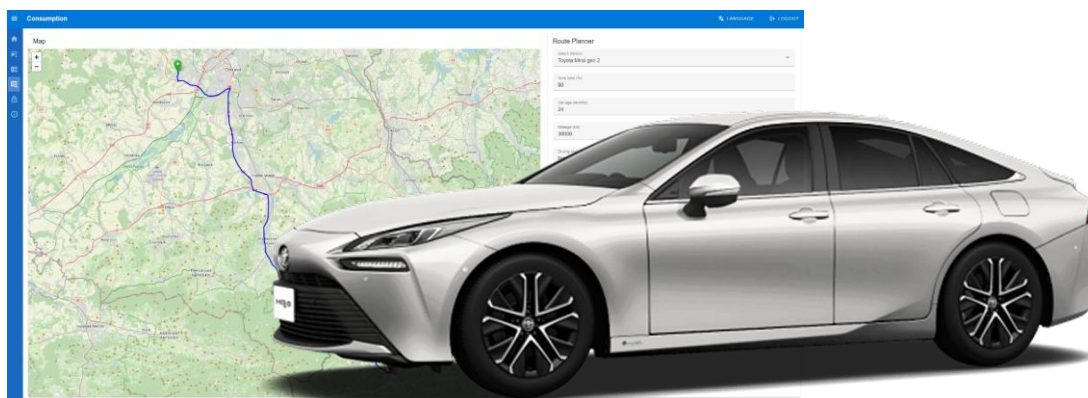


Software s implementovaným matematickým modelem části vozidla na vodíkový pohon

R – Software



Tento projekt Národní centrum vodíkové mobility je spolufinancován se státní podporou Technologické agentury ČR v rámci Programu Národní Centra kompetence. Tento software byl financován v rámci Národního plánu obnovy z evropského Nástroje pro oživení a odolnost.

NÁZEV SOFTWARE

Software s implementovaným matematickým modelem části vozidla na vodíkový pohon

PROJEKT

Národní centrum vodíkové mobility (TN02000007)

NÁZEV DÍLČÍHO PROJEKTU

Podpůrný SW pro akceleraci zavádění vodíkové mobility

AUTOŘI

Lukáš Prokop, VŠB-TU Ostrava, Centrum ENET

David Seidl, VŠB-TU Ostrava, Centrum ENET

Tomáš Kapusňák, VŠB-TU Ostrava, Centrum ENET

AUTORSKÝ TÝM KONCEPČNÍCH MATERIÁLŮ K SOFTWARE

Lukáš Prokop, David Seidl, Tomáš Kapusňák

Obsah

1	Úvod	5
2	Dosavadní stav techniky.....	6
3	Novost řešení	7
4	Popis SW.....	8
4.1	Koncepční materiály SW	8
4.1.1	Architektura softwaru	8
4.1.2	Struktura projektu	9
4.1.3	Datové modely a API	9
4.1.4	Endpoint pro simulaci.....	10
4.2	Matematický model simulace	10
4.3	Analýza funkčních požadavků.....	11
4.3.1	Uživatelské role	11
4.3.2	Funkční požadavky	12
4.3.3	Nefunkční požadavky	14
4.4	Programátorská dokumentace	15
4.4.1	Použité technologie.....	15
4.4.2	Struktura projektu	15
4.4.3	Popis klíčových komponent.....	16
4.4.4	Specifikace API (OpenAPI)	18
5	Popis ověření funkčnosti SW	20
5.1	Cíl ověření	20
5.2	Testovací prostředí	20
5.3	Metodika testování.....	20
5.4	Testovací scénáře a ověření funkčnosti (F1–F6)	21
5.4.1	Ověření F1 – Správa uživatelů a zabezpečení	21
5.4.2	Ověření F2 – Správa vozidel	22
5.4.3	Ověření F3 – Plánování tras.....	22
5.4.4	Ověření F4 – Simulační jádro (Fyzikální model)	22
5.4.5	Ověření F5 a F6 – Vizualizace a UI.....	23
5.5	Validace fyzikálního modelu	23
5.6	Ověření nefunkčních požadavků (N1–N6).....	24

5.7	Závěr k ověření funkčnosti SW	24
6	Umístění SW	25
7	Způsob využití	26
7.1	Práce s grafickým rozhraním (Web App)	26
7.1.1	Přihlášení do systému	26
7.1.2	Správa vozidel.....	26
7.1.3	Plánování trasy a simulace	27
7.1.4	Interpretace výsledků.....	27
7.2	Práce s REST API (Pro vývojáře)	28
7.2.1	Autentizace.....	28
7.2.2	Endpoint: Simulace spotřeby	28
7.3	Screenshoty aplikace	30
8	Seznam příloh.....	32
8.1	Uživatelská příručka případně manuál	32
8.2	Videosoubor se záznamem funkčnosti.....	32
9	Seznam obrázků	33

1 Úvod

V současné době dochází k postupné implementaci strategických dokumentů Evropské komise jako jsou Green Deal a Fit for 55, které definují parametry postupného přechodu ke klimaticky neutrálnímu hospodářství do roku 2050. Součástí těchto změn je bezprecedentní změna energetického mixu, rozvoj technologií pro akumulaci elektrické energie nebo maximální efektivní využívání lokální surovinové základny, včetně energetického využití odpadů a maximalizace využití obnovitelných zdrojů energie. Velký význam je kladen také na rozšíření bezemisních technologií v průmyslu. V řadě průmyslových odvětví, ale také v různých typech dopravy, je za perspektivní součást této snahy považován rozvoj vodíkových technologií.

Výsledkem této části projektu je „Software s implementovaným matematickým modelem části vozidla na vodíkový pohon“ (H2SIM), který slouží k simulaci provozu vozidel na vodíkový pohon v reálných provozních podmínkách. Software integruje vlastní matematické modely fyzikální dynamiky vozidla, spotřeby vodíku a degradace palivového článku, které byly vyvinuty v rámci výzkumných aktivit projektu.

V kontextu tohoto dokumentu bude „Software s implementovaným matematickým modelem části vozidla na vodíkový pohon“ nazýván zkráceným názvem H2SIM.



2 Dosavadní stav techniky

Na rozdíl od běžných simulačních nástrojů, které uvažují konstantní parametry pohonu, „Software s implementovaným matematickým modelem části vozidla na vodíkový pohon“ (H2SIM) umožňuje simulaci reálného technického stavu vozidla v průběhu jeho životního cyklu, což představuje nový přístup v oblasti softwarové simulace vodíkové mobility.

