|  |  |
| --- | --- |
| Министерство науки и высшего образования РФ  Федеральное государственное автономное образовательное учреждение высшего образования  «Пермский государственный национальный исследовательский университет» | |
| Механико-математический факультет  Кафедра информационных технологий | |
| УДК 004.9 | |
| **Отчет о преддипломной практике**  По направлению 09.03.02 «Информационные системы и технологии» | |
|  | Работу выполнил:  cтудент 4 курса  группы ММ/О ИТХ-2019 НБ  Васильев Игорь Дмитриевич  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Научный руководитель:  Кандидат педагогических наук, доцент кафедры ИТ  Василюк Надежда Николаевна  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Пермь 2023 | |

Оглавление

[ВВЕДЕНИЕ 3](#_Toc136205032)

[Раздел 1. Анализ предметной области 5](#_Toc136205033)

[1.1 Обзор и сравнение существующих аналогов ИС 5](#_Toc136205034)

[1.2 Требования к проектируемой системе 8](#_Toc136205035)

[1.3 Вывод 9](#_Toc136205036)

[Раздел 2. Проектирование модели информационной системы учета СРС студента 10](#_Toc136205037)

[2.1 Обзор и выбор методологии моделирования 10](#_Toc136205038)

[2.2 Выбор CASE-средства проектирования ИС 12](#_Toc136205039)

[2.3 Моделирование информационной системы 14](#_Toc136205040)

[2.4 Вывод 17](#_Toc136205041)

[Раздел 3. Разработка прототипа информационной системы учета СРС студента 18](#_Toc136205042)

[3.1 Прототипирование интерфейса ИС 18](#_Toc136205043)

[3.1.1 Интерфейс студента 18](#_Toc136205044)

[3.1.2 Интерфейс лаборанта 20](#_Toc136205045)

[3.2 Выбор средств разработки ИС 22](#_Toc136205046)

[3.2.1 Система управления базой данных 22](#_Toc136205047)

[3.2.2 Серверная часть 23](#_Toc136205048)

[3.2.3 Клиентская часть 23](#_Toc136205049)

[3.3 Создание базы данных 23](#_Toc136205050)

[3.4 Реализация прототипа ИС 24](#_Toc136205051)

[3.5 Рекомендации по совершенствованию ИС 27](#_Toc136205052)

[3.6 Вывод 28](#_Toc136205053)

[Заключение 29](#_Toc136205054)

[Библиографический список 30](#_Toc136205055)

[ПРИЛОЖЕНИЕ А 33](#_Toc136205056)

[ПРИЛОЖЕНИЕ Б 34](#_Toc136205057)

[ПРИЛОЖЕНИЕ В 35](#_Toc136205058)

[ПРИЛОЖЕНИЕ Г 36](#_Toc136205059)

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 преддипломной практики – 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.
11. Анализ предметной области
    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.

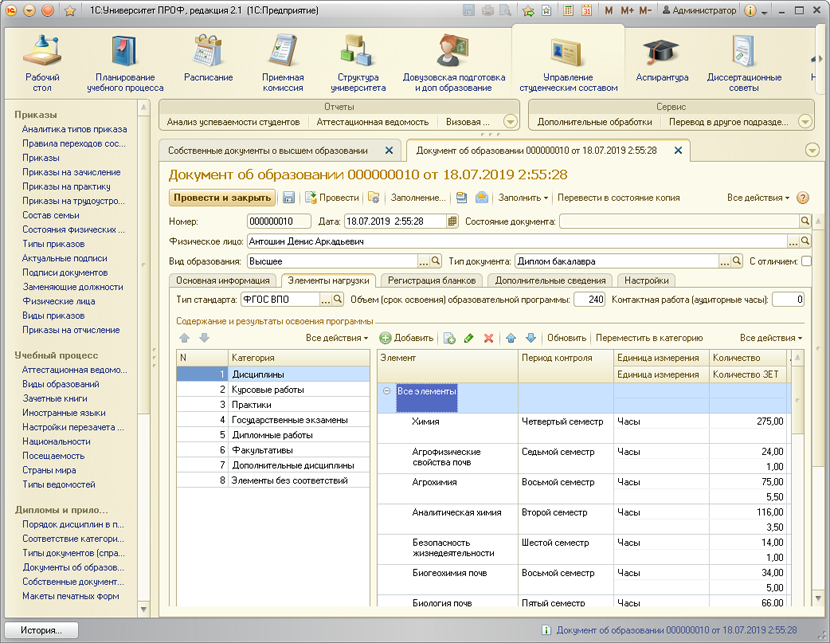
1. «1С: Университет» [5]

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].

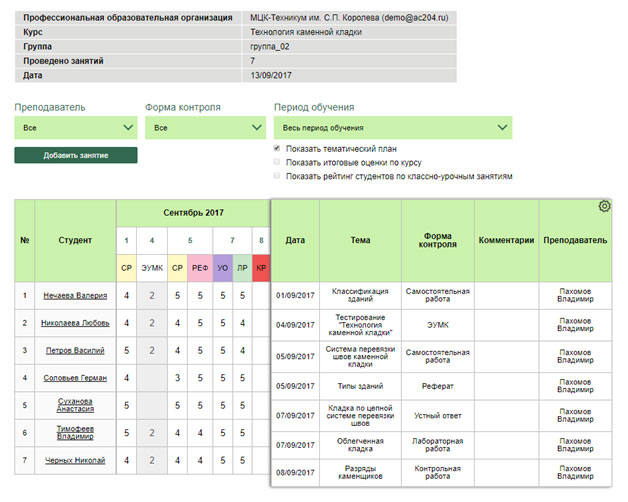


Рисунок

– 1C:Университет

1. «Электронный журнал»

Информационная система «Электронный журнал» предназначена для ведения учета и анализа учебной деятельности студентов (Рис. 2).



Рисунок

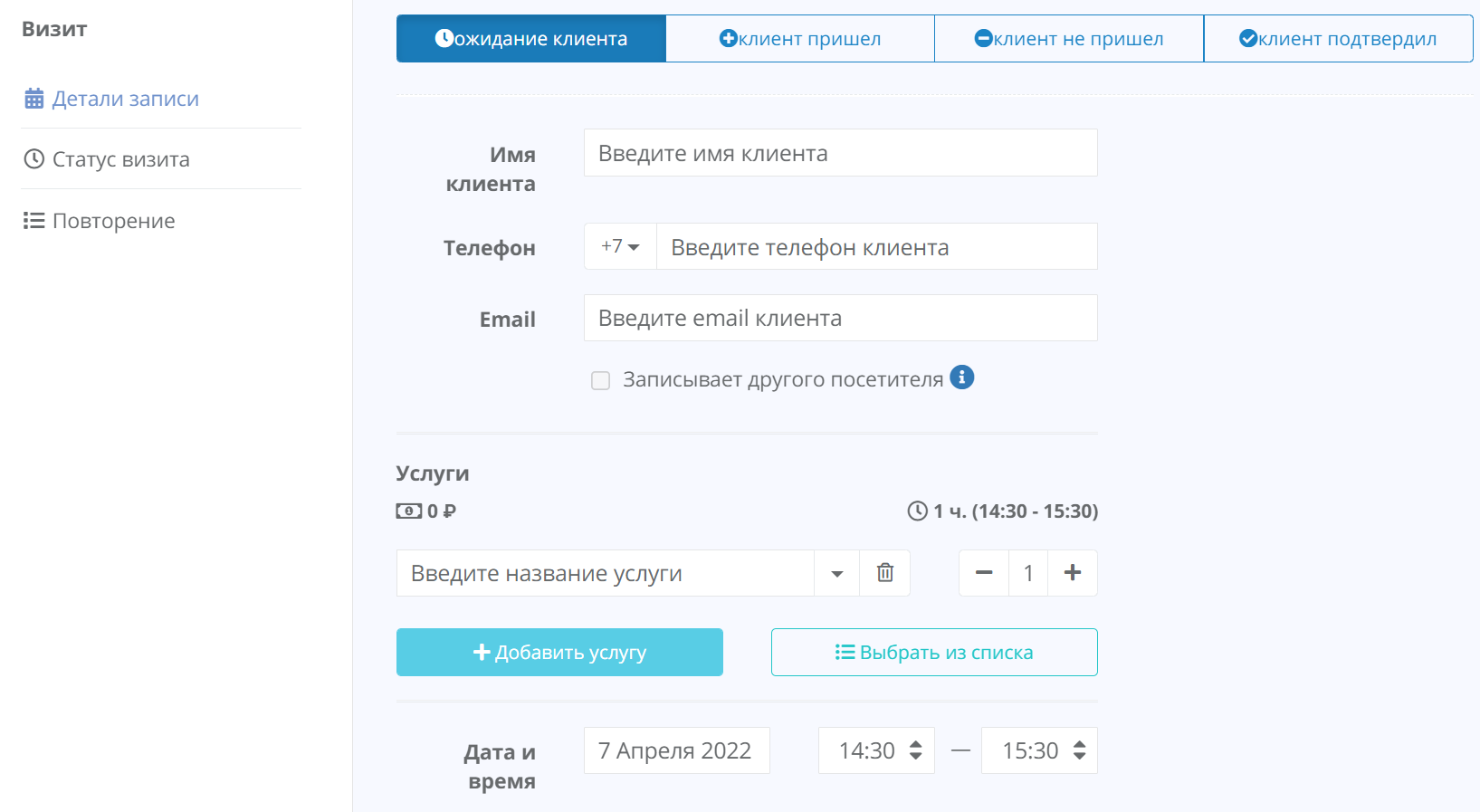
– «Цифровой колледж Подмосковья»

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.

