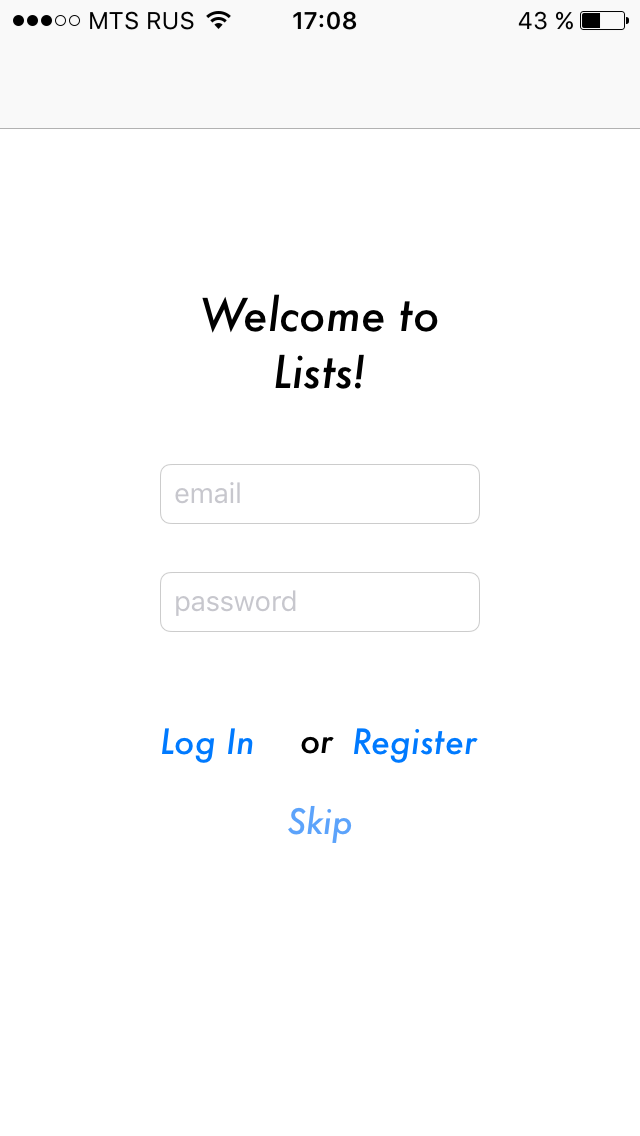
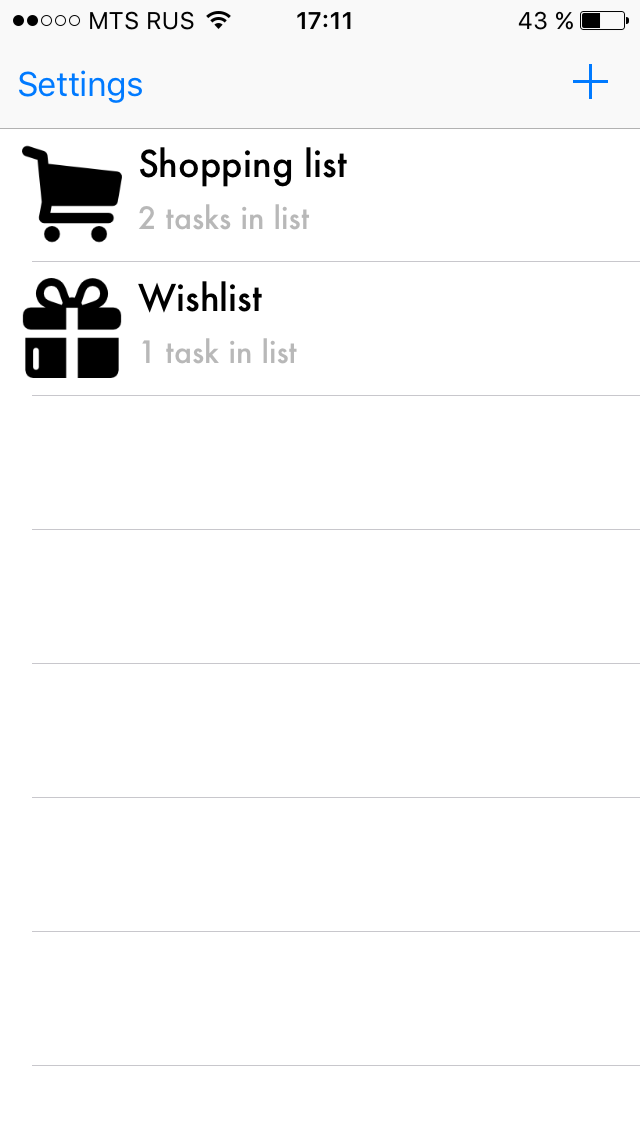
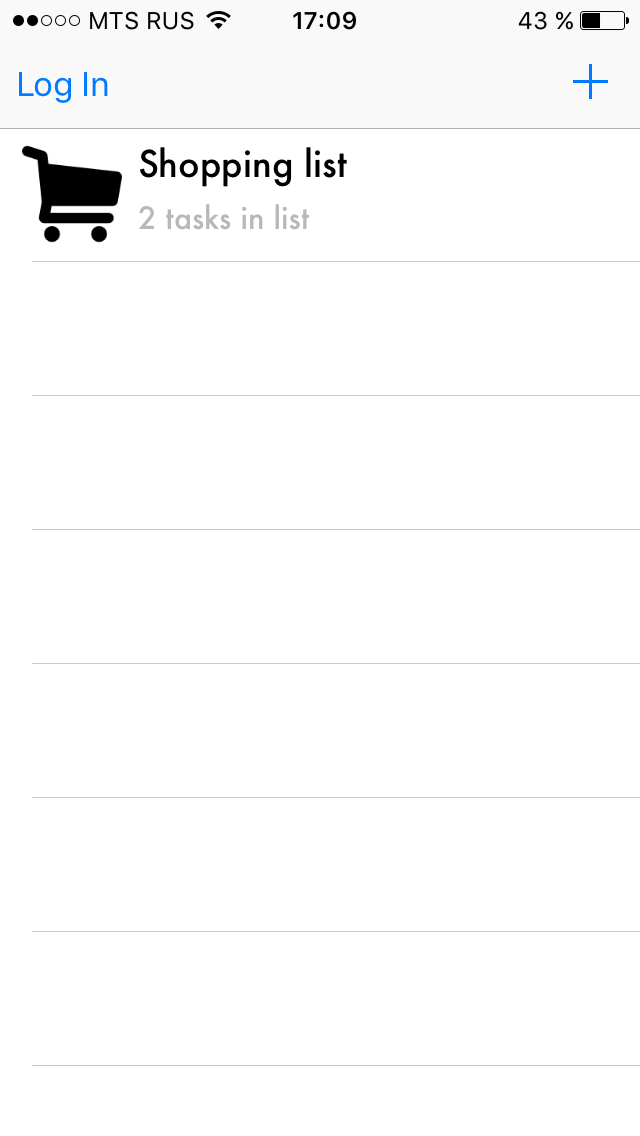
**Коровина Мария, ББИ168**

