

КОМИТЕТ ПО ОБРАЗОВАНИЮ ПРАВИТЕЛЬСТВА САНКТ-ПЕТЕРБУРГА

Санкт-Петербургское государственное
бюджетное профессиональное образование учреждение
«Колледж информационных технологий»

ОТЧЕТ
по практической работе
«Разработка приложения для анализа спектров»
по МДК01.03.

Работу выполнил студент 493 гр.:
Кашицын Артем Андреевич
Преподаватель:
Фомин Александр Валерьевич

Санкт-Петербург 2023

ОГЛАВЛЕНИЕ

ОПИСАНИЕ ИНТЕРФЕЙСА	3
СТРУКТУРА БАЗЫ ДАННЫХ.....	9
ОПИСАНИЕ ФУНКЦИЙ API	13
ДЕМОНСТРАЦИЯ ФУНКЦИЙ	21
ИТОГ РАБОТЫ	45

ОПИСАНИЕ ИНТЕРФЕЙСА

Макет экрана со списком экспериментов, который отображается при запуске приложения отображен на рисунке 1.

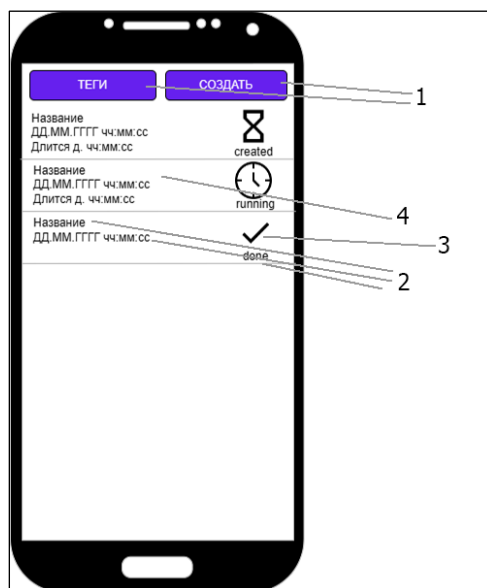


Рисунок 1 – Макет экрана со списком экспериментов

На экране со списком экспериментов расположены элементы под следующими номерами:

- 1 – Button
- 2 – TextView
- 3 – ImageView
- 4 – ListView

Макет главного экрана с просмотром спектра отображен на рисунке 2.

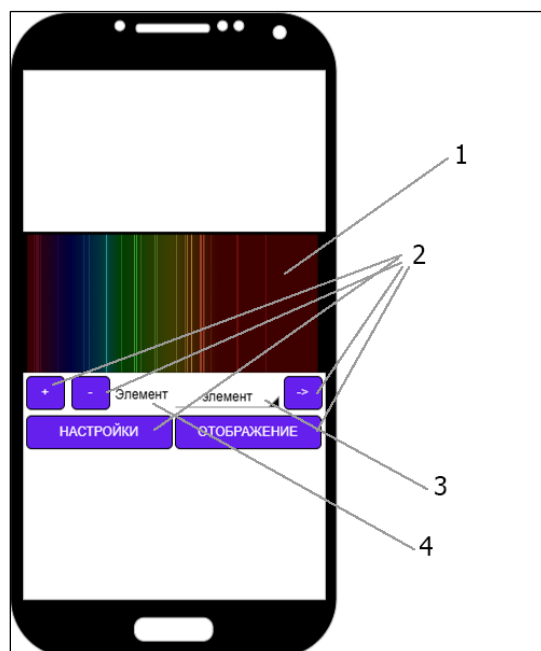


Рисунок 2 – Макет экрана просмотра спектра

На экране просмотра спектра расположены элементы под следующими номерами:

- 1 – SpectraView
- 2 – Button
- 3 – Spinner
- 4 – TextView

Макет экрана настроек отображен на рисунке 3.

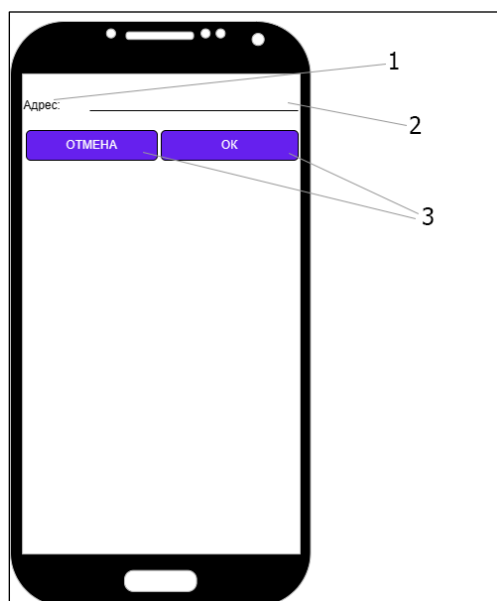


Рисунок 3 – Макет экрана настроек

На экране настроек расположены элементы под следующими номерами:

1 – TextView

2 – EditText

3 – Button

Макет диалогового окна с настройками отображения эксперимента или спектра отображен на рисунке 4.

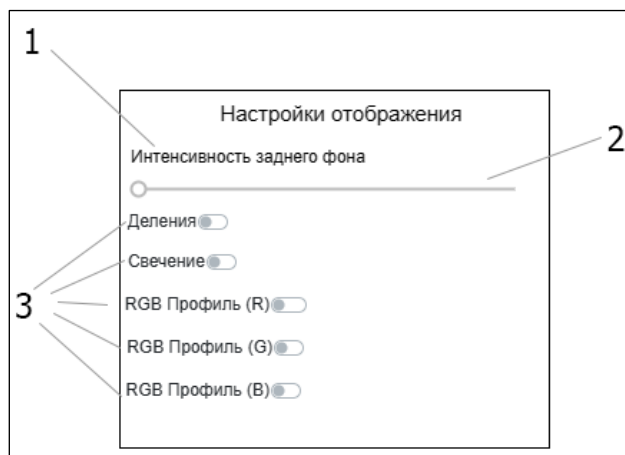


Рисунок 4 – Макет диалогового окна с настройками отображения

На диалоговом окне расположены элементы под следующими номерами:

1 – TextView

2 – Slider

3 – Switch

Макет экрана просмотра заверченного эксперимента отображен на рисунке 5.

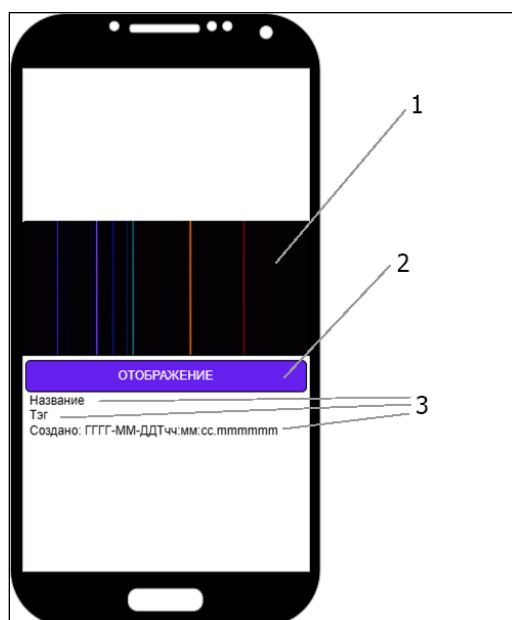


Рисунок 5 – Макет экрана просмотра эксперимента

На экране просмотра заверченного эксперимента расположены элементы под следующими номерами:

1 – SpectraView

2 – Button

3 – TextView

Макет экрана отправки эксперимента отображен на рисунке 6.

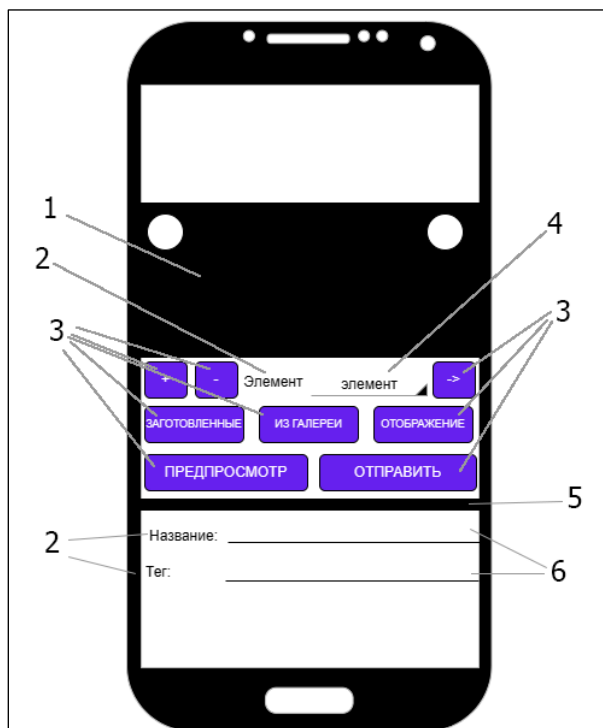


Рисунок 6 – Макет экрана отправки эксперимента

На экране отправки эксперимента расположены элементы под следующими номерами:

1 – SpectraView

2 – TextView

3 – Button

4 – Spinner

5 – ImageView

6 – EditText

SpectraView отображает работу со спектром. На нем отрисовывается задний план, спектральные линии, деления в нанометрах, готовые изображения экспериментов, графики цветового профиля и свечения, точки для кадрирования

изображения. Пользователь имеет возможность изменять масштаб отрисовки, изменяя параметры минимальной и максимальной координаты ширины изображения, а также перемещаться по изображению горизонтально.

Button – это кнопка, по нажатию которой выполняется определенная функция. На экране со списком экспериментов расположены следующие действия: открытие диалогового окна с выбором тега эксперимента, открытие экрана отправки изображения. На экране просмотра спектра кнопки отвечают за увеличение и уменьшение масштаба, загрузку спектральных линий выбранного элемента (как и на экране создания эксперимента), переход на экран настроек, открытие диалогового окна с настройкой отображения (как и на экранах создания и просмотра эксперимента). На экране настроек расположены действия с сохранением адреса конечной точки API, отмену сохранения. На экране отправки эксперимента также имеются кнопки для выбора изображения из заготовленных ресурсов, галереи, для предпросмотра кадрированного снимка с SpectraView, отправки снимка с введенным названием и тэгом эксперимента.

TextView используется как подпись для определения пользователем поля ввода данных, а также для отображения данных экспериментов.

