

**Министерство науки и высшего образования Российской Федерации**  
федеральное государственное автономное образовательное учреждение высшего образования  
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**

**Отчет**

по лабораторной работе №5 «Разработка поведенческих UML-диаграмм»

по дисциплине **«Проектирование информационных систем»**

Автор: Ковалев Г. П.

Факультет: ПИН

Группа: К3341

Преподаватель: Духанов А.В.

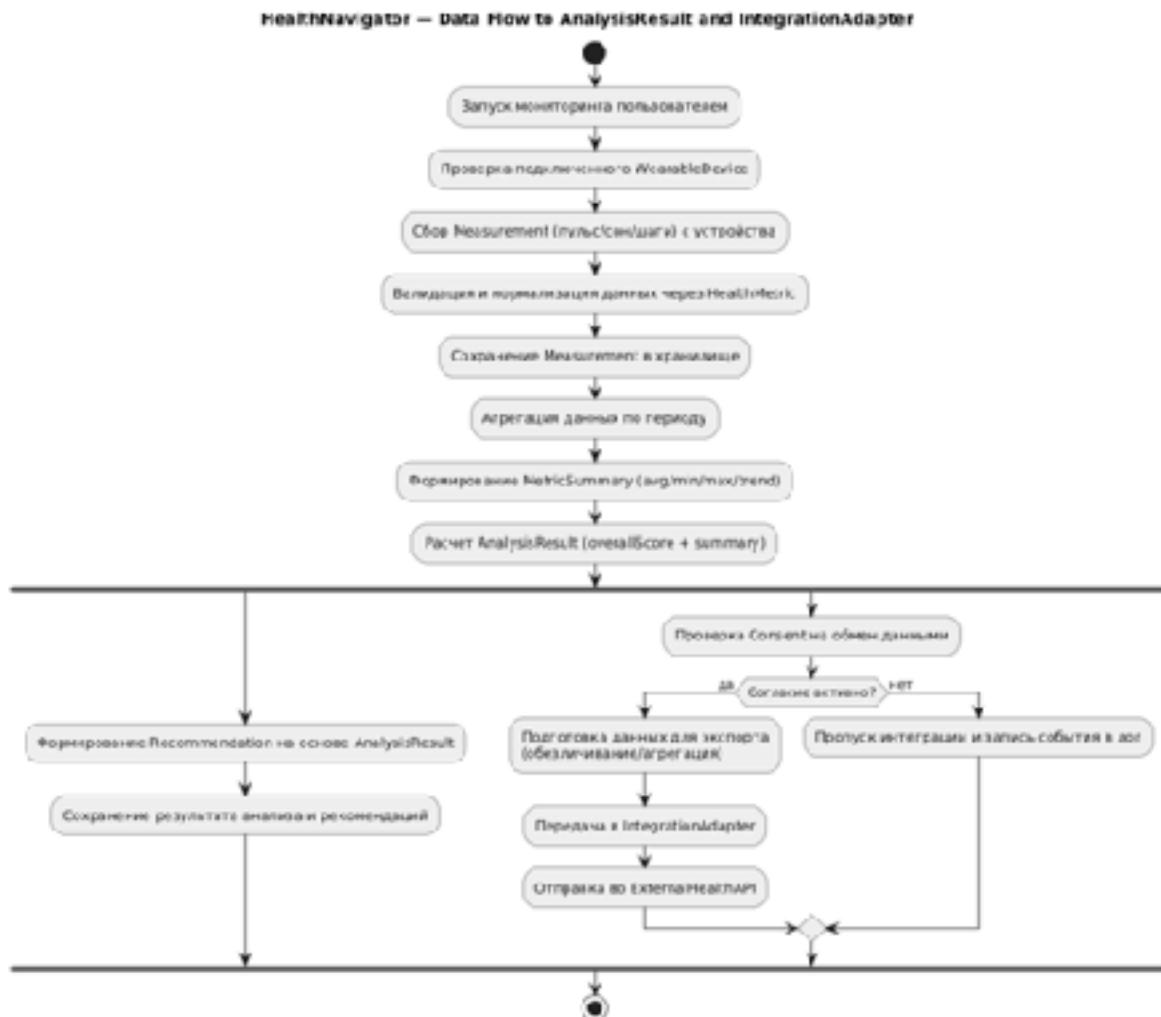
**ИТМО**

Санкт-Петербург 2025

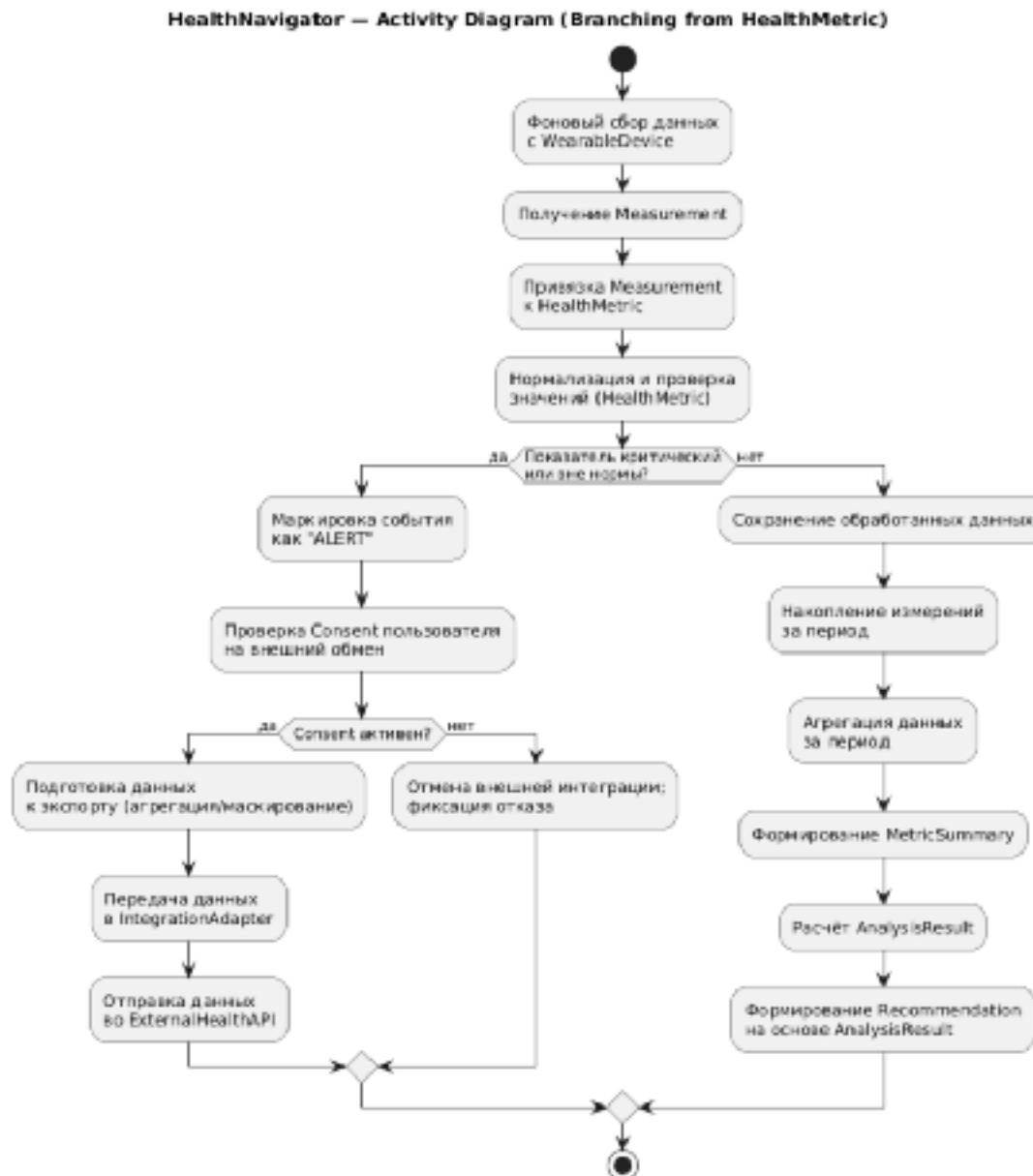
Я сделал промпт в ChatGPT, в тот же чат, что я использовал в четвертой лабораторной работе. Так как он помнит классы, которые использовались в ней и помнит презентацию с правилами для написания диаграммы активности. Вот мой промпт:

Теперь мы будем писать новый код для PlantUML, мы будем делать диаграмму активностей. Должно быть минимум 6 состояний. Давай напишем диаграмму активностей для нашего Health приложения. Смотри, давай рассмотрим как у нас идут данные с самого начала до AnalysisResult и IntegrationAdapter, мы пройдем сквозь несколько состояний и выйдем с данными в данных точках