**Пояснительная записка к контрольному домашнему заданию по программированию**

1. **Тема проекта:** «Создание приложения управления списками»
2. **Адрес репозитория:** <https://github.com/eyentei/lists>
3. **Аннотация:** Пользователь может зарегистрироваться в приложении с помощью своей электронной почты и пароля (или пропустить этот шаг), авторизоваться; Основной функционал приложения состоит в создании, редактировании и удалении списков (покупок, задач, желаний и т.д.). Если пользователь пропустил регистрацию/авторизацию, то созданные им списки сохраняются в локальной базе данных. Если после этого он решит зарегистрироваться, созданные им до этого списки будут перенесены в его аккаунт. Аккаунт и все связанные с ним данные хранятся на сервере, в онлайн-версии базы данных и обновляются в реальном времени. Кроме этого, для авторизованного пользователя существует возможность поменять пароль и выйти из аккаунта.
4. **Используемые технологии и библиотеки:** Мною было создано приложение на языке Swift для платформы iOS. Из внешних библиотек использовалась СУБД Realm, а также ее дополнение для хранения данных онлайн Realm Object Server. Я выбрала Realm как СУБД для своего приложения, потому что она легка в использовании, быстрее других вариантов (SQLite, Core Data), а также хорошо документирована. Онлайн-версия БД хранится на хостинге DigitalOcean (сервер на Ubuntu 16.04.2 x64).
5. **Интерфейс программы:**

