

Описание системы	2
Платформа для идентификации личности и работы с документами	2
Общие требования	2
Поддерживаемые устройства и платформы	3
Публикация приложения в App Store	3
Требования к SDK	4
Библиотека Sign To Login	4
Документация по использованию	4
Требования к Приложению	5
Регистрация	5
Аутентификация	9
Работа с PDF документами	10
Отправка документа по почте	13
Создание и регистрация iOS приложения	13
Требования к серверному API	14
Список запросов	14

Описание системы

Платформа для идентификации личности и работы с документами

Задача состоит в разработке SDK и Демонстрационного Приложения (далее Приложения) для iOS (iPhone & iPad), взаимодействующего с системой <http://signtologin.com/>

Роль SDK - предоставить крупным компаниям возможность создать собственное iPhone / iPad приложение, работающее в рамках их инфраструктуры и взаимодействующее с их копией <http://signtologin.com/>, настроенной под конкретные производственные нужды.

Роль Приложения - дать широкому кругу лиц и организаций доступ ко всем возможностям идентификации личности по подписи и совместной работы с документами, предоставляемым облачной системой <http://signtologin.com>

Общие требования

Функциональные возможности Приложения:

1. Регистрация пользователей в системе signtologin.com, создание образца подписи.
2. Аутентификация пользователей в системе по подписи.
3. Регистрация в системе iOS приложения, основанного на SDK.
4. Открытие PDF документов из приложения Mail и Documents директории.
5. Вставка подписи в PDF документ, сопровождающаяся проверкой по образцу.
6. Отправка подписанного PDF документа по электронной почте.
7. Проверка неизменности документа после подписания и достоверности поставленных на нём подписей.
8. Получение информации о личностях участников, поставивших подписи на PDF документе.

Приложение будет поставляться в виде универсальной (Universal) сборки: один *.app файл, работающий на iPhone, iPad и iPod touch.

Все перечисленные в ТЗ экраны имеют поддержку Portrait и Landscape режимов, если в комментариях не указано обратное.

Все долгие операции выполняются в отдельном потоке, в очереди, создаваемой механизмом GCD. При этом на экране появляется стандартный крутящийся индикатор прогресса.

Взаимодействие iOS SDK с сервисом signtologin.com осуществляется по JSON REST API. Рекомендуемая библиотека для работы с сервером - AFNetworking.

Приложение локализовано. В поставку включены Английский и Русский переводы. Интерфейс автоматически переключается на язык, установленный в настройках iOS.

Для построения графического интерфейса используются стандартные компоненты UI Kit. Поля позволяют вводить только разрешённые символы, клавиатуры имеют соответствующий тип, данные проходят валидацию по регулярным выражениям, форма автоматически скроллируется, чтобы текущее поле ввода не перекрывалось клавиатурой. Перечисленные практики применяются ко всем элементам UI, если явно не указано иное.

Поддерживаемые устройства и платформы

SDK и демонстрационное приложение должны работать на

- * iPhone 3GS, iPhone 4, iPhone 4S, iPhone 5
- * iPod touch 3G, 4G, 5G
- * iPad 1, 2, 3, 4, iPad mini

Версия операционной системы iOS 5.0 и выше, вплоть до iOS, выпущенной Apple на момент сдачи готового приложения.

Поддержка будущих версий iOS с большой вероятностью выполнится автоматически, но в случае возникновения несовместимости из-за изменений на стороне Apple, должна обсуждаться в рамках дополнения к данному ТЗ.

Публикация приложения в App Store

В объём работ входит помощь в публикации Приложения в Apple Store под профайлом Заказчика.

Поскольку приложение создаётся как универсальное (iPhone / iPod touch + iPad), то в App Store оно будет отмечено знаком “+”.

Требования к SDK

Библиотека Sign To Login

SDK должно быть оформлено в виде статической библиотеки signtologin.a, собранной для i386 и armv7 архитектур. Библиотека поставляется с набором заголовочных файлов *.h.

Демонстрационное приложение не должно иметь своего кода взаимодействия с Sign To Login сервером, отрисовки интерфейса пользователя или бизнес логики, необходимого для реализации [Требований к Приложению](#). Весь перечисленный в спецификации функционал работает за счёт использования интерфейсов из SDK. Таким образом, разработчику iOS приложения, основанного на Sign To Login SDK нужно написать минимум кода.