1. «Yclients»

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;
* онлайн-запись клиентов;
* ведение статистики по дням, месяцам, кварталам и годам;
* сохранение информации о клиенте в базу данных.
* 

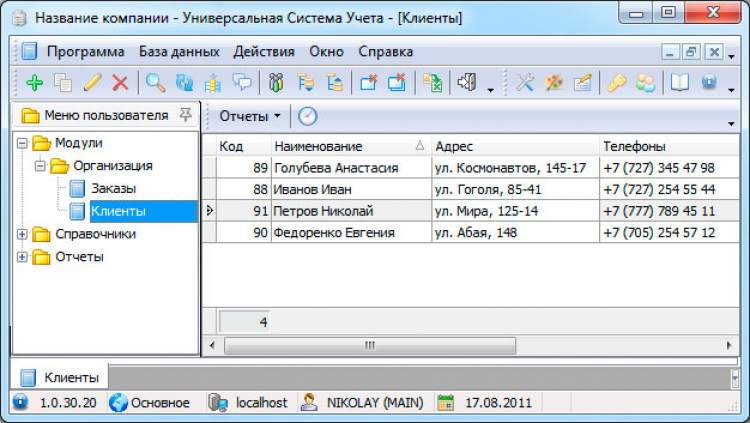
Рисунок

– Запись

1. «Универсальная система учета»

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.



Рисунок

– Универсальная система учета

Таким образом, рассмотренные выше системы являются многофункциональными и имеют как преимущества, так и ряд недостатков.

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
   1. Обзор и выбор методологии моделирования

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

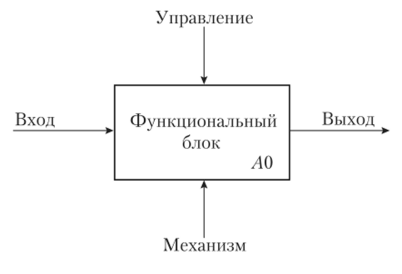
* полнота и выразительность;
* простота и понятность;
* гибкость и расширяемость.

Рассмотрим наиболее популярные методологии проектирования [10]:

IDEF (Integrated Definition for Function Modeling) [11]

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.



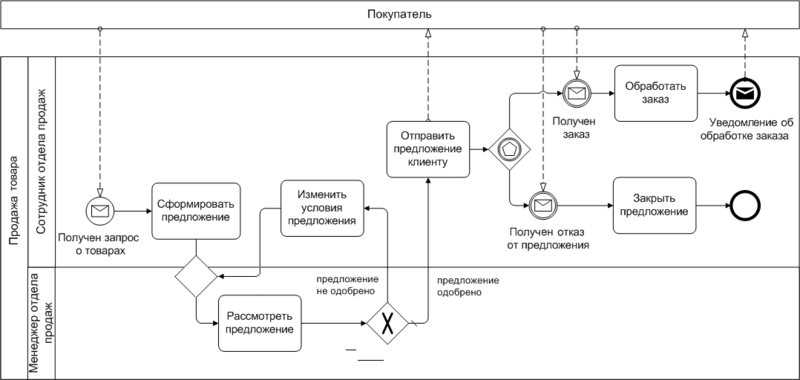
Рисунок

– Пример диаграммы IDEF0

BPMN (Business Process Management Notation)

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.



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.

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.

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.

* + 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.
  + 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.

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.

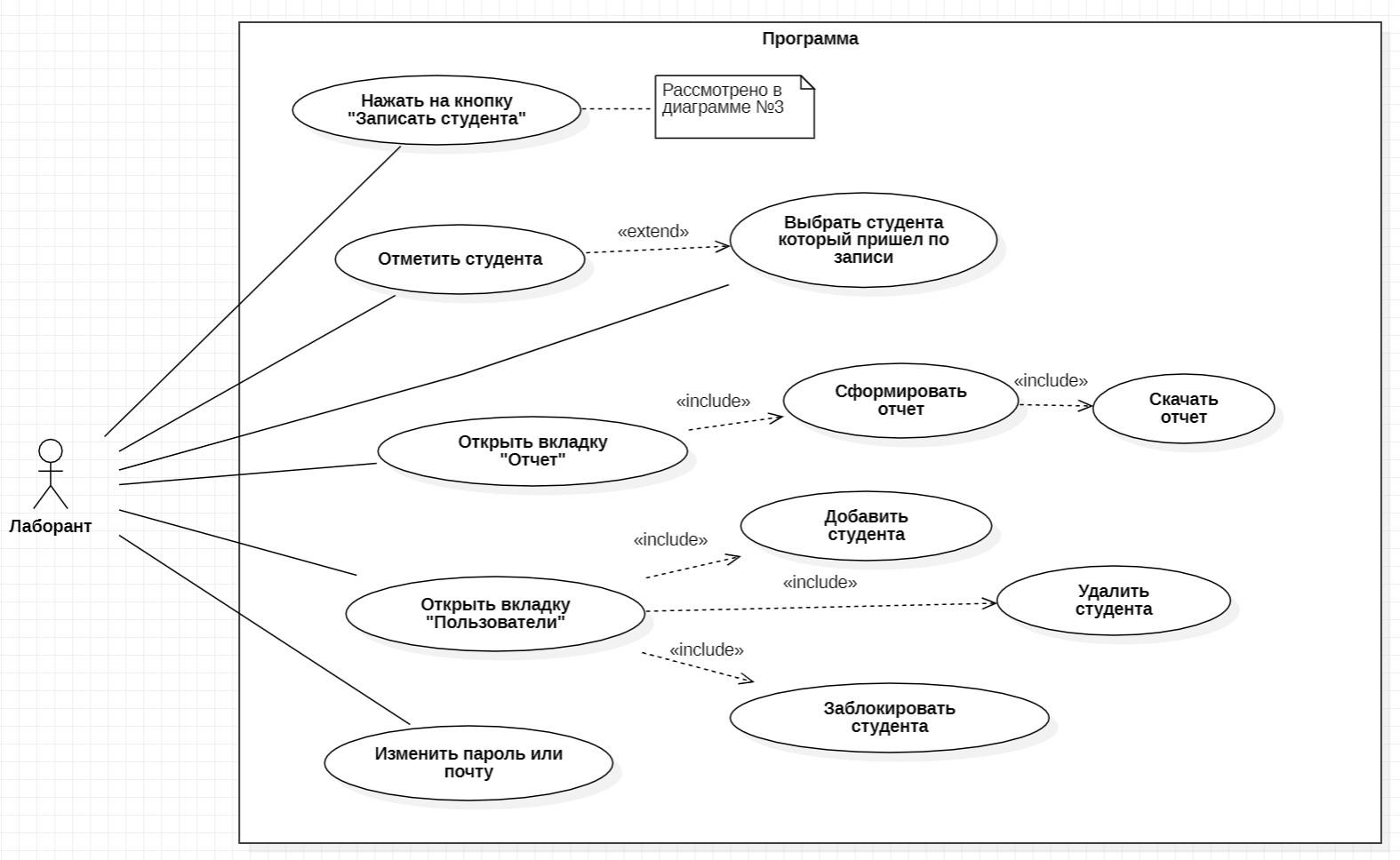
Учитывая эти факторы, CASE-средство StarUML является наиболее подходящим для проектирования информационных систем с учетом указанных критериев.

* 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.