EditText является полем для ввода данных пользователем. Например, ввод адреса сервера, названия и тэга эксперимента.

Spinner – это выдвигающийся список, элемент которого может стать активным после выбора пользователем. Используется для выбора элемента со спектральными линиями.

ListView – компонент, отображающий список. В данном приложении используется пользовательский вид адаптера – компонента для отображения слоев с линиями в определенном виде. Используется для отображения списка экспериментов с выбранным тэгом, состоит из TextView с отображением названия, даты и времени создания, длительности, статуса с сопровождением соответствующих изображений.

Switch – переключатель, который принимает два значения (активное и не активное). Служит для включения отображения делений, графиков свечения, цветового профиля RGB.

Slider – ползунок, который изменяет определенное значение от одной единицы к другой в зависимости в какую сторону пользователь переместил его. Используется для выбора коэффициента яркости заднего плана спектра от 0 до 1.

ImageView – компонент для отображения изображений, используется для отображения кадрированного слепка с SpectraView, а также отображения иконки для каждого статуса.

СТРУКТУРА БАЗЫ ДАННЫХ

ER-диаграмма отображена на рисунке 7.

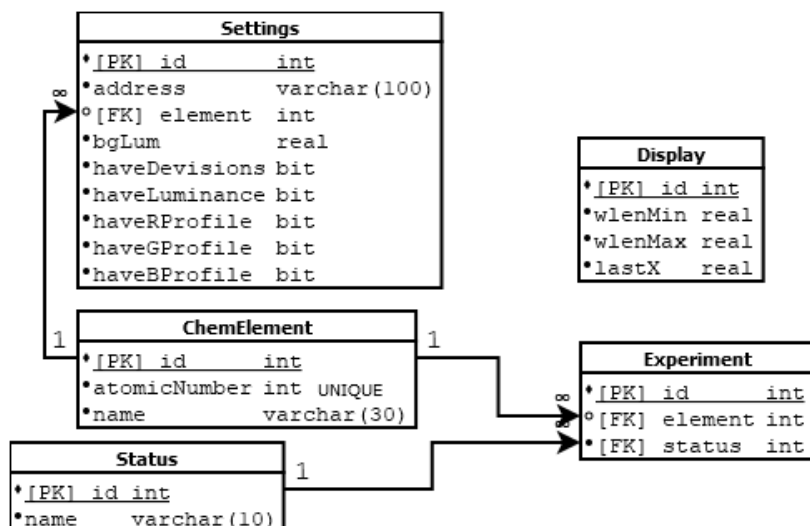


Рисунок 7 – ER-диаграмма

Таблица Display состоит из следующих столбцов:

id – идентификатор настроек отображения

wlenMin – координата x начала спектра

wlenMax – координата x конца спектра

lastX – координата x, на которой пользователь последний раз перемещался

Имя	Тип	Размер	Уникальное	Пустое	Ключ
id	int	-	да	нет	pk
wlenMin	real	-	нет	нет	-
wlenMax	real	-	нет	нет	-
lastX	real	-	нет	нет	-

Примеры входных данных:

1|380.0|780.0|0.0

1|500.0|700.0|120.0

1|313.02|713.02|669.38

Таблица Status состоит из следующих столбцов:

id – идентификатор статусов эксперимента

name – название статуса

Имя	Тип	Размер	Уникальное	Пустое	Ключ
id	int	-	да	нет	pk
name	varchar	10	нет	нет	-

Примеры входных данных:

1|created

2|running

3|done

Таблица ChemElement состоит из следующих столбцов:

id – идентификатор химического элемента

atomicNumber – зарядовое число, идентификатор элемента

name – название химического элемента

Имя	Тип	Размер	Уникальное	Пустое	Ключ
id	int	-	да	нет	pk
atomicNumber	int	-	да	нет	-
name	varchar	30	нет	нет	-

Примеры входных данных:

1|13|aluminum

2|20|calcium

3|18|argon

4|6|carbon

Таблица Experiment состоит из следующих столбцов:

id – идентификатор эксперимента

element – элемент эксперимента

status – статус эксперимента

Имя	Тип	Размер	Уникальное	Пустое	Ключ
id	int	-	да	нет	pk
element	int	-	нет	нет	fk
status	int	-	нет	нет	fk

Примеры входных данных:

27|2|2

18|1|2

10|3|2

9|4|2

Таблица Settings состоит из следующих столбцов:

id – идентификатор настроек

address – адрес конечной точки API

element – последний элемент

bgLum – коэффициент яркости заднего фона

haveDevisions – отображение делений

haveLuminance – отображение графика свечения

haveRProfile – отображение красного в графике цветового профиля

haveGProfile – отображение зеленого в графике цветового графика

haveBProfile – отображение синего в графике цветового профиля

Имя	Тип	Размер	Уникальное	Пустое	Ключ
id	int	-	да	нет	pk
address	varchar	100	нет	нет	-
Element	Int	-			
bgLum	Real	-			
haveDevisions	Bit	-			
haveLuminance	Bit	-			
haveRProfile	Bit	-			

haveGProfile	Bit	-			
haveBProfile	Bit	-			

Примеры входных данных:

1|http://labs-api.spbcoit.ru:80/lab/spectra/api|4|0.31|1|0|1|1|1

1| http://labs-api.spbcoit.ru:80/lab/spectra/api| |0|0|1|0|0|0

1| http://labs-api.spbcoit.ru:80/lab/spectra/api|3|1|0|1|1|1|1

ОПИСАНИЕ ФУНКЦИЙ API

Приложение взаимодействует с сервером по протоколу HTTP, отправляя POST-запросы с телом в формате JSON и получая ответы в виде этого же формата.

Набор входных данных:

atomic_num – число, которое идентифицирует элемент

nm_from – координата x начала спектра

nm_to – координата x конца спектра

steps – количество точек от nm_from до nm_to

tagname – тэг эксперимента

experiment – id эксперимента

experiment_id – id эксперимента

b64image – изображение в формате jpg, закодированного в base64

note – название эксперимента

tag – тэг эксперимента

x0 – левый-верхний угол

x1 – правый-верхний угол

y0 – левый-нижний угол

y1 – правый-нижний угол

Существует следующий набор запросов:

Функция rpc/get_tags

Тип запроса: POST

Получение списка существующих тэгов

Тело запроса: { }

Пример вызова отображен на рисунке 8

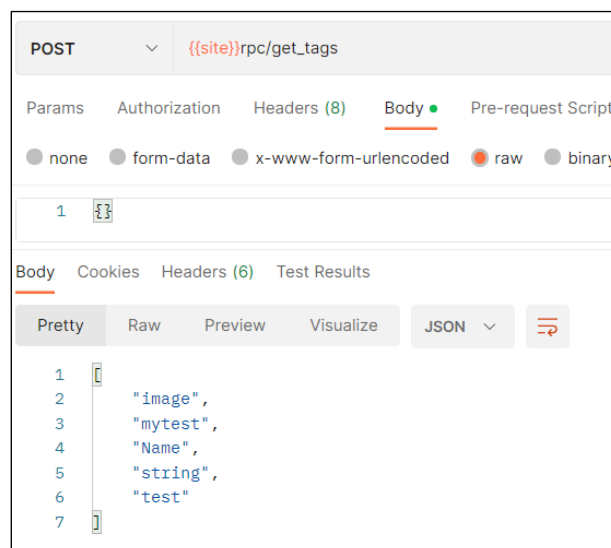


Рисунок 8 – Пример вызова функции rpc/get_tags

Пример ответа: [«tag1»,«tag2»...]

Функция rpc/get_experiments

Тип запроса: POST

Получение массива с id, датой создания, названием, статусом экспериментов с выбранным тэгом

Входные параметры: tagname

Пример вызова отображен на рисунке 9

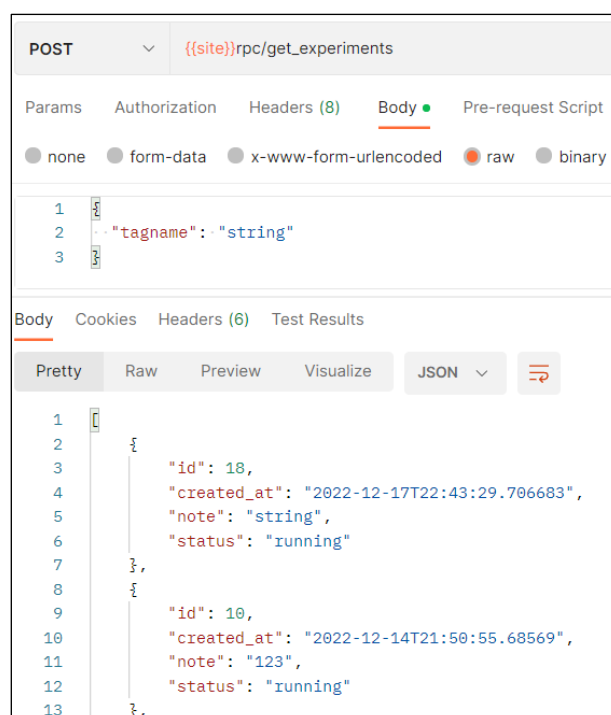


Рисунок 9 – Пример вызова функции rpc/get_experiments

Пример ответа: [{«id»:1, «created_at»:«2022-12-17T22:43:29.000000», «note»:«note», «status»:«created»},...]

Функция `rpc/get_status`

Тип запроса: POST

Получение статуса эксперимента

Входные параметры: `experiment`

Пример вызова отображен на рисунке 10

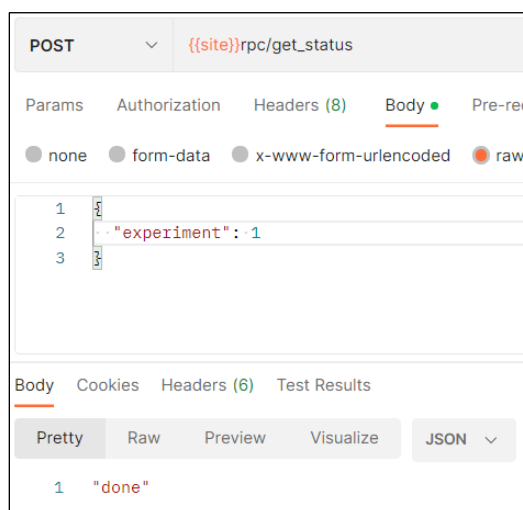


Рисунок 10 – Пример вызова функции `rpc/get_status`

Пример ответа: «created», «running», «done»

Функция `rpc/get_elements`

Тип запроса: POST

Получение массивов с номерами и названиями химических элементов

Входные параметры: `{}`

Пример вызова отображен на рисунке 11

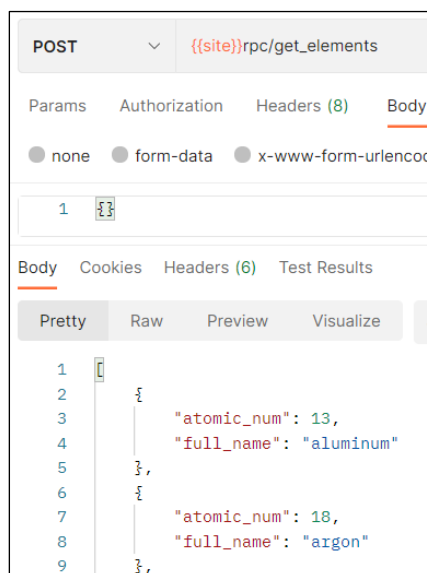


Рисунок 11 – Пример вызова функции rpc/get_elements

Примеры ответов: [{«atomic_num»: 13, «full_name»: «aluminum»},...]

