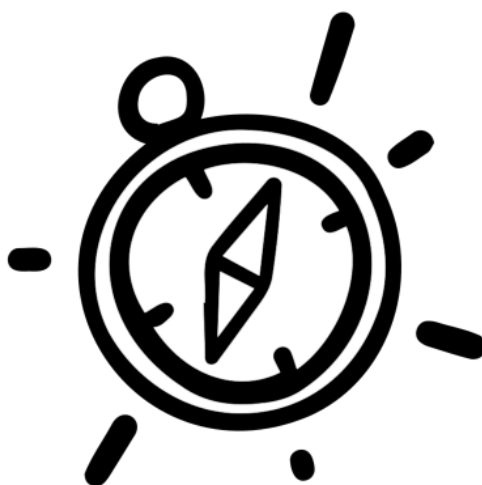


ZÁVĚREČNÁ STUDIJNÍ PRÁCE

dokumentace

LOTR - LOfcation TRacking System



Autor: Štěpán Balner
Obor: 18-20-M/01 INFORMAČNÍ TECHNOLOGIE
se zaměřením na počítačové sítě a programování
Třída: IT4
Školní rok: 2025/26

Poděkování

Děkuji těm, jimž poděkováno nebylo.

Prohlášení

Prohlašuji, že jsem závěrečnou práci vypracoval samostatně a uvedl veškeré použité informační zdroje.

Souhlasím, aby tato studijní práce byla použita k výukovým a prezentačním účelům na Střední průmyslové a umělecké škole v Opavě, Praskova 399/8.

V Opavě 1. 1. 2026

.....
Podpis autora

Anotace

Cílem této práce je návrh a realizace sledovacího systému LOTR (LOcation TRacking System), který se skládá z fyzického hardwarového zařízení, serverové infrastruktury a klientské mobilní aplikace.

Hardwarová část je postavena na platformě ESP32 s využitím modemu SIMCOM A7670 pro komunikaci přes síť LTE a modulu GPS pro získávání polohy. Firmware je optimalizován pro nízkou spotřebu energie a spolehlivý přenos dat. Serverová část, realizovaná v prostředí Node.js s databází MySQL, zajišťuje sběr telemetrických dat, správu uživatelů a poskytuje REST API pro komunikaci s koncovými zařízeními. Součástí systému je také nativní aplikace pro OS Android vyvinutá v jazyce Kotlin, která slouží jako plnohodnotná softwarová náhrada hardwarového trackeru.

Výsledkem projektu je funkční prototyp IoT řešení umožňující sledování polohy v reálném čase, historii pohybu a vzdálenou konfiguraci připojených zařízení.

Klíčová slova

GPS, IoT, sledovací systém, ESP32, LTE, Node.js, MySQL, Android, Kotlin, REST API

