



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

WEBOVÁ A MOBILNÍ APLIKACE PRO SBĚR A VIZUALIZACI METEOROLOGICKÝCH DAT

WEB AND MOBILE APPLICATION FOR METEOROLOGICAL DATA COLLECTION AND
VISUALISATION

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Martin Šturala

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Tomáš Caha

BRNO 2022

Diplomová práce

magisterský navazující studijní program **Telekomunikační a informační technika**

Ústav telekomunikací

Student: Bc. Martin Šturala

ID: 186212

Ročník: 2

Akademický rok: 2021/22

NÁZEV TÉMATU:

Webová a mobilní aplikace pro sběr a vizualizaci meteorologických dat

POKYNY PRO VYPRACOVÁNÍ:

Prostudujte problematiku meteorologických stanic a měřených veličin. Vytvořte webovou aplikaci pro sběr a vizualizaci meteorologických dat. Dále vytvořte mobilní aplikaci pro Android pro zobrazení vybraných dat.

DOPORUČENÁ LITERATURA:

[1] Linux Dokumentační projekt. 4. vyd. Computer Press, 2008. 1336 s. ISBN: 978-80-251-1525-1.

[2] PILGRIM, M. Ponořme se do Python(u) 3. CZ.NIC, 2010. 435 s. ISBN: 978-80-904248-2-1.

Termín zadání: 7.2.2022

Termín odevzdání: 24.5.2022

Vedoucí práce: Ing. Tomáš Čaha

prof. Ing. Jiří Mišurec, C.Sc.
předseda rady studijního programu

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Abstrakt

Práce se zabývá vývojem webové aplikace v .NET 5 a mobilní aplikace pro operační systém Android. Aplikace byly vytvořeny pro sběr a vizualizaci dat z meteostanice, která byla vyvinuta v předchozích letech v rámci diplomové práce Ing. Petra Mlčáka. Webová aplikace poskytuje REST API pro sběr dat z meteostanice a odesílání dat pro mobilní aplikaci, také poskytuje uživatelské webové rozhraní pro vizualizaci dat z meteostanice a jejich správu. V první části jsou popsány veličiny, které meteostanice měří a odesílá na server. V druhé části je popsán návrh databázové struktury pro ukládání dat. Třetí část práce je zaměřena na vývoj webové aplikace v ASP.NET, její části a strukturu. Popsány jsou také funkce a jejich význam ve webovém rozhraní. Čtvrtá část práce popisuje návrh a vývoj mobilní aplikace a principy využití pro získávání a zobrazení dat. V závěru jsou zhodnoceny výsledky práce a způsob testování aplikací.

Klíčová slova

Webová aplikace, mobilní aplikace, Android, online meteostanice, počasí, sběr dat, vizualizace dat, databáze, .NET 5, C#, IoT, Java, REST API, JSON, Entity Framework

Abstract

The thesis deals with the development of a web application in .NET 5 and a mobile application for the Android operating system. The applications were created for the collection and visualization of data from the weather station, which was developed in previous years within the diploma thesis of Ing. Petr Mlčák. The web application provides a REST API for collecting data from the weather station and sending data for the mobile application, it also provides a user web interface for visualizing data from the weather station and managing it. The first part describes the quantities that the weather station measures and sends to the server. The second part describes the design of a database structure for data storage. The third part of the work is focused on the development of a web application in ASP.NET, its parts and structure. The functions and their meaning in the web interface are also described. The fourth part describes the design and development of a mobile application and the principles used for data acquisition and display. In the end, the results of the work and the method of testing applications are evaluated.

Keywords

Web application, mobile application, Android, online weather station, weather, data collection, data visualization, database, .NET 5, C #, IoT, Java, REST API, JSON, Entity Framework

Bibliografická citace

ŠTURALA, Martin. *Webová a mobilní aplikace pro sběr a vizualizaci meteorologických dat* [online]. Brno, 2022 [cit. 2022-05-15]. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/141380>. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce Tomáš Čaha.

Prohlášení autora o původnosti díla

Jméno a příjmení studenta: *Martin Šturala*

VUT ID studenta: *186212*

Typ práce: *Diplomová práce*

Akademický rok: *2021/22*

Téma závěrečné práce: *Webová a mobilní aplikace pro sběr a vizualizaci meteorologických dat*

Prohlašuji, že svou závěrečnou práci jsem vypracoval samostatně pod vedením vedoucího závěrečné práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne: 20. května 2022

podpis autora

Děkuji vedoucímu semestrální práce Ing. Tomáši Cahovi za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé semestrální práce.

V Brně dne: 20. května 2022

podpis autora

Obsah

ÚVOD	11
1 METEOSTANICE A JEJÍ VLASTNOSTI	12
1.1 VELIČINY MĚŘENÉ METEOSTANICÍ	12
1.1.1 Teplota	12
1.1.2 Rosný bod.....	12
1.1.3 Pocitová teplota	13
1.1.4 Vlhkost vzduchu	13
1.1.5 Atmosférický tlak.....	13
1.1.6 Rychlost a směr větru	14
Srážky	15
1.1.7 UV index	15
1.1.8 Teplota a vlhkost půdy	15
1.1.9 Napětí baterie.....	15
1.2 PRINCIP ODESÍLÁNÍ DAT NA SERVER	16
2 NAVRŽENÁ DATABÁZOVÁ STRUKTURA	17
3 WEBOVÁ APLIKACE	20
3.1 STRUKTURA APLIKACE.....	20
3.1.1 Modely webové aplikace	21
3.1.2 Kontroléry webové aplikace.....	24
3.2 UŽIVATELSKÉ WEBOVÉ ROZHRANÍ	35
3.3 NASTAVENÍ A KONFIGURACE APLIKACE	43
3.3.1 Konfigurace databáze	44
3.3.2 Chybové stránky a adresa serveru	45
3.4 POUŽITÉ PRVKY.....	46
3.4.1 Bootstrap.....	46
3.4.2 Chart JS	46
3.4.3 MapQuest.....	47
3.5 TESTOVÁNÍ APLIKACE	47
4 MOBILNÍ APLIKACE	49
4.1 HLAVNÍ APLIKACE.....	49
4.1.1 Design aplikace a layout.....	49
4.1.2 Třídy a životní cyklus aplikace	51
4.1.3 Třída pro ukládání přijatých dat.....	53
4.1.4 Stahování dat	53
4.1.5 Nastavení	53
4.1.6 Konfigurace aplikace	53
4.2 WIDGET APLIKACE	54
4.2.1 Design	54
4.2.2 Třídy a funkce widgetu	55
4.2.3 Aktualizace a stahování dat	55
4.2.4 Konfigurace widgetu.....	56
4.3 TEXTY A LOGA V APLIKACI	56

4.4 TESTOVÁNÍ APLIKACE	56
ZÁVĚR	57
LITERATURA.....	58
SEZNAM SYMBOLŮ A ZKRATEK	60
SEZNAM PŘÍLOH.....	61

Seznam obrázků

Obrázek 1.1 Data z meteostanice interpretována programem Wireshark	16
Obrázek 2.1 Diagram návrhu tabulek v databázi	18
Obrázek 3.1Vzájemné závislosti mezi jednotlivými částmi aplikace	21
Obrázek 3.2Diagramy tříd, které jsou použity jako modely v kategorii Station.....	22
Obrázek 3.3 Diagramy tříd sloužící jako modely v kategorii Api	24
Obrázek 3.4 Logické rozdělení hlavního kontroléru.....	25
Obrázek 3.5 Diagram pro metodu FirstRun.....	25
Obrázek 3.6 Diagram pro metodu Login	26
Obrázek 3.7 Diagram pro metodu Index.....	27
Obrázek 3.8 Diagram pro metodu Map	28
Obrázek 3.9 Diagram pro metodu History	29
Obrázek 3.10 Diagram pro metodu Settings.....	30
Obrázek 3.11 Diagram pro metodu CreateStation	31
Obrázek 3.12 Diagram pro metodu DeleteStation	32
Obrázek 3.13 Rozdělení tříd v API kontroléru	32
Obrázek 3.14 Diagram pro třídu LastMeasurement HTTP POST	33
Obrázek 3.15 Diagram pro třídu LastMeasurement HTTP GET	34
Obrázek 3.16 Hlavní stránka webového rozhraní	36
Obrázek 3.17 Rozbalovací seznam pro výběr stanic v menu.....	36
Obrázek 3.18 Zobrazení stanic na mapě	37
Obrázek 3.19 Volba dat pro zobrazení historie měření.....	38
Obrázek 3.20 Graf pro zobrazení teploty	38
Obrázek 3.21 Graf pro zobrazení rychlosti a směru větru	39
Obrázek 3.22 Graf pro zobrazení úhrnu srážek	39
Obrázek 3.23 Společný graf pro zobrazení teploty a vlhkosti	40
Obrázek 3.24 Rozbalovací formuláře pro přihlášení a odhlášení	40
Obrázek 3.25 Přihlášení na samostatné stránce	41
Obrázek 3.26 Nastavení pro aktivní stanici a kartou pro vytvoření nové stanice	42
Obrázek 3.27 Poslední potvrzení při odstraňování stanice	43
Obrázek 3.28 Formulář pro vložení uživatelských údajů při prvním spuštění.....	43
Obrázek 3.29 Balíčky pro podporu Entity Frameworku a MySQL	44
Obrázek 3.30 Vlastní chybová stránka aplikace	46
Obrázek 3.31 Výstup nástroje curl při pokusu o vložení nové stanice bez zadání hesla	47
Obrázek 3.32Výstup nástroje curl při pokusu o vložení naměřených dat s neplatným API klíčem	48
Obrázek 4.1 Okno s nastavením v mobilní aplikaci	49
Obrázek 4.2 Hlavní okno mobilní aplikace se staženými daty	50
Obrázek 4.3 Ukazatel stahování dat.....	51
Obrázek 4.4 Zobrazení informace o výsledku stahování dat	51
Obrázek 4.5 Životní cykly fragmentu aplikace [16]	52
Obrázek 4.6 Aktualizovaný widget.....	54
Obrázek 4.7 Widget při neúspěšném stažení dat	54
Obrázek 4.8 Přidání widgetu na domovskou stránku.....	55

Seznam tabulek

Tabulka 1.1 Úhel natočení sensoru směru větru a odpovídající směr větru[1]	14
Tabulka 1.2 Hodnoty UV indexu a odpovídající riziko pro lidský organismus[9]	15
Tabulka 2.1 Struktura dat v tabulce stations	17
Tabulka 2.2 Struktura dat v tabulce logins	18
Tabulka 2.3 Struktura dat v tabulce measurements	19
Tabulka 3.1 Pohledy webové aplikace a jejich význam.....	35
Tabulka 3.2 Formáty času a matematické operace pro zvolené období.....	37
Tabulka 4.1 Metody třídy MainActivity a jejich vlastnosti	52

ÚVOD

Webové a mobilní aplikace jsou již každodenní součástí našeho života a jsou využívány téměř v každém odvětví. Na jedno z těchto mnoha odvětví, a to konkrétně na meteorologii je zaměřen i vývoj aplikací v této práci. Když si představíme domácí meteorologickou stanici, vybavíme si zpravidla malé zařízení, na kterém se zobrazuje teplota, vlhkost a tlak. Taková stanice funguje ovšem pouze v okruhu, kde se nachází a to může být omezením, pokud má být stanice umístěna na jiném místě, než kde se obvykle nacházíme. S rozšířením internetu a vývojem IoT zařízení je však možné mít o datech z vlastní meteostanice pořád přehled. Proto je ovšem nutné mít aplikaci, která by byla schopná data ze stanice ukládat a následně zobrazovat.

Právě vývojem takové aplikace se zabývá následující práce, která tímto navazuje na předchozí diplomovou práci Ing. Petra Mlčáka, který vytvořil IoT systém pro zahrádkáře. Tato meteostanice umožňuje měřit jak standartně měřené veličiny, tak i veličiny, které jsou užitečné pro pěstování plodin na zahrádce. Meteostanice od Ing. Petra Mlčáka umožňuje odesílání dat na vlastní server a díky této vlastnosti je možné data z meteostanice sbírat a ukládat pomocí aplikace do databáze a následně zpětně prohlížet historická data.

Cílem práce je vytvoření webové aplikace, ve které bude možné zobrazit aktuální i historická data z meteostanice v graficky srozumitelném a uživatelsky intuitivním prostředí. Dále bude také možné spravovat aktivní meteostanice, které budou do systému zaregistrovány. Aplikace také bude poskytovat API, které bude sloužit jako rozhraní pro mobilní aplikaci. V první části této práce jsou popsány veličiny, které meteostanice měří a jaký je jejich význam, také je popsán princip odesílání dat na server. V druhé části práce se seznámíme s návrhem a strukturou databáze navržené pro webovou aplikaci. Třetí část práce je zaměřena na tvorbu webové aplikace, v této části je popsána struktura aplikace, jak z pohledu vývoje, tak z pohledu funkcí pro uživatele. Ve čtvrté části je popsán vývoj a funkce mobilní aplikace pro operační systém Android. V poslední části práce jsou shrnuty výsledky práce a zhodnocení možností pro další využití vytvořených aplikací.

1 METEOSTANICE A JEJÍ VLASTNOSTI

Aplikace, jež byly vytvořeny v rámci této práce, byly navrženo na míru pro meteostanici, která vznikla v rámci diplomové práce s názvem IOT systém pro zahrádkáře, kterou vypracoval Ing. Petr Mlčák v předešlých letech. Tato meteostanice je vybavena sensory pro měření široké škály veličin od teploty až po vlhkost půdy ve třech úrovních. Meteostanice je postavena na platformě ESP32 a její komponenty jsou vyrobeny především pomocí 3D tisku. Meteostanice umožňuje odesílání naměřených dat na vlastní server a z toho důvodu byla vytvořena webová aplikace, která slouží ke sběru a vizualizaci těchto dat.

1.1 Veličiny měřené meteostanicí

Meteostanice odesílá na server 24 hodnot. Hodnoty jsou buď získány měřením, nebo výpočtem z naměřených hodnot [1]. Jelikož je meteostanice zaměřena na zahrádkáře, je měřena i teplota a vlhkost půdy ve třech hloubkách. V poslední řadě odesílá stanice také informaci o stavu své baterie, a protože je dobíjena pomocí solárního panelu, může to být v některých případech užitečná informace.

1.1.1 Teplota

Teplota je fyzikální vlastnost látek, která souvisí s jejich vnitřní energií. V souvislosti se sledováním změn teploty je tato veličina pro člověka spíše ukazatelem pocitů tepla nebo chladu. Ovšem samotná teplota není o těchto pocitech přímo vypovídající. Teplota tedy vyjadřuje fyzikálně aktuální stav měřené látky, což je v našem případě vzduch a půda.

1.1.2 Rosný bod

Rosný bod je synonymum pro teplotu, při které jsou vodní páry maximálně koncentrované ve vzduchu. Pokud teplota klesne pod teplotu rosného bodu, dochází ke kondenzaci vodních par ve vzduchu a tím ke tvorbě oblačnosti, mlhy nebo rosy[10]. Kondenzaci spojenou s rosným bodem je možné nejčastěji sledovat v létě na sklenici se studenou vodou, která se po chvíli orosí. Rosný bod je možné vypočítat podle rovnice(1.1)[1]z teploty a relativní vlhkosti. V rovnici jsou konstanty, které byly empiricky určeny dle výzkumu NOAA[11]. Rovnice je použita v meteostanici pro výpočet rosného bodu. Do vzorce dosazujeme hodnoty relativní vlhkosti RH [%] a teploty T[°C]. Výpočtem získáme hodnotu rosného bodu T_{dp} [°C].

$$T_{DP} = \frac{243.12 * \ln\left(\frac{RH}{100} * e^{\frac{17.62 * T}{243.12 + T}}\right)}{17.62 - \ln\left(\frac{RH}{100} * e^{\frac{17.62 * T}{243.12 + T}}\right)} \quad (1.1)[1]$$

1.1.3 Pocitová teplota

Pocitová teplota je pro člověka nejvíce vypovídající informace o tom, jak se bude venku cítit. Tuto teplotu lze vypočítat z teploty a rychlosti větru, takto získaná pocitová teplota je pojmenována v angličtině jako „windchill“ – chlad větru[13]. Setkat se můžeme také s názvem „heat index“ - index horka, který lze vypočítat z teploty a vlhkosti vzduchu. Takto získaný údaj je více vypovídající v letních obdobích[12].

Rovnice pro výpočet se mohou mírně lišit konstantami, které především souvisí s výškou, ve které je rychlost větru měřena. Konstanty v rovnici jsou opět získány empiricky z výzkumu dlouhodobých měření. Pro výpočet pocitové teploty je také možné využít rovnici, ve které jsou využity všechny tři zmíněné veličiny, teplota, vlhkost a rychlost větru. Tato rovnice je ideální pro klimatické podmínky v české republice a také je nejvíce vypovídající, protože počítá s nejvíce veličinami. Do rovnice(1.2)[1] se kterou počítá meteostanice je dosazována teplota T [°C], relativní vlhkost RH [%] a rychlost větru V [km/h]. Výpočtem je získána pocitová teplota T_{feel} [°C].

$$T_{feel} = T + \frac{0,33 * RH}{100 * 6,105 * e^{\left(\frac{17,27 * T}{237,7 + T}\right)}} - \frac{0,7 * V}{3,6} - 4 \quad (1.2)[1]$$

1.1.4 Vlhkost vzduchu

Vlhkost vzduchu je hodnota, která značí, jaké množství vodních par je ve vzduchu obsaženo. Relativní vlhkost udává poměr mezi aktuálním nasycením vzduchu vodními parami oproti tomu, pokud by byl vzduch nasycen vodními parami na 100%[15]. Tato hodnota se udává v procentech. Pokud by nebyla vlhkost vzduchu měřena, nebylo by možné vypočítat některé z předešlých rovnic.

1.1.5 Atmosférický tlak

Atmosférický tlak je vyvolán sloupcem vzduchu, který tlačí na zemský povrch. Tlak tedy úzce souvisí s nadmořskou výškou. Je ovšem ovlivňován různými faktory, jako je teplota a vlhkost vzduchu. Při měření je tedy získána hodnota, která je vypovídající pouze pro konkrétní polohu. Aby bylo možné hodnoty tlaku srovnávat, je tlak přepočítáván na tlak u hladiny moře. Pak lze jednoduše srovnat například měření z více stanic.

