

An Open Source Solution for Education Management

EduXes

José Antonio Salgueiro Aquino <info@joseantonio.org>

*Student of the V Master on Free Software Projects Development and
Management 2011-2012*

Copyright (cc) 2012 José Antonio Salgueiro Aquino. Some rights reserved. This work is free and is licensed under the conditions of Creative Commons Attribution – Share Alike 3.0 Unsupported license. You can use, distribute and reuse this work if the same license is applied and the author is quoted

Full text of the license can be read on <http://creativecommons.org/licenses/by-sa/3.0/deed.en>



Contents

Description of the practicum	4
Work plan.....	6
Description and objectives:.....	6
Methodology:.....	9
Work plan:.....	9
Results.....	11
Tasks completed.....	12
State-of-art solutions.....	12
Develop database structure: tables and relationships.	16
Preparation of development:.....	16
Application Work-flow	19
Development:.....	20
Tasks to be done.....	22
Test into real hardware.....	27
Save or download data from database to disk.....	27
Xade web interface.....	27
Another developments.....	27
Test units.	27
User documentation.....	27
Developer's documentation.....	27
Community.....	28
Personal evaluation of the practicum	28

Description of the practicum

Name: José Antonio Salgueiro Aquino

Birth date: 5 august 1970

Education: B.Sc. in Fundamental Physics, University of Santiago de Compostela University 1988-1993.

Current address: Marin (Pontevedra)

Current job: Secondary School teacher in Technology.

I have done on my spare time at home a management application for high school teachers.

Working times (planned): 300 hours.

From 6th August, to 30 September, on an eight hours day basis.

This application involves several technologies:

- Java language.
- Android skeleton application.
- PhoneGap framework to develop multi-platform applications.
- JQuery and JQueryMobile to develop mobile oriented applications.
- JavaScript with Web Databases.
- Git for version control system.

Meetings:

One meeting on August: Technologies to be used were stated, work methodologies, first application windows (pages), Android version to be used (2.3.3) because is the most popular.

Several emails and *gtalk* conversations about organization, general problems were written.

Teleworking is done.

Materials and special equipment used:

My own computer for main development:

Hardware: Intel Quad, 6GiB RAM, 500GiB HD.

Software: Debian Linux Wheezy (testing), Eclipse Juno, JQuery 1.8.1,

jQueryMobile 1.1.1, and PhoneGap-Cordova 1.8.1, Android Virtual Manager 2.3.3, Git 1.7.10.4-1.

For testing Sony-Ericsson Xperia V mobile phone, with USB cable.

Work plan

Description and objectives:

A multi-platform management application for high school teachers is developed. It can be run on a smart-phone or tablet, or even a personal computer.

The actual objectives of the applications are student's management:

- Attendance control.
- Misbehaviour control.
- Activities evaluation.

Application should include this features:

- Data visualization. As table-like.
- Server synchronization with a custom application or Xade¹.

The final goal is to develop an application to make teacher's work easier and comfortable. Also an objective is to write extensible code, which allow another developers to take part into application development.

Tasks:

The current list of tasks are:

1. Study state-of-art solutions.
 - a) Find out other solutions: PDAs and smart-phone or tablet related and web-based applications.
 - b) Download to study and reuse graphical user interfaces, code or/and database structure.
2. Develop database structure: tables and relationships.
3. Preparation of development:
 - a) Build development environment: install Eclipse, Android Virtual Machine, Aptana Plugin, JQuery, JQueryMobile and Phonegap.
 - b) Choose application name and folder's policy.

¹ Xade is the web application used to management by educational community: teacher's, headmasters, administrative staff.

- c) Make a simple application: only a blank page.
 - d) Upload simple application into a git repository²
4. Development:
- a) Populate database with sample data.
 - b) Groups:
 - Make list of groups window.
 - Groups management window.
 - c) Students:
 - Make list of students window.
 - Students management window (insert-update-delete students)
 - d) Timetable for actual date: list of groups for each day.
 - e) Add attendance, misbehaviour for each student.
 - f) Add error handling.
 - g) Retrieve and insert data from and to database
 - h) List of attendance, misbehaviour incidents.
 - i) Add activities grades for each student.
 - j) List students marks and final mark.
 - k) Activities management window (add-update-remove activities)
 - l) Management of student notes.
 - m) List of student notes.
5. Test into real hardware: Android 2.3.3 mobile phone.
6. Save or download data from database to disk.
7. Xade web interface.
- a) Study Xade web interface.
 - b) Develop an ad-hoc application for retrieve Xade's data.
8. Develop an ad-hoc application for store data.
9. Synchronization with a custom server or with Xade.
10. Test units.
11. User documentation. Manual with images.

2 <https://github.com/joseantoniosa/EduXes/>

12. Developer's documentation.
13. Find out a website to host a forum, a bug report system, documentation and application download.

In the following table a broad estimation of time spent in each task are shown.

Tasks	Time (hours)
State-of-art solutions	10
Develop database	8
Preparation for development	40
Development.	
Populate database with sample data.	20
Groups. List and management	50
Students. List and management	30
Timetable for actual date	80
Add attendance, behaviour	50
Add error handling	2
Retrieve and insert data from and to database	30
List of attendance, misbehaviour incidents.	12
Add activities grades for each student.	12
List students marks and final mark.	14
Activities management window	14
Management of student notes.	20
List of student notes.	2
Test into real hardware	20
Save data into disk	10*
Total	394

Motivation:

I am a Technologies teacher, in my daily work I have to evaluate students work such as working with tools, cooperative work, cooperative work with other classmates etc., besides usual activities as written exercises. It

* Estimated

could use a long sheet, or an awkward long spreadsheet, but a portable device with a custom application should be desirable.

This application increases teacher's productivity because teacher only has to write attendance, or unpunctuality two times (on official report and on application's window), and classroom notes and activity grades on very easy way.

The most important feature is to be as easy, fast and intuitive as possible. It could be desirable to be platform independent (Android, iOS, Windows RT), but Android is preferred because it is open source, has a high market share, and to buy an Apple Macintosh computer is not mandatory.

