. Кроме того, есть еще ряд менее значимых, например, использование по умолчанию Entity Framework 6, некоторые изменения при создании проекта (концепция One ASP.NET), дополнительные компоненты и т.д.

В любом случае все полученные при работе с MVC 4 навыки можно успешно применять при использовании MVC 5, учитывая, конечно, нововведения.

1. Практическая реализация
   1. Программная реализация прототипа приложения

Для удобства использования веб-приложение обладает графическим интерфейсом***.***

Приложение обладает определённым ранее функционалом. Т. к. ранее было определено, что основной вид домашних работ это письменный, то остановимся на сдаче текстовых домашних заданий.

Итак, по нижеприведённым рисункам можно определить функционал приложения.

Реализована возможность регистрации пользователя и входа под своей учётной записью (Рисунок 4.1).

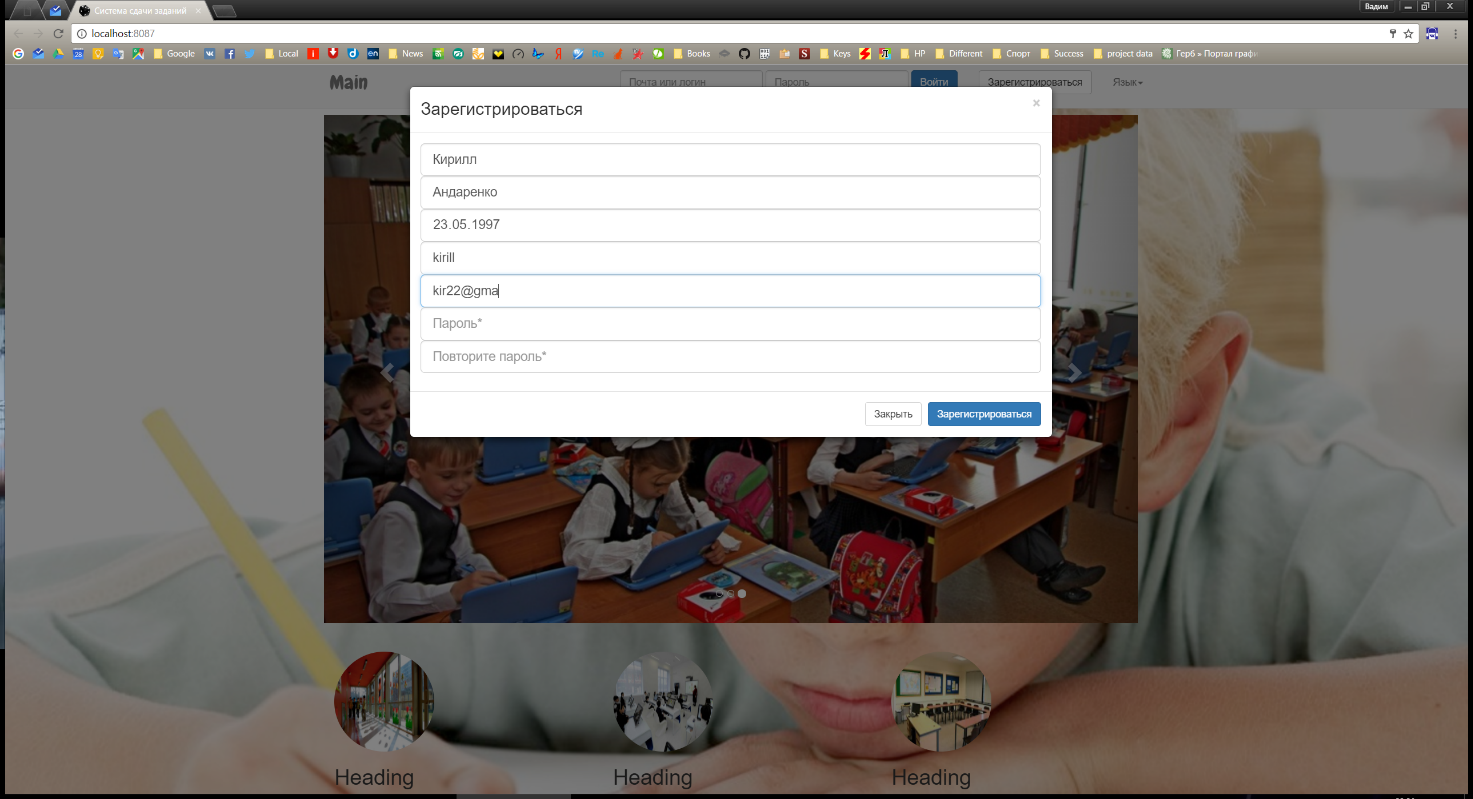


Рисунок 4.1 – Процесс регистрации пользователя

Как было замечено ранее, студент и преподаватель имеют разные уровни доступа. Можно ознакомиться с доступными страницами(действиями) посмотрев на главную страницу студента (Рисунок 4.2).

