

Univerzita Jana Evangelisty Purkyně
v Ústí nad Labem
Přírodovědecká fakulta



Prototyp aplikace pro zobrazení digitálních
dvojčat stavebních projektů v rozšířené realitě

BAKALÁŘSKÁ PRÁCE

Vypracoval: Radek Šmejkal

Vedoucí práce: Ing. Mgr. Pavel Beránek

Studijní program: Aplikovaná informatika

Studijní obor: Aplikovaná informatika

ÚSTÍ NAD LABEM 2025

Namísto žlutých stránek vložte digitálně podepsané zadání kvalifikační práce poskytnuté vedoucím katedry.

Zadání musí zaujímat právě dvě strany.

Zadání je nutno vložit jako PDF pomocí některého nástroje, který umožňuje editaci dokumentů (se zachováním elektronického podpisu).

V Linuxu lze například použít příkaz pdftk.

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně a použil jen pramenů, které cituji a uvádím v přiloženém seznamu literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., ve znění zákona č. 81/2005 Sb., autorský zákon, zejména se skutečností, že Univerzita Jana Evangelisty Purkyně v Ústí nad Labem má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Jana Evangelisty Purkyně v Ústí nad Labem oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladu, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

V Ústí nad Labem dne 9. prosince 2025

Podpis:

Děkuji vedoucímu práce Ing. Mgr. Pavlovi Beránkovi
za neocenitelné rady a pomoc při tvorbě bakalářské práce.

Dále děkuji své rodině a nejbližším přátelům
za podporu během studia a psání závěrečné práce.

PROTOTYP APLIKACE PRO ZOBRAZENÍ DIGITÁLNÍCH DVOJČAT STAVEBNÍCH PROJEKTŮ V ROZŠÍŘENÉ REALITĚ

Abstrakt:

Tato bakalářská práce se zabývá vývojem prototypu mobilní aplikace určené pro zobrazení digitálních dvojčat stavebních projektů v prostředí rozšířené reality. V rámci řešení byl navržen a implementován systém založený na architektuře klient–server, který zahrnuje mobilní aplikaci, serverové REST API a webové rozhraní pro správu digitálních dvojčat a souvisejících dat. Aplikace umožňuje vyhledávání projektů prostřednictvím vlastního mapového systému, interaktivní zobrazení 3D modelů v rozšířené realitě a odesílání zpětné vazby ve formě komentářů a hodnocení. Součástí práce je také návrh a odvození vztahu pro transformaci GPS souřadnic do lokálního souřadnicového systému a implementace cache mechanismu, který optimalizuje načítání mapových dlaždic a zlepšuje výkon aplikace. Výsledkem je funkční prototyp, který demonstruje možnosti využití digitálních dvojčat v oblasti územního plánování a komunikace stavebních záměrů s veřejností.

Klíčová slova: územní plánování, digitální dvojčata, mobilní aplikace, rozšířená realita

PROTOTYPE OF AN APPLICATION FOR DISPLAYING DIGITAL TWINS OF CONSTRUCTION PROJECTS IN AUGMENTED REALITY

Abstract: This bachelor's thesis focuses on the development of a prototype mobile application designed for displaying digital twins of construction projects in an augmented reality environment. As part of the solution, a client–server system was designed and implemented, including a mobile application, a server-side REST API, and a web interface for managing digital twins and related data. The application enables project search through a custom map system, interactive 3D model visualization in augmented reality, and feedback submission in the form of comments and ratings. The work also includes the design and derivation of a relationship for transforming GPS coordinates into a local coordinate system, as well as the implementation of a cache mechanism to optimize map tile loading and improve application performance. The result is a prototype that demonstrates the potential use of digital twins in urban planning and in communicating construction projects to the public.

Keywords: urban planning, digital twins, mobile application, augmented reality

Obsah

Seznam obrázků	13
Seznam ukázek kódu	15
1. Úvod	17
2. Teoretická část	19
2.1. Digitální dvojčata a územní plánování	19
2.2. Rozšířená realita	20
2.3. Přehled existujících řešení	24
3. Praktická část	27
3.1. Metodika práce	27
3.2. Návrh systému	28
3.3. Vybraný technologický zásobník	40
3.4. Implementace systému	41
4. Diskuse a výsledky	47
4.1. Ověření funkčnosti systému	47
4.2. Hodnocení splnění cílů	52
5. Závěr	53
Seznam použitých zdrojů	56
A. Externí přílohy	57

Seznam obrázků

2.1.	Spektrum imerzivních technologií [4]	20
2.2.	Ukázka markeru v praxi. [5]	21
2.3.	Ukázka aplikace bez markerů <i>IKEA Place</i> [6]	22
2.4.	Ukázka interaktivního pískoviště IQLANDIA [7]	23
2.5.	Překlad pomocí aplikace <i>Google Lens</i> [8]	23
2.6.	Ukázka pokládání modelu v aplikaci Augment. [9]	24
2.7.	Ukázka zobrazení stavebního projektu v aplikaci Trimble Connect AR. [10]	25
2.8.	Ukázka zobrazení stavebního projektu pomocí modulu Dalux TwinBIM [12]	26
3.1.	Diagram užití systému	28
3.2.	Diagram aktivit	29
3.3.	Diagram komponent systému	30
3.4.	Diagram balíčků serverové části	31
3.5.	Entitně relační diagram	33
3.6.	Ukázka skici návrhu uživatelského rozhraní	35
3.7.	Prototyp návrhu mobilní aplikace	37
3.8.	Sekvenční diagram systému	39
3.9.	Ukázka klíčových tříd pro fungování AR	40
4.1.	Webové rozhraní s přehledem projektů	47
4.2.	Formulář pro přidání nového projektu	48
4.3.	Potvrzovací okno pro smazání	48
4.4.	Formulář pro úpravu existujícího projektu	49
4.5.	Zobrazení hodnocení a komentářů projektu	49
4.6.	Ukázka jednotlivých obrazovek v mobilní aplikaci	51

Seznam ukázek kódu

3.1. Základní struktura endpointu	41
---	----

1. Úvod

Zapojení veřejnosti do územního plánování je důležitou součástí rozvoje měst. Veřejnost má mít možnost znát plánované změny ve svém okolí a vyjádřit k nim své stanovisko. V praxi však často nemají dostatečnou představu o skutečném dopadu plánovaných stavebních projektů, protože tradiční způsoby prezentace formou technických výkresů, půdorysů nebo statických vizualizací nejsou pro laickou veřejnost dostatečně srozumitelné a neumožňují pochopit prostorové vztahy mezi plánovanou stavbou a okolním prostředím.

Technologie rozšířené reality v kombinaci s konceptem digitálních dvojčat nabízí možnost výrazně zlepšit komunikaci mezi projektanty a veřejností. Rozšířená realita umožňuje zobrazit virtuální 3D model stavby přímo v reálném prostředí pomocí mobilního zařízení, což poskytuje intuitivní a snadno pochopitelnou představu o budoucím stavu. Model digitálního dvojčete přitom může obsahovat nejen vizuální podobu stavby, ale i doprovodné informace o projektu, jeho stavu a plánovaném průběhu realizace.

Cílem této bakalářské práce je vyvinout funkční prototyp mobilní aplikace pro zobrazení digitálních dvojčat stavebních projektů v rozšířené realitě. Aplikace má umožnit vizualizaci 3D modelů staveb, poskytovat informace o projektech a umožnit uživatelům zasílat zpětnou vazbu formou hodnocení a komentářů. Součástí řešení jsou také serverová část a webové rozhraní pro správu digitálních dvojčat.

Práce je rozdělena do pěti hlavních kapitol. Teoretická část se zaměřuje na koncept digitálních dvojčat a jejich využití v územním plánování, technologii rozšířené reality a přehled existujících řešení v této oblasti. Praktická část popisuje metodiku vývoje, návrh architektury systému, volbu technologického zásobníku a implementaci systému. Kapitola diskuse vyhodnocuje výsledky implementace. Závěr shrnuje dosažené výsledky a nastiňuje možnosti budoucího rozvoje.

2. Teoretická část

2.1. Digitální dvojčata a územní plánování

V dnešní době, kdy města čelí významným výzvám jako je rychlý populační růst, omezené zdroje, klimatické změny a tlak na udržitelný rozvoj, je nezbytné využívat pokročilé nástroje pro územní plánování. Tradiční přístupy často nedostačují k řešení složitých a dynamických problémů moderních měst. v tomto kontextu se stále více prosazují technologie jako Internet věcí (IoT), umělá inteligence (AI) a imerzivní technologie jako například virtuální a rozšířená realita. [1]

Digitální dvojče

Digitální dvojče představuje procesy a metody popisující a modelující vlastnosti, chování, vznik a fungování fyzických objektů prostřednictvím digitálních technologií. Model digitálního dvojče plně odpovídá svému fyzickému protějšku v reálném světě a je schopen v reálném čase simulovat jeho chování a výkon. Digitální dvojče tak tvoří dynamický a přesný digitální obraz fyzického objektu, který je průběžně aktualizován na základě dat z reálného prostředí. Tento model může umožňovat nejen monitorování, ale i predikci jeho chování a optimalizaci provozu. Digitální dvojče se tak nemusí omezovat pouze na popis geometrie a funkcí, ale může zahrnovat i celý životní cyklus fyzického objektu, včetně výstavby, údržby a provozu.[2]

Využití digitálních dvojčat v územním plánování

Digitální dvojčata umožňují simulaci různých urbanistických scénářů a předpovídání dopadů plánovaných rozhodnutí, což vede k optimalizaci využití zdrojů, zvýšení udržitelnosti a zlepšení kvality života obyvatel. Tato technologie podporuje datově podložené rozhodování a umožňuje testování inovativních řešení v bezpečném virtuálním prostředí, čímž minimalizuje náklady a rizika reálných experimentů. Díky tomu se digitální dvojčata stávají klíčovým nástrojem moderního územního plánování a správy měst.[1]

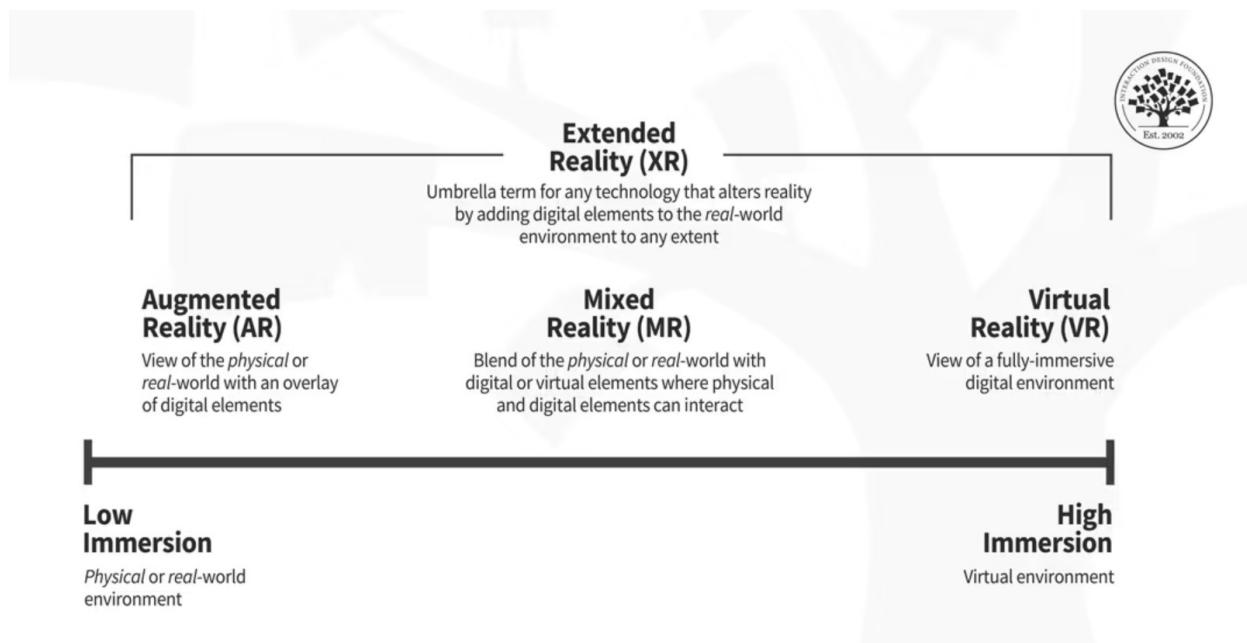
2.2. Rozšířená realita

Rozšířená realita (*Augmented Reality*) je technologie, která v reálném čase spojuje fyzický svět s digitálními prvky, a to ve formě 2D i 3D objektů, které jsou přirozeně začleněné do reálného prostředí. Umožňuje uživatelům interagovat s těmito digitálními objekty pomocí několika metod ovládání, jako jsou například pohyb zařízení, dotykové ovládání nebo hlasové příkazy. [3]

V českém jazyce se běžně používá termín „rozšířená realita“ jako překlad anglického *Augmented Reality* (AR). Tento překlad však není zcela přesný. Pojem *augmented* doslova znamená „posílený“ či „rozšířený o něco navíc“, a vhodnějším překladem by tak mohl být například „posílená realita“. Nicméně, v češtině se již ustálil výraz „rozšířená realita“, a tak jej budu užívat v rámci této práce. [4]

Zároveň je třeba rozlišovat pojem *Extended Reality* (XR), který označuje zastřešující termín pro technologie jako virtuální realita (*Virtual Reality*, VR), smíšená realita (*Mixed Reality*, MR) a právě rozšířená realita (*Augmented Reality*, AR). XR tedy pokrývá celé spektrum imerzivních technologií od posílené a smíšené reality po čisté virtuální. [4]

Spektrum na následujícím obrázku přehledně ukazuje přechod mezi jednotlivými technologiemi.