I wish to learn from this application how to develop a mobile application, and several technologies: JQuery, jQueryMobile, PhoneGap/Cordova, SQLITE, git repository management.

Methodology:

This work was carried on building little blocks, also called pages, and make up it into final application. Database structure was separated from interface, and interface was also separated into dynamic and static. Each new functionality was written, tested, and polished. Each new function was written from previous one, and so on.

Tools involved were Eclipse IDE (with plug-ins) and Android Virtual Manager (AVM) on Debian GNU/Linux Wheezy. When a new functionality was developed, application was tested on AVM, if it worked, source code was polished, application was tested again, if it was satisfactory a new change was committed into git repository.

Work plan:

Several problems were faced:

- Eclipse environment: A stable, reliable and up-to-date IDE, with several

plug-ins is needed. Download vanilla Eclipse Juno from its web-site is chosen because it is more stable, reliable, compatible with newer versions. Aptana Javascript plugin was chosen because Aptana allows source code auto-completion in JQuery.

- PhoneGap and Android incompatibilities. Android 2.3.3 requires JQuery-1.8.1 and does not work on higher versions.
- Error handlers. I have had several problems with *tx.executeSql(...)* function, it confused me with *db.transaction(...)*:

```
tx.executeSql(sql, [parameters], successHandler, errorHandler)
```

and

```
db.transaction(queryFunction, errorHandler, successHandler)
```

have up to four and three parameters respectively, only first one is mandatory. I rather use success and error handlers for *tx.executeSql* function, atomic error control could be better choice.

- Passing variables to functions: Only if another solution is not known or feasible, global variables are used: named after *global_**, and in block capitals.
- Deadline. Development was delayed because I have no enough spare time and above problems were time costly.

Results

Application is evolving from list, edit students and groups, to its final goals.

These objectives were fulfilled:

- Access to any workday, any group and student.
- Management of attendance and misbehaviour of each student. The students information is still hard-coded into source files.

There are obvious and easy to reach objectives:

- Links to student and group management. These pages were done but links are not missing into main application window.

There are several objectives not fulfilled yet, but I am on the way to get those done, those are, in priority order:

- Data visualization. Student attendance and misbehaviour have to be shown in table-like window.
- Test into real hardware. *EduXes.apk* has to be copied into mobile phone.
- Activities evaluation per student. A window to display activities marks and final mark.
- Timetable management. A window to manage groups timetable. When a group has class with this teacher.
- Server synchronization with a custom application or

These objectives were not fulfilled because time and skills lack.

Tasks completed

State-of-art solutions

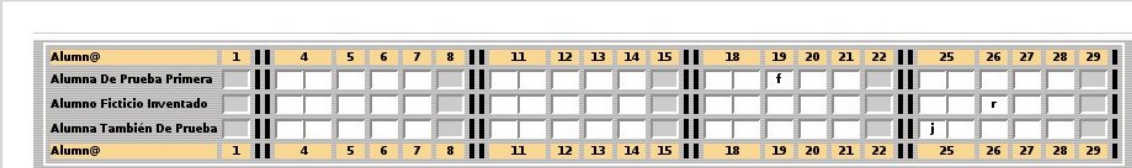
Only an open source application was found for study, Siestta³, nevertheless there are a lot of educational software (Sixa⁴, Unisoft⁵) but they are privative, Microsoft Windows freeware or both (SAS académico⁶).

Siestta was evaluated.

Technically it is an GPL'ed old style PHP-based web application with Ajax, an interactive editor, fckeditor and fpdf to generate reports.

From user point-of-view there are online documentation⁷. This application includes management of students, attendance, marks, tasks, incidents, general queries, letters to parents, interviews with parents, messages, appointments, exams and more.

Several screen-shots were taken and will be reused in current application:



Alumn@	1	4	5	6	7	8	11	12	13	14	15	18	19	20	21	22	25	26	27	28	29
Alumna De Prueba Primera													f								
Alumno Ficticio Inventado																		r			
Alumna También De Prueba																	j				
Alumn@	1	4	5	6	7	8	11	12	13	14	15	18	19	20	21	22	25	26	27	28	29

Illustration 1: Attendance list. (From Siestta webpage)

³ <http://siestta.sourceforge.net/>

⁴ <http://www.sixa.es/es>

⁵ <http://www.unisoftcolombia.com/unisoftcolombia/index.php>

⁶ <http://www.rafaelvarela.com/sas-academico-software-notas-boletines-matriculas.html>

