|  |  |
| --- | --- |
| Министерство науки и высшего образования РФ  Федеральное государственное автономное образовательное учреждение  высшего образования  «Пермский государственный национальный исследовательский университет» | |
| Кафедра информационных технологий | |
| УДК 004.9 | |
| Проектирование, разработка и документирование информационной системы учета времени самостоятельной работы студентов в компьютерном классе  Выпускная квалификационная работа бакалавра  По направлению 09.03.02 «Информационные системы и технологии» | |
| Электронная версия работы помещена в электронную библиотеку кафедры ИТ название файла \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ дата \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Работу выполнил студент  группы ММ/О ИТХ-19 4 курса механико-математического факультета  Васильев Игорь Дмитриевич  Научный руководитель:  Кандидат педагогических наук, доцент кафедры ИТ  Василюк Надежда Николаевна  «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_202\_г. |
| Пермь 2023 | |

**Аннотация**

Васильев Игорь Дмитриевич

Направление «Информационные системы и технологии», группа ИТХ-19.

Проектирование, разработка и документирование информационной системы учета времени студентов для самостоятельной работы в компьютерном классе. Выпускная квалификационная работа. 2023 год.

В данной работе проанализированы аналоги существующих систем учета времени. В результате анализа составлены требования к проектированию системы, выявлены задачи на проектирование.

Работа содержит описание логики работы информационной системы в виде диаграмм, модель данных, а также возможный интерфейс пользователя.

Во введении обосновывается актуальность темы, поставленные задачи и цель, определяется предмет и объект исследования.

В первой главе дана теоретическая основа для разработки информационной системы учета времени студентов. В ней проведен обзор и анализ аналогов, а также выявлены требования к разрабатываемой системе.

Во второй главе описан процесс выбора инструментальных средств проектирования, создания модели информационной системы и описания используемых технологий.

В третьей главе представлен выбор инструментальных средств разработки, а также сама разработка информационной системы. Кроме того, в главе представлены основные рекомендации по совершенствованию разработанной информационной системы.

В заключении сформированы выводы по теме выпускной квалификационной работы.

Спроектированная информационная система может быть использована для работника компьютерного класса и для студентов, желающих записаться в компьютерный класс на самостоятельную работу.

Работа содержит: стр. ил., библ. назв

Ключевые слова: учет времени, самостоятельная работа студента, компьютерный класс, информационная система, uml-диаграммы.

Оглавление

[ВВЕДЕНИЕ 4](#_Toc136036699)

[Обзор литературы 6](#_Toc136036700)

[Глава 1 Анализ предметной области 8](#_Toc136036701)

[1.1 Теоретическое обоснование разработки 8](#_Toc136036702)

[1.2 Обзор и сравнение существующих аналогов ИС 8](#_Toc136036703)

[1.3 Требования к проектируемой системе 14](#_Toc136036704)

[1.4 Вывод 15](#_Toc136036705)

[Глава 2 Проектирование модели информационной системы учета СРС студента 16](#_Toc136036706)

[2.1 Обзор и выбор методологии моделирования 16](#_Toc136036707)

[2.2 Выбор CASE-средства проектирования ИС 19](#_Toc136036708)

[2.3 Моделирование информационной системы 21](#_Toc136036709)

[2.4 Вывод 28](#_Toc136036710)

[Глава 3 Разработка прототипа информационной системы учет СРС студента 29](#_Toc136036711)

[3.1 Прототипирование интерфейса ИС 29](#_Toc136036712)

[3.1.1 Интерфейс студента 29](#_Toc136036713)

[3.1.2 Интерфейс лаборанта 32](#_Toc136036714)

[3.2 Выбор средств разработки ИС 35](#_Toc136036715)

[3.2.1 Система управления базой данных 35](#_Toc136036716)

[3.2.2 Серверная часть 36](#_Toc136036717)

[3.2.3 Клиентская часть 38](#_Toc136036718)

[3.3 Создание базы данных 38](#_Toc136036719)

[3.4 Реализация прототипа ИС 39](#_Toc136036720)

[3.5 Рекомендации по совершенствованию ИС 42](#_Toc136036721)

[3.6 Вывод 43](#_Toc136036722)

[Заключение 44](#_Toc136036723)

[Библиографический список 45](#_Toc136036724)

[ПРИЛОЖЕНИЕ А 48](#_Toc136036725)

[ПРИЛОЖЕНИЕ Б 49](#_Toc136036726)

[ПРИЛОЖЕНИЕ В 50](#_Toc136036727)

[ПРИЛОЖЕНИЕ Г 51](#_Toc136036728)

ВВЕДЕНИЕ

В современном мире наиболее решающим фактором развития является информация. Используя её, человек может строить заключения и формировать свое личное и коллективное будущее. Для того чтобы не запутаться и не потеряться в многочисленных информационных потоках, человек создает и совершенствует инструменты управления информацией. Основным инструментом управления сведениями об окружающем мире и протекающих в нем процессах является информационная система.

Информационная система (ИС) – система, которая организует хранение и манипулирование информацией о предметной области. Такое определение в широком смысле описывает ГОСТ 34.321-96 [1]. В узком смысле, информационная система – это система, предназначенная для автоматизации бизнес-процессов компании или отдельных пользователей, позволяющая обработку данных, включая удаление, изменение и добавление. Главной задачей ИС является полное удовлетворение информационных потребностей конечного пользователя.

Разработка информационной системы для учета времени студентов в компьютерном классе становится актуальной задачей, поскольку в высших учебных заведениях происходит компьютеризация и переход к электронному документированию, а также возникает потребность в быстром и удобном доступе к данным и их обработке. Кроме того, важным аспектом является возможность доступа к ресурсу из любого места, где имеется интернет-соединение.

Таким образом, объектом исследования является учет времени самостоятельной работы студентов.

Предметом исследования является автоматизация учета времени самостоятельной работы студентов в компьютерном классе.

Цель данной выпускной квалификационной работы – проектирование, разработка и документирование ИС учета времени студентов для самостоятельной работы в компьютерном классе, для автоматизации работы сотрудника, отвечающего за компьютерный класс.

Исходя из цели работы, выделим следующие задачи, которые необходимо решить в ходе проектирования:

1. изучить существующую литературу по теме исследования;
2. изучить аналоги системы учета времени;
3. провести сравнительный анализ этих систем;
4. выявить и определить требования к разрабатываемой ИС;
5. провести анализ существующих инструментов для проектирования ИС;
6. определить набор инструментальных средств в соответствии с требованиями;
7. спроектировать модель ИС;
8. разработать прототип интерфейса ИС;
9. разработать прототип ИС;
10. сформировать рекомендации по совершенствованию ИС.

Обзор литературы

Целью данного обзора является выяснение основных проблем в управлении процесса учета самостоятельной работы студентов (СРС) в компьютерном классе. Самостоятельная работа имеет огромное значение в жизни студентов, так как она способствует развитию творческого мышления и формированию уникального взгляда на дисциплины, которые нельзя достичь без личного развития. Данное утверждение подтверждается учебным планом для направления "Информационные системы и технологии" [2], в котором отведено 5042 часа для самостоятельной работы из общей планируемой трудоемкости в 7996 часов. Это составляет около 63% всей учебной деятельности студента в университете. Так как во множестве дисциплин требуется дорогостоящее компьютерное оснащение и программное обеспечение, учащимся приходится пользоваться вычислительными возможностями университета. По причине этого, студент должен правильно распределять свои временные ресурсы. Поэтому особенно важно знать наиболее надежные и эффективные методы создания благоприятных условий для организации и учета СРС в компьютерном классе.

Данную тему частично рассматривает методист Чебаркульского профессионального техникума Т. А. Кудрявцева в своей статье «Проблемы организации самостоятельной работы студентов в условиях реализации федеральных государственных образовательных стандартов среднего профессионального образования третьего поколения» [3]. Основными трудностями при организации СРС в компьютерном классе стали:

1. недостаток навыков планирования самостоятельной работы у студентов;
2. отсутствие достаточных навыков у студентов по оформлению СРС;
3. ограниченные знания технологий у сотрудников и студентов;
4. неумение рассчитать трудоемкость заданий СРС.

Таким образом, одним из ключевых решений является обеспечение необходимыми материально-техническими условиями, включающих наличие необходимого уровня оснащенности кабинетов, предоставление методических материалов для ознакомления работников с бизнес-процессом и наглядных инструкций для оформления СРС, обеспечение необходимой информацией для обучения студента самостоятельному планированию своего времени в компьютерном классе и организацию учета для СРС.

В статье "Организация самостоятельной работы студентов в условиях реализации ФГОС" [4], заместитель директора по учебно-методической работе Н. В. Ананьина рассмотрела вопрос о реализации учета СРС с точки зрения локальных актов. В статье было указано, что основными документами, регламентирующими этот процесс, являются:

1. положение о самостоятельной работе студентов, определяющее сущность самостоятельной работы студентов, ее назначение, планирование, формы организации и виды контроля;
2. положение о журнале учета самостоятельной работы студентов, определяющее порядок оформления и ведения журналов учета СРС.

В связи с этим, можно предположить, что заполнение журналов учета СРС, подготовка каждого студента к самостоятельной работе без использования современных информационных технологий в сфере обработки данных замедляет процесс осуществления СРС. В свою очередь, возникает проблема хранения, сортировки, доступа к данным и их своевременной передачи. Хранение бумажных документов требует значительных затрат и усилий, поскольку необходимо обеспечить их безопасное хранение, оценить риски повреждения информации и предусмотреть дублирование на бумажных носителях. Кроме того, заполнение документов в письменной форме может привести к многочисленным ошибкам и опечаткам, которые могут иметь неблагоприятные последствия.

Важно отметить, что проектирование, разработки и документирование информационной системы учета времени студентов для самостоятельной работы в компьютерном классе является необходимостью, потому что использование информационных технологий и вычислительной техники существенно снизит имеющиеся недостатки в процессе заполнения документов, записи СРС. Это позволит сократить трудозатраты и повысить достоверность данных.

1. Анализ предметной области

В этой главе описана актуальность проектируемой системы, проанализированы существующие информационные системы, решающие проблему учёта времени самостоятельной работы студентов выявлены и определены требования к разрабатываемой ИС.

* 1. Теоретическое обоснование разработки

В настоящее время все большее число учебных учреждений внедряют и используют различные ИС для автоматизации работы преподавателей и подразделений университета. Компьютеризация бизнес-процессов позволяет добиваться высокой скорости ведения документов и предотвращает большинство допускаемых ошибок при фиксации.

Как правило, для того чтобы студенту записаться на самостоятельную работу в компьютерный класс, ему каждый раз необходимо лично подходить к лаборанту или ответственному за компьютерный класс, который, в свою очередь, по разным причинам может отсутствовать на рабочем месте. Использование информационной системы учета СРС может решить эту проблему, предоставляя возможность студентам осуществлять запись через сеть Интернет.

Более того, сотрудник компьютерного класса должен понимать, сколько студентов записалось и в какое время, за какими компьютерами они будут работать, сколько времени потребуется для решения их задач, и с какой целью им потребовался компьютер. Сохранение такой информации на бумажных носителях не является эффективным, поскольку анализировать такие данные гораздо сложнее по сравнению с использованием информационных систем.

Таким образом, проектирование такой ИС является актуальной проблемой, которая позволит упростить доступ студентов к записи и облегчит анализ отчетов.

* 1. Обзор и сравнение существующих аналогов ИС

На сегодняшний день в открытом доступе существует некоторое количество систем для учета времени СРС студента. Многие такие системы разрабатываются внутри учебных учреждений и являются частью системы университета. Однако большую часть рынка занимают коммерческие продукты для учета клиентов в разных предметных областях. Рассмотрим системы, предназначенные для СРС студентов, а также изучим наиболее подходящие и универсальные продукты, которые можно приобрести и использовать в компьютерном классе.

1. «1С: Университет» [5]

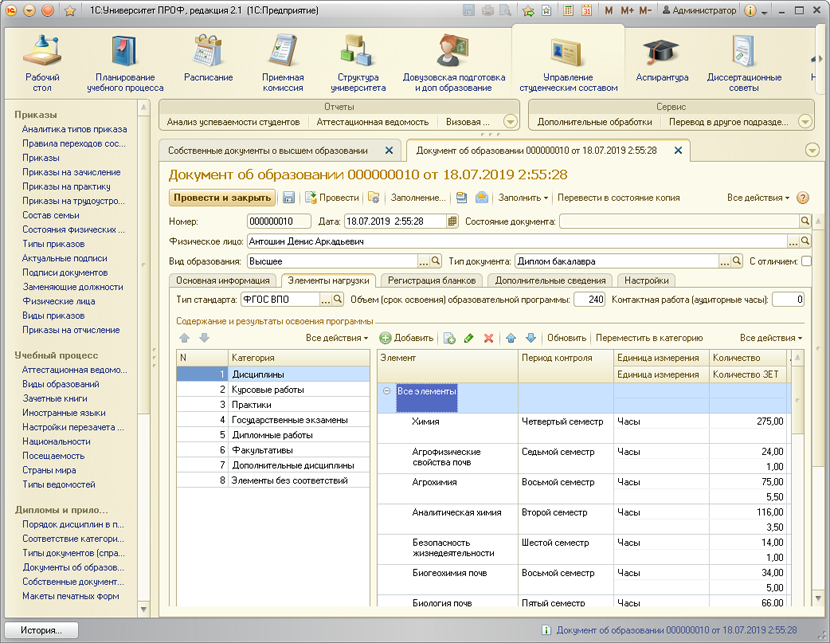
ИС разработана фирмой «1С». Представляется обширным программным обеспечением, которое предоставляет возможность анализировать и обрабатывать данные студентов. Оно также может использоваться для учета времени самостоятельной работы студентов.

Наиболее значимым преимуществом этой системы является ее гибкость и возможность настройки под конкретные задачи. Пользователи этой системы зависят от начальной конфигурации, поскольку она определяет функциональность и параметры работы системы. Перечислим основные особенности системы:

* хранение и обработка сведений о контингенте студентов вуза (Рис. 1);
* создание расписаний для компьютерного класса;
* возможность распределения студентов по компьютерам в классе;
* учет посещения студентов;
* формирование графиков работы студентов и компьютерных классов;
* создание отчетов о посещениях студентов на почасовой и суточной основе.

Тем не менее, следует отметить, что данная система не является оптимальным вариантом для нашей предметной области. Дело в том, что она имеет множество недостатков, таких как:

* внедрение и развертывание занимают значительное количество времени;
* трудоемкий процесс обучения персонала;
* необходимость постоянной поддержки IT-инфраструктуры.;
* высокая стоимость владения (приблизительно 96 000 рублей за копию) [6].



Рисунок

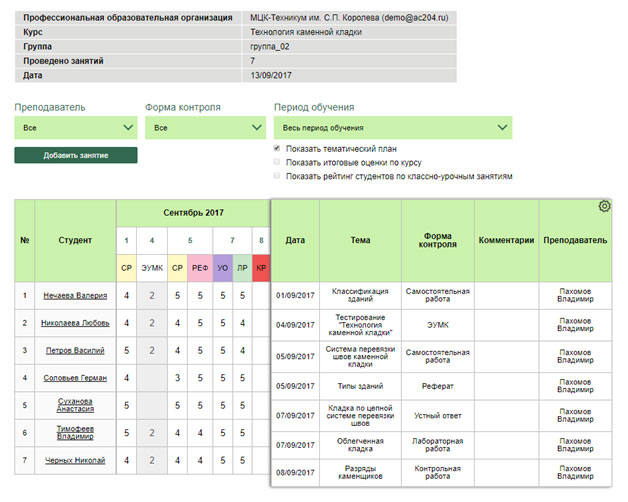
– 1C:Университет

1. «Электронный журнал»

Данная система разработана ГАПОУ МО "МЦК — Техникум имени С. П. КОРОЛЁВА" и является частью большого программного комплекса платформы «Цифровой колледж Подмосковья».

Информационная система «Электронный журнал» предназначена для ведения учета и анализа учебной деятельности студентов (Рис. 2) в образовательных организациях. Она способствует более объективному оцениванию учебных результатов студентов, а также обеспечивает учет посещаемости занятий. ИС осуществляет автоматизированный подсчет рейтинга студентов, а также позволяет управлять настройками для изменения выводимой информации. Данные электронного журнала сохраняются в системе по окончании учебного года.

Данная система предназначена для использования, как студентами, так и преподавателями, которые могут войти в нее с помощью логина и пароля. Интерфейс системы разработан удобным образом для взаимодействия с данными и обладает всеми необходимыми функциями для эффективного ведения учета и анализа учебной деятельности. Система также обладает возможностью анализа данных, которая помогает выявлять тенденции, проводить сравнительный анализ успеваемости студентов и осуществлять мониторинг учебной деятельности. Это обеспечивает принятие обоснованных решений на основе объективной информации.



Рисунок

– «Цифровой колледж Подмосковья»

Основной функционал ИС:

* обеспечение контроля над посещаемостью студентов;
* систематизация и отображение информации о ходе и результатах образовательного процесса студентов;
* просмотр итоговых результатов посещения студентов за определенный период обучения;
* осуществление гибкой системы настроек структуры учебного года;
* опциональное размещение комментариев и замечаний;
* просмотр и редактирование данных в электронном журнале в соответствии с уровнем доступа.