Tento aspekt přímo naplňuje požadavek MET-12 na „zvýšení objemu znalostí v oblasti softwarových řešení“.

3 Novost řešení

Novost „Software s implementovaným matematickým modelem části vozidla na vodíkový pohon“ (H2SIM) spočívá v:

- dynamické vazbě mezi provozní historií vozidla (stáří, nájezd, styl jízdy) a dostupným výkonem pohonu,
- modulární softwarové architektuře umožňující oddělení fyzikálního modelu jízdy a modelu degradace,
- schopnosti provádět predikční analýzy provozu vodíkových vozidel a jejich ekonomiky v čase.

„Software s implementovaným matematickým modelem části vozidla na vodíkový pohon“ (H2SIM) je **aplikační software** ve smyslu MET-12, neboť:

- běží nad standardním operačním systémem,
- poskytuje **specializované analytické a simulační funkce**,
- není pouhou webovou prezentací ani informačním systémem,
- implementuje **vlastní matematické a algoritmické jádro**, nikoliv pouze integraci cizích nástrojů.

4 Popis SW

4.1 Koncepční materiály SW

Tato technická dokumentace popisuje implementaci a architekturu softwaru H2SIM, který simuluje energetickou náročnost a spotřebu vodíkových vozidel na základě implementovaného fyzikálního modelu.

Model slouží k ověření spotřeby paliva (vodíku) v závislosti na profilu trasy, parametrech vozidla a fyzikálních odporech. Aplikace je navržena jako modulární systém, který umožňuje v budoucnu nahradit zjednodušené výpočetní jádro komplexnějšími modely zahrnujícími i tepelný management a degradaci komponent.

4.1.1 Architektura softwaru

H2SIM je implementován jako webová aplikace využívající architekturu klient-server. Komunikace probíhá prostřednictvím REST API. Systém je rozdělen do následujících hlavních vrstev:

Prezentační vrstva (Front-end)

- Aplikace je přístupná prostřednictvím webového rozhraní postaveného na frameworku Vue.js 3.
- Pro vizualizaci mapových podkladů a tras je využita knihovna Leaflet, která komunikuje s mapovými dlaždicemi OpenStreetMap.
- Uživatel zadává parametry vozidla, vybírá trasu v mapě a zobrazuje výsledky simulace (grafy spotřeby, výškový profil) pomocí interaktivních komponent.

Aplikační vrstva (Back-end)

- Serverová část je napsána v jazyce Go (Golang), což zajišťuje vysokou rychlost matematických výpočtů.
- API je implementováno pomocí frameworku Gin, který obsluhuje HTTP požadavky a formátuje data do JSON.
- Hlavní logika simulace (výpočet sil a spotřeby) je zapouzdřena v modulu ModelV2. Backend zároveň funguje jako prostředník (proxy) pro komunikaci s externí routovací službou GraphHopper.
- Výpočetní jádro (Backend) při každém spuštění simulace jízdy aktivně komunikuje s degradačním modelem. Na základě vstupních parametrů zadaných uživatelem (stáří vozu, nájezd, styl jízdy) backend sestaví požadavek a odešle jej na API h2car. Získaná hodnota FinalPower (výkon po degradaci) je následně backendem použita jako limitní faktor pro výpočet dynamiky vozidla a spotřeby v jednotlivých segmentech trasy. Tím je zajištěno, že simulace jízdy odpovídá reálnému technickému stavu vozidla v čase.

Datová vrstva

- Pro trvalé uložení dat o uživatelích a definicích vozidel je využita relační databáze PostgreSQL.
- Data o terénu (výškový profil) jsou získávána dynamicky z externí služby a nejsou trvale ukládána, což snižuje nároky na úložiště.

4.1.2 Struktura projektu

Projekt je organizován v kontejnerizované struktuře (Docker):

H2SIM/

```
|
| └─ Backend/          # Serverová logika (Go)
|   └─ server/controllers/ # API handlers a výpočetní jádro
|   └─ server/models/     # Datové struktury (ORM Gorm)
|   └─ Dockerfile        # Konfigurace kontejneru
|
| └─ Frontend/          # Klientská aplikace (Vue.js)
|   └─ src/views/       # UI stránky (Mapa, Vozidla)
|   └─ src/components/  # Grafy, Formuláře
|   └─ Dockerfile       # Konfigurace kontejneru
|
| └─ docker-compose.yml  # Orchestrace celého řešení
```

4.1.3 Datové modely a API

Model vozidla

Představuje fyzikální parametry nutné pro výpočet.

Popis parametrů:

```
{
  "name": "H2 Truck Prototype",
  "total_mass": 18000, // Hmotnost [kg]
  "dragCoef": 0.6,    // Součinitel odporu vzduchu Cx
  "frontalArea": 8.5,  // Čelní plocha [m2]
  "tankCapacity": 4.5, // Kapacita nádrže [kg H2]
  "width": 1890.0,     // šířka vozidla
  "height": 3500.0,    // výška vozidla
  "frontal_area": 0.0, //vypočtená hodnota čelní plochy vozidla
  "motor_efficiency": 98.0, //účinnost motoru
  "drive_line_efficiency": 95.0, //účinnost hnací soustavy
  "battery_efficiency": 98.0, //účinnost baterie
  "battery_capacity": 1.6,   //kapacita baterie
```

```
"fuel_cell_type": "EcoDrive 60kW Long-Life", //typ palivového článku ze systému h2car  
"number_of_cells": 5 // počet palivových článků  
}
```

4.1.4 Endpoint pro simulaci

Endpoint: POST /api/v1/run Popis: Přijímá ID vozidla, pole GPS souřadnic trasy a v případě H2 vozidel také stáří a nájezd vozidla.

Vrací vypočtenou spotřebu.

Příklad výstupu:

```
{  
  "status": "success",  
  "total_consumption_kg": 4.2,  
  "data_points": [  
    { "dist": 0, "elev": 200, "speed": 0, "tank_status": 100 },  
    { "dist": 0.1, "elev": 205, "speed": 45, "tank_status": 99.98 }  
  ]  
}
```

4.2 Matematický model simulace

Jádro softwaru (implementované v Backend/server/controllers/ModelV2) využívá fyzikální model podélné dynamiky vozidla. V každém kroku simulace (segment trasy) jsou počítány následující síly:

Jízdní odpory

Celková síla F_{total} , kterou musí motor překonat, je součtem dílčích odporů:

$$F_{total} = F_{aero} + F_{roll} + F_{slope} + F_{inert}$$

Aerodynamický odpor (F_{aero})

Závisí na čtverci rychlosti, hustotě vzduchu a tvaru vozidla.

$$F_{aero} = \frac{1}{2} * \rho * C_x * A * v^2$$

Kde: ρ je hustota vzduchu, C_x koeficient odporu, A čelní plocha, v rychlost.

