Федеральное государственное бюджетное образовательное учреждение высшего образования

«Сибирский государственный университет телекоммуникаций и информатики»

(СибГУТИ)

Курсовая работа по теме

«Учёт посещений компьютерного клуба»

Вариант №14

Выполнил: студент 3 курса

группы ИП-213

Дмитриев Антон Александрович

Проверил: Кандидат технических наук

Мерзлякова Екатерина Юрьевна

Новосибирск 2024 г.

**СОДЕРЖАНИЕ**

Оглавление

[ВВЕДЕНИЕ 4](#_Toc185497200)

[Выбор темы 4](#_Toc185497201)

[Приложения для компьютерного клуба и их возможности 4](#_Toc185497202)

[Перспективы развития 4](#_Toc185497203)

[1. Анализ задач и пользователей 5](#_Toc185497204)

[1.1. Анализ целевой аудитории 5](#_Toc185497205)

[1.2. Профиль пользователя 5](#_Toc185497206)

[1.3. Общие требования и задачи 6](#_Toc185497207)

[2. Выбор репрезентативных задач 8](#_Toc185497208)

[2.1. Список задач пользователя 8](#_Toc185497209)

[2.2. Подробное описание задач в терминах пользователя 8](#_Toc185497210)

[2.3. Список необходимых материалов для выполнения задач 8](#_Toc185497211)

[3. Заимствование 9](#_Toc185497212)

[3.1. Описание аналогичных приложений 9](#_Toc185497213)

[3.2. Применимые низкоуровневые решения 9](#_Toc185497214)

[3.3. Список принятых решений и их обоснование 9](#_Toc185497215)

[4. Черновое описание интерфейса 11](#_Toc185497216)

[4.1. Выбор моделей для построения интерфейса 11](#_Toc185497217)

[4.2 Описание интерфейса 14](#_Toc185497218)

[5. Анализ интерфейса 15](#_Toc185497219)

[5.1. Выбор репрезентативных задач 15](#_Toc185497220)

[5.2. CWT-анализ 15](#_Toc185497221)

[5.3. GOMS-анализ 16](#_Toc185497222)

[5.4. Исследование информационной производительности (метод Дж. Раскина) 19](#_Toc185497223)

[6.Прототип Интерфейса 22](#_Toc185497224)

[6.1 Экранные формы и окна сообщений 22](#_Toc185497225)

[6.2 Оценка интерфейса с помощью законов Фитса и Хика 23](#_Toc185497226)

[7 Тестирование с пользователями 26](#_Toc185497227)

[7.1 Описание пользователей 26](#_Toc185497228)

[7.2 Тестирование с пользователем 1 (Низкий уровень подготовки) 26](#_Toc185497229)

[7.3 Тестирование с пользователем 2 (Высокий уровень подготовки) 27](#_Toc185497230)

[7.4 Выводы 28](#_Toc185497231)

[Программная реализация 30](#_Toc185497232)

[Техническое задание 30](#_Toc185497233)

[Описание используемых классов Qt 30](#_Toc185497234)

[Основные функции 31](#_Toc185497235)

[Скриншоты приложения 32](#_Toc185497236)

[Заключение 35](#_Toc185497237)

[Список используемых источников 36](#_Toc185497238)

# ВВЕДЕНИЕ

## Выбор темы

Современные технологии и цифровизация стремительно изменяют множество сфер человеческой жизни, включая индустрию развлечений. Одной из ключевых составляющих этой индустрии являются компьютерные клубы, которые предоставляют пользователям доступ к мощным игровым компьютерам, комфортной атмосфере и возможности для общения с единомышленниками. С ростом популярности киберспорта и коллективных игр компьютерные клубы становятся все более востребованными, привлекая не только молодежь, но и аудиторию старшего возраста.

Моё решение разработать приложение для администратора клуба связано с желанием улучшить процесс управления такими заведениями. Личный интерес к игровой индустрии и опыт посещения подобных клубов помогли мне лучше понять их специфику, выявить основные задачи администраторов и предложить пути их оптимизации. Администраторы клубов сталкиваются с множеством вызовов: управление бронированием, контроль загруженности устройств, взаимодействие с клиентами, организация мероприятий и поддержание актуального расписания. Создание удобного цифрового инструмента для решения этих задач стало моей главной целью.

## Приложения для компьютерного клуба и их возможности

На рынке существует несколько приложений, связанных с компьютерными клубами. Например, приложение популярной сети Colizeum предлагает пользователям возможность бронирования мест, просмотра загруженности клубов и участия в мероприятиях. Однако такие решения, как правило, ориентированы на клиентов и крупные сети, оставляя локальные клубы без необходимой поддержки. Администраторы небольших клубов часто используют устаревшие системы или вовсе обходятся без цифровых инструментов, что снижает их эффективность.

## Перспективы развития

Перспективы разработки приложения для администраторов клуба включают возможность автоматизации задач управления, улучшения взаимодействия с клиентами и расширения функционала клубов. Интеграция с современными технологиями, такими как искусственный интеллект, может значительно упростить анализ данных и улучшить принятие решений. Например, приложение может рекомендовать оптимальные расписания или предлагать идеи для привлечения клиентов. Кроме того, растущая популярность виртуальной реальности и киберспортивных мероприятий создает спрос на специализированные инструменты для работы с новым оборудованием и планирования турниров. Разработка приложения для администратора клуба позволит не только улучшить управление, но и повысить уровень сервиса, предоставляемого клиентам. Это шаг вперед к профессионализации индустрии, который сделает компьютерные клубы более конкурентоспособными в современных условиях.

# 1. Анализ задач и пользователей

## 1.1. Анализ целевой аудитории

Целевая аудитория приложения для администратора клуба включает администраторов и владельцев компьютерных клубов. Основные характеристики аудитории:

* **Социально-демографические характеристики:**  
  Администраторы компьютерных клубов чаще всего — это молодые люди и взрослые в возрасте от 20 до 40 лет, работающие в сфере управления игровыми сервисами. В эту категорию входят как владельцы небольших локальных клубов, так и сотрудники крупных сетей. Уровень дохода может варьироваться, но основное внимание уделяется оптимизации работы клуба.
* **Психографические характеристики:**  
  Целевая аудитория ориентирована на повышение эффективности управления клубом, создание комфортной среды для посетителей и увеличение доходов. Эти пользователи ценят функциональные и интуитивные инструменты для автоматизации задач, такие как управление бронированием, учёт посещений, мониторинг загрузки оборудования и планирование мероприятий.
* **Поведенческие характеристики:**  
  Администраторы ищут надежные инструменты, которые облегчают выполнение рутинных задач. Они ожидают быстрого доступа к данным клуба, возможности в реальном времени управлять ресурсами, а также получать аналитические отчёты о работе клуба. Многие из них уже используют конкурирующие системы, такие как программы Colizeum Manager, и ожидают аналогичного или более высокого уровня надёжности и функциональности.
* **Географические характеристики:**  
  Основная аудитория сосредоточена в городах с активной индустрией развлечений и значительным числом компьютерных клубов. Приложение должно учитывать региональные особенности, включая специфику ценообразования и популярность определённых видов мероприятий.