Отличительной чертой данного программного решения является то, что он специально разработан для использования только колледжами Подмосковья в рамках системы "Цифровой колледж Подмосковья". Оно не предоставляет возможности интеграции с другими системами и ориентировано исключительно на потребности колледжей Подмосковья.

1. «Yclients»

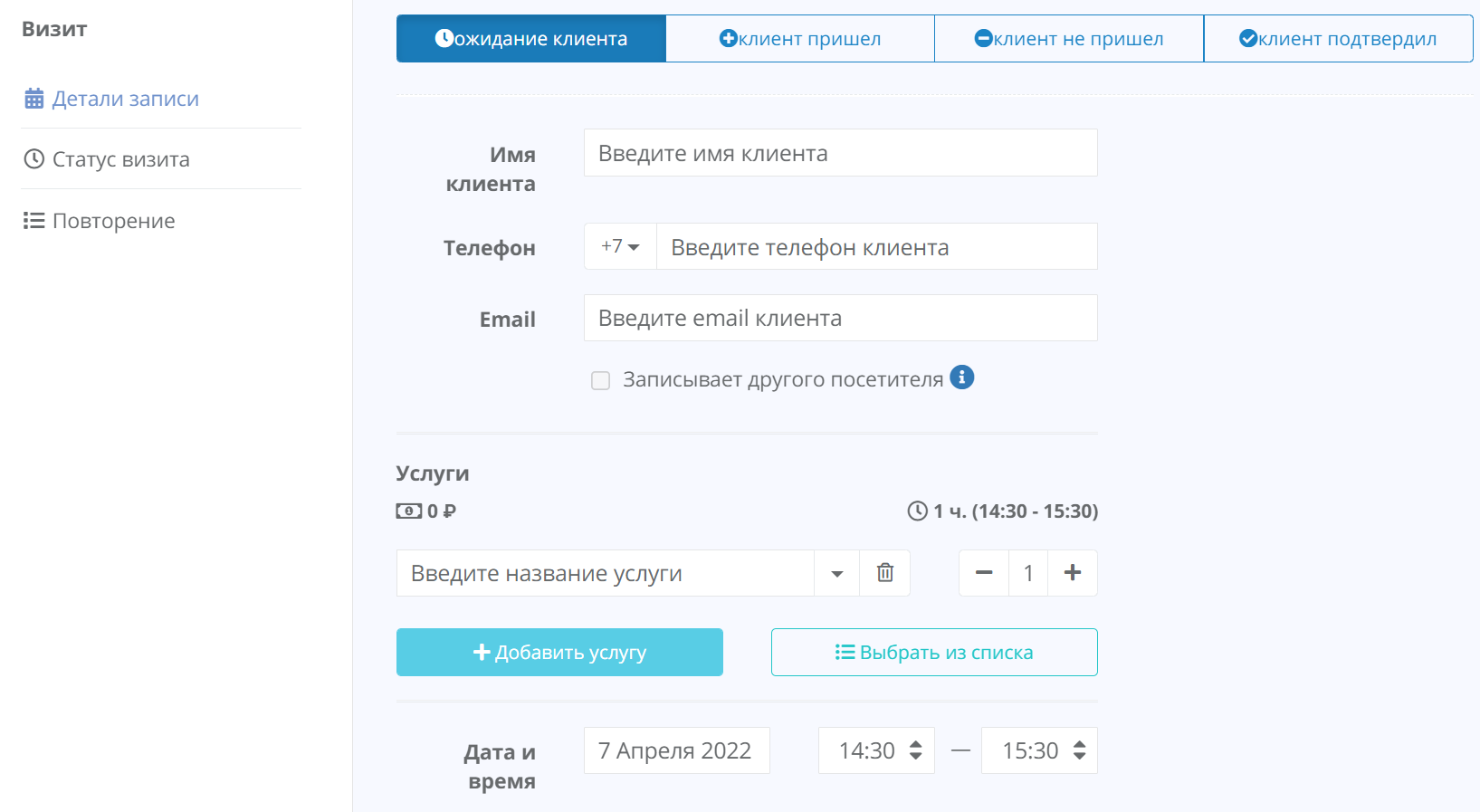
ООО "УАЙКЛАЕНТС" разработала данное программное решение, которое предназначено для учета клиентов в различных предприятиях и сферах услуг, где большое значение придается записи посетителей. Система Yclients помогает эффективно планировать расписание на весь день для клиентов и вести весь учет в электронном формате.

Данный программный продукт предоставляет возможность использования, как сотрудниками, так и клиентами. Он позволяет настроить онлайн-запись через интернет, что обеспечивает удобство и доступность использования системы для всех пользователей.

Основная функциональность данного программного продукта включает в себя следующие возможности:

* записать человека на определенный день и время (Рис. 3);
* учет количества клиентов и даты их записи (Рис. 4);
* онлайн-запись клиентов;
* ведение статистики по дням, месяцам, кварталам и годам;
* сохранение информации о клиенте в базу данных.

Дополнительно можно отметить, что система может предоставлять возможности для отправки уведомлений клиентам о подтверждении записи, изменениях в расписании или других важных событиях, а также для генерации отчетов и аналитики для более детального анализа деятельности и принятия управленческих решений.

* 

Рисунок

– Запись



Рисунок

– Расписание

Стоимость системы рассчитывается по подписке. Цены варьируются от 714 руб. до 1250 руб. за месяц использования.

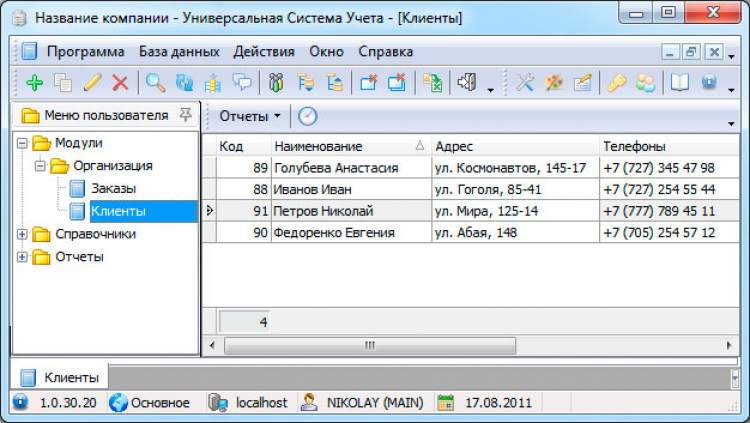
1. «Универсальная система учета»

Данная система разработана компанией «USU Software», является разработкой программного обеспечения для автоматизации бизнес-процессов. Представляет собой программу, в которой администратор может оперативно принимать заявки от клиентов и вести запись в электронном формате.

Основные функциональные возможности данной системы включают:

* формирование единой базы клиентов с полной информацией о контактах и другими необходимыми данными (Рис. 5);
* обозначение как выполненной, так и планируемой работы по любому клиенту;
* напоминания о важных делах;
* планирование резервного копирования;
* импорт/экспорт информации.

Стоимость программы составляет 60 американских долларов.



Рисунок

– Универсальная система учета

Таким образом, рассмотренные выше системы являются многофункциональными и имеют как преимущества, так и ряд недостатков. Например, одна из рассмотренных систем не предполагает использование в учебных учреждениях, не являющихся колледжами Подмосковья, к тому же, эта система рассчитана на форму и структуру организации только для этих образовательных организаций.

Вместе с тем, решение от «1С», хоть и является универсальным, но, тем не менее внедрение такой огромной системы является ресурсозатратным процессом, и не является правильным решением в данной предметной области.

Кроме того, многие из представленных оболочек не ориентированы на учет времени самостоятельной работы студента, что является критическим в выборе программного продукта. Особенно важной чертой продукта должна быть запись студента через интернет, что присутствует только в 2-х рассмотренных системах.

Анализ существующих аналогов демонстрирует, что разработка своей ИС по учету времени СРС в компьютерном классе предпочтительнее, чем покупка аналогичного продукта у стороннего разработчика. При этом проектирование, документирование и разработка программного продукта позволит организовать именно те функциональные требования, которые необходимы работникам компьютерного класса и студентам, а также избежать излишней функциональности, присутствующей в других программных продуктах.

* 1. Требования к проектируемой системе

На основе анализа существующих систем выбраны следующие функциональные требования, которые должны осуществляться разрабатываемой ИС:

* учет времени работы студента в компьютерном классе;
* учет времени бронирования компьютера студентом;
* контроль свободных компьютеров в классе;
* хранение данных о компьютерах, студентах и времени их работы;
* предоставление отчетов по запросу руководителя подразделения;
* удобное представление информации работнику образовательного учреждения;
* дистанционный доступ к записи для студентов;
* отметка о посещении.
  1. Вывод

В первой главе исследована актуальность проектируемой системы, проанализированы существующие аналоги информационных систем, выявлены и определены основные требования к ИС.

1. Проектирование модели информационной системы учета СРС студента

В данной главе описаны методологии моделирования и выбрана подходящая, а также рассмотрены средства моделирования для данной методологии и спроектированы модели в рамках текущей темы работы.

* 1. Обзор и выбор методологии моделирования

В настоящее время современные информационные системы становятся все более сложными в связи с обилием функциональности, которой они предоставляют. Довольно часто, функции и процедуры, классы и методы, могут видоизменяться, объединяться, выделяться в независимые модули, а также модифицироваться. В целях повышения организации проектирования ИС наиболее оптимальным способом и улучшения качества приложения, прежде всего, нужно смоделировать основные процессы. Для этой цели используются различные методологии моделирования.

В связи с этим крайне важно выбрать подходящую методологию для проектирования, чтобы в достаточной степени описать протекающие процессы внутри системы. В конечном итоге, именно от этого зависит, достаточно ли наглядной, удобной и понятной окажется модель.

Выделим основные критерии, которые будут играть основную роль при выборе методологии моделирования:

* полнота и выразительность: Методология должна обеспечивать достаточную гибкость и возможности для полного описания информационной системы, ее компонентов, взаимодействия и поведения;
* простота и понятность: Методология должна обеспечивать понятные обозначения, правила и соглашения, чтобы все люди могли эффективно использовать и воспринимать созданные диаграммы;
* гибкость и расширяемость: Методология должна предоставлять возможность адаптации и расширения в соответствии с требованиями.

Рассмотрим наиболее популярные методологии проектирования [10]:

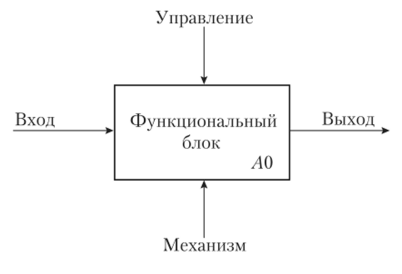
IDEF (Integrated Definition for Function Modeling) [11]

IDEF (Integrated Definition) – это целый набор аналитических средств, применяемых не только в управлении бизнесом, но и во многих других сферах. . С использованием диаграмм IDEF можно эффективно представлять и анализировать модели деятельности сложных систем в разных аспектах. Пользователь сам определяет глубину и широту исследования процессов в системе, чтобы избежать излишней загруженности модели ненужной информацией.

В методологии IDEF есть множество стандартов, но IDEF0 – наиболее часто используемая методология функционального моделирования.

IDEF0 (Рис. 6) имеет несколько преимуществ:

* наглядное представление процесса, имеющего входы/выходы, ресурсы и управление;
* декомпозиция, с помощью которой можно разбить сложный процесс на составляющие ее функции, при этом уровень детализации определяется разработчиком;
* глоссарий является набором из соответствующих определений, которые характеризуют объект и описывают сущность элемента IDEF0.



Рисунок

– Пример диаграммы IDEF0

Таким образом, при проведении сложных проектов обследования предприятий, разработка моделей в стандарте IDEF0 позволяет наглядно и эффективно отобразить весь механизм деятельности проекта в нужном разрезе.

BPMN (Business Process Management Notation) [12]

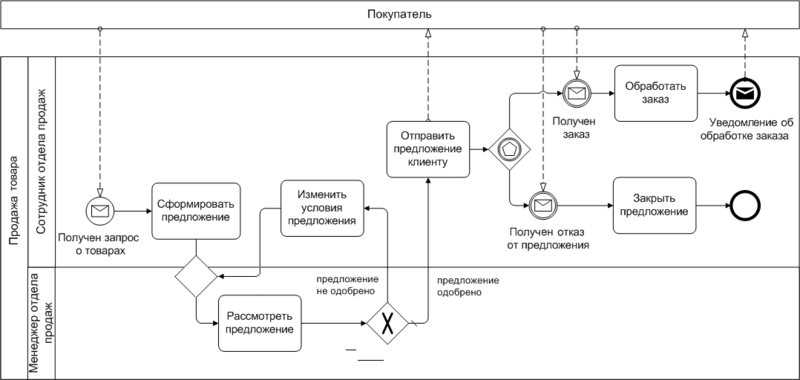
BPMN (Рис. 7) – это язык моделирования бизнес-процессов, который используется для формализации и визуализации различных аспектов процессов в организации. Он служит промежуточным звеном между абстрактным описанием бизнес-процесса и его конкретной реализацией.

Одной из важных особенностей BPMN является его способность поддерживать автоматизацию бизнес-процессов на основе созданной модели. Это означает, что разработанная схема BPMN может быть использована для реализации процесса в автоматизированной системе, где определенные шаги и действия могут быть выполнены компьютером или другими программными средствами.

Достоинства BPMN:

* простота трансляции диаграмм в исполняемые модели с помощью языка формального описания бизнес-процессов;
* доступность и понятность диаграмм для большинства участников бизнес-процессов, зачастую не требующих дополнительного разъяснения;
* импорт готовых схем бизнес-процессов в BPM-системы.

Иначе говоря, BPMN-диаграммы рассчитаны на всех участников бизнес-процессов, с целью получения доступной картины задействованных этапов. Отличительной особенностью является то, что схемы, созданные в среде моделирования BPMN, можно преобразовать в исполняемый файл в BPM-системах.



Рисунок

– Пример диаграммы BPMN

UML (Unified Modeling Language) [13]

UML – стандартизированный язык моделирования, предназначенный для визуализации, специфицирования, конструирования и документирования процессов, создаваемых в ходе разработки. Методология UML – это определенное множество уникальных диаграмм, которые используются для демонстрации ИС с разных сторон.

Главная цель UML-диаграмм – это визуализация. Благодаря простым правилам моделирования и интуитивно понятным элементам диаграмм, такие схемы могут понимать все заинтересованные лица. Также схемы могут использоваться как средство общения между людьми. Вместе с тем, можно изобразить информационную систему с разных сторон в разных диаграммах, которые можно показать заказчику и обсудить с ним.

К тому же UML позволяет строить модели программных систем. По этим моделям потом может производиться генерация каркасного кода проектируемых приложений. Более того, возможен процесс «реверс-инжиниринга», создание UML-модели из существующего кода приложения.

Важно отметить, что UML-модели являются самостоятельными документами. Они предоставляют информацию о системе и ее процессах. Каждый элемент на диаграмме может быть дополнен пояснительной запиской, что способствует более полному пониманию описываемого процесса. Это делает UML не только инструментом визуализации, но и средством документирования, которое помогает в дальнейшем разработке и понимании системы.

Таким образом, UML можно использовать для рисования диаграмм, которые применить для коммуникаций внутри команды и в ходе взаимодействия с заказчиком, т. е. он может служить средством обмена информацией. Он также может использоваться для спецификации систем в процессе разработки. Важным преимуществом UML является возможность повторного использования разработанных архитектурных решений, которые были задокументированы с помощью этого языка.

В результате обзора наиболее популярных методологий проектирования, для проектирования и документирования информационной системы учета времени самостоятельной работы студентов в компьютерном классе больше всего подойдет UML, поскольку совокупность методов, которая содержит данная методология, позволит описать ИС наиболее полно, в виду того, что UML является набором множества диаграмм, с помощью которых можно представить систему с разных точек зрения. Таким образом, UML соответствует требованиям полноты, выразительности, простоты, понятности, гибкости и расширяемости, что делает его наиболее подходящим выбором для проектирования и документирования данной информационной системы.

В том числе, созданные диаграммы позволяют решить проблему документирования системной архитектуры и всех ее деталей. Схемы, разработанные в UML, будут служить документами, которые объясняют, как устроена система, даже для людей, не относящихся к этой предметной области.

* 1. Выбор CASE-средства проектирования ИС

Для эффективного проектирования с методологией UML, нам потребуется соответствующее CASE-средство.

Под термином CASE-средства [14] понимаются программные средства, поддерживающие процессы создания и сопровождения ИС, включая анализ и формулировку требований, проектирование прикладного ПО (приложений) и баз данных, генерацию кода, тестирование, документирование, обеспечение качества, конфигурационное управление и управление проектом, а также другие процессы. CASE-средства вместе с системным ПО и техническими средствами образуют полную среду разработки ИС.

В настоящее время создано множество CASE-средств, каждое из которых ориентировано на те или иные процессы жизненного цикла разработки ИС.

Для проектирования и документирования ИС учета времени СРC в компьютерном классе, выделим основные критерии для выбора программного продукта:

* поддержка UML;
* наличие бесплатной версии;
* поддержка операционной системы Windows;
* простой и интуитивно понятный интерфейс.

