МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«Курганский государственный университет»

Институт математики и интеллектуальных систем

Кафедра «Программное обеспечение автоматизированных систем»

**ПРОЕКТ**

«Физкультурно-оздоровительный комплекс»

Работу выполнил:

студент группы ИТ-1035223

специальности 10.05.03 - "Информационная безопасность автоматизированных систем" (очная форма обучения)

Чупров Александр Денисович

Проверил:

доц. Волк В. К.

Дата сдачи:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Оценка: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Курган 2024

**ОГЛАВЛЕНИЕ**

[1 Стадия “Техническое задание” 3](#_Toc167981068)

[1.1 Общая постановка задачи 3](#_Toc167981069)

[1.2 Разработка терминологического словаря 4](#_Toc167981070)

[1.3 Общие требования к проектируемой системе 8](#_Toc167981071)

[1.4 Категории пользователей проектируемой системы 8](#_Toc167981072)

[2 Стадия “Эскизный проект” 9](#_Toc167981073)

[2.1 Разработка программной архитектуры системы 10](#_Toc167981074)

[2.2 Разработка диаграмм вариантов использования компонентов 11](#_Toc167981075)

[3 Стадия “Технический проект” 14](#_Toc167981076)

[3.1 Разработка диаграмм концептуальных классов 14](#_Toc167981077)

[3.2 Разработка диаграмм программных классов 15](#_Toc167981078)

[3.3 Разработка диаграмм состояний 17](#_Toc167981079)

[4 Обзор результатов программного проекта 19](#_Toc167981080)

[Список литературы 20](#_Toc167981081)

# **1 Стадия “Техническое задание”**

# **Общая постановка задачи**

**Наименование разработки:** Информационная система для управления клиентами и персоналом спортивной организации.

**Назначение системы:** автоматизация процессов взаимодействия с клиентами, управления тренерским составом, ведения расписания занятий и формирования отчетной документации.

**Область применения:** автоматизация учета клиентов и сотрудников, формирование расписаний и ведение отчетности в спортивной организации (например, фитнес-клубе или спортивной секции).

**Масштаб развёртывания системы:** внутренняя сеть организации или облачное развертывание с доступом через интернет.

Таблица 1.1 – клиенты и бизнес-процессы предметной области проекта

|  |  |
| --- | --- |
| **Пользователи** | **Основные бизнес-процессы** |
| **Клиенты** (гости, зарегистрированные, постоянные) | - Просмотр прайс-листа и секций- Запись в группы- Просмотр расписания занятий- Контроль посещаемости и оплаты |
| **Менеджер по работе с клиентами** | - Регистрация клиентов- Контроль посещений и платежей- Поддержка связи с клиентами |
| **Менеджер по работе с персоналом** | - Прием и увольнение тренеров- Распределение тренеров по группам- Анализ загруженности тренерского состава |
| **Администратор системы / руководство** | - Формирование расписания занятий- Анализ загруженности залов- Получение аналитических и отчетных данных (финансовая отчетность, популярность услуг по категориям клиентов) |

# **Разработка терминологического словаря**

Таблица 1.2 – Терминологический словарь

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Термин** | **Применение** | **Комментарий** |
| 1 | Клиент | Посетитель спортивной организации, заинтересованный в получении услуг | Может быть гостем, зарегистрированным или постоянным клиентом |
| 2 | Менеджер по работе с клиентами | Сотрудник, отвечающий за регистрацию клиентов, сопровождение и контроль посещаемости | Осуществляет взаимодействие с клиентами и следит за оплатами |
| 3 | Менеджер по работе с персоналом | Сотрудник, управляющий тренерским составом | Ведёт кадровый учет и распределяет тренеров по группам |
| 4 | Тренер | Специалист, проводящий групповые или индивидуальные занятия | Может быть назначен на несколько групп |
| 5 | Группа клиентов | Объединение клиентов по типу занятий или уровню подготовки | Используется для формирования расписаний и контроля посещений |
| 6 | Занятие | Единица тренировочного процесса, проводимая по расписанию | Содержит информацию о времени, месте, тренере и группе клиентов |
| 7 | Расписание | График проведения занятий в спортивных залах | Формируется с учетом загруженности залов и тренеров |
| 8 | Прайс-лист | Перечень услуг, предлагаемых организацией, с указанием стоимости | Может содержать акции и скидки |
| 9 | Услуга | Определённый вид тренировочного или вспомогательного обслуживания | Пример: групповое занятие, индивидуальная тренировка |
| 10 | Финансовая отчётность | Совокупность документов, отражающих денежные потоки организации | Используется для анализа прибыли и расходов |
| 11 | Популярность услуг | Показатель, отражающий востребованность определённых услуг среди клиентов | Вычисляется по количеству посещений и записей |
| 12 | Кадровый учёт | Система учета данных о сотрудниках | Включает приём, увольнение и распределение тренеров |
| 13 | Регистрация клиента | Процесс занесения данных нового клиента в систему | Включает информацию об оплате, контактные данные, выбранные услуги |
| 14 | Контроль посещений | Слежение за фактическим присутствием клиентов на занятиях | Используется для внутренней отчетности и обратной связи с клиентами |
| 15 | Спортивный зал | Помещение, где проводятся занятия | Может иметь разную вместимость и доступность по времени |