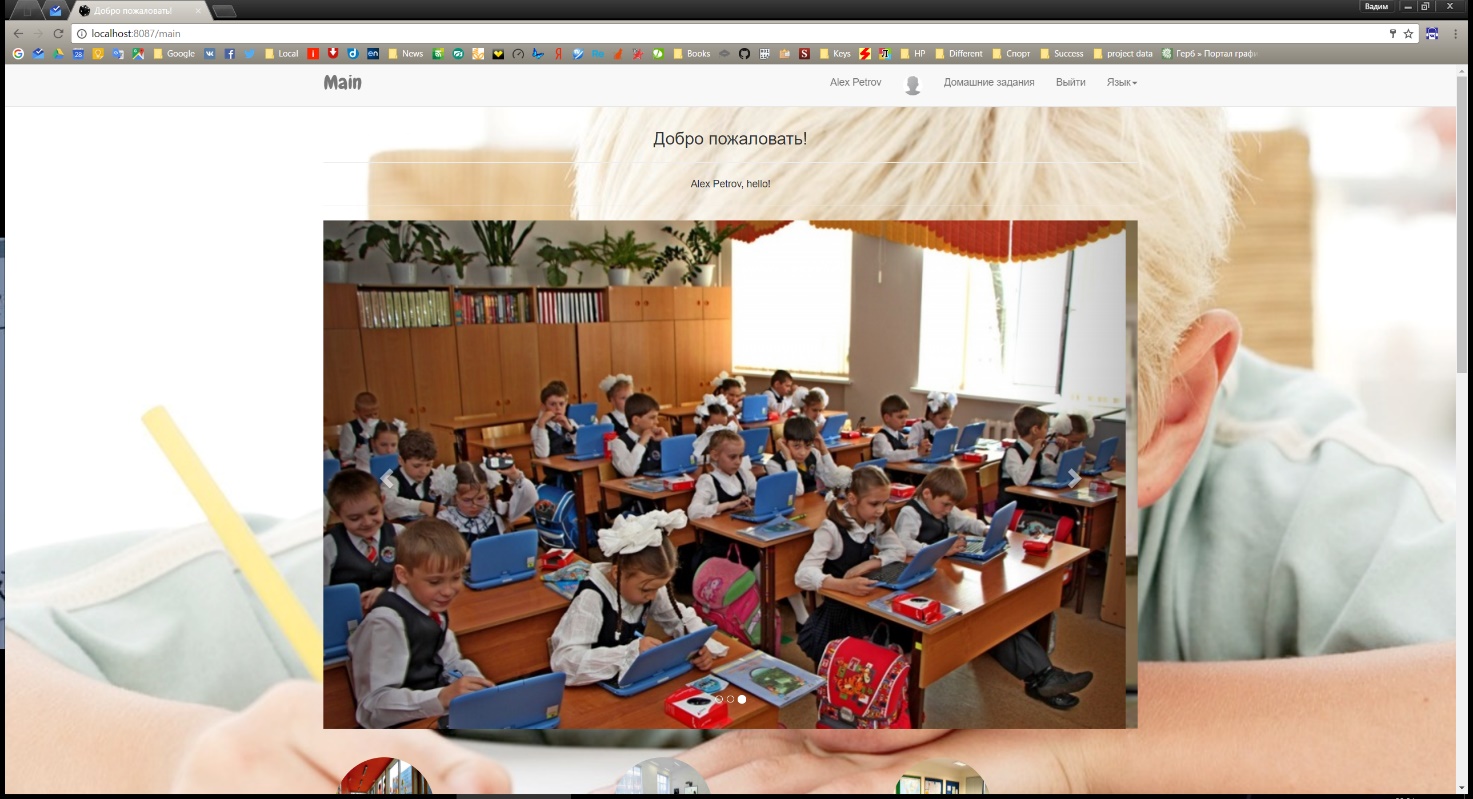


Рисунок 4.2 – Главная страница для студента

Также реализована система настроек. К настройкам своего аккаунта имеют доступ все пользователи. Доступные настройи включают в себя изменение имении фамилии, выбор и сброс аватарки, смену пароля (Рисунок 4.3).

