Министерство образования Республики Беларусь Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

КАФЕДРА ИНФОРМАТИКИ

Отчёт по лабораторной работе №3

По теме “Проектирование и разработка ПО на языке UML”

Выполнила: студентка гр. 053501 Шурко Т.А.

Проверил: ассистент кафедры информатики Гриценко Н. Ю.

Минск 2023

**СОДЕРЖАНИЕ**

[1 Цель работы 3](#_Toc147680381)

[2 Диаграмма обзора 4](#_Toc147680382)

[2.1 Теоретические сведения 4](#_Toc147680383)

[2.2 Построение диаграммы 4](#_Toc147680384)

[3 Диаграмма вариантов использования 6](#_Toc147680385)

[3.1 Теоретические сведения 6](#_Toc147680386)

[3.2 Построение диаграммы 6](#_Toc147680387)

[4 Диаграмма последовательностей 8](#_Toc147680388)

[4.1 Теоретические сведения 8](#_Toc147680389)

[4.2 Построение диаграммы 8](#_Toc147680390)

[5 Диаграмма активности 11](#_Toc147680391)

[5.1 Теоретические сведения 11](#_Toc147680392)

[5.2 Построение диаграммы 11](#_Toc147680393)

[6 Диаграмма состояний 12](#_Toc147680394)

[6.1 Теоретические сведения 12](#_Toc147680395)

[6.2 Построение диаграммы 12](#_Toc147680396)

[7 Диаграмма классов 14](#_Toc147680397)

[7.1 Теоретические сведения 14](#_Toc147680398)

[7.2 Построение диаграммы 14](#_Toc147680399)

[Заключение 15](#_Toc147680400)

[Список использованных источников 16](#_Toc147680401)

# **1 ЦЕЛЬ РАБОТЫ**

Целью лабораторной работы №3 является разработать диаграмму обзора (Overview Diagram), диаграмму вариантов использования (Use Case Diagram), диаграмму последовательностей (Sequence Diagram), диаграмму активности (Activity Diagram), диаграмму состояний (State Diagram), диаграмму классов (Class Diagram) и подготовить отчет о проделанной работе.

# **2 ДИАГРАММА ОБЗОРА**

## **2.1 Теоретические сведения**

Диаграмма обзора (Interaction overview diagram) — диаграмма, которая предназначена для представления взаимодействия только в контексте потока управления в некоторой агрегированной форме. Диаграммы обзора взаимодействия, вместо узлов действий и объектов диаграмм деятельности, имеют фреймы, каждый из которых может соответствовать взаимодействию или использованию взаимодействия. Альтернативные комбинированные фрагменты представляются узлом решения и соответствующим узлом слияния. Параллельные комбинированные фрагменты представляются узлом разделения и соответствующим узлом соединения. Комбинированные фрагменты типа Цикл представляются простыми циклами. Ветвление и слияния ветвлений на диаграммах обзора взаимодействия должны быть должным образом вложенными. Диаграммы обзора взаимодействия заключаются во фрейм, аналогично другим видам диаграмм взаимодействия с тегом sd и ref [1].

## **2.2 Построение диаграммы**

На рисунке 2.1 изображена диаграмма обзора. Левая часть диаграммы показывает начальное взаимодействие пользователей с программным продуктом: необходимо зарегистрироваться или войти, если аккаунт существует.

После ввода валидных данных и успешной аутентификации пользователь получает доступ к основной функциональности приложения: может изменить профайл, записать тренировку, изменить настройки системы, просматривать тренировки других пользователей, создать/редактировать группу или уведомить участников группы, а также выйти из приложения.

При просмотре тренировок можно оставить комментарий или отметить понравившуюся.

После записи тренировки можно опубликовать или просмотреть статистику.

При создании/редактировании группы добавлять или удалять участников.