Функция rpc/nm_to_rgb_range

Тип запроса: POST

Получение массивов с набором координаты между nm_from и nm_to и значениями красного, зеленого, синего цвета для отрисовки фона.

Входные параметры: nm_from, nm_to, steps

Пример вызова отображен на рисунке 12

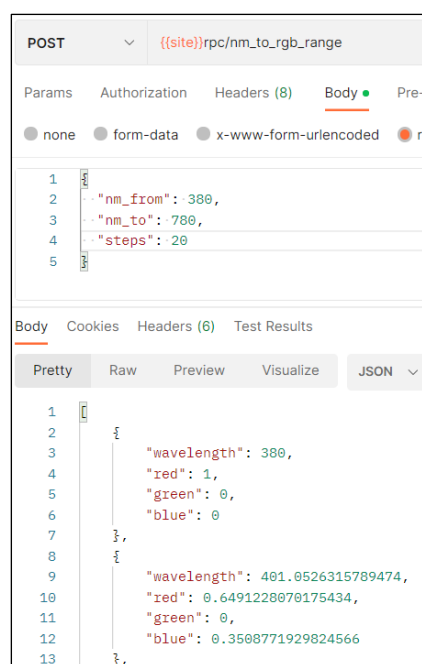


Рисунок 12 – Пример вызова функции rpc/nm_to_rgb_range

Примеры ответов: [{«wavelength»:380, «red»:1, «green»:0 «blue»:0},
...]

Функция rpc/get_lines

Тип запроса: POST

Получение массивов с набором координаты между nm_from и nm_to, относительной интенсивности и значениями красного, зеленого, синего цвета для отрисовки спектральных линий

Входные параметры: atomic_num

Пример вызова отображен на рисунке 13

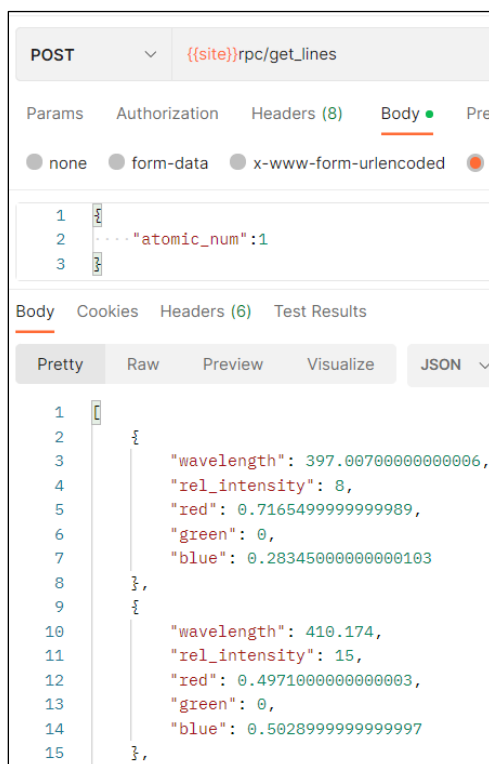


Рисунок 13 – Пример вызова функции rpc/get_lines

Примеры ответов: [{«wavelength»:300, «rel_intensity»:18, «red»:1, «green»:0 «blue»:0}, ...]

Функция rpc/get_experiment_data

Тип запроса: POST

Получение даты создания, тэга, названия, статуса, изображения, x0, y0, x1, y1 выбранного эксперимента

Входные параметры: experiment

Пример вызова отображен на рисунке 14.

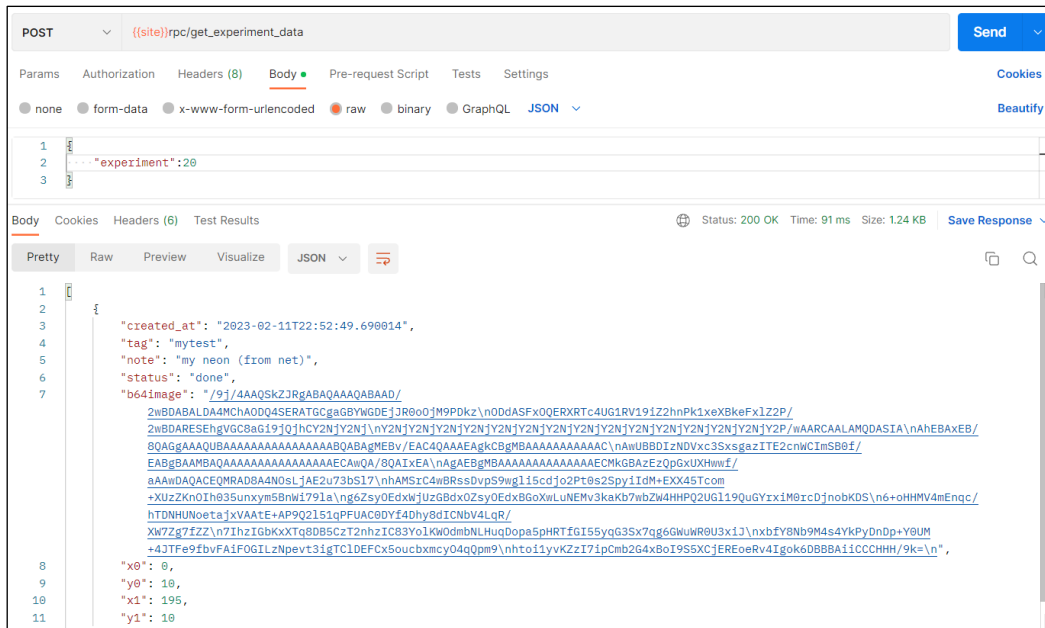


Рисунок 14 – Пример вызова функции `rpc/get_experiment_data`

Примеры ответов: {«created_at»: «2022-12-17T22:43:29.000000», «tag»: «mytest», «note»: «note», «status»: «done», «b64image»: «b64image», «x0»:0, «y0»:10, «x1»:100, «y1»:10}

Функция `rpc/get_luminance_profile`

Тип запроса: POST

Получения массивов с набором координаты x на изображении, а также значение свечения от 0 до 1.

Входные параметры: `experiment_id`

Пример вызова отображен на рисунке 15.

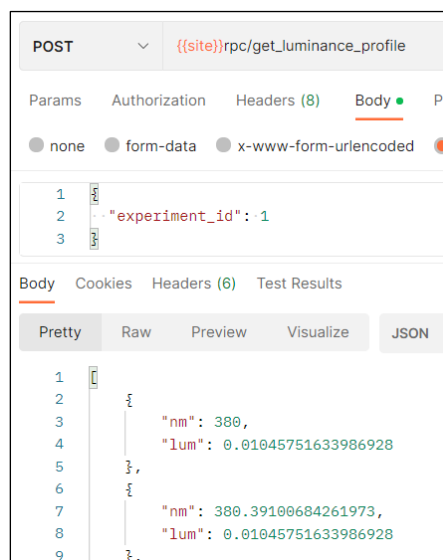


Рисунок 15 – Пример вызова функции rpc/get_luminance_profile

Примеры ответов: [{«nm»: 380, «lum»: 0.5},...]

Функция rpc/get_color_profile

Тип запроса: POST

Получение массивов с набором координаты x на изображении, значение красного, зеленого, синего цвета.

Входные параметры: experiment_id

Пример вызова отображен на рисунке 16

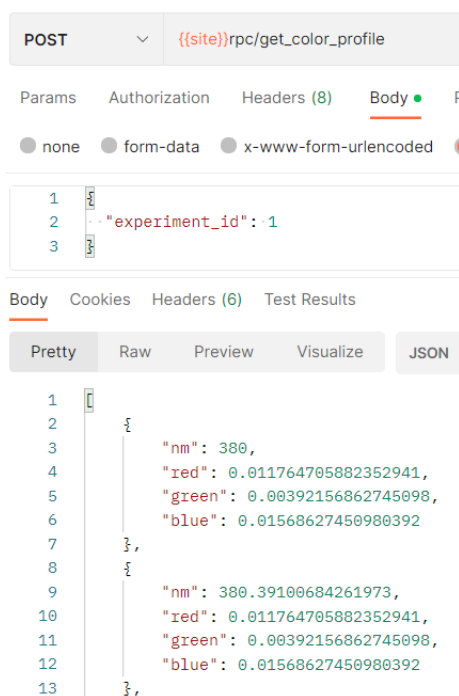


Рисунок 16 – Пример вызова функции rpc/get_color_profile

Примеры ответов: [{«nm»:380, «red»:0.1, «green»:0.1, «blue»:0.1}, ...]

Функция `grpc/run_experiment`

Тип запроса: POST

Отправка изображения, его координат, названия и тэга

Входные параметры: `b64image`, `note`, `tag`, `x0`, `x1`, `y0`, `y1`

Пример вызова отображен на рисунке 17

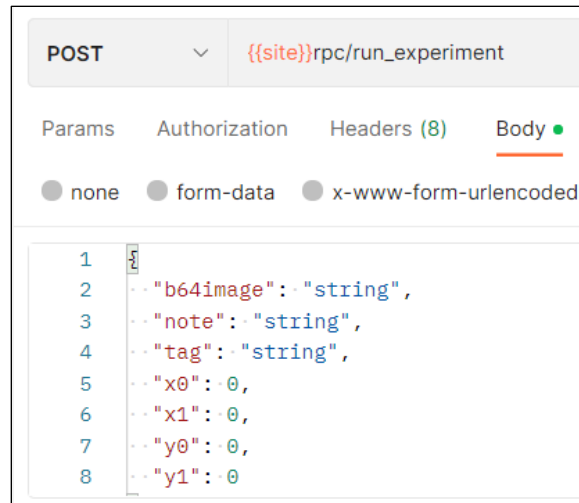


Рисунок 17 – Пример вызова функции `grpc/run_experiment`

Примеры ответов: `true`

ДЕМОНСТРАЦИЯ ФУНКЦИЙ

Иконка приложения отображена на рисунке 18.



