

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

SEMESTRÁLNÍ PRÁCE z UZI

Večeře

1. Formulace úlohy (zadání)	3
2. Analýza úlohy	4
3. Popis algoritmu řešení	5
4. Popis programu (programová dokumentace)	7
5. Popis obsluhy programu (uživatelská dokumentace)	8
6. Rozbor výsledků, zhodnocení	9
7. Závěr	10

1. Formulace úlohy (zadání)

Původní formulace:

Dnes večer přijdu domů a budu mít chuť na Kung-pao. Znalostní systém mi potvrdí, zda je to možné (na základě stavu mé ledničky a spíže – určete). Na rozdíl od mechanického porovnání seznamu obsahu ledničky s receptem, znalostní systém mi potvrdí, že je to možné i v případě, že bude chybět nějaká nepodstatná záležitost, nebo že bude možné nahradit požadovanou surovinu něčím jiným, event. poradí, kde je možné chybějící surovinu obstarat (která prodejna v blízkosti má tuto v nabídce a má ještě otevřeno).

Cíl práce:

Vyvinout znalostní systém na vaření, který umožňuje uživatelům efektivně spravovat inventář ingrediencí, prohlížet a vyhledávat recepty, získávat doporučení obchodů pro nákup chybějících surovin a zobrazovat tutoriály pro přípravu pokrmů. Systém by měl poskytovat uživatelsky přívětivé rozhraní pro interakci s uživatelem a efektivně využívat dostupné zásoby k doporučení receptů.

Výchozí podmínky a vstupní data:

- Uživatel má digitální inventář různých ingrediencí, rozdělených do skladovacích prostor (lednice, mrazák, spíže, koření, nápoje).
- K dispozici je báze dat receptů, každý recept obsahuje seznam potřebných ingrediencí s množstvím a jednotkami, případně substitucemi.
- Informace o dostupnosti ingrediencí v různých obchodech, jejich cenách, vzdálenosti a otevírací době.
- Uživatelské rozhraní bude konzolové s barevným výstupem pro lepší přehlednost.

Požadavky na výstupy:

- Doporučení receptů na základě aktuálního inventáře.
- Zobrazení chybějících ingrediencí s možností doporučení substitucí.
- Doporučení obchodů pro nákup chybějících surovin s ohledem na cenu, vzdálenost a otevírací dobu.
- Zobrazení kroků přípravy receptu ve formě tutoriálu.
- Aktualizace inventáře po nákupu a přípravě receptu.

2. Analýza úlohy

Možné postupy řešení:

1. **Manuální správa inventáře a receptů:** Uživatel sám spravuje data bez podpory systému.
2. **Automatizovaný systém s databází:** Využití báze dat pro správu receptů a inventáře s možností vyhledávání a doporučení.
3. **Integrovaný systém s inferenčním enginem:** Využití logiky pro rozhodování o doporučeních na základě zásob a dostupnosti surovin v obchodech.

Alternativy řešení:

- **Webová aplikace vs. konzolová aplikace:** Webová aplikace by poskytla grafické uživatelské rozhraní, ale vyžadovala by složitější implementaci.
- **Lokální vs. cloudová databáze:** Lokální báze dat by byla jednodušší na nasazení, cloudová by poskytovala lepší škálovatelnost.

Výběr nejvhodnější alternativy:

Zvolena konzolová aplikace s lokální databází kvůli jednoduchosti implementace a snadné správy. Kombinace modulů pro inventář, recepty, doporučení obchodů a uživatelské rozhraní umožňuje efektivní správu a interakci.

Zdůvodnění:

Konzolová aplikace je vhodná pro rychlé prototypování a jednoduché použití bez potřeby složitější infrastruktury. Modularita umožňuje snadnou údržbu a rozšiřování funkcionality.