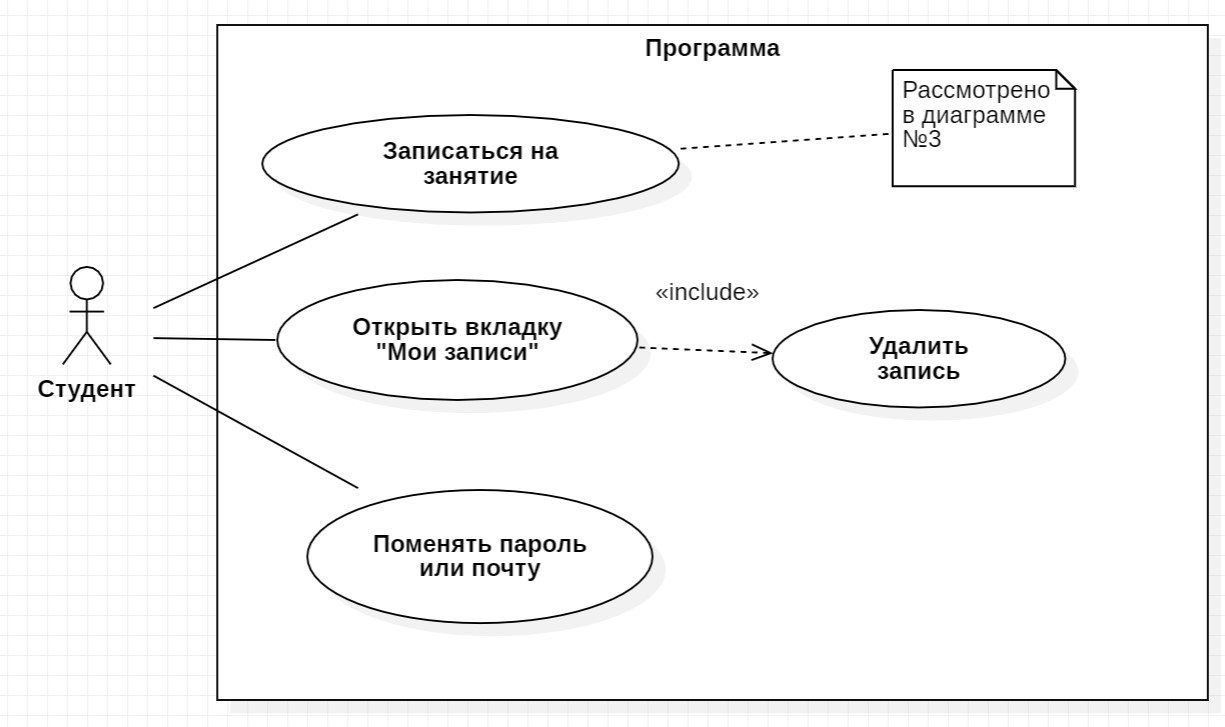
Для достижения целей визуального проектирования с использованием нотации UML вначале строится диаграмма вариантов использования. Диаграмма вариантов использования помогает исследовать и документировать различные случаи использования системы и ее функциональные возможности.

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



Рисунок

– Диаграмма прецедентов «Лаборант» №1

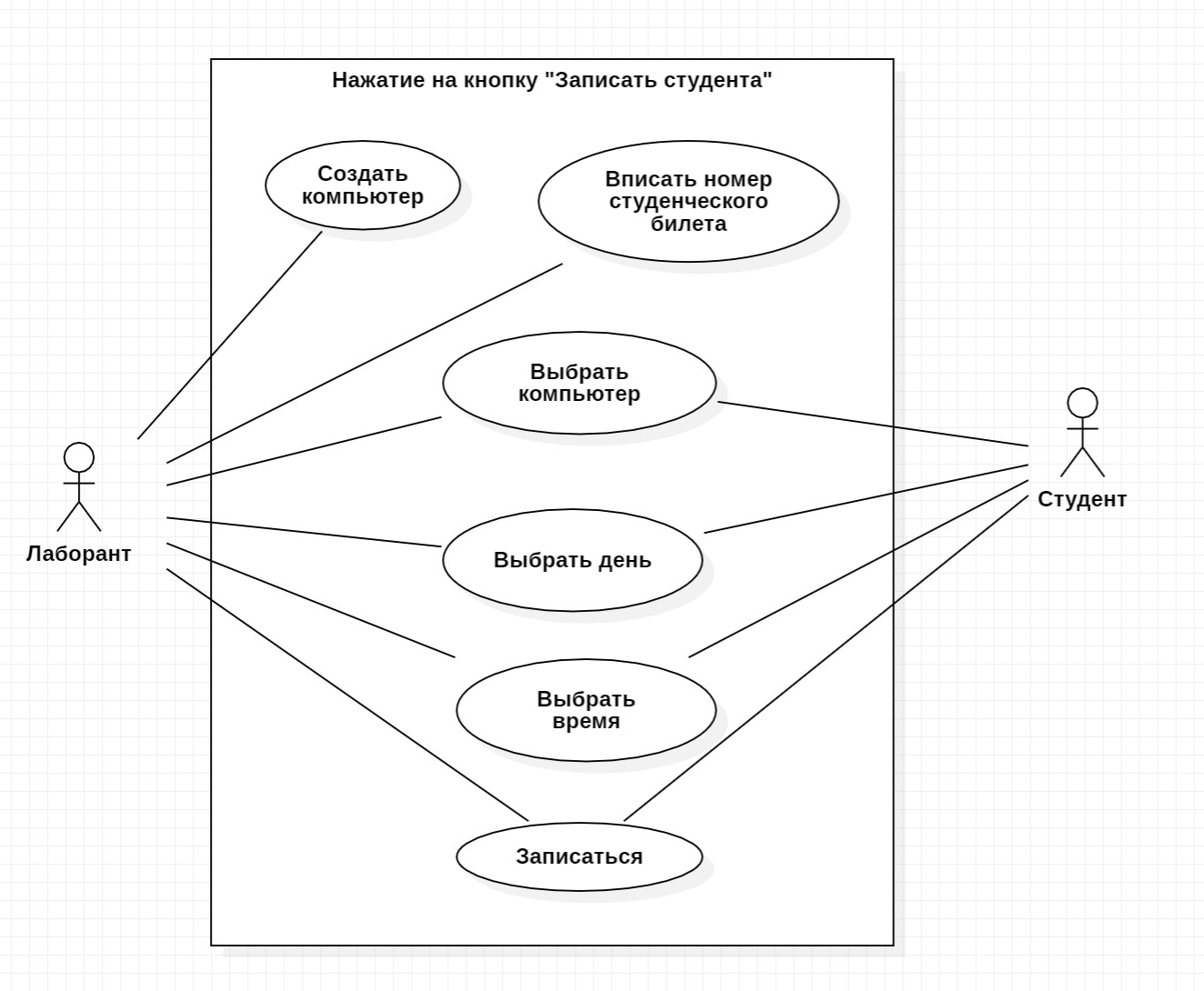


Рисунок

8 – Диаграмма прецедентов «Студент» №2

Рассмотрим диаграмму прецедентов «Студент» №2 (Рис. 8). 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.

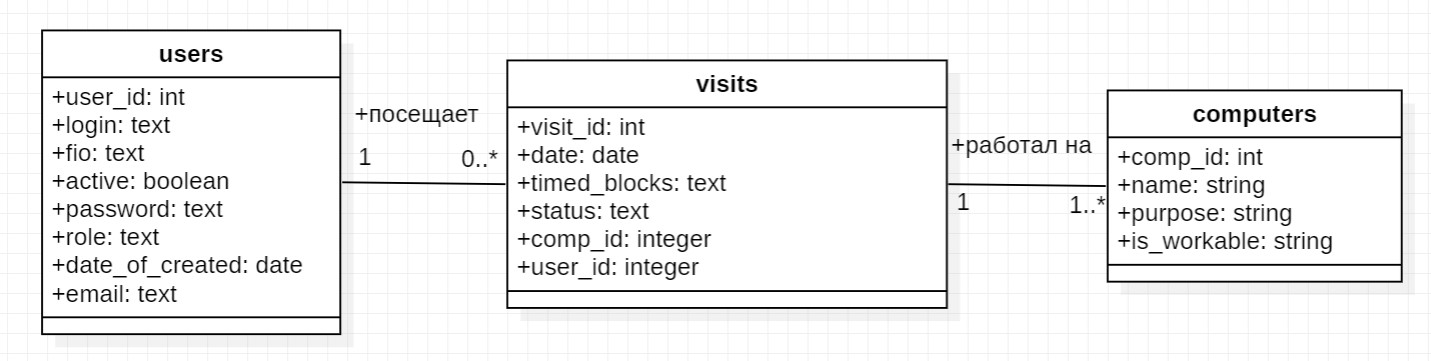


Рисунок

9– Диаграмма прецедентов «Записать студента» №3

Рассмотрим диаграмму прецедентов №3 (Рис. 9). 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 на диаграмме.

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].



Рисунок

– Диаграмма классов

Следовательно, на диаграмме классов (Рис. 10) 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.

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

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.

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

* 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.

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.

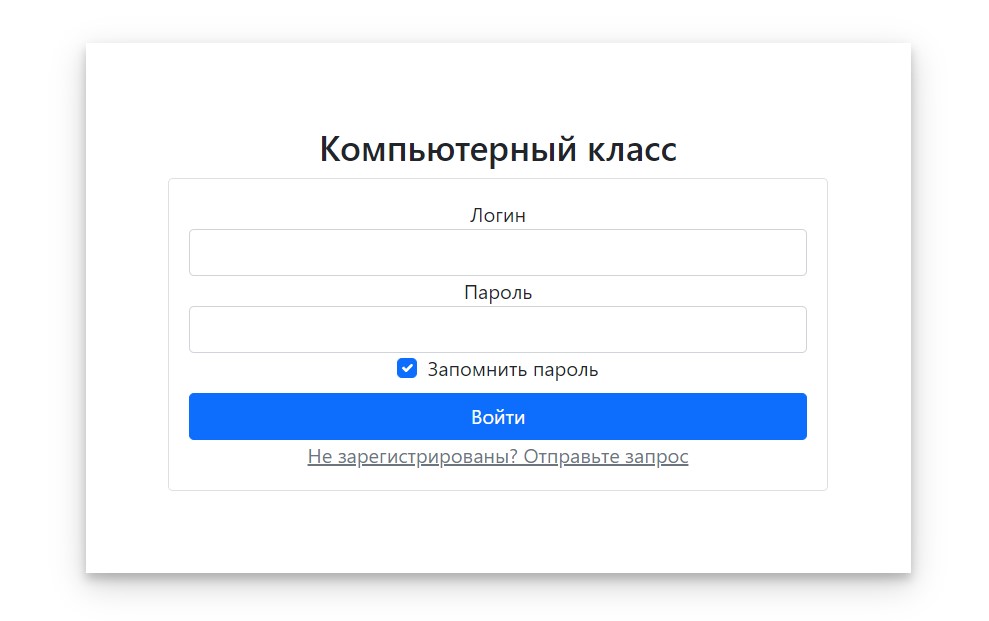
1. Разработка прототипа информационной системы учета СРС студента
   1. Прототипирование интерфейса ИС

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.