Рисунок 18 – Иконка приложения

После запуска приложения, настройки приложения считываются с локальной базы данных: адрес сервера, свечение заднего градиента спектра, также настройки отображения в виде значения в нанометрах начала и конца рассматриваемого спектра, координата места, на котором остановился пользователь.

Класс `ApiHelper` отвечает за отправку запроса и получения ответа. Необходимо ввести адрес, тело запроса. При отправке в отдельном потоке создается подключение, настраивается тип отправляемых данных, тип запроса, добавляется тело запроса, через цикл происходит получение ответа, отключение соединения. Метод класса `onReady` наследуется для произведения действий с полученным ответом.

Первым делом, отправляется запрос на получение массива `string` существующих тэгов эксперимента.

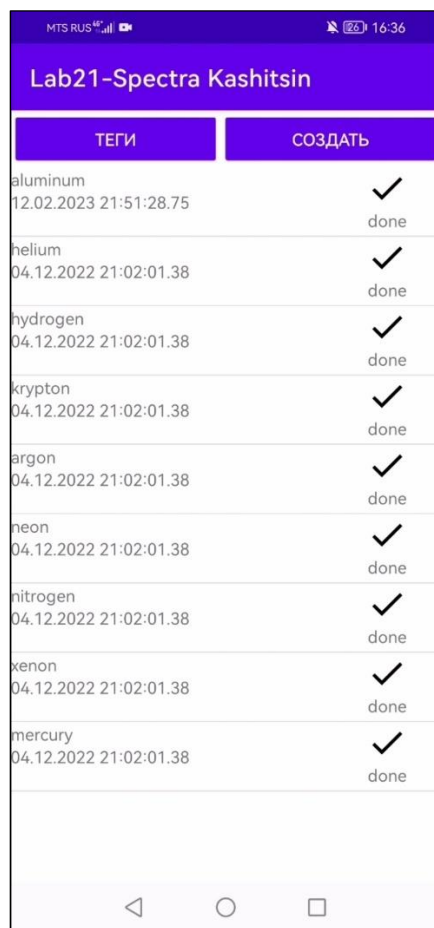
Класс `Experiment` имеет поля с `id`, датой создания, названием, и статусом эксперимента. При создании, отображение дата форматируется для предоставления в стандартном виде с указанием миллисекунд в точности до 2 символов.

Созданный `ExperimentAdapter` будет предоставлять данные `Experiment` отображаются в виде текста на элементе. В зависимости от статуса в `ImageView` отображается соответствующая картинка. Для созданного – изображение пустых песочных часов, обрабатывающегося – изображение циферблата, завершеного – галка. Если статус эксперимента не является завешенным, то определяется его

продолжительность (из даты, текущей вычитается дата создания и отображается в количестве дней, оставшихся часов, минут и т.д.)

Отправляется запрос на получения списка экспериментов по выбранному тэгу, в список экспериментов добавляются полученные эксперименты, а в список обрабатывающихся добавляются эксперименты без статуса «done». Если список обрабатывающихся не пуст, то создается таймер, который каждый секунду будет отправлять запрос на получение статуса по списку, а также обновляется адаптер, благодаря которому у пользователя каждую секунду изменяется длительность экспериментов.

Список экспериментов после запуска приложения с тегом «test» показан на рисунке 19.



The screenshot shows a mobile application interface with a purple header bar containing the text "Lab21-Spectra Kashitsin". Below the header are two buttons: "ТЕГИ" (Tags) and "СОЗДАТЬ" (Create). The main content area displays a list of experiments, each with a tag name, a timestamp, and a status indicator. The status indicator consists of a checkmark and the word "done".

ТЕГИ	СОЗДАТЬ
aluminum 12.02.2023 21:51:28.75	✓ done
helium 04.12.2022 21:02:01.38	✓ done
hydrogen 04.12.2022 21:02:01.38	✓ done
krypton 04.12.2022 21:02:01.38	✓ done
argon 04.12.2022 21:02:01.38	✓ done
neon 04.12.2022 21:02:01.38	✓ done
nitrogen 04.12.2022 21:02:01.38	✓ done
xenon 04.12.2022 21:02:01.38	✓ done
mercury 04.12.2022 21:02:01.38	✓ done

Рисунок 19 – Список экспериментов после запуска приложения

По кнопке «Теги» отображается диалоговое окно со списком тэгов полученных после запроса.

Список тэгов на 16.02 показан на рисунке 20.



Рисунок 20 – Список тэгов 16.02

После выбора «string», список экспериментов обновился, после получения данных от сервера.

Список экспериментов по тэгу string показан на рисунке 21.

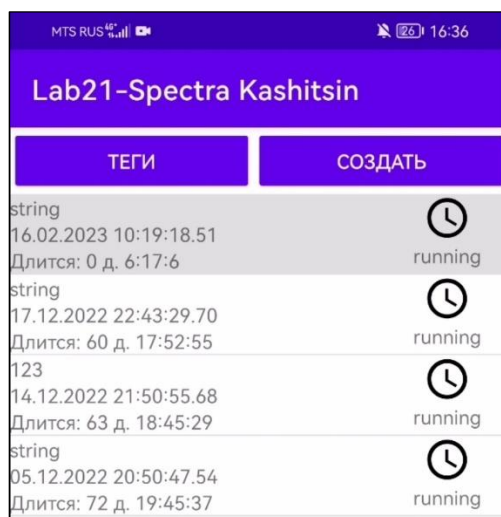


Рисунок 21 – Список экспериментов по тэгу string

Так как эксперимент не завершен, то открывается экран анализа спектра.

Элемент, на котором будет отображаться спектр – SpectraView, унаследованный от SurfaceView.

Класс, настраивающий SpectraView на определенное отображение – SpectraHelper. Настройка под анализ спектра представляет собой определение текущего экрана, отключение рисования изображения (так как пользователь только получает от сервера задний план и спектральные линии), отключение точек кадрирования, получение булевого значения из бд, определяющего наличие делений.

Если задний фон не загружен, то отправляется запрос на получение массива значений rgb от начала до конца спектра размером равному ширине экрана телефона, и по линиям размером в пиксель отрисовывается набор rgb.

Экран анализа спектра эксперимента «string» с загруженным фоном показан на рисунке 22.

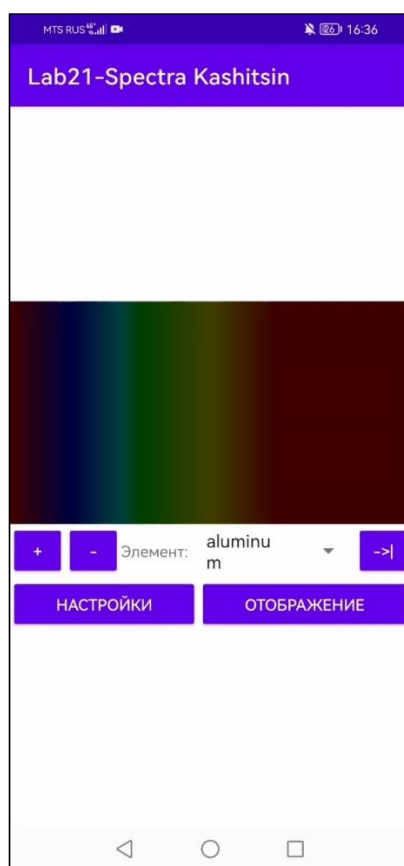


Рисунок 22 – Экран анализа спектра

Благодаря сохранению ширины спектра и последней координаты остановки, пользователь может перемещаться по спектру. Если пользователь прикоснется к элементу, спектр не отображается до тех пор, пока палец не будет убран. Во время движения, определяется изменение координаты на экране, сколько

нанометров приходится на пиксель, и изменяется координаты начала и конца спектра на полученные нанометры.

Отключение отрисовки фона во время движения показано на рисунке 23.

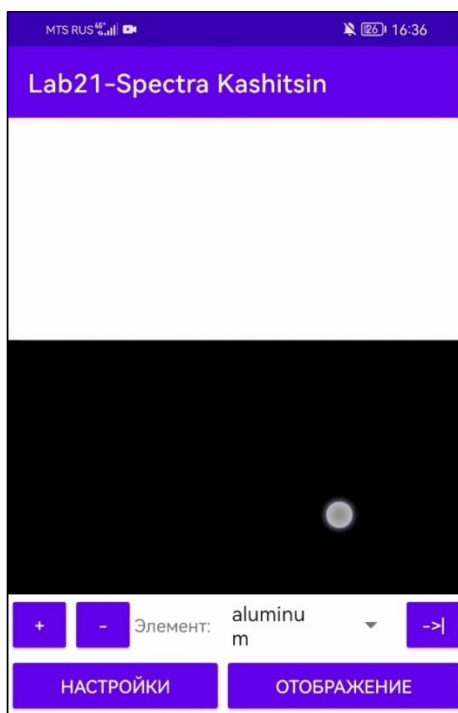


Рисунок 23 – Отключение отрисовки фона во время движения

Когда пользователь отпустит палец, можно увидеть результат перемещения на рисунке 24.

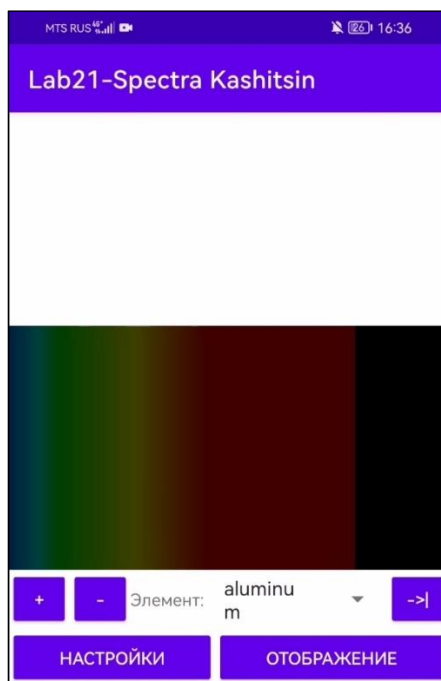


Рисунок 24 – Перемещение по спектру

Масштабирование происходит от и к центру отображения спектра. Приближение показано на рисунке 25.

