

Техническое задание

Разработка социального сервиса “Amare”

Введение	3
Назначение и область применения	3
Общее описание	3
Краткая сводка возможностей	3
Технические требования	3
Приложение IOS	3
Приложения Android	4
Бекенд	4
Описание приложения	4
Список разделов клиентского МА	4
Регистрация и Авторизация	4
Подтверждение личности	6
Специальные требования	8
Аккаунт пользователя	9
Remote Configs	10
Контент создаваемый пользователями	10
Live Story	11
Party	12
Plans	13
Основное рабочее пространство МА	14
Feed Screen	15
Правила отображения контента	15
Список пользователей	15
Список пользователей представлен горизонтальным списком вверху экрана.	15
При скроле ленты ивентов вниз, данный список прячется вверх, при сколе списка ивентов вверх, список появляется поверх ленты.	15
Из списка должны быть исключены пользователи:	15
Пользователи online	15
Список ивентов	16
Из списка ивентов должны быть исключены ивенты:	17
Карточка ивента	17
Архив контента пользователя	18
Весь контент пользователя - Sandwich	19
INVITE - Приглашение пользователя	19
Карта	20
Коммуникация пользователей	22
Inbox	22
Сортировка списка чатов	24
Сортировка списка запросов и уведомлений	24
Chat Room	24

Сортировка сообщений в чате	25
Удаление чата	25
Блокирование пользователя и жалобы	25
Настройки	26
Обратная связь	26
Удаление собственно аккаунта	27
Типы пользователей МА	27
Newbie	27
Member	28
Escapee	28
In-App Purchases	28
Проверка статуса	28
PayGate - покупка подписок	28
Restore Purchases	32
Уведомления	32
Нативные системные Push уведомления	32
In-App уведомления	33
Аналитика	33
Подключаемые сервисы	33
Список ивентов	34
Список клиентских ивентов для Google Analytics и Amplitude:	34
Список серверных ивентов только для Amplitude:	36
Backend	37
Валидация чеков покупок	38
Abuses	38
Административная часть	38
Разделы	38
Users	39
Settings	39
System Configs	39
Remote Configs	39
Content	40
Users	40
Content	41
Unit	41
Фоновые сервисы на сервере	42
Регулярная проверка Membership	42
Чистка пользователей	42

Введение

Назначение и область применения

Сервис предназначен для предоставления пользователям нового способа для знакомств. Пользователи системы являются пользователями мобильных приложений (далее mobile application или “**МА**”) на базе iOS или Android.

Общее описание

Сервис представляет собой клиент-серверное приложение типа “социальная сеть для знакомств и общения”. Клиентская часть представлена **МА** на базе iOS и Android сообщающаяся с бэкендом через его API или SDK *(в зависимости от принятого решения по бекенду и БД)*.

Краткая сводка возможностей

МА позволяет пользователям:

- Зарегистрироваться
- Загрузить и хранить свой контент (текст, изображения, видео)
- Находить контент других пользователей
- Контактировать и общаться с другими пользователями внутри приложения (чаты)
- Отправлять или делиться своим контентом с другими пользователями
- Совершать внутренние покупки, дающие доступ к скрытому контенту или функционалу
- Прочее.

Технические требования

Платформа работы приложения: iOS / Android

Приложение IOS

Совместимость с ОС : IOS 10.3 и старше

Поддержка устройств: iPhone 5S+, iPad Air 2,

Верстка iPhone: Портретная только

Верстка iPad Air 2: Портретная только

Приложения Android

Совместимость с Android : Android 4.4. и старше

Верстка телефон: Портретная только

Наиболее приоритетные разрешения экранов: **720x1280, 1080x1920, 480x854**

Устройства: Xiaomi Redmi Note 4, Xiaomi Redmi 4X, Samsung J5, J7, S8/S8+, S7, S9, S9+, Google Pixel 2

Бекенд

Бекенд сервиса предлагается по типу Backend as a Service - на базе готовой облачной серверной инфраструктуры для приложения.

Предпочтительное решение: Google Firebase + Google Firestore + Cloud Functions + Cloud Functions и прочие необходимые сервисы Google Cloud

Медиафайлы (фото, видео и проч.) должны храниться во внешнем облачном хранилище файлов (например Google Cloud Storage или Firebase Storage)

<https://db-engines.com/en/system/Amazon+DynamoDB%3BGoogle+Cloud+Firestore>

Описание приложения

Список разделов клиентского МА

Приложению условно можно разделить на следующие функциональные блоки:

1. Регистрация и Авторизация
2. Подтверждение личности
3. Создание основного контента
4. Поиск контента других пользователей - **Feed Screen**
5. Коммуникация и взаимодействия пользователей
6. Онбординг
7. Настройки
8. Покупки продуктов
9. Уведомления
10. Взаимодействие с бекендом
11. Аналитика пользовательского поведения

Регистрация и Авторизация

Logic:

Регистрация пользователя происходит “анонимно” - без использования привязки к email, аккаунту социальной сети или номеру телефона. Пользователь идентифицируется при помощи уникального идентификатора девайса (для Android) или инсталляции (для iOS).

Пользователю необходимо выбрать свой пол (далее “**gender**”), предпочитаемый пол пользователей, которых пользователь будет искать (далее “**target**”), указать свой **Nickname** и загрузить свою фотографию-профиля (далее “**selfie**”), на которой обязательно должно быть опознано лицо стандартными способами мобильной платформы. После этого в облачной

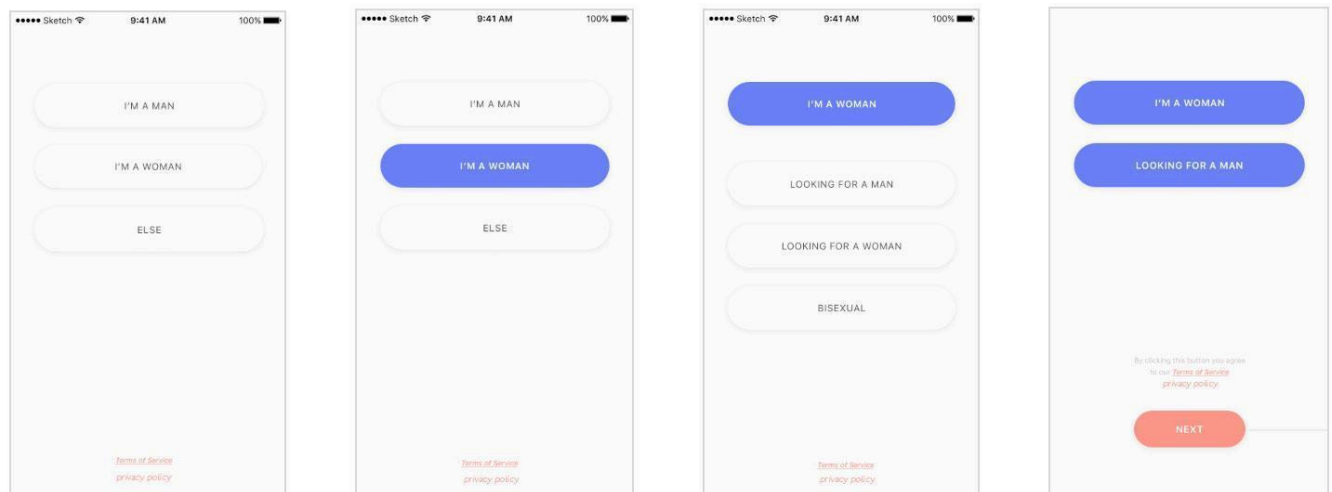
удаленной базе данных (далее **cluod-data-base** или “**CDB**”) создается новый объект пользователя.

Выбор пола хранится локально, до создания объекта в облаке. Если пользователь выбрал **gender** и **target**, но не загрузил **selfie** и закрыл приложение, объект в облаке, соответственно, не создается. Объект создавать только после успешной загрузки **selfie**.

При регистрации пользователь случайным образом получает номер **expGroup** [0, 23], который остается с ним на всю жизнь. Это значение будет использоваться далее для того, чтобы относить пользователя к той или иной выборке для проведения экспериментов (*например см. покупки продуктов*).

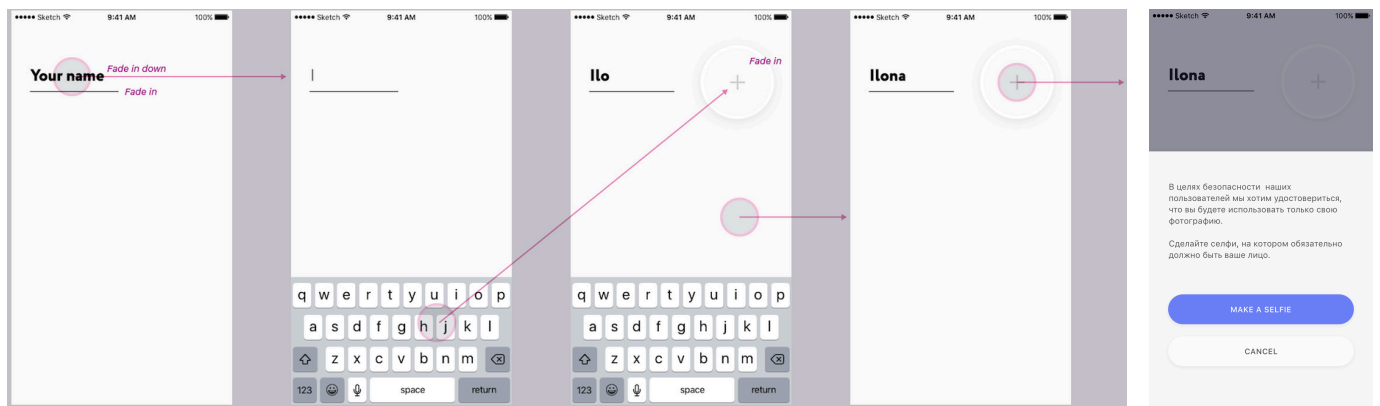
UX/UI:

Сначала на экране появляются три кнопки выбора **gender**. При нажатии на одну из кнопок она подсвечивается - выделяется цветом и после паузы 0.5 сек плавно становится на верхнюю позицию - плавной анимацией длительностью в 0.5 сек. Остальные кнопки выбора пола исчезают эффектом fade-out. Под ней эффектом fade-in появляются кнопки выбора **target**. При выборе **target**, выбранная кнопка подсвечивается - выделяется цветом и после паузы 0.5 сек плавно становится под выбранную кнопку **gender**. Остальные кнопки **target** исчезают эффектом fade-out. При нажатии на кнопку **gender** или **target** повторно, нажатая кнопка становится не выделенной и появляются эффектом fade-in остальные кнопки этого типа, кнопка другого типа, если это **target** съезжает вниз под кнопки выбора **gender**.



Внизу экрана всегда присутствует текстовая ссылки “privacy policy” и “terms of service”, которые открывают http ссылки на web-site с соответствующим контентом.

В момент, когда выбраны и **gender** и **target** появляется кнопка **NEXT**. Ссылки “privacy policy” и “terms of service” занимают другое положение.



После нажатия кнопки NEXT, появляется экран требующий ввести свой Nickname. После введения трех символов справа появляется иконка-кнопка добавления **selfie**.

После нажатия на кнопку добавления selfie появляется предупреждающее диалоговое окно. После прочтения предупреждения пользователь подтверждает и переходит в нативную камеру устройства.

Текст будет изменен. Будет заголовок и ссылка на terms&conditions внизу.

Это сообщение появляется только один раз при первой загрузке selfie. В последствие при смене фото предупреждение уже не показывать.

На входе в приложение на Splash экране, когда объект пользователя получен с сервера, следует проверить поле **status** объекта пользователя. Если пользователь имеет статус **banned** или **suspended**, то он сразу видит экран о том, что забанен и не может двигаться далее. Все что он может отправить письмо в службу поддержки с аппилицией.

Подтверждение личности

Logic:

На данном этапе развития приложения подтверждение личности пользователя происходит путем отправки пользователем второй фотографии (далее "**proof**"), на котором также должно присутствовать лицо, но пользователь должен выполнить одно из случайно выбранных системой заданий. Например, показать определенное количество пальцев, сложить определенным образом руки и тому подобное. Таким образом, **proof** с большой долей вероятности окажется не поддельной фотографией.

Далее модератор админ-панели сервиса сравнит **selfie** и **proof**. И, если окажется, что на обоих фотографиях один и тот же человек, модератор, отметит аккаунт такого пользователя, как подтвержденный. После чего **proof** удаляется из системы.

Для указания статуса selfie есть три иконки для статуса фотографии (есть в дизайне) - красный восклицательный знак - не подтверждена, желтые песочные часы - на модерации, зеленый чек - подтверждена. В случае, если фото не прошло модерацию, значок желтый часов снова меняется на красный восклицательный знак.

Если пользователь с подтвержденным аккаунтом сменил **selfie**, его аккаунт перестает быть подтвержденным и ему предлагается пройти процедуру подтверждения личности повторно.

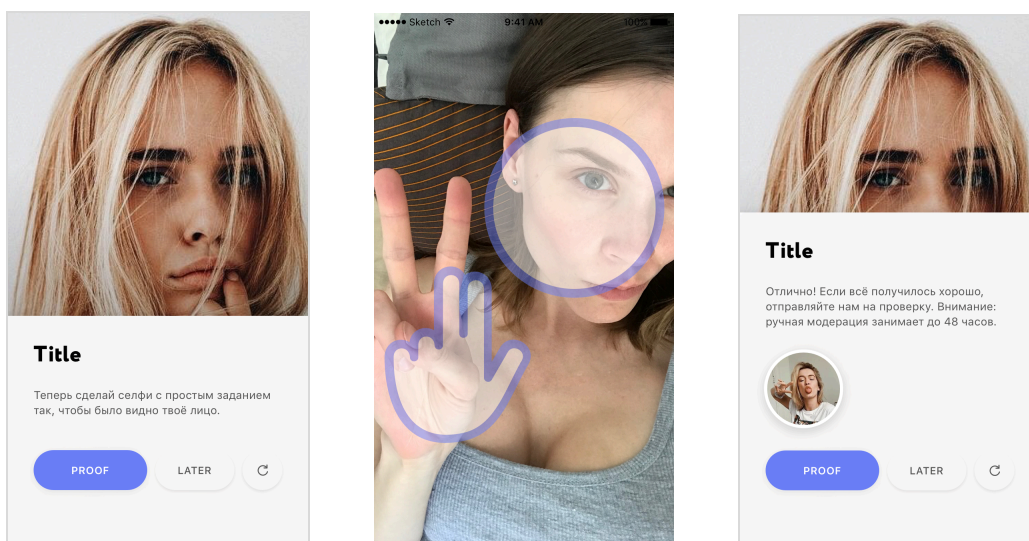
Пользователь может пользоваться приложением без подтвержденного аккаунта, однако, для него будут недоступны некоторые функции приложения.

Пользователь может отправить подтверждение несколько раз - отправить, затем переделать фото и снова отправить, тогда на сервере предыдущее подтверждение “затирается” и дата создания подтверждения обновляется, таким образом модератор всегда будет видеть только последнюю отправленную версию **proff**-фото.

