



Zadání bakalářské práce

Název:	Věnná města českých královen – Webový klient pro schvalování a publikaci 3D modelů II
Student:	Tereza Opálková
Vedoucí:	Ing. Jiří Chludil
Studijní program:	Informatika
Obor / specializace:	Webové a softwarové inženýrství, zaměření Softwarové inženýrství
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	do konce letního semestru 2023/2024

Pokyny pro vypracování

1. Analyzujte aktuální stav projektu Věnná města českých královen a existující práce relevantní k tématu této práce.
2. Analyzujte požadavky na uživatelské rozhraní pro schvalovací proces 3D modelů. Analyzujte požadavky v rolích modelář, grafik, historik a administrátor.
3. Pomocí metod softwarového inženýrství navrhnete prototyp webové aplikace pro schvalování 3D modelů (proces nahrání modelů, postupného schvalování a publikace – označení modelu za veřejný). Přizpůsobte návrh webové technologie React.js.
4. Implementujte prototyp klienta dle návrhu a pomocí vybraných technologií.
5. Podrobte aplikaci uživatelským a akceptačním testům.

Bakalářská práce

**VĚNNÁ MĚSTA
ČESKÝCH KRÁLOVEN –
WEBOVÝ KLIENT PRO
SCHVALOVÁNÍ
A PUBLIKACI 3D
MODELŮ II**

Tereza Opálková

Fakulta informačních technologií
Katedra softwarového inženýrství
Vedoucí: Ing. Chludil Jiří
10. května 2023

České vysoké učení technické v Praze

Fakulta informačních technologií

© 2023 Tereza Opálková. Všechna práva vyhrazena.

Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí a nad rámec oprávnění uvedených v Prohlášení, je nezbytný souhlas autora.

Odkaz na tuto práci: Opálková Tereza. *Věnná města českých královen – Webový klient pro schvalování a publikaci 3D modelů II*. Bakalářská práce. České vysoké učení technické v Praze, Fakulta informačních technologií, 2023.

Obsah

Poděkování	viii
Prohlášení	ix
Abstrakt	x
Seznam zkratk	xii
Úvod	1
1 Cíle práce	3
2 Analýza	5
2.1 Popis a cíle projektu VMCK	5
2.2 Analýza Private API	6
2.2.1 Popis Private API	6
2.2.2 Funkcionality Private API	6
2.2.2.1 Struktury a 3D objekty	6
2.2.2.2 Schvalovací proces	7
2.2.2.3 Publikace modelů	9
2.2.3 Dodatečné informace	9
2.3 Analýza závěrečných prací	10
2.3.1 Backend administrační části – Ing. Dominik Sivák	10
2.3.1.1 Dílčí úkoly závěrečné práce	10
2.3.1.2 Analýza provedená v diplomové práci	10
2.3.1.3 Návrh provedený v diplomové práci	11
2.3.1.4 Realizace provedená v diplomové práci	15
2.3.1.5 Funkcionality neimplementované v diplomové práci	16
2.3.1.6 Použitelnost	16
2.3.2 Webová aplikace pro schvalovací proces 3D modelů – Bc. Pavel Antoš	17
2.3.2.1 Dílčí úkoly závěrečné práce	17
2.3.2.2 Návrh provedený v bakalářské práci	17
2.3.2.3 Implementace provedená v bakalářské práci	18
2.3.2.4 Aktuální stav aplikace	19
2.3.2.5 Použitelnost	21
2.3.3 Ostatní práce	21
2.3.3.1 Práce z roku 2020	21
2.3.3.2 Práce z roku 2021	22
2.4 Analýza softwarových týmových projektů	22
2.4.1 Týmový projekt 1	23
2.4.2 Týmový projekt 2	24
2.4.3 Použitelnost	24
2.5 Analýza webového průvodce	25
2.5.1 Stránky webového průvodce	25

2.5.2	Použitelnost	26
2.6	Analýza uživatelských požadavků	27
2.6.1	Uživatelské role	27
2.6.2	Uživatelské cíle	27
2.6.2.1	Obecné cíle	27
2.6.2.2	Modelář	28
2.6.2.3	Schvalovatel	28
2.6.2.4	Administrátor	29
2.7	Funkční a nefunkční požadavky	30
2.7.1	Funkční požadavky	30
2.7.2	Nefunkční požadavky	33
2.8	Analýza technologií	33
2.8.1	React JS	33
2.8.1.1	Popis	33
2.8.1.2	Výhody a nevýhody	33
2.8.2	React TS	34
2.8.2.1	Popis	34
2.8.2.2	Výhody a nevýhody	34
3	Návrh	35
3.1	Návrh schvalovacího procesu	35
3.1.1	Existující návrhy	35
3.1.2	Vlastní návrh schvalovacího procesu	36
3.2	Návrh obrazovek uživatelského rozhraní	39
3.2.1	Stránky společné všem uživatelům	39
3.2.2	Stránky modeláře	40
3.2.3	Stránky schvalovatele	41
3.2.4	Stránky administrátora	44
3.3	Pokrytí funkčních požadavků	45
4	Realizace	47
4.1	Technologie	47
4.1.1	React JavaScript VS React TypeScript	47
4.1.2	Create React App	47
4.1.3	EmailJS	48
4.1.4	React OAuth2 Google	49
4.1.4.1	Nastavení	49
4.1.4.2	Propojení s Private API	50
4.1.5	Liteprůvodce a IFrame	50
4.2	Private API	50
4.2.1	Problémy	50
4.3	Schéma komunikace aplikace	51
4.4	Stránky webové aplikace	52
4.4.1	Stránky společné všem uživatelům	52
4.4.2	Stránky modeláře	53
4.4.3	Stránky schvalovatele	55
4.4.4	Stránky administrátora	57
4.5	Nedokončené funkcionality	58
4.5.1	Živé zobrazení ve 3D scéně	58
4.5.2	Zobrazování všech formátů 3D objektů	58
4.5.3	Upozornění přes email	58
4.5.3.1	Problémy	59

4.5.3.2	Budoucí rozvoj	59
4.5.4	Úprava podpůrných souborů 3D objektu	59
4.5.5	Nekompletní tvorba 3D objektu	60
4.5.6	Stahování souboru s 3D modelem	60
4.5.7	Správa procesů a jejich uživatelů	60
4.5.8	Zabezpečení aplikace	61
4.5.9	Properties u 3D objektů	61
5	Testování	63
5.1	Testovací scénáře	63
5.1.1	Tvorba scénářů	63
5.1.2	Scénář 1 – Přihlášení a nahrání modelu	64
5.1.3	Scénář 2 – Nová verze a odeslání	65
5.1.4	Scénář 3 – Schválení iterace a zrušení rozhodnutí	65
5.2	Statistiky z testování	66
5.2.1	Testeři	66
5.2.2	Hodnocení aplikace	66
5.3	Výsledky testování	66
5.3.1	Překládání stavů	67
5.3.2	Upozornění o načítání aplikace	67
5.3.3	Zlepšení nápověd	67
5.3.4	Varování u nevratných akcí	68
5.3.5	Doplnění typu u nahrávání textur	68
5.3.6	3D scéna	68
5.3.7	Ikony	69
5.4	Postřehy z BI–TUR	69
5.4.1	Indikátory stavu	69
5.4.2	Stahování souborů pro schvalovatele	69
5.5	Akceptační testy	69
5.5.1	Průběh	70
5.5.2	Výsledek	70
6	Příručky	71
6.1	Instalační příručka	71
6.2	Uživatelská příručka	72
	Závěr	73
A	Další obrazovky aplikace	75
B	Další grafy z uživatelského testování	81
C	Emailové šablony	83

Seznam obrázků

2.1	Návrh datového modelu schvalovacího procesu – převzato z [6]	13
2.2	Návrh životního cyklu 3D objektu v závislosti na akcích schvalovacího procesu – převzato z [6]	14
2.3	Okno s 3D scénou v módu psaní komentářů se souřadnicemi v modelu – screenshot pořízen v aplikaci Pavla Antoše [11]	20
2.4	Kreslení do 3D modelu ve VR scéně – převzato z [14]	22
2.5	Tvorba poznámky pomocí překladu mluvené řeči na text – převzato z [14]	23
2.6	Stránka mapy a dialogové okno vybrané budovy – screenshot z webového průvodce	25
2.7	Stránka scény z webového průvodce – model Katedrála sv. Ducha, screenshot z webového průvodce	26
2.8	Use cases diagram – modelováno v Diagrams.net [20]	32
3.1	Diagram tvorby 3D modelu a odeslání ke schválení – modelováno v Diagrams.net [20]	37
3.2	Diagram procesu schvalování jednou autoritou – modelováno v Diagrams.net [20]	38
3.3	Návrh obrazovky s uloženými modely modeláře	41
3.4	Návrh obrazovky modeláře – formulář pro tvorbu nebo úpravu 3D objektu	42
3.5	Návrh obrazovky s modely ve schvalovacím procesu	43
3.6	Návrh obrazovky se stavem schvalovací iterace	43
4.1	Komunikační schéma aplikace a externích služeb – modelováno v Diagrams.net [20]	52
4.2	Obrazovka s přehledem modelů modeláře	53
4.3	Obrazovka s tvorbou 3D objektu	54
4.4	Obrazovka s přehledem úkolů schvalovatele	56
4.5	Obrazovka s formulářem pro hodnocení iterace	56
4.6	Obrazovka pro publikaci 3D objektu	57
5.1	Průběh uživatelského testování	64
5.2	Graf hodnocení testerů k bodu <i>Jednoduchost nalezení hledané stránky</i>	67
5.3	Graf hodnocení testerů k bodu <i>Výstižnost nadpisů a tlačítek</i>	67
5.4	Nápověda k tlačítkům u modelů modeláře	68
A.1	Úvodní obrazovka pro nepřihlášené uživatele	75
A.2	Úvodní obrazovka pro přihlášeného uživatele, zde modeláře	76
A.3	Obrazovka s detailem uloženého 3D objektu ve formě 3D zobrazení	76
A.4	Obrazovka s detailem uloženého 3D objektu ve formě formuláře	76
A.5	Obrazovka s tvorbou nové verze pro existující 3D objekt	77
A.6	Obrazovka s úpravou uloženého 3D objektu	77
A.7	Obrazovka se zobrazenými soubory uloženého 3D objektu	77
A.8	Obrazovka s formulářem pro odeslání 3D objektu ke schválení	78
A.9	Obrazovka s detailem iterace, která byla ohodnocena – zobrazena ve formátu pro schvalovatele	78
A.10	Obrazovka s přehledem veškerých schvalovacích procesů schvalovatele	78
A.11	Obrazovka s přehledem publikovaných iterací	79

B.1	Graf hodnocení testerů k bodu Kvalita nápovědy (otazníčků)	81
B.2	Graf hodnocení testerů k bodu Vzhled aplikace	82
B.3	Graf hodnocení testerů k bodu Rychlost odezvy aplikace	82
C.1	Šablona pro zaslání upozornění schvalovateli	84
C.2	Šablona pro zaslání upozornění modeláři	84

Seznam tabulek

3.1	Tabulka s kontrolou pokrytí funkčních požadavků F1 a F2	46
3.2	Tabulka s kontrolou pokrytí funkčních požadavků F3 a F4	46

Chtěla bych poděkovat především vedoucímu práce Ing. Jiřímu Chludilovi za skvělé vedení, ochotu a pomoc při tvorbě práce. Také chci poděkovat oponentovi práce, Ing. Petru Benešovi, za cenné rady nejen k implementaci v této práci.

Dále chci poděkovat své rodině a svému příteli za to, že po celou dobu studia a během psaní práce stáli po mém boku a byli mi neochvějnou oporou. Chci poděkovat i svým domácím mazlíčkům za jejich chlupatou a hebkou útěchu.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracovala samostatně a že jsem uvedla veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 2373 odst. 2 zákona č. 89/2012 Sb., občanský zákoník, ve znění pozdějších předpisů, tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené.

V Praze dne 10. května 2023

.....

Abstrakt

Tato bakalářská práce se věnuje návrhu a implementaci prototypu webové aplikace pro projekt Věnná města českých královen. Webová aplikace umožňuje členům projektu nahrávat 3D modely a podrobit je schvalovacímu procesu.

Součástí práce je analýza aktuálního stavu projektu Věnná města českých královen, přidružených aplikací a existujících závěrečných prací, které jsou tematicky podobné tématu této práce. U některých prací a aplikací byl nalezen podstatný potenciál navázat na jejich výstupy.

Aplikace implementovaná v této práci umožňuje uživatelům provést akce definované jejich projektovou rolí. Uživatelská role může nabývat hodnot modelář, historik, grafik nebo administrátor. Modelářům umožňuje nahrávat, verzovat a spravovat 3D modely a následně je odeslat ke schválení. Historikům a grafikům aplikace umožňuje schvalovat či zamítnat odeslané 3D modely, případně k nim přidávat poznámky. Administrátorům aplikace umožňuje vybírat mezi schválenými modely ty, jež jsou vhodné pro publikaci do veřejného prostoru. Implementace byla provedena pomocí technologie React TypeScript.

Aplikace byla podrobena uživatelským testům a na základě jejich výsledků byla upravena. Aplikace byla před odevzdáním podrobena akceptačním testům. Na závěr práce jsou uvedeny důležité příručky, instalační a uživatelská.

Klíčová slova webová aplikace, uživatelské rozhraní, Věnná města českých královen, schvalování 3D modelů, React, Typescript

Abstract

This bachelor thesis is focused on design and implementation of a prototype of a web application for the project Dowry Towns of the Queens of Bohemia. The application allows project members to upload 3D models and subject them to an approval process.

A part of the thesis is analysis of the current state of the project Dowry Towns of the Queens of Bohemia, associated applications and existing theses, which are thematically similar to this thesis. For some works and applications, there was found a potential to follow up on their outputs.

The application implemented in this work allows users to perform actions defined by their project role. This role can be either modeller, historian, graphician or administrator. For modellers, it allows them to upload, to verse and to manage 3D models and send them for approval. For historians and graphicians, it allows them to approve or decline 3D models and optionally add notes. For administrators, it allows them to choose among the approved models and pick the models suitable for being published for the public. The implementation was done using React TypeScript.

The application was subjected to user tests and was modified based on their results. Before submission, the application was subjected to acceptance tests. At the end of the work, there are installation and user manuals.

Keywords web application, user interface, Dowry Towns of the Queens of Bohemia, approval of 3D models, React, Typescript

Seznam zkratk

API	Application Programming Interface
CRUD	Create, Read, Update, Delete
HTML	Hypertext Markup Language
ID	identifikátor
JS	JavaScript
MVČ	Muzeum východních Čech v Hradci Králové
SPA	Single Page Application
TS	TypeScript
URL	Uniform Resource Locator
VMCK	Věnná města českých královen
VR	Virtuální realita

Úvod

Tato práce je součástí projektu Věnná města českých královen (zkráceně VMCK), což je projekt vyvíjený za spolupráce Fakulty informačních technologií ČVUT v Praze a Filosofické fakulty Univerzity Hradec Králové [1]. Jedním z hlavních témat projektu VMCK je vizualizace historických budov či místností pomocí tvorby grafických 3D modelů. Dále jedním z hlavních cílů projektu je zpřístupnit tyto modely široké veřejnosti.

Během postupného vývoje projektu byl vynesena závěr, že součástí procesu tvorby a zveřejnění 3D modelů by měla být kontrola kvality. Tato kvalita je určena tím, že jsou modely věrné předloze, jsou graficky kvalitně zpracovány a splňují další projektové podmínky, například zákaz plagiátorství. Cílem této kontroly je, aby se k široké veřejnosti dostaly pouze kvalitní 3D modely. Aby bylo možno tento cíl splnit, byl v projektu aplikován návrh procesu, ve kterém modely nejdříve procházejí postupným schvalováním, a jakmile jsou označeny za schválené, tak je uživatelům umožněno modely zveřejnit do veřejného prostoru.

Tato práce má za cíl umožnit členům projektu podrobit 3D modely procesu schvalování a zveřejnění pomocí webové aplikace. Ke splnění tohoto cíle bude v práci využito již existující struktury projektu, například API (aplikační programové rozhraní) projektu. Dále bude analyzován potenciál navázat na existující závěrečné práce, jež vznikly pod projektem VMCK, či na existující aplikace, jež jsou s projektem propojeny.

V rámci tohoto projektu již vzniklo několik závěrečných prací, jak bakalářských, tak diplomových. Některé z nich se zabývaly tím, jak umožnit členům projektu podrobit 3D modely tomuto schvalování. Tyto práce se zabývaly buďto tvorbou uživatelského rozhraní pro členy projektu, nebo tvorbou potřebných funkcionalit na backendu projektu. Příklad prací, jež se zabývaly tvorbou uživatelského rozhraní, jsou práce kolegy Michala Martínka – *Administrační rozhraní pro projekt Věnná města českých královen*, kolegy Marka Klofáče – *Věnná města českých královen – Schvalovací systém pro 3D modely ve virtuální realitě*, nebo kolegy Pavla Antoše – *Věnná města českých královen – Webová aplikace pro schvalovací proces 3D modelů*. Práce, jež se zabývaly rozvojem backendu projektu, jsou například práce kolegy Dominika Síváka – *Věnná města českých královen – Backend administrační části*, nebo kolegy Daniela Vančury – *Věnná města českých královen – jádro*. Tyto práce zde budou analyzovány, bude ohodnocena jejich relevance pro tuto práci a bude rozhodnuto, zda jejich výstupy bude možno použít v této práci.

Práce bude rozčleněna do jednotlivých částí. V části Analýza bude proveden rozbor aktuálního stavu projektu, bude analyzován potenciál navázat na existující práce a budou analyzována možná řešení problému. K části Návrh bude proveden návrh funkcionalit, vzhledu a tvorby prototypu aplikace. V kapitole Realizace dojde k implementaci řešení problému. V textu budou rozebrány důležité postupy, jež byly aplikovány během tvorby webové aplikace. V kapitole Testování bude aplikace podrobena uživatelským a akceptačním testům dle postupů softwarového inženýrství. Programátorská příručka provede uživatele instalací, konfigurací a spuštěním aplikace. Uživatelská příručka stručně provede čtenáře standardními uživatelskými akcemi v aplikaci.



Kapitola 1

Cíle práce

Prvním cílem je analyzovat aktuální stav Private API projektu Věnná města českých královen. Bude zde zjištěno, jaké jsou aktuálně implementované a plánované funkcionality, jež se týkají tématu této práce. Dále bude zjištěno a otestováno, jaké jsou aktuálně funkční API endpointy, které budou potřeba k implementaci prototypu aplikace.

Druhým cílem je zjistit, jaké existují závěrečné práce a jiné zdroje, jež se věnují tématům podobným, jako je téma této práce. Z těchto zdrojů budou získány dodatečné informace o Private API projektu VMCK, dále budou analyzovány cíle, jež v pracích byly splněny a jak by práce mohly být přínosné pro tuto bakalářskou práci.

Třetím cílem je získat informace o požadavcích uživatelů na uživatelské rozhraní webové aplikace, která v této práci bude implementována. V aplikaci budou zohledněny uživatelské role modelář, historik, grafik a administrátor. Pro každou z těchto rolí bude provedena analýza jejich předpokládaných aktivit a požadavků.

Čtvrtým cílem je provést návrh prototypu webové aplikace. Při návrhu budou zohledněny analyzované požadavky uživatelů v různých rolích a ostatní informace získané během analýzy. Součástí návrhu budou nové požadavky Private API, pokud vyvstane potřeba úprav, aby mohla být aplikace vyvinuta.

Pátým a hlavním cílem je provést implementaci prototypu webové aplikace v návaznosti na analýzu a návrh. Implementace bude provedena pomocí technologií určených v zadání a v analýze. Aplikace bude implementována s cílem být jednoduchá a uživatelsky přívětivá.

Šestým cílem je provést uživatelské a akceptační testy prototypu webové aplikace. Testované atributy budou funkcionality, náročnost používání a zda je aplikace intuitivní.

Kapitola 2

Analýza

V této kapitole bude nejdříve provedena analýza současného stavu projektu Věnná města českých královen a budou zde analyzovány další zdroje informací přidružené k projektu.

Nejdříve zde bude stručně uveden popis projektu VMCK a jeho hlavní cíle. Dále zde bude analyzováno Private API projektu VMCK a jeho aktuální funkcionality, společně s některými funkcionalitami navrženými, rozpracovanými a plánovanými.

Následovat bude analýza vybraných bakalářských, diplomových a týmových prací, které byly součástí projektu VMCK rozvíjely ho. Pozornost bude věnována projektům, které jsou nejnovější a aktuální a které se tematicky týkají procesu schvalování 3D modelů – to značí proces tvorby a nahrání 3D modelu modelářem, jeho postupné schvalování schvalovacími autoritami a označení schváleného modelu za veřejný. Analyzovány budou i některé další aplikace, které jsou součástí projektu VMCK a mohly by být pro tuto práci přínosné.

Následně budou analyzovány uživatelské požadavky na aplikaci, která bude v této práci vznikat.

Všechna témata budou rozebrána pouze v takovém rozsahu, aby byl obsah relevantní k tématu této práce.

2.1 Popis a cíle projektu VMCK

Projekt Věnná města českých královen vznikl za spolupráce Fakulty informačních technologií ČVUT v Praze, Filosofické fakulty Univerzity Hradec Králové, Historického ústavu Akademie věd České republiky a MVČ (Muzeum východních Čech v Hradci Králové). Jedná se o dlouholetý projekt, jehož cílem je zvyšování povědomí o českém historickém fenoménu, o královských věnných městech. Jedním z nástrojů používaných v projektu je počítačová grafika, pomocí níž jsou vytvářeny 3D modely historických budov.

Dlouhodobým cílem projektu je vytvořit mobilní aplikaci a webový portál, jež budou sloužit jako virtuální průvodci po českých věnných městech a budou v nich promítány právě 3D modely vytvořené pomocí pokročilé počítačové grafiky [1]. Některé ze zveřejněných výsledků projektu si lze prohlédnout na stránkách projektu [1] nebo v online průvodci [2]. Aplikace dostupná na stránkách průvodce bude podrobněji rozebrána v kapitole 2.5.

Část projektu, jíž je tato práce součástí, je vyvíjena na Fakultě informačních technologií ČVUT v Praze. Důležitými vedoucími členy projektu na fakultě jsou Ing. Jiří Chludil a Ing. Petr Pauš, Ph.D. Tato část projektu se týká mimo jiné správy 3D modelů, jejich ukládání, manipulace, práce s podpůrnými daty k modelům a podobně.

2.2 Analýza Private API

Tato sekce se bude týkat analýzy současného stavu Private API pro projekt VMCK. Sekce bude obsahovat analýzu současných funkcionalit a funkčních endpointů Private API, ale také některých funkcionalit rozpracovaných nebo pouze navržených.

Dokumentace k Private API existovala již dříve v editoru Swagger a poslední oficiální verzi lze najít na adrese <http://109.123.202.213:3003/> [3]. Nejnovější varianta vznikla jako součást diplomové práce Ing. Dominika Siváka a je zveřejněna na adrese <https://app.swaggerhub.com/apis/sivakdom/private-api/3.0.5#/> [4]. Dokumentace aktuálně není udržovaná, některé části API se však od její tvorby nezměnily [5]. Z toho důvodu bude v této práci část dokumentace aktivně používána – po ověření její správnosti a funkce – a bude důležitým zdrojem informací pro analýzu Private API.

2.2.1 Popis Private API

Private API pro projekt VMCK poskytuje projektu možnost správy 3D modelů a všech k tomu potřebných entit. API je aktuálně dostupné na adrese <https://vmck.sic.cz:3443/> a je chráněno přístupovým tokenem. Umožňuje klasické CRUD operace (create – vytvořit, read – číst, update – upravit, delete – smazat) nad řadou entit, jež jsou potřeba ke správě 3D modelů. Také obsahuje několik „akčních“ endpointů, které umožňují například provést jedním voláním předdefinované akce nebo uvést do provozu nějaké procesy. Příklad předdefinovaného procesu jsou například API endpointy vytvořené pro schvalovací procesy, jež provedou posloupnost akcí v závislosti na zaslaném požadavku [4]. Příklad volání, jež aktivuje proces, je teprve plánovaná aktivace poloautomatických transformací na 3D model [5].

2.2.2 Funkcionality Private API

V této sekci budou uvedeny hlavní funkcionality, které Private API poskytuje. Budou zmíněny funkcionality implementované, navržené a plánované, vše pouze v rámci témat týkajících se této práce. Budou zde vysvětleny i některé pojmy používané na Private API a v projektu obecně. Pro získání informací o API endpointech bude používána hlavně dokumentace [4] a práce kolegy Siváka [6], v jisté míře také konzultace s vedoucím [5], který je členem vývojového týmu backendu a Private API projektu.

2.2.2.1 Struktury a 3D objekty

Private API umožňuje pomocí endpointů spravovat některá data projektu. Umožňuje standardní CRUD operace nad entitami v databázi. Nad velkou částí entit v databázi je umožněna jednoduchá editace pomocí aplikace, jež vznikla v týmových projektech v předmětech Softwarový týmový projekt 1 a 2 v týmu, jehož jsem byla členem. Důležité entity, jejichž existence je stěžejní pro tuto práci, zde budou uvedeny. Dále u nich budou uvedeny některé jejich důležité vlastnosti.

Struktury jsou objekty, které reprezentují reálný objekt a zadání pro tvorbu 3D modelů. Jedná se tedy o souhrn informací například o reálné budově nebo místnosti, pro něž v budoucnu budou vznikat jejich 3D modely. Entita struktury v sobě nese informace jako například název, slovní popis, GPS polohu, město, data postavení a zničení, užitečné odkazy a také eventuálně své 3D modely [4], [5]. Je v plánu implementovat u struktur koncept fotografie či obrázku pro lepší specifikaci zadání.

Pro tuto aplikaci jsou důležité například tyto API endpointy:

- GET /structures poskytuje kompletní seznam všech uložených struktur.

- `POST /structures/{structureId}` poskytuje detail jedné vybrané struktury.
- `POST /structures/{structureId}/3DObjects` umožňuje nahrát ke struktuře první verzi 3D objektu. Objekt vznikne jako verze 1.0.0. [4]

3D objekty jsou reprezentací 3D modelů konkrétních struktur, jež v sobě nese různá metadata k modelu. Nesou v sobě například své jméno, identifikátor struktury, ke které patří, objekty s transformacemi nebo indikátory, zda je model schválen schvalujícími autoritami. Důležité z pohledu počítačové grafiky jsou taktéž textury nebo assety.

U 3D objektů je umožněno verzování, tudíž tvůrce modelu může ke svému modelu vytvořit novou verzi. Objekty jsou ukládány pomocí spojového seznamu, který začíná první verzí a končí nejnovější verzí. Pokud takovéto verzování existuje, tak v sobě objekty nesou odkazy na předchozí či následující verzi. K následné manipulaci, jako čtení, úpravy či aktivaci procesů, je zpřístupněna pouze nejnovější verze [4], [6].

Pro tuto aplikaci jsou důležité například tyto API endpointy získané z dokumentace [4]:

- `GET /3DObjects` poskytuje kompletní seznam všech uložených 3D objektů. Tento endpoint umožňuje volitelnou filtraci podle tvůrce.
- `GET /3DObjects/{3DObjectId}` poskytuje detail vybraného 3D objektu.
- `PUT /3DObjects/{3DObjectId}/{version}` umožňuje upravit vybranou verzi 3D objektu.