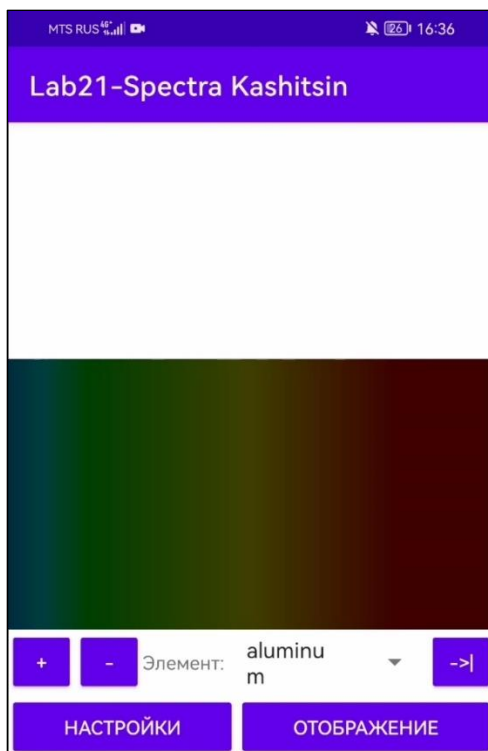


Рисунок 25 – Приближение

Отдаление показано на рисунке 26.

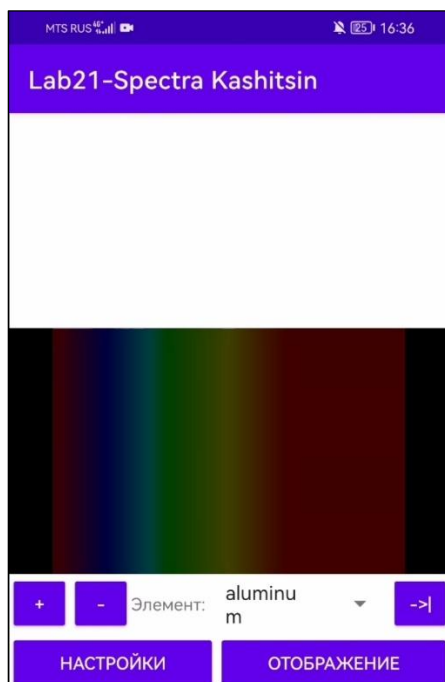


Рисунок 26 – Отдаление

При открытии эксперимента незавершенного, в локальную базу добавляется id и статус этого эксперимента. Отправляется запрос на получение химических элементов. Если в сохраненном эксперименте уже был задан элемент, то показывается он, если элемент не был задан, показывается последний выбранный, если нет последнего сохраненного элемента, показывается первый полученный от сервера.

Элементы представляются в виде объектов класса ChemElement, содержащим номер и название. Список элементов с сервера показан на рисунке 27.



Рисунок 27 – Список элементов для получения линий

По клику на загрузку спектральных линий выбранного элемента, отправляется запрос с номером элемента. Спектральные линии представляет класс SpectraLine, объекты которого напрямую создается из JSON элемента с теми же полями и добавляется в список SpectraView.

Функцией интерполяции преобразовывается координата спектра к координате на экране и рисуется линия высотой в SpectraView и полученным цветом.

Отрисовка спектральных линий calcium показан на рисунке 28.

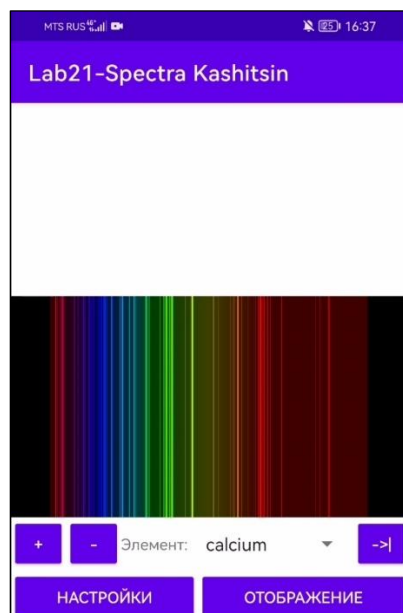


Рисунок 28 – Отрисовка calcium

Сохранение отображения ширины спектра и координаты движения происходит при движении и масштабировании. Загрузка эксперимента после перезагрузки приложения показана на рисунке 29.

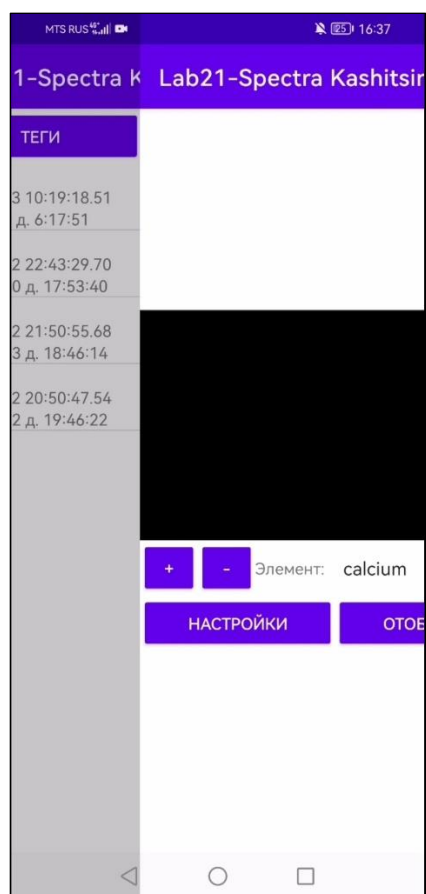


Рисунок 29 – Загрузка эксперимента после перезапуска

Сохранение отображения показано на рисунке 30.

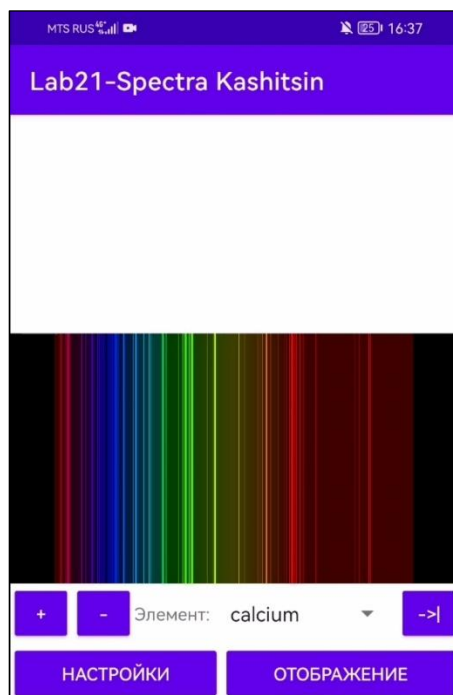


Рисунок 30– Сохранение отображения спектра

По кнопке отображения показывается диалоговое окно с ползунком свечения заднего плана, переключателем делений, значения которых загружаются с бд, и сохраняется в бд после изменения. Во время изменения значений, это отображается на компоненте. Диалоговое окно с отображением показано на рисунке 31.

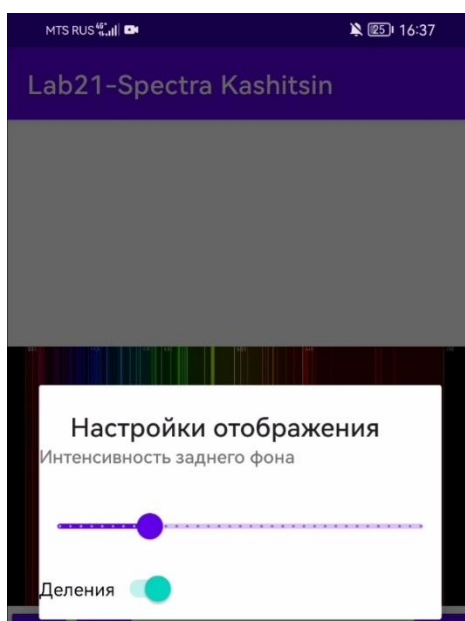


Рисунок 31– Диалоговое окно с настройками отображения

Если отображение делений включено, то методом интерполяции будут рисоваться деления 380,440,490,510,580,645,780 нанометров. Отображение делений показано на рисунке 32.

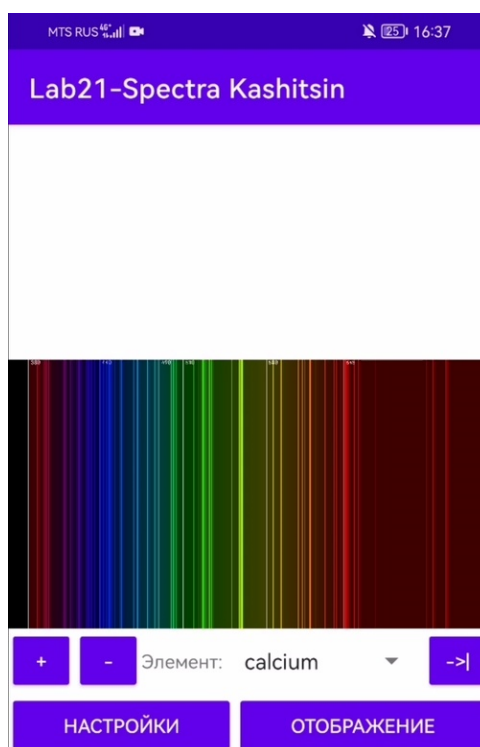


Рисунок 32 – Отображение делений

Отображение заднего фона также зависит от значения ползунка в диалоговом окне, где 0 – полная темнота, 1 – отображение всех цветов.

Отключение заднего фона показано на рисунке 33.

