



# Application Development for Mobile and Ubiquitous Computing

## Seminar Task Second Presentation

GroupNo. 17

Team:

Daniel Olivares

Ian Nunes Rodrigues

- CrowdJammin is a multi-platform app, for music enthusiasts (both amateur and professional) both with or without a band who are looking for inspiration or other people to arrange or jam to their ideas, or jam to other people ideas.

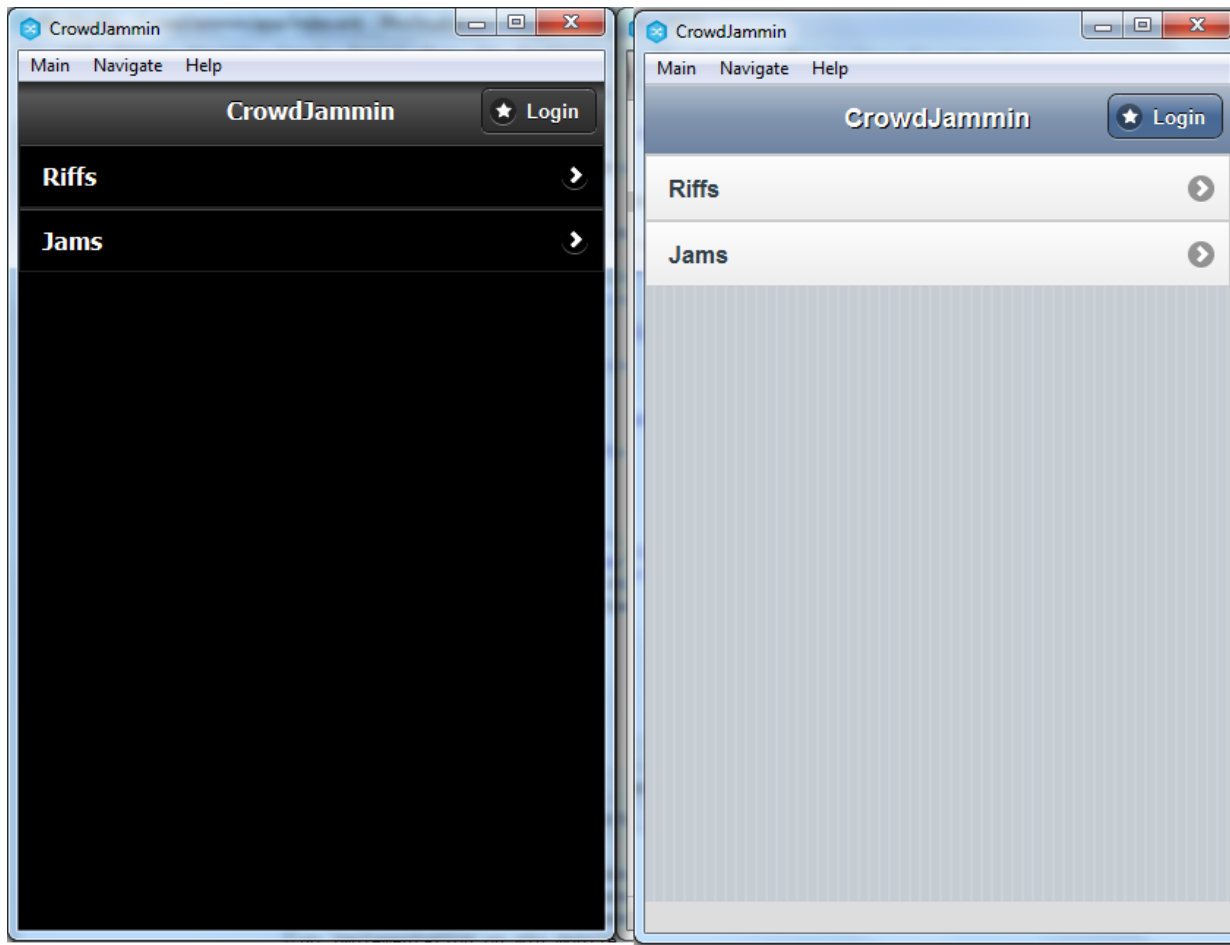
- Screenshots captured by using Rhosimulator, but it can already run in iOS, Android and WP devices or simulators.
- View/design is set to default (native) for now. We can both adjust it to each O.S. or have a general one.