Исследуем несколько CASE-средств наиболее подходящих под критерии для проектирования ИС:

* + 1. StarUML [15]

StarUML - это мощное средство моделирования программного обеспечения, которое обеспечивает быстрое и эффективное моделирование. Оно доступно для платформ macOS, Windows и Linux. Это средство моделирования тесно следует спецификации UML, разработанной OMG для моделирования программ. Оно применимо к любым методологиям и процессам благодаря своему концептуальному подходу. С помощью StarUML легко создавать модели не только для конкретных платформ, таких как .NET или J2EE, но и для других основных структур программных моделей. Кроме того, оно поддерживает обратный инжиниринг, позволяя создавать диаграмму классов на основе уже существующего кода.

Среда разработки StarUML на хорошем уровне настраивается в соответствии с требованиями пользователя и имеет высокую степень расширяемости, особенно в области своих функциональных возможностей.

Распространяется бесплатно, без ограничения по времени, для оценки работоспособности программы. Однако, при сохранении моделей, на них наносятся водяные знаки с пометкой «UNREGISTERED».

* + 1. Violet UML Editor [16]

Violet UML Editor – редактор UML диаграмм, предназначенный для разработчиков, студентов, учителей и авторов, которым необходимо быстро создавать простые UML диаграммы. Данный редактор имеет следующие преимущества:

* простой и понятный интерфейс;
* поддержка основных видов диаграмм UML;
* распространяется бесплатно;
* кроссплатформенность.

Тем не менее, Violet UML Editor имеет ряд существенных недостатков, таких как недостаточно хорошая проверка семантики, т.е. присутствует возможность нарисовать пользователю противоречивые диаграммы. Недостаточная проверка семантики означает, что пользователь может случайно создать диаграммы, которые противоречат основным принципам моделирования. Это может привести к неправильному пониманию модели и возникновению ошибок в процессе разработки программного обеспечения. Также, у Violet UML Editor, нет импорта/экспорта XMI документов, нет генерации кода из диаграммы, нет обратного инжиниринга. Таким образом, эти недостатки могут затруднить процесс моделирования и разработки программного обеспечения, а также ограничить возможности совместной работы и автоматизации.

* + 1. Diagram Designer [17]

Diagram Designer – бесплатное решение для создания блок-схем, диаграмм и слайд-шоу. Данная программа распространяется с открытым кодом, и любой человек может модифицировать код, что определенно является достоинством. Основные функции приложения:

* построение простых диаграмм классов;
* проверка орфографии;
* настраиваемая палитра шаблонов;
* «карманный» калькулятор для решений уравнений;
* импорт/экспорт изображений.

Однако, следует отметить, что Diagram Designer не является лучшим выбором для проектирования информационных систем, поскольку она поддерживает только диаграммы классов и не предоставляет поддержку других типов UML-диаграмм. Это является существенным недостатком. Кроме того, отсутствует функция генерации кода на основе диаграммы, что также является значительным ограничением.

Вместе с этим, Diagram Designer представляет собой полезное решение для создания простых диаграмм и блок-схем, однако, его возможности ограничены и не подходят для более сложных проектов, особенно в области проектирования информационных систем.

* + 1. ArgoUML [18]

ArgoUML — средство UML моделирования. ArgoUML является открытым программным обеспечением и распространяется под лицензией EPL. ArgoUML полностью написан на Java и для работы ему подходит любая операционная система с установленной Java 2 JRE или JDK версии 1.4 или выше. Поддерживает спецификации UML 1.4, генерацию исходного кода, обратный инжиниринг из исходного кода.

Однако, интерфейс ArgoUML не интуитивно понятен, что делает его сложным для новых пользователей системы.

Исходя из всех вышеперечисленных CASE-средств, StarUML наиболее подходит для проектирования ИС, учитывая следующие факторы:

* поддержка UML: StarUML обеспечивает полную поддержку UML, что является важным критерием для CASE-средства при проектировании ИС;
* наличие бесплатной версии: Это CASE-средство распространяется бесплатно без ограничения по времени;
* поддержка операционной системы Windows: Он полностью совместим с операционной системой Windows;
* простой и интуитивно понятный интерфейс: StarUML обладает дружелюбным и понятным интерфейсом, что делает его доступным и удобным в использовании для новых пользователей системы.

Учитывая эти факторы, CASE-средство StarUML является наиболее подходящим для проектирования информационных систем с учетом указанных критериев.

* 1. Моделирование информационной системы

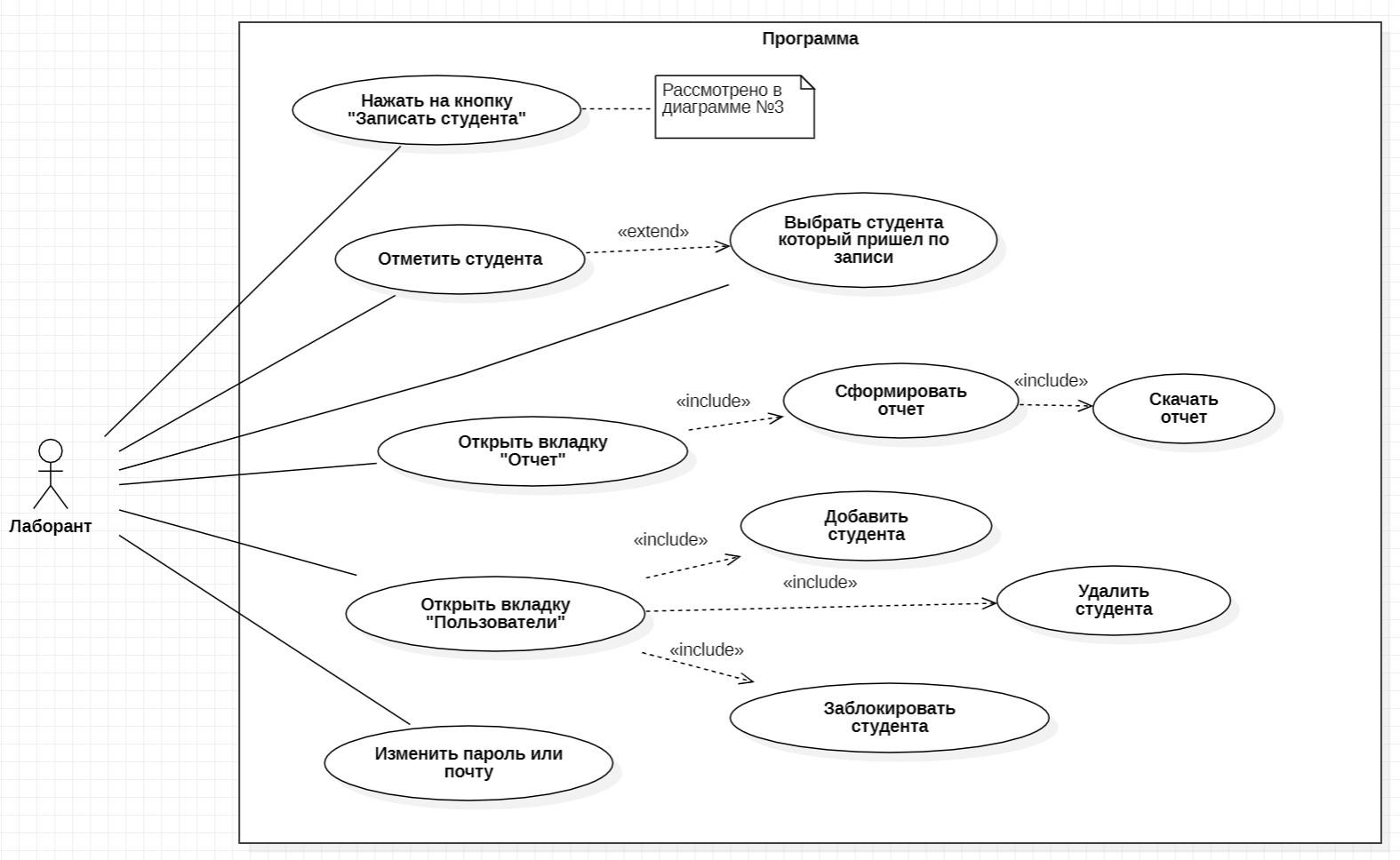
При моделировании ИС будем опираться на задачи, определенные ранее. Доступ к ИС для лаборанта и для студентов будет осуществляться через интернет-сайт.

Для достижения целей визуального проектирования с использованием нотации UML вначале строится диаграмма вариантов использования. Диаграмма вариантов использования помогает исследовать и документировать различные случаи использования системы и ее функциональные возможности.

Диаграмма вариантов использования – это исходное концептуальное представление или концептуальная модель системы в процессе ее проектирования и разработки [19]. Можем выделить главные задачи диаграммы:

* определить общие границы и контекст моделируемой предметной области на начальных этапах проектирования системы;
* сформулировать общие требования к функциональному поведению проектируемой системы;
* разработать исходную концептуальную модель системы для ее последующей детализации в форме логических и физических моделей;
* подготовить исходную документацию для взаимодействия разработчиков системы с ее заказчиками и пользователями.

Таким образом, логика работы информационной системы представлена на следующих рисунках (Рис. 8,Рис. 9).

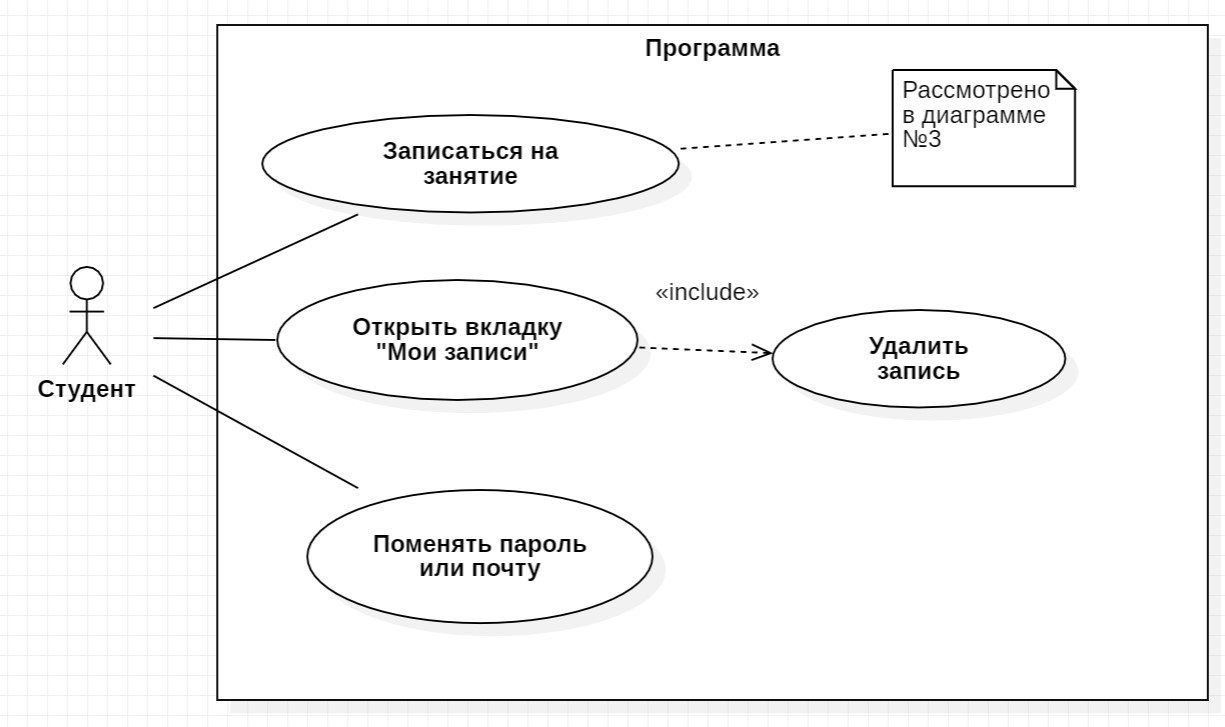


Рисунок

– Диаграмма прецедентов «Лаборант» №1

В процессе изучения диаграммы прецедентов «Лаборант» №1, изображенной выше (Рис. 8), можно выделить 5 основных действий, которые может совершить лаборант при работе с ИС:

* записать студента;
* отметить присутствие студента, т. е. изменить статус посещения;
* открыть отчет о студентах, и, при надобности, скачать;
* изучить пользователей-студентов, а также добавить, заблокировать и удалить. Заблокированный пользователь больше не сможет входить в систему.
* изменить email или пароль.

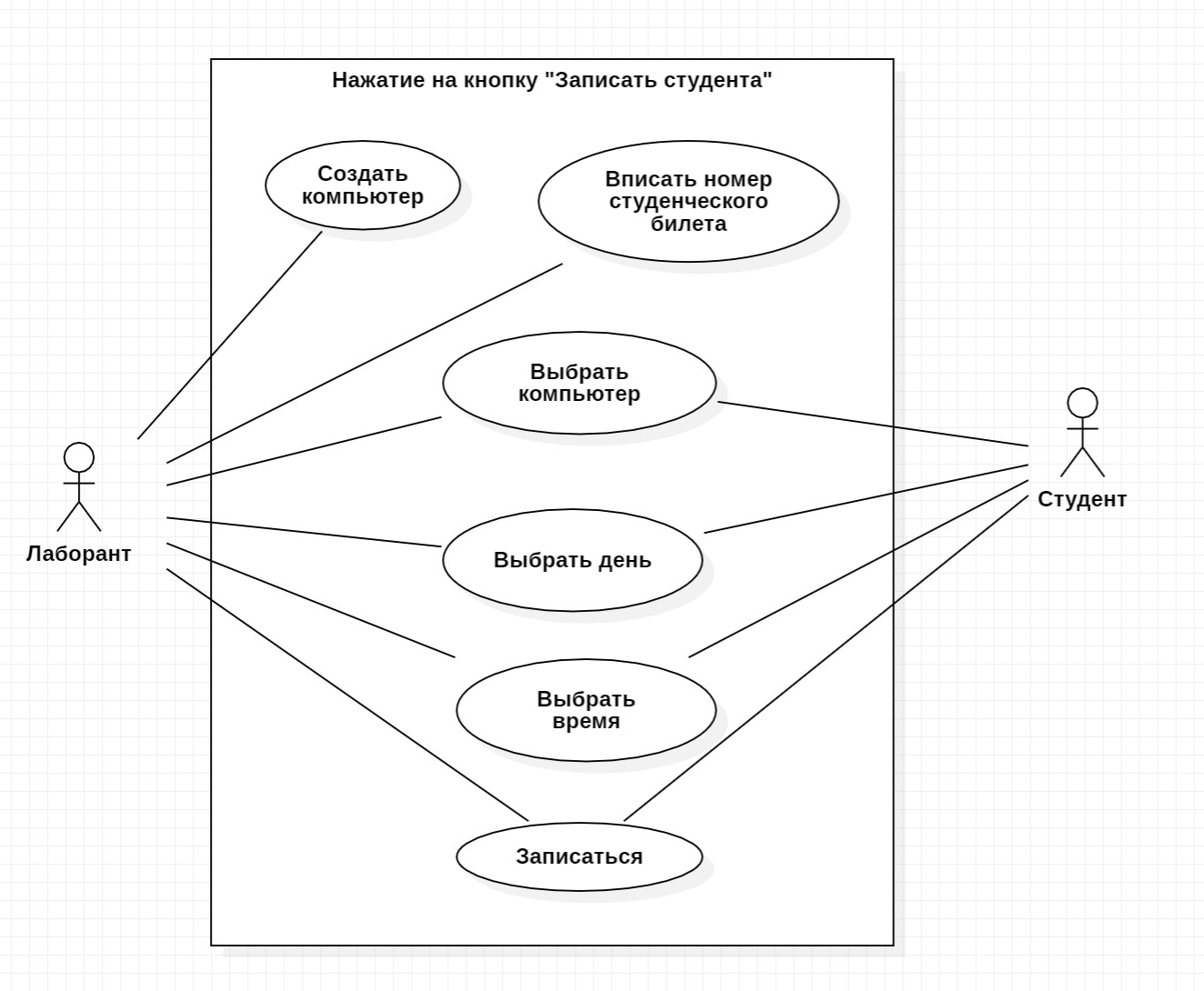


Рисунок

9 – Диаграмма прецедентов «Студент» №2

Рассмотрим диаграмму прецедентов «Студент» №2 (Рис. 9). На ней можно выделить 3 действия:

* записаться на занятие в компьютерный класс;
* посмотреть свои записи и удалить, при надобности, уже существующую запись;
* поменять email или почту.