Z tlaku vzduchu lze určit proudění vzduchu, kdy vzduch proudí z místa s vyšším tlakem do míst s nižším tlakem[14]. Podle klesající a rostoucí tendence tlaku lze také předpovídat počasí. Při přepočtu tlaku vzduchu na hladinu moře byl stanoven tzv. normální tlak, jehož hodnota je 1013.25 hPa. Pokud je tlak přepočtený na hladinu moře

vyšší, než normální tlak je předpoklad, že bude slunečno, a naopak pokud je hodnota tlaku nižší, pravděpodobně bude zataženo až deštivo. Meteostanice využívá pro přepočítání absolutního tlaku na tlak u hladiny moře rovnici (1.3)[1]. Do rovnice je dosazován absolutní tlak P [hPa], nadmořská výška A [m] a teplota vzduchu T [°C]. Výpočtem je získána hodnota tlaku u hladiny moře $P_{seeLevel}$ [hPa] pro zadané vstupní veličiny.

$$P_{seeLevel} = P * \left(1 - \frac{0,0065 * A}{T + 0,0065 * A + 273,15}\right)^{-5,257} \quad (1.3)[1]$$

1.1.6 Rychlost a směr větru

Rychlost větru je veličina, která se měří pomocí anemometru. Jedná se o rychlost proudění vzduchu, která je udávána vůči zemskému povrchu. Směr proudění vzduchu určuje sensor směru větru. U sensoru směru větru je nutná kalibrace nebo správné umístění vůči světovým stranám tak, aby hodnoty odpovídaly realitě. Meteostanice měří i tyto veličiny.

Tabulka 1.1 Úhel natočení sensoru směru větru a odpovídající směr větru[1]

Úhel[°]	Směr(EN)	Směr(CZ)
0	N	S
22,5	NNE	SSV
45	NE	SV
67,5	ENE	VSV
90	E	V
112,5	ESE	VJV
135	SE	JV
157,5	SSE	JJV
180	S	J
202,5	SSW	JJZ
225	SW	JZ
147,5	WSW	ZJZ
270	W	Z
292,5	WNW	ZSZ
315	NW	SZ
337,5	NNW	SSZ

Směr větru měří meteostanice v 16 krocích[1]. Výsledky směru větru jsou meteostanicí interpretovány dvojím způsobem, a to tak, že vrací úhel natočení a světovou stranu, kdy úhel natočení 0° odpovídá směru proudění na sever viz Tabulka 1.1. Směr větru vyjadřuje údaj „odkud vítr vane“, to znamená, že směr S udává proudění vzduchu ze severu na jih. Meteostanice vrací hodnoty v anglickém znění, v aplikacích je tedy směr větru překládán do češtiny dle Tabulka 1.1

Srážky

Meteostanice disponuje také senzorem pro měření intenzity deště. Množství srážek je udáváno v mm. Z měření lze následně vypočítat úhrn srážek za určité období a zpětně získat statistiky například za rok. Pro grafické zobrazení úhrnu srážek je vhodné použít sloupcové grafy.

1.1.7 UV index

UV index udává hodnotu UV záření dopadající na zemský povrch. 1 stupeň UV indexu je možné srovnat s intenzitou UV záření o hodnotě 25 mW/m²[9]. Podle hodnoty UV indexu lze přizpůsobit ochranu proti slunečnímu záření. Hodnota UV indexu je také závislá na geografické poloze, kdy blíže k rovníku UV index vzrůstá. Hodnoty UV indexu jsou rozděleny podle úrovně rizika pro lidský organismus viz Tabulka 1.2. Barvy v tabulce jsou všeobecně využívány pro označení rizika UV indexu.

Tabulka 1.2 Hodnoty UV indexu a odpovídající riziko pro lidský organismus[9]

Hodnota UV indexu	Riziko
0-2	Nízké
3-5	Střední
6-7	Vysoké
8-10	Velmi vysoké
11+	Extrémní

1.1.8 Teplota a vlhkost půdy

Teplota a vlhkost půdy jsou velmi užitečné údaje pro zahrádkáře. Tyto veličiny měří meteostanice ve třech úrovních. Pro tyto čidla je nutné vybrat vhodnou polohu, a to s ohledem na to, jak chce uživatel tyto hodnoty využívat. Je možné sledovat například míru zavlažení přímo v záhonu nebo lze sledovat tyto hodnoty na půdě, která není ovlivněna zaléváním.

1.1.9 Napětí baterie

Stav baterie meteostanice je udáván ve voltech. Tento údaj je v případě velmi nízké spotřeby meteostanice spíše informativní a může být pro uživatele brán jako ukazatel pro výměnu baterie.

1.2 Princip odesílání dat na server

Meteostanice odesílá data na server pomocí protokolu HTTP metodou POST ve formátu typu „application/x-www-form-urlencoded“ což znamená, že jsou proměnné odděleny znakem „&“ a hodnota proměnné znakem „=“. Takto odeslaná data jsou kódovaná do validního URL formátu[27] a mohou vypadat pak následovně: „teplota=20&vlhkost=54“. Meteostanice tímto způsobem odesílá všech 24 hodnot. Data, která nejsou celočíselná, využívají pro oddělení desetinnou tečku. Zachycená příchozí data z meteostanice pomocí programu Wireshark a jejich interpretace jsou zobrazena na Obrázek 1.1.

```
soil_temp_3=10.06&battery_voltage=4.06%0D&soil_moisture_depth_3=-5+cm&soil_moisture_1=35.71&soil_temp_depth_3=0+cm&soil_moisture_2=33.94&soil_moisture_depth_1=-35+cm&soil_moisture_depth_2=-20+cm&rain_amount=0.00&pressure=949.49&temperature_feels_like=8.21&soil_temp_depth_2=-20+cm&soil_moisture_3=24.34&soil_temp_2=16.06&soil_temp_1=13.88&soil_temp_depth_1=-35+cm&wind_angle=45.0&uv_index=0.0&dew_point=8.54&wind_angle_str=NE&temperature=8.54&humidity=100&wind_speed=0.00&sea_level_pressure=1011.42
```

```
▼ HTML Form URL Encoded: application/x-www-form-urlencoded
  > Form item: "soil_temp_3" = "10.06"
  > Form item: "battery_voltage" = "4.06"
  > Form item: "soil_moisture_depth_3" = "-5 cm"
  > Form item: "soil_moisture_1" = "35.71"
  > Form item: "soil_temp_depth_3" = "0 cm"
  > Form item: "soil_moisture_2" = "33.94"
  > Form item: "soil_moisture_depth_1" = "-35 cm"
  > Form item: "soil_moisture_depth_2" = "-20 cm"
  > Form item: "rain_amount" = "0.00"
  > Form item: "pressure" = "949.49"
  > Form item: "temperature_feels_like" = "8.21"
  > Form item: "soil_temp_depth_2" = "-20 cm"
  > Form item: "soil_moisture_3" = "24.34"
  > Form item: "soil_temp_2" = "16.06"
  > Form item: "soil_temp_1" = "13.88"
  > Form item: "soil_temp_depth_1" = "-35 cm"
  > Form item: "wind_angle" = "45.0"
  > Form item: "uv_index" = "0.0"
  > Form item: "dew_point" = "8.54"
  > Form item: "wind_angle_str" = "NE"
  > Form item: "temperature" = "8.54"
  > Form item: "humidity" = "100"
  > Form item: "wind_speed" = "0.00"
  > Form item: "sea_level_pressure" = "1011.42"
```

Obrázek 1.1 Data z meteostanice interpretována programem Wireshark

2 NAVRŽENÁ DATABÁZOVÁ STRUKTURA

Při tvorbě aplikace byla využita databáze MySQL[6]. Tabulky, které slouží pro ukládání dat do databáze byly navrženy tak, aby bylo možné do aplikace v budoucnu přiřadit více meteostanic. Struktura tabulek je definována ve webové aplikaci. Protože byl zvolen přístup pro psaní webové aplikace „codefirst“ není nutné tyto tabulky ručně vytvářet, tabulky jsou automaticky vytvořeny při inicializaci aplikace.

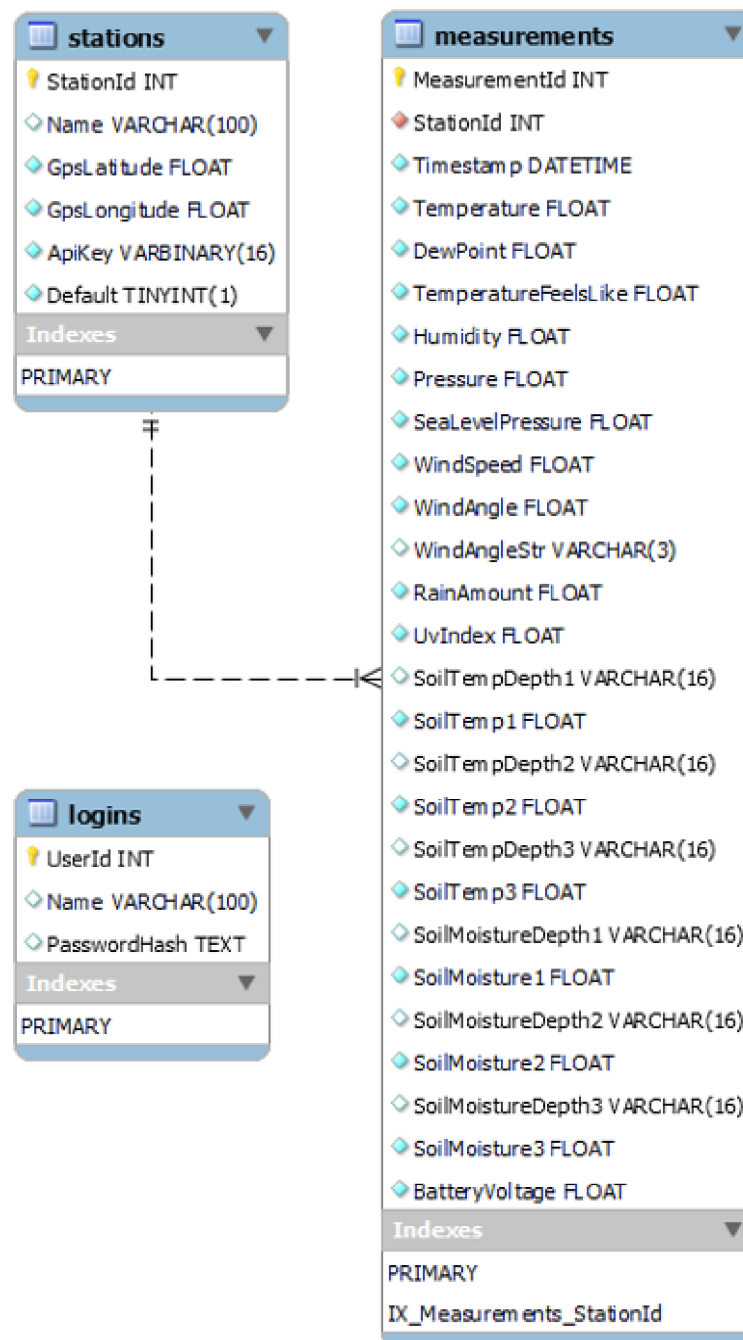
Pro aplikaci jsou v databázi vytvořeny tři tabulky. Tabulka stations viz Tabulka 2.1, která slouží k evidenci registrovaných meteostanic. Tato tabulka obsahuje unikátní primární klíč StationId typu INT, který slouží k identifikaci meteostanice a je využit k propojení 1:N pomocí nevlastního klíče s tabulkou measurements viz Obrázek 2.1. Dále obsahuje položku Name typu VARCHAR o maximální velikosti 100 znaků pro uložení názvu meteostanice. Pro uložení GPS souřadnic jsou zde položky GpsLatitude a GpsLongitude typu FLOAT. Pro uložení API klíče slouží položka ApiKey typu VARBINARY o velikosti 16 B, velikost 16 B odpovídá velikosti GUID pro danou stanici. Poslední položka Default typu TINYINT o velikosti 1 slouží pro uložení logické hodnoty o výchozí stanici pro zobrazení.

Tabulka measurements obsahuje všechna data, která jsou meteostanicí měřena, včetně časového razítka viz Tabulka 2.3. Pro identifikaci dat vůči tabulce stations slouží položka typu INT StationId, která je zároveň nevlastním klíčem v tabulce. Položka MeasurementId typu INT pak slouží jako primární klíč v tabulce. Položka Timestamp typu DATETIME je využita k ukládání data a času, kdy jsou data vkládána do tabulky. Dále jsou v tabulce položky typu FLOAT a VARCHAR, které slouží k ukládání naměřených dat. Položky typu VARCHAR mají stanovenou maximální velikost podle dat, které meteostanice odesílá, kvůli úspoře místa.

Tabulka logins obsahuje přihlašovací údaje uživatele. Tabulku tvoří položky UserId typu INT, Name typu VARCHAR o velikosti 100 znaků a položku PsswordHash typu TEXT pro uložení hashe hesla v textové podobě.

Tabulka 2.1 Struktura dat v tabulce stations

Stations		
StationId	INT PRIMARY KEY	Id meteostanice, unikátní klíč
Name	VARCHAR(100)	Vlastní název stanice
GpsLatitude	FLOAT	GPS zeměpisná šířka
GpsLongitude	FLOAT	GPS zeměpisná délka
ApiKey	VARBINARY(16)	API klíč pro identifikaci při sběru dat
Default	TINYINT(1)	Určení výchozí stanice pro zobrazení



Obrázek 2.1 Diagram návrhu tabulek v databázi

Tabulka 2.2 Struktura dat v tabulce logins

Logins		
UserId	INT PRIMARY KEY	Id uživatele, unikátní klíč
Name	VARCHAR(100)	Uživatelské jméno
PasswordHash	TEXT	Hash uživatelského hesla

Tabulka 2.3 Struktura dat v tabulce measurements

Measurements		
MeasurementId	INT FOREIGN KEY	Id měření, unikátní klíč
StationId	INT PRIMARY KEY	Id stanice, nevlastní klíč
Timestamp	DATETIME	Čas měření/uložení dat
Temperature	FLOAT	Teplota vzduchu
DewPoint	FLOAT	Rosný bod
TemperatureFeelsLike	FLOAT	Pocitová teplota
Humidity	FLOAT	Vlhkost vzduchu
Pressure	FLOAT	Atmosférický tlak
SeaLevelPressure	FLOAT	Tlak přepočtený na hladinu moře
WindSpeed	FLOAT	Rychlost větru
WindAngle	FLOAT	Úhel větru
WindAngleStr	VARCHAR (3)	Směr větru
RainAmount	FLOAT	Množství srážek
UvIndex	FLOAT	UV index
SoilTempDepth1	VARCHAR (16)	Hloubka čidla teploty v úrovni 1
SoilTemp1	FLAT	Teplota půdy v úrovni 1
SoilTempDepth2	VARCHAR (16)	Hloubka čidla teploty v úrovni 2
SoilTemp2	FLOAT	Teplota půdy v úrovni 2
SoilTempDepth3	VARCHAR (16)	Hloubka čidla teploty v úrovni 3
SoilTemp3	FLOAT	Teplota půdy v úrovni 3
SoilMoistureDepth1	VARCHAR (16)	Hloubka čidla vlhkosti v úrovni 1
SoilMoisture1	FLOAT	Vlhkost půdy v úrovni 1
SoilMoistureDepth2	VARCHAR (16)	Hloubka čidla vlhkosti v úrovni 2
SoilMoisture2	FLOAT	Vlhkost půdy v úrovni 2
SoilMoistureDepth3	VARCHAR (16)	Hloubka čidla vlhkosti v úrovni 3
SoilMoisture3	FLOAT	Vlhkost půdy v úrovni 3
BatteryVoltage	FLOAT	Napětí baterie

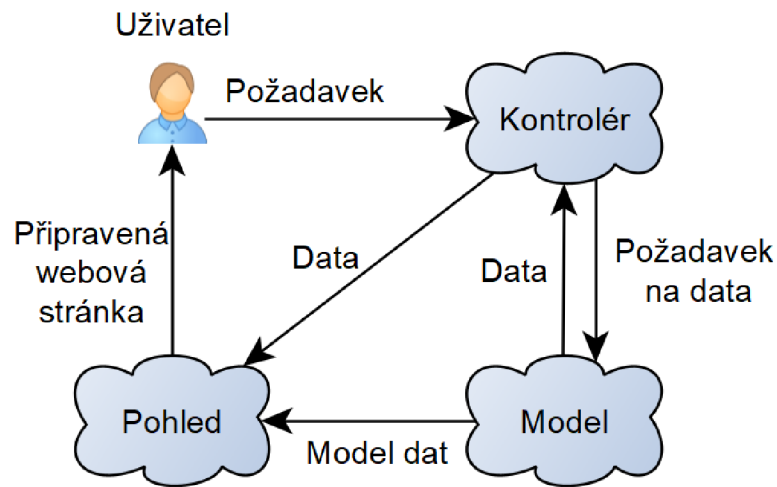
3 WEBOVÁ APLIKACE

Webová aplikace byla vytvořena za pomoci frameworku .NET 5, což je framework, který umožňuje tvorbu webových aplikací v jazyce C#. Výhodou dynamické webové stránky nebo také webové aplikace je, že vše běží na straně serveru a uživatelé jsou dodávány vygenerované statické stránky a není tak zatěžován klientský počítač[4]. Skutečnost, že je aplikace spuštěna na serveru, také umožňuje implementaci API pro odběr nebo vkládání dat do databáze z IoT zařízení. Webová aplikace byla navržena se zaměřením na cílovou architekturu .NET 5, která umožňuje multiplatformní využití[20]. Díky multiplatformní podpoře .NET 5 je jednodušší instalace výsledné aplikace

i v domácím prostředí na vlastní server nebo počítač. Navržená webová aplikace poskytuje uživateli webové rozhraní s možností přidávání nových meteostanic a následné zobrazování aktuálních i historických hodnot měření. Webová aplikace poskytuje pro zadané meteostanice API pro vkládání dat. Pro mobilní aplikaci pak poskytuje API pro odběr dat pro zvolenou stanici. Uživateli je díky webovému rozhraní aplikace poskytnut přehled o sledovaných meteostanicích.

