

# MOBILNÍ APLIKACE PRO MONITOROVÁNÍ PRŮMYSLOVÝCH PROCESŮ (INDUSTRIAL PROCESSES MONITORING APP)

*Daniel Šebík<sup>1</sup>*

<sup>1</sup> Ústav přístrojové a řídicí techniky, fakulta strojní, ČVUT v Praze, [daniel.sebik@fs.cvut.cz](mailto:daniel.sebik@fs.cvut.cz)

*Abstrakt: V této práci je řešena problematika rychlé orientace na pracovišti s využitím mobilních SCADA systémů. Pro tento účel je navržena aplikace formátu SCADA, ve které je implementováno rozpoznání QR-kódů přiřazených jednotlivým částem sledovaného procesu. Aplikace je psána pro operační systémy iOS v jazyce Swift. Hlavními využitými frameworky jsou Core Image pro rozpoznání obrazu a Macaw pro implementaci vektorově řešené vizualizace procesu.*

*Klíčová slova: QR-kód, scanování, Swift, mobilní SCADA, SwiftUI, Core Image, AV Foundation*

## 1. Úvod

Monitorování průmyslových procesů je důležitou součástí potřebnou pro řízení komplexních průmyslových procesů. Tato oblast spadá převážně pod SCADA systémy, které mají za účel centralizovat monitorování a řízení těchto procesů. Za pomoci různých operátorských panelů a zobrazovacích zařízení potom může být monitorování dostupné i přímo na daném pracovišti. V některých případech tomu tak ale není. Ať už z důvodu finančních úspor, prostorových omezení nebo z jiných důvodů. Pro technika pohybujícím se na daném pracovišti potom může být v takových případech poněkud složité získávání informací o daném procesu. Tato myšlenka tak vytváří prostor pro návrh aplikace pro různá přenosná zařízení, která může mít pracovník na daném stanovišti stále k dispozici, a pomocí kterých by mohl jednoduše nascanovat identifikační kód dané části procesu, a tím ji identifikovat a získat potřebné informace. Jednalo by se tak o rozšíření mobilních SCADA systémů právě o rychlou identifikaci. Cílem tohoto projektu řešeného v rámci studentského projektu bylo seznámení se s programovacím jazykem Swift za pomoci tvorby mobilní aplikace řešící popsáný problém.

## 2. Koncept aplikace

Navržená aplikace je rozdělena do tří částí. V první části je vizualizace samotného procesu s interaktivními prvky, pomocí kterých může uživatel přímo přejít na informační stránku dané části procesu, například sebrat data z daného snímače. Obdobným způsobem již přímo SCADA systémy fungují, např. [4]. Ve druhé části si uživatel může navolit seznam několika preferovaných stanovišť, které může bez hledání ve vizualizaci, či scanování identifikačního kódu rychle otevřít. Ve třetí části je implementována inovativní složka. Každá část procesu, snímač či akční člen budou mít vlastní QR kód představující identifikační číslo daného prvku a uživatel jej bude moci přímo na pracovišti nascanovat, čímž se přímo dostane na stránku s

parametry daného prvku v procesu. Nebude tak muset ve vizualizaci hledat, který prvek je ten správný, čímž dojde ke zrychlení hledání. Navíc se zajistí i snížení chybovosti lidského faktoru, protože uživatel nebude moci ve vizualizaci omylem zvolit jiný prvek systému a pracovat se špatnými informacemi.

