

**Slovenská technická univerzita v Bratislave**  
**Fakulta informatiky a informačných technológií**  
**Ilkovičova 2, 842 16 Bratislava 4**

**Predmet**

**– Mobilné technológie a aplikácie –**

**- Milestone 2 -**

**Inteligentná chladnička**

**Ak. Rok : 2021/2022, letný semester**

**Cvičiaci:**  
**Ing. Miroslav Bahleda, PhD.**

**Študenti:**  
**Martin Rudolf 97029**  
**Michael Absolon 96650**

# Obsah

[Návrh databázy](#)

[Návrh obrazoviek](#)

[API](#)

[Public](#)

[Private](#)

[Testy](#)

[Backend akceptačné testy](#)

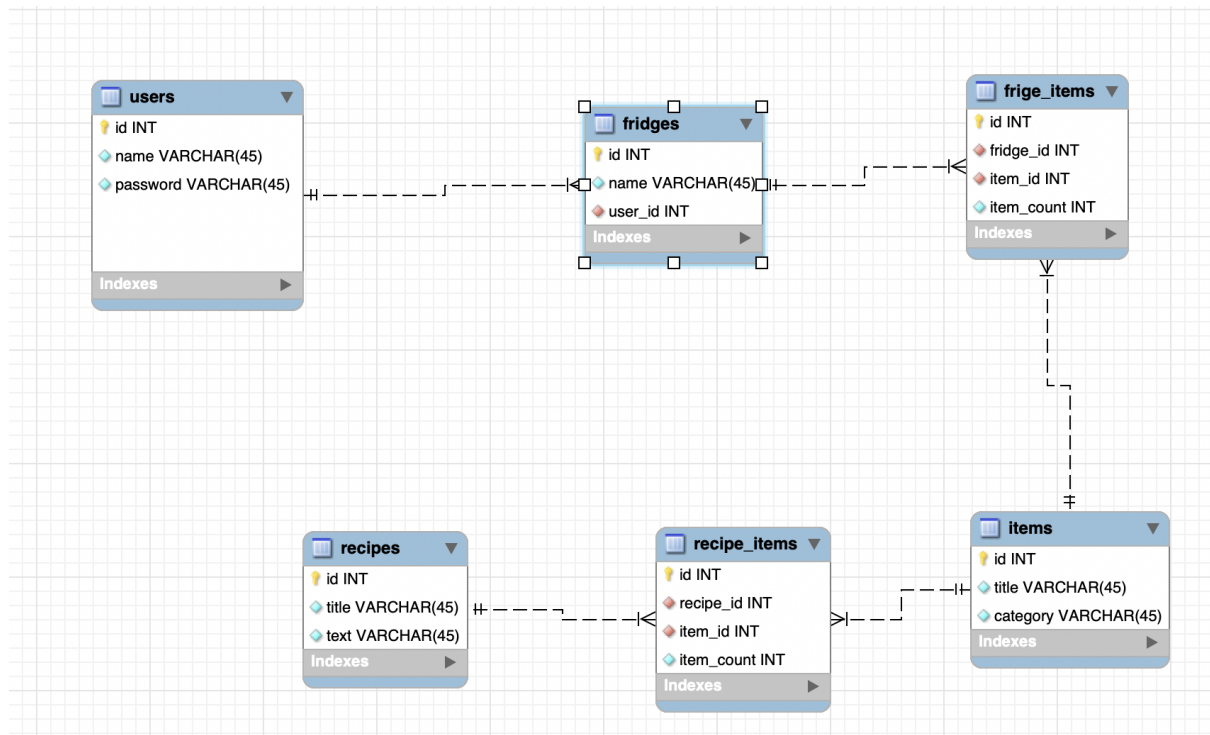
[Frontend akceptačné testy](#)

[Backend - Implementačný opis](#)


[Changelog](#)

[Repozitár](#)

## Návrh databázy



## Návrh obrazoviek



Welcome to intelligent refrigerator app!

We can help you organize your fridge and recommend meaningful recipes.

Get started →

### Login

Email

Password

Login

### Create a refrigerator

Name

Continue →

## Recommended recipes

### Chicken nuggets

2x chicken breast  
100g breadcrumbs  
2x egg

More

### Vegetable salad

2x tomatoes  
1x cucumber  
150g ham

More



Search for food



Apple

Add 1



Ham

Add 1



## Chicken nuggets

2x chicken breast  
100g breadcrumbs  
2x egg

1. In a plastic bag, mix the flour, seasoned salt and some salt and pepper. In a bowl, mix the eggs with 2 tablespoons water. In a separate bowl, add the breadcrumbs.