3.1 Struktura aplikace

Aplikace byla navržena podle konceptu MVC(Model-View-Controller)Model, Pohled a Kontrolér. Aplikace je tedy logicky rozdělena do tří částí, kdy každá z nich plní jednotlivé funkce při běhu aplikace. Model obsahuje definici tříd s vlastnostmi pro propojení a práci s databází. Kontrolér zpracovává požadavky od uživatele, komunikuje s modely a získaná data z modelů zpracovává a předává pohledům pro zobrazení. Pohled pak můžeme brát jako poslední krok, při kterém jsou data z kontroléru vyčtena a vložena do výsledné stránky, která je následně zobrazena uživateli viz Obrázek 3.1.



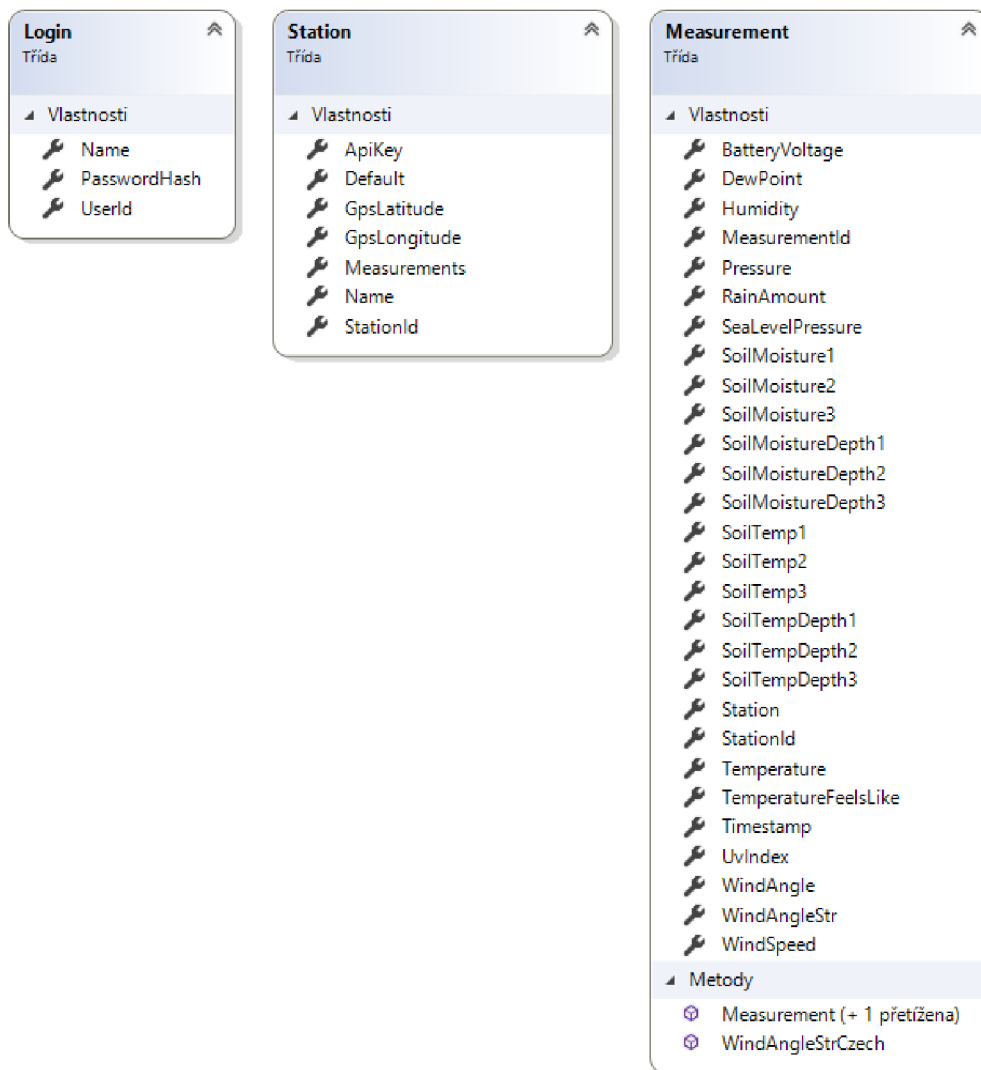
Obrázek 3.1 Vzájemné závislosti mezi jednotlivými částmi aplikace

3.1.1 Modely webové aplikace

Pro webovou aplikaci byly použité modely rozděleny podle použití na dvě části. První část modelů je použita pro data meteostanice a propojení s databází. Tato část modelů pojmenovaná jako Station se skládá ze tří tříd. Tyto třídy obsahují vlastnosti pro popis meteostanice, měřených data a uživatelských údajů pro přihlášení. Vlastnosti těchto tříd jsou zároveň opatřeny datovými anotacemi pro popis vlastností dat v databázové struktuře. Druhá část modelů je využívána v API. Tato část modelů obsahuje dvě třídy, jedna je využívána pro příjem dat z meteostanice a druhá je využita pro odesílání dat do mobilní aplikace.

Třídy sdílených modelů

Sdílená kategorie modelů je pojmenována Station a skládá ze tří tříd. První třída Login obsahuje vlastnosti `UserId` pro ukládání uživatelského id, `Name` pro uživatelské jméno a `PasswordHash` pro uložení hashe uživatelského hesla v textové podobě viz Obrázek 3.2



Obrázek 3.2 Diagramy tříd, které jsou použity jako modely v kategorii Station

Druhá třída Station obsahuje vlastnosti, které nesou informace o fyzické stanici. StationId je unikátní id pro každou stanici, Name slouží pro uložení názvu stanice dle volby uživatele, GpsLongitude a GpsLatitude slouží pro vložení GPS souřadnic, kde se stanice nachází. Tyto souřadnice jsou následně využity pro zobrazení stanice na mapě. ApiKey slouží pro uložení vygenerovaného Api klíče, který se následně používá pro identifikaci stanice při příjmu dat, Default je identifikátor, kterým je určeno, která stanice bude zobrazena na hlavní stránce webového rozhraní. Measurements je kolekce dat typu Measurement a používá se pro identifikaci nevlastního klíče při tvorbě databáze viz Obrázek 3.2. Vlastnosti v těchto třídách jsou opatřeny datovými anotacemi, pokud je to potřeba. Je tak určen například primární klíč pro danou tabulku nebo maximální délka textového řetězce. Například takto:

```
[Key]
public int StationId { get; set; }
[StringLength(100)]
public string Name { get; set; }
```

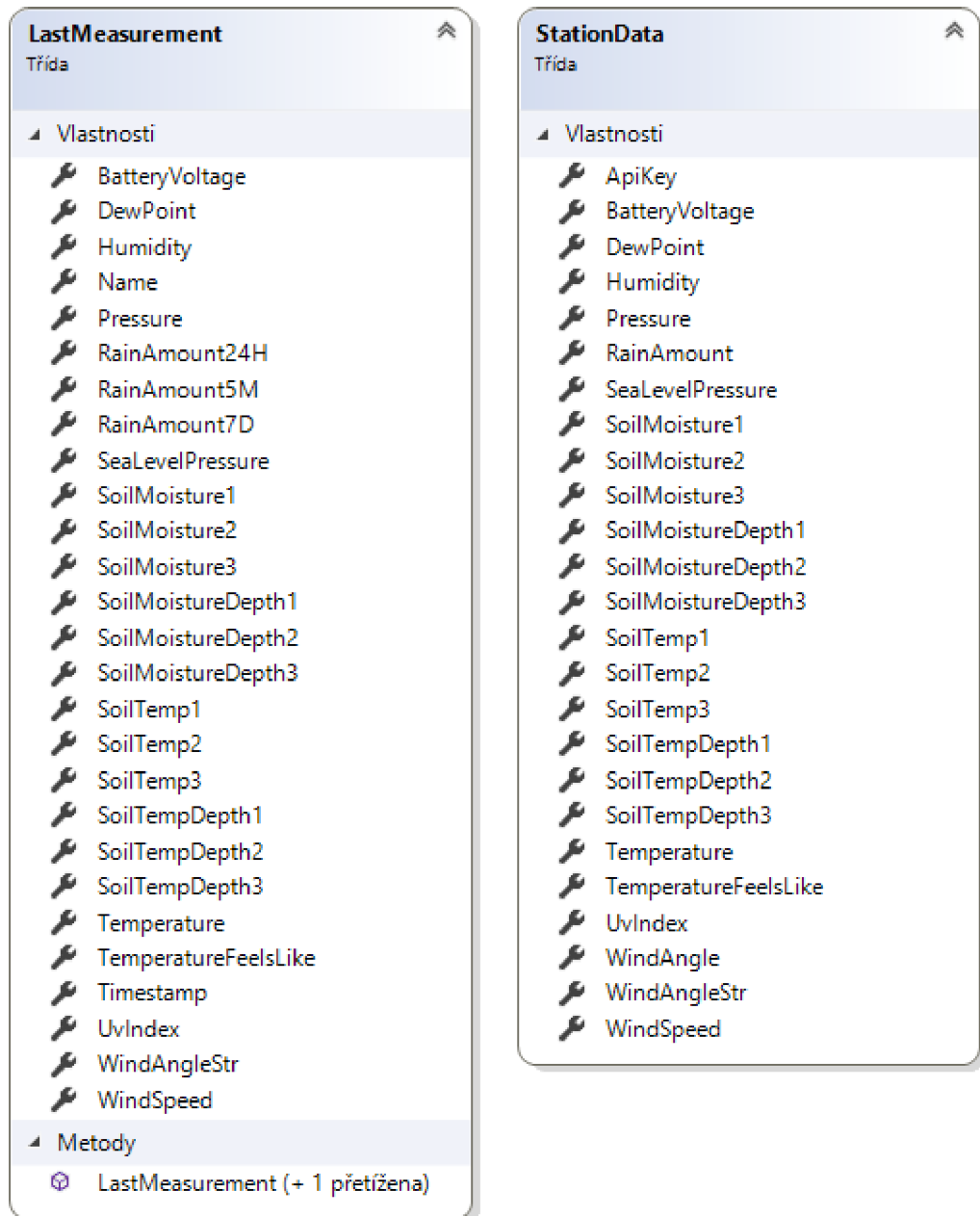
Třetí třída `Measurement` obsahuje vlastnosti popisující naměřená data ze stanice. Tato třída oproti datům z meteostanice obsahuje navíc vlastnost `MeasurementId` pro unikátní identifikaci dat v databázi, `StationId` pro identifikaci nevlastního klíče vůči třídě `Station` a položku `Timestamp`, která je přidělena přijatým datům při zpracování. Třída obsahuje také metody, kdy jedna z nich je přetížitelný konstruktor, který přijímá parametry `StationData` viz Obrázek 3.3 a `StationId`. Tento přetížitelný konstruktor je využíván při příjmu dat ze stanice a poskytuje možnost jednoduchého naplnění vlastností dle parametrů již při vytvoření instance třídy. Další metoda `WindAngleStrCzech` slouží pro překlad anglické zkratky směru větru do češtiny. Diagram třídy `Measurement` je znázorněn na Obrázek 3.2

Modely určené pro API

Další kategorie modelů je pojmenována `Api`, protože je využita v API kontroléru. Kategorie zahrnuje dvě třídy, třídu `StationData` a třídu `LastMeasurement`. Třída `StationData` obsahuje vlastnosti, které přesně odpovídají svým typem datům, které odesílá meteostanice viz Obrázek 3.3. Vlastnosti v této třídě jsou navíc opatřeny atributem `BindProperty`[26], aby svým názvem přesně odpovídaly názvům přijatých dat. Atribut tak zajistí, že při požadavku na příjem dat jsou vlastnosti instance třídy automaticky naplněny přijatými hodnotami. Využití Atributu `BindProperty` u vlastností třídy je zapsána níže.

```
[BindProperty(Name = "api_key")]  
public GuidApiKey { get; set; }
```

Třída `LastMeasurement` je v API kontroléru využita pro odesílání dat. Oproti třídě `StationData` obsahuje navíc vlastnosti `Name` s názvem meteostanice, `RainAmount5M`, `RainAmount24H` a `RainAmount7D`, které obsahují úhrn srážek za dané období a `Timestamp` s časovým údajem o záznamu měření viz Obrázek 3.3.



Obrázek 3.3 Diagramy tříd sloužící jako modely v kategorii Api

3.1.2 Kontroléry webové aplikace

Kontrolér je pro webovou aplikaci nejdůležitější částí pro zpracování dat. Jedná se o mezičlánek, který zpracovává uživatelské požadavky, pracuje s modely a předává zpracovaná data do pohledů pro zobrazení. Kontrolér obsahuje metody, které mají stejný název jako jednotlivé stránky webového rozhraní. Tyto metody jsou následně volány, pokud přijde požadavek pro zobrazení některé ze stránek webového rozhraní. Metody mají návratovou hodnotu typu „ActionResult“, což je objekt, který je odeslán po provedení metody zpět do prohlížeče. Tento objekt je v tomto případě view (pohled) pro danou stránku webového rozhraní. Návratová hodnota, která je předána prohlížeči

ovšem může být i stavový HTTP kód nebo čistý text. V kontroléru mohou existovat také metody, které slouží k obsluze POST požadavků a nemusí mít stejná název jako pohled. Navržená webová aplikace obsahuje dva kontroléry. Hlavní kontrolér HomeController a kontrolér pro API ApiController.

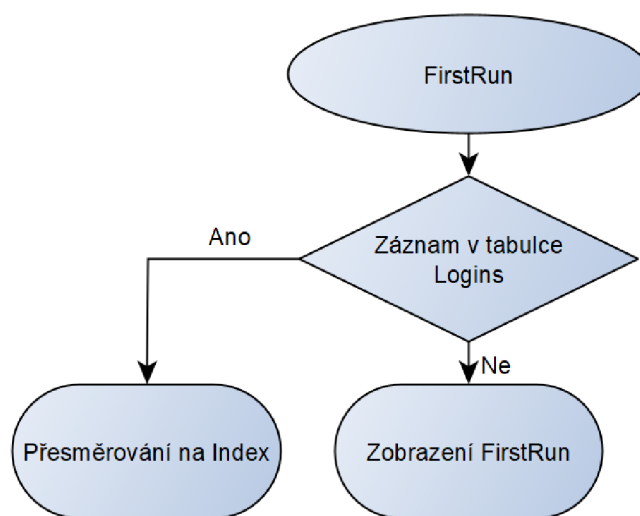
Hlavní kontrolér

Hlavní kontrolér je v navržené webové aplikaci pojmenován jako HomeController. Tento kontrolér obsahuje metody, pro obsluhu uživatelských požadavků na zobrazení daného pohledu a také metody pro zpracování POST požadavků z odeslaných formulářů. HomeController lze logicky rozdělit do sekcí podle toho, jaký zpracovávají požadavek a ke kterému patří pohledu viz Obrázek 3.4.



Obrázek 3.4 Logické rozdělení hlavního kontroléru

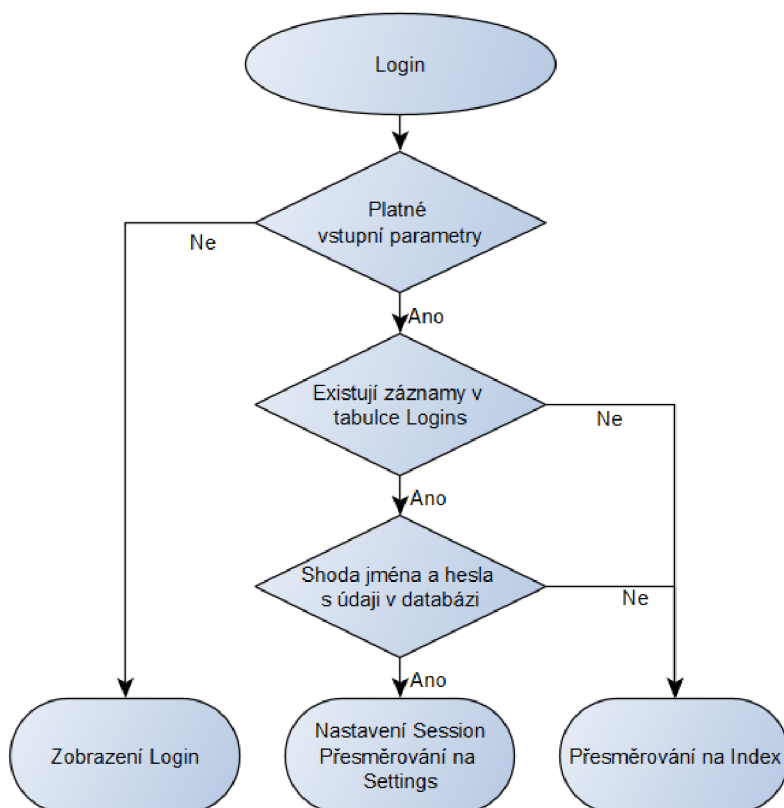
Sekce FirstRun obsahuje metodu, která vrací pohled FirstRun jen pokud nejsou v databázi v tabulce Logins žádné záznamy. Pokud záznam existuje, je vyvolána metoda Index pro zobrazení hlavní stránky viz Obrázek 3.5.



Obrázek 3.5 Diagram pro metodu FirstRun

Sekce Login se skládá z metod Login, SetLogin a Logout. Sekce zahrnuje všechny metody, které mají na starosti správu přihlášení. Metoda Login je vyvolána na základě POST požadavku nebo přesměrování z jiné sekce, přijímá nepovinné parametry name a pass. Pokud nejsou parametry přijaty, je vrácen pohled Login pro zadání jména a hesla. Pokud jsou parametry přijaty, je zjištěno, zda existují záznamy v tabulce Logins a přijaté údaje jsou porovnány s údaji v databázi. Při shodě je nastaven Session klíč Authorized na hodnotu true a klíč User na hodnotu jména uživatele a následně

je vyvolána metoda Settings pro zobrazení nastavení. Pokud není nalezena shoda, je vyvolána metoda pro zobrazení hlavní stránky viz Obrázek 3.6.