Valivý odpor (F_{roll})

Závisí na hmotnosti vozidla a typu povrchu/pneumatik.

$$F_{roll} = C_{rr} * m * g * \cos(\alpha)$$

Kde: C_{rr} je součinitel valivého odporu, m hmotnost vozidla s posádkou, g gravitační zrychlení, α úhel sklonu vozovky.

Odpor Stoupání (F_{slope})

Gravitační síla působící proti směru jízdy při jízdě do kopce.

$$F_{slope} = m * g * \sin(\alpha)$$

Setrvačná síla (F_{inert}): Síla potřebná pro zrychlení vozidla (nebo brzdná síla při zpomalení).

$$F_{inert} = m * a$$

Výpočet spotřeby vodíku

Mechanický výkon P_{mech} potřebný na kolech se vypočítá jako součin celkové síly a rychlosti:

$$P_{mech} = F_{total} * v$$

Pro vodíkový pohon je následně vypočten hmotnostní průtok paliva $\dot{m}_{H_2}$ s využitím účinnosti palivového článku η_{H_2} a výhřevnosti vodíku ($LHV_{H_2} \approx 120 \text{ MJ/kg}$):

$$\dot{m}_{H_2} = \frac{P_{mech}}{\eta_{fc} * LHV_{H_2}}$$

Výsledná spotřeba za celou trasu je integrálem okamžité spotřeby v čase.

4.3 Analýza funkčních požadavků

Cílem této analýzy funkčních požadavků je popsat, jakým způsobem splňuje software H2SIM požadavky na simulaci spotřeby vodíku během jízdy v reálných podmínkách. Software je určen k simulaci a vizualizaci spotřeby vodíkového vozidla na základě nově implementovaného a unikátního fyzikálního modelu, který zohledňuje vlastnosti vozidla, parametry vodíkového pohonu (palivový článek) i trasy.

Tento model slouží k ověření funkčnosti v běžných podmínkách a představuje nástroj pro aplikovaný výzkum v oblasti elektromobility a vodíkových technologií.

4.3.1 Uživatelské role

U1 – Neregistrovaný uživatel (Guest)

- Uživatel, jenž má oprávnění vidět úvodní stránku a informace o projektu.
- Uživatel nemá žádná oprávnění k práci se systémem.

U2 – Registrovaný uživatel

- Uživatel má plný přístup ke všem částem webové aplikace (Správa vozidel, Plánovač tras, Nastavení).
- Uživatel má možnost spouštět výpočetní model.

4.3.2 Funkční požadavky

F1 – Správa uživatelů a zabezpečení

- F1.1: Systém musí umožnit uživateli přihlášení pomocí definovaného uživatelského jména a hesla.
- F1.2: Komunikace s API musí být zabezpečena pomocí JWT (JSON Web Token). Systém musí provádět validaci přístupového tokenu s každým požadavkem a automaticky obnovovat sezení pomocí obnovovacího tokenu.
- F1.3: Systém musí umožnit přihlášenému uživateli změnu hesla po zadání stávajícího hesla.
- F1.4: Systém musí umožnit uživateli bezpečné odhlášení a zneplatnění přístupu na klientovi.

F2 – Správa vozidel

- F2.1: Systém umožní zobrazit seznam všech dostupných vozidel systému s možností filtrování, řazení a stránkování.
- F2.2: Systém musí udržovat parametry jednotlivých vozidel po dobu životnosti systému. Každé vozidlo musí obsahovat následující parametry:
 - Název vozidla,
 - Typ pohonu (EV – Elektrické, H2 – Vodíkové) ,
 - Celková hmotnost (kg),
 - Šířka a výška (mm)
 - Slouží k výpočtu přibližné čelní plochy vozidla,
 - Koeficient odporu vzduchu,
 - Efektivita motoru (%),
 - Efektivita hnací soustavy (%),
 - Efektivita baterie (%),
 - Kapacita baterie (kWh)/ objem vodíkové nádrže (kg),
 - Typ palivového článku (pouze H2),
 - Počet palivových článků (pouze H2).
- F2.3: Systém umožní editování existujících vozidel a také možnost přidání nového vozidla. Systém musí provádět validaci vstupů a informovat uživatele o chybějících parametrech.
- F2.4: Systém umožní uživateli smazat vozidlo po potvrzení dialogu.

F3 – Plánovač tras a mapové podklady

- F3.1: Systém musí zobrazit interaktivní mapu uživateli.
- F3.2: Systém musí umožnit textové vyhledávání adres (Start, Cíl, Průjezdny body) s podporou našeptávání relativních adres.
- F3.3: Systém musí umožnit uživateli zvolit body trasy kliknutím na interaktivní mapu.
- F3.4: Uživatel musí být schopen definovat trasu skládající se z počátečního bodu, koncového bodu a libovolného počtu průjezdných bodů (zastávek).
- F3.5: Systém musí na základě definovaných bodů vypočítat trasu pro automobil, včetně výškového profilu trasy a průměrné rychlosti.

F4 – Simulační model

- F4.1: Systém musí vyžadovat zvolení vozidla, bodů trasy a počáteční stav energie (Baterie v % nebo Nádrž v %) před spuštěním simulace, v případě H2 vozidel také zadání stáří vozidla a aktuální nájezd.
- F4.2: Systém musí vypočítat spotřebu v každém segmentu trasy na základě:
 - Silových působení:
 - Odpor vzduchu,
 - Valivý odpor,
 - Gravitační síla,
 - Setrvačná síla.
 - Efektivity:
 - Zohlednění účinnosti motoru, hnací soustavy a baterie.
 - Rekuperace:
 - Výpočet zpětného získání energie při brždění nebo jízdě z kopce (pouze pro EV).
- F4.3: Systém musí provádět simulaci spotřeby vodíku.
- F4.4: Systém musí na základě dostupných navigačních dat vypočítat jízdní rychlost a čas potřebný k překonání úseku.
- F4.5: Systém musí být schopen regulovat rychlost dle dostupného výkonu baterie nebo na základě degradace palivových článků.