- Splash screen

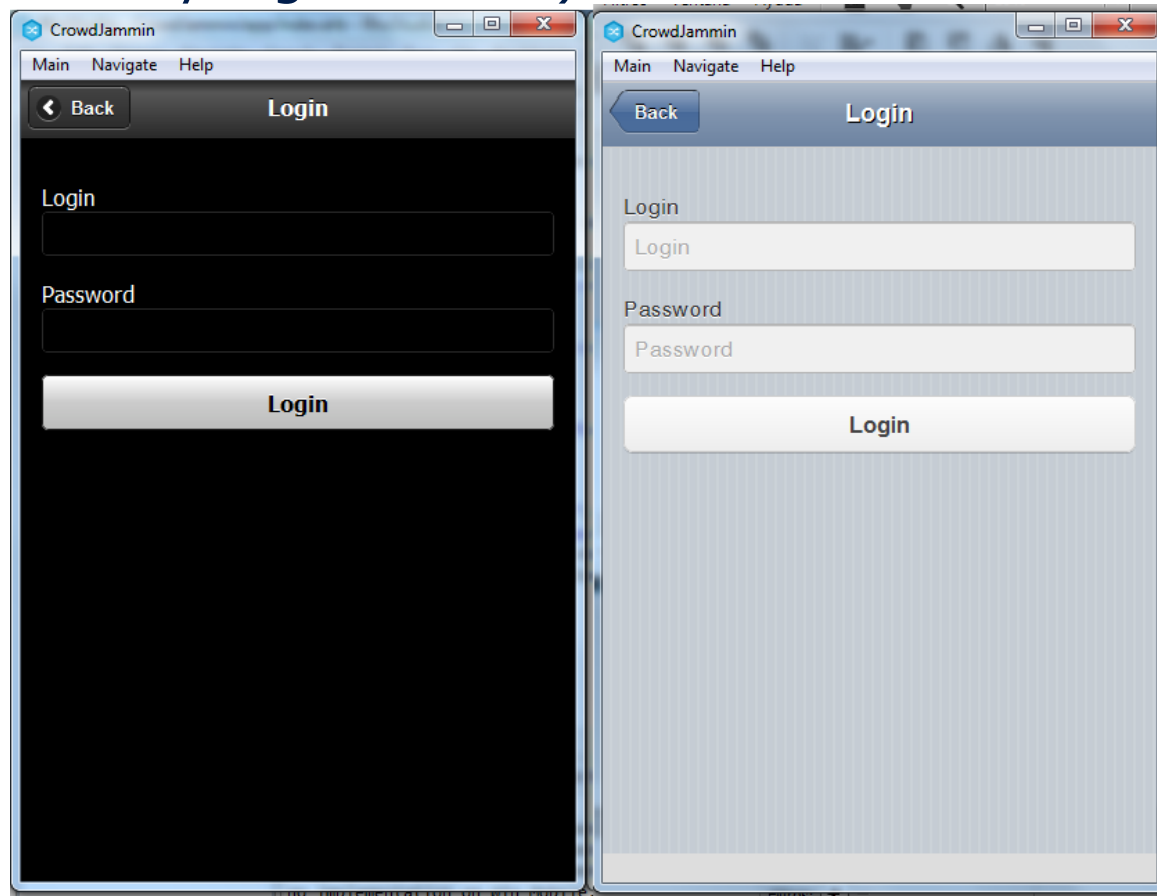


Crowd  
Jammin

- Landing screen / Menu

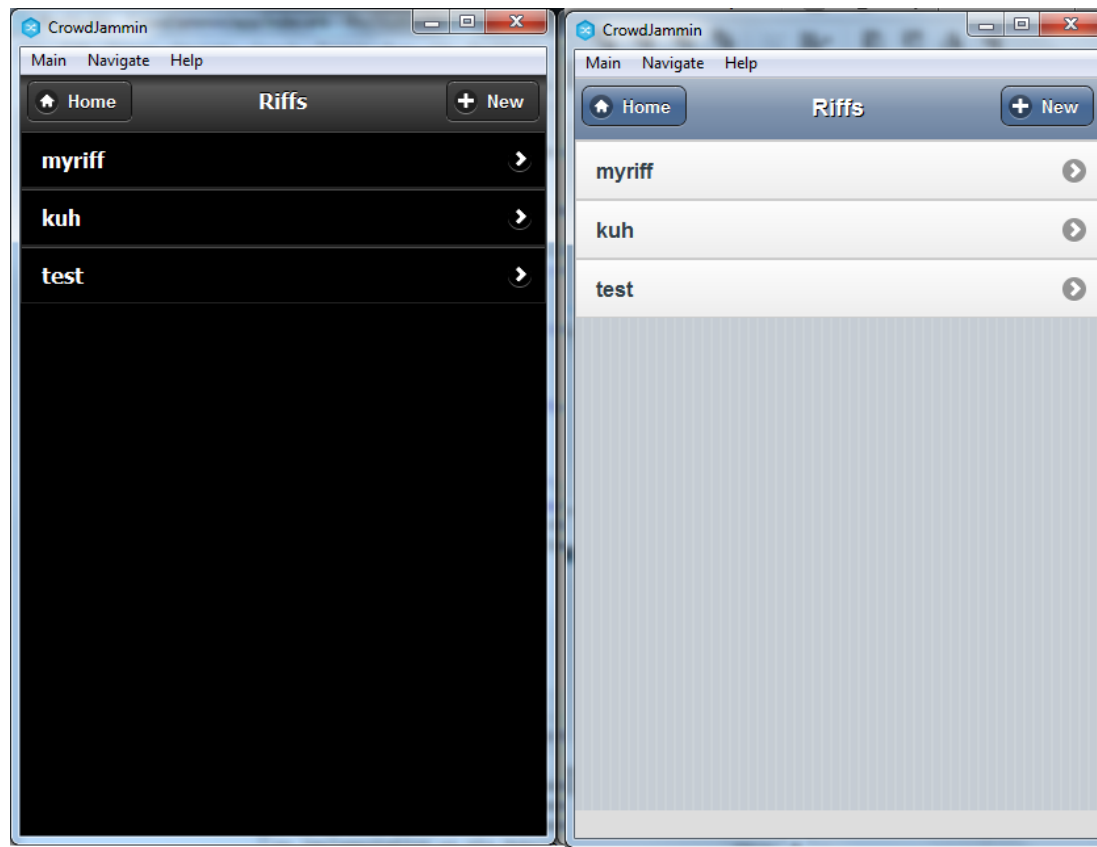


- Login screen. (Facebook and/or Twitter APIs will be used for easy login as well).