Obsah

Úvod	3
1 FYZICKÉ ZAŘÍZENÍ „TRACKER“	5
1.1 TEORETICKÁ VÝCHODISKA A POUŽITÉ TECHNOLOGIE	5
1.1.1 MIKROKONTROLÉRY A SoC (ESP32)	5
1.1.2 MOBILNÍ KOMUNIKACE V IoT A AT PŘÍKAZY	5
1.1.3 GLOBÁLNÍ NAVIGAČNÍ SYSTÉMY (GPS)	5
1.1.4 OPERAČNÍ SYSTÉMY REÁLNÉHO ČASU (FREERTOS)	5
1.1.5 PRINCIPY ŘÍZENÍ NAPÁJENÍ (POWER LATCH)	5
1.2 NÁVRH HARDWARE	5
1.2.1 VÝBĚR KOMPONENT (LILYGO, NEO-6M)	5
1.2.2 SCHÉMA ZAPOJENÍ A PERIFERIE	5
1.3 IMPLEMENTACE FIRMWARE	5
1.3.1 ARCHITEKTURA A POUŽITÉ KNIHOVNY	5
1.3.2 ŘÍZENÍ SPOTŘEBY A PRACOVNÍ CYKLUS	5
1.3.3 SPRÁVA DATOVÉHO ÚLOŽIŠTĚ (LITTLEFS)	5
1.4 KOMUNIKACE A DATA	5
1.4.1 KOMUNIKAČNÍ PROTOKOL A ZABEZPEČENÍ	6
1.4.2 STRUKTURA DAT A HANDSHAKE	6
1.5 KONFIGURACE A SERVISNÍ REŽIM	6
1.5.1 OTA REŽIM A WEBOVÉ ROZHRANÍ	6
1.5.2 PARAMETRIZACE ZAŘÍZENÍ	6
2 SERVEROVÁ ČÁST	7
2.1 ÚVOD A KONCEPCE SYSTÉMU	7
2.1.1 ROLE SERVERU V SYSTÉMU LOTR	7
2.1.2 MONOLITICKÁ ARCHITEKTURA A MVC VZOR	7
2.2 POUŽITÉ TECHNOLOGIE	7
2.2.1 RUNTIME PROSTŘEDÍ NODE.JS A FRAMEWORK EXPRESS	7
2.2.2 RELAČNÍ DATABÁZE A ORM (MYSQL + SEQUELIZE)	7
2.2.3 PRINCIPY AUTENTIZACE A OAUTH 2.0	7
2.2.4 REST API ARCHITEKTURA	7
2.3 NÁVRH A ARCHITEKTURA BACKENDU	7
2.3.1 DATABÁZOVÉ SCHÉMA A MODELÝ	7

2.3.2	STRUKTURA API ENDPOINTŮ	7
2.3.3	BEZPEČNOSTNÍ MECHANISMY	7
2.4	IMPLEMENTACE KLÍČOVÝCH FUNKCÍ	7
2.4.1	PŘÍJEM A VALIDACE TELEMETRICKÝCH DAT	7
2.4.2	ALGORITMY PRO ZPRACOVÁNÍ POLOHY (GEOFENCING, CLUSTERING)	8
2.4.3	SPRÁVA UŽIVATELŮ A OPRÁVNĚNÍ	8
2.5	PREZENTAČNÍ VRSTVA (FRONTEND)	8
2.5.1	ŠABLONOVACÍ SYSTÉM EJS	8
2.5.2	VIZUALIZACE DAT A MAPOVÉ PODKLADY	8
3	APLIKACE PRO ANDROID	9
3.1	KONCEPT A CÍLE APLIKACE	9
3.1.1	NÁHRADA HARDWAROVÉHO TRACKERU	9
3.1.2	PRINCIP OFFLINE-FIRST A ODOLNOST PROTI VÝPADKŮM	9
3.2	POUŽITÉ TECHNOLOGIE A PLATFORMA	9
3.2.1	JAZYK KOTLIN A ASYNCHRONNÍ PROGRAMOVÁNÍ (COROUTINES) .	9
3.2.2	SYSTÉMOVÉ KOMPONENTY (FOREGROUND SERVICE, WORKMANAGER)	9
3.2.3	LOKÁLNÍ PERSISTENCE (ROOM DATABASE)	9
3.2.4	ZABEZPEČENÉ ÚLOŽIŠTĚ (ENCRYPTEDSHAREDREFERENCES) . . .	9
3.3	ARCHITEKTURA APLIKACE	9
3.3.1	VRSTEVNATÁ ARCHITEKTURA A REAKTIVNÍ MODEL (STATEFLOW) .	9
3.3.2	VLASTNÍ IMPLEMENTACE HTTP KLIENTA	9
3.4	IMPLEMENTACE KLÍČOVÝCH FUNKCÍ	9
3.4.1	SBĚR POLOHY NA POZADÍ (LOCATIONSERVICE)	9
3.4.2	DÁVKOVÁ SYNCHRONIZACE DAT (SYNCWORKER)	9
3.4.3	ŘÍZENÍ SPOTŘEBY A VZDÁLENÉ VYPÍNÁNÍ (POWERCONTROLLER) .	10
3.4.4	HANDSHAKE PROTOKOL A KONFIGURACE	10
3.5	UŽIVATELSKÉ ROZHRANÍ	10
3.5.1	HLAVNÍ OVLÁDACÍ PANEL (DASHBOARD)	10
3.5.2	DIAGNOSTICKÁ KONZOLE PRO TERÉNNÍ TESTOVÁNÍ	10
	ZÁVĚR	11
	SEZNAM POUŽITÝCH INFORMAČNÍCH ZDROJŮ	13
	SEZNAM OBRÁZKŮ	15
	SEZNAM TABULEK	15
	PŘÍLOHY	15
	A SCHÉMA ZAPOJENÍ HARDWARE	17
	B ER DIAGRAM DATABÁZE	19
	C UŽIVATELSKÝ MANUÁL APLIKACE	21