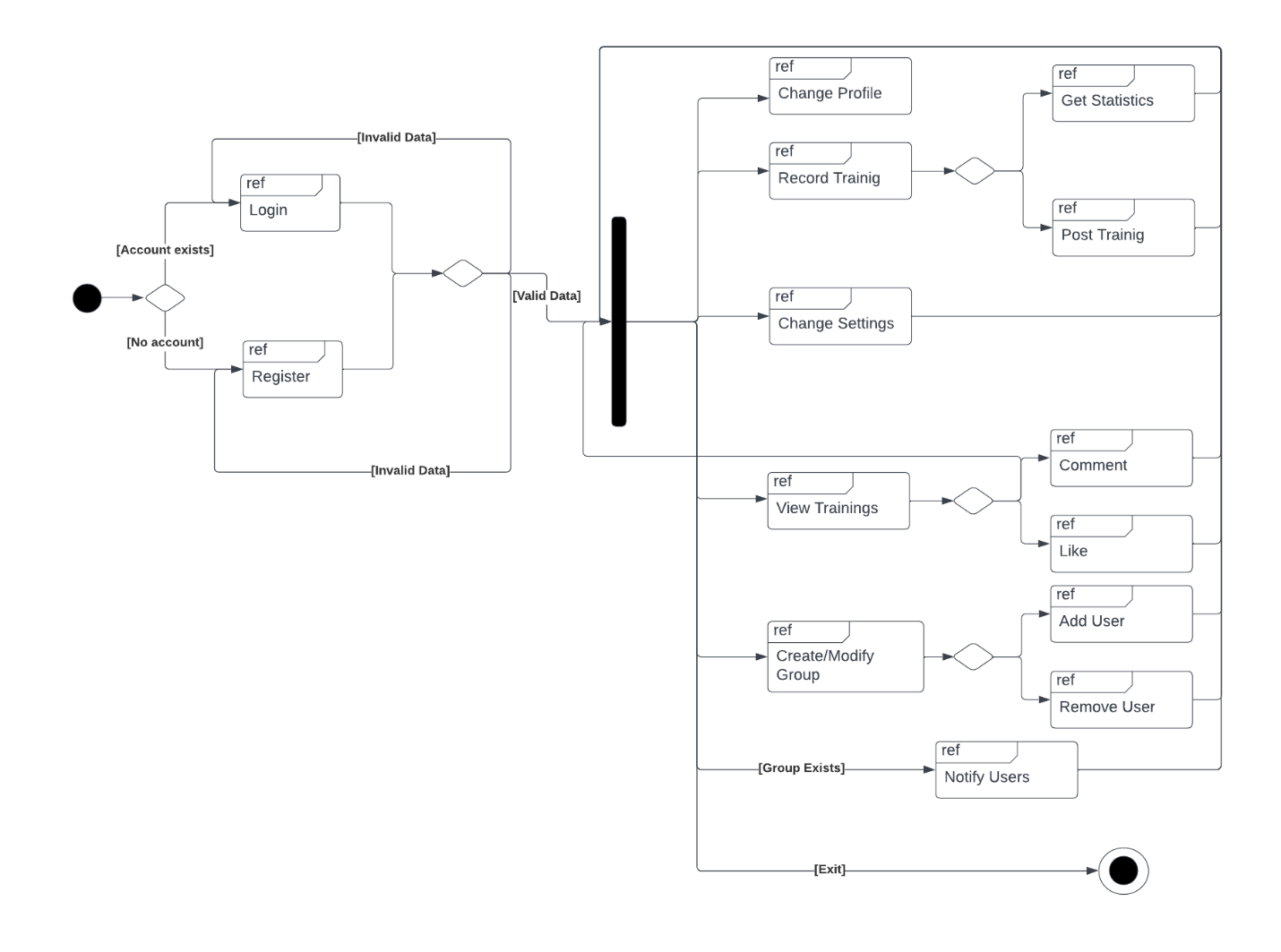


Рисунок 2.1 — Диаграмма обзора (Interaction overview diagram)

# **3 ДИАГРАММА ВАРИАНТОВ ИСПОЛЬЗОВАНИЯ**

## **3.1 Теоретические сведения**

Диаграмма вариантов использования (Use Case Diagram) в [UML](https://ru.wikipedia.org/wiki/UML) — диаграмма, отражающая отношения между [акторами](https://ru.wikipedia.org/wiki/%D0%90%D0%BA%D1%82%D0%BE%D1%80_(UML)) и [прецедентами](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B5%D1%86%D0%B5%D0%B4%D0%B5%D0%BD%D1%82_(UML)) и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне.

Прецедент — возможность моделируемой системы (часть её функциональности), благодаря которой пользователь может получить конкретный, измеримый и нужный ему результат. Прецедент соответствует отдельному сервису системы, определяет один из вариантов её использования и описывает типичный способ взаимодействия пользователя с системой. Варианты использования обычно применяются для спецификации внешних [требований](https://ru.wikipedia.org/wiki/%D0%A2%D1%80%D0%B5%D0%B1%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F_%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%BC%D1%83_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D1%8E) к системе.

Актор — стилизованный человечек, обозначающий набор ролей пользователя (понимается в широком смысле: человек, внешняя сущность, класс, другая система), взаимодействующий с некоторой сущностью (системой, подсистемой, классом). Акторы не могут быть связаны друг с другом (за исключением отношений обобщения/наследования).

Основное назначение диаграммы — описание функциональности и поведения, позволяющее [заказчику](https://ru.wikipedia.org/wiki/%D0%97%D0%B0%D0%BA%D0%B0%D0%B7%D1%87%D0%B8%D0%BA), [конечному пользователю](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BD%D0%B5%D1%87%D0%BD%D1%8B%D0%B9_%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8C) и [разработчику](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%81%D1%82) совместно обсуждать проектируемую или существующую [систему](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0_(%D0%BA%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80)).

При моделировании прецедентов [системный аналитик](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%BD%D1%8B%D0%B9_%D0%B0%D0%BD%D0%B0%D0%BB%D0%B8%D1%82%D0%B8%D0%BA) стремится: чётко отделить систему от её окружения, а также определить язык или система программирования) при составлении модели прецедентов опускаются (для них составляется другой документ) [2].

## **3.2 Построение диаграммы**

На рисунке 3.1 изображена диаграмма вариантов использования, на которой изображено 2 основных актора: авторизированный и неавторизированный пользователь. Для неавторизированного пользователя доступен только вход или авторизация. Для авторизированного вся функциональность приложения.

При создании группы, есть возможность добавить другого авторизованного пользователя.

При редактировании группы добавить или удалить авторизированного пользователя.

При клике уведомить группу о местоположении, приходят сообщения всем текущим участникам группы.

