

Mobile Computing (CSE 40814/60814)

Topic: Using local and remote databases (SQL, Parse) for both Android and iOS

1. Introduction

1.1. Local databases (on the device)

- CoreData for iOS
- SQLite for Android and iOS

1.2. Remote databases (for both iOS and Android)

- Parse.com
- Remote Server (e.g. mysql)

2. Demos

2.1. Remote Databases

- Parse for Android and iOS
- MySQL on a remote server

3. Resources

Mobile Computing (CSE 40814/60814)

1. Local databases (on the device)



Core Data

SQLite

Core Data is a schema-driven object graph management and persistence framework. Fundamentally, Core Data helps you to save model objects (in the sense of the model-view-controller design pattern) to a file and get them back again.



SQLite

SQLite is a relational database management system and is a popular choice for embedded systems (i.e. mobile devices)

“Android provides full support for SQLite databases.”



<https://developer.apple.com/technologies/mac/data-management.html>



<http://developer.android.com/guide/topics/data/data-storage.html>

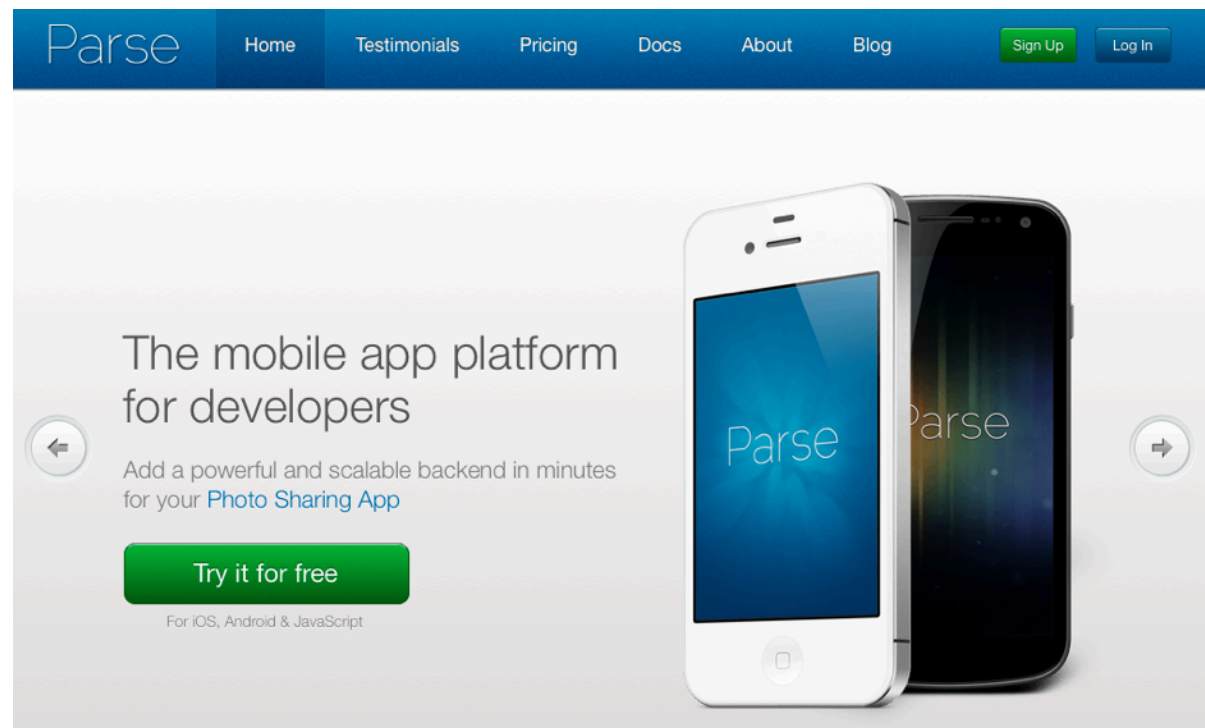
Mobile Computing (CSE 40814/60814)

2. Remote databases

Remote Server



Parse.com



In Class Demo

Mobile Computing (CSE 40814/60814)

In-class Demos

1. Parse.com

Suitable for
iOS & Android
Projects

2. Remote Server

Suitable for
iOS & Android
Projects

3. iOS Core Data

Core Data Tutorial for iOS
Getting Started

4. SQLite Android

Starting Point

Mobile Computing (CSE 40814/60814)