Рисунок

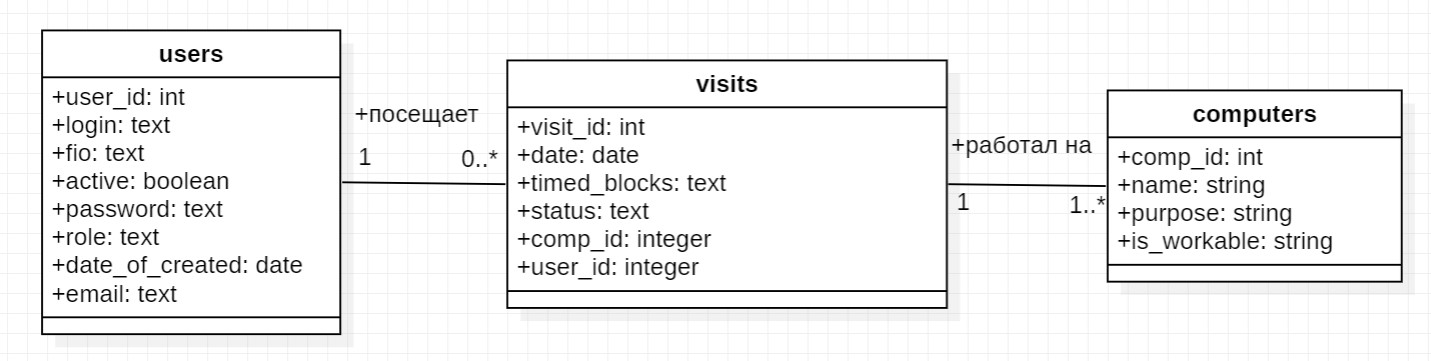
10– Диаграмма прецедентов «Записать студента» №3

Рассмотрим диаграмму прецедентов №3 (Рис. 10). На ней изображены два актера: студент и лаборант. Лаборант, помимо записи в компьютерный класс, может создать компьютер для последующей записи студентов. ИС подразумевает, что для записи студента на свободный компьютер лаборанту, необходимо осуществить следующие операции в порядке, который указан далее:

* + 1. выбрать компьютер, соответствующий требованиям и пожеланиям студента;
    2. выбрать пришедшего студента, по номеру его студенческого билета. Однако, если студента нет в системе, указать информацию о студенте, а именно: имя, фамилию, номер студенческого билета, после чего пароль сгенерируется автоматически;
    3. выбрать интересующий день для работы в компьютерном классе;
    4. выбрать свободное время;
    5. сохранить запись.

Для студентов, которые хотят записаться в компьютерный класс, не требуется указывать свой студенческий номер, поскольку система авторизации уже точно определяет, какой пользователь работает с системой. При создании записи автоматически подставляется информация о текущем студенте. Таким образом, процесс записи для студента аналогичен процессу для лаборанта, за исключением ввода студенческого номера.

Далее рассмотрим диаграмму классов. Диаграмма классов — это диаграмма языка UML, на которой представлена совокупность декларативных или статических элементов модели, таких как классы, с атрибутами и операциями, а также связывающие их отношения [19]. Диаграмма классов предназначена для представления статической структуры модели системы в терминологии классов объектно-ориентированного программирования.



Рисунок

– Диаграмма классов

Следовательно, на диаграмме классов (Рис. 11) представлены 3 класса:

* users - «Пользователи»;
* computers - «Компьютеры»;
* visits - «Посещения компьютерного класса».

Сущность «Пользователи» нужна для хранения информации о пользователях, и поможет организовать авторизацию и аутентификацию в проектируемой системе. Данный класс имеет следующие атрибуты:

* user\_id(int) – является первичным ключом, именно по этому атрибуту можно определить запись в таблице;
* login(string) – для студента является студенческим номером, по которому можно определить студента; для лаборанта – любой набор символов, не пересекающийся с другими логинами пользователей;
* fio(string) – ФИО пользователя для упрощенного понимания, чей это аккаунт;
* active(boolean) – статус аккаунта определяет, может ли пользователь работать с системой. Если статус установлен в «true», то пользователь имеет доступ и может авторизоваться на сайте. В противном случае, если статус установлен в «false», пользователь заблокирован и не имеет права на доступ к системе. Заблокировать пользователя может лаборант;
* password(string) – этот атрибут необходим для авторизации студента через сеть, хранение будет осуществляться в виде захэшированной строки для того, чтобы злоумышленники не смогли понять, какой пароль у пользователя;
* role(string) – в разрабатываемой системе может принимать значения «ROLE\_STUDENT» и «ROLE\_WORKER» для разграничения информации, представляемой в системе, а также для доступа к разным веб-страницам;
* date\_of\_created(date) – дата создания пользователя;
* email(string) – электронная почта для связи с пользователем, также почта будет использоваться для отправки пароля для студента.

Сущность «Компьютеры» необходима для хранения информации о компьютерах в классе и имеет 4 атрибута:

* comp\_id(int) – является первичным ключом, именно по этому атрибуту можно определить запись в таблице;
* name(string) – наименование компьютера;
* purpose(string) – назначение компьютера (подходит ли этот компьютер для задач студента)
* is\_workable(bool) – отвечает за работоспособность компьютера.

И главная сущность «Посещения компьютерного класса», с которой ИС будет работать чаще всего. Она требуется, потому что именно в ней мы будем хранить историю посещений и записи студентов на самостоятельную работу. Благодаря этой таблице, мы сможем определять свободные компьютеры в информационной системе. Данная таблица имеет 6 атрибутов:

* visit\_id(int) – является первичным ключом, именно по этому атрибуту можно определить запись в таблице;
* date(date) – дата посещения студента;
* timed\_blocks(string) – интервал блоков времени, на которое записался студент: рабочее время компьютерного класса с 9-00 до 18-00, студент может записываться с промежутком в 15 минут, и если он запишется с 9-00 до 9-30, то в базу данных это время запишется, как «0-2». Таким образом, это поле может принимать интервал от 0 до 35, так как в расписании с 9-00 до 18-00 – 36 пятнадцатиминутных блоков;
* status(string) – состояние, в котором находится эта запись:

1. («sign»(Записался(лась)) – студент записался, но еще не пришел и время не просрочено;
2. «attended»(Посетил(а)) – студент пришел в компьютерный класс;
3. «absent»(Не посетил(а)) – студент не пришел в заявленное время;
4. «undefined»(Не установлено) – вспомогательное состояние.

* comp\_id(int) – идентификатор компьютера для определения за каким устройством будет работать или уже работал студент;
* user\_id(int) – идентификатор студента, чтобы понять, какой студент записался на занятие или уже его отработал.

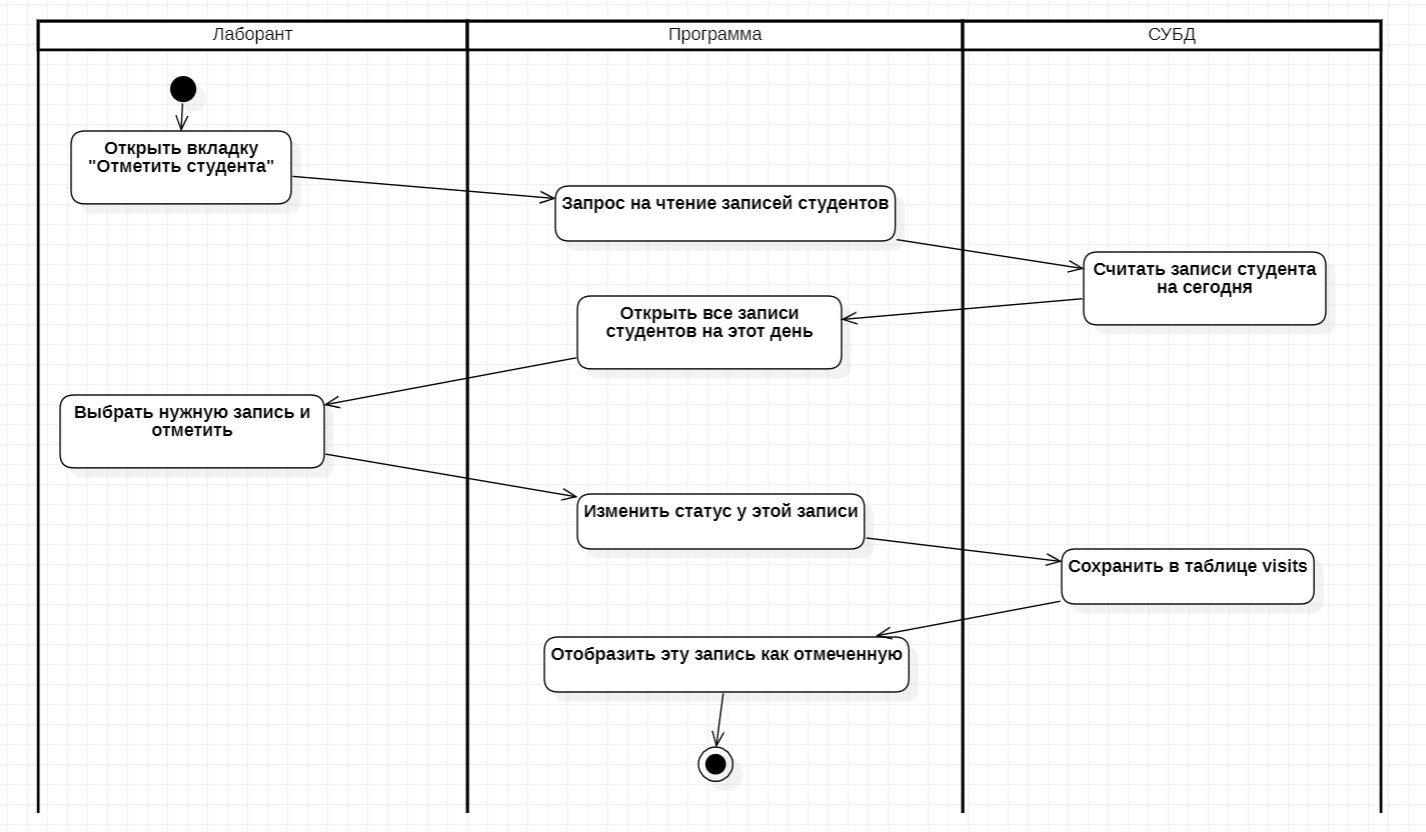
Для полного представления процесса записи студента на СРС через информационную систему рассмотрим диаграммы последовательностей с точки зрения лаборанта (Приложение А) и студента (Приложение Б).

Рассматривая диаграмму последовательностей с точки зрения лаборанта, можно заметить, что если студента нет в ИС, лаборант может внести его в базу данных, тем самым добавив нового пользователя. Тогда этого студента можно будет записать на занятие в компьютерный класс. Также при создании, пароль сгенерируется системой автоматически, и будет отправлен на email студенту. Вместе с тем при выборе компьютера, дня и времени, пользователю не будет предложено неактуальных для записи данных, тем самым ускоряя процесс обработки информации, и не перегружая пользователя ненужной информацией. Даже если студента не устраивает предложенный день или время, лаборант может вернуться на шаг назад и выбрать другой компьютер.

Обратив внимание на запись с точки зрения студента, пользователю не нужно регистрировать себя и вписывать свой логин каждый раз, чтобы записаться на занятие. Вся необходимая информация о студенте хранится в базе данных, и при необходимости система сама может определить, кто с ней сейчас работает. Поэтому ИС сразу начинает запись с определения необходимого компьютера. Как только студент выберет компьютер, дату и время, пользователя перенаправит на его существующие записи для того, чтобы он был в курсе, когда и на какой компьютер ему нужно приходить в ближайшее время. Также на этой вкладке можно удалить запись, если студент передумает идти на занятие.

При записи на самостоятельную работу в компьютерный класс, может возникнуть множество ситуаций, когда студента не будут устраивать компьютер, день или время, поэтому в диаграмме присутствует ряд возвратов к тем вопросам, которые решались ранее, для того чтобы наиболее полно решить вопрос записи в удобное время.

Также рассмотрим некоторые тонкости при функции ИС «Отметить студента» (Рис. 12), когда студент приходит в назначенное время в компьютерный класс, и лаборант должен указать, что студент пришел в назначенное время. Для этого лаборант отмечает нужную запись, тем самым меняя статус данного посещения на «Посетил(а)» или «Не посетил(а)».



Рисунок

– Диаграмма последовательностей «Отметить студента»

* 1. Вывод

Итак, в ходе проведения исследования был выполнен анализ существующих методологий моделирования с целью выбора оптимального подхода. На основании данного анализа была выбрана конкретная методология, которая лучше всего соответствует требованиям и целям нашего проекта.

Дополнительно, с учетом выбранной методологии, был произведен выбор и определение CASE-средства, в соответствии с его возможностями и соответствием требованиям проекта, которое будет использоваться для моделирования информационной системы.

С целью разработки функциональности информационной системы в различных сценариях, был применен объектно-ориентированный подход моделирования, основанный на языке UML. В процессе моделирования были учтены различные аспекты задачи, с учетом требований, как со стороны студента, так и со стороны лаборанта.

Результаты полученного моделирования были подробно задокументированы с использованием UML-диаграмм. Эти диаграммы выполняют функцию пояснительной записки для заинтересованных лиц, не обладающих предварительным знанием о данной информационной системе.

1. Разработка прототипа информационной системы учета СРС студента

В данной главе описан этап прототипирования интерфейса, выбор инструментальных средств, разработка прототипа информационной системы, а также представлены основные рекомендации по совершенствованию разработанной информационной системы

* 1. Прототипирование интерфейса ИС

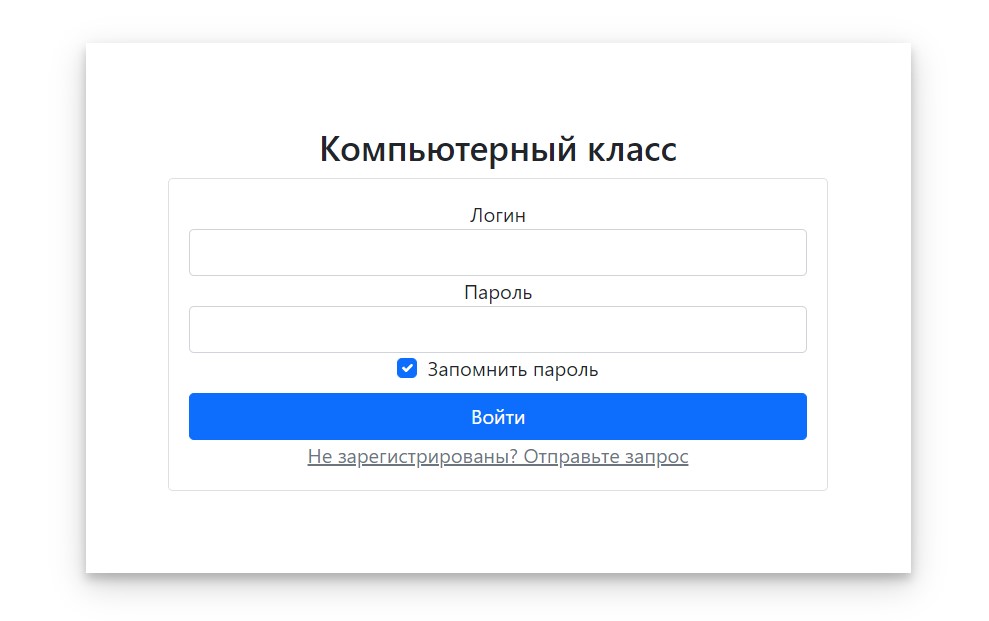
Прототипирование интерфейса информационной системы играет важную роль в процессе разработки. Оно позволяет визуализировать и представить концепцию ИС на ранней стадии, помогая команде разработчиков и заинтересованным сторонам лучше понять, как будет выглядеть, и работать окончательный продукт. Создание прототипа также позволяет уточнить требования и получить обратную связь от пользователей, что помогает выявить и исправить потенциальные проблемы и улучшить пользовательский опыт.

В рамках данной работы прототипирование интерфейса является важной частью, поскольку оно позволяет проверить и протестировать взаимодействие пользователей с ИС, выявить возможные проблемы и несоответствия на ранних этапах разработки, что способствует экономии времени и ресурсов. В связи с этим, прототипирование интерфейса является важным шагом для демонстрации и оценки работоспособности, удобства использования и соответствия требованиям разработанной информационной системы.

* + 1. Интерфейс студента

Первым делом рассмотрим интерфейс для удаленной записи студента через сайт, так как именно таким способом чаще всего будут взаимодействовать пользователи с ИС.

При открытии любого каталога сайта студент увидит первым следующее окно (Рис. 13)

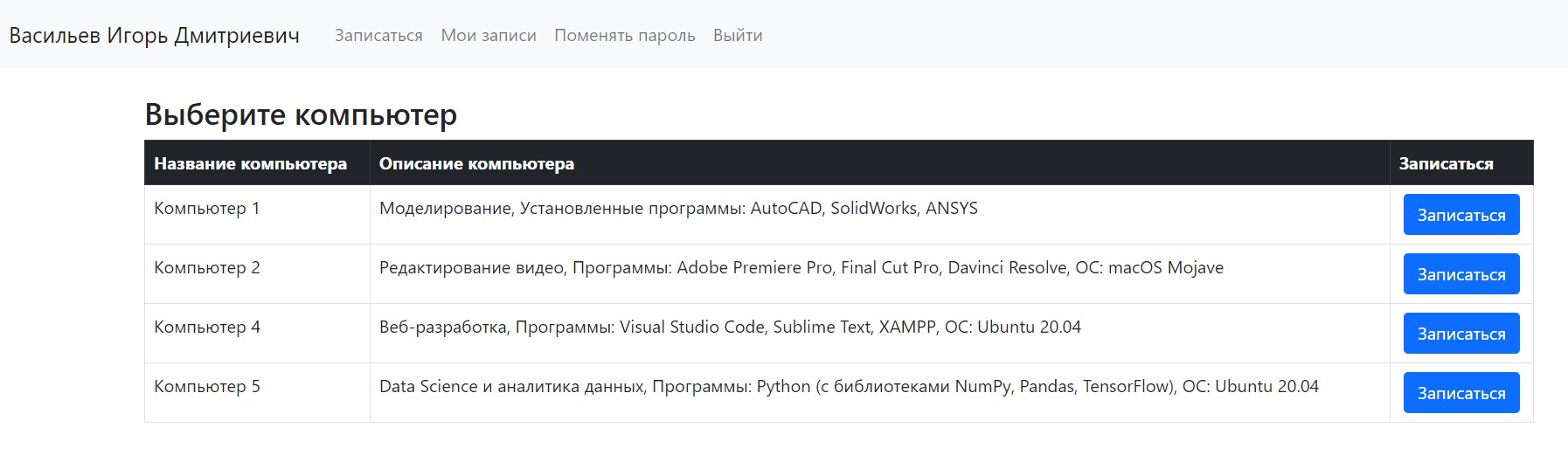


Рисунок

– Авторизация

Если студент уже записывался ранее, ему необходимо ввести номер своего студенческого билета и пароль, тогда он сможет нажать кнопку «Войти» и перейти непосредственно записи на СРС в компьютерный класс. Но в то же время, если студент впервые заходит на этот сайт ему следует отправить запрос лаборанту, подтвердив свою личность, назвав номер студенческого билета, ФИО и дату поступления в университет, а также почту для регистрации. Таким же образом можно подойти к лаборанту лично, чтобы он зарегистрировал студента в информационной системе для дистанционной записи через сайт. С другой стороны, если у ВУЗа есть возможность массово проинформировать студентов, то можно рассмотреть возможность генерации данных авторизации всех студентов в БД и сообщить студентам их пароли в других информационных системах для дальнейшего использования.