⁷ <http://siestta.sourceforge.net/doc/index.html>

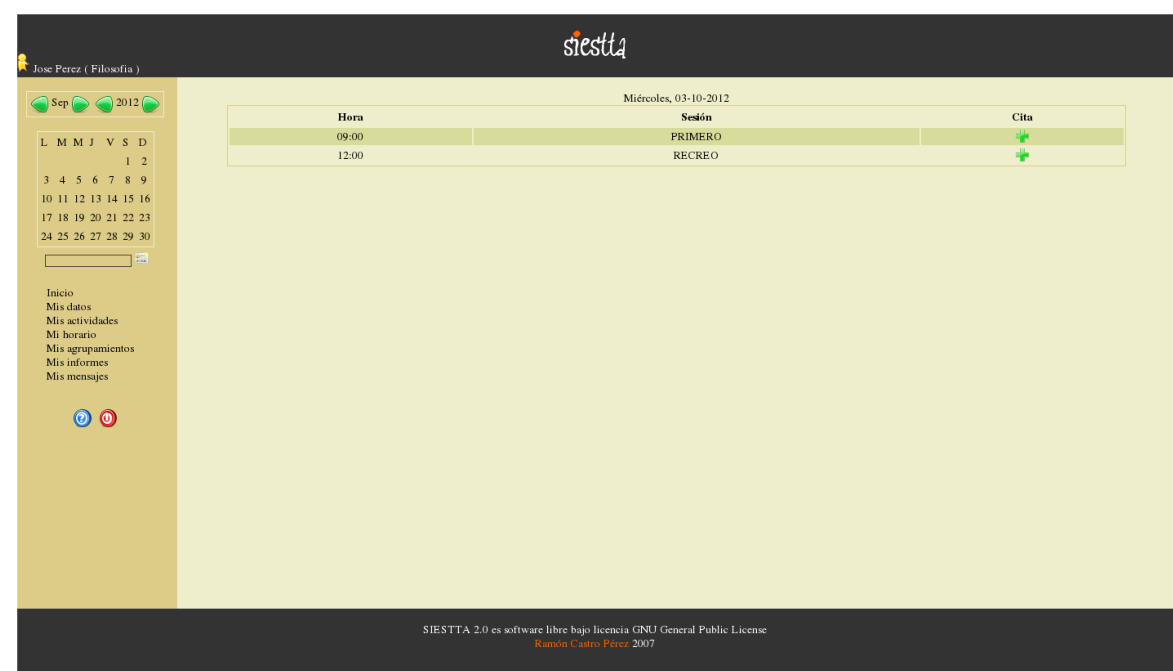


Illustration 2: Main window (From local installation)

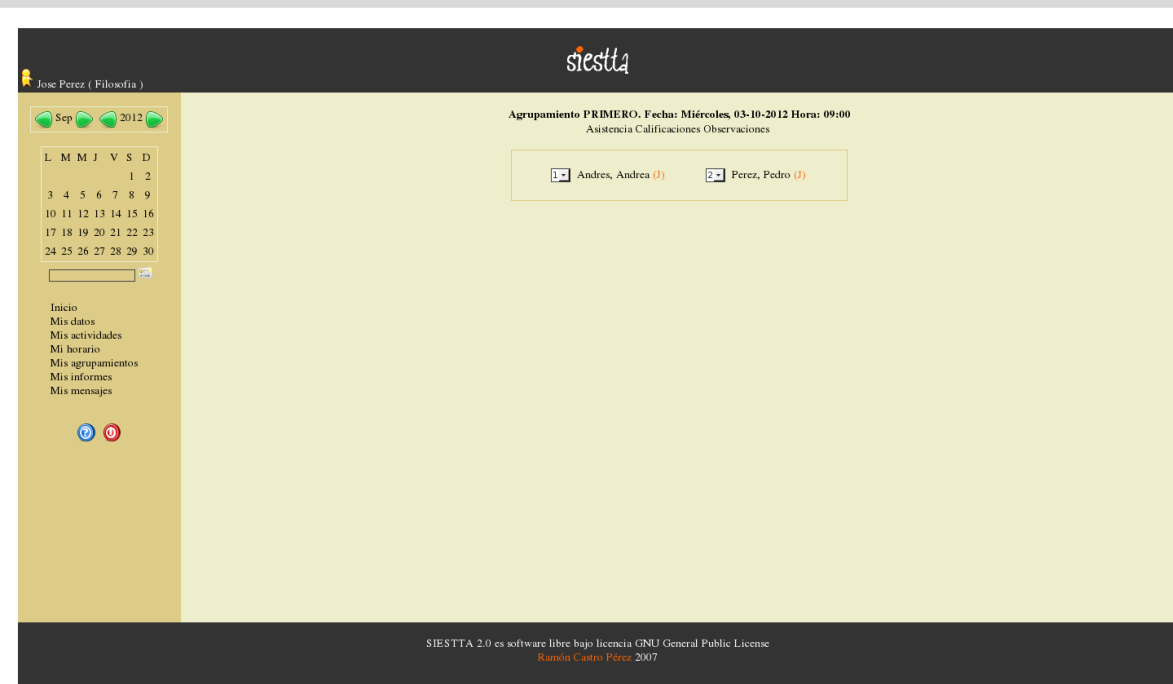


Illustration 3: Attendance (From Siesta local installation)