### 3. Tvorba UI

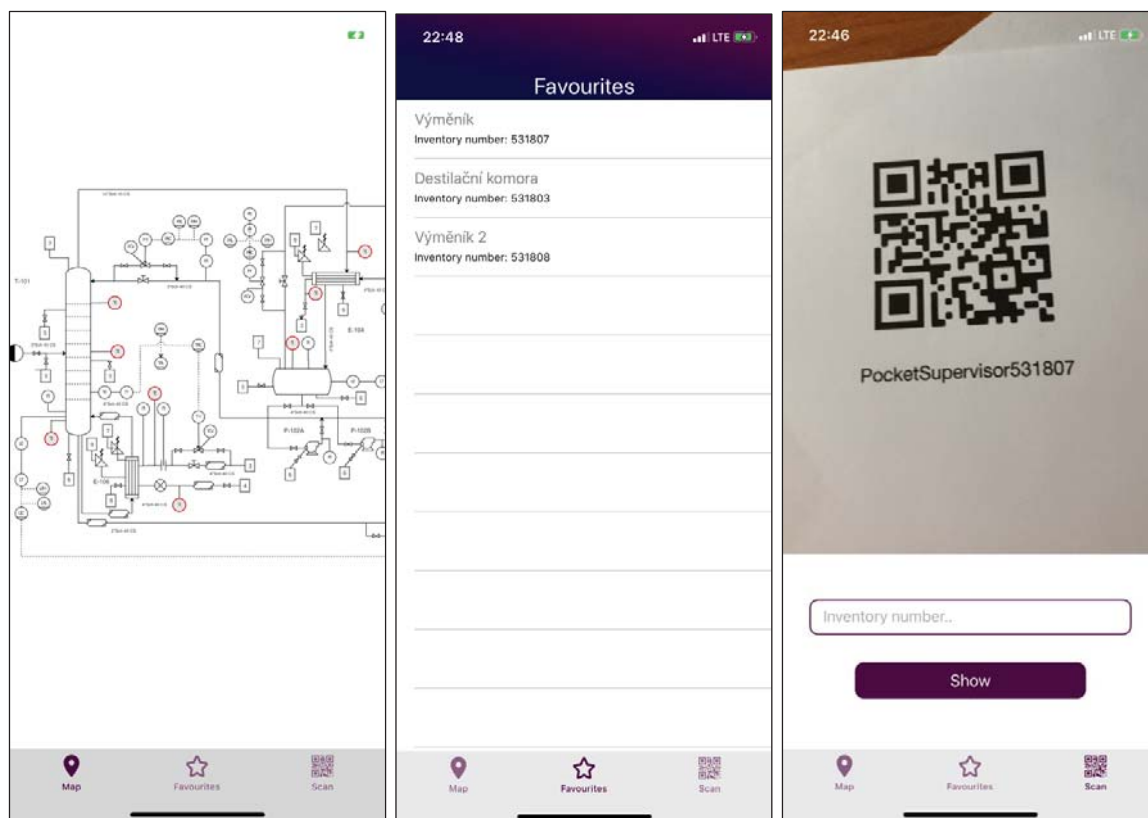
Tvorba uživatelského rozhraní se v nynější době u operačních systémů iOS přesouvá k novému frameworku SwiftUI, nicméně v době tvorby aplikace ještě nebyl tento framework k dispozici a využil jsem frameworku UIKit [2].

Po otevření aplikace se uživateli zobrazí přihlašovací obrazovka. Ta je vytvořena pomocí objektu třídy `UIViewController`. Po zadání správné kombinace přihlašovacího jména a hesla se následně zavře přihlašovací okno a skrze notifikaci objekt třídy `UISceneDelegate`, který spravuje události během životního cyklu aplikace, otevře nové okno. Pro rozdělení aplikace do tří částí je pro toto okno voleno objektu třídy `UITabBarController`, který spravuje tři rozdílné `UITabViewControllery` jednotlivých částí, mezi nimiž lze přepínat pomocí lišty na spodní straně obrazovky.

První z oken je zmiňovaná vizualizace procesu. Tu jsem řešil pomocí grafického frameworku `Macaw` [1], který mi dovolil využít vektorových formátů grafiky vizualizace ve formátu `SVG`. Zoomování a posouvání ve vizualizaci je řešeno pomocí transformací souřadnic a velikostí v daném vektorovém obrázku. Snímek obrazovky z okna vizualizace je vidět na obr. 1.a.