## 1.2. Профиль пользователя

Демографические и психологические особенности:

Пользователи приложения — это администраторы компьютерных клубов, которые активно управляют бронированиями мест, следят за состоянием оборудования, проводят турниры и акции. Возрастная категория администраторов варьируется от 18 до 35 лет. Основной акцент делается на молодых и опытных профессионалов в возрасте 20–30 лет.

Большинство администраторов обладают высокими цифровыми навыками, так как работают с компьютерной техникой и специализированным программным обеспечением. Им важно иметь быстрый и удобный доступ к управлению бронированиями, интуитивно понятный интерфейс для просмотра расписания и данных о клиентах, а также простые инструменты для просмотра или отмены бронирований.

Задачи и цели администратора:

Основные задачи пользователей приложения включают:

* **Управление бронированиями:** создание, просмотр, изменение и удаление записей о бронировании мест в клубе.
* **Контроль доступности оборудования:** возможность редактировать статус доступности компьютеров и периферийных устройств.
* **Проведение акций и мероприятий:** управление информацией о скидках, турнирах и других событиях.
* **Обеспечение удобного взаимодействия с клиентами:** предоставление информации о доступных местах, обработка запросов пользователей, решение возможных проблем с оборудованием.

Цели администратора — оптимизация рабочего процесса в компьютерном клубе, упрощение управления ресурсами и предоставление качественного сервиса для клиентов.

Рабочая среда использования:

Администраторы используют приложение в условиях компьютерного клуба, преимущественно на стационарных компьютерах или ноутбуках. Возможна работа с приложением на планшетах для большей мобильности. Среда обычно характеризуется динамичным рабочим процессом: администраторы принимают запросы от клиентов, отслеживают расписание и оперативно решают технические вопросы.

Приложение должно обеспечивать минимальное время на выполнение рутинных задач, таких как создание новых бронирований или внесение изменений, чтобы администратор мог больше внимания уделять клиентам и техническому состоянию оборудования.

Термины пользователя:

Основные термины, которые могут быть использованы администраторами, включают:

* **Бронирование, расписание, клиенты.**
* **Место, компьютер, оборудование.**
* **Турниры, акции, скидки.**
* **Управление, изменения, удаление.**
* **Доступность, технические проблемы.**
* **Сеанс, время, дата.**

Приложение должно поддерживать использование этих терминов в интерфейсе, чтобы максимально соответствовать ожиданиям пользователей и облегчить их работу.

## 1.3. Общие требования и задачи

На основе анализа целевой аудитории выделены ключевые задачи:

* **Функциональные задачи:**
  + Управление бронированием игровых мест.
  + Мониторинг доступности и загрузки оборудования.
  + Ведение базы данных клиентов.
* **Требования к системе:**
  + Простой, интуитивно понятный интерфейс.
  + Адаптация под мобильные устройства и ПК.
  + Поддержка интеграции с базами данных клуба.
  + Возможность работы в режиме реального времени.
* **Перспективы развития:**
  + Уведомления о предстоящих событиях или технических сбоях.
  + Автоматизация учета доходов и расходов.
  + Поддержка аналитики: отчеты о популярности услуг, посещаемости и выручке.
  + Интеграция с социальными сетями для продвижения клуба и взаимодействия с клиентами.

Эти данные создают основу для разработки приложения, максимально адаптированного под задачи администраторов компьютерных клубов.

# 2. Выбор репрезентативных задач

## 2.1. Список задач пользователя

На основании анализа целевой аудитории выделены основные задачи, которые администраторы клубов решают с помощью приложения:

1. Просмотр и управление загрузкой оборудования.
2. Управление бронированиями: создание. отмена.

## 2.2. Подробное описание задач в терминах пользователя

1. **Просмотр и управление загрузкой оборудования:**  
   Администратор видит список всех компьютеров клуба с текущим статусом (свободен, забронирован) и может оперативно вносить изменения в расписание.
2. **Управление бронированиями:**  
   Администратор вручную добавляет бронирования для посетителей, изменяет время или оборудование в случае необходимости и отменяет бронирования по запросу.

## 2.3. Список необходимых материалов для выполнения задач

Для реализации описанных задач необходимы следующие ресурсы:

1. **База данных клуба:**
   * Список компьютеров с характеристиками (процессор, видеокарта, монитор и периферия).
   * Текущий статус каждого компьютера (свободен, занят).
2. **Информация о расписании:**
   * График работы клуба.
   * Временные интервалы для бронирования.
3. **Функциональные элементы приложения:**
   * Интерфейс для управления бронированиями, статусами оборудования и событиями.

Детальное описание задач позволяет сфокусироваться на разработке интерфейса и функционала, который будет максимально полезен для администраторов. Подготовленные материалы станут основой для проектирования системы.

# ****3. Заимствование****

## 3.1. Описание аналогичных приложений

**Colizeum**

Это приложение популярной сети игровых клубов, которое предоставляет пользователям возможность бронировать игровые места. Оно позволяет выбирать клубы по всей стране, видеть доступность компьютеров, просматривать характеристики оборудования и участвовать в акциях и турнирах. Приложение имеет удобный интерфейс с акцентом на минималистичный дизайн, отображение карт клубов и систему уведомлений.

## 3.2. Применимые низкоуровневые решения

1. **Окно с календарём для выбора даты**
   * Позволяет пользователям легко выбирать дату бронирования.
2. **Поля ввода времени начала и конца**
   * Удобный элемент для выбора времени начала и конца бронирования.
3. **Всплывающие окна с подсказками**
   * Простой способ предоставления пользователю уведомлений об ошибках или успешных действиях.
4. **Сетка с кнопками, обозначающими устройства**
   * **Позволяет выбрать удобные места для посетителей**

## 3.3. Список принятых решений и их обоснование

1. **Использование QCalendarWidget для выбора даты**
   * Решение принято из-за удобства работы с датами, что критично для бронирования.
   * Легкость кастомизации под задачи пользователя, включая выделение доступных и недоступных дней.
2. **QTimeEdit для выбора времени**
   * Позволяет пользователям быстро и точно выбирать временной интервал, исключая ошибки ввода.
   * Виджет интуитивно понятен и легко интегрируется с другими элементами.
3. **QTableWidget для отображения списка бронирований**
   * Это решение обеспечивает компактный и структурированный вывод данных.
   * Возможность выделения строк и отображения подробностей в одной таблице соответствует привычкам целевой аудитории.
4. **Отключение и окрашивание кнопок с использованием QPushButton**
   * Взаимодействие с недоступными компьютерами исключено за счет отключения соответствующих кнопок.
   * Цветовые изменения помогают пользователям быстрее ориентироваться в интерфейсе.
5. **Использование QMessageBox для уведомлений**
   * Позволяет информировать пользователя о состоянии его действий (успех или ошибка).
   * Простой и универсальный механизм, который не требует дополнительной визуализации.

Принятые решения были основаны на анализе привычек целевой аудитории и опыте аналогичных приложений. Ключевой задачей стало упрощение взаимодействия пользователей с системой, а заимствованные элементы позволили создать интерфейс, соответствующий современным стандартам удобства.