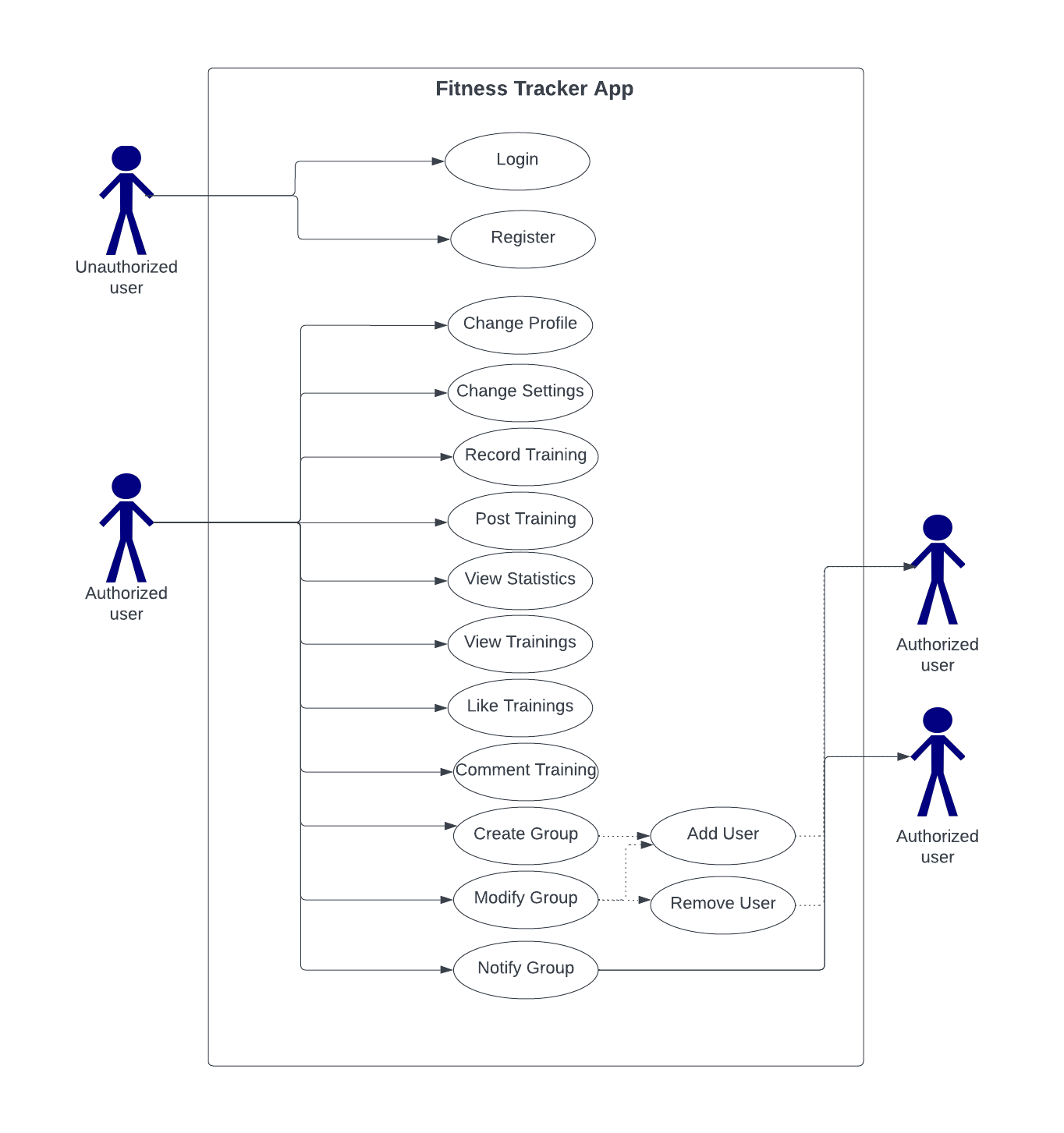


Рисунок 3.1 — Диаграмма вариантов использования (Use Case diagram)

# **4 ДИАГРАММА ПОСЛЕДОВАТЕЛЬНОСТЕЙ**

## **4.1 Теоретические сведения**

Диаграмма последовательности (Sequence Diagram) — [UML-диаграмма](https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0_(UML)), на которой для некоторого набора объектов на единой временной оси показан жизненный цикл объекта (создание-деятельность-уничтожение некой сущности) и взаимодействие акторов (действующих лиц) информационной системы в рамках [прецедента](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B5%D1%86%D0%B5%D0%B4%D0%B5%D0%BD%D1%82_(UML)).

Основными элементами диаграммы последовательности являются обозначения [объектов](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)) (прямоугольники с названиями объектов), вертикальные «линии жизни», отображающие течение времени, прямоугольники, отражающие деятельность объекта или исполнение им определенной функции (прямоугольники на пунктирной «линии жизни» — фокусы контроля , и стрелки, показывающие обмен сигналами или сообщениями между объектами [3].

## **4.2 Построение диаграммы**

На рисунке 4.1 изображена диаграмма последовательности для регистрации пользователей в приложении. Пользователь вводит данные на экран регистрации и подтверждает ввод. Далее поля проверяются на валидность и отправляется запрос на создание пользователя в базу данных. После успешного создания происходит аутентификация пользователя.

