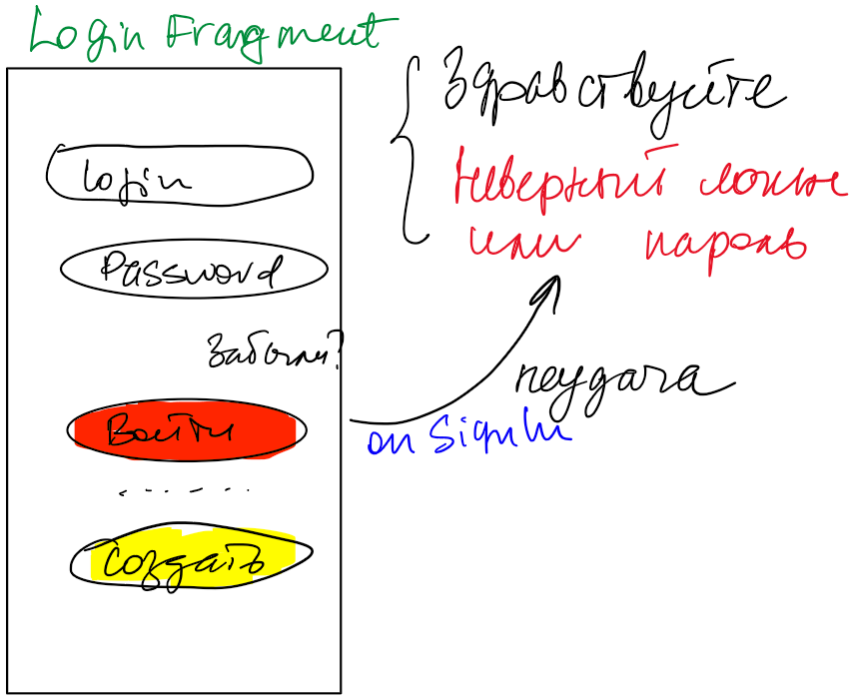
Техническое задание. Вход и Регистрация

Техническое задание на разработку экранов, viewModel-ей и логики представления для подсистемы входа и регистрации пользователя в системе

Приложение использует Single Activity – Multiple Fragment навигацию, LiveData и DataBinding (не ViewBinding), логика уровня представления реализована viewModel-ями, бизнес логика на сервере и частично на клиенте в классе Server

Фрагменты и навигация

Все фрагменты используют общий экземпляр LoginViewModel, который привязан к жизненному циклу вложенного навигационного графа LoginGroup. В этот граф входят все описанные ниже фрагменты, навигация в графе соответствует процессу входа или регистрации новой учетной записи



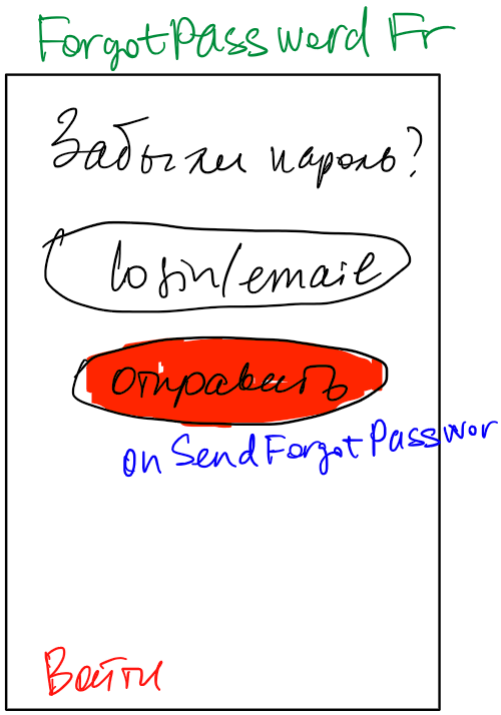
Пользователь переадресуется на этот экран каждый раз, когда требуется ввести данные для входа в систему (первый вход после установки, после выхода или если не подходит ранее установленный пароль)

Нажатие кнопки «Войти» обрабатывается во viewModel:

- Кнопка не активна, если login или password не заполнены
- При успешном входе:
    - Если почтовый адрес пользователя все еще не подтвержден, то переход на экран ConfirmEmailRequestFragment
    - Если все в порядке, то производится выход из подграфа навигации наружу
- При неудаче приветственное сообщение меняется на «Неверный логин или пароль»