# ****4. Черновое описание интерфейса****

## 4.1. Выбор моделей для построения интерфейса

Для описания интерфейса будет использована комбинация нескольких моделей, каждая из которых охватывает определенный аспект взаимодействия пользователя с системой. Выбранные модели включают:

1. **Модель прямого манипулирования** – позволит описать объекты, с которыми пользователи взаимодействуют непосредственно (например, кнопки, поля ввода, таблицы).

**Метафоры:**

1. **Бронирование** — процесс выбора и резервирования игровых мест в клубах. Метафора «игровое место» будет представлять собой элемент, который можно "забронировать" или "освободить".
2. **Таблицы с данными** — используются для отображения записей о бронированиях и устройствах. Метафора «таблица» здесь представляется как организованный способ отображения информации.
3. **Кнопки и элементы управления** — эти элементы будут использоваться для выполнения действий, например, «Забронировать», «Удалить запись», и будут работать как активные объекты, с которыми пользователи напрямую взаимодействуют.

**Объекты и представления:**

1. **Календарь (QCalendarWidget)**: объект для выбора даты бронирования.
2. **Время (QTimeEdit)**: объекты для выбора времени начала и окончания бронирования.
3. **Таблица бронирований (QTableWidget)**: объект для отображения текущих записей о бронированиях.
4. **Кнопки управления (QPushButton)**: кнопки для выполнения действий (например, «Забронировать», «Удалить», «Выбрать столы»).

**Матрица прямого манипулирования:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | **Конечные объекты** |  |  |  |
| **Исходные объекты** | **Бронирование (запись в таблице)** | **Дата (календарь)** | **Время (ввод времени)** | **Устройства (список устройств)** | **Кнопки действий (например, "Забронировать")** |
| **Бронирование (таблица)** | - | Просмотр даты бронирования | Просмотр времени бронирования | Просмотр связанных устройств | Удаление бронирования |
| **Дата (календарь)** | Выбор даты для нового бронирования | - | Установка времени бронирования | Просмотр доступных устройств | Подтверждение выбора даты |
| **Время (ввод времени)** | Выбор времени для нового бронирования | Уточнение доступности даты | - | Просмотр доступных устройств | Подтверждение выбора времени |
| **Устройства (список)** | Привязка к бронированию | Фильтрация по дате | Фильтрация по времени | - | Выбор устройства для бронирования |
| **Кнопки действий** | Создание/удаление бронирования | Подтверждение выбора даты | Подтверждение выбора времени | Подтверждение выбора устройств | - |

1. **Модель действий пользователя** – определит последовательность действий, которые пользователи могут выполнять для решения своих задач в системе.

#### Роли пользователей:

1. **Администратор клуба** – единственная роль, выполняющая все действия по управлению системой, включая бронирования, управление устройствами, редактирование и удаление данных.

#### Список действий:

1. **Администратор**:
   * Просмотр и управление всеми бронированиями.
   * Удаление записей о бронированиях.
   * Управление доступностью устройств.

#### Модель действий пользователя:

| **Роль** | **Действие** | **Описание** |
| --- | --- | --- |
| Администратор | Просмотр всех бронирований | Просмотр, сортировка и фильтрация данных о бронированиях. |
|  | Удаление бронирования | Удаление записи о бронировании из базы данных. |
|  | Управление доступностью устройств | Изменение статуса устройств (доступно/недоступно). |

1. **Структурная модель** – будет использоваться для представления и организации структуры интерфейса: расположение элементов на экране, окна и меню.

**Элементы интерфейса:**

1. **Главное окно(начальный экран)**: содержит кнопку начала брони.
2. **Текстовая строка (QLineEdit)**: поле ввода имени.
3. **Календарь (QCalendarWidget)**: отображение месяца с выделением выбранных и доступных дней.
4. **Время (QTimeEdit)**: элементы ввода для времени начала и окончания бронирования.
5. **Таблица бронирований (QTableWidget)**: для отображения информации о текущих бронированиях.
6. **Кнопки (QPushButton)**: для выбора устройств и подтверждения действий, например, бронирование или удаление записи.

**Схема экранных кадров:**

1. **Экраны бронирования**:
   * Календарь для выбора даты.
   * Время для выбора интервала.
   * Кнопки выбора устройств
   * Кнопка для подтверждения бронирования.
2. **Экран администратора**:
   * Таблица с данными о текущих бронированиях.
   * Кнопки для удаления записей.

Изображение выглядит как текст, снимок экрана, Шрифт, дизайн

Автоматически созданное описание

Рисунок 1 Графическое представление

1. **Последовательно-динамическая модель** – поможет проанализировать динамику взаимодействия пользователя с системой и вовлеченные объекты интерфейса.

#### Роли пользователей:

1. **Администратор:** Управляет всеми аспектами работы системы, включая бронирования, состояние оборудования и аналитику.

**Список объектов интерфейса:**

1. Календарь.
2. Время (QTimeEdit).
3. Таблица бронирований.
4. Кнопки управления (QPushButton).

#### Таблица динамического взаимодействия:

|  |  |  |  |
| --- | --- | --- | --- |
| **Роль** | **Объект** | **Действие** | **Результат** |
| Администратор | Таблица | Просмотр бронирований | Отображение всех записей с возможностью фильтрации. |
|  | Кнопка | Нажатие "Удалить" | Удаление выбранного бронирования из таблицы. |
|  | Календарь | Выбор даты | Отображение данных о бронированиях на указанную дату. |
|  | Таблица устройств | Изменение статуса устройства | Изменение доступности устройства для клиентов. |

## 4.2 Описание интерфейса

Программа представляет собой приложение для администрирования компьютерного клуба, с возможностью создания новых бронирований, просмотра и удаления существующих.

Начальный экран представляет собой логотип клуба, кнопку начала новой бронирования и меню администрирования.

После нажатия на кнопку начала новой бронирования открывается экран с полем ввода имени посетителя, на которого будет производиться бронирование. Помимо поля ввода присутствуют две кнопки, кнопка перехода на предыдущий экран, и кнопка далее для перехода на экран выбора даты.

Экран ввода даты представляет собой виджет календаря, две кнопки, кнопка перехода на предыдущий экран, и кнопка далее для перехода на экран выбора времени.

Экран ввода времени представляет собой два поля выбора времени, две кнопки, кнопка перехода на предыдущий экран, и кнопка далее для перехода на экран выбора мест.

Экран выбора мест представляет собой несколько кнопок обозначающих устройства, которые имеют разный цвет, в зависимости от их статуса (свободно, забронировано и выбрано), список выбранных устройств и две кнопки, кнопка перехода на предыдущий экран, и кнопка подтверждения бронирования для сохранения данных в базу и перехода на начальный экран.

Меню администрирования представляет собой таблицу с данными о текущих бронированиях, поля ввода фильтров, кнопка поиска по этим фильтрам, кнопка удаления выбранного бронирования.