2. Shake the chicken pieces in the bag to coat. In batches, dip them in the egg wash, then in the breadcrumbs.

3. Heat about 2 inches of vegetable oil in a large skillet over medium-high heat until a few breadcrumbs dropped in the oil start to sizzle immediately. Fry the chicken, in batches if needed, until golden brown and cooked through, a couple minutes per side. Drain on a paper towel and serve with ketchup.

Done!

Note: This will remove all used ingredients from your virtual refrigerator

## Ingredients

Apple

x10



Milk bottles

x10



Potatoes

x10



Add food



## Options

### Switch refrigerator

Michael's refrigerator

select

delete

Martin's refrigerator

select

delete

Add refrigerator



# API

## Public

/login - POST - Log in user. Create a new login session if necessary. Accepts email and password.

/register - POST - Register a new user. Accepts email, first name, password.

/refrigerator/<ID>/recipes - GET - Returns paginated food recipes for specific refrigerator

/refrigerators/<ID>/food - GET - Returns all food in the specific refrigerator

/refrigerators/<ID>/food - POST - Allows adding new food to user's refrigerator

Example body: {"item\_id": 123,  
                  "item\_count": 12}

/refrigerators/<ID>/food/<ID> - PATCH - Allows modifying count of existing food

Example body: {"item\_count": 20}

/refrigerators/<ID>/food/<ID> - DELETE - Allows deleting food from user's refrigerator

/users/<ID>/refrigerators - POST - Allows adding a new refrigerator

Example body: {"name": "misova"}

refrigerators/<ID> - DELETE - Allows deleting a refrigerator

/users/<ID>/refrigerators - GET - Returns all refrigerators of the specific user

/foodTypes - GET - Returns all available food types

## Private

/foodTypes - POST - Allows adding a new food type

Example Multipart-Form: {"file": image.png  
                          "title": "hrasok"  
                          "category": "zelenina"}

/recipes - POST - Creates a new recipe

Example body: {"title": "hraskove pyre",  
                  "text": "spravime hraskove pyre",  
                  "items": [  
                      {  
                          "id": 6,  
                          "item\_count": 7  
                      }  
                  ]  
          }

# Testy

## Backend akceptačné testy

| Test 1: Úprava počtu kusov jedného typu jedla v chladničke |  |
|--|--|
| Vstupné podmienky:   | <ul style="list-style-type: none"><li>Máme k dispozícii platné ID používateľa</li></ul>  |
| Výstupné podmienky:  | Počet kusov jedného jedla v chladničke sa zmení  |
| Postup:  | <ol style="list-style-type: none"><li>Aplikácia zavolá HTTP GET /users/&lt;id&gt;/refrigerators pričom poskytne platné id používateľa</li><li>Volanie vráti kód 200 OK a v response je zoznam chladničiek, ktoré patria tomuto používateľovi. Každá položka zoznamu obsahuje ID a meno každej chladničky.</li><li>Aplikácia zavolá HTTP GET /refrigerators/&lt;ID&gt;/food pričom poskytne platné id chladničky z predchádzajúceho volania.</li><li>Volanie vráti kód 200 OK a v response bude obsah chladničky vo forme zoznamu jedál, kde každá položka zoznamu bude obsahovať ID, category, item_count a title.</li><li>Aplikácia zavolá HTTP PATCH /refrigerators/&lt;ID&gt;/food/&lt;ID&gt; pričom poskytne platné ID chladničky a platné ID jedla, pre ktoré chce upraviť počet kusov. Do body zada item_count, ktorý sa nastaví na zmenený počet kusov.</li><li>Volanie vráti kód 204 NO CONTENT.</li></ol> |
| Výsledok: PASS / FAIL                                      |  |

| Test 2: Pridanie jedného jedla do chladničky |  |
|--|--|
| Vstupné podmienky:                           | <ul style="list-style-type: none"><li>Máme k dispozícii platné ID používateľa</li></ul>                          |
| Výstupné podmienky:                          | Do chladničky je pridané nové jedlo  |
| Postup:                                      | <ol style="list-style-type: none"><li>Aplikácia zavolá HTTP GET /users/&lt;id&gt;/refrigerators pričom</li></ol> |