Druhé z oken je věnováno uloženým prvkům procesu, tak aby se uživatel rychleji dostal k částem procesu, které nejčastěji sleduje. Seznam je řešený pomocí objektu třídy `UITableView`, do kterého se načítají objekty uložených částí procesu podle identifikačních kódů, které jsou uloženy do paměti po přidání na seznam preferovaných objektů. Ukládání do paměti mobilního zařízení je řešeno pomocí frameworku `Core Data`. Okno s uloženými položkami je vidět na obr. 1.b.

V posledním okně `UITabViewControllera` může uživatel scanovat QR kód přiřazený určité části procesu. Scanování QR kódů je řešeno pomocí frameworků `Core Image` a `AV Foundation`. Okno s nástrojem scanování je vidět na obr. 1.c.



Obr. 1 Okna hlavního UITabViewControlleru zleva: a) vizualizace procesu, b) uložené prvky procesu, c) scanování QR kódů.

Posledním oknem (obr. 2) uživatelského rozhraní je okno samotného prvku v procesu, kde se zobrazují potřebné informace, a kde se nachází tlačítka pro přidání do preferovaných položek nebo také jeho zobrazení přímo ve vizualizaci. Data pro testování funkce byla brána z pražských meteostanic [3].



Obr. 2 Stránka s parametry zvoleného prvku procesu.

#### 4. Scanování QR kódů

Pro scanování identifikačních QR kódů jsem využil frameworků Core Image a AV Foundation, jak jsem již zmínil výše. První ze zmiňovaných frameworků je určený ke zpracování a analýze obrazu. V tomto případě jsem využil konkrétně objektu třídy CIDetector, který jsem využil k identifikaci QR kódu z dodaného snímku a jejich přečtení.

Framework AV Foundation potom slouží k přístupu k audiovizuálním prvkům mobilních zařízení, v tomto případě k zadnímu fotoaparátu. Po výběru konkrétního vstupu/výstupu je třeba toto zařízení nakonfigurovat podle požadovaných cílů (počet snímků za sekundu, kvalita snímků a podobně). Po nastavení požadovaného zařízení je možné spustit takzvanou CaptureSession, během které dochází k opakovanému snímání audiovizuálního signálu, v tomto případě tedy ke snímání obrázků. Snímané obrázky se potom za pomoci objektu třídy AVCaptureVideoPreviewLayer zobrazují přímo na obrazovce uživateli, tak aby mohl namířit na identifikovaný kód. Obrázek je po sejmutí předán také objektu třídy CIDetector, který identifikuje, zdali je přítomný nějaký QR-kód a případně přečte zapsaný identifikační kód.

#### 5. Závěr

Během projektu byla v jazyce Swift vytvořena aplikace s prvky vizualizace obdobné SCADA systému pro mobilní zařízení, rozšířená o možnost scanování identifikačního čísla jednotlivých částí průmyslových procesů za pomoci QR kódů pro rychlou orientaci na pracovišti.

#### Literatura

- [1] P.W.D. Charles, Project Title, (2013), GitHub repository, Dostupné z: <https://github.com/charlespwd/project-title>
- [2] STEELKIWI Inc., “SwiftUI vs UIKit: Benefits and Drawbacks”, 2020, *Medium*.

[3] “Meteostanice”, *Opendata.praha.eu*, Dostupné z: <https://opendata.praha.eu/en/dataset/tsk-meteostanice>

[4] TRIMAX SYSTEMS, “Mobile SCADA”, *Trimaxsystems.com*, Dostupné z: <http://www.trimaxsystems.com/solutions/mobile-scada.html>



**Selected article from**

**Tento dokument byl publikován ve sborníku**

**Nové metody a postupy v oblasti přístrojové  
techniky, automatického řízení a informatiky 2020  
New Methods and Practices in the Instrumentation,  
Automatic Control and Informatics 2020**

**14. 9. – 16. 9. 2020, Zámek Lobeč**

**ISBN 978-80-01-06776-5**

Web page of the original document:

<http://iat.fs.cvut.cz/nmp/2020.pdf>

Obsah čísla/individual articles:

<http://iat.fs.cvut.cz/nmp/2020/>

Ústav přístrojové a řídicí techniky, FS ČVUT v Praze, Technická 4, Praha 6