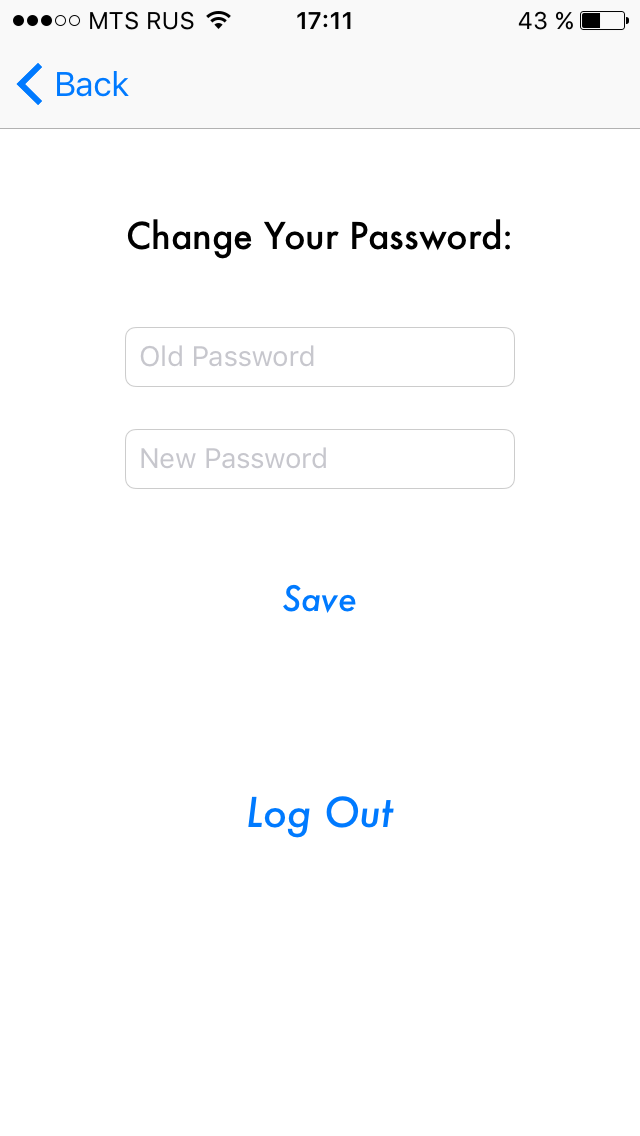
****

Первый экран приложения. Здесь пользователь может зарегистрироваться, войти в аккаунт или пропустить этот шаг и приступить к созданию списков без авторизации.

