

Vývoj systému chytré ledničky s funkcemi pro evidenci produktů a návrh receptů

Martin Mahdal

Bakalářská práce
2025



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
Ústav automatizace a řídicí techniky

Akademický rok: 2024/2025

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení:	Martin Mahdal
Osobní číslo:	A22195
Studijní program:	B0714A150006 Aplikovaná informatika v průmyslové automatizaci
Specializace:	Inteligentní systémy s roboty
Forma studia:	Prezenční
Téma práce:	Vývoj systému chytré ledničky s funkcemi pro evidenci produktů a návrh receptů
Téma práce anglicky:	Development of a Smart Refrigerator System with Product Inventory and Recipe Suggestions

Zásady pro vypracování

- Prostudujte technologie pro skenování čárových kódů a rozpoznávání produktů.
- Navrhněte systém správy potravin v ledničce s funkcemi pro sledování doby spotřeby.
- Implementujte kamerový systém pro vizuální identifikaci potravin.
- Vytvořte funkci pro upozornění na blížící se expiraci produktů.
- Navrhněte a implementujte algoritmus pro návrh receptů na základě dostupných zásob.
- Otestujte systém a zhodnotte jeho přínos pro správu potravin.

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam doporučené literatury:

1. PECINOVSKÝ, Rudolf. *Python: Kompletní příručka jazyka pro verzi 3.11*. Grada, 2022. ISBN 978-80-271-3891-3. Dostupné také z: <https://www.bookport.cz/AccountSaml/SignIn/?idp=https://shibboleth.utb.cz/idp/shibboleth&returnUrl=/kniha/python-11285/>
2. HOWSE, Joseph. *OpenCV Computer Vision with Python*. Packt Publishing, 2013. ISBN 1299545009. Dostupné také z: <https://proxy.k.utb.cz/login?url=https://app.knovel.com/hotlink/toc/id:kpOCVCVPL1/opencv-computer-vision?kpromoter=marc>
3. DONAT, Wolfram. *Learn Raspberry Pi Programming with Python: Learn to Program on the World's Most Popular Tiny Computer*. Second edition. *Technology in action*. [Berkeley, CA]: Apress, [2018]. ISBN 9781484237694. Dostupné z: <https://doi.org/9781484237694>.
4. ČÁDA, David. *Využití 2D kódů v běžné praxi*. 2016. Dostupné také z: <http://hdl.handle.net/10563/38447>.
5. WELLING, Luke a THOMSON, Laura. *MySQL: průvodce základy databázového systému*. Brno: CP Books, 2005. ISBN 8025106713.

Vedoucí bakalářské práce: **Ing. Jakub Husár, Ph.D.**
Ústav automatizace a řídicí techniky

Datum zadání bakalářské práce: **13. prosince 2024**
Termín odevzdání bakalářské práce: **27. května 2025**

doc. Ing. Jiří Vojtěšek, Ph.D. v.r.
děkan



prof. Ing. Vladimír Vašek, CSc. v.r.
ředitel ústavu

Ve Zlíně dne 13. prosince 2024

PROHLÁŠENÍ AUTORA BAKALÁŘSKÉ PRÁCE

Beru na vědomí, že

- odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb., v platném znění bez ohledu na výsledek obhajoby;
- bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému a bude dostupná k nahlédnutí;
- jedno vyhotovení bakalářské práce v listinné podobě bude ponecháno Univerzitě Tomáše Bati ve Zlíně k uložení;
- na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- podle § 60 odst. 1 autorského zákona má Univerzita Tomáše Bati ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- podle § 60 odst. 2 a 3 mohu užít své dílo – bakalářskou práci – nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tj. k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá; neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji, že

- jsem na bakalářské práci pracoval(a) samostatně a použitou literaturu jsem řádně citoval(a); v případě publikace výsledků budu uveden(a) jako spoluautor;
- odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou obsahově totožné.

Ve Zlíně, dne

.....
podpis autora

ABSTRAKT

Tato bakalářská práce se zabývá vývojem systému pro evidenci produktů v lednici pomocí skenování čárových kódů. Cílem systému je rovněž upozorňovat na blížící se data expirace pomocí emailu a také vizuálně. Systém je složen z jednoduchých komponentů, jako jsou display, Raspberry Pi, kamera a ventilátor – vše umístěno v tištěném krytu. Systém poskytuje možnost zjednodušení skladování produktů v domácnosti a snížení plýtvání potravinami.

Klíčová slova: Databáze, Evidence produktů, Chytrá Lednice

ABSTRACT

This bachelor's thesis aims to develop a system for recording products in the refrigerator using barcode scanning. The system also aims to notify about approaching expiration dates via email and also visually. The design is composed of simple components such as a display, Raspberry Pi, camera and a fan – all placed inside a printed cover. The system allows the option to simplify the storage of products in the household and reduce food waste.

Keywords: Database, Recordings of items, Smart Fridge

Chtěl bych poděkovat škole za umožnění realizace této bakalářské práce a také mému vedoucímu za zachování chladné hlavy a rady při situacích, kdykoliv jsem nevěděl a za jeho skvělý přístup a ochotu. Chtěl bych také poděkovat mé snoubence za podporu a prostor při realizaci této práce a pomoc v těžkých situacích.

OBSAH

ÚVOD.....	12
I TEORETICKÁ ČÁST	13
1 STUDIE TECHNOLOGIÍ PRO ČTENÍ ČÁROVÝCH KÓDŮ	14
1.1 ČÁROVÝ KÓD	14
1.2 ČTENÍ ČÁROVÝCH KÓDŮ	15
1.3 JINÉ TYPY 1D A 2-D KÓDŮ	15
2 EVIDENCE POTRAVIN.....	16
2.1 SQL DATABÁZE	16
2.1.1 MySQL.....	16
3 MIKROPOČÍTAČE	18
3.1 ZÁKLADNÍ INFORMACE O MIKROPOČÍTAČÍCH	18
3.1.1 Raspberry-Pi.....	19
3.2 POUŽITÍ MIKROPOČÍTAČŮ	20
4 OPERAČNÍ SYSTÉM	21
4.1 LINUX.....	21
4.2 UBUNTU	21
5 PROGRAMOVACÍ JAZYKY	22
5.1 HISTORIE JAZYKU PYTHON	22
5.2 POUŽITÍ PROGRAMOVACÍCH JAZYKŮ	22
5.3 JEDNODUCHÝ POPIS JAZYKA PYTHON.....	23
6 DISPLAY A KAMERA RPI	24
6.1 RASPBERRY PI TOUCH DISPLAY	24
6.2 ARDUCAM 2MPX IMX291	24
6.3 ENDOSKOP INSPEKČNÍ KAMERA USB-C 5M LED	26
7 CHYTRÉ LEDNICE.....	27
8 3D MODEL KRYTU NA RPI.....	28
8.1 SOLIDWORKS	28
II PRAKTICKÁ ČÁST	29
9 KONSTRUKCE.....	30
10 RPI, APLIKACE A POUŽITÉ MODULY.....	31
10.1 RASPBERRY PI 5.....	31
10.2 VISUAL STUDIO CODE	31
11 MYSQL	32

12	POPIS PROGRAMU	34
12.1	POUŽITÉ KNIHOVNY	34
12.1.1	OpenCV	34
12.1.2	Mysql.connector	34
12.1.3	Tkinter	34
12.1.4	Tkcalendar	36
12.1.5	Counter	36
12.1.6	Datetime	36
12.1.7	Email	37
12.1.8	OpenAI	37
12.1.9	Onboard	37
13	FUNKCE PROGRAMU	38
13.1	DTB_CODE.PY	38
13.1.1	Add_new_bar	38
13.1.2	Add_new_item	38
13.1.3	Delete_item	39
13.2	SEND_EMAIL.PY	39
13.2.1	Send_email	39
13.3	AI.PY	40
13.3.1	AI_prompt	41
13.4	POPUP.PY	41
13.4.1	Open/close_keyboard	41
13.4.2	Create_popup	42
13.4.3	Diss_but / Enab_but	42
13.4.4	List_items	42
13.4.5	Detection	43
13.4.6	Vlozit_bar_dtb	44
13.4.7	Vlozit	45
13.4.8	Znovu	45
13.4.9	Storno	45
13.4.10	Select_date	45
13.4.11	Vlozit_item	45
13.4.12	Odebrat	46
13.4.13	Recepty	46
14	KRYT	48
15	ROZŠÍŘENÍ	50
15.1	DETEKCE OBJEKTŮ BEZ ČÁROVÉHO KÓDU	50
15.1.1	Funkce detection_yolo()	55
15.1.2	Funkce vlozit_yolo	55
15.2	DETEKCE POMOCÍ QR KÓDU	56
15.2.1	Funkce detection_QR()	57
15.2.2	Funkce vlozit_QR	57

ZÁVĚR	58
SEZNAM OBRÁZKŮ	61
SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	62

ÚVOD

Moderní technologie stále více pronikají do každodenního života a nacházejí své uplatnění i v běžném chodu domácností. Jednou z oblastí, kde digitální systémy mohou výrazně přispět k vyšší efektivitě a komfortu, je správa potravin. Plýtvání potravinami je celosvětovým problémem s dopady nejen ekonomickými, ale i ekologickými. Řešení, která poskytují lepší přehled o stavu zásob a upozorní na blížící se datum expirace, mohou být důležitým krokem k zmírnění tohoto problému.

Tato bakalářská práce se zaměřuje na návrh a realizaci systému chytré ledničky, která uživatelům umožní efektivní evidenci produktů pomocí čárových kódů, automatické upozorňování na expiraci a v neposlední řadě i návrh receptů na základě aktuálního obsahu lednice. V práci je popsán výběr použitých komponent, jako jsou Raspberry Pi, kamera či dotykový displej, dále návrh uživatelského rozhraní a implementace jednotlivých softwarových modulů v jazyce Python. Výsledkem je systém, který může reálně zjednodušit správu potravin v domácnosti, přispět ke snížení plýtvání a poskytnout uživatelům inspiraci pro každodenní vaření.

I TEORETICKÁ ČÁST

1 STUDIE TECHNOLOGIÍ PRO ČTENÍ ČÁROVÝCH KÓDŮ

Čárový kód je technologický prostředek pro sběr dat. Patent byl přidělen pánům Normanovi Josephu Woodlandovi a Bernardu Silverovi poprvé v roce 1952, avšak poprvé byl vznesen nápad na tento typ sběru dat v roce 1949.

Kód byl původně určen k zaznamenávání informací o produktu u pokladen. Myšlenka o čárovém kódu, který využívá čáry s různými tloušťkami, pana Woodlanda údajně napadla na pláži, kde vycházel z principu Morseovy abecedy. Chtěli požádat firmu IBM, u které pan Woodland pracoval, avšak jejich nápad nebyl komerčně proveditelný.

Patent následně získala firma RCA, která se neúspěšně pokoušela vyvinout komerční aplikaci. Nápad na tuto aplikaci ale nevyšel a roku 1969 patent na čárový kód propadl. V tom samém roku se ale americká Národní asociace potravinářských řetězců začala zajímat o tento nápad a založila komisi zástupců supermarketů, kteří měli cíl vytvořit jednotný kód potravinářských produktů. Firma IBM se v roce 1971 začala zajímat o pana Woodlanda a společně vyvinuli technologii UPC (Universal Product Code).

Poprvé se čárový kód objevil a byl použit na obalu žvýkaček Wrigley's ve městě Troy.[4]

1.1 Čárový kód

Čárové kódy jsou jedním z nejrozšířenějších komerčně používaným způsobem automatické identifikace produktů v ochodech po celém světě.

Čárové kódy mají spoustu výhod pro jejich použití. Tyto výhody jsou třeba:

- Přesnost

Čtením pomocí čtečky dochází k eliminaci lidské chyby při zadávání produktů. Při čtení kódu je zavedena kontrolní číslice, která ověřuje, která slouží k ověření správnosti načtení.

- Flexibilita

Tisknutí kódu může být provedeno na různé materiály a můžou být použity v různých podmínkách.

- Rychlost

- cena, produktivita a další výhody [4]

1.2 Čtení čárových kódů

Čárový kód je tvořen tištěnými černými pruhy dle definované šířky. Je určen pro čtení pomocí skenerů či čteček. Čárové kódy se dají rozlišit na jednodimenzionální (1D) a dvoudimenzionální (2D), přičemž 1D kódy mají díky jednodimenzionálnímu provedení menší kapacitu na uložené informace v kódu.