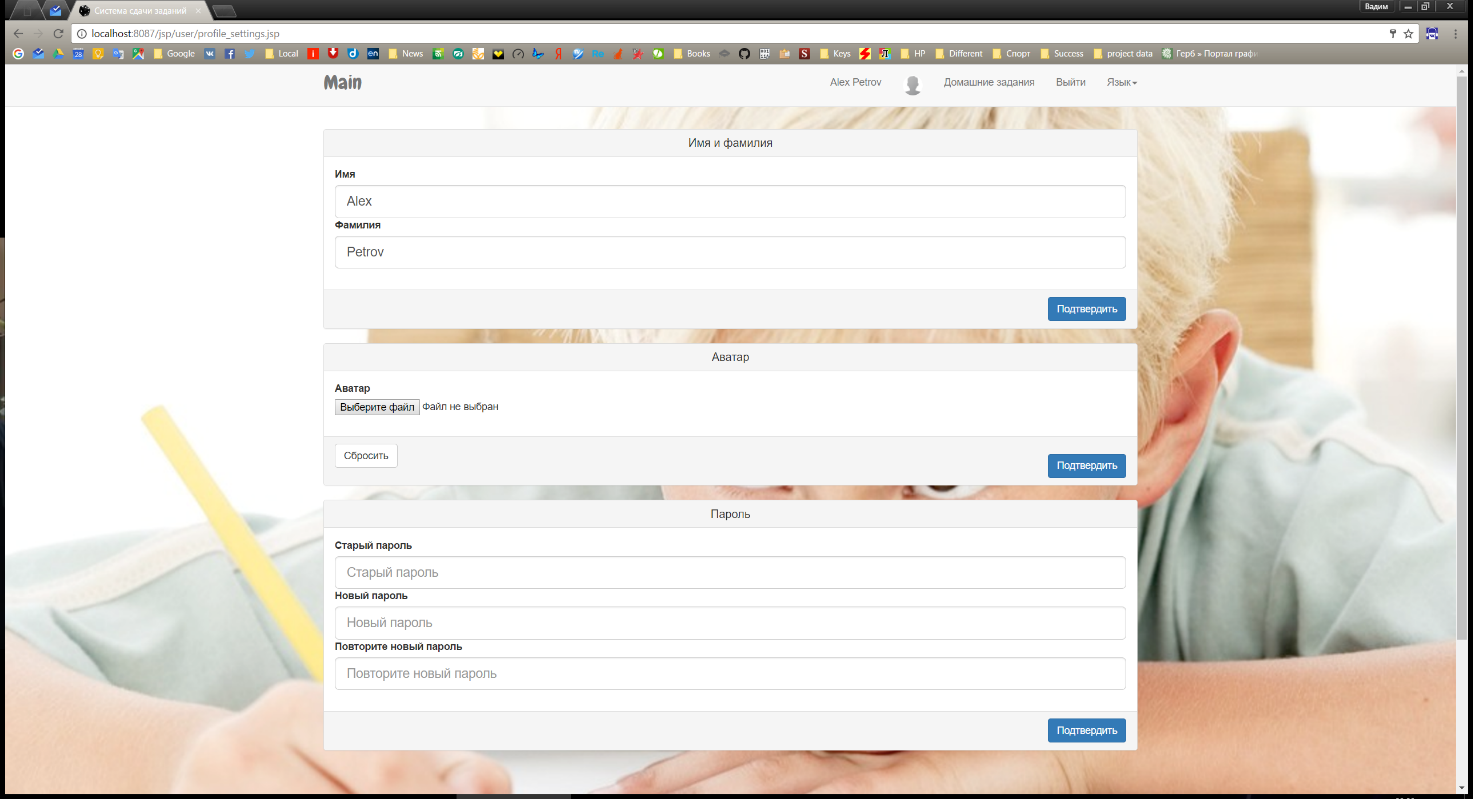


Рисунок 4.3 – Страница настроек

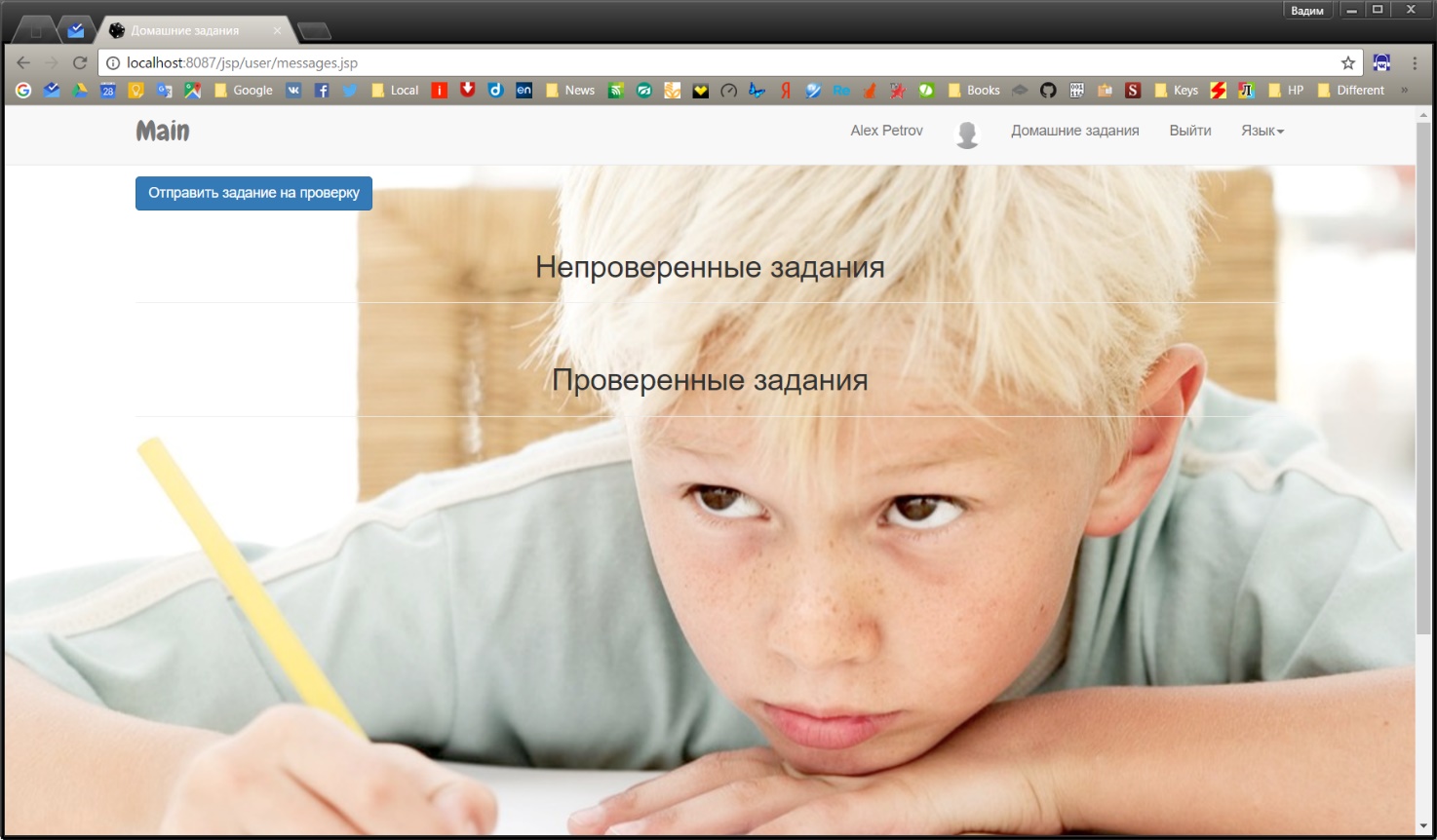


Рисунок 4.4 – Страница отправки готовых заданий

Форма отсылки текстовых заданий отсылки текстовых заданий включает в себя название и текст самого задания (Рисунок 4.5).