В заголовочные *.h файлы должны быть включены только те свойства и методы, которые действительно необходимы. Все поля и детали реализации выносятся в *.m файлы за счёт использования анонимных категорий.

Поскольку SDK может быть использовано как в iPhone, так и в iPad приложениях, нужно иметь возможность открывать контроллеры с UI через стэк навигации, модально, поверх существующих экранов (popover), через корневой контроллер и посредством простого добавления view в иерархию.

Документация по использованию

В поставку входит инструкция для разработчиков по интеграции библиотеки в iOS приложения, создаваемые в XCode 4.6. Инструкция представлена на русском и английском языках.

Каждый интерфейс, свойство и метод, доступные в *.h файлах SDK должны быть тщательно прокомментированы на английском языке.

Требования к Приложению

Регистрация

При запуске, на время инициализации Приложения, пользователь видит экран - заставку в течение 1-3 секунд.

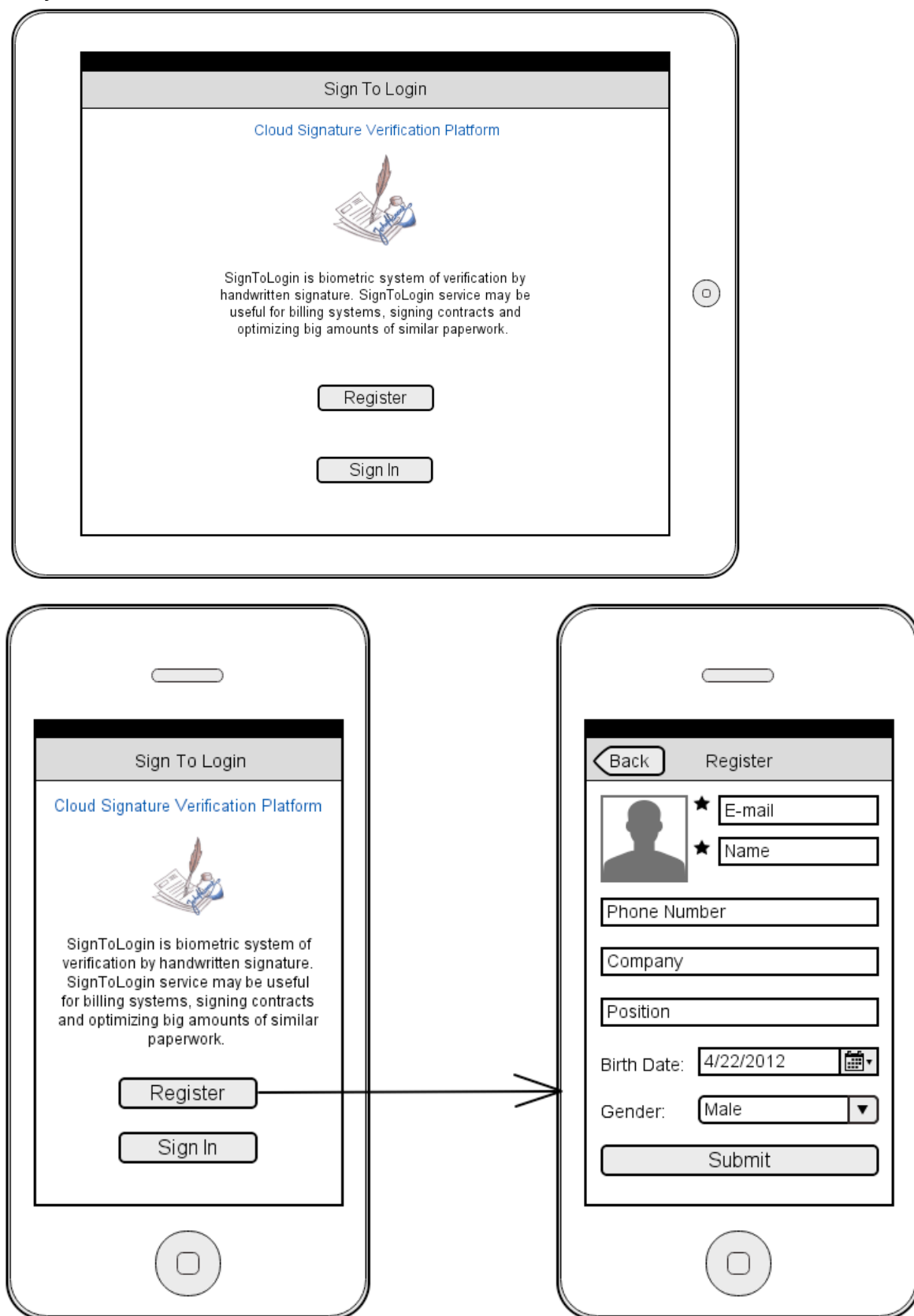
Рисунок 1



Настройки пользователя сохраняются в NSUserDefaults. Таким образом, если регистрация в Приложении уже пройдена, и образец подписи создан, то Приложение запомнит e-mail и все введенные данные. После заставки сразу откроется экран со списком PDF документов, хранящихся в Documents Directory на iOS устройстве (*Рисунок 7*)

В случае, когда приложение открывается впервые, пользователь попадает на стартовый экран, с которого можно перейти на регистрацию, либо пройти авторизацию, если регистрация уже была осуществлена на другом iOS устройстве.

