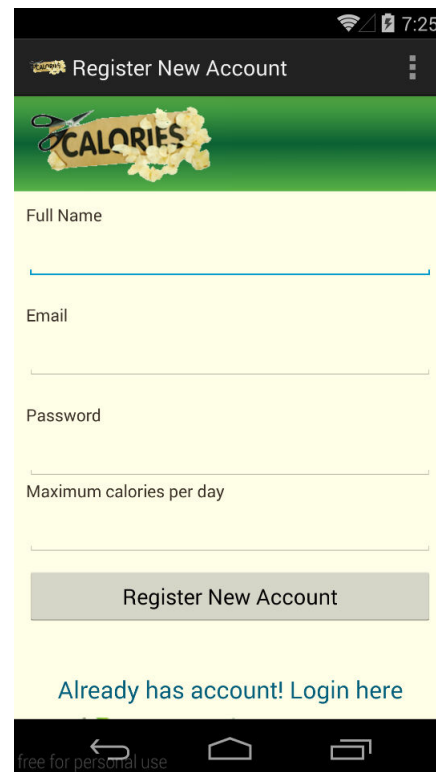


MyCalories is a small client-server type of application built using latest version of Android Studio. It is based on Parse.com backend for storing data and also for providing REST API functions that makes this app scalable and robust.

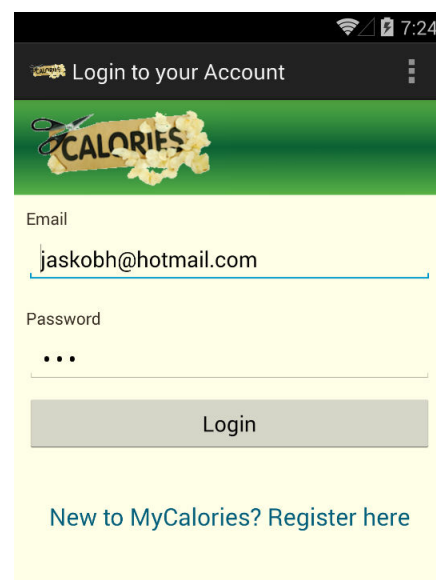
## Usage

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



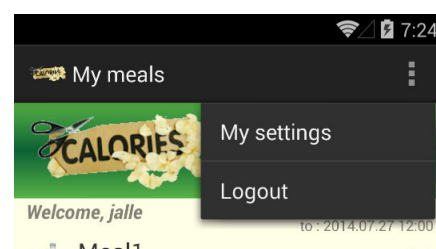
The screenshot shows the 'Register New Account' screen of the MyCalories app. At the top, there's a status bar with signal, battery, and time (7:25). Below the title bar, there's a green header with the 'MyCalories' logo. The main area is a light yellow form with four input fields: 'Full Name', 'Email', 'Password', and 'Maximum calories per day'. Each field has a blue underline. Below the fields is a grey button labeled 'Register New Account'. At the bottom, there's a blue link that says 'Already has account! Login here'. The bottom of the screen shows the Android navigation bar with a back arrow, home icon, and recent apps icon, along with the text 'free for personal use'.

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



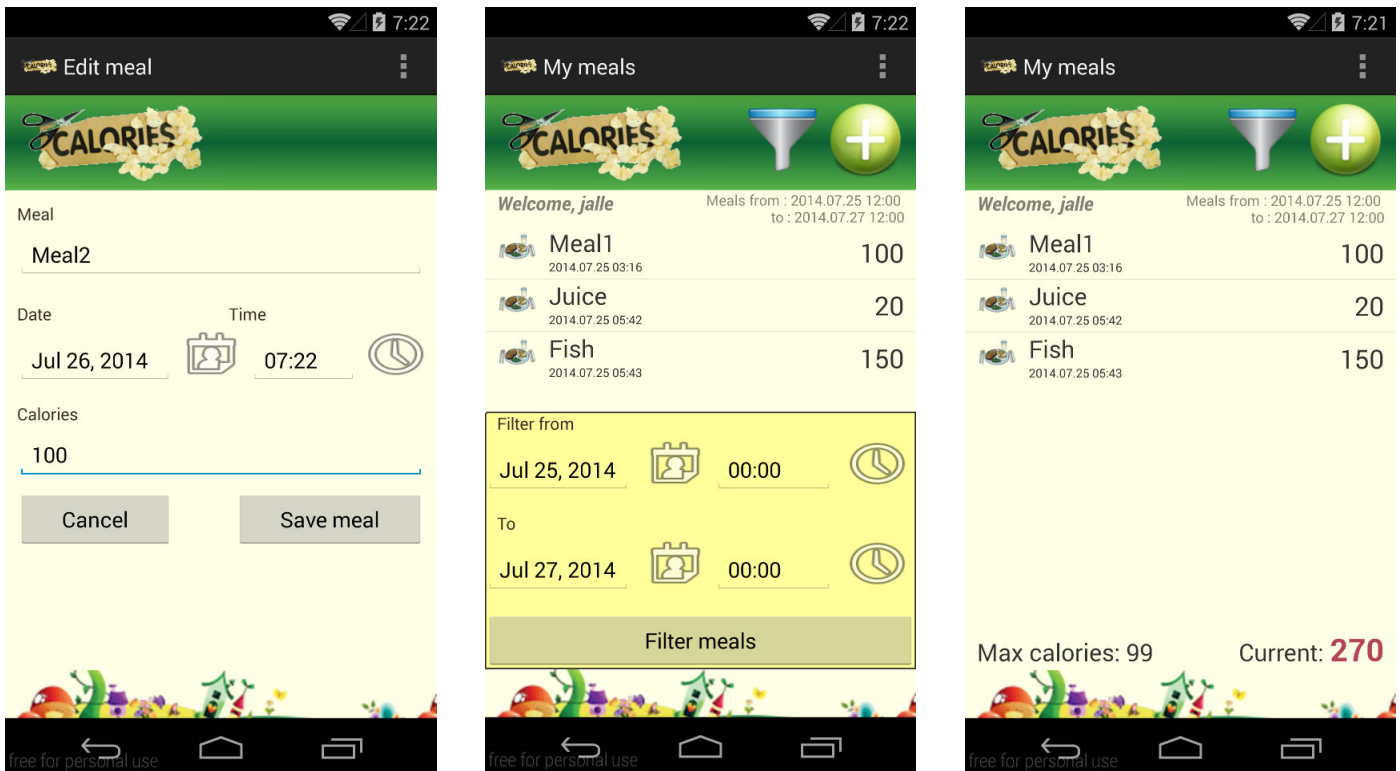
The screenshot shows the 'Login to your Account' screen of the MyCalories app. At the top, there's a status bar with signal, battery, and time (7:24). Below the title bar, there's a green header with the 'MyCalories' logo. The main area is a light yellow form with two input fields: 'Email' and 'Password'. The 'Email' field contains the text 'jaskobh@hotmail.com' and has a blue underline. The 'Password' field has three dots indicating it's hidden. Below the fields is a grey button labeled 'Login'. At the bottom, there's a blue link that says 'New to MyCalories? Register here'.

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



The screenshot shows the 'My meals' screen of the MyCalories app. At the top, there's a status bar with signal, battery, and time (7:24). Below the title bar, there's a green header with the 'MyCalories' logo. The main area is a light yellow form with a 'Welcome, jalle' message and a 'Meal1' entry. A dark grey menu is open on the right side, showing two options: 'My settings' and 'Logout'. The bottom of the screen shows the Android navigation bar with a back arrow, home icon, and recent apps icon, along with the text 'free for personal use'.

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



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");  
    }  
  });  
});
```