

Министерство науки и высшего образования РФ
Федеральное государственное автономное образовательное учреждение
высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)
Кафедра автоматизированных систем управления (АСУ)

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ПРОСМОТРА ДАННЫХ С РАДАРА, С
ВОЗМОЖНОСТЬЮ ФИЛЬТРАЦИИ И КЛАССИФИКАЦИИ

Пояснительная записка к курсовому проекту
по дисциплине «Основы разработки программного обеспечения»

Студент гр. 431-3
_____ А.В. Гурулёв
«__» _____ 2024

Руководитель
доцент кафедры АСУ, к.т.н.
_____ А.К. Лукьянов
(оценка) «__» _____ 2024

Томск 2024

Министерство науки и высшего образования РФ
Федеральное государственное автономное образовательное учреждение
высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)
Кафедра автоматизированных систем управления (АСУ)

Задание к курсовому проекту
по дисциплине «Основы разработки программного обеспечения»

Студенту группы __431-3__ факультета систем управления

Гурулёву Александру Владиславовичу

1. Тема курсовой работы: ПО просмотра данных с радара, с возможностью фильтрации и классификации.
2. Срок сдачи: 28 декабря 2024 г.
3. Исходные данные: Записи данных с радара.
4. Содержание работы (перечень вопросов, подлежащих разработке):
 - 4.1. Описание среды разработки.
 - 4.2. Описание языка программирования.
 - 4.3. Описание проекта.
5. Требования:
 - 5.1. Программа должна обеспечить возможность редактирования данных.
 - 5.2. Программа должна обеспечить возможность провести разметку данных
6. Требования к оформлению работы: в соответствии с ОС ТУСУР 01-2021

Дата выдачи задания: 10 сентября 2024 г.

Руководитель:

доцент кафедры АСУ, к.т.н., Лукьянов А.К. _____

подпись

Задание принял к исполнению: _____

10.09.2024

Оглавление

Введение.....	5
1 АНАЛИЗ ТРЕБОВАНИЙ.....	6
1.1 Модель бизнес - процесса.....	6
1.1.1 Фильтрация трасс.....	6
1.1.2 Выбор трассы для отображения.....	6
1.1.3 Отрисовка трассы.....	6
1.1.4 Разметка трассы.....	7
1.2 Формулировка требований к системе.....	7
1.2.1 Требования к составу выполняемых функций.....	7
1.2.2 Организация входных и выходных данных.....	8
1.2.3 Временные характеристики.....	9
1.2.4 Требования к надёжности.....	9
1.2.5 Требования к информационной и программной совместимости.....	9
1.3 Сравнение системы с аналогами.....	10
2 ОПРЕДЕЛЕНИЕ СПЕЦИФИКАЦИЙ.....	11
2.1 Модель проектирования и разработки системы.....	11
2.2 Инструменты разработки.....	12
2.3 Входные и выходные данные.....	13
3 ПРОЕКТИРОВАНИЕ.....	14
3.1 Фильтрация трасс.....	14
3.2 Выбор трассы для отображения.....	14
3.3 Отрисовка трассы.....	14
3.4 Разметка трассы.....	15
4 КОДИРОВАНИЕ.....	16
5 ТЕСТИРОВАНИЕ.....	19
Заключение.....	21
Список использованных источников.....	22

Введение

В виду трудности при получении данных с радара точно определить является обнаруженный объект реальным или принятая информация является отражением от помех появилась потребность для разработки модели машинного обучения для бинарной классификации зафиксированных трасс. Для обучения модели данные необходимо разметить, для чего необходима программа, предоставляющая удобную навигацию по имеющимся трассам, а также позволяющая просматривать данные по трассе в удобном для восприятия человеком виде и присваивать трассе класс истина/ложь.

Цель проекта — разработать программное обеспечение, позволяющее пользователю производить разметку данных для дальнейшего обучения моделей машинного обучения.

Задачи проекта на этапах реализации:

- сформировать требования к ПО;
- сформировать спецификации к ПО;
- спроектировать ПО;
- реализовать ПО;
- провести тестирование ПО.