Рисунок 2



Регистрация состоит из двух шагов:

1. Заполнение профайла
2. Создание образца подписи

Профайл включает в себя обязательные поля:

- * Адрес электронной почты
- * Имя пользователя

И дополнительные поля, которые позже можно отредактировать на signtologin.com:

- * Картинку
- * Номер телефона
- * Название компании
- * Должность
- * Дату рождения
- * Пол

Помимо полей, введенных пользователем, приложение по возможности определяет страну и город, а также широту и долготу места, из которого была произведена регистрация. Если пользователь запретил узнавать его географические координаты, то эти данные на сервер не отсылаются.

Также автоматически посылаются тип устройства, API Key приложения, версия iOS, текущая дата. (см. [Требования к серверному API](#))

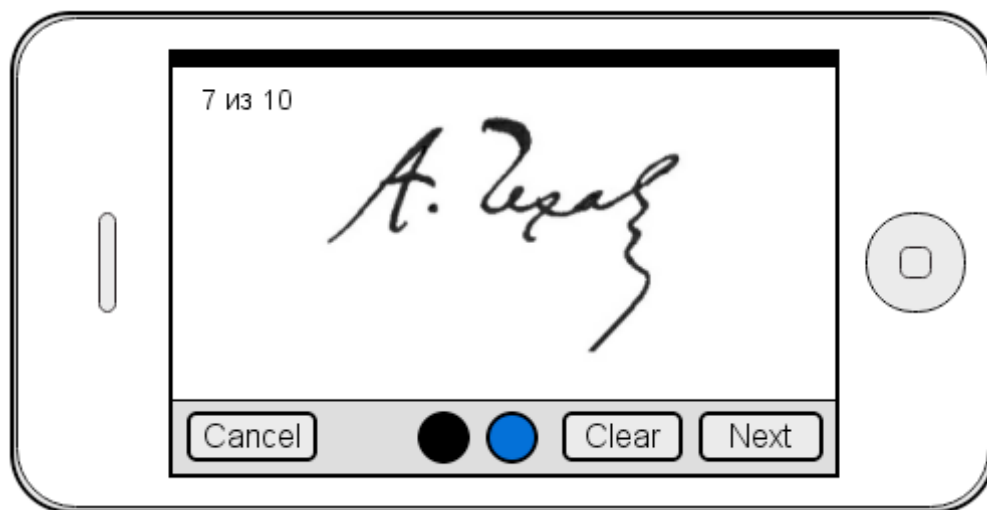
Экран для создания образца подписи реализован на Quartz. Толщина линии зависит от скорости движения пальца: чем медленнее, тем тоньше. Есть возможность выбирать синий и чёрный цвет пера.

По производительности и внешнему виду нужно ориентироваться на приложение <https://itunes.apple.com/us/app/signnow-sign-fill-pdf-word/id489262811>

Для создания образца на сервере, необходимо получить 10 подписей пользователя. Точки по каждой подписи сохраняются в отдельный массив. Поля с данными профайла и 10 подписей посылаются на сервер единым запросом.

При нажатии на кнопку Cancel пользователь попадёт обратно на форму заполнения профайла.

