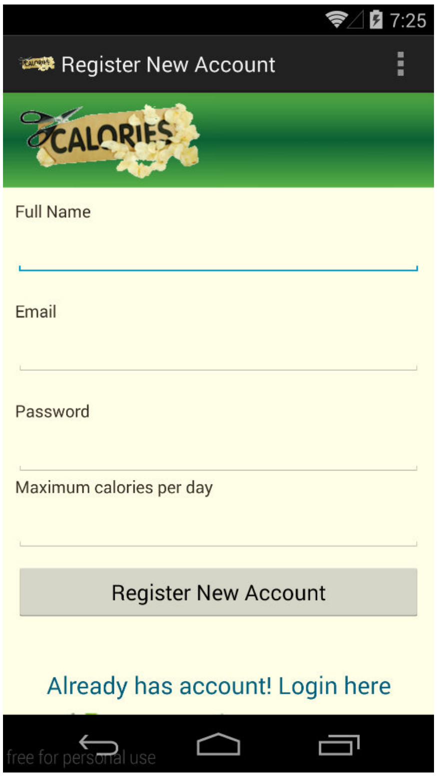
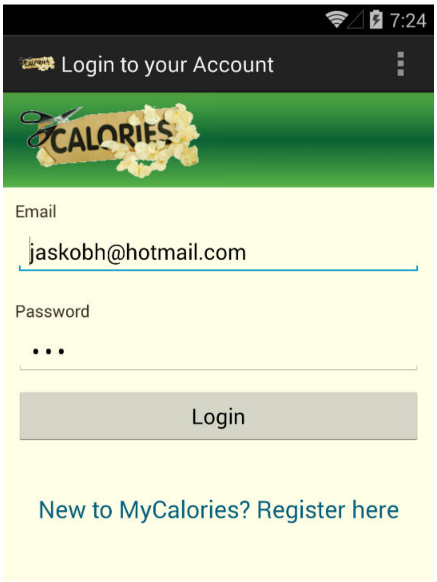
MyCalories is a small client-server type of application built using latest version of Android Studio. It is based on Parse.com backend for storing data and also for providing REST API functions that makes this app scalable and robust.

Usage

1. First there is Register screen where you enter your details and create account.



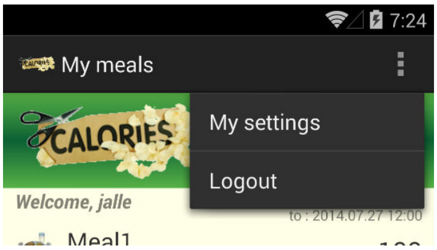
2. Then you have login screen. After 1st successful login credentials are remembered so the user does not have to enter that again.



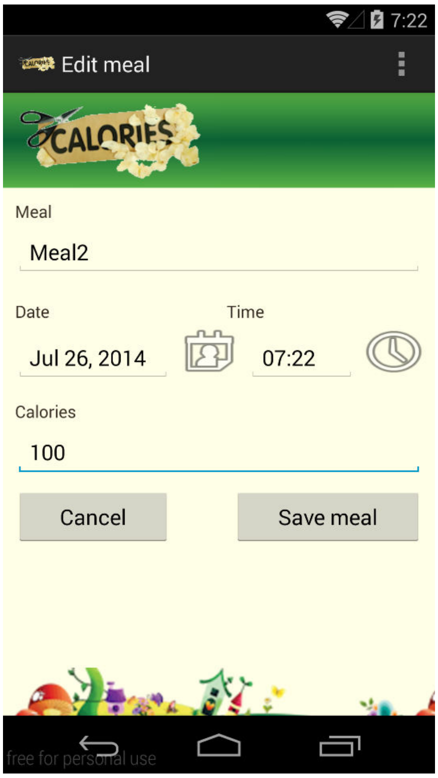
Email
jaskobh@hotmail.com

Password
•••

3. In settings menu there is option to set Max Calories and Logout from the application.



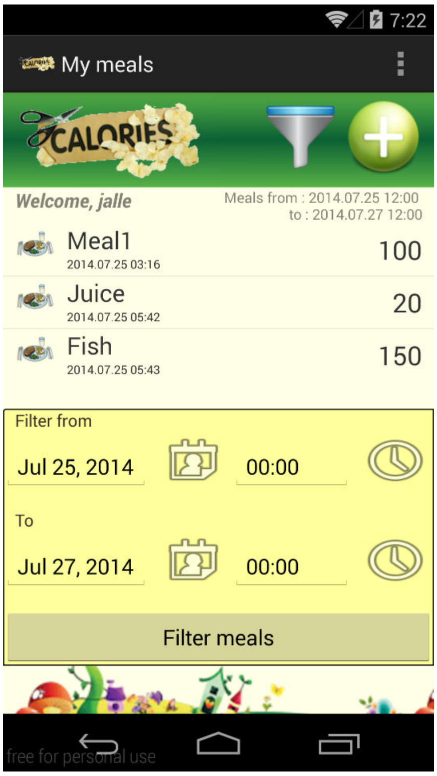
4. The following are options to adding meals, deleting, editing and filtering.