Дизайн интерфейса минималистичен и современен:

* Простые и заметные кнопки действий
* Яркий и привлекательный интерфейс отражающий причастность к играм

Черновое описание интерфейса позволит создать удобный и интуитивно понятный продукт, который будет соответствовать ожиданиям целевой аудитории и обеспечит эффективное выполнение задач по бронированию игровых мест и управлению ими.

# 5. Анализ интерфейса

Для анализа интерфейса выберем три главные репрезентативные задачи, проведем CWT и GOMS анализы, а также исследуем производительность экранных форм по методу Дж. Раскина.

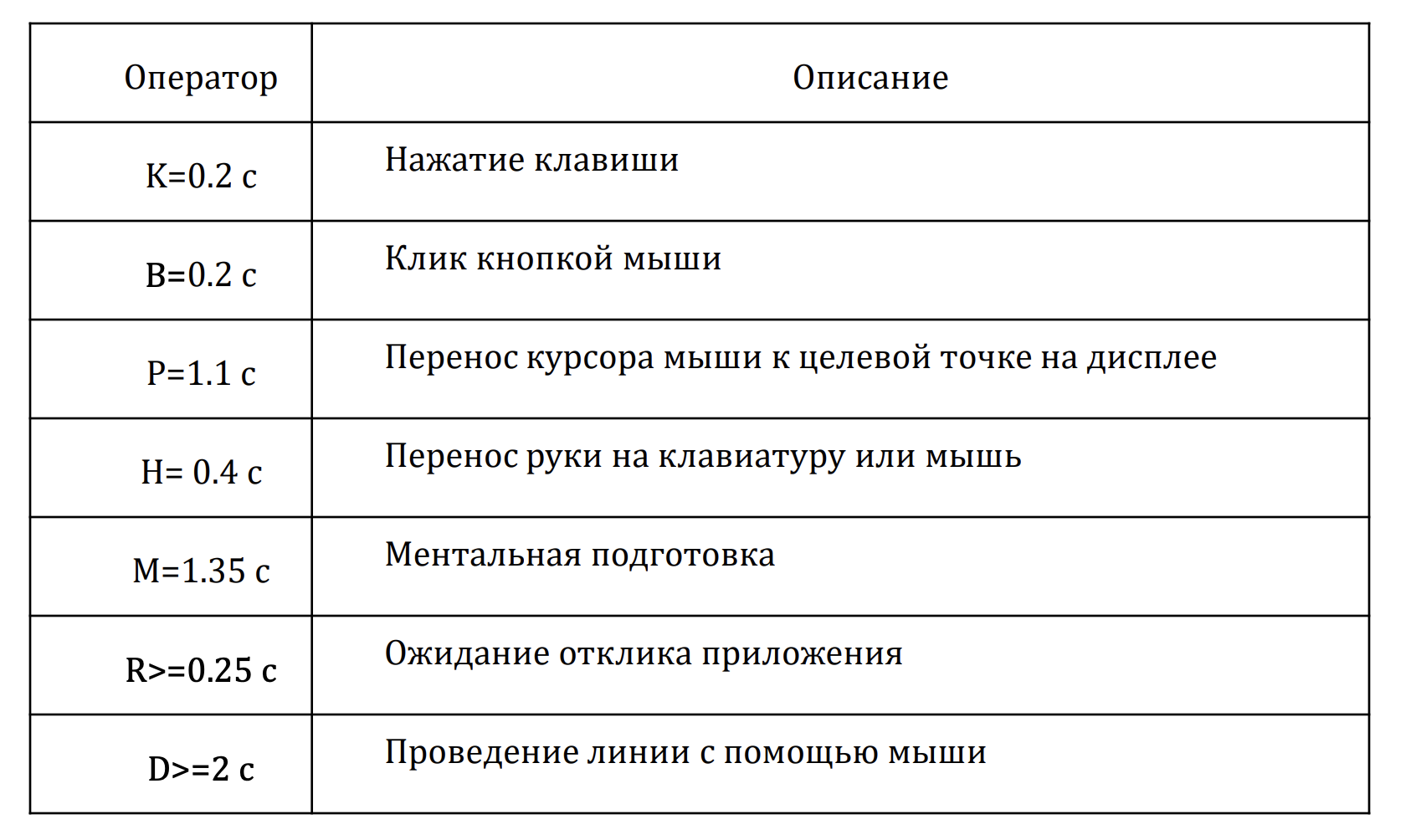
## ****5.1. Выбор репрезентативных задач****

1. **Задача 1:** Создание нового бронирования.
2. **Задача 2:** Просмотр списка бронирований
3. **Задача 3:**.Удаление бронирования.

## ****5.2. CWT-анализ****

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Формулировка задачи** | **Действие** | **История** | **Проблемы** | **Пути решения** |
| **Просмотр списка бронирований** | Открыть экран бронирований. | Администратор выбирает пункт меню "Бронирования". | Возможное затруднение при поиске нужной записи в большом списке. | Добавить фильтры по дате, имени и устройствам. |
|  | Применить фильтры к списку. | Ввод данных (например, даты) в поля фильтра. | Сложность при использовании интерфейса фильтра, если он перегружен элементами. | Упростить интерфейс фильтра: оставить только ключевые параметры для поиска. |
| **Удаление бронирования** | Найти бронирование. | Администратор использует фильтр для выбора нужной записи. | Проблема, если интерфейс не показывает полную информацию о записи до её выбора. | Добавить предварительный просмотр записи в списке бронирований. |
|  | Удалить запись. | Нажатие кнопки "Удалить" и подтверждение действия. | Пользователь может случайно удалить запись без осознания последствий. | Добавить окно подтверждения удаления |
| **Создание Бронирования** | Добавить запись | Администратор вводит параметры записи вручную | Пользователь может ошибиться с создать запись на несуществующее время или уже занятое в это время место | Поэтапно вводить данные о бронировании, создание программных проверок на занятость устройства и проверки текущего времени |

## ****5.3. GOMS-анализ****



Для каждой задачи подсчитаем количество действий, необходимых для её выполнения:

1. **Цель 1: Добавление нового бронирования**

**Подцель 1: Начало бронирования**

* 1. Переместить руку на мышь (H = 0.4 с).
  2. **Переместить курсор к кнопке начала (P = 1.1 с.).**
  3. **Кликнуть на кнопку (В = 0.2 с).**

**Подзадача 2: Ввод имени**

* 1. **Переместить курсор к полю ввода имени (P = 1.1 с).**
  2. Переместить руку на клавиатуру (H = 0.4 с).
  3. **Ввести Имя (15 символов) (К = 0.2 \*15 = 3 с).**
  4. Переместить руку на мышь (H = 0.4 с).
  5. **Переместить курсор к кнопке далее (P = 1.1 с).**
  6. **Кликнуть на кнопку далее (В = 0.2 с).**

**Подцель 3: Выбор даты**

* 1. **Переместить курсор к полю календаря (P = 1.1 с).**
  2. **Кликнуть на кнопку с датой (В = 0.2 с).**
  3. **Переместить курсор к кнопке далее (P = 1.1 с).**
  4. **Кликнуть на кнопку далее (В = 0.2 с).**