После авторизации студенту открывается окно с выбором подходящего компьютера для его конкретной задачи (Рис. 14).



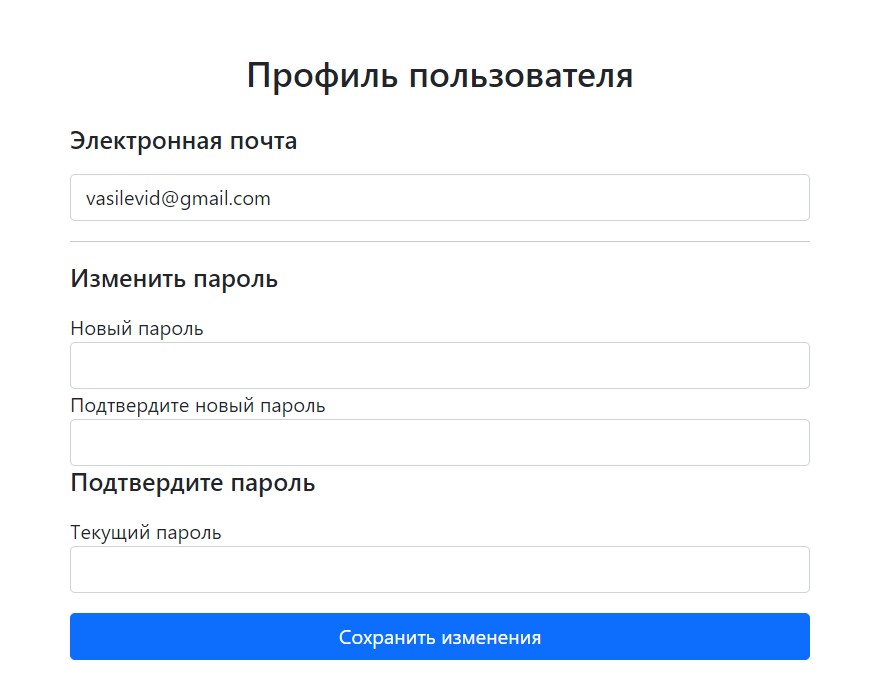
Рисунок

14 – Выбор ПК (Студент)

На данном рисунке (Рис. 14) видно, что в навигации присутствует его ФИО, кроме этого присутствуют ссылки на другие вкладки сайта, а также система сразу же предлагает на выбор компьютеры, указывая необходимую информацию о них. В первом столбце указывается название компьютера, во втором – необходимое описание, для того чтобы студент смог подобрать компьютер в связи со своей задачей, а в третьем столбце кнопка «записаться», привязанная к каждому компьютеру. Также стоит отметить, что в представленной таблице не будут отображаться неработоспособные компьютеры.

В то же время присутствует возможность поменять электронную почту или пароль (Рис. 15). Этот пароль должен соответствовать требованиям безопасности [20] , таким как:

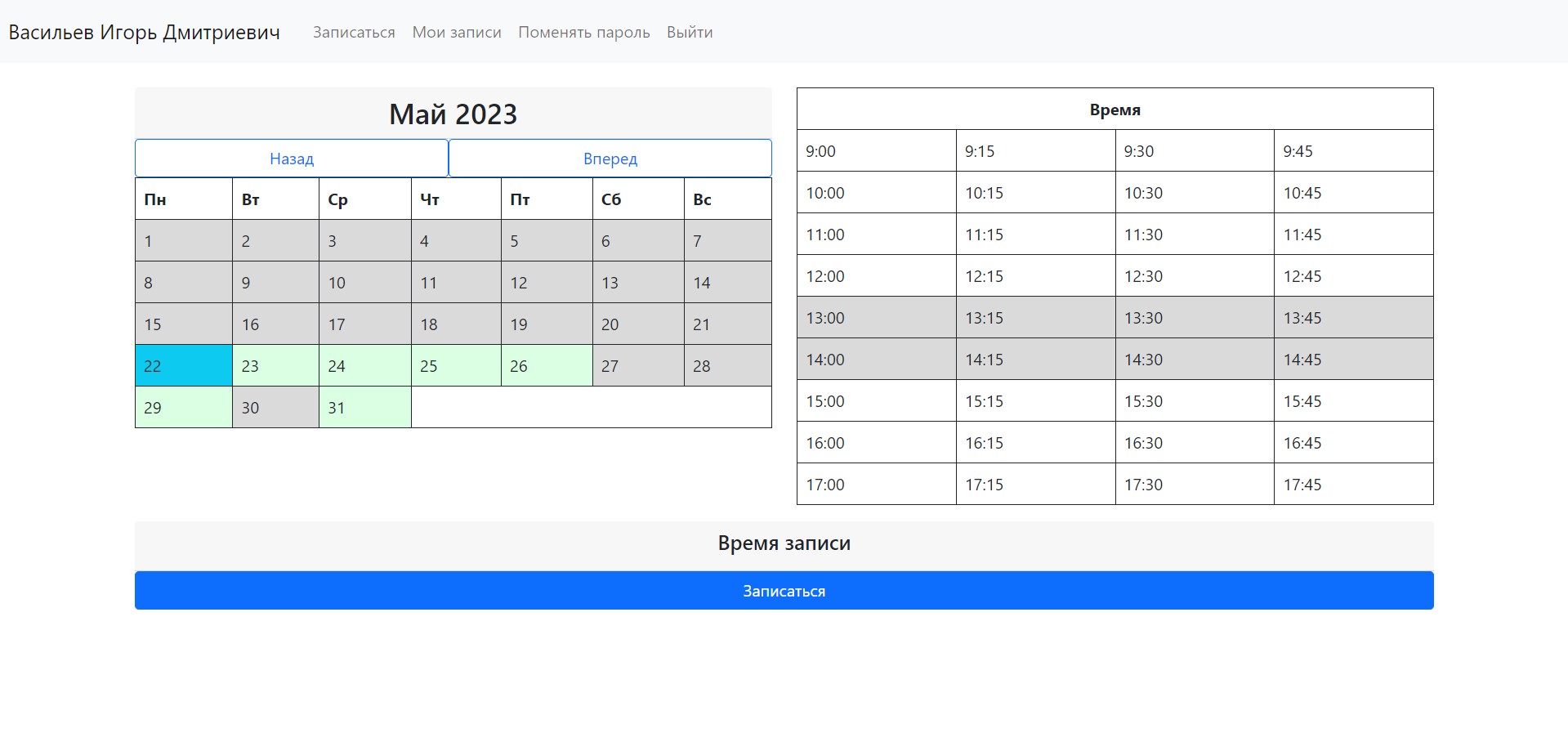
* минимальная длина пароля 8 символов;
* не более 128 символов;
* как минимум одна заглавная и одна строчная буква;
* только латинские или кириллические буквы;
* как минимум одна цифра;
* только арабские цифры;
* без пробелов;



Рисунок

– Изменение почты или пароля

После выбора необходимого компьютера (Рис. 14) перед студентом отображается новая информация о выборе дня и времени посещения (Рис. 16).

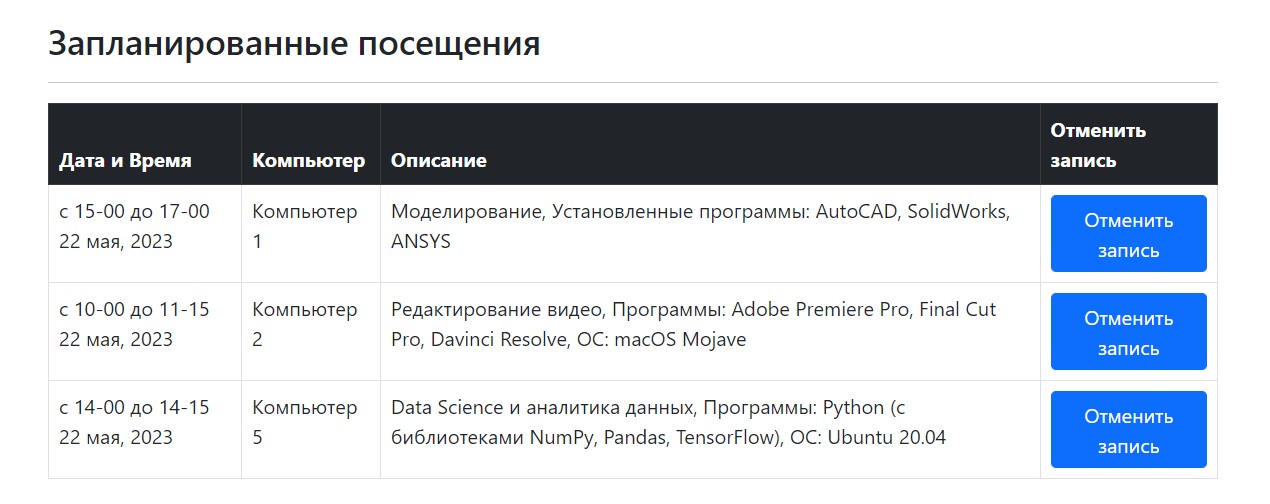


Рисунок

– Запись

На представленном рисунке (Рис. 16) студенту предложено выбрать подходящий день и время. При выборе дня меняются и часы записи (становятся активными или наоборот деактивируются), также стоит уточнить, что нельзя выбрать день, в котором нет свободной записи. В свою очередь, на календаре слева отображается текущий день (темно-синяя ячейка), свободные дни (зеленые ячейки), и недоступные дни (светло-серые). Вместе с этим, справа отображаются часы записи: светло-серым выделены часы, на которые записаться не представляется возможным, также у них отключена возможность выбрать, а все записи, где часы доступны, выделить можно. Выделение времени происходит по блокам, каждый блок – это 15 минут, т.е. 10:15 в ячейке справа – это время с 10:15 до 10:30. Сначала пользователь выделяет первое время, например 10:15, а потом следующее время, например 11:15, и таким образом пользователь записывается на время с 10:15 до 11:30. Таким образом, происходит запись на соответствующий день и время. Вместе с тем, пользователю дублируется время над кнопкой «Записаться», чтобы он не ошибся с записанным временем. Также всегда можно вернуться к выбору компьютера, если студента не устраивает расписание на этом компьютере.

После выбора даты и времени студенту предложено посмотреть его существующие записи (Рис. 17), а также предоставляется возможность отменить запись.

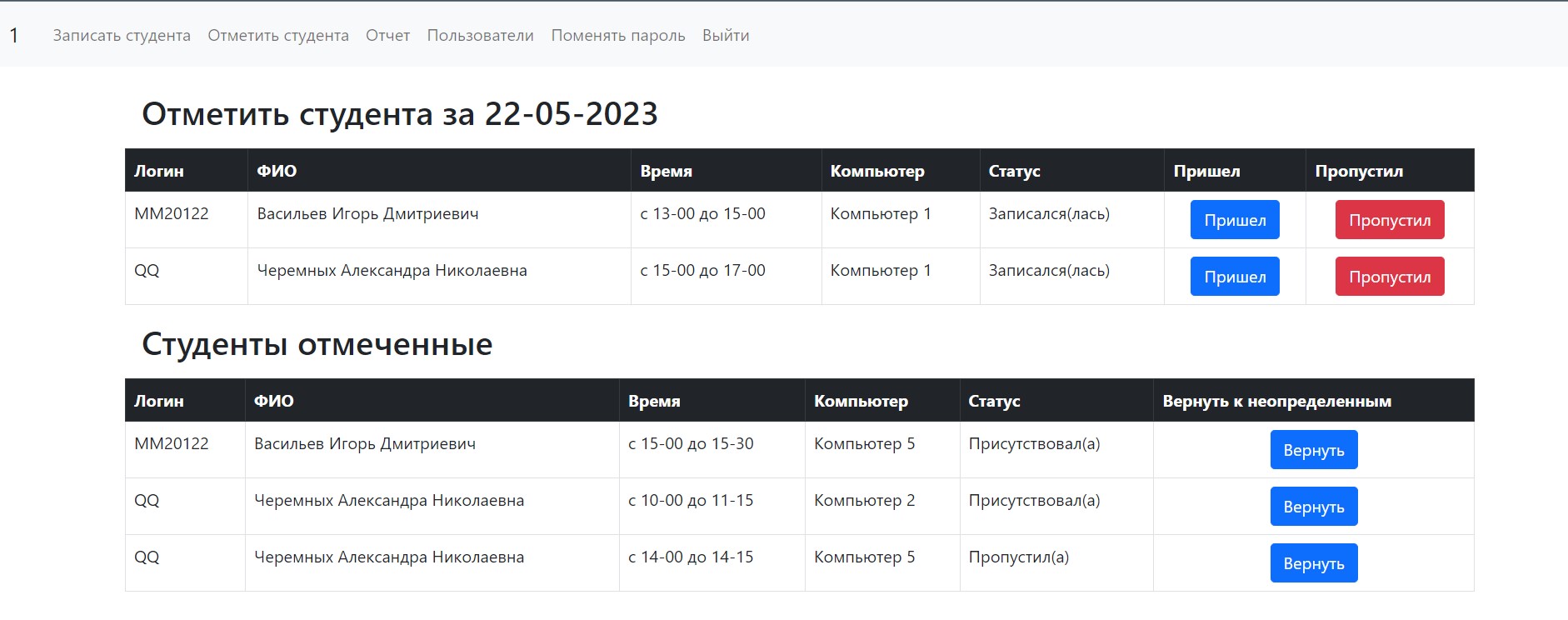


Рисунок

17 – Запланированные посещения студента

* + 1. Интерфейс лаборанта

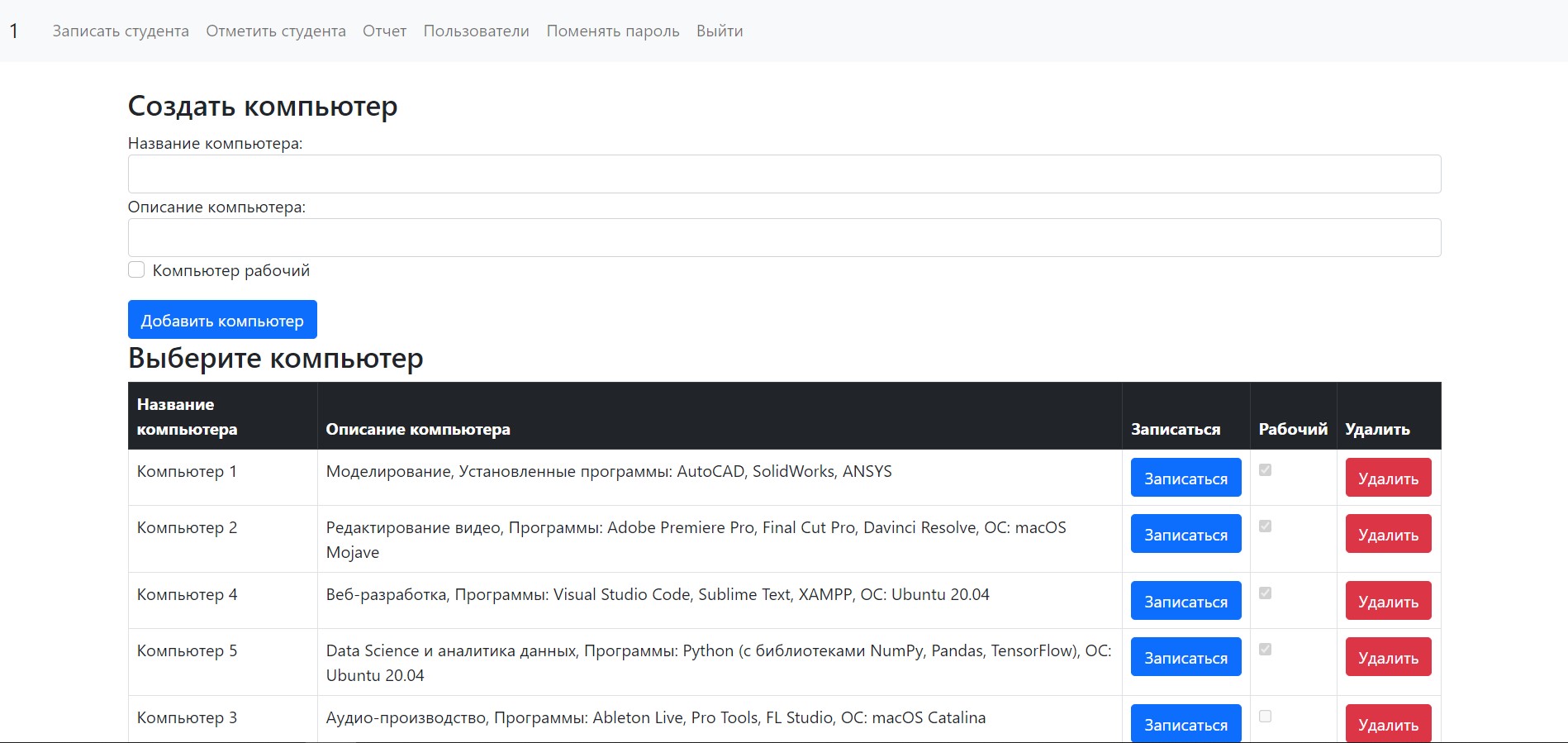
Теперь рассмотрим интерфейс лаборанта. После авторизации лаборанта перенаправит, на страницу «Отметить студента», так как пользователь с ролью «ROLE\_WORKER» чаще всего будет взаимодействовать с системой именно через эту страницу (Рис. 18). На данной странице изображены 2 таблицы, в которых отмечено 2 группы студентов: отмеченные и не отмеченные. Работник может отметить запись в таблицу неотмеченных в зависимости от того, пришел ли студент или пропустил занятие. Таким образом, он попадет в таблицу «отмеченных». Тем же методом можно вернуть студента обратно в таблицу «неотмеченных».



