Проект: Нативный модуль фото-видео редактора для приложения на React Native

. Тех. задание

1. Фотовидео редактор

Детально задача в следующем. У нас в разработке мобильное приложение с социальными механиками, в котором пользователи могут загружать фото или видео в свою ленту.
Для работы с фото или видео нужны редакторы. Модули на React Native или те опенсорсные, что есть в сети, не оправдали себя и нужны нативные модули, которые интегрируются в наше приложение на React Native .

Интеграция с React Native:
https://medium.com/mindorks/create-native-modules-for-your-react-native-android-app-cfe5ea7230f4
https://medium.com/wix-engineering/creating-a-native-module-in-react-native-93bab0123e46

Видео-редактор
https://github.com/shahen94/react-native-video-processing
Есть такой проект, но у нас не получилось его запустить и интегрировать. Возможно есть смысл попробовать его допилить и взять за основу.

Описание модулей:

На вход модуля приложение на React Native отдаст исходное пользовательское фото или видео: с камеры устройства, из галереи или интернета. Это некий стандартный формат из общепринятых на мобильных устройствах или в интернете (для изображений: png, jpeg и gif(первый кадр); на выходе только JPEG. Видео: mp4, webm. На выходе только mp4. Предусмотреть базовый набор распространенных кодеков, например для некоторых устройств есть проблемы с кодированием в h.264)

На выходе отдается обратно в приложение ReactNative уже обработанное фото или видео, с примененными операциями обрезанное как по размеру, так и по длительности (если речь о видео), и приложение уже отправляет результат на сервер. Все манипуляции с фото и видео максимально, насколько возможно, перенести в приложение и разгрузить бэкенд. Для быстрой отправки на сервер предусмотреть настройки степени сжатия для JPEG и настройки для снижения веса и качества видео (уменьшать битрейт до приемлемого, fps если 60 и прочие, снижать до 30 например)

Функционал:

Пользовательский интерфейс полностью на стороне React Native. От нативного модуля нужны только непосредственно операции с фото и видео.

Фото-редактор
scale & crop — масштабирование и обрезка,
rotate — поворот,
flip — зеркальное отражение как по вертикали так и по горизонтали.
Так как изначально пользователи практически всегда передают изображение избыточного размера, предусмотреть одновременное применение параметров масштабирования, поворота и отражения.

Если это поможет производительности, возможно сразу при получении исходного изображения от пользователя, уменьшать его в некий достаточный размер, который будет обрабатываться быстрее.
Должна быть возможность возврата в исходное состояние картинки (в то что загрузил пользователь или к первоначальному достаточному размеру, из параграфа выше, если такой функционал поможет в производительности)

Примеры такого редактора:
Онлайн редактор. По функционалу тут то что надо и можно использовать как рефренс: https://pqina.nl/doka/#top

Видео-редактор
scale & crop — масштабирование и обрезка,
flip — зеркальное отражение как по вертикали так и по горизонтали.
Так как изначально пользователи практически всегда передают видео избыточного размера, предусмотреть одновременное применение параметров масштабирования, поворота и отражения.

Если это поможет производительности, возможно сразу при получении исходного видео от пользователя, уменьшать его в некий достаточный размер, который будет обрабатываться быстрее. (снижать fps битрейт, масштабировать размер)
Должна быть возможность возврата в исходное состояние видео, чтобы заново начать редактирование, если что-то не понравилось.(в то что загрузил пользователь или к первоначальному достаточному размеру, из параграфа выше, если такой функционал поможет в производительности)

cut timeline — возможность выбора фрагмента видео (десятисекундный отрезок), интерфейс как в Instagram — получение кадров из видео и расположение их на таймлайне, возможность двигать границы отрезка.
Пример: https://github.com/shahen94/react-native-video-processing
А также редакторы в приложениях Instagram, TikTok и тп.

Для очень длинных видео. Например если вставят фильм, лента с кадрами будет очень большой и выемка такого количества кадров займет очень много времени, поэтому делаем в несколько шагов, когда сперва выбирается например полчаса, потом 5 минут, потом 10 секунд, например.
Посмотреть по производительности можно ли это делать не обрезая видео на промежуточных шагах, а только ограничивая время воспроизведения и отдавая кадры из заданного отрезка.
Ограничение и интервал получения кадров задается в запросе к редактору из React Native.

Что ожидаем

Нужна примерная оценка по времени (**проект срочный, поэтому сразу лучше указывать максимально сжатые сроки, в которые вы способны сделать подобную задачу**) и стоимости отдельно каждого из модулей (фото и видео).
Если есть известные Вам ограничения для реализации функционала, напишите, возможно получится что-то упростить, перенести на следующий этап или отменить.
Также напишите возможные способы оплаты.
По результатам работы нужно будет предоставить исходный код, описание ключевых классов и инструкцию как запускать компиляцию модуля и API для взаимодействия с приложением на React Native. Если это влияет на стоимость, учтите в оценке.