**Подцель 4: Выбор времени**

* 1. **Переместить курсор к полю времени начала (P = 1.1 с).**
  2. **Кликнуть на кнопку выбора времени(10 раз так как изначально время = 0:00) (В = 0.2 \* 10 = 2 с).**
  3. **Переместить курсор к полю времени окончания (P = 1.1 с).**
  4. **Кликнуть на кнопку выбора времени(10 раз так как изначально время = 0:00 а чаще всего берут 2 часа) (В = 0.2 \* 12 = 2.4 с).**
  5. **Переместить курсор к кнопке далее (P = 1.1 с).**
  6. **Кликнуть на кнопку далее (В = 0.2 с).**

**Подцель 6: Выбор устройств**

* 1. **Переместить курсор к свободному устройству (P = 1.1 с).**
  2. **Кликнуть на устройство (В = 0.2 с).**
  3. **Переместить курсор к кнопке подтверждения (P = 1.1 с).**
  4. **Кликнуть на кнопку подтверждения (В = 0.2 с).**

Исходная запись операций (до применения правил)

Цель: **Добавление нового бронирования**  
Подцели:

1. **Начало бронирования**
   * MH MP MB
2. **Ввод имени**
   * MP MH M15K MH MP MB
3. **Выбор даты**
   * MP MB MP MB
4. **Выбор времени**
   * MP M10B MP M12B MP MB
5. **Выбор устройств**
   * MP MB MP MB

Применение правил:

1. **Применение правила 0**:  
   Операторы **M** добавлены перед всеми действиями **K**, **P**, и **B**. Это уже выполнено в исходной записи.
2. **Применение правила 1**:  
   Удаляем операторы **M**, если следующие действия полностью ожидаемы.
   * **Подцель 1: Начало бронирования**  
     Перемещение курсора к кнопке "Начало" и клик очевидны: **M H M P M B** → **H P B**.
   * **Подцель 2: Ввод имени**  
     После перехода к полю ввода имени, перемещение руки на клавиатуру предсказуемо:  
     **M P M H M K15** M H M P M B→ **P H K15 H P B**.
   * **Подцель 3: Выбор даты**  
     Клик по кнопке даты предсказуем: **M P M B** M P M B→ **P B P B**.
   * **Подцель 4: Выбор времени**  
     Клик по кнопкам изменения времени (многократные нажатия) очевиден:  
     **M P M B10 M P M B** → **P B10 P B** , **M P M B12** M P M B→ **P B12 P B**.
   * **Подцель 5: Выбор устройств**  
     Перемещение к устройству и клик ожидаемы: **M P M B** M P MB→ **P B P B**.
3. **Применение правила 2**:  
   Внутри когнитивных единиц (последовательности действий, таких как ввод текста), удаляем все операторы **M**, кроме первого.
   * **Подцель 2: Ввод имени**  
     Ввод имени 15 символов — когнитивная единица:  
     **P H K15 H P B** (без изменений, все операторы **M** уже удалены на шаге 1).
4. **Применение правила 3**:  
   Удаляем **M** перед разделителями, если они лишние или завершение когнитивной единицы.
   * Не применяется, так как в действиях нет лишних разделителей.
5. **Применение правила 4**:  
   Удаляем **M** перед разделителями, если они завершают постоянную строку (команду).
   * Не применяется, так как клик завершает каждую подцель.
6. **Применение правила 5**:  
   Удаляем перекрывающие операторы **M**, совпадающие с оператором **R** (ожидание).
   * Нет операторов ожидания **R**, поэтому не применяется.

Оптимизированная запись операций

1. **Начало бронирования**  
   **H P B**
2. **Ввод имени**  
   **P H K15 H P B**
3. **Выбор даты**  
   **P B P B**
4. **Выбор времени**  
   **P B10 P B12 P B**
5. **Выбор устройств**  
   **P B P B**

Итоговая запись операций

**H P B → P H K15 H P B → P B P B → P B10 P B12 P B → P B P B**

Теперь запись операций учитывает оптимизацию взаимодействия и соответствует правилам, что снижает общее количество действий и время выполнения.

## ****5.4. Исследование информационной производительности (метод Дж. Раскина)****

Исследование информационной производительности для Цели 1: **Добавление нового бронирования**

#### 1. Описание действий и расчет информационной нагрузки:

1. **Запуск приложения**  
   Предположим, пользователь выбирает приложение из списка на устройстве с **10 приложениями**.  
   Количество вариантов: N=10.  
   Информация для выбора:

бита

1. **Выбор начала бронирования**  
   Предположим, пользователь выбирает действие "Начало бронирования" из **2 вариантов** (например, "Добавить бронирование", "Просмотр бронирований").  
   Количество вариантов: N=2.  
   Информация для выбора:

бит

1. **Ввод имени**  
   Ввод текста: **15 символов**. Используем алфавит из **46 символов** (латиница, цифры, спец. символы).  
   Информация для ввода:

бита

1. **Выбор даты**  
   Выбор даты из календаря (в среднем, N=30N = 30, количество дней в месяце).  
   Информация для выбора:

бита

.

1. **Выбор времени начала и окончания**  
   Выбор времени из диапазона 00:0000:00–23:5923:59 (всего N=24×60=1440 вариантов).  
   Информация для выбора:

бита

1. **Выбор устройства**  
   Выбор из доступных **10 устройств** (например, столы в компьютерном клубе).  
   Информация для выбора:

бита

#### 2. Итоговая информационная нагрузка:

Hобщая=Hзапуск+Hначало+Hимя+Hдата+Hвремя+Hустройство

Hобщая=3.32+1.58+82.8+4.91+20.98+3.32=116.91 бит

#### 3. Время выполнения действий:

1. **Запуск приложения**: Tзапуск=2 с.
2. **Выбор начала бронирования**: Tначало=1.5 с.
3. **Ввод имени**: Tимя=10 с.
4. **Выбор даты**: Tдата=3 с.
5. **Выбор времени**: Tвремя=4.4 с.
6. **Выбор устройства**: Tустройство=2 с.

Общее время:

Tобщая=Tзапуск+Tначало+Tимя+Tдата+Tвремя+Tустройство

Tобщая=2+1.5+10+3+4.4+2=22.9 с.

#### 4. Эффективность интерфейса:

Формула Дж. Раскина:

E=Hобщая/Tобщая

Подставляем значения:

E=116.9122.9≈5.11 бит/с

Итоги и выводы:

* **Общая информационная нагрузка**: Hобщая≈116.91 бит.
* **Общее время выполнения**: Tобщая≈22.9 с.
* **Эффективность интерфейса**: E≈5.11 бит/с

Интерфейс приложения показывает **высокую эффективность (5.11 бит/с)**. Улучшить показатели можно за счет:

* Уменьшения времени на ввод текста (например, автозаполнение имени).
* Упрощения выбора времени (предоставление наиболее популярных вариантов).