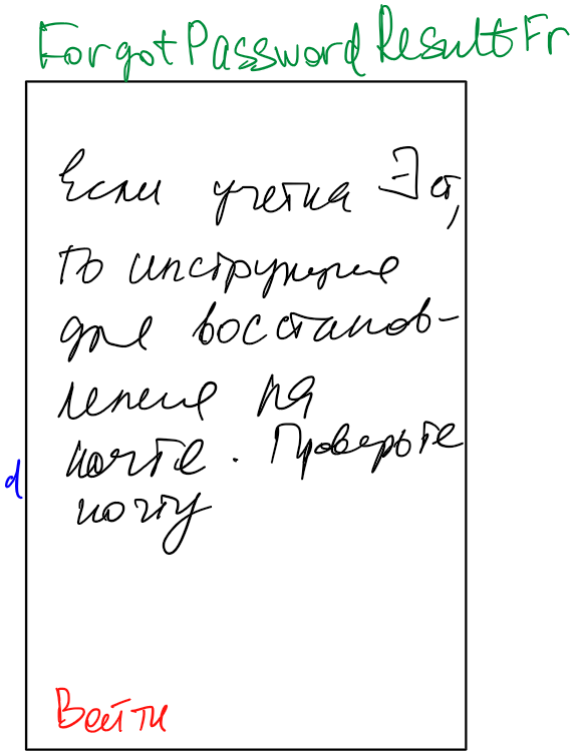
Фрагмент обрабатывает кнопки:

- «Создать» – переходит в экран создания новой учетной записи
- «Забыли пароль?» – переходит в экран восстановления пароля



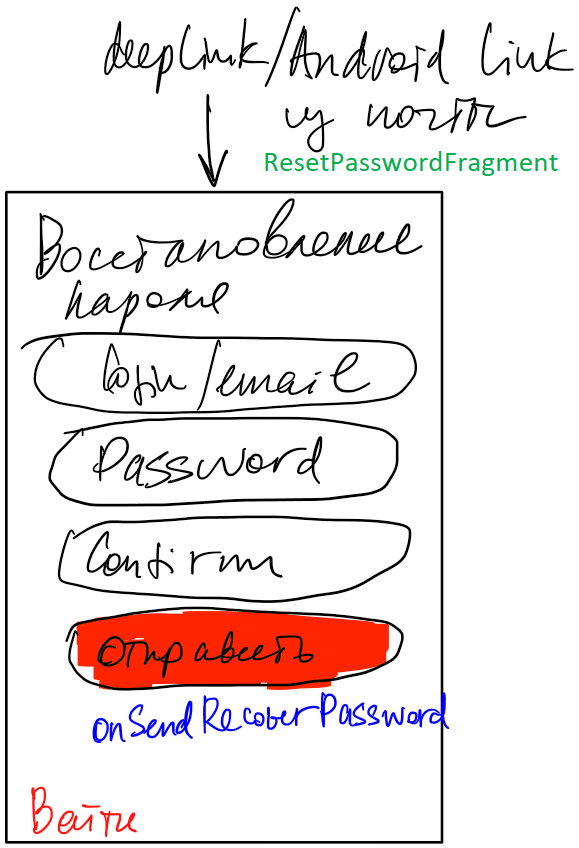
Пользователю надо ввести почту и нажать «Отправить», которое обрабатывается во viewModel. По результатам обработки viewModel производит переход на фрагмент ForgotPasswordResultFragment

Если пользователь вспомнил пароль, то он может вернуться на экран ввода пароля, нажав на «Войти», обрабатывается фрагментом



Вне зависимости от результата обработки на сервере пользователю сообщается, что если почта существует, то ему надо туда войти для получения дальнейших инструкций

Если пользователь вспомнил пароль, то он может вернуться на экран ввода пароля, нажав на «Войти», обрабатывается фрагментом