Parse.com

1

<https://parse.com/apps/quickstart>

2

Get started

mobcom

Individual Developer ▼

Start using Parse ▶

1 Sign up for Parse
Set up your Parse account

2 Get started
Create your first Parse app

Mobile Computing (CSE 40814/60814)

Parse on Android - Example/Demo

```
package com.parse.starter;

import com.parse.Parse;
import com.parse.ParseACL;

import com.parse.ParseUser;

import android.app.Application;

public class ParseApplication extends Application {

    @Override
    public void onCreate() {
        super.onCreate();

        // Add your initialization code here
        public void onCreate() {
            Parse.initialize(this,
                "dUlyYjwjUQNYkx8rX2nD1P7B6Ij3hnXMbeSCUx5i",
                "mVmevaUxuFh1xVkPlwaqCtZCD8heIzbN92Y0vZc");
        }

        ParseUser.enableAutomaticUser();
        ParseACL defaultACL = new ParseACL();
        // Optionally enable public read access
        // by default.

        // defaultACL.setPublicReadAccess(true);
        ParseACL.setDefaultACL(defaultACL, true);

    }
}
```

```
package com.parse.starter;

import com.parse.ParseObject;
import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.Toast;

public class ParseStarterProjectActivity extends Activity implements
OnClickListener{
    /** Called when the activity is first created. */
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        Button tstBtn = (Button) findViewById(R.id.button1);
        tstBtn.setOnClickListener(this);

    }

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        if (v.isPressed()){
            Toast.makeText(v.getContext(), "Button Pressed",
                Toast.LENGTH_SHORT).show();

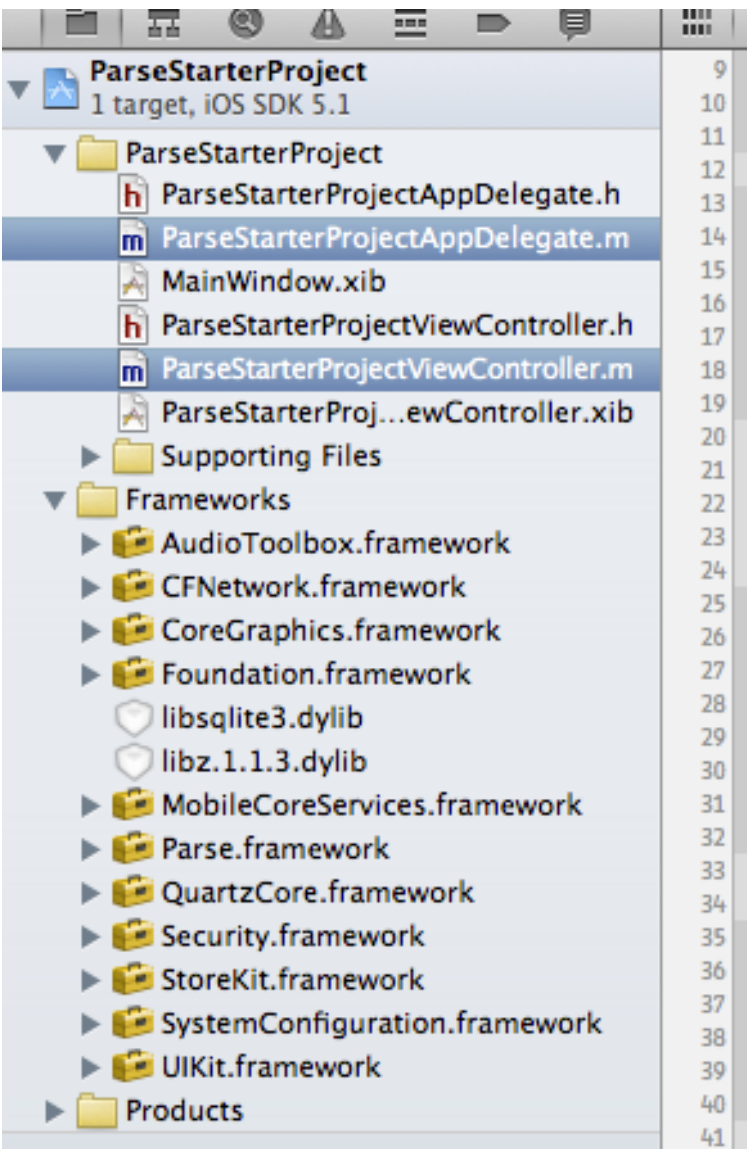
            ParseObject testObject = new ParseObject("TestObject");
            testObject.put("Sal", "Rules!");
            testObject.saveInBackground();

        }

    }
}
```