Obrázek 2.1.: Spektrum imerzivních technologií [4]

Klíčové aspekty

Pro správné fungování rozšířené reality jsou zásadní tyto klíčové aspekty: Jedním z nich je snímání reality, při kterém kamery a senzory zachycují reálné prostředí a detekují klíčové body a plochy v prostoru. Tento proces umožňuje správné zobrazení a interakci digitálních objektů s fyzickým světem, čímž se vytváří dojem jejich skutečné přítomnosti v daném prostředí. Dalším důležitým prvkem je pozicování, které zajišťuje přesné určení polohy zařízení v prostoru. Rozšířená realita využívá různé senzory, jako jsou GPS, akcelerometry či gyroskopy, díky nimž dokáže určit orientaci a pohyb zařízení. Tato kombinace senzorických údajů je klíčová pro správné umístění a zobrazení digitálních prvků v reálném světě. Poslední zásadní složkou je vykreslování, které umožňuje realistické zobrazení digitálních objektů v souladu s perspektivou a měřítkem okolního prostředí. Díky tomu se digitální objekty zobrazují v přesné poloze a působí dojmem, že jsou přirozenou součástí reality. [3]

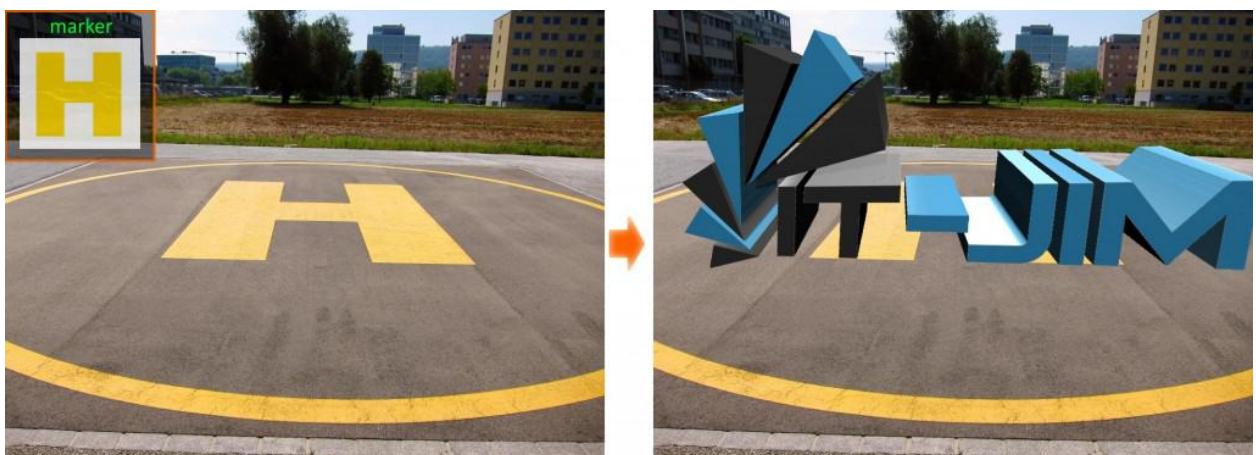
Typy rozšířené reality

Rozšířená realita pomocí markerů (marker-based)

Tento přístup je také často nazýván rozpoznáváním obrázků, protože využívá markery umístěné v reálném světě jako referenční body pro určení polohy a orientace zařízení. Kamery a senzory v zařízení detekují marker, který slouží k identifikaci a zobrazení specifického digitálního objektu přímo na pozici markeru. Tato metoda je jednoduchá a efektivní pro aplikace, které vyžadují interakci s konkrétními objekty v reálném světě. [3]

Markerem může být prakticky jakýkoliv grafický prvek, který je jednoznačně rozpoznatelný a odlišný od okolí. Například speciální symbol, logo, QR kód nebo běžný obrázek s dostatečně výraznými rysy. [5]

Jako příklad slouží obrázek 2.2, kde markerem je H na přistávací ploše a proto se logo položí přesně na pozici markeru.



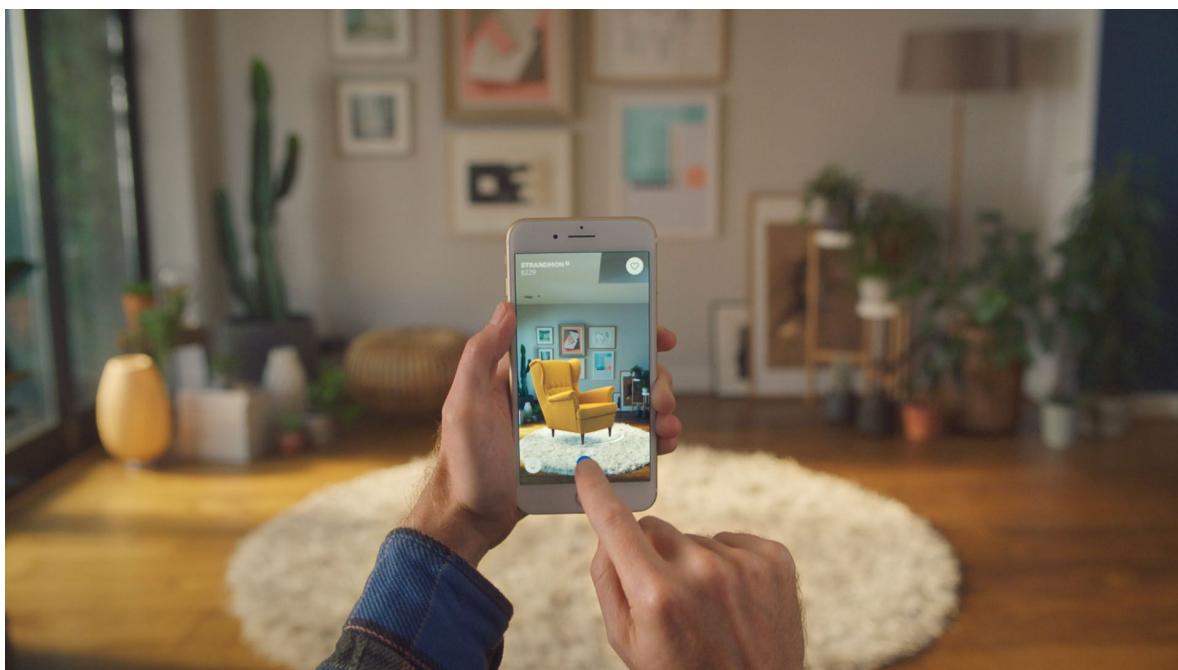
Obrázek 2.2.: Ukázka markeru v praxi. [5]

Rozšířená realita bez markerů (markerless)

Aplikace bez markerů využívá senzory, jako jsou GPS, akcelerometry a gyroskopy pro určení polohy a orientace zařízení v prostoru bez použití vizuálních markerů. Tento přístup je běžně používán v aplikacích, které závisí na geolokaci (například navigace a mapování) a umožňuje uživatelům interagovat s virtuálními objekty na základě jejich skutečné polohy v reálném světě. [3]

Příkladem již existující markerless aplikace je *IKEA Place*, která umožňuje uživatelům vizualizovat nábytek přímo v jejich domácnosti pomocí rozšířené reality.

Na obrázku 2.3 lze vidět ukázku, kde si uživatel vybral produkt na e-shopu a následně ho vizualizoval v samotné aplikaci. [6]



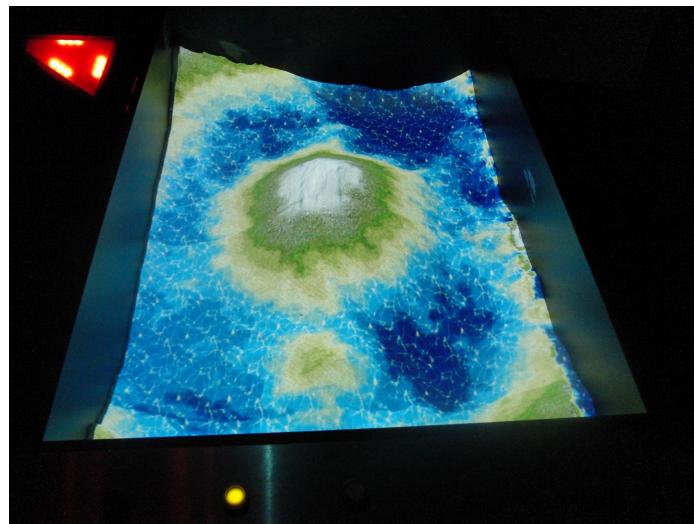
Obrázek 2.3.: Ukázka aplikace bez markerů *IKEA Place* [6]

Rozšířená realita na základě projekce

Tento přístup využívá projekci světla na povrchy v reálném světě k interakci s digitálními prvky. Senzory detekují změny v projekci a umožňují uživatelům reagovat na změny, což vytváří interaktivní zážitky. Tento typ pozicování je mnohem náročnější na výpočetní výkon, protože vyžaduje analýzu změn v projekcích a jejich korelace s polohou uživatele. [3]

Reálným příkladem rozšířené reality na základě projekce je interaktivní pískoviště, které se nachází ve vědeckozábavním centru v Liberci. Toto pískoviště poskytuje uživatelům možnost modelovat vlastní krajinu z píska a pozorovat, jak v ní probíhají různé přírodní procesy.

Na obrázku 2.4 lze vidět pobřeží, které představuje vyvýšené části pískoviště a vytvořené nížiny jsou zatopeny oceánem [7]



Obrázek 2.4.: Ukázka interaktivního pískoviště IQLANDIA [7]

Rozšířená realita pomocí překrytí

Tento přístup umožňuje úplné nebo částečné překrytí reálného objektu digitálními informacemi. U tohoto typu je klíčové rozpoznání objektů, protože aplikace musí správně identifikovat, kde se objekt nachází, aby ho mohla správně překrýt. [3]

Příkladem překrytí je *Google Lens*, již existující aplikace která poskytuje nástroje pro počítání příkladů, překlad textů a identifikaci předmětů prostřednictvím rozšířené reality. Pro lepší představu je obraz z kamery zpracován a následně dochází k překrytí reálného textu v cizím jazyce jeho překladem. [8]



Obrázek 2.5.: Překlad pomocí aplikace *Google Lens* [8]

2.3. Přehled existujících řešení

S rychlým rozvojem technologií rozšířené reality (AR) roste i počet aplikací, které umožňují zobrazovat digitální modely staveb přímo v reálném prostředí uživatele. Existují však nejen aplikace určené pro prezentaci hotových či plánovaných stavebních projektů, ale také sofistikovaná řešení, která stavitelé používají přímo na staveništi. Tyto aplikace slouží například k vizualizaci rozvodů kanalizace, elektroinstalací a vody. To přispívá k efektivnější koordinaci a kontrole stavebních prací.

Augment

Dalším zajímavým řešením je aplikace Augment, která se zaměřuje na jednoduchou vizualizaci 3D modelů podobně jako prototyp vyvýjený v rámci této práce. Na rozdíl od sofistikovaných nástrojů umožňuje snadno a rychle zobrazit modely bez složitého nastavování. Aplikace pouze snímá povrch podlahy a na něj položí uživatelem vybraný 3D model. [9]

Uživatelé mohou libovolně upravovat velikost, přesouvat a otáčet objekty. Tento přístup však snižuje přesnost vztahu mezi digitálním dvojčetem a fyzickým prostředím, a proto není ideální pro přesnou vizualizaci stavebních projektů. Spíše se hodí na zobrazení produktů pro lepší představu například při nakupování na e-shopu. [9]

Na obrázku 2.6 je zobrazený model obchodního stojanu v rozšířené realitě. Uživatel může interagovat s modelem, otáčet ho a měnit jeho velikost přímo na obrazovce.



Obrázek 2.6.: Ukázka pokládání modelu v aplikaci Augment. [9]

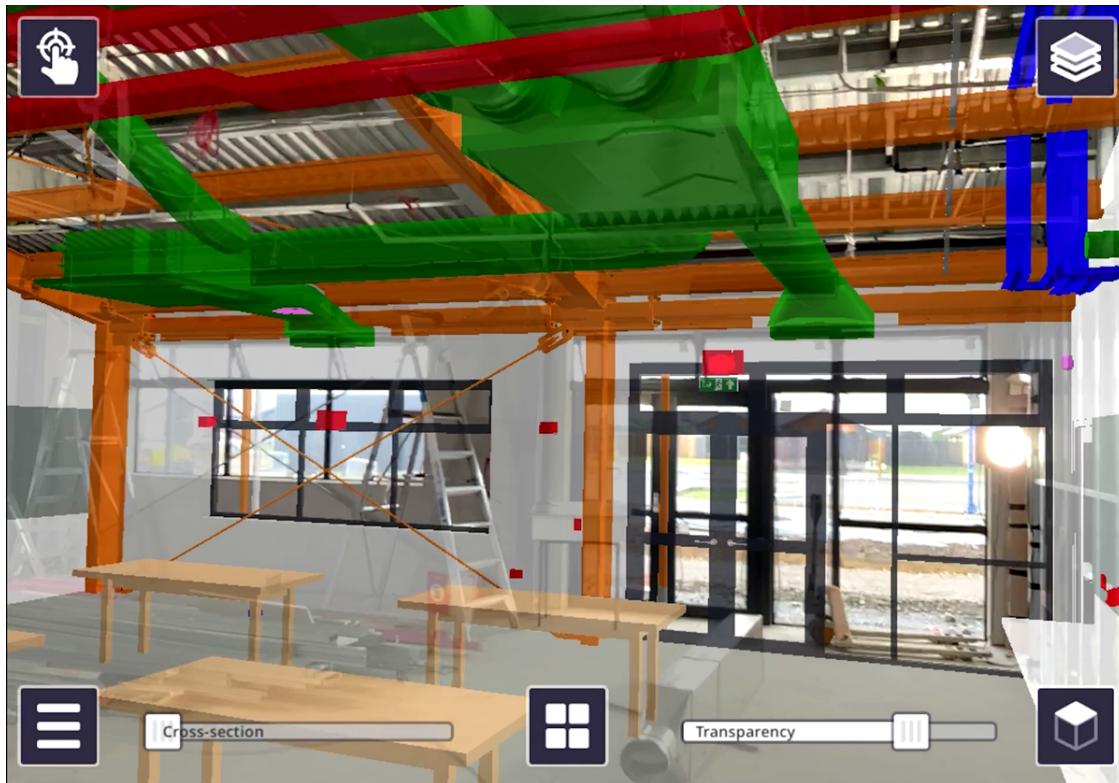
Trimble Connect AR

Trimble Connect AR je pokročilá aplikace využívající rozšířenou realitu pro vizualizaci stavebních projektů přímo na staveništi. Tato platforma umožňuje uživatelům zobrazovat 3D modely staveb v reálném prostředí přes mobilní zařízení nebo AR brýle. Na rozdíl od jednoduchých AR vizualizací se Trimble Connect AR zaměřuje na podporu profesionálů ve stavebnictví. [10]

Mezi hlavní funkce aplikace patří možnost přesného umístění digitálního modelu na skutečné místo výstavby, synchronizace s cloudovým repozitářem projektové dokumentace a spolupráce více uživatelů v reálném čase. Díky integraci s BIM (Building Information Modeling) umožňuje také zobrazovat detailní informace o konstrukčních prvcích, stavebních materiálech a dalších parametrech přímo v AR prostředí. [10]

Tato aplikace je ukázkou moderního využití rozšířené reality v praxi, která přesahuje pouhou vizualizaci a přináší přidanou hodnotu při řízení stavebních projektů. I když je funkčně mnohem pokročilejší než prototyp vyvýjený v rámci této práce, slouží jako inspirace a důkaz potenciálu AR v oblasti digitálních dvojčat staveb. [10]

Obrázek 2.7 znázorňuje interiér rozestavěné budovy propojený s 3D modelem technických instalací. Barevně odlišené vrstvy zobrazují jednotlivé systémy technického zařízení budovy, například zelené prvky představují vzduchotechnické vedení, oranžové části označují nosnou konstrukci a podobně.