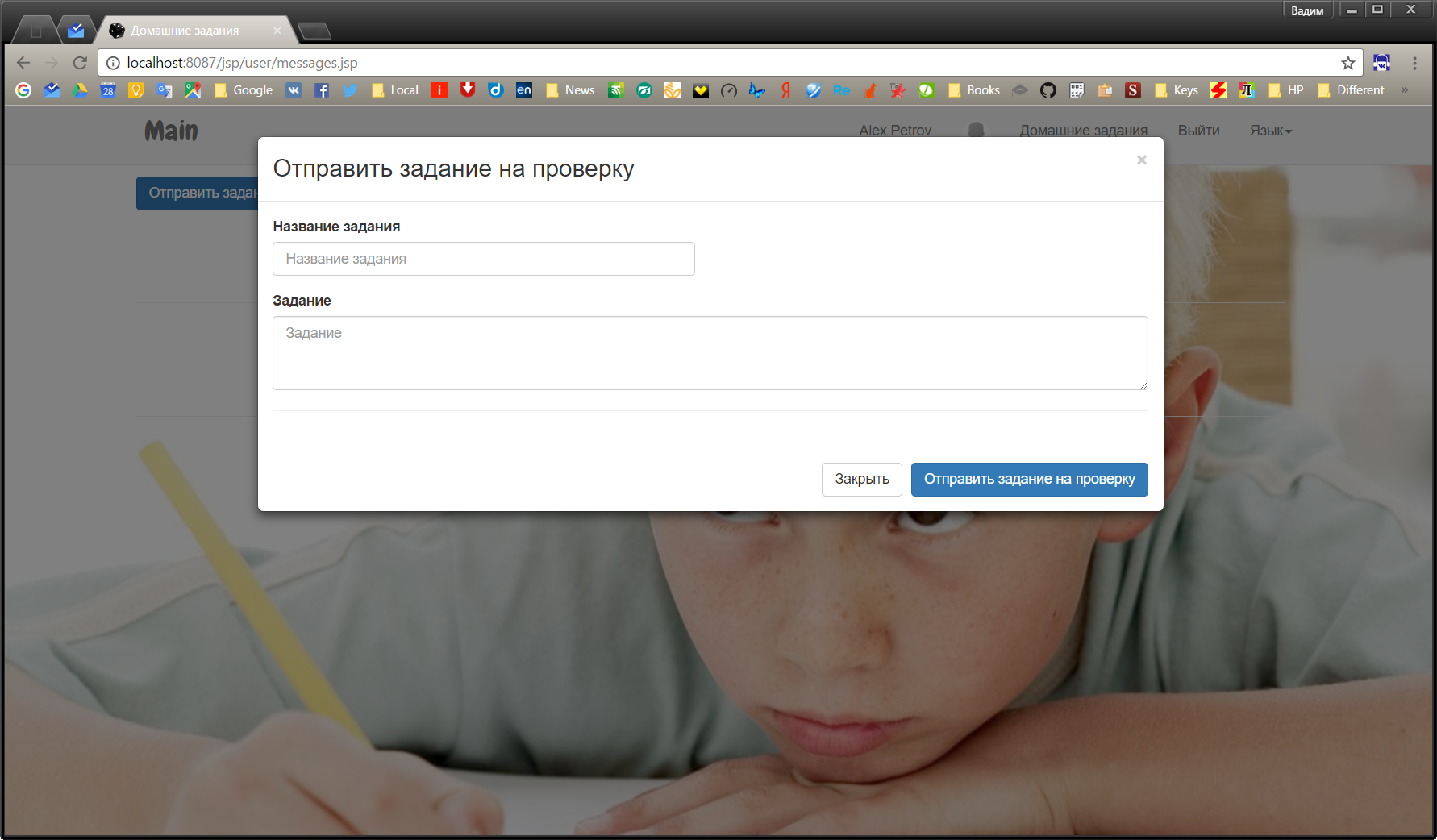


Рисунок 4.5 – Форма отправки задания

Преподаватель имеет доступ к администрированию, где он может блокировать и разблокировать студентов. Также он имеет возможность работать с прислаными работами (Рисунок 4.6).

