



Vyšší odborná škola
a Střední průmyslová škola elektrotechnická
Plzeň, Koterovská 85

DLOUHODOBÁ MATURITNÍ PRÁCE S OBHAJOBOU

Téma:

System chytré ledničky "Easy Freezy"

Autor práce: Martin Robb
Třída: 4.L
Vedoucí práce: Ing. Pavel Jedlička
Dne: 1.1.2030
Hodnocení:

ZDE VLOŽIT LIST ZADÁNÍ

Z důvodu správného číslování stránek

Anotace a poděkování

Maturitní práce se zabývá vývojem mobilní aplikace pro Android, která organizuje domácí zásoby potravin. Aplikace skenuje čárový kód a datum spotřeby produktu. Dále najde informace produktu pomocí naskenovaných dat v externí databázi potravin (Open Food Facts). Získané informace následně ukládá do databáze. Aplikace umožňuje uživateli tuto databázi upravovat.

Prohlašuji, že jsem tuto práci vypracoval samostatně a použil literárních pramenů a informací, které cituji a uvádím v seznamu použité literatury a zdrojů informací.

Prohlašuji, že jsem nástroje UI využil v souladu s principy akademické integrity a že na využití těchto nástrojů v práci vhodným způsobem odkazuji.

Souhlasím s využitím mé práce učiteli VOŠ a SPŠE Plzeň k výuce.

V Plzni dne 1. 1. 2030

Podpis:

Obsah

Úvod	5
1 Získávání dat o produktech	6
1.1 Kalendář	7
1.2 Čtení čárového kódu	11
2 Databáze	12
2.1 Uživatelská databáze	12
2.2 Komunikace s databází	12
3 Aplikace na Android	13
3.1 Hlavní aplikace	13
3.1.1 Seznam použitých knihoven, frameworků a nástrojů	14
Závěr	15

Úvod

Motivace

Aplikací, které napodobují chytrou lednici není mnoho. Zároveň také popularita chytrých ledniček rok od roku vzrůstá. V této práci jsem se chtěl pokusit o napodobení těchto chytrých lednic bez toho aby uživatel musel tyto drahé a prostor zabírající lednice kupovat. Pokusím se hlavně napodobit databázový systém produktů a expiračních dat, práce a upravování dat v databázi z uživatelského rozhraní (dále jen UI) i samotné zobrazování dat produktů z databáze.

Výběr programovacího jazyku

Pro realizaci aplikace jsem se rozhodl využít programovací jazyk Python [**python**]. Hlavním důvodem tohoto rozhodnutí je skutečnost, že Python je jediným jazykem který znám. Jako ideální jazyky pro vývoj aplikací pro Android je často uváděn jazyk Java nebo Kotlin. Python sice uváděn není pro vývoj Android aplikací ho ale použít lze. Musíme však využít vhodný framework, který se postará o kompatibilitu s Androidem.

Kivy

Kivy je vývojová platforma pro jazyk Python, která se je využívána pro vývoj aplikací na Android, IOS ale i Windows. Existuje ještě alternativní vývojová platforma BeeWare, která je ještě v rané fázi vývoje a proto také méně používaná. Kivy je proto momentálně nejlepší volbou pro vývoj aplikace na Android pomocí Pythonu. Dále také používám KivyMD. KivyMD je rozšíření frameworku Kivy, které usnadňuje úpravy a vytváření grafického rozhraní aplikace.

1 Získávání dat o produktech

Je nutné uvědomit si jaké data o produktech je potřeba. Je potřeba datum expirace a čárový kód produktu. Čárový kód umožňuje dohledání dalších informací v externí databázi jako například jméno, výrobce nebo také ale počet kalorií na 100g atd. Já využívám veřejně dostupnou databázi OpenFoodFacts [**openfoodfacts**]. Pro mé účely stačí jen jméno produktu. Nejdříve jsem plánoval stáhnout obsah celé přístupné databáze. Naskytla se ale mnohem lepší varianta. OpenFoodFacts podporují získávání dat přímo z jejich webových stránek. Stačí tedy zadat určitou URL adresu, která obsahuje čárový kód daného produktu, do vyhledávání a jako odpověď získáme JSON soubor s informacemi o daném produktu viz Listing 1.1.