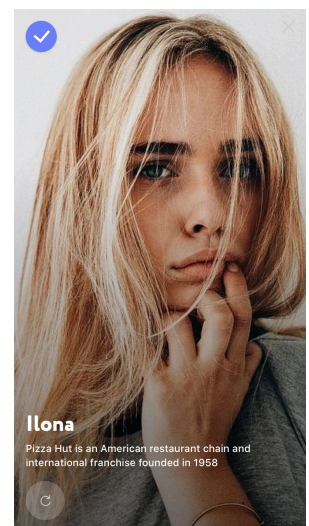
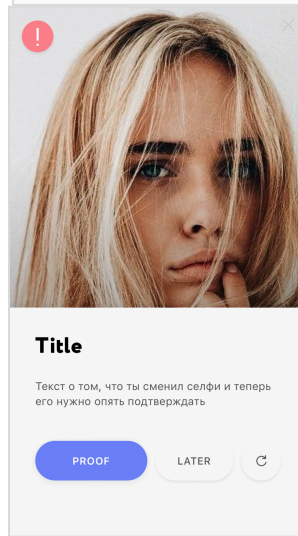
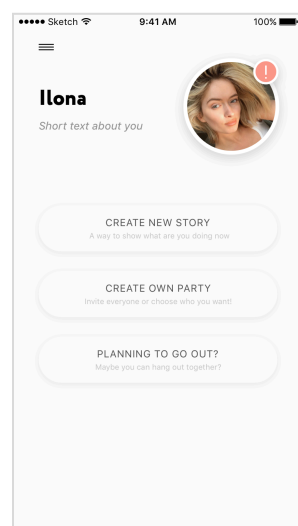
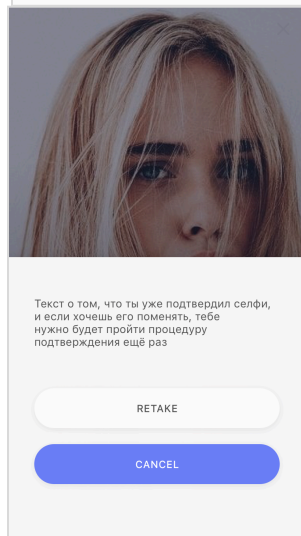
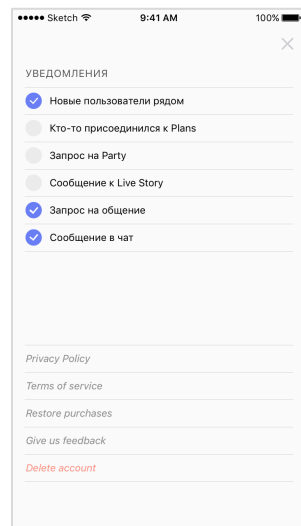
UX/UI:

После успешного создания своего первого **selfie** пользователь видит его на весь экран с поздравлениями и с предложением подтвердить его подлинность. Также пользователь может переснять фото-**selfie** или пропустить этап подтверждения.

Выполнив задание подтверждения - сфотографировавшись с определенным кол-вом пальцев показанных на фоне своего лица, пользователь видит экран, на котором имеется его основное **selfie** и **proof**. Далее он имеет возможность переснять **proof**, отправить **proof** на модерацию или отказаться от подтверждения **selfie**.



В случае отправки на модерацию, пользователь увидит диалоговое окно с предупреждением о том, что модерация займет до 48 часов. И перейдет на экран **profile** со своим **nickname**, фотографией профиля и на этот раз здесь уже будут кнопки создания **events**. Также на экране появляется кнопка **settings**, открывающая экран с настройками. При клике на свое **selfie** пользователь попадет на экран редактирования своего селфи.



На экране редактирования selfie изображено фото профайла пользователя на весь экран, его Nickname и короткое описание, а также кнопки:

- Если **selfie** еще подтверждено модератором - “Confirm photo” и “Retake”
- Если **selfie** уже подтверждено модератором - “Retake”

Текст короткого описания и Nickname можно редактировать, если тапнуть на них. Если **selfie** еще подтверждено модератором и пользователь нажимает на кнопку “Retake”, появляется предупреждение о том, что если пользователь сменит основное **selfie**, то

профайл нужно будет снова подтверждать отправляя подтверждение модератору.

На экране также присутствует иконка соответствующая статусу подтвержденности **selfie** (не подтверждена, отправлена на ревью, подтверждена).

Специальные требования

Сохранение основной фотографии **selfie** должно происходить с обработкой размеров и качества. При сохранении **selfie** нужно сохранить два изображения - **main** и **preview**.

Main изображение должно быть конвертировано в jpeg с максимальным размером большей стороны 900px. Изображение также не должно быть более 2 Mb.

Preview изображение должно быть конвертировано в jpeg с максимальным размером большей стороны 100px. Изображение также не должно быть более 50 Kb.

Все изображения загружаемые пользователями в чатах или при создании ивентов обрабатываются по принципу изображения **Main**.

Аккаунт пользователя

Logic:

Информация, сохраняемая о пользователе, может быть изменена, дополнена или урезана в будущем. Основными свойствами пользователя на данный момент являются:

- **id** - уникальный идентификатор (привязанный к девайсу на Android или инсталляции на iOS)
- **gender**
- **target**
- **nickname**
- **description** (короткий текст о пользователе)
- **location** (координаты)
- **membership** (newbie, member, escapee)
- **memberTime** (timestamp - время до которого валиден его статус **membership** в случае, если он **member**)
- **os** (android, iOS)
- **status** (active, banned, suspended...)
- **expGroup** (случайное в диапазоне [0, 23] - не меняется в течение жизни пользователя)
- **country** (страна в формате ISO 3166-1 alpha-2 - две буквы: UA, PT)
- **region**
- **city**
- ... другие необходимые свойства (видимость на карте, видимость в поиске, включенные нотификации, проверенный аккаунт и т.д.)

** Для того, чтобы было удобно искать пользователя в админ-панели стоит сделать поля (Страна и регион - штат, область или город)*

После того, как пользователь разрешил доступ к своей локации, и необходимо в фоновом потоке определить код страны и регион пользователя (**country, region, city**).

Это делается путем запроса на специальный сервис Google API

<https://developers.google.com/maps/documentation/geocoding/intro?hl=ru> с указанием координат.

Запрос делать с ключом **[gooGeocodeKey]** полученным из **Remote Configs**. Если ключ не указан, то в приложении не применять никакой. Данный запрос будет часто отвечать ошибкой, поэтому следует делать до 10 попыток запросов с паузой длиной равной **attempt + 1 сек**, где **attempt** - номер попытки. Получив ответ в виде json нужно вытащить из **address_components** значения

- **country/short_name** для получения значения **country**
- **administrative_area_level_1/short_name** для получения значения **region**
- **locality/short_name** для получения значения **city**

Если не удалось получить значение **locality**, то выставляется такое же как и **region**

Если не удалось получить значение **region**, то выставляется такое же как и **administrative_area_level_1**

Если не удалось получить значение **administrative_area_level_1**, то выставляется такое же как и **country**

Если ничего не удалось получить за 10 попыток, то нужно вписать значения 0, 0, 0

После того, как значения получены в фоновом потоке нужно произвести апдейт полей пользователя. В том числе объект пользователя в аналитике.

Remote Configs

Сразу после авторизации пользователя следует получить с сервера специальный объект Remote Configs - список значений параметров, которые необходимы для особой работы мобильного приложения, задаваемый администратором в админ-панели системы.

Однако, **MA** может полноценно функционировать не получив каких-либо или всех Remote Configs, используя дефолтные значения. Таким образом, любые параметры из списка Remote Configs лишь изменяют уже заданные по умолчанию в приложении переменные.

См. Remote Configs в описании соответствующего раздела админ-панели.

Таким образом для приложений разных платформ будут разные наборы параметров. Кроме того, они будут разные для разных версий приложений одной и той же платформы.

Список уже известных переменных:

- **[OS]** (String)
 - **[appVers]** (String)
 - **[expGroup]** (Object)
 - **[exampleUsers]** (Boolean)
 - **[reviewMod_1]** (Boolean)
 - **[uxCam]** (Integer)
 - **[maxRadius]** (Integer)

Контент создаваемый пользователями

Logic:

Контент пользователей представлен событиями их жизни, планами или намерениями (все три далее **“events”**), которыми они желают поделиться - сделать информацию об этом доступной для других пользователей сервиса.

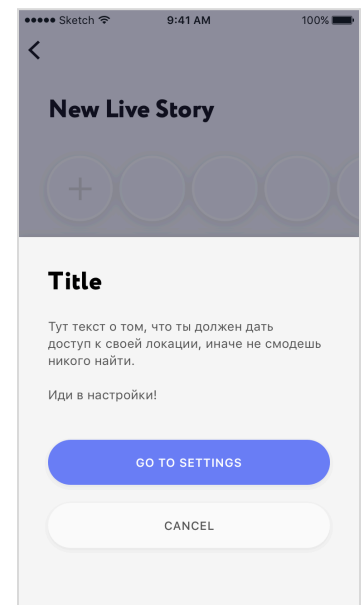
Events могут быть трех типов:

1. **Live Story**
2. **Party**
3. **Plans**

UX/UI:

При нажатии на одну из кнопок создания **events** на экране профиля, происходит проверка наличия разрешения пользователя использовать локацию. Если разрешения нет, показывается диалоговое окно, которое просит дать разрешения. Если пользователь не дал разрешения в этот момент или ранее, появляется экран объясняющий, что без разрешения использовать локацию пользователь не сможет нормально пользоваться приложением. Ему будет доступна кнопка, которая перебрасывает пользователя в настройки девайса, где он может включить разрешения на использования локации, и кнопка “назад”, которая вернет пользователя на экран профиля.

Такое окно должно всплывать и в других местах приложения. Нажатие на Cancel - должно приводить пользователя на экран Profile. Проверку на наличие этих разрешений следует делать (кроме нажатия на кнопки создания **events**) также при развороте приложения после свернутого состояния или при переходе на главный экран приложения - Feed Screen.



Live Story

Logic:

Набор контента (несколько фотографий, короткое описание, комментарии, стикеры, местоположение и проч.), существование которого ограничено по времени (от 1 часа до 1 дня), и позволяет пользователю привлечь внимание других пользователей к образу своей жизни, текущему временному настроению или времяпрепровождению.

Данный тип **event** должен содержать:

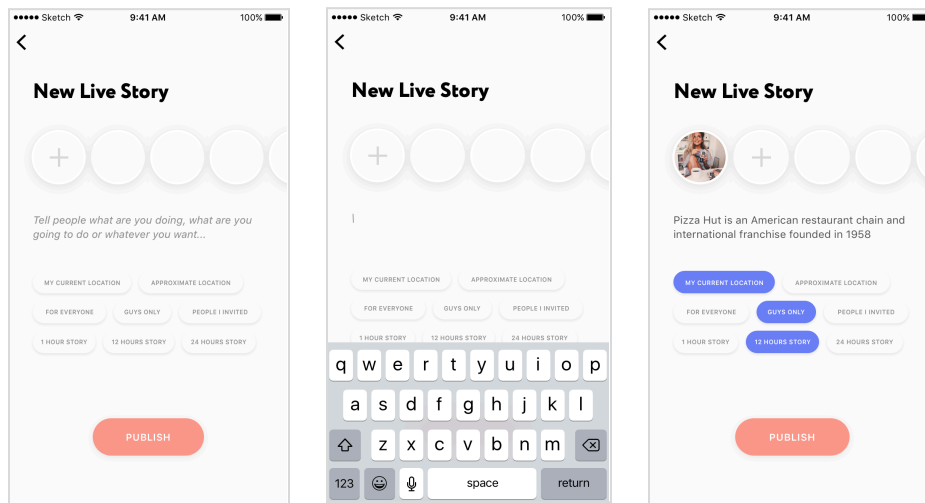
- Минимум одну фотографию или короткое видео (5 сек). (пока только фото)

Самый простой способ создать **Live Story** - это снять одну фотографию при помощи нажатия кнопки внизу основного экрана и затем указав, что эта фотография является **Live Story**. Однако, при желании, пользователь может добавить дополнительную информацию об этом событии.

Дополнительная информация, которую можно добавить:

- Дополнительные фотографии или короткие видео (до 5-ти штук в сумме)
- Короткое текстовое описание происходящих событий
- Стикер, добавленный поверх фото или видео (из набора предоставленных стикеров)
- Локация: можно указать точность показа локации: точное положение или примерное (круг радиусом 1 км на карте)
- Время жизни **Live Story**
- Видимость (все, только противоположный пол - по умолчанию, только пользователи с которыми еще нет чатов)

Все пользователи, которые видят **Live Story** у себя в ленте событий могут написать одно сообщение пользователю - создателю события. Если пользователь ответит на такое сообщение, то между данными двумя пользователями создастся приватный чат.



UX/UI:

Кнопка PUBLISH

изначально должна быть неактивна. После добавления первой фотографии в Story кнопка становится активной. По умолчанию должны быть выбраны следующие опции “my current location”, “24 hours story”, и кнопка соответствующая полу **target**.

Введение описания **не обязательно**. Максимальное кол-во символов: 200 символов.

Также по умолчанию включена кнопка: “не показывать тем, с кем есть чат”

Если фотографий загружено несколько, то их порядок можно менять путем перетаскивания кружков вдоль горизонтального списка

Party

Logic:

Объявление частной вечеринки, мероприятия или митинга каким либо пользователем с целью пригласить других пользователей посетить данное мероприятие, прийти на вечеринку, прийти в гости.

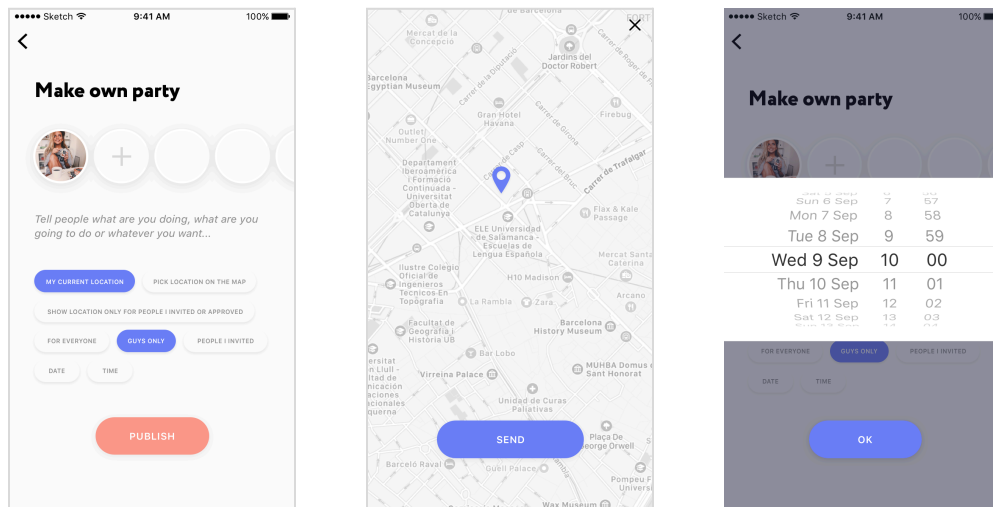
Данный тип **event** должен содержать:

- Минимум одну фотографию или короткое видео (5 сек), описание
- Короткое текстовое описание предстоящей вечеринки
- Точная локация
- Дата и время вечеринки

Дополнительная информация, которую можно добавить:

- Дополнительные фотографии или короткие видео (до 5-ти штук в сумме)
- Стикер, добавленный поверх фото или видео (из набора предоставленных стикеров)
- Отметка, могут ли пользователи видеть локацию сразу или только после утверждения запроса на посещение
- Видимость (все, только противоположный пол - *по умолчанию*, только пользователи с которыми еще нет чатов)

Все пользователи, которые видят **Party** у себя в ленте событий могут отправить запрос на посещение данного мероприятия, а также опционально вместе с запросом написать одно сообщение пользователю - создателю события. Если пользователь ответит на такое сообщение, то между данными двумя пользователями создается приватный чат. Пользователь-создатель события может утвердить запрос или отклонить.