Tradičně se pro čtení čárových kódů používají laserové snímače. Vyzařují červené světlo, které je pohlcováno černými pruhy kódu a odraženo světlými mezerami. Snímač dále snímá reflexi a převádí je na elektrické signály odpovídající šířce čar a mezer. Od 21. století se začali také kromě tradičních snímačů také používat digitální snímače. Fungují na principu vyfocení čárového kódu a následného dekodování pomocí dekodéru, který je součástí snímače. Hlavní výhodou digitálních snímačů je jejich schopnost číst kódy mnoha směrně a také možnost číst jak jednodimenzionální kódy, tak i dvoudimenzionální. [4]



Obrázek 1 Vizualizace čtení čárového kódu

Zdroj: [4]

1.3 Jiné typy 1D a 2-D kódů

Způsobem, jakým se kódují konkrétní znaky do kódu se liší na více typů. V dnešní době se nachází v této skupině kódu přibližně 200 standardů čárových kódů. Různé typy čárových kódů mohou obsahovat různé informace. Některé typy obsahují číslce a některé mohou obsahovat i speciální znaky a písmena.

Nejpoužívanější typ jednodimenzionálního čárového kódu, který je používán pro potraviny v obchodech je EAN 8 a EAN 13 (obsahuje číselný kód, který je po načtení odkázán na databázi). Další typy používané pro označování obchodní a logistických jednotek jsou UCC a EAN 128. Pro označování malých objektů se používá GS1 DataBar. Další jednodimenzionální používané kódy jsou třeba Code 39, ITF a ITF-14.

Mezi nejrozšířenější dvoudimenzionální kód patří QR kód. Další typy jsou například DataMatrix a PDF 417. [4]

2 EVIDENCE POTRAVIN

Evidence potravin a výrobků je klíčová pro funkci skladování a prodeje produktů. V dnešní době se pro evidenci potravin v supermarketech používají databáze, ve kterých jsou uloženy informace o produktech, které se nacházejí v obchodech. Tyto databáze jsou nezbytnou součástí produkce a skladování, jelikož uchovávají informace potřebné ke správnému a přesnému provozu skladů a obchodů.

Nejčastěji se v systému evidence produktů nacházejí: název produktu, cena, nutriční hodnoty, datum spotřeby a informace potřebné k funkci prodeje a skladování v obchodech.

2.1 SQL databáze

Systém SQL byl vytvořen v roce 1970 a byl dříve označován jako „structured English query language“ (SQUEL). Následně pojem SEQUEL byl zkrácen na SQL. Firma Oracle byla první, která nabízela komerční použití SQL databázového systému pro správu dat.

Systém SQL (Structured query language) je velmi populární a používaný relační databázový systém, který se používá ve velkém rozsahu aplikací a použití. Programátoři datových analýz a vývojáři používají SQL z důvodu lehké integrace do ostatních programů a programovacích jazyků. Výhodou SQL jazyku je jeho jednoduchost na pochopení, protože používá pro své instrukce jednoduchá anglická slova jako jsou třeba CREATE, DESTROY, INSERT a mnoho dalších. Pracuje na principu relačních databází, tudíž informace jsou ukládány v tabulkové formě, která má řádky a sloupce. Díky své architektuře jsou databáze přehledné a lehce se v nich vyhledávají informace.

Systém SQL se používá v mnoha databázových systémech jako jsou Microsoft SQL Server, SQLite, MySQL a PostgreSQL. [5]

2.1.1 MySQL

Původně vyvinut v roce 1995, MySQL je jeden z nejpopulárnějších databázových systémů – často používaný pro webové aplikace. MySQL je zdarma a často používaný v „open source“ balíčcích.

Oproti nerelační databázi jako je třeba NoSQL, který funguje pro představu jako složka s dokumenty, které nejsou strukturované a nemají žádné vztahy mezi sebou, MySQL je relační databázový systém, což znamená, že každá uložená informace má nějaký „vztah“

s jinými daty. Databáze uvnitř systému obsahuje „tables“ neboli tabulky, což jsou tabulky obsahující řady a sloupce dat.

Oba typy databází pracují na takzvaném Klient-Server modelu, což znamená že klient (neboli uživatel) požaduje data a server s databází mu je dodá.

Základní operace pro SQL neboli CRUD jsou:

- CREATE: operace se používá pro vytváření databází a tabulek.
- READ: jsou operace, která nám vypíše námi zvolená data a zobrazí je.
- UPDATE: tato operace se používá při aktualizování dat, které již existují v databázi a chceme je změnit.
- DELETE: operace smaže specifikovaná data. [14]

3 MIKROPOČÍTAČE

Dříve pojem mikropočítač znamenal protiklad slova „velký počítač“, avšak v dnešní době se tento pojem přeměnil na počítač, který je kompaktní zařízení menší než klasický stolní počítač.

V roce 1974 byl představen společností MITS první komerční mikropočítač Altair 8800. Byl to 8bitový mikropočítač, který obsahoval procesor Intel 8080. Procesor byl taktovaný na 2 MHz a jeho paměť byla 256 B. Data se v něm ukládala na magnetické pásky a později diskety. Tento mikropočítač měl provedené připojení mezi komponenty pomocí sběrnice S-100 bus, která se později stala standardem pro sběrnice v počítačích. Mikropočítač se prodával za přibližně 400 amerických dolarů a balíček neobsahoval žádné vestavěné výstupní periferie kromě LED diod na předním panelu, proto musel být připojen na sériový terminál, aby měl počítač nějaký digitální výstup. Pro toto připojení musela být do mikropočítače nainstalována karta pro sériovou komunikaci. [6]



Obrázek 2 Altair 8800

Zdroj: [6]

3.1 Základní informace o mikropočítačích

Mikropočítač je jednodeskový počítač (SBC – Single Board Computer), který se vejde na stůl, ale dá se i přenášet sebou. Aby zařízení mohlo být klasifikováno jako počítač, musí splňovat určité technické požadavky, mezi které patří přítomnost centrální procesorové jednotky CPU (Central processing unit), která má za úkol veškeré výpočetní operace. Dále musí obsahovat I/O periferie, paměť, úložiště, jako je SSD nebo Hard Drive (dříve magnetické pásky, nebo diskety) a jejich propojení pomocí sběrnice, která slouží pro komunikaci a přenos dat mezi komponenty počítače.

- Procesor (CPU): Je to mozek počítače, který provádí výpočty systému a zajišťuje řízení celého systému počítače.

- Paměť RAM a ROM: Rychlá paměť RAM (Random Access Memory) slouží v počítači jako dočasné úložiště pro běžné programy, naopak ROM (Read Only Memory) a také FLASH paměť slouží jako úložiště pro trvalý obsah dat v počítači. Mezi tyto trvalá data patří firmware.

- I/O periférie: Mezi vstupní a výstupní rozhraní patří porty pro připojení obrazovek, klávesnice, myš a další vstupní nebo výstupní zařízení, které uživatel může připojit.

- Operační systém (OS): Operační systém je základní programové vybavení počítače. Mezi tyto systémy patří například Windows, Linux nebo Mac. Jeho funkce jsou správa procesů, pamětí, souborového systému, hardwaru, uživatelského rozhraní a uživatelů. [9] [10]

3.1.1 Raspberry-Pi

Mikropočítač od firmy Raspberry Pi Foundation, která vyvinula první verzi SBC Raspberry Pi Model B v roce 2012. Poprvé byl účel podpořit výuku informačních technologií ve školách. Raspberry Pi potřebuje pro svou funkčnost operační systém a pro tyto účely jsou velmi často použity operační systémy Raspberry Pi OS (dříve Raspbian), Ubuntu nebo DietPi. Operační systém Raspberry Pi je uložen na paměťové kartě, která je vložena do slotu na desce.

Raspberry Pi je skvělý malý samostatný mikropočítač, který na jedné desce implementuje všechny systémové komponenty, jako jsou mikroprocesor, který slouží jako výpočetní jednotka, nadále má paměť RAM, která slouží pro uchování dat pro probíhající úlohu v počítači, řadiče pro rozhraní jako je USB, Ethernet a další. Raspberry Pi má také integrovaný řadič pro bezdrátové připojení k WIFI a bezdrátové Bluetooth. Díky všem těmto komponentům a funkcím tvoří Raspberry Pi skvělou pomůcku pro IT kutily, studenty a vývojáře.

Pro naše účely je použit model Raspberry Pi 5 (dále jen RPI) 8 GB má procesor ARM Cortex-A76 64-bit, který je taktován na 2.4 GHz, který se nachází na čipu Broadcom BCM2712. Paměť RAM LPDDR4X-4267 8 GB. Tato pátá edice čipu obsahuje dva microHDMI porty, dva USB 3.0 porty, dva USB 2.0 porty a jako jeho předchůdce Raspberry

Pi 4 je napájen konektorem USB C. Tento model má také mnoho dalších portů (CSI, DPI, GPIO, ...) a funkcí. [8] [19]



Obrázek 3 Raspberry Pi 5 8GB

Zdroj: [19]

3.2 Použití mikropočítačů

Mikropočítače mají dnes široké využití v mnoha oblastech – od domácností až po specializované systémy a průmyslové aplikace.

Mikropočítače může použít i v mnoha oblastech vzdělávání a výzkumu díky jejich cenové dostupnosti a flexibilnímu použití. Dají se využít v oblasti programování, robotiky a práce s elektrickými obvody, což je perfektní pro studenty na středních a vysokých školách. Studenti mohou experimentovat s operačními systémy, kde mohou tvořit a vyvíjet softwarové aplikace.

V průmyslovém využití se mikropočítače dají skvěle využít pro řízení a monitorování výrobních procesů, fungování výrobních linek, snímání dat ze senzorů a jejich následné zpracování. V automobilovém průmyslu můžeme běžně vidět použití mikropočítačů v palubních deskách automobilu, kde řídí motor, bezpečnostní systémy automobilu nebo jeho multimediální zařízení.

V lékařském průmyslu pro monitorování a diagnostiku pacientů, i pro řízení laboratorního vybavení.

V dnešní době se mikropočítače také uplatňují v rámci internetu věcí - IoT (Internet of Things). S jejich pomocí lze vytvářet chytrá zařízení, která pomocí připojení do sítě dokážou komunikovat mezi sebou a s uživatelem. Jako takovéto zařízení, která jsou běžně známá a používaná patří třeba chytrí asistenti, chytré termostaty, chytré hodinky či náramky a spousta dnes už běžných zařízení, která patří do našeho běžného života. [7]

4 OPERAČNÍ SYSTÉM

4.1 Linux

V roce 1991 měl vysokoškolský student Linus Torvalds na svém osobním počítači tehdejší operační systém MINIX, což byla taková odlehčená verze systému UNIX. UNIX byl ale placený operační systém, který zvládal multitasking, ale Linusovi to nestačilo. Vytvořil jádro operačního systému, které čerpalo inspiraci ze systému UNIX, které pojmenoval LINUX (Linux Is Not UniX). Linux není jen jeden operační systém, ale rodina různých operačních systémů, které pracují osvobozeně na principu Linuxového kernelu. Různé verze a distribuce Linuxu vycházejí od různých vývojářů s vlastními desktopovými vzhledy a balíčky systémových aplikací. Mezi tyto verze patří třeba Ubuntu, Debian, Fedora, Arch Linux a mnoho dalších.

Systém Linux je skvělý pro použití v serverových počítačích díky možnostem, jak systém upravit pro specifické potřeby. Systém je zcela zdarma a skvěle konfigurovatelný, což z něj dělá perfektní systém pro servery. Díky těmto funkcím pracuje na systému Linux téměř 70 % všech serverů na celém světě. U serverového použití můžeme najít systém Linux v podobě LAMP serverů, což je zkratka pro „Linux Apache MySQL PHP server“.

Systém Linux lze také použít pro desktopové účely, kde může být skvělou náhradou pro systémy jako jsou třeba Windows nebo Mac v určitých použitích. Linux pro účetní účely je jeden ze skvělých příkladů, protože u většiny účetních aplikací nezáleží, na jakém operačním systému běží, proto neplacený systém může být skvělá úspora pro firmu. Linux pro desktop může mít ale pro běžné používání pár problémů, jako je třeba instalace ovladačů a konkrétních aplikací. Desktopové distribuce Linux-u používá přibližně 4 % uživatelů. [17] [18]

4.2 Ubuntu

Ubuntu je linuxový operační systém, který byl pojmenován po starém africkém slovu, které mělo význam „humanity to others“, v překladu „lidskost ostatním“. Poprvé v roce 2004 Mark Shuttleworth shromáždil malý tým vývojářů distribuce Debian aby spolu vytvořili lehce ovladatelný desktopový operační systém založený na Linux-u, který nazvali Ubuntu. Tento operační systém byl zdarma a všichni ho mohli používat rovnoměrně. [16]

5 PROGRAMOVACÍ JAZYKY

Pro programování určitých úloh na počítači je potřeba sdělit procesoru, co má vykonávat za operace a výpočty. Pro tyto účely byly vyvinuty programovací jazyky. Programovací jazyk je určitý způsob algoritmizace postupu provedení instrukcí. Jde o prostředek mezi programátorem a počítačem, který umožní programátorovi zadat počítači instrukce pro provedení určité požadované úlohy nebo aplikace.

