

VYSOKÁ ŠKOLA EKONOMICKÁ V PRAZE

FAKULTA INFORMATIKY A STATISTIKY

KATEDRA INFORMAČNÍCH TECHNOLOGIÍ

STUDIJNÍ PROGRAM: APLIKOVANÁ INFORMATIKA

OBOR: INFORMAČNÍ SYSTÉMY A TECHNOLOGIE

Prototyp aplikace pro ověření identity studenta u písemné zkoušky

DIPLOMOVÁ PRÁCE

Student: Bc. Ondřej Schrek
Vedoucí: Ing. Libor Gála, Ph.D.

2019

Prohlášení

Prohlašuji, že jsem diplomovou práci zpracoval samostatně a že jsem uvedl všechny použité prameny a literaturu, ze kterých jsem čerpal

V Praze dne 10. února 2019

.....
Bc. Ondřej Schrek

Poděkování

Text poděkování.

Abstrakt

Doplnit text.

Klíčová slova

Doplnit klíčová slova.

Abstract

Doplnit anj text.

Keywords

Doplnit anj ključová slova.

Obsah

Úvod	1
1 Charakteristika problému	2
1.1 Současný stav systému	2
1.2 Problémy stávajícího systému	3
1.3 Souhrn problémů a podmínek řešení	3
1.4 Návrh řešení	3
1.5 Cíl práce	3
Závěr	4

Úvod

V dnešní době se téměř každá instituce nějakým způsobem potýká s problémem efektivního identifikování osob. Vysoké školy jsou jednou z těchto institucí, kde se také potýkají s tímto problémem. Jelikož je zde potřeba často identifikovat větší skupiny studentů v co nejkratším čase. Příkladem je identifikace studentů u zkoušek, kdy zkoušející musí identifikovat každého studenta, zda-li je zapsán na zkoušku a také jestli se jedná skutečně o osobu daného studenta. Na Vysoké škole ekonomické v Praze je potřeba řešení pro efektivnější a spolehlivější způsob identifikace. Řešení které bude reflektovat současné technické možnosti a rozšiřovat systém, který je již na VŠE nastaven. Mezi další požadavky patří také to, aby řešení bylo možné snadno plošně nasadit při zachování nízkých nákladů.

Cílem práce je vytvořit prototyp aplikace, který umožní identifikovat studenty u zkoušky. Tento prototyp bude realizován v podobě mobilní aplikace pro zařízení se systémem Android. Implementace bude provedena v jazyce Kotlin. Aplikace bude umožňovat načítání informací z identifikačních karet používaných studenty. Aplikace bude propojená s informačním systémem školy, ze kterého bude získávat informace o zkouškách a jejich účastnících.

Součástí této práce je analýza současného stavu řešení systému identifikace a možností jeho zlepšení. Podle analýzou nalezených slabých míst a problémů bude navrhnut cílový stav systému identifikace. Poté budou zvoleny technologie pro implementaci a přístupy jakými lze dosáhnout cílového stavu systému. V souladu s vybraným přístupem bude navržena integrace prototypu s informačním systémem InSIS. Dále se vydefinují úkoly k implementaci a vytvoří se samotný postup implementace. Podle navrhovaného postupu bude popsána implementace jednotlivých úkolů. V další části bude navrženo a popsáno testování prototypu. Poslední část práce nastíní možnosti dalšího rozvoje aplikace.

Kapitola 1

Charakteristika problému

1.1 Současný stav systému

V současné době se na VŠE používá k identifikaci studentů u zkoušek kontrola proti vytištěným seznamům přihlášených studentů. Zkoušející musí procházet seznam a vyhledávat studenty dle jejich školních identifikačních karet, nebo dokladů totožnosti, kterými se na začátku zkoušky prokazují. Při zkouškách kde je nahlášeno mnoho studentů je tato činnost velmi zdlouhavá a vyžaduje po zkoušejícím přípravu navíc. Minimálně v podobě tisku aktuálních seznamů studentů těsně před zkouškou. Současný systém umožňuje identifikaci studenta pomocí načtení karty při vstupu do místnosti, avšak toto není dostatečně přesný způsob pro identifikaci konkrétního studenta z důvodu nemožnosti ztotožnit zda-li se skutečně jedná o danou osobu, která se prokázala identifikační kartou. Tento způsob se na VŠE nově používá pro automatickou docházku na výuku, kde je tento způsob identifikace dostatečný. Další nevýhodou je, že se tento systém dá využít pouze v učebnách, kde je přístup řízen pomocí karet. Na VŠE existují učebny, které tento přístup nemají a v těchto učebnách by ani nebylo možné tuto metodu identifikace použít.