F5 – Vizualizace výsledků simulace

- F5.1: Systém musí zobrazit uživateli vypočítanou trasu v interaktivní mapě
- F5.2: Systém musí vygenerovat interaktivní graf zobrazující průběh veličin po délce trasy:

- Nadmořská výška
- Rychlost vozidla
- Kumulativní spotřeba (kWh nebo kg pro H₂)
- Stav baterie/nádrže (%)

F6 – Lokalizace

- F6.1: Webová aplikace musí podporovat volbu českého a anglického jazyka

4.3.3 Nefunkční požadavky

N1 – Výkon

- Systém musí provádět výpočty v reálném čase. Délka výpočtu by měla být úměrná délce trasy, přičemž simulace běžné trasy by měla proběhnout v řádu jednotek sekund.
- API systému musí být optimalizováno pro rychlou odezvu i při zpracování velkého množství bodů trasy.

N2 – Rozšiřitelnost

- Systém musí být navržen tak, aby umožňoval snadné přidání nových typů vozidel a úpravu jejich fyzikálních parametrů bez nutnosti zásahu do zdrojového kódu výpočetního jádra.
- Architektura aplikace musí podporovat budoucí rozšíření o další jazykové mutace a případné nové moduly pro detailnější analýzu dat.

N3 – Stabilita a Spolehlivost

- Systém musí být odolný vůči chybám ve vstupních datech (např. při výpadku externích služeb) a musí zaručovat stabilitu serverové části i při neočekávaných stavech výpočtu.

N4 – Bezpečnost

- API musí být chráněno proti neautorizovanému přístupu; manipulace s daty a spouštění simulací musí být dostupné pouze ověřeným uživatelům.
- Uživatelská hesla a citlivé údaje nesmí být ukládány v čitelné podobě.

N5 – Interoperabilita

- Výměna dat mezi aplikační a serverovou částí musí probíhat v univerzálním formátu (JSON) pro zajištění komunikace
- Serverová část musí nabízet přístupné rozhraní také mimo klientskou aplikaci.

N6 – Uživatelská přívětivost

- Uživatelské rozhraní musí být responzivní a použitelné na různých typech zařízení (desktop, laptop, tablet).

- Systém musí uživateli poskytnout jasnou zpětnou vazbu o stavu probíhajících operací (výpočet, ukládání, chyba)

4.4 Programátorská dokumentace

Tato část dokumentace poskytuje detailní popis interní struktury kódu a implementace softwaru H2SIM, který slouží k simulaci energetické náročnosti a spotřeby vodíkových vozidel na reálných trasách. Dokument popisuje použité technologie, architekturu projektu, datové modely a algoritmy výpočetního jádra, které provádí fyzikální simulaci jízdy.

Software je navržen jako modulární systém s oddělenou serverovou částí (Backend) a klientskou částí (Frontend), běžící v kontejnerizovaném prostředí.

4.4.1 Použité technologie

Backend (Serverová část):

- Go (Golang) 1.21+: Primární jazyk pro implementaci aplikační logiky a výpočetního jádra. Zvolen pro svou rychlost a efektivitu při matematických operacích.
- Gin Framework: Webový framework pro tvorbu REST API.
- GORM: ORM knihovna pro práci s databází.

Frontend (Klientská část):

- Vue.js 3: Progresivní JavaScript framework pro tvorbu uživatelského rozhraní.
- Vuetify 3: UI knihovna implementující Material Design.
- Pinia: State management pro správu stavu aplikace (uživatel, vozidla).
- Leaflet / OpenLayers: Knihovny pro práci s interaktivními mapami.

Infrastruktura a Data:

- PostgreSQL: Relační databáze pro ukládání uživatelských dat a definic vozidel.
- Docker & Docker Compose: Kontejnerizace aplikace pro snadné nasazení a izolaci prostředí.
- Nginx: Webový server sloužící jako reverse proxy a pro servírování statických souborů frontendu.

4.4.2 Struktura projektu

Projekt je rozdělen do dvou hlavních adresářů (Backend, Frontend) a konfiguračních souborů pro nasazení.

```
H2SIM/  
|  
├── Backend/          # Serverová část (Go)  
|   ├── db/          # Konfigurace databáze a seedování dat
```

```
| ├── server/
| | ├── controllers/      # Logika API endpointů
| | |   ├── ModelV2/      # Fyzikální výpočetní jádro (Simulace)
| | |   |   ├── Calculations.go # Fyzikální vzorce (odpory, síly)
| | |   |   └── Consumption.go # Hlavní smyčka výpočtu
| | |   └── vehicle.go      # CRUD operace pro vozidla
| | ├── models/           # Definice datových struktur (Structs)
| | ├── routes.go         # Definice API cest (Routing)
| | └── security/         # Autentizace a JWT tokeny
| ├── main.go            # Vstupní bod aplikace
| ├── Dockerfile         # Definice obrazu pro Backend
| └── .env                # Konfigurace prostředí
|
| ├── Frontend/ConsumptionUI/ # Klientská část (Vue.js)
| | ├── src/
| | |   ├── components/    # Znovupoužitelné UI komponenty
| | |   ├── views/         # Hlavní stránky (Map, Vehicles, Login)
| | |   ├── stores/        # Pinia stores (User, Vehicle)
| | |   └── api/           # Axios wrapper pro komunikaci s API
| | ├── Dockerfile        # Definice obrazu pro Frontend
| | └── nginx.conf        # Konfigurace Nginx serveru
|
| └── docker-compose.yml    # Orchestrace kontejnerů
```

4.4.3 Popis klíčových komponent

Výpočetní jádro (Backend/server/controllers/ModelV2)

Modul ModelV2 zapouzdřuje veškerou logiku simulace. Je navržen jako bezstavová služba, která na vstupu přijímá parametry vozidla a trasy, a na výstupu vrací vypočtená data pro jednotlivé body.

- Calculations.go:
 - Obsahuje pomocné funkce pro výpočet jednotlivých odporových sil působících na vozidlo.
 - Implementuje funkce `aeroResistanceForce` (aerodynamický odpor), `rollingResistanceForce` (valivý odpor), `inertialForce` (setrvačná složka) a `hillClimbingForce` (odpor stoupání).
 - Tyto funkce pracují s datovými typy `float64` a jako vstup přebírají parametry z modelu `Vehicle` a aktuální stav z modelu `Trip`.
- Consumption.go:
 - Hlavní řídící skript simulace.