ÚVOD

Můžeme říci, že již od začátku cílem projektu bylo vytvořit kompletní a hlavně uzavřený systém, co by mohl žít svým vlastním životem a jehož použití by bylo možné nasadit v reálných podmínkách.

Již kdysi jsem se zabýval "sběrnými" nebo "sledovacími" systémy. Poohlížel jsem se z variabilních důvodů po možnostech jak by šlo sestavit zařízení schopné nahrávat video a ukládat na SD kartu nebo jiné úložiště. Sledování polohy je jenom dalším krokem v tomto směru. Možnost že bych měl krabíčku jež bych mohl dát někomu do auta, nebo jej připnout k nápravě kamionu a sledoval jeho pohyb do doby vybití baterie byla, a stále je, velmi přitažlivá.

Bylo jasné, že již existují řešení a "GPS-trackery" jsou běžně dostupné a velmi dobré kvality a výdrže baterie. Avšak protože jsem dumal nad tématem pro závěrečný projekt, výroba a řešení konstrukce vlastního, sic nízkokvalitního a dosti poruchového, zařízení se ukázal jako vhodný nápad. Existovala možnost vytvořit opravdu pouze "krabici" trackeru s ukládáním dat na apevné úložiště, avšak rozhodl jsem se to spojit s vývojem serveru jež by umožňoval sledování v reálném čase. Na tomto "webovém" serveru by tak šlo zobrazovat a spravovat data posílaná zařízením přes mobilní síť. To se pak přirozeně rozvinulo do plného systému s uživatelskými účty a možností spravovat více zařízení na jednom z nich. Již z počátků pak bylo také na výledni, že pokud by server měl fungovat a přijímat data spolehlivě, tak jeho jádrem by muselo být responzivní a hlavně "robustní" API. Jeho struktura se však, hlavně kvůli postupnému přehodnocování principu komunikace mezi serverem a zařízeními, dosti drasticky měnila. Fyzické zařízení totiž bylo hlavním pilířem celého systému a jeho konstrukce, složení, a hlavně samotný kód, udávaly směr vývoje všeho ostatního.

Jako méně důležitou, ale přesto užitečnou součást systému jsem pak viděl souběžný vývoj mobilní aplikace pro systém Android. Ta měla sloužit jako plnohodnotná náhrada hardwarového zařízení a možná jako náhrada více schopná.