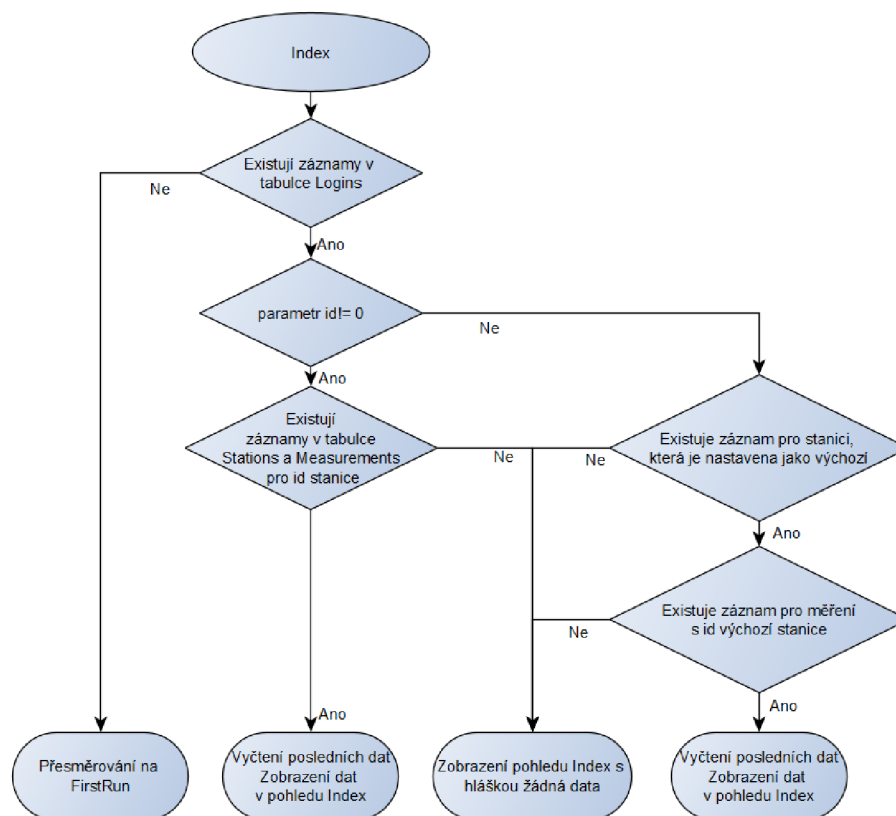


Obrázek 3.6 Diagram pro metodu Login

Metoda SetLogin je vyvolána pouze na základě formuláře, který je uživateli zobrazen v pohledu FirstRun. Metoda přijímá parametry name a pass. Pokud existuje v databázi v tabulce Logins záznam je vrácen pohled Index. Pokud je tabulka prázdná, je jméno a heslo vloženo do databáze a je vyvolána metoda pro zobrazení nastavení. Pro heslo je vytvořen hash pomocí hashovací funkce SHA256 hash hesla je uložen v databázi v textové podobě.

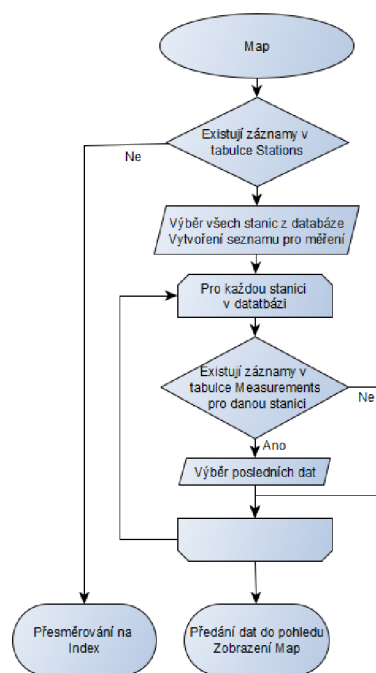
Metoda Logout je vyvolána na základě požadavku na odhlášení od uživatele. Pro odhlášení a zamezení přístupu do zabezpečené sekce jsou vymazány klíče Session.

Sekce Index zahrnuje metodu Index a přijímá nepovinný parametr id, který slouží pro výběr stanice, ze které mají být na pohledu Index zobrazena data. Nejprve je provedena kontrola, zda existuje záznam v tabulce Logins, pokud záznam neexistuje je vyvolána metoda pro zobrazení pohledu FirstRun. V případě, že záznam existuje, je zjištěno, zda existují záznamy pro vybranou stanici. Pokud ano, jsou z databáze vyčtena poslední naměřená data a informace o stanici jsou předána do pohledu pro zobrazení. Pokud v databázi neexistují záznamy pro vybranou meteostanici nejsou pohledu předána žádná data a uživateli je v pohledu oznámeno, že pro danou stanici neexistují žádné záznamy viz Obrázek 3.7.



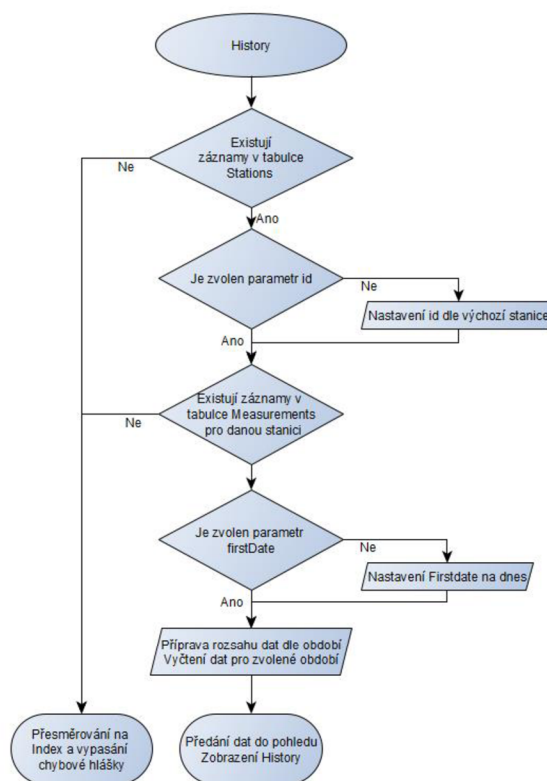
Obrázek 3.7 Diagram pro metodu Index

Sekce Map zahrnuje metodu Map pro zobrazení pohledu s mapou meteostanic. V metodě je nejdříve zjištěno, zda existují záznamy v tabulce pro Stanice. Pokud záznamy existují, jsou z databáze vybrány údaje o stanicích a jejich poslední záznamy. Tyto data jsou předány do pohledu a pohled s mapou stanic je vrácen uživateli viz Obrázek 3.8.



Obrázek 3.8 Diagram pro metodu Map

Sekce History obsahuje metodu History, která přijímá nepovinné parametry z formuláře pro zobrazení historie měření. Pokud uživatel poprvé zadá požadavek z hlavního menu pro zobrazení historie měření, pak nejsou metodě v kontroléru předány žádné parametry, z toho důvodu jsou parametry nepovinné. Při této akci jsou uživateli vybrána základní data pro zobrazení v pohledu. Vstupními parametry jsou stationId pro identifikaci stanice, reportType pro určení, jaké časové rozmezí pro data má být zvoleno, firstDate pro určení počátku období pro zobrazení a List graphs s názvy grafů, které se mají uživateli vykreslit. Na začátku je ověřeno, jestli existují záznamy pro stanice a měření v databázi. Pokud ano, pokračuje se přípravou dat na základě přijatých parametrů. Pokud parametry přijaty nebyly, jsou použity výchozí hodnoty pro zobrazení grafů historie měření. Podle těchto údajů jsou data vyčtena z databáze a jsou předána pohledu pro zobrazení viz Obrázek 3.9.



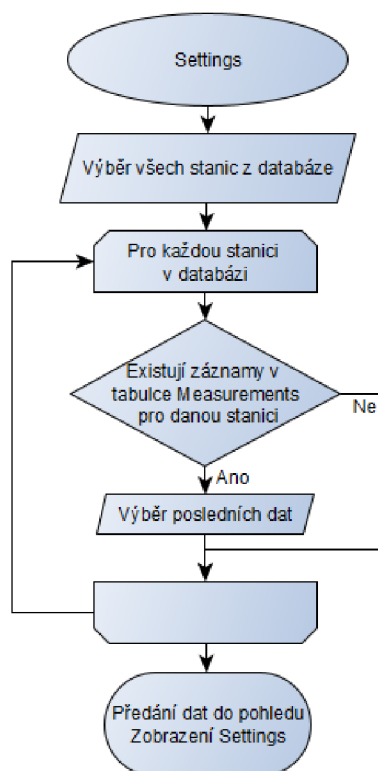
Obrázek 3.9 Diagram pro metodu History

Sekce Settings zahrnuje čtyři metody a jelikož se jedná o sekci, kde je možné manipulovat s daty v databázi, upravovat a mazat naměřená data je těmto metodám předřazen filtr akcí pojmenovaný jako Login. Kód filtru akcí je vždy spuštěn před vykonáním metody, která je tímto filtrem opatřena. Kód filtru ověřuje, zdali je v Session nastaven klíč Authorized na hodnotu true. Pokud klíč není nastaven je uživatel automaticky přesměrován na výchozí přihlašovací stránku Login. Filtr akcí pro přihlášení je zapsán níže.

```

public class LoginAttribute : ActionFilterAttribute
{
public override void OnActionExecuting(ActionExecutingContext
filterContext)
{
byte[] sessionKey;
if (filterContext.HttpContext.Session.TryGetValue("Authorized",
out sessionKey))
{
if (Encoding.UTF8.GetString(sessionKey) == "true")
{
// User is authorized - do nothing
return;
}
}
// User is not authorized redirect to default login page
string loginUrl = "/Home/Login";
filterContext.Result = new RedirectResult(loginUrl, true);
}}
  
```

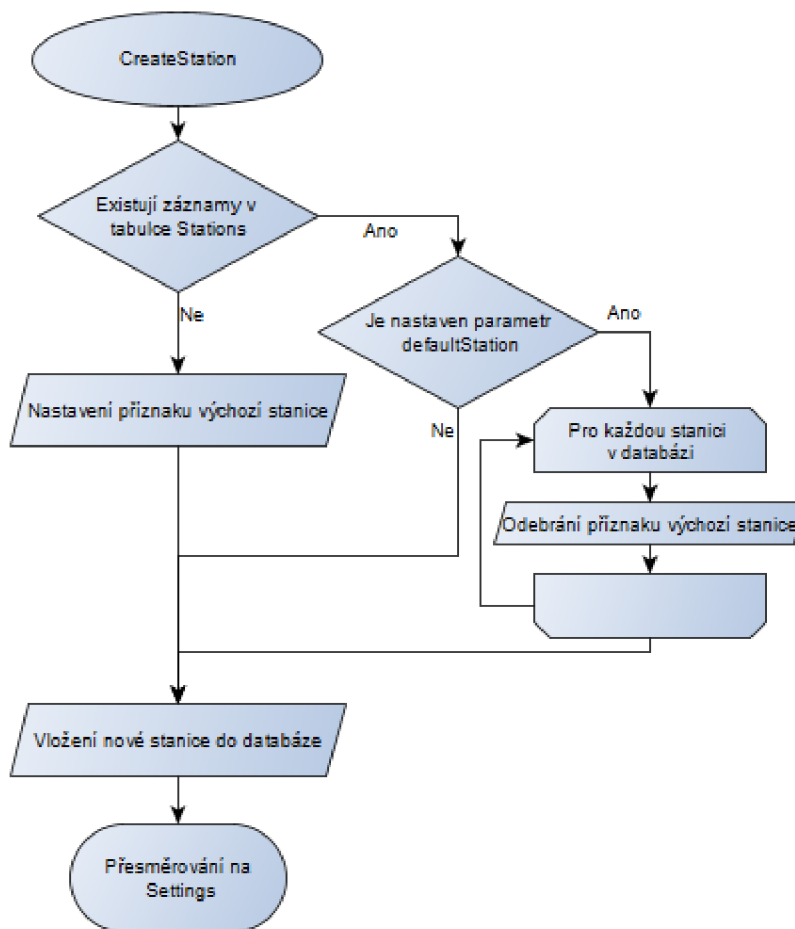
Sekce obsahuje metodu Settings, která je vyvolána při dotazu na zobrazení nastavení. V této metodě jsou vyčteny všechny stanice. Pro každou stanici je vyčteno poslední měření, pokud existuje. Tyto data jsou předána pohledu Settings pro zobrazení. Z těchto dat jsou důležité především informace o stanici, kterou tvoří třída Station a z posledního měření hodnota Timestamp a BatteryVoltage. Tyto hodnoty jsou zobrazeny v nastavení pro každou stanici a některé z nich je zde možné měnit viz Obrázek 3.10.



Obrázek 3.10 Diagram pro metodu Settings

Dále sekce zahrnuje metodu CreateStation zpracovávající POST požadavky. Metoda přijímá povinné parametry name, gpsLatitude a gpsLongitude. Tyto parametry jsou předány z formuláře pro vytvoření nové stanice a jsou pro uživatele volitelné. Dále přijímá nepovinný parametr defaultStation, který slouží k indikaci stanice pro výchozí zobrazení. Pokud není tento parametr předán je automaticky nastaven na hodnotu false. Při vyvolání metody je zjištěno, zdali je stanice zvolena jako výchozí, pokud ano, pak je všem ostatním stanicím v databázi nastavena tato hodnota na false.

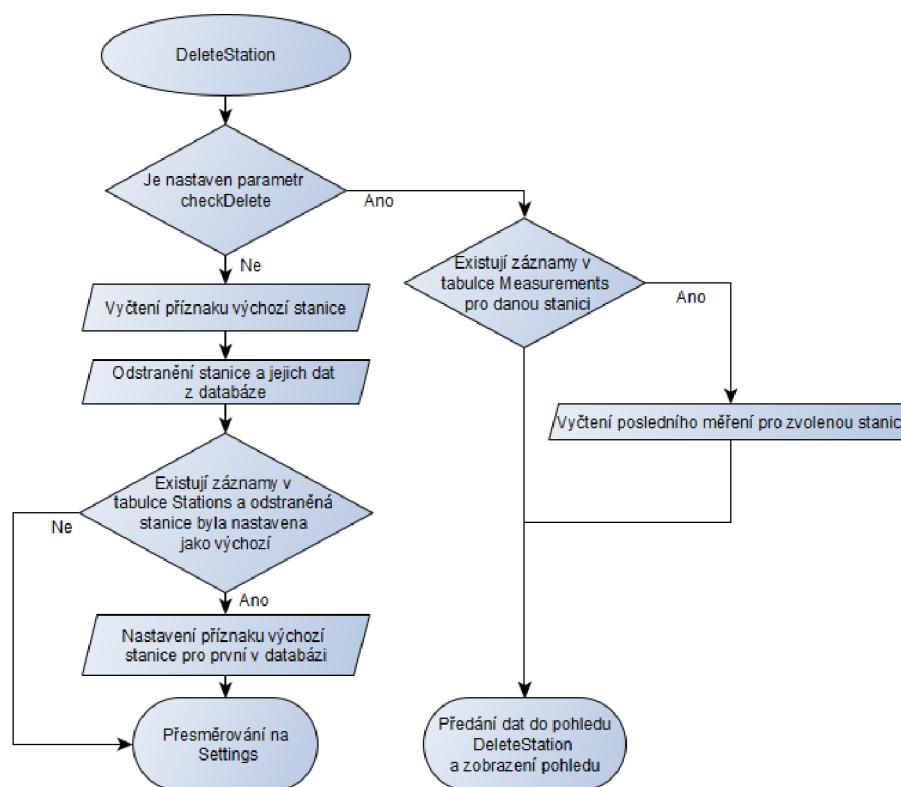
Pokud se jedná o první vytvářenou stanici, je jí automaticky nastaven příznak výchozí stanice. Následně jsou data z přijatých parametrů vložena do instance třídy Station a tento objekt je vložen do databáze jako záznam o nové stanici. Nakonec je uživatel přeměrován opět do nastavení, kde je zobrazena karta s novou stanicí viz Obrázek 3.11.



Obrázek 3.11 Diagram pro metodu CreateStation

Metoda ModifyStation slouží pro úpravu údajů o stanici. Přijímá parametry StationId pro identifikaci stanice, dále name, gpsLatitude, gpsLongitude a nepovinný defaultStation. Při vyvolání metody je opět jako u předchozí metody zjištěno, zdali je stanice zvolena jako výchozí, pokud ano pak je všem ostatním stanicím v databázi nastavena tato hodnota na false. Následně je vybrán z databáze záznam zvolené stanice dle id a jsou upraveny hodnoty dle vstupních parametrů. Uživateli je po dokončení vrácen pohled s nastavením a již upravenou stanicí.

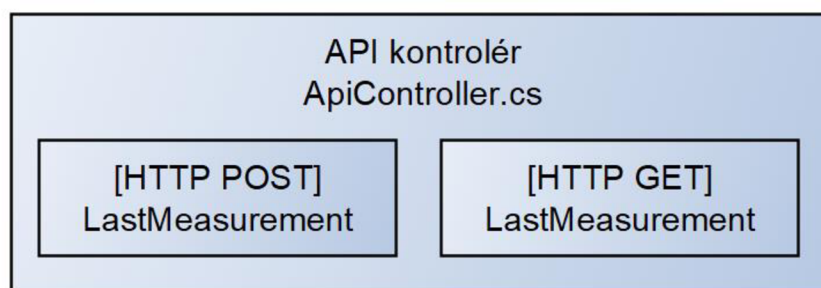
Metoda DeleteStation přijímá parametr stationId pro určení stanice a nepovinný parametr checkDelete, kterým je určeno, jestli má být uživateli zobrazen pohled s posledním upozorněním o smazání stanice nebo jestli mají být data z databáze odstraněna. Pokud uživatel potvrdil v předchozím dotazu odstranění stanice a naměřených dat jsou z databáze odstraněny záznamy z tabulky Measurements a z tabulky Stations pro vybranou stanici. Tyto změny jsou nevratné viz Obrázek 3.12.



Obrázek 3.12 Diagram pro metodu DeleteStation

Api kontrolér

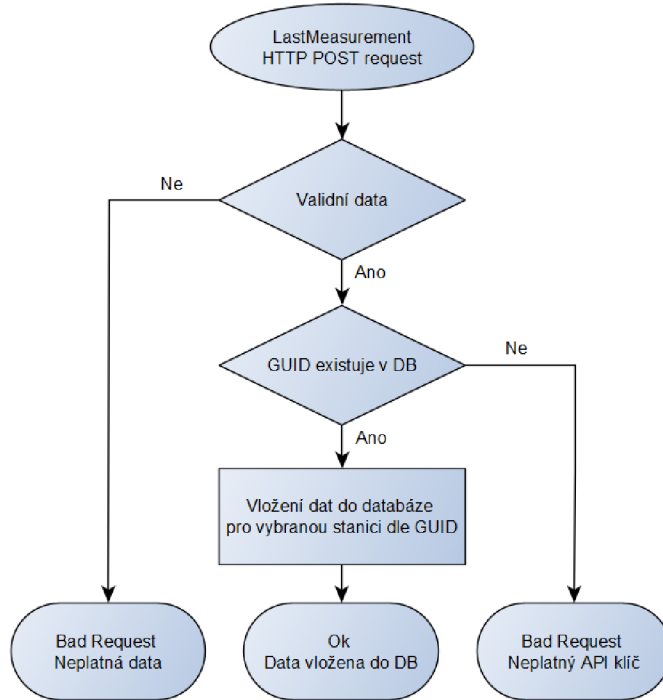
Api kontrolér byl vytvořen za účelem vytvoření REST API[30] pro webovou aplikaci. V případě navržené aplikace se jedná o API, pomocí kterého je možné vkládat naměřená data ze stanice a číst poslední naměřená data pro zvolenou stanici. O funkci API kontrolérů se starají dvě metody pojmenované jako LastMeasurement podle toho, že zpracovávají poslední měření jak při příjmu, tak při odesílání viz Obrázek 3.13.