UX/UI:

Кнопка PUBLISH изначально должна быть неактивна. После добавления первой фотографии в Party кнопка становится активной. По умолчанию должны быть выбраны следующие опции “my current location” и кнопка соответствующая полу **target**.

Введение описания **обязательно**. Также обязательно ввести время и дату.

Локацию можно отметить отличную от текущего местоположения пользователя, отметив локацию на карте в ручную. Максимальное кол-во символов в описание: 500 символов.

Также по умолчанию включена кнопка: “не показывать тем, с кем есть чат”

Если фотографий загружено несколько, то их порядок можно менять путем перетаскивания кружков вдоль горизонтального списка.

Plans

Logic:

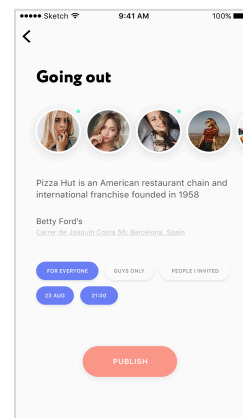
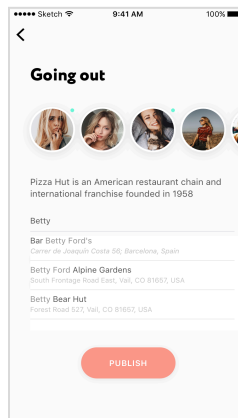
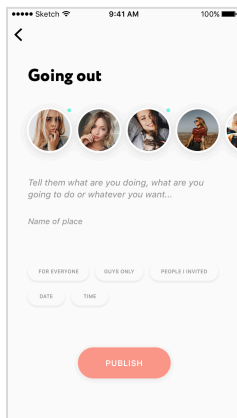
Публикация намерения пользователя посетить какое-либо публичное место (клуб, бар, фестиваль) с целью привлечь других пользователей присоединиться.

Данный тип **event** должен содержать:

- Название места или заведения, которое пользователь собирается посетить. Название не прописывается вручную, но выбирается из существующих. Для этого используется Google Places API. (в будущем это возможно будет Foursquare API)
- Дата и время похода
- Видимость (все, только противоположный пол - *по умолчанию*, только пользователи с которыми еще нет чатов)

Все пользователи, которые видят **Plans** у себя в ленте событий могут написать одно сообщение пользователю - создателю события. Если пользователь ответит на такое сообщение, то между данными двумя пользователями создается приватный чат.

Также другие пользователи могут нажать кнопку **“Join”**, тогда у нажавшего пользователя автоматически создается такой же план похода в это же место на эту же дату и время. Пользователь затем может отредактировать данные (изменить дату/время, видимость или добавить короткий комментарий). Кроме того, все пользователи, которые имеют у себя планы на ту же дату и время с погрешностью 3 часа получают уведомление о том, что этот пользователь также собирается посетить это же место.



UX/UI:

Кнопка PUBLISH изначально должна быть неактивна. После добавления первой фотографии в Party кнопка становится активной. По умолчанию должны быть выбраны следующие опции “my current location” и кнопка соответствующая полу **target**.

Введение описания **обязательно**. Также обязательно ввести время и дату.

Локацию можно отметить отличную от текущего местоположения пользователя, отметив локацию на карте вручную. Максимальное кол-во символов в описании: 500 символов.

Также по умолчанию включена кнопка: “не показывать тем, с кем есть чат”.

Основное рабочее пространство МА

После того, как пользователь загрузил свое **selfie** и создал свой первый ивент, он переходит в основное рабочее пространство **МА**. Далее при запуске приложения повторно, если у пользователя уже есть **selfie**, приложение загружается сразу с этого пространства после **Splash** экрана. Здесь же проверяется наличие разрешения на использование локации повторно.

основное рабочее пространство **МА** представлено пятью экранами. В центре композиции пространства находится центральный экран Feed. Остальные экраны вызываются нажатием навигационных кнопок или соответствующими свайпами - имитирующими “вытягивание” соответствующего экрана слева, справа, сверху или снизу.

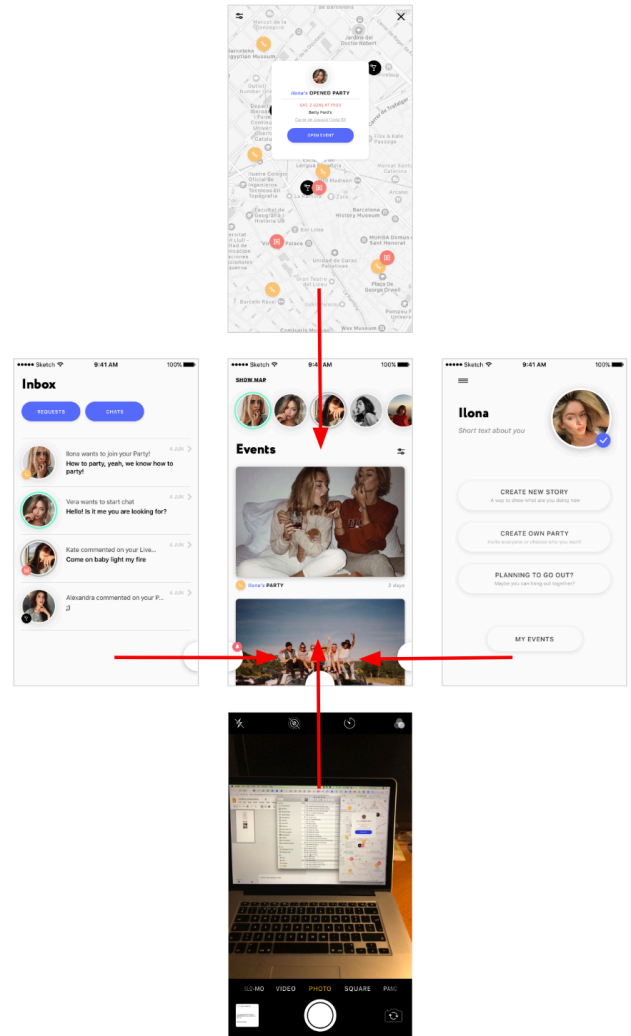
Верхний экран: Map - карта событий

Левый экран: Inbox - запросы пользователей и чаты

Правый экран: Profile - профиль, настройки, создание ивентов, архив ивентов

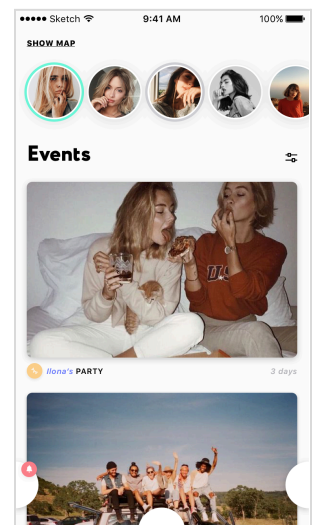
Нижний экран: Камера - создание фото для Live Story

Центральный экран: Feed - список пользователей и список ивентов.



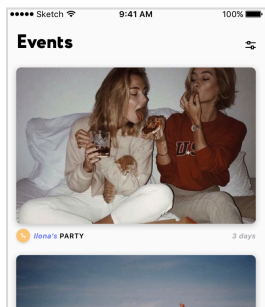
Feed Screen

Данный экран представлен кнопками навигации, вытягивающими дополнительные экраны, горизонтальным списком пользователей, и вертикальным списком-лентой ивентов, а также кнопок вызывающей экран с настройками сортировки и фильтрации ивентов.



Правила отображения контента

Список пользователей



Список пользователей представлен горизонтальным списком вверху экрана.

При скроле ленты ивентов вниз, данный список прячется вверх, при скроле списка ивентов вверх, список появляется поверх ленты.

Из списка должны быть исключены пользователи:

- С которыми уже есть активные чаты
- Заблокированные пользователи

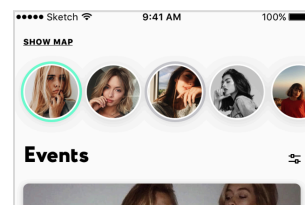
Если другой пользователь заблокировал текущего, то текущий будет видеть заблокировавшего и сможет отправлять ему сообщения и запрос и приглашения, но заблокировавший не будет их получать.

Пользователи online

Пользователи, фотографии которых отображаются в приложении могут быть помечены статусом - **online**.

Online - приложение пользователя на данный момент открыто (**зеленая обводка круга**).

Was online - приложение пользователя закрыто, но последний раз оно было открыто не менее 24 часа назад (**серая обводка круга**).



Пользователи подгружаются пакетами по **50 шт.**

Сортировка: **по удаленности** от пользователя **МА**.

Максимальный радиус поиска пользователей **1000 км.**

Обсуждается!

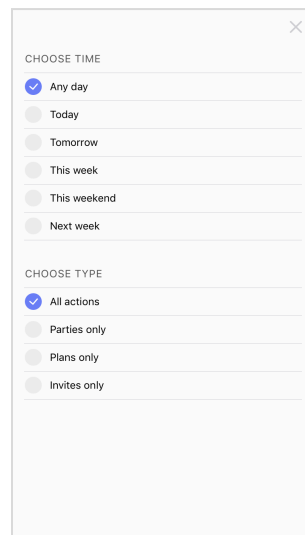
Дополнительные результаты: Если результаты в радиусе 100 км найдено всего менее 50 пользователей, то необходимо произвести дополнительный поиск с радиусом $R = R + 100 \cdot (N \text{ попытки})$, в случае если нашлось менее 50 пользователей, повторить с увеличением номера попытки. Производить попытки с увеличением радиуса до тех пор, пока значение $R(i)$ не достигнет значения 1000км или значения **[maxRadius]** из **Remote Configs**. Если в какой-то из попыток нашлось 50 или более пользователей, то продолжать искать в этом радиусе.

$$R_0 = 0 + 100 \cdot 1 = 100 \text{ км}$$

$$R_1 = R_0 + 100 \cdot 2 = 300 \text{ км}$$

$$R_2 = R_1 + 100 \cdot 3 = 600 \text{ км}$$

$$R_3 = R_2 + 100 \cdot 4 = 1000 \text{ км}$$



Поиск пользователей производится со следующими ограничениями:

M - Male

F - Female

Bi - Else или Bisexual

Первое значение - **gender** пользователя

“->” - это означает кого ищет пользователь

Второе значение - **target** пользователя

“.” результат получаемый при поиске системой

Например: для пользователя с комбинацией (gender_1 -> target_1) будут получены пользователи с комбинациями (gender_2 -> target_2) и (gender_N -> target_N)

M -> F : F -> M - если мальчик ищет девочку, то он будет видеть только девочек, которые ищут мальчиков

F -> M : M -> F

F -> F : F -> F, F -> Bi - если мальчик ищет мальчиков, то он будет видеть только мальчиков, которые ищут мальчиков и мальчиков, которые ищут бисексуалов

M -> M : M -> M, M -> Bi

Bi -> Bi : Bi -> Bi, M -> Bi, F -> Bi

Bi -> F : F -> Bi

Bi -> M : M -> Bi

M -> Bi : Bi -> M, Bi -> Bi, M -> M

F -> Bi : Bi -> F, Bi -> Bi, F -> F

То же правило распространяется на поиск по ивентам, за исключением тех ивентов, которые пользователя установили доступными для всех. Такие ивенты показываются всем независимо от комбинации target/gender.

Список ивентов

Ивенты подгружаются пакетами по **50 шт.**

Максимальный радиус поиска ивентов **100 км**

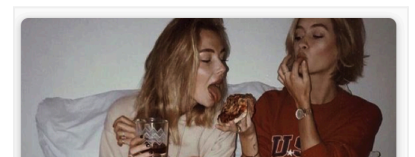
Сортировка:

- Дата создания (наиболее свежие сверху)
- Дата истечения (вверх то, до свершения или истечения которого осталось меньше всего времени)
- Удаленность (с минимальным расстоянием сверху)

Из списка ивентов должны быть исключены ивенты:

- От пользователей, с которыми уже есть активные чаты, если в настройках ивента выбрана опция “не показывать тем, с кем у меня чат”
- Ивенты от заблокированных пользователей
- Ивенты, на которые пользователь отреагировал - написал сообщение, отправил запрос на присоединения или заблокировал и пожаловался

Также пользователь может указать дополнительные правила отображения ивентов в ленте, нажав на кнопку фильтров и перейдя на соответствующий экран. Выбранные правила сортировки сохраняются только на период сессии приложения - выгруженное из памяти приложение возвращает эти настройки к дефолтным.



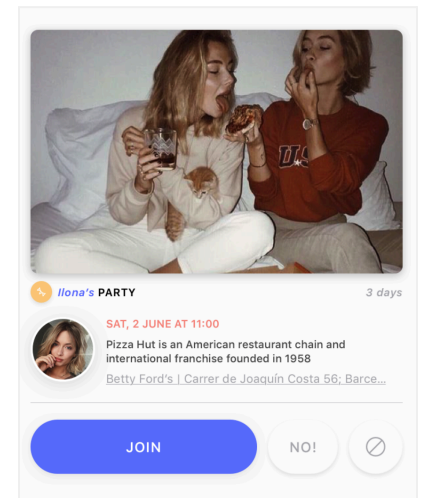
Карточка ивента

Ивент типа **Story** или **Party** представлены прямоугольными горизонтальными карточками с изображением ивента. Если загружено более одной фотографии к ивенту, то отображается первый.

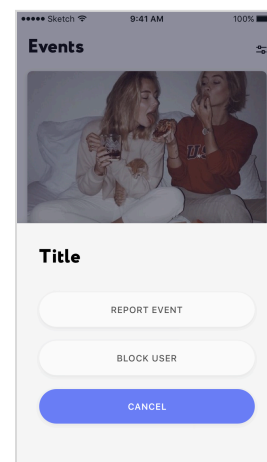
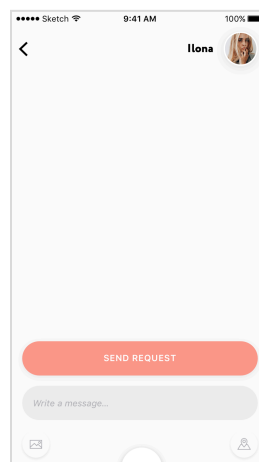
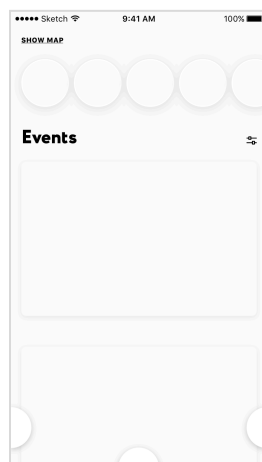
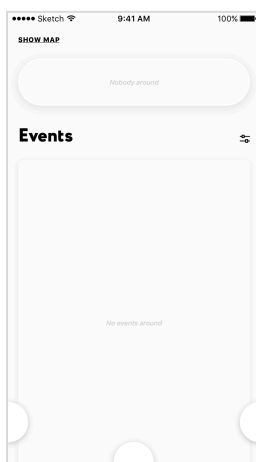
Ивент типа **Plan** представлен карточкой с изображением карты, маркера по середине и название заведения над маркером. Под карточкой написан nickname и тип ивента, а также сколько дней осталось до **Party** или **Plan**.