Listing 1.1: Ukázka získání JSON souboru produktu

```
url = f"https://world.openfoodfacts.org/api/v2/product/{
    carovy_kod}.json"
odpoved = requests.get(url)
odpoved_json = odpoved.json()
produkt_api_data = api_json_file.get("product", "Unknown")
}
```

Na začátku vloženého ukázkového programu (viz Listing 1.1 1.1) lze vidět string (posloupnost textových charakterů) pojmenovaný "url". Do tohoto stringu je vložen čárový kód mezi složené závorky. Dále je využita knihovna Requests [**requests**], která umožňuje vyhledávání, získání odpovědi serveru a následné převedení (přeparsování) na formát JSON. Do proměnné "response" se uloží odpověď serveru na poslanou URL a následně se ukládá přeparsovaná verze odpovědi do proměnné "odpoved_json". V posledním řádku se z této odpovědi získávají všechny informace o produktu a ukládají do proměnné "produkt_api_data". Jednotlivé informace se dále získávají z proměnné "produkt_api_data" stejným způsobem.

1.1 Kalendář

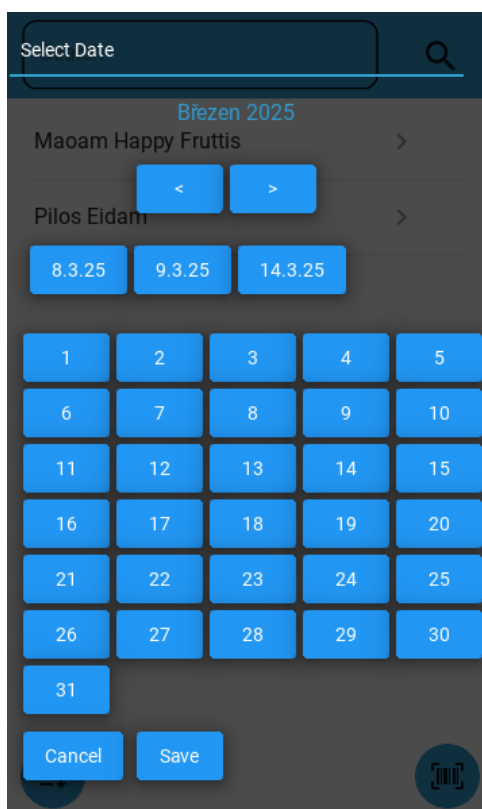
Pro získání data expirace produktu jsem se rozhodl použít kalendář. Uvažoval jsem, že bych mohl využít kamery zařízení pro naskenování data expirace. Kvůli tečkovanému tisku dat expirace a lesklému obalu byla ale často nepřesná a tak jsem se od tohoto nápadu odklonil. Rozšíření KivyMD [**kivymd**] již v sobě jistý kalendář (anglicky datepicker) pro výběr dat obsahuje, bohužel ale nepodporuje některé funkce, které chci aby kalendář měl jako například vybrání více dat spotřeby. Rozhodl jsem se tedy zkonstruovat vlastní.

Kalendář je rozdělen na několik sekcí. Každá z těchto sekcí má jinou náležitou funkci. Sekce popisují podle obrázku 1.1. Nejdříve je 1. vrstva popisek, který zobrazuje rok a měsíc, ve kterém vybíráme datum. Pod touto sekcí jsou tlačítka, které slouží pro změnu měsíce. V další sekci každé tlačítko představuje jednotlivé dny v měsíci. V poslední sekci jsou umístěna tlačítka pro uložení nebo pro zrušení výběru a následné ukončení aplikace. Výsledný kalendář je zachycen na Obrázku 1.2.