Pro různé účely byly vyvinuty různé programovací jazyky, které se od sebe liší na nižší a vyšší programovací jazyky podle míry abstrakce. Dále také můžeme programovací jazyky dělit dle jejich způsobu překladu a spuštění na komplikované, jako je třeba Java, C, atd... a interpretované, jako je třeba Basic, Perl nebo Python.

5.1 Historie jazyku python

Jazyk Python byl inspirován jazykem ABC a byl vytvořen v roce 1990-1991 panem Guido van Rossumem v Nizozemsku.

Jako jeho logo používá had krajta, jelikož je to další význam slova Python v anglickém jazyce. Dnes se používá verze jazyku Python 3. Tato verze se poprvé objevila v roce 2008. [11]



Obrázek 4 Logo hada jazyku python

Zdroj: <https://www.python.org/community/logos/>

5.2 Použití programovacích jazyků

V dnešní době se programovací jazyky používají pro velmi rozsáhlou škálu aplikací a funkcí. Pomocí určitého programovacího jazyku je možno vytvořit jednoduché i složité aplikace, webové stránky, a mnoho dalších typů aplikací pro různorodé použití.

Příklady programovacích jazyků pro back-end webových stránek jsou třeba Python, JavaScript, PHP, Java, C# a mnoho dalších.

5.3 Jednoduchý popis jazyka Python

V jazyku Python lze importovat různé moduly pro určité funkce. Tyto moduly pak mají v sobě již funkce, které můžeme používat v našem kódu. Stejným způsobem lze také importovat soubory typu „*.py“, které pokud obsahují vytvořené funkce, pokud tyto soubory obsahují funkce, lze je následně používat v jiném kódu.

V Pythonu lze tvořit uživatelské funkce pomocí slova „def“, kde tyto funkce mohou obsahovat kód pro určitý účel, který pak můžeme snadno volat pomocí názvu funkce a jejích parametrů v kódu.

Jazyk Python je takzvaný dynamicky typovaný jazyk to znamená, že na rozdíl od většiny programovacích jazyků (např. C nebo Java) při vytváření proměnných se nedefinuje její datový typ. Místo toho jazyk sám přiřadí datový typ proměnné po přiřazení hodnoty. Například: pokud by jsme chtěli proměnnou „cislo“ nastavit jako integer proměnnou, stačí napsat „cislo = 5“, pak po spuštění programu Python přiřadí proměnné hodnotu a určí její datový typ. Python od verze 3.5 podporuje type hinting, což umožňuje vývojářům naznačit datový typ proměnných pomocí dvojtečky. Type hinting slouží hlavně ke zlepšení čitelnosti kódu a statické analýze, ale nemá vliv na samotné běhové chování programu.

Například:

```
cislo: int = 5
```

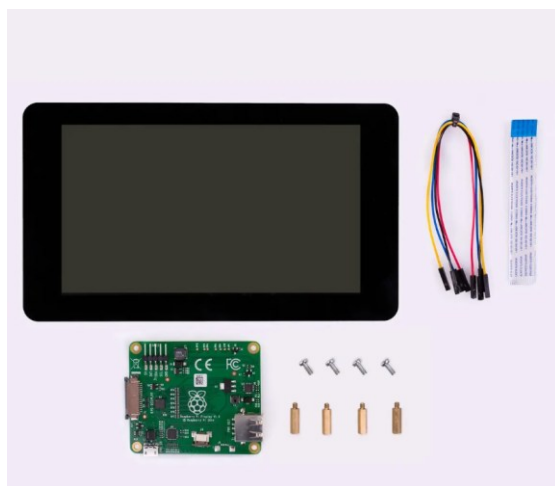
Po spuštění kódu se kód vykonává řádek po řádku, ale může být spuštěn v takzvaném ladicím režimu („Debugging mode“). Tento režim/mód má za účel interaktivně spouštět kód řádek po řádku na základě vstupu od uživatele, což umožňuje lehkou experimentaci a testování. [11] [3] [1]

6 DISPLAY A KAMERA RPI

6.1 Raspberry Pi Touch Display

RPI Touch Display je kapacitní dotykový displej připojený k RPI přes DSI (Display Serial Interface) konektor. Displej má úhlopříčku 7" s rozlišením 800x480 pixelů a je typu LCD. Díky jeho kapacitním vlastnostem nabízí tento displej stejný komfort použití jako displej mobilního telefonu nebo tabletu, což vede k usnadněnému uživatelskému použití. Jelikož je kapacitní, nepotřebuje při dotyku uživatel vyvíjet tlak na displej pro zaznamenání dotyku.

Na jeho zadní straně se nachází řídicí deska pro připojení konektoru DSI pro přenos signálu a také piny pro napájení displeje (+5 V a GND), které se připojí na GPIO piny, což napájí celý displej přímo z RPI, čímž odpadá potřeba externího napájení. Řídicí deska má také kovové distanční sloupky pro možnost namontování RPI přímo na desku displeje, což umožňuje kompaktnost celého systému. [12]



Obrázek 5 Obsah balení displeje Raspberry Pi Touch Display

Zdroj: [12]

6.2 Arducam 2Mpx IMX291

Kamera Arducam IMX291 je připojená k RPI přes USB kabel a nepotřebuje instalovat žádné ovladače, což vede k velmi usnadněnému použití kamery pro jakýkoliv čip, který má na desce USB port. Má nativní ovladače UVC pro systémy Windows, Linux a Mac, což znamená, že tyto systémy jsou po připojení do USB portu s kamerou kompatibilní a kamera je ihned připravena k použití po zapojení. Kamera obsahuje obrazový senzor Sony IMX291 s parametry 1/2,8", 2Mpx. Snímač CMOS barevný má dobrý dynamický rozsah 80 dB a

dokáže také zachytit minimální osvětlení 1 mLux, což výrazně usnadňuje použití kamery i za šerého osvětlení. Tato kamera zvládá nahrávání v kvalitách a formátech:

- H.264 se 30 fps s rozlišením 1920×1080
- MJPG se 30 fps s rozlišením 1920×1080
- YUY2 se 30 fps s rozlišením 640×360

Disponuje také širokým zorným polem (FOV – Field Of View) 160° s zaostřením od 2 m do nekonečna. Samotný objektiv také obsahuje integrovaný IR filtr. Systém kamera má také automatické ovládání vlastností jako je sytost, kontrast, ostrost, vyvážení bílé barvy a expozice. Deska má také 2 mikrofony pro záznam zvuku. [13]



Obrázek 6 Kamera Arducam 2Mpx IMX291

Zdroj: [13]

6.3 Endoskop inspekční kamera USB-C 5M LED

Endoskop disponuje rozlišením 640x480 px a nahrává video ve 30 snímcích za sekundu. Průměr kamery je 5,5 mm a má voděodolnost IP67. Její pozorovací úhel je 70 ° s pracovní teplotou od 0 ° do 70 ° C.

Pro propojení využívá USB a má LED osvětlení s ovladatelnou intenzitou. Nepotřebuje instalaci ovladačů, a tak lze po připojení ihned používat kameru a všechny její funkce. Délka kabelu pro připojení je 5 metrů. [20]



Obrázek 7 Endoskop inspekční kamera

Zdroj: [20]

7 CHYTRÉ LEDNICE

V dnešní době bychom mohli chytré lednice popsat jako lednice s integrovaným multimediálním systémem. Chytré lednice mají většinou systém pro napouštění vody či vytváření a dispenzi ledu a dotykovou plochu s obrazovkou pro multimediální účely. Lednice se většinou dokážou připojit k domácí síti Wi-Fi a díky ní mít přístup k internetu a možnosti přehrávání videí, vyhledávání informací na internetových stránkách a mnoho dalších funkcí. Tento systém se podobá rozhraní chytrých telefonů nebo televizí.



Obrázek 8 Americká lednice LG GSGV80PYLD

Zdroj: <https://www.alza.cz/lg-gsgv80pyld-d7951209.htm>

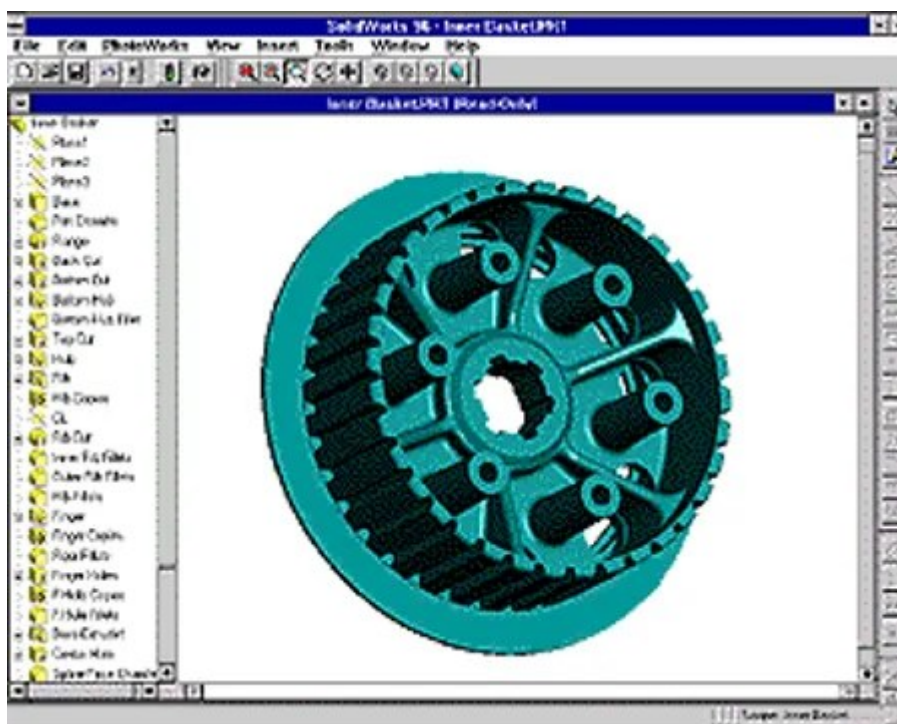
Pro naše účely budeme pod pojmem chytrá lednice rozumět zařízení se systémem pro evidenci potravin, návrh receptů a kontrolu a upozornění datumu expirace.

8 3D MODEL KRYTU NA RPI

Modelování vzhledu krytu pro RPI a display byl použit software SolidWorks, který umožňuje tvoření 3D modelů s přesnými rozměry pro naše účely. Kryt pro lednici byl vytvořen, aby měl vzhled retro lednice. Finální model krytu byl připraven pro výrobu pomocí 3D tisku.

8.1 SolidWorks

Historie programu Solid Works sahá do roku 1995 kdy vyšla první verze softwaru. V tehdejší době cena za tento software byla pouhých 4000 USD oproti ostatním tehdejším CAD softwarům, které stály kolem 18000 USD. Program byl velmi intuitivní a snadný na používání oproti ostatním CAD softwarům v tehdejší době. Modely se zobrazovaly v plném 3D zobrazení se stínováním oproti bílému pozadí což usnadnilo modelování a tvoření modelů.



Obrázek 9 Solid Works 1995

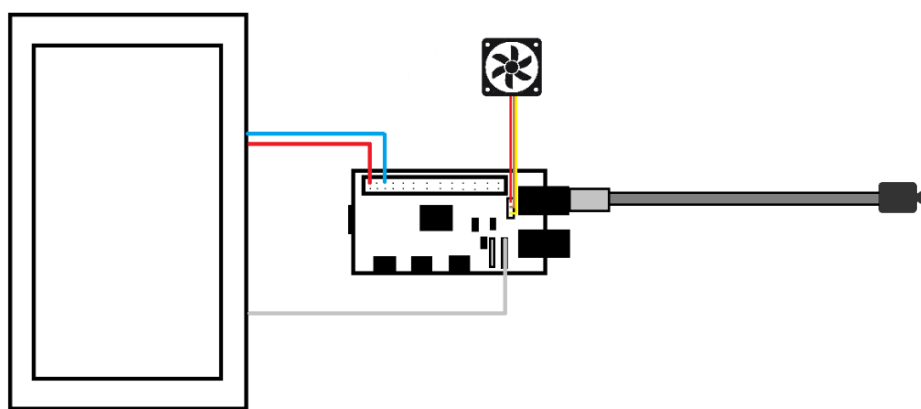
Zdroj: [15]

Nová verze softwaru vycházela každý rok až do dnešního dne, kdy dnes již existuje přes 30 verzí softwaru. [15]

II PRAKTICKÁ ČÁST

9 KONSTRUKCE

Pro zobrazení aplikace na evidenci potravin je použit kapacitní displej, který umožňuje intuitivní a pohodlné ovládání dotykem. Díky vysoké citlivosti a kvalitnímu rozlišení displeje je zajištěna dobrá čitelnost a snadná navigace v uživatelském rozhraní. Pro snímání čárových kódů slouží kamera připojená přes USB, která umožňuje načítání kódů z obalů potravin. Tato kamera je optimalizována pro různé světelné podmínky, což zajišťuje spolehlivost skenování i v méně ideálních prostředích. Celý systém tak usnadňuje evidenci a správu zásob potravin, což přispívá k efektivnějšímu hospodaření. [12] [13]



Obrázek 10 Schéma zapojení

Zdroj: Vlastní

10 RPI, APLIKACE A POUŽITÉ MODULY

Pro začátek byly dodány RPI 4, kabeláž a kamera. Úvodní testování probíhalo na RPI 4 a standardní kameře. Pomocí software-u Raspberry Pi Imager byl na SD kartu nainstalován operační systém Ubuntu, který byl následně na samotném RPI nakonfigurován.