При тапе на карточку ивента, она анимационно (0.5 сек) “нажимается” и под ней открываются некоторые детали ивента:

- **Selfie** пользователя, разместившего ивент
- Дата и время ивента **Party** или **Plan** или дата создания **Story**
- Описание, если пользователь ввел его (только первые две строки, которые помещаются)
- Адрес, если есть локация и удалось определить адрес или город, если выбрана приблизительная локация, а не точная.
- Кнопки:
 - “Join” для **Party** или **Plan**, и “Send Message” для **Story** (открывают экран создания запроса с опциональным сообщением)
 - “No” - удаляет ивент из списка, пользователь больше не должен видеть этот ивент, но может видеть другие ивенты этого пользователя
 - Кнопка блокирования или жалобы на пользователя - открывается диалоговое окно с выбором.



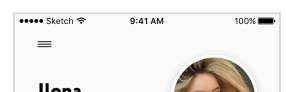
Последующий тап на развернутой карточке откроет окно “**Sandwich**” всего контента данного пользователя. Sandwich откроется в том месте, которое соответствует выбранному ивенту (см. Sandwich - весь контент пользователя)



Развернутая

ивента сворачивается до исходного состояния, когда она прокручивается за пределы экрана при скролле вверх или вниз.

карточка



Архив контента пользователя

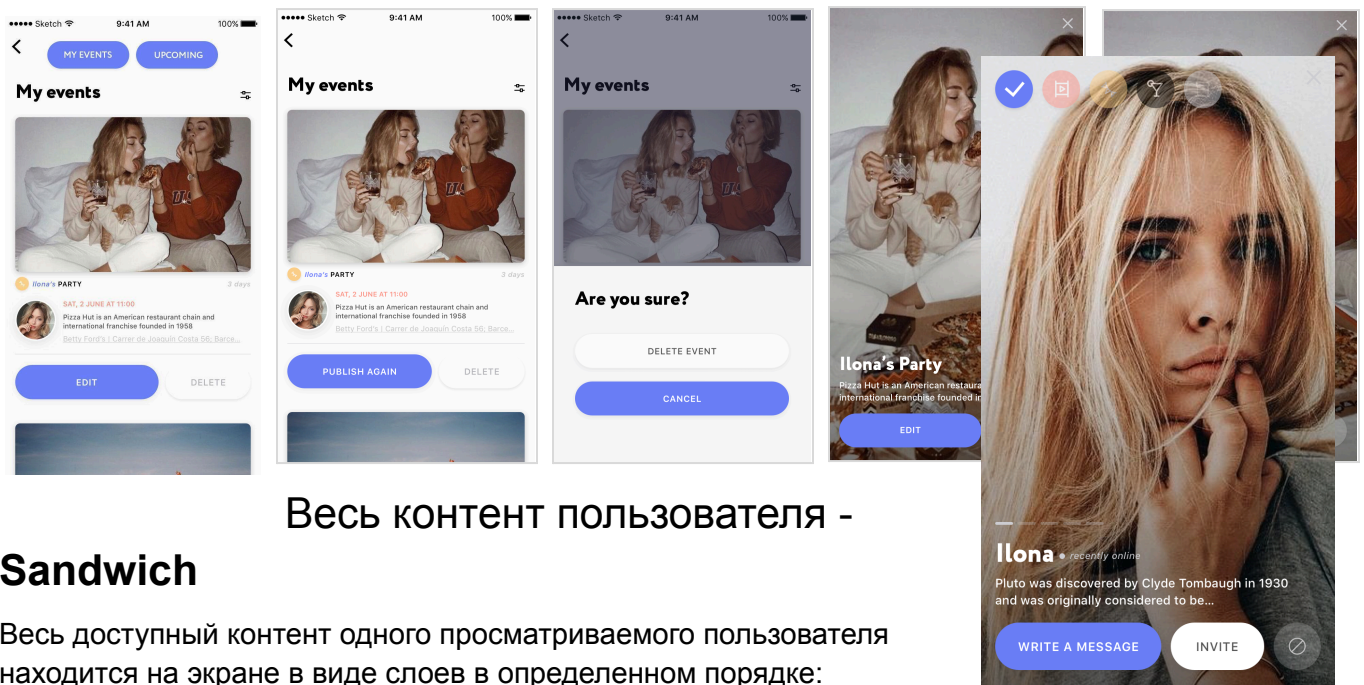
Если у пользователя создан хотя бы один ивент, то на экране **Profile** будет доступна кнопка “**My Events**” и появится доступ соответственно к архиву **events** созданных пользователем ранее.

Экран архива - это лента событий, выполненная аналогично ленте Feed.

Раздел делится на две части: Мои события и события на которые меня пригласили и пользователь утвердил или попросился и его утвердили.

Порядок отображения контента - только хронологический - по дате публикации (или пере-публикации) - свежее сверху. Вся информация и поведение карточек - аналогичное ленте Feed. Кнопки взаимодействия в этих карточках отличаются от Feed. Для действующих Live Story или еще не наступивших Parties или Plans - имеется кнопка EDIT, для архивных ивентов PUBLISH AGAIN. Также для всех типов ивентов есть кнопка DELETE.

Также свой ивент можно открыть на весь экран и с помощью тапов на правой или левой стороне проходить по изображениям вперед или назад. Свайп вправо/влево переводит с одного ивента на другой, Аналогично “**Sandwich**” всего контента пользователя (см. далее).



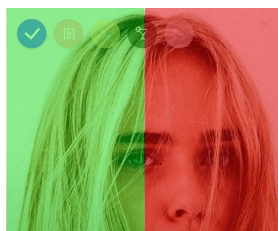
Весь контент пользователя -

Sandwich

Весь доступный контент одного просматриваемого пользователя находится на экране в виде слоев в определенном порядке:

1. **Selfie** пользователя
2. Текущее активное **Live Story**
3. Активные **Parties** - может быть несколько (в порядке отдаления даты проведения - чем больше времени до ивента, тем глубже слой)
4. Активные **Plans** - может быть несколько (в порядке отдаления даты проведения - чем больше времени до ивента, тем глубже слой)
5. Прошлые **Stories** - в хронологической последовательности - свежие выше.

Вверху экрана находятся иконки соответствующие разделам: **Live Story, Parties, Plans, Archive** (старые stories). Если профиль пользователя подтвержден, то первая иконка обозначает это состояние.



При нажатии на эти иконки, происходит переход сразу в соответствующие разделы. В каждом из разделов может быть несколько изображений (например в Live Story может быть 5 фото).

При тапе на фотографии происходит переход вперед (правая половина изображения) или над (левая половина) по слоям сендвича. Прокликаяв все изображения Live Story, следующая фотография будет первой фотографией ближайшей Party пользователя. Plan представлен картой с отметкой заведения и времени, возможно с комментарием пользователя.

На данном экране также должен присутствовать свайп вправо и влево, что должно открывать Sandwich других пользователей, только если сендвич открыт из списка пользователей.

За аналог визуального отображения тапов и свайпов переключения между контентом следует взять Snapchat.

На экране имеется Nickname пользователя, короткое описание, статус онлайнности, кнопки взаимодействия (Write message, Join), Invite, Block/Report.

Возле имени на экране присутствуют черточки обозначающие количество изображений в текущем разделе.

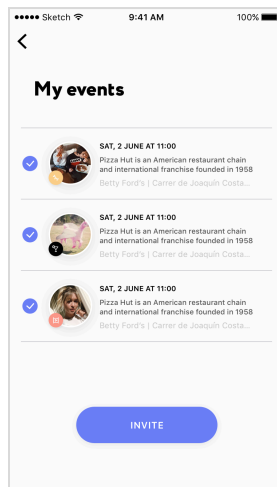
INVITE - Приглашение пользователя

В сендвиче пользователя имеется кнопка INVITE, которая позволяет пригласить пользователя присоединится к **Party** или **Plan**.

Если у пользователя нет ни одного активного **Party** или **Plan** то приглашать другого пользователя некуда, о чем будет соответствующее сообщение. И будет предложено создать **Party** или **Plan**.

Если пользователь решит создать ивент, то откроется соответствующий экран создания ивента, на котором вместо кнопки PUBLISH будет кнопка INVITE.

Однако, ивент все равно создастся и появится в списке ивентов пользователя, но одновременно с этим выбранный пользователь сразу получит запрос-приглашение на присоединение его к ивенту.

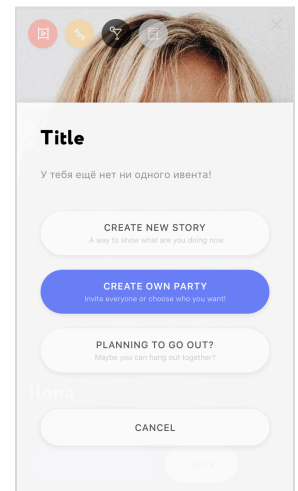


Если у пользователя уже есть один или более **Party** или **Plan**, то появится экран с выбором ивентов, на которые можно пригласить пользователя

Можно выбрать только какой-то один ивент из списка.

Если пользователю уже был отправлен **invite** на какой-то **event**, то такого ивента уже не должно быть в списке из которого нужно выбирать.

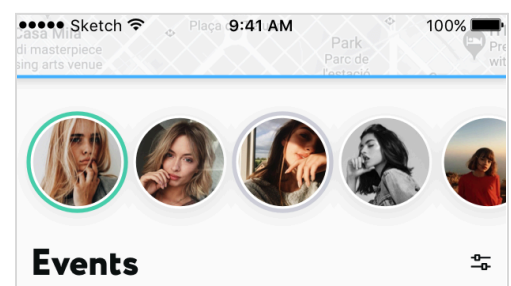
После того, как пользователю будет отправлено приглашение, должно появиться уведомление о том, что приглашение успешно отправлено и кнопка INVITE должна стать неактивной.



Можно приглашать пользователя на какой-либо ивент не чаще, чем 1 раз в 24 часа.

Карта

Карта ивентов расположена в основной рабочей зоне приложения сверху от экран **Feed**. Экран появляется в результате тапа на верхнюю полосу карты вверху экрана Feed или свайпом сверху-вниз - вытягивая эту карту.



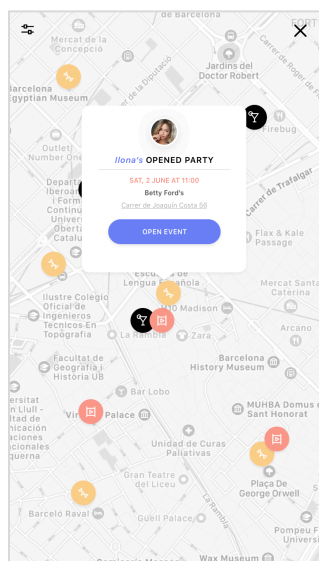
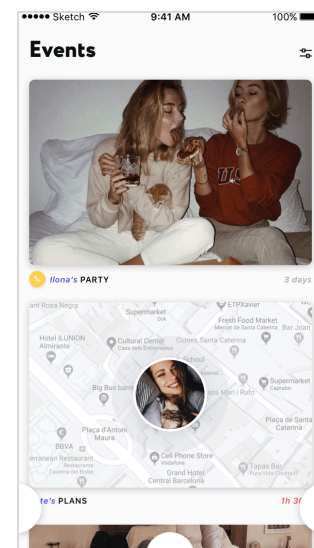
Полоска этой карты - это картинка полученная через static api подрезанная под необходимый размер экрана и обработана перех появлением так, чтобы она стала черно/белой.

Получить картинку карты можно при помощи GET https запроса URL следующего формата:
[https://maps.google.com/maps/api/staticmap?center=\\(lat\),\\(\\(lng\)&scale=2&zoom=\\(zoom\)&size=\\(width\)x\\(\\(height\)&sensor=false&key=\\(StaticMapKey\)](https://maps.google.com/maps/api/staticmap?center=\(lat),\(\(lng)&scale=2&zoom=\(zoom)&size=\(width)x\(\(height)&sensor=false&key=\(StaticMapKey)\)

Карточка ивента типа **Plan** представлена в списке ивентов как карта с круглым **selfie** пользователя по центру. Изображение карты также следует получить как картинку через staticmap, а затем сделать ее ч/б.

Карта центрирована по местоположению пользователя **MA**. На ней нанесены все events, местоположение которых доступно для пользователя **MA**.

- **Story** с отметкой “показывать приблизительное местоположение” - на карте **не указано**
- **Party**, у которого скрыто местоположение для неподтвержденных и не приглашенных пользователей - на карте **не указано**
- **Party**, у которого скрыто местоположение для неподтвержденных, но пользователь MA приглашен на мероприятие - на карте **указано**
- Все Plans - на карте **указаны**



Все ивенты, которые указаны на карте обозначаются при помощи соответствующих иконок. Тап по каждой из них открывает окошко с деталями ивента и кнопкой, позволяющей открыть детали на весь экран - открывается Сэндвич, который не переходит при свайпе на Сэндвич следующего пользователя - здесь только один.

Для ивентов типа **Plan** - если несколько пользователей имеют Plan в одно и тоже место, следует нанести только один маркер на карте и в окошке с деталями, появляющемся при клике, нужно сделать кнопки вперед/назад для пролистывания всех пользователей планирующих посетить это место. (только для **Plans**)

Эффект черно/белой карты: Обсуждается!

Поскольку было принято решение использовать Google Maps и данная карта по умолчанию является цветной, а в дизайне нам хотелось бы иметь ч/б, есть следующее предложение. После того, как пользователь “вытянул” экран с картой, пользователь видит на настоящую цветную карту, но картинку полученную из Staticmap. На самом деле на экране уже есть реальная карта GoogleMap, а сверху на ней лежит Image точно такого же масштаба, что и карта и точно такого же центрирования, так что объекты на карте совпадают. После того, как экран с картой полностью открывается полностью, через 1-1,5 секунды ч/б картинка исчезает медленным фейдом и обнажается цветная точно такая же карта под ней. Это для пользователя должно выглядеть, как будь-то ч/б карта медленно стала цветной. И с этого момента пользователь может работать с жестами уже реальной карты - передвигать и менять масштаб.