Рисунок

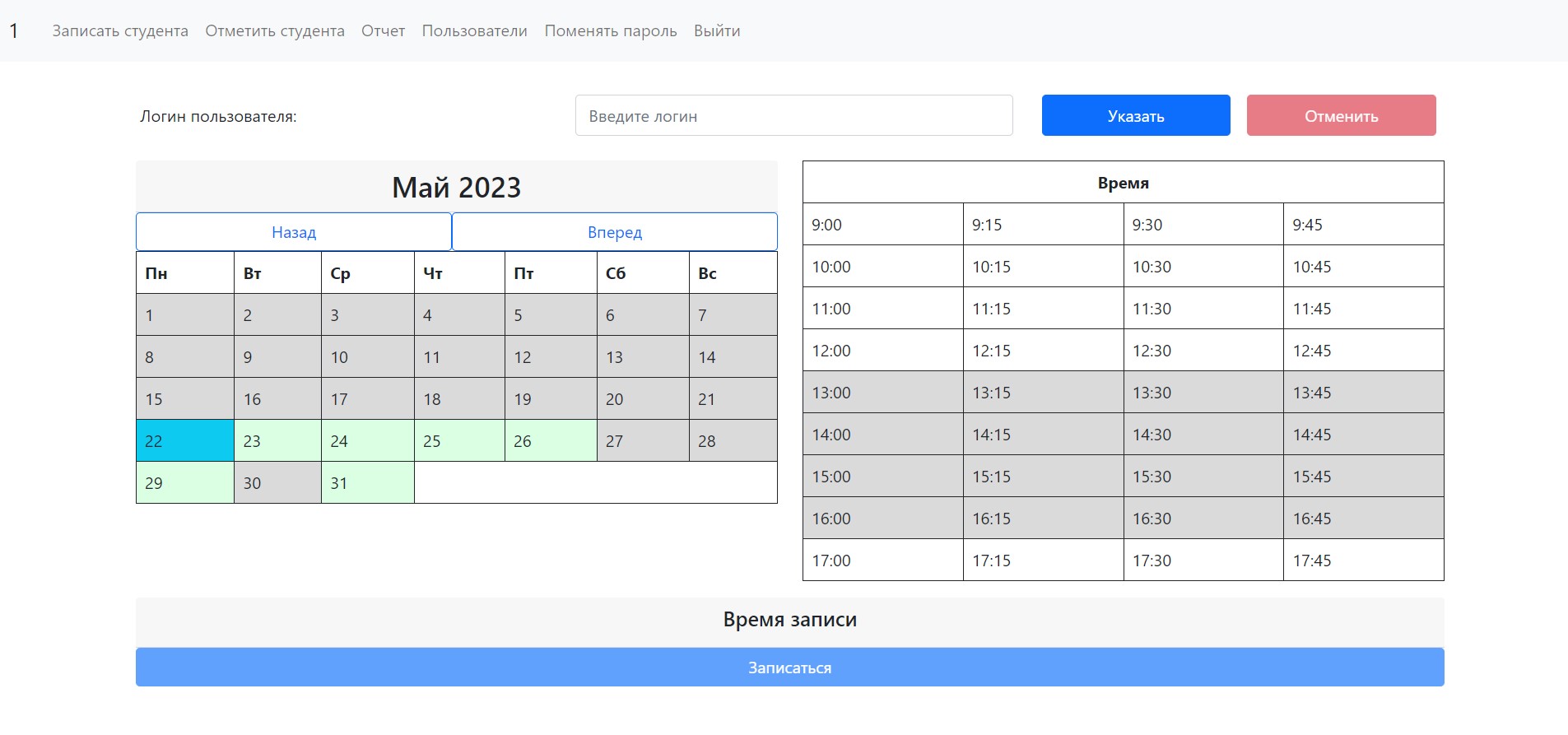
– Отметить студента

Запись студента на конкретный компьютер, день и время, со стороны лаборанта представлена на Рис. 19 и на Рис. 20. При выборе компьютера, у работника университета есть возможность создать новый компьютер, удалить уже существующий, а также у него есть возможность смотреть является ли компьютер рабочим или нет. Во время выбора даты и времени, лаборант обязан выбрать студента с помощью поля ввода логина пользователя, также внизу календаря продублируется ФИО студента, чтобы понять, что это действительно тот студент, которого записывает лаборант.



Рисунок

– Выбор ПК (Лаборант)



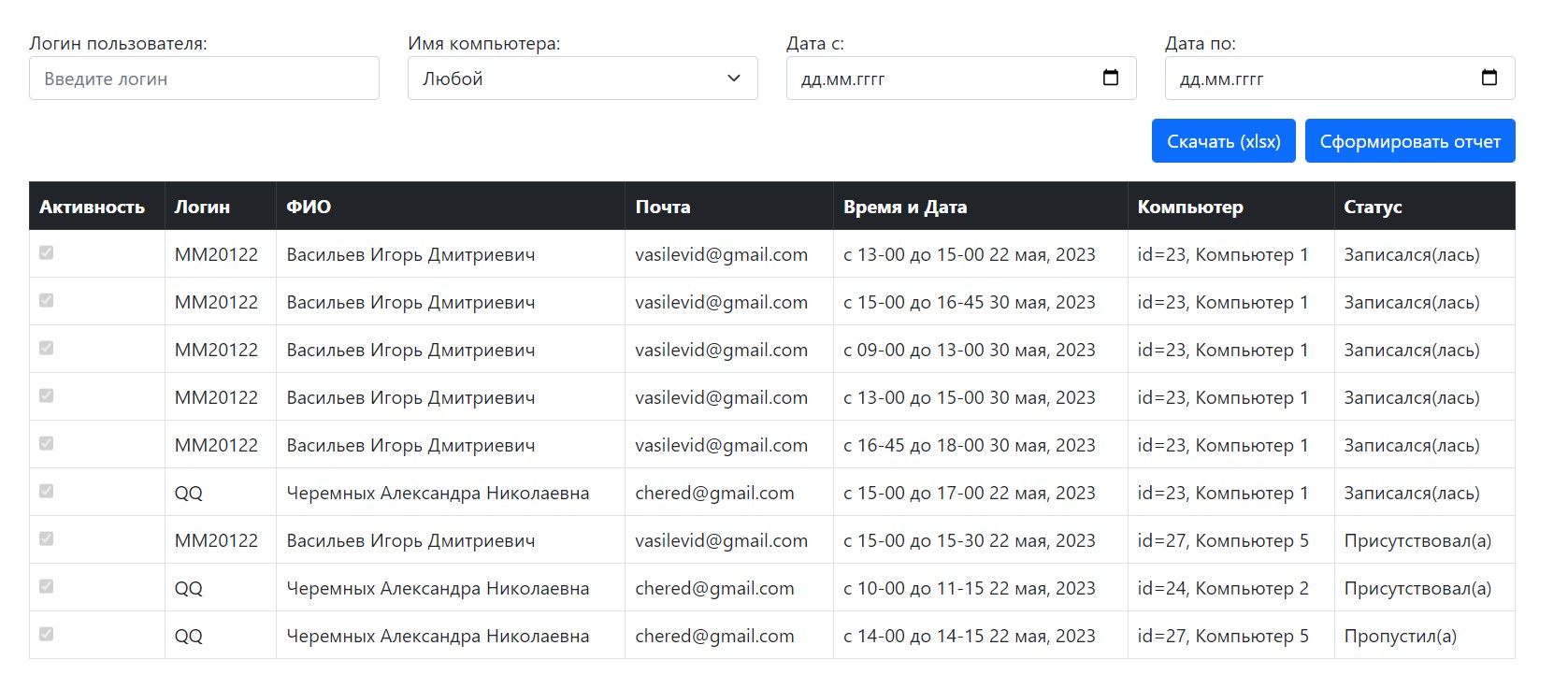
Рисунок

– Отметить студента (Лаборант)

Работник университета имеет возможность просмотра отчета о студентах на странице, которая представлена на Рис. 21. Эта страница позволяет задавать несколько настроек для фильтрации отчета:

* по логину: Работник может указать конкретного человека, по которому нужно произвести фильтрацию отчета. Это позволяет отобразить информацию только для выбранного студента.
* по компьютеру: Работник может выбрать определенный компьютер для фильтрации отчета. Таким образом, отчет будет содержать информацию только о студентах, которые использовали выбранный компьютер.
* по дате "от" и "до": Работник может задать диапазон дат для фильтрации отчета. Отчет будет содержать информацию только о записях, сделанных в указанном диапазоне.

Эти настройки фильтрации позволяют работнику университета получать более точные и специфические данные из отчета о студентах, в зависимости от требуемых критериев и условий. Также сформированный отчет можно скачать в формате электронных таблиц (.xlsx). Пример фильтрации и сгенерированного отчета продемонстрирован в приложении В.



Рисунок

– Отчет

Страница с изменением электронной почты и пароля у лаборанта идентична странице студента. Кроме того, в панели навигации присутствует кнопка "Выйти". При нажатии на эту кнопку, пользователь выходит из аккаунта и перенаправляется на страницу авторизации, где ему требуется ввести учетные данные для входа в систему снова.

* 1. Выбор средств разработки ИС

Необходимо начинать выбор с СУБД потому, что она является фундаментом всей информационной системы, и определение ее характеристик, функций и возможностей помогает определить последующие шаги при выборе других компонентов и инструментов для разработки ИС.

* + 1. Система управления базой данных

Существует множество СУБД, которые можно использовать для создания информационной системы, однако, для данной работы лучше всего подойдет PostgreSQL [21]. Это бесплатная и открытая реляционная СУБД, которая обладает высоким уровнем надежности и безопасности, благодаря использованию транзакций, механизмов резервного копирования и аутентификации пользователей [22]. PostgreSQL также обладает широким набором инструментов и расширений, что позволяет настроить базу данных под любые требования проекта. Благодаря многопоточной обработке эта СУБД может распараллеливать выполнение запросов и ускорять работу с базой данных. Более того, она обладает хорошей масштабируемостью и производительностью, что позволяет ей обрабатывать большой объем данных и обеспечивать высокую скорость выполнения запросов. Таким образом, PostgreSQL является идеальным выбором для создания информационной системы об учете студентов для записи в компьютерный класс.

* + 1. Серверная часть

В информационной системе для учета работы студентов в компьютерном классе наиболее приоритетными составляющими являются безопасность, производительность, надежность и количество разработанных библиотек. Для сочетания всех свойств системы были выбраны следующие технологии: язык программирования Java, фреймворк Spring Boot, библиотека Spring JPA, библиотека Spring Security, обработчик шаблонов FreeMarker и другие библиотеки. Важно отметить, что все эти инструменты являются бесплатными и могут быть использованы в коммерческих проектах без необходимости платить за лицензии.

Java [23] является одним из наиболее широко используемых языков программирования в мире и предлагает множество преимуществ для разработки приложений. В частности, этот язык обеспечивает высокую производительность, надежность и безопасность, что является важным для систем, используемых множеством пользователей. Кроме того, Java имеет масштабируемую архитектуру, что позволяет легко расширять и изменять функциональность приложения.

Spring Boot [24], в свою очередь, является фреймворком, который был разработан для упрощения процесса создания веб-приложений на Java. Он предоставляет широкий набор инструментов и библиотек, которые сокращают время и усилия, необходимые для конфигурации приложения, и позволяют быстро разрабатывать функциональность.

Spring JPA [25] - это библиотека Spring Framework, которая предоставляет удобный способ работать с базами данных в Java приложениях. Она предоставляет высокоуровневый API для управления сущностями. Использование Spring JPA позволяет значительно ускорить и упростить процесс разработки, а также снизить количество ошибок, связанных с работой с базами данных.

Spring Security [26]- это фреймворк безопасности, предназначенный для обеспечения защиты веб-приложений, разработанных на платформе Spring. Spring Security предоставляет множество видов защиты для веб-приложений, включая:

* аутентификацию: Проверка подлинности пользователей для доступа к защищенным ресурсам, используя различные виды аутентификации, такие как форма входа, аутентификация на основе токена и др;
* авторизацию: Управление правами доступа к защищенным ресурсам для различных категорий пользователей, определяя права доступа на основе ролей, принадлежности к группам и других критериев;
* защиту от атак типа CSRF: Предотвращение атак, связанных с межсайтовой подделкой запросов, путем генерации случайных токенов и проверки их валидности при каждом запросе;
* защиту от атак типа XSS: Предотвращение атак, связанных с внедрением вредоносного кода на стороне клиента, путем применения различных механизмов фильтрации и экранирования данных;
* защиту паролей: Хеширование паролей пользователей и использование солей для защиты от атак типа подбора пароля;
* межсерверную защиту: Предоставление механизмов безопасной коммуникации между серверами, таких как SSL и HTTPS;
* логирование и мониторинг: Предоставление инструментов для записи и анализа событий безопасности, а также мониторинга работы приложения в целом.

FreeMarker [27]- это бесплатный обработчик шаблонов, который позволяет разработчикам создавать динамические HTML страницы и другие виды текстовых документов в Java приложениях. Он интегрируется с фреймворком Spring и позволяет разделять логику и представление, улучшая читаемость и поддерживаемость кода. Использование FreeMarker в фреймворке Spring обеспечивает высокую производительность и масштабируемость, так как он может кэшировать результаты обработки шаблонов и быстро отображать динамические страницы для пользователей. Это делает FreeMarker одним из лучших выборов для создания пользовательского интерфейса в веб-приложениях на базе фреймворка Spring Boot.

Также, в работе был использован git [28] для удобства отслеживания версионности приложения, а библиотека Lombok [29] для упрощения написания кода и ускорения разработки.

В итоге, благодаря объединению этих технологий, конечное приложение будет удовлетворять потребностям пользователя, соответствовать нормам безопасности и надежности, а также позволит повысить читаемость и поддерживаемость кода, что послужит дальнейшему развитию проекта.

* + 1. Клиентская часть

Для реализации клиентской части веб-приложения было решено использовать технологии HTML, CSS и JavaScript.

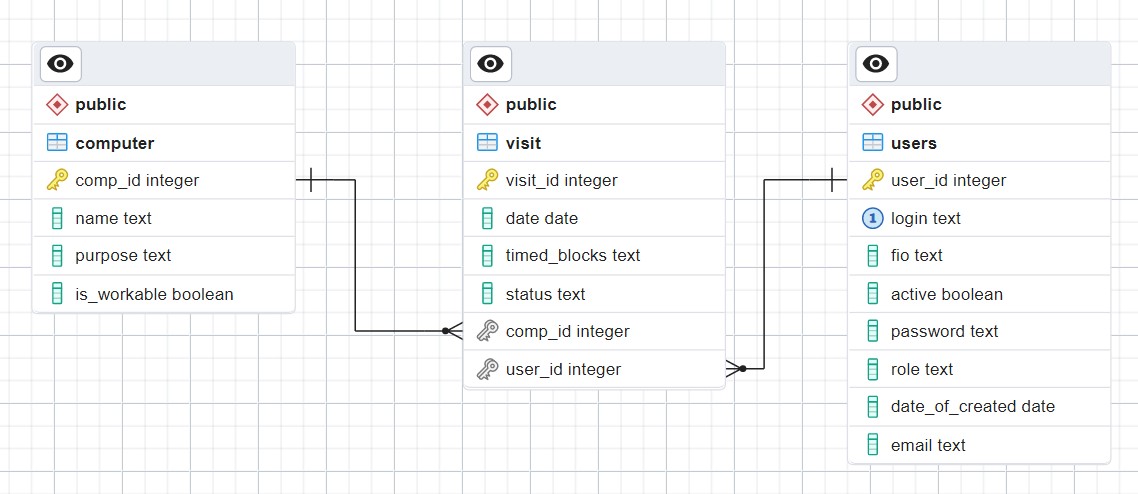
Для ускорения процесса разработки клиентской части веб-приложения был использован Bootstrap. Bootstrap [30] — это популярный фреймворк, который позволяет быстро создавать адаптивные и красивые пользовательские интерфейсы. Фреймворк содержит готовые CSS-стили, JavaScript-компоненты и шаблоны, которые можно использовать для быстрого создания интерфейса приложения. Преимуществами Bootstrap являются удобство в использовании, высокая скорость разработки и возможность создания адаптивного интерфейса для различных устройств.