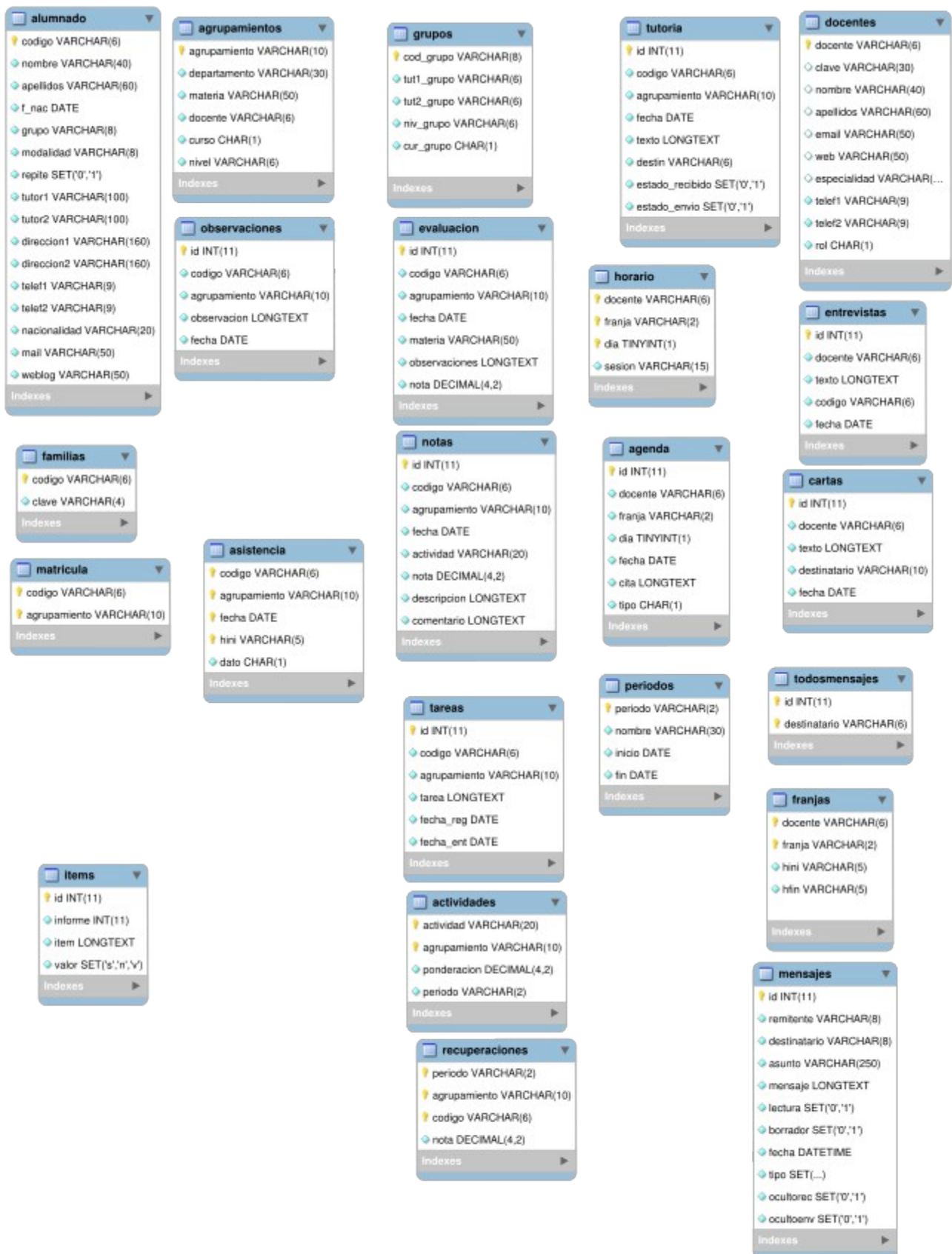


Illustration 4: Siesta database structure.

This application (Siestta) are also available for PDAs, it could be a valid solution but it is server-side with outdated technologies. Data structure from Siestta is standard and fully functional, and it could be partially reused by EduXes.

Source code are also shown: *calendario.php*. It shows us a PHP application which uses sessions variables and is not Model-View-Controller oriented.

```
<?php
session_start();
require('config.php');
require('idioma/'.$idioma.'');
include('funciones_calendario.php');
$docente = $_SESSION['usuario_sesion'];
//recogemos variables
$mes_actual = $_POST['mes'];
$anyo_actual = $_POST['anyo'];
if($mes_actual || $anyo_actual) {
    include('funciones.php');
    conecta();
}
//si es la primera vez que entramos, cargamos la fecha actual
if(!isset($mes_actual)) $mes_actual = date('m');
if(!isset($anyo_actual)) $anyo_actual = date('Y');
//presentamos ahora el calendario del mes actual o cargado
//tabla con nombre mes y año y las flechas para navegar
echo '
<br />
<table class="tablacentrada_i">
<tr>
<td>
<a href="#" onclick="navegaMes(\'\'.'$mes_actual.'\'\'.'
$anyo_actual.'\'\'.'menos\'')" title="\'\'.$id_anterior.'\'\'></a>
';
$nombre_mes = numero_mes_a_nombre($mes_actual);
....
```

Develop database structure: tables and relationships.

Data base structure looks like Illustration 5: EduXes Database structure

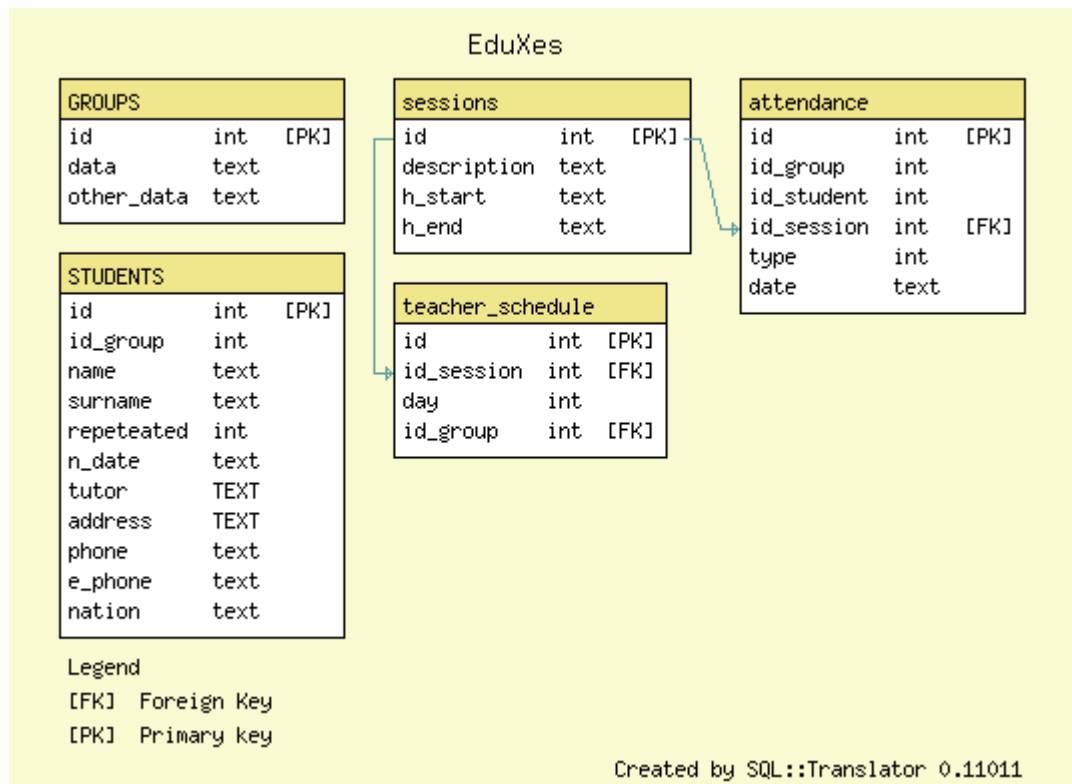


Illustration 5: EduXes Database structure

Preparation of development:

a) Build development environment:

Java Development Kit (JDK) version 1.6 is downloaded from:

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

As root user that file is unpacked into `/usr/lib/jvm` and configured to be the Java default:

```
# update-java-alternatives -s JDK_1.6_NAME
```

Eclipse Juno (4.6) is downloaded from its web-page:

<http://www.eclipse.org> Download->Linux 64 bits

Android Development Toolkit (ADT) is downloaded following instructions

on this page:

<http://developer.android.com/sdk/installing/installing-adt.html>

A new line is included into repository software (Help → Install New Software → Add):

<http://dl-ssl.google.com/android/eclipse/>

Next step is to select all the related software listed.