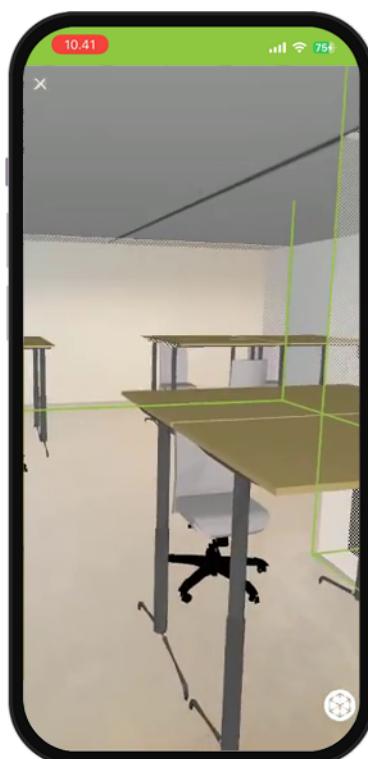
Obrázek 2.7.: Ukázka zobrazení stavebního projektu v aplikaci Trimble Connect AR. [10]

Dalux

Dalux nabízí několik nástrojů pro práci s BIM (Building Information Model) modely, z nichž nejrozšířenějším je Dalux BIM viewer. Tento nástroj umožňuje přehledné zobrazování stavebních projektů ve 2D a 3D formě, prohlížení výkresů, provádění měření, vytváření řezů, filtrování prvků a zobrazování jejich vlastností. [11]

Pro práci v rozšířené realitě nabízí Dalux funkci TwinBIM, která je součástí mobilní aplikace Dalux. TwinBIM umožňuje umisťovat stavební modely do skutečného prostředí pomocí rozšířené reality a zároveň podporuje interakce s objekty, například zobrazení popisu po kliknutí na objekt.[12]

Na obrázku 2.8 se nachází ukázka vizualizace v Dalux TwinBIM, na které lze vidět rozmístění nábytku konstrukce budovy, reálná konstrukce je přitom zvýrazněna zelenou barvou.



Obrázek 2.8.: Ukázka zobrazení stavebního projektu pomocí modulu Dalux TwinBIM [12]

Shrnutí

Analýza existujících řešení ukázala, že technologie pro vizualizaci stavebních projektů v rozšířené realitě jsou již dostupné, avšak většina z nich je zaměřena na profesionální použití, je technicky složitá, nebo finančně náročná. Současná komerční řešení se často orientují na komplexní projektové řízení, simulace stavebního procesu nebo interní firemní komunikaci.

Z vypracovaného přehledu tedy vyplývá, že na trhu chybí jednoduchá, dostupná a uživatelsky přívětivá aplikace, která by umožnila široké veřejnosti nahlížet na plánované projekty formou digitálních dvojčat v rozšířené realitě a zároveň poskytovat zpětnou vazbu. Tato skutečnost potvrzuje relevanci a praktický význam této práce.

3. Praktická část

3.1. Metodika práce

Práce na vývoji prototypu aplikace byla rozdělena do několika etap, které na sebe navazovaly. Cílem bylo postupovat systematicky, od úvodní analýzy a rešerše až po návrh, implementaci a testování aplikace. Při plánování metodického postupu byl zvolen vodopádový model s prvky iterativního přístupu. Vodopádový model je pro akademické práce vhodný kvůli pevnému zadání, které je potřeba nadefinovat před začátkem práce a prvky iterativního modelu jsou zase vhodné při jednotlivých fázích vývoje.

V první fázi byla provedena analýza zadání a dostupných technologií. Tato etapa zahrnovala rešerší týkající se rozšířené reality, digitálních dvojčat a mobilních aplikací, s důrazem na využití v oblasti stavebnictví a územního plánování. Součástí této analýzy bylo také prostudování existujících řešení a výběr vhodného přístupu k návrhu vlastní aplikace.

Na základě zjištěných poznatků následovala fáze specifikace požadavků, ve které byly stanoveny hlavní funkce, které má aplikace splňovat. Důraz byl kladen na využití rozšířené reality pro zobrazení 3D modelů stavebních projektů, přístup k těmto modelům prostřednictvím mapy nebo QR kódu a možnost zpětné vazby ze strany uživatele.

Následovala fáze návrhu systému, kde byl na základě definovaných požadavků vytvořen návrh architektury celé aplikace. Tento návrh zahrnoval návrh klientské i serverové části.

Po dokončení návrhu byla zahájena fáze implementace systému. Vývoj probíhal postupně, přičemž průběžné konzultace s vedoucím práce a zpětná vazba hrály klíčovou roli při ladění funkcionality. Součástí procesu implementace byl také jednoduchý CI proces, který při každém zapsání změn do repozitáře automaticky spouštěl linter, instaloval závislosti a kontroloval, zda je projekt sestavitelný.

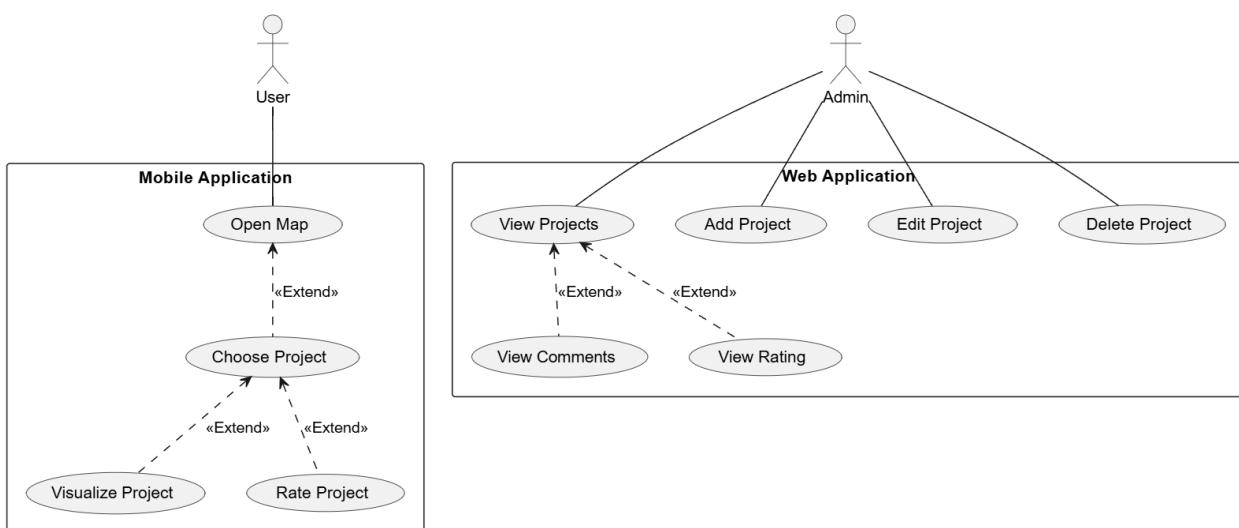
Závěrečnou část metodiky tvořilo manuální testování, během kterého byly ověřeny všechny klíčové funkcionality aplikace, jako jsou rozšířená realita, načítání 3D modelů, komunikace se serverem a mapová navigace.

3.2. Návrh systému

Specifikace požadavků

Před realizací samotného návrhu systému bylo nutné specifikovat požadavky, které aplikace musí splňovat. K tomuto účelu velmi dobře slouží diagram případů užití, který ilustruje interakce mezi uživatelem a samotným systémem.

Na obrázku 3.1 je diagram případů užití systému. Administrátor má možnost provádět různé akce, jako jsou prohlížení, přidávání, úprava nebo mazání projektů. Samotné zobrazení projektů je ještě rozšířeno o zobrazení komentářů a hodnocení jednotlivých projektů. V mobilní aplikaci může uživatel otevřít mapu, to je rozšířeno o možnost výběru projektu a následně ještě o možnost vizualizace a ohodnocení.

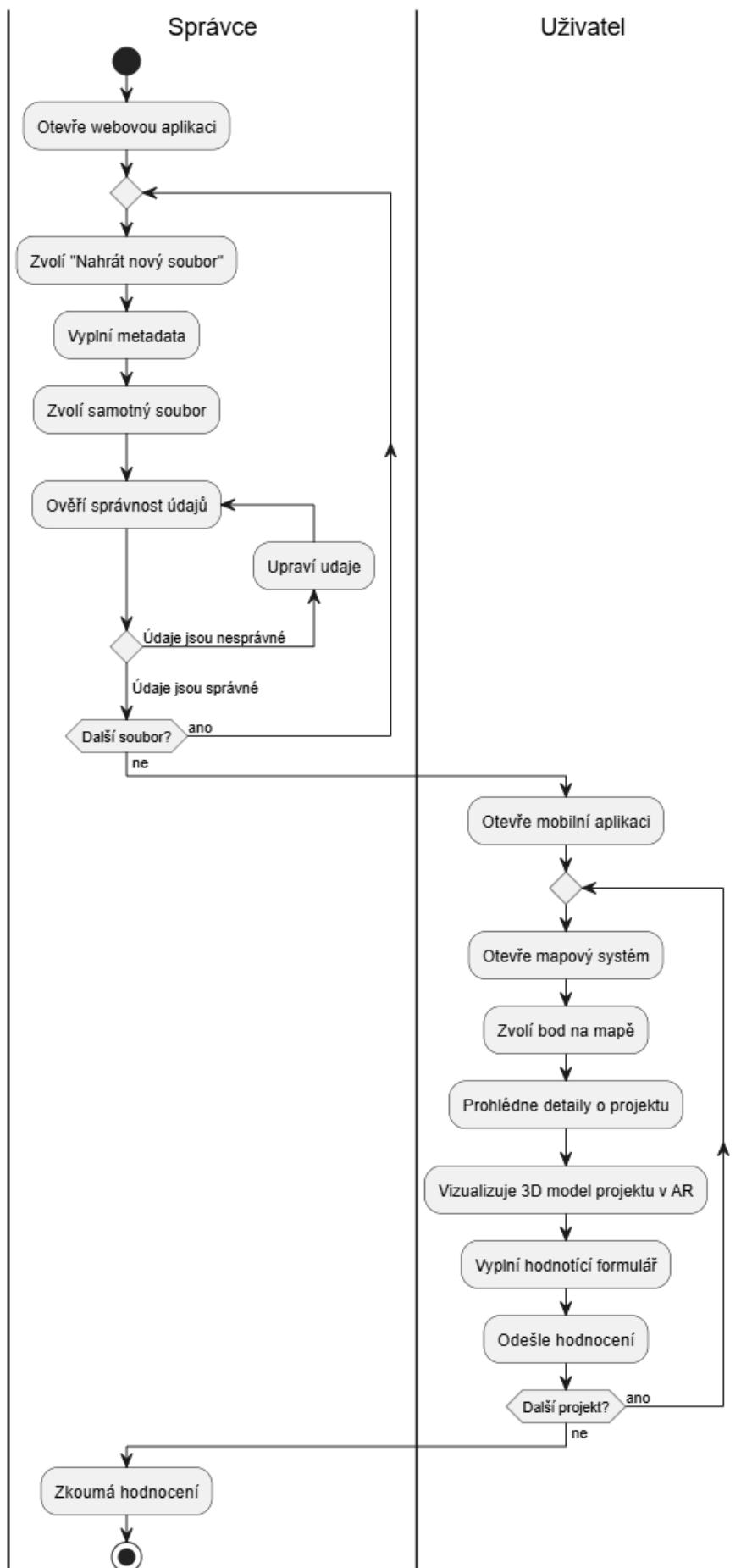


Obrázek 3.1.: Diagram užití systému

Diagram aktivit na obrázku 3.2 je rozdělen do dvou plaveckých dráh, které oddělují činnosti správce a běžného uživatele. V první plavecké dráze je zachyten proces, kdy správce otevře webovou aplikaci, zvolí možnost „Nahrát nový soubor“ a vyplní formulář s metadaty nového digitálního dvojčete. Po vyplnění provádí kontrolu správnosti údajů, v případě nesrovnalostí může data opakováně upravovat.

Druhá plavecká dráha obsahuje chování uživatele mobilní aplikace. Uživatel může opakovaně vybírat jednotlivé body na mapě, zobrazovat podrobnosti o projektech, vizualizovat jejich 3D modely v rozšířené realitě a odesílat svá hodnocení.

Na závěr se proces vrací zpět do plavecké dráhy správce, který může zobrazovat a analyzovat získaná uživatelská hodnocení.