* 1. Создание базы данных

Созданная база данных в PostgreSQL представлена на Рис.22. Связи между таблицами представлены выглядят следующим образом:

* Таблица «users» связана с таблицей «visit» по полю user\_id.
* Таблица «computer» связана с таблицей «visit» по полю comp\_id.

При попытке удаления информации из таблиц «users» и «computer», если есть записи с этим идентификатором в таблице, «visits», запись не удалится и выдаст ошибку, т.е. удаление будет неуспешным. Это поможет поддерживать целостность данных и избегать неконсистентности информации в базе данных.



Рисунок

– База данных студентов

* 1. Реализация прототипа ИС

Разработанный прототип имеет множество классов, которые изображены на диаграмме классов в приложении Г. Большинство из них представлены классами, которые реализуют функциональность, а именно:

* Controller отвечает за управление потоком приложения. Он принимает запросы пользователей, обрабатывает их, и затем отвечает соответствующим образом клиенту. Для получения или изменения данных он взаимодействует с уровнем Service.
* Service – это бизнес-логика приложения, которая обрабатывается сервисным уровнем. Данные, полученные из слоя DAO, обрабатываются и затем передаются обратно контроллеру. Для того чтобы получить или изменить данные, он также взаимодействует с уровнем DAO.
* DAO – это взаимодействие с базой. Он возвращает данные в класс Service после получения их из базы данных. При необходимости он также сохраняет данные в базе данных.
* DTO – это объект из базы данных. С помощью него производится обмен информации между слоями, а также осуществляется чтение и запись в базу данных.

При разработке во фреймворке Spring boot используется паттерн проектирования Model-View-Controller (MVC) [31], и поставляемые классы соответствуют этому паттерну. Каждый слой взаимодействует только с выше- или нижележащим слоем, что делает приложение более модульным и более простым в обслуживании.

При рассмотрении логики информационной системы, отдельное внимание уделим организации защиты доступа к веб-сайту. Благодаря настройке класса SecurityConfig, каждого пользователя, заходящего неавторизованным на любой каталог сайта, автоматически перенаправляет на страницу авторизации, где его просят ввести логин и пароль. После того, как пользователь аутентифицировался, сервер генерирует csrf-токен [32] , и теперь при отправке post-запросов от клиента к серверу вместе с данными передается csrf-токен, который используется для проверки того, что аутентифицированный пользователь действительно является лицом, выполняющим запросы к приложению. Поскольку токен хранится в сессии пользователя и изменяется каждый раз при повторном создании сессии, вредоносное приложение не может получить к нему доступ.

После прохождения формы авторизации, пользователю становятся доступны только те веб-страницы, которые относятся к его роли. Например, пользователю с ролью «ROLE\_STUDENT» доступны следующие веб-документы, не считая формы login.html:

* computers.html (выбор компьютера);
* shedule.html (выбор даты и времени);
* visits.html (просмотр своих записей);
* profile.html (изменение почты/пароля).

В свою очередь, в базе данных хранятся логин и пароль, захэшированный функцией Bcrypt [33]. Bcrypt - это популярная криптографическая функция хэширования, которая часто используется для безопасного хранения паролей в базах данных. Это распространенный вариант хэширования паролей во многих приложениях, поскольку он не требует больших вычислительных затрат и устойчив к атакам методом перебора.

Теперь рассмотрим, процесс записи студента в компьютерный класс с точки зрения работы сервера. При входе на сайт после авторизации его встречает таблица с выбором компьютеров – эта таблица сформирована путем чтения всех компьютеров, у которых поле «is\_workable» принимает значение «true», так как студентам не нужно видеть неработающие компьютеры. При выборе компьютера студент нажимает кнопку около нужного устройства и переходит на следующий документ, в котором изображен календарь и время для записи. В это же время на сервере считываются все записи, начиная с текущей даты, и с помощью метода getShedulesByCompIdOnDate создается HashTable, в котором хранятся значения даты и занятые дни. Если этот день полностью занят (нет свободного времени), то тогда напротив даты проставляется значение «out». После этого, HashTable передается в шаблон FreeMarker и обрабатывается двумя скриптами showCalendar и showTimeTable для отображения информации, полученной из базы данных, пользователю. Следующим шагом пользователю следует выбрать подходящую дату и время, но есть ограничения, которые не позволят выбрать неподходящую дату, а именно будут недоступны следующие варианты:

1. даты в календаре, которые находятся раньше сегодняшней;
2. суббота и воскресенье;
3. даты, в которых все время уже выбрано другими студентами;
4. время, которое превосходит 4 часа за одну запись;
5. время, в интервале которого, уже есть другая запись студента.

При выборе времени студенту также дублируется время записи на странице. После выбора даты и времени из доступных пользователь нажимает на кнопку «Записаться», и информация о выбранных компьютере, времени и дате отправляются post-запросом на сервер, где и сохраняются в базе данных.

Для работника компьютерного класса, если он хочет записать студента в компьютерный класс, на странице с выбором даты и времени добавляется сверху форма с вводом логина пользователя, а в остальном, порядок действий тот же.

На странице с собственными записями студента есть таблица записей, где он может отменить запись, отправляя post-запрос c идентификатором записи серверу.

Обратим внимание на действия работника компьютерного класса и его взаимодействие с системой с точки зрения работы сервера. После входа в систему перед пользователем открываются две таблицы с отмеченными и неотмеченными посещениями студентов. В первой таблице перед каждой записью располагаются кнопки «Пришел» и «Пропустил», во второй, соответственно, кнопка «Вернуть». Путем нажатия пользователь генерирует запрос серверу в следующем виде: «/check/{status}/{visit\_id}», где {status} – это статус, на который нужно поменять запись, а {visit\_id} – идентификатор записи. Далее, этот запрос отправляется в функцию checkUser, где изменяется статус записи и сохраняется в базе данных.

Следующим шагом рассмотрим формирование отчета для работника. Путем выбора логина студента, имени компьютера, даты «с» и даты «по», пользователь может сформировать отчет обо всех записях по заданным критериям. Если пользователь предпочел не вводить данных, тогда это поле не учитывается при выборе. Таким образом, SQL-запрос к базе данных будет сформирован так (Лист. 1):

SELECT a.\* FROM shedule a

LEFT JOIN User u ON a.user\_id = u.id

LEFT JOIN Computer c ON a.comp\_id = c.id

WHERE 1=1

AND (u.login LIKE :login OR :login IS NULL OR :login = '')

AND (c.compId = :compId OR :compId IS NULL OR :compId = 0)

AND (a.date >= :dateFrom OR DATE(:dateFrom) IS NULL)

AND (a.date <= :dateTo OR DATE(:dateTo) IS NULL)

Листинг

1 – Запрос к базе данных

В дополнение к этому, пользователь может скачать отчет в формате электронных таблиц (.xlsx) путем отправки post-запроса. На сервере создается объект Workbook из библиотеки «org.apache.poi» в функции createXslxFile, далее из объекта Shedule преобразуется к странице электронной таблицы и сохраняется в Workbook. Вторым шагом, этот объект сериализуется в битовый формат и отправляется запросом пользователю с «Content-Type: application/octet-stream» и «Content-Disposition: attachment», что позволяет браузеру автоматически начать скачивание файла пользователю. Сформированная таблица в точности выглядит, как таблица из веб-документа.

Работник компьютерного класса может управлять пользователями. На странице users.html может редактировать, удалять и добавлять новых пользователей. При регистрации пользователя, пароль отправляется на почту, указанную при регистрации, и сохраняется в базе данных в зашифрованном виде с помощью BCrypt, чем обеспечивает безопасное хранение паролей.

* 1. Рекомендации по совершенствованию ИС

Эффективное управление записями учащихся имеет решающее значение для образовательных учреждений, чтобы обеспечить бесперебойное функционирование административных процессов и предоставить точную и достоверную информацию. В условиях компьютерного класса, где записи учащихся хранятся и доступ к ним осуществляется в цифровом виде, становится необходимым постоянное совершенствование информационной системы для удовлетворения меняющихся потребностей образовательных учреждений.

Первая рекомендация предлагает включить в информационную систему функцию поиска нерабочих дней. Интегрировав надежный источник, например, «isdayoff.ru» [34], система сможет предоставлять точную информацию о нерабочих днях, праздниках и выходных. Эта функция предотвратит ввод ошибочных данных студентами в дни, когда компьютерный класс закрыт, тем самым повысив, убрав недостоверность студенческих записей в нерабочие дни.

Вторая рекомендация посвящена использованию запросов по месяцам с помощью jQuery Ajax [35]. Это усовершенствование позволит системе эффективнее получать записи студентов за определенные периоды времени. При выборе клиентское приложение будет получать ограниченную информацию об одном месяце, а не обо всех записях.

Третья рекомендация подчеркивает необходимость оповещения студентов о записи на занятие. За несколько дней или часов, электронное письмо с основной информацией о дате, времени и компьютере, будет отправлено студенту на email, который указан у пользователя. Эта доработка поможет студенту напомнить о самостоятельной работе в компьютерном классе, а также позволит ответственным студентам отменять запись, если забыли о ней.

* 1. Вывод

В главе 3 был спроектирован интерфейс, включая подробное описание всех используемых элементов управления. Каждый инструмент был выбран на основе конкретных причин и преимуществ, которые он предлагает. Кроме того, в главе подчеркивалась важность безопасности информационной системы и приведены способы защиты информационной системы от злоумышленников. Наконец, разработан прототип ИС и определены рекомендации по ее совершенствованию.

Заключение

В результате проделанной работы была спроектирована, разработана и задокументирована информационная система учета времени самостоятельной работы студентов в компьютерном классе. Данная информационная система может быть использована как для студентов ВУЗа, так и для учеников школ, для записи на самостоятельную работу в компьютерном классе.

В ходе выпускной квалификационной работы были реализованы следующие задачи:

1. изучена существующая литература по теме исследования;
2. изучены аналоги системы учета времени;
3. проведен сравнительный анализ этих систем;
4. выявлены и определены требования к разрабатываемой ИС;
5. проведен анализ существующих инструментов для проектирования ИС;
6. определен набор инструментальных средств в соответствии с требованиями;
7. спроектирована модель ИС;
8. разработан прототип интерфейса ИС,
9. разработан прототип ИС;
10. сформированы рекомендации по совершенствованию ИС.

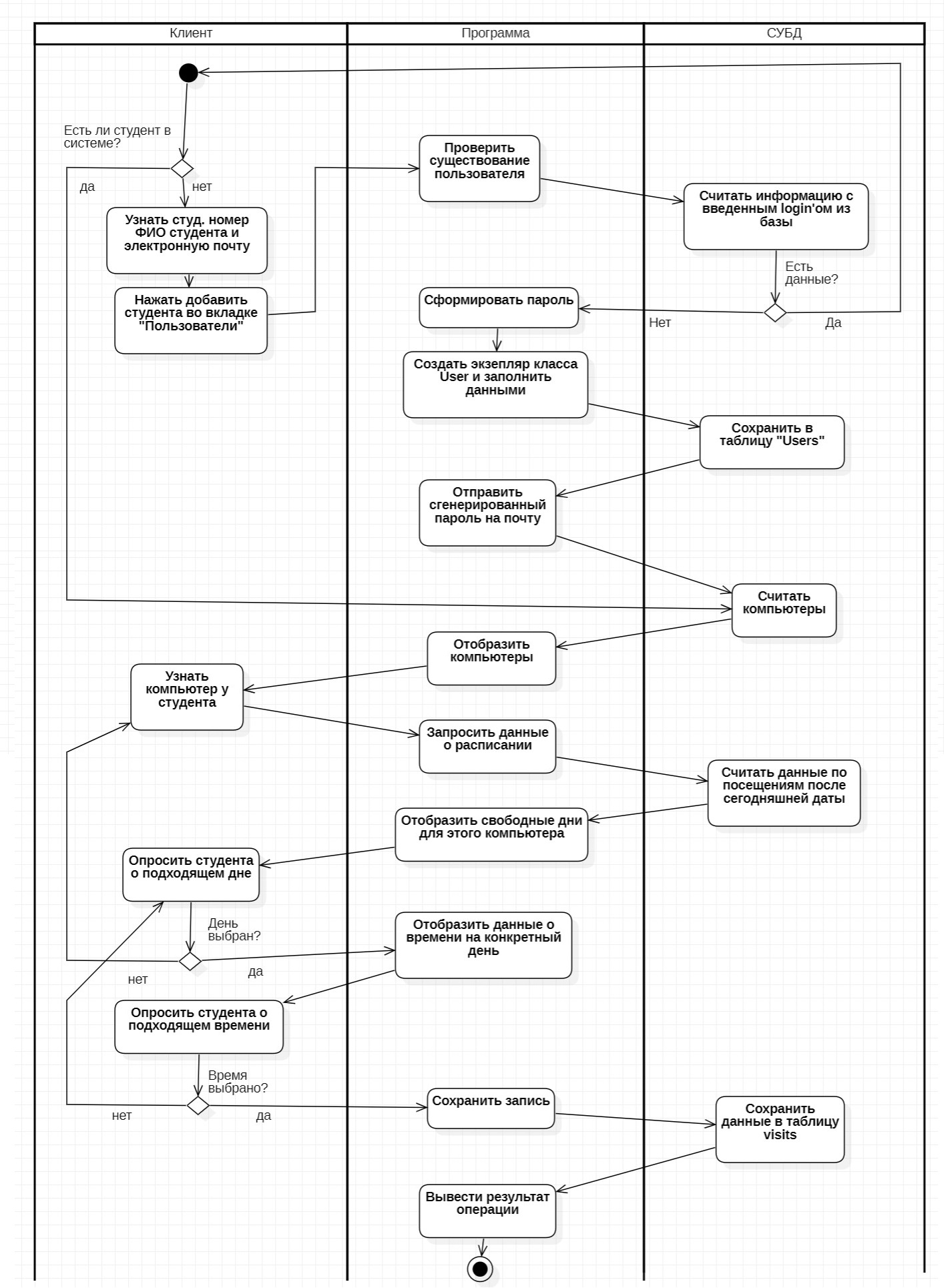
Подводя итоги, можно сказать, что все поставленные задачи были выполнены. Цель выпускной квалификационной работы, а именно, проектирование, разработка и документирование ИС учета времени студентов для самостоятельной работы в компьютерном классе, была достигнута. В будущем планируется продолжить развитие данного проекта с целью его внедрения в реальную образовательную среду.