3. Popis algoritmu řešení

Formulace algoritmu:

Algoritmus řešení se skládá z následujících kroků:

1. **Inicializace systému:**
 - Načtení dat o zásobách z různých skladovacích prostor.
 - Načtení receptů z databáze.
 - Načtení informací o obchodech a jejich sortimentu.
2. **Interakce s uživatelem:**
 - Zobrazení hlavního menu s možnostmi:
 - Výběr receptu k přípravě.
 - Zobrazení inventáře.
 - Dokoupení ingrediencí.
 - Konec programu.
3. **Doporučení receptů:**
 - Po výběru receptu systém kontroluje dostupnost ingrediencí.
 - V případě nedostatku ingrediencí nabízí substituce nebo doporučení obchodů pro nákup.
4. **Správa zásob:**
 - Po úspěšném nákupu nebo přípravě receptu se aktualizuje inventář.
 - Odečítají se použité ingredience.
5. **Zobrazení tutoriálu:**
 - Při přípravě receptu se zobrazí krok za krokem návod na přípravu pokrmu.

Dekompozice na dílčí kroky (moduly):

- **Modul inventáře:** Správa a kontrola dostupnosti ingrediencí.
- **Modul receptů:** Správa receptů a jejich ingrediencí.
- **Modul doporučování obchodů:** Doporučení nejvhodnějších obchodů pro nákup.
- **Modul uživatelského rozhraní:** Interakce s uživatelem a zobrazení informací.
- **Modul tutoriálů:** Zobrazení kroků přípravy receptu.
- **Modul inferenčního enginu:** Kontrola dostupnosti ingrediencí a rozhodování o možnosti přípravy receptu.

Schématické znázornění návazností:

1. Uživatelské rozhraní → Výběr receptu
2. Inference Engine → Kontrola dostupnosti → Výsledek (dostupné nebo nedostupné)
3. Pokud nedostupné → Doporučení substitucí nebo obchodů
4. Po nákupu → Aktualizace inventáře
5. Zobrazení tutoriálu → Příprava receptu

4. Popis programu (programová dokumentace)

Volba programovacího jazyka a technologie:

Pro implementaci systému byl zvolen programovací jazyk Python díky jeho jednoduchosti, bohaté knihovně modulů a snadné manipulaci s datovými strukturami. Python umožňuje rychlé prototypování a podporuje objektově orientované programování, což usnadňuje modularitu a rozšiřitelnost systému.

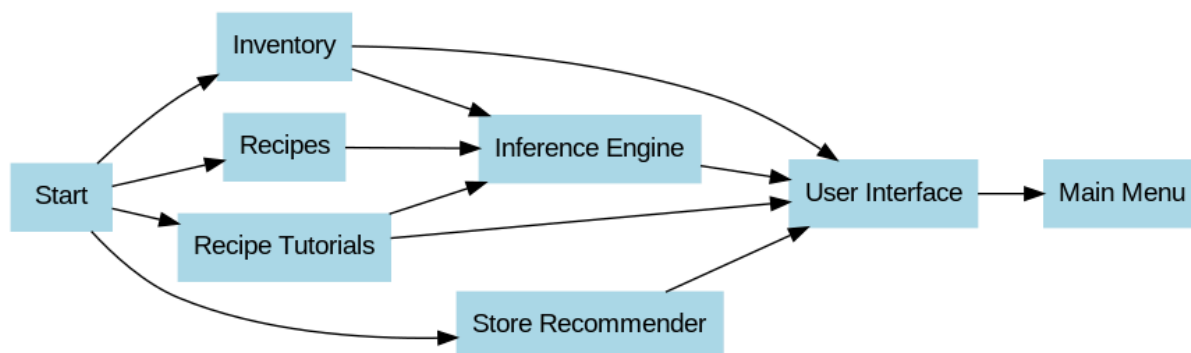
Volba implementujících datových typů a datových struktur:

- **Slovníky (dict):** Pro uložení inventáře, receptů a informací o obchodech. Slovníky poskytují rychlý přístup k datům podle klíče, což je efektivní pro vyhledávání a aktualizace.
- **Seznamy (list):** Pro uchovávání seznamů ingrediencí a kroků receptů. Seznamy jsou vhodné pro pořadí kroků a iteraci přes prvky.
- **Třídy (class):** Pro modelování modulů systému (Inventory, Recipes, StoreRecommender, UserInterface, RecipeTutorials, InferenceEngine). Třídy umožňují objektově orientovaný přístup, který zvyšuje srozumitelnost a organizaci kódu.