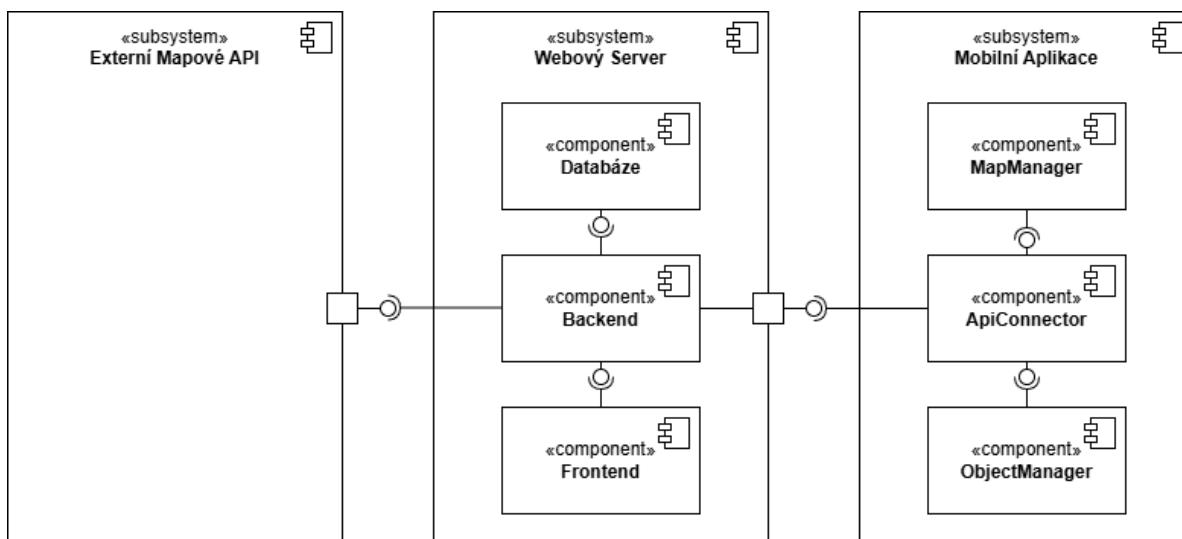
Obrázek 3.2.: Diagram aktivit

Architektura systému

Systém využívá architekturu klient-server. Klientskou část tvoří mobilní aplikace a webový frontend, který je sice součástí serverové části, ale z hlediska komunikace se serverem funguje jako klient. Serverová část poskytuje aplikační logiku prostřednictvím REST API a zajišťuje perzistenci dat v databázi.

Klientská část je realizována dvěma způsoby: mobilní aplikací pro rozšířenou realitu určenou koncovým uživatelům a webovou aplikací pro správu digitálních dvojčat. Mobilní aplikace zajišťuje interakci s uživatelem, vizualizaci digitálních dvojčat v rozšířené realitě, práci s mapovým rozhraním a komunikaci se serverem. Webová aplikace umožňuje nahrávat, upravovat a mazat digitální dvojčata, určovat jejich souřadnice a zobrazovat uživatelská hodnocení.

Serverová část přijímá požadavky od obou typů klientů, poskytuje data digitálních dvojčat, uchovává zpětnou vazbu uživatelů a spravuje příslušné datové modely. Základní architektura systému je znázorněna na obrázku 3.3. Webový server se skládá ze tří hlavních komponent: backendu poskytujícího aplikační logiku a REST API, databázové vrstvy zajišťující perzistenci dat a frontendu, jenž tvoří webové rozhraní. Komunikace s daty probíhá výhradně prostřednictvím backendu, který tak představuje jediný vstupní bod. Backend dále komunikuje s externím mapovým API za účelem získávání mapových dlaždic.

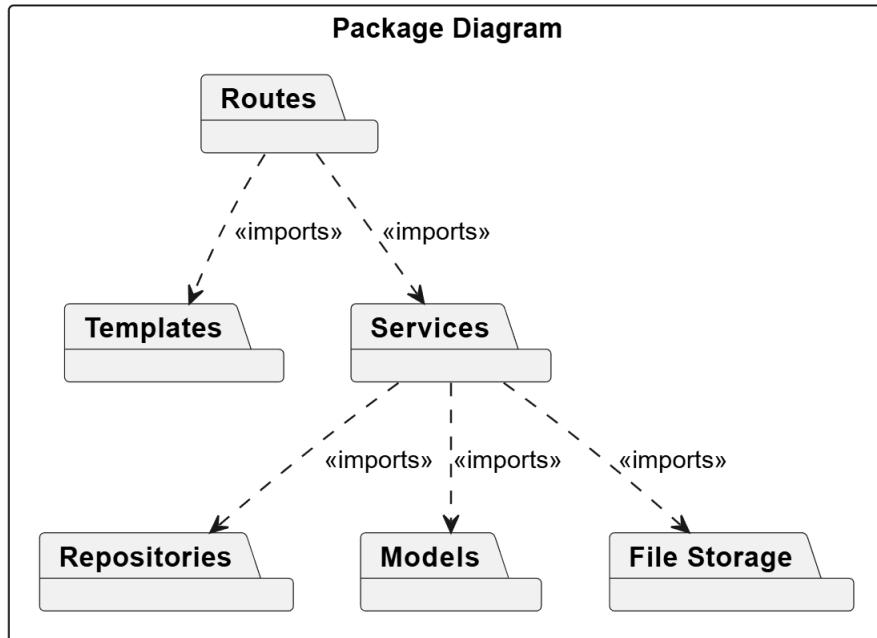


Obrázek 3.3.: Diagram komponent systému

Modulární návrh systému umožňuje samostatný a nezávislý rozvoj jednotlivých částí aplikace, čímž se usnadňuje jejich údržba, rozšiřování a případná nahraďba. Tento přístup rovněž zajišťuje efektivní a spolehlivou výměnu dat mezi klientskou a serverovou částí, což přispívá k vyšší robustnosti, flexibilitě a škálovatelnosti systému.

Architektura serverové části

Serverová část je navržena jako vrstevnatá aplikace, která zajišťuje rozdělení zodpovědností a usnadňuje údržbu. Aplikace je rozdělena do šesti hlavních balíčků, jejichž struktura je znázorněna na obrázku 3.4.



Obrázek 3.4.: Diagram balíčků serverové části

Modul *routes* zpracovává HTTP požadavky od obou typů klientů a definuje jak API endpointy pro mobilní aplikaci, tak webové endpointy pro administrační rozhraní. S ním úzce spolupracuje modul *templates*, který obsahuje šablony pro webové rozhraní určené pro správu digitálních dvojčat. Byznys logiku a validační pravidla implementuje modul *services*, který zajišťuje správnost zpracovávaných dat a koordinuje operace mezi ostatními vrstvami.

Přístup k datům a komunikaci s databází zajišťuje modul *repositories*, který poskytuje abstrakci pro datovou vrstvu a umožňuje nezávislou změnu databázové technologie. Doménové entity systému, jako jsou metadata digitálních dvojčat, metadata mapových dlaždic nebo uživatelská hodnocení, definuje modul *models*. Fyzické uložení 3D modelů a souvisejících souborů spravuje modul *file storage*.

Tato vrstevnatá architektura umožňuje nezávislý vývoj jednotlivých balíčků. Prezentační vrstva je oddělena od byznys logiky, což zajišťuje možnost snadné změny API nebo přidání nových rozhraní bez nutnosti modifikace hlavní aplikační logiky.

Návrh datových modelů

Pro zajištění konzistentní struktury dat byl navržen systém tří datových modelů, které reprezentují základní entity systému. Tyto modely definují, jakým způsobem jsou data ukládána v databázi

a jaké atributy jednotlivé entity obsahují.

FileMetadata

Model *fileMetadata* uchovává metadata projektů, konkrétně názvy GLB a obrazových souborů, zeměpisné souřadnice ve formátu latitude a longitude, název projektu a jeho textový popis. Každý záznam je identifikován pomocí UUID o které se stará server.

- **_id: str** – Unikátní identifikátor
- **glb_filename: str** – Název GLB souboru
- **img_filename: str** – Název obrázkového souboru
- **lat: double** – Zeměpisná šířka
- **lon: double** – Zeměpisná délka
- **name: str** – Název položky
- **description: str** – Popis položky

Ratings

Model *ratings* ukládá uživatelská hodnocení. Každé hodnocení obsahuje referenci na ID projektu, počet hvězdiček (1-5), volitelný textový komentář a časové razítko vytvoření. Tato struktura umožňuje efektivní dotazování všech hodnocení pro konkrétní projekt a výpočet průměrného hodnocení pomocí agregačních funkcí databáze.

- **_id: str** – Unikátní identifikátor
- **file_id: str** – ID hodnoceného souboru
- **rating: int** – Počet hvězdiček (1-5)
- **comment: str** – Komentář k hodnocení
- **created_at: Optional[datetime]** – Datum vytvoření (výchozí: aktuální datum a čas)

MapsCache

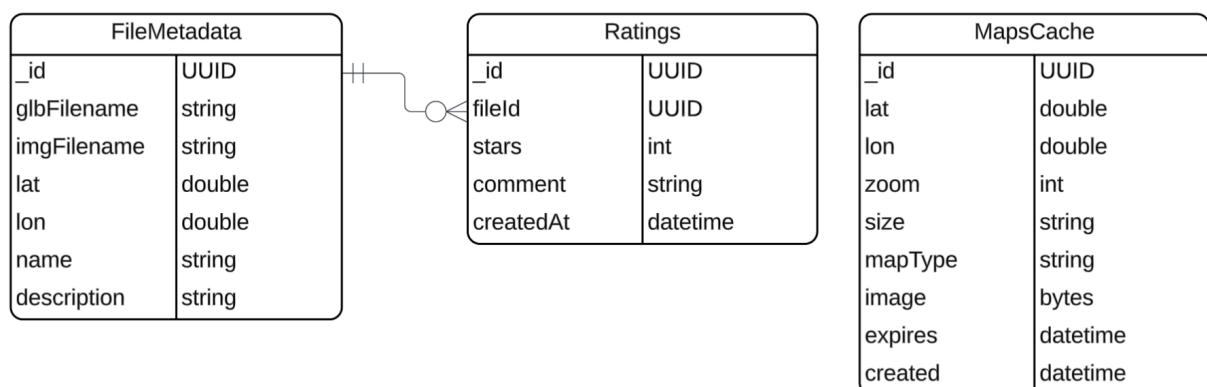
Model *mapsCache* slouží k optimalizaci načítání mapových podkladů. Ukládá stažené mapové dlaždice z Google Maps API společně s jejich parametry, kterými jsou zeměpisná šířka, zeměpisná délka a úroveň přiblížení a časové razítko expirace. Při dotazu na mapovou dlaždici server nejprve ověří, zda existuje platná dočasně uložená verze. Pokud ano, vrátí ji přímo z databáze, čímž se eliminují opakování dotazy na externí API a výrazně se zrychlí odezva aplikace.

- **_id: str** – Cache ID (automaticky generované)
- **lat: str** – Zeměpisná šířka

- lon: str – Zeměpisná délka
- zoom: str – Úroveň přiblížení
- size: str – Velikost mapy
- maptype: str – Typ mapy
- image: bytes – Binární data obrázku
- expires: Optional[datetime] – Datum expirace (výchozí: aktuální datum a čas + 10 dní)
- created: Optional[datetime] – Datum vytvoření (výchozí: aktuální datum a čas)

Entitně relační diagram

Entitně relační diagram na obrázku 3.5 znázorňuje strukturu entit systému a vztahy mezi jednotlivými entitami. Primární klíče jednotlivých entit jsou označeny atributem `_id`, zatímco vazby mezi entitami jsou realizovány pomocí cizích klíčů, například `file_id` ve vazbě mezi entitami. Diagram ukazuje logické propojení dat, což umožňuje přehledné pochopení struktury databáze a integritu dat v systému.



Obrázek 3.5.: Entitně relační diagram

Návrh REST API

REST API bylo navrženo jako rozhraní pro komunikaci mezi klientskými aplikacemi a serverem. API poskytuje následující endpointy:

- **GET /files** – vrací seznam všech dostupných projektů ve formátu JSON. Tento endpoint je využíván mobilní i webovou aplikací při inicializaci pro získání seznamu digitálních dvojčat.
- **GET /files/<id>** – poskytuje detailní informace o konkrétním projektu včetně metadat (název, popis, souřadnice) a seznamu všech hodnocení včetně průměrného hodnocení. Tento endpoint je volán při zobrazení náhledové obrazovky projektu.

- **GET /files/<filename>/download** – poskytuje 3D modely a náhledové obrázky ke stažení.
- **GET /files/<id>/ratings** – vrací všechna hodnocení pro daný projekt včetně průměrného hodnocení. Hodnocení jsou seřazena od nejnovějšího po nejstarší.
- **GET /maps/staticmap** – poskytuje mapové dlaždice na základě parametrů latitude, longitude, zoom. Dlaždice jsou dočasně ukládány v databázi po dobu 10 dní, což minimalizuje počet dotazů na Google Maps API a zrychluje načítání mapy.
- **POST /files** – přijímá nový projekt včetně GLB souboru, náhledového obrázku a metadat (název, popis, souřadnice). Tento endpoint je využíván webovým rozhraním pro správu projektů.
- **POST /files/<id>/ratings** – přijímá hodnocení od uživatelů ve formátu JSON obsahujícím počet hvězdiček (1–5) a volitelný textový komentář.
- **PUT /files/<id>** – aktualizuje existující projekt. Umožňuje změnu metadat a volitelně i nahrazení GLB souboru nebo náhledového obrázku.
- **DELETE /files/<id>** – umožňuje smazat projekt ze systému včetně všech souvisejících souborů a hodnocení.