Коммуникация пользователей

Пользователи **МА** могут:

- отправлять друг другу сообщения
- запросы на общение
- запросы на присоединение к **events**
- приглашения на свои **events**
- блокировать и жаловаться на других пользователей

Inbox

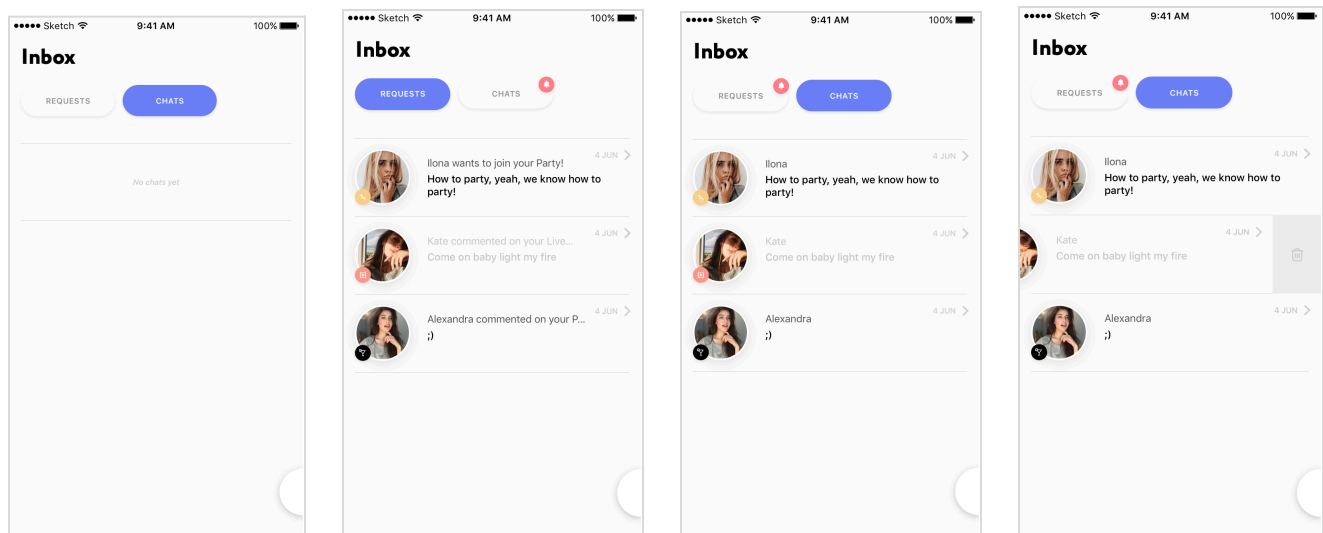
Во вкладке коммуникации **МА** присутствуют разделы:

- входящие сообщения без ответа, запросы и приглашения
- активные чаты

Пользователи могут получать:

- Прямые сообщения со страниц профилей других пользователей
- Сообщения-реакции на **Live Story**
- Сообщения-комментарии к запросу на присоединение к **Party**
- Сообщения-комментарии к присоединению к **Plans**
- Приглашения присоединиться к чьему-то **Plans** или **Party**

Конкретному пользователю другой пользователь может отправить только одно прямое сообщение, одно сообщение на каждый созданный **Live Story**, **Party** и **Plans** и приглашение на каждое из своих **Plans** или **Party**, но не чаще одного приглашения на один ивент в сутки.



UX/UI:

Строка списка входящих **событий** состоит из:

- Аватара (с обводкой зеленым - online, серым - was online, белым - остальные)
- Значок соответствующий ивенту
- Фразе кто, что хочет сделать ([Nickname] + wants + [join your party | to chat | to join your plans | comment to your story])
- Текст сообщения (если есть)
- Дата (если сообщение пришло в течение последних 24 часов, то время вместо даты)

Строка списка входящих **сообщений** состоит из:

- Аватара (с обводкой зеленым - online, серым - was online, белым - остальные)
- Nickname
- Текст сообщения (если есть)
- Дата (если сообщение пришло в течение последних 24 часов, то время вместо даты)

Удаление чата или ивента из списка происходит **свайпом вправо** с подтверждением.

Все сообщения от одного и того же пользователя, присланные для разных events объединяются в одном “**потенциальном**” чате, которые располагается во вкладке **запросов**.

Активный чат между пользователями создается только тогда, когда пользователь, который получил входящее сообщение или запрос или приглашение ответил хотя бы одним сообщением или принял хотя бы одно приглашение или утвердил присоединение хотя бы к одному своему ивенту. Активный чат располагается во вкладке **чатов**.

В активном чате можно писать неограниченное количество сообщений с обеих сторон, до тех пор, пока один из пользователей не заблокирует другого или не удалит чат с ним. (см. *Блокировку пользователей и удаление чатов*)

Когда потенциальный чат становится активным - он переносится из вкладки **запросов** во вкладку **чатов**.

Если у пользователей уже есть активный чат и собеседник присылает запрос или приглашение, то запрос или приглашение уже отображается в активном чате как новое входящее сообщение и чат не переносится во вкладку запросов. Только если активный чат был удален, но собеседник-пользователь не был заблокирован, то в таком случае новый запрос от собеседника поступит во вкладку **запросов**.

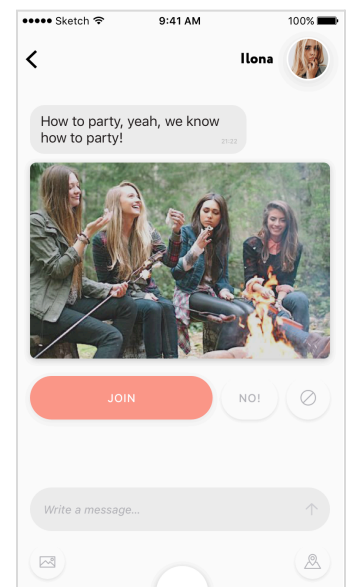
Например:

1 August 2018: У пользователя **A** есть **Live Story** и **Plans**. Пользователь **B**, может отправить одно прямое сообщение пользователю **A**. одно сообщение-реакцию на **Live Story** пользователя **A** и одно для его **Plans**

2 August 2018: У пользователя **A** просрочилась вчерашняя **Live Story**, но он создал новое **Party**. Пользователь **B**, может отправить запрос на присоединение к вечеринке и написал к нему сообщение комментарий. Прямое сообщение уже писать больше нельзя, теперь писать можно только к новым созданным событиям.

Таким образом, пользователь **B** написал пользователю **A** - 4 сообщения.

Пользователь **A** видит все входящие сообщения от пользователя **B** как входящие сообщения и запросы разных типов, но все они находятся в одном “потенциальном” чате в разделе “входящие сообщения без ответа и запросы”.



- Если пользователь **A** отвергнет один из запросов, то остальные входящие от этого пользователя останутся и он может принять одно из них.
- Если пользователь **A** **заблокирует** (см. *Блокировку пользователей*) пользователя **B**, то все входящие от пользователя **B** будут автоматически удалены из списка входящих пользователя **A**.
- Если пользователь **A** откроет “потенциальный” чат с пользователем **B**, там он увидит в хронологическом порядке все его (**B**) входящие сообщения и запросы.
- Если пользователь **A** ответит в таком “потенциальном” чате, то между этими пользователями **A** и **B** создается активный чат.

Сортировка списка чатов

Чаты подгружаются пакетами по **50 шт.**

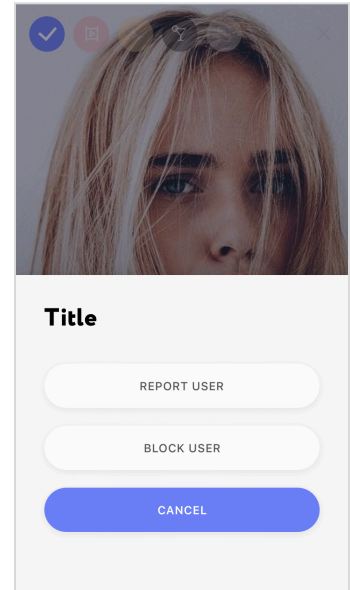
Сортировка: чат с последним сообщением по времени вверх.

Если пользователь на данный момент открыл вкладку с чатами и смотрит на нее, он должен увидеть как строка чата переходит со своего положения на верхнее в списке, если в этот момент в этот чат поступило новое входящее сообщение.

Сортировка списка запросов и уведомлений

Запросы и уведомления подгружаются пакетами по **50 шт.**

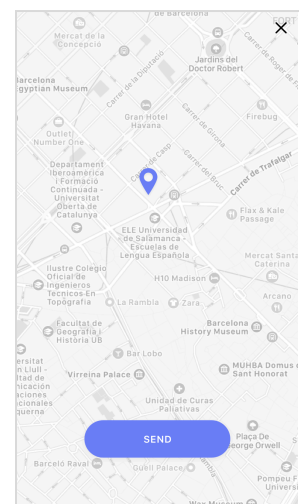
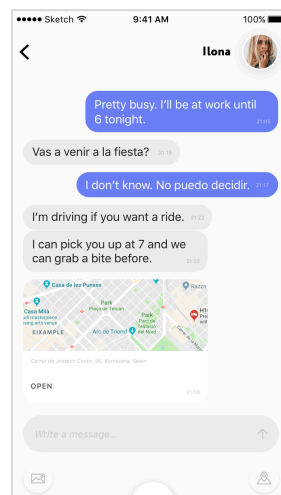
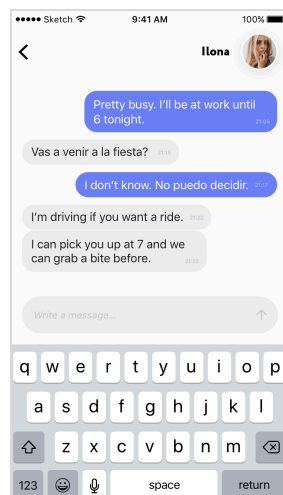
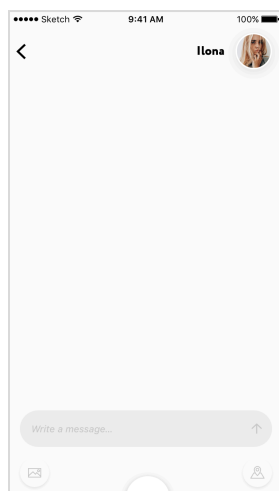
Сортировка: свежие вверх



Chat Room

Внутри активного чата, пользователи могут обмениваться:

- текстовыми сообщениями с использованием встроенных **emoji** (**emoji не в этом релизе**)
- встроенными **стикерами** (**не в этом релизе**)
- изображениями
- короткими видео (**не в этом релизе**)
- локацией
- запросами или приглашениями



Если какой-либо из пользователей вписал внутри своего текстового сообщения телефонный номер (определить по самым примитивным паттернам), то опознанный номер выделяется, подчеркивается и становится активным для нажатия. При нажатии должен производиться переход девайса телефонный вызов.

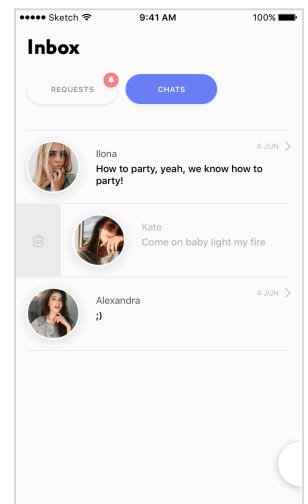
Сортировка сообщений в чате

Сообщения подгружаются пакетами по **25 шт.** Чтобы получить еще 25 более старых сообщений нужно скролить вверх по чату.

Удаление чата

При желании пользователь **A** может “покинуть” чат - удалить его. Это можно сделать внутри чата через меню блокирования или жалобы на пользователя или находясь в списке чатов свайпом. В таком случае, собеседник (пользователь **B**) получает соответствующее уведомление о том, что пользователь **A** покинул чат и для второго пользователя **B** прекращается возможность писать прямые сообщения удалившему. Однако, сама история переписки для пользователя **B** остается доступной до тех пор, пока сам пользователь **B** также не удалит данный чат. Несмотря на то, что писать напрямую пользователь **B** больше не может, он по-прежнему может отправлять пользователю **A** сообщения или комментарии с запросами к **events**. Пользователь **A** будет снова получать такие входящие у себя в inbox. И в случае, если он переходит по такому уведомлению в “потенциальный” чат, то видит не только последние запросы и сообщения, но и отметку о том, что он удалил этот чат и всю переписку, которая была ранее. Если пользователь **A** ответит на новые запросы от пользователя **B**, то чат возобновится.

Если пользователь заблокировал другого пользователя (пожаловался) то чат тоже будет удален, но в таком случае ни запросы ни сообщения больше никогда не будут доходить пользователю от заблокированного собеседника.



Блокирование пользователя и жалобы

При желании пользователь **A** может заблокировать пользователя **B**. Это можно сделать с жалобой или без.

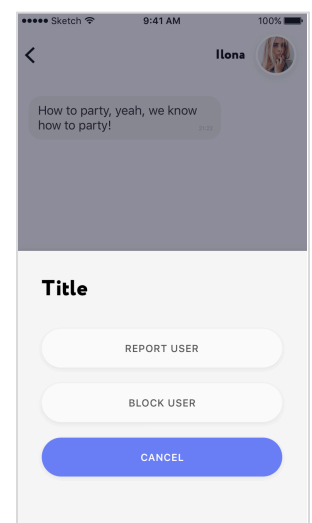
Блокировка может быть произведена:

- из чата (при открытии контента пользователя)
- со страницы аккаунта пользователя
- со страницы какого-либо из **events** пользователя или из списка **events** всех пользователей.

При блокировке пользователя инициатор блокировки должен выбрать заблокировать пользователя или заблокировать и пожаловаться. При выборе варианта с жалобой нужно также выбрать причину жалобы (например, сообщить о нарушении правил, законов и проч.)

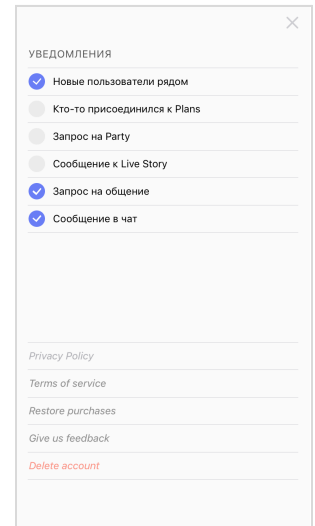
Все жалобы отправляются на рассмотрение модераторов сервиса.

При блокировке пользователя **B** пользователем **A**, пользователь **A** больше никогда не увидит никакой контент пользователя **B** (ни аккаунта, ни чатов, ни events) все существующие чаты, запросы и входящие сообщения удаляется и пользователь более никогда не получает никаких упоминаний



от пользователя **B**.

Пользователь **B** в свою очередь получает уведомление в чат, если он есть о том, что пользователь **A** покинул чат. Однако, он будет видеть весь контент созданный или создаваемый пользователем **A**.



Настройки

В соответствующем разделе с настройками пользователь может изменять работу некоторых функций приложения или сервиса.

Пользователь имеет возможность изменить:

- **Target**
- **Nickname**
- **Description**
- **Selfie**

Включить или выключить следующие нотификации:

- Получать уведомления о новых пользователях рядом (радиус 5 км)
- Получать уведомления о том, что кто-то присоединился к **Plans** пользователя
- Получать уведомления о том, что кто-то сделал запрос на **Party** пользователя
- Получать уведомления о том, что кто-то написал сообщение к **Live Story**
- Получать уведомления о том, что кто-то сделал запрос на общение
- Получать уведомления о том, что кто-то написал сообщение в чат

Пользователь также может перейти по ссылкам **privacy policy**, **terms of service** - открывающие в webview страницу по https ссылке с соответствующей информацией.

Обратная связь

Пользователь может написать письмо в службу поддержки клиентов - должен открыться почтовый клиент девайса с заготовленным текстом вставленным в письмо-шаблон.

Шаблон формируется по следующему правилу:

U:[User_id];[App_name];[App_version];[OS];[OS_vers];[Lang];[Country]

Пример: U:06E188BC-2C06-4CFE-BE5C-FE26E032CC05v;Amare;1.1.3;iOS;10.3;EN;US

User_id - уникальный идентификатор пользователя в **CDB**

App_name - уникальный идентификатор приложения в **CDB**

App_version - версия приложения

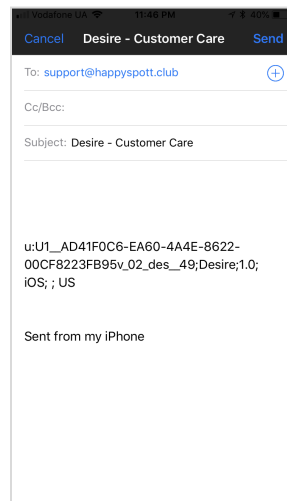
OS - операционная система девайса пользователя (iOS, Android)

OS_vers - версия операционной системы девайса пользователя

Lang - язык девайса пользователя

Country - страна местонахождения пользователя

Тема письма также является переменным значением и устанавливается в зависимости от языка девайса пользователя.