Modely reprezentují samotné stáhnutelné soubory s 3D modely, ukládané bez textur a s informací o formátu [4].

Textury a assety jsou entity obsahující podpůrné soubory pro 3D modely. Ukládají se do databáze samostatně a v 3D objektech jsou zapsány odkazy na tyto soubory, společně se základními informacemi o nich. Přidání souboru se provádí pomocí API volání, po němž bude soubor uložen k vybranému 3D objektu. [4]

2.2.2.2 Schvalovací proces

Aktuálně na API existuje základ implementace funkcionalit pro schvalovací proces pro 3D modely. Tento proces však ještě není kompletně zpřístupněn uživatelům v rámci UI (uživatelského rozhraní). Vznikly pouze prototypy, které nepokryly všechny části procesu – více v kapitole o analýze starších prací (2.3). Jediný způsob, jak používat funkcionality implementace schvalovacího procesu, je skrz jednoduché nástroje pro provádění HTTP požadavků a přímých volání na Private API. Takový nástroj je například Postman [7]. Tento způsob je vysoce nepraktický a je reálný pouze pro uživatele, kteří mají hlubokou znalost Private API projektu VMCK, jeho dokumentace a jeho backendu. Právě proto, že proces není zatím kompletně zpřístupněn uživatelům, je cílem této práce takovou aplikaci navrhnout a implementovat.

Návrhu a implementaci tohoto procesu a potřebných funkcionalit na backendu se věnoval ve své diplomové práci nejvíce kolega Sivák [6], dříve také ve svých pracích kolegové Vančura [8] a Martínek [9]. Jejich práce budou analyzovány ve vlastní kapitole věnované existujícím pracím (2.3).

API endpointy k tomuto procesu jsou téměř kompletně zdokumentovány v editoru Swagger a zveřejněny v dokumentaci na adresách zmíněných v popisu API. Tuto dokumentaci společně se svou diplomovou prací upravil a zveřejnil kolega Sivák. Tato dokumentace nebyla od té doby aktualizována a značná část z ní tedy není aktuální, proces schvalování se však od tvorby kolegovy verze dokumentace téměř nezměnil [5]. V kontextu schvalovacích procesů bude tedy dokumentace využívána, po ověření funkce a konzultace s vedoucím, zda jsou význam a použití stále stejné.

Zde je uveden seznam některých důležitých entit z procesu schvalování [4] a jejich krátký popis [5], [6].

ApprovalProcess – v překladu Schvalovací proces – je entita reprezentující schvalovací proces. Váže se na jednu strukturu, tedy na jednu budovu a na jedno zadání. Obsahuje všechna odevzdání 3D modelů od všech tvůrců, jež pro vybranou strukturu nějaký model odevzdali. Jedno odevzdání 3D objektu se nazývá iterace. Schvalovací proces obsahuje také množinu členů, vybraných z uživatelů registrovaných v projektu, a informaci o jejich rolích. V procesech jsou uloženy i další dodatečné informace, které lze získat, pokud se dotážeme na správných endpointech.

Dle aktuálního návrhu se schvalovací proces dělí na dvě části. V první části, v iteracích modelu, je tvořen samotný 3D model objektu. V druhé části, v iteracích variant modelu, má probíhat transformace modelů z první části procesu. Jedná se například o aplikování vlivů počasí či stáří na předtím schválené modely.

ApprovalIteration – ve zkráceném překladu Iterace – se týká jednoho odevzdání 3D modelu ke schválení v první části procesu. V první části schvalovacího procesu je tvořen samotný 3D model modelářem a je schvalován dvěma autoritami, historikem a grafikem. Základní myšlenka cyklu je, že modelář – autor 3D objektu – odešle verzi 3D modelu ke schválení a schvalovatelé – historik a grafik – model buďto schválí, nebo vrátí k přepracování. Schvalující authority mají za úkol ověřit, zda je model věrný reálné předloze a zadání a zda je model správně proveden z pohledu počítačové grafiky.

ApprovalVariant – ve zkráceném překladu Varianta – označuje jedno odevzdání ve druhé fázi procesu. Zde mají být dle aktuálního návrhu na schválené modely aplikovány poloautomatické transformace, které by měly probíhat na serveru a jejich aktivace by měla být provedena pomocí volání API. Jedná se například o aplikace efektů počasí nebo ročního období na již schválený model. Variantu modelu je možno odevzdat i tak, že je modelář požádán o ruční aplikaci efektu a 3D model odevzdává podobně jako v první části schvalovacího procesu.

Tato funkcionální je na backendu implementována tak, jak je zde popsáno. V průběhu vývoje se však změnil návrh variant – nyní se jako varianta 3D objektu bere nový 3D objekt, který bude mít určené „properties“, neboli vlastnosti. Jedná se například o dobu postavení nebo efekt počasí.

Zde jsou uvedeny některé důležité endpointy pro schvalovací procesy a jejich iterace. Iterace variant nebudou rozebírány, jelikož není požadováno, aby je aplikace explicitně oddělovala od variant modelu.

- GET /approvals poskytuje seznam všech vytvořených schvalovacích procesů. Automaticky zde probíhá filtrace, která zajistí, že uživateli se vrátí pouze ty procesy, jichž je členem.
- GET /approvals/{approvalId} poskytuje detail vybraného schvalovacího procesu. V entitě jsou i záznamy o iteracích a případně i variantách v tomto schvalovacím procesu.
- POST /approvals/{structureId} vytvoří nový schvalovací proces pro vybranou strukturu. Na backendu je povolena existence maximálně jednoho schvalovacího procesu pro jednu strukturu.
- POST /approvals/{approvalId}/users/{approvalRole}/{userId} umožňuje přidat uživatele do procesu. Uživatel získá v procesu roli určenou v požadavku.
- POST /approvals/{approvalId}/iterations/model umožňuje vytvořit iteraci schvalovacího procesu a odeslat ke schválení vybraný 3D model.
- POST /approvals/iterations/model/{modelIterationId}/approval umožňuje schválit vybranou iteraci. Po úspěšném provedení bude iterace schválena rolí, jež má volající uživatel. Podobným způsobem je možné odeslat zamítnutí iterace nebo finalizaci iterace.
- DELETE /approvals/iterations/model/{modelIterationId}/approval umožní vzít zpět schválení vybrané iterace. Podobným způsobem je možné vzít zpět zamítnutí iterace nebo finalizaci iterace. [4], [6]

Členy je možno přidávat do libovolného ApprovalProcess. Při přidávání člena do schvalovacího procesu musí uživatel dostat přidělenou roli v procesu – výběr je z rolí modelář, grafik a historik. Podle této role má uživatel v procesu rozdílná práva a rozdílné úkoly. Modelář má za úkol splnit zadání a vytvořit vhodný model. Historik zde musí kontrolovat odevzdané modely a hodnotit, zda jsou dostatečně věrné originální předloze a historickému kontextu, do nějž budova patří. Úkol uživatele v roli grafik je u odevzdaných modelů kontrolovat, zda jsou dostatečně kvalitní z pohledu počítačové grafiky. [5]

Zde jsou uvedeny některé API endpointy, jež se věnují členům procesu a jsou důležité pro tuto aplikaci:

- `POST /approvals/{approvalId}/users/{approvalRole}/{userId}` umožňuje přidat uživatele do procesu. Uživatel získá v procesu roli určenou v požadavku. [4]

ApprovalComment je entita, která se váže k iteraci v libovolné fázi procesu (tvorba či transformace). Obsahuje atributy **type** (typ entity vybraný jako komentář) a **content** (obsah komentáře, který musí být typu **type**).

Zde jsou uvedeny některé API endpointy, jež jsou důležité pro správu komentářů:

- `GET /comments/{commentableType}/{commentableId}` umožňuje získat veškeré komentáře pro vybranou entitu. V kontextu této práce bude `commentableType` typicky „model“, což symbolizuje iteraci schvalovacího procesu.
- `POST /comments/{commentableType}/{commentableId}` umožňuje přidat komentář k vybrané entitě, v této práci opět typicky k iteraci schvalovacího procesu. [4], [6]

2.2.2.3 Publikace modelů

Proces zveřejňování modelů, jež byly schváleny autoritami, byl navržen a implementován jako součást schvalovacího procesu v práci kolegy Siváka [6]. Proces je navržen tak, že jakmile je iterace, tudíž i obsažený 3D objekt, schválena, tak může být finalizována. Iterace takto získá atribut **finalizedDate**, v překladu „datum finalizace“. 3D objekt změní svůj stav z **approved** na **finalized**. Zároveň je možné tuto akci vzít zpět, stejně jako u schvalování a zamítání iterací.

Tuto akci má právo provádět pouze uživatel s rolí administrátor. Jedná se o finální kontrolu modelu, kde administrátor rozhodne, zda je vhodné model publikovat na veřejné API. Těmto uživatelům je umožněno libovolně finalizovat schválené iterace a případně brát zpět toto rozhodnutí.

Zde jsou uvedeny některé důležité API endpointy získané z dokumentace [6]:

- `POST /approvals/iterations/model/{modelIterationId}/finalization` umožňuje finalizovat iteraci a tím i 3D objekt v ní obsažený. Volání je umožněno pouze administrátorům.
- `DELETE /approvals/iterations/model/{modelIterationId}/finalization` umožňuje negovat finalizaci iterace. Tím se iterace a 3D objekt stanou opět pouze schválenými. Volání je opět umožněno pouze administrátorům.

2.2.3 Dodatečné informace

Na vývoji Private API se podílelo hned několik studentů. Práce dokončené v roce 2020 a později jsou uvedeny v kapitolách níže.

Hlavní část API, která bude potřeba v této práci, vyvinul kolega Sivák, který položil základ backendovým operacím pro proces schvalování 3D modelů. Jeho práci zde bude věnováno více prostoru. V případě potřeby budou v této práci navrženy změny API či backendu.

Pokud bude zjištěn nedostatek, který přímo či nepřímo brání implementaci či správné funkci vyvíjené aplikace, budou nové požadavky zdokumentovány a předány aktuálním vývojářům backendu projektu. Pro tento účel byl vybrán ticketovací systém ¹ Redmine.

2.3 Analýza závěrečných prací

V této kapitole bude provedena analýza bakalářských a diplomových prací relevantních k tématům, jichž se bude týkat tato práce. Práce obsahující informace, které jsou pro tuto práci relevantnější, zde dostanou více prostoru než práce, které jsou neaktuální nebo tematicky vzdálenější.

2.3.1 Backend administrační části – Ing. Dominik Sivák

V roce 2021 vznikly tři závěrečné práce rozvíjející projekt Věnná města českých královen. Jednou z nich je tato, která rozvíjí backend projektu.

Diplomová práce se zabývá návrhem a implementací schvalovacího procesu v projektu VMCK na backendu. Proces umožňuje na backendu spravovat entity schvalovacích procesů (entita **ApprovalProcess** a entity, které se jí týkají). Cílem je umožnit pomocí API volání tvořit schvalovací procesy, spravovat jejich obsah, a ve výsledku umožnit na backendu provádět proces schvalování. Zde uživatelé v různých rolích pracují na schvalování 3D modelů, a to pomocí zamítání, komentování a schvalování. Na konci procesu je možno publikovat schválené modely.

Ing. Dominik Sivák rozvíjel a opravoval již existující základ procesu na backendu projektu. Navrhl a implementoval opravy a nové části. Na konci práce poskytl pro další využití procesu dokumentaci Private API VMCK (odkazy byly zmíněny v kapitole 2.2) a programátorskou příručku.

Práce kolegy Siváka zde dostane, společně s prací kolegy Antoše, nejvíce prostoru, jelikož nejlépe ze všech prací reprezentuje stav API (aktuální v době jejího psaní a z části i v době psaní této práce).

2.3.1.1 Dílčí úkoly závěrečné práce

V této sekci je uveden stručný výčet úkolů, které byly autorovi práce uděleny a které byly v práci splněny. Jsou zde vynechány úkoly málo relevantní k zadání této práce.

1. Analyzovat projekt VMCK, backend projektu a současné funkcionality a nedostatky.
2. Navrhnout nové funkční požadavky a potřebné úpravy starého návrhu a implementace.
3. Implementovat změny z návrhu.
4. Otestovat implementaci. [6]

2.3.1.2 Analýza provedená v diplomové práci

V této sekci je proveden rozbor kapitoly Analýza z práce Ing. Siváka. Kapitola neobsahuje kompletní rozbor všech částí, ale pouze informace z částí, které jsou relevantní pro téma této práce. Stav funkčních požadavků, jako *Splněno při převzetí projektu*, jsou uvedeny tak, jak je analyzoval a navrhl kolega ve své práci.

¹prostředek pro systematizaci úkolů [10]

Analýza starých a návrh nových funkčních požadavků

F1: Management 3D objektů včetně modelů a textur: API umožní uživateli kompletní CRUD operace (Create, Read, Update, Delete) nad entitami v databázi.

Splněno při převzetí projektu.

F1.1: Verzování 3D objektů: API umožní prohlížet, upravovat a mazat verze 3D objektů.

Splněno při převzetí projektu.

F1.2: Ukládání 3D objektů v různých podobách: API musí umožnit ukládat 3D objekty v různých podobách, jako za použití různých textur nebo různých úrovní detailu.

Splněno při převzetí projektu.

F2: Správa uživatelů: Uživatelům bude umožněno projít registrací, po níž musí administrátor API účet schválit. Podle úrovně přístupu získá nový uživatel přístupová práva.

Nesplněno při převzetí projektu: registrace uživatelů není implementovaná.

F2.1: Přístupová práva a uživatelské role: Registrovaný uživatel dostane přiděleny role a přístupová práva. Správa obého je umožněna pouze administrátorovi.

Nesplněno při převzetí projektu: uživatel může mít maximálně jednu roli.

F5: Schvalování modelů: (*upraveno*) Nad API bude postaven schvalovací proces 3D modelů. Proces bude složen z iterací, kde ke každé iteraci náleží jeden 3D objekt. Iterace musí být schváleny autoritami *grafik* a *historik*. V první fázi, *model*, jsou potřeba souhlasy obou autorit, v druhé fázi, *variant*, pouze grafika.

F5.1: Správa uživatelů ve schvalovacím procesu: (*nové*) Pro proces bude možné spravovat uživatele, kteří budou jako členové mít přístup k iteracím a komentářům. Uživatel může být členem v různých rolích a tím získat různé pravomoce v procesu.

F5.2: Komentáře ve schvalovacím procesu: (*nové*) API umožní členům k iteracím připojovat komentáře za účelem poskytnutí zpětné vazby. V této práci jsou zohledněny komentáře textové nebo textové s přiloženými souřadnicemi v modelu.

F5.3: Statistiky schvalovacího procesu: (*nové*) U každého procesu bude možné získat jeho statistiky.

F6: Alternativní přihlášení: (*nové*) Uživatelům bude umožněno přihlásit se pomocí účtu třetích stran, konkrétně Google. Pokud se přihlásí uživatel s novým účtem, bude registrován. [6]

2.3.1.3 Návrh provedený v diplomové práci

V této sekci je proveden rozbor kapitoly Návrh z práce kolegy Siváka. Kapitola, stejně jako předchozí, obsahuje pouze informace z částí, které jsou pro tuto práci relevantní. V návrhu jsou řešeny tehdejší nedostatky a návrhy jejich oprav.

Nový návrh schvalovacího procesu byl v práci proveden, aby nahradil nedostatečný návrh tehdejší. Dle nových požadavků v sekci Analýza byl proveden nový návrh procesu, aby se proces stal přehlednějším a aby byla vylepšena jeho funkčnost. V práci kolegy Siváka byla část funkčních požadavků, která se týká schvalovacího procesu, implementována v celém rozsahu tak, jak byla navržena.

Byla navržena nová entita, **ApprovalProcess**, která v sobě drží mimo jiné i vazbu na schvalovaný objekt a na iterace procesu, **ApprovalIteration**. Přehled datového návrhu struktur okolo schvalovacího procesu je znázorněn v obrázku 2.1. Obrázek je přímo převzat z práce kolegy Siváka [6].

První fáze procesu obsahuje iterace, jejichž obsahem jsou modely postupně vytvářeny modelářem a následné hodnocení grafikem a historikem. Schvalovací autority iteraci mohou schválit a nebo zamítnout. K oběmu je umožněno přidat poznámky. Výsledkem schvalování bude 3D model, jenž bude historicky věrný reálné předloze a bude vhodně zpracován z pohledu počítačové grafiky.

Druhá fáze procesu má za cíl vytvořit a schválit různé *varianty* modelu, jako např. model po aplikaci efektů počasí. Takovou variantu zastřešuje jedna **ApprovalVariant**, v níž je opět několik **ApprovalIteration**, každá s jedním modelem. Entita také obsahuje informace o poloautomatických transformacích (transformace na serveru), pokud byly použity. Pokud je grafik nespokojen s vygenerovaným výsledkem, může modeláře požádat o ruční aplikaci efektu. Nebo může modeláře pořádat o ruční zpracování rovnou. V obou případech bude transformovaný model nedostupný až do doby, kdy bude dokončen modelářem nebo serverem, tzn. nepůjde sledovat postup transformace v reálném čase. V této části schvalování je vyžadována kontrola pouze grafikem. [6]

Statistiky jsou zde navrženy jako způsob sledování údajů v procesu. Jde například o délku trvání automatické transformace nebo zátěž vyvinutá na server při takové transformaci. Pro tuto statistiku vznikla nová entita **ApprovalAction**, která má vazbu na entitu **ApprovalProcess**. V rozsahu diplomové práce nebylo tyto funkcionality implementovat, ale připravit backend na případná rozšíření. Statistiky pro tuto bakalářskou práci nejsou podstatné, nebudou proto dále rozebírány.

Správa uživatelů zde byla navržena v upravené formě. Vznikal návrh způsobu, jak spravovat účastníky ve schvalovacím procesu a jejich přístupová práva. Na databázové úrovni byla navržena nová vazební tabulka, pro vztah M:N (many to many) mezi **User** a **ApprovalProcess**. Tabulka bude obsahovat i roli, ve které User v procesu vystupuje.

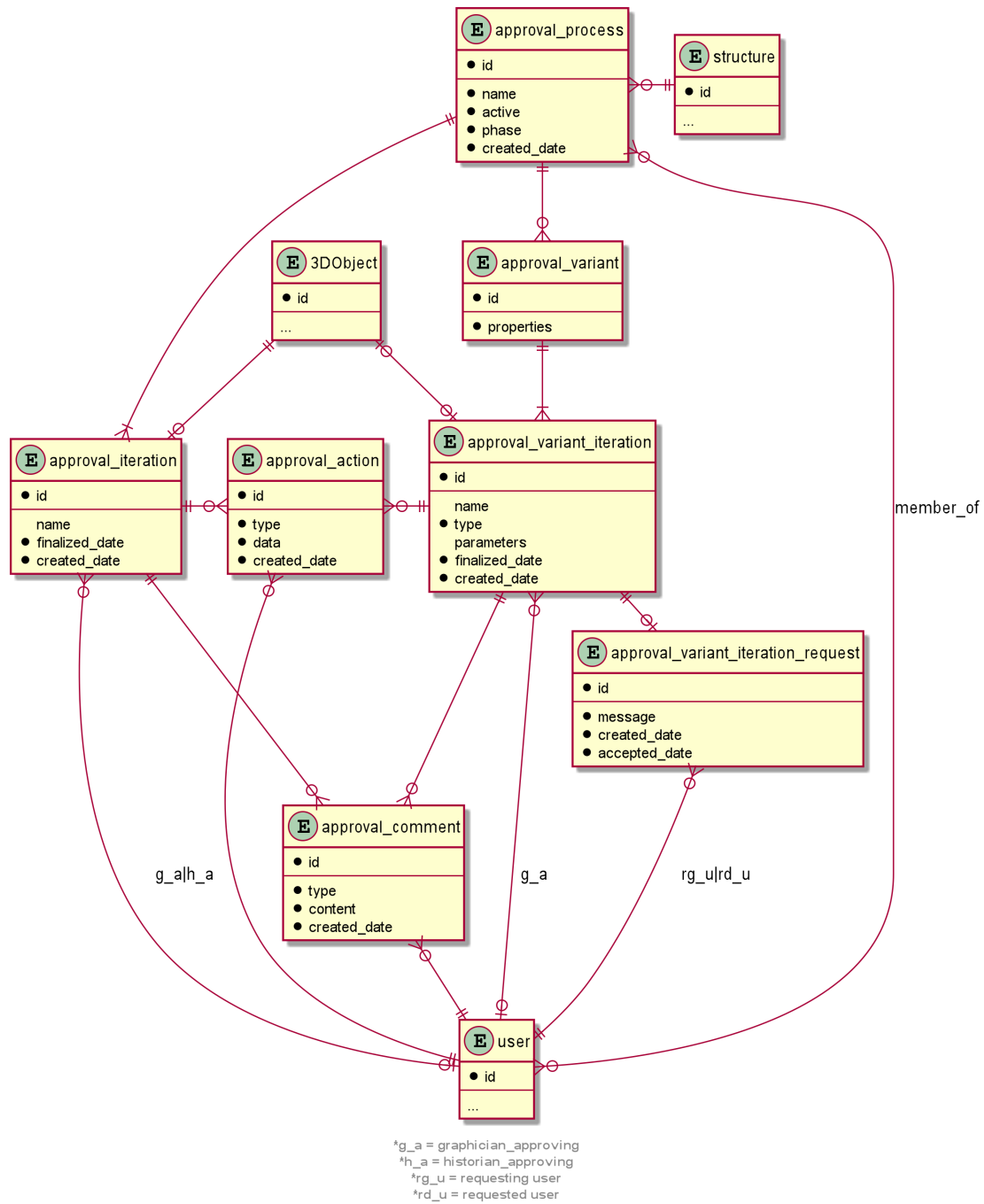
Dále byla navržena nová třída, jejíž úkolem bylo kontrolovat přístup uživatelů. Přístup zkontroluje na základě vstupu, což je **uživatel** (subjekt), **akce** (predikát), kterou chce provést a **záznam** entity, na které chce akci vykonat (objekt).

Komentáře byly původně pouze textové a proto byly označeny za nedostatečné. Proto vznikla nová entita **ApprovalComment**, který má vazbu na **ApprovalIteration**, atribut **type** (určující typ komentáře) a **content**, u něž se předpokládá objekt typu type.

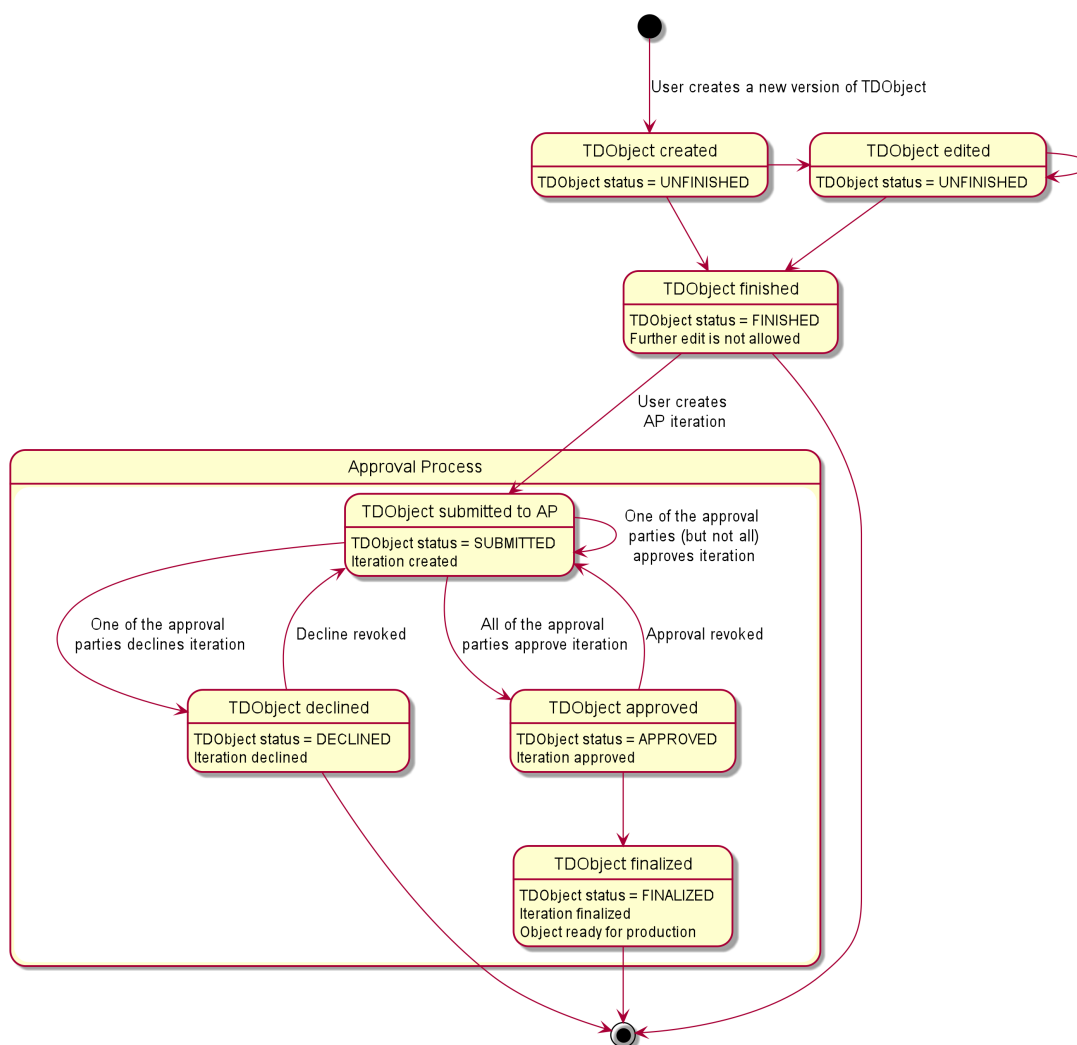
Úpravy 3D objektů byly označeny jako nutné změny pro další rozvoj implementace procesu. Autor v uvedl, že pro lepší orientaci v procesech a iteracích je potřeba změnit množinu stavů, jež entita TDOject nabývá. Důvodem bylo, že v tehdejší návrhu se entita pohybovala mezi stavy UNFINISHED (nedokončený), SUBMITTED (odeslaný), DECLINED (zamítnutý), FIXED (opravený) a APPROVED (schválený). Obzvláště hodnota FIXED byla problematická, jelikož jakmile byl nahrán nový opravený model, ztratily se informace o původním zamítnutém modelu. Také například u komentářů, v nichž byly explicitně zmíněny souřadnice ve 3D modelu, představuje FIXED stav problém – 3D objekt se změní, ale souřadnice v komentáři zůstanou stejné a jsou dále nerelevantní. Také bylo rozhodnuto, že informace o stavu schvalování se přesunou do iterace, místo dosavadního 3D objektu. Níže jsou uvedeny důležité změny implementace 3D objektů.

Entita TDOject byla osvobozena od zodpovědnosti za držení stavu schvalování. Původně v entitě byly informace, zda je model schválen jednotlivými autoritami. Tato odpovědnost se přesouvá do jednotlivých iterací procesu a ve 3D objektu se stav schvalování promítá pouze jako změny stavu.