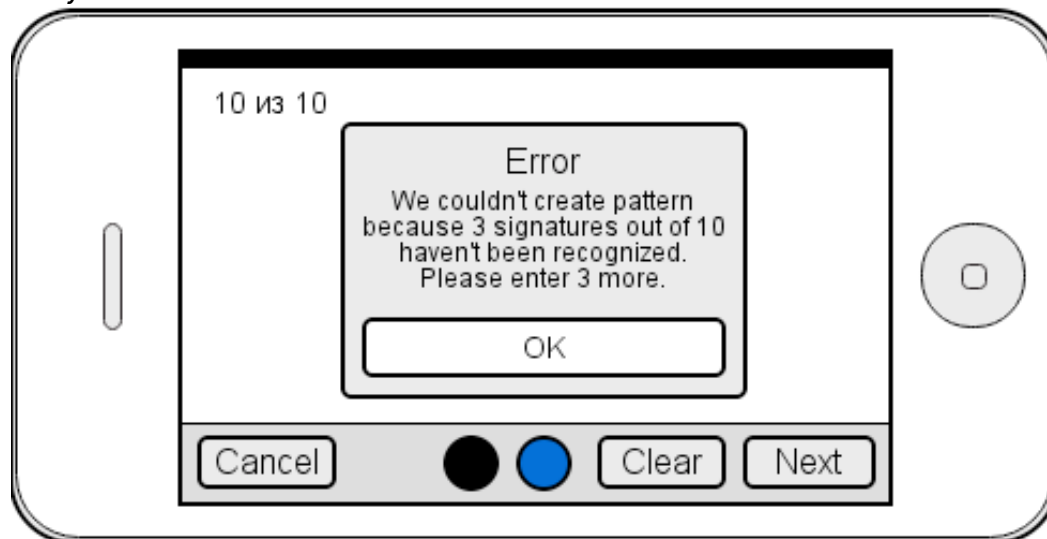
Рисунок 3



В случае, если запрос на создание образца подписи прошёл успешно, пользователь получает UIAlertView с сообщением “Образец создан” и переходит на экран со списком PDF документов (Рисунок 7)

Если сервер не смог создать паттерн на основе 10 подписей, пользователь получает уведомление об ошибке и просьбу заново ввести несколько подписей (от 1 до 9). В результате формируется аналогичный запрос на регистрацию и осуществляется очередная попытка.

Рисунок 4



Аутентификация

Аутентификация состоит из двух отдельных запросов, поскольку мы должны удостовериться, что пользователь уже проходил регистрацию именно на iOS устройстве.

Таким образом, первым запросом отправляется e-mail пользователя:

Рисунок 5



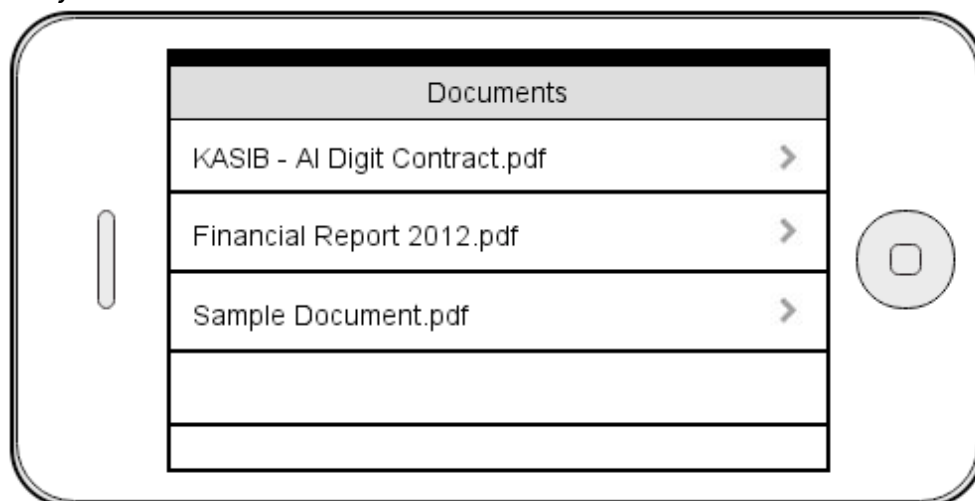
В случае, если e-mail не существует, Приложение уведомляет пользователя об ошибке и переводит на стартовый экран (Рисунок 2). Если e-mail уже существует, но образец подписи был создан не в iOS приложении, то пользователю будет предложено создать новый образец (Рисунок 3) Если e-mail и образец подписи есть в системе, но пользователь впервые открывает приложение с данного устройства, то переходим на авторизацию:

Рисунок 6



Синхронизация Documents директории между несколькими iOS устройствами не предусмотрена. Способ передачи и хранения PDF документов определяется пользователем. В системе Sign To Login документы не хранятся.