1 АНАЛИЗ ТРЕБОВАНИЙ

1.1 Модель бизнес - процесса

Полная картина взаимодействия между модулями показана на рисунке А.1. Схема используемой базы данных представлена на рисунке А.2.

Данные в системе проходят через следующие этапы:

1. Фильтрация трасс.
2. Выбор трассы для отображения.
3. Отрисовка трассы.
4. Разметка трассы.

1.1.1 Фильтрация трасс

На основе выставленных фильтров по классу цели, сигнатуры цели, количеству отметок в трассе, типу цели формируется SQL-запрос к базе данных, для получения списка id трасс, что удовлетворяют фильтрам и сохраняются в оперативной памяти. По умолчанию все фильтры неактивны.

1.1.2 Выбор трассы для отображения

По умолчанию выбирается первая трассы из выборки, далее пользователь может переключать трассы. И после по выбранному id формируется sql запрос на получение данных о трассе, в которые входят значения координат отметок трассы, их скорости, ускорения, их дисперсии, а также время получения отметки. Данные сохраняются в оперативную память.

1.1.3 Отрисовка трассы

Для отрисовки используются данные о координатах отметок трассы, при этом высота пересчитывается для учёта кривизны земли, по формуле:

$$(x^2 + y^2 + z^2) + 8500000^2 - 2 \times (x^2 + y^2 + z^2) \times 8500000 \times \sqrt{\cos(\pi)} - 8500000$$

Так же указывается общее время жизни трассы и id в базе.

1.1.4 Разметка трассы

Пользователь при анализе трассы присваивает ей класс истинная/ложная. Если присвоенный класс не был уже установлен ранее — формируется SQL-запрос на обновление класса в базе данных.

1.2 Формулировка требований к системе

1.2.1 Требования к составу выполняемых функций

ПО просмотра данных с радара, с возможностью фильтрации и классификации должно выполнять следующие функции:

1. Функции для работы с пользователем:

- настройка программы:
 - выбор базы данных, с данными которой будет идти работа;
 - авторизация в базе данных;
 - выбор режима отображения (круговое, простое);
- отображение трассы в двух проекциях (вид сверху, вид сбоку);
- отображение графика по изменению во времени каждой характеристики отметок трассы;
- выделение отметок трассы;
- редактирование трассы
 - разделение трассы по выделенной отметке;
 - удаление трассы целиком;
 - удаление выделенных отметок;
- просмотр данных выделенных отметок;
- присвоение класса (ложный/истинный) отображаемой трассы;
- подключение модуля классификации моделью машинного обучения;

– классификация отметок трассы с помощью подключенного модуля классификации моделью машинного обучения;

– фильтрация трасс по признакам:

- тип цели;
- сигнатура;
- класс;
- файла записи;
- количества отметок;
- тип радиолокационного комплекса (РЛК);
- номер трассы;

– переход к трассе по номеру в выборке по активным фильтрам;

– переход к трассе по номеру в базе;

– фильтр трасс по загруженному файлу лога процесса тестирования модели машинного обучения.

– выгрузка данных по активным фильтрам в файлы формата `.csv` для обучения, тестирования и объединенный.

2. Функции администрирования не предусмотрены.

На рисунке А.3 представлена диаграмма вариантов использования.

1.2.2 Организация входных и выходных данных

1. Входные данные:

– записи трасс. Организация поступления — запросы к базе данных;

– `.log` файл содержащий `id` трасс, в которых модель машинного обучения допустила ошибки при тестировании. Организация поступления — считывание по указанному пути, по требованию пользователя;

2. Выходные данные:

- траектория трассы в плоскостях xu , z , в виде графиков.
- Организация получения — отображение на экране пользователя;
- файлы формата `.csv` для обучения, тестирования и объединенный. Организация получения — загрузка на устройство пользователя по требованию.

1.2.3 Временные характеристики

Работа функционала классификация отметок трассы с помощью подключенного модуля классификации моделью машинного обучения определяется сложностью модели.

Работа функционала пользователя по выгрузке данных определяется количеством трасс в выборке.