# 

# **1.3 Общие требования к проектируемой системе**

**Архитектурные и эксплуатационные требования:**

* **Масштаб развертывания:** внутренняя сеть спортивной организации или облачный сервер с доступом через интернет;
* **Интерфейс системы:** веб-приложение с возможной адаптацией под мобильные устройства (через адаптивный дизайн или отдельное приложение);
* **Хранилище данных:** централизованное, реализовано с использованием СУБД, содержащей таблицы клиентов, сотрудников, занятий, расписаний и финансовых операций;
* **Безопасность:** обязательная аутентификация пользователей и разграничение прав доступа в зависимости от их роли (клиент, менеджер по персоналу, менеджер по клиентам).

**Требования к функциональным характеристикам:**

* **Управление доступом:** разграничение прав на просмотр и редактирование данных (например, менеджеры не имеют доступа к аналитике, предназначенной для руководства);
* **Формирование прайс-листа:** возможность ввода, редактирования и публикации информации об услугах, ценах, акциях;
* **Кадровый учёт:** добавление и удаление тренеров, распределение их по группам, учёт нагрузки;
* **Регистрация клиентов:** фиксация анкетных данных, выбор услуг, контроль оплат и посещений;
* **Расписание занятий:** создание и редактирование расписаний с учётом загруженности залов и тренерского состава;
* **Аналитика и отчётность:** сбор статистики по услугам, посещаемости и финансам для формирования управленческих решений;
* **Авторизация пользователей:** доступ к функционалу осуществляется только после входа в систему под учётной записью, соответствующей роли пользователя.

# **Категории пользователей проектируемой системы**

Схема взаимодействия пользователей системы с её функциональными модулями, реализующими основные бизнес-процессы (см. таблицу 1.1), представлена в обобщённой диаграмме вариантов использования (рисунок 1.1).

* **Клиент** выполняет функции просмотра прайс-листа, записи в группы, контроля посещений и оплаты занятий.
* **Менеджер по работе с клиентами** осуществляет регистрацию клиентов, ведёт контроль посещаемости и оплат, а также взаимодействует с клиентами.
* **Менеджер по работе с персоналом** выполняет задачи кадрового учёта: прием и увольнение тренеров, их распределение по группам.
* **Руководитель** управляет расписаниями, анализирует популярность услуг и просматривает отчётность по финансовым и организационным показателям.

.

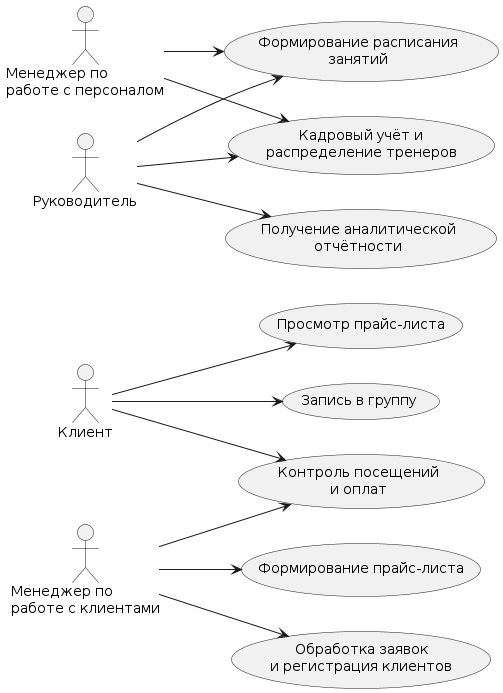


Рисунок 1.1 – Обобщённая UML-диаграмма вариантов использования

# **2 Стадия “Эскизный проект”**

В соответствии с обобщенной диаграммой вариантов использования (рисунок 1.1) производится функциональная декомпозиция проектируемой системы. Результаты декомпозиции представляются диаграммой пакетов системы (рисунок 2.2), отражающей ее программную архитектуру, и диаграммой вариантов использования (рисунок 2.3), детализирующей процессы функционирования отдельных ее подсистем.

# **2.1 Разработка программной архитектуры системы**

Изображение выглядит как текст, диаграмма, План, Параллельный

Автоматически созданное описание

Рисунок 2.2 – UML-Диаграмма пакетов

**Хранилище данных** включает множество баз данных, управляемых единым сервером и хранящий в себе данные сотрудников, клиентов компании, администраторов, классификатора услуг, истории оказания услуг клиентам.