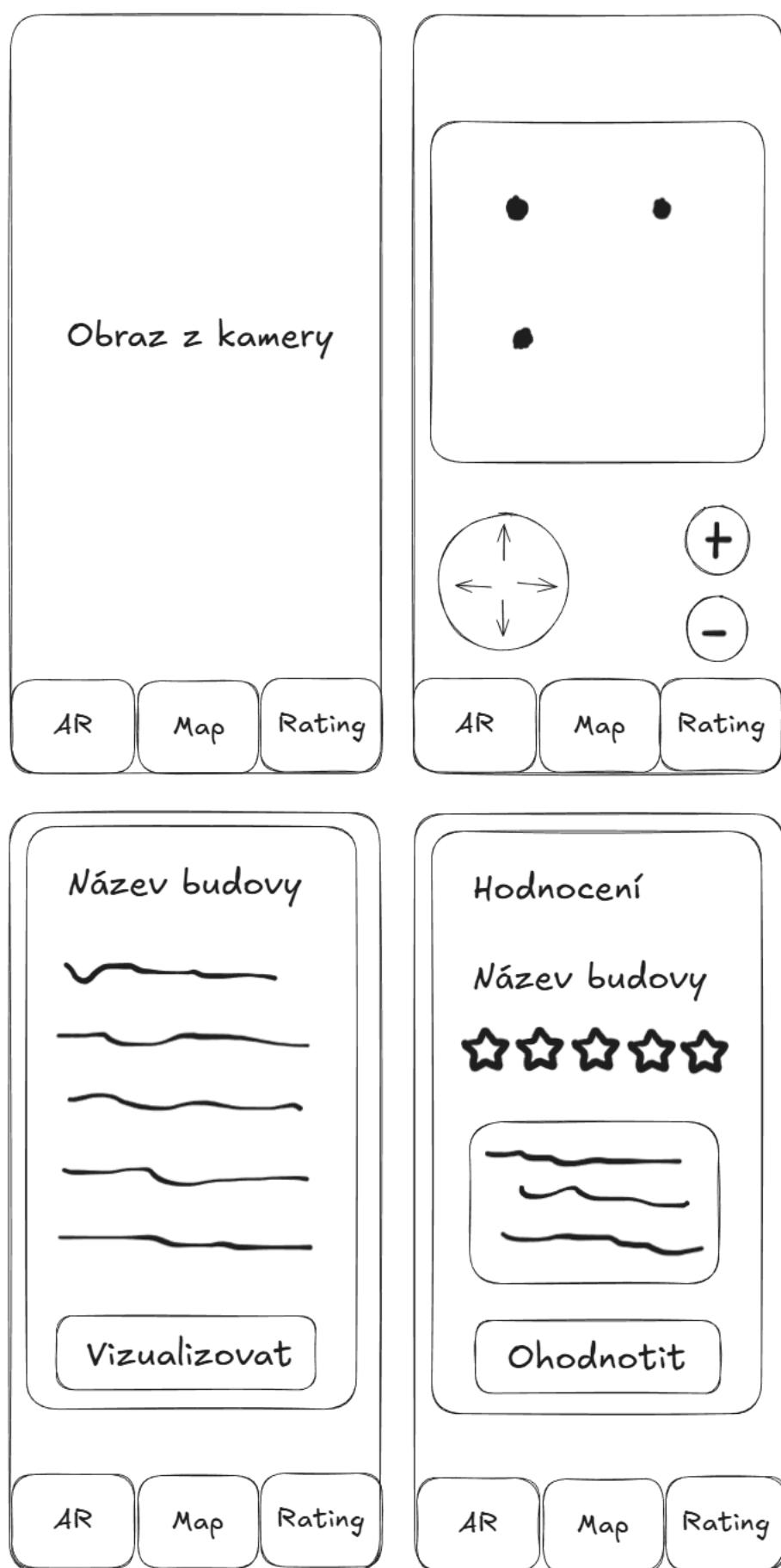
Klientská část (mobilní aplikace)

Klientská část mobilní aplikace slouží jako hlavní prostředí, ve kterém uživatel pracuje s digitálními dvojčaty. Umožňuje vyhledávat dostupné projekty pomocí interaktivní mapy. Po výběru je 3D model načten ze serveru a zobrazen v rozšířené realitě, přičemž není nutné používat fyzické markery. Model lze volně prohlížet z různých úhlů, což uživateli poskytuje lepší prostorovou představu o jeho podobě i fungování. Veškerá data jsou získávána dynamicky ze serveru, aby byla zajištěna aktuálnost jednotlivých projektů, a po prohlédnutí může uživatel odeslat zpětnou vazbu prostřednictvím jednoduchého formuláře.

Návrh uživatelského rozhraní

Uživatelské rozhraní je koncipováno jako přehledné a intuitivní. Obsahuje hlavní AR obrazovku s digitálním dvojčetem, mapové rozhraní pro výběr projektů, informační panel s detailem projektu a formulář pro zpětnou vazbu. Navigace v jednotlivých obrazovkách je řešena spodním navigačním panelem. Ostatní ovládací prvky jsou ukotveny k okrajům obrazovky tak, aby byla zajištěna alespoň minimální forma responzivity rozhraní.

Před samotnou tvorbou prototypu byl vytvořen návrh ve formě jednoduchých skic, které sloužily k ověření rozvržení prvků a logiky navigace mezi obrazovkami. Na obrázku 3.6 je vidět ukázka jedné z těchto skic, která poskytla vizuální vodítko pro následnou tvorbu digitálního prototypu.



Obrázek 3.6.: Ukázka skici návrhu uživatelského rozhraní

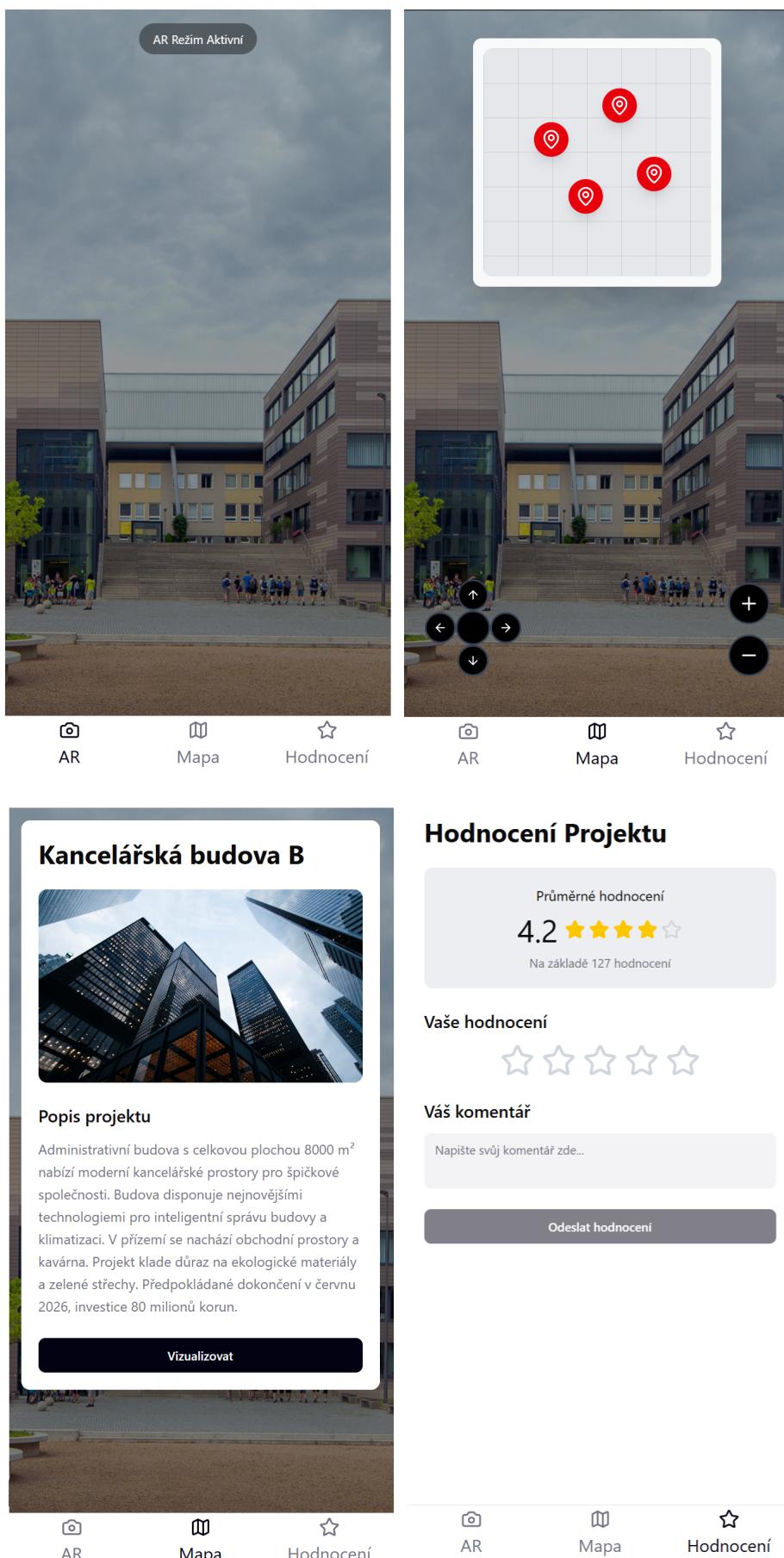
Návrh uživatelského rozhraní na obrázku 3.7 zahrnuje čtyři klíčové obrazovky, které společně pokrývají celý proces práce s digitálním dvojčetem. První z nich je AR režim, který umožňuje zobrazit 3D model přímo v reálném prostředí. Uživatel si může model zobrazit přímo v reálném prostředí před sebou. Tento pohled je navržen s důrazem na minimální množství ovládacích prvků, aby nerušil vizuální dojem a umožnil maximální prostor pro samotnou vizualizaci.

Druhá obrazovka představuje mapový systém, který slouží jako hlavní způsob výběru dostupných digitálních dvojčat. Uživatel zde může posouvat, přibližovat či oddalovat mapu, vybírat projekty podle jejich polohy.

Třetí obrazovka poskytuje náhled projektu, tedy stránku s důležitými informacemi o vybraném digitálním dvojčeti. Zobrazuje název projektu, ilustrační obrázek a stručný popis, který uživateli pomáhá pochopit kontext před vstupem do rozšířené reality. Tato obrazovka slouží jako logický mezíkrok mezi mapou a samotným AR režimem.

Poslední obrazovkou je hodnocení projektu, které umožňuje uživateli poskytnout zpětnou vazbu. Kombinuje textové pole pro komentář a hvězdičkové hodnocení. Tato část rozhraní podporuje jednoduché a rychlé vyplnění tak, aby zpětná vazba nebyla pro uživatele zbytečnou zátěží.

Napříč všemi obrazovkami byla věnována pozornost také volbě velikosti písma. Nadpisy a klíčové informace využívají větší, výraznější typografii, zatímco doprovodné texty a sekundární prvky pracují s menší velikostí písma. Tento přístup zajišťuje zřetelné hierarchické členení informací, dobrou čitelnost a jednotný vizuální styl napříč celou aplikací.



Obrázek 3.7.: Prototyp návrhu mobilní aplikace

Sekvenční diagram systému

Sekvenční diagram na obrázku 3.8 znázorňuje sekvenci zpráv při hlavních scénářích použití. Jedná se o samotné zapnutí aplikace, zobrazení mapy, výběr projektu, vizualizaci a hodnocení projektu. Na diagramu vystupuje pět klíčových aktérů/objektů, kterými jsou uživatel, mobilní aplikace, serverová část, databáze a externí služba Google Maps API.

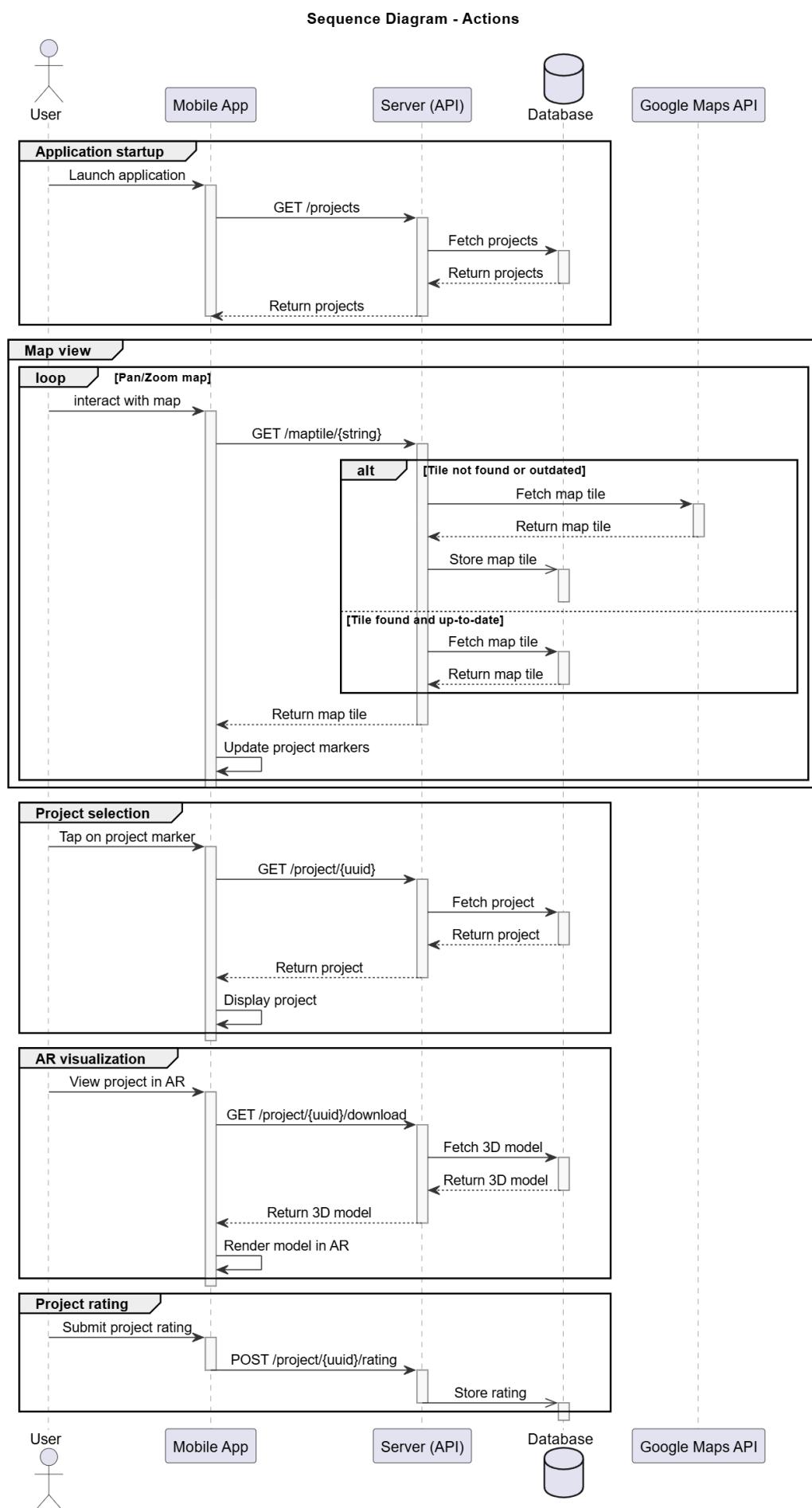
Část diagramu *Application startup* obsahuje odeslání požadavku z mobilní aplikace na server a žádost o seznam dostupných projektů. Server této žádosti vyhoví, data načte z databáze a vrátí je zpět mobilní aplikaci, aby bylo možné přejít na obrazovku s mapovým systémem.