1.2.4 Требования к надёжности

- система не должна предоставлять доступ к данным записей, если пользователь не может авторизоваться в базе данных;
- система должна иметь режим работы без сохранения изменений;
- система не должна искажать правки данных пользователя;
- система должна сменять отображение одной трассы на следующую не более чем за секунду, в случаях когда следующая трасса включает не более двух сотен отметок, в иной ситуации допустима задержка.

1.2.5 Требования к информационной и программной совместимости

Разработка ПО ведётся на языке программирования Python версии 3.12 с использованием сторонних библиотек `easygui`, `matplotlib`, `psycopg`. При получении данных идёт работа с СУБД PostgreSQL.

1.3 Сравнение системы с аналогами

На текущий момент прямых зарубежных и отечественных аналогов систем с покрывающим все нужды функционалом нет.

Существуют различные ПО для разметки данных, такие как SuperAnnotate [1], Telus Data Annotation [2], но данные инструменты больше ориентированы для изображений и анализа речи и не удовлетворяют всем требованиям, а также не распространяются свободно.

2 ОПРЕДЕЛЕНИЕ СПЕЦИФИКАЦИЙ

2.1 Модель проектирования и разработки системы

Для разработки ПО была предпочтена эволюционная модель.

Эволюционная модель разработки программного обеспечения (ПО) основана на следующей идее: разрабатывается первоначальная версия продукта, которая передаётся на испытание пользователям, затем она дорабатывается с учётом мнения пользователей, получается промежуточная версия продукта, которая также проходит «испытание пользователем», снова дорабатывается и так несколько раз, пока не будет получен необходимый программный продукт.

Различают два подхода к реализации эволюционного метода разработки:

1. Подход пробных разработок. Здесь большую роль играет постоянная работа с заказчиком (или пользователями) для того, чтобы определить полную систему требований к ПО, необходимую для разработки конечной версии продукта. В рамках этого подхода вначале разрабатываются те части системы, которые очевидны или хорошо специфицированы. Система эволюционирует (дорабатывается) путём добавления новых средств по мере их предложения заказчиком.

2. Прототипирование. Здесь целью процесса эволюционной разработки ПО является поэтапное уточнение требований заказчика и, следовательно, получение законченной спецификации, определяющей разрабатываемую систему. Прототип обычно строится для экспериментирования с той частью требований заказчика, которые сформированы нечётко или с внутренними противоречиями.

При разработке использовался подход пробных разработок

Данная модель была выбрана по причине постоянного изменения ситуации при работе с машинным обучением, появляются новые данные признаки и все это необходимо анализировать при разметке

2.2 Инструменты разработки

Для разработки ПО выбран ряд инструментов:

Текстовый редактор VS Code [3]. Имеет подсветку и подсказки для синтаксиса Python, также имеет интеграцию с git и поддерживает режим отладки.

СУБД PostgreSQL [4]. Базируется на языке SQL и поддерживает многие из возможностей стандарта SQL:2011 и ряд возможностей SQL:2016 в части работы с данными в формате JSON.

Язык программирования Python [5]. мультипарадигмальный высокоуровневый язык программирования общего назначения с динамической строгой типизацией и автоматическим управлением памятью. Язык ориентирован на повышение производительности разработчика, читаемости кода и его качества, а также на обеспечение переносимости написанных на нём программ.

Сообществом Python было реализовано огромное количество библиотек, в том числе для работы с базами данных, графиками и реализации графического интерфейса. Для реализации ПО просмотра данных с радара, с возможностью фильтрации и классификации используются следующие библиотеки:

- psycopg, позволяет работать с базами данных PostgreSQL [6];
- matplotlib, библиотека с открытым исходным кодом для визуализации данных в Python. С её помощью можно создавать точечные и круговые диаграммы, линейные графики, гистограммы, диаграммы ошибок, 3D-графики [7];
- easygui, позволяет реализовать простой графический интерфейс [8];

- keras [9], tensorflow [10], библиотеки для машинного обучения.

2.3 Входные и выходные данные

Используются следующие входные и выходные данные.