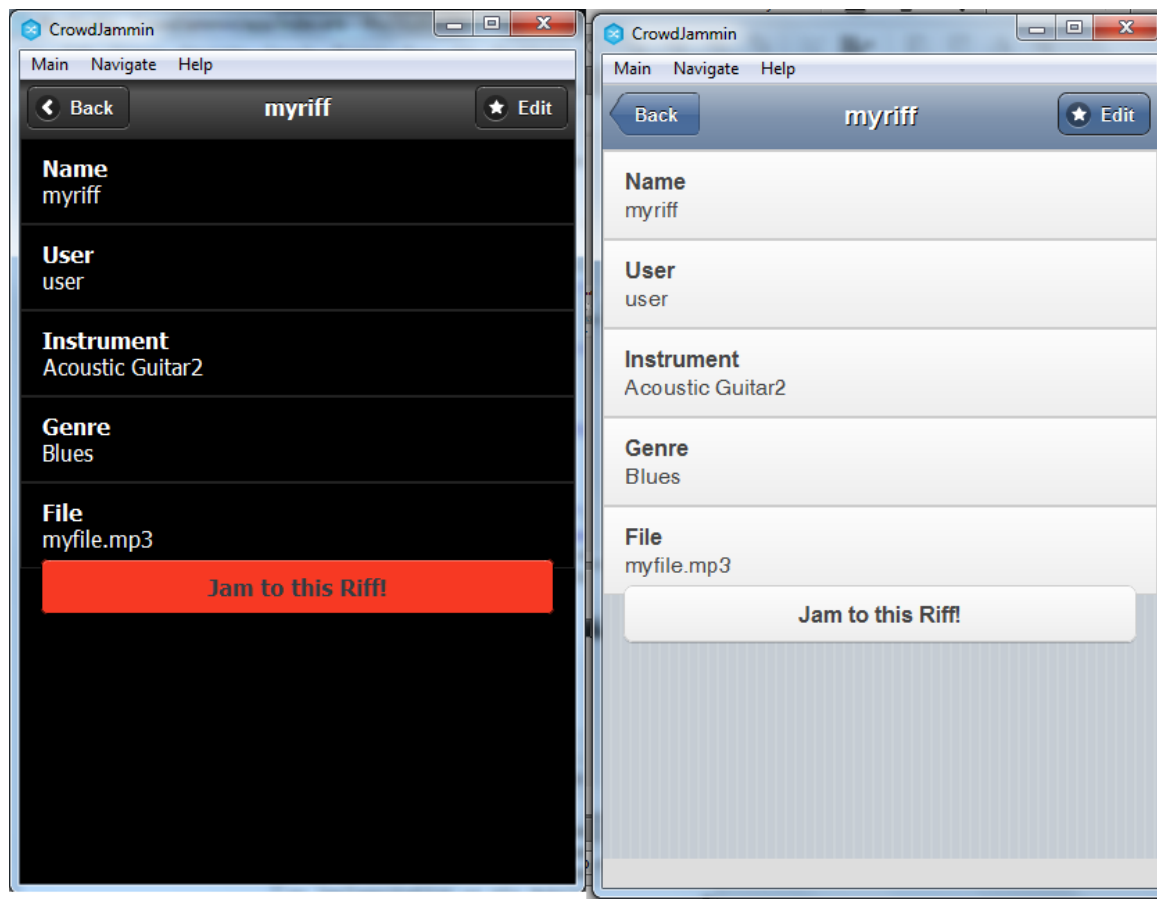
- Riffs are pieces of music uploaded by a single user.
- Jams are various riffs together (when a user joins another).

- Riffs Browsing Screen. Here the user will be able to browse and filter riffs according to their properties (genre, instrument, location, etc)