For Aptana Plugin line to add into Eclipse is:

<http://download.aptana.com/studio3/plugin/install>

Furthermore JQuery, JQueryMobile and Phonegap are needed, and were downloaded from their web sites:

JQuery 1.8.1 (no newer versions):

<http://jquery.com/>

JQuery will be copied into *assets/www/js* folder.

JQueryMobile version 1.1.1 from

<http://jquerymobile.com/>

JQueryMobile is a zip file which will be uncompressed and copied into *assets/www/js* folder.

PhoneGap - Cordova 1.8.1 is downloaded from this URL:

<https://github.com/phonegap/phonegap/zipball/1.8.1>

To create a PhoneGap application this very important instructions (Getting Started with Android) should be followed step by step:

http://docs.phonegap.com/en/1.8.1/guide_getting-started_android_index.md.html

b) Choose application name and folder's policy:

EduXes stands for "Educación" and "Xestión", is a educational

management software.

A folder is created (*assets/www/js*) which contains javascript (*.js) files except JQuery and JQueryMobile which is included into another folder (*assets/www/js/jquery*), do not forget style sheet files (*.css)

c) Make a simple application: only a blank page.

Getting started with Android is followed step by step.

d) Upload simple application into a git repository.

A Github⁸ account is created and a new application is initialized. This are the source code project page:

<https://github.com/joseantoniosa/EduXes/>

Source code are upload to GitHub:

```
$ git init
$ git add -A *
$ git remote add EduXes git@github.com:joseantoniosa/EduXes.git
$ git push origin master
```

Every time an update is going to be uploaded:

```
$ git add -A *
$ git commit -m 'CHANGES_DESCRIPTION'
$ git push origin master
```

e) JQueryMobile⁹ applications are structured in pages (<div data-role="page">), which are very similar to desktop applications windows, therefore, from Javascript code, to change to a new page `$.mobile.changePage("#daily_work")` opens daily_work p page.

⁸ <http://www.github.com>

⁹ <http://jquerymobile.com/demos/1.1.1/docs/>

Application Work-flow

Next illustration try to be self-explicative. Beginning at `onDeviceReady()` from `interface.fs` file. Inside each page several actions are performed: e.g., open and populate database, load list of groups (`loadSchedule()`), and user choose next step according options shown.

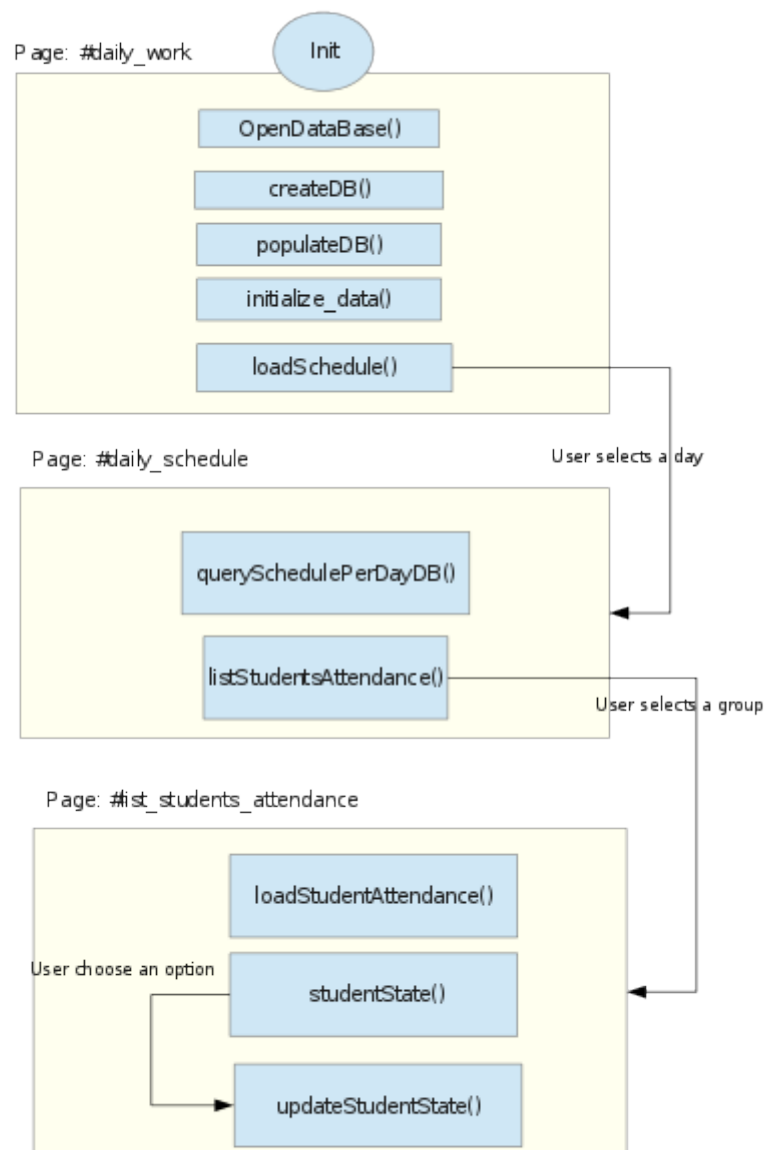


Illustration 6: Application Skeleton

Development:

There are two JavaScript files:

- *interface.js*: It contains information and decisions related to interface and application workflow, completely independent from database.
- *database.js*: It contains database related code: SELECT, INSERT, etc.

There is only one HTML file:

index.html only contains HTML framework, page properties, and static content.

There are three important files which contain documentation:

- TODO.txt. List of goals to be achieved and milestone reached.
- DATABASE.sql Data-base structure in SQL format.
- REAME.txt. Only contains general information about this application.

Next step in development is populate database.

f) Populate database with sample data. To test application, sample data are needed.

g)Groups: These pages are not active.

- Make a list of groups window (aka “page”).
- Make a group management window.

h)Students: These pages are not active.

