

## Úvodem

Je skvělé, že jste se rozhodl posunout své znalosti. Následující úkol je zcela dobrovolný a jeho excelentní vypracování otevírá možnost přidat se k týmu TechFides. Úkol popisuje pokročilejší variantu webové aplikace, která bude obsahovat všechny důležité technické vychytávky, se kterými pracujeme v rozsáhlejších webových aplikacích.

## Vesmírná platforma

Cílem je vytvořit webovou aplikaci pro kosmonauty. Aplikace bude rozdělená na frontend a backend s využitím REST API nebo GraphQL. Backend bude postavený v Node.js s MongoDB. Frontend bude postaven na Reactu/Angular/Vue s uložením dat v nějakém single state systému (redux, vuex).

Dokumentace REST API bude provedena na Apiary.io a na úvodní stránce aplikace bude možné přepnout, k jakému serveru se frontend má připojovat s možností připojit na mock server apiary.

## Role uživatelů

K webové aplikaci budou přistupovat **kosmonauti** a řídicí středisko (**operátoři**). Operátoři mohou přidávat a editovat kosmonauty.

## Entity systému

V systému jsou především tyto hlavní entity:

### Kosmonaut

- Jméno
- Datum narození
- Váha
- Spotřeba jídla na hodinu letu

### Raketa

- Počet míst
- Objem nádrže
- Stav nádrže
- Spotřeba paliva na 1km letu
- Rychlost rakety
- Nosnost (v praxi jí plní hmotnost kosmonautů plus váha jídla v ledničce)
- Objem ledničky (kg)

### Uživatel

- Email

- Heslo
- Role

Let do vesmíru

- Vzdálenost
- Vazba na raketu
- Čas startu

## Důležité případy užití

### Registrace operátora

Operátoři se mohou do systému zaregistrovat z internetu, kde musí proběhnout ověření jejich emailové adresy. Operátoři musí mít možnost obnovit heslo přes mail, pokud ho zapomenou.

### Vytvoření a start letu

Operátor může v systému vytvořit kosmonauty a rakety. Dále může založit let do vesmíru a tomuto letu přiřadit raketu. Na základě zadaného času startu se dopočítá předpokládaný čas přistání, který je odvozen od rychlosti rakety. Operátor může naplnit raketu kosmonauty a odstartovat let, následně má přehled o provedených letech do vesmíru. Během zakládání letu také operátor doplní aktuální stav paliva v nádrži rakety a také stav naplnění ledničky (v procentech). Samozřejmě není možné, aby jeden kosmonaut ve stejný čas byl ve dvou raketách, přičemž je možné plánovat více navazujících letů dopředu s jedním kosmonautem.

Každý kosmonaut má evidováno, kolik spotřebuje jídla na hodinu letu. Zároveň je u rakety a konkrétního letu operátorem zadáno kolik poletí kosmonautů. Tedy v součtu víme, jaká bude spotřeba jídla na hodinu letu. Také známe délku letu podle rychlosti rakety. Celkově tedy dokážeme spočítat, kolik toho kosmonauti snědí za celý let – výsledná hmotnost musí stačit pro objem ledničky v raketě, v opačném případě nesmí být umožněno spuštění letu, respektive přidání dalšího kosmonauta do plánovaného letu.

Pozor! Mladí kosmonauti (mezi 20-40 lety věku) mohou 30% letu hladovět, je tedy možné takový let odstartovat, nicméně operátor musí být upozorněn, že budou kosmonauti hladovět.

Nesmí být povoleno operátorovi odstartovat raketu pokud:

- Je raketa přetížená
- Na palubě rakety je málo jídla pro kosmonauty na celý let
- Raketa má nedostatek paliva

Operátorovi se hezky zobrazí, co není splněno, aby mohl napravit konfigurace letu, aby byl následný let úspěšný.

Po přihlášení vidí operátor dashboard aktuálních letů, kdy u každého je zobrazen aktuální stav, tedy kolik kilometrů má raketa za sebou, kolik zbývá paliva a jídla v ledničce rakety (dopočítáno na základě času startu rakety a spotřeb entit). Pokud kterýkoliv kosmonaut spustí autodestrukci rakety, pak se okamžitě (využití socketů) operátorovi tento let na dashboardu vyznačí jako zrušený.

## Kosmonauti

Kosmonaut vidí po přihlášení své plánované, aktuální i uskutečněné lety. Kosmonaut také může chatovat s ostatními kosmonauty v rámci jednotlivých aktuálních letů. Chat bude řešen pomocí socketů. Chat musí umožnit kosmonautům posílat fotky z letu.

Kosmonauti mají možnost u aktuálního letu spustit autodestrukci rakety. V takovém případě všem přihlášeným kosmonautům začne běžet viditelný odpočet 10 vteřin, aby opustili raketu (využití socketů). Po autodestrukci bude raketa označena za zničenou a všechny plánované lety budou označeny jako zrušené a to včetně aktuálního.

## Automatizované testy

Ve světě je trendem provádět kontinuální deployment, což znamená nutnost pečlivého testování. Je tedy nutné pokrýt a zavést alespoň tyto typy testu:

- Jednotkové testy na frontendu i na backendu
- Selenium webdriver testy (psát v Javě) pro důležité případy užití aplikace s Allure reporty (<https://github.com/allure-framework/>)

Pozor, je potřeba dát pozor, aby psané testy dávaly smysl! Je potřeba testovat především implementovanou logiku.

## Doplňující informace

Jako každá světová aplikace, i vesmírná platforma potřebuje splňovat následující:

- 100% responzivita
- Jazykové mutace česká/slovenská + anglická
- Prosím o využití nějaké pěkné šablony postavené na responzivním frameworku (Bootstrap, Material design)
- Kód prosím vystavit veřejně na github
- Finální implementace musí být hezká a čistá
- Aplikaci prosím dockerizovat (<https://www.docker.com/>)

Veškeré nepopsané funkcionality či scénáře prosím doplnit podle vlastního uvážení.