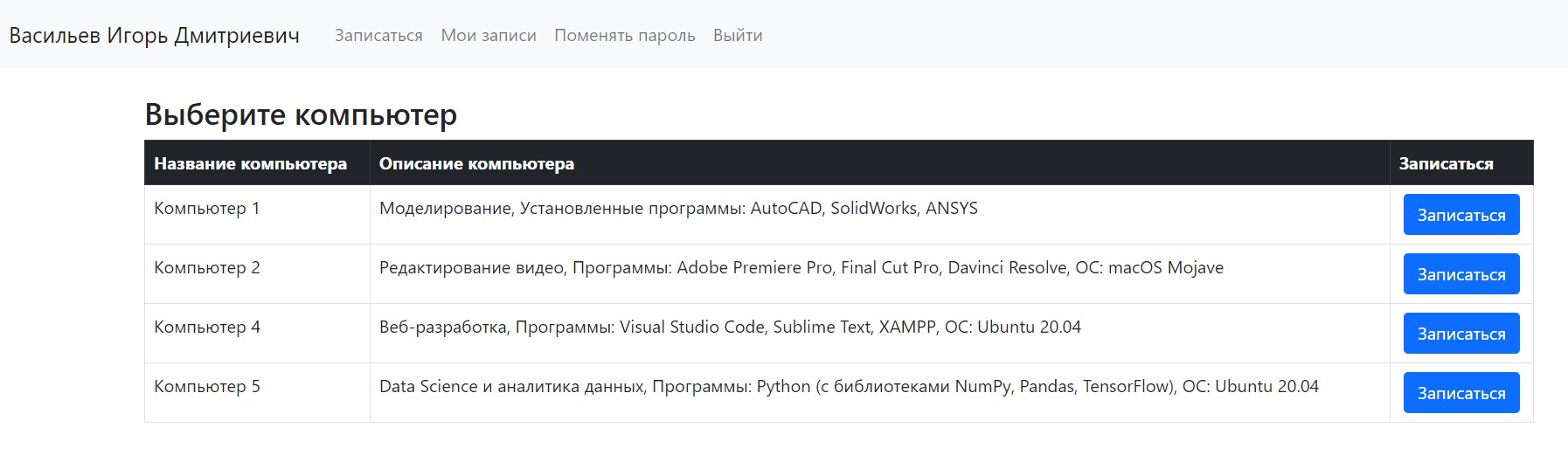
При открытии любого каталога сайта студент увидит первым следующее окно (Рис. 13)



Рисунок

– Авторизация

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

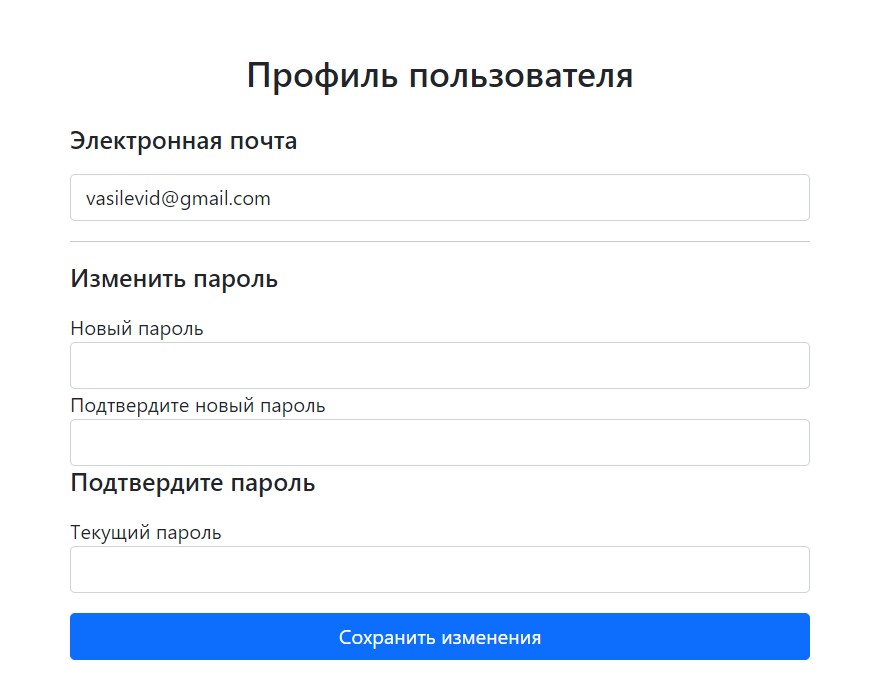


Рисунок

12 – Выбор ПК (Студент)

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

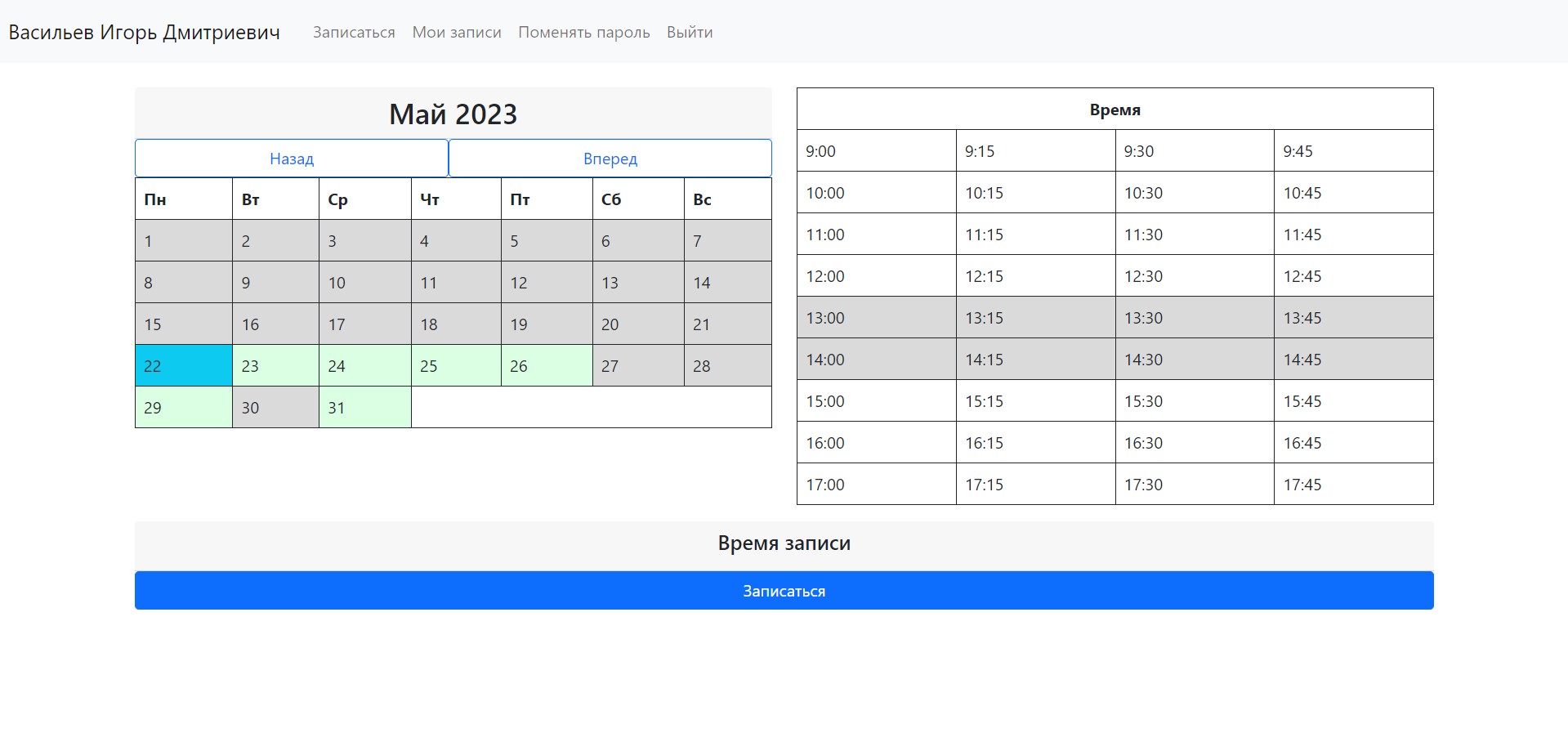
* минимальная длина пароля 8 символов;
* не более 128 символов;
* как минимум одна заглавная и одна строчная буква;
* только латинские или кириллические буквы;
* как минимум одна цифра;
* только арабские цифры;
* без пробелов;