|                              |  |
|------------------------------|--|
|                              | <p>poskytne platné id používateľa</p> <ol style="list-style-type: none"> <li>2. Volanie vráti kód 200 OK a v response je zoznam chladničiek, ktoré patria tomuto používateľovi. Každá položka zoznamu obsahuje ID a meno každej chladničky.</li> <li>3. Aplikácia zavolá HTTP GET /foodTypes</li> <li>4. Volanie vráti kód 200 OK a v response je zoznam typov jedla. Každá položka zoznamu obsahuje ID typu jedla.</li> <li>5. Aplikácia zavolá HTTP POST /refrigerators/&lt;id&gt;/food pričom poskytne platné uuid chladničky z predchádzajúceho volania a do body sa vloží ID typu jedla a počiatočný počet kusov.</li> <li>6. Volanie vráti kód 201.</li> </ol> |
| <b>Výsledok: PASS / FAIL</b> |  |

| <b>Test 3: Pridanie novej chladničky</b> |   |
|--|---|
| <b>Vstupné podmienky:</b>                | <ul style="list-style-type: none"> <li>• Máme k dispozícii platné ID používateľa</li> </ul>   |
| <b>Výstupné podmienky:</b>               | Do zoznamu používateľových chladničiek sa pridá nová chladnička   |
| <b>Postup:</b>                           | <ol style="list-style-type: none"> <li>1. Aplikácia zavolá HTTP POST /users/&lt;id&gt;/refrigerators pričom poskytne platné ID používateľa a do body pošle vloží meno chladničky.</li> <li>2. Volanie vráti kód 201 CREATED a response bude prázdny.</li> </ol> |
| <b>Výsledok: PASS / FAIL</b>             |   |

| <b>Test 4: Pridanie jedného jedla do chladničky (negative test)</b> |   |
|---|---|
| <b>Vstupné podmienky:</b>   | <ul style="list-style-type: none"> <li>• Máme k dispozícii platné ID používateľa</li> </ul> |
| <b>Výstupné podmienky:</b>  | Obsah chladničky sa nezmení   |

|                              |   |
|------------------------------|---|
| <b>Postup:</b>               | <ol style="list-style-type: none"> <li>1. Aplikácia zavolá HTTP POST /refrigerators/&lt;id&gt;/food v body uvedie neplatné ID typu jedla.</li> <li>2. Volanie vráti kód 400 bad request.</li> </ol> |
| <b>Výsledok: PASS / FAIL</b> |   |

| <b>Test 5: Pridanie novej chladničky (negative test)</b> |   |
|--|---|
| <b>Vstupné podmienky:</b>                                | <ul style="list-style-type: none"> <li>• Máme k dispozícii platné ID používateľa</li> </ul>   |
| <b>Výstupné podmienky:</b>                               | Zoznam používateľových chladničiek zostane nezmenený  |
| <b>Postup:</b>   | <ol style="list-style-type: none"> <li>1. Aplikácia zavolá HTTP POST /users/&lt;ID&gt;/refrigerators pričom poskytne platné ID používateľa a do body nepošle nič.</li> <li>2. Volanie vráti kód 400 BAD REQUEST.</li> </ol> |
| <b>Výsledok: PASS / FAIL</b>                             |   |

## Frontend akceptačné testy

| <b>Test 1: Pridanie jedného jedla do chladničky</b> |  |
|---|--|
| <b>Vstupné podmienky:</b>                           | Používateľ je prihlásený do aplikácie  |
| <b>Výstupné podmienky:</b>                          | Zmení sa zoznam jedla v chladničke   |
| <b>Postup:</b>                                      | <ol style="list-style-type: none"> <li>1. Používateľ stlačí tlačidlo search v dolnej lište</li> <li>2. Používateľ si vyberie zo zoznamu jedla alebo vyhľadá pomocou search baru požadované jedlo</li> <li>3. Používateľ si zvolí požadovaný počet daného jedla</li> <li>4. Používateľ klikne na plusko pri danom jedle.</li> </ol> |



|                              |  |
|------------------------------|--|
|                              | 5. Zobrazí sa toast o úspešnom pridaní jedla do chladničky |
| <b>Výsledok: PASS / FAIL</b> |  |