V části diagramu *Map View* dochází k opakovanému načítání mapových dlaždic podle toho, jak uživatel s mapou interaguje (pohyb, přibližování a oddalování). Mobilní aplikace nejprve ověří dostupnost dlaždice na serveru. Pokud požadovaná dlaždice není dostupná, nebo není aktuální, server ji nejprve stáhne z Google Maps API, uloží do databáze a následně odešle klientovi. V opačném případě je načtena přímo z databáze. Klient poté aktualizuje pozice projektových bodů na mapě.

Při akci *Project selection* odešle mobilní aplikace dotaz na server a získá detailní informace o projektu. Po jejich načtení z databáze se uživateli zobrazí náhledová obrazovka, která umožňuje přejít do režimu rozšířené reality.

V okamžiku, kdy uživatel požádá o zobrazení digitálního dvojčete v AR v části diagramu *Project visualization*, aplikace si od serveru vyžádá 3D model příslušného projektu. Server jej získá z databáze, odešle zpět do mobilní aplikace a ta následně model vizualizuje v AR.

Poslední část diagramu *Project rating* popisuje odeslání uživatelské zpětné vazby. Po vyplnění hodnocení odešle aplikace data na server, který uloží hodnocení do databáze.



Obrázek 3.8.: Sekvenční diagram systému

3.3. Vybraný technologický zásobník

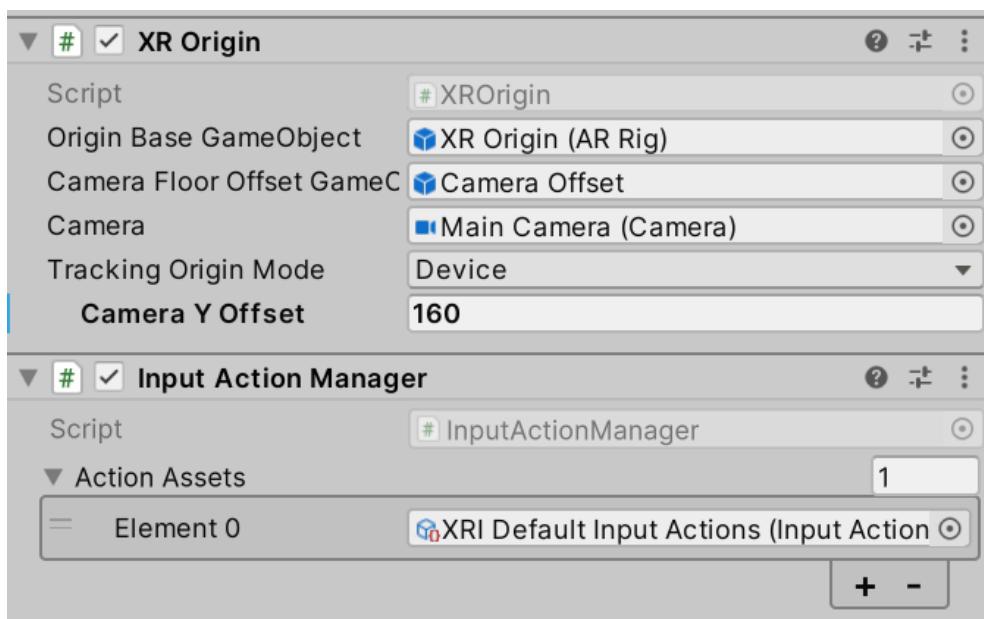
Pro vývoj aplikace jsem zvolil technologický zásobník, který kombinuje nástroj pro tvorbu AR aplikací, spolehlivý serverový backend a dokumentově orientovanou databázi. Jednotlivé technologie jsem vybral primárně kvůli dostupnosti kvalitní dokumentace a podpoře funkcionalit, které tato práce vyžaduje.

Klientská část

Základ klientské časti tvoří herní engine Unity, který je v současnosti jedním z nejpoužívanějších nástrojů pro vývoj AR a VR aplikací. Hlavní výhodou Unity je jeho multiplatformnost, podpora velkého množství XR zařízení a stabilní ekosystém knihoven. [13]

K implementaci logiky aplikace byl využit objektově orientovaný jazyk C#, který Unity nativně podporuje. C# nabízí jasnou syntaxi, dobrou práci s objekty a bohaté množství knihoven. [14]

Obrázek 3.9 zobrazuje třídy, které zajišťují samotné fungování AR. Třída XR Origin je zodpovědná za zpracování vstupů, které jí předává Input Action Manager. Díky těmto třídám je celý základ VR či AR projektu implementován, protože tyto třídy zajišťují plnou integraci kamery do prostředí, správné fungování pohybu a interakce s objekty.



Obrázek 3.9.: Ukázka klíčových tříd pro fungování AR

Serverová část

Pro serverovou část aplikace byl zvolen programovací jazyk Python v kombinaci s frameworkem Flask. Flask představuje minimalistický a snadno rozšiřitelný framework, který umožňuje rychlou

tvorbu REST API i jednoduchou správu jednotlivých endpointů. Díky nízké režii je vhodný pro projekty, které vyžadují lehkou, přehlednou a snadno udržovatelnou serverovou logiku. [15]

Na ukázce 3.1 je vidět základní struktura API endpointu implementovaného ve frameworku Flask. Endpoint je zodpovědný za získávání metadat všech souborů spravovaných službou file_service. Díky dekorátorům Flask automaticky zajišťuje routování požadavků a umožňuje vracet data ve formátu JSON bez nutnosti manuální manipulace s HTTP odpovědí. Funkce `get_file_metadata` volá metodu `get_all_files_metadata` služby a předává její výstup přímo klientovi.

```
@bp.route("/files", methods=["GET"])
def get_file_metadata():
    """API_endpoint_pro_ziskani_metadat_vsech_souboru"""
    metadata = file_service.get_all_files_metadata()
    return jsonify(metadata)
```

Ukázka kódu 3.1: Základní struktura endpointu

Pro zajištění přenositelnosti a konzistentního prostředí je server provozován pomocí kontejnerizační technologie Docker. Kontejnery umožňují izolovat běhové prostředí, usnadňují nasazení a umožňují jednotnou konfiguraci nezávislou na cílovém zařízení. [16]

Datová vrstva je řešena dokumentově orientovanou databází MongoDB. Tento typ databáze umožňuje ukládat nestrukturovaná či částečně strukturovaná data. Výhodou MongoDB je také jednoduchá integrace s Pythonem a možnost dynamicky přizpůsobovat datový model podle potřeb aplikace. [17]

Datové formáty

Pro přenos metadat mezi klientem a serverem byl zvolen formát JSON díky jeho jednoduché struktuře a široké podpoře. 3D modely jsou ukládány a přenášeny ve formátu GLB, který je standardizovaným formátem pro web a mobilní aplikace. GLB kombinuje geometrii, textury a materiály do jediného binárního souboru, což zjednoduší přenos a zajišťuje rychlé načítání. [18]

Komunikace mezi klientem a serverem probíhá přes HTTPS protokol pomocí REST API, které poskytuje jednotné rozhraní pro všechny operace se systémem.

3.4. Implementace systému

Implementace probíhala podle předem navržené architektury a metodiky, přičemž jsem se snažil o co největší modularitu a přehlednost kódu. Samotná implementace probíhala v několika etapách, při kterých jsem zajišťoval jednotlivé funkcionality.

Mobilní Aplikace

Integrace rozšířené reality

Pro zajištění správné interakce s 3D objekty v rozšířené realitě jsem použil XR Interaction Toolkit, což je knihovna poskytující třídy pro vytváření aplikací s imerzivními technologiemi, které zajišťují interakce se vstupy v reálném čase.

Po vložení balíčku XR Interaction Toolkit jsem vytvořil XR rig, což je třída, která se automaticky stará o správu kamery a pohyb zařízení. XR rig také zajišťuje, že všechny vstupy, jako jsou pohyby kamery nebo interakce s objekty, jsou správně přeneseny do 3D prostoru aplikace. Tento prvek významně zjednodušil práci s kamerou a senzory, což je zásadní pro plynulé fungování aplikace.

Import modelu a správné pozicování

Další krok představovalo zajištění správné knihovny pro načítání 3D modelů v reálném čase. Většina dostupných knihoven nepodporuje asynchronní stahování modelu, proto jsem určil GLTFast jako nevhodnější knihovnu pro tuto práci.

Po úspěšném načtení modelu bylo nutné zajistit jeho správné umístění v reálném prostoru. Rozhodl jsem se pro manuální přepočítání pozice tak, aby se model objevil přímo v záběru kamery. Zde hraje roli pozice kamery, směr pohledu a vzdálenost modelu.

Mapový systém

Po vyhodnocení dostupných balíčků pro integraci mapového podkladu do Unity jsem zjistil, že žádné z bezplatných řešení nesplňuje požadavky projektu. Z tohoto důvodu jsem se rozhodl vytvořit mapový systém vlastní implementací. Pro generování mapových dlaždic bylo využito rozhraní Google Maps Static API, které na základě zeměpisné šířky, zeměpisné délky a úrovně přiblížení poskytuje statický obraz odpovídající mapové oblasti. Získaná dlaždice byla následně převedena na texturu a integrována do prostředí Unity.

Následným krokem byla vizualizace jednotlivých projektů ve formě bodů umístěných na mapě. Tato část se ukázala jako náročnější, než bylo původně očekáváno, protože bylo nutné převést geografické souřadnice na pozice v lokálním souřadnicovém systému Unity.

Ačkoli existují standardní vzorce pro převod zeměpisné šířky a délky do 2D projekce, nelze je přímo aplikovat na výpočet pozice v prostředí Unity. Unity nepoužívá geografickou projekci, ale lokální kartézský souřadnicový systém. Bylo tedy nezbytné vytvořit vlastní transformační vztah, který umožňuje přepočítávat pozice bodů tak, aby se správně zobrazovaly na mapě.

Pro odvození tohoto vztahu jsem vybral několik referenčních bodů se známými geografickými souřadnicemi. Jako referenční body jsem použil rohy mapové dlaždice a její střed, jejichž pozice v Unity lze přesně určit. Průměrem poměrů jejich geografických souřadnic s pozicemi v Unity

jsem odvodil transformační koeficient C, jeho přesnost jsem následně ověřil vložením bodů na velmi vzdálená místa od středu původní mapové dlaždice.

Tento koeficient následně umožnil vypočítat relativní posun objektů v rámci mapové dlaždice. Rovnice pro výpočet souřadnice X vychází z rozdílu mezi zeměpisnou šírkou objektu $\text{Lat}_{\text{objekt}}$ a středem dlaždice $\text{Lat}_{\text{dlaždice}}$, normalizovaného pomocí koeficientu C a dále škálovaného podle rozdílu mezi aktuální úrovní přiblížení Z a základní úrovně přiblížení Z_0

$$X = \frac{\text{Lat}_{\text{objekt}} - \text{Lat}_{\text{dlaždice}}}{C} \times 2^{Z-Z_0}$$

Analogicky je určena i souřadnice Y. Obě vypočtené hodnoty určují konečné umístění vizualizovaných projektů na mapě a zajišťují korektní zobrazení při všech úrovních přiblížení v prostředí Unity.

Asynchronní stahování 3D modelu

Po výběru projektu uživatelem odešle mobilní aplikace HTTP GET požadavek na endpoint `/files/<filename>/download` pro stažení GLB souboru. Server vrátí binární data modelu s MIME typem `model/gltf-binary`.

Unity přijme data a předá je knihovně GLTFast, která zajišťuje parsing GLB formátu. GLTFast asynchronně zpracovává binární data, extrahuje geometrii, textury a materiály a vytváří z nich Unity GameObject se vsemi potřebnými komponentami (Mesh, MeshRenderer, Material).

Po úspěšném načtení je model umístěn do scény podle vypočtené pozice vzhledem ke kameře. Model se objeví ve vzdálenosti přibližně metr před kamerou. Toto umístění zajišťuje, že model je okamžitě viditelný v zorném poli bez nutnosti pohybu ze strany uživatele.

Pro zajištění plynulého načítání je celý proces prováděn asynchronně pomocí C# korutiny, takže aplikace zůstává responzivní i během stahování a parsování větších modelů.

Serverová část

Implementace datových modelů

Datové modely byly implementovány jako Python dataclass třídy, které poskytují automatickou generaci inicializační metody a dalších pomocných metod. Každý model obsahuje metody `to_dict()` a `from_dict()`, které umožňují jednoduchou serializaci a deserializaci dat pro uložení do MongoDB.

Pro práci s databází byla využita knihovna PyMongo, která poskytuje nativní rozhraní pro komunikaci s MongoDB z Pythonu. Všechny operace s databází jsou v repository třídě, což zajišťuje separaci datové logiky od byznys logiky aplikace.

Validace a zpracování požadavků

Při příjmu dat od klientů probíhá validace na několika úrovních. Flask router nejprve ověřuje správnost HTTP metody a přítomnost povinných parametrů. Následně Service vrstva kontroluje datové typy, rozsahy hodnot (např. počet hvězdiček 1-5) a správnost souborů (přípona, velikost).

Pro upload souborů je využita knihovna Werkzeug, která je součástí Flaska. Soubory jsou ukládány do souborového systému s unikátním názvem UUID, což předchází konfliktům a zajišťuje bezpečnost.

