

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Tvorba uživatelských rozhraní  
Specifikace zadání a uživatelských požadavků

9. října 2021

**Lukáš Hais** (xhaisl00)  
**Tadeáš Kachyňa** (xkachy00)  
**Lukáš Neupauer** (xneupa01)

# Obsah

<b>1</b>	<b>Analýzy a návrhy členů týmu</b>	<b>2</b>
1.1	Lukáš Hais	2
1.1.1	Způsob hledání tématu	2
1.1.2	Způsob zkoumání uživatele a jeho potřeby	2
1.1.3	Identifikované potíže	2
1.1.4	Návrh nového řešení	2
1.2	Tadeáš Kachyňa	2
1.2.1	Způsob hledání tématu	2
1.2.2	Způsob zkoumání uživatele a jeho potřeby	2
1.2.3	Identifikované potíže	2
1.2.4	Návrh nového řešení	2
1.3	Lukáš Neupauer	3
1.3.1	Spôsob hľadania témy	3
1.3.2	Spôsob skúmania užívateľa a jeho potreby	3
1.3.3	Identifikované nedostatky	3
1.3.4	Návrh nového riešenia	3
<b>2</b>	<b>Důvod vybrání zvoleného tématu</b>	<b>4</b>
<b>3</b>	<b>Analýza uživatele</b>	<b>4</b>
<b>4</b>	<b>Potřeby uživatele</b>	<b>4</b>
<b>5</b>	<b>Popis současného řešení</b>	<b>4</b>
<b>6</b>	<b>Návrh zadání</b>	<b>5</b>
6.1	Funkcionalita	5
6.2	Grafika	5
<b>7</b>	<b>Návrh řešení</b>	<b>5</b>
7.1	Datový model	5
7.2	API	5
7.2.1	Create	6
7.2.2	Read	6
7.2.3	Update	6
7.2.4	Delete	7

# **1 Analýzy a návrhy členů týmu**

## **1.1 Lukáš Hais**

### **1.1.1 Způsob hledání tématu**

Obepisování blízkých přátel s dotazem, zda nemají nějakou aplikaci u které by věděli o nějakých funkčních/grafických nedostacích. Následně byla 1 aplikace vybrána dle osobních preferencí.

### **1.1.2 Způsob zkoumání uživatele a jeho potřeby**

Sraz s uživatelem a sledování jeho využívání aplikace, doptávání se na očekávanou funkcionalitu. Uživatel požaduje aplikaci, která mu umožní zaznamenávat data z různých oblastí (počet uběhnutých kilometrů, objem vypité vody za den, hodiny strávené učením, . . . ) a následné zobrazení grafů podle časového období, filtrované pomocí kategorií spravovaných uživatelem.

### **1.1.3 Identifikované potíže**

Neintuitivnost samotné aplikace. Chybí kategorie pro položky podle kterých by bylo možné filtrovat. Nepřehlednost zobrazených grafů.

### **1.1.4 Návrh nového řešení**

Znovuvytvoření aplikace s novým, modernějším designem. Přidání chybějících funkcionalit a implementace stávajících.

## **1.2 Tadeáš Kachyňa**

### **1.2.1 Způsob hledání tématu**

Oslovení někoho z mých přátel nebo členů rodiny, zda-li používají nějakou aplikaci u které vědí o nějakých funkčních či grafických nedostacích. Následně byla mnou jedna aplikace vybrána.

### **1.2.2 Způsob zkoumání uživatele a jeho potřeby**

Navštívení vybraného uživatele a sledování jeho každodenní práce s aplikací (konkrétně s aplikací Fitbit App, která slouží pro detailnější zobrazení statistik naměřených wearable zařízení, které je s touto aplikací spárováno pomocí bluetooth). Bylo vypořádováno, že uživatel není zcela spokojen s aktuální aplikací, má několik nedostatků, které by bylo potřeba vyřešit.

### **1.2.3 Identifikované potíže**

Používání aplikace je v některých ohledech docela zdlouhavé a neintuitivní. Při vyhledávání aplikací či ciferníků chybí více filtrů podle kterých by si mohl uživatel snadněji vybrat. Při sledování záznamu svého spánku, srdečního tepu atd. by byl potřeba slider, aby uživatel nemusel zdlouhavě naklikávat jednotlivé naměřené časové úseky.

### **1.2.4 Návrh nového řešení**

V případě této aplikace by byl potřeba redesign a přidání některých funkcí.

## **1.3 Lukáš Neupauer**

### **1.3.1 Spôsob hľadania témy**

Pozorovanie okolia a hľadanie spôsobu zprehľadnenia a väčšieho dosahu informácií. Skúmanie možnosti pridania funkcií do sociálnych médií.