Функции подсистемы «**Администрирование**»:

* Управление доступом
* Чат с клиентом

Функции подсистемы «**Управление работами**»:

* Управление заявками
* Распределение работ

Функции подсистемы «**Подготовка аналитической отчётности**»:

* Учёт количества клиентов
* Анализ оплаты услуг клиентов

Функции подсистемы «**Управление прайс-листом**»:

* Поиск по прайс-листу
* Добавление услуг
* Формирование тарифов

Функции подсистемы «**Управление контактами**»:

* Оформление документов
* Управление заявками

# **2.2 Разработка диаграмм вариантов использования компонентов**

В качестве примеров детализации процессов функционирования подсистем ниже приведена диаграмма вариантов использования для компонентов, содержащихся в пакете “Управление работами”

Изображение выглядит как диаграмма, рисунок, текст, линия

Автоматически созданное описание

Рисунок 2.3 – UseCase – диаграмма подсистемы “Управление работами”

Таблица 2.1 – Сценарий вариантов использования “Управление заявками”

|  |  |
| --- | --- |
| **Главный раздел** | |
| Наименование | Управление заявками |
| Тип | Базовый |
| Акторы | Клиент |
| Цель | Реализация процесса управление заявками |
| Краткое описание | Клиент проходит авторизацию на сайте, затем переходит на главную страницу. После этого пользователь ищет подходящие услуги или тарифы, рассматривает условия, стоимость. Далее клиент начинает оформлять заявку, которую в последствии одобрят или отклонят. Также клиент имеет возможность начать чат с менеджером. |
| Связанные варианты использования | Расширяющие: “Чат клиента с менеджером”  Использующие: “Регистрация заявки”, “Отклонение заявки”, “Одобрение заявки” |
| Раздел “**Типичный ход событий**” | |
| Действия актеров | Отклик системы |
| 1 Активизирует сайт.  3 Вводит данные (вход/регистрация).  Исключение 1: отказ в доступе.  6 Выбирает услугу/тариф  8 Рассматривает условия, стоимость и варианты услуг  10 Оставляет заявку  Исключение 2: Отклонение заявки и “Завершение сеанса” | 2 Запрашивает данные клиента.  4 Выполняет процедуру аутентификации.  5 Предлагает меню услуг и тарифов  7 Предоставляет сведения об услугах  9 Переходит на страницу оформления заявки  11 Одобряет заявку |
| Раздел “**Исключения**” | |
| Действия актеров | Отклик системы |
| Исключение 1: отказ в доступе. | |
|  | 12 Повторно запрашивает данные клиента |
| Исключение 2: Отклонение заявки и “Завершение сеанса” | |
| 15 Отказывается от услуги, отменяет заявку | 16 Отклоняет заявку  17 Закрывает заявку |

# **3 Стадия “Технический проект”**

На стадии технического проекта проводится объектная декомпозиция системы, результаты которой предоставляются в виде UML-диаграммы.

# **3.1 Разработка диаграмм концептуальных классов**

На рисунке 3.4 приведена диаграмма концептуальных (пассивных) классов для пакета подсистемы классификатора услуг (рисунок 2.2). Имена таких классов называются “Entity” (сущность), у каждого из них имеется раздел атрибутов и отсутствует раздел действий. Такие диаграммы могут использоваться для реализации схем баз данных.

Диаграмма концептуальных классов пакета “Классификатор услуг” представляет структурную модель, которая описывает свойства услуг в нашей системе.

Изображение выглядит как текст, число, диаграмма, снимок экрана

Автоматически созданное описание

Рисунок 3.4 – Диаграмма концептуальных классов для пакета “Классификатор услуг”

Объекты класса “Услуга” отношениями ассоциации с объектами “Параметры услуг”, “Категории услуг”, “Тарифы”. Таким образом, каждая услуга принадлежит определённой категории и имеет множество параметров, которые уточняют её характеристики.

Категории услуг включают в себя множество услуг, каждая услуга принадлежит одной категории, что упрощает поиск по категориям. Услуги имеют свои специфические параметры. Каждая услуга может иметь несколько таких параметров.

# **3.2 Разработка диаграмм программных классов**

На рисунке 3.5 представлена диаграмма классов подсистемы “История обработки заявок”, разработанная в соответствие с диаграммой пакетов (рисунок 2.2). В отличие от рассмотренных ранее диаграмм концептуальных классов, эта диаграмма содержит программные классы, в которых есть не только атрибуты, но и действия – операции классов.

Изображение выглядит как текст, диаграмма, Параллельный, число

Автоматически созданное описание

Рисунок 3.5 – Диаграмма программных классов подсистемы “История обработки заявок”

Объект класса “Клиент” представляет собой множество клиентов. Атрибуты данного класса определяют его персональные данные и ID. Методы позволяют вводить данные или подавать заявку.