Входные данные:

- данные трасс, содержащие подробную информацию о отметках;
- .log файл содержащий id трасс, в которых модель машинного обучения допустила ошибки при тестировании;
- модуль машинного обучения для просмотра результатов вызова автоматической классификации на текущей трассе, выбранной пользователем.

Выходные данные:

- траектория трассы в плоскостях x, y, z , в виде графиков;
- файлы формата .csv для обучения, тестирования и объединенный.

3 ПРОЕКТИРОВАНИЕ

Полная картина взаимодействия между модулями показана на рисунке А.1.

Данные в системе проходят через следующие этапы:

1. Фильтрация трасс.
2. Выбор трассы для отображения.
3. Отрисовка трассы.
4. Разметка трассы.
5. Редактирование трассы.

3.1 Фильтрация трасс

На основе выставленных фильтров по классу цели, сигнатуры цели, количеству отметок в трассе, типу цели формируется SQL-запрос к базе данных, для получения списка id трасс, что удовлетворяют фильтрам и сохраняются в оперативной памяти. По умолчанию все фильтры неактивны.

3.2 Выбор трассы для отображения

По умолчанию выбирается первая трассы из выборки, далее пользователь может переключать трассы. И после по выбранному id формируется sql запрос на получение данных о трассе, в которые входят значения координат отметок трассы, их скорости, ускорения, их дисперсии, а также время получения отметки. Данные сохраняются в оперативную память.

3.3 Отрисовка трассы

Для отрисовки используются данные о координатах отметок трассы, при этом высота пересчитывается для учёта кривизны земли, по формуле:

$$(x^2 + y^2 + z^2) + 8500000^2 - 2 \times (x^2 + y^2 + z^2) \times 8500000 \times \sqrt{\cos(\pi)} - 8500000$$

Так же указывается общее время жизни трассы и id в базе.

3.4 Разметка трассы

Пользователь при анализе трассы присваивает ей класс истинная/ложная. Если присвоенный класс не был уже установлен ранее — формируется SQL-запрос на обновление класса в базе данных.

4 КОДИРОВАНИЕ

Разработка ПО проходит в несколько этапов. Сначала проектируется база данных, как источник данных, реализуется первичная версия приложения, далее происходит соединение двух компонент. Сформированный таким образом прототип проходит этап согласования с руководителем разработки. Проверяются:

- корректность отображения данных;
- скорость переключения трасс с различной длиной;
- корректность обновления трассы в базе на указанный признак.

По результатам согласования происходит либо утверждение прототипа, либо его переработка. Таким образом, в течение итераций были добавлены:

- возможность фильтрации трасс;
- возможность прямого перехода к трассе по её номеру в выборке;
- возможность прямого перехода к трассе по её id в базе;
- возможность выделения отметок трассы для просмотра подробных данных;
- возможность разделять трассу на части;
- возможность удалять выделенные отметки;
- возможность удалять трассу целиком;
- возможность подключать модуль машинного обучения для автоматической классификации трассы;
- возможность загружать .log файл для просмотра трасс, где ошибается тестируемый модуль.

Взаимодействие с пользователем рассмотрено на рисунке 4.1.



Рисунок 4.1 - Диаграмма последовательности взаимодействий пользователя и ПО

Аутентификация

Введите логин и пароль:

Логин

Пароль

Cancel OK

Рисунок 4.2 - Авторизация в базе

На рисунках 4.2-4.4 представлен примеры состояния ПО после взаимо-
действия с пользователем

