

Západočeská univerzita v Plzni  
Fakulta aplikovaných věd  
Katedra informatiky a výpočetní techniky

## **Diplomová práce**

### **Docházka a výkazy práce pro systém IMIS na platformě Android**

# Prohlášení

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 11. dubna 2013

Maxipes Fík

# Abstract

Text of abstract.

# Obsah

<b>1</b>	<b>Úvod</b>	<b>1</b>
<b>2</b>	<b>Současný systém</b>	<b>2</b>
2.1	Oracle forms . . . . .	2
2.2	Architektura . . . . .	2
2.2.1	Triggery . . . . .	4
2.2.2	Seznam hodnot . . . . .	5
2.3	Datový model . . . . .	6
2.4	Uživatelské rozhraní . . . . .	6
2.4.1	Zápis příchodů a odchodů . . . . .	6
2.4.2	Výkaz práce . . . . .	7
<b>3</b>	<b>Analýza</b>	<b>8</b>
3.1	Architektura . . . . .	8
3.1.1	Přímé připojení k databázi . . . . .	8
3.1.2	Oracle Database Mobile Server . . . . .	8
3.1.3	Webová služba . . . . .	8
3.2	Datová vrstva . . . . .	8
3.2.1	Práce s datumem a časem . . . . .	8
3.2.2	Kritika datové vrstvy . . . . .	9
3.3	Business logika . . . . .	9
3.3.1	Triggery . . . . .	9
3.3.2	Databázové balíčky a uložené procedury . . . . .	10
3.3.3	Forms knihovny . . . . .	10
3.4	Uživatelské rozhraní . . . . .	10
3.4.1	LOV . . . . .	10
<b>4</b>	<b>Zabezpečení</b>	<b>11</b>
<b>5</b>	<b>Webová služba</b>	<b>12</b>

<b>6</b>	<b>Android aplikace</b>	<b>13</b>
6.1	Funkcionalita . . . . .	13
6.1.1	Nastavení a konfigurovatelnost . . . . .	14
6.1.2	Uživatelská přívětivost . . . . .	14
6.2	Architektura . . . . .	14
6.3	Business logika . . . . .	15
6.4	Android komponenty . . . . .	15
6.5	Ukládání dat . . . . .	15
6.6	SQLite . . . . .	16
6.7	REST . . . . .	17
6.8	Synchronizace . . . . .	17
6.9	Zabezpečení . . . . .	17
6.10	Zpětná kompatibilita . . . . .	17
6.11	Budoucí rozšiřitelnost . . . . .	17
6.12	Vytváření grafů . . . . .	17
6.13	Chybové reporty . . . . .	17
<b>7</b>	<b>Testování</b>	<b>18</b>
7.1	O čem psát... . . . .	18
7.2	Zásady pro vypracování . . . . .	18

# 1 Úvod

## 2 Současný systém

IMIS = Integrovaný manažerský informační systém

### 2.1 Oracle forms

Oracle forms je softwarový produkt vyvinutý společností Oracle. Slouží k vytváření formulářů, které interagují s Oracle databází. Jako programovací jazyk využívá PL/SQL. Produkt byl původně požíval terminálové rozhraní pro komunikaci se serverem. Později byl přepracován do architektury klient-server.

Prostředí běhu zajišťuje defaultní správu transakcí. Díky tomu je Oracle Forms silný nástroj pro efektivní vývoj aplikací, jejichž primárním cílem je přístup k datům uložených v databázi.

**PL/SQL** PL/SQL (Procedural Language/Structured Query Language) je procedurální nadstavba jazyka SQL od firmy Oracle založená na programovacím jazyku Ada.

### 2.2 Architektura

Z hlediska architektury se Oracle Forms aplikace skládá z těchto celků:

**Modul formuláře** A form (or form module) is the main component that anchors an application and provides the necessary code to interact with the datasource and the user interface. The underlying database data is reflected in multiple items, including fields, check boxes, and radio groups. A form is logically organized into blocks.