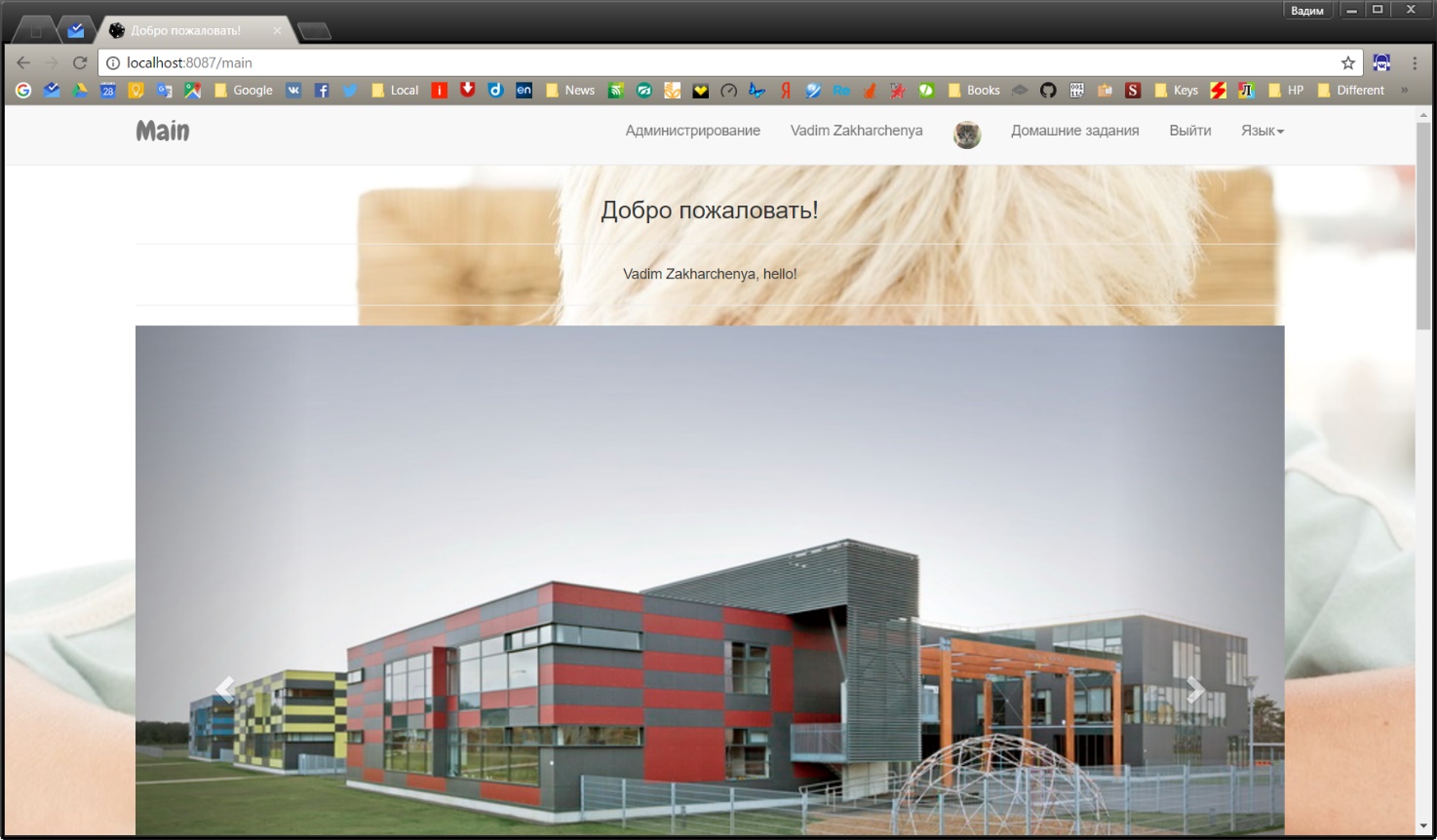


Рисунок 4.6 – Главная страница для преподавателя

На странице администрирования преподаватель просматривает список пользователей, а также, при необходимости, блокирует или разблокирует их (Рисунок 4.7).