Pro práci byly následně staženy a nainstalovány různé aplikace (např. Visual Studio Code) a v průběhu práce byly dle potřeby doinstalovány knihovny jazyka Python jako jsou třeba Onboard, OpenCV a další. Tyto knihovny byly staženy a nainstalovány pomocí příkazu:

```
sudo apt install python-*
```

Pomocí tohoto způsobu byli také instalovány všechny ostatní potřebné knihovny. Některé aplikace a knihovny šli také stáhnout pomocí aplikace App center systému Ubuntu. Po nějakém čase se naskytla možnost pracovat na novější verzi RPI, konkrétně Raspberry Pi 5 8 GB, takže od tohoto bodu bude popisován postup práce jen za použití této novější verze RPI. Už po úvodním testování RPI 5 se systémem Ubuntu bylo v porovnání s RPI 4 pozorováno výrazné zrychlení všech aplikací.

10.1 Raspberry PI 5

Na novou verzi RPI byl stejným způsobem nainstalován operační systém Ubuntu ve verzi 24.04, aby se zaručil chod dosavadního programu tvořeného v programu Visual Studio Code. Pro Python verze 3.12.3 byl otestován jednoduchý program pro vypsání textu do konzole, který běžel správně.

10.2 Visual Studio Code

Pro vývoj funkcí programu byl pomocí oficiálních stránek Visual Studio nainstalován program Visual Studio Code. Program slouží jako vyhovující programovací prostředí pro účely v jazyku Python díky jeho schopnosti spouštět program, debugging a instalaci knihoven pro různé účely.

Programovací prostředí aplikace je také nápomocné s funkcemi chybových hlášení při překladu kódu, nápovědy pro automatické doplňování kódu a mnoho podobných funkcí. Jedna z hlavních výhod pro účely práce s databázovým systémem byla možnost jednoduše nainstalovat rozšíření SQL Viewer. Toto rozšíření umožnilo zobrazit obsah a strukturu databáze, a také psát do SQL Query.

11 MYSQL

Na RPI byl nainstalován relační databázový systém MySQL, který slouží pro uložení názvů, čárových kódů a data expirace do databáze a následné pracování s těmito daty. Pro instalaci byly použity příkazy:

```
sudo apt install mysql-server -y
```

Tento příkaz nainstaluje do systému databázový systém MySQL se všemi potřebnými soubory pro správný běh systému. Následně byl zadán příkaz pro správné nastavení pro zabezpečení systému:

```
sudo mysql_secure_installation
```

Tento příkaz umožňuje nastavit zabezpečení systému v podobě přihlašovacích údajů – jméno a heslo, možnost připojení jako správce a další možnosti. Pro přihlášení do systému MySQL lze použít příkaz:

```
sudo mysql -u root -p
```

Tento příkaz umožní přihlášení a následnou práci v systému jako je tvorba databází, tvorba uživatelů a přiřazování práv pro správu systému uživatelům.

Pro účel bakalářské práce byl systém používán na správu potravin a ukládání informací o potravinách do databáze pro jednoduchou přehlednost a funkčnost.

Funkce MySQL jsou srozumitelné a umožňují efektivní práci s databázemi. Tyto funkce mají různé účely, které vykonávají různé operace v systému. Například:

```
CREATE DATABASE název_databáze;
```

Instrukce pro vytvoření nové databáze s libovolným jménem.

```
DROP DATABASE <název_databáze>;
```

Instrukce pro smazání databáze.

```
CREATE TABLE <název_tabulky> (<sloupec1> datový_typ_sloupce,  
<sloupec2> datový_typ_sloupce, ...);
```

Instrukce pro vytvoření nové tabulky s libovolným jménem a jeho sloupce s libovolnými názvy.

```
DROP TABLE název_tabulky;
```

Instrukce pro smazání tabulky.

```
ALTER TABLE název_tabulky ADD název_sloupce datový_typ_sloupce;
```

Instrukce pro upravení libovolného parametru v tabulce s libovolným jménem, přidání sloupce s libovolným názvem a přiřazením datového typu.

```
SELECT * FROM název_tabulky;
```

Instrukce pro výpis veškerého obsahu z databázové tabulky. Lze napsat názvy sloupců, či řádků na místo * pro výpis ze volených sloupců, či řádků.

```
DELETE FROM název_tabulky WHERE podmínka;
```

Instrukce pro smazání obsahu z tabulky a podmínkou pro smazání.

```
INSERT INTO název_tabulky(název_sloupce1, název_sloupce2) VALUES  
('data1', 'data2');
```

Instrukce pro vložení libovolných dat do sloupců tabulky.

12 POPIS PROGRAMU

Program slouží k evidenci produktů v lednici, upozorňování na dobu expirace produktů vizuálně a také pomocí zaslání emailu a navrhování receptů na základě produktů v lednici. Tyto funkce zajišťují programy v jazyku Python verze 3.12.3. Každý program obsahuje funkce, které vykonávají různé procesy pro účely přidání, odstranění, sledování a podobně.

12.1 Použité Knihovny

Při vývoji chytrých vlastností lednice byly do souborů importovány různé knihovny, které poskytují specifické nástroje pro zpracování obrazu, databáze, GUI a dalších funkcí programu.

12.1.1 OpenCV

OpenCV (Open Computer Vision Library) je multiplatformní knihovna zaměřena na zpracování obrazu a počítačové vidění. „cv2“ je zkratka používaná pro import aktuální verze knihovny. V bakalářské práci je použita knihovna na detekci čárových kódů z živého videa, které snímá kamera připojená pomocí USB k RPI 5. OpenCV dokáže nejen snímat z živého videa zachyceného z kamery, ale také i z fotek uložených v počítači. Tento způsob je pro využití na vizuální identifikaci produktů velmi užitečný a efektivní, jelikož OpenCV má funkci pro tento typ detekce čárových kódů z videa s dostatečnou rychlostí a přesností.

12.1.2 Mysql.connector

Knihovna mysql.connector v Pythonu slouží k propojení Python aplikace s databází MySQL. Umožňuje:

- Připojit se k MySQL databázi
- Spouštět SQL dotazy (např. SELECT, INSERT, UPDATE, DELETE)
- Číst výsledky dotazů
- Spravovat databázové připojení

12.1.3 Tkinter

Knihovna Tkinter je jednoduchá knihovna pro vytváření uživatelských rozhraní pro aplikace v jazyku Python. Poskytuje rychlý způsob, jak vytvářet uživatelská rozhraní v Pythonu ať už program běží na operačním systému Windows, MAC nebo Linux. Funkce knihovny

používají běžná anglická slova jako je třeba „Button“ pro tvorbu tlačítka, „Label“ pro jednoduchý text a mnoho dalších funkcí. Knihovna má také funkci „pack“ která se používá pro zobrazení a takzvaného „zabalení“ funkce na obrazovce. Pro příklad:

```
button_vlozit_item = tk.Button(popup, text="vlozit",
command=vlozit_item, bg="black", fg="white")

button_vlozit_item.pack(fill= "x",side="top", padx=10, pady=0)
```

Tyhle dva řádky vytvoří tlačítko s názvem „button_vlozit_item“ v prvním řádku a nadefinují jeho atributy:

- `popup` určí v jakém okně se má tlačítko vytvořit
- `text="vlozit"` určuje že v tlačítku se zobrazí text „vlozit“
- `command=vlozit_item` udává jakou akci má stlačení tlačítka vykonat, v tomto případě je to zavolat funkci se jménem „vlozit_item“
- `fg = "white"` určuje jakou barvou se vykreslí „foreground“ neboli v překladu popředí s textem, v tomto případě bude text tlačítka bílý
- `bg = "black"` určuje jakou barvou se vykreslí naopak „background“ neboli v překladu pozadí tlačítka, v tomto případě bude pozadí za textem tlačítka černé
- `side="top"` připne tlačítko k vrchní straně okna. Tlačítko tam zůstane, i kdyby se okno zvětšilo nebo zmenšilo.
- `Fill="x"` roztáhne tlačítko po ose X od kraje okna do druhého kraje okna a vždy ho bude roztahovat i kdyby se okno roztáhlo nebo zmenšilo
- `padx=10` určuje, že od tlačítka na ose Y neboli nahoře a dole od tlačítka se bude udržovat mezera od ostatních prvků aplikace, která v tomhle případě bude veliká 10 pixelů
- `pady=5` určuje, že od tlačítka na ose X neboli vpravo a vlevo od tlačítka se bude udržovat mezera od ostatních prvků aplikace, která v tomhle případě bude veliká 10 pixelů

Tohle je jen pár příkladů, co lze zapsat za atributy do funkcí „button“ a „pack“. Funkci „pack“ lze také použít pro „zapomenutí“ neboli odstranění prvku z okna aplikace. Tohoto lze dosáhnout pomocí „forget“:

```
button_1_1.pack_forget()
```

12.1.4 Tkcalendar

Knihovna Tkcalendar přidává funkci „calendar“, která funguje s Tkinter a umožňuje vytvoření funkčního interaktivního kalendáře, který lze vykreslit v okně vytvořeném pomocí Tkinter. Tento kalendář lze nakonfigurovat dle potřeby s funkcemi pro minimální povolený zadaný datum a také uložit volitelné datum na stlačeném datumu v kalendáři.

```
cal = Calendar(popup, mindate = date.today(), selectmode="day",
               year=2025, month=1, day=28)
cal.pack(fill = "x", pady=10)
selected_date = cal.selection_get()
```

Tyto tři řádky mají za účel vytvořit a vykreslit kalendář v okně „popup“ s minimálním možným datem pro výběr (přiřazení atributu „mindate“), který je roven dnešnímu datumu (pomocí knihovny „datetime“) a módem pro zobrazení dnů, měsíců a roku. Uloží se do proměnné „cal“ a ta se pak následně „zabalí“ do okna pomocí funkce „pack“.

Po stisknutí políčka v kalendáři s datem políčko změní barvu na modrou, aby indikovalo zvolené pole/datum. Toto zvolené pole/datum se následně uloží do proměnné „selected_date“ ve tvaru „ROK-MĚSÍC-DEN“.

12.1.5 Counter

Knihovna „Counter“ je použita pro počítání při načítání čárových kódů pomocí kamery.

```
codes.append(result)
nej_z_codes = Counter(codes).most_common(1)[0][0]
```

Counter slouží pro vybrání nejčastěji se objevovaného kódu z proměnné pole „codes“, což pomáhá k přesnější detekci čárového kódu.

12.1.6 Datetime

Knihovna „datetime“ je využívána pro snímání datumu a času v přítomném čase ale také pomocí ukládání do proměnných může být využita pro uložení jiných ať už dřívějších či pozdějších datumů a časů.

```
today = datetime.now().date()
one_week_away = today + timedelta(days=7)
```

Pro účely této bakalářské práce je využívána ve funkcích pro sledování doby expirace a zasílání emailů s upozorněním na blížící se datum expirace produktů.

12.1.7 Email

Knihovna Email je využívána k zaslání emailu z předdefinované emailové adresy na předem definovanou emailovou adresu s libovolným textem emailu a předmětem.

12.1.8 OpenAI

Pro funkci na navrhování receptů z produktů vložených do lednice je využita knihovna OpenAI, která umožňuje komunikaci s chatbotem ChatGPT verze „gpt-3_5-turbo-0125“ pomocí Python-u. Pro funkci této knihovny je ale potřeba generace osobního API klíče z oficiálních stránek openai: <https://platform.openai.com/settings/organization/api-keys>. Pomocí API klíče lze odesílat dotazy a přijímat odpovědi prostřednictvím modelu ChatGPT použitím tokenů. Tokeny se odečítají z předplaceného kreditu, který je možné spravovat přes uživatelský účet OpenAI. Balanci lze nabíjet a kontrolovat na oficiálních stránkách openai: <https://platform.openai.com/settings/organization/usage>

12.1.9 Onboard

Pro účely zadávání názvů zcela nově přidaných produktů do databáze čárových kódů je využita virtuální klávesnice Onboard. Tuto virtuální klávesnici lze otevírat a zavírat pomocí jazyku python a je funkční s aplikacemi vytvořenými pomocí knihovny Tkinter. Pro správné použití klávesnice musí být nejprve ale řádně nastavena. Klávesnice obsahuje vlastní grafické rozhraní pro konfiguraci rozložení a stylu, což usnadňuje její použití s Tkinterem.