Рисунок

– Изменение почты или пароля

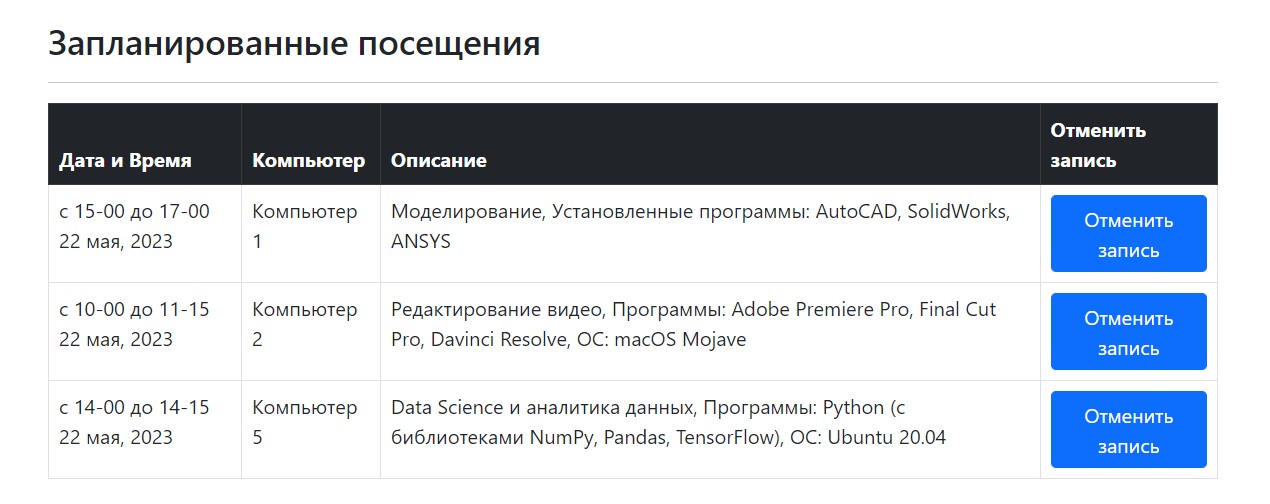
После выбора необходимого компьютера (Рис. 12) перед студентом отображается новая информация о выборе дня и времени посещения (Рис. 16).



Рисунок

– Запись

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

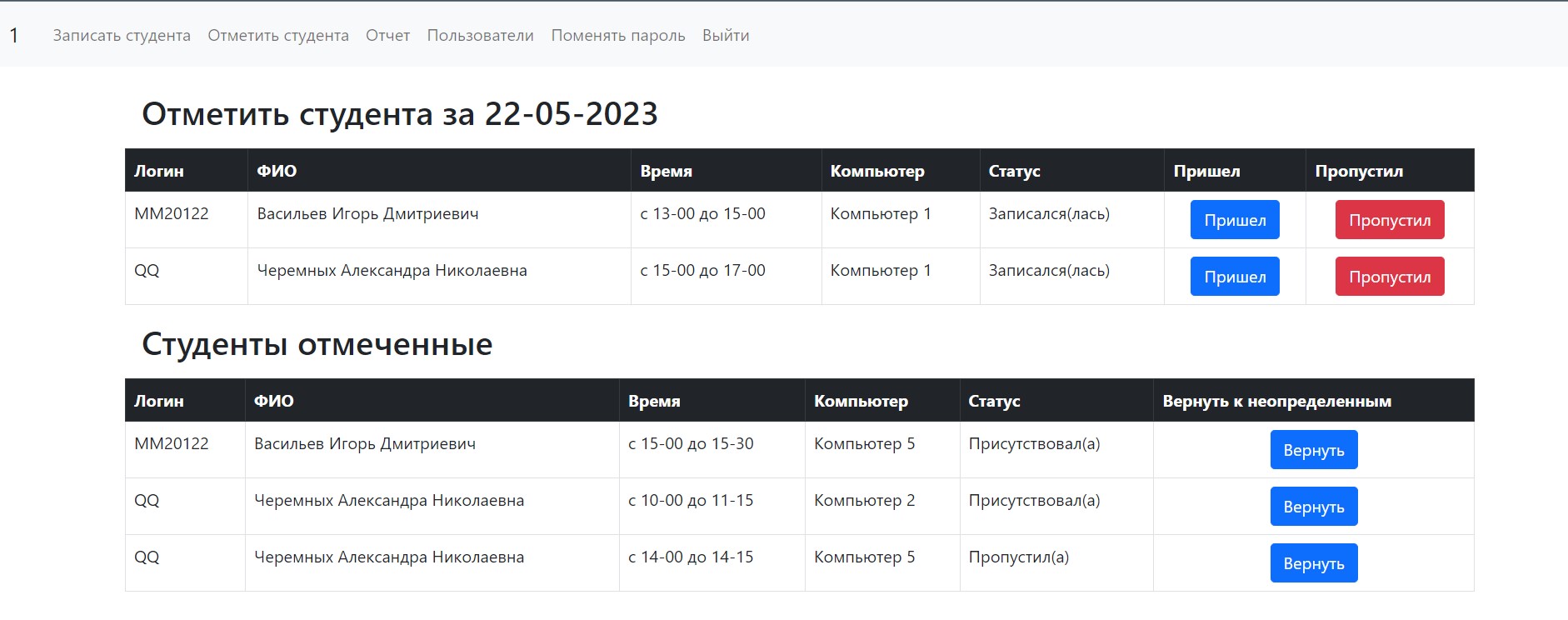


Рисунок

15 – Запланированные посещения студента

* + 1. Интерфейс лаборанта

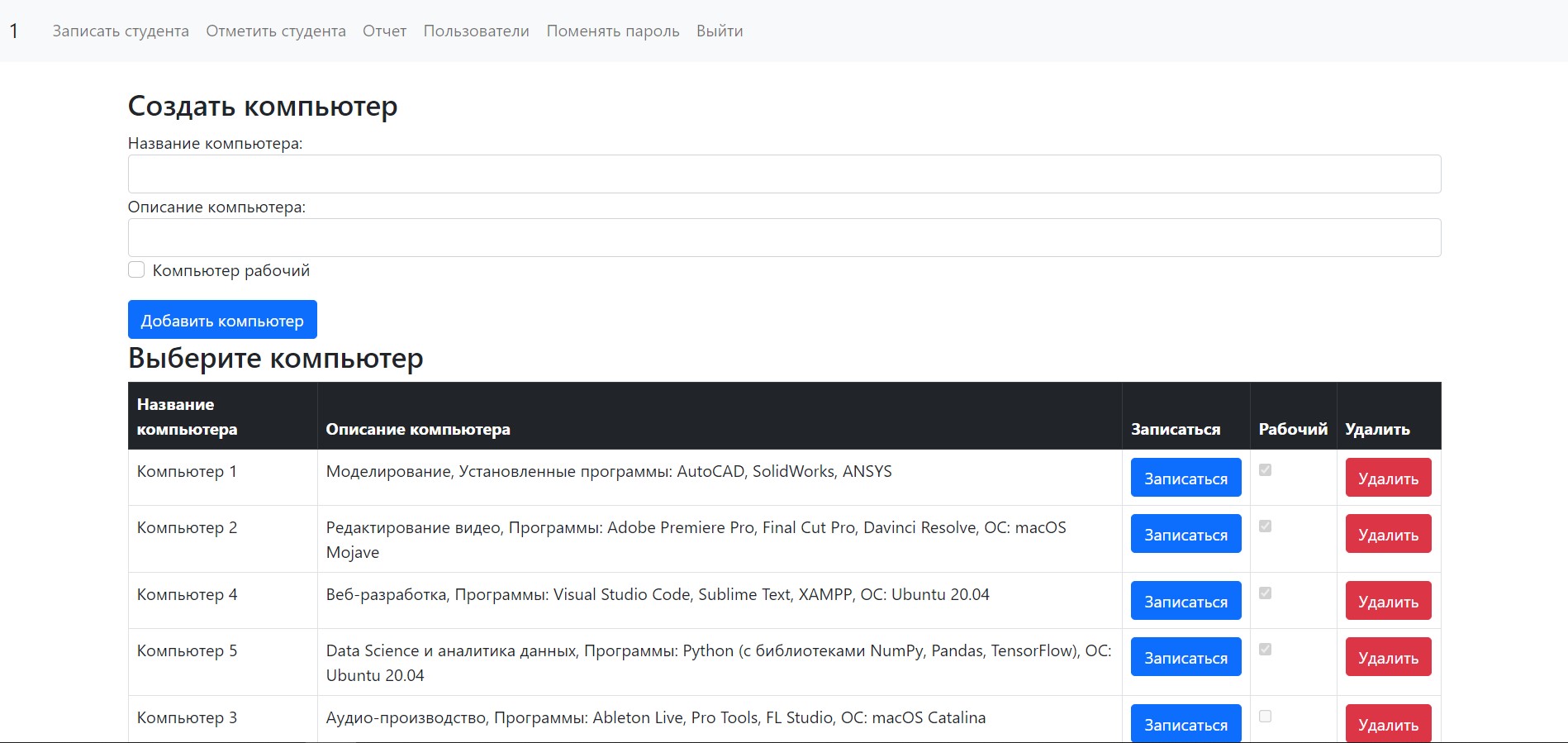
Теперь рассмотрим интерфейс лаборанта. После авторизации лаборанта перенаправит, на страницу «Отметить студента», так как пользователь с ролью «ROLE\_WORKER» чаще всего будет взаимодействовать с системой именно через эту страницу (Рис. 18). На данной странице изображены 2 таблицы, в которых отмечено 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