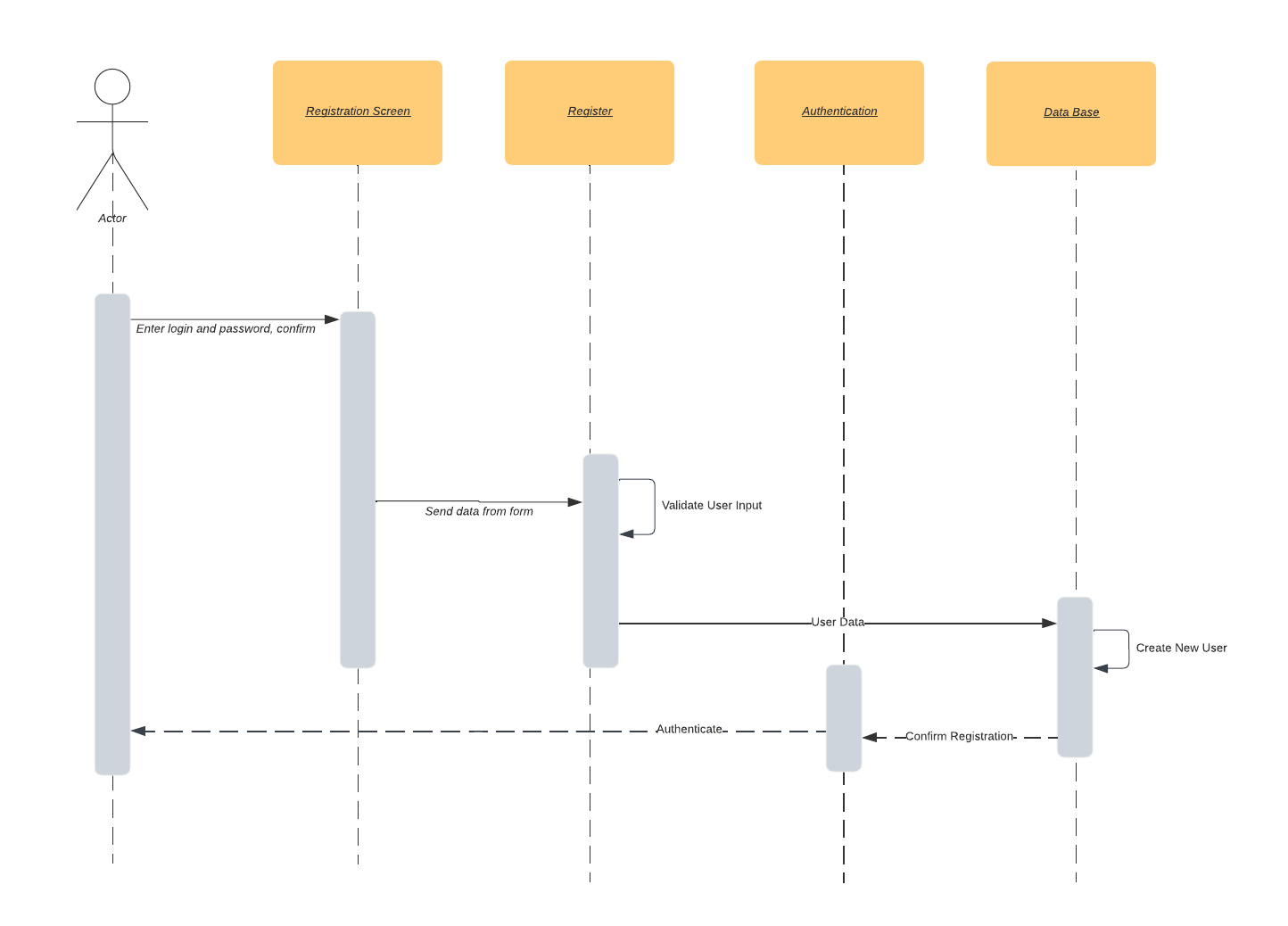


Рисунок 4.1 — Диаграмма последовательности для регистрации пользователей (Sequence Diagram)

На рисунке 4.2 изображена диаграмма последовательности для входа пользователя в приложение. Пользователь вводит данные на экран авторизации и подтверждает ввод. Далее поля проверяются на валидность и отправляется запрос на проверку существования пользователя в базе данных. После успешного ответа от базы данных, происходит аутентификация пользователя.