| <b>Test 2: Pridanie novej chladničky</b> |   |
|--|---|
| <b>Vstupné podmienky:</b>                | Používateľ je prihlásený do aplikácie   |
| <b>Výstupné podmienky:</b>               | Do zoznamu chladničiek v nastaveniach sa pridá nová chladnička  |
| <b>Postup:</b>                           | <ol style="list-style-type: none"> <li>1. Používateľ stlačí tlačidlo options v dolnej lište</li> <li>2. Používateľ stlačí tlačidlo "Add refrigerator"</li> <li>3. Používateľ vyplní meno chladničky</li> <li>4. Používateľ stlačí tlačidlo vytvoriť</li> <li>5. Používateľ sa ocitne znova na obrazovke options a vidí svoju novo pridanú chladničku</li> </ol> |
| <b>Výsledok: PASS / FAIL</b>             |   |

| <b>Test 3: Úprava počtu kusov jedného typu jedla v chladničke</b> |   |
|---|---|
| <b>Vstupné podmienky:</b>   | Používateľ je prihlásený do aplikácie   |
| <b>Výstupné podmienky:</b>  | Zmení sa počet daného jedla v chladničke  |
| <b>Postup:</b>  | <ol style="list-style-type: none"> <li>1. Používateľ stlačí tlačidlo chladničky v dolnej lište</li> <li>2. Používateľ si vyberie položku zo zoznamu jedla v danej chladničke</li> <li>3. Používateľ klikne na plusko alebo minusko podľa toho či chce ubrať alebo pridať počet kusov</li> <li>4. Automaticky sa zmení číselko symbolizujúce počet kusov pri danej položke.</li> </ol> |
| <b>Výsledok: PASS / FAIL</b>                                      |   |

|  |
|--|
| <b>Test 4: Pridanie novej chladničky (negative test)</b> |
|--|

|                              |   |
|------------------------------|---|
| <b>Vstupné podmienky:</b>    | Používateľ je prihlásený do aplikácie   |
| <b>Výstupné podmienky:</b>   | Do zoznamu chladničiek v nastaveniach sa nepridá nová chladnička  |
| <b>Postup:</b>               | <ol style="list-style-type: none"> <li>1. Používateľ stlačí tlačidlo options v dolnej lište</li> <li>2. Používateľ stlačí tlačidlo "Add refrigerator"</li> <li>3. Používateľ stlačí tlačidlo vytvoriť</li> <li>4. Chladnička sa nevytvorí, pretože používateľ nevyplnil povinné pole "meno chladničky" a pod vstupom na meno sa zobrazí chybová hláška</li> </ol> |
| <b>Výsledok: PASS / FAIL</b> |   |

|   |   |
|---|---|
| <b>Test 5: Prihlásenie do aplikácie (negative test)</b> |   |
| <b>Vstupné podmienky:</b>                               | Používateľ je registrovaný  |
| <b>Výstupné podmienky:</b>                              | Používateľ sa neprihlási do aplikácie   |
| <b>Postup:</b>  | <ol style="list-style-type: none"> <li>1. Používateľ zadá meno a heslo do textboxov "username" a "password"</li> <li>2. Používateľ klikne na tlačidlo "Log in"</li> <li>3. Používateľovi sa znovu načíta login screen s červeným textom "invalid password or username"</li> </ol> |
| <b>Výsledok: PASS / FAIL</b>                            |   |

# Backend - Implementačný opis

Backend sme sa rozhodli implementovať v programovacom jazyku *Python*. Na implementáciu a vystavenia REST API sme použili webový framework *Flask*. Komunikáciu s databázou zabezpečuje knižnica *SQLAlchemy*, ktorá nám taktiež slúži ako abstraktná vrstva databázy (ORM). Všetky knižnice potrebné na spustenie projektu su vylistované v súbore *requirements.txt*.

Celý projekt je rozdelený do štyroch súborov: *app.py*, *database.py*, *models.py* *database\_functions.py*. V súbore *database.py* je funkcia ktorá inicializuje spojenie s databázou, ktorú volá vo svojej main funkcii súbor *app.py*, ktorý okrem inicializácie implementuje definované endpointy a ich spracovanie. *Model.py* implementuje modely datáby, ktoré definujú navrhnuté vzťahy a atribúty tabuliek. Súbor *database\_functions.py* obsahuje implementované funkcie na komunikáciu s DB, ktoré voláme pri handlovaní requestov v súbore *app.py*.

## Changelog

- Pridaný DELETE endpoint pre odstránenie chladničky
- Aktualizované všetky testy pre backend
- Pre každý endpoint REST API doplnená špecifikácia body
- Zmena relationship medzi users a fridges, z many to many na many to one.
- pridanie endpointu /recipes na pridanie noveho receptu

## Repozitár

<https://github.com/xabsolon/mtaa-backend>