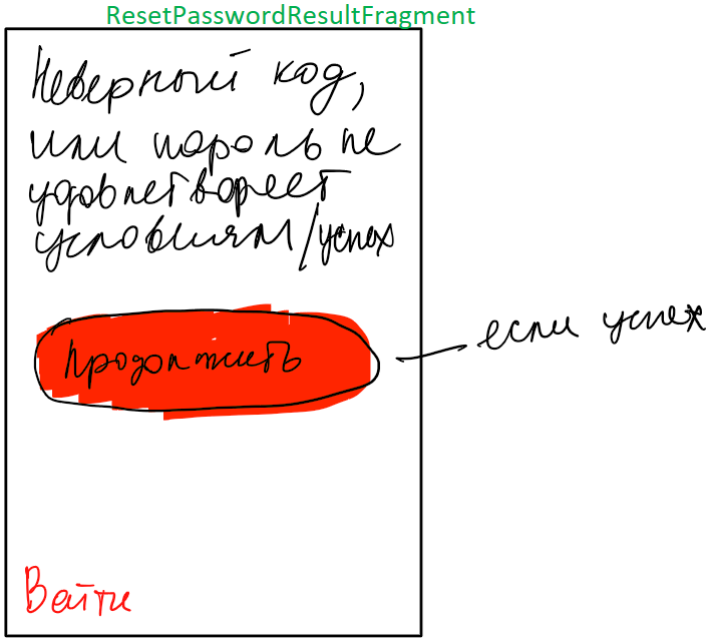


В этот фрагмент пользователь попадает, нажав ссылку вида https://my.goodwan.ru/auth/ResetPassword/{token}, которая пришла ему в почту в результате запуска процесса восстановления пароля.

Пользователю нужно ввести почтовый адрес, новый пароль два раза и нажать «Отправить», которая обрабатывается во viewModel. Возвращенный сервером результат сброса пароля viewModel передает в ResetPasswordResultFragment

Если пользователь передумал менять пароль и вспомнил старый, то он может вернуться на экран ввода пароля, нажав на кнопку «Войти», которая обрабатывается фрагментом

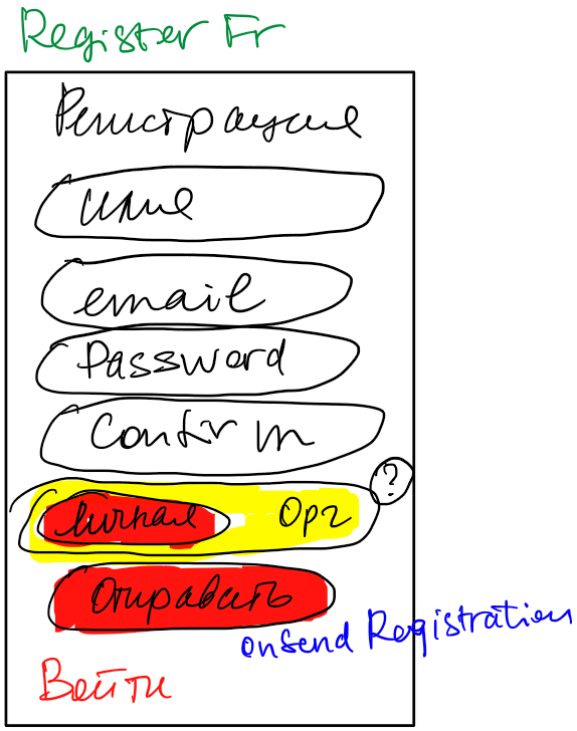
ResetPasswordResultFragment



Показываем пользователю результат сброса пароля:

- В случае успешной установки нового пароля пишем «все хорошо» и показываем кнопку «Продолжить», которая обрабатывается фрагментом и производит выход из подграфа Login
- Если смена пароля не удалась, то пишем «Все плохо»

Пользователь может заново нажать «Войти» (обрабатывается фрагментом), там ввести пароль или заново перейти в экран сброса пароля

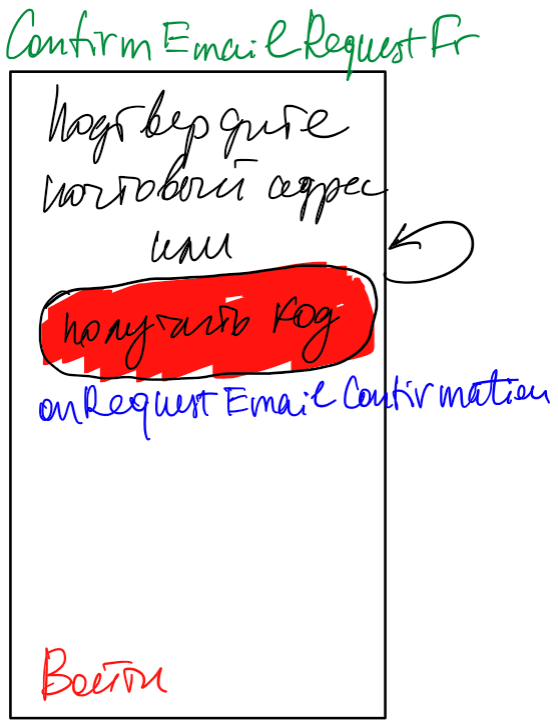


На этом экране пользователь может зарегистрировать новую учетную запись:

- Надо представиться, указать почтовый адрес и два раза ввести желаемый пароль
- Выбрать тип учетной записи «Личная» или «Организация»

    - Вывести маленькую круглую кнопку со знаком «?», которая выводит (обрабатывается фрагментом) модальное окно с примерным текстом: личная – только один пользователь, организация – может иметь несколько пользователей, но нужно управлять правами доступа. в дальнейшем личную можно преобразовать в организацию
- Кнопка «Отправить» обрабатывается во viewModel, которая передает введенные данные для обработки на сервер. По окончании обработки viewModel переходит в экран ConfirmEmailRequestFragment

Если пользователь вдруг вспомнил, что уже регистрировался, то он может перейти на экран ввода пароля (обрабатывается фрагментом)

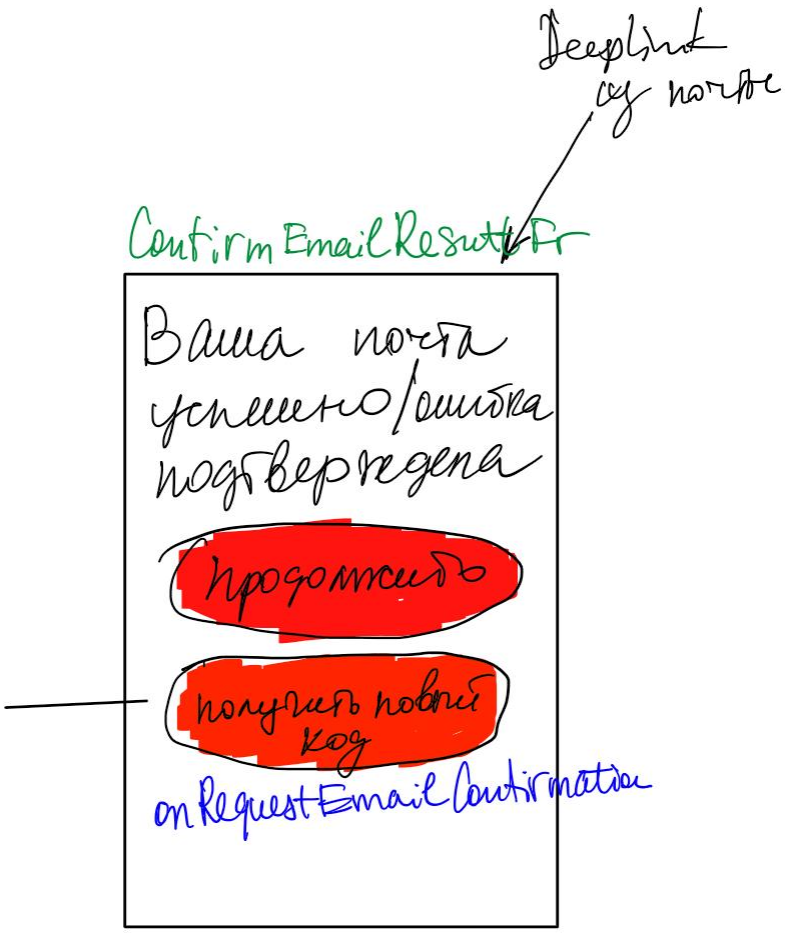


Сообщаем пользователю, что ему нужно подтвердить свой почтовый адрес по инструкциям в своей почте.

Подождав, пока с момента отправки кода пройдет 1 минута, viewModel показывает кнопку «Получить код», нажав на которую пользователь может повторно запросить отправку подтверждения своего почтового адреса, если по каким-то причинам ему ничего не пришло

Кнопка «Получить код» обрабатывается viewModel-ю, которая запрашивает сервер, скрывает кнопку и заново запускает счетчик

Войти – фрагмент переходит на экран ввода пароля



В этот фрагмент пользователь попадает, нажав ссылку вида https://my.goodwan.ru/auth/confirmemail?email={email}&token={token}, которая пришла ему в почту в результате запуска процесса подтверждения электронной почты

viewModel обрабатывает ссылку:

- Если произведен вход в систему, но почтовый адрес в ссылке не совпадает с адресом текущего пользователя, то выводим «адреса не совпадают» и показываем кнопку «Продолжить»
- Если вход в систему не был произведен или адреса совпадают, то viewModel запрашивает сервер, передав ему почтовый адрес и токен из ссылки
- По результатам обработки показываем «Успех» или «Ошибка» с причиной
- Если успех:
    - был произведен вход в систему – показываем кнопку «Продолжить»
    - не был произведен вход в систему – то показываем кнопку «Войти»
- В случае неудачи:
    - был произведен вход в систему – показываем «Получить новый код»
    - не было входа в систему – показываем кнопку «Войти»

ViewModel обрабатывает кнопку «Получить новый код» аналогично экрану ConfirmEmailRequestFragment

Фрагмент обрабатывает кнопки:

- «Продолжить» – выходит из подграфа Login
- «Войти» – переходит на экран ввода пароля

Верстка

https://www.figma.com/file/ozeUAxyhE9GBdHtNmkbCIB/Goodwan-Air?node-id=436%3A1795

Все экраны сверстаны в едином стиле на основе верстки экрана ввода пароля (его верстка сделана):

- Ориентация – только вертикальная
- Градиент на фоне
- Toolbar скрыт, полный экран
- ConstraintLayout
- Отступы по краям справа и слева 24dp
- Все элементы разбиваются на функциональные блоки, которые по смыслу должны находиться рядом при масштабировании экрана
- Верхний элемент каждого блока привязывается top-to-bottom к соответствующей горизонтальной направляющей (GuideLine), которая пропорционально позиционируется, исходя из формулы h/667, где h – расстояние от верхнего элемента блока до верхнего края экрана в figma