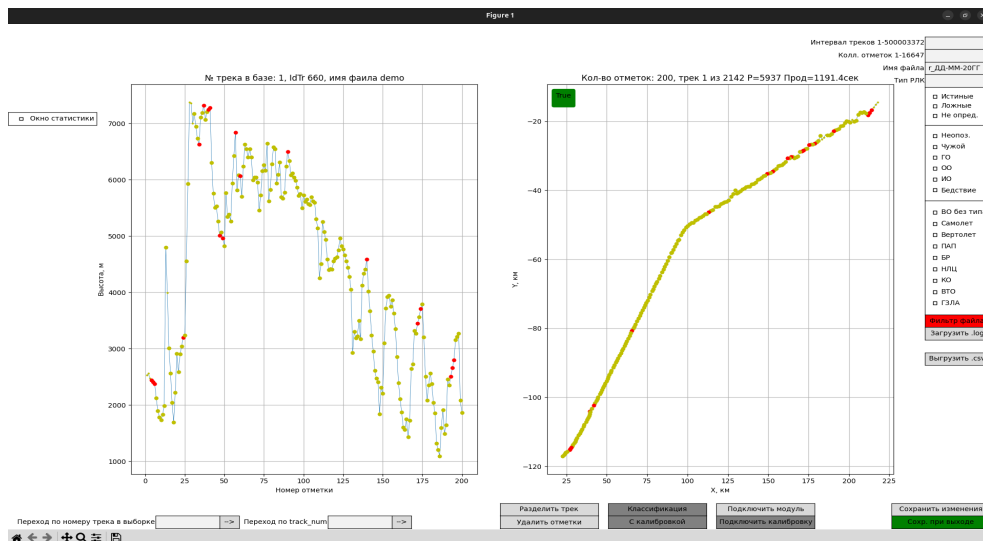


Рисунок 4.3 - Начальное состояние ПО

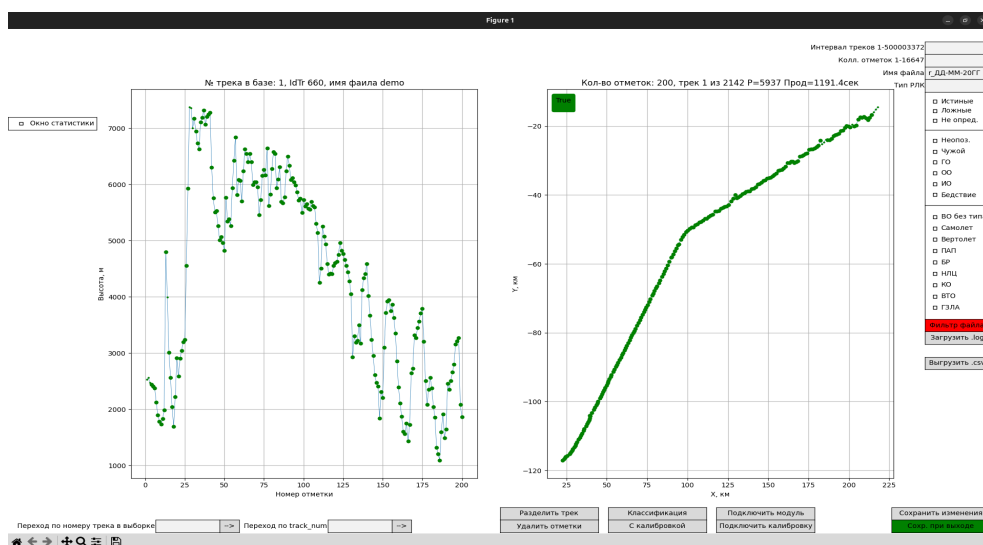


Рисунок 4.4 - После вызова модуля машинного обучения для
классификации отметок

5 ТЕСТИРОВАНИЕ

В данной работе предусмотрено тестирование по технологии чёрного ящика с использованием методологии эквивалентного разбиения.

Тестирование будет затрагивать проверки корректности входных данных, чтобы обеспечить корректную работу ПО.