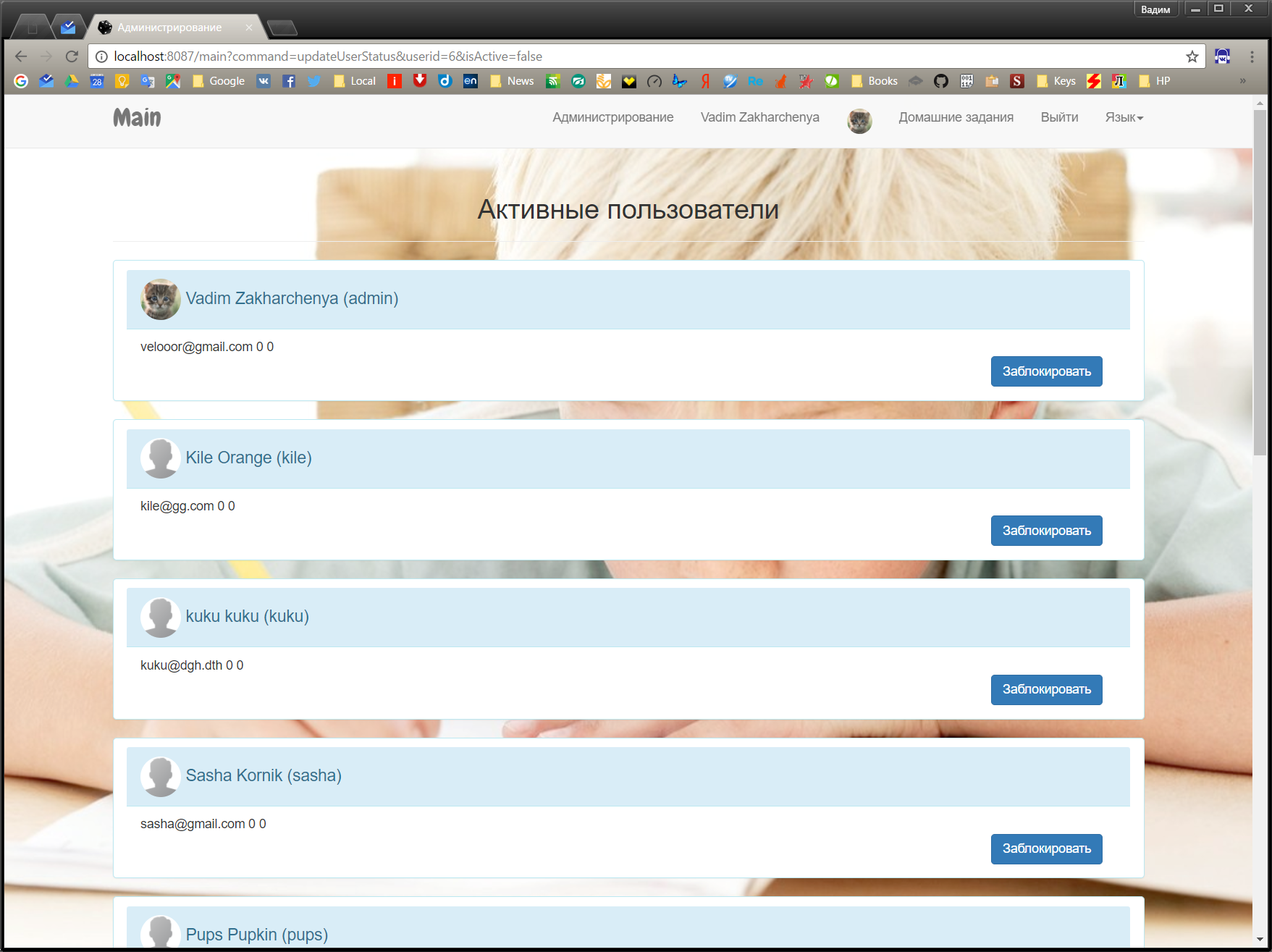


Рисунок 4.7 – Страница управления студентами

Данное веб приложение оснащено информацонным стендом, где может публиковаться различный контент. Информацию выдавать различную разным типам пользователей (Рисунок 4.8).