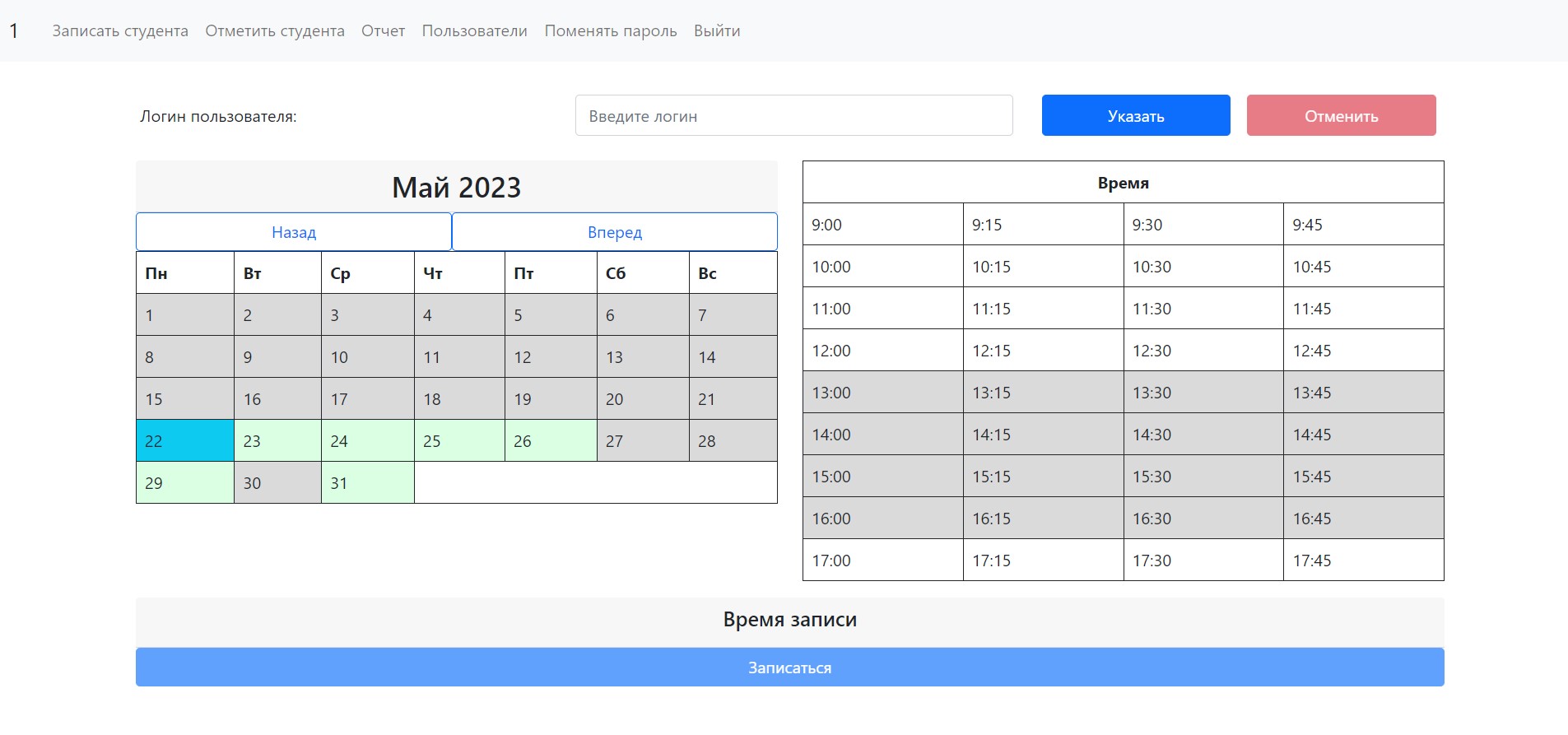
– Отметить студента

Запись студента на конкретный компьютер, день и время, со стороны лаборанта представлена на Рис. 19 и на Рис. 20.



Рисунок

– Выбор ПК (Лаборант)

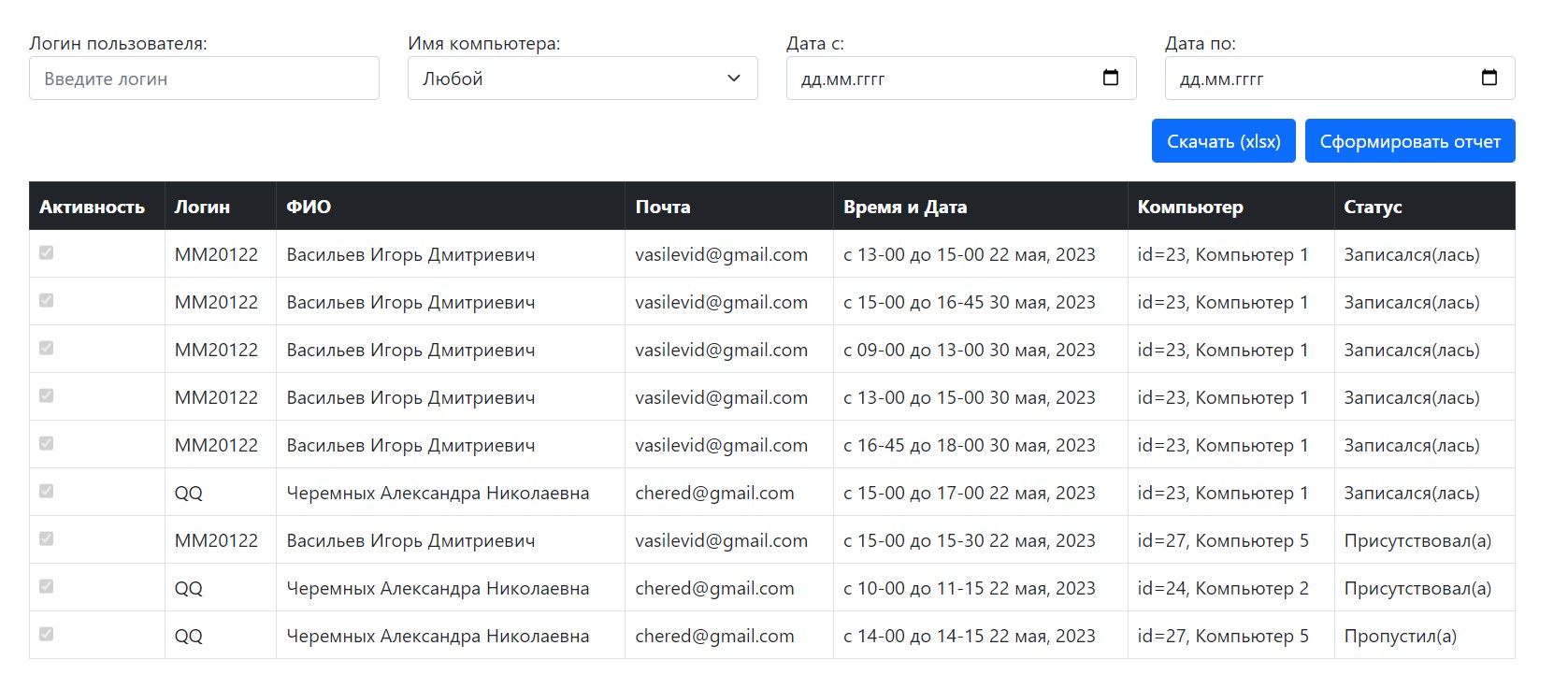


Рисунок

– 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.

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.

* + 1. Клиентская часть

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

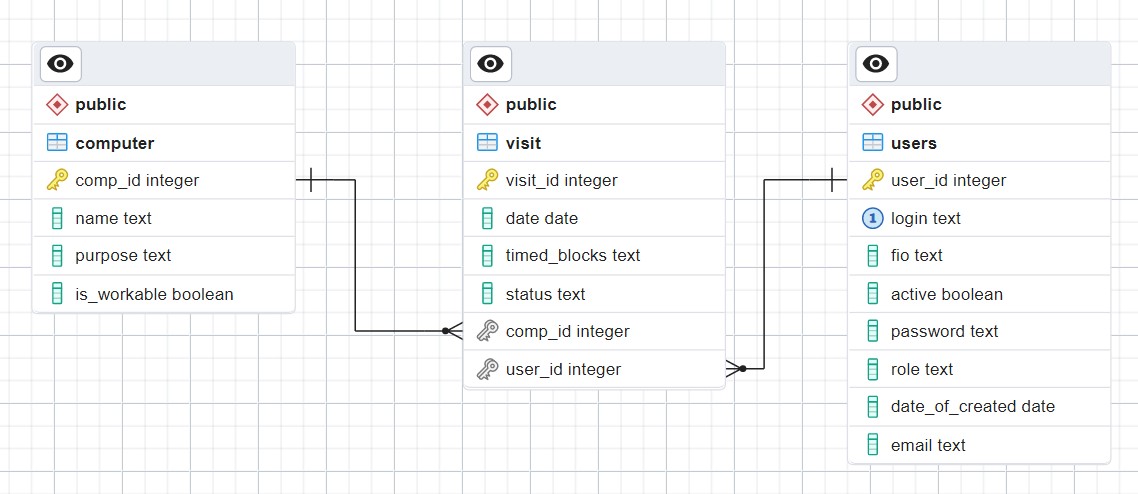
Для ускорения процесса разработки клиентской части веб-приложения был использован Bootstrap. Bootstrap [30] — это популярный фреймворк, который позволяет быстро создавать адаптивные и красивые пользовательские интерфейсы. Фреймворк содержит готовые CSS-стили, JavaScript-компоненты и шаблоны, которые 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;
* service;
* dao;
* dto.