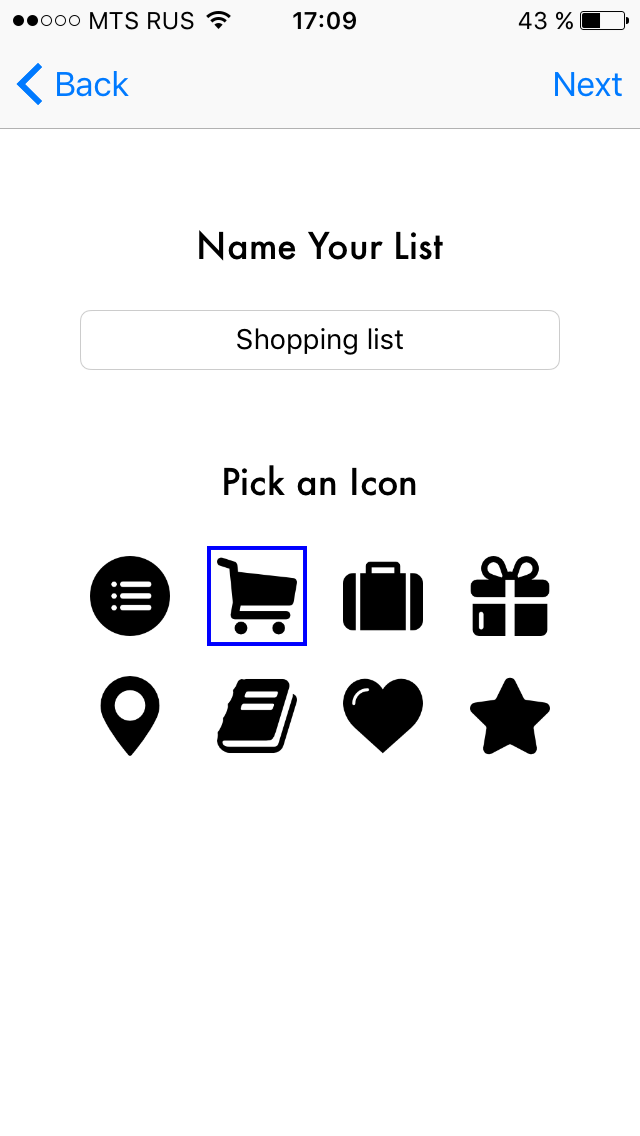
****

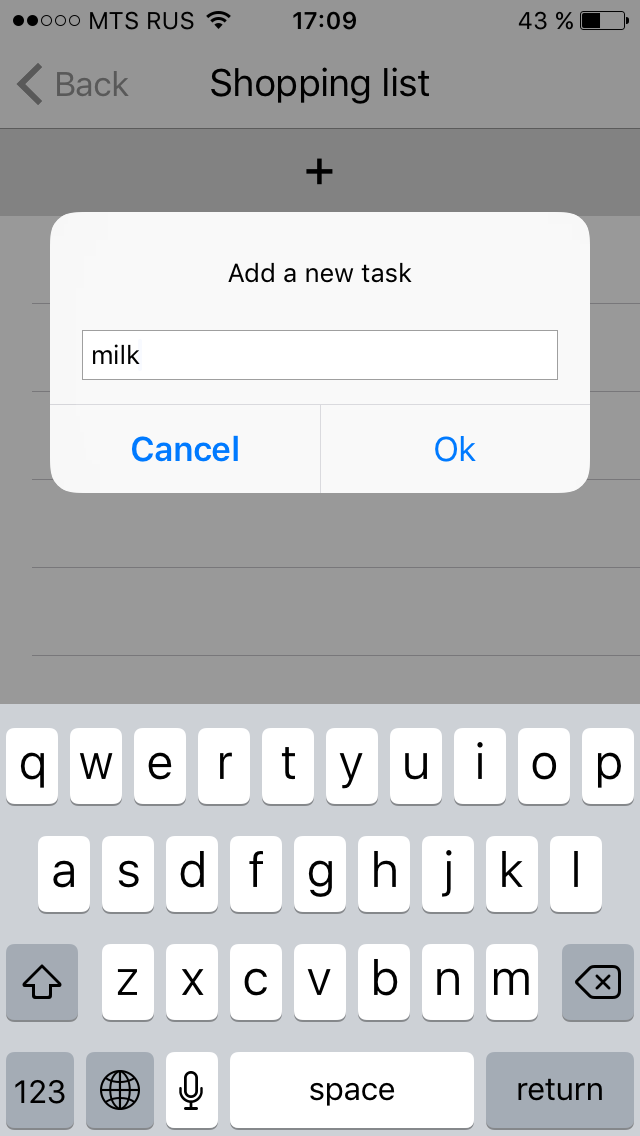
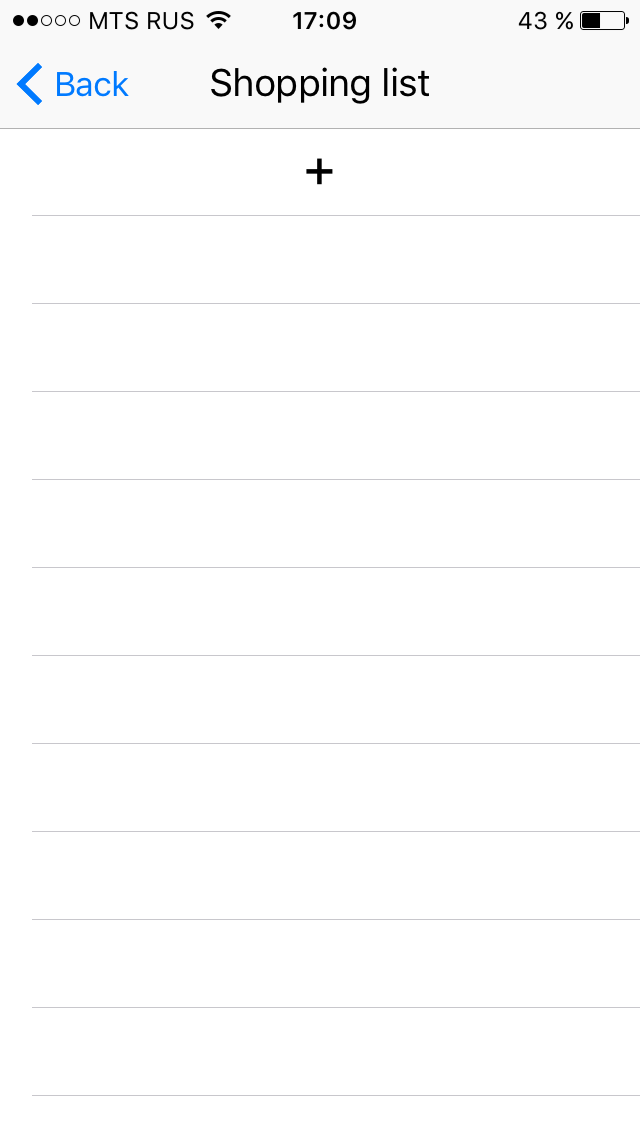
1. Второй экран неавторизованного и 2) авторизованного пользователя.

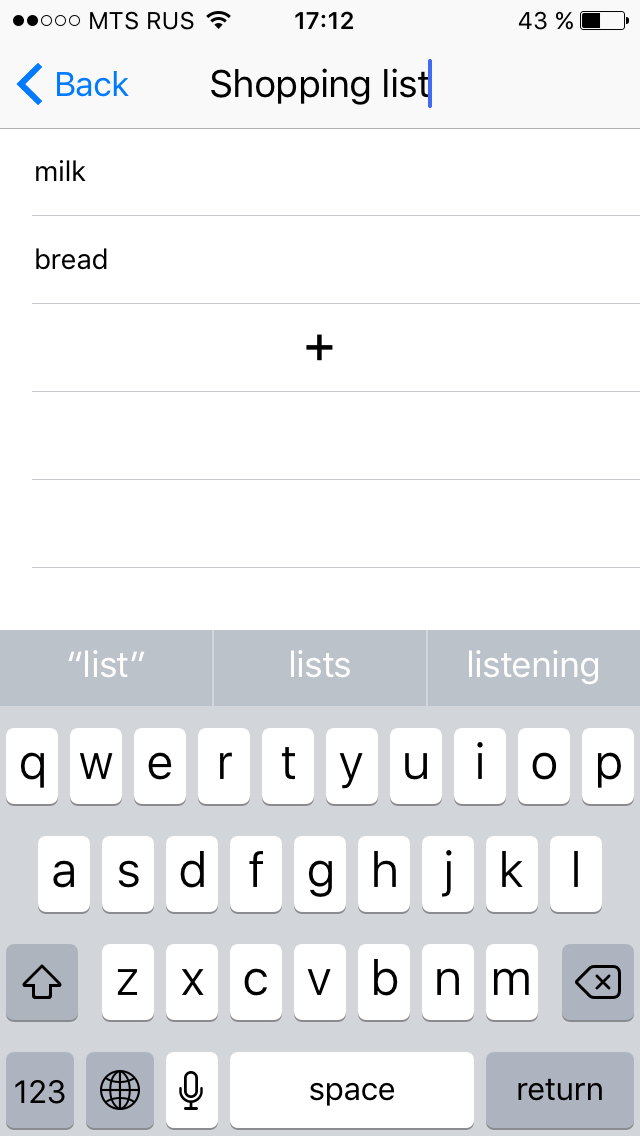
Разница в функционале левой верхней кнопки – у неавторизованного пользователя она ведет на первый экран, где он может зарегистрироваться или войти в аккаунт. У авторизованного она ведет на экран настроек:



На экране настроек пользователь может изменить свой пароль или выйти из аккаунта с переходом на первый экран.

При нажатии кнопки “+” в правом верхнем углу второго экрана происходит переход на экран инициализации списка. На этом экране можно задать название списка и иконку, подходящую по тематике. Кнопка Back возвращает к предыдущему экрану без сохранения, кнопка Next записывает пока что пустой список в базу данных и переводит к следующему экрану.



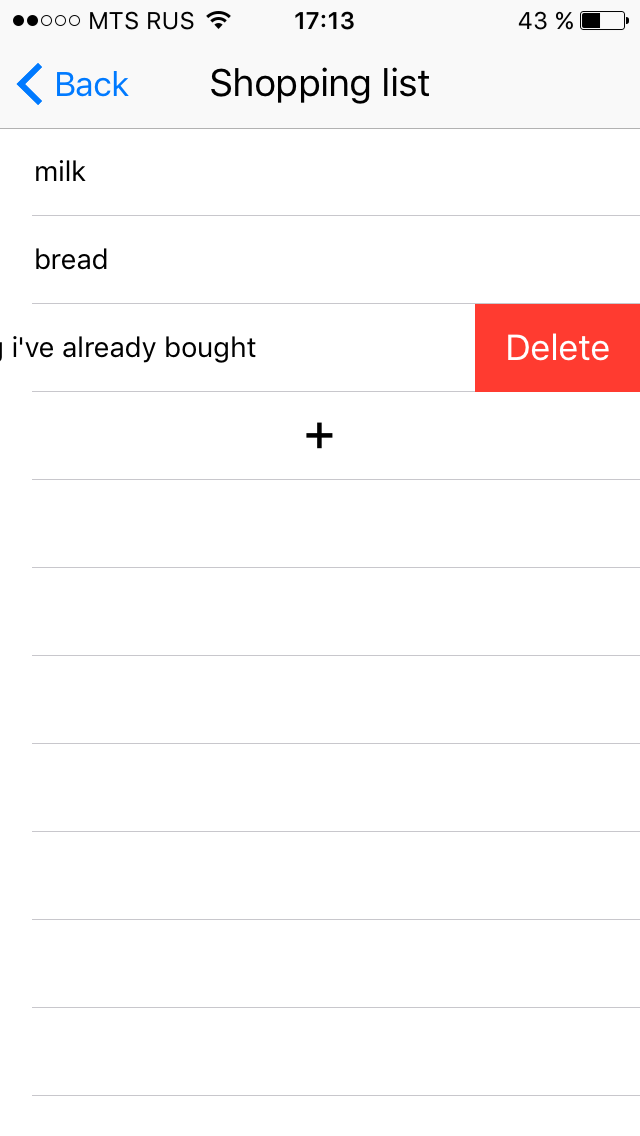
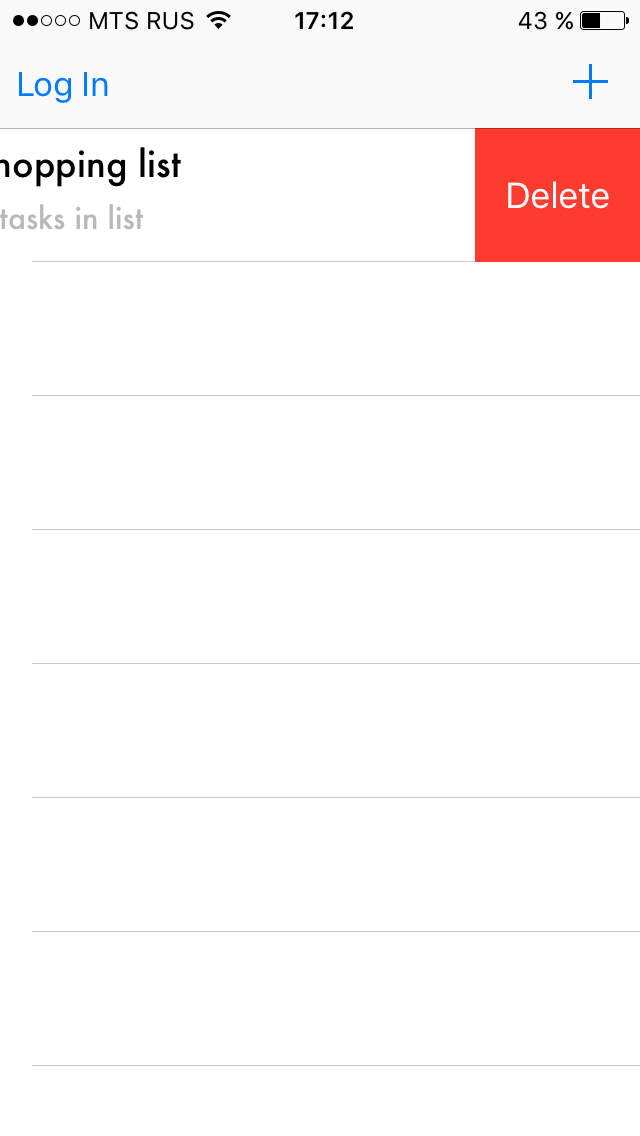