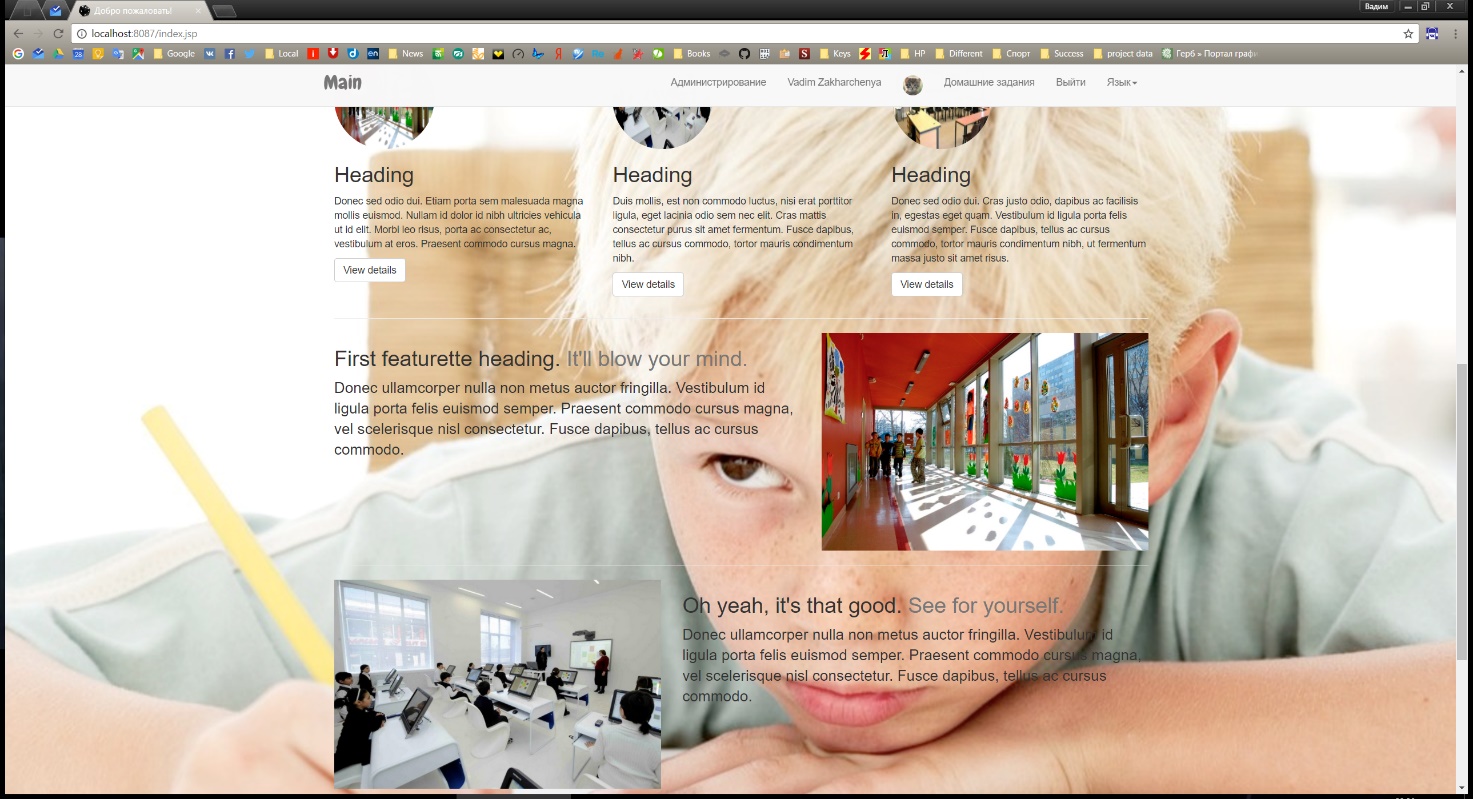


Рисунок 4.8 – Информационный стенд на главной странице

При получении задания от студента, оно автоматически попадает в раздел «непроверенные» и соотвественно подсвечивается. Имеется возможность увидеть всю необходимую информацию: время создания, название задания, отправителя и т. д.

Преподаватель может перевести задание в раздел проверенные, а также полность удалить при необходимости (Рисунок 4.9).

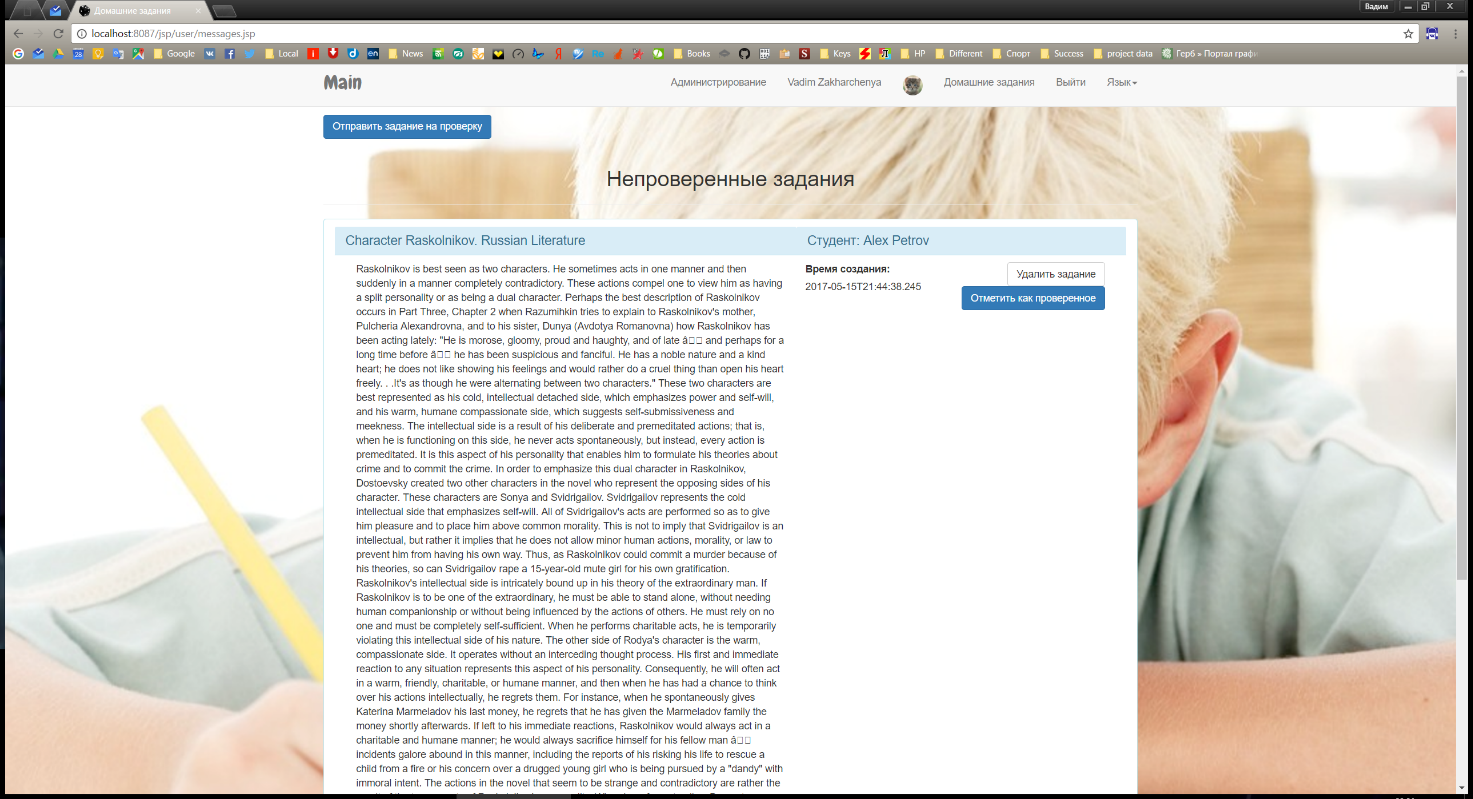


Рисунок 4.9 – Страница домашних заданий у преподавателя

В ходе работы приложения естественным образом возникает необходимсть хранения информации. Можно привести несложную диаграмму используемой баз данных (Рисунок 4.10).