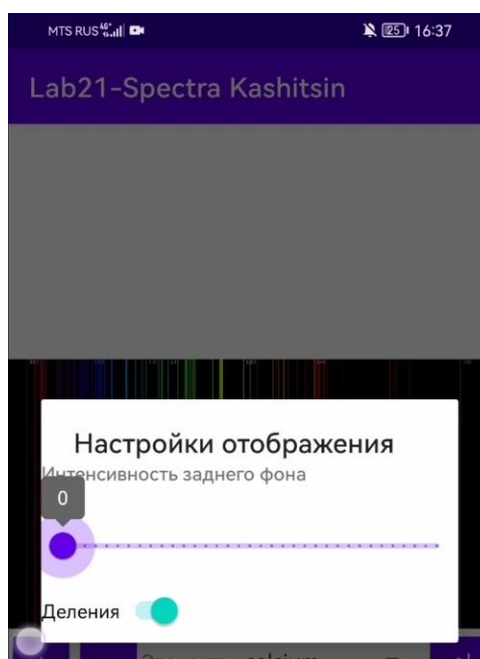


Рисунок 33 – Изменение отображения фонового градиента

Пока было открыто окно анализа спектра, прошло достаточно времени, и в списке экспериментов под тэгом «string» обновилась длительность.

Изменение длительности экспериментов показана на рисунке 34.

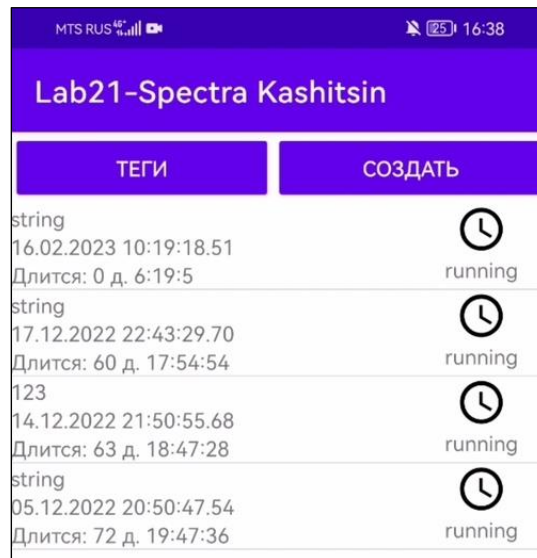


Рисунок 34 – Изменение длительности экспериментов

Перейдем к экспериментам под тегом «test» и откроем «krypton». Так как эксперимент завершен, необходимо загрузить его изображение. Отправляется запрос на получение данных эксперимента под выбранным id. На экране под SpectraView отображается название, тэг, дата создания эксперимента. Декодируется изображение и при помощи SpectraHelper инициализируется другой принцип работы с элементом. Устанавливается, что эксперимент имеет изображение, а значит имеет и различные графики, изображение передается, отключаются точки кадрирования, с бд принимаются сохраненные настройки отображения графиков. Изображение подтягивается или растягивается до размеров элемента и отрисовывается.

Отображение завершенного эксперимента показано на рисунке 35.



Рисунок 35 – Экран отображения эксперимента

По кнопке отображения открывается диалоговое окно на рисунке 36.

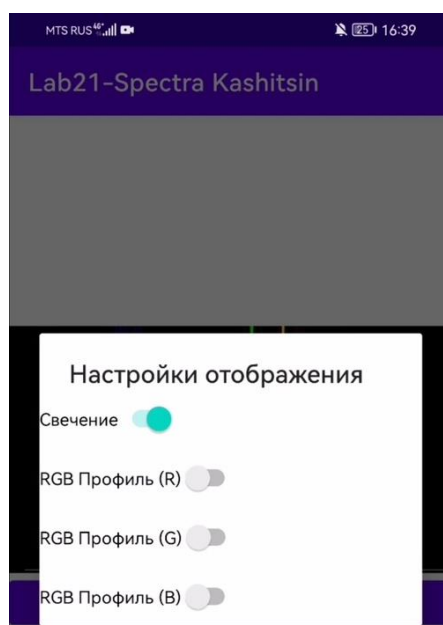


Рисунок 36 – Диалоговое окно отображение графиков

Для эксперимента существует два графика: график освещения (где пик – самые яркие пиксели) и график цветового профиля (с красными, зелеными, си-

ними линиями). Когда изменяется значение переключателей, изменяется значение и в локальной базе данных. Принцип отрисовки графиков один, за исключением, что цветовой профиль был разбит на отдельные 3 графика. Если был включен какой-то из графиков, то отправляется соответствующий запрос и сервер возвращает массив набора координаты на спектре, и самого значения графика от 0 до 1. Значение также интерполируется, где 1 – вершина SpectraView, 0 – расположение в самом низу. Массивы перебираются, и точка за точкой соединяется, где x у точки – координата спектра, y – значение.

Отображение графика свечения у эксперимента «krypton» показано на рисунке 37.

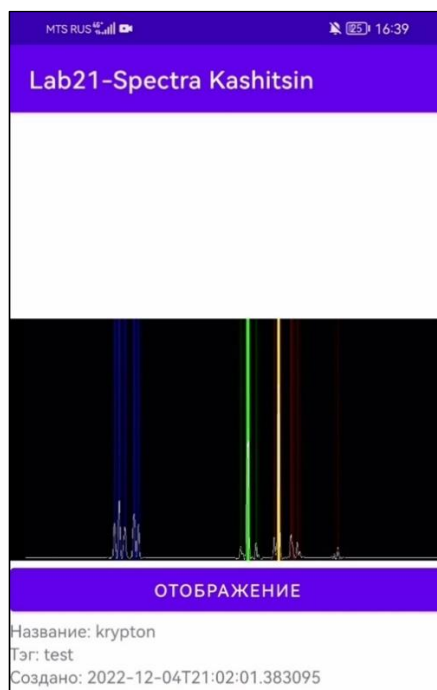


Рисунок 37 – График свечения эксперимента krypton

Отображение графика цветowego профиля у эксперимента «krypton» показано на рисунке 38.

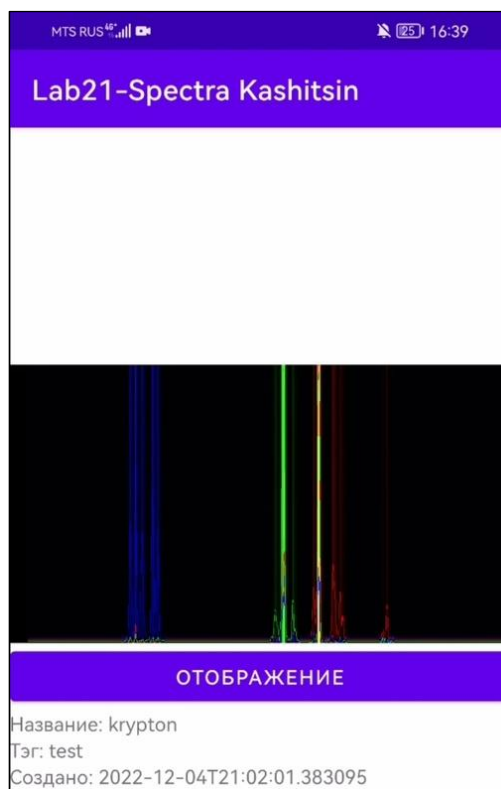


Рисунок 38 – График цветового профиля эксперимента krypton

После выбора элемента для незавершенного эксперимента, он прикрепляется к этому эксперименту (сохраняется).

Перейдем к экспериментам под тегом «string». Зададим второму эксперименту элемент aluminum на рисунке 39.

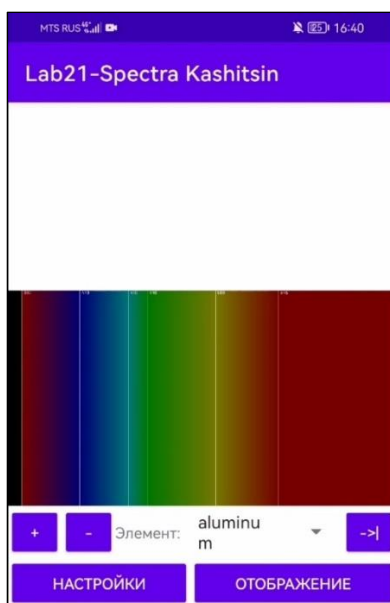


Рисунок 39 – Сохранение элемента aluminum для второго эксперимента

Выберем третий эксперимент, и так как для него не был сохранен элемент, то по умолчанию загружается последний выбранный – aluminum.

Загрузка последнего элемента показана на рисунке 40.

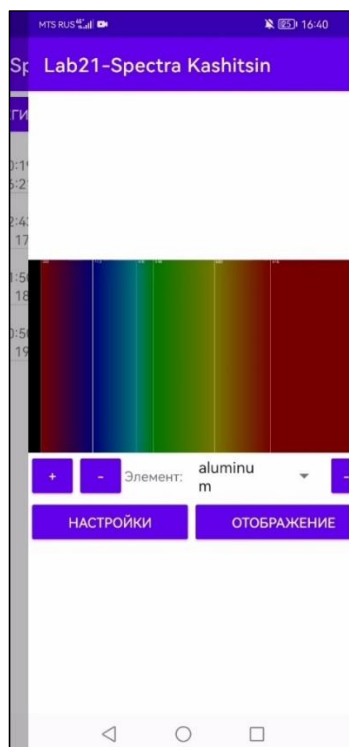


Рисунок 40 – Загрузка последнего элемента

На последнем эксперименте укажем элемент carbon на рисунке 41.

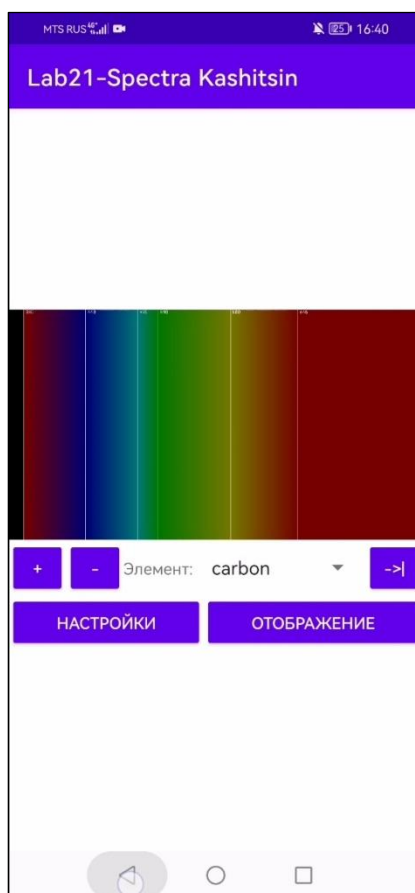


Рисунок 41 – Сохранение элемента для последнего эксперимента

После перезагрузки приложения, открытии последнего эксперимента, отображается выбранный для этого эксперимента элемент на рисунке 42.