13 FUNKCE PROGRAMU

13.1 Dtb_code.py

Soubor „dtb_code.py“ slouží k provádění operací s databázovým systémem MySQL. Nachází se v něm propojení s databází.

```
mydb = mysql.connector.connect(  
    host="localhost",  
    user="mahdal",  
    password="fai",  
    ssl_disabled=True,  
    database="mydatabase"  
)
```

Pomocí tohoto způsobu se při spuštění programu systém připojí k databázi a její tabulkám „barcode_dtb“ a „items“. Dále se s databází pracuje pomocí funkcí.

13.1.1 Add_new_bar

Funkce slouží pro zavedení nového čárového kódu, který ještě doposud nebyl do systému přidán. Pracuje na principu předání hodnot proměnných pro jméno a čárový kód produktu. Tyto parametry obdrží při zavolání funkce z programu „popup.py“.

```
def add_new_bar(name, code):  
    curr.execute('''INSERT INTO barcode_dtb (ItemName, Barcode) VALUES  
    (%s, %s)''', (name, code))  
    mydb.commit()
```

Funkce provede příkaz „INSERT INTO“ což vytvoří v databázi „barcode_dtb“ nový řádek s parametry ID, ItemName, Barcode. Parametr ID je zaveden, aby se samostatně zvyšoval pomocí funkce „AUTOINCREMENT“ databázového systému MySQL.

13.1.2 Add_new_item

Tato funkce má podobný účel jako funkce „add_new_bar“ přidat nový řádek s parametry do databáze. Přidává ale produkty do databáze „items“.

```
def add_new_item(name, code, expdate):  
    curr.execute('''INSERT INTO items (name, barcode, expdate) VALUES  
    (%s, %s, %s)''', (name, code, expdate))  
    mydb.commit()
```

Na rozdíl od předchozí funkce si tato přebírá i parametr „expdate“ což pomocí kalendáře obsahuje datum expirace produktu v podobě „ROK-MĚSÍC-DEN“.

13.1.3 Delete_item

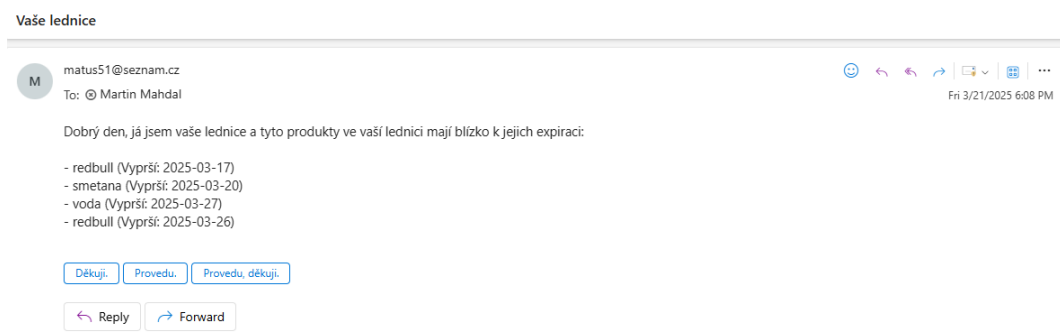
Pracuje na principu odstraňování řádků z databáze „items“ na základě předaného parametru „id“, který je předán funkci v souboru „popup.py“.

```
def delete_item(id):  
    try:  
        curr.execute(''DELETE FROM items WHERE ID = %s'', (id,))  
        mydb.commit()  
        print(f"\nDeleting: ID={id}\n")  
    except mysql.connector.Error as e:  
        print(f"error: {e}\n")
```

Jelikož během testování probíhaly komplikace s odstraňováním produktů byl přidán blok „try/except“, který slouží pro odchycení chyb a vypsání do konzole.

13.2 Send_email.py

Soubor „send_email.py“ obsahuje funkce pro zasílání emailu s dobou expirace kratší než týden od aktuálního data. Tato funkcionality je realizována pomocí knihovny „email“ a „datetime“.



Obrázek 11 Ukázka zaslaného emailu

Zdroj: Vlastní

13.2.1 Send_email

Tato funkce zajišťuje funkčnost celého programu pro zasílání emailu. Hlavní princip je přihlášení pomocí zadání parametrů „sender_email“ a „password“ (hodnota proměnné „password“ v ukázce zdrojového kódu byla nahrazena znakem hvězdičky z důvodu ochrany osobních údajů). Skrz tento přihlášený email pak může zaslat na adresu „reciever_email“ požadovanou zprávu.

```

def send_email():
    sender_email = "matus51@seznam.cz"
    receiver_email = "m_mahdal@utb.cz"
    password = "*"

    msg = MIMEMultipart()
    msg['From'] = sender_email
    msg['To'] = receiver_email
    msg['Subject'] = 'Vaše lednice'
    today = datetime.now().date()
    one_week_away = today + timedelta(days=7)
    query = """SELECT name, expdate FROM `mydatabase`.`items` WHERE
expdate <= %s OR (%s <= expdate AND expdate <= %s)"""
    dtb_code.curr.execute(query, (today, today, one_week_away))
    result=dtb_code.curr.fetchall()

    body = "Dobrý den, já jsem vaše lednice a tyto produkty ve vaší
lednici mají blízko k vypršení jejich datumu expirace:\n\n"
    for name,expdate in result:
        body += f"- {name} (Vyprší: {expdate})\n"
    msg.attach(MIMEText(body, 'plain'))

    try:
        server = smtplib.SMTP('smtp.seznam.cz', 587)
        server.starttls()
        server.login(sender_email, password)
        text = msg.as_string()
        server.sendmail(sender_email, receiver_email, text)
        server.quit()
        print("Email odeslan!")
    except Exception as e:
        print(f"error: {e}")

```

Do proměnné „body“ lze napsat požadovanou zprávu pro zaslání. Využívá soubor „dtb_code.py“ pro funkčnost databáze a následné získání jmen a expiračních dob produktů pro zaslání. Pomocí změn hodnot proměnných lze měnit odesílatele, příjemce, font textu, text zprávy a předmět emailu.

13.3 AI.py

Program má za princip funkce propojení s chatbotem ChatGPT aby navrhnul recept pomocí zaslání jmen z lednice za pomocí programu pro databázi „dbt_code“. Pro jeho funkčnost ale musí být v programu definována proměnná pro propojení pomocí API klíče.

13.3.1 AI_prompt

Funkce má za úkoly vzít jména a doby expirace produktů v lednici z databáze „items“

```
def AI_prompt():
    dtb_code.curr.execute("SELECT name FROM `mydatabase`.`items`")
    names_v_lednici = dtb_code.curr.fetchall()

    items_list = [item[0] for item in names_v_lednici]

    items_str = ", ".join(items_list)

    prompt = f"napiš recept který by šel udělat z věcí v lednici a  
napiš potřebné gramáže a postup: . V lednici je: {items_str}"

    response = client.chat.completions.create(
        model="gpt-3.5-turbo", # Change to "gpt-4" if needed
        messages=[{"role": "user", "content": prompt}]
    )
    return (response.choices[0].message.content)
```

Tyto údaje zašle v proměnné „prompt“ jako vstup do chatbota verze gpt-3.5-turbo a ten jako output zašle odpověď s receptem formátovaným, jak bylo specifikováno ve vstupu.

13.4 Popup.py

V souboru „popup.py“ běží hlavní funkce celého programu pro zobrazení a funkčnost okna na obrazovce. Kód je strukturován tak, že funkce pro provádění různých úkolů jsou ve funkci „create_popup“, která se opakuje ve smyčce. Na startu programu jsou importovány všechny knihovny popsané dříve a soubory „send_email“, „dtb_code“ a „AI“. Kromě funkcí také obsahuje definování parametrů okna aplikace „popup“.

13.4.1 Open/close_keyboard

Tyto dvě funkce slouží pro spouštění virtuální klávesnice Onboard pomocí příkazů:

```
subprocess.Popen(["onboard"]) a os.system("pkill onboard").
```

Klávesnice je použita pro zadávání názvů nových produktů do zadávacího pole aplikace.



Obrázek 12 Ukázka Onboard klávesnice

Zdroj: Vlastní

13.4.2 Create_popup

Ve funkci „create_popup“ je obsažen celý program a tato funkce běží ve smyčce. Kromě všech ostatních funkcí také obsahuje tvorbu a funkčnost hlavních tlačítek a listu v hlavním okně.

13.4.3 Diss_but / Enab_but

Názvy těchto dvou funkcí znamenají „Disable buttons“ a „Enable button“ což je přesně jejich funkčnost. Jedna funkce tlačítka na obrazovce zapíná a druhá je vypíná, aby se nedali používat v určitých situacích, ale stále byli zobrazení.

Tohoto dokáží docílit pomocí volání názvu tlačítka a pak jeho konfigurování pomocí slova „config“ za tečkou a v závorce nastavení stavu „state“ tlačítka na „disabled“ což je vypnuto a „active“ což je aktivováno. Příklad tohoto příkazu vypadá následovně:

```
button_vlozit_bar_dtb.config(state="disabled")
```

13.4.4 List_items

Pomocí zavolání této funkce v kódu je možné obnovit list produktů („tree“) na obrazovce. List zobrazuje produkty, které si dokáže pomocí využití funkcí z programu „dtb_code“ vzít jméno, dobu expirace a id z databáze aktuálních produktů s názvem „items“ a vypsát je na listu na obrazovce. Pro tuto funkčnost je použit příkaz:

```
tree.insert("", tk.END, values=(id[0], name[0], expdate[0]), tags = ("almost_expired",))
```

K těmto produktům také pomocí podmínky pro porovnání s proměnnou „today“ a proměnnou „one_week_away“ přiřazuje „tagy“ neboli štítky které mají přiřazeny barvy pro vizuální rozlišení dob expirace.

```
tree.tag_configure("almost_expired", foreground="orange")
tree.tag_configure("expired", foreground="red")
```

```
tree.tag_configure("OK", foreground="black")
```

Barva textu značí, jestli má produkt v lednici dobu expirace menší nebo rovno než aktuální dnešní datum, dobu expirace větší než dnešní datum ale menší než 7 dní od aktuálního datumu a větší než 7 dní od aktuálního datumu. Barvy znamenají:

Černá = produkt má dobu expirace větší než 7 dní od aktuálního datumu

Oranžová = produkt má menší datum expirace menší než 7 dní od aktuálního datumu

Červená = produkt má datum expirace menší nebo rovno aktuálnímu datumu

Podle těchto barev lze jednoduše vizuálně rozlišit prošlé produkty a ty kterým brzo vyprší datum expirace.

ID	Název	expdate
84	máslo	2025-04-25
85	hovězí maso	2025-05-09
86	smetana	2025-05-14
87	paprika	2025-05-22
90	rajče	2025-04-21

Obrázek 13 Ukázka zobrazení produktů v listu

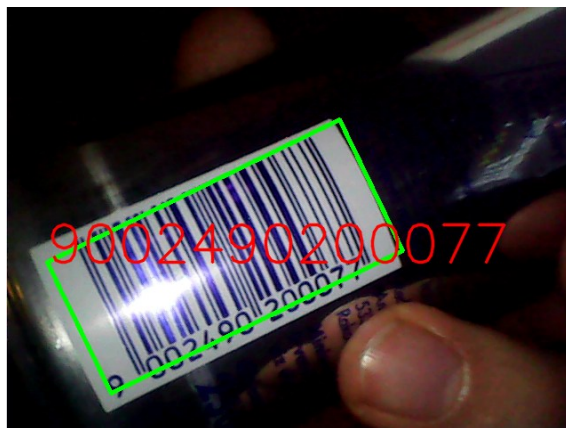
Zdroj: Vlastní

13.4.5 Detection

Jedná se o funkci pro samotnou vizuální detekci čárových kódů produktů pomocí kamery připojené k RPI 5. Využívá knihovny „OpenCV“, která již obsahuje potřebné příkazy pro funkčnost vizuální detekce čárových kódů:

```
cv2.barcode.BarcodeDetector
```

Pokud se na zorném poli kamery objeví čárový kód tak kolem něj v zobrazovacím okně „Detektor EAN-ov“ vykreslí obdélník a detekovaný kód zobrazí jako text nad vykresleným obdélníkem. Tento proces čtení kódu provádí do doby, než se detekce kódu nesplní pět krát. Těchto pět načtených kódů ukládá do proměnné „codes“, ze které se nakonec uloží do proměnné „nez_z_codes“ nejvíce krát objevující se kód v proměnné. Tímto procesem načítání se eliminuje šance načtení nesprávného čárového kódu a udržuje správné načtení kódu 100 % času.