Studentům jsou v současné době poskytovány dva typy bezkontaktní identifikačních karet, interní karta školy a mezinárodní karta ISIC. Obě karty jsou v rámci školního užívání rovnocenné. Obě tyto karty využívají technologii RFID-NFC.

1.2 Problémy stávajícího systému

Současný systém je neflexibilní a nabízí příležitosti pro jeho automatizaci. Současné řešení neodpovídá technickým možnostem, které jsou již na VŠE zavedeny např. elektronické identifikační karty. Další nevýhodou je, že zkoušející musí zadávat prezenci do systému InSIS manuálně.

1.3 Souhrn problémů a podmínek řešení

1.4 Návrh řešení

1.5 Cíl práce

Závěr

Text závěru.

Literatura

```
1  import javax.swing.JApplet;
2  import java.awt.Graphics;
3
4  public class Hello extends JApplet {
5      public void paintComponent(Graphics g) {
6          g.drawString("Hello, world!", 65, 95);
7      }
8  }
```

```
1
2
3
4
5  package com.example.admin.androidactivity;
6  import android.os.Bundle;
7  import android.support.annotation.Nullable;
8  import android.support.design.widget.FloatingActionButton;
9  import android.support.design.widget.Snackbar;
10 import android.support.v7.app.AppCompatActivity;
11 import android.support.v7.widget.Toolbar;
12 import android.view.View;
13 import android.view.Menu;
14 import android.view.MenuItem;
15
16 public class ActivityExample extends AppCompatActivity {
17     @Override
18     protected void onStart() {
19         super.onStart();
20     }
21
22     @Override
23     protected void onCreate(@Nullable Bundle savedInstanceState) {
```

```

24         super.onCreate(savedInstanceState);
25     }
26
27     @Override
28     protected void onStop() {
29         super.onStop();
30     }
31
32     @Override
33     protected void onResume() {
34         super.onResume();
35     }
36
37     @Override
38     protected void onPause() {
39         super.onPause();
40     }
41
42     @Override
43     protected void onDestroy() {
44         super.onDestroy();
45     }
46 }

```

Listing 1.1: Simple code listing.

```

1     /* Block comment */
2     package hello
3     import kotlin.collections.* // line comment
4
5     /**
6      * Doc comment here for 'SomeClass'
7      * @see Iterator#next()
8      */

```

```

9      @Deprecated("Deprecated class")
10     private class MyClass<out T : Iterable<T>>(var prop1 : Int) {
11         fun foo(nullable : String?, r : Runnable, f : () -> Int,
12             fl : FunctionLike, dyn: dynamic) {
13             println("length\nis ${nullable?.length} \e")
14             val ints = java.util.ArrayList<Int?>(2)
15             ints[0] = 102 + f() + fl()
16             val myFun = { -> "" };
17             var ref = ints.size
18             ints.lastIndex + globalCounter
19             ints.forEach lit@ {
20                 if (it == null) return@lit
21                 println(it + ref)
22             }
23             dyn.dynamicCall()
24             dyn.dynamicProp = 5
25         }
26
27         val test = """hello
28                     world
29                     kotlin"""
30         override fun hashCode(): Int {
31             return super.hashCode() * 31
32         }
33     }
34     fun Int?.bar() {
35         if (this != null) {
36             println(message = toString())
37         }
38         else {
39             println(this.toString())
40         }
41     }
42     var globalCounter : Int = 5

```

```
43         get = field
44     abstract class Abstract {
45     }
46     object Obj
47     enum class E { A, B }
48     interface FunctionLike {
49         operator fun invoke() = 1
50     }
```