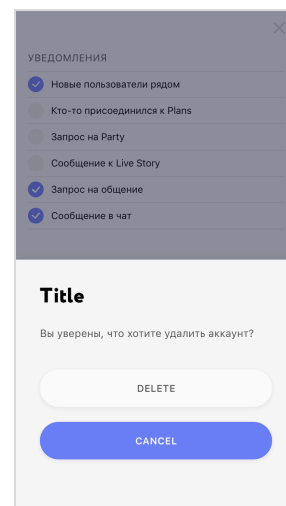


Удаление собственно аккаунта

Пользователь имеет право удалить свой аккаунт и все данные о нем из системы. Нажав на соответствующую кнопку в настройках, пользователь увидит предупреждение, опрос и возможность отказаться от удаления.

После подтверждения пользователем удаления аккаунта, из системы **CDB** должны быть удалены все объекты относящиеся к пользователю, если они не нарушают целостность базы данных. Если пользователь состоял в каких либо чатах, то происходит удаление чата (аналогично ситуации, когда пользователь покинул чат). Всем собеседникам в активный чат отправляется сообщение о том, что пользователь покинул чат и возможность писать в такой чат прекращается.

Следует удалить все созданные пользователем **events**. Все медиафайлы, загруженные пользователем также должны быть удалены.



Типы пользователей MA

В приложении на данный момент предусмотрены следующие типы пользователей:

- Newbie
- Member
- Escapee

Newbie

Пользователь, который установил приложение и никогда не покупал ни одного продукта-подписки ранее.

Пользователь **может**:

- Пройти процесс авторизации
- Отправлять на утверждение **selfie**
- Создать свою **Live Story**
- Просматривать контент других пользователей
- Менять свои личные настройки или **selfie**
- Получать сообщения и уведомления от других пользователей
- Блокировать или жаловаться на других пользователей
- Удалить свой аккаунт

Пользователь **не может**:

- Создавать **Party** или **Plans**
- Открывать, отвечать, удалять запросы, сообщения или комментарии.
- Писать другим пользователям, отправлять запросы или присоединяться к **Party** или **Plans**

При попытках взаимодействия с запрещенными для данного типа пользователя функциями появляется экран покупки продуктов (далее "**PayGate**"). Кроме того, **PayGate** появляется в определенных местах автоматически.

Member

Данному типу пользователей доступны все существующие на данный момент функции приложения. *(В будущем планируется появиться еще один тип пользователи, которые будут иметь дополнительные преимущества перед типом **member** - им будут доступны свех-функции)*

Escapee

Данному типу пользователю разрешено и запрещено все то же, что и типу **Newbie**. Однако, ему могут быть показаны некоторые другие экраны, тексты и продукты для повторных покупок.

In-App Purchases

Проверка статуса

MA является бесплатным для скачивания и установки. Однако, некоторые функции сервиса доступны только особым типам пользователей.

Приобретение особого статуса **Member**, позволяющего пользователю получить доступ ко всем функциям приложения осуществляется путем приобретения платной авто-продляемой подписки iTunes или Google Play. Если пользователь типа Member, отменит автопродление подписки, то, когда время действия последнего оплаченного периода закончится, система должна сменить пользователю тип с **Member**, на **Escapee**.

При запуске в приложение на Splash экране после авторизации, должна произойти проверка наличия подписки (запрос на iTunes или Google Play серверы)

- Если подписка присутствует и она валидная, то если пользователь не **Member**, то должен произойти апдейт статуса пользователя на **Member**.
- Если подписка не присутствует и пользователь **Member**, то его статус меняется на **Escapee**. (Newbie остается Newbie)
- Пользователь Free не проверяется

Если пользователь **Newbie** или **Escapee**, но его тип (**gender** = "female", "**target**" = "male"). Такие пользователи получают доступность использования функций **MA** как и пользователи типа **member**. Проверять подписку для таких пользователей не требуется.

В случае, если пользователь не **Member** и у него нет активной валидной подписки, следует получить все данные о продуктах (названия, цены, валюты и т.д.) - сделать это нужно в фоновом потоке без задержки UI. Данные нужно закешировать на период сессии - они потребуются при показе PayGate.

PayGate - покупка подписок

Экраны **PayGate** появляются для пользователей со статусом **Newbie** и **Escapee** в следующих случаях:

- Пользователь нажал на запрос или приглашение в списке чатов (*кроме системного - см. онбординг*) ([Action](#))
- Пользователь нажал на чат в списке чатов (*кроме системного - см. онбординг*) ([Action](#))
- Пользователь нажал на любую кнопку взаимодействия в любом месте (Join, Write message, Invite и т.д.) ([Action](#))
- Пользователь пытается создать Plan или Party ([Action](#))
- При переходе на **Feed** экран, после того как были показаны первые результаты поиска, с задержкой 3 секунды (если результаты не найдены вообще - без задержки) ([Popup](#))

[Action](#), [Popup](#) - триггеры появления PayGates

PayGate делится на 4 группы по типу продуктов продаваемых на нем:

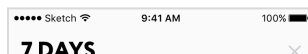
- **Free Trial**: продается только один продукт - подписка с бесплатным пробным периодом ([FT](#))
- **Paid Trial**: продается только один продукт - подписка с платным пробным периодом ([PT](#))
- **Base Multi**: продаются три продукта - подписки без пробных периодов ([BM](#))
- **Base Single**: продается только один продукт - подписка без пробного периода ([BS](#))

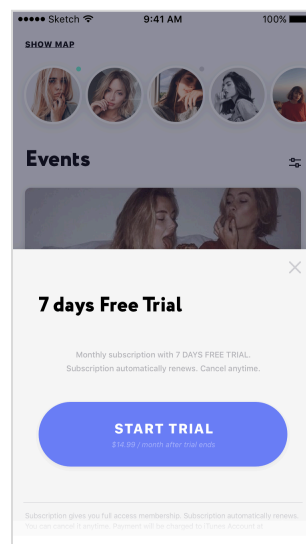
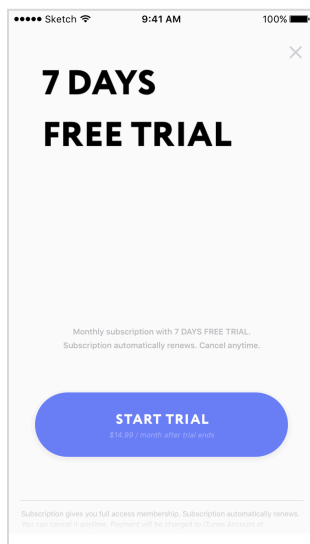
[FT](#), [PT](#), [BM](#), [BS](#) - типы комбинаций продуктов

Дизайн PayGate может быть 12-ти типов (на самом деле 6-ти типов: 12 - получается, только из-за комбинации переменных текстов:

- **Free Trial PayGate** на весь экран без результатов ([D1](#))
- **Free Trial PayGate** на весь экран с результатами ([D2](#))
- **Free Trial PayGate** на часть экрана ([D3](#))
- **Paid Trial PayGate** на весь экран без результатов ([D1](#))
- **Paid Trial PayGate** на весь экран с результатами ([D2](#))
- **Paid Trial PayGate** на часть экрана ([D3](#))
- **Base Single: PayGate** на весь экран без результатов ([D1](#))
- **Base Single: PayGate** на весь экран с результатами ([D2](#))
- **Base Single: PayGate** на часть экрана ([D3](#))
- **Base Multi: PayGate** на весь экран без результатов ([D4](#))
- **Base Multi: PayGate** на весь экран с результатами ([D5](#))
- **Base Multi: PayGate** на часть экрана ([D6](#))

[D1-D6](#) - типы дизайна экрана





Free Trial PayGate и

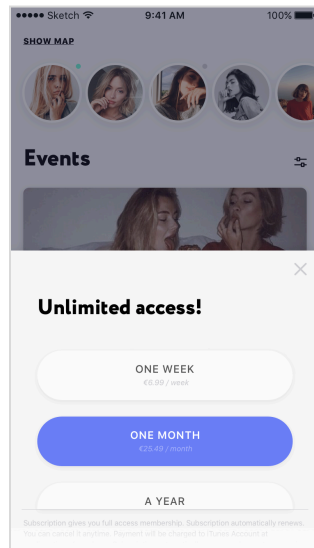
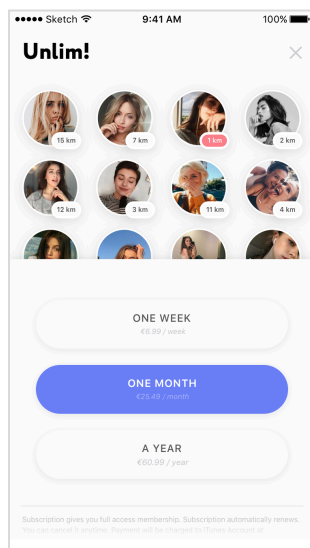
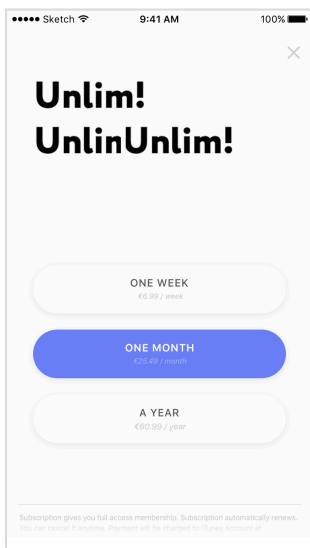
Paid Trial PayGate -

отличаются только текстом и продуктом.

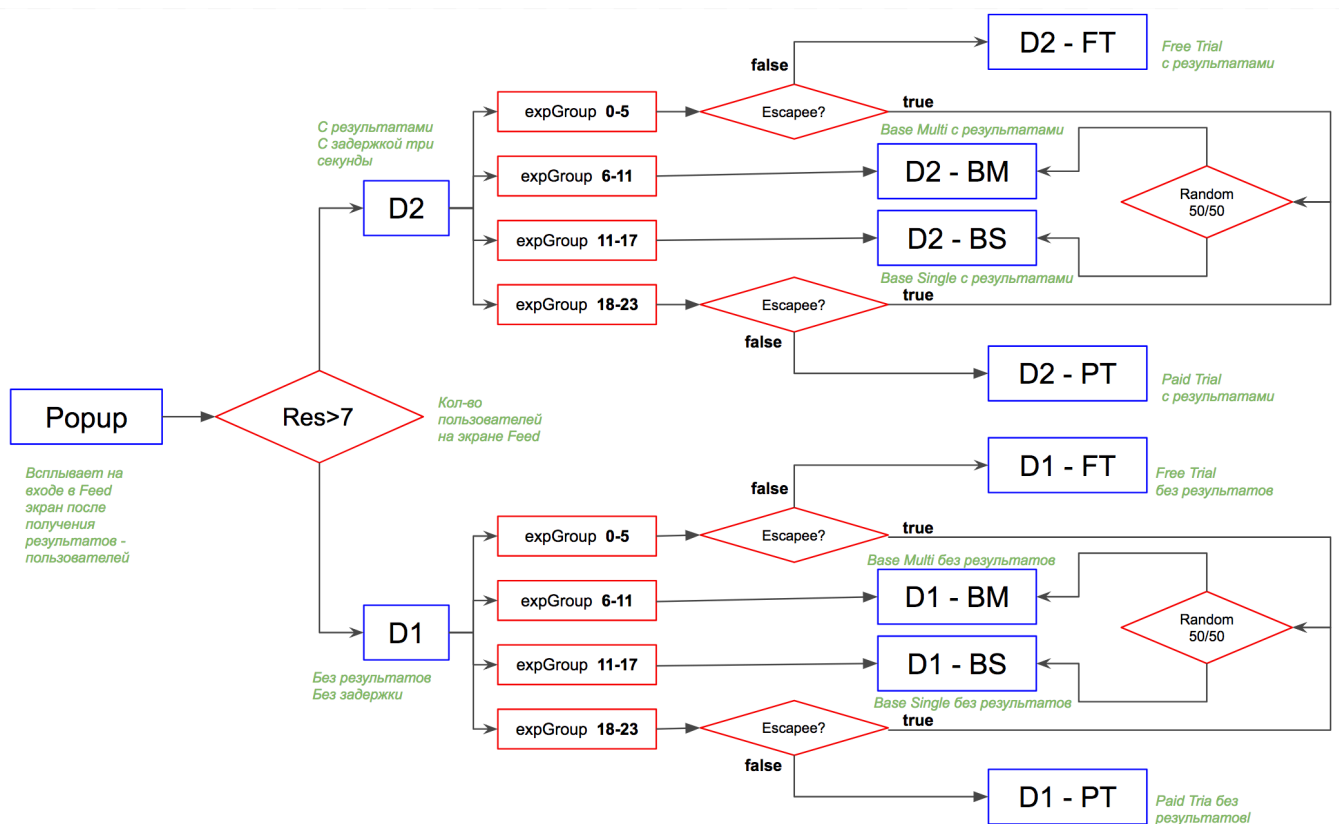
В будущем добавится еще тип PayGate с одним Base продуктом - будет.

PayGate на часть экрана означает, что часть экрана прозрачная с легким затемнением и под ней видно тот интерфейс, который перекрыл PayGate.

Результатами считаются **selfies** пользователей найденных вокруг пользователя **MA**. Внизу экрана располагается длинный текст, который едва виден, но его можно прочитать проскролив весь экран PayGate вниз. В случае частичного экрана, он также скроллится поднимаясь вверх - перекрывая полупрозрачный участок.

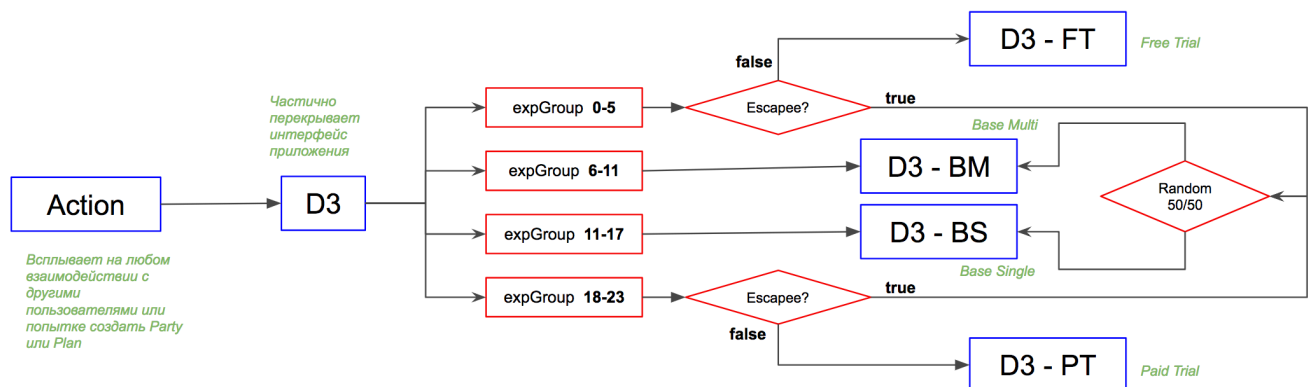


На экране, где отрисованы круглые фотографии пользователей и указано расстояние до них, пользователи должны быть отсортированы по расстоянию. Если расстояние до пользователя



более 50 км, то значок с указанием расстояния не показывать - только selfie пользователя. Если расстояние до пользователя менее или равно 3 км, то значок с расстоянием показывается красным.