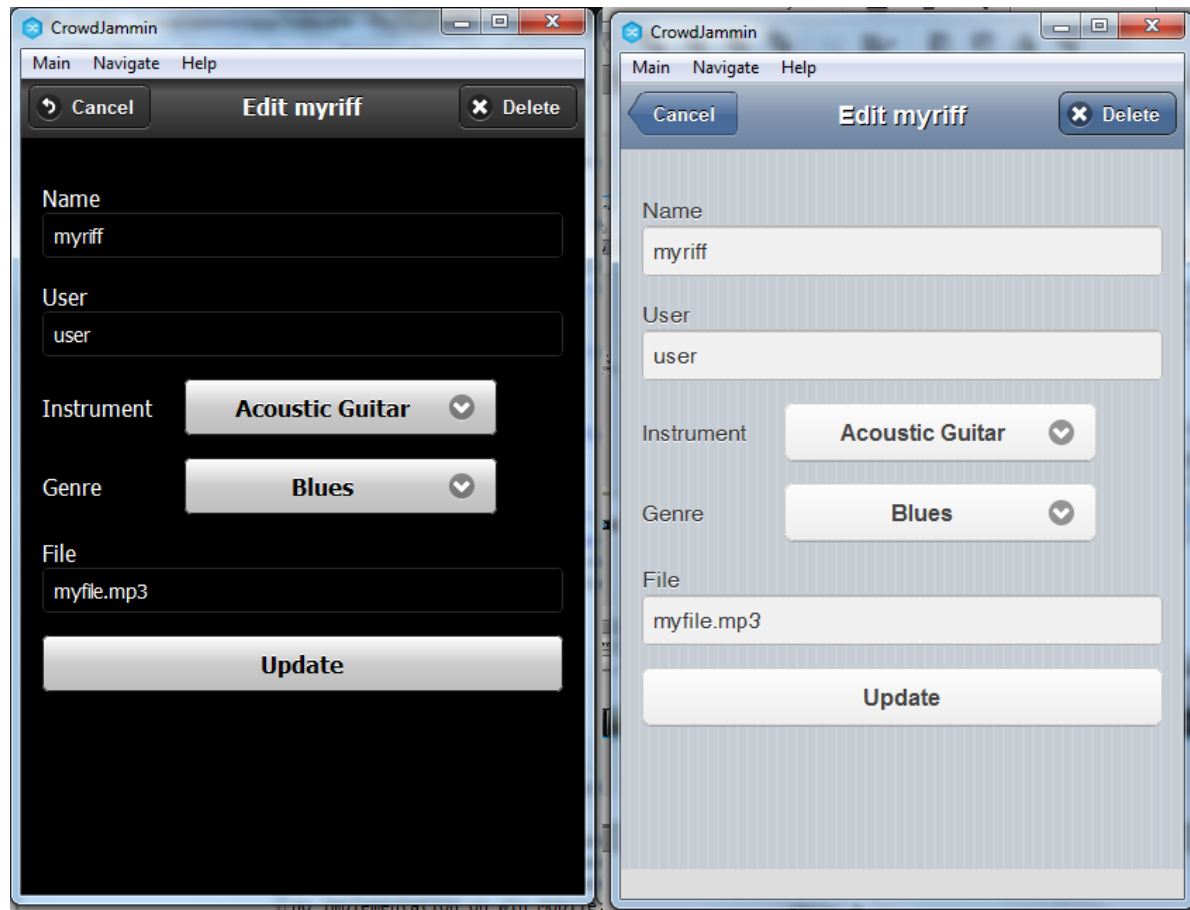




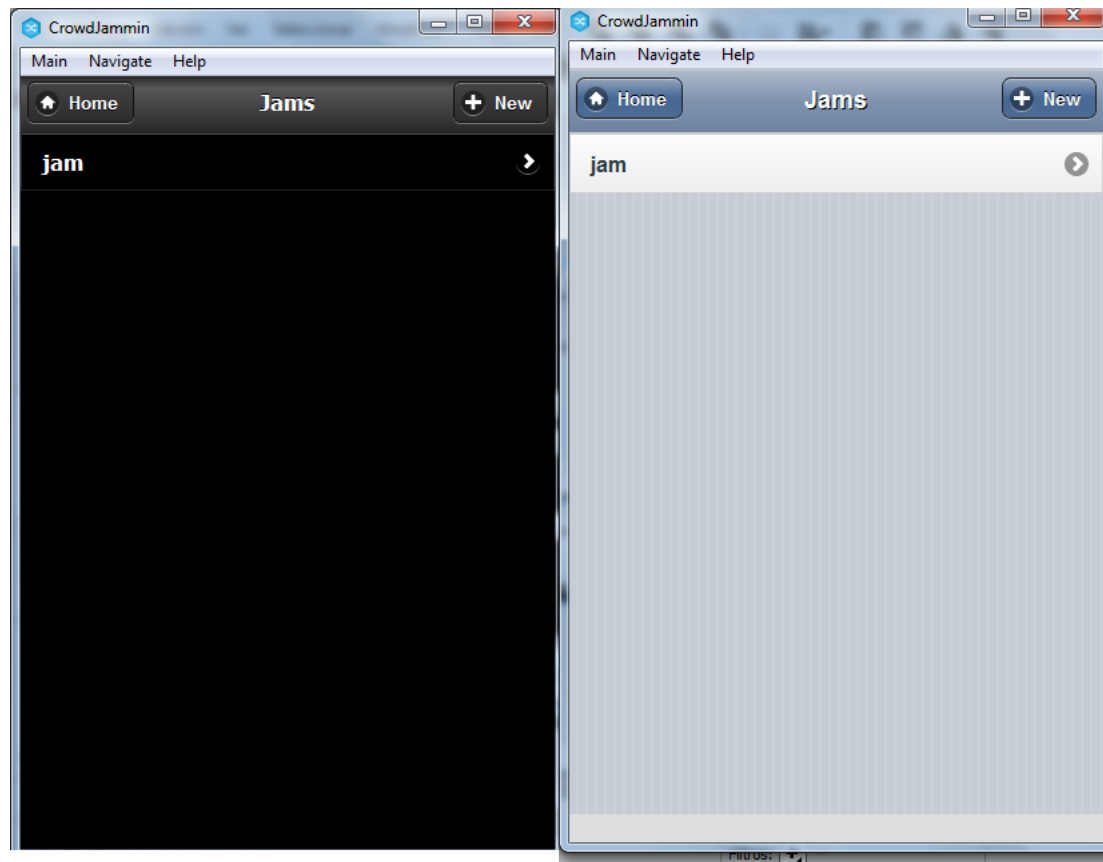
- Riff display. Here the user will be able to listen to the riff and if he likes it, he can join in.