Библиографический список

1. ГОСТ 34.321-96 Информационные технологии. Система стандартов по базам данных. Эталонная модель управления данными. URL: <https://docs.cntd.ru/document/1200017662> (дата обращения: 22.05.2022)
2. Учебный план. Направление 09.03.02. Информационные системы и технологии направленность (профиль) Безопасность информационных систем степень бакалавр [Электронный ресурс]. – 2020. – URL: <http://www.psu.ru/files/docs/obrazovanie/bachelors/2020/up/09_03_02_bis_up.pdf> (дата обращения 03.03.2022)
3. Кудрявцева Т.А., Пуртова Т.И., Соколова М.Г. Проблемы организации самостоятельной работы студентов в условиях реализации федеральных государственных образовательных стандартов среднего профессионального образования третьего поколения // Инновационное развитие профессионального образования. – 2015. – N 2. – C. 39-42.
4. Ананьина Н. В. Организация самостоятельной работы студентов в условиях реализации ФГОС // Образование. Карьера. Общество. – 2013. – N 4. – C. 51-55.
5. 1С:Университет. Возможности продукта [Электронный ресурс]. – 2022. – URL: <https://solutions.1c.ru/catalog/university/features> (дата обращения 03.03.2022)
6. 1С:Университет. Приобретение продукта [Электронный ресурс]. – 2022. – URL: <https://solutions.1c.ru/catalog/university/buy> (дата обращения 03.03.2022)
7. Платформа «Цифровой колледж Подмосковья». Руководство пользователя [Электронный ресурс]. – 2022. – URL: <https://e-learning.tspk-mo.ru/seo/help/module_2_2_4.php?st=MA%3D%3D&sct=MA%3D%3D&mw=NTg3&ms=AwAAAAAABAI%3D> (дата обращения 05.03.2022)
8. Yclients. Описание продукта [Электронный ресурс]. – 2022. – URL: <https://www.yclients.com/ru> (дата обращения 05.03.2022)
9. Универсальная система учета [Электронный ресурс]. – 2022. – URL: usu.kz (дата обращения 05.03.2022)
10. Методологии моделирования бизнес-процессов [Электронный ресурс]. – 2022. – URL: <https://hsbi.hse.ru/articles/metodologii-modelirovaniya-biznes-protsessov-osnovnye-metody-i-metodiki/> (дата обращения 05.03.2022)
11. Основные методологии обследования организаций. Стандарт IDEF0. [Электронный ресурс]. – 2022. – URL: <https://www.cfin.ru/vernikov/idef/idef0.shtml> (дата обращения 05.03.2022)
12. Нотация BPMN 2.0: ключевые элементы и описание [Электронный ресурс]. – 2021. – URL: <https://www.comindware.com/ru/blog-нотация-bpmn-2-0-элементы-и-описание/> (дата обращения 05.03.2022)
13. Бабич А. Введение в UML [Электронный ресурс]. – 2022. – URL: <https://intuit.ru/studies/courses/1007/229/info> (дата обращения 10.03.2022)
14. Вендров. А.М. CASE-технологии. Современные методы и средства проектирования информационных систем [Электронный ресурс]. – 2022. – URL: <http://citforum.ru/database/case/introduction.shtml> (дата обращения 10.03.2022)
15. StarUML documentation [Электронный ресурс]. – 2022. – URL: <https://docs.staruml.io/> (дата обращения 10.03.2022)
16. Classic Violet Review [Электронный ресурс]. – URL: <https://horstmann.com/violet/> (дата обращения 10.03.2022)
17. Diagram Designer [Электронный ресурс]. – 2020. – URL: <https://logicnet.dk/DiagramDesigner/> (дата обращения 10.03.2022)
18. Проектирование UML (ArgoUML) [Электронный ресурс]. – 2022. – URL: <https://studwood.net/1707669/informatika/proektirovanie> (дата обращения 10.03.2022)
19. Леоненков А. Нотация и семантика языка UML [Электронный ресурс]. – 2022. – URL: <https://intuit.ru/studies/courses/32/32/info> (дата обращения 10.03.2022)
20. Требования к паролю // MyKaspersky URL: https://support.kaspersky.com/KPC/1.0/ru-RU/183862.htm (дата обращения: 22.05.2023).
21. Documentation // PostgreSQL URL: https://www.postgresql.org/docs/ (дата обращения: 22.05.2023).
22. Чем PostgreSQL лучше других SQL баз данных с открытым исходным кодом. // Хабр URL: https://habr.com/ru/articles/282764/ (дата обращения: 22.05.2023).
23. Java Documentation // Oracle URL: https://docs.oracle.com/en/java/ (дата обращения: 22.05.2023).
24. Spring Boot Reference Documentation // Spring URL: https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/ (дата обращения: 22.05.2023).
25. Spring Data JPA - Reference Documentation // Spring URL: https://docs.spring.io/spring-data/jpa/docs/current/reference/html/ (дата обращения: 22.05.2023).
26. Spring Security // Spring URL: https://docs.spring.io/spring-security/reference/ (дата обращения: 22.05.2023).
27. Apache FreeMarker Manual // FreeMarker URL: https://freemarker.apache.org/docs/ (дата обращения: 22.05.2023).
28. git // git URL: https://git-scm.com/ (дата обращения: 22.05.2023).
29. Project Lombok // Project Lombok URL: git // git URL: https://projectlombok.org/ (дата обращения: 22.05.2023).
30. Bootstrap // Bootstrap URL: https://getbootstrap.com/ (дата обращения: 22.05.2023).
31. Model View Controller (MVC) Design Pattern in Java // Java Guides URL: https://www.javaguides.net/2019/08/model-view-controller-mvc-design-in-java.html (дата обращения: 22.05.2023).
32. A Guide to CSRF Protection in Spring Security // Baeldung URL: https://www.baeldung.com/spring-security-csrf (дата обращения: 27.05.2023).
33. Bcrypt Step by Step // medium URL: https://medium.com/bootdotdev/bcrypt-step-by-step-99dd76a3b16a (дата обращения: 22.05.2023).
34. API производственного календаря // isDayOff() URL: https://www.isdayoff.ru/ (дата обращения: 22.05.2023).
35. jQuery.ajax() // jQuery URL: https://api.jquery.com/jquery.ajax/ (дата обращения: 22.05.2023).

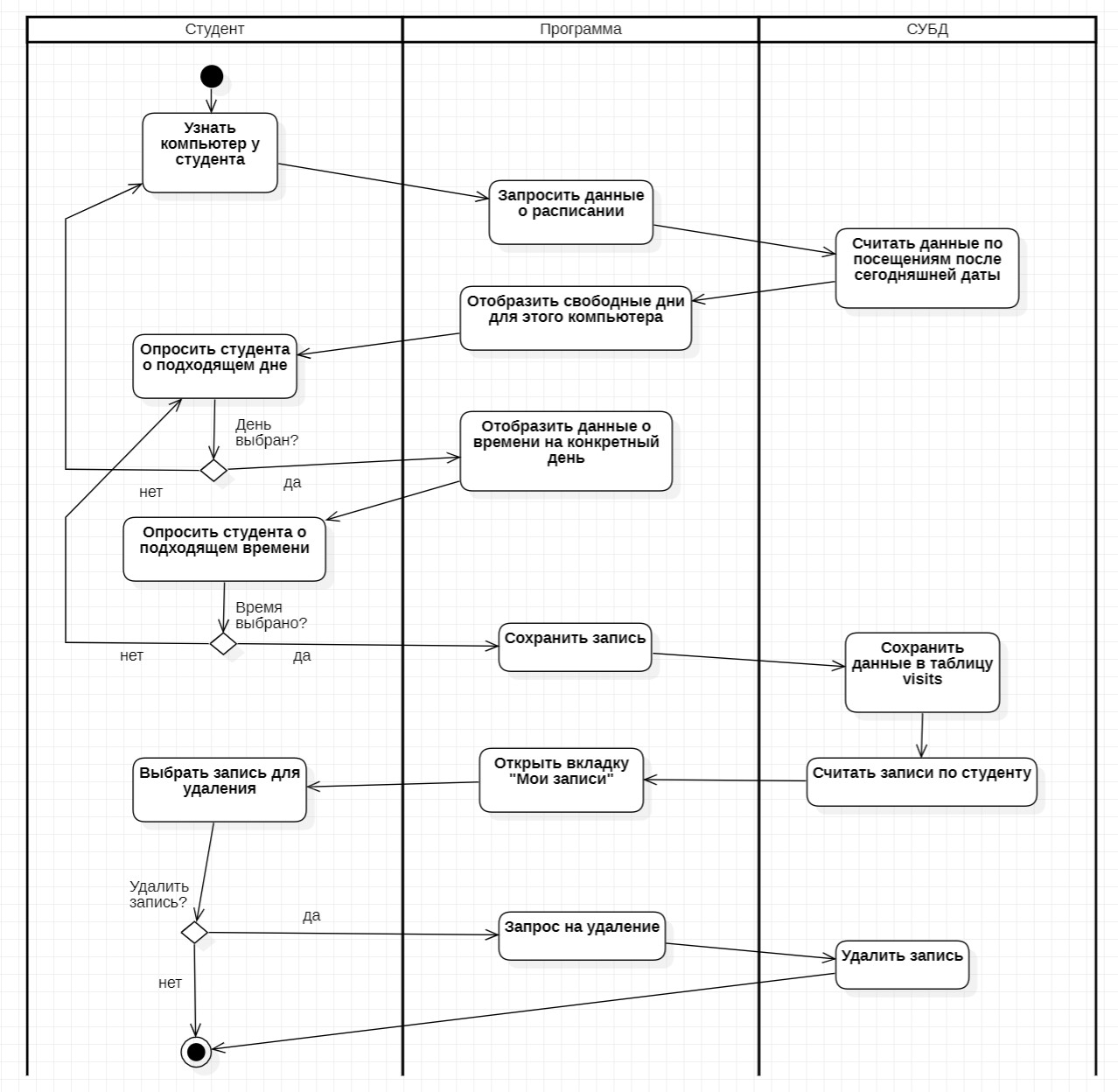
ПРИЛОЖЕНИЕ А

**Диаграмма последовательности «Записать студента». Лаборант**



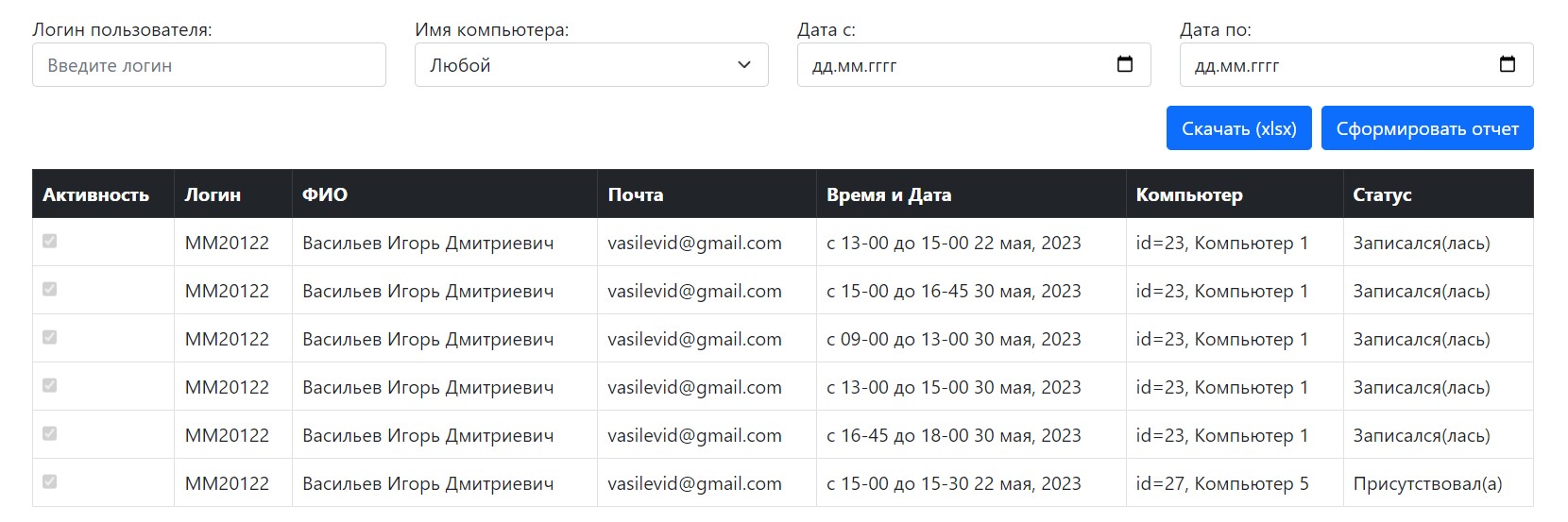
ПРИЛОЖЕНИЕ Б

**Диаграмма последовательности «Записаться». Студент.**

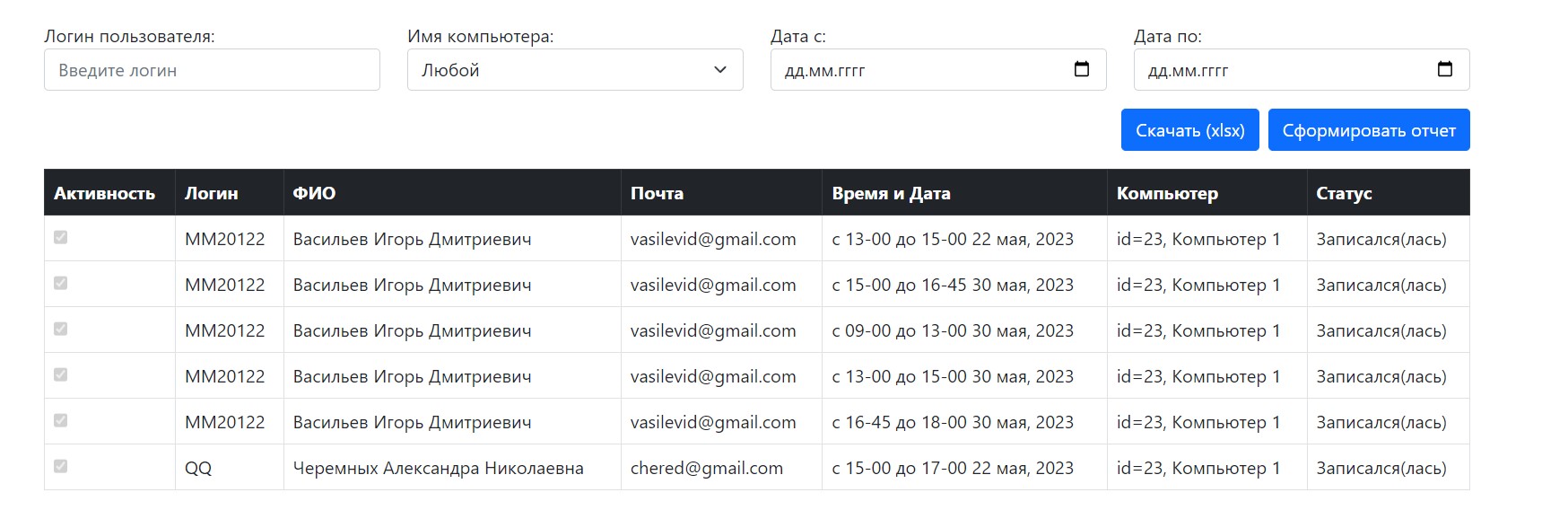


ПРИЛОЖЕНИЕ В

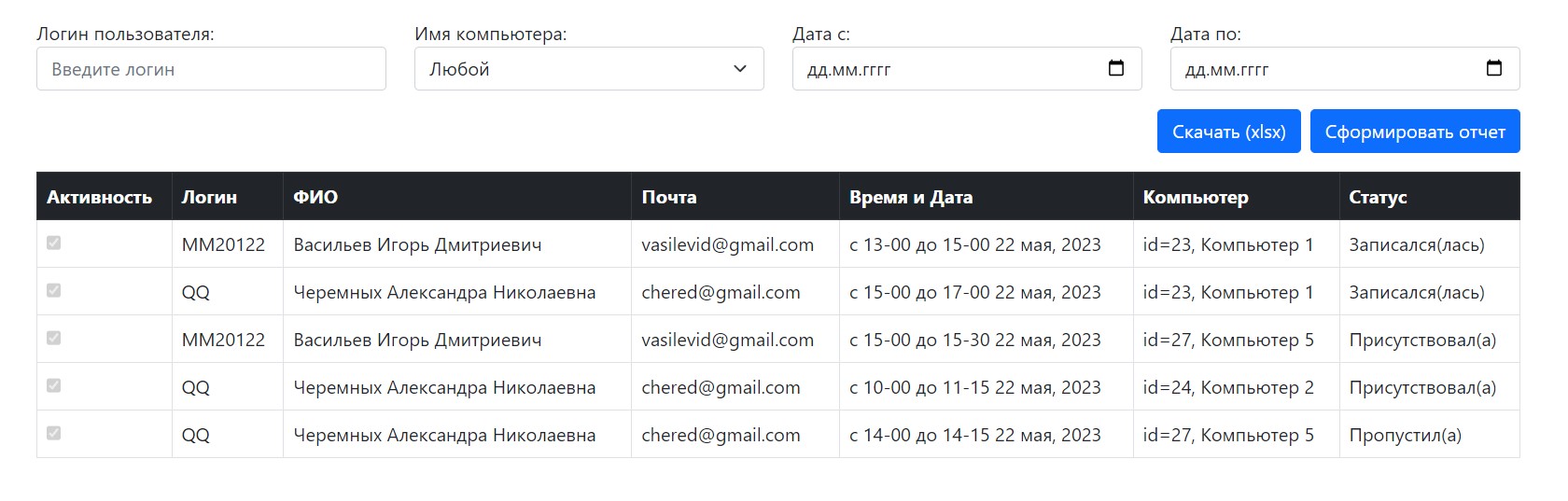
**Отчет. Сортировка по логину.**



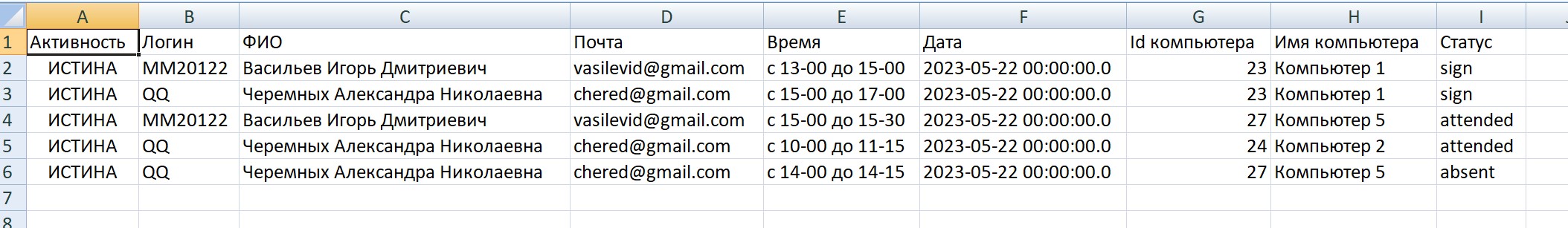
**Отчет. Сортировка по компьютеру.**



**Отчет. Сортировка по дате.**



**Отчет. Формирование xlsx.**



ПРИЛОЖЕНИЕ Г

**Диаграмма классов приложения**