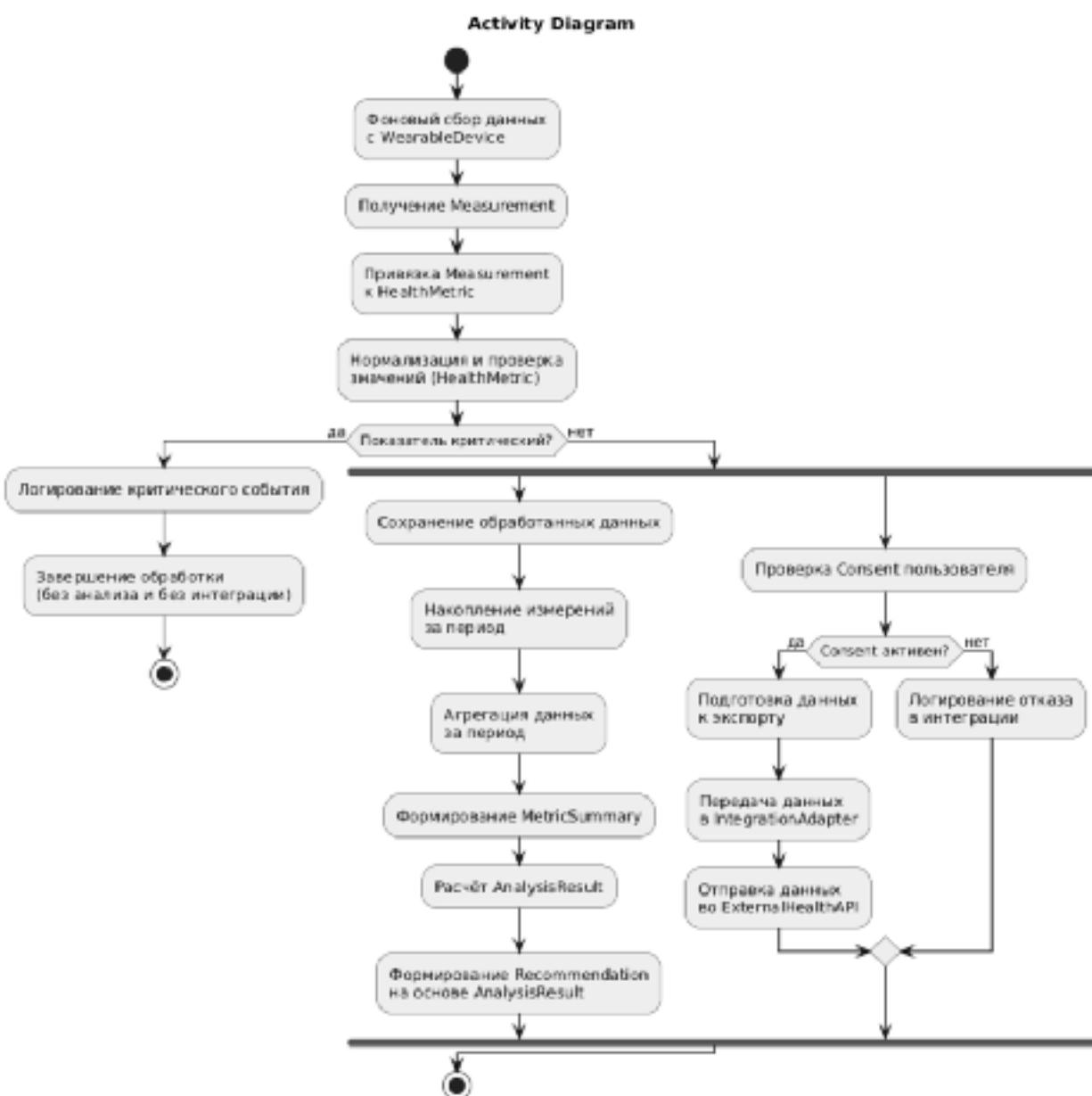
Вот пример PlantUML-диаграммы активностей для HealthNavigator, где показан путь данных от начала (сбор) до AnalysisResult и до IntegrationAdapter, с разветвкой по согласию (Consent). Здесь больше 6 состояний и есть две конечные "точки выхода" по данным.