- Make list of students window.
- Students management window (insert-update-delete students)

i) Timetable for actual date: list of groups for selected day.

Below `queryScheduleSuccess()` and `querySchedulePerDayDB()` functions are written, these functions fills daily_schedule page as shown in Illustration 6: Application Skeleton.

```
/*  
 * Main Window  
 */
```

```

function queryScheduleSuccess(tx, results) {
    var len = results.rows.length;
    $('#groups_day_ul').empty();
    var html;
    var id=0;
    var description="";
    var start = "";
    var t_id_session=-1;
    for (var i=0;i<len;i++) {
        id = results.rows.item(i).id;
        t_id_session = results.rows.item(i).t_id_session;
        start = results.rows.item(i).s_h_start;
        description = results.rows.item(i).description;
        html = "<li>";
        html += "<div data-role='fieldcontain'>";
        html += start;
        html += "<a data-role='button' data-iconpos='notext' style='float: right;";
        href='index.html#list_students_attendance' ";
        html += " onClick=\"listStudentsAttendance(" + results.rows.item(i).t_id_group +
        ","+t_id_session + ");\">" + description + "</a>";
        html += " ";
        html += "</div>";
        html += "</li>";
        $('#groups_day_ul').append(html);
    }
    $('#groups_day_ul').listview('refresh');
}

```

```

/* Query groups per day - Main Window -
*/
function querySchedulePerDayDB(tx){
    var query = "SELECT teacher_schedule.id_session, teacher_schedule.day,
teacher_schedule.id_group as t_id_group, teacher_schedule.id_session as t_id_session,
";
    query += " groups.id as g_id, groups.data as description, sessions.id as s_id,
sessions.h_start as s_h_start, sessions.h_end as s_h_text FROM teacher_schedule,
groups, sessions ";
    query += " WHERE day=" +week_day_global + " AND g_id=t_id_group AND t_id_session=s_id
ORDER BY t_id_session;";
    console.log("querySchedulePerDayDB:" + query);
    tx.executeSql(query,[], dbSuccessFunc = function(tx,rs){
        queryScheduleSuccess(tx, rs);},

```

```

    dbErrorFunc = function(tx, e) {
        if (tx.message) e = tx;
        alert(" There has been an error QuerySchedulePerDayDB: " + e.message);
        return false;
    });
}

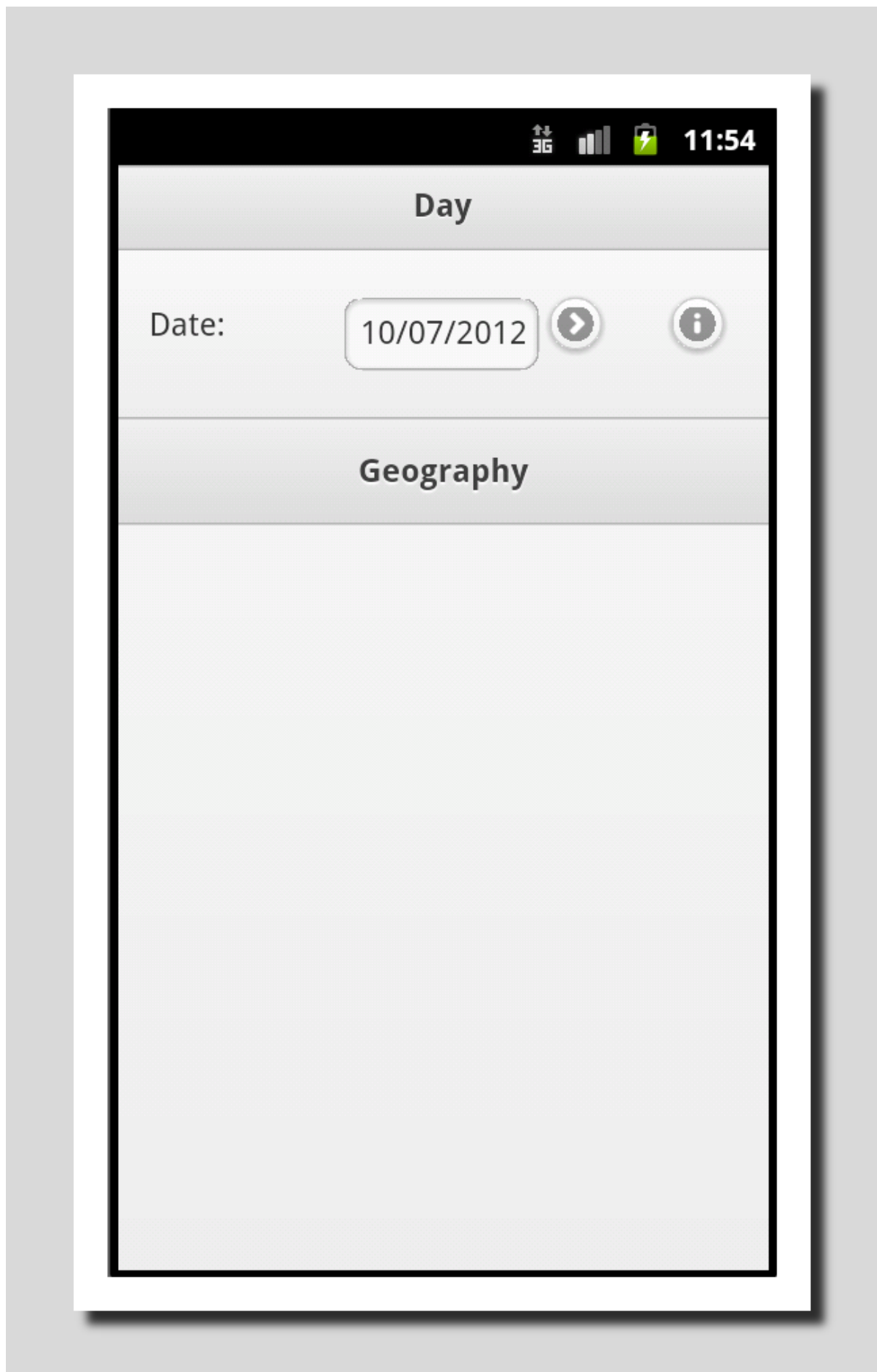
```

- j) Add attendance, misbehaviour for each student. Student attendance, misbehaviour, punctuality are set here.