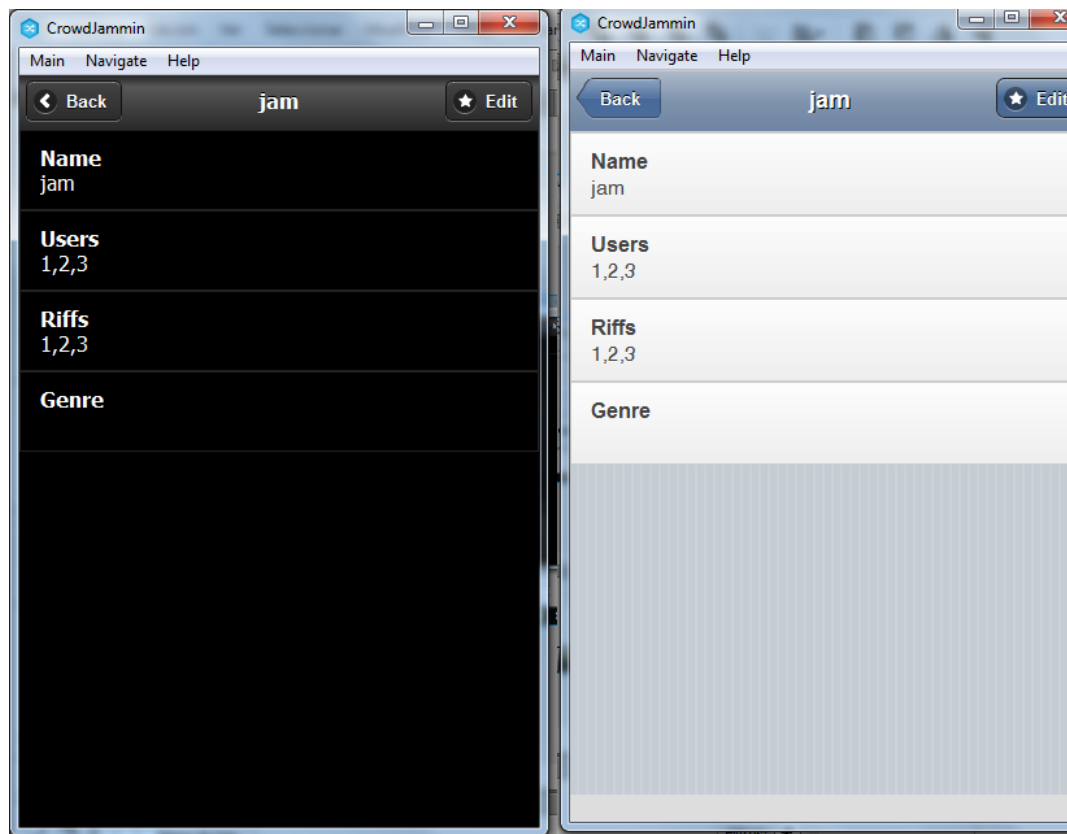
- Edit (or delete) own riff.



- Jams browsing. Here the user will be able to browse and filter jams according to their properties.