### **1.3.2 Spôsob skúmania užívateľa a jeho potreby**

Oslovenie potencionálnych užívateľov a zisťovanie nedostatkov a možností vylepšení aplikácií v ich každodennom používaní. Najčastejšie Discord, Facebook, WhatsApp a Instagram.

### **1.3.3 Identifikované nedostatky**

Neprehľadnosť alebo úplne absencia udalostí v okolí. Užívateľ si pri Facebooku nemôže filtrovať udalosti podľa kategórií a väčšina užívateľ túto vymoženosť ani nevyužíva pre komplikovanosť platformy. Pri ostatných táto platformách bola pozorovaná absencia tejto funkcionality.

### **1.3.4 Návrh nového riešenia**

Vytvorenie jednoduchšej platformy s udalosťami v okolí, s možnosťou zoradenia do kategórií, funkciou zobrazenia udalosti iba určitým ľuďom (napr. systém priateľov) a ďalšími funkciami, ktorá bude slúžiť výužívajú, iba na zvýšenie povedomia a sprehľadnenie udalostí v okolí.

## 2 Důvod vybrání zvoleného tématu

Po společném schůzi se tým demokraticky rozhodl pro volbu aplikace od *Lukáše Haise*. Důvodem k tomuto kroku bylo celkové zalíbení členů pro daný projekt.

## 3 Analýza uživatele

Uživatel pracuje jakožto číšník a ve volném čase sportuje. Má rád data a výstupy z nich, které mu mohou dát nějaký retrospektivní pohled na různé části jeho života.

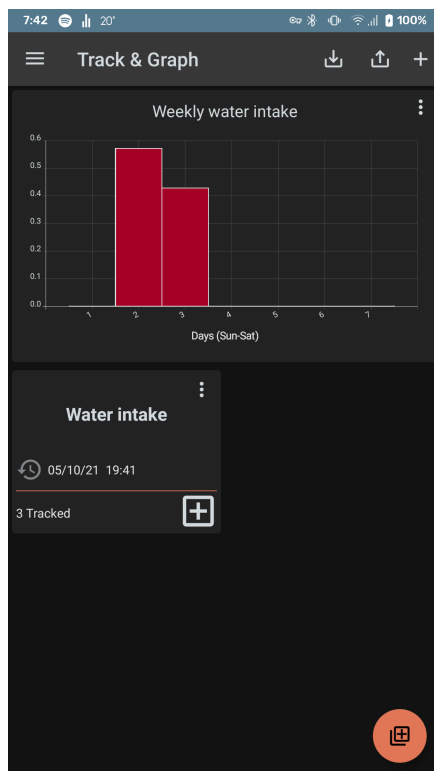
## 4 Potřeby uživatele

Uživatel chce využívat aplikaci pro zaznamenávání dat a následné retrospektivní vyhodnocování zaznamenaných dat.

Může jít např. o sledování četnosti výskytu zlozvyků, udržování nastavených pravidel (např. denní příjem vody), sledování zlepšení fyzických výkonů, „RPG statistiky“ při dlouhodobém využívání aplikace a zaznamenávání velkého množství dat.

## 5 Popis současného řešení

Uživatel využívá aplikaci *Track and Graph* [obrázek 1] (ke stažení zde: <https://f-droid.org/en/packages/com.samco.trackandgraph/>).



Obrázek 1: *Track and Graph* aplikace

## 6 Návrh zadání

V našem případě půjde o znovuvytvoření již existující aplikace, modernizaci designu a přidání chybějících funkcionalit.

### 6.1 Funkcionalita

Přidání kategorií pro lepší filtraci při vytváření grafů. Kategorie budou mít vlastní měrné jednotky (např. litry pro vypité pití).