Рисунок 7



Работа с PDF документами

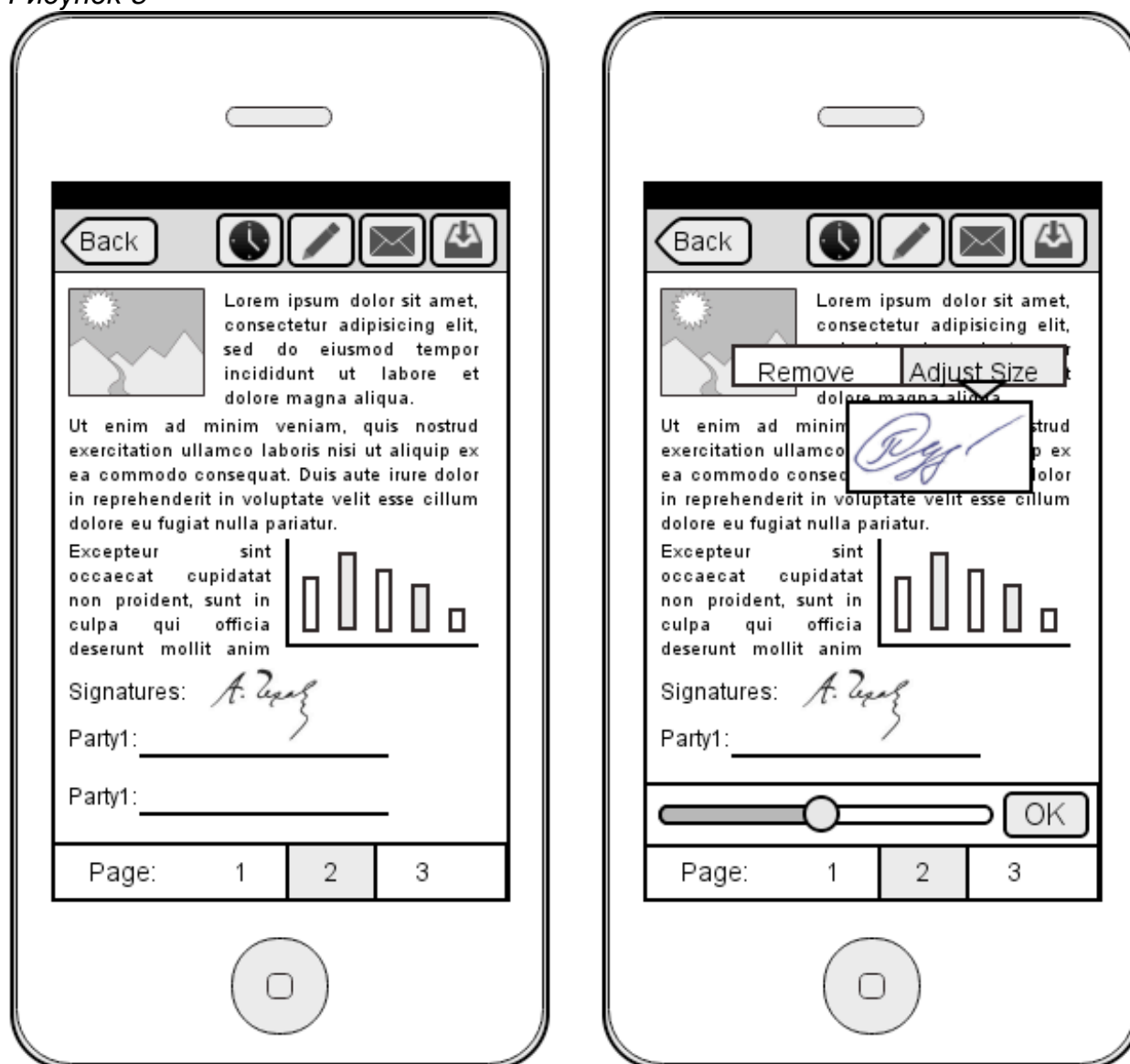
При открытии PDF из почты, или выбора из списка, пользователь попадает в экран просмотра (Рисунок 8).

При нажатии на кнопку “Подписать”, пользователю предлагается поставить новую подпись (см. Рисунок 6). Подпись проверяется сервером. Авторизация всегда проходит удачно, однако подпись совпадает с образцом с некоторой вероятностью. Таким образом, следующий пользователь, открывший документ, будет предупреждён, если процент совпадения низок.