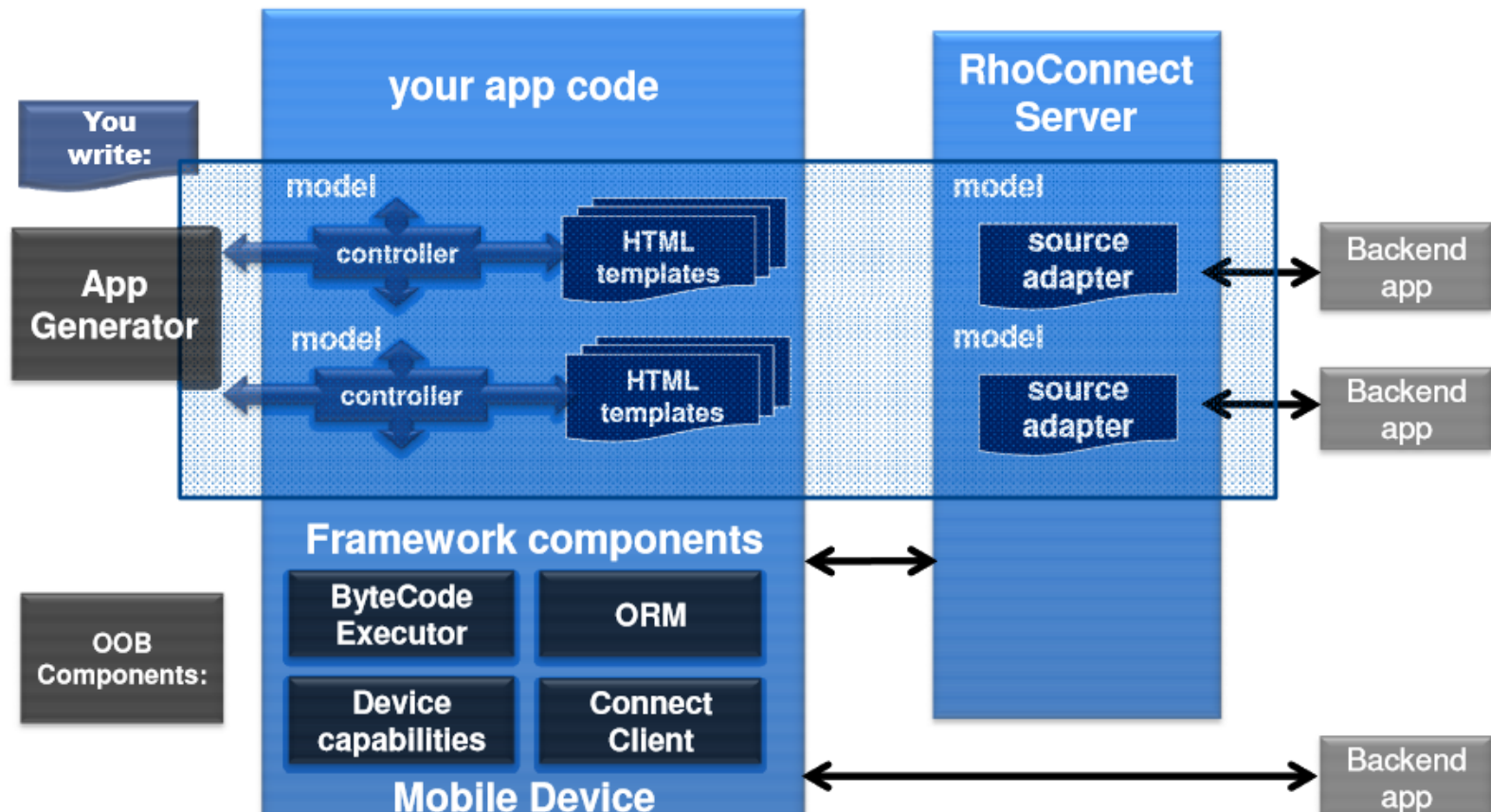


- Jam display

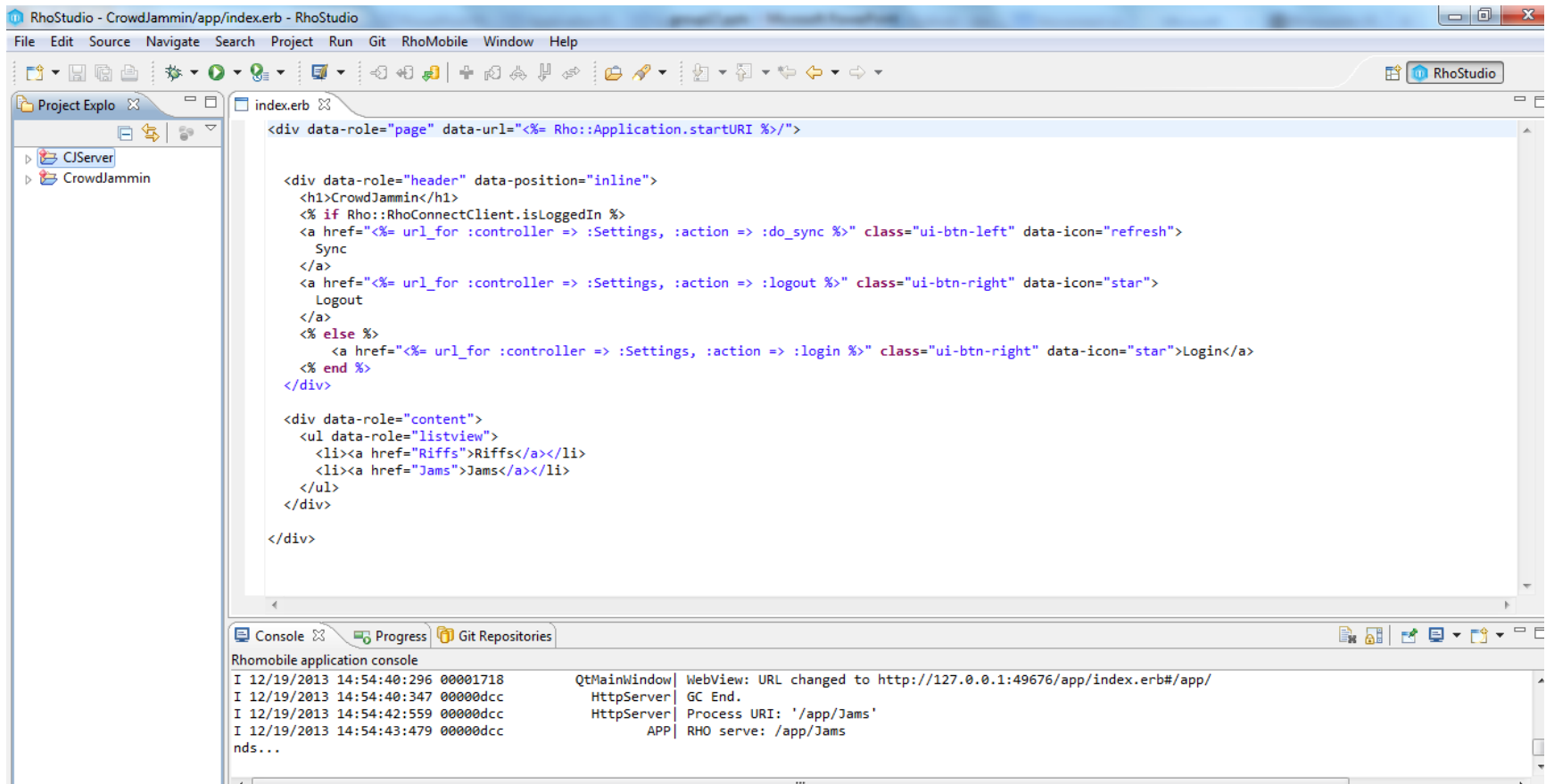




## RhoConnect in the RhoMobile Suite



- Client part of the application: Rails (HTML). Includes CRUD displays and a controller for each model.