Таблица 5.1 — Классы эквивалентности

Входные условия	Классы эквивалентности	
	<i>Правильные</i>	<i>Неправильные</i>
Фильтрация трасс		
Параметры интервала id трасс	\d+-\d+ (1)	\w+-\d+ (2), \d+-\w+ (3), \w+-\w+ (4), \w+ (5), \d+ (6)
Параметры интервала количества отметок в трассе	\d+-\d+ (7)	\w+-\d+ (8), \d+-\w+ (9), \w+-\w+ (10), \w+ (11), \d+ (12)
Имя файла	r_DD_MM_20YY (13)	Иные сочетания символов (14)
Редактирование трасс		
Количество выделенных отметок для разделения трассы	1 (15)	!=1 (16)
Количество выделенных отметок для их удаления	>0 (17)	<1 (18)
Подключение модуля машинного обучения		
Содержимое каталога с модулем	Два каталога: «module», «model», где находятся файл модуля и файлы моделей соответственно (19)	Отсутствует каталог «module» (20), отсутствует каталог «model» (21), каталог «module» пуст (22), каталог «model» пуст (23)
Загрузка .log файла		
Формат содержимого файла	Json-структура файла с наличием набора ключей: [Local stats][Model errors][TrId] (24)	Файл не соблюдает структуру Json (25), файл не содержит ключей [Local stats][Model errors][TrId] (26).

Тестирование проводится в ручном режиме.

Далее приведены тестовые сценарии, покрывающие различные классы эквивалентности.

1. Первый сценарий включает правильные параметры фильтров:

- Параметр интервала id трасс — 20-10000;
- Параметры интервала кол-ва отметок в трассе — 20-100;
- Имя файла — r_12_12_2024.

В данном сценарии покрываются классы эквивалентности 1, 6, 11, соответствующие правильным параметрам.

2. Сценарий покрывающий классы эквивалентности 4, 5, 10, 11, 14:

- Параметр интервала id трасс — abc-def;
- Параметры интервала кол-ва отметок в трассе — abc-def;
- Имя файла — r_dd_gb_202f.

3. Сценарий покрывающий класс эквивалентности 15:

- Выделить одну отметку на трассе.

4. Сценарий покрывающий классы эквивалентности 16, 17:

- Выделить три отметки на трассе.

5. Сценарий покрывающий класс эквивалентности 18:

- Не выделять отметки на трассе.

6. Сценарий покрывающий класс эквивалентности 19:

- Указать путь к любому модулю.

7. Сценарий покрывающий класс эквивалентности 24:

- Указать путь к файлу, сгенерированному скриптом тестирования

модуля машинного обучения.

8. Сценарий покрывающий классы эквивалентности 20, 21, 22, 23:

- Указать путь к пустому каталогу.

Заключение

В процессе курсового проекта был исследован процесс разработки программного обеспечения. Был разработан прототип ПО

ПО в текущем состоянии позволяет просматривать трассы, редактировать их, а также присваивать им класс истинная/ложная, как в ручном так и в автоматическом, при помощи машинного обучения, режимах.

Список использованных источников

1. Официальный сайт SuperAnnotate [Электронный ресурс]: Официальный сайт SuperAnnotate. URL: <https://www.superannotate.com/> (дата обращения 7.11.2024).
2. Официальная страница Telus Data Annotation [Электронный ресурс]: Официальный сайт Telus. URL: <https://www.telusdigital.com/solutions/ai-data-solutions/data-annotation> (дата обращения 8.11.2024).
3. Официальный сайт VS Code [Электронный ресурс]: Официальный сайт VS Code. URL: <https://code.visualstudio.com/> (дата обращения 5.09.2024).
4. Официальная документация PostgreSQL [Электронный ресурс]: Официальный сайт PostgreSQL. URL: <https://www.postgresql.org/docs/> (дата обращения 29.09.2024).
5. Официальная документация Python3 [Электронный ресурс]: Официальный сайт Python. URL: <https://docs.python.org/3/> (дата обращения 5.09.2024).
6. Официальная документация Psycopg [Электронный ресурс]: Официальный сайт Psycopg. URL: <https://www.psycopg.org/psycopg3/docs/> (дата обращения 5.09.2024).
7. Официальная документация Matplotlib [Электронный ресурс]: Официальный сайт Matplotlib. URL: <https://matplotlib.org/stable/api/index> (дата обращения 5.09.2024).
8. Официальная документация Easygui [Электронный ресурс]: Официальный сайт Easygui. URL: <https://easygui.sourceforge.net/> (дата обращения 5.09.2024).
9. Официальная документация Keras [Электронный ресурс]: Официальный сайт Keras. URL: <https://keras.io/api/> (дата обращения 13.10.2024).