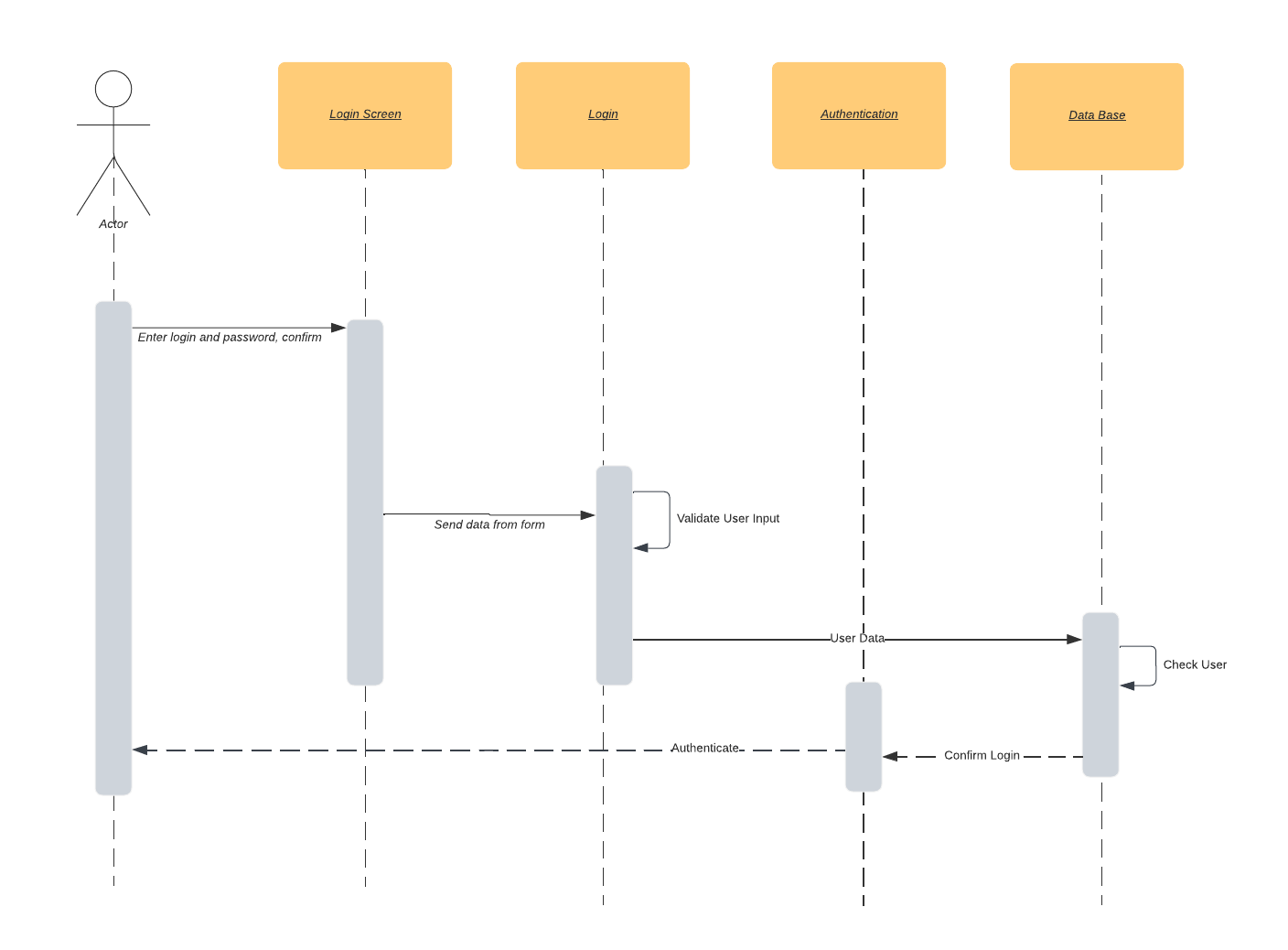


Рисунок 4.2 — Диаграмма последовательности для входа пользователя (Sequence Diagram)

На рисунке 4.3 изображена диаграмма последовательности для создания тренировки пользователем и получения статистики. Пользователь начинает тренировку. Далее асинхронно обрабатываются данные и возвращается статистика при прохождении определенного этапа дистанции.

По окончании тренировки пользователь может просмотреть данные по всем таким этапам, а также узнать какие пользователи пробегали данную дистанцию лучше или хуже него.