# 6.Прототип Интерфейса

## 6.1 Экранные формы и окна сообщений

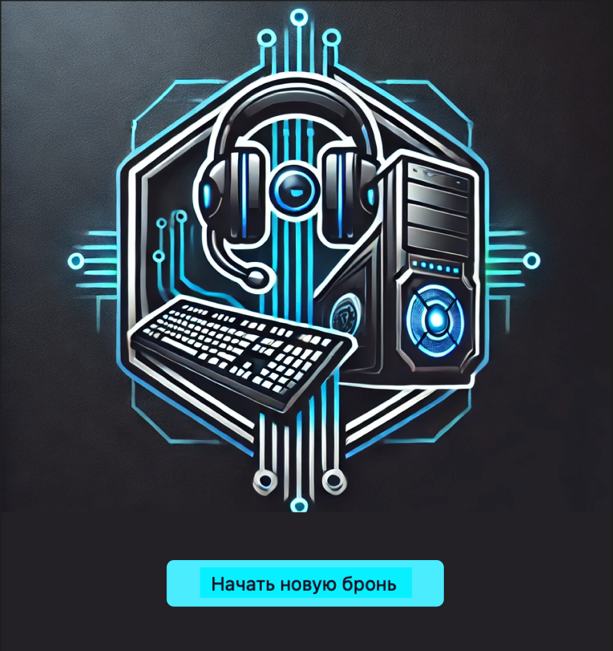


Рисунок 2 Начальный экран

Изображение выглядит как текст, снимок экрана, дизайн

Автоматически созданное описание

Рисунок 3 Экран ввода имени

Изображение выглядит как текст, снимок экрана, дисплей, мультимедиа

Автоматически созданное описание

Рисунок 4 Экран выбора даты

Изображение выглядит как снимок экрана, текст, дизайн

Автоматически созданное описание

Рисунок 5 Экран ввода времени

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание

Рисунок 6 Экран выбора мест

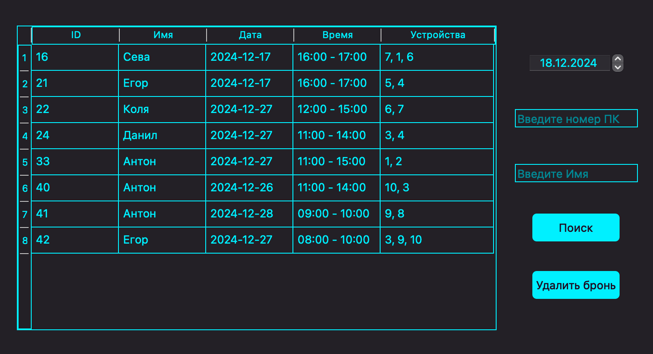


Рисунок 7 Экран с бронированиями

## 6.2 Оценка интерфейса с помощью законов Фитса и Хика

#### 1. **Закон Хика** (Время выбора между вариантами)

Закон Хика определяет время, которое требуется пользователю для выбора между несколькими вариантами. Это время зависит от количества альтернатив NN, причем оно растет логарифмически:

T=a+b×log2(N+1)T

где:

* T— время выбора (в секундах),
* N— количество вариантов,
* a и b — эмпирические константы (зависят от интерфейса и пользователя).

Мы будем использовать следующие данные:

* **Выбор устройства**: Пусть пользователь выбирает одно из 10 устройств (например, компьютеров в клубе).
  + N = 10,
  + a = 0.4,
  + b = 0.2.

Tустройство=0.4+0.2×log2(10+1)=0.4+0.2×log2(11)≈0.4+0.2×3.46=0.89 с.

* **Выбор даты и времени**: Пусть у нас есть 30 возможных вариантов для выбора даты (количество дней в месяце), и 1440 вариантов для выбора времени (24 часа × 60 минут).
  + Для даты: N = 30,
  + Для времени: N = 1440.

Tдата=0.4+0.2×log2(30+1)=0.4+0.2×log2(31)≈0.4+0.2×4.95=1.39 с.

Tвремя=0.4+0.2×log2(1440+1)=0.4+0.2×log2(1441)≈0.4+0.2×10.47=2.49 с.

**Суммарное время выбора по закону Хика**:

Tвыбор= Tустройство+Tдата+Tвремя=0.89+1.39+2.49=4.77 с.

#### 2. **Закон Фитса** (Время нажатия на цель)

Закон Фитса определяет время, необходимое для перемещения указателя на цель (например, кнопку). Оно зависит от расстояния DD до цели и её ширины WW:

T=a+b×log2(D/W+1)T

где:

* T — время нажатия (в секундах),
* D — расстояние до цели (в пикселях),
* W — ширина цели (в пикселях),
* a и b — эмпирические константы.

Для наших кнопок:

* **Кнопка "Подтвердить"** (ширина 100px, расстояние 75px):

Tподтвердить=0.05+0.15×log2(75100+1)=0.05+0.15×log2(1.75)≈0.05+0.15×0.77=0.16 с.

* **Кнопка устройства** (ширина 36px, расстояние 75px):

Tустройство=0.05+0.15×log2(7536+1)=0.05+0.15×log2(3.08)≈0.05+0.15×1.63=0.29 с.

* **Кнопка "Далее"** (ширина 100px, расстояние 75px):

Tдалее=0.05+0.15×log2(75100+1)=0.05+0.15×log2(1.75)≈0.05+0.15×0.77=0.16 с.

**Суммарное время нажатия по закону Фитса**:

Tнажатие=Tподтвердить+Tустройство+Tдалее=0.16+0.29+0.16=0.61 с.

#### 3. **Общее время взаимодействия**:

**Общее время взаимодействия** = Время выбора по закону Хика + Время нажатия по закону Фитса:

Tобщая=Tвыбор+Tнажатие=4.77+0.61=5.38 с.

Выводы:

* **Время выбора (Закон Хика)**: 4.77 с.
* **Время нажатия (Закон Фитса)**: 0.61 с.
* **Общее время взаимодействия**: 5.38 с.

Это время взаимодействия является достаточно хорошим, но можно уменьшить его за счет:

1. **Оптимизации выбора**: Уменьшение числа вариантов (например, уменьшение количества доступных устройств или сокращение списка дат).
2. **Увеличения размеров кнопок**: Увеличение ширины кнопок (например, кнопки "Далее" и "Подтвердить") для снижения времени, необходимого для наведения и нажатия.
3. **Оптимизации расстояний**: Уменьшение расстояния между полями ввода и кнопками.

Для тестирования вашего приложения, которое является частью компьютерного клуба, я подготовил описание тестирования с двумя типами пользователей — с низким и высоким уровнем подготовки. Основной акцент в этом тестировании сделан на удобство интерфейса, точность взаимодействия с элементами и понимание пользователями функциональности на разных экранах.

# 7 Тестирование с пользователями

## 7.1 Описание пользователей

1. **Пользователь 1:**
   * **Возраст:** 18 лет.
   * **Уровень подготовки:** низкий (часто пользуется компьютером, но только для базовых задач, например, соцсети, браузинг).
   * **Не знаком с подобным приложением.**
   * **Ожидания пользователя:**
     + Интуитивно понятный интерфейс.
     + Простота взаимодействия с приложением, без необходимости в обучении.
2. **Пользователь 2:**
   * **Возраст:** 30 лет.
   * **Уровень подготовки:** высокий (хорошо разбирается в компьютерах, в том числе в приложениях для бронирования).
   * **Знаком с подобными приложениями.**
   * **Ожидания пользователя:**
     + Минимальная нагрузка на внимание, быстрый и удобный доступ ко всем функциям приложения.