Cachování mapových dlaždic

Implementace cachování mapových dlaždic výrazně optimalizuje výkon systému. Při požadavku na mapovou dlaždici server nejprve ověří, zda existuje platný záznam v kolekci `maps_cache`. Identifikátor je generován pomocí MD5 hashe z parametrů požadavku (`latitude`, `longitude`, `zoom`, `size`, `maptype`).

Pokud cache obsahuje platný záznam (expirace > aktuální čas), server vrátí uloženou dlaždici přímo z databáze. V opačném případě server stáhne novou dlaždici z Google Maps Static API, uloží ji do databáze s expirací 10 dní a odešle klientovi. Tato strategie minimalizuje náklady na volání externího API a zrychluje odezvu aplikace.

Implementace webové aplikace

Detailní návrh uživatelského rozhraní pro webovou aplikaci nebyl v rámci této práce vypracován, neboť se jedná o nástroj pro správu digitálních dvojčat s omezeným okruhem uživatelů, kde je kláden důraz primárně na funkčnost a efektivní správu dat. Pro vývoj webového rozhraní byl využit iterativní přístup, kdy bylo rozhraní průběžně upravováno podle aktuálních potřeb samotné implementace.

Webové rozhraní pro správu digitálních dvojčat bylo vytvořeno jako součást serverové aplikace s využitím šablonovacího systému Flask. Uživatelské rozhraní je implementováno jako jednoduché webové rozhraní, které umožňuje uživatelům kompletní správu projektů.

Hlavní stránka zobrazuje přehled všech nahraných projektů ve formě karet, které obsahují náhledový obrázek, název, popis, geografické souřadnice a průměrné hodnocení. Pro stylování byla využita CSS knihovna Tailwind, která poskytuje utility třídy pro rychlou tvorbu responzivního rozhraní.

Nahrávání nového projektu probíhá prostřednictvím modálního okna s formulářem, kde uživatel vyplní název, popis, souřadnice a nahraje GLB soubor spolu s náhledovým obrázkem. Po odeslání formuláře jsou soubory validovány na serveru (kontrola přípony, velikosti) a uloženy do souborového systému se zachováním UUID jako názvu souboru. Metadata jsou následně uložena do MongoDB databáze.

Úprava existujících projektů je řešena podobným modálním formulářem, který je předvyplněn aktuálními daty projektu. Uživatel může změnit jakákoli metadata nebo nahradit aktuální soubory.

Pro zobrazení hodnocení jednotlivých projektů slouží další modální okno, které dynamicky načítá hodnocení pomocí dotazu na endpoint `/files/<id>/ratings`. Hodnocení jsou zobrazena včetně počtu hvězdiček, komentáře a data vytvoření.

Veškerá interakce s uživatelem (otevírání modálních oken, potvrzení smazání, odeslání formulářů) je řešena pomocí JavaScriptu, který komunikuje se serverem asynchronně pomocí Fetch API, což zajišťuje plynulý uživatelský zážitek bez nutnosti obnovovat stránku.

Testování systému

Testování aplikace probíhalo formou nasazení na fyzické mobilní zařízení s operačním systémem Android. Pro testování byl použit telefon Xiaomi Redmi 10T Pro s verzí Androidu 12.

Testování na reálném zařízení bylo zvoleno jako primární metoda, protože emulátory neumožňují plnohodnotné ověření funkcí rozšířené reality, které vyžadují fyzické senzory a kameru zařízení. Webové rozhraní bylo také testováno vzdáleně z mobilního telefonu, nikoliv z hostujícího zařízení.

Během testování byly manuálně ověřeny všechny klíčové funkcionality systému:

- Správné zobrazení 3D modelů v rozšířené realitě a jejich pozicování v prostoru
- Funkčnost mapového rozhraní včetně posunu, přibližování a výběru projektů
- Komunikace se serverem (stahování modelů, náhledových obrázků, metadat a mapových dlaždic)
- Asynchronní načítání a renderování GLB souborů
- Odesílání uživatelských hodnocení
- Správa projektů prostřednictvím webového rozhraní (vytváření, úprava, mazání)
- Zobrazení hodnocení a komentářů prostřednictvím webového rozhraní

Všechny testované funkcionality pracovaly správně a systém prokázal základní funkčnost potřebnou pro účely prototypu.

Nasazení systému

Systém se skládá ze serverové části běžící v Docker kontejnerech a mobilní aplikace pro platformu Android. Pro úspěšné nasazení serverové části je nutné mít nainstalovaný Docker a funkční internetové připojení pro přístup k Google Maps API.

Před spuštěním serveru je třeba nakonfigurovat přístup k Google Maps Static API. V kořenovém adresáři složky `WebServer` je umístěn soubor `.env`, který obsahuje konfigurační proměnnou

`GOOGLE_MAPS_API_KEY`. Hodnotu této proměnné je nutné nahradit platným API klíčem získaným z Google Cloud Console.

Pro spuštění serverové části spušťte příkaz `docker-compose up` v kořenovém adresáři složky `WebServer`. Docker Compose automaticky stáhne potřebné obrazy, nainstaluje závislosti, vytvoří kontejnery `flask` a `mongo` a spustí flask server na portu 5000. Webové rozhraní pro správu digitálních dvojčat je tedy přístupné na adrese `http://localhost:5000`.

Před sestavením mobilní aplikace je nutné v Unity editoru otevřít projekt ze složky `ARDigitalTwins`, vyhledat objekt `DBConnector` a nastavit adresu serveru. Pro testování v Android emulátoru se používá adresa `http://localhost:5000`, pro fyzické zařízení je třeba použít IP adresu hostujícího zařízení v lokální síti. Prototyp nebyl navržen pro produkční nasazení na veřejnou IP adresu bez dodatečných bezpečnostních opatření.

Sestavení mobilní aplikace probíhá v Unity prostřednictvím menu `File → Build Settings`, kde je třeba vybrat platformu `Android` a spustit samotné sestavení. Unity vytvoří instalovatelný APK soubor, který lze nainstalovat na Android zařízení.

Po úspěšném nasazení je vhodné provést základní ověření funkčnosti. U serverové části je nutné ověřit dostupnost webového rozhraní a v mobilní aplikaci je třeba zkонтrolovat načtení mapového systému. Úspěšné ověření těchto základních funkcí indikuje správnou funkčnost celého systému.

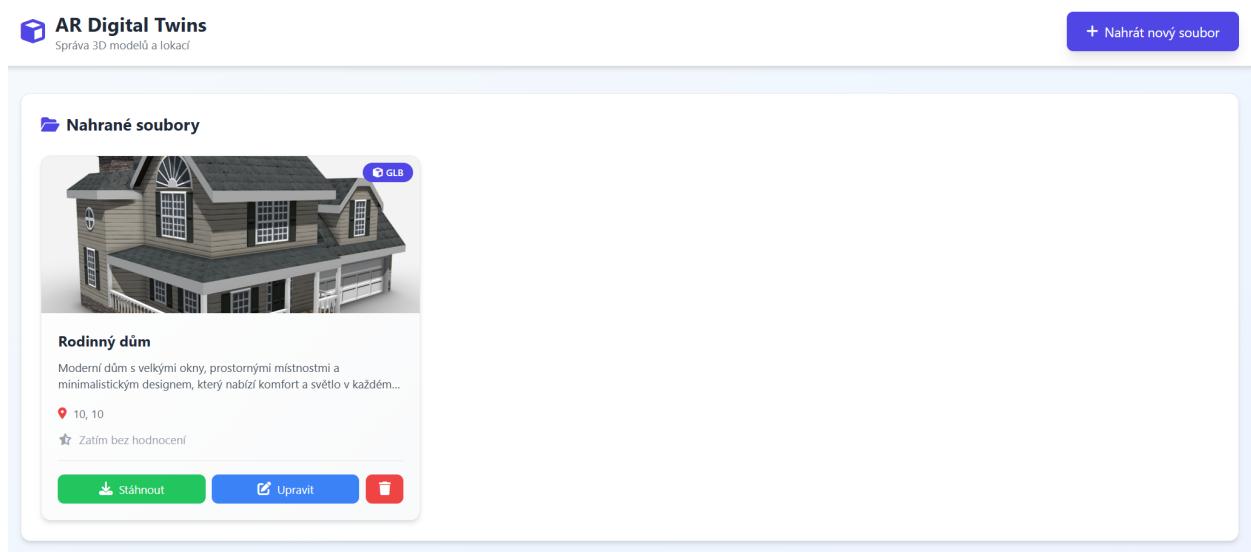
4. Diskuse a výsledky

4.1. Ověření funkčnosti systému

Tato kapitola demonstruje praktickou funkčnost implementovaného systému na konkrétním příkladu hypotetického projektu venkovního posezení. Město plánuje instalaci stolu s integrovanými lavicemi na místním sídlišti a před realizací chce zjistit názor občanů. Správce proto vytvoří digitální dvojče tohoto posezení, které si mohou obyvatelé prohlédnout v rozšířené realitě a vyjádřit svůj názor. Následující sekce ukazují proces správy digitálního dvojčete správcem a interakci koncového uživatele s mobilní aplikací, čímž ověřují, že systém splňuje navržené požadavky.

Správa digitálních dvojčat

Správce zahajuje proces přidání nového projektu otevřením webové aplikace. Hlavní stránka webového rozhraní zobrazuje přehled projektů ve formě karet obsahujících náhledový obrázek, název, popis, souřadnice, průměrné hodnocení a tlačítka pro správu, viz obrázek 4.1.



Obrázek 4.1.: Webové rozhraní s přehledem projektů

4. Diskuse a výsledky

Správce klikne na tlačítko pro přidání projektu, které otevře modální formulář. Zde vyplňuje metadata projektu vyplní pole název, do popisu uvádí informace o plánované instalaci venkovního posezení, doplňuje souřadnice lokality na sídlišti a nahrává soubory (GLB model stolu s lavicemi a náhledový obrázek), viz obrázek 4.2.

Nahrát nový soubor

Název
Stůl s lavicemi

Popis
Plánovaná instalace venkovního posezení kombinující stůl s integrovanými lavicemi. Projekt je určen pro veřejný prostor na sídlišti. Před realizací chce město zjistit zájem o tento typ posezení.

Zeměpisná šířka
50

Zeměpisná délka
14

GLB Soubor
posezeni.glb

Náhledový obrázek
posezeni.png

Zrušit Nahrát soubor

Obrázek 4.2.: Formulář pro přidání nového projektu

V případě potřeby má správce možnost projekt zcela odstranit. Smazání vyžaduje potvrzení a odstraní projekt včetně všech přidružených souborů a hodnocení, viz obrázek 4.3.

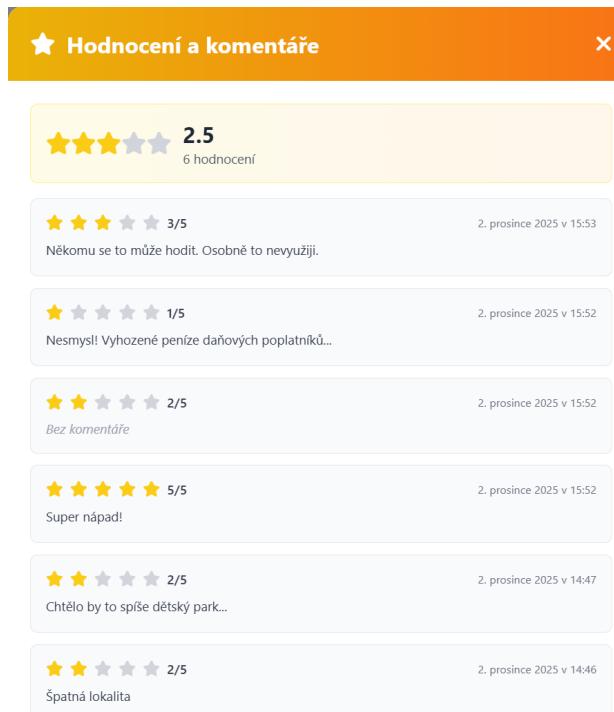


Obrázek 4.3.: Potvrzovací okno pro smazání

Správce může také data následně upravit pomocí editačního formuláře. Pokud by například zadal špatné souřadnice nebo chtěl změnit popis projektu, může pomocí tohoto formuláře údaje opravit. Formulář je předvyplněn aktuálními daty a umožňuje úpravu všech metadat i nahrazení souborů, viz obrázek 4.4.

Obrázek 4.4.: Formulář pro úpravu existujícího projektu

Po zveřejnění projektu s posezením může veřejnost odesílat svou zpětnou vazbu. Správce má k dispozici funkci pro zobrazení všech hodnocení projektu, kde vidí veškerou zpětnou vazbu. Hodnocení jsou seřazena chronologicky od nejnovějšího a obsahují počet hvězdiček, volitelný komentář a časové razítko. Tato zpětná vazba slouží městu k vyhodnocení, zda obyvatelé sídliště instalaci posezení vítají, viz obrázek 4.5.