- **Datový blok** The data block serves as a bridge to the underlying data and provides an abstraction for how the data is reached. A data block can be associated with a specific database table (or a view), a stored

procedure, a FROM clause query, or a transactional triggers. By default, the association between a data block and the database allows for automatic access and manipulation of data in the database.

- **Řídící blok** A control block is a block that does not have a relationship to a database table. Control blocks can contain items of any type (buttons, text items, check boxes, etc.), however, regardless of item type, all of the items in a control block are control items—that is, items that do not correspond to columns in a database table.

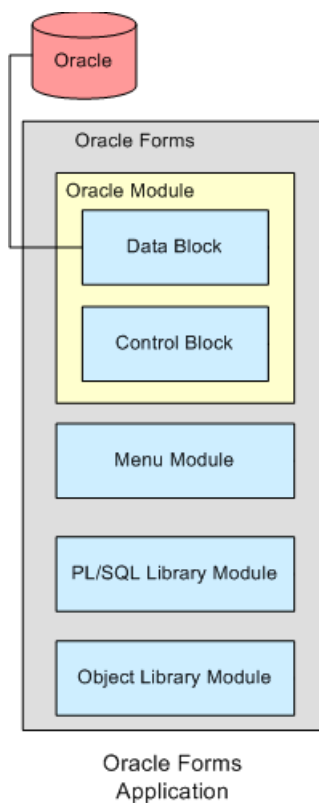
**Modul menu** The menu module consists of a hierarchy of menus. Each menu consists of the items that can be selected. Menu modules are usually attached to form modules. Every form module has a default menu that includes the commands for all basic database operations, such as insert, update, delete, and query.

**Modul PL/SQL knihovny** The PL/SQL library contains reusable code invoked by other form, menu, or library modules. The code, also called a program unit, can be user-named functions, procedures, or packages. The program unit is stored and executed on the client. The program units are not for database-related code alone. They may contain purely business logic. Libraries make all the PL/SQL code available to the form without being part of the form. The libraries are loaded dynamically. Multiple forms can attach the same PL/SQL library.

**Modul knihovny objektů** Object Library

The object-oriented programming concept of reusing code is implemented in Oracle Forms by the object library and PL/SQL library modules. The Object library contains reusable objects and the PL/SQL library contains reusable code.





TODO obrazek upravit - prepsat Oracle Module na Form Module

### 2.2.1 Triggery

Aplikace v Oracle pracuje s následujícími typy triggerů:

- Block-processing triggers: - Block processing triggers fire in response to events related to record management in a block.
- Interface event triggers: - Interface event triggers fire in response to events that occur in the form interface.
- Master-detail triggers: - Form Builder generates master-detail triggers automatically when you define a master-detail relation between blocks. The default master-detail triggers enforce coordination between records in a detail block and the master record in a master block.

- Message-handling triggers: - Form Builder automatically issues appropriate error and informational messages in response to runtime events.
- Navigational triggers: - Navigational triggers fire in response to navigational events.
- Query-time triggers: - Query-time triggers fire just before and just after the operator or the application executes a query in a block.
- Validation triggers: - Validation triggers fire when Form Builder validates data in an item or record.

### **2.2.2 Seznam hodnot**

A (LOV) is a pop-up window that provides the user with a selection of values. The values can be static or populated by querying the database. LOVs are populated using columns returned by record groups. Check the Record Group property of the LOV for the record group which is used to provide values.