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.

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 преддипломной практики 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.

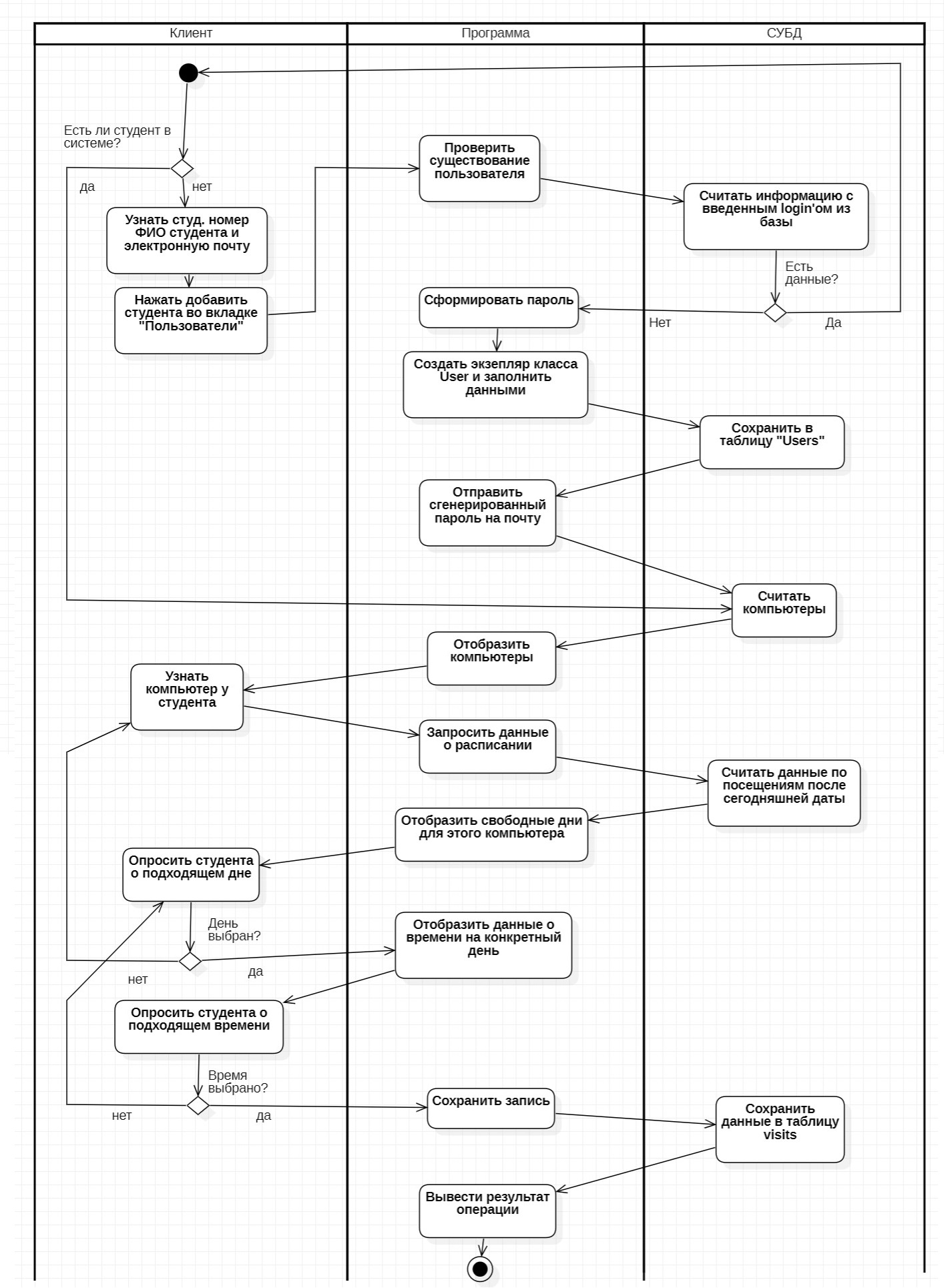
Подводя итоги, можно сказать, что все поставленные задачи были выполнены. Цель преддипломной практики, а именно, проектирование, разработка и документирование ИС учета времени студентов для самостоятельной работы в компьютерном классе, была достигнута. В будущем планируется продолжить развитие данного 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).