Obrázek 14 Načítání čárového kódu

Zdroj: Vlastní

Během vytváření této funkce se vyskytovali problémy se zavíráním okna pro zobrazení zorného pole kamery a probíhaly testy různých kamer. Dle testů bylo vyhodnoceno, že pro nejpřesnější detekci dosahuje endoskop inspekční kamera USB-C, která byla zabudována do krytu kamerového skeneru.

13.4.6 Vložit_bar_dtb

V této funkci se využívá propojení se souborem „dtb_code.py“ pro vložení názvu a čárového kódu do databáze všech produktů které jsou již načteny pomocí této funkce. Jako hlavní účel má za úkol při stisknutí tlačítka vkládat zcela nové produkty pro funkci lednice. Jelikož v jednodimenzionálních čárových kódech není obsaženo nic jiného než číslo, které je odkazem do databáze, která obsahuje ostatní informace o produktu, tak je tato funkce potřeba pro funkci načítání produktů do lednice.

Funkce načte čárový kód z funkce „detection“ a pomocí „Tkinter“ vytvoří 3 tlačítka, text pro informace o načtení a instrukce a textové pole sloužící pro vložení názvu pro načtený čárový kód.

A screenshot of a Tkinter window with a light gray background. At the top, it says 'Zadejte jméno nového produktu' in a small font. Below that, it says 'načtený kod: 20346355'. There is a white text input field. Below the input field are three buttons: 'vložit produkt' (light blue), 'načíst znovu' (light blue), and 'Storno' (red).

Obrázek 15 Ukázka funkce „vlozit_bar_dtb“

Zdroj: Vlastní

Během procesu načítání a psaní názvu jsou pro vkládání, odebírání a návrh receptu vypnuta pomocí funkce „diss_but“.

13.4.7 Vložit

Jednoduchá funkce pro funkčnost vkládání do databáze pomocí:

```
dtb_code.add_new_bar(name, result_code)
```

Převezme načtený kód z funkce „detection“ v proměnné „result_code“ a proměnnou „name“ z textového pole vytvořeného pomocí Tkinter.

13.4.8 Znovu

Funkce slouží vytvoření tlačítka pro případ, že během funkce „vlozit_bar_dtb“ dojde ke špatnému načtení čárového kódu, který se zobrazí po načtení jako text. Když k tomuto případu dojde je možné stlačit tlačítko „nacist znovu“, které stornuje načtení a znovu spustí funkci „detection“ pro nové načítání a přepsání načteného kódu.

Tento proces je důležitý, jelikož pokud dojde ke špatnému načtení a uložení čárového kódu do databáze čárových kódů, tak následné správné načítání čárového kódu produktu pomocí tlačítka „vlozit“ vede k tomu, že databáze nenajde čárový kód ve své tabulce a produkt nelze načíst. Špatně načtený kód se musí pomocí konzole MySQL nebo softwaru VS Code ručně odstranit nebo přepsat. Účel tohoto tlačítka je proto velmi důležitý pro správné načtení čárového kódu a funkci celého systému.

13.4.9 Storno

Funkce obsahující tvorbu a zobrazení jednoduchého tlačítka pro stornování procesu po načtení čárového kódu pomocí funkce „vlozit_bar“ a vrácení do stavu hlavní nabídky.

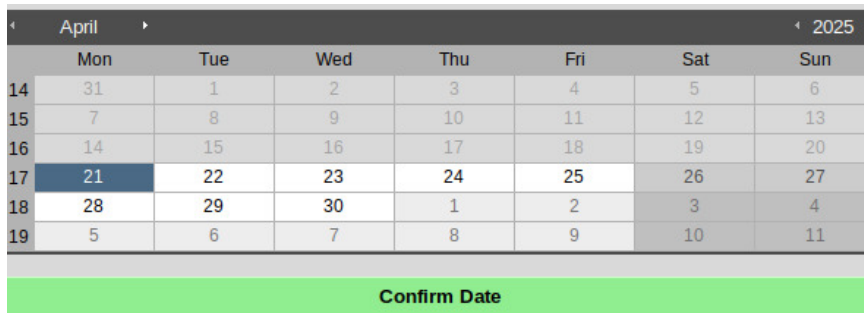
13.4.10 Select_date

Vytváří a zobrazuje kalendář pro výběr datumu expirace pro následující funkci „vlozit_item“.

13.4.11 Vložit_item

Funkčnost je stejná jako u předchozí funkce „vlozit_bar_dtb“ s tím rozdílem že namísto zadávání názvu pro nový produkt již má načtený čárový kód z funkce „detection“ převzat z databáze čárových kódů název načteného produktu. Také se vytvoří a zobrazí pomocí

„Tkinter-calendar“ kalendář, pomocí kterého se volí datum expirace načteného produktu. V kalendáři nelze zvolit datum, které je menší než aktuální datum.



	Mon	Tue	Wed	Thu	Fri	Sat	Sun
14	31	1	2	3	4	5	6
15	7	8	9	10	11	12	13
16	14	15	16	17	18	19	20
17	21	22	23	24	25	26	27
18	28	29	30	1	2	3	4
19	5	6	7	8	9	10	11

Confirm Date

Obrázek 16 Kalendář a tlačítko potvrzení

Zdroj: Vlastní

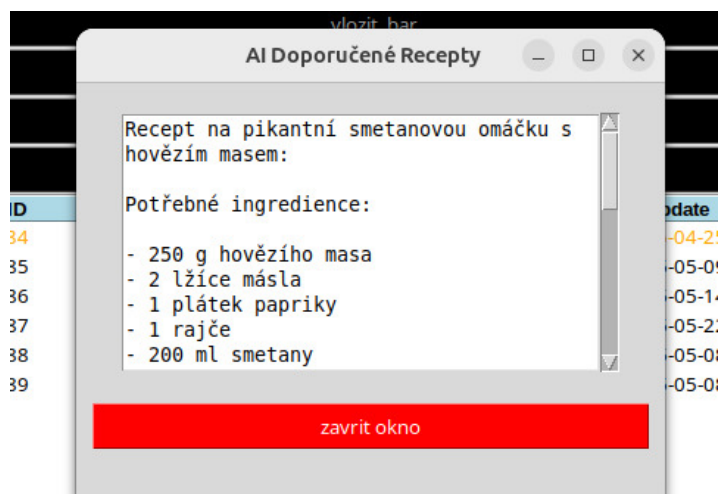
13.4.12 Odebrat

Funkce má za cíl odebírat produkty z listu produktů vložených v lednici. Produkty jsou vypsané v listu, který umožňuje označování řádků po stisknutí na řádek. Funkce neudělá nic, pokud je tlačítko „odebrat“ stlačeno, když není žádný řádek označen.

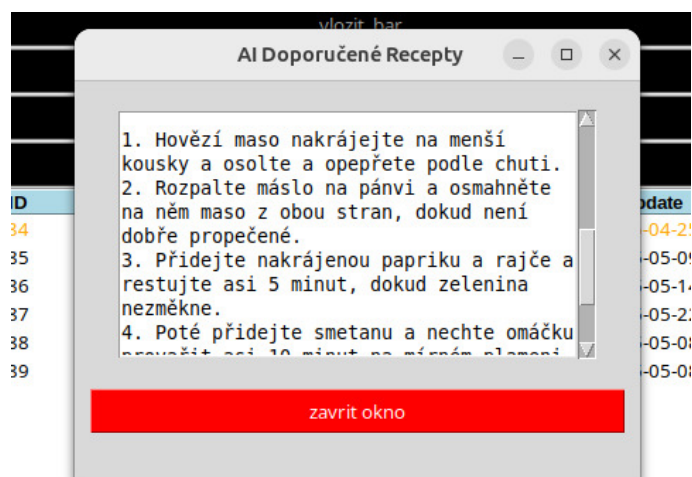
Pokud je ale libovolný řádek označen a stlačeno tlačítko „odebrat“, tak pomocí ID se vyhledá v databázi produktů v lednici produkt, který je následně smazán z tabulky a odebrán ze zobrazení v listu.

13.4.13 Recepty

Funkce sloužící pro volání funkcí ze souboru „AI“, které uloží do proměnné recept text, který umělá inteligence vygenerovala jako výsledek promptu na vytvoření receptu. Tento text je pak vložen do nového vyskakovacího okna vytvořeného pomocí „Tkinter“ ve kterém je tlačítko a posouvací textové pole obsahující hodnotu proměnné „recept“. Toto textové pole obsahuje celý recept, gramáž produktů potřebnou pro vaření a postup přípravy jídla.



Obrázek 17 Ukázka vypsání receptu 1

Zdroj: Vlastní

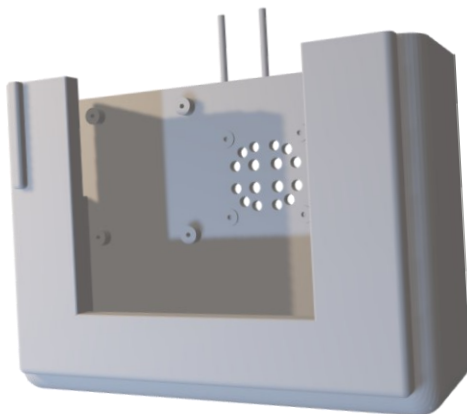
Obrázek 18 Ukázka vypsání receptu 2

Zdroj: Vlastní

Tlačítko slouží pro jednodušší zavírání vyskakovacího okna a je zabarveno červeně. Účel této funkce je mířen na usnadnění každodenního života. Funkce může být jednoduše přeprogramována pomocí změny promptu, který se posílá umělé inteligenci na věci jako jsou návrhy koktejlových nápojů, jídel jen bez masa receptů na pečení a mnoho dalších využití.

14 KRYT

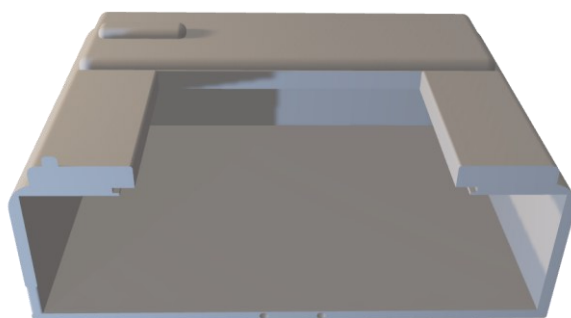
Kryt pro displej, kameru, RPI, kabeláž a větrák na chlazení je vytvořen v softwaru SolidWorks 2024 za účelem efektivního používání a vzhledu chytrého systému. Kryt obsahuje držáky a otvory pro RPI 5 a otvor na straně pro připojení napájení. Na dnu krytu se také nachází držáky a otvory pro upevnění větráku pro chlazení.



Obrázek 19 Spodní model krytu

Zdroj: Vlastní

Spodní i vrchní část krytu systému také obsahuje drážky pro zasunutí displeje, do kterých se displej jednoduše zasune a drážky drží displej, aby se nepropadl do vnitřní části krytu.

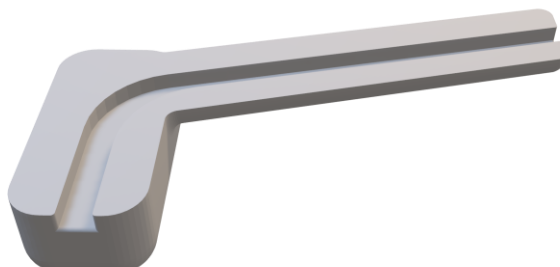


Obrázek 20 Vrchní část krytu

Zdroj: Vlastní

Spojení obou částí vznikne celek, který má vzhled lednice. Vytvořený celek je možné umístit kdekoliv v domácnosti.

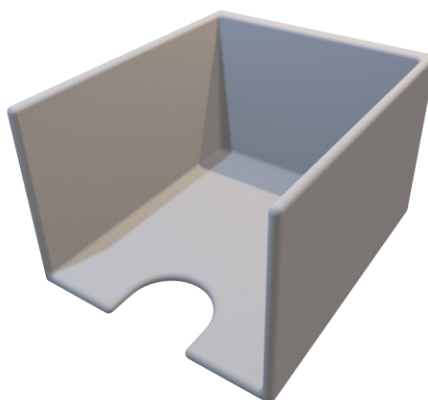
Ke krytu pro RPI, displej a větrák byl také navržen a vytištěn kryt pro kameru skládající se ze 2 částí s otvory pro vložení kamery. Celý kryt na kameru je pak slepen k sobě lepidlem, aby připomínal ruční skener čárových kódů v běžných obchodech.