Obrázek 3.13 Rozdělení tříd v API kontroléru

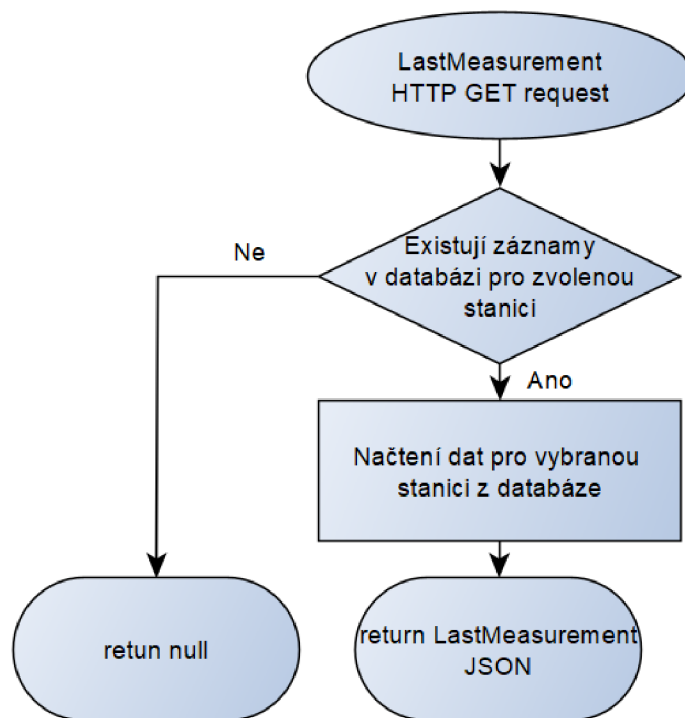
Metoda LastMeasurement pro příjem dat přijímá POST požadavky a jako vstupní parametr třídu StationData, která byla popsána výše a díky definovaným vazbám vlastností jsou data POST požadavku automaticky přiřazena vlastnostem instance třídy. Metoda je vyvolána při POST požadavku a jako první je ověřeno, zdali byla přijatá data validní a jestli vlastnosti instance třídy byly těmito hodnotami naplněny. Pokud ano,

je podle API klíče z požadavku vyhledána stanice v databázi. Pokud je nalezena stanice s odpovídajícím API, jsou pro stanici z databáze vyčteny informace a je vytvořena nová instance třídy Measurement s vlastnostmi, které byly přijaty v POST požadavku. Následně je tento objekt vložen to databázové tabulky Measurements. Uživatel nebo zařízení je o výsledku operace informováno odpovídajícím stavovým HTTP kódem a textovou hláškou. Při úspěšném vložení dat je výsledek s kódem 200 OK, pro neplatný API klíč a špatný formát dat je odpověď 400 BadRequest viz Obrázek 3.14 Diagram pro třídu LastMeasurement HTTP POST.



Obrázek 3.14 Diagram pro třídu LastMeasurement HTTP POST

Pro obsluhu požadavků na odběr dat byla vytvořena metoda s názvem LastMeasurement, ovšem v tomto případě pro zpracování GET požadavku. Přijímá parametr id pro určení stanice a vrací objekt LastMeasurement. Při přijetí požadavku je nejprve ověřena existence stanice pro zvolené id. Při existenci jsou načtena poslední data a informace o dané stanici do instance třídy LastMeasurement a jsou ve formátu JSON vráceny jako odpověď na požadavek. Pokud pro stanici záznamy neexistují, je vrácena pouze hodnota null místo objektu viz Obrázek 3.15.



Obrázek 3.15 Diagram pro třídu LastMeasurement HTTP GET

Pohledy

Vytvořené pohledy pro webovou aplikaci představují výsledné webové uživatelské rozhraní, do kterého jsou vkládána data od kontroléru. V pohledech je definována struktura HTML stránek s propojením se C# kódem. Toto propojení využívá syntaxi Razor engine. Obsah stránky, který se nachází v pohledu, je pouze specifická část z celku webové stránky. Tato část je ještě vkládána do těla, které je společné pro všechny stránky. Tato společná stránka je pojmenována `_Layout`. Takovýmto přístupem jsou zjednodušené úpravy na prvcích stránky, které jsou pro všechny pohledy společné, například záhlaví a zápatí stránky. Pro dynamické vykreslení položek v menu nebo přihlašovacího formuláře v menu je využita takzvaná `ViewComponent`[25] neboli komponenta zobrazení. Ta umožňuje vložit část dynamického kódu do statické části sdílené komponenty `_Layout`. V Tabulka 3.1 je vypsán přehled pohledů a jejich funkce v uživatelském rozhraní.

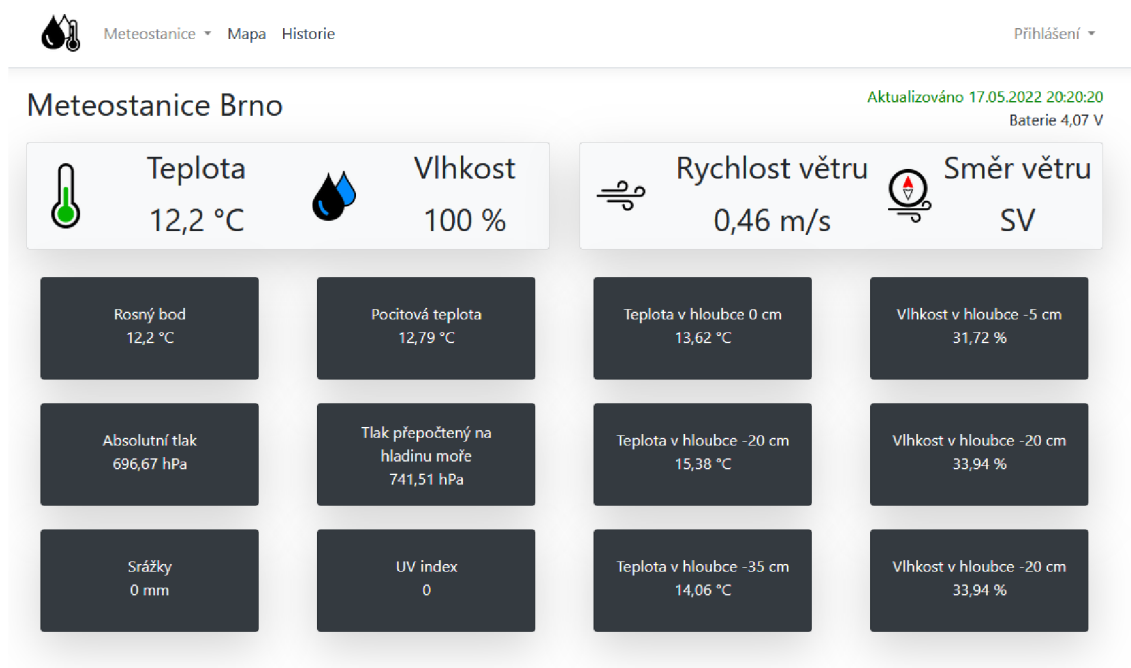
Tabulka 3.1 Pohledy webové aplikace a jejich význam

Jméno pohledu	Funkce pohledu
_Layout	Sdílený pohled, obsahuje záhlaví a zápatí webové stránky, jsou zde vkládány jednotlivé pohledy
Error	Výchozí chybová stránka pro webovou aplikaci
Index	Hlavní stránka webové aplikace se zobrazením posledních naměřených dat
Map	Stránka pro zobrazení mapy stanic
History	Stránka pro zobrazení historie měření
Settings	Stránka pro správu meteostanic
FirstRun	Stránka s formulářem pro vložení uživatelských údajů při prvním spuštění
Login	Stránka s formulářem pro přihlášení do sekce nastavení
DeleteStation	Stránka pro potvrzení při odebírání stanice

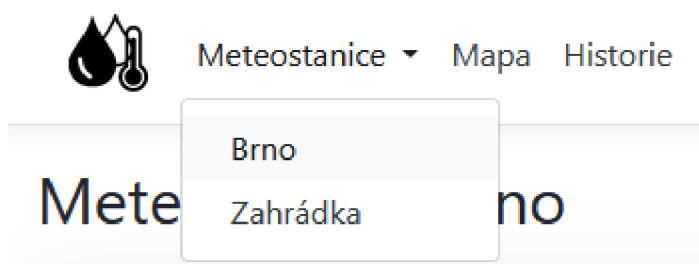
3.2 Uživatelské webové rozhraní

Z uživatelského pohledu je důležité webové rozhraní aplikace a jeho obsluha. Protože se jedná o zobrazování dat z meteostanice, která disponuje velkým množstvím měřených dat, byla hlavní stránka pro zobrazení posledních hodnot z meteostanice rozvržena do bloků pro přehledné zobrazení, podobně jako u některých hardwarových meteostanic viz Obrázek 3.16.

Hlavní stránka je tvořena z pohledu Index. Uživatel má na hlavní stránce přehledně zobrazeny všechny data z meteostanice. Na hlavní stránce je zobrazen název stanice, datum poslední aktualizace, jehož text mění barvu podle stáří dat, a tudíž má uživatel okamžitý přehled o tom, jestli je stanice aktivní. Tento text je pro aktuální data zobrazen zeleně, pro data starší než 30 minut je text oranžový a pokud jsou poslední data starší více než 1 hodinu, je text s datem aktualizace zobrazen červeně. Pod datem aktualizace je zobrazena hodnota napětí baterie. Dále následují karty se zobrazením meteorologických hodnot. Karty s hodnotami byly logicky uspořádány, tak jak jsou data pro člověka vypovídající. Z levé strany do pravé podle toho, co má pro uživatele obvykle největší informativní hodnotu. Nejvíce zjišťována bývá teplota, a proto je barva loga teploty proměnná podle teploty. Pro teploty nižší než 0 °C je logo modré, pro teploty mezi 0-25 °C je zelená a pro teploty nad 25 °C je logo červené. Hlavní stránka umožňuje takto zobrazit data pro všechny přidané stanice. Výběr stanice, pro kterou mají být data zobrazena je možný v menu pod rozbalovací položkou s názvem Meteostanice viz Obrázek 3.17. Tato položka menu je automaticky naplněna názvy meteostanic, které jsou v aplikaci přidány.



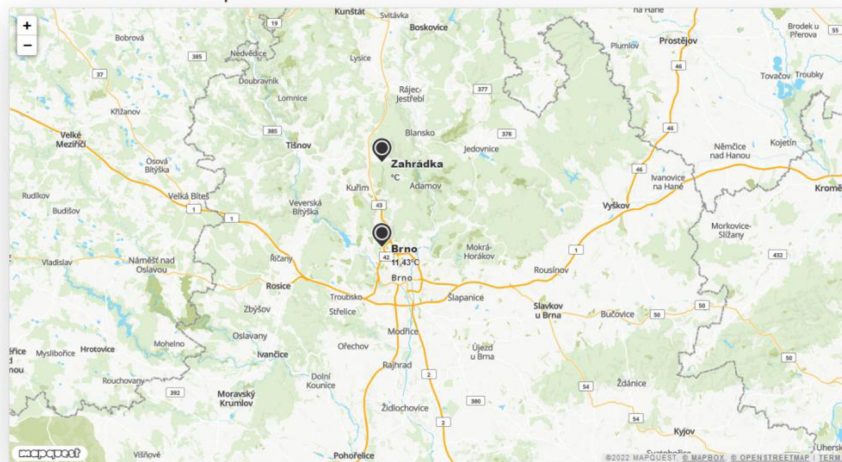
Obrázek 3.16 Hlavní stránka webového rozhraní



Obrázek 3.17 Rozbalovací seznam pro výběr stanic v menu

Další položkou v menu je položka Mapa. Na této stránce jsou zobrazeny meteostanice na mapě. Poloha GPS souřadnic pro umístění bodu se stanicí je vyčtena z databáze. Pro zobrazení interaktivních bodů, které jsou dynamicky přidávány podle stanic v databázi, byl zvolen mapový podklad od společnosti MapQuest[19]. MapQuest nabízí API pro vkládání vlastní bodů i tras do mapových podkladů. Do určitého počtu prokliků je toto API přístupné zdarma. Stanice je pak zobrazena jako bod na mapě s aktuální teplotou a vlhkostí. Mapa se stanicemi viz Obrázek 3.18.

Poloha stanic na mapě



Obrázek 3.18 Zobrazení stanic na mapě

Další stránkou webového rozhraní pro vizualizaci naměřených dat je stránka pojmenována historie. Zde má uživatel možnost procházet všechna v minulosti naměřená data a zobrazovat jejich průběh v grafech. Při prvním vstupu do sekce historie jsou uživateli zobrazeny grafy pro všechna data a jejich kombinace pro výchozí stanici a aktuální den. Uživatel má možnost zvolit si meteostanici, jaké průběhy chce zobrazit, počáteční datum a časové období viz Obrázek 3.19. Pro časové období 1 den jsou zobrazena data ze stanice o půlnoci zvoleného dne do půlnoci dne následujícího. Pro toto období jsou zobrazena měření bez průměrování. Pro další časové období jsou data průměrována, a navíc je přidán rozsah minimálních a maximálních hodnot pro zvolený graf. Pro srážky je potřeba provádět sumu nadhodnotami, aby výsledný graf měl vypovídající hodnotu. Pro každé období se také liší formát data a času[31], který bude zobrazen na ose x a u popisku dat. Data jsou pro zvolené období zpracována a zobrazena podle Tabulka 3.2.

Tabulka 3.2 Formáty času a matematické operace pro zvolené období

Období	Krok pro matematické operace	Operace/Zobrazení	Dlouhý formát data a času (popisek dat)	Krátký formát data času (Popisek osy x)
1 den	-	Zobrazení naměřených dat	dd.MM.yyyyHH:mm	HH:00
1 Týden	1 Hodina	Průměr, Min, Max, pro srážky Suma	dd.MM.yyyy HH:00-HH:00	dd.MM.yyyy HH:00
1 Měsíc	1 Den	Průměr, Min, Max, pro srážky Suma	dd.MM.yyyy	dd.MM.yyyy
1 Rok	1Týden	Průměr, Min, Max, pro srážky Suma	dd.MM.yyyy-dd.MM.yyyy	MMMM yyyy



Historie měření

Výběr dat pro zobrazení

Stanice:

Období: Den
 Týden
 Měsíc
 Rok

Od data:

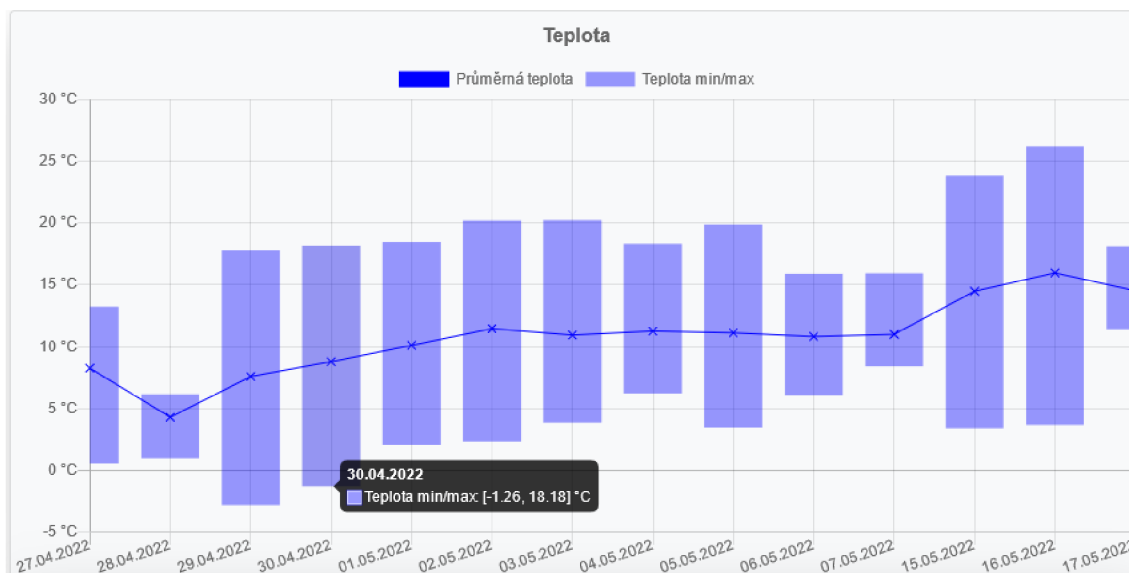
Grafy pro zobrazení

<input checked="" type="checkbox"/> Teplota	<input checked="" type="checkbox"/> Rychlost a směr větru	<input checked="" type="checkbox"/> Napětí baterie
<input checked="" type="checkbox"/> Pocitová teplota	<input checked="" type="checkbox"/> Srážky	<input checked="" type="checkbox"/> Teplota + Pocitová teplota
<input checked="" type="checkbox"/> Rosný bod	<input checked="" type="checkbox"/> UV index	<input checked="" type="checkbox"/> Teplota + Teplota půdy
<input checked="" type="checkbox"/> Vzdušná vlhkost	<input checked="" type="checkbox"/> Teplota půdy	<input checked="" type="checkbox"/> Vlhkost + Vlhkost půdy
<input checked="" type="checkbox"/> Tlak vzduchu	<input checked="" type="checkbox"/> Vlhkost půdy	<input checked="" type="checkbox"/> Teplota + Vlhkost
<input checked="" type="checkbox"/> Tlak vzduchu přepočtený na hladinu moře		

Zobrazit

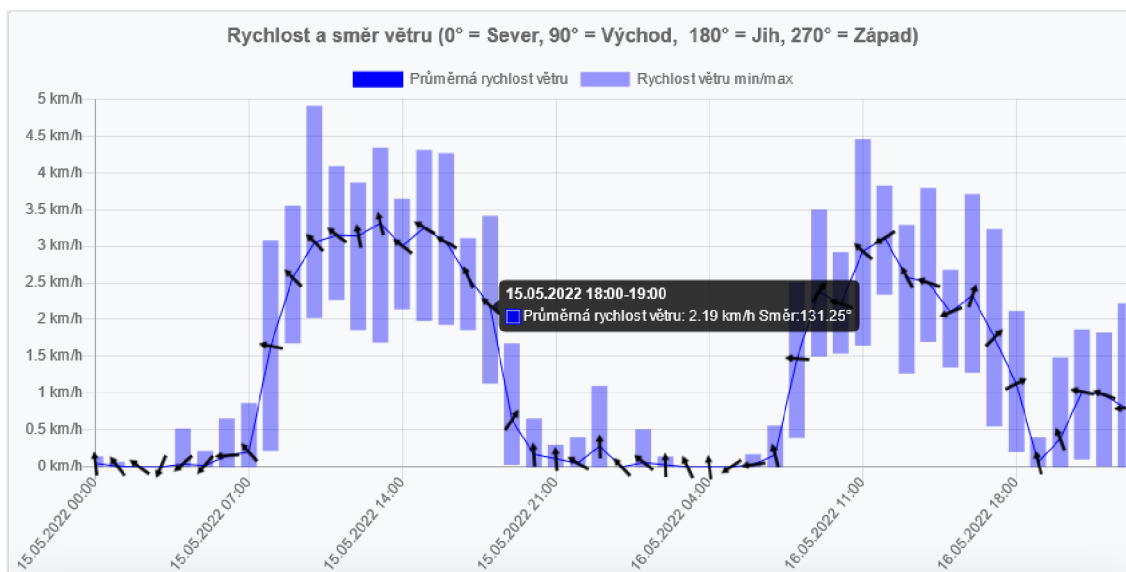
Obrázek 3.19 Volba dat pro zobrazení historie měření

Zvolený časový úsek začíná počátečním datem, které uživatel zvolí. Datum lze volit pouze v rozmezí, pro které existují data v databázi. Pro volbu datumu je ve formuláři použit prvek *input type="date"* [31], který po rozkliknutí umožňuje volbu pomocí kalendáře. Pro zobrazení naměřených a průměrných hodnot jsou u všech veličin použity spojnicové čárové grafy s body. Jako bod byl zvolen symbol ‚x‘. Pro zobrazení minima a maxima jsou použity plovoucí sloupcové grafy viz Obrázek 3.20. Na Obrázek 3.20 je znázorněn jeden z mnoha grafů, které jsou použity pro vykreslení historie měření, kde jsou vykresleny průměrné, minimální a maximální hodnoty za zvolené období.