The screenshot shows the RhoStudio IDE interface. The main editor displays the `index.erb` file with the following code:

```
<div data-role="page" data-url="<%= Rho::Application.startURI %>"/>

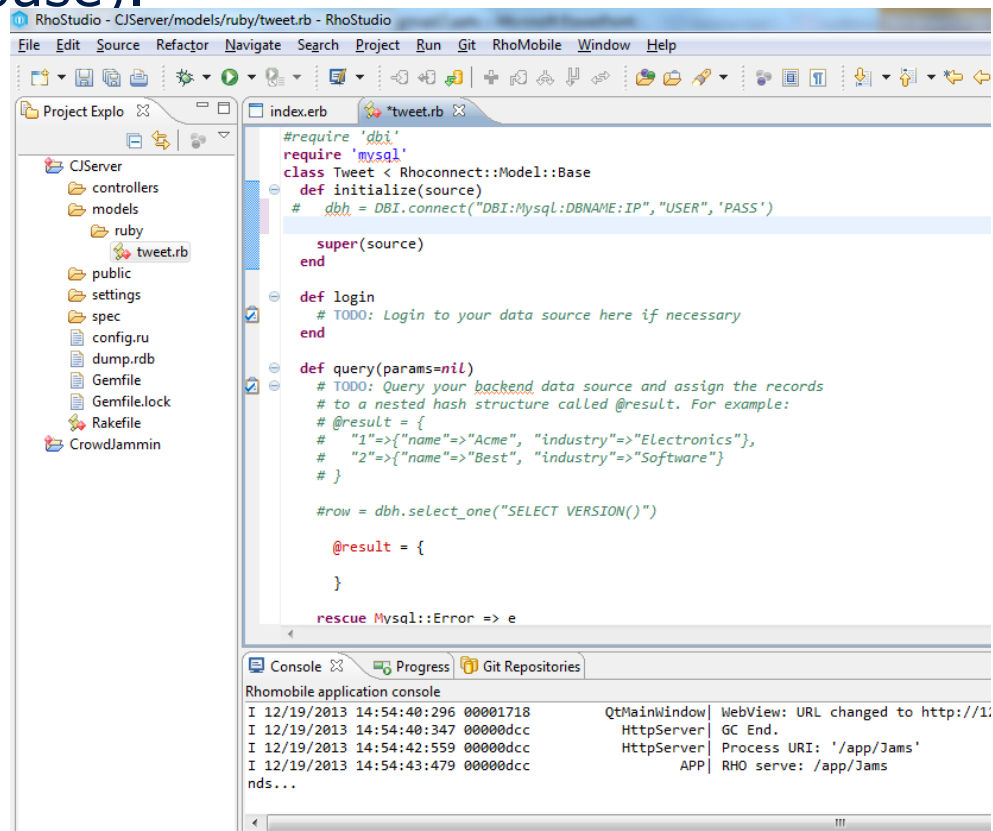
  <div data-role="header" data-position="inline">
    <h1>CrowdJammin</h1>
    <% if Rho::RhoConnectClient.isLoggedIn %>
      <a href="<%= url_for :controller => :Settings, :action => :do_sync %>" class="ui-btn-left" data-icon="refresh">
        Sync
      </a>
      <a href="<%= url_for :controller => :Settings, :action => :logout %>" class="ui-btn-right" data-icon="star">
        Logout
      </a>
    <% else %>
      <a href="<%= url_for :controller => :Settings, :action => :login %>" class="ui-btn-right" data-icon="star">Login</a>
    <% end %>
  </div>

  <div data-role="content">
    <ul data-role="listview">
      <li><a href="Riffs">Riffs</a></li>
      <li><a href="Jams">Jams</a></li>
    </ul>
  </div>
</div>
```

The console output at the bottom shows the following logs:

```
Rhmobile application console
I 12/19/2013 14:54:40:296 00001718 QtMainWindow WebView: URL changed to http://127.0.0.1:49676/app/index.erb#/app/
I 12/19/2013 14:54:40:347 00000dcc HttpServer GC End.
I 12/19/2013 14:54:42:559 00000dcc HttpServer Process URI: '/app/Jams'
I 12/19/2013 14:54:43:479 00000dcc APP RHO serve: /app/Jams
nds...
```