Množina stavů entity TDOject byla proto upravena. Nový návrh stavů byl upraven tak, aby odpovídal novému návrhu procesu, kde v entitě **ApprovalProcess** figurují jednotlivé entity



■ Obrázek 2.1 Návrh datového modelu schvalovacího procesu – převzato z [6]



■ **Obrázek 2.2** Návrh životního cyklu 3D objektu v závislosti na akcích schvalovacího procesu – převzato z [6]

ApprovalIteration a **ApprovalVariant**. V novém návrhu totiž proces obsahuje posloupnosti iterací, kde každá iterace spravuje pouze jeden model, který bude schválen nebo zamítnut. Níže jsou uvedeny nové stavy a jejich význam dle nového návrhu. Životní cyklus objektu je zobrazen v obrázku 2.2) [6].

UNFINISHED objekt – v překladu „nedokončený“ – je takový, který je stále ve fázi tvorby modelářem. Modely jsou ukládány tak dlouho, než je autor spokojen. Je také umožněno je editovat nebo mazat, primárně autorovi, ale také administrátorovi.

FINISHED – v překladu „dokončený“ – je následující stav objektu, který autor označil za dokončený a který již nelze upravovat. Může to být stav finální, pokud si autor nepřeje posílat model do schvalovacího procesu. Autorovi a administrátorovi je umožněno tento model smazat, a pouze autor může model odeslat ke schválení.

SUBMITTED – v překladu „odeslaný“ – označuje objekt, jenž byl odeslán ke schválení jako předmět iterace. Tento objekt ještě nebyl ohodnocen – ani schválen, ani zamítnut.

DECLINED – v překladu „zamítnutý“ – je stav po tom, co byl objekt v rámci iterace zamítnut jednou z autorit. Tudíž stačí, aby buď historik nebo grafik zamítli vybranou iteraci.

APPROVED – v překladu „schválený“ – je objekt, který byl schválen všemi schvalovacími autoritami. Počet autorit záleží na tom, o kterou fázi se jedná – v první fázi, fázi modelu, musí model schválit grafik i historik. Ve druhé fázi pak pouze grafik.

FINALIZED – v překladu „finalizovaný“ – je stav objektu, který úspěšně prošel schvalovacím procesem a byl finalizován administrátorem. Tento objekt byl vybrán jako verze modelu, která bude v budoucnu umístěna na veřejné API. [6]

Úpravy API endpointů zúčastněných na schvalovacím procesu byly provedeny jako nutná součást změny návrhu zmíněného výše. Kompletní výčet endpointů tak, jak je kolega zanechal, je uveden v dokumentaci [4].

Způsob přihlašování uživatelů byl v práci pozměněn. Bylo rozhodnuto, že do aplikace bude možno se přihlašovat a registrovat pomocí účtu třetí strany a že touto OAuth autoritou bude Google. Pro účel přihlašování vznikl nový **Controller**, který přijímá požadavky na přihlášení. Tyto požadavky budou zasílány frontendovou klientskou aplikací, od které se očekává, že bude využívat kódy a komponenty poskytnuté přímo Google.

Po přihlášení získá webová aplikace od Google pro jednotlivého uživatele dočasný ověřovací token, který pošle backendu, kde bude token ověřen. Tím bude uživatel ověřen nebo mu bude přístup odepřen. Pokud se pomocí Google přihlášení pokusí přihlásit uživatel, jehož účet v databázi uživatelů ještě není, bude mu vytvořen nový profil (jeho data budou uložena do databáze).

2.3.1.4 Realizace provedená v diplomové práci

V této sekci bude proveden rozbor kapitoly Realizace z práce kolegy Siváka. Opět jsou zde probrány pouze části, které jsou relevantní k tématům mého projektu. Pro stručný rozbor důležitých vzniklých API endpointů vizte kapitolu, kde je rozebíráno Private API projektu (2.2.2). Pro detailní rozbor vizte dokumentaci [4] nebo práci kolegy Siváka [6].

Implementace schvalovacího procesu byla provedena tak, jak byla navržena u kolegy Siváka v kapitole Návrh (2.3.1.3). Vnikly nové endpointy a na některých existujících byly provedeny potřebné úpravy. K oběmu vznikly nové nebo upravené integrační testy.

Dokumentace na portálu Swagger vznikla v nové verzi jako bonusová součást práce. Z důvodu cenové politiky byla vytvořena kopie specifikace, která byla upravena, místo modifikace existujícího souboru. Sivákova osobní kopie je dostupná na adrese <https://app.swaggerhub.com/apis/sivakdom/private-api/3.0.5#/>. V jeho programátorské příručce je vysvětlen postup, jak vytvořit novou verzi dokumentace, jak Swagger používat v lokálním prostředí nebo jak lokální instanci testovat.

Správa procesů je umožněna pomocí standardních endpointů na CRUD operace (create, read, update, delete). V dokumentaci jsou umístěny pod kapitolou **approval process**. Specifičtější funkcionality, které se týkají například iterací či členů, jsou uvedeny níže.

Správa uživatelů ve schvalovacím procesu byla také realizována tak, jak byla navržena. Nad uživateli ve schvalovacím procesu byly implementovány základní CRUD operace. V dokumentaci jsou informace umístěny pod kapitolou **approval process members**.

Správa komentářů pro procesy a jejich iterace je také umožněna přes klasické endpointy ke CRUD operacím. V dokumentaci je lze najít pod **approval process comments**.

Správa iterací modelů v procesu má taktéž svou sadu endpointů na CRUD operace. Iterace modelu se týkají se iterací v **první fázi**, kde je schvalován 3D model samotný, a historik a grafik kontrolují jeho historickou věrnost a grafickou povedenost. V dokumentaci jsou pod kapitolou **approval process model iterations**.

Správa variant modelů ve schvalovacích procesech má také vlastní endpointy. Jedná se o iterace v **druhé fázi**, kde je schvalována nová varianta již schváleného modelu - například po aplikaci poloautomatických transformací nebo po aplikování efektů, které mohl provést server nebo modelář ručně. Jedná se například o efekty počasí, jako déšť a sníh, nebo jiné podmínky, jako například bláto. Endpointy lze v dokumentaci najít pod kapitolou **approval process variants**.

Správa jednotlivých iterací variant modelů má taktéž vlastní endpointy a v dokumentaci je nalezneme pod kapitolou **approval process variant iterations**.

Implementace autentizace přes Google autoritu byla v návrhu označena jako nejlogičtější možnost, jak uživatele registrovat a přihlašovat.

V programátorské příručce kolegy je uveden zkrácený postup, jak použít Google přihlášení v klientské aplikaci. Pro detailní rozbor jsme odkázáni na samotný Google, přes odkaz <https://developers.google.com/identity/sign-in/web/sign-in>. Kolega ve své práci uvedl stručnou příručku, jak zavést Google autentizaci do frontendové aplikace. Zde bude jeho rozbor vynechán, jelikož detailnější příručka bude uvedena ve vlastní sekci 4.1 v kapitole **Realizace**.

Obecný postup se řídí Google příručkou, dostupnou na adrese <https://developers.google.com/identity/sign-in/web/backend-auth>.

2.3.1.5 Funkcionality neimplementované v diplomové práci

Zde bude proveden výčet funkcionalit, které v práci byly navrženy, ale nebyly implementovány, nebo je kolega při tvorbě práce poznamenal jakožto nedostatky. Kolega je navrhuje jako další vhodná vylepšení ve své kapitole **Návrhy do budoucna**. Budou zde uvedeny pouze ty funkcionality, jejichž neexistence se může týkat této práce.

Zavedení pokročilejší autorizace uživatelů soudí kolega jako bezpečnostní krok. Nyní je jeho autorizační token každého uživatele v DB od vytvoření stále stejný a API token je pouze jeden globální.

Zavedení správy rolí uživatelů by bylo vhodné přehodnotit. Aktuálně backend umožňuje každému uživateli mít pouze jednu roli. Zároveň kolega považuje kontrolu přístupu při volání endpointů za nedostatečnou.

Systém notifikace by byl vhodným nástrojem pro oznámení změn v procesu zúčastněným uživatelům. Jednalo by se o oznámení nových iterací, schválení nebo zamítnutí a podobně. Vhodným prvním krokem by bylo oznámení emailem. Doporučuje využití eventů a eventbus, jejichž příklad uvádí v souboru `src/Utils/ApprovalEventBus.ts`.

2.3.1.6 Použitelnost

Využití výstupu této práce je naprosto stěžejní pro implementaci v této práci. V práci byl implementován backend, který bude bezpochyby používán v aplikaci, která bude implementována

jako součást této práce. I z tohoto důvodu je jasnou volbou, že pro návrh v této práci bude využíván návrh procesu a entit provedený v této práci.

Zároveň důležitým výstupem této práce je dokumentace [4], která bude stěžejním zdrojem informací pro návrh i implementaci v této práci.

2.3.2 Webová aplikace pro schvalovací proces 3D modelů – Bc. Pavel Antoš

Tato práce je jednou ze závěrečných prací z roku 2021, které rozvíjely projekt Věnná města českých královen. Tato bakalářská práce měla za úkol zpřístupnit 3D modely v projektu VMCK jeho uživatelům pomocí webového uživatelského rozhraní a zároveň umožnit jejich schvalování. Hlavním cílem bylo zjednodušit uživatelům prohlížení 3D objektů a eliminovat potřebu používání speciálních nástrojů pro 3D zobrazení. Motivací bylo také ušetřit uživatelům manuální importování 3D modelů do těchto nástrojů. [11]

U této práce bude analyzován potenciál navázat na ni či využít a přepracovat některé její části, jelikož některé části zadání jsou podobné zadání této práce.

2.3.2.1 Dílčí úkoly závěrečné práce

V této sekci je uveden stručný výčet úkolů, které byly autorovi práce uděleny a které byly v práci splněny.

1. Analyzovat projekt proces schvalování 3D modelů z práce Michala Martínka a Daniela Vančury.
2. Analyzovat Private API projektu VMCK.
3. Analyzovat a vybrat vhodné technologie na zobrazení 3D modelů ve webovém prostředí.
4. Navrhnout prototyp webové aplikace.
5. Implementovat prototyp webové aplikace pomocí vybraných technologií.
6. Podrobit prototyp uživatelským testům. [11]

2.3.2.2 Návrh provedený v bakalářské práci

V této sekci je proveden stručný rozbor kapitoly Návrh z práce kolegy Antoše. Návrh zde není pro tuto práci tak stěžejní, jako návrh backendu kolegy Siváka, rozbor bude proto kratší.

Nový návrh schvalovacího procesu byl proveden v návaznosti na práci kolegy Michala Martínka, ve které on i kolega Sivák viděli nedostatky v návrhu bodů procesu. V tomto novém návrhu se opět počítá s tím, že proces vytváření zadání Historikem je proveden v externím procesu a Modelář přichází do procesu s již připraveným zadáním. Tento návrh se v několika bodech neshoduje s návrhem od kolegy Siváka, pravděpodobně z toho důvodu, že práce vznikaly paralelně a kolegové neměli potřebu se v tomto shodovat. Jejich práce na sebe totiž nejsou navázané.

Byly zde navrženy pozměněné stavy modelů.

UNFINISHED objekt – v překladu „nedokončený“ – je takový, který je stále ve fázi tvorby modelářem. Modely jsou ukládány tak dlouho, než je autor spokojen. Význam je zde stejný jako u kolegy Siváka.

FINISHED – v překladu „dokončený“ – je stav objektu, který byl autorem dokončen a který již nelze upravovat. Význam je zde stejný jako u kolegy Siváka.

SUBMITTED – v překladu „odeslaný“ – označuje objekt, jež byl odeslán ke schválení jako předmět iterace. Význam také souhlasí s kolegou Sivákem.

DISAPPROVED – v překladu „neschválený“ – je stav, který odpovídá stavu DECLINED v práci kolegy Siváka. Tento objekt byl zamítnut některou z autorit.

FIXED – v překladu „opravený“ – je stav, který se kolega Sivák snažil eliminovat. Zde označuje model, který byl modelářem opraven a lze jej opět poslat ke schválení, čímž se znova stane SUBMITTED. Stav FIXED existuje proto, aby bylo možné odlišit model, který je schvalován poprvé a model, který byl opraven a cyklem prochází znova.

APPROVED – v překladu „schválený“ – je objekt, který byl schválen všemi autoritami. Opět souhlasí se stavem u kolegy Siváka.

FINALIZED – v překladu „finalizovaný“ – je stav objektu, který úspěšně prošel schvalovacím procesem a byl finalizován administrátorem. Tento objekt byl vybrán jako verze modelu, která bude v budoucnu umístěna na veřejné API. [11]

Návrh komunikace aplikace byl proveden jako optimalizace komunikace s Private API VMCK. Byl zde navržen jednoduchý backend, který má za úkol zpracovávat komunikaci s Private API. Samotný webový klient tedy komunikuje pouze s Google API (které zpracovává část autentizace uživatelů) a se zmíněným malým backendem.

Návrh React komponent byl proveden jako příprava na implementaci webové aplikace. Byla zde popsána stromová struktura React komponent s kořenem *App*. Z veřejného React prostředí je zde využita knihovna *React Router*, která poskytuje komponenty potřebné k navigaci mezi stránkami aplikace. Z těchto komponent jsou využity *BrowserRouter* a *Switch*. Ty nám umožní navigovat se na stránky podle jejich URL. Pro využívání Google přihlášení jsou využity hotové komponenty poskytnuté přímo knihovnami od Google, a to *GoogleLogin* a *GoogleLogout*. Pro podrobný rozbor vizte práci kolegy Antoše [11].

Návrh uživatelského rozhraní měl za cíl vyřešit, jak aplikace efektivně umožní grafikům a historikům dávat zpětnou vazbu při schvalování a zamítání modelů. Do návrhu byla zahrnuta i myšlenka na to, že schvalovatelé si budou chtít zobrazit předchozí komentáře u modelu, například aby si mohli zkontrolovat, zda nepřidávají připomínku, která již byla vynesena.

Kolega se věnuje nejlepším možnostem, jak umístit jednotlivé ovládací prvky a co v nich zobrazit. Pro označení připomínky navrhl přidat ke komentáři 3D souřadnici v modelu, kdy se připomínka zobrazí na modelu jako barevná koule na povrchu modelu. Pro umístění historie připomínek vybral umístění mimo scénu, aby se udržela co nejlepší přehlednost stránky a 3D modelu.

2.3.2.3 Implementace provedená v bakalářské práci

V této sekci je proveden stručný rozbor kapitoly Implementace z práce kolegy Antoše. Kapitola obsahuje pouze informace z částí, které jsou pro tuto práci relevantní.

Backend je napsán v Node.js frameworku Express.js. Úkolem tohoto malého backendu je zprostředkovat komunikaci mezi Private API VMCK a webovou aplikací. Webová aplikace se musí při každém volání prokázat API tokenem a tento backend token ověřuje a následně komunikuje s Private API projektu. Webová aplikace bude provádět API volání na adresu `url_backend/...` místo volání url samotného Private API projektu VMCK.

Webová aplikace byla podle návrhu napsána ve frameworku React.js. Podle návrhu platí i hierarchie komponent. Autor zde rozebírá mimo jiné i rozmístění a podmíněné vykreslování komponent. Pro detailní rozbor komponent a jejich implementaci vizte práci kolegy Antoše [11].

Manipulace s 3D modelem byla implementována v jazyku TypeScript a frameworku Babylon.js. Pro import 3D modelu se je použit endpoint `/models/{modelId}`, který posílá data ve formátu octet-stream. Autor zde popisuje, jaké datové formáty zpracovává a jaké volání aplikace musí provést.

Autor také popisuje, jak implementoval různé módy zobrazení 3D modelů. Jedná se například o módy **wireframe**, **edge rendering** a jiné.

Autor dále popisuje, jak implementoval umístění prostorové poznámky do modelu. U 3D modelu je umožněno umístit na povrch bod v podobě koule. K tomuto bodu je umožněno přidat textovou poznámku a společně je odeslat jako prostorovou připomínku k modelu. Jelikož však Private API nepodporovalo odesílání souřadnicových komentářů, odesílání a načítání komentářů k modelu je zde simulované. Při obnovení stránky se komentáře nenávratně ztratí.

Uživatelská příručka slouží jako orientace pro uživatele, kde je poučen o všech dostupných funkcionalitách aplikace. Zde se jedná hlavně o stránky, které aplikace poskytuje. Zde jsou uvedeny jen části nejvíce relevantní pro tuto práci.

- **Přihlášení** je uživateli umožněno pomocí tlačítka v horním rohu obrazovky. Toto tlačítko vyvolá Google vyskakovací okno. Jakmile je přihlášení ověřeno pomocí Google authority, aplikace zkontroluje, zda je uživatel registrován v aplikaci. Pokud ne, zobrazí se mu stránka s registrací do aplikace – pomocí emailu a username, kde oba údaje jsou získány z jeho Google účtu. Po zmáčknutí tlačítka Register je registrován v aplikaci. Přihlašování a odhlašování je dále umožněno pomocí tlačítek Login a Logout – přihlásit a odhlásit.

- **Prohlídka 3D modelu** je umožněna po výběru struktury ze seznamu na hlavní stránce. Na úvodní stránce jsou zobrazeny karty se základními informacemi o struktuře (pro význam pojmu „struktura“ vizte kapitolu 2.2.2). Karty umožňují přejít pomocí kliknutí na detail struktury a jejích 3D objektů.

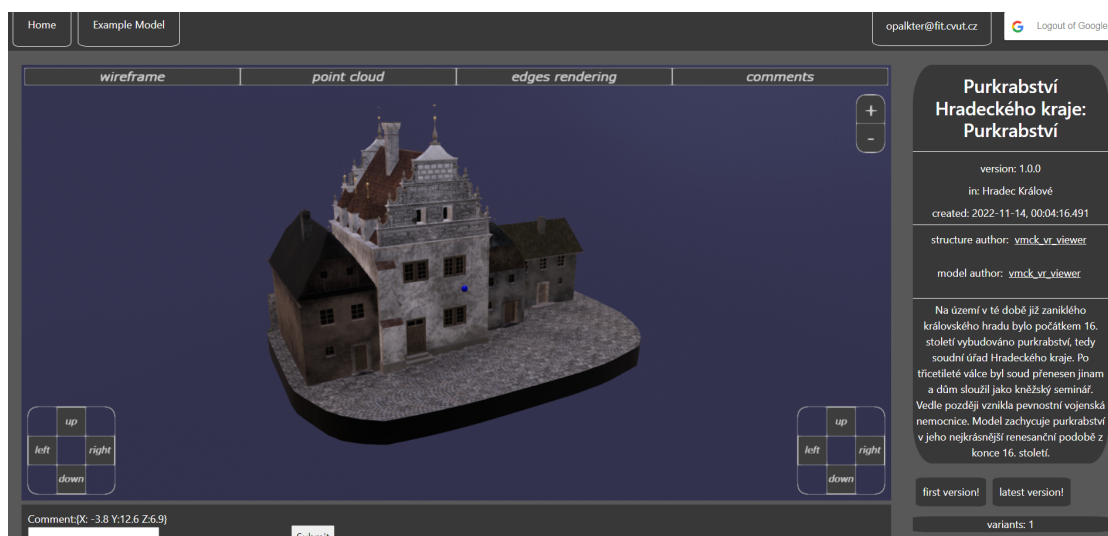
Po výběru struktury se uživateli zobrazí stránka s 3D scénou a blokem s informacemi o vybraném 3D objektu struktury (pro význam pojmu „3D objekt“ vizte kapitolu 2.2.2). Jako výchozí model se uživateli zobrazí první dostupná varianta z 3D objektů struktury. Všechny varianty modelu jsou dostupné v rozbalovacím menu **Variants** a lze mezi 3D modely přepínat. Lze také iterovat mezi modely podle data vytvoření, a to pomocí tlačítek **Next version** a **Previous version**. V okně s 3D modelem se lze pohybovat a otáčet.

Ve scéně lze také aplikovat různé módy zobrazení 3D modelů. Jedním z módů jsou **Comments**. To je speciální zobrazení, kde jsou zobrazeny textové komentáře a i komentáře vytvořené v 3D scéně.

- **Tvorba připomínek** je umožněna ve dvou formách - obyčejný text a připomínka se souřadnicemi. Oba druhy se zadávají ve formuláři pod scénou s 3D modelem. Obyčejná textová připomínka se vytváří tím, že uživatel zadá komentář, použije tlačítko Submit a opustí mód s komentáři. Komentář se souřadnicemi lze vytvořit tak, že uživatel kliknutím na 3D model umístí bod v 3D prostoru, do formuláře zadá text a odešle pomocí Submit. Mód psaní komentářů se souřadnicemi je ukázán na obrázku 2.3.

2.3.2.4 Aktuální stav aplikace

V této sekci bude popsáno, jak byla spouštěna a testována webová aplikace v zimním semestru roku 2022/23. Budou zde uvedeny problémy, které by bylo nutno vyřešit, pokud by padlo roz-



■ **Obrázek 2.3** Okno s 3D scénou v módu psaní komentářů se souřadnicemi v modelu – screenshot pořízen v aplikaci Pavla Antoše [11]

hodnutí tuto práci dále rozvíjet. Také zde budou uvedeny změny, které autor provedl v době po odevzdání práce.

Aplikace v současném stavu byla spouštěla a testována poprvé po téměř dvou letech od obhájení práce, proto nebylo překvapující, že její stav nebyl ideální. Po komunikaci s autorem byly odhaleny jisté problémy. Jak webový klient, tak osobní backend byly ukládány do vlastních privátních repozitářů ve fakultní GIT doméně, <https://gitlab.fit.cvut.cz>. Od autora byla získána přístupová práva k těmto private repozitářům. [12]

Kvůli vzniku nové verze Gitlab přestaly na projektech fungovat pipelines a tím pádem i docker images jak webového klienta, tak osobního backendu. Autor doporučil naklonovat si a spouštět pouze kód z větve DEV, který osobní backend vůbec nepoužívá a jeho chování je údajně téměř stejné, jako tehdejší aplikace (detaily nebyly uvedeny). Po tom, co proběhl první kontakt s autorem, se kolega Antoš ke své práci vrátil a lehce ji upravil, aby byla spustitelná alespoň aplikace na DEV větvi. [12]

Dále nastaly změny v Node.js – ve verzi Node.js v17 byly opraveny bezpečnostní díry týkající se SSL poskytovatelů. Tato oprava však rozbila programy, které používaly SSL ve starší Node.js verzi. To byl případ i zde. První možností bylo provést downgrade na starší Node.js v16, to by ale znamenalo vědomé používání nebezpečného kódu. Dalším řešením bylo ručně aktualizovat použité balíčky pomocí příkazu `npm audit fix --force` [13]. To umožnilo aplikaci spustit.

Testování aplikace, prováděné při zkoumání chyb výše a při organizovaném procházení stránkami, odhalilo další problémy s aplikací.

První očividný problém byl odhalen při prvním zkoumání 3D modelů. Problémem bylo, že aplikace nepodporovala všechny formáty 3D objektů, které byly aktuálně uloženy na Private API. Jednalo se například o formát FBX. Při testování se spolehlivě zobrazovaly formáty GLB a OBJ. Po komunikaci s autorem byla získána informace, že původní aplikace podporovala formáty FBX a OBJ, u nichž manipulaci zpracovával Babylon.js plugin **.obj File Loader Plugin**. Když kolega Antoš po kontaktování dělal rychlé úpravy, aby byla aplikace spustitelná, udělal rozhodnutí ohledně načítání formátů. Kolega místo původního pluginu **.obj File Loader Plugin** vytvořil parametrický loader objektů a přidal pomocí něj do aplikace plugin **GLTFFileLoader**. Výraz **gltf** je zkratka pro **Graphics Language Transmission Format**, což je formát, který dokáže

mimo .gltf soubory pracovat i s .glb soubory, které se v projektu VMCK vyskytují. Tím aplikaci umožnil zpracovat OBJ a GLB formáty. [12]

Další problémy se týkaly zobrazování a funkce stránky s 3D modelem (stránky s detailem struktury). Okno se scénou správně nepodporovalo změnu rozměrů okna prohlížeče. Jakékoliv změny velikosti okna způsobily, že se scéna rozbíjela – jednalo se například o otevírání developer módu pomocí F12 nebo změny rozměrů okna prohlížeče (například zúžení). Při takovéto změně rozměrů okna prohlížeče přestávala správně fungovat registrace myši – reálná pozice neodpovídala tomu, kde ji registrovala scéna. Klikání na tlačítka bylo v této situaci náročné.

Dalším problémem bylo, že když byla stránka s modelem načtena (např. pomocí F5) ve chvíli, kdy stránka v prohlížeči nebyla ve full-screen módu, tak se celá scéna špatně naškálovala (tlačítka byla mnohem větší) a také nastal problém s registrací myši, jako bylo zmíněno výše. Jednalo se například o situaci, kdy v prohlížeči bylo otevřeno developer okno a poté byla stránka načtena.

Drobná chyba byla, že po přepnutí variant modelů přestala fungovat levá skupina navigačních tlačítek a pohyb ve scéně pomocí pravého tlačítka myši. Šlo o pohyb s modelem horizontálně a vertikálně.

2.3.2.5 Použitelnost

Použitelnost výstupu kolegovy práce pro návrh a implementaci v této práci se jeví jako střední. Vznikla zde aplikace, která se soustředila hlavně na práci s nástroji pro 3D zobrazování a samotný proces schvalování byl upozaděn. Důvodem je pravděpodobně hlavně to, že implementace backendu ke schvalovacím procesům probíhala paralelně s implementací práce kolegy Antoše, její spolehlivé využití tedy nebylo příliš proveditelné.

Největším potencionálním přínosem práce kolegy Antoše pro tuto práci je systém na import a zobrazování 3D modelů. Bylo by však potřeba extrahovat jej z existující aplikace a využít ve vlastní aplikaci, jelikož přímo navázat na existující aplikace se nejeví jako efektivní přístup.

2.3.3 Ostatní práce

V této kapitole budou krátce uvedeny práce, které jsou již starší a neaktuální nebo méně relevantní pro tuto práci. Práce jsou uvedeny pro úplnost a nebudou hluboce rozvedeny.

2.3.3.1 Práce z roku 2020

Zde jsou uvedeny práce, které byly napsány v roce 2020 a jejich podoba již dnes není aktuální vůči stavu projektu VMCK.

Jádro private API - Bc. Daniel Vančura Tato bakalářská práce [8] vyvíjela novou část Private API pro projekt VMCK. Byla zprovozněna datová úložiště a propojeny četné moduly do jednoho uceleného komunikačního API.

Tato práce již není aktuální z důvodu, že na Private API od jejího napsání proběhly četné změny z rukou studentů i vedoucích projektu.

Administrační rozhraní pro projekt Věnná města českých královen - Ing. Michal Martínek Tato diplomová práce [9] měla za úkol vytvořit administrační rozhraní pro projekt VMCK - vytvořit webovou aplikaci pro správu dat na backendu. Webová aplikace vznikla, ale z důvodu nedostatečně dodělaného API jsou volání na API pouze simulovaná. Aplikace umožňovala CRUD operace nad simulovaným API, filtrování výsledků, zobrazování 3D struktur, přihlašování uživatelů a přiřazení schvalovacích autorit, vše však nad simulovanými API voláními.

Tato práce již není aktuální z důvodu podobného jako předchozí práce - API se od jejího odevzdání značně rozvinulo a změnilo, a místo úpravy tohoto projektu již vnikly projekty nové,



■ **Obrázek 2.4** Kreslení do 3D modelu ve VR scéně – převzato z [14]

které zastávají jeho práci a pracují s aktuální verzí API (vizte kapitoly Týmové projekty - 2.4.1 a 2.4.2).