При нажатии на кнопку “сохранить” на сервер отправляется хэш код подписанного PDF документа. Хэш на сервере обновляется после добавления каждой подписи. Если, после подписания очередным участником, договор был изменён вне приложения, то при открытии документа в приложении каждый следующий участник получит соответствующее предупреждение.

Картинку с подписью можно удалить, перенести пальцем на нужное место в пределах данной страницы, или изменить её размер. К подписи пользователю предлагается приложить свою фотографию, взятую в момент подписания документа.

Рисунок 8



Вместе с картинкой подписи в PDF вставляется ссылка вида

<http://signtologin.com/documents/?documentid=31a8de70>

Если пользователь открыл PDF на компьютере, то по этой ссылке ему будет показана вся история добавления подписей. Если ссылка была открыта на iOS устройстве (вне зависимости от приложения, в котором пользователь просматривал PDF), откроется Safari Mobile, JavaScript определит, установлено ли Приложение Sign To Login, и если нет, то предложит пользователю скачать его в App Store. Если же Приложение установлено, то будет открыта история добавления подписей на iOS устройстве (рисунок 9).

История подписей также может быть открыта непосредственно из окна с PDF (Рисунок 8) при нажатии на кнопку “История”. На странице <http://signtologin.com/documents/> должен быть вставлен код, определяющий, пришёл ли пользователь с iOS устройства или из обычного браузера:

```
var userAgent = window.navigator.userAgent;

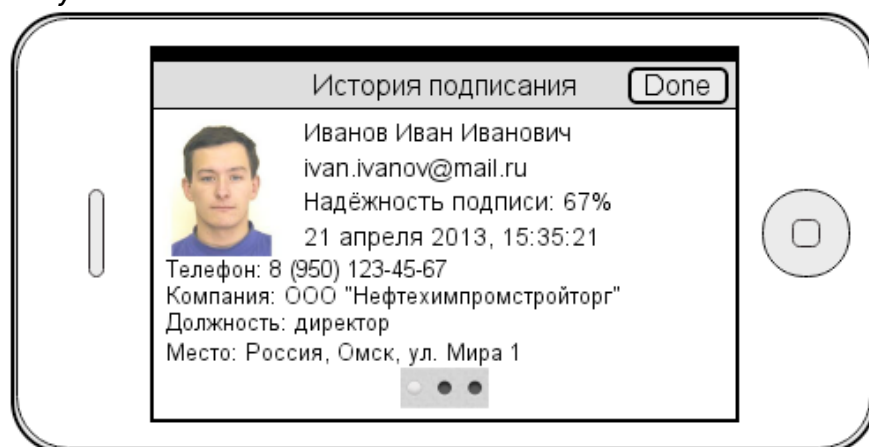
if (userAgent.match(/iPad/i) || userAgent.match(/iPhone/i)) {
    // iPad or iPhone
}
else {
    // Anything else
}
```

Если пользователь открыл ссылку на своём iPad / iPhone, то необходимо вызвать метод startSignToLogin(); определённый следующим образом:

```
<script type="text/javascript">
function startSignToLogin() {
    window.location = 'stl://user/?documentid=31a8de70';
    setTimeout(function() {
        if (confirm('You don't have Sign To Login app. Download it now?')) {
            window.location = "itms-apps://itunes.com/apps/signtologin/";
        }
    }, 300);
}
</script>
```

На iOS устройстве история подписания документа постранично отображает данные по каждой поставленной подписи:

Рисунок 9



Отправка документа по почте

После сохранения подписанный документ можно отправить по e-mail из приложения. При нажатии на полученный по e-mail PDF в стандартной программе Mail на iOS появится диалог, предлагающий список программ для открытия документа, в том числе Приложение Sign To Login, если оно уже установлено.

При открытии PDF документа приложением, файл автоматически копируется в Documents директорию устройства для упрощения последующего доступа.

Создание и регистрация iOS приложения

Пользователи имеют возможность зарегистрировать и разработать собственное iOS приложение на основе SDK.

Для этого необходимо зайти в профайл на signtologin.com и добавить приложение, указав:

- * Название
- * Bundle ID
- * Иконку
- * Адрес электронной почты ответственного за приложение
- * Веб страничку приложения

Сервер сгенерирует уникальный API Key, необходимый для подключения SDK к API системы.

Требования к серверному API

Список запросов

Звёздочкой * помечены обязательные параметры.