PayGate на весь экран появляется независимо от типа продукта каждый раз при переходе на **Feed** экран. Если на Feed экране присутствует более 8 результатов то следует показывать Paygate на весь экран с результатами. Пользователь, который уже ранее покупал какую-либо подписку и отменил ее, соответственно получил статус Escapee - больше не может увидеть никакой из вариантов PayGate с триалом.



Важно: значение expGroup - может быть подменено, не у пользователя, а при выборе типа PayGate или групп продуктов. Это может быть включено как принудительное значение для всех, при условии активации данной функции в админ-панели системы (см. **Remote Configs**). Другими словами, если на сервере установлено принудительное значение 3, то все пользователи будут видеть PayGate соответствующий expGroup = 3, как-только на сервере значение сменят на отлично от диапазона [0, 23] - те же самые пользователи будут видеть ту комбинацию PayGate, которая соответствует их expGroup.

После того, как PayGate экран показался, следует проверить наличие закешированных данных о доступных продуктах и если их нет, то сделать запрос и после того, как данные получены отобразить их на странице - реактивно - нарисуются кнопки, потом на них добавятся цены)

Когда пользователь нажимает на кнопку, соответствующую кнопке “купить подписку”, происходит проверка наличия текущей подписки, на случай если произошла ошибка при проверке на момент авторизации, и если подписка уже есть и она активная и валидная, то статус пользователя обновляется на **Member** и PayGate закрывается. Если подписки нет, отправляется запрос на покупку продукта.

Если в ответ на попытку покупки возвращается **fail** по причине отказа пользователя, то не нужно показывать никакого алерта с ошибкой. PayGate остается открытым.

В случае, если сервер iTunes или GooglePlay ответил **fail** следует показать эту ошибку на экране в диалоговом окне (также отправить в crashlytics).

В случае успешной покупки, пользователь должен передать на сервер для возможности ее немедленной или последующей плановой валидации сервером следующие данные:

Для Google Play:

- **product_id** - id продукта
- **purchase_token** - токен транзакции
- **signature** - подпись транзакции

Для iTunes:

- **product_id** - id продукта
- **receipt** - id чека
- **signature** - подпись транзакции

После успешной покупки, статус **Membership** и **Membertime** пользователя обновляется на **Member** и PayGate закрывается.

Параметры следует уточнить, так как API сервисов могло измениться.

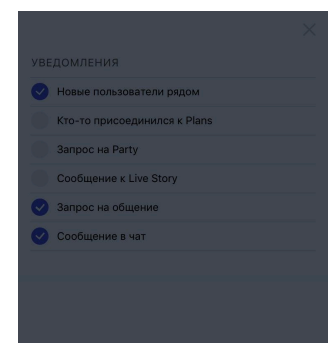
MemberTime - это **expirationDate**, которое должно быть получено от сервиса iTunes или GooglePlay и может называться по другому, но смысл в том, что это время, до которого будет валиден проверяемый чек покупки.

Restore Purchases

На экране настроек приложения имеется пункт “Восстановление покупок”. После нажатия на него пользователь должен увидеть системный лодер, а в фоне должен произойти процесс проверки наличия покупки у пользователя, точно такой же, какой запускается при старте приложения или при запуске PayGate экрана.

В случае, если покупка найдена, происходит апдейт статуса **membership** и **membertime** (на период не от последнего восстановления, а последней транзакции) и пользователь должен увидеть сообщение на экране “Ваши покупки восстановлены и статус обновлен”.

В случае отсутствия покупок у пользователя активных покупок, должна появиться надпись “У вас нет активных покупок”



Purchase Restore

У вас нет активных покупок

OK

В случае системной ошибки ее нужно вывести на экран.

Уведомления

В процессе взаимодействия пользователей между собой и с системой, пользователь получает нотификации: когда приложение закрыто или свернуто, то системные нативные уведомления (Push notifications), когда приложение открыто - внутренние (In-App notifications).

Нативные системные Push уведомления

Пользователь может отказаться от получения тех или иных уведомлений у себя в настройках (см. Настройки). Это означает, что другие пользователи или сервис не будет уведомлять отказавшегося о соответствующих событиях.

Пользователь получает следующие системные сообщения:

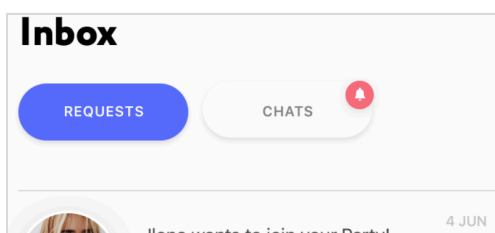
- уведомления о новых пользователях рядом (радиус 5 км)
- уведомления о том, что кто-то присоединился к **Plans** пользователя
- уведомления о том, что кто-то сделал запрос на **Party** пользователя
- уведомления о том, что кто-то написал сообщение к **Live Story**
- уведомления о том, что кто-то сделал запрос на общение
- уведомления о том, что кто-то написал сообщение в чат (отправляется сообщение типа - text, image, location, но не сам текст. Когда сообщение приходит то по типу в зависимости от языка установленного на девайсе будет написано на языке пользователя “new message”, “новое сообщение” и т.д)
- произвольные уведомления отправленные системой (отправленные при помощи бекенда или специально подключенного сервиса)

In-App уведомления

В **MA** существуют специальные значки уведомляющие пользователя о разных новых событиях или обновлении контента (далее **badge**).

Badges уведомляют пользователя о появлении:

- новых созданных чатах (на левой кнопке навигационного меню, на вкладке **chats**, на экране **Inbox**, на строке определенного чата в списке чатов на экране **Inbox**)
- новых входящих сообщениях (на левой кнопке навигационного меню, на вкладке **chats**, на экране **Inbox**, на строке определенного чата в списке чатов на экране **Inbox**)
- Новые входящие запросы (на левой кнопке навигационного меню, на вкладке **requests**, на экране **Inbox**, на строке определенного запроса в списке запросов на экране **Inbox**)



Аналитика

В систему должны быть внедрены разные системы аналитики.

Подключаемые сервисы

- **Google Analytics** (<https://developers.google.com/analytics/devguides/collection/>)
- **Amplitude** (<https://amplitude.zendesk.com/hc/en-us/sections/115000961027-SDK-References>)
- **USCAM** (<https://uxcam.com/>)
- **Deep Linking**
- **Crashlytics** (<https://fabric.io/kits/ios/crashlytics>)

Google Analytics и Amplitude должны дублировать друг друга - будут отправляться одинаковые ивенты.

Список ивентов

Список **клиентских** ивентов для Google Analytics и Amplitude:

- **Показ экрана** (встроенные методы в SDK обоих сервисов)
 - **Splash Screen**
 - **Gender Screen**
 - **Profile Screen**
 - **Selfie Screen**
 - **Proof Screen**
 - **Feed Screen**
 - **Map Screen**
 - **Inbox Screen**
 - **Story Event Screen**
 - **Party Event Screen**
 - **Plan Event Screen**
 - **User List Screen**
 - **Event Details Screen**
 - **User Details Screen**
 - **Settings Screen**
 - **My Events Screen**
 - **Chatroom Screen**
 - **PayGate Screen**
 - **type** (String) - T1-FT, T2-PT (см. PayGate Screen)
- **Действия**
 - **Sign Up Done** - произошло создание пользователя на сервере
 - **Sign Up Tap** - нажата кнопка NEXT на экране с выбором пола
 - **Login Done** - пользователь идентифицирован и получен с сервера
 - **Gender Done** - введены оба gender и target
 - **gender** (String) - M,F,Bi
 - **target** (String) - M,F,Bi
 - **First Selfie Tap** - впервые нажата кнопка создать селфи
 - **First Selfie Done** - впервые создано селфи
 - **Proof Tap** - нажата кнопка создания подтверждения селфи
 - **Proof Done** - нажата кнопка отправки proof на проверку
 - **Story Btn Tap** - нажата кнопка создания Live Story на экране профайла

- **Party Btn Tap** - нажата кнопка создания Party на экране профайла
- **Palans Btn Tap** - нажата кнопка создания Plans на экране профайла
- **Location Ask** - показан запрос на разрешение использовать локацию
- **Location Ask Done** - показан запрос на разрешение использовать локацию
 - **result** (Boolean) - дал или нет разрешения
- **Story Done** - успешно создан event типа Live Story
 - **photos** (Integer) - количество добавленных фотографий
 - **exact_location** (Boolean) - true, если оставил точное местоположение
 - **target** (String) - **all** (если для всех), **target** (если для противоположного пола), **invited** (если только по инвайтам)
 - **hidden** (Boolean) - true, если оставил не показывать там, с кем уже есть чат
 - **lifetime** (Integer) - длина публикации стори в фиде - 1, 12, 24
 - **count** (Integer) - который по счету это ивент данного типа у пользователя за все время его существования
 - **count_all** (Integer) - который по счету это ивент любого типа у пользователя за все время его существования
- **Party Done** - успешно создан event типа Party
 - **photos** (Integer) - количество добавленных фотографий
 - **current_location** (String) - true, если оставил текущее местоположение
 - **hidden_location** (String) - true, если локация показывается только утвержденным пользователям
 - **target** (String) - **all** (если для всех), **target** (если для противоположного пола), **invited** (если только по инвайтам)
 - **hidden** (Boolean) - true, если оставил не показывать там, с кем уже есть чат
 - **today** (Boolean) - true, если Party создано и планируется в один и тот же день
 - **count** (Integer) - который по счету это ивент данного типа у пользователя за все время его существования
 - **count_all** (Integer) - который по счету это ивент любого типа у пользователя за все время его существования
- **Palans Done** - успешно создан event типа Plans
 - **target** (String) - **all** (если для всех), **target** (если для противоположного пола), **invited** (если только по инвайтам)
 - **hidden** (Boolean) - true, если оставил не показывать там, с кем уже есть чат
 - **today** (Boolean) - true, если Plans создан и планируется в один и тот же день
 - **count** (Integer) - который по счету это ивент данного типа у пользователя за все время его существования
 - **count_all** (Integer) - который по счету это ивент любого типа у пользователя за все время его существования
- **Location Permission Shown**
 - **success** (Boolean) - true, если получены разрешения
- **First Search Request** - впервые отправлен запрос на поиск пользователей и ивентов.
 - **gender** (String) - M,F,Bi
 - **target** (String) - M,F,Bi
- **First Search Done** - впервые получены результаты от первого запроса поиска

- **users** (Integer) - количество пользователей, которое удалось найти
- **events** (Integer) - количество ивентов, которое удалось найти
- **time** (Integer) - количество секунд, которое длился поиск
- **gender** (String) - M,F,Bi
- **target** (String) - M,F,Bi
- **Request Sent** - успешная отправка приглашения
 - **type** (String) - **party, plan, story_msg, direct** - прямое сообщение
 - **msg** (Boolean) - true, если с запросом отправлено опциональное сообщение
 - **gender** (String) - M,F,Bi
 - **target** (String) - M,F,Bi
- **Request Approved** - успешная отправка приглашения
 - **type** (String) - **party, plan, story_msg, direct** - прямое сообщение
 - **msg** (Boolean) - true, если с запросом отправлено опциональное сообщение
 - **gender** (String) - M,F,Bi
 - **target** (String) - M,F,Bi
- **Chat Created** - ответ на входящий чат или запрос (так создается чат)
 - **type** (String) - **party, plan, story_msg, direct** - прямое сообщение
 - **msg** (Boolean) - true, если с запросом отправлено опциональное сообщение
- **Chat Activated** - когда, каждый из собеседников написал другому [chat_activated_count] количество сообщений (см. System Configs в админ панели)
- **Invite Tap** - тап по кнопке Invite в сендвиче пользователя
 - **gender** (String) - M,F,Bi
 - **target** (String) - M,F,Bi
- **Invite Done** - успешная отправка приглашения по кнопке INVITE (отправляется вместе с Request Sent)
 - **just_created** (Boolean) - true, если ивент на который пригласили был создан в процессе приглашения
 - **type** (String) - **party, plan**
 - **gender** (String) - M,F,Bi
 - **target** (String) - M,F,Bi
- **Account Deleted** - пользователь удалил свой аккаунт
- **PayGate Tap**
 - **product_id** (String) - product_id продукта, который соответствует кнопке, по которой был тап
- **Purchase Done**
 - **product_id** (String) - product_id продукта, который был успешно приобретен

Список **серверных** ивентов только для Amplitude:

Следует увязать пользователей по id - см. Документацию к Amplitude Analytics.

- **Payment Done** - Успешная валидация покупки. Не отправляется при покупке бесплатного триала.
 - **product_id** (String) - product_id продукта, который был успешно приобретен
 - **count** (Integer) - который по счету раз у данного пользователя происходит оплата.

- **chats_count** (Integer) - количество чатов у пользователя на момент оплаты
- **Payment Canceled** - Проверка показала, что пользователь отменил продление и покупка не совершилась (для повторной плановой проверки чеков).
 - **product_id** (String) - product_id продукта, который был успешно приобретен
 - **count** (Integer) - сколько раз ранее у этого пользователя проходила оплата.
 - **chats_count** (Integer) - количество чатов у пользователя на момент оплаты

Backend

Взаимодействие клиентской части - мобильного приложения и серверной части должно быть спроектирована по принципу оптимальной точки пересечения экономии и удобства расширения дополнительных возможностей приложения, а также переиспользуемости решений.

Все функции необходимые для полноценного функционирования мобильного приложения должны быть реализованы на клиенте или при помощи облачных функций.

Если можно избежать траты процессорного времени облачной функции и сделать вычисление на клиенте, при этом не создать значительных ограничений в возможностях данной функции, а также не получить БОльшого расхода средств, связанных с платой за исходящий/входящий трафик необходимый для выполнения расчетов на клиенте, то следует перевести такие вычисления на клиент.

Например, валидация подписки iTunes или GooglePlay происходит на клиенте, без пересчета ее на сервере, хотя это и является узким местом возможного недобросовестного использования приложения пользователями, которые смогут подделать электронный чек продукта, в действительности совокупный ущерб наносимый подобными инцидентами настолько мал, что им можно пренебречь. Тем не менее, это не исключает необходимости валидации чеков на сервере, а лишь уменьшает кол-во плановых проверок.

Если можно делать записи в БД или читать из нее без дополнительных логических действий, валидаций, триггеров и проч. То следовало бы избежать создание дополнительных облачных функций.

Существуют также функции, которые необходимо выполнять для полноценного функционирования системы и админ-панели, некоторые из которых будут работать постоянно, а некоторые требуется запускать периодически через планировщик в определенной время в фоне. Такие функции необходимо создать на выделенном сервере.

При таком разделении, в случае, если выделенный сервер, временно недоступен, то мобильные приложения продолжат работать стабильно, так как они будут зависеть только от стабильности работы Amazon или Google, а фоновые процессы, которые могут оказаться "тяжелыми" и исчерпать запасы оперативной памяти, и соответственно вывести из строя сервер, не повлияют на работоспособность мобильных приложений.

Функции которые требуется реализовать на выделенном сервере:

- Периодическая проверка - валидация чеков iTunes или GooglePlay
- Чистка базы данных от пользователей и их контента, срок годности которых истек
- Все функции необходимые для работы админ-панели

Валидация чеков покупок

Если покупка прошла успешно - получен **success** на клиенте, то клиент присылает параметры необходимые для валидации чека через API сервисов.

Следует провалидировать чек и пропатчить параметры пользователя **Membership** и **MemberTime** - timestamp до которого будет валидным его покупка.