## 2.3 Datový model

## 2.4 Uživatelské rozhraní

### 2.4.1 Zápis příchodů a odchodů

[illegible]

+popsat z pohledu uzivatele

## 2.4.2 Výkaz práce

VX42 - Výkaz práce (01.03.2012) CCA

**VÝKAZ PRÁCE**

Uživatel: 06/2012

Zaměst. 1520 MARTIN KADLEC BC Kód KDA Stav : V P S Z  
 Vedoucí 1429 JANSÁ JIŘÍ Střed PRG (počet zázn.) 0 15 0 0  
 Celkem za období z docházky (vyhodn.) 056:40 z docházky 060:54 z výkazu 056:40

**Výkaz práce - hodiny**

Datum	S	Zakázka	Prac.	Hodin	Požad.	Hláš.	Organizace	Popis činnosti
PÁ 01/06/2012	P	R-CCA-FIRMA	1 1	KDAREZC	02:00			sm
SO 02/06/2012	P	K-VV-N-2012	4 1	KDAREZC	01:00			firemni smernice
NE 03/06/2012								
PO 04/06/2012								
ÚT 05/06/2012								
ST 06/06/2012								
ČT 07/06/2012								
PA 08/06/2012	S	Zakázka	Auto	Km	Organizace	Popis činnosti		
SO 09/06/2012	V							
NE 10/06/2012								
PO 11/06/2012								
ÚT 12/06/2012								
Celkem z doch.		Km	Hodin	Rozdíl				
03:00		A	03:00	00:00	14:00(P)-15:00(P),16:59(P)-17:00(P),			

Alt1=PŘÍSTUPY Alt2=ZMĚNA OBDOBÍ Alt3=ODMĚNY OSV Alt4=SUMA Alt5=POTVRZENÍ  
 Alt6=IMP:KNIHY J. Alt7=IMP:VÝKAZU ^P=TISK

+popsat z pohledu uzivatele

## 3 Analýza

### 3.1 Architektura

#### 3.1.1 Přímé připojení k databázi

#### 3.1.2 Oracle Database Mobile Server

Oracle Database Mobile Server 11g - zajišťuje synchronizaci mezi Oracle db a mobilním zařízením, zamítнуто z licenčních důvodů, možná by stálo za to to víc prozkoumat a něco o tom napsat

#### 3.1.3 Webová služba

### 3.2 Datová vrstva

#### 3.2.1 Práce s datem a časem

Při návrhu datového modelu jsem řešil problém pomocí jakého datového typu vyjadřovat údaj o čase či datu. V Oracle databázi je použit datový typ Date. SQLite databáze nabízí tři způsoby jako ukládat informaci o čase:

- **TEXT** podle ISO8601 normy ve formátu "YYYY-MM-DD HH:MM:SS.SSS".
- **REAL** podle Juliánského kalendáře, počet dní od poledne 24. Listopadu roku 4714 před Kristem (Greenwichského času).
- **INTEGER** jako Unix Time, počet sekund 1970-01-01 00:00:00 UTC.

Pro uložení v SQLite databázi jsem zvolil typ INTEGER. V aplikaci (Android klient, webová služba) jsem se rozhodl reprezentovat časový údaj pomocí primitivního typu long. Měl jsem k tomu řadu dobrých důvodů:

- odpadá starost s formátem datumu při serializaci a deserializaci JSON řetězce
- snadné porovnávání hodnot pomocí relačních operátorů
- sníží se počet konverzí v aplikaci (např. pro výpočet pozice pro vykreslení komponenty v UI)

Také jsem se ujistil, že rozsah typu long je pro potřeby aplikace dostačující. Srovnání použitých datových typů je znázorněno v tabulce 3.1.

Datový typ	Minimální hodnota	Maximální hodnota	Přesnost
Oracle Date	January 1, 4712 BCE	December 31, 4712 CE	sekundy
SQLite INTEGER			sekundy
Java long	2.12.292269055 BC	17.8.292278994 AD	milisekundy

Tabulka 3.1: Datové typy reprezentující časový údaj

### 3.2.2 Kritika datové vrstvy

co se mi nelibilo a co bych navrhl jinak a jak, návrh prichody/odchody - jeden radek, chybi primarni klic - ROWID jako unikatni identifikator, problemy ktere to prinasi, format casu - problemy s prevodem

## 3.3 Business logika

existuje nekajá možnosť prevodu formsů do javy - oracle adf - co to je, co to resi, proc to neresi muj problem

### 3.3.1 Triggery

jen ty, jejichž funkčnost bude muset být implementována.