Mobile Computing (CSE 40814/60814)

Parse for iOS - Example/Demo



Changes to AppDelegate.m file

```
// Implement viewDidLoad to do additional setup after loading the
view, typically from a nib.
- (void)viewDidLoad
{
    [super viewDidLoad];
    PFObject *testObject = [PFObject
objectWithClassName:@"TestObject"];
    [testObject setObject:@"ios" forKey:@"foo"];
    [testObject save];
}
```

Changes to ViewController.m file

```
- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:
(NSDictionary *)launchOptions
{
    // *****
    // Uncomment and fill in with your Parse credentials:
    // [Parse setApplicationId:@"your_application_id"
clientKey:@"your_client_key"];
    [Parse setApplicationId:@"dU1yYwjUQNykx8rX2nD1P7B6Ij3hnXMbeSCUx5i"
clientKey:@"mVmevaUxuFh1xVkPlwaqCtZCD8heLIzbN92Y0vZc"];
    //
    // If you are using Facebook, uncomment and fill in with your Facebook App Id:
    // [PFFacebookUtils initializeWithApplicationId:@"your_facebook_app_id"];
    // *****
}
```


Mobile Computing (CSE 40814/60814)

2. *Remote databases*

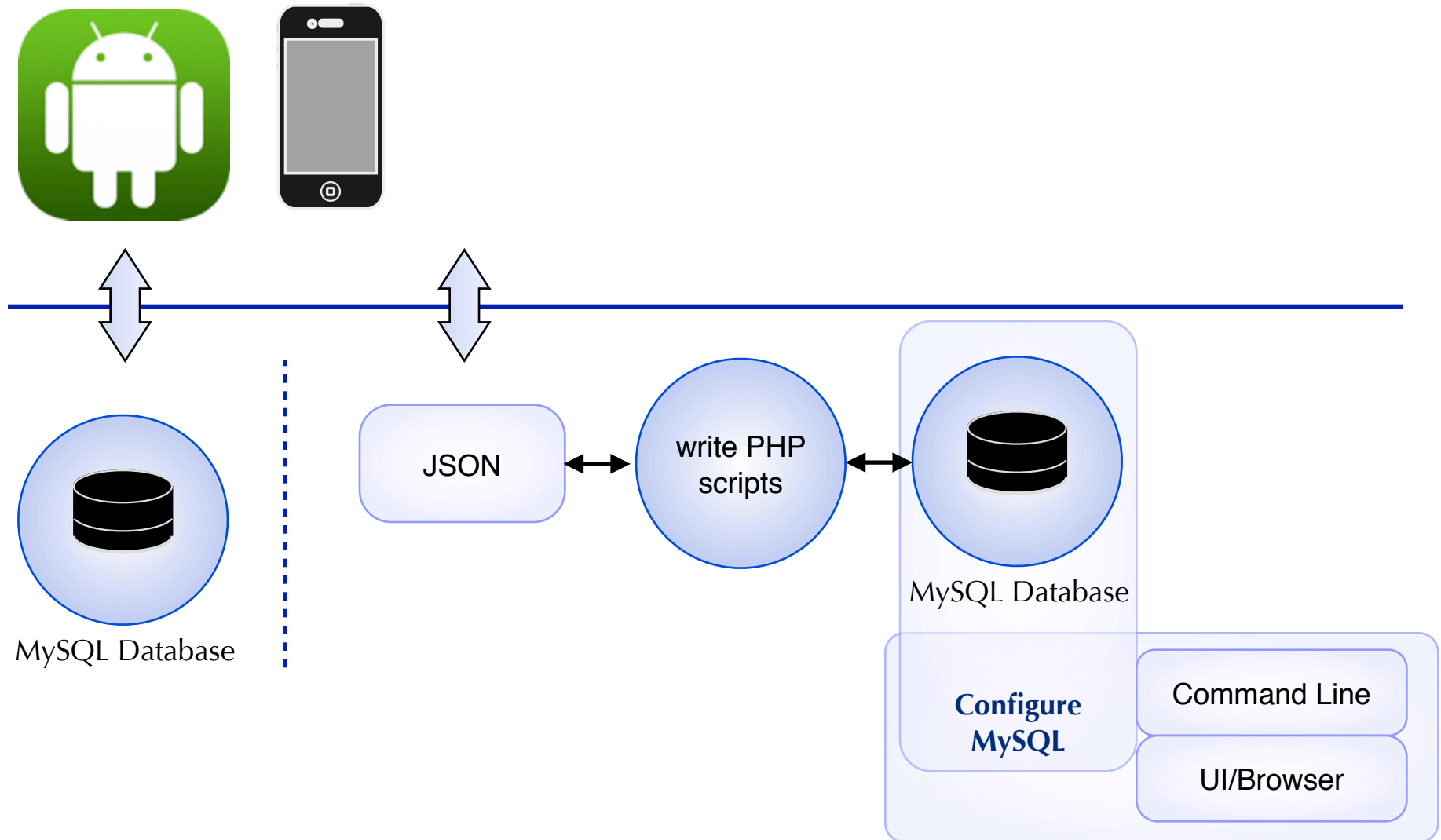
1. Hosted on a server reachable over the internet
2. Configured and maintained as a service on a Windows or Linux system
3. Pushing/Fetching Data is typically done using PHP scripts
4. May be accessed on Android & iOS apps
5. The database engine is typically MySQL