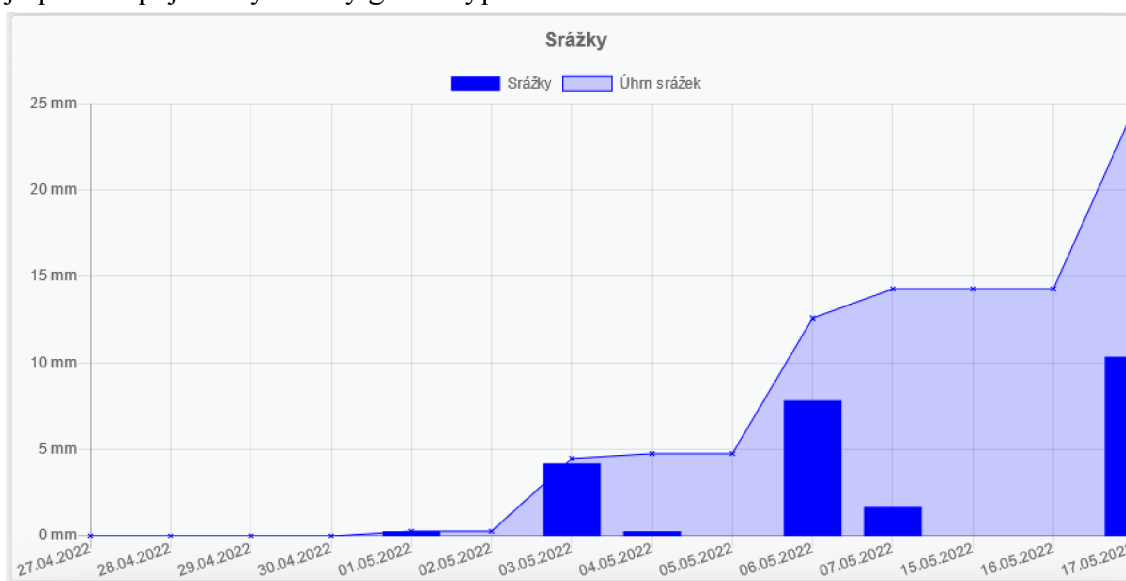
Obrázek 3.20 Graf pro zobrazení teploty

Výjimkou jsou grafy pro zobrazení rychlosti a směru větru a srážek. Kde pro rychlost a směr větru je místo symbolu ‚x‘ použit obrázek šipky, která svým natočením odpovídá úhlu směru větru viz Obrázek 3.21.



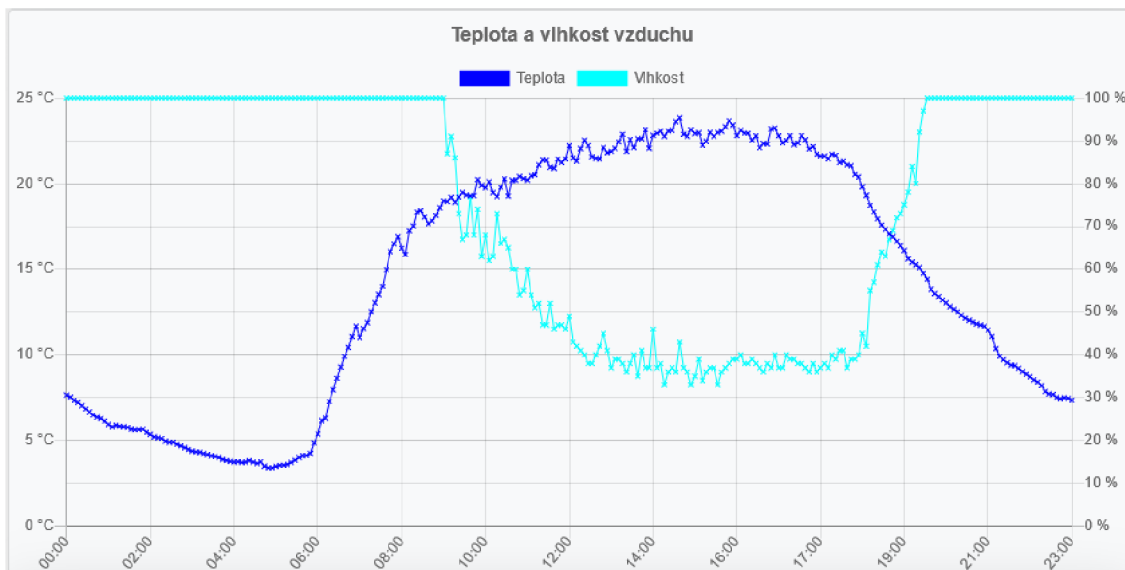
Obrázek 3.21 Graf pro zobrazení rychlosti a směru větru

Pro krátkodobé úhrny srážek je použit sloupcový graf, pro úhrn za celkové období je použit spojnicový čárový graf s výplní viz Obrázek 3.22.



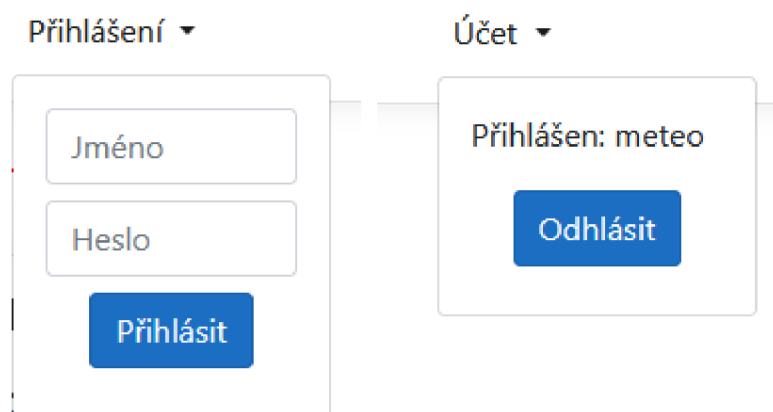
Obrázek 3.22 Graf pro zobrazení úhrnu srážek

Pro vybrané měřené veličiny jsou hodnoty zobrazeny ve společném grafu. Společné grafy zahrnují teplotu a pocitovou teplotu, Teplotu a teplotu půdy, vlhkost a vlhkost půdy a teplotu s vlhkostí. V těchto grafech jsou zobrazeny pro přehlednost pouze průměrné nebo aktuální hodnoty viz Obrázek 3.23.



Obrázek 3.23 Společný graf pro zobrazení teploty a vlhkosti

Další částí uživatelského rozhraní je Nastavení. Jak již bylo popsáno výše, je tato část přístupná pouze po přihlášení, aby nebylo umožněno neoprávněným uživatelům mazat nebo upravovat údaje o stanicích. Přihlášení je součástí menu v podobě rozbalovacího formuláře a je zde vkládáno pomocí komponent zobrazení ve společném pohledu `_Layout`. Rozbalovací formulář může mít dvě podoby, a to před přihlášením kdy obsahuje vstupní pole pro zadání jména a hesla a tlačítko přihlásit a po přihlášení kdy je zobrazeno uživatelské jméno a tlačítko odhlásit viz Obrázek 3.24. Přihlášení může být také zobrazeno na samostatné stránce viz Obrázek 3.25. Přihlášení na samostatné stránce je zobrazeno pouze v případě, že se nepřihlášený uživatel snaží vstoupit do sekce s nastavením přímým zadáním URL adresy nebo po vypršení Session.



Obrázek 3.24 Rozbalovací formuláře pro přihlášení a odhlášení



Přihlášení

Pro vstup do této sekce je nutné přihlášení

Přihlášení

Jméno:

Heslo:

Obrázek 3.25 Přihlášení na samostatné stránce

Po úspěšném přihlášení je uživateli zobrazena sekce s nastavením. V této sekci jsou zobrazeny všechny přidané stanice a informace o nich. Pro každou ze stanic je zde samostatné nastavení v podobě karty. V této kartě jsou zobrazeny základní informace o stavu baterie a status, zda je stanice aktivní a kdy naposledy odeslala data. Jsou zde zobrazeny i URL adresy pro odesílání a příjem dat pro danou stanici. Uživatel zde má možnost měnit název, GPS souřadnice a také lze volit výchozí stanici pro zobrazení na hlavní stránce viz Obrázek 3.26. Pro přidání nové stanice je na konci seznamu karta s touto možností. Zde uživatel zadává název a souřadnice viz Obrázek 3.26. Při vytvoření jsou pro novou stanici automaticky vygenerovány URL adresy.

Nastavení

Meteostanice: Brno

Status: **Aktivní**
Poslední aktualizace: 18.05.2022 0:34:56
Baterie: 4,06 V
Název stanice:
GPS - Zeměpisná šířka:
GPS - Zeměpisná délka:
URL pro sběr dat: http://31.30.12.197:30939
/Api/LastMeasurement?api_key=a139c273-
e27d-4e69-97d0-7fdc6b46e96f
URL pro mobilní aplikaci: http://31.30.12.197:30939
/Api/LastMeasurement?StationId=1
Nastavit jako výchozí stanici:

Vytvoření nové meteostanice

Název stanice:
GPS - Zeměpisná šířka:
GPS - Zeměpisná délka:
Nastavit jako výchozí stanici:

Obrázek 3.26 Nastavení pro aktivní stanici a kartou pro vytvoření nové stanice

Uživatel má možnost odstranění stanice a veškerých dat, která byla naměřena. Tato funkce je užitečná, pokud chce uživatel meteostanici například přemístit a nepřeje si zachovávat naměřená data z jiné lokace. Odstranění je možné provést v nastavení pomocí tlačítka odstranit. Aby nedošlo k nechtěnému odstranění dat, je uživatel po kliknutí na tlačítko odstranit přesměrován na stránku, kde je uživatel naposledy dotázán, zdali si přeje stanici odstranit. Tuto volbu může buď potvrdit, nebo se vrátit zpět. Na této záchytné stránce jsou také zobrazena základní data s názvem a stavem stanice, aby bylo jasné, o kterou stanici se jedná viz Obrázek 3.27. Po tomto kroku jsou data nevratně odstraněna z databáze.

Odstranění stanice

Změny jsou nevratné, potvrďte vymazání stanice a veškerých záznamů.

Odstranit meteostanici: Zahrádka

Status:	Neaktivováno
Poslední aktualizace:	Neaktivováno
Baterie:	Neaktivováno
GPS - Zeměpisná šířka:	49,3268
GPS - Zeměpisná délka:	16,5758
Výchozí stanice:	Ne

Obrázek 3.27 Poslední potvrzení při odstraňování stanice

Webové rozhraní obsahuje také stránku, které je uživateli zobrazena pouze pokud nejsou v databázi v tabulce Logins žádné záznamy. Předpokládá se, že tento stav nastane pouze při prvním spuštění aplikace. Uživatel je na tuto stránku přesměrován při každém dotazu na jakoukoli jinou stránku. Tato stránka obsahuje formulář pro vytvoření uživatelského jména a hesla pro vstup do administrace viz Obrázek 3.28.

První spuštění

Nastavte jméno a heslo pro vstup do administrace

Nové přihlašovací údaje

Jméno:	<input type="text" value="meteo"/>
Heslo:	<input type="password" value="•••••"/>
<input type="button" value="Pokračovat"/>	

Obrázek 3.28 Formulář pro vložení uživatelských údajů při prvním spuštění

3.3 Nastavení a konfigurace aplikace

Pro spuštění a provoz webové aplikace .NET 5, která pracuje s MySQL databází je nutné také správně nakonfigurovat náležitě závislosti a doinstalovat balíčky pro propojení s databází[7]. V neposlední řadě je nutné mít nainstalovaný MySQL server.

3.3.1 Konfigurace databáze

V prvním kroku je nutné nainstalovat MySQL server a nastavit přihlašovací údaje pro správce a pro uživatele[6]. Údaje o připojení k databázi jsou v aplikaci nastaveny v souboru appsettings.json. Nastavení pro databázi, která je spuštěna na stejném hostujícím počítači nebo serveru jako webová aplikace vypadá následovně:

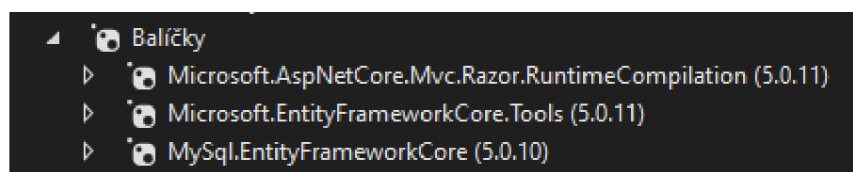
```
"AllowedHosts": "*",
"ConnectionStrings": {
  "DefaultConnection":
"server=127.0.0.1;port=3306;database=MeteoWebApp;user=root;password=pa
ss"
}
```

V této konfiguraci je nastavena adresa, port a uživatelské údaje pro připojení k serveru. Tato konfigurace ve formě textu je využita v souboru Startup.cs kde je v konfiguraci služeb přidáno připojení k databázi. V konfiguraci služeb je také přidáno použití session, které je využíváno pro přihlášení.

```
public void ConfigureServices(IServiceCollection services)
{
    // Configure connection with Db while startup
    services.AddDbContextPool<MeteoDbContext>(options =>options.UseMySQL
(Configuration.GetConnectionString("DefaultConnection")));

    services.AddControllersWithViews();
    // Use session
    services.AddSession();
}
```

Aby bylo možné databázi využívat jsou součástí webové aplikace balíčky pro podporu závislostí viz Obrázek 3.29.



Obrázek 3.29 Balíčky pro podporu Entity Frameworku a MySQL

V konfiguraci připojení k databázi je využívána třída MeteoDbContext, která vychází ze třídy DbContext[29]. Tato třída obsahuje popis tabulek, které mají být v databázi vytvořeny a také jsou zde popsány relace mezi jednotlivými tabulkami. Například závislost tabulky Station a tabulky Measurements, kdy je v tabulce Measurements využitý prvek StationId jako nevlastní klíč.

Třída MeteoDbContext:

```
public class MeteoDbContext : DbContext
{
    public MeteoDbContext(DbContextOptions<MeteoDbContext> options) :
    base(options)
    {}

    public DbSet<Station> Stations { get; set; }
    public DbSet<Measurement> Measurements { get; set; }
    public DbSet<Login> Logins { get; set; }

    /* Create relation for table Stations and Measurements */
    protected override void
    OnModelCreating(ModelBuilder modelBuilder)
    {
        // 1:N relation
        modelBuilder.Entity<Measurement>().HasOne<Station>(M
        =>M.Station).WithMany(S
        =>S.Measurements).HasForeignKey(M
        =>M.StationId);
    }
}
```

Pro vytvoření tabulek v databázi je nutné přidat migraci s popisem a aktualizovat databázi. Migrace vygeneruje konfigurační soubor, který je použit při aktualizaci databáze. Migrace a aktualizace mohou být provedeny například i při úpravě názvu tabulek. Pro vložení tabulek do databáze je nutné provést následující dva kroky v konzoli balíčků.

```
Add-Migration inicializace_databaze
Update-Database
```

3.3.2 Chybové stránky a adresa serveru

Aby bylo možno používat vlastní chybovou stránku viz Obrázek 3.30 pro zobrazení uživateli, je nutno k těmto stránkám zadat vlastní cestu v souboru web.config. Tento soubor je ve formátu XML a jsou zde nastaveny cesty k souborům nebo akcím pro daný chybový kód.