- On-Delete, On-Insert, On-Update, Pre-Delete, Pre-Insert, Pre-Update
- When-Validate-Item

### **3.3.2 Databázové balíčky a uložené procedury**

### **3.3.3 Forms knihovny**

## **3.4 Uživatelské rozhraní**

### **3.4.1 LOV**

jaka alternativa v androidu

## 4 Zabezpečení



## **5 Webová služba**

## 6 Android aplikace

### 6.1 Funkcionalita

Na základě analýzy současného systému a potřeb zaměstnanců byla vybrána k implementaci následující funkčnost:

#### Docházka

- Přehledné zobrazení událostí docházky daného zaměstnance
- Uživatel má možnost přidávat, ediovat a mazat svoje události
- Aplikace zajišťuje automatickou synchronizaci těchto údajů s firemní databází
- Zobrazení poměru typů docházkových událostí za dané období

#### Aktuální přítomnost na pracovišti

- Zobrazení seznamu zaměstnanců aktuálně přítomných na pracovišti
- Uživatel má možnost spravovat seznam svých "oblíbených" zaměstnanců a tento seznam zobrazovat přednostně

#### Výkazy práce

- Zobrazení poměru typů zakázek za dané období
- Zobrazení vývoje vývoje daného typu zakázky v daném období
- Možnost zobrazení těchto údajů i za jiné zaměstnance

### 6.1.1 Nastavení a konfigurovatelnost

Aplikace si musí pamatovat údaje nutné pro snadnou obsluhu tzn. uživatelské jméno a heslo, adresu umístění webové služby a tyto údaje jsou konfigurovatelné.

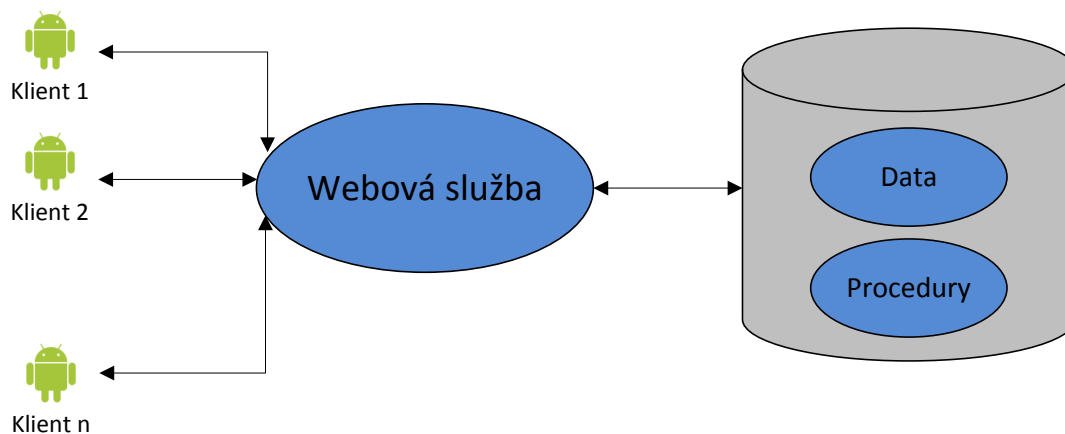
Dále aplikace umožní uživateli konfigurovat vzhled některých komponent, jako je barva typu události v docházce a typu záznamu ve výkazech.

### 6.1.2 Uživatelská přívětivost

Uživatelské rozhraní aplikace klade důraz na přehlednost, ergonomii a časově efektivní obsluhu.

## 6.2 Architektura

Android aplikace funguje jako tenký klient, který se připojuje k webové službě. Webová služba používá REST architekturu a přistupuje k samotné databázi.