Объект класса “Менеджер” представляет собой множество менеджеров. Общедоступные атрибуты этого класса определяют его персональные данные, ID и должность. Методы позволяют открывать, закрывать заявку.

Объект класса “Заявки” представляет собой множество заявок. Атрибуты данного класса определяют ID, тип, статус и дату подачи заявки. Методы позволяют создавать и редактировать заявку.

Объект класса “Чат клиента с менеджером” представляет собой чат. Общедоступные атрибуты этого класса определяют клиента, менеджера, статус, дату создания и закрытия чата, ID чата. Методы позволяют создать, открыть чат. Добавить собеседника, обсудить, закрыть чат.

# **3.3 Разработка диаграмм состояний**

Рассмотренные выше диаграммы классов представляют статические модели компонентов проектируемой системы и отражают её объектную декомпозицию. Результаты статического моделирования, полученные на этой стадии проекта, позволяют приступить к более детальному проектированию системы.

В качестве примера реализации динамической модели логического уровня, приведены упрощённые диаграммы состояний (рисунок 3.6) для пакета “История обработки заявок”, разработанные в соответствии с диаграммой классов этого пакета (рисунок 3.5).

Изображение выглядит как диаграмма, текст, План, Технический чертеж

Автоматически созданное описание

Рисунок 3.6 – Диаграмма состояний пакета “История обработки заявок”

Первое из целевых состояний “Регистрация” выполняется входное (entry) действие – полное имя, возраст, номер телефона/email. После этого (exit) действие, которое завершает ввод данных. Из этого состояния осуществляется переход в состояние “Проверка на подлинность”.

В состояние “Проверка на подлинность” выполняется (entry) действие обработки данных. Последующее (do) действие производит проверку введённых данных. После этого (exit) действие подтверждает вход пользователя, либо отказывает ему. В случае, если пользователю отказали при входе, он переходит в состояние “Регистрация”, если система подтверждает вход, то он переходит в состояние “Выбор услуги/Подача заявки” или в состояние “Чат клиента с менеджером”.

В состоянии “Выбор услуги/Подача заявки” выполняется (do) действие – получение прайс-листа, выбор услуги/тарифа, оставление заявки, оставить/отменить заявку. В случае, если у пользователя остались вопросы он переходит в состояние “Чат клиента с менеджером”, если у пользователя нет вопросов касательно услуг, он оставляет заявку и переходит в состояние “Статус заявки” или покидает сайт и закрывает его.

В состоянии “Чат с менеджером” выполняется (entry) действие – вопрос и последующие (do) действия – открытие чата, решение вопроса. После этого выполняется (exit) действие – закрытие чата. В случае, если клиент передумал приобретать услуги, он выходит из системы. Если же он обсудил вопрос и намерен продолжить, он переходит обратно в состояние “Выбор услуги/Подача заявки”

В состоянии “Статус заявки” выполняется (entry) действие – показать текущий статус и последующее (do) действие – изменить статус заявки. В случае, если заявка не обработана или клиент решил оставить новую заявку, происходит переход в состояние “Выбор услуги/Подача заявки”, в противном случае, происходит переход в состояние “Отчёт о заявках”.

В состоянии “Отчёт о заявка” выполняется (do) действие – открытие истории заявок, добавление заявки, формирование отчёта, добавление в список аналитического отчёта, последующее (exit) действие – закрытие аналитического отчёта. В случае, если менеджер желает дополнить отчёт, он переходит обратно в состояние “Отчёт о заявках”, в противном случае, происходит завершение сеанса обработки заявки.

# **4 Обзор результатов программного проекта**

Подводя итоги выполнения трёх стадий программного проекта, перечислим основные результаты.

На стадии технического задания были определены назначение и область применения проектируемой системы, был разработан терминологический словарь, определён состав пользователей и разработана обобщённая UML-диаграмма вариантов использования.

На стадии эскизного проекта были разработаны модели проектируемой системы. Была проведена её декомпозиция, результаты которой были представлены в диаграмме пакетов, отражающие её программную архитектуру. Сценарии вариантов использования детализировали процессы функционирования отдельных систем.

На стадии технического проекта были разработаны структурные и динамические модели логического уровня, детализирующие концептуальное представление о системе. Были разработаны диаграммы концептуальных классов для компонентов системы, классификатора услуг. Также были разработаны диаграммы программных классов для компонентов системы истории обработки заявок. Для компонентов разработаны диаграммы состояний, отражающие реакцию системы на внутренние и внешние события.

Из всего вышеперечисленного можно сделать вывод, что при работе над проектом был получен набор артефактов проекта, включающий в себя UML-диаграммы и их подробные описания. Данный набор подходит для программной реализации компонентов проектируемой системы на завершающей стадии проекта.