2.3.3.2 Práce z roku 2021

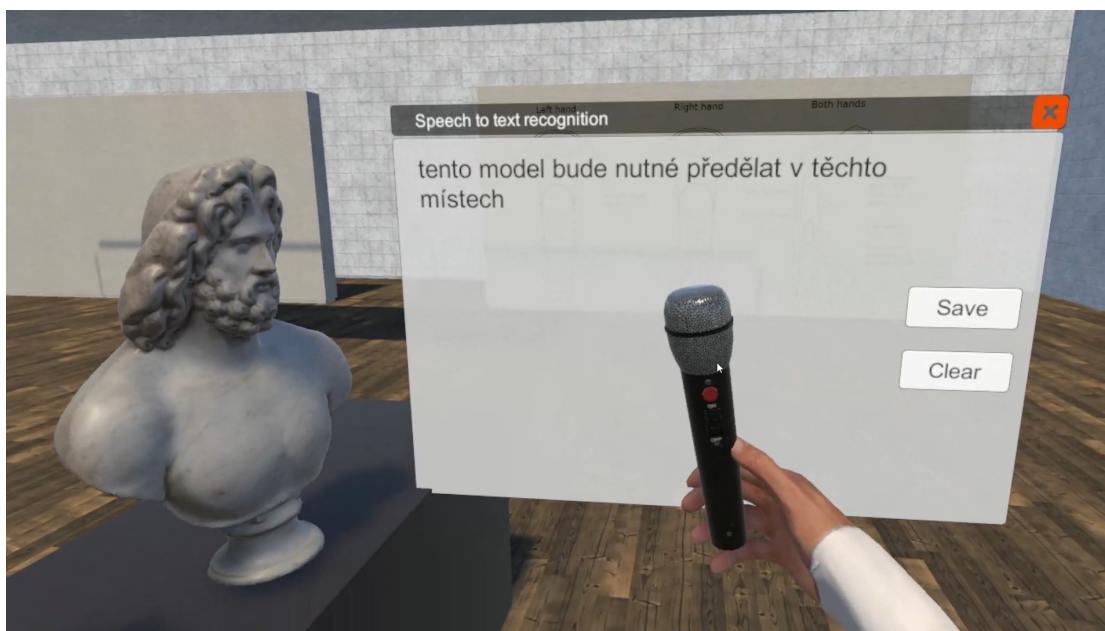
Níže uvedená práce se sice týkala schvalovacího procesu, ale je zaměřená více na studenty počítačové grafiky, což byl kolega Klofáč a což autor této práce není. Nejsou zde tedy ambice tuto práci dále rozvíjet, práce je zde však uvedena hlavně kvůli grafickým ukázkám.

Schvalovací systém pro 3D modely ve virtuální realitě - Bc. Marek Klofáč Tato práce [14] měla za úkol zobrazit 3D modely ve virtuální realitě a umožnit si je prohlížet a přidávat připomínky. Tato práce je zde uvedena hlavně z důvodu, že celý VMCK projekt se zaměřuje i na virtuální realitu a konkrétně Ing. Petr Pauš Ph.D. se i aktivně věnuje vývoji VR funkcionalit v projektu. Také protože nápady malovat na modely a vytvářet poznámky mluveným slovem byly mezi pracemi originální.

V této práci vznikla VR aplikace, kde bylo možné se přihlásit, a pomocí rozhraní od kolegy Antoše umožnit uživateli si vybrat strukturu a model. Prostředí umožnilo do 3D modelů přidávat textové a hlasové poznámky, a také na 3D model kreslit. Obě tyto připomínky jsou zobrazené na obrázcích 2.4 a 2.5.

2.4 Analýza softwarových týmových projektů

V této kapitole bude provedena stručná analýza práce týmových projektů z předmětů SP1 a SP2 (Softwarový týmový projekt 1 a 2) a které vyvíjely webového klienta ke správě dat v Private API projektu VMCK. Jsou zde uvedeny pro lepší orientaci v existujících aplikacích pro projekt.



■ **Obrázek 2.5** Tvorba poznámky pomocí překladi mluvené řeči na text – převzato z [14]

2.4.1 Týmový projekt 1

V rámci předmětu Softwarový týmový projekt 1 se vyvíjela webová aplikace, která živě komunikuje s Private API VMCK a umožňuje nad ním provádět základní CRUD operace nad vybranými entitami.

Proběhla analýza, návrh a implementace prototypu webového klienta. K implementaci byl použit JavaScript framework Vue 3 a CSS knihovna Bootstrap. Proběhly tři iterace odevzdání – analýza, návrh a implementace. Implementace tedy dostala jen třetinu času. Prototyp aplikace byl tudíž ve velice základní podobě. Níže budou uvedeny některé uživatelské stránky, které klient zpřístupňoval. [15]

- **Seznam struktur** je výchozí stránka aplikace, která zobrazuje struktury získané z API volání GET `/structures`. Umožňuje pomocí tlačítek u každého záznamu ho smazat, upravit nebo zobrazit jeho detail. Stránka umožňuje filtraci podle jmen struktur.
- **Vytvoření struktury** je jednoduchý formulář, který má povinná a nepovinná pole. Výsledek odesílá pomocí API endpointu POST `/structures`
- **Detail struktury** zobrazuje veškerá data struktury získaná pomocí API volání GET `/structures/{structureId}`. Zobrazuje také tlačítko s odkazem na stránku s úpravami a tlačítko zpět.
- **Úprava struktury** je také přímočarý - pomocí formuláře umožňuje upravovat data ve struktuře a odesílá je na API pomocí PUT `/structures/{structureId}`. Poskytuje odkaz na úpravu 3D modelů struktury.
- **Úprava modelů struktury** umožňuje uživateli zobrazit a upravit 3D modely struktury a její assety a textury. Stránka získává informace z endpointů `/models/{modelId}` a `/3D0bjects/{td0bjectId}`.

Na závěr předmětu byl klient schopen provádět CRUD operace na strukturami a jejich 3D modely.

2.4.2 Týmový projekt 2

V rámci předmětu Softwarový týmový projekt 2 se pokračovalo ve vývoji webového klienta pro přístup k Private API VMCK. [16]

Nově bylo na API potřeba spravovat nové entity, **Areas** – oblasti. K tomu vznikla podobná sada stránek jako již existovala ke strukturám. Aplikace byla také byla rozšířena o navigační menu, více možností vyhledávání a další potřebné menší úpravy, například lepší vzhled.

- **Seznam Areas** zobrazuje výsledky API volání `GET /areas`. Podobně jako u struktur zobrazuje u každého záznamu tlačítka Odstranit, Upravit a Detail.
- **Přidání Area** zobrazuje formulář ke tvorbě nové oblasti a jeho obsah posílá na API endpoint `POST /areas`.
- **Detail Area** opět zobrazuje data získaná od API při volání `GET /areas/{areaId}`. Poskytuje tlačítka s odkazem na úpravu area.
- **Úprava Area** zobrazuje formulář na úpravu informací o záznamu. Nově také zobrazuje dvě mapy, kde na každé lze upravovat jiné potřebné informace. Mapy byly implementovány pomocí využití API od Mapy.cz společnosti Seznam.cz [17].

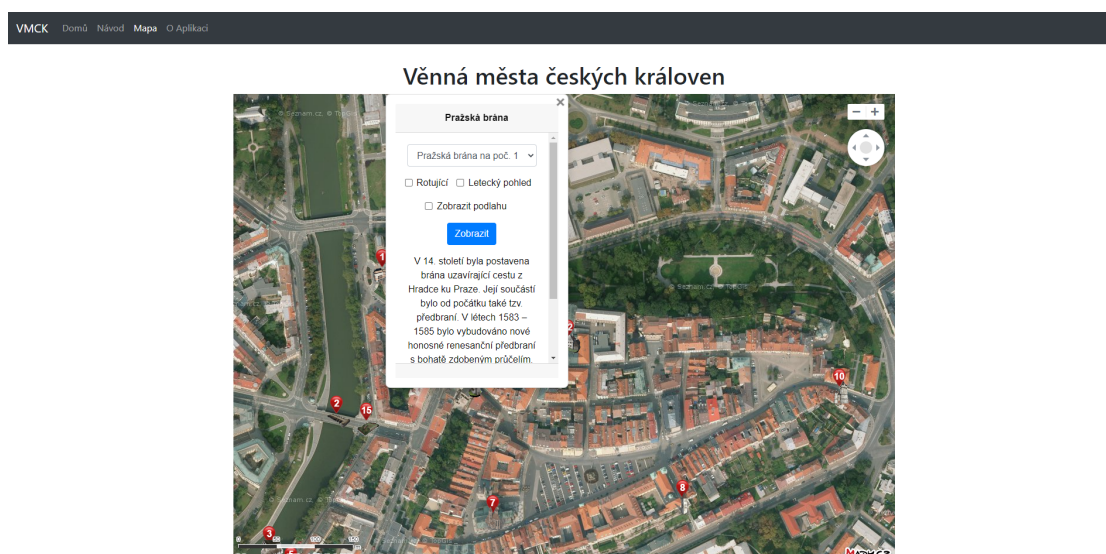
Dalším bodem zadání bylo vytvořit stránky pro správu **Effect** (efektů), **EffectGroup** (kombinací několika atomických efektů) a **Task** (úloh pro serverovou část). Vývoj backendu k těmto funkcionalitám aktuálně stále probíhá, zadáno tedy bylo navrhnout API endpointy pro tyto funkcionality a komunikaci s API si v těchto případech mockovat – simulovat. Ve stylu předchozích stránek pro Area a Structures tedy vznikly nové stránky pro mockovanou implementaci spravování těchto entit. Mockování probíhá tak, že aplikace při volání místo API volání dostává obsah textového souboru s předdefinovanými odpověďmi.

- **Effect** je zde zpracován jako spustitelný plugin. Klient umožňuje takové soubory vytvářet, nahrávat a stahovat. Jsou zde interaktivní okna, kde lze definovat výchozí parametry pro efekt, například parametry typu String, Vector, Enum a podobně. Doplnují se zde další informace, které by uživatel spouštějící efekt pomocí softwaru Blender mohl potřebovat.
- **EffectGroup** je skupina efektů. Ve skupině záleží na pořadí efektů, jelikož pluginy se budou na cílový objekt volat tak, v jakém pořadí byly vloženy. Klient umožňuje tvořit a ukládat skupiny efektů.
- **Task** je entita navržená tak, aby reprezentovala serverovou úlohu. Vyžaduje zvolení cílového 3D objektu a EffectGroup, dále pak vyplnění parametru pro každý jednotlivý Effect ve skupině.

2.4.3 Použitelnost

Použitelnost výstupů z týmových projektů je v této práci nízká. Aplikace vytvořená v těchto týmových projektech umožňuje provádět základní úpravy nad daty na Private API projektu VMCK v jednoduchém uživatelském rozhraní. Může některým uživatelům usnadňovat práci, obzvláště pokud tyto akce provádějí často. Podobné funkcionality umožňuje nástroj **Postman** a z důvodu osobní preference bude také při návrhu a implementaci v této práci používán.

Postman je nástroj pro vytváření a využívání API. Je možno si jej nainstalovat jako počítačovou aplikaci a používat jej mimo jiné pro provádění API volání. Cenová politika tohoto nástroje nabízí spoustu potřebných funkcionalit již v nejmenším balíčku, který je zdarma [7]. Další užitečná funkcionalita je možnost ukládat a organizovat svá API volání pro urychlení práce. Také je zde možnost vytvářet projekty a týmy, kde členům týmu můžeme sdílet své kolekce.



■ Obrázek 2.6 Stránka mapy a dialogové okno vybrané budovy – screenshot z webového průvodce

2.5 Analýza webového průvodce

Tato kapitola se bude věnovat webové stránce, která poskytuje široké veřejnosti část výsledků projektu VMCK. Tento průvodce je zveřejněn na této stránce [2]. V této kapitole budou popsány funkcionality této aplikace společně s obrazovými ukázkami. Bude provedena analýza potenciálu využití aplikace či jejích částí v tomto projektu.

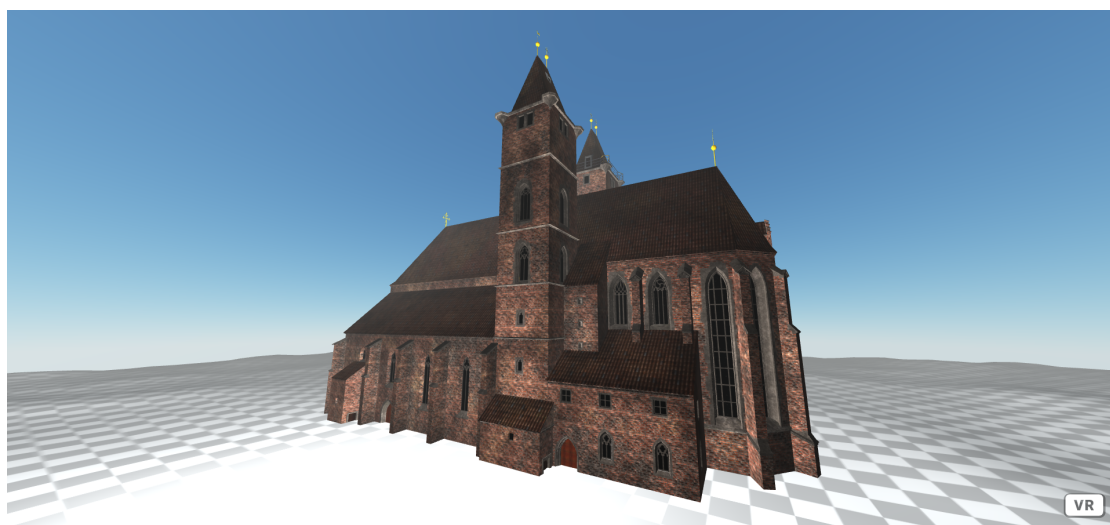
Jak bylo zmíněno v popisu projektu VMCK (2.2.1), tato stránka je výsledkem spolupráce několika organizací a je to první veřejně dostupná aplikace na prohlížení 3D modelů v projektu. Hlavní funkcí je zobrazování veřejných 3D modelů budov, aktuálně (v zimním a letním semestru roku 2022/23) pouze v centru města Hradec Králové.

2.5.1 Stránky webového průvodce

Zde budou stručně představeny některé důležité stránky, které lze navštívit na webové stránce <https://litepruvodce.kralovskavennamesta.cz/>. Budou stručně popsány jejich funkcionality a vzhled, který bude doplněn obrázky.

Mapa je hlavním přehledem modelů v aplikaci. Stránka zobrazuje interaktivní mapu typu Mapy.cz od společnosti Seznam. Při vývoji bylo pravděpodobně použito oficiální API pro Mapy.cz dostupné na <https://api.mapy.cz/> [17]. Na mapě se zobrazují speciální ikony ve tvaru budov společně s markery a identifikačním číslem. Každou ikonku lze rozkliknout, čímž se otevře malé dialogové okno. Toto okno se týká vybrané budovy a obsahuje krátký textový popis budovy a drobný formulář s prvky, které ovlivňují 3D zobrazení. Jde například o volbu modelu (jelikož budova může mít více variant, například z různých století) nebo zda chce uživatel zobrazit podlahu nebo využít letecký pohled. Stránka je předvedena v obrázku (2.6). Po zvolení možnosti **Zobrazit** je uživatel přeměrován na scénu, do které se mu načte zvolený 3D model, dle jeho požadavků ve formuláři.

3D scéna je interaktivní prostor, ve kterém se uživateli načte jeho zvolený model. Pomocí myši se lze ve scéně otáčet na místě a pomocí šipek nebo alternativy pro levou ruku (tlačítka w, a, s, d místo up, left, down, right) se lze po scéně pohybovat. Kombinace obého umožňuje



■ **Obrázek 2.7** Stránka scény z webového průvodce – model Katedrála sv. Ducha, screenshot z webového průvodce

relativně intuitivní pohyb kolem budovy, pro extrémní uživatele případně i skrz. V prostoru se nelze pohybovat nahoru a dolů, jediná alternativa je scénu zavřít a změnit nastavení zobrazení – zaměnit letecký pohled za obyčejný nebo naopak. Rychlejší varianta je změnit URL stránky, kde do parametru `camera2` vložíme požadovanou možnost. Nastavení `camera2=on` reprezentuje zapnutí leteckého pohledu a možnost `camera2=off` mód vypíná, což znamená, že stojíme ve výšší země. Podobným způsobem lze v URL přepínat parametr `rotation` pro rotaci modelu. Scéna s modelem Katedrály svatého Ducha je zobrazena v obrázku 2.7. Scéna má navíc i užitečnou funkcionalitu ukládání scény jako obrázek. Toho lze docílit kliknutím pravého tlačítka myši a volbou mezi *Uložit obrázek jako..* a *Kopírovat obrázek* do clipboardu počítače.

2.5.2 Použitelnost

Použitelnost výstupů aplikace v této práci se jeví jako vysoká. Tato aplikace je v relativně aktivním vývoji, je tedy docela malá šance, že bude rychle stárnout. Stránka se scénou přímo komunikuje s Private API a je schopná pomocí změny parametru `objectId` v URL zobrazit většinu 3D objektů z API, pokud zde vložíme platnou hodnotu identifikátoru entity `3DObject`. Aktuálně však scéna zobrazuje pouze modely ve formátech GLB a nikoliv OBJ a FBX. V databázi jsou aktuálně ve převážné většině modely ve formátu GLB (konkrétně gltf).

Pokud bylo rozhodnuto tuto scénu využít v této aplikaci, bylo by to pravděpodobně provedeno pomocí technologie `iframe`. Technologie `iframe` umožňuje do webové aplikace přidat externí HTML objekty pomocí jednoduchého tagu `<iframe>`. Tento tag je poté schopen zobrazovat externí zdroj, například zobrazí na naší stránce to, co zobrazuje stránka, jejíž URL `iframe` obdrží. [18]

Dále pokud by byla tato aplikace využívána v implementaci této práce, bylo by potřeba opravit zobrazování formátů. Aplikace by musela umět zobrazit modely ve všech formátech dostupných na Private API, tudíž OBJ, GLB a FBX. Tato změna je reálně proveditelná, jelikož jedním z vývojářů aplikace je Ing. Jiří Chludil.

2.6 Analýza uživatelských požadavků

Tato kapitola bude obsahovat analýzu požadavků, které na aplikaci budou mít její uživatelé. Hlavním zdrojem informací budou rozhovory se zadavatelem (vedoucím práce, Jiřím Chludilem) [5] a také dosavadní práce, které již uživatelské požadavky obsahovaly. Zdrojem je také analýza provedená v předmětu *Tvorba uživatelských rozhraní*, kde autor práce prováděl analýzu uživatelských požadavků a cílů [19].

2.6.1 Uživatelské role

Zde budou rozebrány uživatelské role, které mohou členové projektu VMCK v aplikaci mít přiřazeny. Každá role určuje jiné požadavky a jinou sestavu cílů, které uživatel v aplikaci bude mít a bude chtít splnit. Roli uživatel získává při registraci do projektu. Výchozí rolí je role **modelář**, její práva však nejsou aktivní, dokud není uživatelský účet schválen administrátorem [5].

Modelář je výchozí rolí pro nově registrované uživatele. Hlavním úkolem modeláře v projektu je vytvářet 3D objekty, nahrávat je do projektu a případně je odevzdávat ke schválení, pokud je jeho cílem nechat své výtvořiny zpřístupnit pro veřejnost.

U uživatele s rolí **modelář** se tedy předpokládá znalost počítačového 3D modelování. Tento uživatel tedy musí rozumět běžným termínům potřebným ke tvorbě grafických 3D modelů, jako je například textura, asset, transformace, translace, rotace apod. Také by měl rozumět tomu, jakou roli tyto termíny hrají ve 3D modelování. Očekává se, že používá počítačové grafické programy, jež mu umožní vytvořit a exportovat 3D modely v podporovaných formátech, jako jsou například OBJ, FBX nebo GLTF, a případně doplňující soubory, jako textury nebo assety.

Historik je role, která v rozsahu této práce spadá do nadmnožiny **schvalovatel**. Předpokladem pro tohoto uživatele je znalost modelovaných předloh, tzn. reálných budov. Případně se očekává schopnost si informace o reálných předlohách dohledat. Tento předpoklad zajišťuje, že v roli schvalovatele bude schopen objektivně hodnotit věrnost 3D modelů předloze.

Grafik je role, jež taktéž jako historik spadá do nadmnožiny **schvalovatel**. Předpokladem pro tohoto uživatele je být schopen dohledat si informace o předloze 3D modelu. Navíc je předpokladem, že uživatel bude znát pojmy a standardní postupy z prostředí 3D modelování a bude schopen objektivně ohodnotit kvalitu grafického zpracování 3D modelu.

Administrátor je role, která označuje uživatele, jenž v kontextu této aplikace rozhoduje, které modely budou zveřejněny pro širokou veřejnost. Uživatel musí být obeznámen s tím, jsou-li v projektu nějaká psaná či nepsaná pravidla, jež by veřejné 3D modely měly splňovat.

2.6.2 Uživatelské cíle

Zde budou rozebrány uživatelské cíle, které budou uživatelé mít v aplikaci, která vzniká v této práci. Cíle budou rozděleny podle uživatelských rolí v projektu. Role **historik** a **grafik** zde budou sloučeny do jedné zastřešující role **schvalovatel**, protože jejich požadavky na tuto aplikaci splývají do stejné množiny.

2.6.2.1 Obecné cíle

Zde budou uvedeny uživatelské cíle, jejichž splnění musí být umožněno všem uživatelům bez závislosti na roli. Jedná se o základní požadavky.

Registrace a přihlášení do aplikace

- **Registrovat se** do aplikace pomocí Google účtu, pokud účet ještě neexistuje.
- **Přihlásit se** do aplikace pomocí Google účtu, pokud již účet v aplikaci existuje.

Získání práv v projektu

- **Získat roli v projektu** po tom, co se registruje.
- **Získat práva** v projektu. Jedná se o práva na nahrávání a úpravy dat v projektu a o práva na určitá API volání.

2.6.2.2 Modelář

Uživatel s touto rolí má v aplikaci potřebu vytvářet 3D objekty a spravovat je. Dále bude své modely odesílat ke schválení a sledovat postup schvalování jednotlivých 3D objektů.

Uživatelské cíle pro tuto roli jsou uvedeny níže. Veškerá manipulace s 3D objekty je povolena pouze autorovi nad jeho vlastními výtvoři.

Tvorba a správa 3D objektů

- **Nahrát** do aplikace **nový** 3D objekt se všemi náležitostmi 3D objektu, například texturami, assety, transformační maticí apod.
- **Upravit** uložený model.
- **Vytvořit** novou **verzi** existujícího modelu.
- **Smazat** verzi 3D objektu.

Prohlížení vytvořených 3D objektů

- **Zobrazit veškeré** uložené 3D objekty.
- **Zobrazit** detail jednoho vybraného 3D objektu ve **3D formě**.
- **Zobrazit** detail jednoho vybraného 3D objektu ve **formulářové formě**. Tím zobrazí veškeré detaily, které byly vyplněny při odeslání.

Schvalování 3D objektů

- **Odeslat** nejnovější verzi 3D objektu ke schválení.
- **Zobrazit** všechny 3D objekty, jež jsou aktuálně registrované v procesu schvalování. To obsahuje jak 3D objekty ohodnocené, tak neohodnocené.
- **Sledovat stav** schvalování vybraného 3D objektu – zda byl již schválen nebo zamítnut a dodatečné informace o iteraci a 3D objektu.
- **Být upozorněn** na změny stavu schvalování u jednotlivých 3D objektů. Jedná se například o situaci, když je jeho model schválen nebo zamítnut. Preferovaným prostředkem je email.

2.6.2.3 Schvalovatel

Uživatel v této uživatelské skupině má roli **grafik** nebo **historik**. Uživatel bude v aplikaci dostávat přiřazené úkoly – iterace schvalovacích procesů, ke kterým se musí vyjádřit. Uživatelské cíle pro tuto skupinu uživatelů jsou uvedeny níže.

Přehled schvalovacích procesů

- **Zobrazit všechny** schvalovací procesy, jichž je členem.
- **Zobrazit stav** všech schvalovacích procesů, jichž je členem.

Přehled úkolů

- **Zobrazit všechny** iterace, které čekají na jeho hodnocení, ve stručném přehledu.
- **Zobrazit jeden** vybraný úkol. To značí jednu iteraci schvalovacího procesu, jež je navázána na jeden 3D objekt. Mít přehled o aktuálním stavu schvalování této iterace. Mít možnost zobrazit si zúčastněný 3D objekt.
- **Zjistit stav** iterace, kterou má ohodnotit. Mít dostupnou informaci o postupu schvalování, například zda ji již někdo schválil nebo jestli někdo přidal komentáře.

Schvalování

- **Ohodnotit** jeden 3D objekt – jednu iteraci schvalovacího procesu. Mít při hodnocení zobrazeny důležité informace o 3D objektu a iteraci, například komentáře, které již u iterace existují. Mít dostupnou informaci o stavu schvalování – například zda ji již někdo schválil nebo zamítnul. Mít možnost jako součást hodnocení odeslat poznámku.
- **Vzít zpět** hodnocení iterace.
- **Opět ohodnotit** iteraci, u níž vzal zpět své rozhodnutí.
- **Být upozorněn** ve chvíli, kdy je přidán nový úkol. Jedná se novou odevzdanou iteraci ve schvalovacím procesu, jehož je uživatel členem. Preferovaným prostředek je email.

2.6.2.4 Administrátor

Uživatel v této uživatelské roli má v kontextu této práce za úkol kontrolovat schválené 3D objekty a rozhodovat, zda jsou vhodné k publikaci nebo nikoliv. Uživatelské cíle pro tohoto uživatele jsou uvedeny níže.

Veřejné 3D objekty

- **Zobrazit** stručný přehled všech 3D objektů, které byly označeny jako veřejné, v kontextu této práce tedy ve FINALIZED stavu.
- **Zobrazit** detail vybraného 3D objektu, který byl zveřejněn. Mít možnost zobrazit si 3D formu objektu.

Přehled úkolů

- **Zobrazit** stručný přehled 3D objektů, jež byly schváleny a čekají na publikaci (finalizaci).
- **Zobrazit** detail vybraného 3D objektu, který čeká na zveřejnění. Mít možnost zobrazit si 3D formu objektu. Mít možnost zobrazit si komentáře ze schválené iterace.

Publikace 3D objektů

- **Publikovat** vybraný 3D objekt. Jedná se o objekt z iterace, jež byla schválena. Mít možnost si u publikace zobrazit 3D formu objektu a komentáře z iterace.
- **Vzít zpět** publikaci 3D objektu.
- **Znovu publikovat** 3D objekt, u něž byla zrušena publikace.

2.7 Funkční a nefunkční požadavky

Z analýzy domény a uživatelských cílů lze postavit pro aplikaci požadavky. Ty budou dle principů softwarového inženýrství rozděleny na funkční a nefunkční. Požadavky budou označeny svou prioritou. **Vysoká** priorita označuje požadavek, jenž je stěžejní pro použitelnost aplikace. **Nízká** priorita označuje „nice to have“ funkcionality, které by použití aplikace zpříjemnily nebo zjednodušily.