- Webová služba - Java EE 6, aplikační server GlassFish

- Databáze - Oracle 10g, obsahuje navíc databázové procedury, které se používají v současných formulářích
- Android - obsahuje persistentní úložiště, obsahuje záznamy o docházce, úložiště se bude automaticky synchronizovat ve stavu online s databázovým serverem prostřednictvím webové služby

TODO prepsat srozumitelneji TODO schema komunikace -HHTP, JDBC

## 6.3 Business logika

prijde to do webové služby - důvody

## 6.4 Android komponenty

1. komponenty pro sync a auth, provazani s android ucetm
2. CursorLoader
3. Async task
4. nestandardni UI
5. modifikace adapterview
6. custom UI - viewgroup

+ nejaka ukazka konkretniho pouziti

## 6.5 Ukládání dat

### Sdílené preference

ukládá primitivní datové typy ve tvaru klíč-hodnota. Slouží k uložení nastavení specifických pro aplikaci. Toto nastavení může být uloženo jako soukromé, kdy mohou k datům přistupovat pouze aplikace sdílející stejné *Linux*

*user ID.*

V aplikaci používám toto úložiště pro nastavení síťového připojení (doména a port webové služby) a barevného nastavení pro typy docházkových událostí.

Načtení dat se typicky odehrává v `onCreate()` metodě aktivity:

```
%\begin{lstlisting}
SharedPreferences settings = getSharedPreferences(PREFS_NAME,
Context.MODE_PRIVATE);
int color = settings.getInt("color", defaultColor);
%\end{lstlisting}
```

Uložení dat se typicky odehrává v `onStop()` metodě aktivity:

```
SharedPreferences settings = getSharedPreferences(PREFS_NAME,
Context.MODE_PRIVATE);
SharedPreferences.Editor editor = settings.edit();
editor.putInt("color", userColor);
editor.commit();
```

**Interní úložiště**

**Externí úložiště**

**SQLite databáze**

**Cloudové úložiště**

## 6.6 SQLite

je třeba resit delku dat napríklad stringu?, dynamic typing

V knihovnách pro Forms aplikace se nachází další kód, který bude nutné přepsat do webové služby.

## **6.7 REST**

1. REST operace - davkove vs jednotlivé
2. REST, tabulka URI,

## **6.8 Synchronizace**

1. sync algoritmus - 2 algoritmy (jeden ideální, druhý reálný), srovnání
2. sync architektura - komponenty

## **6.9 Zabezpečení**

-authentikace Server webové služby je dostupná v síti VPN. Další zabezpečení bude řešeno později...

## **6.10 Zpětná kompatibilita**

## **6.11 Budoucí rozšiřitelnost**

## **6.12 Vytváření grafů**

knihovny, cloudové řešení, vlastní komponenty

## **6.13 Chybové reporty**

## 7 Testování

### 7.1 O čem psát...

1. popsat IMIS
2. připraveno webové služby na další mobilní platformy
3. činnost aplikace online/offline
4. flow diagramy pro různé činnosti
5. přístupová práva
6. úspora persistentní paměti na straně androida
7. chybové reporty a opravy na aplikaci v ostrem prostředí, obrázků + ukázka
8. jak zjistit změnu záznamu, v datech se ukládá pouze datum poslední změny, nikoli přesný čas
9. perioda automatického mazání dat
10. budoucnost formsu?
11. datový model schema

### 7.2 Zásady pro vypracování

1. Prozkoumejte systém IMIS pro evidenci docházky a pracovních výkazů. Vyberte činnosti, které by bylo vhodné implementovat i pro mobilní zařízení.
2. Navrhněte mobilní aplikaci pro platformu Android, které bude obsahovat vybrané funkce z předchozího bodu zadání. Zvažte aspekty zabezpečení komunikace aplikace se systémem.
3. Implementujte navržené řešení, berte přitom v úvahu možnou rozšiřitelnost o další funkce.

4. Ověřte funkcionální vytvořené aplikace.