Obrázek 21 Model kamerového skeneru

Zdroj: Vlastní

Pro tento kamerový skener byl také vytištěn držák, do kterého po vytištění byl lepidlem přilepen také magnet na dno držáku pro snadnější uchycení skeneru do držáku. Ke kamerovému skeneru na spodní stranu jeho „hlavičky“ byl také následně přilepen magnet s orientací opačného pólu směrem dolů jako je magnet na držáku pro zaručení magnetického držení kamerového skeneru v držáku.



Obrázek 22 Model držáku na kamerový skener

Zdroj: Vlastní

15 ROZŠÍŘENÍ

15.1 Detekce objektů bez čárového kódu

Většina produktů, které se prodávají v obchodech má na sobě čárový kód a datum expirace, avšak některé produkty, které se prodávají na kusy, čárový kód nemají. Produkty prodávané na váhu čárový kód získají dodatečně při vážení nebo přímo na pokladně.

Pro účel přidávání produktů bez čárového kódu do systému lednice byl navržen způsob, který využívá vizuální identifikaci pomocí kamery. Pro tyto účely již existují způsoby, jak detekovat zeleninu či ovoce, avšak tyto modely detekce vyžadují vysoký výpočetní výkon pro zpracování obrazu a detekci objektů v reálném čase. Při testování dostupných přednaučených modelů mělo RPI velké potíže s výkonem, konkrétně při inferenci, tj. využití knihoven potřebných k detekci objektu v obraze z kamery. Pro detekci jediného snímku se objevovala přibližně jedna a půl sekundová mezera mezi snímky z kamery.

```
0: 480x640 (no detections), 1449.6ms  
Speed: 5.6ms preprocess, 1449.6ms inference, 1.3ms postprocess per image at shape (1, 3, 480, 640)
```

Obrázek 23 Ukázka detekce s 1,5 s mezerou mezi snímky

Zdroj: Vlastní

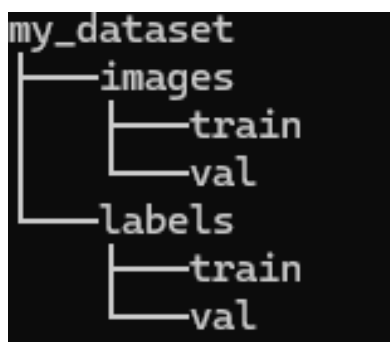
Po testech různých modelů byl vybrán nejlepší model, který ale měl stále čtyři sekundy zpoždění mezi snímky. Tento model nebyl přesný, jelikož snímky, na kterých byl trénován byly pořizovány v jiných světelných podmínkách a také tvarově a barevně jiných objektech, než byly použity při testování detekce.

Proto byl implementován způsob pomocí serveru vytvořeném pomocí služby Colab, který má mnohem větší výkon než RPI, na kterém běží program pro detekci. Tento server dostane snímky z kamery a následně ve snímku pomocí modelu vyhodnotí předměty nacházející se ve snímku z kamery. Po vyhodnocení server zašle odpověď, jaký objekt byl detekován.

Tento způsob se ukázal jako efektivní řešení problému s výkonem RPI a také jako přijatelné řešení pro další kroky postupu pro vizuální detekci produktů bez čárového kódu.

Tyto přednaučené modely ale nebyli velmi přesné a často se objevovala chyba v detekci předmětu ze snímku kamery, a proto byl zvolen jako další postup trénování vlastního modelu.

Pro detekci byl z knihovny „ultralytics“ zvolen model „yolov8“ pro který byl následně trénován vlastní model pro detekci produktů bez čárového kódu. Trénování modelu probíhá pomocí dodání modelu „yolov8“ velké množství fotek objektů, které požadujeme detekovat a také textové soubory obsahující takzvaný „label“ (štítek) objektu ve fotce. Pro úvodní trénování byla pro začátek zvolena mrkev, jablko a vejce. Na trénink modelu pomocí „ultralytics“ je vyžadován velmi specifický způsob dodání fotek a štítků pro trénink. Musí být vytvořena specifická struktura složek a souboru pro správnou funkčnost tréninku. Nejdříve musí být vytvořena složka s libovolným názvem která slouží jako takzvaný „dataset“ ostatních složek. Pro tento účel byla vytvořena složka „my_dataset“ která obsahuje složku „images“, složku „labels“ a soubor typu „.yaml“ s názvem „data“.



Obrázek 24 Struktura složek ke trénování modelu

Zdroj: Vlastní

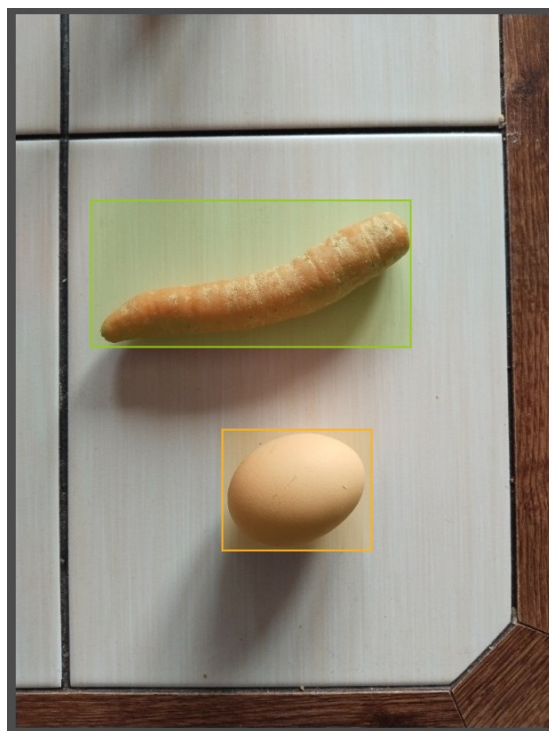
Složky „labels“ a „images“ musí nadále obsahovat další 2 složky s názvy „train“ a „val“. Tyto složky obsahují trénovací fotky a jejich příslušné štítky a validační fotky a jejich příslušné štítky. Fotky by měli být typu .jpg nebo .png. Fotky slouží pro model jako plátno s pixely s různými barvami a štítky jako rozměry požadovaných objektů pro detekci. Labels musí mít určitou strukturu pro správné označení objektu ve fotce. Tato struktura je:

0 | 0.528227 | 0.372976 | 0.567870 | 0.407223

Id labelu | střed X | střed Y | šířka | délka

Tento štítek musí být tvořen ručně, za pomoci asistenčních softwarů nebo webových aplikací. Pro tento úkon byl zvolen web www.makesence.ai, který sloužil k vytvoření tzv. štítku každé fotky pro detekci. Tvoření štítků fotek ale musí být prováděno manuálně pro každou fotku, takže tento způsob je velmi zdoluhavý. Použitý web má jednoduché uživatelské prostředí, které ulehčuje práci při tvoření štítků. Tento proces funguje tak, že fotky jsou na web vloženy a také jsou na webu ručně vytvořeny třídy štítků podle potřeby.

Následně se na každé fotce musí požadovaný objekt detekce obkreslit obdélníkem co nejpřesněji. Obkreslování pomocí nástroje „laso“ je mnohem přesnější způsob pro následnou detekci, avšak model „yolov8“ neumožňuje štítek v tomto formátu, proto byl zvolen obdélník.



Obrázek 25 Ukázka tvoření štítku pomocí www.makesence.ai

Zdroj: Vlastní

Po vytvoření obdélníku okolo každého požadovaného objektu ve fotce se k obdélníku přiřadí třída neboli „class“ objektu. V tomto případě obdélník okolo mrkve jako „třída: mrkev“ a pro další objekty jejich vlastní třída. Tyto třídy pak mají v souboru TXT štítků vlastní ID, podle kterého model následně detekuje, o jaký objekt se jedná. Je důležité, aby tyto třídy štítků byli správně pro zaručení správného fungování detekce pomocí modelu. Tento proces tvoření obdélníku a přiřazení třídy musí proběhnout u každé vložené fotky. Webová stránka makesence.ai má usnadňující funkci, a to je exportovat štítky fotek zabalené v souboru ZIP v požadovaném formátu pro funkci trénování pomocí „yolov8“. Tyto soubory s příponou TXT obsahující štítky fotek musí mít stejný název jako fotka pro kterou byl štítek tvořen, aby se zajistilo správné odkázání štítku na fotku.

Následně jsou příslušné fotky a štítky rozděleny do složek train a val pro následné trénování. Tyto soubory by ještě ale měly být před trénováním samotného modelu správně rozděleny do dvou složek. Složka „train“ slouží modelu pro samotné trénování na fotkách a

složka „val“ slouží modelu jako reference na validaci. Tyto fotky a jejich příslušné modely by ale měly být náhodně vybrány. Poměry rozdělení fotek do těchto složek se může lišit na základě velikosti datové sady/datasetu, tj. počtu fotek určených pro trénování modelu. Možnosti mohou být:

- 90% train a 10% val pro přibližně 10–30 trénovacích fotek
- 80% train a 20% val pro přibližně 50–100 trénovacích fotek
- 70% train a 30% val pro přibližně 100+ trénovacích fotek

Tyto poměry zaručí vyváženější trénink modelu a přesnější detekci objektů. Také je důležité, aby v případě více tříd objektů byli reprezentováni všechny třídy, které se mohou vyskytnout při detekci, ve složce „val“.

Pro účely této práce bylo nafoceno přes 300 fotografií mrkve, vejce a jablka, a pro tyto fotky byli následně pomocí webové stránky makesense.ai vytvořeny štítky. Následně byl zvolen poměr souborů ve složkách 70% train a 30% val. Jakmile byli fotky rozděleny do jejich příslušných souborů byl vytvořen ve složce „my_dataset“ soubor „data.yaml“, který je potřebný, aby model věděl, jaký objekt je přiřazen k příslušnému štítku. Obsah souboru musí být také velmi specifický, jelikož model pomocí tohoto souboru také rozeznává, z jaké složky si přebírá fotky na trénování a na validaci.

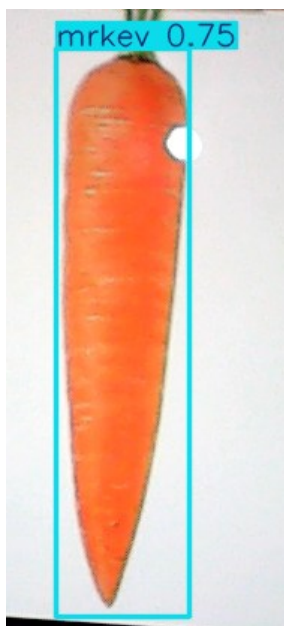
```
path: C:/Users/Ryzen/Desktop/projekt_detekce/my_dataset
train: images/train
val: images/val
nc 3:
names: ['jablko', 'mrkev', 'vejce']
```

V souboru lze vidět, že modelu se musí přiřadit cesta, kde se nacházejí fotky. Musí být také popsány třídy pro detekci, kde v tomto souboru je jim přiřazen název, který se následně bude zobrazovat u detekovaného objektu. Také musí být specifikováno jaký je počet tříd, které se mají detekovat pomocí „nc 3:“. Jelikož byly všechny soubory nastaveny tak jak je potřeba, mohl se zahájit trénink modelu. Samotné trénování bylo také provedeno v softwaru Visual Studio Code. Ke trénování modelu je také potřeba specifikovat velikosti fotek pomocí „imgsz“, na jakou velikost se fotky mají během trénování upravit. Dále proměnná „batch“, která modelu říká že se má trénovat na 8 fotkách najednou což celý proces zrychlí. Důležitý příkaz je také proměnná „epochs“, která modelu a funkci train() říká, že má provést trénink na padesáti opakováních.

```
from ultralytics import YOLO
model = YOLO('yolov8n.pt')
model.train(
    data =
    r'C:\Users\Ryzen\Desktop\projekt_detekce\my_dataset\data.yaml',
    epochs=50,
    imgsz=640,
    batch=8,
    name='yolov8-moje'
)
```

Pomocí tohoto kódu byl spuštěn trénink, který zabral přibližně 40 minut na padesát opakování tréninku po osmi fotkách najednou. Finální model se uložil jako „yolov8-moje“ a pro detekci byl použit soubor „best.pt“ vytvořen uvnitř tohoto modelu.

Model byl testován na fotkách zobrazených pomocí mobilního zařízení, kde model díky osvětlení obrazovky a kvalitním fotkám na google.com detekoval správně při přibližně 70-80% případech.



Obrázek 26 Ukázka detekce mrkve zobrazené na mobilním zařízení

Zdroj: Vlastní

Následně byl model testován přes živé video na reálných objektech, kde detekce upadala na malé procento a kvůli světelným podmínkám a natočení předmětů, detekce byla často chybná a nepřesná. Toto může být způsobeno nedostatečnými pozicemi nafocení předmětů pro trénování a nedostatečným osvětlením předmětů.