2.7.1 Funkční požadavky

F1: Správa uživatelů: Aplikace umožní základní manipulaci pro správu uživatelů.

F1.1: Registrace: Aplikace umožní novým uživatelům registrovat se do projektu.

Priorita: vysoká

F1.2: Přihlášení: Aplikace umožní registrovaným uživatelům přihlásit se do aplikace svým projektovým účtem.

Priorita: vysoká

F1.3: Získání role: Aplikace přidělí nově registrovaným uživatelům projektovou roli a projektová práva patřící k roli.

Priorita: vysoká

F2: Správa 3D objektů: Aplikace umožní základní manipulaci s 3D objekty, včetně jejich souborů.

F2.1: Správa jednoho 3D objektu: Aplikace umožní uživateli s rolí **modelář** provádět CRUD operace (create, read, update, delete) nad 3D objekty. U read, update a delete se jedná pouze o 3D objekty vytvořené tímto uživatelem.

U **create** – tvorby – aplikace umožní nahrávat veškeré informace a soubory pro entitu 3D objekt.

Pro **read** – čtení – aplikace umožní zobrazit jak data zadaná při vytváření objektu, tak 3D zobrazení objektu.

U **update** – úpravy – aplikace umožní upravovat data a soubory uložených 3D objektů.

Pro **delete** – smazání – aplikace umožní smazat 3D objekt. U mazání se dle návrhu backendu jedná o nejnovější verzi.

Priorita: vysoká

F2.2: Tvorba nové verze 3D objektu: Aplikace umožní uživateli s rolí **modelář** vytvořit a uložit novou verzi 3D objektu. Jedná se o tvorbu nového 3D objektu, který bude po vytvoření zařazen do seznamu verzí vybraného 3D objektu.

Priorita: vysoká

F2.3: Přehled 3D objektů: Aplikace umožní uživateli s rolí **modelář** zobrazit veškeré 3D modely, jež vytvořil. Umožní mu provádět s těmito 3D modely vybrané akce.

Priorita: vysoká

F3: Schvalování 3D objektů: Aplikace umožní schvalování a zamítání 3D objektů.

F3.1: Odeslání 3D objektu ke schválení: Aplikace umožní uživateli s rolí **modelář** odeslat 3D objekt, který je **dokončený**, do schvalovacího procesu. 3D objekt bude odeslán v samostatné iteraci schvalovacího procesu.

Priorita: vysoká

F3.2: Ohodnocení schvalovací iterace: Aplikace umožní uživateli s rolí **historik** nebo **grafik** zobrazit si odeslaný 3D objekt – odeslaný v iteraci schvalovacího procesu – a rozhodnout, zda jej schválí nebo zamítne. Aplikace umožní společně s hodnocením odeslat poznámky.

Priorita: vysoká

F3.3: Zrušit hodnocení schvalovací iterace: Aplikace umožní uživateli s rolí **historik** nebo **grafik** vzít zpět své hodnocení k 3D objektu ve schvalovací iteraci. Akce bude umožněna pouze, pokud iterace a model nebyly finalizovány – zveřejněny.

Priorita: vysoká

F3.4: Sledování stavu iterace: Aplikace umožní uživateli s rolí **modelář**, **historik** nebo **grafik** zobrazit si stav schvalování iterace, ke které má uživatelská práva. Do stavu iterace patří jak vyjádření autorit, tak dosavadní komentáře. Uživatelská práva získá uživatel tak, že se stane členem schvalovacího procesu.

Priorita: vysoká

F3.5: Přehled schvalovacích procesů: Aplikace umožní uživateli s rolí **modelář**, **historik** nebo **grafik** zobrazit si přehled všech schvalovacích procesů, jichž je členem a na něž má práva. Aplikace umožní prohlížet jednotlivé iterace těchto procesů.

Priorita: vysoká

F3.6: Přehled úkolů: Aplikace umožní uživateli s rolí **historik** nebo **grafik** zobrazit si přehled všech schvalovacích iterací, které čekají na jeho ohodnocení.

Priorita: vysoká

F3.7: Zasílání upozornění: Aplikace upozorní členy schvalovacího procesu na změny. Uživateli s rolí **modelář** zašle upozornění, pokud se změnil stav jeho 3D objektu ve schvalování. Uživateli s rolí **historik** nebo **grafik** pošle upozornění, pokud do procesu, jehož je členem, byla odevzdána nová iterace. Uživateli s rolí **administrátor** pošle upozornění, pokud byl nějaký 3D model schválen a je připraven na publikaci.

Priorita: nízká

F4: Publikace 3D modelů: Aplikace umožní spravovat, zda jsou 3D modely označeny jako veřejné.

F4.1: Přehled publikovatelných 3D objektů: Aplikace umožní uživateli s rolí **administrátor** zobrazit všechny 3D modely (iterace), které byly schváleny a je možno je publikovat – označit za **finalizované**.

Priorita: vysoká

F4.2: Publikace 3D objektů: Aplikace umožní uživateli s rolí **administrátor** zobrazit si schválenou iteraci a její 3D objekt. Uživatel bude moci tento model označit za **finalized** – veřejný.

Priorita: vysoká

F4.3: Přehled publikovaných 3D objektů: Aplikace umožní uživateli s rolí **administrátor** zobrazit všechny 3D modely (iterace), které byly schváleny a poté publikovány – označeny za **finalized**.

Priorita: vysoká

F4.4: Zrušení publikace 3D objektů: Aplikace umožní uživateli s rolí **administrátor** zrušit publikaci 3D objektu (iterace), který byl publikován. Tím se stav objektu vrátí ze stavu **finalized** do **approved**.

Priorita: vysoká



■ **Obrázek 2.8** Use cases diagram – modelováno v Diagrams.net [20]

2.7.2 Nefunkční požadavky

N1: Přihlášení a registrace pomocí Google: Aplikace dle návrhu projektu bude používat k přihlášení účet třetí strany, **Google**.

N1.1: Registrace: Aplikace umožní novým uživatelům použít k registraci účet třetí strany, **Google**.

Priorita: vysoká

N1.2: Přihlášení: Aplikace umožní existujícím uživatelům použít k přihlášení účet třetí strany, **Google**.

Priorita: vysoká

N2: Jazyk Aplikace bude poskytovat uživatelské rozhraní v **češtině**.

Priorita: vysoká

N3: Responzivita Aplikace bude responzivní. Aplikace bude dávat uživateli zpětnou vazbu o stavu aplikace.

Priorita: vysoká

2.8 Analýza technologií

Zde bude provedena analýza technologií, mezi kterými lze volit při implementaci aplikace.

Tato práce má v zadání určenou technologii **React** – je tedy požadavkem zadavatele, aby při implementaci byla používána tato technologie. Z tohoto důvodu zde budou analyzovány různé možnosti, jak React použít. Naopak zde nebudou analyzovány jiné technologie typicky používané pro implementaci frontendu.

2.8.1 React JS

Zde budou uvedeny některé důležité vlastnosti, pozitiva a negativa JavaScript knihovny React.js.

2.8.1.1 Popis

React JS je JavaScript knihovnou pro zjednodušení vývoje frontendu. Je používán ke tvorbě interaktivních webových aplikací, přičemž vyžaduje mnohem méně kódu než klasický JavaScript.

Princip Reactu je postaven na tvorbě znovupoužitelných komponent, jejichž vzájemným skládáním se tvoří uživatelské rozhraní aplikace. React se snaží udělat aplikaci co nejefektivnější z pohledu renderingu² tak, že pomocí hierarchické struktury jednotlivých komponent vybírá, které komponenty se musí aktualizovat a které ne. Tím se výrazně zrychluje odezva aplikace a dělá ji vysoce responzivní.

2.8.1.2 Výhody a nevýhody

Výhody použití React JS. Jednou z velkých výhod Reactu je výše zmíněný systém komponent, který umožňuje psát vysoce responzivní aplikace s efektivním systémem renderování.

Další výhodou Reactu je například to, že dává developerům spoustu prostoru na vlastní realizaci. Nevnučuje totiž vývojářům striktní pravidla a konvence, jako například framework Angular. [21]

²vykreslování

React také disponuje velkým množstvím knihoven, které výrazně usnadňují a urychlují práci tím, že nám poskytují hotové komponenty či funkcionality. Také například dobře podporuje vývoj na mobilních zařízeních. [21]

React JS také podporuje formát souborů JSX. Formát JSX označuje JavaScript XML, což je rozšíření jazyka JavaScript. Toto rozšíření je oblíbeným nástrojem pro strukturaci JavaScript kódu tak, aby syntakticky připomínal formát HTML a zjednodušoval tak programátorům implementaci. [22]

React se také těší čím dál tím větší popularitě, což znamená, že se rozrůstá jeho komunita. Z toho důvodu je čím dál tím jednodušší najít v této komunitě například rady nebo doporučení. V této komunitě také vznikají nové moderní knihovny a jsou udržovány ty stávající.

Nevýhody použití React JS. Některé výhody React JS mohou být zároveň i nevýhodami. Příkladem je rychle se rozrůstající knihovna. Rychlé tempo změn v použití frameworku nutí vývojáře se často přizpůsobovat změnám. To se také projevuje v dokumentaci, kterou je čím dál tím těžší v komunitě udržovat aktuální.

Také flexibilita a volnost je zároveň nevýhodou. Když neexistují konvence a standardy, může být složité svůj styl programování a své vlastní konvence synchronizovat s někým jiným. [23]

Velkou nevýhodou u použití React JS (a jazyka JS obecně) je jeho dynamické typování. Při používání JS je velice jednoduché udělat chybu a vytvořit tím závažné runtime chyby.

2.8.2 React TS

Zde budou uvedeny některé vlastnosti, pozitiva a negativa nadstavby knihovny React, React TypeScript.

2.8.2.1 Popis

TypeScript je programovací jazyk, který je typovaným supersetem (nadstavbou) JavaScriptu. Je to staticky typovaný jazyk a jelikož je supersetem, každý JS kód je také validním TS kódem. [24]

TypeScript se v poslední době stává populárním, proto také knihovna React podporuje mimo JS i tento jazyk.

2.8.2.2 Výhody a nevýhody

Výhody použití React TS jsou významné. Použití TS v React aplikaci značně zvyšuje přehlednost a bezpečnost aplikace díky svému statickému typování. To například umožňuje kompilátoru TS detekovat chyby již za kompilace místo za runtime, jak je tomu u JS. Typescript také přináší prvky objektově orientovaného programování, jako například třídy, moduly, generika, pattern matching nebo typovou inferenci. [25]

React TS podporuje formát TSX, který odpovídá typované verzi formátu JSX. Vývojáři tedy při použití React TS neztratí tuto funkcionality.

Nevýhody použití React TS je potřeba také uvést. TypeScript je složitější na naučení než JavaScript a nováčkům proto potrvá výrazněji déle začít v něm pracovat. Také je někdy náročné použít některé JS knihovny nebo externí pluginy, protože kvůli svému striktnímu typování může být problém s kompatibilitou. [25]

Kapitola 3

Návrh

V této kapitole bude proveden návrh procesů a návrh důležitých částí aplikace, která v práci bude implementována.

Nejdříve bude proveden návrh schvalovacího procesu pro aplikaci. Bude zde rozebrána volba, na který starší návrh procesu je vhodné navázat.

Dále bude proveden návrh obrazovek aplikace a jejich funkcionality. Důležité stránky budou ukázány na prototypních wireframe obrázcích.

V poslední části kapitoly bude proveden rozbor, zda návrh aplikace pokrývá veškeré funkční požadavky.

3.1 Návrh schvalovacího procesu

Návrhů schvalovacího procesu již v projektu VMCK vniklo několik a nyní musí být provedena volba, který bude aplikován v této práci. Bude zde uveden krátký popis a porovnání existujících návrhů. Následně bude provedeno rozhodnutí, který návrh bude nejlepším výchozím bodem pro vlastní návrh procesu.

3.1.1 Existující návrhy

Zde budou uvedeny návrhy z analyzovaných závěrečných prací. Zastaralé návrhy z prací kolegů Martínka a Vančury již nejsou považovány za kandidáty, zbývají tedy návrhy kolegů Siváka a Antoše. Budou zde uvedeny nejdůležitější rozdíly v jejich návrzích a bude rozhodnuto, který návrh bude použit v této práci a proč. Podrobný rozbor návrhů a jejich diagramy byl proveden v kapitole o analýze závěrečných prací (2.3).

Návrh Pavla Antoše vycházel z návrhu kolegy Martínka. Cyklus schvalování zde probíhá tak, že pokud je model zamítnut a je poté autorem přepracován, získá nová verze modelu stav **FIXED** (opravený). Při každém opakování schvalování se čeká, než se obě schvalovací autority kladně vyjádří, nebo než jedna zamítne. Při zamítnutí je model vrácen autorovi k přepracování. Ve chvíli, kdy je model schválen oběma autoritami, stává se model **APPROVED** (schválený). Spouštění automatických transformací a publikování modelu jsou v návrhu uvedeny pouze jako blíže nerozvedené kroky. V návrhu se nepočítá s tím, že by se modely po aplikaci transformací měly znova schvalovat.

Návrh není ve shodě s aktuálními backendovými procesy, jelikož tyto procesy na backendu vytvářel kolega Sivák a jelikož obě tyto práce byly psány paralelně. Velkým rozdílem také je, že od samotné tvorby modelu až po publikaci se jedná o jeden cyklický a nepřerušovaný proces.

Návrh Dominika Siváka rozvíjel návrh kolegy Vančury. Kolega Sivák ve svém návrhu eliminoval stav modelu **FIXED** (opravený). Docílil tím toho, že každé zaslání ke schválení reprezentuje jednu iteraci procesu a skončí výsledkem **APPROVED** (schváleno) nebo **DECLINED** (zamítnuto), případně **FINALIZED** (finalizováno), pokud byl schválený model zveřejněn. Zavedl také dvě fáze schvalování, kde v první se schvaluje samotný 3D model a v druhé jeho transformace. V první iteraci je potřeba být schválen historikem i grafikem, v druhé fázi pak pouze grafikem.

Jeho návrh byl implementován na backendu, což znamená, že takto návrh vypadá dodnes, až na drobné změny. Proto je jasnou volbou v této práci použít tento návrh, s případnými úpravami.

3.1.2 Vlastní návrh schvalovacího procesu

V předchozí části byl označen návrh kolegy Siváka jako vhodný kandidát na návrh procesu v této práci. Hlavním důvodem, proč zvolit tento návrh, bylo, že takto je proces aktuálně funkční na backendu aplikace a dá se tedy pojmenovat jako nejaktuálnější. Obsahem této práce není upravovat backendovou část projektu, pouze případně navrhnout změny, pokud budou nutné pro splnění požadavků aplikace. Proto je vhodná volba vybrat takový návrh, aby odpovídal návrhu na backendu aplikace.

Budou zde uvedeny nové diagramy, které podrobněji rozeberou průchod některými procesy se zaměřením na jednotlivé role a jednotlivé fáze procesu. Jedná se o diagramy procesů, jak je navrhl kolega Sivák [6]. Jelikož však ve své práci vlastní diagramy těchto konkrétních procesů neměl, budou zde nově vytvořeny. Cílem je zvýšit přehlednost v aplikaci a v jejím návrhu. Na diagramech bude znázorněno, jak se změnou návrhu zjednodušily jednotlivé procesy oproti návrhům, které byly v projektu před kolegou Sivákem. Pro tvorbu diagramů byl používám nástroj Diagrams.net [20], [26].

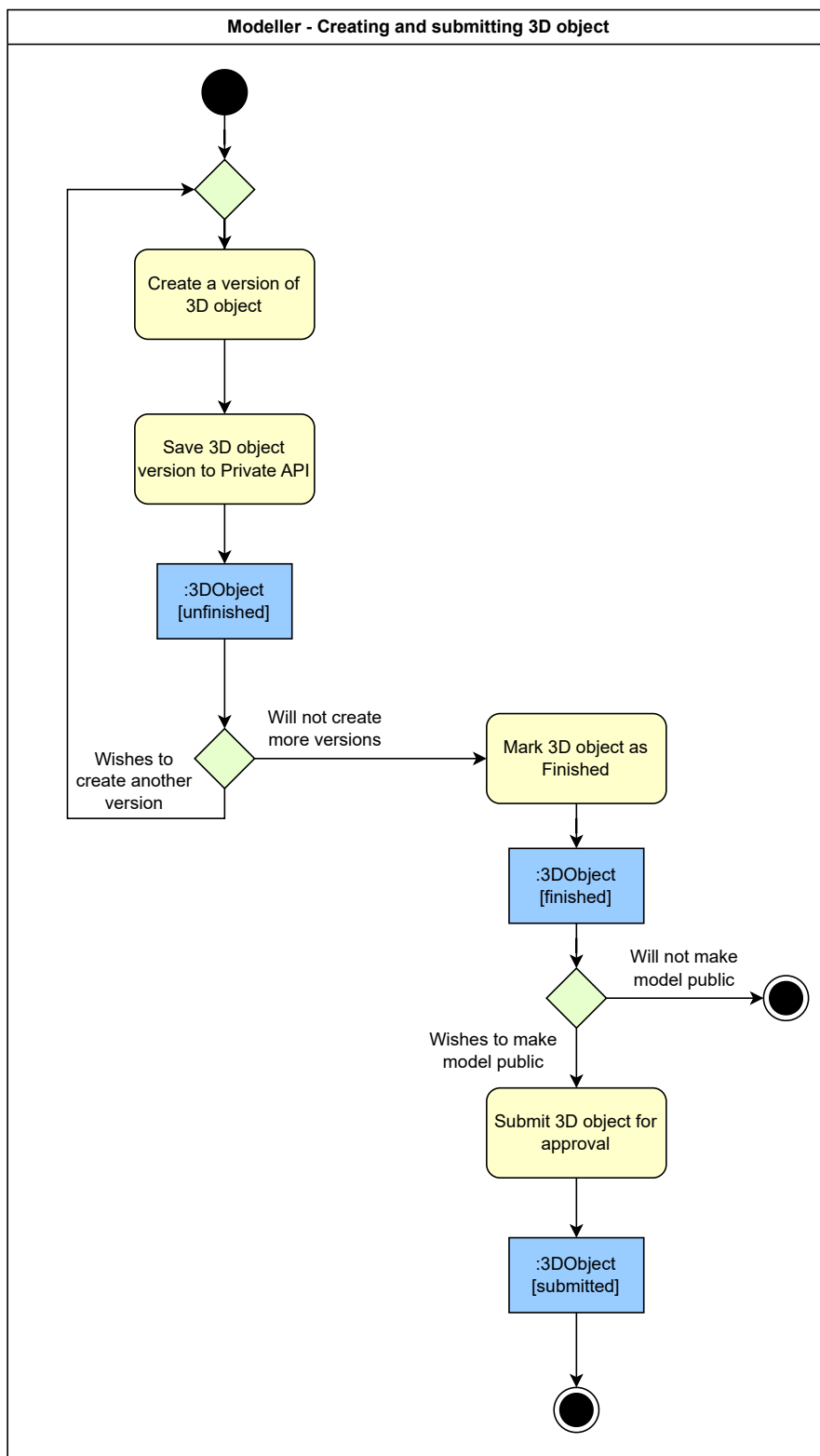
Stavy modelu budou používány tak, jak byly popsány v rozboru práce kolegy Siváka v kapitole 2.3.1.3. Jedná se o stavy **UNFINISHED**, **FINISHED**, **SUBMITTED**, **DECLINED**, **APPROVED** a **FINALIZED**.

Je důležité upozornit na to, že na rozdíl od prací kolegů Vančury, Martínka a Antoše je zde změna stavů pouze jednosměrná, nikoliv cyklická. Kolega Sivák navrhl a implementoval proces tak, že každá iterace skončí s výsledkem **approved** nebo **declined** (případně **finalized**) a další schvalování nové opravené verze bude předmětem nové samostatné iterace. Nejedná se tedy o cyklický proces, kde se opravuje stejný model pomocí stavu **FIXED**.

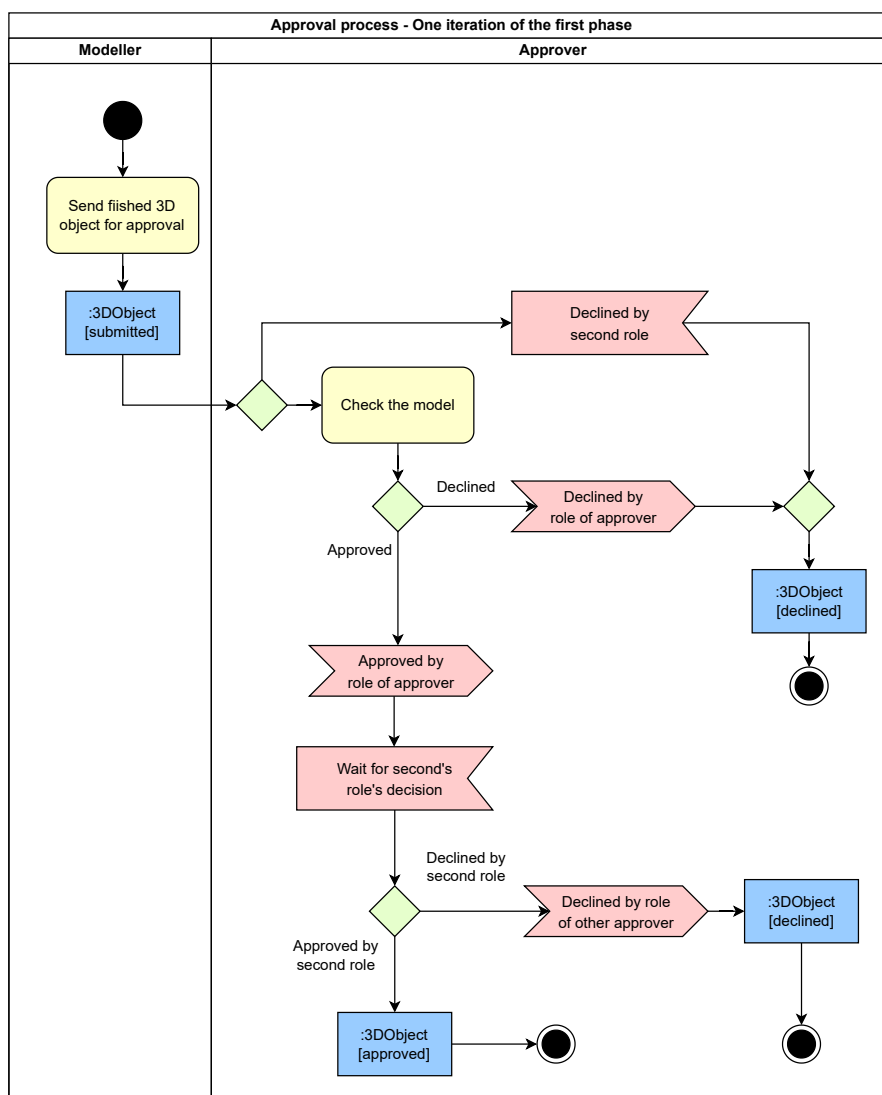
Tvorba modelu modelářem je podproces, který je jako jediná část procesu cyklický. Modelář je zde jediným aktérem. Modelář v tomto procesu nahrává na Private API verzi 3D modelu, dokud není s výsledkem spokojen. Poté označí nejnovější verzi 3D objektu jako **FINISHED** a buďto tímto proces ukončí, nebo odešle model ke schválení. Cílem schvalování je, aby 3D objekt získal stav **APPROVED** a mohl být zveřejněn na veřejné API. Po odeslání ke schválení získává model stav **SUBMITTED**. Proces je modelování, ukládání a odeslání je znázorněn na diagramu 3.1.

Proces schvalování v první fázi vyžaduje schválení od grafika a historika. Pokud jedna autorita zamítne, 3D objekt se přesouvá do stavu **DECLINED** a proces je automaticky ukončen u obou autorit. To znamená, že je stažen z front úkolů u grafika i historika.

Pokud obě autority model schválí, obdrží 3D objekt stav **APPROVED** a bude tímto umožněno administrátorovi tento model zveřejnit. V první fázi je proces stejný z pohledu Grafika i Historika, v diagramu 3.2 je tedy použit aktér Approver, neboli Schvalovatel, který zastupuje obě role.



■ Obrázek 3.1 Diagram tvorby 3D modelu a odeslání ke schválení – modelováno v Diagrams.net [20]



■ **Obrázek 3.2** Diagram procesu schvalování jednou autoritou – modelováno v Diagrams.net [20]

Proces schvalování ve druhé fázi vyžaduje pouze schválení grafikem. Jedná se o vizuální transformace, jako například efekty počasí nebo ročního období. Průběh je téměř identický jako v první fázi. Jedním rozdílem je, že je potřeba pouze jedna schvalovací autorita. Také jsou zde dvě možnosti tvorby nového modelu. Buďto byla na model zavolaná automatická transformace na serveru, nebo byl modelář požádán o ruční úpravu.

Aktuálně není na backendu v provozu automatická transformace a aplikace efektů. Tyto procesy a akce jsou zatím pouze navrženy. V implementaci v této práci tedy nebudou automatické serverové transformace brány v potaz.

Dle nového návrhu **variant** na backendu projektu budou varianty řešeny tak, že budou obvyčejnými 3D objekty. Budou mít však speciální atributy, **properties**, které budou určovat efekt, který byl na variantu aplikován. Tyto objekty budou tedy moct podléhat stejnému schvalovacímu procesu, jako iterace modelu. Bude však nutné nastavit nová pravidla pro počet schvalujících autorit.

Jelikož však **properties** na backendu projektu zatím nejsou implementované, ani v návrhu aplikace se nepředpokládá, že by byly schvalovány takovéto objekty.

Proces publikace schválených modelů má jediného aktéra, administrátora. Uživatel v této roli si může zobrazit všechny APPROVED modely, které čekají na publikaci. Administrátor může modely libovolně přesouvat mezi stavy APPROVED a FINALIZED. Tudíž schválené modely může udělat veřejnými a veřejné modely může vrátit do stavu „schválený“.

3.2 Návrh obrazovek uživatelského rozhraní

V této kapitole budou navrženy pohledy (views, stránky, nebo také obrazovky) ve webové aplikaci. Stránky budou navrhovány podle uživatelských rolí v projektu VMCK a podle jejich uživatelských požadavků. Cílem aplikace je umožnit uživatelům splnit uživatelské cíle, které mají v kontextu této práce určeny.

Návrh obrazovek bude sloužit jako inspirace pro implementaci. Také bude použit k základnímu ověření, zda obsah návrhu odpovídá požadavkům zadavatele. Návrhy obrazovek budou v průběhu návrhu aplikace také konzultovány se zadavatelem, aby bylo před implementací aplikace ověřeno, zda budou splňovat uživatelské a projektové požadavky.

Pro účel kontroly splnění funkčních požadavků budou stránky zkratkovitě označeny. Písmeno O, M, S, A bude označovat uživatele – obecné, modelář, schvalovatel a administrátor. Dále bude přiděleno pořadové číslo. Takže například stránka M2 bude stránka modeláře definovaná jako druhá v pořadí.

Veškeré grafické návrhy byly modelovány v nástroji **Diagrams.net**[20] a byly vytvářeny z vlastní fantazie, bez existujících šablon.