После успешной валидации, если **expirationDate (MemberTime)** изменился, то следовательно произошла не проверка существующей покупки, а новая покупка (продление с прошлого платежного периода на новый, это тоже новая покупка) следовательно нужно отправить в аналитику соответствующий ивент (с инкрементом кол-ва покупок данным пользователем, за исключением, если предыдущий период был бесплатным триалом).

Abuses

Когда пользователи жалуются на какого-либо пользователя или его пользователя следует инкрементировать счетчики жалоб для каждого пользователя и его контента. У объекта пользователя и каждого объекта ивента должно быть поле для учета количества жалоб.

Если на какого-то пользователя пожаловались более **[user_tolerance]** пользователей, он автоматически должен быть заблокирован системой - статус **suspended**

Если на какой-либо ивент пользователя пожаловались более **[content_tolerance]** пользователей, и контент пользователя и сам пользователь автоматически должен быть заблокирован системой - статус **suspended**

(см. [System Configs](#) в админ панели)

Такой пользователь не сможет пользоваться приложением, но его контент остается до принятия решения модератором.

Административная часть

Админ-панель сервиса необходима для управления контентом, конфигураций системы, поведением мобильных приложений, управления фоновыми сервисами и модерацией.

Админ-панель доступна только зарегистрированным пользователям админ-панели.

Разделы

- **Users** (пользователи админки)
- **Settings** (Настройки админки)
- **Content** (контент пользователей приложений)
- **Moderation** (раздел для модерации контента пользователей мобильных приложений)

Users

Пользователи создаются только супер-модератором. При создании следует назначить логин, пароль и отметить в какие разделы получит доступ пользователь. В последствии, супер-модератор может менять все эти параметры.

Типы пользователей:

- **Superadmin** - доступны все существующие разделы и опции, а также возможность создавать других пользователей и назначать им права
- **Admin** - доступны все существующие разделы и опции
- **Editor** - доступен раздел поиска, создания и редактирования контента пользователей, а также раздел модерирования
- **Moderator** - доступен только раздел модерирования

Settings

Настройки делятся на два типа

- **System Configs** - настройки работы всей системы
- **Remote Configs** - настройки мобильного приложения передаваемые на клиент при авторизации пользователя в зависимости от платформы и версии мобильного приложения

System Configs

- **[user_tolerance]** - количество жалоб на пользователя до того, как он будет автоматически забанен
- **[content_tolerance]** - количество жалоб на пользователя до того, как он будет автоматически забанен
- **[suspended_time]** - количество дней, после которых аккаунт и весь контент пользователя находящегося в статусе suspended удаляются.
- **[chat_activated_count]** - количество сообщений с каждой стороны собеседников в чате, после которого можно считать чат “активным” (см. Аналитику)
- **[moderator_clean_time]** - количество дней, после которых созданный модератором бекап забаненного пользователя удаляется.
- **[moderator_session]** - количество минут, после которых сессия модератора считается прерванной

Remote Configs

Администратор может добавлять любое количество таких переменных, и если **MA** встречает незнакомую переменную, или ее тип не соответствует ожидаемому типу, или не соответствует диапазон, то такая переменная просто игнорируется и в работу принимается значение заданное по умолчанию в Remote Configs.

Список уже известных переменных:

- **[OS]** (String) - операционная система, для которой подготовлена ветка параметров
 - **[appVers]** (String) - версия приложения, для которой действуют параметры (-1 - для всех версий, кроме указанных)

- **[expGroup]** (Object) - если значение имеется и оно в диапазоне [0, 23] то оно используется для всех пользователей вместо того, которое было установлено пользователям, само expGroup пользователей не меняется и как только админ изменит это значение или версия приложения смениться, пользователи станут снова использовать то expGroup, которое у них было создано в свойствах при регистрации. Значения задаются через вложенное значение **all** - для всех стран, кроме тех которые указаны отдельно по стандарту ISO 3166-1 alpha-2 - две буквы: UA, PT, US....

Пример:

- **all: 0 (Integer)** - для всех стран, которые не указаны специально
 - **US: 3**
 - **PT: 13**
- **[exampleUsers]** (Boolean) - наличие фейковых пользователей с контентом, наполняющих приложение для демонстрации функциональности приложения модератору проверяющему приложение.
- **[reviewMod_1]** (Boolean) включение дополнительных настроек, например, показ кнопки определенным цветом или показ какого-то текста, отличного от дефолтного.
- **[uxCam]** (Integer) - значение в диапазоне [0, 100] - процент пользователей, для которых будет активирован сервис **UXCAM** для записи действий пользователей. Задается через вложенное значение **all** - для всех стран, кроме тех которые указаны отдельно.
- **[maxRadius]** (Integer) - максимальный радиус используемый для поиска пользователей
- **[gooMapsKey]** (String) - ключ к API Google Maps (в приложении будет хардкодом также указан дефолтный ключ)
- **[gooGeocodeKey]** (String) - ключ к API Google Geocode (если не указан, то в приложении не применять никакой)

Content

В данном разделе есть два подраздела:

- Users
- Content

Users

Результаты показываются в таблице постранично с пейджингом по 50 результатов на странице. Здесь можно просматривать, создавать и редактировать пользователей. А также банить пользователей.

Пользователи отображаются в таблице в виде строк с параметрами:

- **Изображение Selfie** - маленькая картинка размером 80 px большая их сторон (загружать изображение **preview**)
- **Gender**
- **Target**
- **Membership**
- **Status**
- **Country (ISO-code)**
- **Количество abuses на пользователя**
- **Суммарное количество abuses на контент пользователя**
- **Кнопка BAN**

- **Кнопка INFO** (см. Unit страницу)

По умолчанию сортировка отсутствует (так как отдаст база без специального указания сортировки)

Результаты могут быть пересортированы при нажатии на какой-либо параметр в шапке таблицы.

Также должна быть возможность найти конкретного пользователя по его **id** или по идентификатору покупки - **purchase_token** или **receipt**, которые передает клиент на сервер при валидации покупки.

На странице присутствует возможность фильтрации пользователей по указанным выше параметрам, а также по **user_id**, **expGroup**, **OS**, **Nickname**, **region** и **city** указав одно или несколько значений, для этих параметров.

Content

Результаты показываются в таблице постранично с пейджингом по 50 результатов на странице. Здесь можно просматривать и удалять контент пользователей - ивенты. А также банить пользователей.

Контент отображаются в таблице в виде строк с параметрами:

- **Изображение Selfie** - маленькая картинка размером 100 рх большая их сторон (загружать изображение **preview**)
- **Все изображения ивента** в горизонтальный ряд - 100 рх большая их сторон каждого изображения
- **Описание ивента**
- **Status** пользователя
- **Количество abuses на пользователя**
- **Суммарное количество abuses на контент пользователя**
- **Кнопка SUSPEND** - удаляется данный ивент пользователя и пользователь переходит в статус **suspended**
- **Кнопка BAN** - удаляется весь контент пользователя и пользователь получает статус **banned**
- **Кнопка INFO** (см. Unit страницу)

По умолчанию сортировка выполняется со следующими правилами:

- количество abuses на данный контент пользователя (чем больше тем выше)
- количество abuses на пользователя (чем больше тем выше)
- неподтвержденные аккаунты выше
- дата создания ивента (более старые, созданные раньше - выше)

One Unit

При клике на кнопку **INFO** открывается страница:

- Selfie изображение - крупное (Main)
- Количество abuses на пользователя
- со всеми параметрами одного пользователя
- со всем его контентом (Stories, Parties, Plans, Description, MemberTime - с пометками о количества жалоб на каждый из ивентов)
- дате и времени последнего состояния online
- фотографии proof, если она загружена и не утверждена
- Списанием всех его покупок и количество каждого типа покупки

На данной странице присутствуют кнопки

- **Save** - сохранить введенные данные, страница закрывается
- **Save & Continue** - сохранить введенные данные и остаться на этой странице
- **Close** - закрыть страницу
- **Delete** - удаляет пользователя из системы (и связанные с ним данные, а также все собеседники в чатах с этим пользователем будут видеть, что собеседник покинул чат)
- **Ban** - пользователь получает статус забаненного - весь контент пользователя удаляется.
- **Clean Abuses** - очистить счетчики жалоб с пользователя

Moderation

В данном разделе имеются следующие подразделы:

- Selfies
- Events
- Revise
 - Logs
 - Content

Пользователь типа moderator имеет доступ только в раздел Moderation.

Если модератор сделал BAN пользователю, то весь контент пользователя удаляется из основной базы, но сохраняется в виде бекапа где-то (например в Google Storage). Пользователь админ-панели может вернуть контент из бекапа и восстановить статус забаненного пользователя, до тех пор, пока существует бекап. Бекапы регулярно удаляются (см. Фоновые сервисы). Это сделано на случай, если модератор забанил кого-то по ошибке и для того чтобы другие пользователи могли проверить работу модератора и восстановить забаненного по ошибке пользователя.

Модератор не может создавать пользователя, редактировать его данные и удалять пользователя.

Учет активности пользователя и ведение логов его действий

При входе в раздел модерации пользователем **Editor** или **Moderator** (для этого пользователя сразу после авторизации, в том числе после перезагрузки страницы) должен появиться алерт с вопросом [“Are you ready to start moderation process?”](#) и кнопки [“Start”](#) и [“Cancel”](#). Кнопка [“Cancel”](#) возвращает пользователя **Editor** в раздел Users, а модератора выбрасывает logout на авторизацию. Кнопка [“Start”](#) начинает считать время работы и другие действия пользователя в данном разделе. Для остальных типов пользователей вход в данный раздел не требует никакого предупреждения и подсчета времени.

Подсчет времени начинается сразу после нажатия кнопки [“Start”](#) - начало сессии модератора. Если в течение [\[moderator_session\]](#) минут не нажата ни одна из кнопок **действия** (см. ниже), то сессия считается прерванной. На экране должен появиться тот же алерт [“Are you ready to start moderation process?”](#) и кнопки [“Start”](#) и [“Cancel”](#), с теми же действиями на кнопках.

Для модератора следует вести лог его сессий и счетчик **действий** пользователя каждой сессии пользователя - кликов по кнопкам [“OK”](#), [“BAN”](#), [“APPROVE”](#), [“SUSPENDED”](#). Не нужно писать в журнал каждое действие, как отдельную запись. Одна сессия - это одна запись, но для этой записи нужно инкрементировать значения действий.

Время сессии считать в секундах - отображать в админке в минутах.

Revise

В данном разделе имеется два подраздела Logs - журнал сессий модератора и Content - список пользователей или контента пользователей, которых обработал модератор.

Logs

Модератор может смотреть только информацию о своих сессиях и только свой отмодерированный контент.

Пользователи с правами большими, чем модератор могут выбирать пользователя для которого можно просматривать логи и продуктивность.

Лог представляет из себя список-таблицу строк со следующими свойствами за выбранный промежуток времени:

- дата сессии
- длина сессии в мин
- кол-во проверенных selfies (total)
- кол-во проверенных selfies (Approved) - клик открывает список обработанного контента
- кол-во проверенных selfies (Rejected) - клик открывает список обработанного контента
- кол-во проверенных ивентов (total)
- кол-во проверенных ивентов (Approved) - клик открывает список обработанного контента
- кол-во проверенных ивентов (Suspended) - клик открывает список обработанного контента
- кол-во "BAN" пользователей - клик открывает список обработанного контента
- коэффициент продуктивности модератора в минуту (кол-во действий за сессию / время сессии в секундах * 60)

Вверху таблицы должны быть три выделенные стилистически строки с обобщенными данными по работе данного модератора, которые не должны менять свои данные в зависимости о того какой промежуток времени выбран.

Строки с суммарными значениями сессий за:

1. Все время работы модератора
2. За последние 30 дней
3. За предыдущий рабочий день

Коэффициент продуктивности - **не среднее** значение по сессиям, **но равно** кол-во действий за суммарное время сессий / суммарное время сессий * 60.

Можно менять временной промежуток для отображения логов. По умолчанию показывается последний предыдущие сутки. Можно выбрать любой другой день, или месяц.

Content

Данный раздел представляет собой таблицу-список отработанных модератором пользователей и контента. Здесь есть подразделы:

- Approved Selfies
- Rejected Selfies
- Approved Events
- Suspended Events
- Banned Users

В один из разделов можно попасть при клике на какую-либо из обозначенных ячеек в таблице Logs (см. Выше список свойств таблицы логов).

Содержание подразделов - это строки со свойствами пользователей и изображениями.

Approved/Rejected Selfies:

1. Selfie (80x80px),
2. Proof-photo (80x80px)

3. Nickname
4. Description

Approved/Suspended Event:

1. Photos (80x80px)
2. Description

Banned Users:

1. Если забанен из раздела Selfie - то показывать тоже, что и в Approved/Rejected Selfies
2. Если забанен из Events - то показывать тоже, что и в Approved/Suspended Event

Возле каждого элемента есть кнопка RESTORE, которая восстанавливает пользователя и его контент из бекапа.

Selfies

Результаты показываются в таблице постранично с пейджингом по 50 результатов на странице. Здесь можно просматривать, создавать и редактировать пользователей. А также банить пользователей.

Пользователи отображаются в таблице в виде строк с параметрами:

- **Изображение Selfie** - маленькая картинка размером 80 px большая их сторон (загружать изображение **preview**)
- **Gender**
- **Target**
- **Membership**
- **Status**
- **Country (ISO-code)**
- **Количество abuses на пользователя**
- **Суммарное количество abuses на контент пользователя**
- **Кнопка BAN**
- **Кнопка INFO** (см. Unit страницу)

По умолчанию сортировка выполняется со следующими правилами:

- количество abuses на пользователя (чем больше, тем выше)
- суммарное количество abuses на контент пользователя (чем больше, тем выше)
- неподтвержденные аккаунты выше
- дата изменения селфи (более старые, созданные раньше - выше)

1. Проверить абузы (каждый день)

2. Проверить фото-пруфы (каждый день)

3. Проверить контент пользователей (каждый день)

4. Фиксация времени работы (каждый день)

5. Сохранение архива совершенных действий (каждый день)

6. Если модератор сделал BAN, то весь контент пользователя удаляется из основной базы, но сохраняет в виде бекапа, и admin или выше может вернуть контент из бекапа.

Фоновые сервисы на сервере

Регулярная проверка Membership

Запланировано: раз в 24 часа

Валидация чеков пользователей происходит запросом на соответствующие API iTunes или Apple. Сервис должен проходить по всем пользователям у которых **MemberTime** истек и делать запрос на сервисы iTunes или GooglePlay и при необходимости патчить **MemberTime**, **Membership**, а также отправлять соответствующий ивент в аналитику. Сделать проверку необходимо чуть позже, чем истекает его предыдущий купленный продукт, для того, чтобы маркет успел попытаться снять деньги с пользователя и изменил `expirationDate` или занес ошибку в его платежную историю. Таким образом, нужно сравнить `expirationDate` сохраненную в БД системы и `expirationDate`, которую отдает API iTunes или Apple, если дата та же, то транзакция еще не успела произойти, нужно проверить этого пользователя чуть позже, если `expirationDate` передвинулась вперед и ответ о валидации чека `success`, значит очередная оплата прошла успешно, нужно инкрементировать значение количества покупок пользователя и отправить ивент в аналитику.

Чистка пользователей

Запланировано: раз в 24 часа

Пользователи находящиеся в статусе **suspended** более **[suspended_time]** дней - получают статус **BANNED**, а весь их контент с ними связанный удаляется. Остается только объект пользователя со статусом **BANNED**.

Чистка бекапов

Запланировано: раз в 24 часа

Бекапы удаленных модератором пользователей очищаются, если бекапу более **[moderator_clean_time]** дней.