URL: <http://signtologin.com/register/>

POST параметры:

- * apikey - ключ, выдаваемый при регистрации iOS приложения.
- * email - строка e-mail
- * user_name - строка с именем пользователя
- company_name - компания пользователя
- position - должность в компании
- phone_number - номер телефона с кодом страны
- birthday - дата рождения в формате DD.MM.YYYY
- sex - пол (male / female)
- photo - картинка
- location - { "lat" : 29.1171176, "lon" : 30.1171176 }
- country - страна (определяется по reverse geo coding API)
- city - город
- address - адрес местонахождения пользователя
- * time - текущее время и дата в формате "DD.MM.YYYY HH:MM:SS"
- * os_version - версия операционной системы
- * device_version - версия устройства
- * resolution - { "width" : 1024, "height" : 768 } - размер экрана
- * pressure - давление (всегда = 0 для iOS)
- * "signatures": [
 - {
 - "pressure": 0,
 - "dot_count": 3,
 - "signature_contents": [
 - {
 - "x": 1068,
 - "y": 52,
 - "p": 0
 - },
 - {
 - "x": 1067,
 - "y": 52,
 - "p": 0
 - },
 - {

```

        "x": 1065,
        "y": 53,
        "p": 0
    },
    ...
]
},
...
{
    "pressure": 0,
    "dot_count": 2,
    "signature_contents": [
        {
            "x": 1067,
            "y": 52,
            "p": 0
        },
        {
            "x": 1065,
            "y": 53,
            "p": 0
        },
    ]
}
]

```

Ответы:

{ "res": "0", "sessionid": "f19a4b68d31ec73a" } если приложение допущено для работы с системой и образец подписи успешно создан.

{ "res": "1", "error": "Application with key '550e8400-e29b-41d4-a716-446655440000' is not registered or disabled" } если приложение было запрещено либо API Key указан неверно.

{ "res": "2", "error": "Not all signatures a valid", "count_required": 3, "count_exist": 7, "count_model": 10, "count_incorrect": 3 } если не все подписи пользователя были распознаны и нужно ввести ещё count_required подписей.

URL: <http://signtologin.com/login/>

POST параметры:

- * apikey
- * email
- * os_version

* device_version

Ответы:

{ "res": "1", "error": "Application with key '550e8400-e29b-41d4-a716-446655440000' is not registered or disabled" }

{ "res": "2", "error": "User does not exist" } если пользователя с данным e-mail не существует в базе

{ "res": "3", "error": "Registration is required for this type of device" } если пользователь существует, но ни разу не проходил регистрацию

{ "res": "0" } если пользователь уже зарегистрирован

URL: <http://signtologin.com/authenticate/>

POST параметры:

* apikey

* email

photo

location

country

city

address

* time

* resolution

* pressure

* "signature":

```
{
  "pressure": 0,
  "dot_count": 176,
  "signature_contents": [
    {
      "x": 1068,
      "y": 52,
      "p": 0
    },
    {
      "x": 1067,
      "y": 52,
      "p": 0
    },
    {
      "x": 1065,
```



```

        "y": 53,
        "p": 0
      },
      ...
    ]
  }

```

Ответы:

{ "res": "0", "similarity": 0.74, "signatureid": "d31ec73af19a4b68" } если подпись совпала с образцом на 74% и пользователь допущен до работы. При этом данные о подписи запоминаются сервером.

URL: <http://signtologin.com/savedocument/>

POST параметры:

- * apikey
- * email
- * previous_hash - хэш документа, открытого в редакторе подписей
- * document_hash - хэш код документа, сохранённого в приложении.
- * signatures : ["d31ec73af19a4b68", ...] - данные о подписях, поставленных пользователем. При первом сохранении документа подписи отсутствуют.

Ответы:

{ "res": "0", "documentid": "31a8de70" } если запись о документе успешно внесена в базу и документу назначен уникальный идентификатор.

URL: <http://signtologin.com/opendocument/>

POST параметры:

- * apikey
- * email
- * documentid - уникальный идентификатор документа
- * document_hash - хэш код документа, вычисленный на момент открытия в редакторе подписей

Ответы:

{ "res": "0", "history": [JSON с информацией о всех подписях, поставленных на документ] } если документ не был изменён после последнего сохранения в приложении

{ "res": "1", "error": "Document has been updated outside of the application", "history":
[JSON с информацией о всех подписях, поставленных на документ] } если
документ был изменён вне приложения