3.2.1 Stránky společné všem uživatelům

Zde budou uvedeny stránky a funkcionality aplikace, jež budou dostupné všem uživatelům aplikace. Jedná se tedy o uživatele přihlášené i nepřihlášené, a v případě přihlášených uživatelů o libovolnou uživatelskou roli.

Úvodní obrazovka (O1) bude hlavní stránkou, kde bude uživatel přivítán a bude uveden do aplikace. Budou mu objasněny důležité informace, jako například o jakou aplikaci se jedná nebo co se v ní dá dělat. Úvodní obrazovka bude mít dvě podoby, jednu pro přihlášené uživatele a jednu pro nepřihlášené.

Stránka pro **nepřihlášené** uživatele bude velice stručná. Sdělí uživateli, o jakou aplikaci se jedná. Dále bude uživatel požádán o Google přihlášení. Bude také seznámen s tím, jak funguje registrace a přihlášení do aplikace.

Stránka pro **přihlášené** uživatele zobrazí opět základní informace o projektu. Dále mu bude vysvětleno, jak se může po aplikaci pohybovat pomocí navigačního menu. Bude mu také stručně popsáno, jaký je význam jednotlivých uživatelských rolí v projektu.

Navigační menu (O2) se bude také zobrazovat všem uživatelům, přihlášeným i nepřihlášeným. Bude však obsahovat rozdílné informace a různá navigační tlačítka v závislosti na stavu uživatele a jeho uživatelských právech. Návrh vzhledu navigačního tlačítka bude proveden s ohledem na aktuálně moderní praktiky.

„Navigační menu by mělo být jednoduché a jasné, s textem, který bude snadno čitelný. ... Navigační menu by mělo vyčnívat a upoutat pozornost. Nebojte se používat kontrast, barevná schémata a další vizuální triky na zvýraznění stisknutelných tlačítek. ... Co je nejdůležitější, navigační menu by mělo zodpovídat otázky, nikoliv je vytvářet.“ (přeloženo autorem práce)[27]

Pro **nepřihlášeného** uživatele se zobrazí pouze tlačítko pro **Google přihlášení**. Pro **přihlášeného** uživatele se zobrazí tlačítko pro **odhlášení**, dále pak navigační tlačítka, která budou odkazovat na vybrané důležité stránky v aplikaci.

Google přihlášení (O3) není vlastní stránka, ale funkcionalita, která po použití tlačítka **Přihlásit se** zobrazí „popup“ (v překladu „vyskakovací“) okno pro přihlášení pomocí Google účtu. Jediné dostupné tlačítko pro nepřihlášeného uživatele bude právě **Google přihlášení** v navigačním menu. Takto se může uživatel přihlásit se do webové aplikace, případně být registrován do projektu VMCK a následně přihlášen do aplikace.

Pro přihlášené uživatele se bude zobrazovat tlačítko **odhlásit**.

3.2.2 Stránky modeláře

Zde budou uvedeny stránky, které budou přístupné uživatelům s rolí **commonUser**, v kontextu této práce tedy **Modelář**.

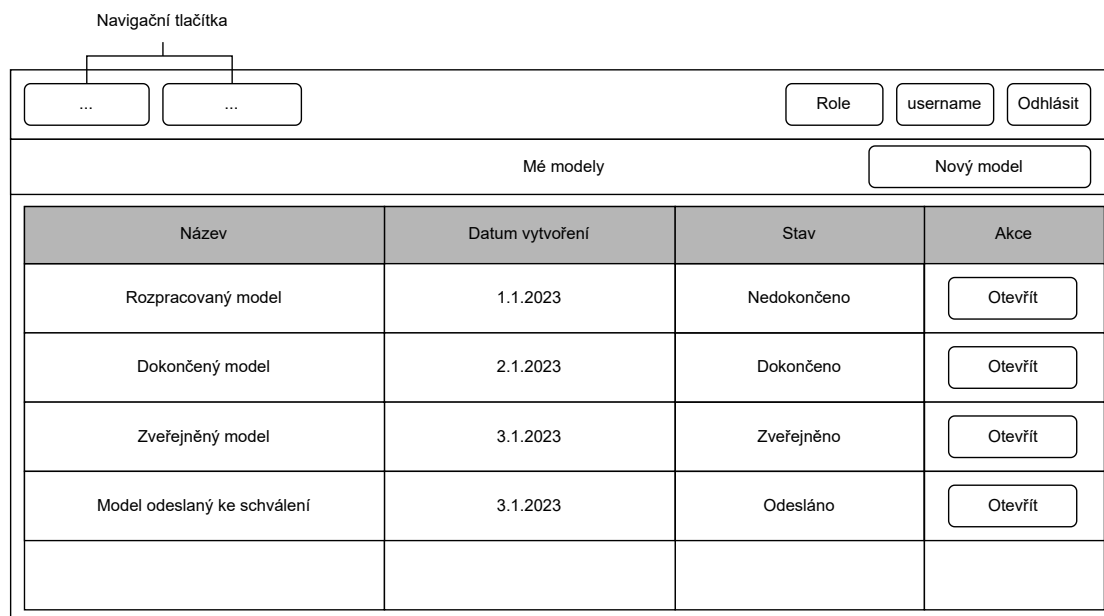
Stránka s modely (M1) je navržena tak, aby zobrazila všechny modely, jichž je uživatel autorem. To značí modely ve všech stavech – nedokončené, dokončené, veřejné a další. Je důležité upozornit na to, že se jedná o stavy 3D objektu na API, nikoliv o stavy schvalování, ačkoliv ten se také promítne do stavu modelu. Na sledování stavu schvalování bude mít modelář vlastní stránku s přehledem vývoje jednotlivých schvalovacích procesů. Detaily o průběhu schvalování jsou totiž spravovány v iteraci schvalovacího procesu, nikoliv v 3D objektu samotném. Stránka s přehledem uložených modelů je navržena na obrázku 3.3.

Modelář bude mít na této stránce také tlačítko pro tvorbu nového modelu. To značí tvorbu modelu, který nebude navazovat na žádný existující model a bude to tedy první verze. U každého již uloženého modelu budou zobrazeny základní informace a akční tlačítka. Pomocí těchto tlačítek budou umožněny důležité akce s 3D objektem – například zobrazit 3D objekt, zobrazit data vyplněná při odeslání, upravit nebo smazat.

Formuláře pro vytvoření nebo úpravu 3D objektu (M2) jsou navrženy tak, aby umožnily autorovi vyplnit všechna pole, která jsou navržena u entity 3D objekt. Formulář je navržen na obrázku 3.4. Konkrétní množina polí formuláře se může změnit podle toho, která pole bude povoleno vyplňovat nebo měnit při úpravě nebo tvorbě nového 3D objektu.

Formuláře budou sloužit k tvorbě modelů, k tvorbě nových verzí nebo k úpravám dat v uloženém 3D objektu.

Detail uloženého modelu (M3, M4) je vhodné navrhnout ve dvou formách, ve formě dat a ve formě 3D modelu. Budou zde tedy dvě rozdílné stránky.



■ **Obrázek 3.3** Návrh obrazovky s uloženými modely modeláře

V první formě (**M3**) je vhodné zobrazit uživateli formulářová data modelu – to značí zobrazit mu 3D objekt tak, jak jej nahrál ve vytvářecím formuláři. Zobrazí se mu tak detailní informace, jako například transformační matice nebo nahrané soubory. Tato stránka bude velice podobná formuláři na obrázku 3.4. Také zde budou akce, které může autor se svým modelem provádět – například přejít na úpravu modelu, přejít do tvorby nové verze, dokončit tento model a podobně.

V druhé formě (**M4**) bude umožněno zobrazit si 3D objekt ve 3D scéně. Tato stránka je navržena tak, aby neobsahovala zbytečné rušivé informace, bude tedy obsahovat pouze základní data o 3D objektu, 3D scéně a několik tlačítek, které budou odkazovat na důležité akce s modelem. Akce budou například smazat, upravit nebo přejít na detail. Obrázek s návrhem stránky nebude pro její jednoduchost uveden. Nejlépe se dá přirovnat k obrázku 3.6, kde jsou také uvedeny základní informace o 3D modelu a 3D scéně. Místo informací o iteraci zde však budou navigační tlačítka.

Přehled schvalovacích procesů (M5) bude stránka, kde se zobrazí veškeré procesy, jichž je uživatel členem. Tato stránka bude primárně sloužit k tomu, aby si modelář mohl zobrazit stav schvalování modelů, které odeslal ke schválení. Dle návrhu schvalovacího procesu z kapitoly 3.1 tedy každý záznam na stránce bude odpovídat jedné **iteraci**, která se váže na jeden 3D objekt. Stránka je navržena na obrázku 3.5. V návrhu je zohledněno to, že informací o iteraci je spousta a musí být zváženo, jaké informace budou zobrazeny v tomto stručném přehledu.

Stránka pro zobrazení iterace (M6) bude sloužit jako přehled veškerých důležitých informací o jedné iteraci procesu, která se váže na jeden odevzdaný 3D objekt. Pro modeláře bude mít pouze informativní charakter. Bude obsahovat zobrazení 3D objektu ve 3D scéně, důležité informace o iteraci a také aktuální stav hodnocení. Stránka je navržena na obrázku 3.6.

3.2.3 Stránky schvalovatele

Zde budou uvedeny stránky, které budou přístupné uživatelům s rolí **historian** nebo **approver**, popořadě tedy **historik** a **grafik**.

Navigační tlačítka

Role username Odhlásit

Tvorba a úprava modelu

Název modelu Model rok 1500 Struktura Pevnost 1 ▾ Soubor Nahrát...

Verze 1 0 0

Transformace

Translace 0 0 0 Scale 1 1 1 Rotace 0 0 0 1

Vynulovat

Textury Assety Tagy

Soubor Nahrát... ✕ Soubor Nahrát... ✕ Tag 1 Hodnota 1 ✕

Přidat Přidat Přidat

Akce

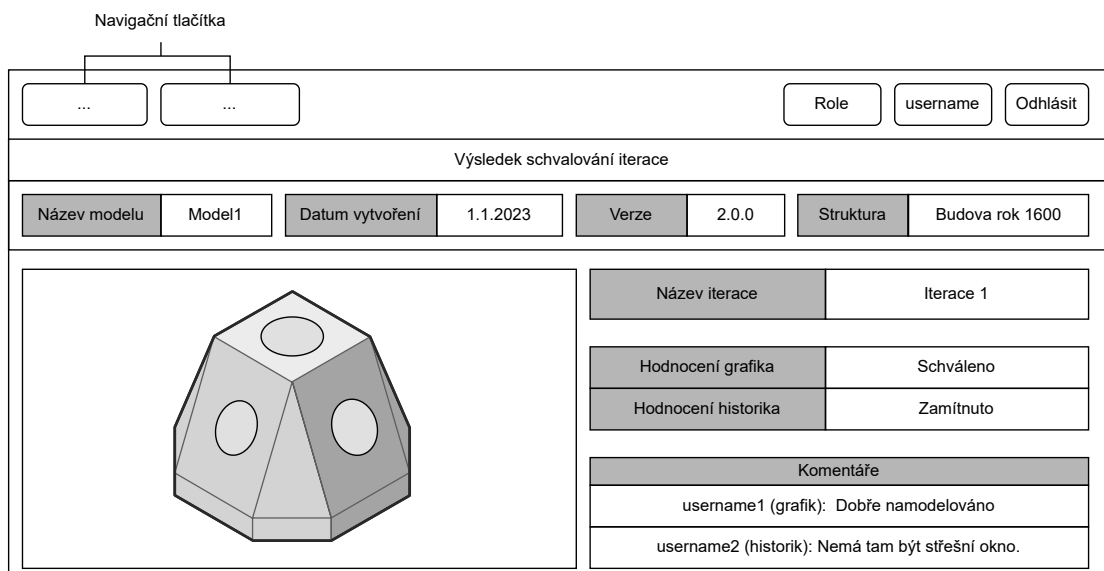
Uložit

Tlačítka s akcemi

■ **Obrázek 3.4** Návrh obrazovky modeláře – formulář pro tvorbu nebo úpravu 3D objektu



■ **Obrázek 3.5** Návrh obrazovky s modely ve schvalovacím procesu



■ **Obrázek 3.6** Návrh obrazovky se stavem schvalovací iterace

Přehled úkolů (S1) bude stránka, kde uživatel uvidí veškeré modely, které čekají na jeho ohodnocení. Konkrétně se tedy jedná o jednotlivé iterace schvalovacích procesů, jichž je uživatel členem a které ještě nebyly ohodnoceny rolí uživatele. Také se zde nezobrazí iterace zamítnuté druhou rolí, které jsou již ve stavu **zamítnuto**. Návrh vzhledu stránky bude téměř stejný jako **přehled schvalovacích procesů** u modelářů, tudíž jej lze přirovnat k návrhu na obrázku 3.5.

Přehled schvalovacích procesů (S2) bude stránka, kde uživatel uvidí veškeré procesy, jichž je členem. Budou zde tedy veškeré iterace z těchto procesů, nehledě na to, zda jsou schváleny, zamítnuty nebo neohodnoceny. Vzhled bude opět velice podobný návrhu na obrázku 3.5.

Stránka pro ohodnocení iterace (S3) se zobrazí po výběru konkrétní iterace ze seznamu úkolů nebo přehledu procesů. Uživatel si zde může prohlédnout 3D objekt, případně komentáře, které byly k této iteraci odeslány.

Hlavním účelem stránky je, aby schvalovatel ohodnotil tuto iteraci. Bude zde tedy malý formulář pro ohodnocení – bude potřeba zvolit rozhodnutí **Schvaluji** nebo **Zamítám** a bude možná přidat k hodnocení poznámky. Po ohodnocení iterace zmizí z fronty úkolů tohoto uživatele, bude však stále dohledatelná v přehledu.

Stránka bude téměř stejná jako stránka na obrázku 3.6. Rozdílem bude to, že schvalovatel uvidí schvalovací formulář. Do něj bude vyplňovat své rozhodnutí – **Schvaluji** nebo **Zamítám** – a případně komentáře, které chce k iteraci přidat.

Stránka s ohodnocenou iterací (S4) bude podobná stránce pro přehled iterace, kterou má uživatel s rolí modelář (3.6). Schvalovatel navíc dostane možnost vzít zpět své rozhodnutí, pokud tuto iteraci ohodnotil on.

3.2.4 Stránky administrátora

Zde budou uvedeny stránky, které budou poskytnuty uživatelům s rolí **admin**, v tomto kontextu tedy **administrátor**.

Stránka s modely k publikaci (A1) zobrazí všechny schválené modely, které čekají na publikaci na veřejné API. Z návrhu procesu tedy vyplývá, že se bude jednat o iterace, které byly schváleny. Stránka bude v návrhu podobná jako přehled iterací z obrazovek pro schvalovatele (obrázek 3.5). Není zde již potřeba indikátorů stavů schvalování, jelikož se zde zobrazí pouze schválené iterace.

Stránka pro publikaci modelu (A2) umožní uživateli zobrazit si schválený model (iteraci) a rozhodnout, zda jej bude publikovat. Bude potřeba zobrazit základní informace o modelu, 3D scénu s modelem a základní informace o iteraci. Také bude vhodné zobrazit, kdo iteraci schválil a jaké byly přidány komentáře. Pro publikaci zde bude tlačítko, které umožní označit model za veřejný.

Stránka je v návrhu velice podobná návrhu na obrázku 3.6. Pouze zde budou informace navíc o jednotlivých schvalovatelích a akční tlačítko.

Přehled veřejných modelů (A3) bude stránka, kde se opět podobným způsobem zobrazí 3D objekty (iterace), které byly schváleny a poté publikovány.

Návrh je opět podobný, jako přehled iterací u modelářů či schvalovatelů (3.5). Stejně jako u publikovatelných modelů zde nebude potřeba mít indikátory stavu schvalování, bude zde však datum publikace.

Stránka pro zrušení publikace modelu (A4) bude přístupná pomocí otevření vybrané iterace na stránce s přehledem veřejných modelů. Administrátor zde může vzít zpět publikaci.

Návrh stránky je podobný jako u **publikace modelu**, bude zde však umožněno provádět „privatizaci“ místo publikace.

3.3 Pokrytí funkčních požadavků

V této kapitole bude provedena kontrola, zda navržená aplikace pokrývá veškeré uživatelské cíle a funkční požadavky. Bude zde využito označení jednotlivých stránek z kapitoly 3.2 (tedy například M1 označuje stránku modeláře definovanou jako první v pořadí). Kontrola pokrytí funkčních požadavků bude znázorněna pomocí tabulky.

Řádky tabulky označují jednotlivé role (uživatelskou skupinu pro schvalovatele) – N pro nepřihlášen, M pro modelář, S pro schvalovatel, A pro administrátor. Sloupce tabulky označují jednotlivé funkční požadavky. Zkratky v buňkách označují stránky, které splňují tento požadavek pro určenou roli (případně skupinu). Pokud se tento požadavek na tuto skupinu neaplikuje a není žádoucí jej plnit, je buňka vyplněna pomlčkou. Pokud aplikace požadavek splní, ale splní jej automaticky v návaznosti na akce uživatele, buňka je vyplněna textem „Automaticky“. Tabulka je pro přehlednost rozdělena do dvou menších tabulek, znázorněných zde: 3.1 a 3.2.

	F1.1	F1.2	F1.3	F2.1	F2.2	F2.3
N	O1, O2, O3	O1, O2, O3	Automaticky	-	-	-
M	-	-	-	M1, M2, M3, M4	M3	M1
S	-	-	-	-	-	-
A	-	-	-	-	-	-

■ **Tabulka 3.1** Tabulka s kontrolou pokrytí funkčních požadavků F1 a F2

	F3.1	F3.2	F3.3	F3.4	F3.5	F3.6	F3.7	F4.1	F4.2	F4.3	F4.4
N	-	-	-	-	-	-	-	-	-	-	-
M	M3	-	-	M5, M6	M5	-	Automaticky	-	-	-	-
S	-	S3	S4	MS1, S2, S3, S4	S2	S1	Automaticky	-	-	-	-
A	-	-	-	-	-	-	-	A1	A2	A3	A4

■ **Tabulka 3.2** Tabulka s kontrolou pokrytí funkčních požadavků F3 a F4

Kapitola 4

Realizace

V této kapitole bude rozebrána implementace výsledné aplikace. Nejdříve budou rozebrány použité technologie a odůvodněna rozhodnutí týkající se jich. Dále budou uvedeny změny provedené na Private API projektu VMCK. Následně budou popsány veškeré stránky aplikace a budou předvedeny obrázky vybraných stránek.

Na konec budou rozebrány funkcionality, které se během implementace nepodařilo splnit a důvody neúspěchu.

4.1 Technologie

V této kapitole budou rozebrány technologie, které byly použity během implementace. Pokud u technologií došlo k nějaké volbě, budou zde uvedeny důvody k této volbě.

4.1.1 React JavaScript VS React TypeScript

V analýze byly rozebírány tyto dvě technologie jako hlavní nástroj pro implementaci. V implementaci byla jako jazyk zvolena možnost **React TypeScript**.

Důvody k volbě byly následující. Typescript je bezpečnější varianta programovacího jazyka, za cenu pomalejšího učení a studia navíc. Přidává však do aplikace například datové typy a statickou kontrolu typování.

Typy v jazyku TypeScript umožní této aplikaci být bezpečnější, lépe přehlednou, organizovanou a do budoucna také rozšiřitelnou, až už autorem práce nebo následníky. Důvodem je také to, že backend projektu je psán v Node.js s nadstavbou TypeScript. To znamená, že kdyby bylo potřeba tuto aplikaci upravovat po zakončení této práce, vývojáři backendu budou mít lepší předpoklady ji udržovat a upravovat.

4.1.2 Create React App

Create React App je nástroj, který je dodáván samotnou společností Meta Platforms, dříve Facebook, která spravuje a dodává samotný React. Je to tedy oficiální, udržovaný a spolehlivý nástroj, jehož úkolem je vývojářům usnadnit tvorbu nových SPA (Single Page Application) v Reactu.

Tento nástroj umí pomocí jednoho příkazu vytvořit funkční jednoduchou aplikaci a odstínit vývojáře od složitého nastavování složek a souborů, od instalací potřebných nástrojů a jejich konfigurace. Při použití nástroje vznikne aplikace s připraveným jednoduchým App.js / App.ts

(mimo spoustu jiných souborů), kde už lze dle libosti psát kód. V základní konfiguraci souborů jsou obsaženy i základní scripty pro důležité akce, jako například `npm start` pro spuštění vývojového serveru, defaultně na adrese `http://localhost:3000`, nebo `npm run build` pro sestavení aplikace. [28]

Použití Create React App je velice jednoduché. Stačí ve zvolené složce v terminálu zavolat příkaz `npx create-react-app my-application`, kde název vytvořené aplikace bude stejný jako poslední prvek příkazu, zde by to bylo `my-application`. Příkaz `npx` reprezentuje nástroj **Node Package eXecute**, který je součástí instalace knihovny `npm`, tedy **Node Package Manager**. Oba nástroje jsou obsaženy v instalaci `Node.js`. Příkaz `npx` nám umožňuje použít nástroj `create-react-app` bez toho, aby bylo nutné jej pokaždé instalovat a mazat, jak by tomu muselo být při použití obyčejného `npm`.

Typescript byl zvolen jako jazyk pro implementaci aplikace. Nástroj Create React App našťástí počítá s tím, že vývojáři v Reactu budou chtít tento jazyk používat. Do aplikace lze přidat TypeScript jak při vytváření, tak kdykoliv později. Pro přidání TypeScriptu hned při tvorbě aplikace lze použít příkaz `npx create-react-app my-application --template typescript`. Pokud je potřeba jej doinstalovat do již existující aplikace, lze provést příkaz `npm install --save typescript @types/node @types/react @types/react-dom @types/jest`. [29]

4.1.3 EmailJS

Pro implementaci emailových upozornění byla vybrána knihovna a služba EmailJS [30]. Tato knihovna a online služba poskytuje možnost snadno a rychle vytvořit emailovou službu pro frontendové aplikace v JavaScriptu, případně TypeScriptu. Také má vstřícnou cenovou politiku – na účtu zdarma umožňuje vytvořit až dvě emailové šablony a poslat až 200 emailů měsíčně.

Instalace knihovny a nastavení služby je jednoduchá. Autor práce použil jako nápomocné zdroje k instalaci a nastavení dokumentaci knihovny [31], také online video návod [32] a online fórum [33]. Bude zde uveden stručný postup, jak nainstalovat tuto knihovnu do vlastní React aplikace. Předpokladem pro použití služby je disponovat vlastním emailovým účtem u jedné z podporovaných domén – nejznámější poskytoval je například Google a jeho **Gmail**.

- **Instalace knihovny** do existujícího projektu je provedena pomocí příkazu:


```
npm install @emailjs/browser --save.
```
- **Nastavení emailové služby** lze provést na stránce <https://www.emailjs.com/>. Zde je potřeba založit uživatelský účet pod vybraným emailem. Dále je potřeba provést následující akce:
 - Nejdříve je potřeba **vytvořit novou službu**. Zde je proveden výběr poskytovatele (například Gmail), připojení služby k vybranému emailu a výběr názvu služby.
 - Nyní lze získat `service_id`, identifikátor služby.
 - Nyní je potřeba vytvořit emailovou **šablonu**. Do šablony je možno posílat z frontendové aplikace proměnné zabalené do `props` objektu. V šabloně tyto proměnné lze použít takto: `{{prop_name}}`. Takto lze do šablony vyplnit i proměnný email odesílatele a příjemce.
 - Z vytvořené šablony lze získat `template_id`, její identifikátor.
 - V nastavení uživatelského účtu lze získat `public_key`.
- **Import** funkcionalit knihovny `emailjs` do komponenty v kódu je proveden pomocí příkazu `import emailjs from '@emailjs/browser'`.

- **Odeslání emailu** je provedeno pomocí volání `emailjs.send(service_id, template_id, emailProps, public_key)`. Parametry `service_id`, `template_id` a `public_key` získáme z nastavení emailové služby z dřívějšího bodu.

Pro účel požití této knihovny byl vytvořen Google účet `vmck.notifications@gmail.com`. Přihlašovací údaje k tomuto emailu a do účtu EmailJS budou poskytnuty vedoucímu práce.

4.1.4 React OAuth2 | Google

Pro Google autentifikaci, přihlašování a odhlašování byla použita knihovna **React OAuth2** pro Google autentifikaci. Knihovna se instaluje pomocí příkazu:

```
npm install @react-oauth/google@latest.
```

Z této knihovny byly využity tyto prvky:

- `GoogleAuthProvider` – poskytovatel, do kterého je potřeba zabalit aplikaci.
- `GoogleLogin` – hotová komponenta, která má podobu tlačítka, které automaticky zobrazí vyskakovací přihlašovací okno a po úspěšném přihlášení vrátí user credentials a ověřovací údaje.
- `googleLogout` – funkce, která zařídí Google odhlášení uživatele.
- `CredentialResponse` – interface, který definuje podobu odpovědi od úspěšného průběhu přihlášení pomocí `GoogleLogin`.

4.1.4.1 Nastavení

Pro použití těchto komponent v kódu je potřeba připravit aplikaci pro Google autentifikaci. Část postupu již byla popsána v práci Dominika Siváka. Zde bude uveden vlastní popis, jak umožnit Google autentifikaci v React frontendové aplikaci.

Google console je prostředí poskytované společností Google, které je potřeba využít pro registraci naší aplikace jako entity využívající Google OAuth ověřování. Je potřeba se na adrese <https://console.cloud.google.com/> přihlásit přes vlastní Google účet a vytvořit nový projekt pro webovou aplikaci.

OAuth consent screen je potřeba konfigurovat proto, aby byly uživatelům pomocí vyskakovacího okna předány informace, kam se pokouší přihlásit a jaké informace budou zpracovávány. V Google console v záložce **OAuth consent screen** je potřeba nakonfigurovat projekt.

Client ID potřebuje aplikace k tomu, aby byla jednoznačně identifikovatelná jako uživatel Google OAuth a aby pomocí něj Google autorita mohla ověřovat přihlašovací tokeny. V okně **Credentials** je možno své Client ID založit. Je potřeba vyplnit informační formulář. Je důležité správně vyplnit pole **Authorized JavaScript origins**, kam je potřeba napsat URI, kde bude hostována aplikace. V základním nastavení aplikace od Create React App aplikace běží na portu `http://localhost:3000`, pokud by aplikace sídlila na veřejné adrese, tak by bylo potřeba ji zde doplnit. Pro tuto aplikaci to bude adresa `https://vmck-dev.sic.cz:8000/`. Po úspěšném vyplnění formulářů bude mít aplikace vlastní Client ID, které uvidíme na stránce Credentials. Toto Client ID je potřeba v kódu předat do `GoogleAuthProvider`.

4.1.4.2 Propojení s Private API

Propojení lokálního Google přihlašování ve webové aplikaci s Private API projektu VMCK je poslední krok k tomu, aby bylo možné přihlašovat uživatele registrované v databázi projektu. Postup je zde uveden proto, aby potenciální následníci měli jednoduchý návod, jak propojení zařídit.

Registrace Client ID je důležitý první krok. Je potřeba přidat Client ID do backendu aplikace, aby API přijímalo požadavky o autentizaci přicházející ze stránky aplikace. Tento krok obnáší kontaktovat někoho z vývojářů, kteří spravují backend projektu, a předat jim své Client ID. Po aktualizaci backendu bude aplikace připravena na autentifikaci uživatelů.