Следующий экран – экран добавления и редактирования задач. 1) Пустой список

2) Добавление задачи (по нажатию «+»). При нажатии Ok задание запишется в БД

3) Редактирование заголовка. При окончании редактирования запись в БД обновится (окончание редактирования – нажатие return или тап в любом месте экрана).

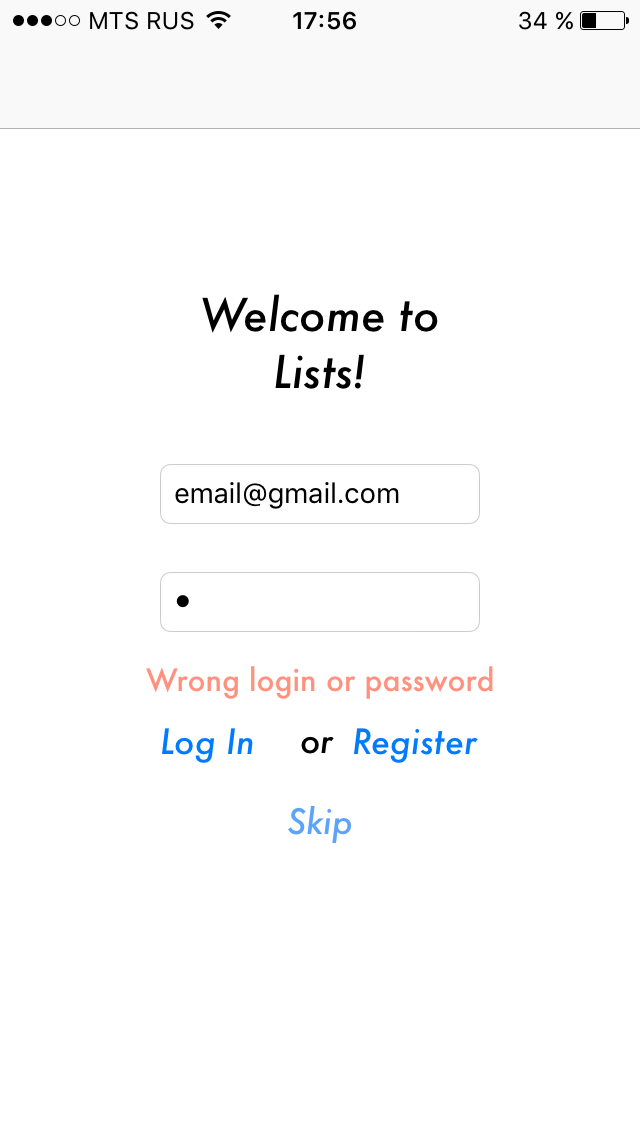
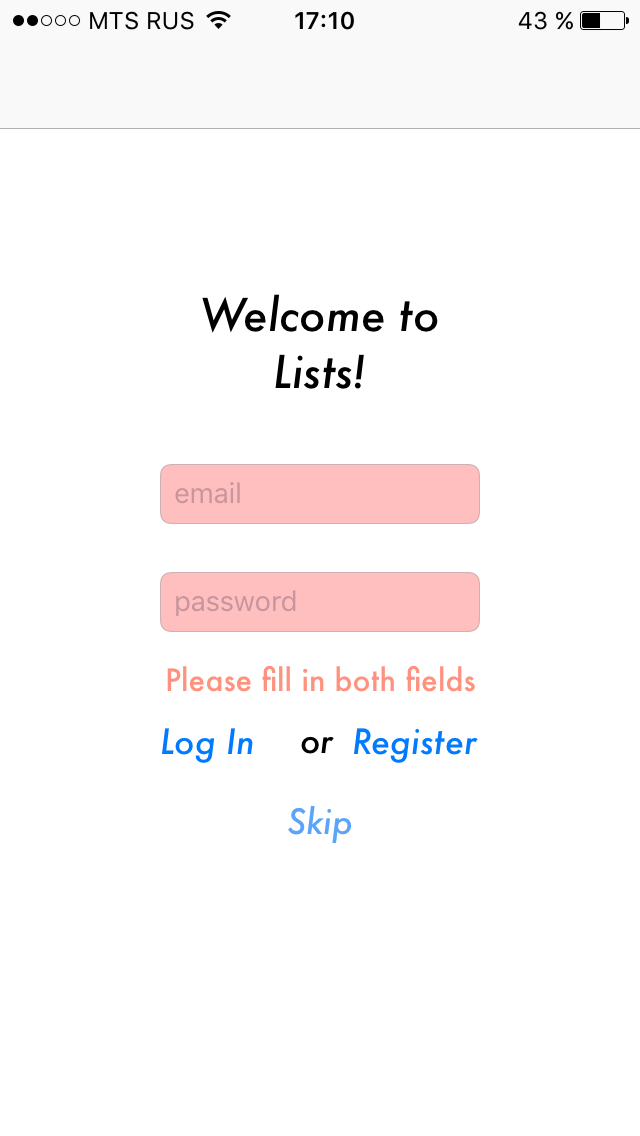
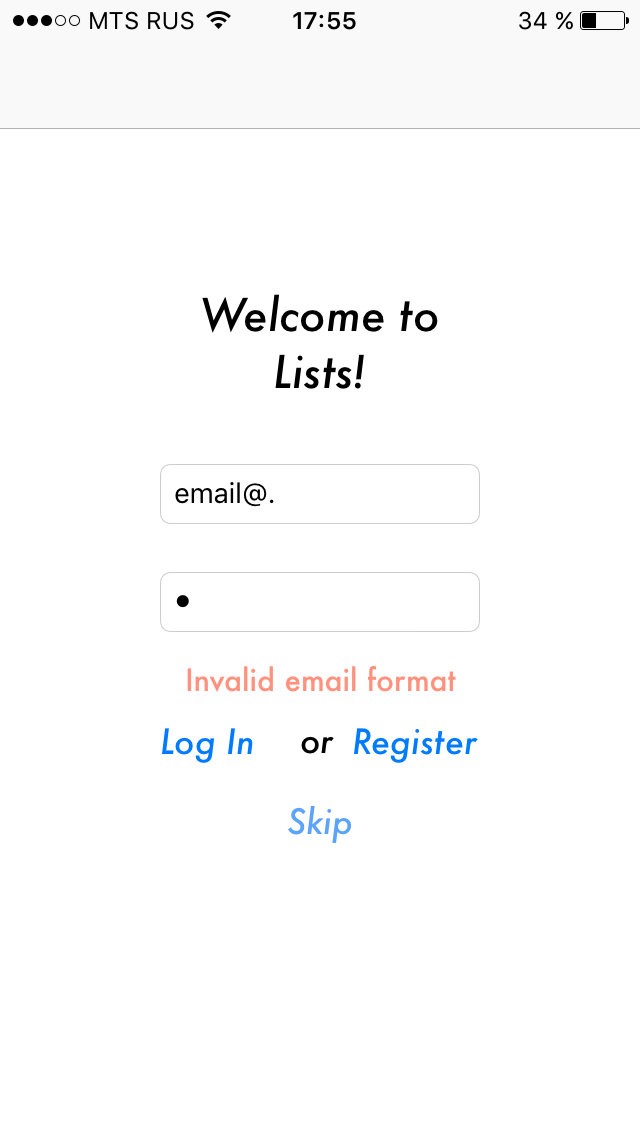
Кроме редактирования, возможно также удаление – как отдельной задачи, так и списка целиком.