Obrázek 27 Ukázka špatné detekce vejce

Zdroj: Vlastní

Odstranění této chyby se může realizovat pomocí lepšího nafocení fotek pro trénink a nafocení většího počtu fotek. Pro tyto fotky se ale musí vytvořit štítky a tento proces může být zdoluhavý. Také pro přidání nových objektů pro detekci musí být nafoceny a vytvořeny nové štítky a model nanovo natrénován. Při tomto počtu fotek by se model trénoval hodiny, a proto se tato metoda detekce implementovala jen pomocí tlačítka pro zkoušení a není určena pro klasické použití pro detekci ovoce a zeleniny. Jelikož zelenina a ovoce ztrácí chuť při nízkých teplotách tak je tato funkce jen pro rozšíření a testování.

15.1.1 Funkce `detection_yolo()`

Pomocí kamerového systému se přenáší snímky na cloudový server, na kterém se detekují produkty ve snímcích. Tento server následně zašle zpět data o detekci a uloží je do proměnné „detections“.

15.1.2 Funkce `vlozit_yolo`

Tato funkce vypíná tlačítka při detekci a pomocí proměnné „detections“ z funkce pro detekci pomocí modelu yolo přes cloud ukládá jen název detekovaného produktu s největším procentem jistoty detekce.

15.2 Detekce pomocí QR kódu

Jelikož klasické jednodimenzionální čárové kódy běžně používané na produktech v obchodech (jako je EAN) neobsahují nic jiného než číselný kód, který se odkazuje po načtení na databázi, která obsahuje ostatní informace o produktu, byla navržena funkce detekce produktů pomocí QR kódu. Tato funkce má za účel ukázkou vylepšení správy produktů v obchodech a zlepšení načítání dat o produktech.

Načítání informací produktů, které mají vytvořen QR kód ve správném formátu je mnohem efektivnější pro skladování a sbírání informací o produktech, ale jelikož dnešní produkty v obchodech neobsahují na obalech QR kódy tak z tohoto důvodu bylo navrženo generování vlastních QR kódu, které obsahují data o produktu ve specifickém formátu pro následné snadné dekódování textu obsaženém v QR kódu.

Detekce probíhá pomocí python scriptu, který využívá funkce knihovny „OpenCV“, která již obsahuje funkci pro detekci QR kódů.



Obrázek 28 Vlastně generovaný QR kód pro produkt: smetana

Zdroj: Vlastní

Tento testovací QR kód obsahuje text, který popisuje produkt „smetana“ ve formátu „název|datum expirace ve formátu ROK-MESÍC-DEN|EAN PRODUKTU“ a obsahuje informace jako je název produktu, datum expirace produktu a jeho čárový kód. Jednotlivé slova jsou odděleny znakem „|“ pro následné jednoduché rozdělení textu do jednotlivých slov, které se mohou uložit do vlastních proměnných. S těmito proměnnými lze pak pracovat stejným způsobem jako kdyby si je systém bral z databáze.

Pro testovací účely byl QR kód vytištěn na papír formátu A4 a následně skenován pomocí vytištěného kamerového skeneru. Detekce, která probíhala v testech měla 100% úspěšnost s přesností čtení QR kódu i za různých světelných podmínek.

Tento způsob rozšíření by v budoucnu za případu návrhu tištění QR kódů na produkty v ochodech velmi zjednodušil evidenci produktů a také uživatelské použití tohoto systému. Funkce pro detekci byla do systému implementována jako možnost načítání pomocí tlačítka v uživatelském rozhraní.

15.2.1 Funkce detection_QR()

Zde probíhá detekce QR kódu pomocí kamery, která když kód detekuje tak kolem něj nakreslí modré pole a jeho data přečte a uloží do proměnné „detections“. Tato proměnná je následně rozdělena na jednotlivé slova pomocí znaku „|“ a jednotlivá slova jsou uloženy do jejich příslušných proměnných. Funkce vrací tyto názvy pro následné ukládání pomocí funkce „vlozit_QR“ do databázového systému lednice.

15.2.2 Funkce vlozit_QR

Jednoduchá funkce, která vypíná tlačítka pro ovládání systému během detekce a volání detekce s následným ukládáním dat, které se detekují. Tyto data jsou následně přidána do databáze systému lednice.

ZÁVĚR

Cílem práce byl návrh a realizace funkčního systému chytré lednice, který umožňuje evidenci potravin v lednici pomocí skenování čárových kódů, vizuální rozlišení produktů podle data expirace, upozornění na blížící se vypršení data expirace a návrh receptů podle aktuálního obsahu lednice. V rámci řešení byl využit mikropočítač Raspberry Pi, kamera a dotykový displej. Tyto komponenty byly následně uloženy do krytu, který byl pro tyto účely modelován a vytištěn pomocí 3D tisku. Pro softwarové funkce byly použity různé knihovny, zejména OpenCV, Tkinter a databázový systém MySQL. Funkce návrhů receptů využívá komunikaci s jazykovým modelem od společnosti OpenAI.

Systém byl úspěšně navržen, implementován a otestován. V praxi může tento systém sloužit jako užitečný nástroj pro evidenci potravin v domácnosti a může přispět ke snížení plýtvání potravinami.

Systém je připraven do budoucna, v práci se uvažuje využití QR kódů na obalech produktů, které by kromě identifikace obsahovaly i informaci o datu expirace, což by výrazně usnadnilo použití tohoto systému v domácnosti. Tato funkce však nemůže být v dnešní době plně využita, protože běžně prodávané produkty zatím tyto navržené QR kódy na obalech neobsahují.

Do systému bylo také implementováno vizuální rozpoznání pro produkty bez čárového kódu, jako je například nebalená zelenina a ovoce. K tomuto byl vytvořen a natrénován vlastní yolov8 model, robustnějším trénováním lze zlepšit detekci různých objektů.

Konečná verze systému je plně funkční a je připravena k použití v běžné domácnosti.

- [1] PECINOVSKÝ, Rudolf. *Python: Kompletní příručka jazyka pro verzi 3.11*. Grada, 2022. ISBN 978-80-271-3891-3. Dostupné také z: <https://www.bookport.cz/AccountSaml/SignIn/?idp=https://shibboleth.utb.cz/idp/shibboleth&returnUrl=/kniha/python-11285/>
- [2] HOWSE, Joseph. *OpenCV Computer Vision with Python*. Packt Publishing, 2013. ISBN 1299545009. Dostupné také z: <https://proxy.k.utb.cz/login?url=https://app.knovel.com/hotlink/toc/id:kpOCVCVPL1/open-cv-computer-vision?kpromoter=marc>
- [3] DONAT, Wolfram. *Learn Raspberry Pi Programming with Python: Learn to Program on the World's Most Popular Tiny Computer*. Second edition. *Technology in action*. [Berkeley, CA]: Apress, [2018]. ISBN 9781484237694. Dostupné z: <https://doi.org/9781484237694>.
- [4] ČÁDA, David. *Využití 2D kódů v běžné praxi*. 2016. Dostupné také z: <http://hdl.handle.net/10563/38447>.
- [5] WELLING, Luke a THOMSON, Laura. *MySQL: průvodce základy databázového systému*. Brno: CP Books, 2005. ISBN 8025106713.
- [6] *MITS Altair 8800*. Online. 2002. Dostupné z: <https://oldcomputers.net/altair-8800.html>. [cit. 2025-03-24].
- [7] UNIVERSITY OF TECHNOLOGY. *INTRODUCTION TO COMPUTER AND MICROCOMPUTERS*. Online. Dostupné z: <https://www.uotechnology.edu.iq/depeeee/lectures/4th/Electronic/Microprocessor%20engineering%202/part1.pdf>. [cit. 2025-03-24].
- [8] VETR, Michal. *Systém evidence zboží s využitím QR kódů*. Vedoucí Malaník, David. Zlín: Univerzita Tomáše Bati ve Zlíně. Fakulta aplikované informatiky, Ústav elektroniky a měření, 2017. Dostupné také z: <http://hdl.handle.net/10563/41077>.
- [9] SOBOTKA, Zdeněk a STARÝ, Jaroslav. *MIKROPOČÍTAČE*. Online. 1981. Dostupné z: https://sam.speccy.cz/olddocs/mikropocitace_pt1.pdf. [cit. 2025-03-24].
- [10] PALACKÝ, Petr. *Mikropočítačové řídicí systémy I*. Ostrava: Vysoká škola báňská - Technická univerzita, 2007. ISBN 978-80-248-1494-0. Dostupné také z: <http://www.elearn.vsb.cz/archivcd/FEI/MRS1/>.

- [11] MACH, Petr. *PYTHON – programování zábavou*. Online. Dostupné z: <http://python.wraith.cz>. [cit. 2025-03-24].
- [12] RPISHOP. *Raspberry Pi Touch Display*. Online. Dostupné z: <https://rpishop.cz/lcd-oled-displeje/243-7-oficialni-kapacitni-lcd-displej-800x480.html>. [cit. 2025-03-24].
- [13] RPISHOP. *Arducam 2Mpx IMX291*. Online. Dostupné z: <https://rpishop.cz/usb-kamerove-moduly/2517-arducam-2mpx-imx291-usb-low-light-160-camera-module.html>. [cit. 2025-03-24].
- [14] UDEMY. *What is MySQL? A Beginner's Guide to SQL Databases*. Online. Dostupné z: <https://blog.udemy.com/what-is-mysql/>. [cit. 2025-03-24].
- [15] GOENGINEER. *The History of SOLIDWORKS*. Online. Dostupné z: <https://www.goengineer.com/blog/history-of-solidworks>. [cit. 2025-03-24].
- [16] *The story of Ubuntu*. Online. Dostupné z: <https://ubuntu.com/about>. [cit. 2025-03-24].
- [17] MACH, Petr. *ÚSL úvod do systému Linux*. Online. 2004. Dostupné z: <http://usl.wraith.cz/usl.html>. [cit. 2025-03-24].
- [18] ITNETWORK. *Lekce 1 - Úvod do operačního systému Linux*. Online. Dostupné z: <https://www.itnetwork.cz/linux/zaklady/uvod-do-operacniho-systemu-linux>. [cit. 2025-03-24].
- [19] RPISHOP. *Raspberry Pi 5*. Online. Dostupné z: <https://rpishop.cz/raspberry-pi-5/6498-raspberry-pi-5-8gb-ram.html>. [cit. 2025-03-24].
- [20] *ENDOSKOP INSPEKČNÍ KAMERA*. Online. Dostupné z: https://allegro.cz/nabidka/endoskop-inspekcní-kamera-usb-c-5m-led-usb-android-3v1-16495104962?bi_s=ads&bi_m=showitem:desktop:top:active&bi_c=MDIyNjgxYWQQtNzA5OS00ZDg4LTlkYmMtMWU2ZmJmM2NiYWYyAA&bi_t=ape&referrer=proxy&emission_unit_id=3f96f602-d44b-47c5-bc69-022c18daed3c. [cit. 2025-05-10].

SEZNAM OBRÁZKŮ

Obrázek 1 Vizualizace čtení čárového kódu.....	15
Obrázek 2 Altair 8800.....	18
Obrázek 3 Raspberry Pi 5 8GB.....	20
Obrázek 4 Logo hada jazyku python	22
Obrázek 5 Obsah balení displeje Raspberry Pi Touch Display	24
Obrázek 6 Kamera Arducam 2Mpx IMX291	25
Obrázek 7 Endoskop inspekční kamera.....	26
Obrázek 8 Americká lednice LG GSGV80PYLD	27
Obrázek 9 Solid Works 1995	28
Obrázek 10 Schéma zapojení.....	30
Obrázek 11 Ukázka zaslaného emailu	39
Obrázek 12 Ukázka Onboard klávesnice.....	42
Obrázek 13 Ukázka zobrazení produktů v listu.....	43
Obrázek 14 Načítání čárového kódu.....	44
Obrázek 15 Ukázka funkce „vlozit_bar_dtb“	44
Obrázek 16 Kalendář a tlačítko potvrzení	46
Obrázek 17 Ukázka vypsání receptu 1	47
Obrázek 18 Ukázka vypsání receptu 2	47
Obrázek 19 Spodní model krytu	48
Obrázek 20 Vrchní část krytu	48
Obrázek 21 Model kamerového skeneru	49
Obrázek 22 Model držáku na kamerový skener	49
Obrázek 23 Ukázka detekce s 1,5 s mezerou mezi snímky	50
Obrázek 24 Struktura složek ke trénování modelu	51
Obrázek 25 Ukázka tvoření štítku pomocí www.makesence.ai	52
Obrázek 26 Ukázka detekce mrkve zobrazené na mobilním zařízení	54
Obrázek 27 Ukázka špatné detekce vejce.....	55
Obrázek 28 Vlastně generovaný QR kód pro produkt: smetana	56

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

RPI Raspberry Pi