Blokové schéma programových kroků:

- **Main modul (riedl_main.py):** Inicializace a spuštění uživatelského rozhraní.
- **UserInterface:** Interakce s uživatelem, výběr akcí, zobrazení menu.
- **Inventory:** Správa zásob, převody jednotek, kontrola dostupnosti.
- **Recipes:** Správa receptů, vyhledávání receptů podle názvu.
- **StoreRecommender:** Doporučování obchodů na základě dostupnosti ingrediencí a aktuální doby.
- **RecipeTutorials:** Zobrazení kroků přípravy receptů.
- **InferenceEngine:** Kontrola dostupnosti ingrediencí a rozhodování o možnosti přípravy receptu.

Vývojový diagram inicializace programových modulů:



Popis významu použitých symbolů:

- **Konstanty:** Definovány v třídě Inventory (např. UNIT_CONVERSIONS) pro převod jednotek.
- **Proměnné:** Instance tříd jako inventory, recipes, recommender uchovávají stav systému a slouží k interakci mezi moduly.
- **Funkce/metody:** Implementují logiku pro interakci s uživatelem, správu zásob, vyhledávání receptů a doporučení obchodů.

Výčet specifických vlastností programových modulů:

- **Modul Inventory:**
 - Převod jednotek s podporou více jednotek (g, kg, ml, l, ks).
 - Komplexní správa zásob s možností aktualizace a odečítání ingrediencí.
 - Podpora substitucí ingrediencí při nedostatku.
- **Modul Recipes:**
 - Centralizace receptů do slovníku pro rychlé vyhledávání.
 - Snadné rozšiřování o nové recepty přidáním do databáze.
- **Modul StoreRecommender:**
 - Dynamické doporučování obchodů na základě aktuální doby a dostupnosti surovin.
 - Informace o cenách, vzdálenosti a otevírací době obchodů.
- **Modul UserInterface:**
 - Barevný výstup pro lepší přehlednost a čitelnost.
 - Intuitivní konzolové menu pro snadnou interakci s uživatelem.
- **Modul RecipeTutorials:**
 - Formátované zobrazení kroků receptů jako tutoriálů.
 - Přehledné a snadno čitelné návody pro přípravu pokrmů.
- **Modul InferenceEngine:**
 - Logika pro kontrolu dostupnosti ingrediencí a rozhodování o možnosti přípravy receptu.

- Podpora substitucí pro nepodstatné ingredience.

5. Popis obsluhy programu (uživatelská dokumentace)

Instalace:

1. Požadavky:

- Python 3.x
- Standardní knihovny Pythonu (žádné externí knihovny nejsou vyžadovány).

2. Instalace:

- Stáhněte si zdrojový kód projektu ve formátu .zip a rozbalte jej do požadovaného adresáře.
- Ujistěte se, že máte nainstalovaný Python 3.x. Pokud ne, stáhněte a nainstalujte jej z python.org.

Spuštění programu:

1. Otevřete terminál nebo příkazový řádek.
2. Přejděte do adresáře s rozbaleným projektem.

Spustíte hlavní spustitelný soubor příkazem:

```
python riedl_main.py
```

3.

Obsluha programu:

1. Hlavní menu:

- **Výběr receptu:** Zobrazí seznam dostupných receptů. Zadejte číslo receptu, který chcete připravit.
- **Zobrazení inventáře:** Zobrazí aktuální zásoby ingrediencí v jednotlivých skladovacích prostorech.
- **Dokoupení ingredience (nákup):** Umožňuje dokoupit chybějící ingredienci manuálně. Program doporučí obchody, kde je možné ingredienci zakoupit.
- **Konec:** Ukončí program.