Registrace a přihlášení uživatelů je hlavním cílem celého procesu s Google autentizací. Na obojí se používá stejné API volání, metoda POST odesílaná na endpoint `/auth/googlelogin`. Součástí musí být tělo, ve kterém je obsaženo `clientId` s hodnotou Client ID webové aplikace, a `token`, kde hodnotou musí být platný token uživatele, který nám vrátil Google při přihlášení a ověření uživatele. Pokud tento účet není registrován v databázi projektu, bude po tomto volání vytvořen s rolí `commonUser`, tedy modelář, a API volání vrací objekt informacemi o uživateli a jeho ověřovací token. Pokud uživatel již existuje, nedojde k žádným změnám, pouze se vrátí stejný objekt jako u registrace a uživatel bude přihlášen v aplikaci.

4.1.5 Litepřevodce a IFrame

Jako nástroj pro zobrazování 3D scény byl vybrán litepřevodce projektu VMCK z kapitoly 2.5. Důvodem je, že poměr mezi funkcionalitami a jednoduchostí použití byl nejlepší mezi zobrazovacími nástroji.

Tento nástroj lze použít pomocí vložení stránky `view.html` do technologie IFrame (také Inline Frame). Stránka `view` bere jako jeden z parametrů v URL i ID 3D objektu uloženého na API, a pokud tento objekt podle ID nalezne, tak zobrazí scénu s vybraným 3D objektem.

IFrame je HTML element, který umožňuje zobrazit přímo obsah jiné webové stránky [18]. Použití je jednoduché, stačí do aplikace nainstalovat `react-iframe`, z této knihovny získat element `Iframe` a do HTML/JSX/TSX kódu jednoduše vložit tag `<Iframe ... />`. Do parametrů komponenty je potřeba doplnit URL cílové stránky, také například volitelné parametry jako rozměry nebo parametry zobrazení.

4.2 Private API

Pro potřeby vývoje bylo založeno nové API. Proběhla kopie live API na adrese `https://vmck.sic.cz:3443` a příslušné databáze. Byly zkopírovány funkcionality a data z tohoto API a umístěny na nové DEV (vývojové) API, které je dostupné na adrese `https://vmck-dev.sic.cz:3443/`. Stejně jako původní API je i toto nové chráněno přístupovými tokeny.

Toto nové API bylo během vývoje používáno a testováno. Pokud byly nalezeny chyby, vytvořeny připomínky nebo proveden návrh na změnu, byl vytvořen ticket v ticketovacím systému Redmine. Tímto způsobem jsou návrhy a úkoly zorganizovány a je jim přidělena priorita. Také i kdyby se některé úkoly nestihly splnit v termínu této práce, budou i nadále zdokumentovány a nebudou zapomenuty.

4.2.1 Problémy

Během testování a používání backendu bylo objeveno několik problémů. Všechny jsou zdokumentovány v systému Redmine. Některé chyby však přímo ovlivňují správnost chodu této aplikace.

Zde budou uvedeny některé kritické problémy a jak se k nim aplikace zachová.

Chyba v právech modeláře je naprosto stěžejní pro funkci aplikace. Na backendu totiž uživatel s rolí **modelář** nemá práva na nahrávání 3D objektů ke strukturám. Jedná se o veškerá POST volání, které se týkají nahrávání souborů a dat. Pro testování musela být používána jiná role než modelář.

Dále modeláři chybí práva na mazání verzí 3D objektu. Dle návrhu má autor 3D objektu mít práva na smazání nejnovější verze 3D objektu. Práva na jakákoliv mazání má však aktuálně pouze administrátor. Pro účely implementace a testování byla opět využívána jiná uživatelská role.

Oba problémy budou automaticky vyřešeny, až se opraví backend.

Chyba ve vytváření 3D objektu je dalším problémem při implementaci. Při vytváření 3D objektu totiž API neregistruje poslané parametry **transformation** a **transformation web**. Ať tato pole jsou nebo nejsou poslána v těle requestu, backend je nastaví na defaultní hodnoty.

Tato chyba v aplikaci není řešena, jelikož si vedoucí práce nepřál tento problém obcházet. Tělo requestu se tedy posílá tak, jako kdyby volání fungovalo korektně. Zmíněné dva parametry se však kvůli chybě backendu neuloží.

Tato chyba bude automaticky opravena, až se opraví backend projektu.

Chyba v úpravě 3D objektu se také týká parametrů **transformation** a **transformation web**. Tyto parametry se aktuálně z uloženého 3D objektu smažou a přenastaví na „{}“, pokud nejsou parametry posílány v každém PUT requestu.

Řešení je v aplikaci takové, že se tyto parametry posílají v každém PUT requestu. Není to však efektivní řešení a bylo by vhodné jej upravit, až se opraví backend projektu.

Finalizace iterací v projektu prošla změnou návrhu. Schvalovací proces je aktuálně nastaven tak, že může existovat pouze jeden proces pro jednu strukturu, mělo by být ale možné v procesu finalizovat více modelů – publikovat více 3D objektů pro jednu strukturu. Právě finalizace více iterací na backendu není implementována a bude potřeba ji implementovat tak, aby pro schvalovací proces bylo možné finalizovat více iterací.

Tato funkcionalita bude v aplikaci fungovat hned, jakmile bude opraven backend.

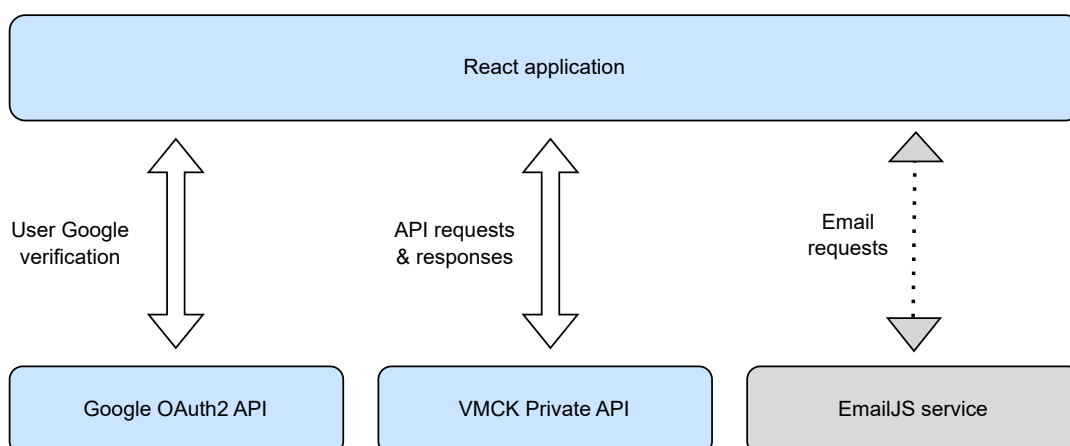
Chyba při vrácení hodnocení nastává tehdy, když je uživatel v roli historik a nastane určitý scénář. V určité situaci se stane, že jakmile se historik rozhodne vzít zpět své rozhodnutí, API mu odpoví, že to nelze. Tuto chybu bude potřeba opravit pro správnou funkci aplikace.

Tato funkcionalita bude v aplikaci fungovat hned, jakmile bude opraven backend.

4.3 Schéma komunikace aplikace

Aplikace komunikuje s externími službami. Pro lepší přehled zde tyto služby budou uvedeny a komunikace bude znázorněna na diagramu 4.1.

VMCK Private API Aplikace komunikuje s Private API projektu Věnná města českých královen. Komunikace s tímto API poskytuje veškerá potřebná data v projektu, například data uživatele, struktur nebo 3D objektů. Aplikace také komunikuje s Private API když požaduje přihlášení, nahrání nových dat nebo aktivaci akcí. Tato komunikace je naprosto stěžejní pro běh aplikace.



■ **Obrázek 4.1** Komunikační schéma aplikace a externích služeb – modelováno v Diagrams.net [20]

Google OAuth2 API Aplikace komunikuje se službou pro Google autentizaci. K této komunikaci jsou používány React komponenty poskytnuté samotným Google.

Služba v aplikaci slouží k tomu, aby přijímala požadavky na Google přihlášení, ověřila požadavek a zaslané uživatelské údaje a nakonec vrátila aplikaci ověřené **User Credentials**.

EmailJS service V aplikaci byly implementovány základy pro zasílání emailových upozornění. K tomu byla zvolena technologie EmailJS. Pro použití tohoto nástroje je potřeba konfigurovat EmailJS službu a také s ní poté komunikovat, když aplikace požaduje zaslání emailu.

Tato služba ve finální aplikaci není aktivně používána, i když základy jsou v aplikaci implementovány. Důvody jsou uvedeny v kapitole o nedokončených funkcionalitách (4.5). Z tohoto důvodu je služba v diagramu 4.1 znázorněna šedě.

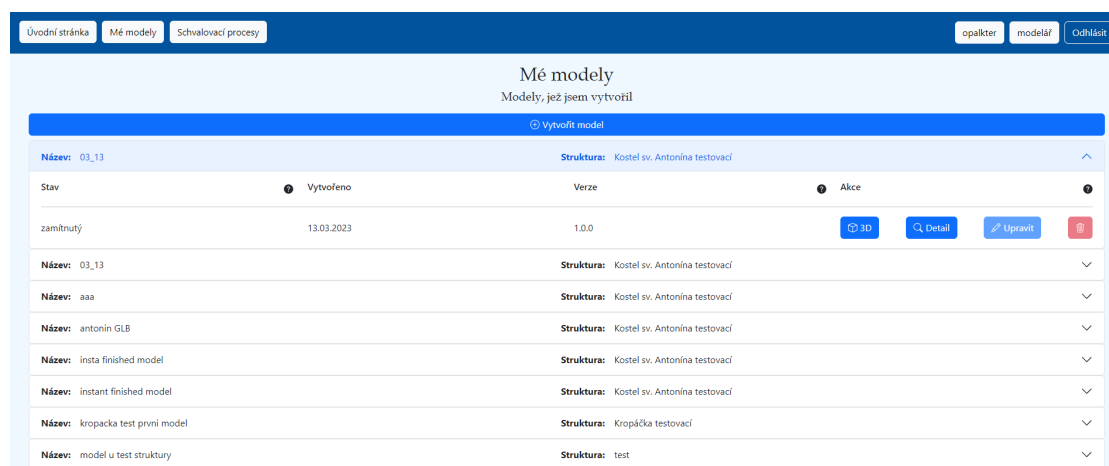
4.4 Stránky webové aplikace

V této kapitole budou rozebrány a předvedeny stránky aplikace implementované podle návrhu v kapitole 3.2. Pro přehlednost bude tato kapitola opět rozdělena podle uživatelských rolí nebo skupin. Stránky budou předvedeny v takovém pořadí, v jakém byly navrženy v kapitole 3.2. Některé obrazovky nebudou uvedeny v textu, jelikož jsou navzájem vizuálně podobné. Takové stránky budou uvedeny v přílohách.

4.4.1 Stránky společné všem uživatelům

Zde jsou uvedeny stránky a funkcionality aplikace, jež jsou dostupné všem uživatelům aplikace. Jedná se tedy o uživatele přihlášené i nepřihlášené, a v případě přihlášených uživatelů o libovolnou uživatelskou roli. Stránky mohou mít různou podobu v závislosti na stavu uživatele.

Úvodní obrazovka je stránka, která sídlí na route /. Pro **nepřihlášené** uživatele zobrazuje základní informace o aplikaci a také žádost o přihlášení. Pro **přihlášené** uživatele zobrazuje informace o aplikaci, základní informace o navigaci v aplikaci a také informace o to, jaké akce můžou jednotlivé role v aplikaci provádět. Stránka pro svou jednoduchost nebude vložena jako obrázek.



■ **Obrázek 4.2** Obrazovka s přehledem modelů modeláře

Navigační menu se zobrazuje různě podle stavu uživatele. Je implementováno podle provedení návrhu. Podrobné popisy stránek budou uvedeny v následujících kapitolách.

Pro **nepřihlášené uživatele** zobrazuje pouze tlačítko pro Google přihlášení. Pro **přihlášené uživatele** zobrazuje tlačítko Odhlásit, také pole s uživatelským jménem a projektovou rolí. Dále má každá role vlastní sadu navigačních tlačítek, tlačítko společné všem je **Úvodní stránka**.

Pro **modeláře** jsou zobrazena tlačítka **Mé modely** a **Schvalovací procesy**. První tlačítko odkazuje na veškeré modely, jež uživatel uložil. Druhé tlačítko odkazuje na výčet všech schvalovacích procesů, jichž je uživatel členem a má práva je zobrazit.

Pro **schvalovatele** jsou zobrazena tlačítka **Čeká na schválení** a **Schvalovací procesy**. První tlačítko odkazuje na přefiltrovaný seznam schvalovacích procesů a iterací. Také je zde políčko s číslem, které označuje počet čekajících úkolů. Druhé tlačítko odkazuje na výčet všech schvalovacích procesů, jichž je uživatel členem a má práva je zobrazit.

Pro **administrátora** jsou zobrazena tlačítka **Čeká na publikaci** a **Veřejné modely**. První tlačítko odkazuje na přefiltrovaný seznam schvalovacích procesů a iterací, kde se zobrazí pouze schválené iterace. Druhé tlačítko odkazuje na podobný seznam, kde jsou pro změnu pouze finalizované iterace.

Google přihlášení je provedeno pomocí tlačítek v navigačním menu. Nepřihlášený uživatel se může pomocí tlačítka přihlásit do aplikace. Registrace nových uživatelů probíhá automaticky na backendu projektu. Uživatel přihlášený se může odhlásit.

4.4.2 Stránky modeláře

Zde budou předvedeny stránky, které byly implementovány pro uživatele s rolí modelář.

Stránka s modely obsahuje nadpis, tlačítko pro tvorbu nového modelu a seznam uložených 3D objektů. Každý prvek v seznamu je rozbalovací položka, kde v zabalené podobě jsou vidět jen nejdůležitější informace, a po kliknutí na položku se objeví detailnější informace a akční tlačítka. V rozbalené položce jsou doplněny nápovědy – malé otazníčky, které „on hover“ zobrazí vyskakovací okénko s nápovědou. V nápovědě se uživatel dozví například to, co tlačítka dělají, kdy jsou tlačítka neaktivní, kam budou uživatele navigovat a co uživatel nalezne na cílové stránce. Stránka je zobrazena na obrázku 4.2.

■ **Obrázek 4.3** Obrazovka s tvorbou 3D objektu

Formuláře pro vytvoření nebo úpravu 3D objekt byly implementovány s malými rozdíly oproti návrhu. Důvodem je, že se změnila některá povinná a nepovinná pole u entity 3D objekt. Některé tyto rozdíly nebyly nijak zdokumentované a byly zjištěny testováním a konzultacemi. Některé rozdíly vyplývají z nedorozumění mezi autorem a dokumentací.

Například **verze** modelu je při vytváření vždy 1.0.0. Dále při úpravě tuto verzi nelze měnit, jelikož uživatel v tu chvíli upravuje jednu vybranou verzi. Také u tvorby nové verze nemá smysl verzi zadávat, jelikož číslo verze se generuje automaticky.

Dále přibyla ve formuláři **transformační síť**, což je objekt stejného typu jako **transformace**. Tento atribut je u 3D objektů definovaný dle slov vedoucího pouze dočasně, nyní je však potřeba mít možnost jej zadat.

Dále u **textur** proběhly změny. Při implementaci bylo zjištěno, že vytváření textur je v nesouladu s dokumentací. V aktuální implementaci je totiž potřeba pro každou texturu vybrat její typ a odeslat jej společně se souborem.

Dále se změnil **tagy**, neboli properties (vlastnosti). Přidávání properties není na backendu implementováno, navzdory tomu, že jsou popsány v dokumentaci.

Tlačítka u formulářů byla implementována následovně:

- **Formulář pro tvorbu nového modelu** obsahuje tlačítka dvě. První tlačítko pro uložení modelu – to uloží model do databáze, společně s jeho soubory, a model bude ve stavu **unfinished**. Druhé tlačítko je **Uložit a dokončit**, které uloží model do databáze a označí jej jako **finished**.
- **Formulář pro úpravu existujícího modelu** obsahuje tlačítka tři. První dvě tlačítka jsou stejná jako v předchozím bodě. Třetí tlačítko je **reset hodnot**, které nastaví formulářová pole do takového stavu, aby odpovídala objektu před začátkem úprav.

Formulář pro tvorbu nového 3D objektu je zobrazen na obrázku 4.3.

Detail uloženého modelu byl dle návrhu implementován ve dvou formách – ve formě 3D scény s modelem a základními informacemi, a ve formě formuláře, kde se zobrazí data tak, jak byla odeslána při uložení. Druhá forma také obsahuje většinu akčních tlačítek.

Detail s 3D modelem je stránka, kde uživatel vidí stručný přehled informací o 3D objektu a 3D scéně. 3D scéna je Také zde vidí akční tlačítka **Detail**, **Uložit** a **Smazat**. **Detail** odkazuje na formulářový detail, **úprava** na formulář s úpravou a **smazat** umožní smazat tento model. Tlačítka Upravit a Smazat však podléhají pravidlům, kdy je lze a kdy nelze použít. Například

upravit lze pouze model ve stavu **unfinished** a smazat lze pouze nejnovější verzi a pouze model neodeslaný ke schválení.

Detail s formulářem vypadá velice podobně jako **formulář pro tvorbu nového modelu**. Některá pole byla zaměněna za jiná a například soubory byly pro přehlednost schovány pod vlastní tlačítko **Soubory**. Tento formulář však obsahuje většinu akčních tlačítek, kterou modelář u modelu dostává. Tlačítka opět podléhají kontrole, zda je možno je provést. Pokud ne, jsou tlačítka neaktivní. Tlačítka jsou popsána níže:

- **Upravit** přesměruje uživatele do formuláře pro úpravu 3D objektu.
- **Nová verze** otevře uživateli formulář pro tvorbu nové verze tohoto 3D objektu. Některá pole ve formuláři jsou neměnná – například název nebo strukturu již nelze měnit, jelikož tato pole jsou definovaná z předchozí verze. Akční tlačítka jsou zde **Uložit** a **Uložit a dokončit**.
- **Dokončit** dá do pohybu akce, které označí tuto verzi modelu za dokončenou. Dokončení 3D objektu je nevratná akce a model již nebude možno upravovat.
- **Odeslat** otevře okno, kde uživatel zadá informace pro odeslání 3D modelu do schvalování. Po potvrzení aplikace odešle požadavek a odešle 3D model do schvalovacího procesu v nové iteraci.
- **Soubory** otevře okno, kde uživatel vidí všechny uložené soubory a je mu umožněno si je stáhnout.
- **Smazat** umožňuje smazat vybranou verzi 3D objektu.

Přehled schvalovacích procesů byl také implementován dle návrhu. Zobrazují se zde veškeré schvalovací procesy, jichž je uživatel členem. Pro přehlednost jsou použita rozbalovací pole – každý řádek reprezentuje jeden proces. Po rozbalení se zobrazí dostupné iterace a legenda. Obrazovka je téměř identická jako **procesy schvalovatele** nebo **úkoly schvalovatele** (obrázek 4.4). Pouze zde nejsou iterace filtrovány.

Stránka pro zobrazení iterace byla implementována tak, že zobrazí uživateli 3D scénu, základní informace o obsaženém 3D modelu a důležité informace o vybrané iteraci. Stránka vypadá téměř identicky jako **stránka pro hodnocení iterace** na obrázku 4.5. Rozdílený je pouze formulář napravo od scény – místo formuláře pro odeslání se zobrazují informace o iteraci.

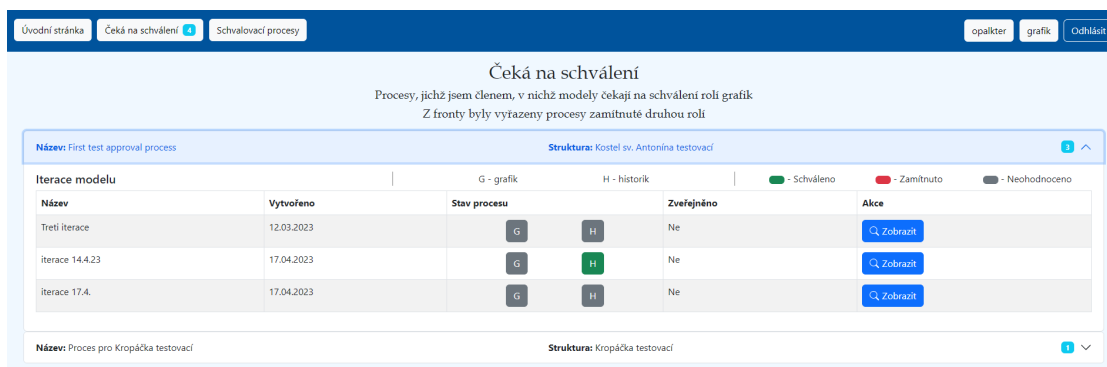
4.4.3 Stránky schvalovatele

Zde budou předvedeny stránky, které byly implementovány pro uživatele s rolí historik nebo grafik, tudíž uživatelskou skupinu **schvalovatel**.

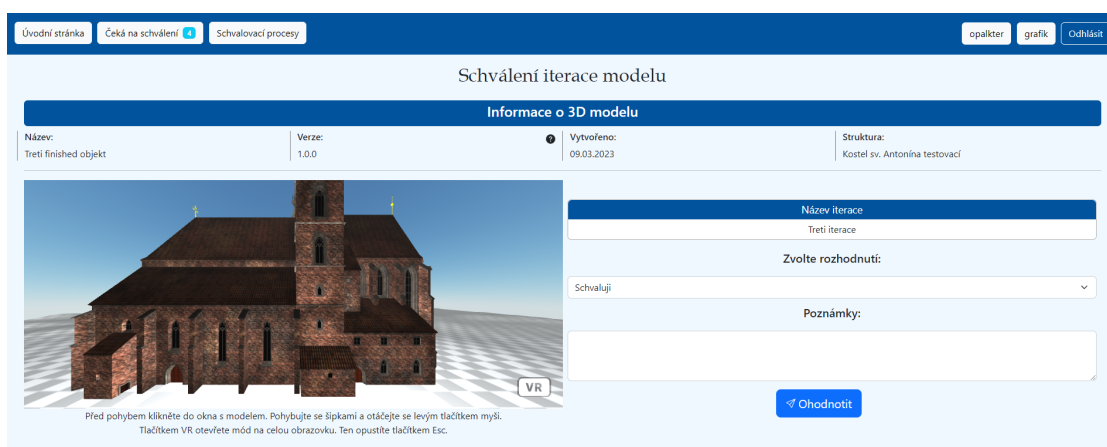
Přehled úkolů je stránka s přefiltrovanými schvalovacími procesy uživatele. Zobrazí se zde pouze ty iterace, které čekají na hodnocení rolí uživatele a zároveň nejsou zamítnuté druhou autoritou. Dále pokud v procesu žádná taková iterace není, proces se nezobrazí.

Každý **proces** se dle implementace backendu váže na jednu strukturu, tento proces tedy obsahuje veškeré odevzdané modely pro vybranou strukturu (zadání). Zde u úkolů se iterace a procesy filtrují dle kritérií výše.

Počet iterací, které čekají na ohodnocení, se projevuje jako barevné upozornění v navigačním menu a v rádcích pro procesy. V navigačním menu se u tlačítka „Čeká na schválení“ zobrazuje barevné políčko, kde číslo uvnitř znamená celkový počet iterací čekajících na schválení. Dále v každém řádku pro proces je podobné upozornění, kde číslo označuje počet iterací v tomto procesu, které čekají na schválení.



Obrázek 4.4 Obrazovka s přehledem úkolů schvalovatele



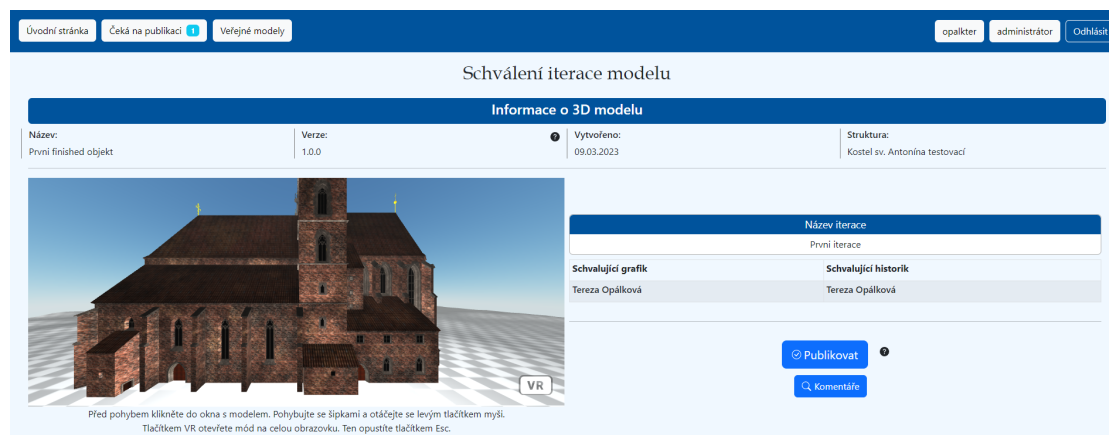
Obrázek 4.5 Obrazovka s formulářem pro hodnocení iterace

U každé iterace jsou zobrazeny důležité informace o iteraci a také tlačítko, které uživatelé naviguje na stránku pro ohodnocení iterace. Důležitou zobrazenou informací je například indikátor stavu schvalování – rozhodnutí jednotlivých autorit. Stránka je zobrazena na obrázku 4.4. Lze zde vidět i navigační menu, kde se zobrazuje počet úkolů.

Přehled schvalovacích procesů je zobrazen na stránce, kde jsou vidět veškeré procesy uživatele a veškeré jejich iterace. Návrh rozbalovacích položek je podobný jako na obrázku s úkoly schvalovatele (4.4). Opět se zde lze pomocí tlačítka dostat do detailu vybrané iterace.

Stránka pro ohodnocení iterace zobrazuje 3D scénu, základní informace o 3D objektu, základní informace o iteraci a formulář. Zde uživatel vybírá rozhodnutí k iteraci a vyplňuje poznámky. Stránka je ukázána na obrázku 4.5.

Stránka s ohodnocenou iterací zobrazuje 3D scénu s modelem, informace o modelu a o iteraci. Jsou zde uvedena rozhodnutí jednotlivých autorit a jejich komentáře. Stránka vypadá podobně jako **Stránka pro ohodnocení iterace** (4.5), místo formuláře jsou zde informace o rozhodnutích autorit a komentáře odeslané k iteraci. Také zde autor rozhodnutí může vzít své rozhodnutí zpět, pokud iterace nebyla finalizována – zveřejněna.



■ **Obrázek 4.6** Obrazovka pro publikaci 3D objektu

4.4.4 Stránky administrátora

Zde budou uvedeny stránky, které byly implementovány pro uživatele s rolí administrátor.

Stránka s modely k publikaci zobrazuje iterace, které byly schváleny a je možno je publikovat. Iterace jsou zabaleny do jednotlivých schvalovacích procesů. Počet těchto iterací se promítá do počtu úkolů pro administrátora – v tlačítku „Čeká na publikaci“ se, podobně jako u schvalovatele, zobrazuje číslo reprezentující počet úkolů. Pro každý řádek s procesem je zobrazeno podobné upozornění, které značí počet čekajících úkolů v tomto procesu.