- Obsahuje volání API systému h2car s cílem získat maximální výkon palivových článků vodíkového vozidla.
- Obsahuje funkci RunSimulation, která iteruje přes pole souřadnic (RoutePoints).
- V každém kroku cyklu:
 - Dopočítá vzdálenost a převýšení mezi body.
 - Volá funkce z Calculations.go pro určení energetické náročnosti úseku.
 - Provádí validaci a případnou limitaci dle dostupného výkonu degradovaných palivových článků.
 - Agreguje výsledky do výstupní struktury JSON.

Správa dat a API (Backend/server/controllers)

Tato vrstva zajišťuje validaci vstupů a komunikaci s databází.

- vehicle.go:
 - Implementuje handlers pro REST API endpointy /vehicles.
 - Funkce CreateVehicle zajišťuje parsování JSON requestu do Go struktury a validaci povinných polí (např. kontrola, zda hmotnost > 0).
- navi.go:
 - Slouží jako adaptér pro externí navigační služby.
 - Funkce GetRoute sestavuje HTTP požadavek na GraphHopper API, zpracovává odpověď a extrahuje z ní geometrii trasy (encoded polyline) a výškový profil.

Vizualizace (Frontend)

Frontendová část je implementována v komponentách Vue.js a zajišťuje interakci s uživatelem.

- MapDisplay.vue:
 - Zodpovídá za vykreslení mapového podkladu (Leaflet) a zobrazení trasy.
 - Přijímá pole souřadnic z API a vykresluje je jako L.polyline.
- GraphDisplay.vue:
 - Vizualizuje numerické výsledky simulace. Využívá knihovnu Chart.js (nebo ekvivalent) pro vykreslení průběhu spotřeby a nadmořské výšky v závislosti na ujeté vzdálenosti.
- CellView.vue:
 - Zodpovídá za zobrazení dostupných palivových článků v systému h2car.

Datové modely (Databáze)

Data jsou uchovávána v relační databázi PostgreSQL. Struktura je definována pomocí Go struktur (Structs) v adresáři models/.

Tabulka: Vehicles (Vozidla)

- ID (UUID): Unikátní identifikátor.
- Name (String): Název vozidla.
- Type (String): Typ pohonu.
- Width (Float): Šířka vozidla.
- Height (Float): Výška vozidla
- Weight (Float): Celková hmotnost.
- DragCoef (Float): Koeficient odporu vzduchu.
- FrontalArea (Float): Čelní plocha vozidla.
- BatteryCapacity (Float): Kapacita baterie.
- TankCapacity (Float): Kapacita nádrže na vodík.
- Fuel_cell_type (String): Typ palivového článku.
- Number_of_cells (Int): Počet palivových článků

Tabulka: Users (Uživatelé)

- ID (UUID): Unikátní identifikátor.
- Username (String): Uživatelské jméno
- PasswordHash (String): Hash uživatelského hesla

4.4.4 Specifikace API (OpenAPI)

Autentizace

- POST /login
 - Vstup: { "username": "...", "password": "..." }
 - Výstup: { "accessToken": "eyJ...", "refreshToken": "..." }

Správa vozidel

- GET /api/v1/vehicles – Získá seznam dostupných vozidel.
- POST /api/v1/vehicles – Vytvoří nové vozidlo.

Simulace spotřeby

- POST /api/v1/run
 - Popis: Spustí fyzikální simulaci pro danou trasu a vozidlo.
 - Vstup (Body):

```
{  
  "vehicle_id": "uuid-vozidla",  
  "From": ["49.831515", "18.160785"],  
  "To": ["49.799864", "18.303964"],  
  "Stops": [  
    ["49.815234", "18.256732"]  
  ],  
  "BatteryLevel": 95.0,  
  "CarAge": 12,  
  "Mileage": 15000,  
  "DrivingStyle": 3  
}  
  ◦ Výstup:  
{  
  "status": "success",  
  "total_consumption_kg": 1.25,  
  "data_points": [  
    { "dist": 0.1, "elev": 250, "speed": 50, "tank_status": 99.9 },  
    { "dist": 0.2, "elev": 252, "speed": 52, "tank_status": 99.8 }  
  ]  
}
```

5 Popis ověření funkčnosti SW

Ověření bylo prováděno za účelem potvrzení, že aplikace správně implementuje fyzikální model a že výstupy odpovídají reálným podmínkám a dostupným testovacím datům. Testování zahrnovalo unit testy, integrační testy, funkční testy a testování na reálných scénářích.

Cílem ověření funkčnosti bylo otestovat, že software poskytuje přesné výsledky pro simulaci spotřeby vodíkových vozidel na libovolných trasách. Všechny testy byly provedeny v souladu s požadavky pro výsledek R – Software podle MET-12 a prokázaly, že software splňuje očekávání.

5.1 Cíl ověření

Cílem ověření funkčnosti je potvrdit, že:

- Software správně implementuje fyzikální model spotřeby (F4), přičemž výstupy odpovídají teoretickým předpokladům a výsledkům reálného měření.
- Všechny definované funkční požadavky (F1-F6) jsou splněny a aplikace pracuje v souladu se specifikací.
- Kvalitativní aspekty, definované v nefunkčních požadavcích (N1-N6), jsou dodrženy.
- API správně zpracovává vstupy, provádí výpočty a vrací očekávané výstupy ve formátu JSON.

5.2 Testovací prostředí

Ověření bylo prováděno v prostředí, které odpovídá cílové architektuře softwaru.

- Backend: Go (Golang), Gin Framework, Docker kontejner
- Frontend: Vue.js 3, Vuetify 3, Docker kontejner, Nginx
- Databáze: PostgreSQL
- Testovací nástroje:
 - Insomnia pro volání API požadavků
 - Go pro validaci fyzikálního modelu

5.3 Metodika testování

Ověřování probíhalo formou testování shora dolů (od jádra k uživatelskému rozhraní) a bylo rozděleno do následujících fází:

1. Unit Testy výpočetního jádra (White-Box Testing): Ověření základních matematických funkcí fyzikálního modelu (F4). Kontrola výpočtu sil (aerodynamický odpor, valivý odpor, gravitační síla) pro definované vstupní body.