Оплата по договору (возможен вариант разбиения общей суммы на несколько частей пропорционально функциональным частям и оплата по мере готовности каждой)

Планы на будущее

Не входит в текущую задачу но в будущем необходимо
Фильтры и работа с цветом: яркость, контраст, насыщенность. На всё фото и видео.
Сейчас это считать не надо, но в случае успешного выполнения задачи, планируется дополнять и развивать функционал. Это добавление масок, надписей, стикеров.
Интерактив: голосования, лайки. Примерно как в Instagram Stories и подобных.
Также возможна постоянная удаленная работа.

API взаимодействия модулей редактора и приложения

Необходимо реализовать 2 отдельных нативных компонента:
1. Фото компонент (ImageEditView) — умеет отображать фото и редактировать его
2. Видео компонент (VideoEditView) — умеет отображать видео и редактировать его

Размеры компонентов должны либо растягиваться автоматически по родительскому компоненту либо пропсами задавать размеры.

Фото компонент (ImageEditView)

Props:
path — uri на локальный файл телефона или url на файл на нашем сервере
bgColor — цвет которым заполнять пустое пространство на результирующей картинке.

Event:
onLoad — отрабатывает, когда компонент загружен и картинка отобразилась

изначально переданная картинка отображается по центру компонента пропорционально уменьшенная, чтобы бОльшая сторона умещалась в компонент (Contain)

Все методы возвращают промисы, которые отработают, когда операция завершится
Так же все методы сразу же все изменения применяют внутри нашего компонента

Methods:
rotate — поворот фото. Тип данных Int - угол поворота в градусах 0-360
flipH — отражение по горизонтали (без параметров)
flipV — отражение по вертикали (без параметров)
getInfo — получение информации об изображении: высота ширина size: {width, height}
rollback — возврат на исходное состояние картинки

transformImage - scale & crop & rotate
{
x,y: Float - смещение картинки относительно верхнего левого угла
width,height: Int - желаемый размер результирующей картинки
scale: Float - множитель увеличения картинки
angel: Float - угол поворота
}
функция трансформации картинки состоящая из:
— создание холста указанного размера (width,height)
— масштабирование исходной картинки на заданный коэффициент (scale)

— перемещение картинки на координаты (x y)
— поворот картинки на заданный угол (angle)

Возвращает ссылку на сгенерированную картинку формата JPEG во временном хранилище на устройстве

Видео компонент (VideoEditView)

Props:
path — uri на локальный файл телефона или url на файл на нашем сервере
bgColor — цвет, которым заполнять пустое пространство на результирующем видео
volume — громкость от 0 до 1
repeat — boolean начинать ли с начала видео, когда закончится
paused — boolean остановка видео
progressUpdateInterval — указывается в миллисекундах как часто должен отрабатывать колбек onPlayerProgress (если -1 то колбек не срабатывает)

Event:
onPlayerLoaded — колбек который срабатывает, когда видео загрузилось и начало играть
onPlayerProgress — колбек который срабатывает, когда видео играет с заданным интервалом (progressUpdateInterval) возвращает текущую миллисекунду проигрывания видео (только если видео играет)

Изначально переданное видео отображается по центру компонента пропорционально уменьшенное, чтобы бОльшая сторона умещалась в компонент (Contain)

Все методы возвращают промисы, которые отработают, когда операция завершится. Так же все методы сразу же все изменения применяют внутри нашего компонента

Methods:
flipH - отражение по горизонтали
(без параметров)
flipV - отражение по вертикали
(без параметров)

getInfo - получение информации о видео size: {width, height}, duration, frameRate, bitrate, codecdata

rollback - возврат на исходное состояние видео

seek (Int) - перемотка на указанную миллисекунду

getPreviewTimeline ({
timeStart, timeEnd, (Int) в миллисекундах отрезок видео

count - количество кадров
width,height высота и ширина нужных картинок превью
}) функция которая генерирует превью на заданном интервале и возвращает массив картинок (ссылку на хранилище или base64 — реакту одинаково)

trim ({
start, end - (int) миллисекунды начала и конца
}) функция обрезки видео по длине

transformVideo -
{
x,y: FLoat - смещение видео относительно верхнего левого угла
width,height: Int - желаемый размер результирующего видео
scale: Float - множитель увеличения видео
}
функция трансформации видео состоящая из:
— создание холста указанного размера (width,height)
— масштабирование исходного видео на заданный коэффициент (scale)
— перемещение картинки на координаты (x y)
Поворота видео на произвольный угол не предусматриваем в этом релизе. Только обрезка и масштабирование

Возвращает ссылку на сгенерированное видео формата mp4 во временном хранилище на устройстве. По умолчанию кодек H.264. Видео должно генерироваться и проигрываться на большинстве современных устройств. Предусмотреть проверку и переключение на альтернативный поддерживаемый кодек (H.263 например)
Нужно обсудить