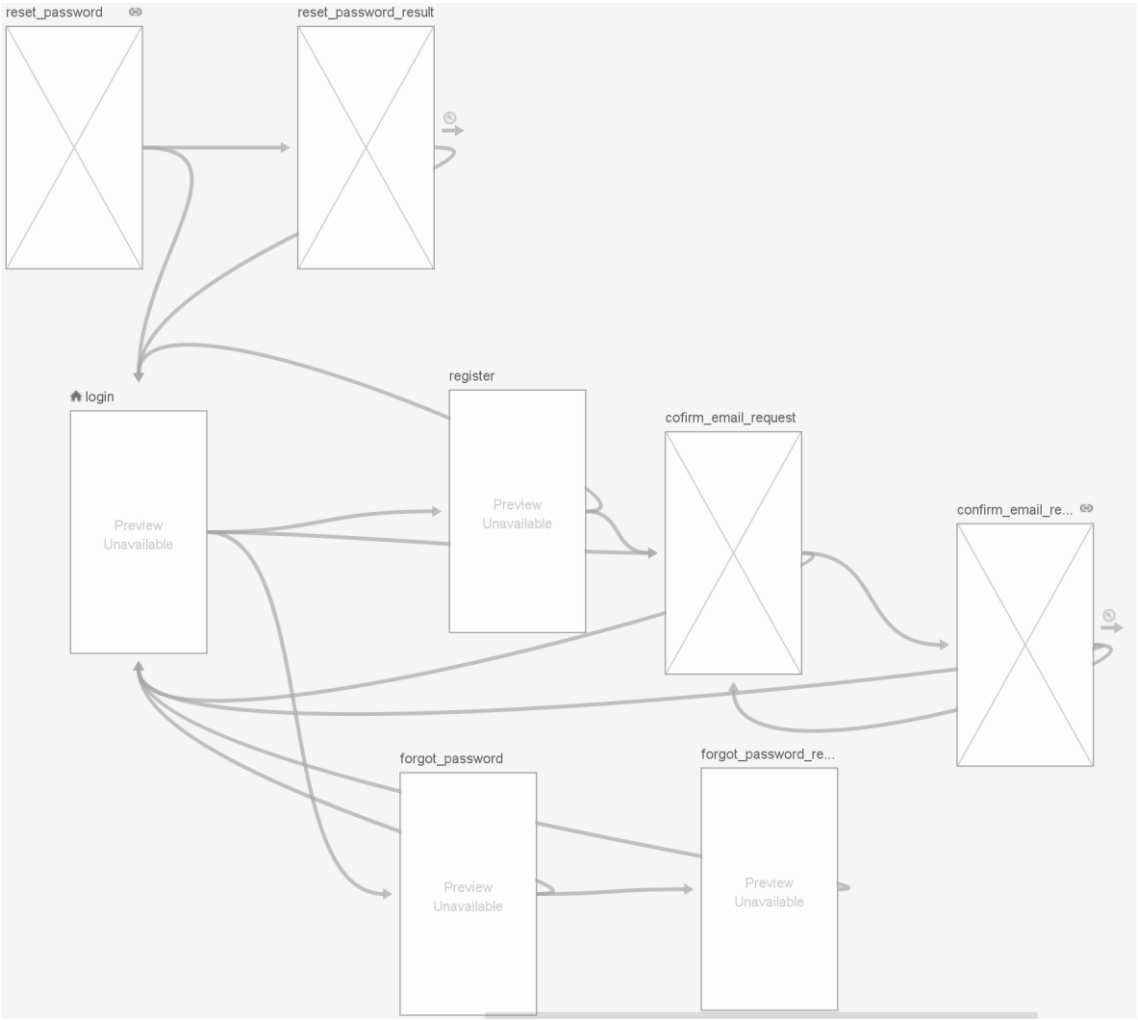
Стили

- TextMain – основные надписи
- TextHeader – заголовки («Здравствуйте»)
- ButtonPrimary – основная кнопка
- ButtonSecondary – прочая кнопка (белая в дизайне)
- ButtonTextPrimary – текстовая кнопка основная (красная «Войти» снизу)
- ButtonTextSecondary – текстовая кнопка прочая (белая «Забыли пароль?»)
- Input – EditText для ввода

Цвета

- ButtonPrimaryBackground – красный «Войти»
- ButtonPrimaryText – белый «Войти»
- ButtonSecondaryBackground
- ButtonSecondaryText
- InputBackground
- InputText
- InputHint

Реализация навигации



Все навигация между этими фрагментами находится внутри навигационного вложенного графа Login с точкой входа на фрагмент LoginFragment. Внутри этого графа отключена полностью кнопка Back и навигация возможна только через нажатия имеющихся на экране кнопок. Переходы не попадают в стек навигации

Навигация реализуется:

- Статическая навигация по нажатию кнопки на экране – в коде фрагмента слушаем нажатия на кнопки и вызываем нужное navigation action
- Динамическая, по результатам обработки – в коде view model вызываем ViewModelBase.navigationCommand.postvalue() для нужного navigation action

Состояние

Состояние живет в синглтоне state и обновляется классом Server по результатам обращения к серверу:

- isSignedIn: bool – был ли произведен вход в систему
- emailCofirmationUtc: datetime? – время отправки запроса на подтверждение почтового адреса

ViewModel и логика обработки

Управление видимостью кнопок и надписей:

- isSignInAvailable: LiveData<bool> – доступность кнопки Войти
- showSigninGreetings: LiveData<bool> – показывать успех или неудачу входа
- showPasswordRecoverySuccess: LiveData<bool> - показывать успех или неудачу процедуры восстановления пароля
- showEmailConfirmationSuccess: LiveData<bool> - показывать успех или неудачу процедуры подтверждения почтового адреса

Привязка к полям ввода:

- login: LiveData<string>
- password: LiveData<string>
- confirmPassword: LiveData<string>
- token: string – приходит в закодированном base64 виде

Обработчики кнопок – вызывают нужную doXXX suspend функцию:

- onSignIn – кнопка Войти
- onSendRegistration – кнопка Отправить экрана Регистрация
- onSendForgotPassword – кнопка Отправить экрана Забыли пароль
- onSendRecoverPassword - кнопка Отправить экрана Восстановить пароль
- onRequestEmailConfirmation – кнопка Получить код экранов «Запросить подтверждение адреса» и «Результат подтверждения почтового адреса»

Логика

Suspend функции, вызывают методы класса Server, подставляют туда нужные значения из полей ввода:

- doSignIn – выполняет вход
- doRegistration – отправляет на сервера запрос на создание новой учетной записи
- doRequestPasswordRecovery – запрашивает у сервера высылку токена на сброс пароля
- doRecoverPassword – просит сервер сбросить пароль
- doRequestEmailConfirmation – просит сервер выслать нового токена на подтверждение почтового адреса
- doConfirmEmail – просит сервер подтвердить почтовый адрес

Взаимодействие с сервером

Пометки для разработчика backend-а и класса Server

- При создании учетной записи на сервер отправляется локаль и часовой пояс для установки языка и часового пояса в профиле учетной записи
- Возвращать с сервера время последнего подтверждения почтового адреса