- Server side: Ruby CRUD source adapters. Here we can use a DBI adaptor to connect to a database or parse remote data (in this case we will use a MySQL database).



```
RhoStudio - C:\Server\models\ruby\tweet.rb - RhoStudio
File Edit Source Refactor Navigate Search Project Run Git RhoMobile Window Help

Project Explorer
C:\Server
├── controllers
├── models
│   └── ruby
│       └── tweet.rb
├── public
├── settings
├── spec
├── config.ru
├── dump.rdb
├── Gemfile
├── Gemfile.lock
├── Rakefile
└── CrowdJammin

index.erb
tweet.rb

#require 'dbi'
require 'mysql'
class Tweet < Rhoconnect::Model::Base
  def initialize(source)
    # dbh = DBI.connect("DBI:MySQL:DBNAME=IP", "USER", 'PASS')

    super(source)
  end

  def login
    # TODO: Login to your data source here if necessary
  end

  def query(params=nil)
    # TODO: Query your backend data source and assign the records
    # to a nested hash structure called @result. For example:
    # @result = {
    #   "1"=>{"name"=>"Acme", "industry"=>"Electronics"},
    #   "2"=>{"name"=>"Best", "industry"=>"Software"}
    # }

    #row = dbh.select_one("SELECT VERSION()")

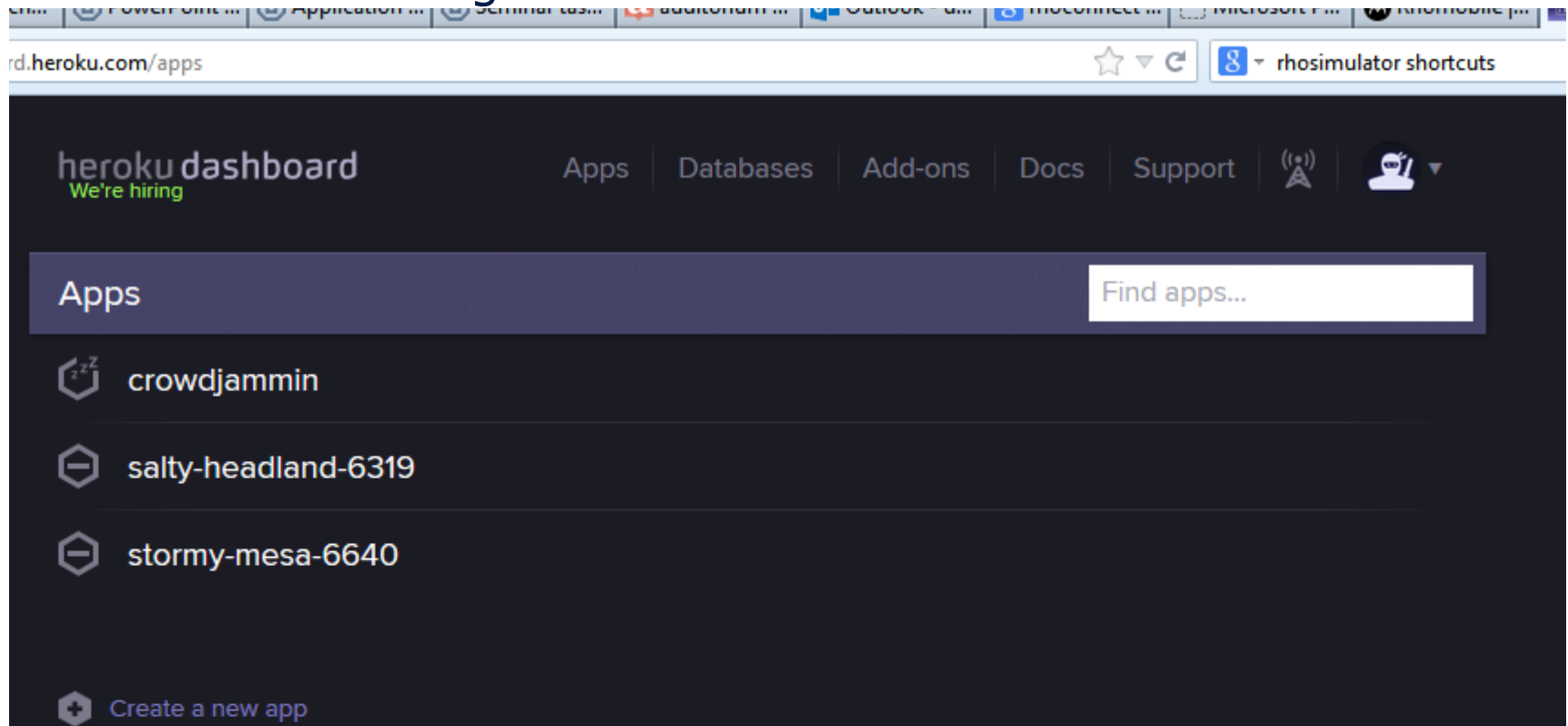
    @result = {

    }

    rescue MySQL::Error => e
  end
end

Rhomobile application console
I 12/19/2013 14:54:40:296 00001718 QtMainWindow WebView: URL changed to http://1/
I 12/19/2013 14:54:40:347 00000dcc HttpServer GC End.
I 12/19/2013 14:54:42:559 00000dcc HttpServer Process URI: '/app/Jams'
I 12/19/2013 14:54:43:479 00000dcc APP RHO serve: /app/Jams
nds...
```

- The backend-app is already uploaded to Heroku, which is a cloud platform as a service (PaaS) supporting several programming languages. It is one of the best known services for app distribution and version management.





- Low resource cost application developed for multiple platforms using general specifications;
- Connectivity with server resolved through on-need-only ruby requests and offline scheduling capabilities of the application;
- Simple and deductive user interface.

- Audio files are recorded and encoded application side;
- Application sends audio file (after validation) to be stored server side;
- Application requests audio data and information from server which is then sent and replicated on the device;
- Requests are done with xmlhttp through ruby to the database;
- Data regarding the audio files (riff or jam) is intrinsically tied to the user data.

- Audio playability on the devices;
- Location awareness regarding the user and/or the riffs and jams;
- Jam data and different settings associated with it;
- Implementation of User-to-User communication.