2. Integrační testy API (Black-Box Testing): Kontrola komunikace mezi frontendem a backendem, ověření správy databázových entit (CRUD operace F2) a správnosti návratových kódů (N4).
3. Funkční testy (Scénářové testování): Provedení komplexních uživatelských scénářů pro ověření splnění F1, F3 a F5 (např. "Plánování trasy Brno–Praha s vozidlem X a zobrazení grafu spotřeby").

5.4 Testovací scénáře a ověření funkčnosti (F1–F6)

5.4.1 Ověření F1 – Správa uživatelů a zabezpečení

- Test: přihlášení validního uživatele
 - Akce: Zadání správného přihlašovacího jména a hesla do formuláře.
 - Očekávaný výsledek: Server vrátí HTTP 200, JSON s přístupovým tokenem a přesměruje uživatele do aplikace.
 - Výsledek: **Splněno**
- Test: přihlášení nevalidního uživatele
 - Akce: Zadání chybných přihlašovacích údajů.
 - Očekávaný výsledek: Server vrátí HTTP 401 Unauthorized a přístup je odepřen.
 - Výsledek: **Splněno**
- Test: Ochrana API endpointů (JWT)
 - Akce: Volání chráněného endpointu (/api/v1/vehicles) bez autorizační hlavičky.
 - Očekávaný výsledek: Server vrátí chybu HTTP 401 (invalid Token).
 - Výsledek: **Splněno**
- Test: Změna hesla
 - Akce: Uživatel zadá staré heslo a nové heslo v sekci Nastavení.
 - Očekávaný výsledek: Heslo je v databázi bezpečně aktualizováno (nový hash). Staré heslo již nelze použít.
 - Výsledek: **Splněno**

5.4.2 Ověření F2 – Správa vozidel

- Test: Vytvoření nového vozidla
 - Akce: Vyplnění formuláře pro nové vozidlo (Název, Hmotnost, Cd, Objem nádrže, typ palivového článku, počet palivových článků) a uložení.
 - Očekávaný výsledek: Vozidlo je uloženo do databáze a zobrazí se v seznamu vozidel.
 - Výsledek: **Splněno**
- Test: Validace vstupů
 - Akce: Pokus o uložení vozidla se zápornou hmotností.
 - Očekávaný výsledek: Systém zobrazí validační chybu a nedovolí data odeslat.
 - Výsledek: **Splněno**
- Test: Editace a smazání
 - Akce: Úprava parametrů existujícího vozidla a následné smazání jiného vozidla.
 - Očekávaný výsledek: Změny jsou perzistentní, smazané vozidlo zmizí ze seznamu.
 - Výsledek: **Splněno**

5.4.3 Ověření F3 – Plánování tras

- Test: **Vyhledání lokality**
 - Akce: Zadání textové adresy do vyhledávacího pole (např. "Brno").
 - Očekávaný výsledek: Našeptávač nabídne relevantní lokality, po výběru se údaj zapíše do formuláře.
 - Výsledek: **Splněno**
- Test: **Výpočet trasy s průjezdními body**
 - Akce: Zadání trasy Start – Průjezdní bod – Cíl.
 - Očekávaný výsledek: Na mapě se vykreslí trasa procházející body ve správném pořadí.
 - Výsledek: **Splněno**

5.4.4 Ověření F4 – Simulační jádro (Fyzikální model)

- Test: Simulace vodíkového vozidla
 - Akce: Spuštění simulace pro vozidlo s palivovým článkem.
 - Očekávaný výsledek: Výsledek obsahuje úbytek vodíku v nádrži (kg) a zohledňuje degradaci palivového článku.

- Výsledek: **Splněno**
- Test: Integrace s degradačním modelem (h2car)
 - Akce: Spuštění simulace jízdy s definovaným stářím vozidla a stylem jízdy.
 - Očekávaný výsledek: Backend H2SIM před výpočtem trasy úspěšně zavolá API h2car, získá hodnotu degradovaného výkonu (FinalPower) a tu použije jako limit pro fyzikální model jízdy. Výsledná doba jízdy nebo spotřeba se liší oproti novému vozidlu.
 - Výsledek: **Splněno**.
- Test: Vliv terénu (Stoupání/Klesání)
 - Akce: Porovnání spotřeby na rovině vs. v kopcovitém terénu.
 - Očekávaný výsledek: Spotřeba do kopce je výrazně vyšší. Při jízdě z kopce je spotřeba nižší.
 - Výsledek: **Splněno**

5.4.5 Ověření F5 a F6 – Vizualizace a UI

- Test: Vykreslení výsledků
 - Akce: Spuštění simulace.
 - Očekávaný výsledek: Zobrazí se interaktivní graf (Rychlost, Výška, Energie) a souhrnné statistiky jízdy.
 - Výsledek: **Splněno**
- Test: Lokalizace (Jazyk)
 - Akce: Přepnutí jazyka aplikace (CZ/EN).
 - Očekávaný výsledek: Texty v rozhraní se okamžitě změní do zvoleného jazyka.
 - Výsledek: **Splněno**

5.5 Validace fyzikálního modelu

Byla provedena validace výpočtového jádra se zaměřením na vodíkový pohon pro potvrzení správnosti výzkumného výsledku:

- Test teoretických konstant:
 - Ověřeno, že model správně aplikuje fyzikální zákony (gravitační síla, odpor vzduchu, valivý odpor) při výpočtu energetické náročnosti pohybu vozidla.
- Test charakteristiky spotřeby:

- Ověřeno, že simulovaný úbytek vodíku (kg) odpovídá energetickému výdeji potřebnému k překonání jízdních odporů, přepočítanému přes účinnost palivového článku.
- Porovnání s referenčními daty:
 - Výsledné hodnoty spotřeby (kg H₂ / 100 km) byly porovnány s dostupnými experimentálními daty a technickými listy reálných vodíkových vozidel. Výsledky simulace vykazují shodu s očekávanými hodnotami pro danou kategorii vozidla.