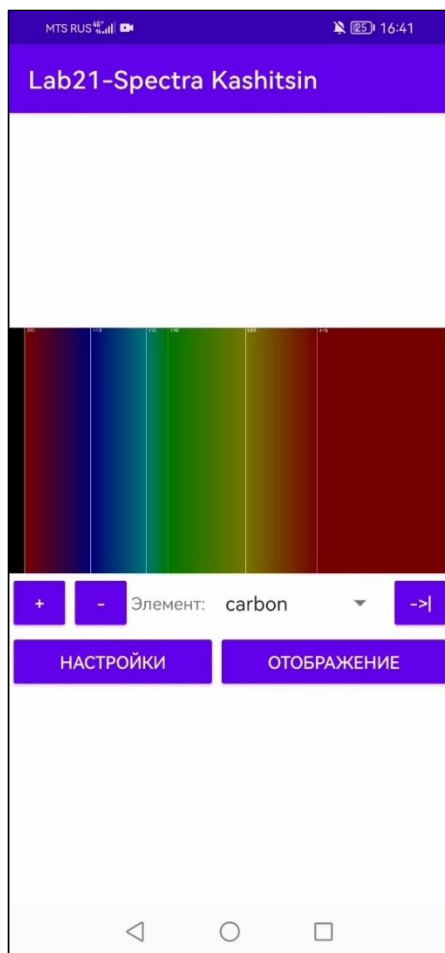


Рисунок 42 – Загрузка элемента для последнего эксперимента

Кнопка «Создать» осуществляет переход к экрану создания эксперимента. На этом экране отображение SpectraView работает в два режима: со спектральными линиями химического элемента, и с готовым изображением (по аналогии с экспериментами, находящимися в обработке и завершенными экспериментами), но с добавлением точек для кадрирования.

Точка – объект класса Dot с полями x и y (ее координаты). Она отрисовывается на элементе с заданным радиусом. Если пользователь нажимает на точку, то запоминаются ее координаты для перемещения. Пользователь может перемещать точки, отпуская, они встают на один уровень (вторая точка справа выравнивается по первой).

Выберем элемент sodium, уменьшим масштаб, и выберем функцию предпросмотра кадрированного изображения. Левая точка – нижний левый угол изображения, правая – нижний правый угол. Верхние точки находятся параллельно на верхней линии SpectraView.

Рисование происходит на холсте и отображается на элементе SpectraView, для снимка с этого элемента использовался отдельный холст на основе нового Bitmap с теми же характеристиками. Рисование на элементе дублируется в отдельный холст, но без рисования точек. Затем на основании отдельного bitmap создается новый размер с заданными точками.

Отображение кадрированного изображения в ImageView показывается после нажатия предпросмотра, а для отправки bitmap сжимается в jpg и представляется в виде массива байтов, которые кодируются в Base64. Стоит учитывать, что у сервера есть ограничение на размер, превышающий 4096 бит.

На рисунке 43 отображено, что эксперимент не может быть отправлен из-за большого размера изображения.

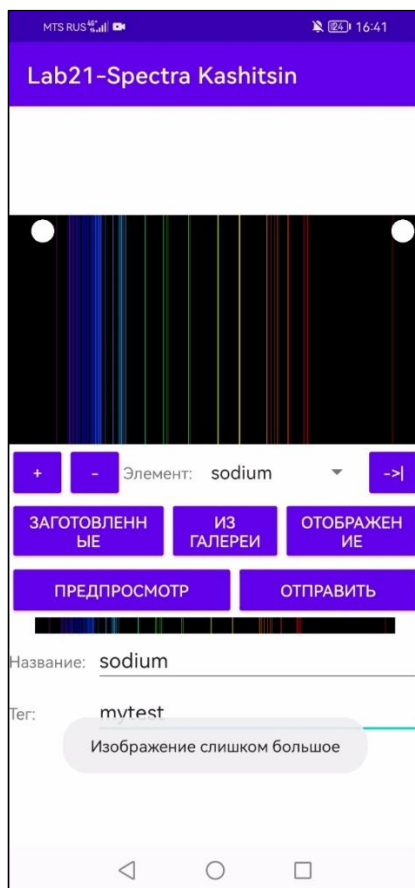


Рисунок 43 – Ограничение на объем изображения

В этом случае можно сузить масштаб и сделать ширину изображения короче. Для отправки эксперимента, помимо изображения, необходимо ввести название и тэг эксперимента.

Отправка эксперимента показана на рисунке 44.



Рисунок 44 – Отправка эксперимента

Теперь созданный эксперимент можно увидеть в списке по тегу mytest на рисунке 45.

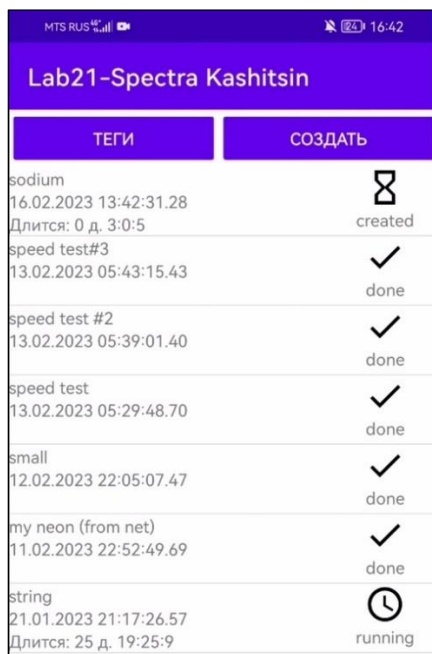
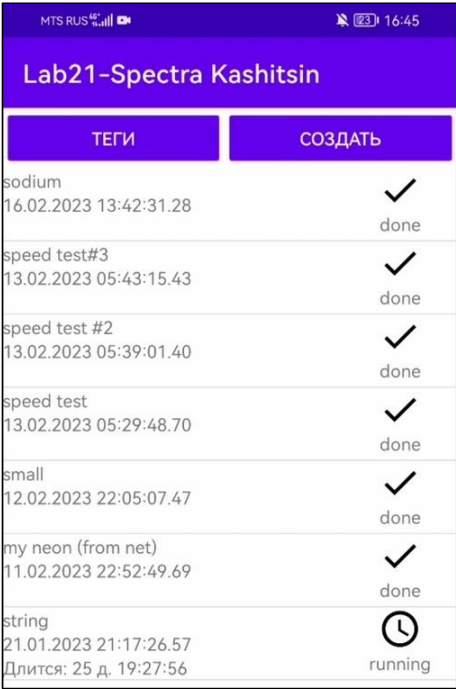


Рисунок 45 – Созданный эксперимент с sodium

Статус эксперимента обновляется в реальном времени, отправкой запроса на получение статусов обрабатывающихся экспериментов. Когда сервер рассчитывает пиксели и создаст все графики, эксперимент перейдет в статус завершенного на рисунке 46.



ТЕГИ	СОЗДАТЬ
sodium 16.02.2023 13:42:31.28	✓ done
speed test#3 13.02.2023 05:43:15.43	✓ done
speed test #2 13.02.2023 05:39:01.40	✓ done
speed test 13.02.2023 05:29:48.70	✓ done
small 12.02.2023 22:05:07.47	✓ done
my neon (from net) 11.02.2023 22:52:49.69	✓ done
string 21.01.2023 21:17:26.57 Длится: 25 д. 19:27:56	🕒 running

Рисунок 46 – Завершенный эксперимент с sodium

Изображение было обработано за 3 минуты. Отображение готового эксперимента показано на рисунке 47.

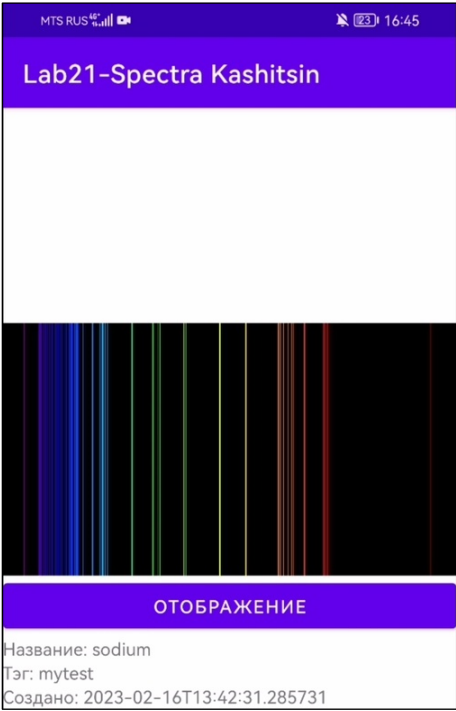


Рисунок 47 – Отображение завершенного эксперимента с sodium

Отображение графика свечения эксперимента sodium показано на рисунке 48.

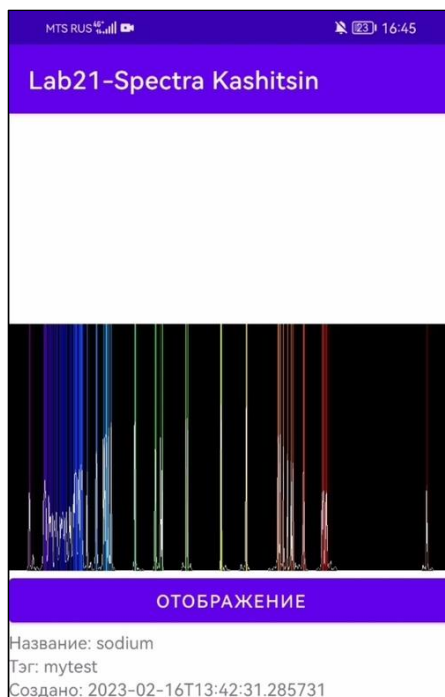


Рисунок 48 – Отображение графика свечения эксперимента sodium

Отображение графика цветового профиля (красный цвет) эксперимента sodium показано на рисунке 49.