Obrázek 4.5.: Zobrazení hodnocení a komentářů projektu

Interakce uživatele s digitálními dvojčaty

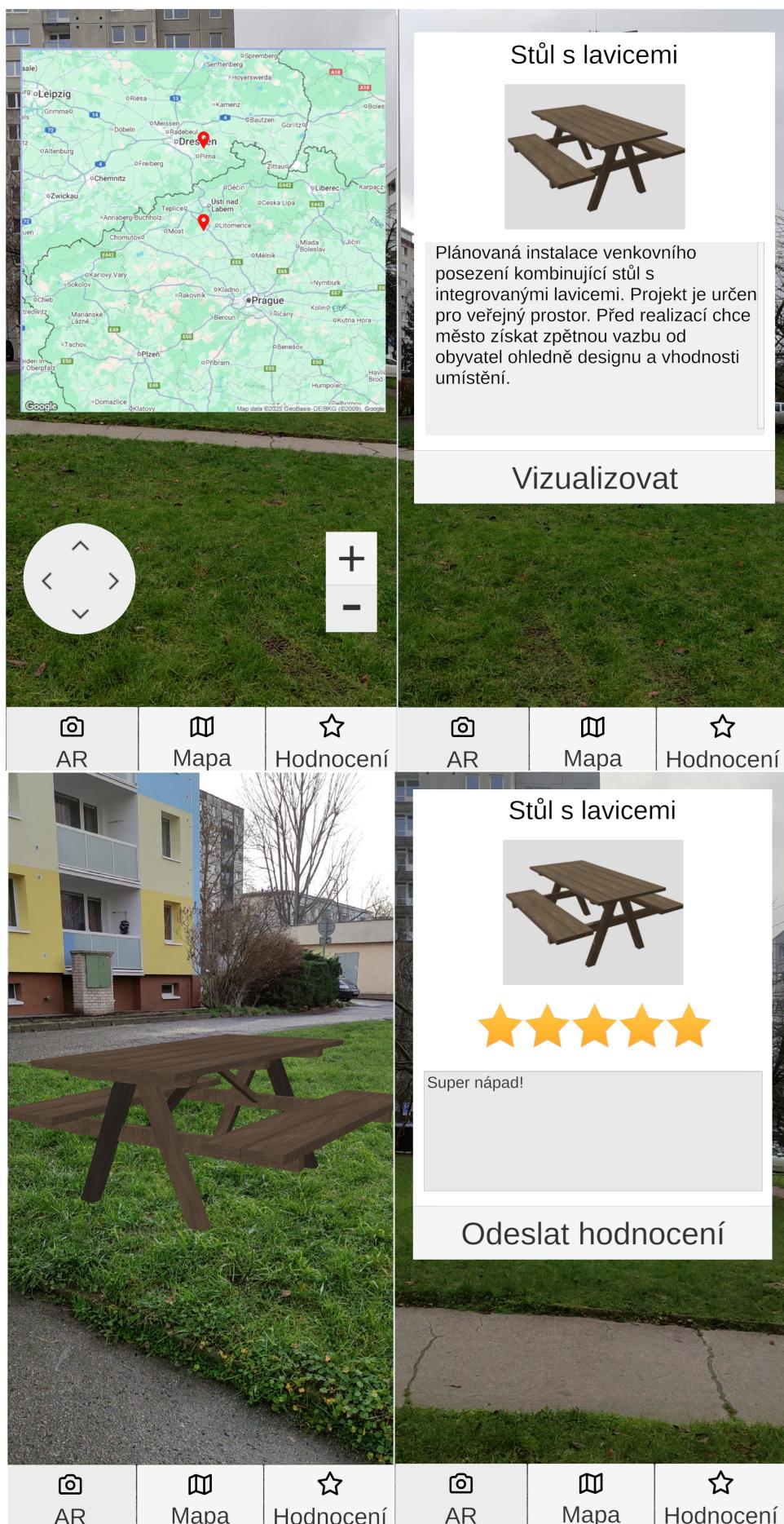
Obyvatel sídliště si chce prohlédnout plánované posezení a vyjádřit svůj názor. Otevře mobilní aplikaci, která automaticky načte všechny dostupné projekty z databáze. Aplikace zobrazí interaktivní mapové rozhraní s vyznačenými pozicemi všech digitálních dvojčat, včetně bodu označujícího plánovanou lokaci posezení na sídlišti. Navigace v mapě je intuitivní, uživatel ovládá zobrazení pomocí tlačítka pro přiblžení (+/-) a čtyřsměrného ovladače pro posouvání. Aplikace dynamicky načítá nové mapové dlaždice při každé změně zobrazení a průběžně přepočítává pozice bodů tak, aby odpovídaly aktuální mapové dlaždici.

Uživatel klikne na bod označující projekt s posezením, čímž se otevře náhledový panel. Panel obsahuje název projektu, náhledový obrázek modelu, popis plánované instalace a tlačítko pro zahájení vizualizace.

Po stisknutí tlačítka pro zahájení vizualizace aplikace plynule přejde do režimu rozšířené reality. Během tohoto přechodu se na pozadí stahuje 3D model stolu s lavicemi ve formátu GLB. Jakmile je stahování dokončeno, model se automaticky zobrazí ve fyzickém prostředí uživatele, v tomto případě přibližně ve vzdálenosti jednoho metru před ním. Toto umožňuje uživateli získat realistickou představu o velikosti posezení a posoudit, jak by mohlo vypadat v reálném prostředí. V AR režimu uživatel interaguje s modelem prostřednictvím přirozených pohybů zařízení, může se volně pohybovat kolem objektu, přiblížovat se pro detailnější prohlídku nebo se vzdalovat pro komplexnější pohled.

Po prohlédnutí modelu posezení může uživatel vyjádřit svůj názor prostřednictvím hodnotícího formuláře dostupného ze spodního navigačního panelu. Uživatel uděluje hodnocení pomocí hvězdiček v rozsahu 1–5 a může přidat volitelný textový komentář, například zda se mu samotný projekt líbí nebo zda má nějaké připomínky. Po odeslání systém validuje zadaná data a ukládá je do databáze společně s časovým razítkem, čímž umožňuje městu sledovat zpětnou vazbu obyvatel v čase.

Uživatel se může vrátit na mapové rozhraní prostřednictvím navigačního panelu, kde může vybrat další projekt a celý proces se opakuje.



Obrázek 4.6.: Ukázka jednotlivých obrazovek v mobilní aplikaci

4.2. Hodnocení splnění cílů

Hlavním cílem bylo vytvořit funkční prototyp pro zobrazení digitálních dvojčat v rozšířené realitě.

Mobilní aplikace umožňuje zobrazení 3D modelů v AR bez fyzických markerů, mapové rozhraní pro výběr projektů a zpětnou vazbu formou hodnocení a komentářů. Uživatelské rozhraní je intuitivní a aplikace je kompatibilní s Androidem.

Serverová část zajišťuje komunikaci přes REST API, správu digitálních dvojčat a implementuje cache mechanismus pro mapové dlaždice, což optimalizuje výkon a snižuje náklady na volání Google Maps API.

Webové rozhraní umožňuje správu projektů včetně nahrávání modelů, úprav a mazání. Webové rozhraní obsahuje také přehled všech projektů, který obsahuje všechny důležité údaje včetně zpětné vazby.

Během vývoje byly upraveny některé požadavky. Ukládání do osobního repozitáře bylo nahrazeno přímým načítáním ze serveru, což eliminuje zastaralá data a zjednoduší architekturu. Skenování QR kódů nebylo implementováno, protože mapa opouští od nutnosti vylepovat QR kódy na reálná místa projektů. Detailní projektové informace (termíny, náklady, odpovědné firmy) nebyly zahrnuty, prototyp obsahuje obecný popis, který tyto informace může zahrnovat. V případě potřeby je datový model připraven na rozšíření. Otáčení modelu gestem nebylo implementováno, uživatel model prohlíží pohybem zařízení, což odpovídá přirozenému používání AR.

5. Závěr

Cílem této bakalářské práce bylo vyvinout funkční prototyp mobilní aplikace pro zobrazení digitálních dvojčat stavebních projektů v rozšířené realitě. Aplikace měla umožnit veřejnosti prohlížet plánované stavby v reálném prostředí a poskytovat zpětnou vazbu prostřednictvím hodnocení a komentářů.

Navržený systém využívá architekturu klient-server a skládá se ze tří hlavních komponent: mobilní aplikace implementované v prostředí Unity s využitím jazyka C#, serverového REST API vytvořeného ve frameworku Flask a webového rozhraní pro správu modelů digitálních dvojčat. Datová vrstva je realizována dokumentově orientovanou databází MongoDB, která umožňuje flexibilní ukládání metadat projektů, uživatelských hodnocení a cachovaných mapových dlaždic.

Mobilní aplikace umožňuje uživatelům vyhledávat dostupné projekty pomocí interaktivního mapového rozhraní, zobrazovat 3D modely ve formátu GLB v prostředí rozšířené reality a odesílat zpětnou vazbu formou hvězdičkového hodnocení a textových komentářů. Webové rozhraní poskytuje kompletní správu projektů včetně nahrávání, úpravy a mazání. Webové rozhraní také umožňuje procházet seznam jednotlivých projektů, kde je zobrazen jejich popis, průměrné hodnocení, název a náhledový obrázek, a zároveň poskytuje možnost zobrazení všech hodnocení a komentářů.

Hlavním technickým přínosem práce je implementace vlastního mapového systému včetně odvození matematického vztahu pro transformaci geografických souřadnic do souřadnicového systému v Unity. Tento přístup byl nutný kvůli absenci vhodných bezplatných mapových knihoven pro Unity a umožňuje přesné zobrazení projektů na mapě při různých úrovních přiblížení. Implementace cache systému pro mapové dlaždice s expiračním časem 10 dní výrazně snížila počet dotazů na Google Maps Static API a zrychlila načítání mapy.

Srovnání s existujícími řešeními (Trimble Connect AR, Dalux BIM Viewer, Augment) ukazuje, že vytvořený prototyp obsazuje dosud nevyužitou kombinaci. Jednoduché, veřejně přístupné řešení zaměřené na hodnocení v územním plánování. Profesionální nástroje nabízejí pokročilé funkce a vysokou přesnost, ale jsou finančně a technicky náročné a cílí na stavební firmy, nikoli na veřejnost.

Prototyp demonstruje technickou proveditelnost využití rozšířené reality pro vizualizaci digitálních dvojčat stavebních projektů. Budoucí vývoj by měl bezpochyby zahrnovat i autentizaci uživatelů v mobilní aplikaci, což by zamezilo nežádoucímu spamování komentářů a hodnocení. Dále by mohla být implementována autentizace oprávněných uživatelů ve webové aplikaci pro lepší kontrolu nad správou projektů a integrace analytických nástrojů pro spracování zpětné vazby.

Seznam použitých zdrojů

1. MARIN, G. G. *Digital Twin: An Emerging Tool for Urban Planning and Design* [online]. 2023. [cit. 2025-03-15]. ISSN 2069-6469. Dostupné z: <https://uac.incd.ro/Art/v15n1a1.pdf>.
2. WANG, Z. Digital Twin Technology. In: *Industry 4.0 - Impact on Intelligent Logistics and Manufacturing* [online]. IntechOpen, 2020 [cit. 2025-03-15]. Dostupné z DOI: [10.5772/intechopen.80974](https://doi.org/10.5772/intechopen.80974).
3. FABIO, Arena; COLLOTTA, Mario; PAU, Giovanni; TERMINE, Francesco. An Overview of Augmented Reality. *Computers* [online]. 2022 [cit. 2024-12-17]. Dostupné z DOI: [10.3390/computers11020028](https://doi.org/10.3390/computers11020028).
4. TREMOSA, Laia. *Beyond AR vs. VR: What is the Difference between AR vs. MR vs. VR vs. XR?* [online]. 2025. [cit. 2025-02-05]. Dostupné z: <https://www.interaction-design.org/literature/article/beyond-ar-vs-vr-what-is-the-difference-between-ar-vs-mr-vs-vr-vs-xr>.
5. SHARAPOV, Dmytro. *Marker-Based Augmented Reality - It-Jim* [online]. c2024. [cit. 2025-02-11]. Dostupné z: <https://www.it-jim.com/blog/augmented-reality-tracking-with-different-markers>.
6. IKEA. *Launch of new IKEA Place app – IKEA Global* [online]. 2017. [cit. 2024-12-27]. Dostupné z: <https://www.ikea.com/global/en/newsroom/innovation/ikea-launches-ikea-place-a-new-app-that-allows-people-to-virtually-place-furniture-in-their-home-170912/>.
7. IQLANDIA. *Geo - expozice / iQLANDIA.CZ* [online]. c2024. [cit. 2024-12-28]. Dostupné z: <https://iqlandia.cz/iqlandia/expozice/geo>.
8. GOOGLE. *Google Lens – vyhledávejte, co vidíte* [online]. [cit. 2024-12-28]. Dostupné z: <https://lens.google/intl/cs/>.
9. AUGMENT. *AR Viewer / Augment* [online]. [cit. 2025-03-18]. Dostupné z: <https://www.augment.com/blocks/ar-viewer/>.
10. BUILDINGPOINT NORTHEAST. *Trimble Connect AR - BuildingPoint Northeast — Your Trimble Distribution Partner from Maryland to Maine* [online]. [cit. 2025-03-17]. Dostupné z: <https://www.buildingpointne.com/field-technology/mixed-reality/trimble-connect-ar/>.
11. DALUX. *Nejrychlejší bezplatný prohlížeč BIM na světě* [online]. [cit. 2025-03-17]. Dostupné z: <https://www.dalux.com/cs/produkty/bim-viewer/>.

12. DALUX. *TwinBIM - Augmented Reality (AR) – Dalux HelpCenter* [online]. [cit. 2025-03-18]. Dostupné z: <https://support.dalux.com/hc/cs/articles/360015612953-TwinBIM-Roz%C5%A1%C3%AD%C5%99en%C3%A1-realita-AR>.
13. UNITY TECHNOLOGIES. *Unity Documentation*. c2025. Dostupné také z: <https://docs.unity.com/en-us>.
14. MICROSOFT. *C# docs - get started, tutorials, reference*. [online]. c2025. [cit. 2025-07-19]. Dostupné z: <https://learn.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/>.
15. PALLETS. *Welcome to Flask – Flask Documentation (3.1.x)* [online]. c2010. [cit. 2025-07-22]. Dostupné z: <https://flask.palletsprojects.com/en/stable/>.
16. DOCKER INC. *Docker Documentation* [online]. c2013-2025. [cit. 2025-07-25]. Dostupné z: <https://docs.docker.com/>.
17. MONGODB INC. *MongoDB Documentation - Homepage* [online]. c2025. [cit. 2025-07-29]. Dostupné z: <https://www.mongodb.com/docs/>.
18. KHRONOS GROUP. *gltF - Runtime 3D Asset Delivery* [online]. c2025. [cit. 2025-07-20]. Dostupné z: <https://www.khronos.org/gltf/>.

A. Externí přílohy

Zdrojové kódy této bakalářské práce jsou umístěny ve veřejném repozitáři na adrese:
<https://github.com/radeksmejky9/BcThesis>.

WebServer/	Adresář se serverovou částí práce
ARDigitalTwins/	Adresář s klientskou částí práce
Thesis/	Adresář s psanou částí práce
thesis.pdf	Text bakalářská práce
README.md	Návod pro nasazení