5.6 Ověření nefunkčních požadavků (N1–N6)

- N1 – Výkon:
 - Simulace běžné trasy (do 300 km) proběhla v řádu jednotek sekund. **Splněno.**
- N2 – Rozšiřitelnost:
 - Ověřena možnost úpravy fyzikálních parametrů H₂ vozidla bez zásahu do zdrojového kódu. **Splněno.**
- N3 – Stabilita:
 - Systém zvládl opakované spouštění simulací bez výpadku služby. **Splněno.**
- N4 – Bezpečnost:
 - Hesla jsou hashována, API je chráněno tokeny. **Splněno.**
- N5 – Interoperabilita:
 - Systém komunikuje s API GraphHopper a vrací data ve standardním JSON formátu. **Splněno.**
- N6 – Uživatelská přívětivost:
 - Aplikace je plně responzivní a ovladatelná na mobilním telefonu. **Splněno.**

5.7 Závěr k ověření funkčnosti SW

Ověření funkčnosti softwaru H2SIM prokázalo, že aplikace je plně funkční a splňuje všechny definované požadavky. Klíčový fyzikální model poskytuje validní výsledky simulace spotřeby pro vodíková vozidla, které jsou v souladu s fyzikálními principy a referenčními daty. Na základě skutečností lze konstatovat, že software H2SIM je funkčně správný a připravený pro simulaci spotřeby vodíkových vozidel. V budoucnu bude model rozšířen o modul komplexnější simulace komfortních systému vozidla, které umožní podrobnější analýzu vlivu konkrétních prvků na spotřebu.



6 Umístění SW

<https://h2car.vsb.cz>

Přihlašovací údaje jsou k dispozici na vyžádání u původců software.

7 Způsob využití

Tato část dokumentu slouží jako uživatelská příručka k softwaru H2SIM, který simuluje energetickou náročnost a spotřebu vodíkových vozidel (FCEV) v reálných provozních podmínkách.

Software slouží k predikci spotřeby paliva (vodíku) na základě fyzikálního modelu, který zohledňuje:

- Parametry trasy: Vzdálenost, rychlostní limity a výškový profil (stoupání/klesání).
- Parametry vozidla: Aerodynamický odpor, valivý odpor, hmotnost a účinnost palivového článku.

Aplikace je dostupná ve dvou formách: jako webová aplikace s grafickým rozhraním pro interaktivní práci a jako REST API pro automatizované výpočty.

7.1 Práce s grafickým rozhraním (Web App)

Pro běžného uživatele je určeno webové rozhraní, které umožňuje intuitivní zadávání dat a vizualizaci výsledků na mapě.

7.1.1 Přihlášení do systému

Aplikace je zabezpečena. Pro přístup je nutné se přihlásit přiděleným uživatelským jménem a heslem.

- Otevřete webový prohlížeč a zadejte adresu aplikace.
- Zadejte své přihlašovací údaje.

7.1.2 Správa vozidel

Před spuštěním simulace je nutné definovat vozidlo, se kterým chcete jet.

- Přejděte do sekce Vozidla (Vehicles) v hlavním menu.
- Pro vytvoření nového vozidla klikněte na tlačítko „Přidat vozidlo“.
- Vyplňte technické parametry:
 - Název vozidla: Uživatelská identifikace modelu (např. Toyota Mirai II).
 - Typ pohonu: Volba typu energetického systému (H2 – Vodíkový pohon)
 - Hmotnost (kg): Celková provozní hmotnost vozidla včetně posádky a nákladu.
 - Rozměry (mm): Výška a šířka vozidla, sloužící k automatickému výpočtu čelní plochy (A) pro určení aerodynamického odporu.
 - Aerodynamický koeficient (C_d): Bezrozměrná hodnota vyjadřující tvarovou aerodynamiku karoserie (u moderních aut typicky 0,25 – 0,35).
 - Kapacita nádrže: Kapacita nádrže v kg vodíku.

- Parametry účinnosti:
 - Účinnost motoru (%): Efektivita elektromotoru (standardně cca 95–98 %).
 - Účinnost hnací soustavy (%): Mechanické ztráty v převodovce a ložiscích (standardně cca 95 %).
 - Průměrná účinnost palivového článku (%): Statická hodnota reprezentující průměrnou efektivitu přeměny vodíku na elektřinu (zadejte např. 55 pro 55 %).
- Typ palivového článku: Volba palivového článku z možností systému h2car.
- Aplikace v tomto kroku automaticky komunikuje se systémem h2car a uživateli nabízí vždy aktuální seznam dostupných konfigurací palivových článků. Díky tomu není nutné při přidání nového typu článku zasahovat do klientské aplikace.
- Počet palivových článků: počet palivových článků (odpovídá co počtu dosahujícímu co nejpřesnější shodu s výkonem vozidla)
- Uložte vozidlo. Nyní je dostupné pro simulace.

7.1.3 Plánování trasy a simulace

- Přejděte na Plánovač.
- Vyberte vozidlo.
- Nastavte počáteční stav nádrže.
- Zadání trasy:
 - Klikněte do mapy pro výběr bodu Start.
 - Zadejte Cíl, případně přidejte průjezdní body (zastávky).
 - Klikněte do mapy pro výběr odpovídajícího bodu.
 - Alternativně využijte zadání adresy s pomocí vyhledávání.
- Klikněte na tlačítko „Spustit simulaci“ pro výpočet spotřeby.

7.1.4 Interpretace výsledků

Po dokončení výpočtu se zobrazí:

- Trasa na mapě: Barevná křivka kopírující silniční síť.
- Interaktivní grafy:
 - Výškový profil: Zobrazuje terén trasy.
 - Stav nádrže: Ukazuje úbytek paliva v průběhu cesty. Pokud křivka klesne na 0, vozidlo nedojede do cíle.

- Spotřeba paliva
- Průměrná rychlost

7.2 Práce s REST API (Pro vývojáře)

Software H2SIM poskytuje REST API, které umožňuje provádět simulace automatizovaně z jiných aplikací nebo skriptů (např. Python, MATLAB, cURL). Rozhraní přijímá a vrací data ve formátu JSON.

7.2.1 Autentizace

Pro volání API je nutné získat přístupový token (JWT).

- Endpoint: POST /login
- Body: {"username": "...", "password": "..."}
• Response: {"accessToken": "..."}
• Poznámka: Token vložte do hlavičky každého dalšího požadavku: Authorization: Bearer <token>.

7.2.2 Endpoint: Simulace spotřeby

Tento endpoint provádí výpočet fyzikálního modelu pro zadanou trasu a vozidlo.

- URL: POST /api/v1/run
- Popis: Spustí simulaci spotřeby na základě bodů trasy a ID vozidla.