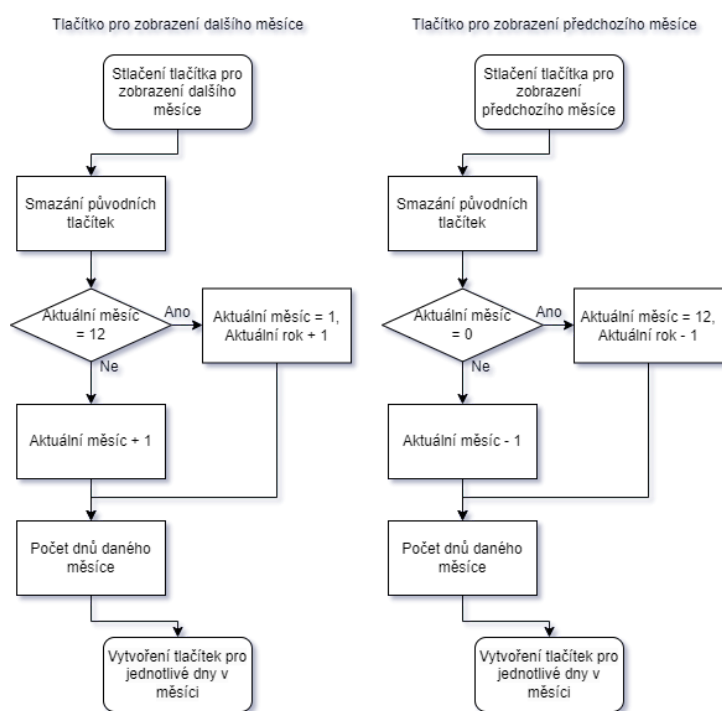
Hlavní část kalendáře jsou tlačítka pro výběr jednotlivých dnů v měsíci (na Obrázku 1.1 označeno číslem 4). Počet těchto tlačítek se musí přizpůsobit podle toho v jakém měsíci uživatel právě vybírá. Proto kalendář vždy musí smazat předchozí a vygenerovat nové tlačítka když uživatel stiskne jedno ze dvou tlačítek pro změnu měsíce (na Obrázku 1.1 označeno číslem 2). Funkce tlačítek je znázorněna v podobě diagramu na obrázku 1.3.



Obrázek 1.1: Kalendář rozdělený na jednotlivé sekce



Obrázek 1.2: Kalendář



Obrázek 1.3: Diagram změny měsíce

1.2 Čtení čárového kódu

2 Databáze

2.1 Uživatelská databáze

2.2 Komunikace s databází

```
class Content(MDBoxLayout):
    def __init__(self, items, id, product_container, **kwargs
    ):
        #zdedeni parametru materske tridy
        super().__init__(**kwargs)

        self.orientation = "vertical"
        self.size_hint_y = None
        self.id = id
        self.product_container = product_container

        # prizpusobeni vysky podle poctu dat expirace
        self.dates_array = items
        self.items = len(items)
        self.height = dp(35 * self.items) #old dp 48

}
```

3 Aplikace na Android

3.1 Hlavní aplikace

```
class Content(MDBoxLayout):
    def __init__(self, items, id, product_container, **kwargs
    ):
        #zdedeni parametru materske tridy
        super().__init__(**kwargs)

        self.orientation = "vertical"
        self.size_hint_y = None
        self.id = id
        self.product_container = product_container

        # prizpusobeni vysky podle poctu dat expirace
        self.dates_array = items
        self.items = len(items)
        self.height = dp(35 * self.items) #old dp 48

}
```

3.1.1 Seznam použitých knihoven, frameworků a nástrojů

- OpenCV (CV2) [**opencv**]
- Ubuntu [**ubuntu**]
- KivyMD [**kivymd**]
- MySQL connector [**mysql-connector**]
- Collections [**python-collections**]
- Datetime [**python-datetime**]
- Buildozer [**buildozer**]
- Python [**python**]
- Requests [**requests**]
- Kivy [**kivy**]

Závěr

Tolik k rychlému úvodu a ukázce. Tento template by měl být rovnou použitelný pro vaši práci. Pouze upravíte titulní stránku a text v práci. Níže je ještě použitá literatura. Pokud byste měli nějaké dotazy, zkuste google (latex je velice dobře komunitně podporován a najdete odpovědi na všechny problémy online). Případně se obraťte na autora dokumentu.