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
<system.webServer>
    <httpErrorserrorMode="Custom" existingResponse="Replace">
<remove statusCode="404" subStatusCode="-1" />
<remove statusCode="500" subStatusCode="-1" />
<error statusCode="404" path="/Home/Error" responseMode="ExecuteURL"
/>
<error statusCode="500" path="/Home/Error" responseMode="ExecuteURL"
/>
    </httpErrors>
</system.webServer>
</configuration>
```



Chyba.

Požadavek nemohl být zpracován.

Požadovaná stránka neexistuje, nebo nastala jiná chyba.

Pro řešení problému se obraťte na správce s ID vašeho požadavku.

ID požadavku: 00-b9674b0866cbe344a644003de82189b7-6f50566108386c4e-00

Obrázek 3.30 Vlastní chybová stránka aplikace

Poslední konfigurací pro spuštění je soubor `launchSettings.json`. V tomto souboru je nastavena URL adresa aplikace, včetně portu, ve výchozím stavu je zde nastavena adresa `localhost`. Tuto konfiguraci je ovšem potřeba měnit pouze ve specifických případech, například pokud je vyžadováno, aby aplikace byla spuštěna na jiném než základním přiděleném portu.

3.4 Použité prvky

Aplikace byla vyvinuta v prostředí Microsoft Visual Studio 2022, které umožňuje využití stylů pro webové rozhraní. Sada těchto stylů se nazývá `bootstrap`. Dále byly využity již hotové prvky pro vizualizaci dat v grafech a zobrazení mapy.

3.4.1 Bootstrap

V rámci předpřipravených stylů ve vývojovém prostředí Visual Studia 2022 je možné využít pro návrh webového rozhraní stylování pomocí sady `bootstrap` [8]. Tato sada umožňuje jednoduše konfigurovat styl webové stránky pomocí tříd u jednotlivých značek. Usnadňuje tvorbu tabulek a podporuje responzivní design webové stránky.

3.4.2 Chart JS

Pro vizualizaci dat v podobě grafů byl zvolen open source Javaskriptový doplněk, který umožňuje tvorbu velké škály grafů s širokou možností nastavení zobrazení dat. Pro webovou aplikaci byla zvolena verze 2, protože má velkou podporu pro některé funkce. Knihovna nebyla do aplikace instalována, ale byl využit online zdroj knihovny [5]. Adresa knihovny je vložena do hlavičky webové stránky ve sdíleném pohledu `_Layout`.

3.4.3 MapQuest

Pro zobrazení stanic na mapě byly využity mapové podklady od společnosti MapQuest[19], Firma se zabývá tvorbou mapových podkladu stejně jako například mapy.cz

nebo Google. Pro vývojáře poskytuje API pro Java Script. Aby bylo možno využívat toto API, je nutno založit si účet a zvolit si plán, podle kterého jsou účtovány poplatky za využívání služeb API. Plán pro 15000 transakcí za měsíc je poskytován zdarma. Pro aplikaci je nutné vygenerovat API klíč, který je vkládán do Java scriptu.

3.5 Testování aplikace

Testování aplikace bylo provedeno v několika krocích. Webové rozhraní bylo ručně otestováno na tzv. extrémní situace, kdy uživatel zadá nesmyslná data do vstupních polí a byl sledován postup v kódu, zda zachytí neočekávané vstupu dle předpokladů případné výjimky byly ošetřeny. Následně byl proveden test, zdali není přístupováno do databáze, pokud neexistují žádné záznamy pro stanice a měření a zdali je uživateli zobrazen správný výpis. Dále byl otestován přístup do zabezpečené sekce, kdy byl pomocí nástroje curl odeslán pouze POST požadavek pro přidání nebo odstranění stanice. Očekávaným výstupem je přesměrování na hlavní stránku a neprovedení POST požadavku, čemuž odpovídá i výstup z nástroje curl viz Obrázek 3.31.

```
C:\Users\Martin>curl -X POST -k -i "http://192.168.88.114:30939/Home/CreateStation?name=d"
--data "gps_latitude=10.4&gps_longitude=23.1"
HTTP/1.1 301 Moved Permanently
Transfer-Encoding: chunked
Location: /Home/Login
Server: Microsoft-IIS/10.0
X-Powered-By: ASP.NET
Date: Wed, 18 May 2022 00:54:40 GMT
```

Obrázek 3.31 Výstup nástroje curl při pokusu o vložení nové stanice bez zadání hesla

Testování API pro sběr dat bylo také provedeno pomocí nástroje curl. Pro případy, kdy jsou data i API klíč ve správném formátu i pro případ nesprávných dat nebo neplatného API klíče bylo testování úspěšné viz Obrázek 3.32.

```

C:\Users\Martin>curl -X POST -k -i "http://192.168.88.114:30939/Api/LastMeasurement?api_key=7796991c-d440-4e43-b5c4-7a449287eec9" --data "temperature=-6.05&dew_point=12.56&temperature_feels_like=21.59&humidity=58&pressure=955.73&sea_level_pressure=1015.35&wind_speed=1.29&wind_angle=67.5&wind_angle_str=ENE&rain_amount=5.10&uv_index=2.7&soil_temp_depth_1=-35 cm&soil_temp_1=18.44&soil_temp_depth_2=-20 cm&soil_temp_2=19.81&soil_temp_depth_3=0 cm&soil_temp_3=21.69&soil_moisture_depth_1=-35 cm&soil_moisture_1=0.00&soil_moisture_depth_2=-20 cm&soil_moisture_2=0.00&soil_moisture_depth_3=-5 cm&soil_moisture_3=0.00&battery_voltage=04.06"
HTTP/1.1 400 Bad Request
Transfer-Encoding: chunked
Content-Type: text/plain; charset=utf-8
Server: Microsoft-IIS/10.0
X-Powered-By: ASP.NET
Date: Wed, 18 May 2022 00:59:54 GMT

Invalid API key

```

Obrázek 3.32 Výstup nástroje curl při pokusu o vložení naměřených dat s neplatným API klíčem

Pro otestování komunikace je možné využít následující vzorový příkaz programu curl.

```

Curl-X POST -k -I "http://192.168.88.114:30939/Api/LastMeasurement?api_key=7796991c-d440-4e43-b5c4-7a449287eec9" --data "temperature=-6.05&dew_point=12.56&temperature_feels_like=21.59&humidity=58&pressure=955.73&sea_level_pressure=1015.35&wind_speed=1.29&wind_angle=67.5&wind_angle_str=ENE&rain_amount=5.10&uv_index=2.7&soil_temp_depth_1=-35 cm&soil_temp_1=18.44&soil_temp_depth_2=-20 cm&soil_temp_2=19.81&soil_temp_depth_3=0 cm&soil_temp_3=21.69&soil_moisture_depth_1=-35 cm&soil_moisture_1=0.00&soil_moisture_depth_2=-20 cm&soil_moisture_2=0.00&soil_moisture_depth_3=-5 cm&soil_moisture_3=0.00&battery_voltage=04.06"

```

API pro odesílání dat bylo otestováno pomocí nástroje curl a také pomocí vytvořené mobilní aplikace. Webová aplikace byla následně spuštěna na testovacím počítači, přijímala data z meteostanice a poskytovala API pro mobilní aplikaci. Při běžném používání z uživatelského pohledu nebyly nalezeny chyby, které by ovlivňovaly běh aplikace.

4 MOBILNÍ APLIKACE

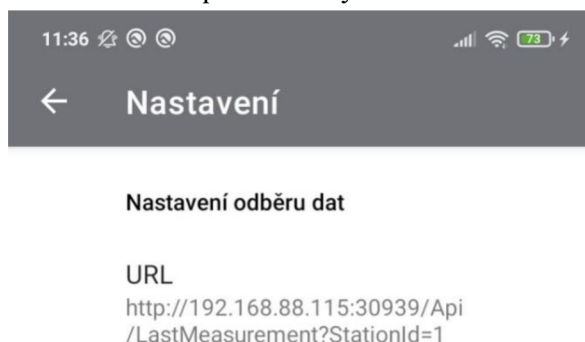
Mobilní aplikace byla vyvinuta pro platformu Android ve vývojovém prostředí Android Studio. Aplikace poskytuje uživateli možnost zobrazovat aktuální data z vybrané stanice. Na hlavní stránce mobilní aplikace jsou zobrazena všechna aktuální data ze stanice. V nastavení má uživatel možnost zadat URL adresu pro odběr dat, kterou získá v nastavení webové aplikace. REST API[30] webové aplikace poskytuje data ve formátu JSON jako odpověď na HTTP GET požadavek z mobilní aplikace. Takto získaná data jsou navázána na připravený datový model a následně jsou uživateli zobrazena. Pro zobrazení základních dat jako je teplota a vlhkost byl také vytvořen widget, který lze umístit na plochu.

4.1 Hlavní aplikace

Hlavní aplikace umožňuje uživateli zobrazení posledních naměřených dat z meteostanice. V nastavení aplikace se pak vkládá uživatel odkaz pro stahování dat.

4.1.1 Design aplikace a layout

Mobilní aplikace poskytuje po otevření jednu hlavní stránku pro zobrazení posledních dat a druhou stránku s nastavením pro vložení URL adresy pro stahování dat. Pro navigaci mezi okny bylo využito navigační menu. Nastavení je přístupné po rozkliknutí rozbalovacího menu, následný návrat zpět je zajištěn navigačním tlačítkem zpět, které automaticky naviguje na okno hlavní stránky, protože je nastavena jako rodičovská pro okno s nastavením viz Obrázek 4.1. Okno s nastavením je navrženo jednoduše podle vzoru nastavení v operačním systému.



Obrázek 4.1 Okno s nastavením v mobilní aplikaci

Hlavní okno aplikace bylo navrženo tak, aby bylo možné zobrazit přehledně všechna měřená data. Z tohoto důvodu bylo zvoleno rozložení dat do tabulky. Protože je zobrazovaných dat mnoho, je pro menší formáty displeje přidáno posouvání

obsahu za okraj, pokud se obsah tabulky neveleze na display. Hlavní okno aplikace se staženými daty viz Obrázek 4.2. Zobrazená tabulka s daty odpovídá vlastnostem v třídě LastMeasurement.

Veličina	Hodnota
Název stanice	Brno
Poslední aktualizace	14.05.2022 18:14:13
Teplota vzduchu	20.03 °C
Teplota rosného bodu	10.66 °C
Pocitová teplota	19.85 °C
Relativní vlhkost vzduchu	55.0 %
Tlak vzduchu na stanici	949.12 hPa
Tlak vzduchu přepočtený na hladinu moře	1008.54 hPa
Rychlost větru	2.1 km/h
Směr větru	SV
Množství srážek za 5 min	0.0 mm
Množství srážek za 24 hodin	0.0 mm
Množství srážek za 7 dní	0.0 mm
UV index	0.0
Teplota půdy v -35 cm	13.31 °C
Teplota půdy v -20 cm	16.62 °C
Teplota půdy v 0 cm	21.25 °C
Vlhkost půdy v -35 cm	35.3 %
Vlhkost půdy v -20 cm	34.92 %
Vlhkost půdy v -5 cm	20.57 %
Napětí baterie	4.08 V

Obrázek 4.2 Hlavní okno mobilní aplikace se staženými daty

Pro pohodlnou aktualizaci dat byl zvolen prvek s názvem SwipeRefresh[23], kdy je aktualizace vyvolána při potáhnutí celého hlavního okna dolů. O tom, že jsou data stahována je uživatel informován kruhovým ukazatelem načítání v horní části okna viz Obrázek 4.3.



Obrázek 4.3 Ukazatel stahování dat

O tom, zdali jsou data stažena nebo ne, je uživatel při dokončení stahování nebo po vypršení časového limitu pro stahování dat, informován plovoucí hláškou viz Obrázek 4.4.

Tlak vzduchu přepočtený na hladinu moře	1008.54 hPa	Tlak vzduchu přepočtený na hladinu moře	--- hPa
Rychlost větru	Aktualizováno 2.1 km/h	Rychlost větru	Data nelze stáhnout --- km/h
Směr větru	SV	Směr větru	---

Obrázek 4.4 Zobrazení informace o výsledku stahování dat

4.1.2 Třídy a životní cyklus aplikace

V aplikaci bylo vytvořeno celkem 6 tříd viz Tabulka 4.1, z čehož 4 z nich jsou spjaty s okny a fragmenty aplikace. Jedná se o třídy MainActivity, MainFragment, SettingsActivity a MeteoWidget.

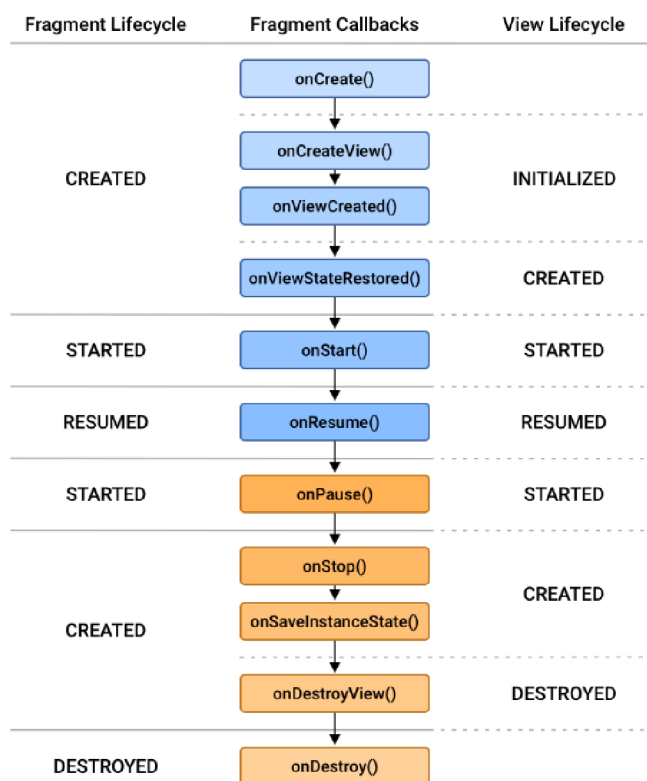
Třída MainActivity je hlavní třídou pro celou mobilní aplikaci, je rozšířením třídy AppCompatActivity a implementuje rozhraní třídy FetchData pro stahování dat. V této třídě jsou přetěžovány metody, které jsou vyvolávány v souvislosti s životním cyklem aplikace viz Obrázek 4.5 a akcemi v aplikaci. Konkrétně jsou to metody onCreate, onResume, processFinish, onCreateOptionsMenu, onOptionsItemSelected, onSupportNavigateUp. Vlastní metoda pro naplnění tabulky staženými daty je pak pojmenována updateView. Z životního cyklu byly vybrány metody onCreate a onResume, protože je možné tyto body cyklu využít pro zlepšení fungování aplikace[17].

Metoda onCreate je vyvolána při startu aplikace, a proto jsou zde nastaveny vlastnosti pro zpětná volání metod, deklarace proměnných a inicializace navigačního menu. Metoda onResume je vyvolávána při vyvolání neboli obnovení hlavního okna aplikace, a proto byla využita pro spuštění stahování dat. Uživatel tedy nemusí data ručně stahovat, ale při každém otevření aplikace jsou automaticky stažena pomocí zpětného volání metody onResume.

Metoda processFinish slouží jako rozhraní pro třídu FetchData a je vyvolána po dokončení stahování dat, které probíhá v pozadí. V této metodě jsou uživatelům zobrazeny hlášky o dokončení stahování a jsou aktualizována data v okně hlavní aplikace.

Tabulka 4.1 Metody třídy MainActivity a jejich vlastnosti

Jméno metody	Funkce metody
onCreate	Deklarace proměnných a nastavení zpětných volání
onResume	Spuštění stahování dat
processFinish	Příjemce stažených dat po dokončení asynchronního stahování
onOptionsItemSelected	Naplnění navigačního menu položkami pro dané fragmenty aplikace
onOptionsItemSelected	Spuštění aktivity podle toho jako uživatel zvolil položku z menu
onSupportNavigateUp	Obsluha navigačního tlačítka zpět v menu, spustí nadřazenou aktivitu
updateView	Aktualizace dat v zobrazení v hlavním okně aplikace



Obrázek 4.5 Životní cykly fragmentu aplikace [16]

Další třídou je MainFragment, kde jsou využity metody pro zpětné volání v závislosti na životním cyklu fragment. Využity jsou onCreateView, onCreateView a onDestroyView. Tyto metody jsou využity pro správné zobrazení obsahu v okně

hlavní aplikace. Třída SettingsActivity využívá metodu zpětného volání onCreate pro nastavení obsahu v okně zobrazení s využitím rozšíření třídy referenceFragmentManager. Je zde nastaven odkaz na soubor s definicí okna pro nastavení, který zároveň obsahuje klíče a hodnoty nastavení.

4.1.3 Třída pro ukládání přijatých dat

Api webové aplikace odesílá data ve formátu JSON, jehož klíče a hodnoty odpovídají struktuře třídy LastMeasurement. Proto byla i pro mobilní aplikaci použita třída LastMeasurement, která svou strukturou a vlastnostmi odpovídá třídě ve webové aplikaci. Jediným rozdílem je, že neobsahuje přetížitelné konstruktory

4.1.4 Stahování dat

Pro stahování dat byla vytvořena třída FetchData, která je rozšířením třídy AsyncTask[21]. Pro stahování dat je přetížena metoda doInBackground[22], která přijímá textový parametr s URL adresou. Pro stahování dat je nastaven časový limit 10 sekund pro připojení a 10 sekund pro čtení dat ze serveru, jelikož se jedná o velmi malé množství dat, není zde pravděpodobnost, že by byl limit v reálném nasazení překročen[2]. Po stažení dat je získán textový řetězec, který je nutné přiřadit podle klíče a hodnoty vlastnostem třídy LastMeasurement. Protože API webové aplikace poskytuje data ve formátu JSON je možné toto přiřazení za využití knihovny GSON [18], která byla vytvořena pro serializaci a deserializaci objektů v JSON formátu. Za pomoci funkce této knihovny jsou přijatá data v JSON formátu navázána na vlastnosti třídy LastMeasurement. Pro správné navázání vlastnosti pro hodnotu data a času je nutné definovat v jakém formátu je datum a čas zapsán.

4.1.5 Nastavení

Nastavení aplikace využívá třídu SharedPreferences, která umožňuje ukládat do paměti zařízení malé množství dat, která jsou tvořena klíčem a hodnotou. Pro účely uložení URL adresy je tato metoda ideální. Díky této třídě je zajištěno, že je URL adresa zachována i po ukončení aplikace. Pro vrácení původní hodnoty URL adresy tak, jak byla po instalaci aplikace je nutné v nastavení telefonu vymazat data aplikace. Pro ukládání URL adresy byl zvolen klíč URL a základní hodnota je vzorová URL adresa „http://your_ip/Api/LastMeasurement?StationId=1“. Hodnota Klíče URL je před každou aktualizací vyčtena a použita pro stahování dat.

4.1.6 Konfigurace aplikace

Pro konfiguraci aplikace slouží soubor AndroidManifest.xml. V tomto souboru jsou nastavena požadovaná povolení pro přístup k internetu[28], nastavení loga aplikace, závislostí mezi jednotlivými okny a nastavení metod pro příjem dat ve widgetu. Pro přístup na internet a využití protokolu HTTP je nutné v konfiguraci nastavit tyto hodnoty:

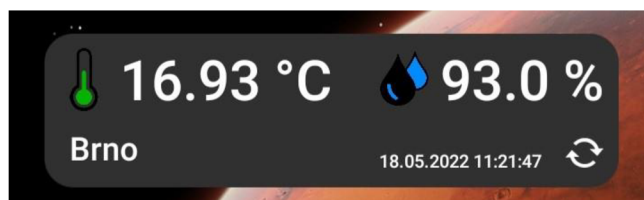
```
<uses-permission android:name="android.permission.INTERNET" />  
android:usesCleartextTraffic="true"
```

4.2 Widget aplikace

Widget aplikace je doplněk pro hlavní aplikaci a poskytuje zobrazení teploty a vlhkosti pro poslední naměřená data. Uživatel má možnost umístit widget kamkoli na domovskou obrazovku.

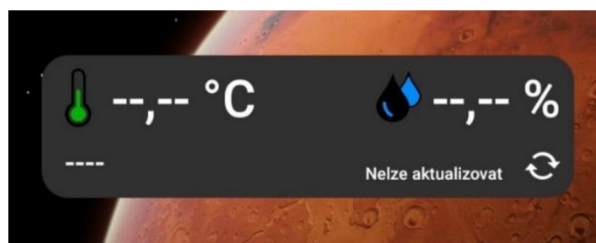
4.2.1 Design

Widget byl navržen tak, aby zobrazované hodnoty bylo přehledné a jasně čitelné. Vedle zobrazení teploty a vlhkosti jsou také loga těchto veličin, tak jako na hlavní stránce webového rozhraní. Logo teploměru také mění barvy podle teploty. Pokud je teplota pod 0 °C, je ikonka modrá, pokud je teplota v rozmezí 0-25 °C, je logo zelené a pro vyšší teploty je logo červené. Takto má uživatel okamžitý vizuální přehled o teplotě. Dále je na widgetu zobrazen název stanice a datum poslední aktualizace. V pravém spodním rohu widgetu je tlačítko pro okamžitou aktualizaci dat. Po kliknutí na hodnotu teploty je spuštěna hlavní aplikace. Minimální velikost widgetu je v závislosti na zobrazovaných datech nastavena na 4 buňky horizontálně a 1 buňku vertikálně. Ve výsledku tedy zabírá jednu řadu na domovské stránce. Pokud jsou data stažena, jsou zobrazena dle předchozího popisu viz Obrázek 4.6.



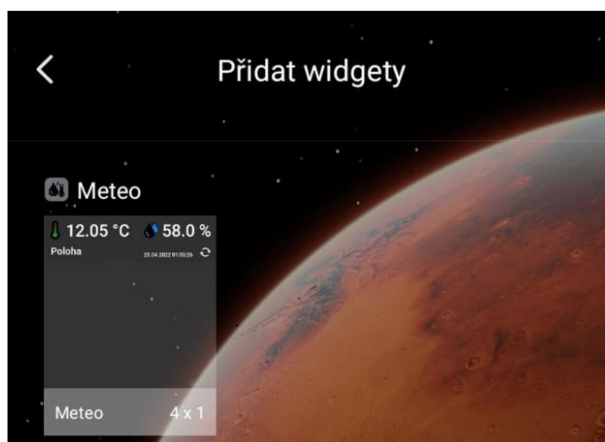
Obrázek 4.6 Aktualizovaný widget

Pokud se data z nějakého důvodu nepodařilo stáhnout, jsou zobrazeny pouze prázdné hodnoty a informace, že nelze data aktualizovat. Viz Obrázek 4.7.



Obrázek 4.7 Widget při neúspěšném stažení dat

Přidání widgetu probíhá tak jako u jiných aplikací. Po dlouhém dotyku na domovské stránce v menu s widgety je zde položka s widgetem pro aplikaci Meteo viz Obrázek 4.8.



Obrázek 4.8 Přidání widgetu na domovskou stránku

4.2.2 Třídy a funkce widgetu

Widget je spjat s třídou `MeteoWidget`. Tato třída je rozšířením pro třídu `AppWidgetProvider`[24] a jsou využity funkce zpětného volání pro aktualizaci dat a zachycení kliknutí na tlačítka ve widgetu. Přetěžovány jsou metody zpětného volání `onUpdate` a `onReceive`. Pro aktualizaci dat byla vytvořena metoda `updateAppWidget`. Metoda `onReceive` je vyvolána, pokud je přijata zpráva od operačního systému. Tato zpráva je odeslána pomocí tlačítka pro aktualizaci na widgetu. Pokud je práva přijata a odpovídá akci kliknutí na tlačítko aktualizovat, je skryto tlačítko pro aktualizaci a je zobrazen kruhový ukazatel načítání dat. Následně je nastavena akce pro odeslání zprávy, která zajistí vyvolání metody `onUpdate`. V metodě `onUpdate` je pro každou instanci widgetu vyvolána metoda `updateAppWidget`, kde jsou stažena data a podle výsledku vložena do widgetu. Také jsou zde nastavena zpětná volání a akce pro tlačítka na widgetu.

4.2.3 Aktualizace a stahování dat

Navržený widget se periodicky aktualizuje po 30 minutách nebo pomocí tlačítka pro okamžitou aktualizaci. Periodická aktualizace je omezena na 30 minut v rámci operačního systému Android. Tato perioda lze nastavit v nastavení widgetu, čas je zde uveden v milisekundách. Aktualizace využívá opět třídu `FetchData` jako hlavní aplikace. Rozdíl je v tom, že widget využívá metodu `doInBackground` pro stažení a získání dat namísto rozhraní.

4.2.4 Konfigurace widgetu

Pro nastavení vlastností widgetu slouží konfigurační soubor `meteo_widget_info.xml`, kde jsou nastaveny všechny vlastnosti včetně doby periodické aktualizace.