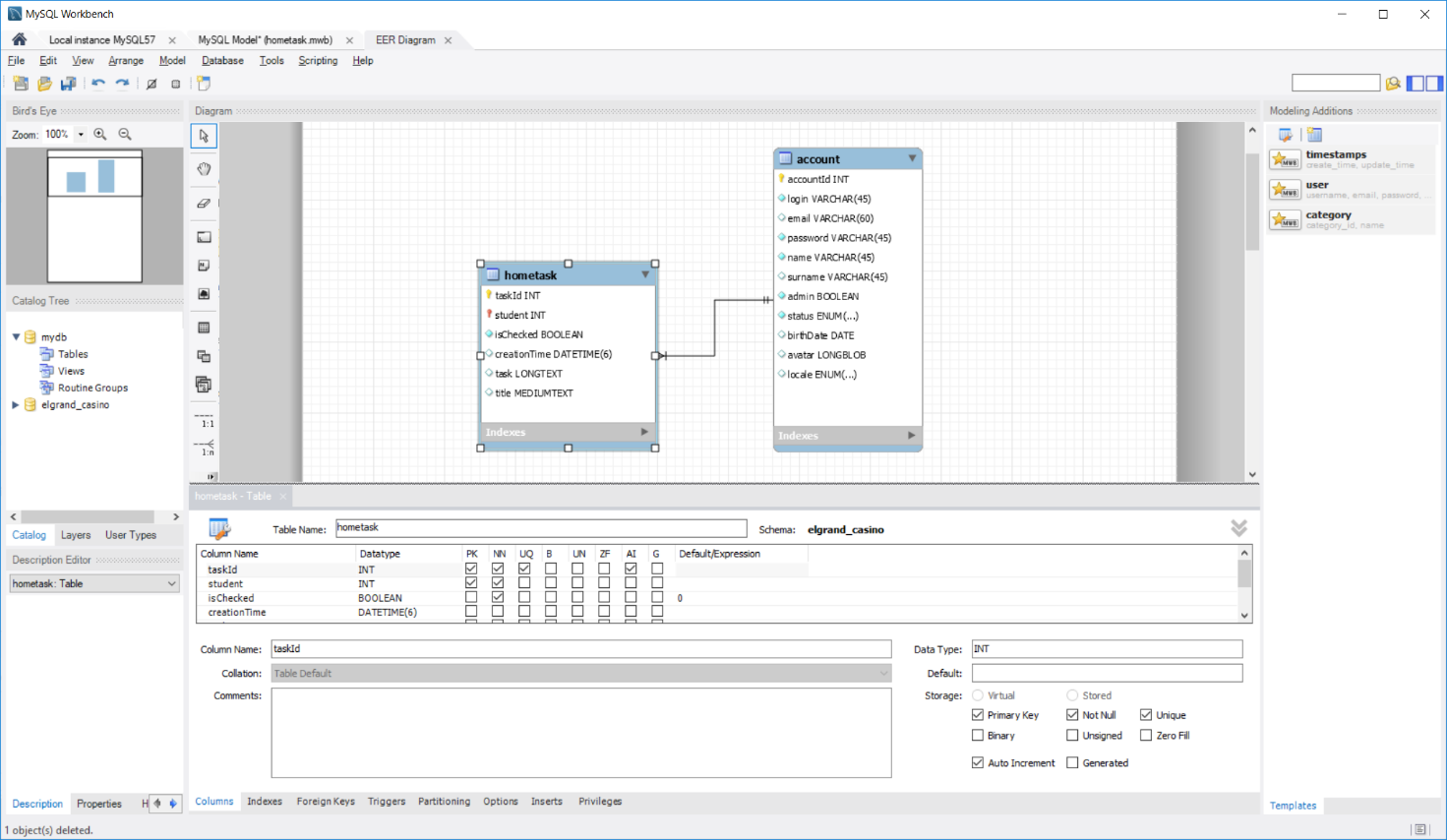


Рисунок 4.10 – Диаграмма базы данных

ЗАКЛЮЧЕНИЕ

В данной курсовой работе:

* изучены виды домашних заданий.
* проанализированы существующие системы сдачи и выдачи домашних заданий.
* выявлены недостатки существующей системы и определить причины и предпосылки для создания улучшенной системы сдачи домашних заданий с использованием информационных технологий.
* рассмотрена необходимость разграничения доступа к аккаунтам для различных видов пользователей.
* определены необходимые технологии для реализации системы.
* реализован прототип программного средства системы выдачи и сдачи домашних заданий.

Список использованных источников

1. Гриффитс, И. Программирование на C# 5.0 / И. Гриффитс. – Москва: Эксмо, 2014. – 1136 с.
2. Макфарланд, Д. Большая книга CSS3 / Д. Макфарланд – СПб: Издательский дом «Питер», 2011. – 560 с.
3. Гонсалвес, Э. Изучаем Java EE 7 / Э. Гонсалвес. – СПб: Питер, 2017. – 640 с.
4. Эккель, Б. Философия Java / Б. Эккель. – СПб: Питер, 2017. – 1168 с.
5. Фримен, А. Pro ASP.NET MVC 4 / А. Фримен, С. Сандерсон. – 4-е изд. – Москва: Вильямс, 2011. –672 с.

Список использованных источников:

1. Роджерс Д., Адамс Д. Математические основы машинной графики. - М.: Мир, 2001. - 604 с.

2. босз орепсу [Электронный ресурс]. - Режим доступа: 1Шр://с1ос5.орепсу.огд.