Typically MySQL Powered

Mobile Computing (CSE 40814/60814)

MySQL Database Demo



Mobile Computing (CSE 40814/60814)

MySQL Database Demo

Command Line

```
mysql -u tremcam -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 384
Server version: 5.1.52 Source distribution
```

Copyright (c) 2000, 2010, Oracle and/or its affiliates. All rights reserved.

This software comes with ABSOLUTELY NO WARRANTY. This is free software, and you are welcome to modify and redistribute it under the GPL.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

```
mysql> use neurobit;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -R
```

```
Database changed
mysql> describe TremCamTbl;
```

Field	Type	Null	Key	Default	Extra
subjectid	int(10)	NO	PRI	NULL	auto_increment
email	varchar(64)	YES		NULL	
datetimestamp	datetime	YES		NULL	
ctrl_tst_category	tinyint(1)	YES		NULL	
accelX	mediumblob	YES		NULL	
accelY	mediumblob	YES		NULL	
accelZ	mediumblob	YES		NULL	
fft_result	mediumblob	YES		NULL	

PhpMyAdmin (Browser)

The screenshot shows the PhpMyAdmin web interface. The top navigation bar indicates the server is 'localhost', the database is 'neurobit', and the table is 'TremCamTbl'. The main panel displays a SQL query: `SELECT * FROM 'TremCamTbl' LIMIT 0, 30`. Below the query, there are options to show 30 rows starting from record # 30. The table view shows columns: subjectid, email, datetimestamp, ctrl_tst_category, accelX, accelY, accelZ, and fft_result. The first two rows of data are visible: one with subjectid 14 and email test@nd.com, and another with subjectid 35 and email gsgdcg.

This is how you design and maintain the database

Next, is an enumerated list of steps to interface with mysql from an Android app.

MySQL Database Demo Continued

1

Configure *MySQL*
– database: *neurobit*
– table: *UsersTbl*
– Fields: *name, email, passwd, datecreated*

2

Write a PHP script to send data to the database

```
<?php
// array for JSON response
$response = array();

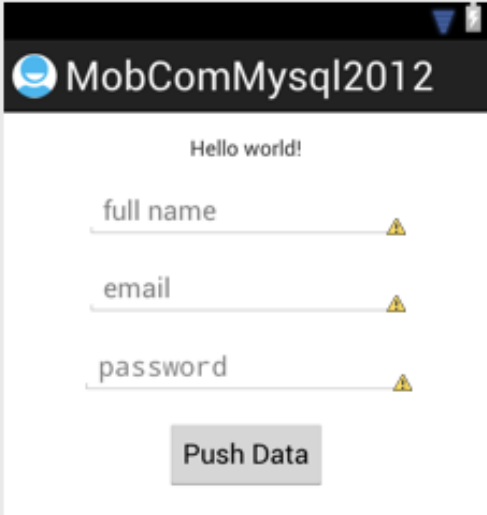
// check for required fields
if (isset($_POST['email']) && isset($_POST['name']) &&
    isset($_POST['passwd']) && isset($_POST['datecreated']))
{

    $name      = $_POST['name'];
    $email     = $_POST['email'];
    $passwd    = $_POST['passwd'];
    $datecreated = $_POST['datecreated'];

    $con = mysql_connect("localhost","mysqluser","userpass");
    if (!$con){
        die('Could not connect: ' . mysql_error());
    }
    mysql_select_db("neurobit", $con);
    .....
```