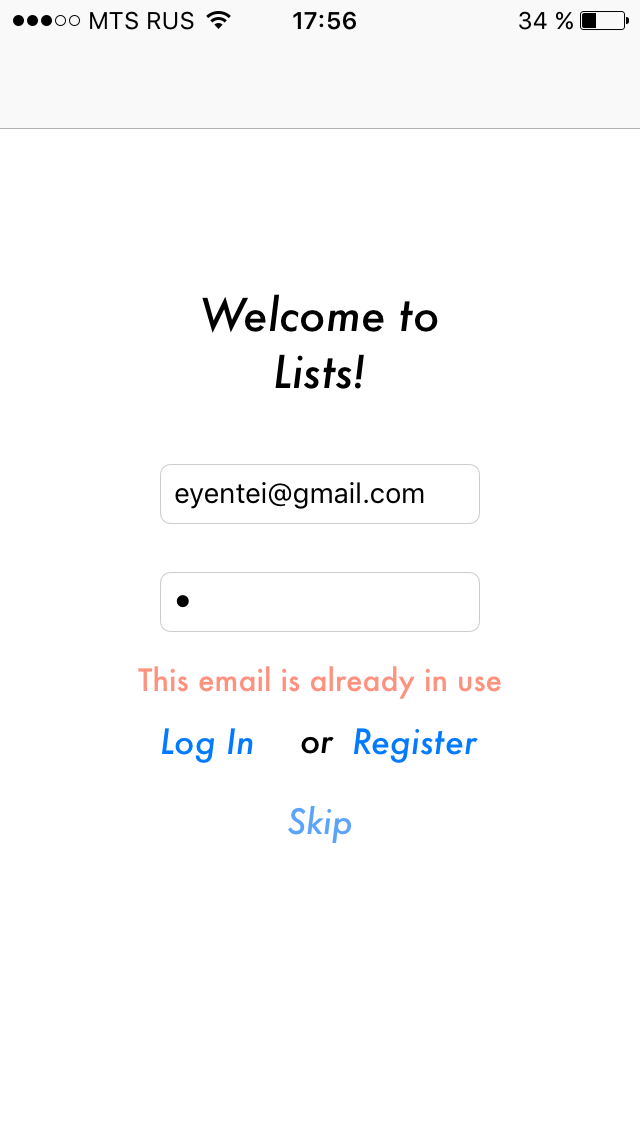
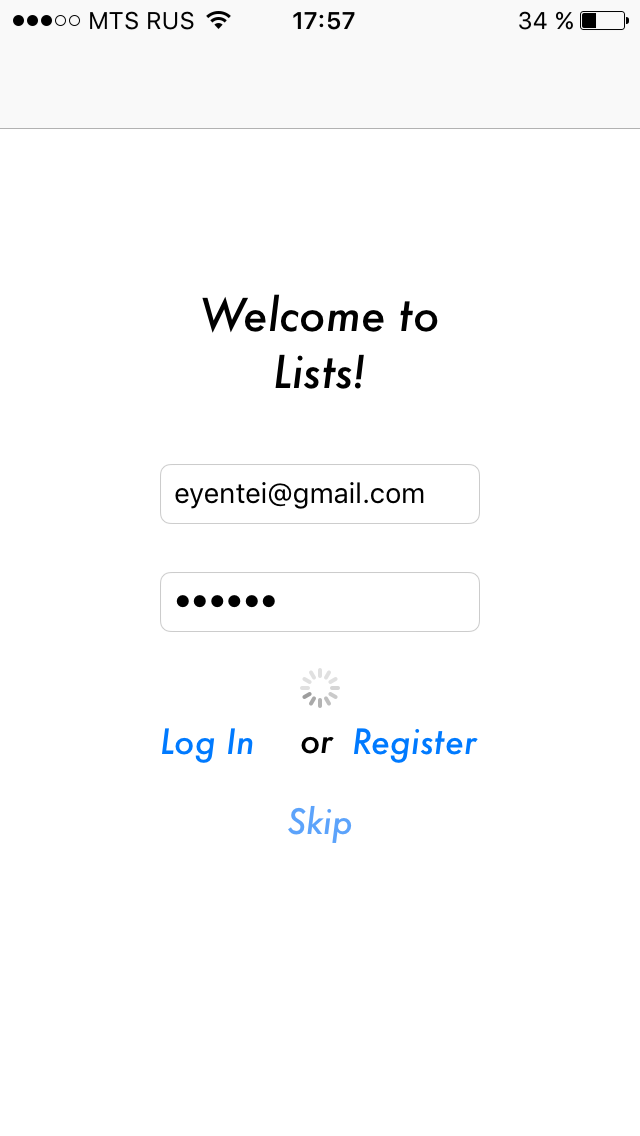


Примеры удаления

1) Списка,

1. Записи.



Изображения 1-4 – примеры ошибок при регистрации или авторизации:

1. Пользователь не заполнил необходимые поля

2. Пользователь ввел почту неправильно

3. Попытка авторизации с неверным паролем или логином

4. Попытка повторной регистрации с тем же адресом электронной почты

5.Ошибок нет – пример анимации загрузки.

1. **Классы:**

**Класс User** – модель пользователя;

Содержит свойства: email (логин, электронная почта), password (пароль), lists (массив списков, созданных пользователем), а также метод, устанавливающий свойство email первичным ключом для БД.

**Класс ToDoList** – модель списка;

Содержит свойства: title (название списка, по умолчанию Untitled), picture (иконка), owner (пользователь, которым был создан этот список), entries (массив записей (заданий) этого списка).

**Класс Entry** – модель записи (задания)

Содержит свойства: text (текст задания).

1. **Список использованных источников:**

<https://realm.io>

<https://stackoverflow.com>