10. Официальная документация TensorFlow [Электронный ресурс]:
Официальный сайт TensorFlow. URL: https://www.tensorflow.org/api_docs/python/tf (дата обращения 13.10.2024).ц

Приложение А

(обязательное)

Схема взаимодействия модулей программы

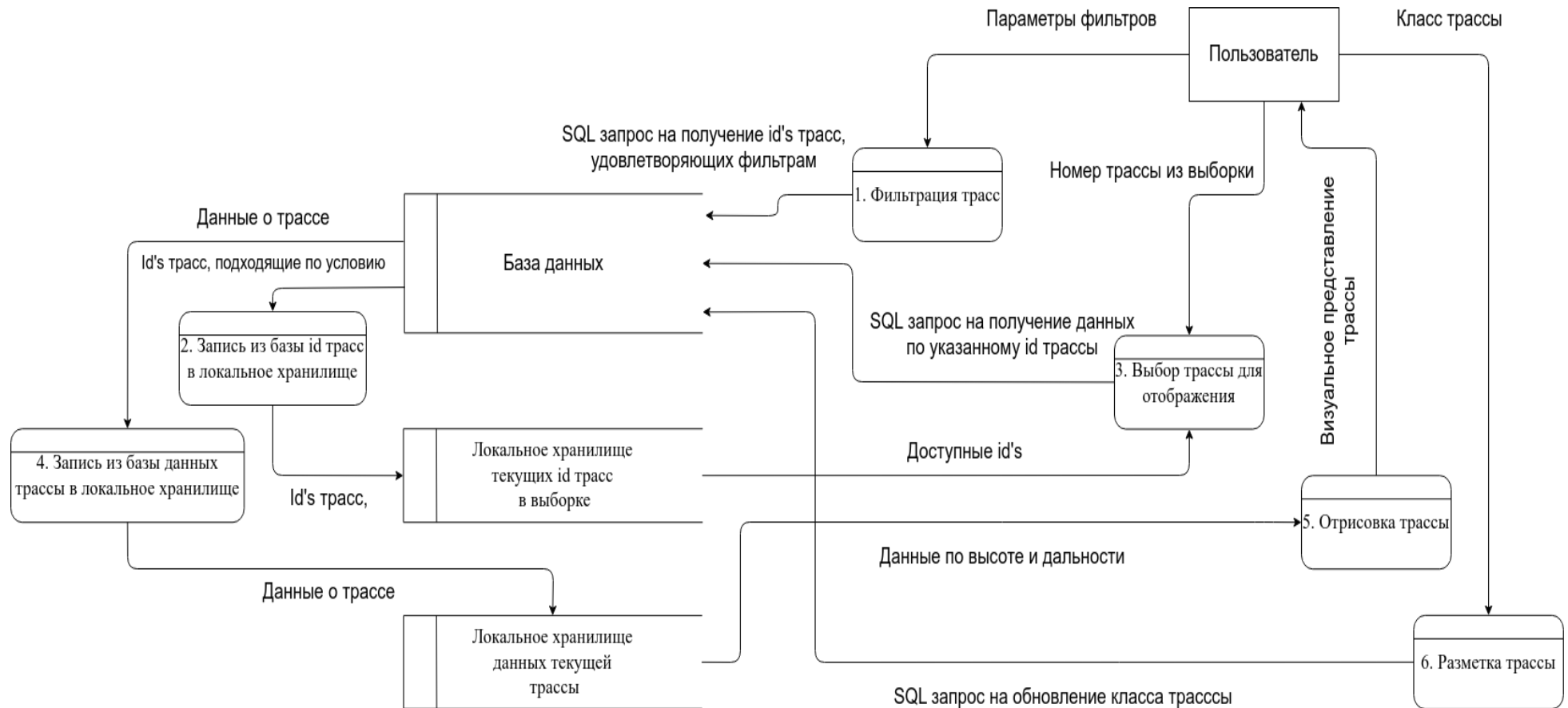


Рисунок А.1 - Схема взаимодействия модулей программы



Рисунок А.2 - Схема базы данных

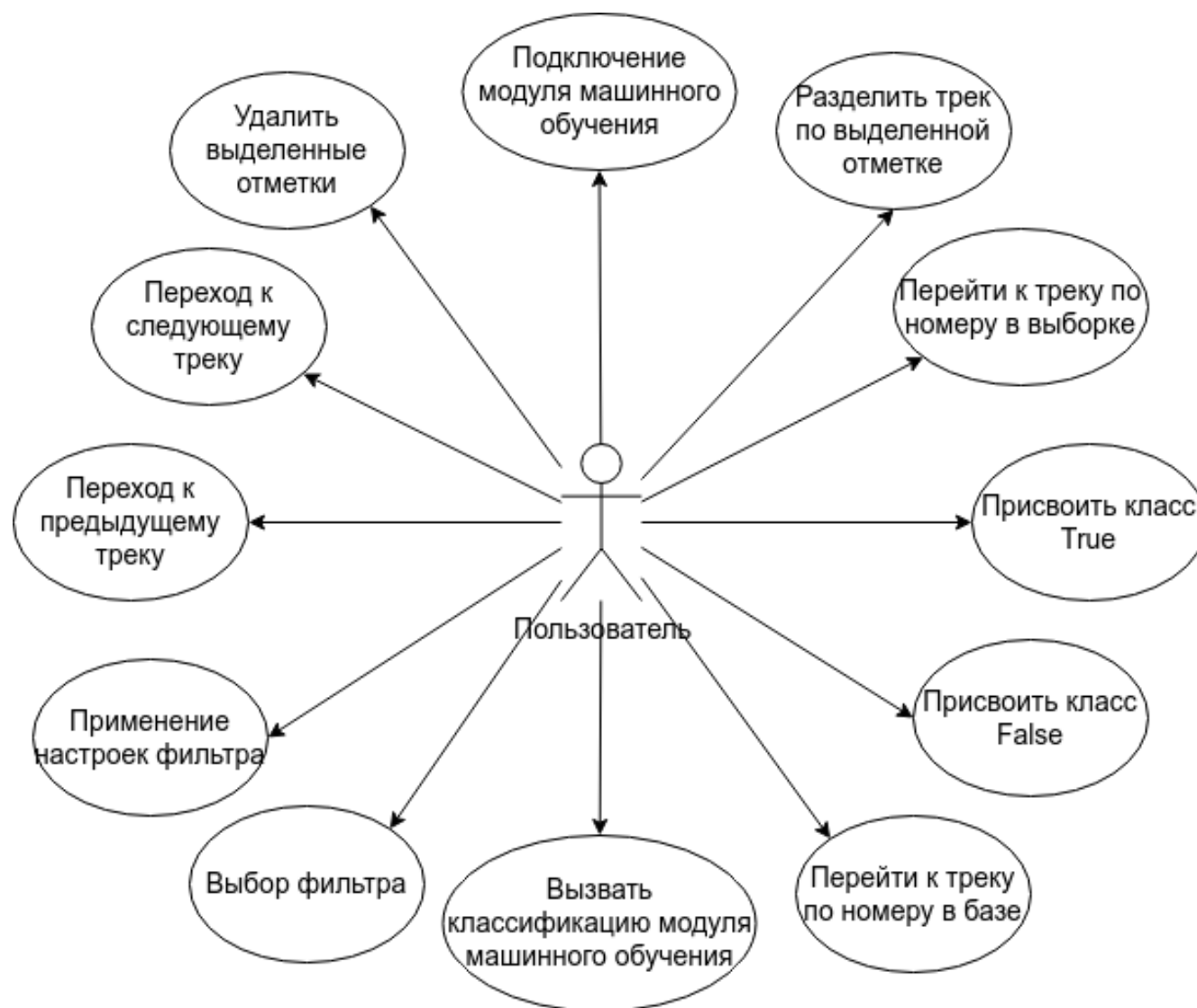


Рисунок А.3 - Диаграмма вариантов использования