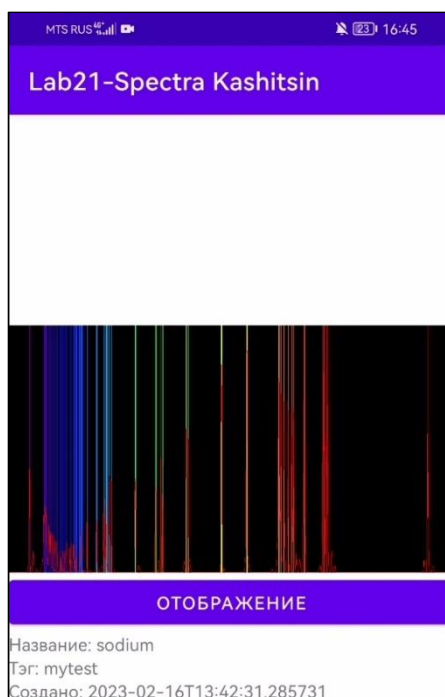


Рисунок 49 – Отображение графика красного эксперимента sodium

Отображение графика цветового профиля (зеленый цвет) эксперимента sodium показано на рисунке 50.

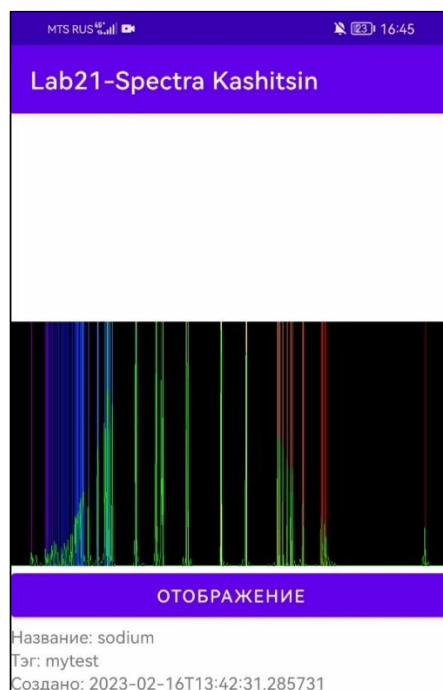


Рисунок 50 – Отображение графика зеленого эксперимента sodium

Отображение графика цветового профиля (синий цвет) эксперимента sodium показано на рисунке 51.

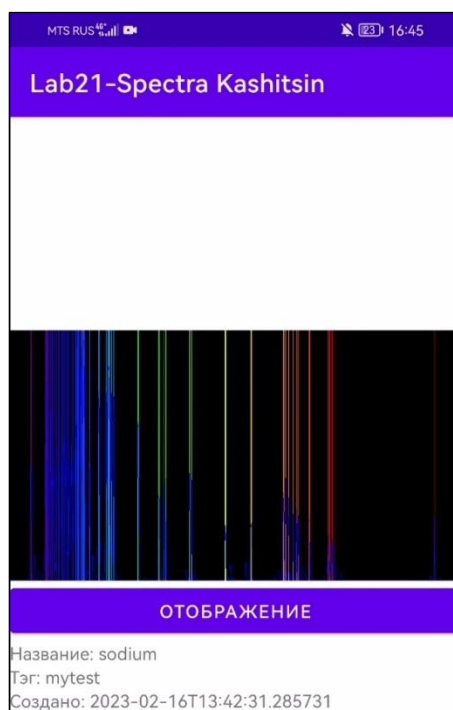


Рисунок 51 – Отображение графика синего эксперимента sodium

На экране создания эксперимента можно использовать готовые изображения.

Отображение списка заготовленных изображений показано на рисунке 52.

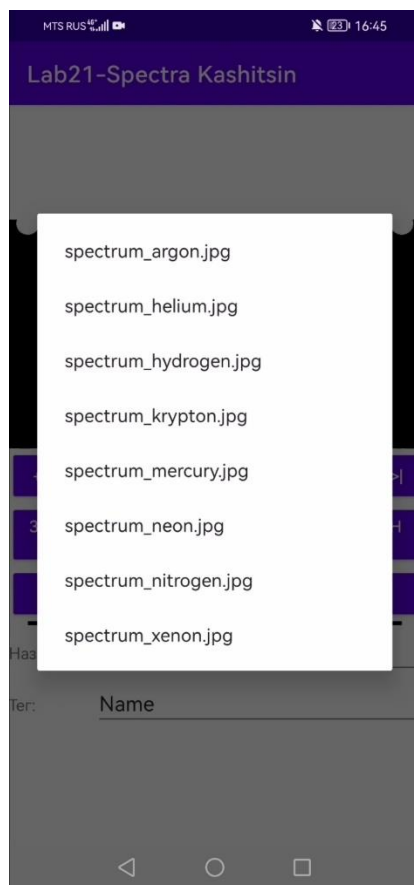


Рисунок 52 – Список заготовленных изображений

Отображение заготовленного изображения spectrum_helium.jpg на рисунке 53.

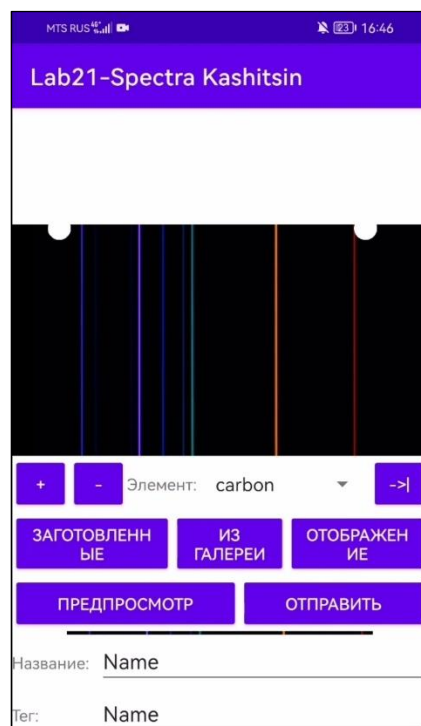


Рисунок 53 – Отображение spectrum_helium.jpg

По нажатию на кнопку для выбора изображения из галереи открывается системный экран выбора изображения на рисунке 54.

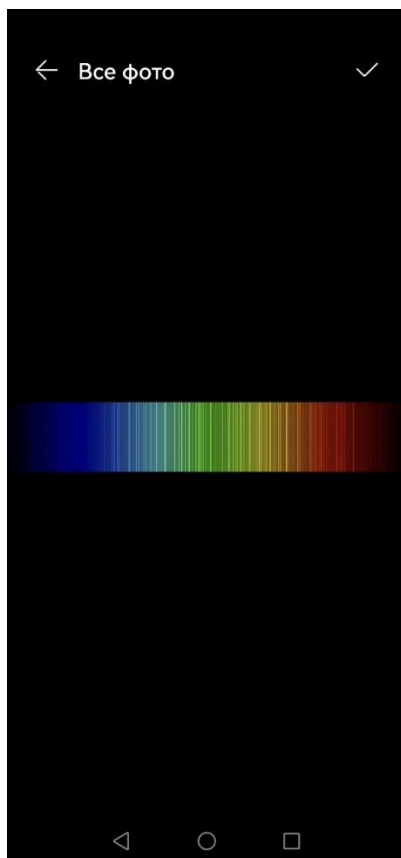


Рисунок 54 – Выбор изображения из галереи

Отправление изображения хепон из интернета показано на рисунке 55.

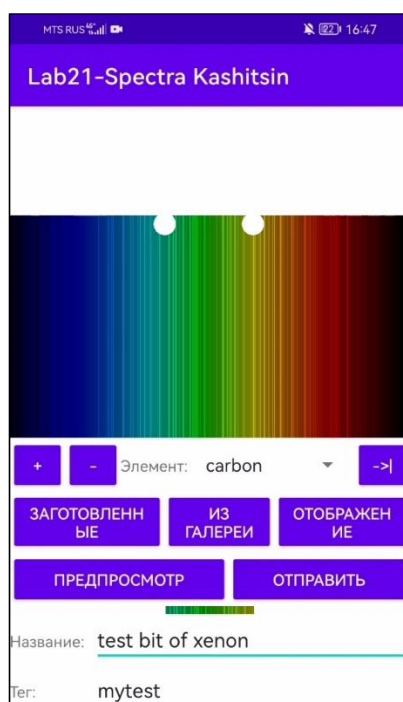


Рисунок 55 – Отправление изображение из галереи

Отображение графика свечения эксперимента с изображением из галереи показано на рисунке 56.

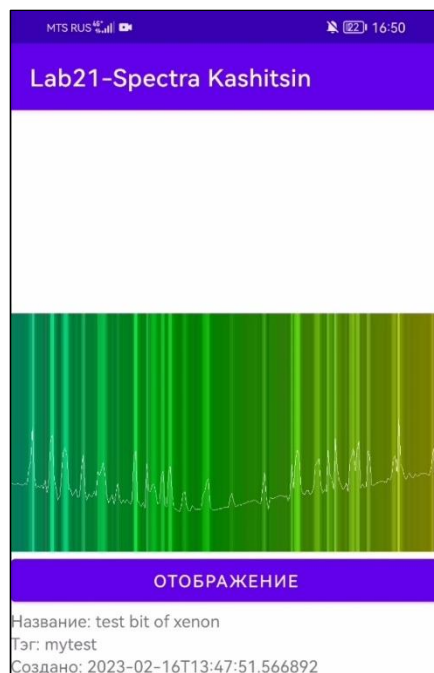


Рисунок 56 – График свечения эксперимента из галереи

Отображение графика цветового графика эксперимента с изображением из галереи показано на рисунке 57.

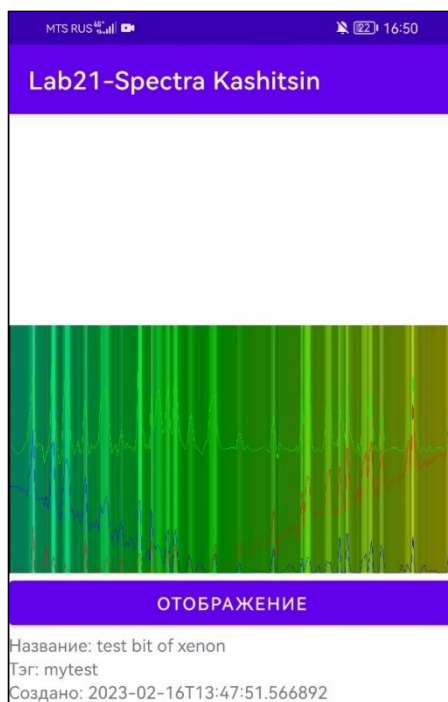


Рисунок 57 – График цветового профиля из галереи

ИТОГ РАБОТЫ

Ссылка на репозиторий с готовым проектом: https://git.spbcoit.ru/git/g493-13-kashitsin/lab30_spectra_kashitsin_493

Альтернативная ссылка (github.com): [aza1rat/LabSpectra](https://github.com/aza1rat/LabSpectra)