## 7.2 Тестирование с пользователем 1 (Низкий уровень подготовки)

| **Действия пользователя** | **Реакция системы** | **Реакция и ожидания пользователя** |
| --- | --- | --- |
| Открытие приложения | Приложение загружается, отображается главный экран с логотипом и кнопками. | Ожидает, что приложение откроется быстро и будет доступно для использования. |
| Нажатие на кнопку "Начать бронь" | Открывается экран с полем для ввода имени пользователя. | Ожидает, что будет предложено ввести имя, и форма будет удобной для ввода. |
| Ввод имени в поле | Имя сохраняется автоматически. | Пользователь ожидает, что поле ввода будет работать без ошибок. |
| Нажатие на кнопку "Далее" | Открывается экран выбора даты. | Ожидает, что календарь будет прост в использовании и позволит выбрать день без сложности. |
| Выбор даты на календаре | Дата выбирается, сохраняется автоматически. | Пользователь ожидает, что будет легко выбрать нужную дату. |
| Нажатие на кнопку "Далее" | Открывается экран выбора времени. | Ожидает, что выбор времени будет простым и понятным. |
| Выбор времени начала и конца бронирования | Время сохраняется автоматически. | Пользователь ожидает, что время будет удобно вводиться, и он сможет корректно установить нужное время. |
| Нажатие на кнопку "Далее" | Открывается экран с кнопками выбора компьютеров. | Пользователь ожидает, что кнопки будут визуально понятны и легки для выбора. |
| Выбор компьютеров (до нескольких устройств) | Кнопки изменяют цвет, и описание устройств отображается слева. | Ожидает, что будет легко выбрать компьютер, и все данные о компьютерах будут доступны. |
| Нажатие на кнопку "Подтвердить" | Сохранение данных в текстовый файл, возврат на главный экран. | Ожидает, что все данные будут сохранены, и он вернется на главный экран без ошибок. |

**Проблемы, которые могут возникнуть у пользователя 1:**

* **Трудности с точностью нажатия:** кнопки и элементы управления могут быть слишком мелкими или неудобно расположенными, что усложнит взаимодействие, особенно при выборах (например, компьютеров).
* **Неинтуитивные переходы:** неопытный пользователь может не сразу понять, как вернуться на предыдущий экран или отменить действия.

**Предложенные улучшения:**

* Увеличение размеров кнопок и элементов управления, чтобы улучшить точность взаимодействия.
* Добавление подсказок или анимаций для более наглядного понимания, что происходит на каждом экране.
* Размещение кнопки "Назад" в более заметном месте.

## 7.3 Тестирование с пользователем 2 (Высокий уровень подготовки)

| **Действия пользователя** | **Реакция системы** | **Реакция и ожидания пользователя** |
| --- | --- | --- |
| Открытие приложения | Приложение быстро загружается и показывает главный экран с логотипом и кнопками. | Понимает интерфейс и ожидает, что все функции будут готовы к быстрому использованию. |
| Нажатие на кнопку "Начать бронь" | Открывается экран для ввода имени пользователя. | Ожидает мгновенный переход к полю ввода и готов к быстрому вводу. |
| Ввод имени | Имя сохраняется автоматически. | Пользователь ожидает, что ввод будет быстрым и не вызовет проблем. |
| Нажатие на кнопку "Далее" | Открывается экран для выбора даты. | Понимает интерфейс и ожидает, что календарь будет работать без задержек. |
| Выбор даты | Выбирает дату на календаре без ошибок. | Ожидает, что весь процесс выбора даты будет быстрым и точным. |
| Нажатие на кнопку "Далее" | Открывается экран выбора времени. | Ожидает быстрого и точного выбора времени без дополнительных шагов. |
| Выбор времени | Время устанавливается без задержек. | Пользователь ожидает, что время будет легко настроить. |
| Нажатие на кнопку "Далее" | Открывается экран с выбором компьютеров. | Ожидает мгновенного появления всех доступных компьютеров и информации о стоимости. |
| Выбор компьютеров | Выбирает нужные компьютеры, и система отображает их описание и стоимость. | Понимает, что можно выбрать несколько компьютеров, и ждет удобного интерфейса для этого. |
| Нажатие на кнопку "Подтвердить" | Все данные сохраняются, и приложение возвращается на главный экран. | Ожидает, что подтверждение будет быстрым и вся информация будет правильно сохранена. |

**Проблемы, которые могут возникнуть у пользователя 2:**

* **Отсутствие горячих клавиш для быстрой навигации**: пользователь, привыкший к быстрым действиям, может почувствовать недостаток быстрого доступа к функциям.
* **Малые задержки при загрузке экрана с компьютерами**: если экраны загрузки с компьютерами или тарифами не будут быстрыми, это может замедлить работу даже опытных пользователей.

**Предложенные улучшения:**

* Внедрение горячих клавиш для перехода между экранами или для выбора действия.
* Оптимизация производительности, чтобы минимизировать задержки на экранах с большими элементами управления.
* Улучшение расположения элементов на экране (например, более очевидное размещение кнопок возврата или подтверждения).

## 7.4 Выводы

**Тестирование с неопытным пользователем (Пользователь 1):**

* **Проблемы:**
  + Трудности с точностью наведения на мелкие кнопки или элементы управления.
  + Недостаток подсказок, которые могли бы облегчить понимание интерфейса.
* **Рекомендации:**
  + Увеличение размеров кнопок и улучшение их расположения.
  + Введение подсказок для улучшения понимания интерфейса.
  + Явное выделение активных элементов, чтобы пользователи могли легко ориентироваться.

**Тестирование с опытным пользователем (Пользователь 2):**

* **Проблемы:**
  + Отсутствие горячих клавиш для быстрого доступа к основным функциям.
  + Замеченные задержки в переходах между экранами или при загрузке информации.
* **Рекомендации:**
  + Внедрение горячих клавиш для улучшения удобства быстрого использования.
  + Оптимизация работы приложения для уменьшения задержек, особенно при загрузке данных.
  + Улучшение расположения и логики переходов между экранами для повышения эффективности взаимодействия.

**Общие выводы:**

* Для **неопытных пользователей** важно предоставить понятный и доступный интерфейс с дополнительными подсказками и более крупными элементами управления.
* Для **опытных пользователей** необходимо ускорить работу системы и предоставить дополнительные возможности для быстрого взаимодействия, такие как горячие клавиши и оптимизация скорости работы.

# Программная реализация

## Техническое задание

Приложение “Учёт посещений компьютерного клуба” должно позволять пользователю:

1. Создание новых бронирований

2. Удобное управление текущими бронированиями.

Технические требования:

1. Использование Qt как основного инструмента для разработки.
2. Программа должна работать на macOS\Windows\Linux.
3. Все пользовательские действия должны сопровождаться визуальной обратной связью.
4. Данные должны сохраняться локально в базе данных SQLite.