Meal
Meal2

Date Time
Jul 26, 2014 07:22

Calories
100



Welcome, jalle
Meals from : 2014.07.25 12:00
to : 2014.07.27 12:00

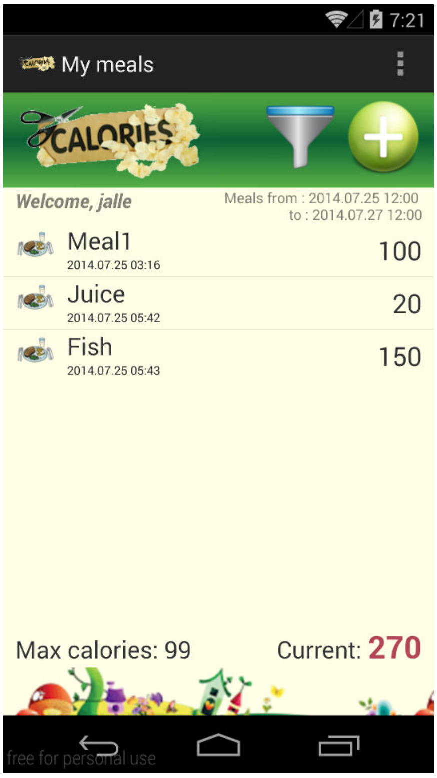
Meal1 100
2014.07.25 03:16

Juice 20
2014.07.25 05:42

Fish 150
2014.07.25 05:43

Filter from
Jul 25, 2014 00:00

To
Jul 27, 2014 00:00



Welcome, jalle
Meals from : 2014.07.25 12:00
to : 2014.07.27 12:00

Meal1 100
2014.07.25 03:16

Juice 20
2014.07.25 05:42

Fish 150
2014.07.25 05:43

Max calories: 99 Current: 270

There are 2 compiled apk files.

1. mycalories.apk - main app that utilize REST API
2. mycalories_wo_api.apk - without REST API

Source code is located here:

http://git.toptal.com/Jasmin-Ibrisimbegovic/jasmin-ibrisimbegovic.git

There are 4 REST API functions located at Parse.com cloud code:

```
//Create meal: parameters Meal object

Parse.Cloud.define("create", function (request, response) {
    var query = new Parse.Query("Meals");

    var Meal = Parse.Object.extend("Meals");
    var meal = new Meal();
    meal.set("meal", request.params.meal);
    meal.set("calories", request.params.calories);
    meal.set("user", request.params.user);
    meal.set("date", request.params.date);

    meal.save(null, {
        success: function (mealsaved) {
            console.log("Create succeed");
            response.success("Meal added");
        },
        error: function (error) {
            response.error(error.code + " " + error.message);
        }
 });
});


// Update meal: parameters Meal object

Parse.Cloud.define("update", function (request, response) {
    var query = new Parse.Query("Meals");
    var meal =request.params.meal

    query.equalTo("objectId", request.params.objectId);

    query.first({
        success: function(object) {
            object.set("meal", request.params.meal);
            object.set("calories", request.params.calories);
            object.set("user", request.params.user);
            object.set("date", request.params.date);
            object.save();
            response.success("Meal updated");
        },
        error: function(error) {
           response.error( error.code + " " + error.message);
        }
    });
});
```

```
//  Select meals, parameters : userId , from , to dates

Parse.Cloud.define("select", function (request, response) {
    var query = new Parse.Query("Meals");
    console.log(request.params.from)

    query.equalTo("user", request.params.user);
    if (request.params.from != null) {
        query.greaterThan("date", request.params.from);
    }
    if (request.params.to != null)
        query.lessThan("date", request.params.to);

    query.find({
        success: function (results) {
           response.success(results);
        },
        error: function () {
          response.error("select failed");
        }
    });
});

// Delete meal, parameters: objectId

Parse.Cloud.define("delete", function (request, response) {
    var query = new Parse.Query("Meals");
    query.equalTo("objectId", request.params.objectid);

    query.find({
        success: function (results) {
            for (var i = 0, len = results.length; i < len; i++) {
                var result = results[i];
                result.destroy();
                console.log("Destroy: " + i);
            }

              response.success("Meal deleted");
        },
        error: function () {
            response.error("deleted failed");
        }
    });
});
```