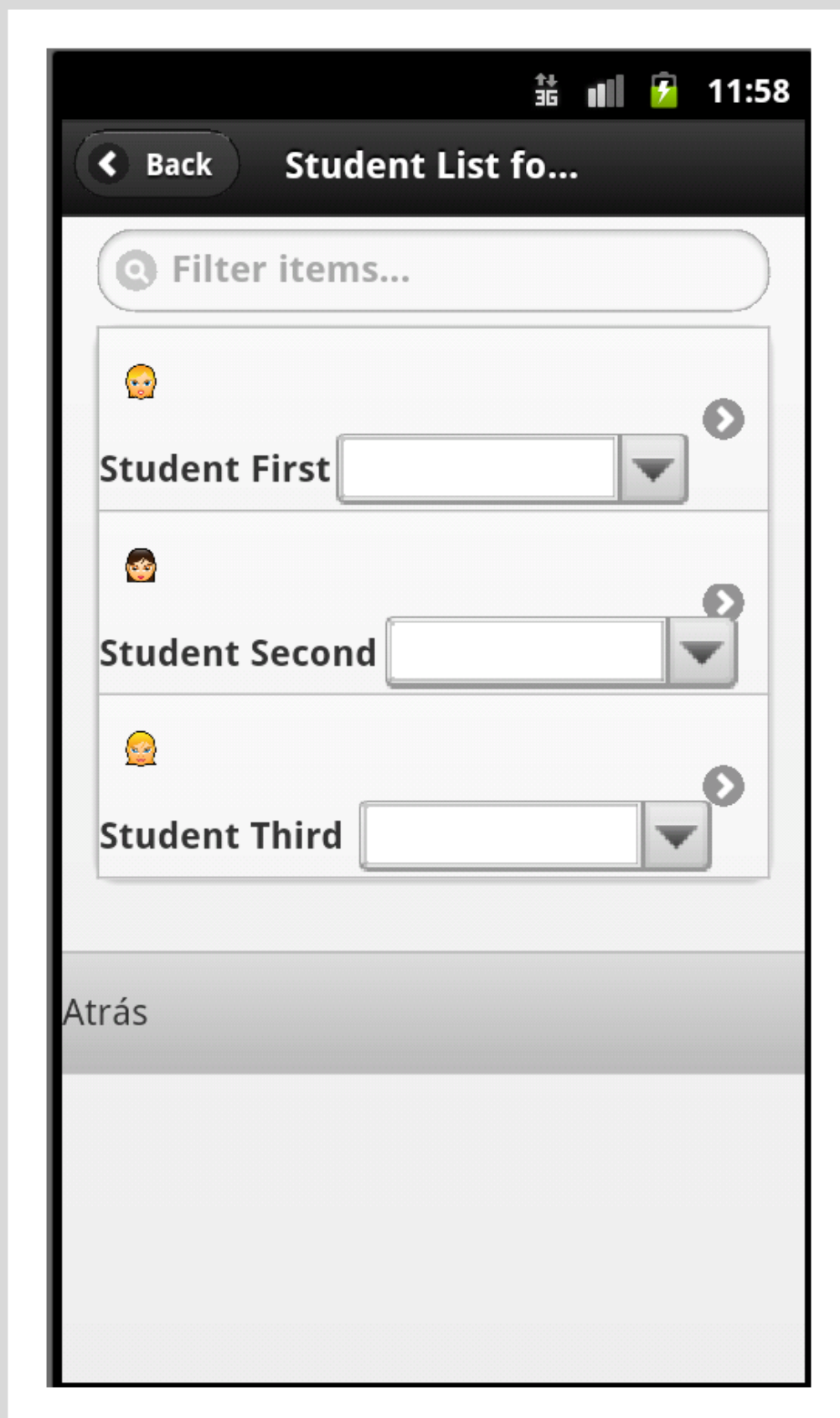
Tasks to be done

- k) Add error handling. Error handling is managed throw `dbErrorFunc`.
- l) Retrieve and insert data from and to database.
- m) List of attendance, misbehaviour incidents. A list of attendance will be carried on, reusing Siestta interfaces.
- n) Add activities grades for each student.
- o) List students marks and final mark. On a window, group, student name and surname will be shown, and a table with his/her marks.
- p) Activities management window. Add, remove and update activities. It includes name of activity and percent weight.
- q) Management of student notes. Student's notes could be included as an option.
- r) List of student notes. In a table-like window, notes will be displayed.

Next pages show several windows/pages already implemented.