ChatGPT выдал неплохую основу, но с ней необходимо поработать лично, ведь тут есть ошибки в логике. Пользователь не запускает мониторинг, ветвление идет от `AnalysisResult`. Данные логические ошибки нужно исправить первыми. После данных изменений GPT выдал мне данную диаграмму.



Она тоже неверная, на этот раз исправим руками. При проверке критических показателей данные не идут ни в анализ, ни в API. Они должны логироваться и все. Сохранение и отправление данных в API идут параллельно. После исправлений у меня получилась данная диаграмма активности.



После получения измерений данные нормализуются и проверяются с использованием логики метрик, где выполняется контроль критических значений. При обнаружении критического показателя процесс завершается после логирования события. В случае отсутствия критических отклонений процесс разделяется на две параллельные ветви: в первой выполняется сохранение и аналитическая обработка данных с формированием AnalysisResult и рекомендаций, а во второй — проверка согласия пользователя и, при его наличии, передача данных через IntegrationAdapter во внешнее API. Данная схема обеспечивает одновременную аналитическую обработку и интеграцию данных при строгом соблюдении условий безопасности и логической целостности процесса.

Данная работа является второй по PlantUML, во второй раз напишу, что с ней очень удобно и круто работать. Ранее я рисовал диаграммы в Figma и Mira. PlantUML сохраняет огромное количество времени.

@startuml

title Activity Diagram

start

:Фоновый сбор данных\nс WearableDevice;

:Получение Measurement;

:Привязка Measurement\nк HealthMetric;

:Нормализация и проверка\nзначений (HealthMetric);

if (Показатель критический?) then (да)

:Логирование критического события;

:Завершение обработки\n(без анализа и без интеграции);

stop

else (нет)

fork

:Сохранение обработанных данных;

:Накопление измерений\nза период;

:Агрегация данных\nза период;

:Формирование MetricSummary;

:Расчёт AnalysisResult;

:Формирование Recommendation\nна основе AnalysisResult;

fork again

:Проверка Consent пользователя;

if (Consent активен?) then (да)

:Подготовка данных\nк экспорту;

:Передача данных\nв IntegrationAdapter;

:Отправка данных\nв ExternalHealthAPI;

else (нет)

:Логирование отказа\nв интеграции;

endif

end fork

endif

stop

@enduml

XLJBRjf05DtFLymIVY4i6jHKQXQHgdJH3PiF38bLsqJs49KT4AMfb5IqgowgDAsK3o04X0Cn\_  
CATF-gvOqCW0NLXMPwvvpxEDUxiL6HQGQ-  
C9xnjInZNVgcPRSzYHHf08XN80GEtHBCqhn3EmbFKZReJkQsRQa8VLndjfBUru9RKT-  
q2fl7tsfq-zgYu1UGFKr9xP2neXCg6HFD0gRaOwqA5n0RQDuu7jqPWcr5-zheGqaM-  
ryitnWJQHL-  
NuQrQcHmXFWHjJttwXnFOa9PBck46IDC8RNC2F8TfVYZ\_8mYfTiII4LvS5boJY7LiNiLDx9M  
d2YU4pOJXAdZ31gw8qnpe41gKNXPGxz0lWUqHbqlsyoWrm3fjnySPqXuxnLQ\_aA7u2vhAsNv  
TjSb5jN72hkwnU2Glu6D5OEfLb0rpf-

KM2ON05zmpbpb1IP8tOD4w4zcEDp5CeTOi2f\_L6z0alJk4lLCwTNhejXUMer-  
PUoWckSSGrpf7Mae3pUtJZo0VWHeACv-umGVnmGV3T9OtKdqQwPg-  
jmDlS4Sz7a0tvVJC8LDHQ1Fy1UyTUs1-  
sAykXybkn5n\_gkEcR\_z4TwceZW2rhodYD4FcadCyCyqmpqstWumPBTQIyC2U\_MNUeVDq8Op  
YT<sub>w</sub>-8zkKAHZEtOlzBYaswZ4IRsJjCGkaBF-Oot1-  
s6oAPZwMUTNCYxJSmTLWPnklQiz8AF1lPqEtE2-  
X0xeT6unV1ZIZNr8dHqHFuDpJdpZORTGVOWJUVWK2picKilzlFqEl-NF1jsQziATpEkfGsMJ-  
d3MjUNRXtvdWXiYFXjc8N7\_W9\_WC0