Příklad volání (cURL):

```
curl -X POST \
  http://localhost:8080/api/v1/run \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer VÁŠ_TOKEN" \
  -d '{
    "vehicle_id": "d290f1ee-6c54-4b01-90e6-d701748f0851",
    "initial_fuel_percent": 80,
    "route_points": [
      {"lat": 50.0755, "lon": 14.4378},
      {"lat": 49.1950, "lon": 16.6068}
    ],
    "CarAge": 12,
    "Mileage": 15000,
    "DrivingStyle": 3
  }'
```

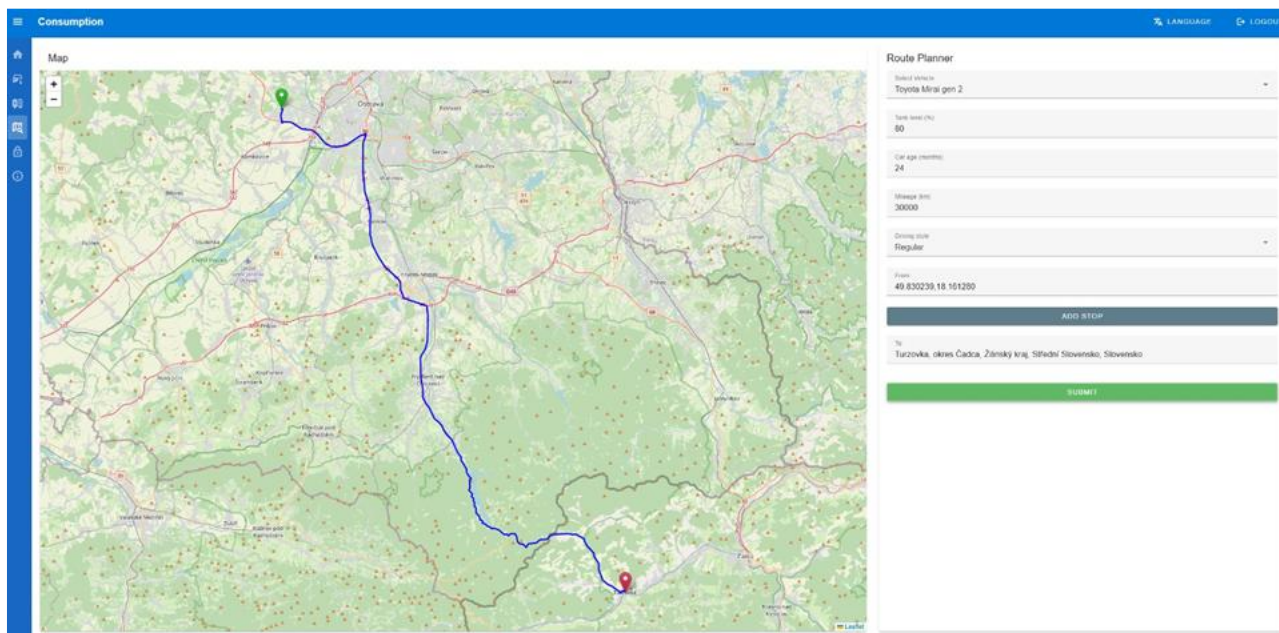
Příklad odpovědi (JSON): Výstup obsahuje celkovou spotřebu a detailní pole bodů pro vykreslení grafů. Struktura data_points odpovídá internímu modelu simulace.

```
{
  "status": "success",
  "total_consumption_kg": 1.25,
  "data_points": [
    {
      "dist": 0.0,
      "elev": 250,
      "speed": 0,
      "tank_status": 100.0
    },
    {
      "dist": 0.1,
      "elev": 252,
      "speed": 45,
      "tank_status": 79.98
    }
  ]
}
```

Vysvětlivky klíčů:

1. dist: Ujetá vzdálenost od startu [km].
2. elev: Nadmořská výška v daném bodě [m].
3. speed: Okamžitá rychlost vozidla [km/h].
4. tank_status: Stav vodíkové nádrže [%].

7.3 Screenshoty aplikace



OBRÁZEK 1: VIZUALIZACE TRASY SIMULOVANÉHO VOZIDLA



OBRÁZEK 2: PRŮBĚH SIMULOVANÝCH PARAMETRŮ

Edit Vehicle

Name	Nexo
Total Mass (kg)	1890
Height (mm)	1630
Width (mm)	1830
Drag Coefficient	0.329
Engine efficiency (%)	98
Drive line efficiency (%)	95
Battery efficiency (%)	98
☒ Hydrogen Vehicle (H2)	
Battery Capacity (kWh)	1.56
Hydrogen Tank Capacity (kg)	6.33
Number of Fuel Cells	2
Fuel Cell Type	EcoDrive 60kW Long Life

OBRÁZEK 3: ZADÁVÁNÍ PARAMETRŮ VOZIDLA

☒ Hydrogen Vehicle (H2)

Battery Capacity (kWh)	1.56
Hydrogen Tank Capacity (kg)	6.33
Fuel Cell Type	EcoDrive 60kW Long Life
Fuel Cell Type	Standard 60kW Fuel Cell
Fuel Cell Type	EcoDrive 60kW Long Life

OBRÁZEK 4: ZADÁVÁNÍ PARAMETRŮ PALIVOVÉHO ČLÁNKU

Consumption		LANGUAGE	LOGOUT
Fuel Cell Models			
Name	Initial Power (kW)		
EcoDrive 60kW Long Life	60		
Standard 60kW Fuel Cell	60		
		Items per page: 10	1-2 of 2

OBRÁZEK 5: VÝBĚR Z DEFINOVANÝCH PALIVOVÝCH ČLÁNKŮ POSKYTNUTÝCH API H2CAR



8 Seznam příloh

8.1 Uživatelská příručka případně manuál

Uživatelská příručka je přiložena v samostatném souboru:

https://www.cistadoprava.cz/files/NAHYC_DP003N-V10-Uzivatelaska_prirucka.pdf

8.2 Videosoubor se záznamem funkčnosti

Videosoubor s názvem H2CAR se záznamem funkce SW je v samostatném souboru:

https://www.cistadoprava.cz/files/NAHYC_DP003N-V10-H2CAR.mp4



9 Seznam obrázků

Obrázek 1: Vizualizace trasy simulovaného vozidla.....	30
Obrázek 2: Průběh simulovaných parametrů	30
Obrázek 3: Zadávání parametrů vozidla	31
Obrázek 4: Zadávání parametrů palivového článku	31
Obrázek 5: Výběr z definovaných palivových článků poskytnutých api h2car	31