## Описание используемых классов Qt

1. *QMainWindow*: Основной класс для создания главного окна приложения. Он предоставляет возможности для добавления меню, тулбаров и центральных виджетов.
2. *QGridLayout*: Класс для создания сеточной компоновки, которая позволяет размещать виджеты в строках и столбцах с возможностью настройки их размеров.
3. *QLabel*: Класс для отображения текста или изображения на экране. Это статический виджет, который не взаимодействует с пользователем.
4. *QHBoxLayout*: Класс для горизонтальной компоновки виджетов. Все элементы, добавленные в этот контейнер, будут выровнены горизонтально.
5. *QLineEdit*: Класс для однострочного текстового ввода, который позволяет пользователю вводить и редактировать текст.
6. *QPushButton*: Класс для кнопок, которые могут реагировать на нажатие и выполнять действия.
7. *QDialog*: Базовый класс для создания диалоговых окон. Диалоги могут быть модальными или немодальными.
8. *QListWidget*: Класс для отображения списка элементов, где каждый элемент может быть представлен текстом, изображением или комбинацией.
9. *QSqlDatabase*: Класс для работы с базой данных. Позволяет устанавливать соединение с базой данных и выполнять операции с ней.
10. *QSqlQuery*: Класс для выполнения SQL-запросов к базе данных и обработки результатов.
11. *QSqlError*: Класс для обработки ошибок, связанных с SQL-запросами или базой данных.
12. *QPixmap*: Класс для работы с изображениями. Используется для хранения и отображения графики, таких как изображения и иконки.
13. *QMessageBox*: Класс для создания стандартных сообщений пользователю, таких как предупреждения, ошибки или подтверждения.
14. *QString*: Класс для работы со строками в Qt. Предоставляет множество методов для манипуляции строками (например, поиск, замену, разделение).
15. ***QDate*:** Класс для представления и работы с датами. Он используется для хранения и манипуляции датами, такими как текущая дата, дата начала и окончания бронирования.
16. ***QTime:*** Класс для представления и работы с временем. Используется для хранения и манипуляции временем, например, для выбора времени начала и окончания бронирования.
17. ***QTimeEdit:*** Класс для создания виджета для выбора времени. Применяется для ввода времени начала и окончания бронирования, предоставляя пользователю удобный интерфейс для выбора времени.
18. ***QCalendarWidget:*** Класс для отображения календаря и выбора даты. Используется на экране выбора даты для удобства пользователя при выборе дня для бронирования.
19. ***QDateEdit:*** Класс для создания виджета, который позволяет пользователю редактировать дату. Это может быть полезно, если требуется вручную вводить дату вместо выбора её из календаря.
20. ***QTableWidget:*** Класс для отображения данных в виде таблицы. Мог бы быть использован для отображения списка бронирований, компьютеров или других данных, связанных с системой бронирования, с возможностью сортировки, поиска и редактирования.

## Основные функции

MainWindow::MainWindow(QWidget \*parent) – отвечает за отрисовку главного экрана

NameScreen::NameScreen(QWidget \*parent) – отвечает за отрисовку экрана ввод имени

CalendarScreen::CalendarScreen(QString name, QWidget \*parent) – отвечает за отрисовку экрана выбора даты

TimeScreen::TimeScreen(QString name, QDate date, QWidget \*parent) – отвечает за отрисовку экрана выбора времени

DeviceScreen::DeviceScreen(QString name, QDate date, QTime startTime, QTime endTime, QWidget \*parent) – отвечает за отрисовку экрана выбора устройства

BookingsScreen::BookingsScreen(QWidget \*parent) – отвечает за отрисовку экрана с бронированиями

void on\_nextButton\_clicked() – отвечает за отрисовку следующего экрана

void CalendarScreen::on\_backButton\_clicked() – отвечает за отрисовку предыдущего экрана

void DeviceScreen::on\_confirmButton\_clicked() – отвечает за запись данных о бронировании в базу данных

void BookingsScreen::on\_deleteButton\_clicked() – отвечает за удаление строки из базы данных

void BookingsScreen::on\_searchButton\_clicked() – отвечает за фильтрацию при просмотре базы данных

## Скриншоты приложения



Рисунок 8 Начальный экран

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Автоматически созданное описание

Рисунок 9 Экран выбора даты

Изображение выглядит как текст, снимок экрана, дизайн

Автоматически созданное описание

Рисунок 10 Экран выбора времени

Изображение выглядит как текст, снимок экрана, число

Автоматически созданное описание

Рисунок 11 Экран с выбором устройств

Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание

Рисунок 12 Таблица бронирований

База данных

Изображение выглядит как текст, снимок экрана, Шрифт, табло

Автоматически созданное описание

Рисунок 13 База данных

# Заключение

Проведённая работа по разработке интерфейса системы бронирования компьютеров для компьютерного клуба продемонстрировала важность создания удобного, интуитивно понятного и функционального приложения, которое соответствует потребностям пользователей с разным уровнем подготовки.

**Ключевые достижения разработки:**

* **Оптимизация пользовательского интерфейса.** Были внесены изменения для повышения удобства и скорости взаимодействия с элементами приложения, включая увеличение размеров кнопок, улучшение их расположения, добавление кнопок возврата и подсказок, а также упрощение структуры экранов.
* **Универсальность дизайна.** Интерфейс адаптирован для разных категорий пользователей. Для новичков предусмотрены визуальные подсказки и понятная структура экранов, а для опытных пользователей обеспечены минимальные задержки и высокая скорость выполнения операций.
* **Расширение функциональности.** Приложение реализует последовательный процесс бронирования, включая ввод имени, выбор даты и времени, выбор компьютеров с разными тарифами и сохранение данных бронирования в текстовый файл. Это делает его эффективным инструментом для управления процессом аренды оборудования.

**Результаты тестирования:**

Проведённое тестирование с участием пользователей разного уровня подготовки подтвердило, что внесённые изменения существенно повысили удобство использования приложения:

* **Неопытные пользователи** быстрее осваивают приложение благодаря интуитивно понятной навигации и упрощённому взаимодействию с интерфейсом.
* **Опытные пользователи** оценили скорость работы системы, логичность переходов между экранами и возможность гибко управлять процессом бронирования.

**Итог:**

Разработанное приложение для бронирования компьютеров обеспечивает комфортное и эффективное взаимодействие для всех пользователей, сочетая простоту интерфейса с функциональными возможностями. Это решение помогает организовать удобный процесс бронирования, способствуя положительному пользовательскому опыту и увеличивая общее качество предоставляемых услуг в компьютерном клубе.

# Список используемых источников

* 1. Документация Qt [URL-Ссылка: <https://doc.qt.io/>]
  2. Mike Owens, M.O. The Definitive Guide to SQLite / M.O. Mike Owens. – Москва : Apress, 2020. – 362 с.
  3. Документация по современным стандартам C++ [URL-ссылка: <https://en.cppreference.com/w/>]
  4. Руководство для начинающих по Qt [URL-Ссылка: <https://www.learncpp.com/cpp-tutorial/qt-for-beginners/>]