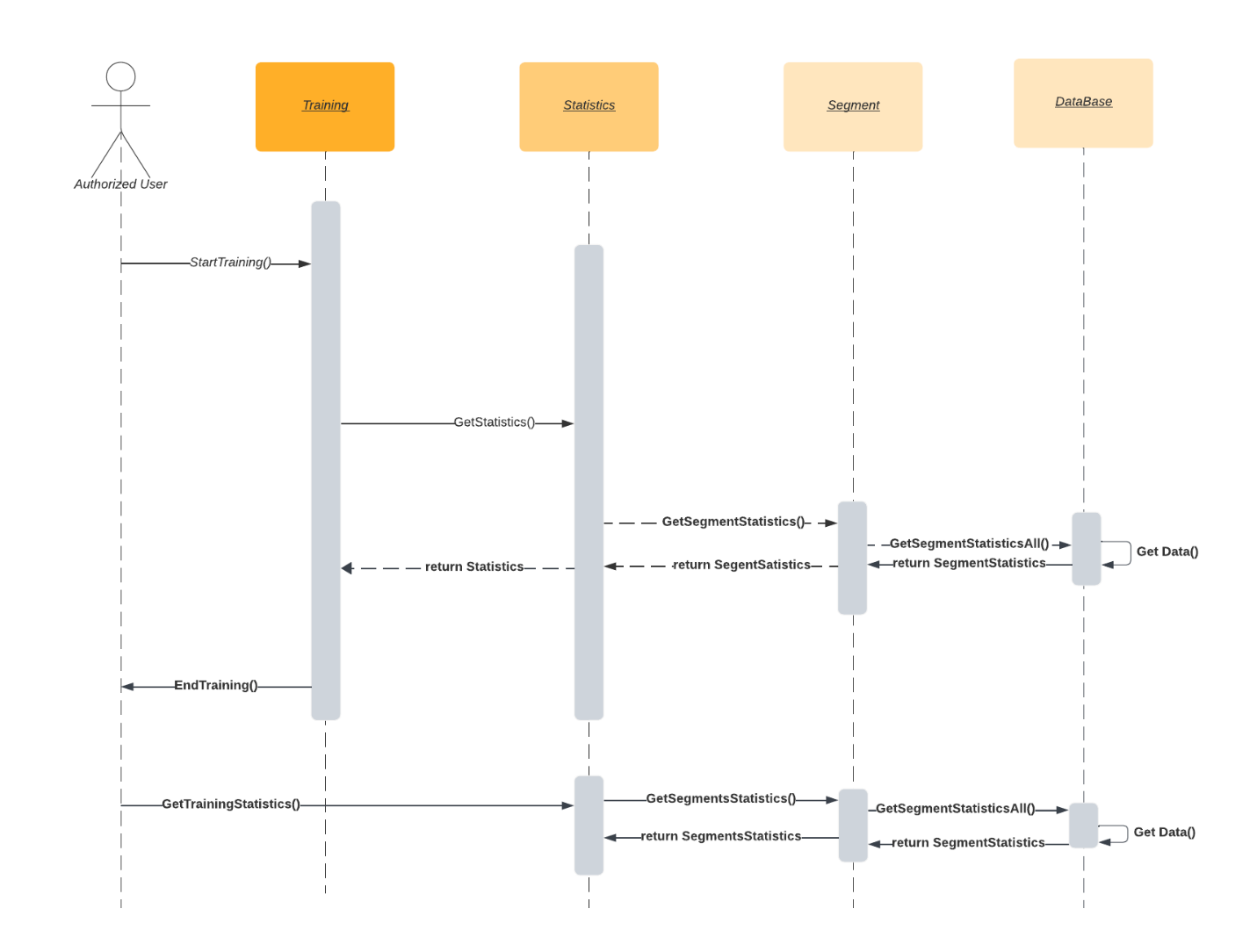


Рисунок 4.3 — Диаграмма последовательности для создания тренировки пользователем и получения статистики (Sequence Diagram)

# **5 ДИАГРАММА АКТИВНОСТИ**

## **5.1 Теоретические сведения**

Диаграмма деятельности (Activity Diagram) — [UML](https://ru.wikipedia.org/wiki/UML)-диаграмма, на которой показаны действия, состояния которых описаны на [диаграммах состояний](https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0_%D1%81%D0%BE%D1%81%D1%82%D0%BE%D1%8F%D0%BD%D0%B8%D0%B9_(UML)). Под деятельностью понимается спецификация исполняемого поведения в виде координированного последовательного и параллельного выполнения подчинённых элементов — вложенных видов деятельности и отдельных действий, соединённых между собой потоками, которые идут от выходов одного узла ко входам другого [4].

Диаграммы деятельности используются при моделировании [бизнес-процессов](https://ru.wikipedia.org/wiki/%D0%91%D0%B8%D0%B7%D0%BD%D0%B5%D1%81-%D0%BF%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81), технологических процессов, последовательных и параллельных вычислений.

## **5.2 Построение диаграммы**

На рисунке 5.1 изображена диаграмма активности, которая разделена на 2 части: первая относится к неавторизованному пользователю, вторая к авторизованному, в которой отображен весь функционал приложения.

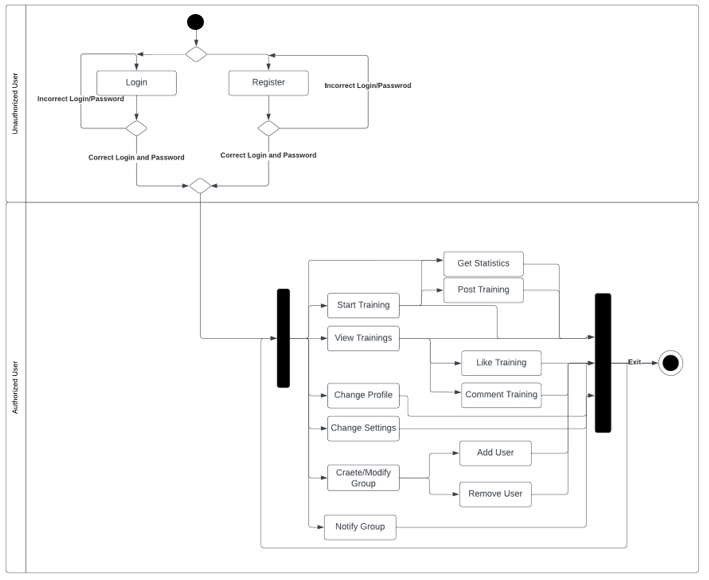


Рисунок 5.1 — Диаграмма активности (Activity Diagram)

# **6 ДИАГРАММА СОСТОЯНИЙ**

## **6.1 Теоретические сведения**

Главное предназначение диаграммы состояний (State Diagram) — описать возможные последовательности состояний и переходов, которые в совокупности характеризуют поведение элемента модели в течение его жизненного цикла. Диаграмма состояний представляет динамическое поведение сущностей, на основе спецификации их реакции на восприятие некоторых конкретных событий.

Диаграмма состояний описывают динамическое представление объекта. Они особенно важны для моделирования поведения интерфейсов, классов или коопераций и подчеркивают событийно-зависимое поведение объекта [5].