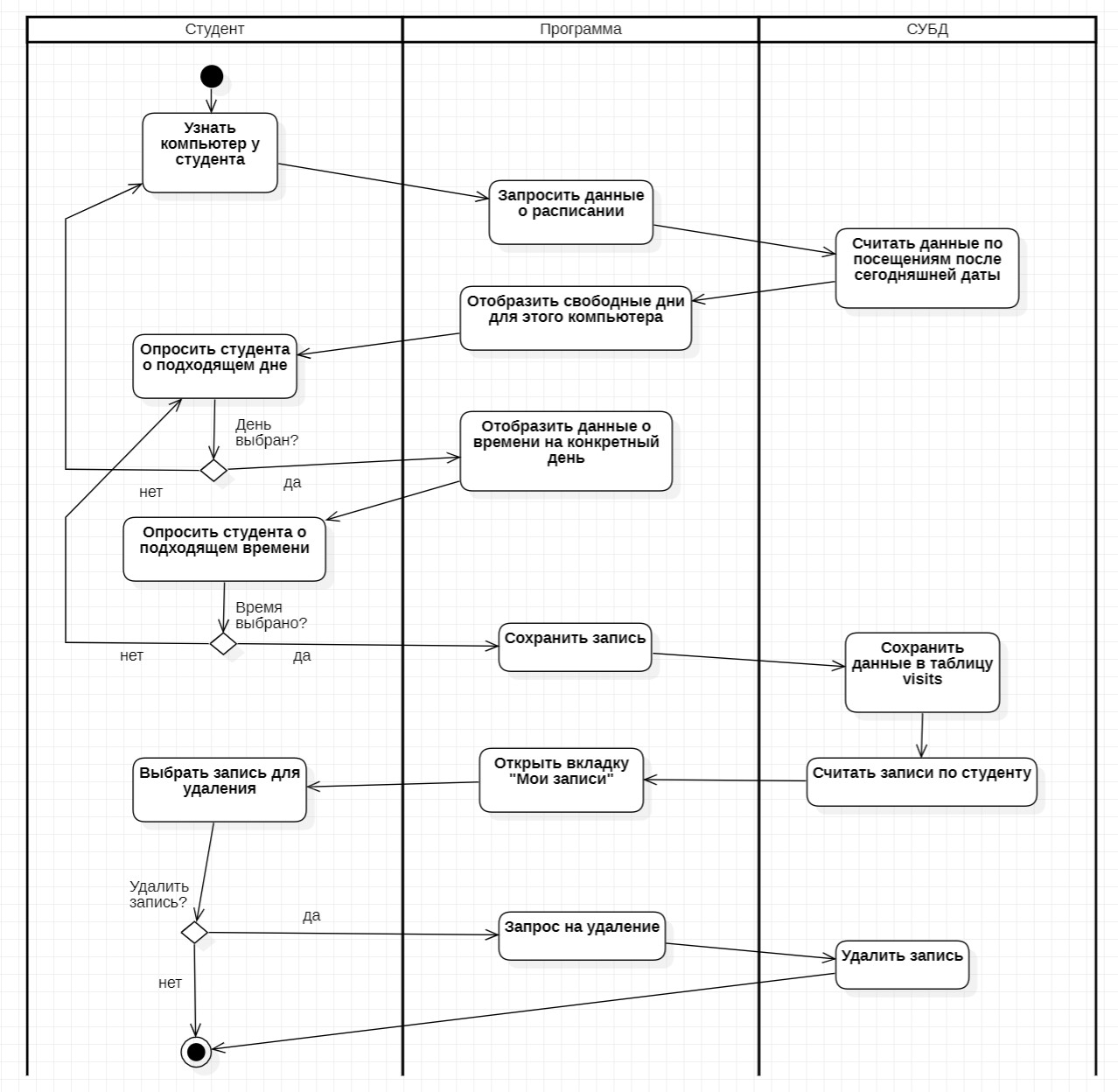
ПРИЛОЖЕНИЕ А

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



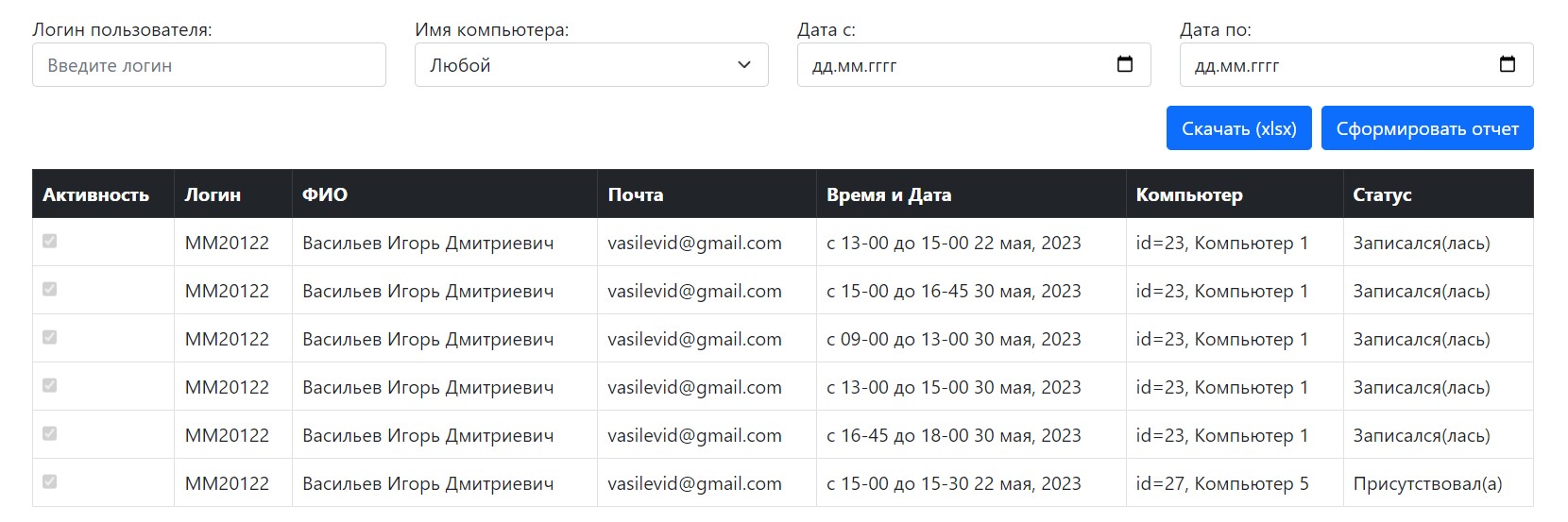
ПРИЛОЖЕНИЕ Б

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

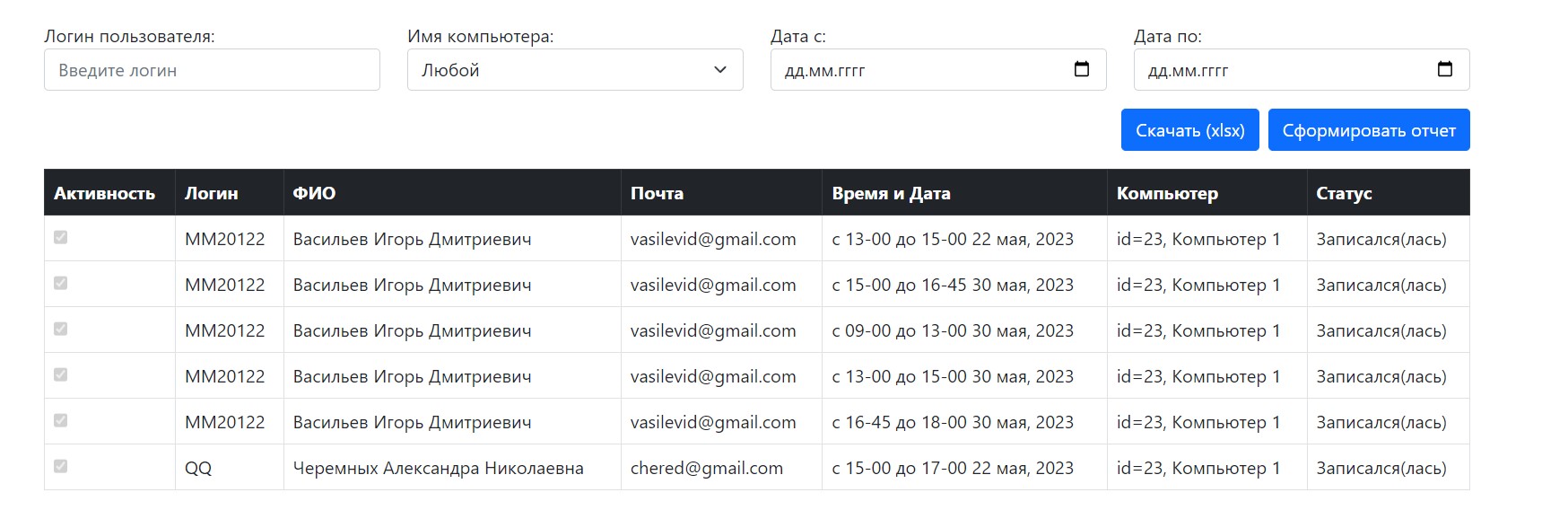


ПРИЛОЖЕНИЕ В

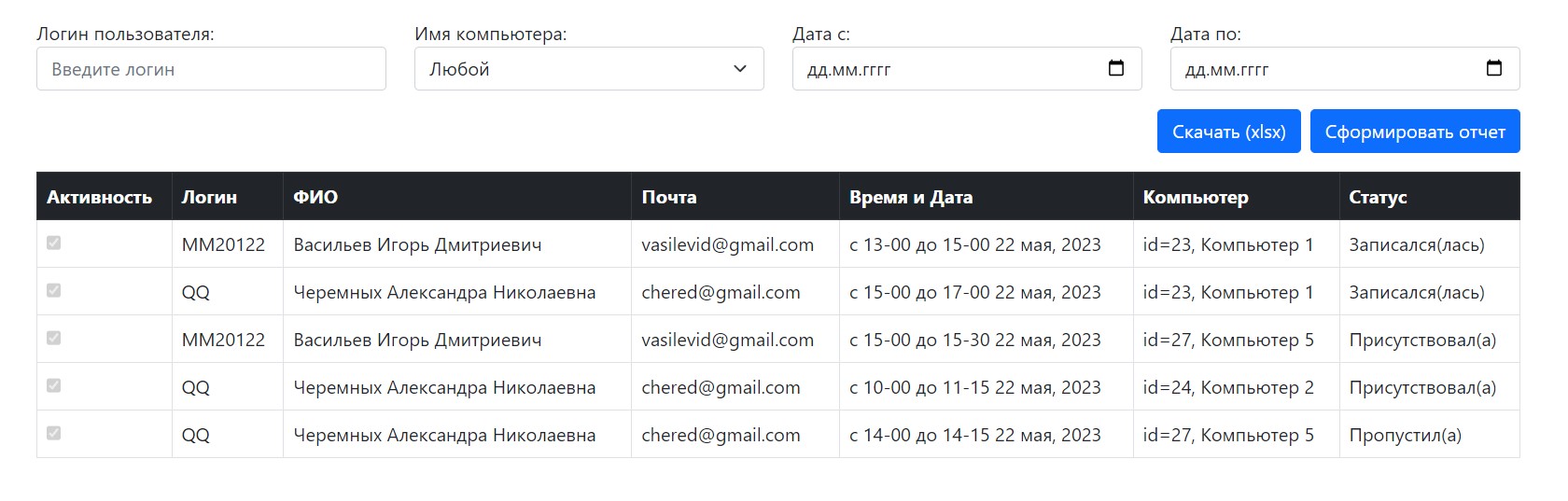
**Отчет. Сортировка по логину.**



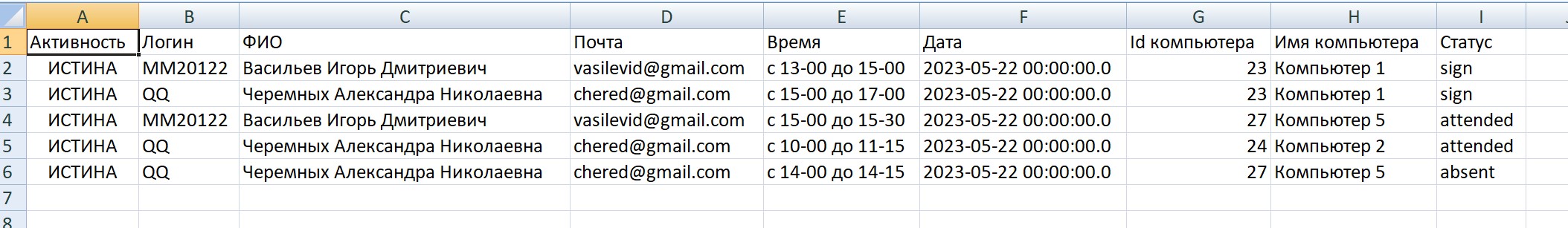
**Отчет. Сортировка по компьютеру.**



**Отчет. Сортировка по дате.**



**Отчет. Формирование xlsx.**



ПРИЛОЖЕНИЕ Г

**Диаграмма классов приложения**