1 FYZICKÉ ZAŘÍZENÍ „TRACKER“

1.1 TEORETICKÁ VÝCHODISKA A POUŽITÉ TECHNOLOGIE

1.1.1 Mikrokontroléry a SoC (ESP32)

1.1.2 Mobilní komunikace v IoT a AT příkazy

1.1.3 Globální navigační systémy (GPS)

1.1.4 Operační systémy reálného času (FreeRTOS)

1.1.5 Principy řízení napájení (Power Latch)

1.2 NÁVRH HARDWARE

1.2.1 Výběr komponent (LilyGO, NEO-6M)

1.2.2 Schéma zapojení a periferie

1.3 IMPLEMENTACE FIRMWARE

1.3.1 Architektura a použité knihovny

1.3.2 Řízení spotřeby a pracovní cyklus

1.3.3 Správa datového úložiště (LittleFS)

1.4 KOMUNIKACE A DATA

1.4.1 Komunikační protokol a zabezpečení

1.4.2 Struktura dat a handshake

1.5 KONFIGURACE A SERVISNÍ REŽIM

1.5.1 OTA režim a webové rozhraní

1.5.2 Parametrizace zařízení

2 SERVEROVÁ ČÁST

2.1 ÚVOD A KONCEPCE SYSTÉMU

2.1.1 Role serveru v systému LOTR

2.1.2 Monolitická architektura a MVC vzor

2.2 POUŽITÉ TECHNOLOGIE

2.2.1 Runtime prostředí Node.js a framework Express

2.2.2 Relační databáze a ORM (MySQL + Sequelize)

2.2.3 Principy autentizace a OAuth 2.0

2.2.4 REST API architektura

2.3 NÁVRH A ARCHITEKTURA BACKENDU

2.3.1 Databázové schéma a modely

2.3.2 Struktura API endpointů

2.3.3 Bezpečnostní mechanismy

2.4 IMPLEMENTACE KLÍČOVÝCH FUNKCÍ

2.4.1 Příjem a validace telemetrických dat

2.4.2 Algoritmy pro zpracování polohy (Geofencing, Clustering)

2.4.3 Správa uživatelů a oprávnění

2.5 PREZENTAČNÍ VRSTVA (FRONTEND)

2.5.1 Šablonovací systém EJS

2.5.2 Vizualizace dat a mapové podklady

3 APLIKACE PRO ANDROID

3.1 KONCEPT A CÍLE APLIKACE

3.1.1 Náhrada hardwarového trackeru

3.1.2 Princip Offline-First a odolnost proti výpadkům

3.2 POUŽITÉ TECHNOLOGIE A PLATFORMA

3.2.1 Jazyk Kotlin a asynchronní programování (Coroutines)

3.2.2 Systémové komponenty (Foreground Service, WorkManager)

3.2.3 Lokální persistence (Room Database)

3.2.4 Zabezpečené úložiště (EncryptedSharedPreferences)

3.3 ARCHITEKTURA APLIKACE

3.3.1 Vrstevnatá architektura a reaktivní model (StateFlow)

3.3.2 Vlastní implementace HTTP klienta

3.4 IMPLEMENTACE KLÍČOVÝCH FUNKCÍ

3.4.1 Sběr polohy na pozadí (LocationService)

3.4.2 Dávková synchronizace dat (SyncWorker)

3.4.3 Řízení spotřeby a vzdálené vypínání (PowerController)

3.4.4 Handshake protokol a konfigurace

3.5 UŽIVATELSKÉ ROZHRANÍ

3.5.1 Hlavní ovládací panel (Dashboard)

3.5.2 Diagnostická konzole pro terénní testování

ZÁVĚR

ZHODNOCENÍ PROJEKTU

MOŽNOSTI BUDOUCÍHO VÝVOJE

SEZNAM POUŽITÝCH INFORMAČNÍCH ZDROJŮ

LITERATURA

- [1] DOKULIL Jakub. *Šablona pro psaní SOČ v programu L^AT_EX* [Online]. Brno, 2020 [cit. 2020-08-24]. Dostupné z: https://github.com/Kubiczek36/SOC_sablona
- [2] ESPRESSIF SYSTEMS. *ESP32 Technical Reference Manual* [Online]. 2024. Dostupné z: <https://www.espressif.com/en/support/documents/technical-documents>
- [3] SIMCOM. *A7600 Series AT Command Manual* [Online]. 2021.
- [4] JETBRAINS. *Kotlin Documentation* [Online]. Dostupné z: <https://kotlinlang.org/docs/home.html>
- [5] GOOGLE. *Android Developers Guides* [Online]. Dostupné z: <https://developer.android.com/guide>

Seznam obrázků

Seznam tabulek

PŘÍLOHA A SCHÉMA ZAPOJENÍ HARDWARE

PŘÍLOHA B ER DIAGRAM DATABÁZE

PŘÍLOHA C UŽIVATELSKÝ MANUÁL APLIKACE