3

Start a new Android Project
– Design the UI



MySQL Database Demo Continued

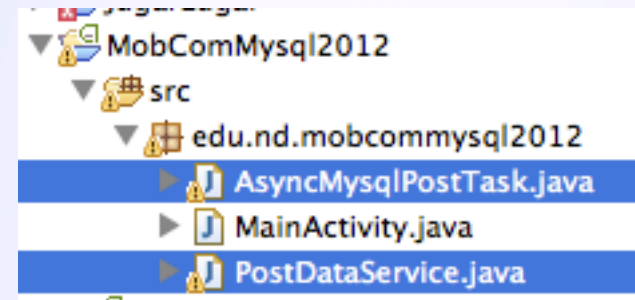
4

Initialize the UI and add OnClickListener to button

```
private void initializeUI(){
    fullnameET = (EditText) findViewById(R.id.editText1);
    emailET    = (EditText) findViewById(R.id.editText2);
    passET     = (EditText) findViewById(R.id.editText3);
    btn        = (Button)   findViewById(R.id.button1);
    btn.setOnClickListener(new OnClickListener(){
        @Override
        public void onClick(View arg0) {
            if (arg0.getId() == R.id.button1) {
                if (editTextFieldsNotEmpty()) {
                    AsyncMysqlPostTask postServiceTask = new
                        AsyncMysqlPostTask();
                    postServiceTask.execute(
                        fullnameET.getText().toString(),
                        emailET.getText().toString(),
                        passET.getText().toString(), this);
                }
            }
        }
    });
}
```

5

Pushing the data over a Http/Json can be done using AsyncTask (these files below handle the internet php data)



The internet interaction with mysql is typically done on a different thread from the UI, so using AsyncTask is one method of implementing the Http post/get calls.

6

Run The App

(see src code for more details)

7

3G 10:30

MainActivity

Hello world!

John Doe

asd@nd.edu

.....

Push Data

Here is what the UI will look like

8

8

192.168.1.148 / localhost / neurobit / UsersTbl | phpMyAdmin 2.11.11.3

192.168.1.148 / localhost / n... x

How to connect Android with ... x

Invalid Query: Column count ... x

Mercedes Sosa Ft Calle 13-C

192.168.1.148/phpmyadmin/index.php?db=neurobit&table=UsersTbl&lang=en-utf-8&target=t

☆

f

android, datetime, format s

Most Visited

Getting Started

insideND Login ...

Latest Headlines

Yahoo! Mail

Gmail

Latest Headlines

phpMyAdmin

Database

neurobit (2)

neurobit (2)

TremCamTbl

UsersTbl

			email	name	passwr	datecreated	dateupdated	recordid
<input type="checkbox"/>			asdf	sal	sal@xzc.bnm	2012-08-29 05:37:38	0000-00-00 00:00:00	1
<input type="checkbox"/>			asdfg	Sal Agui	asder@fghj.j	2012-08-29 06:32:54	0000-00-00 00:00:00	6
<input type="checkbox"/>			asdfgh	chavo	ch@vo.ru	2012-08-29 07:30:42	0000-00-00 00:00:00	7
<input type="checkbox"/>			qw@ert.yui	sadf	sa	2012-08-29 07:36:22	0000-00-00 00:00:00	8
<input type="checkbox"/>			qw@ertt.yui	sadf	sadfr	2012-08-29 07:39:36	0000-00-00 00:00:00	9
<input type="checkbox"/>			asd@zxx.bn	qwertr	wrt	2012-08-31 21:18:35	0000-00-00 00:00:00	10
<input type="checkbox"/>			asd@nd.edu	John Doe	wrtdfghj12	2012-08-31 22:29:13	0000-00-00 00:00:00	11

See evidence that the data was pushed to the remote mysql database (in red)

The same is done to fetch/get data from mysql
(contact me if you want more details)

Mobile Computing (CSE 40814/60814)

Resources

Download Tutorial Source Code



<https://github.com/wndsword/mobcomtutorial/tree/AndroidMysql>



<https://github.com/wndsword/mobcomtutorial/tree/Docs>



***Core Data
Tutorials***

<http://www.raywenderlich.com/934/core-data-on-ios-5-tutorial-getting-started>

<http://timroadley.com/2012/02/09/core-data-basics-part-1-storyboards-delegation/>

***developer.
apple.com***

[Core Data Core Competencies](#)