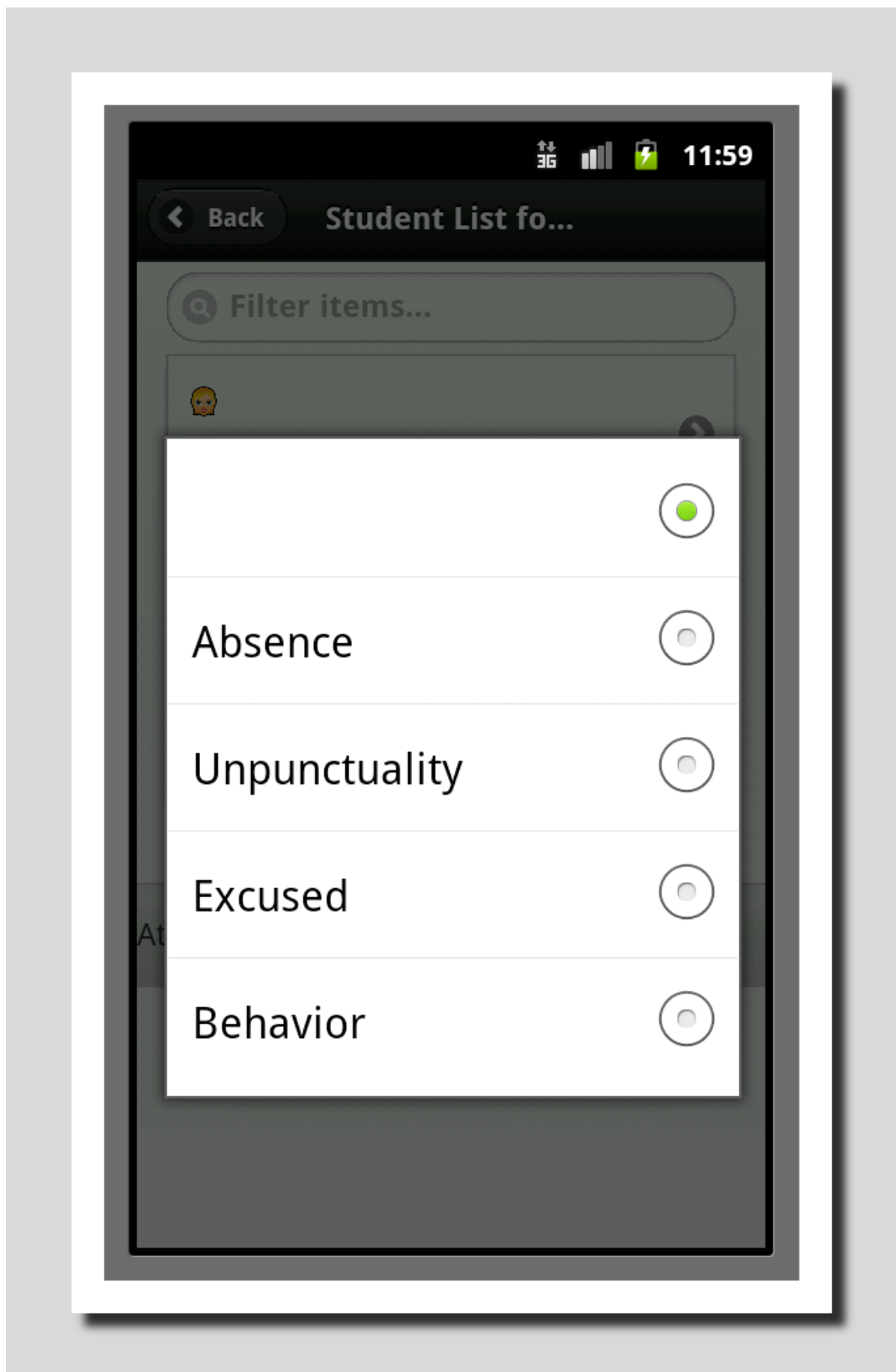


Illustration 10: Student absence/unpunctuality/behaviour.

Test into real hardware

Android 2.3.3 mobile phone will be used to test application. This is not done yet. Seems to be problems copying .apk file into disk phone.

Save or download data from database to disk

SQLite database should be saved into an external file.¹⁰

Xade web interface

- s) Study Xade web interface. Using accessibility tools Xade will be parsed looking for information: students data, groups, etc.
- t) Develop an ad-hoc application for retrieve Xade's data.

Another developments

1. Develop an ad-hoc application for store data. This application will be platform specific.
2. Synchronization with a custom server or with Xade. With above information about Xade interface it could be possible to upload some information into Xade.

Test units.

Need to be written for each capability and function.

User documentation

Manual with images. Need to be written with screen-shots or a screen-cast.

Developer's documentation

API should be documented, application overview will be written and several diagrams will be drawn.

¹⁰ http://docs.phonegap.com/en/1.8.1/cordova_file_file.md.html#File

Community

Find out a site to host: forum, bug reports, documentation and to download application itself.¹¹ This is a very important milestone, because could allow to improve application itself, and to listen users opinions.

Personal evaluation of the practicum

Firstly, it was really difficult to prepare environment, because there a lot of incompatibilities among plug-ins, Eclipse versions and son on, at last I think it was my fault, I used to find out the most complicated solutions for not-so-difficult problems.

On the other hand, coding was effortless, despite of application work-flow confusion and increasing complexity. My mentor, Manuel Rego help me a lot, he told me an overview of how JQueryMobile, PhoneGap works, and how to work step-by-step; I should follow his code-style (e.g. use append function and do not work directly with strings), but I have felt more comfortable with my old-fashioned coding-style. Of course I have reused several SergasApp functions.

Eventually this application could be a never ending story, really difficult and complicated. Several functions should be rewritten.

Honestly, I am not really a good coder, nor a good graphical user interface designer neither a good database designer. At this point I realized that Siestta developer did a good job. I feel far away from be a good coder, but it works, at least for me.

¹¹http://en.wikipedia.org/wiki/Comparison_of_open_source_software_hosting_facilitie
[s](#)

Acknowledgements:

- To my mentor. Manuel Rego.
- To Ramón Castro Perez who sent me a patch to allow Siestta work.
- José Manuel Ciges Regueiro, which memory I used as a template for my own work.

Tools used:

- Sqlfairy. Transforms SQL language into a png image.
- LibreOffice 3.5.4.2 to write this document. ¹²
- Gimp 2.8.2 to get screen-shots.
- GNU/ Debian Wheeze October 2012

References:

- Comparison among open source hosting facilities:
http://en.wikipedia.org/wiki/Comparison_of_open_source_software_hosting_facilities
- W3C Database Specifications: <http://www.w3.org/TR/webdatabase/>
- PhoneGap Storage:
http://docs.phonegap.com/en/1.8.1/cordova_storage_storage.md.html
- JQueryMobile: <http://jquerymobile.com/demos/1.1.1/>
- JQuery: <http://docs.jquery.com/>
- SergasApp: <http://mrego.github.com/sergasapp/>
- EduXes: <https://github.com/joseantoniosa/EduXes/>
- Siestta: <http://siestta.sourceforge.net/doc/index.html>
- Xade Web: <https://auth.edu.xunta.es/cas/login>

This document is hosted at:

https://github.com/joseantoniosa/EduXes/blob/master/docs/PhoneGap_Project_MSWL_Memory.odt

¹² I rather LaTeX, but LibreOffice is straightforward.