## **6.2 Построение диаграммы**

На рисунке 6.1 изображена диаграмма состояний приложения фитнес-трекер, которая позволяет описать основные состояния приложения: регистрация или вход, выбор в меню, изменение профайла или настроек системы, создание или редактирование группы, просмотр или создание тренировок, рассылка уведомлений участникам группы.

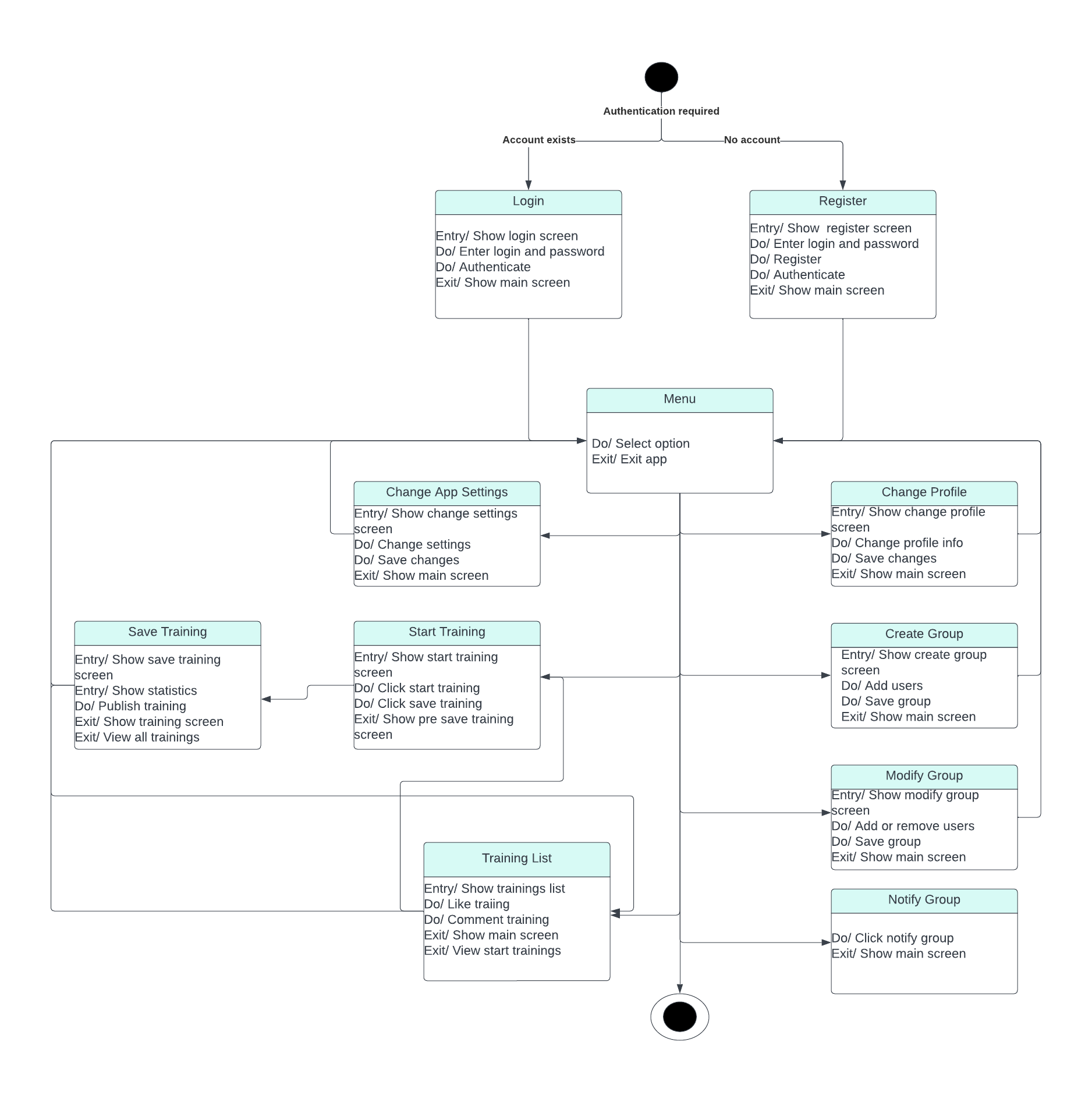


Рисунок 6.1 — Диаграмма состояний (State Diagram)

# **7 ДИАГРАММА КЛАССОВ**

## **7.1 Теоретические сведения**

Диаграмма классов (Сlass Diagram) — структурная [диаграмма](https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0_(UML)) языка моделирования [UML](https://ru.wikipedia.org/wiki/UML), демонстрирующая общую структуру иерархии [классов](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B0%D1%81%D1%81_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)) системы, их коопераций, [атрибутов](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%BB%D0%B5_%D0%BA%D0%BB%D0%B0%D1%81%D1%81%D0%B0) (полей), [методов](https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4_(%D1%8F%D0%B7%D1%8B%D0%BA%D0%B8_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F)), интерфейсов и взаимосвязей (отношений) между ними. Широко применяется не только для документирования и визуализации, но также для конструирования посредством прямого или обратного проектирования.

Целью создания диаграммы классов является графическое представление статической структуры декларативных элементов системы. Она содержит в себе также некоторые элементы поведения (например — операции), однако их динамика должна быть отражена на диаграммах других видов ([диаграммах коммуникации](https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0_%D0%BA%D0%BE%D0%BC%D0%BC%D1%83%D0%BD%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D0%B8), диаграммах состояний). Для удобства восприятия диаграмму классов можно также дополнить представлением [пакетов](https://ru.wikipedia.org/wiki/%D0%9F%D0%B0%D0%BA%D0%B5%D1%82_(UML)), включая вложенные [6].