```
<?xml version="1.0" encoding="utf-8"?>
<appwidget-provider
xmlns:android="http://schemas.android.com/apk/res/android"
android:description="@string/app_widget_description"
android:initialKeyguardLayout="@layout/meteo_widget"
android:initialLayout="@layout/meteo_widget"
android:minWidth="250dp"
android:minHeight="40dp"
android:previewImage="@drawable/widget_preview"
android:previewLayout="@layout/meteo_widget"
android:resizeMode="horizontal|vertical"
android:targetCellWidth="4"
android:targetCellHeight="1"
android:updatePeriodMillis="1800000"
android:widgetCategory="home_screen" />
```

4.3 Texty a loga v aplikaci

Texty a znaky pro veličiny jsou definovány v samostatném souboru `strings.xml` [3], při použití tohoto souboru je pak následná úprava v rámci projektu velmi zjednodušena. Také mohou být definovány soubory pro vícejazyčnou mutaci aplikace. Barevné schéma a loga použité v aplikaci jsou také definovány v samostatných souborech. Logo mobilní aplikace bylo zvoleno stejné jako logo pro webovou aplikaci.

4.4 Testování aplikace

Testování aplikace probíhalo na telefonu s operačním systémem Android. Z uživatelského pohledu je zde jen jedna možnost zadávání dat, a to v nastavení při vkládání URL adresy. Zde byla otestována situace, kdy uživatel zadá neplatnou URL adresu a pokouší se získat data. V tomto případě vyprší časový limit pro připojení a je uživateli zobrazena hláška s tím, že nelze data stáhnout. Stejná situace platí pro widget. Další situace může nastat, pokud uživatel nemá přístup k internetu, v tomto případě jsou také vypsány hlášky o nemožnosti stažení dat. Při každodenní používání aplikace nebylo limitující ani omezení periodické aktualizace pro aktualizaci widgetu.

ZÁVĚR

V rámci diplomové práce bylo cílem vytvořit webovou a mobilní aplikaci, pro vizualizaci a sběr dat z meteostanice podle předložené dokumentace. Teoretická část práce je věnována meteostanici a popisu měřených veličin. Popsán je formát dat, metoda jejich odesílání na server a zobrazena je také zachycená komunikace a interpretace dat za pomoci programu Wireshark.

První část práce byla věnována návrhu databázové struktury, která byla použita pro webovou aplikaci s využitím Entity Frameworku. Výsledná webová aplikace poskytuje uživateli webové a API rozhraní. Na hlavní stránce webového rozhraní jsou uživateli zobrazena aktuální data z meteostanice. V další sekci jsou meteostanice zobrazeny s aktuálními hodnotami z posledního měření na mapě, dle polohy z nastavení stanice. V sekci historie jsou vizualizovány naměřené hodnoty ve volitelných grafech. V zabezpečené sekci se nachází nastavení pro správu meteostanic. Zde je umožněno uživateli přidávat nové meteostanice nebo upravovat již vytvořené. Po vytvoření nové stanice jsou uživateli této sekci zpřístupněny URL adresy pro odesílání a příjem dat. Pro odesílání a příjem dat poskytuje webová aplikace API. Toto rozhraní je navrženo tak, aby bylo možné přijímat data z meteostanice v současném formátu a zároveň aby bylo možné data poskytovat navržené mobilní aplikaci ve formátu JSON.

Druhá část práce byla věnována vývoji mobilní aplikace pro operační systém Android. Mobilní aplikace byla navržena tak, aby byla uživatelsky intuitivní. Navržená mobilní aplikace využívá API rozhraní webové aplikace pro získávání posledních naměřených dat ve formátu JSON. Tyto data následně zobrazuje v hlavním okně mobilní aplikace. Výběr stanice, pro kterou budou data zobrazena, závisí na URL adrese, kterou uživatel vloží do nastavení aplikace. Pro plné využití mobilní aplikace byl také vytvořen widget, který zobrazuje teplotu a vlhkost z posledního měření. Widget je periodicky aktualizován po 30 minutách a pro případ potřeby je vybaven tlačítkem pro okamžitou aktualizaci.

Aplikace byly otestovány ve dvou fázích. Nejprve při vývoji, kdy byly otestovány neobvyklé situace, které mohou nastat ze strany uživatele. Nakonec byly webová i mobilní aplikace nainstalovány na testovací server a telefon a byly otestovány v provozu s reálnými daty z meteostanice.

LITERATURA

- [1] *Dokumentace k závěrečné práci IOT systém pro zahrádkáře* [online]. [cit. 2021-12-10]. Dostupné z:
https://www.vut.cz/www_base/zav_prace_soubor_verejne.php?file_id=226442
- [2] *Linux: dokumentační projekt*. 4., aktualiz. vyd. Brno: Computer Press, 2007. ISBN 978-80-251-1525-1.
- [3] PILGRIM, Mark. *Ponořme se do Python(u) 3: Dive into Python 3*. 4., aktualiz. vyd. Praha: CZ.NIC, c2010. CZ.NIC. ISBN 978-80-904248-2-1.
- [4] *Tutoriál zaměřený na tvorbu webových aplikací* [online]. [cit. 2021-12-10]. Dostupné z: <https://www.itnetwork.cz/csharp/asp-net-mvc/zaklady/>
- [5] *Online dokumentace pro knihovnu chart.js* [online]. [cit. 2021-12-10]. Dostupné z: <https://www.chartjs.org/docs/latest/>
- [6] *Dokumentace pro databázi mysql* [online]. [cit. 2021-12-10]. Dostupné z: <https://dev.mysql.com/doc/>
- [7] *Tipy a triky pro práci s .net MVC* [online]. [cit. 2021-12-10]. Dostupné z: <https://docs.microsoft.com/en-us/aspnet/mvc/>
- [8] *Online dokumentace k sadě bootstrap* [online]. [cit. 2021-12-10]. Dostupné z: <https://getbootstrap.com/docs/5.1/getting-started/introduction/>
- [9] *Informace o UV indexu* [online]. [cit. 2021-12-10]. Dostupné z: <https://www.e-pocasi.cz/uv-index/>
- [10] *Teorie a vzorec pro výpočet rosného bodu* [online]. [cit. 2021-12-10]. Dostupné z: https://cs.wikipedia.org/wiki/Rosn%C3%BD_bod
- [11] *Teorie o rosném bodu* [online]. [cit. 2021-12-10]. Dostupné z: <https://www.in-pocasi.cz/clanky/teorie/rosny-bod-20.2.2015/>
- [12] *Pocitová teplota* [online]. [cit. 2021-12-10]. Dostupné z: <https://www.in-pocasi.cz/clanky/teorie/pocitova-teplota/>
- [13] *Vzorec pro výpočet pocitové teploty* [online]. [cit. 2021-12-10]. Dostupné z: <https://www.alpy4000.cz/tipy-a-metodika/windchill-pocitova-teplota-detail-323>
- [14] *Článek o tlaku vzduch* [online]. [cit. 2021-12-10]. Dostupné z: <https://www.in-pocasi.cz/clanky/teorie/tlak-vzduchu-4.8.2015/>
- [15] *Teorie k vlhkosti vzduchu* [online]. [cit. 2021-12-10]. Dostupné z: https://cs.wikipedia.org/wiki/Vlhkost_vzduchu
- [16] *Obrázek životního cyklu aplikace* [online]. [cit. 2022-05-18]. Dostupné z: <https://developer.android.com/images/guide/fragments/fragment-view-lifecycle.png>
- [17] *Životní cyklus aplikace* [online]. [cit. 2022-05-18]. Dostupné z: <https://developer.android.com/guide/fragments/lifecycle>
- [18] *Dokumentace ke knihovně GSON* [online]. [cit. 2022-05-19]. Dostupné z: <https://github.com/google/gson/blob/master/UserGuide.md>
- [19] *Dokumentace k API MapQuest* [online]. [cit. 2022-05-19]. Dostupné z: <https://developer.mapquest.com/documentation>

- [20] *Informace o .NET 5* [online]. [cit. 2022-05-19]. Dostupné z: <https://www.c-sharpcorner.com/article/introduction-of-net-5/>
- [21] *Diskuze na téma AsyncTask* [online]. [cit. 2022-05-19]. Dostupné z: <https://stackoverflow.com/questions/24399294/android-async-task-to-make-an-http-get-request>
- [22] *Android AsyncTask* [online]. [cit. 2022-05-19]. Dostupné z: <https://stackoverflow.com/questions/12575068/how-to-get-the-result-of-onPostExecute-to-main-activity-because-async-task-is-a>
- [23] *Android SwipeRefreshLayout* [online]. [cit. 2022-05-19]. Dostupné z: <https://guides.codepath.com/android/implementing-pull-to-refresh-guide>
- [24] *Android Widget* [online]. [cit. 2022-05-19]. Dostupné z: <https://developer.android.com/guide/topics/appwidgets>
- [25] *Component view dokumentace* [online]. [cit. 2022-05-19]. Dostupné z: <https://docs.microsoft.com/en-us/aspnet/core/mvc/views/view-components?view=aspnetcore-3.1>
- [26] *Dokumentace BindProperty* [online]. [cit. 2022-05-19]. Dostupné z: <https://docs.microsoft.com/cs-cz/aspnet/core/mvc/models/model-binding?view=aspnetcore-6.0>
- [27] *Informace o url encode* [online]. [cit. 2022-05-19]. Dostupné z: <https://www.urlencoder.io/learn/>
- [28] *Android AppPermission* [online]. [cit. 2022-05-19]. Dostupné z: <https://developer.android.com/guide/topics/permissions/overview#workflow>
- [29] *Dokumentace pro třídu DbContext* [online]. [cit. 2022-05-19]. Dostupné z: <https://docs.microsoft.com/cs-cz/dotnet/api/system.data.entity.dbcontext?view=entity-framework-6.2.0>
- [30] *Informace o REST API* [online]. [cit. 2022-05-19]. Dostupné z: <https://www.itnetwork.cz/programovani/nezarazene/stoparuv-pruvodce-rest-api/>
- [31] *Dokumentace pro Input type date* [online]. [cit. 2022-05-19]. Dostupné z: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input/date>
- [32] *Dokumentace k formátování data a času* [online]. [cit. 2022-05-19]. Dostupné z: <https://docs.microsoft.com/cs-cz/dotnet/standard/base-types/custom-date-and-time-format-strings>

SEZNAM SYMBOLŮ A ZKRATEK

Zkratky:

URL	Uniform Resource Locator(jednotný lokátor zdroje)
JS	Java Script (programovací jazyk)
UV	Ultra Violet (Ultra fialový)
IoT	Internet of Things (Internet věcí)
HTTP	Hypertext Transfer Protocol (hypertextový přenosový protokol)
API	Application Programming interface (Aplikační programovací rozhraní)
GPS	Global Positioning Systém (Globální Polohový Systém)
GUID	Globally Unique Identifier (Univerzální unikátní identifikátor)
MVC	Model-View-Controller (Model-Pohled-Kontrolér)
.NET	Platforma pro vývoj aplikací
INT	Integer (Celočíselný datový typ)
VARCHAR	Datový typ pro ukládání znakových řetězců
FLOAT	Datový typ
VARBINARY	Datový typ pro ukládání binárních hodnot
MySQL	Databázový systém
JSON	JavaScript Object Notation (Způsob zápisu dat)
GSON	Knihovna pro práci s JSON
XML	Extensible Markup Language (Značkovací jazyk)
POST	Metoda HTTP pro odesílání dat na server
GET	Metoda HTTP protokolu pro vyžádání dat ze serveru

SEZNAM PŘÍLOH

PŘÍLOHA A - ZDROJOVÉ KÓDY WEBOVÉ APLIKACE	62
PŘÍLOHA B - ZDROJOVÉ KÓDY PRO MOBILNÍ APLIKACI	63
PŘÍLOHA C - OBRÁZKY POUŽITÉ V APLIKACÍCH.....	64

Příloha A - Zdrojové kódy webové aplikace

Zdrojový kód webové aplikace je součástí elektronické přílohy.

Příloha B - Zdrojové kódy pro mobilní aplikaci

Zdrojový kód mobilní aplikace je součástí elektronické přílohy.

Příloha C - Obrázky použité v aplikacích

Upravené i originální obrázky včetně licence a zdroje jsou součástí přílohy