2. Příprava receptu:

- Po výběru receptu program zkontroluje dostupnost ingrediencí.
- Pokud některé ingredience chybí, nabídne možnosti substituce nebo doporučení obchodů pro nákup.
- Po úspěšném nákupu nebo použití substitucí program zobrazí tutoriál krok za krokem.

3. Dokoupení ingrediencí:

- Vyberte možnost 'Dokoupení ingredience'.
- Zadejte název ingredience a množství, které chcete dokoupit.

- Program doporučí obchody, kde je možné ingredienci zakoupit.
- Vyberte obchod a dokončete nákup.

Vstupní data:

- **Recepty:** Název receptu jako řetězec (str).
- **Ingredience:** Název ingredience a množství s jednotkou (např. 500 g).
- **Obchod:** Výběr obchodu z doporučeného seznamu.

Výstupní data:

- **Doporučené recepty:** Seznam receptů vhodných na základě dostupných surovin.
- **Inventář:** Tabulka s aktuálními zásobami ingrediencí.
- **Tutoriál:** Krok za krokem návod na přípravu zvoleného receptu.
- **Doporučení obchodů:** Seznam obchodů s informacemi o ceně, vzdálenosti a dostupnosti.

6. Rozbor výsledků, zhodnocení

Dosažené výsledky:

- Úspěšně byl vytvořen modulární znalostní systém na vaření s plnou funkcionalitou pro správu zásob, receptů, doporučení obchodů a zobrazení tutoriálů.
- Systém efektivně kontroluje dostupnost ingrediencí, nabízí substituce a doporučuje obchody na základě aktuální doby a dostupnosti surovin.
- Uživatelské rozhraní je intuitivní a poskytuje jasné a barevné výstupy pro lepší přehlednost.

Vyzdvihnutí kladů:

- **Modularita systému:** Umožňuje snadnou údržbu a rozšiřování funkcionality bez zásahu do hlavního kódu.
- **Objektově orientovaný přístup:** Zvyšuje srozumitelnost a organizaci kódu, což usnadňuje jeho údržbu a rozvoj.
- **Podpora substitucí:** Zajišťuje flexibilitu při nedostatku ingrediencí a umožňuje uživatelům připravit pokrm i s omezeným inventářem.

Kritika nedostatků:

- **Konzolové uživatelské rozhraní:** Může být méně přívětivé pro některé uživatele ve srovnání s grafickými aplikacemi.
- **Omezené chybové zprávy:** Momentálně poskytuje základní zpětnou vazbu, která by mohla být rozšířena o podrobnější informace a návrhy na opravu chyb.

- **Ručně spravovaná databáze:** Automatizace aktualizace dat by mohla zvýšit efektivitu a snížit riziko chyb.

7. Závěr

Zhodnocení splnění zadání:

Projekt splnil všechny hlavní požadavky zadání tím, že vytvořil plně funkční znalostní systém na vaření. Systém umožňuje efektivní správu zásob, vyhledávání a doporučování receptů, nákup chybějících ingrediencí a poskytuje uživatelsky přívětivé rozhraní s barevným výstupem. Modularita a objektově orientovaný přístup zajišťují snadnou údržbu a rozšiřitelnost systému.

Možnosti dalšího rozvoje:

- **Grafické uživatelské rozhraní:** Přechod na GUI by zlepšil uživatelskou zkušenost a přístupnost systému.
- **Integrace s online databázemi:** Automatické aktualizace receptů a zásob by zvýšily efektivitu správy dat.
- **Plánování jídelníčku:** Přidání funkce pro plánování jídel a sledování nutričních hodnot by rozšířilo využitelnost systému.
- **Historie nákupů:** Umožnění sledování historie nákupů a receptů by poskytlo uživatelům užitečné informace pro budoucí vaření.