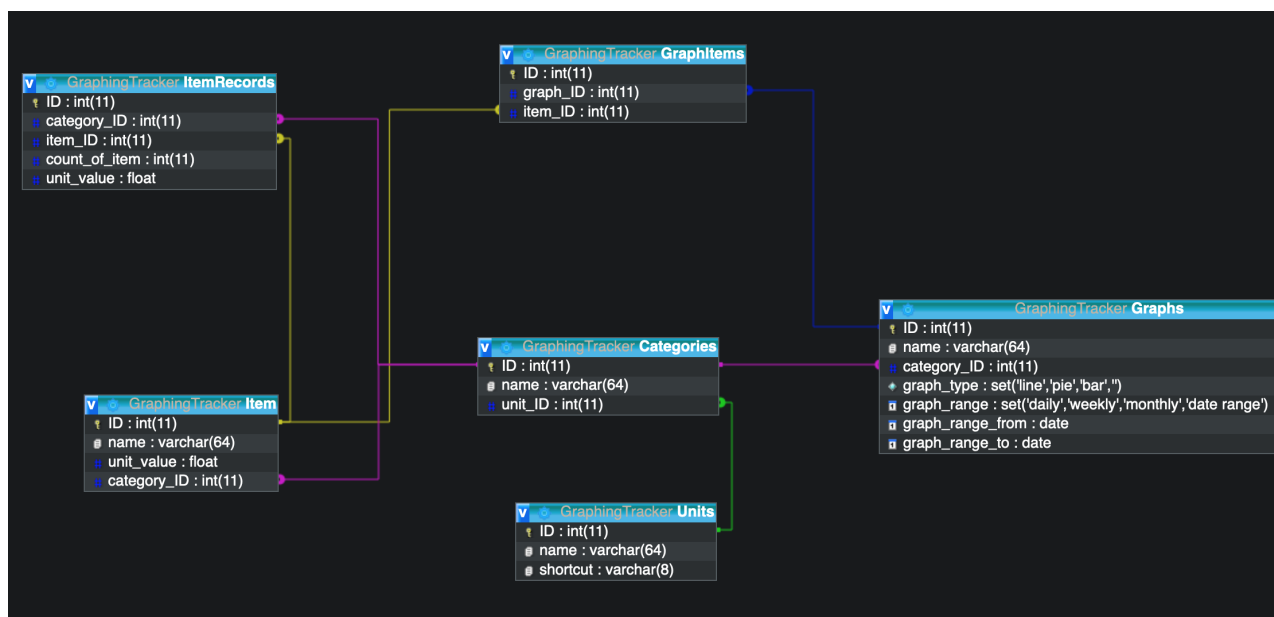
Toto by mělo směřovat k lepším výstupům samotné aplikace a tudíž k většímu zájmu o používání aplikace.

### 6.2 Grafika

Modernizace stávající grafiky. Vytvoření menu ve kterém lze upravovat/přidávat položky do databáze, či zobrazovat grafy.

## 7 Návrh řešení

### 7.1 Datový model



Obrázek 2: Předběžný návrh datového modelu

### 7.2 API

API v našem případě nebude komunikovat s databází, která by byla umístěna na externím serveru, ale naopak bude součástí aplikace jakožto SQLite soubor (není potřeba uchovávat data mimo aplikaci). API bude podporovat CRUD metody nad většinou entit v databázi (viz Obrázek 2) kromě Units, které již jsou předvyplněny základními jednotkami.

### 7.2.1 Create

- `Category::create()` - vytvoří novou kategorii se zadaným jménem a preferovanými jednotkami (např.: kategorie Alkohol s jednotkami v litrech)
- `Item::create()` - vytvoří nový pojmenovaný objekt ve specifikované kategorii se zadaným obsahem v preferovaných jednotkách (např.: Pivo 500ml v kategorii Alkohol)
- `ItemRecord::create()` - vytvoří nový záznam o objektu v kategorii a jeho četnosti (lze přidávat již existující objekty a jejich počet [3x Pivo 500ml] nebo neexistující objekt kde zadáváme samotné jednotky [50km do kategorie Běhání])
- `Graph::create()` - vytvoří nový pojmenovaný graf pro zadanou kategorii, nastaví preferovaný typ grafu (linie, obdelníkový, kruhový), období pro zobrazení (denní, týdenní, měsíční a nebo fixní datum od do)
- `GraphItem::create()` - vytvoří záznam o preferovaných objektech z kategorie, které se mají zobrazovat v grafu

### 7.2.2 Read

- `Category::Read()` - zobrazí informace o existující kategorii
- `Category::ReadAll()` - zobrazí všechny existující kategorie
- `Item::Read()` - zobrazí informace o existujícím objektu v kategorii
- `Item::ReadAll()` - zobrazí všechny existující objekty v kategorii
- `ItemRecord::Read()` - zobrazí informace o existujícím záznamu objektu
- `ItemRecord::ReadAll()` - zobrazí všechny záznamy o přidávaných objektech
- `Graph::Read()` - zobrazí informace o existujícím grafu
- `Graph::ReadAll()` - zobrazí všechny existující grafy
- `GraphItem::Read()` - zobrazí preferované objekty pro zadaný graf

### 7.2.3 Update

- `Category::Update()` - upraví jméno nebo preferované jednotky
- `Item::Update()` - upraví jméno nebo hodnotu u preferovaných jednotek
- `ItemRecord::Update()` - upraví záznam o objektu v kategorii
- `Graph::Update()` - upraví nastavení grafu

#### 7.2.4 Delete

- `Category::Delete()` - odstraní kategorii a všechny objekty v ní
- `Item::Delete()` - odstraní objekt z kategorie
- `ItemRecord::Delete()` - odstraní záznam o objektu z kategorie
- `Graph::Delete()` - odstraní graf včetně jeho nastavení
- `GraphItem::Delete()` - odstraní preferovaný objekt pro zadaný graf