## **7.2 Построение диаграммы**

На рисунке 7.1 диаграмма классов, на которой отображены основные сущности: абстрактный класс «User»; наследуемые от него «AuthenticatedUser» и «UnauthenticatedUser»; класс «Group», который отображает связь группы и авторизированных пользователей; далее класс «Training», отображающий сущность тренировка; класс «Segment» — сегменты, на которые разбивается тренировка; класс «ListTrainings» для работы со списком тренировок.

Также пользователь при авторизации указывает страну и подписку из существующих в перечислении.

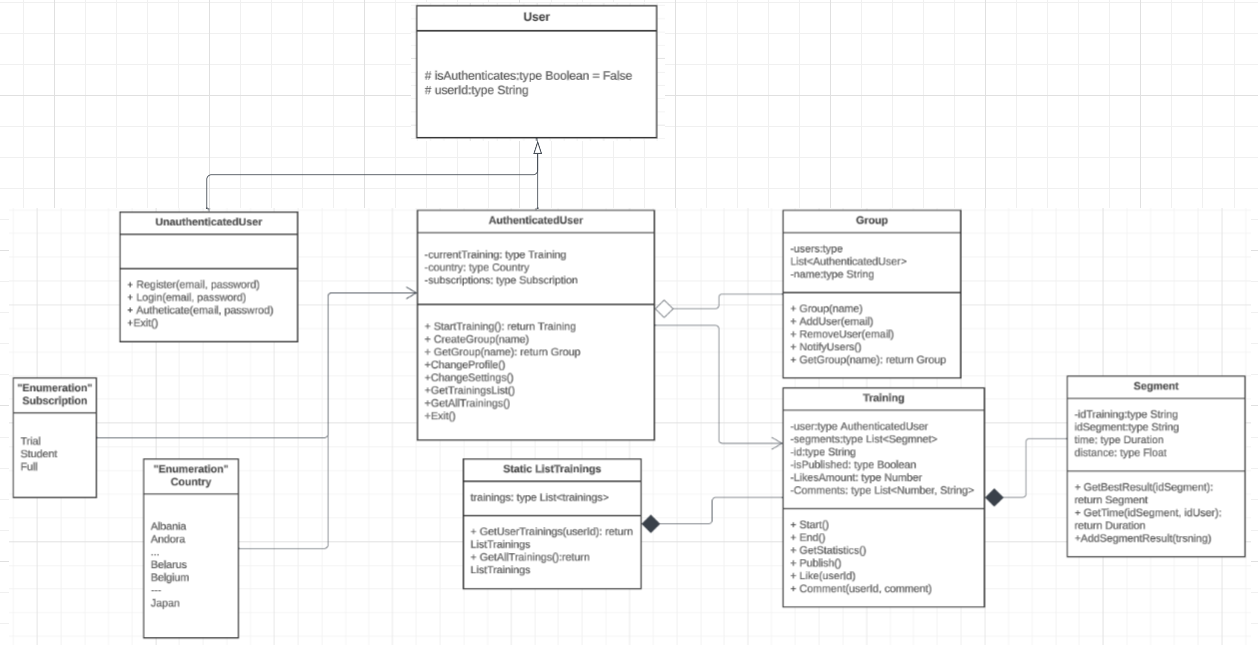


Рисунок 7.1 — Диаграмма классов (Сlass Diagram)

# **ЗАКЛЮЧЕНИЕ**

В результате выполнения лабораторной работы были изучены и разработаны: диаграмма обзора (Overview Diagram), диаграмма вариантов использования (Use Case Diagram), диаграмма последовательностей (Sequence Diagram), диаграмма активности (Activity Diagram), диаграмма состояний (State Diagram), диаграмма классов (Class Diagram). А также подготовлен отчет о проделанной работе.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Studbooks [Электронный ресурс]: Studbooks — Overview diagram (Диаграмма обзора взаимодействия). Режим доступа: https://studbooks.net/2263323/informatika/overview\_diagram\_diagramma\_obzora\_vzaimodeystviya (дата обращения: 07.10.2023).
2. Википедия [Электронный ресурс]: Википедия — Диаграмма прецедентов. Режим доступа: <https://en.wikipedia.org/wiki/Диаграмма_прецедентов> (дата обращения: 07.10.2023).
3. Википедия [Электронный ресурс]: Википедия — Диаграмма последовательности. Режим доступа: <https://en.wikipedia.org/wiki/Диаграмма_последовательности> (дата обращения: 08.10.2023).
4. Википедия [Электронный ресурс]: Википедия — Диаграмма активности. Режим доступа: <https://en.wikipedia.org/wiki/Диаграмма_активности> (дата обращения: 08.10.2023).
5. Википедия [Электронный ресурс]: Википедия — Диаграмма состояний. Режим доступа: <https://en.wikipedia.org/wiki/Диаграмма_состояний> (дата обращения: 08.10.2023).
6. Википедия [Электронный ресурс]: Википедия — Диаграмма классов. Режим доступа: <https://en.wikipedia.org/wiki/Диаграмма_классов> (дата обращения: 08.10.2023).