V záznamu o iteraci byly mírně změněny zobrazené položky – například stav schvalování již není potřeba zobrazovat, jelikož se zde zobrazí pouze **approved** modely. Pomocí tlačítka lze přejít do detailu iterace. Vzhled je podobný stránce 4.4.

Stránka pro publikaci modelu zobrazuje 3D scénu s modelem, základní informace o 3D objektu a o iteraci. Zobrazují se zde jména autorit, které tento model schválily a také jejich komentáře. Komentáře jsou pro přehlednost schovány pod tlačítko **Zobrazit komentáře**. Stránka je zobrazena na obrázku 4.6.

Přehled veřejných modelů zobrazuje iterace, které byly publikovány (finalizovány). Iterace jsou pro přehlednost zabaleny podle jednotlivých procesů. Pro každý řádek s procesem je zobrazeno upozornění, které značí počet publikovaných iterací v tomto procesu. V kontextu projektu tedy veškeré veřejné modely pro strukturu.

Stránka je vzhledově podobná stránce s **přehledem modelů k publikaci** nebo **přehledem úkolů schvalovatele** (4.4). Pouze množina procesů a iterací je rozdílná.

Stránka pro zrušení publikace modelu se zobrazí po otevření iterace na stránce s **přehledem veřejných modelů**. Zde může administrátor vzít zpět publikaci 3D modelu. Tím jeho stav vrátí z **finalized** do **approved**.

Stránka je podobná stránce pro **publikaci 3D modelu** (4.6). Pouze akční tlačítka jsou pozměněna.

4.5 Nedokončené funkcionality

V této sekci budou rozvedeny funkcionality, které nebyly v aplikaci dokončeny nebo implementovány. Důvodem je například nízká priorita požadavku, nebo nedodělky na backendu aplikace, které bránily v dokončení. Konkrétní důvody budou uvedeny níže.

4.5.1 Živé zobrazení ve 3D scéně

Naprosto stěžejním faktorem v této aplikaci je zobrazovat 3D modely ve 3D scéně. K tomuto účelu byl vybrán nástroj **liteprůvodce** z oficiálních stránek projektu VMCK.

Došlo zde však k zásadnímu problému. Tento průvodce je totiž napojený na live API na adrese <https://vmck.sic.cz:3443>, ale aplikace funguje, dle žádosti vedoucího práce, na DEV API na adrese <https://vmck-dev.sic.cz:3443/>. To způsobuje problém, že aplikace aktuálně není schopná zobrazit modely ukládané na nové DEV API.

Tento problém měl být vyřešen v průběhu semestru. Požadavkem na vývojáře průvodce bylo, aby vytvořili novou verzi HTML stránky view, která by byla konfigurovaná na nové DEV API. Tato stránka by pak byla vložena přímo do aplikace.

K této akci však během semestru nedošlo. Aplikace tedy používá pouze IFrame s odkazem na Live průvodce¹, kde zobrazuje stále stejný model.

Komponenta s 3D scénou, tedy `model-viewer.component.tsx`, bude v budoucnu potřebovat úpravy. Až bude poskytnuta předělaná stránka průvodce, bude možné ji umístit přímo do této komponenty místo aktuálního IFrame. HTML stránka může být do komponenty vložena například pomocí techniky `dangerouslySetInnerHTML`, jehož použití lze dohledat například na této stránce [34].

4.5.2 Zobrazování všech formátů 3D objektů

Zobrazení 3D objektů ve všech formátech je stěžejní požadavek na aplikaci. Došlo zde však k problému, úzce spojeným s předchozím bodem.

V aktuálním stavu použitá 3D scéna, tedy IFrame s odkazem na liteprůvodce, nedokáže zobrazit modely ve formátu OBJ a FBX. Na vývojáře liteprůvodce byl vynesena požadavek, aby upravil HTML stránku tak, aby podporovala i tyto dva formáty, společně s aktuálně funkčním formátem GLB/GLTF. Nová HTML stránka by pak byla vložena do samotné aplikace.

K této úpravě taktéž během semestru nedošlo. Dočasné řešení je tedy stejné jako v předchozím bodě – jako 3D scéna se zobrazuje IFrame s odkazem na jednu vybranou 3D scénu z liteprůvodce. V tomto místě bude potřeba aplikaci upravit, až bude opravena implementace průvodce.

Alternativním řešením do budoucna může být využití jiného nástroj k zobrazování 3D scény. Paralelně s touto prací vzniká i diplomová práce kolegy Antoše, *Webová komponenta pro interaktivní vizualizaci 3D modelů*. Zde kolega implementuje React knihovnu, která umožní jednoduše zobrazovat 3D modely a manipulovat s nimi. Využit v budoucnu výstupy jeho práce v aplikaci z této práce má dobrý potenciál.

4.5.3 Upozornění přes email

Tento požadavek měl v aplikaci nízkou prioritu. Byl však implementován zárodek této funkcionality.

Dle kapitoly o technologiích 2.8 a části o **EmailJS** byla nastavena emailová služba pro tuto aplikaci.

¹<https://litepruvodce.kralovskavennamesta.cz/view.html?objectId=ba130f1f-a79a-4fea-8f81-2fbefb8f257d&rotation=off&vr=on&camera2=on>

Registrace do služby Email.js byla provedena pomocí speciálně založeného emailového účtu `vmck.notifications@email.cz`. Až později došlo k zjištění, že služba zasílání emailů přes tohoto poskytovatele nepodporuje, byl proto založen druhý email na samotné posílání emailů – `vmck.notifications@gmail.com`. První email je tedy používán k přihlášení na stránce `https://www.emailjs.com/` a druhý email je používán k posílání emailů. Přihlašovací údaje k účtům budou poskytnuty vedoucímu práce.

Šablony jsou navrženy dvě, což je maximum pro členství zdarma. První šablona je navržena pro modeláře – pokud u jeho odeslaného 3D objektu dojde k nějakému rozhodnutí (schválení nebo zamítnutí iterace autoritou), v emailu se modelář dozví o novém stavu jeho 3D objektu. Druhá šablona je určena schvalovatelům. Slouží k zaslání emailu všem schvalovatelům registrovaných v procesu, pokud je do procesu odevzdána iterace. Cílem je upozornit autority, že jim přibyl nový úkol.

Obrázky šablon jsou vloženy v přílohách práce.

4.5.3.1 Problémy

Prvním problémem, který nastal, byla uživatelská práva na API. Jak v první šabloně, tak ve druhé, je v aplikaci potřeba získat detaily o uživateli, jelikož jejich email je dostupný pouze přes API volání `GET /users/{id}`. Na zavolání tohoto endpointu má však uživatel právo pouze tehdy, když jako `id` zadá své vlastní ID. Aplikace tedy není schopná získat emaily jednotlivých uživatelů a nemůže tedy zaslat emaily s upozorněním.

Druhým problémem je samotný návrh zasílání emailů. Častým „best practise“ je, aby zasílání emailů spravoval backend. Je tak tomu kvůli bezpečnosti a úniku údajů. Pokud by tato frontendová aplikace měla zasílat emaily, musela by nejdříve získat detail uživatele z backendu. Již zde by mohlo dojít ke sledování citlivých údajů v developer okně prohlížeče. Následně při samotném zaslání by bylo možné dohledat v prohlížeči například obsah emailu nebo příjemce. Emailová služba by musela být dobře šifrována a zabezpečena.

4.5.3.2 Budoucí rozvoj

Pokud by se budoucí vývojáři rozhodli, že by chtěli, aby aplikace opravdu zasílala emaily, mohou navázat na zárodek kódu v této aplikaci.

V souboru

```
src/utils/email-handler/email-handler.util.ts
```

jsou připraveny a zakomentovány funkce, které lze použít na zasílání emailů podle dvou výše zmíněných šablon. Jedná se funkce `sendModellerEmail` a `sendApproverEmail`.

Použití těchto funkcí je zakomentováno i v místech, kde by k tomuto volání mělo dojít. Tedy `sendModellerEmail` je použit v komponentě `ApproverIterationApproval`, kde dochází k hodnocení iterace. Funkce `sendApproverEmail` je použita a zakomentovaná v komponentě `SendForApprovalAutomaticModal`.

Autor a vedoucí práce se však shodli na tom, že zasílání upozornění by měla být z bezpečnostních důvodů práce backendu.

4.5.4 Úprava podpůrných souborů 3D objektu

V aplikaci aktuálně není možné upravit textury a assety uložených 3D objektů. Důvodem je to, že na backendu byly k těmto souborům přidány povinné atributy **format** a **type** (typový `MimeType` souboru). API však není připravené na to, aby se všechny tyto atributy daly poslat

v jednom formData společně se zbytkem dat 3D objektu. V poslední dokumentaci jsou například textury definovány jako obyčejné pole souborů, což aktuální API neakceptuje.

Formulář pro úpravu dat tedy aktuálně neposílá ani textury, ani assety. Tyto soubory zůstávají po úpravě stejné.

4.5.5 Nekompletní tvorba 3D objektu

Vytváření nového 3D objektu není provedeno naprosto správně.

Transformace a transformační matice se při volání POST neukládají, vždy se uloží defaultní hodnoty. Toto je chyba backendu. Bylo by možno tento problém obejít tím, že po každém POST se zavolá i úprava, PUT na tentýž objekt. Vedoucí práce si však nepřál realizovat toto řešení a rozhodl, že je potřeba opravit backend.

Aplikace tedy umožňuje tato pole zadat a odesílá je, ale backend je uloží v defaultních hodnotách. Až bude backend opraven, ukládání by mělo automaticky fungovat správně.

Nahrávání podpůrných souborů 3D objektu není provedeno ideálně. Důvodem je to, že k texturám a assetům je potřeba posílat i metadata a aktuálně API není schopné přijmout takto složité objekty v jednom formData objektu. V dokumentaci jsou textury a assety uvedeny pouze jako pole souborů, což API aktuálně nepřijímá. U každé textury je potřeba k souboru doplnit **type** (typový MimeType), **format** a **textureType** (typ textury, jedna z předdefinovaných hodnot). U každého asset souboru je nutno přidat pole **format**.

Aktuálně aplikace funguje tak, že vytvoří nový 3D objekt bez podpůrných souborů, a pak pro každý soubor volá oddělený endpoint. Jedná se o tyto endpointy:

- POST /3DObjects/{TIDObject}/assets
- POST /3DObjects/{TIDObject}/textures

Funkcionalita tedy funguje korektně, není to však efektivní ani rychlé řešení.

4.5.6 Stahování souboru s 3D modelem

Ve formuláři s detailem 3D objektu je tlačítko se soubory. Zde si autor může zobrazit a stáhnout textury a assety uložené s modelem. Není zde však možné stáhnout soubor s modelem. Je tomu tak, protože se u modelu na API neukládá jeho typ a název. Aplikace tedy nemůže z filename souboru zjistit název a formát, jako to aktuálně dělá u textur nebo assetů. Nelze tedy tento soubor zobrazit ani stáhnout.

4.5.7 Správa procesů a jejich uživatelů

Toto je funkcionalita, která je pro aplikaci relativně důležitá. Aktuálně totiž aplikace počítá s tím, že pro strukturu, pro kterou chce uživatel odeslat model ke schválení, již existuje v databázi schvalovací proces. Toto vytváření procesů se tedy musí dělat mimo tuto aplikaci, z doporučení autora například v nástroji Postman. [7] Pokud se uživatel pokusí odeslat schválení 3D objekt pro strukturu, pro niž neexistuje proces, bude o tom informován a nebude možno model odevzdat. Návrhově nedává smysl dávat modeláři práva na to proces vytvořit a upravovat, jelikož se jedná o důležitá data, jako název procesu nebo množina členů schvalovacího procesu.

Stejně tak se předpokládá, že přidání uživatelů do procesů bude provedeno externě. Jedná se o přidávání modelářů, historiků a grafiků do schvalovacího procesu pro vybranou strukturu.

4.5.8 Zabezpečení aplikace

Pro budoucí rozvoj aplikace by bylo vhodné podstoupit některá bezpečnostní opatření.

Bezpečnost routes je záležitost změny frontendu. Nyní je v aplikaci možné ručně změnit URL adresu a tím se dostat i na stránky, které by neměly být dostupné jeho uživatelské roli.

Bezpečnost registrace po nasazení je záležitost backendu. Až bude aplikace nasazená na veřejné adrese, v aktuálním stavu by bylo možné registrovat se s naprosto jakýmkoliv Google účtem. Tento problém není tak závažný, protože nově registrovaný uživatel musí být nejdříve aktivován administrátorem, a až poté může nahrávat modely nebo být členem procesů.

4.5.9 Properties u 3D objektů

Koncept properties, také vlastností nebo tagů, je v novém návrhu procesu zamýšlen jako indikátor různých variant jednoho 3D objektu. Například po automatické nebo manuální transformaci existujícího objektu získá nový 3D objekt tag „zasněžený“, pokud aplikovaný efekt byl snůh.

Tato funkcionality není na backendu implementována. Není implementována ani automatická transformace, ani posílání pole **properties** v tvorbě nebo úpravě 3D objektu.

Formuláře v aplikaci jsou částečně připravené na tuto funkcionality. Ačkoliv jsou tlačítka pro přidání tagu deaktivována, na pozadí si formulář udržuje objekty s vloženými tagy. Jedná se o jedno pole s klíči a druhé pole s hodnotami, kde hodnoty na stejném indexu odpovídají jedné zadané dvojici.

Uživatelské testování probíhalo pomocí moderovaného procházení testovacími scénáři. Tester obdržel testovací scénáře a spuštěnou aplikaci. U testování asistoval moderátor (autor této práce), který dával testerovi případné instrukce a korigoval nastavení a průběh testování. Při testování byl nahráván zvuk a obrazovka testovacího počítače. Tester byl na začátku testování požádán, aby přemýšlel nahlas, aby bylo možné efektivně záznamy z testování analyzovat. Ukázka průběhu testování je na obrázku 5.1.

Cílem uživatelského testování bylo otestovat jednoduchost, intuitivnost a přehlednost rozhraní aplikace.

Akceptační testy probíhaly se zadavatelem práce. Proběhla kontrola, zda aplikace splňuje určené funkční požadavky.

5.1 Testovací scénáře

Pro uživatelské testy byly vytvořeny tři testovací scénáře. Tyto scénáře obsahovaly několik úkolů, které měl tester splnit. Tyto scénáře budou uvedeny níže.

Tester měl také za úkol vyplnit dva krátké dotazníky. V prvním dotazníku tester vyplnil několik základních informací o sobě a svých znalostech. Ve druhém dotazníku vyjádřil své dojmy z aplikace a vnesl případné poznámky.

Pro účel testování (zároveň pro účel posílání emailů s upozorněním) byl vytvořen nový uživatelský účet registrovaný na email **vmck.notifications@gmail.com**. Heslo k účtu bude z bezpečnostních důvodů vynecháno.

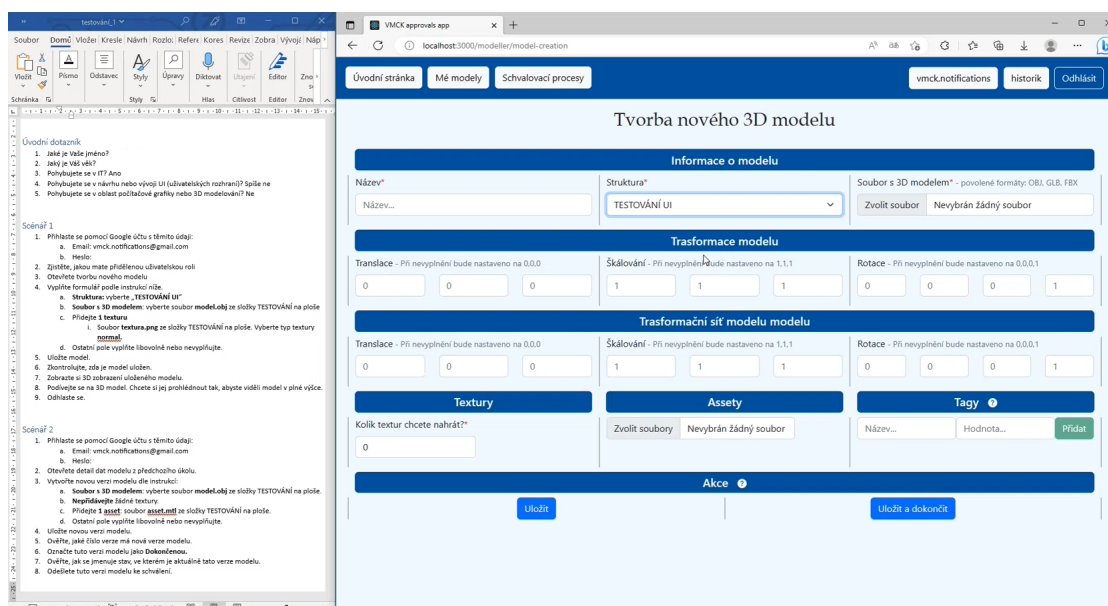
Počátečním bodem je pro veškeré scénáře úvodní stránka aplikace v odhlášeném stavu.

5.1.1 Tvorba scénářů

Scénáře byly postaveny na základně přednášek z předmětu BI-TUR, Tvorba uživatelského rozhraní. Scénáře jsou postaveny na takzvaném „testování použitelnosti“.

Cílem těchto scénářů je ověřit, zda aplikace vhodně vede uživatele k jeho cíli. Scénáře pokrývají některé typické akce v aplikaci. Je zde hodnoceno, jak snadné či náročné je pro testera splnit zadaný cíl.

Použitím scénářů pro testování a následnou analýzou výsledků bude možno určit, zda byl návrh aplikace správně proveden. Hodnocena bude například správná navigace a odezva – zda uživatel pochopil, kudy vede cesta k cíli, zda měl šanci si této cesty všimnout, zda cestu opravdu zvolí nebo jestli porozumí odezvě aplikace. Informace o testování použitelnosti lze dohledat ve studijních materiálech předmětu BI-TUR [35].



■ Obrázek 5.1 Průběh uživatelského testování

5.1.2 Scénář 1 – Přihlášení a nahrání modelu

Pro tento scénář potřebuje být uživatel v roli **modelář**. Kvůli chybám v uživatelských právech na backendu bylo testování prováděno s rolí **historik** a s upraveným nastavením, aby se aplikace pro historika dočasně chovala tak, jak by se měla chovat pro modeláře.

1. Přihlaste se pomocí Google účtu s těmito údaji (email a heslo):
 - a. vmck.notifications@gmail.com
 - b. *****
2. Zjistěte, jakou máte přidělenou **uživatelskou roli** v aplikaci.
3. Otevřete **tvorbou nového modelu**.
4. Vyplňte formulář podle instrukcí níže.
 - a. **Struktura**: vyberte **TESTOVÁNÍ UI**.
 - b. **Soubor s 3D modelem**: vyberte soubor **model.obj** ze složky **TESTOVÁNÍ** na ploše.
 - c. Přidejte **1 texturu**.
 - i. Soubor **textura.png** ze složky **TESTOVÁNÍ** na ploše. Vyberte **typ** textury **normal**.
 - d. Ostatní pole vyplňte libovolně nebo nevyplňujte.
5. **Uložte** model.
6. Zkontrolujte, zda je model uložen.
7. Zobrazte si **3D** zobrazení uloženého modelu.
8. Podívejte se na 3D model. Chcete si jej prohlédnout tak, abyste viděli model v **celé výšce** **najednou**.
9. Odhlaste se.

5.1.3 Scénář 2 – Nová verze a odeslání

Pro tento scénář potřebuje být uživatel v roli **modelář**. Situace s právy a nastavením aplikace je stejná jako ve scénáři 1.

1. Přihlaste se pomocí Google účtu s těmito údaji (email a heslo):
 - a. vmck.notifications@gmail.com
 - b. *****
2. Otevřete **detail dat** modelu z předchozího úkolu.
3. Vytvořte **novou verzi** modelu dle instrukcí:
 - a. **Soubor s 3D modelem**: vyberte soubor **model.obj** ze složky TESTOVÁNÍ na ploše.
 - b. Nepřidávejte žádné textury.
 - c. Přidejte **1 asset**: soubor **asset.mtl** ze složky TESTOVÁNÍ na ploše.
 - d. Ostatní pole vyplňte libovolně nebo nevyplňujte.
4. **Uložte** novou verzi modelu.
5. Ověřte, jaké **číslo verze** má nová verze modelu.
6. Označte tuto verzi modelu jako **Dokončenou**.
7. Ověřte, jak se jmenuje **stav**, ve kterém je aktuálně tato verze modelu.
8. **Odešlete** tuto verzi modelu ke schválení.
 - a. Iteraci pojmenujte libovolně
9. Ověřte, jak se jmenuje **stav**, ve kterém je aktuálně tato verze modelu.
10. Odhlaste se.

5.1.4 Scénář 3 – Schválení iterace a zrušení rozhodnutí

Pro tento scénář potřebuje být uživatel v roli **schvalovatel**. Jelikož role **historik** byla dočasně upravena a použita v předchozích scénářích, zde je použita role **grafik**. Moderátor testování před tímto scénářem upravil uživatelskou roli účtu vmck.notifications@gmail.com na **approver**, neboli **grafik**.

1. Přihlaste se pomocí Google účtu s těmito údaji (email a heslo):
 - a. vmck.notifications@gmail.com
 - b. *****
2. Zjistěte, **kolik** na Vás čeká **úkolů**.
3. Otevřete si libovolný **úkol** tak, abyste byli připraveni úkol **ohodnotit**.
4. Vyberte si libovolné **rozhodnutí** a napište k němu libovolnou **poznámku**.
5. **Odešlete** hodnocení.
6. Zjistěte, jaké je **rozhodnutí** role **Historik** k této iteraci.
7. Přejděte do **přehledu** veškerých schvalovacích procesů.
8. Vyberte některý model, který je **zamítnutý** rolí **Grafik**.
9. **Vezměte zpět** toto rozhodnutí.

5.2 Statistiky z testování

Zde bude provedena stručná analýza zúčastněných testerů a jejich dojmů z aplikace. Budou zde analyzovány dotazníky, jež testeři vyplnili.

5.2.1 Testeři

Aplikaci testovalo šest testerů. Jednalo se o čtyři studenty ČVUT, oponenta práce a vedoucího práce. Uvedu zde stručné statistiky o testerech.

Věk testerů bude rozdělen do tří skupin dle rozmezí věku. Jeden tester patřil do věkové skupiny **do 20 let**. Čtyři testeři patřili do skupiny **do 21 až 30 let**. Jeden tester patřil do skupiny **30+**.

Technické znalosti testerů jsou důležitým faktorem pro jejich schopnost se rychle orientovat a přizpůsobovat v digitálním prostoru. Proto byli testeři dotázáni na některé jejich počítačové znalosti. Dotázané otázky a získané odpovědi jsou uvedeny níže:

- **Pohybujete se v IT?**
 - 5x Ano, 1x Ne.
- **Pohybujete se v návrhu nebo vývoji UI?**
 - 3x Ano, 3x Ne.
 - Poznámka autora – 4 z 6 testerů se někdy věnovali vývoji webových stránek.
- **Pohybujete se v oblasti počítačové grafiky nebo 3D modelování?**
 - 1x Ano, 5x Ne.

5.2.2 Hodnocení aplikace

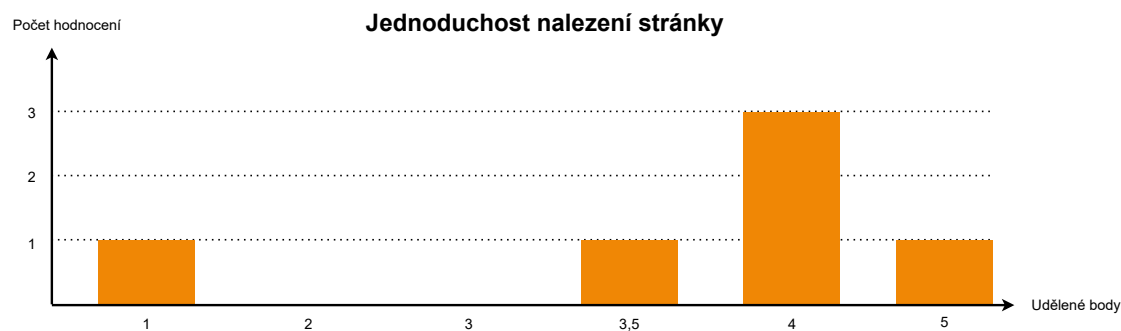
V dotazníku na konci bylo položeno pět otázek týkajících se přívětivosti aplikace. Tester měl ohodnotit otázky číslem 1–5, kde 1 je nejhorší výsledek a 5 je nejlepší. Body hodnocení jsou uvedeny níže:

1. Výstižnost / pochopitelnost popisků (tlačítek, nadpisů).
2. Kvalita vysvětlivek (otazníčků) – vyplňte 0, pokud jste vysvětlivky nepoužívali.
3. Složitost nalezení hledané stránky. Pokud bylo náročné nějakou najít, prosíme vyplňte kterou.
4. Vzhled aplikace.
5. Rychlost odezvy aplikace.

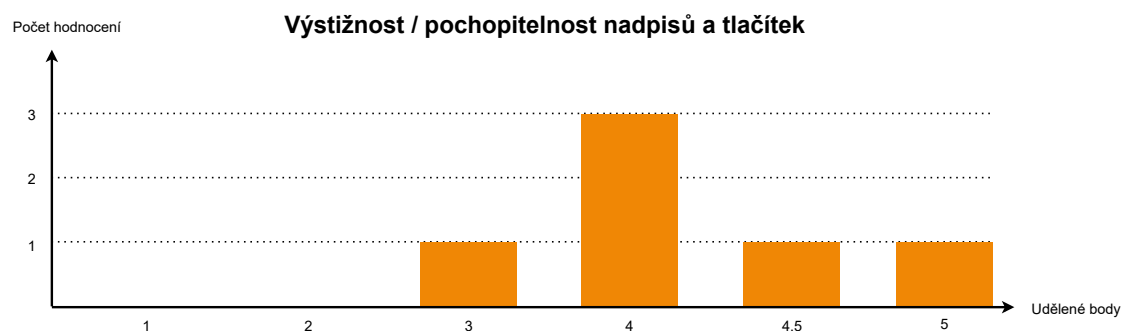
Z těchto otázek byly vytvořeny pro přehlednost grafy. Zde budou uvedeny nejpodstatnější grafy, jenž určují nutnost některé věci v aplikaci změnit. Jedná se o graf **jednoduchosti nalezení stránky** (5.2) a **výstižnost nadpisů a tlačítek** (5.2). Zbytek grafů bude uveden v příloze.

5.3 Výsledky testování

Zde budou uvedeny některé problémy, jež byly objeveny během uživatelských testů. Pokud byl některý problém dostatečně závažný a byla v návaznosti na testování změněna implementace, bude to zde popsáno.



■ **Obrázek 5.2** Graf hodnocení testerů k bodu *Jednoduchost nalezení hledané stránky*



■ **Obrázek 5.3** Graf hodnocení testerů k bodu *Výstižnost nadpisů a tlačítek*

5.3.1 Překládání stavů

V původní implementaci byly stavy 3D objektu zobrazovány tak, jak jsou uloženy v databázi. Jedná se o stavy **unfinished**, **finished**, **submitted**, **declined**, **approved** a **finalized**. Ačkoliv u políček, kde se zobrazuje stav, byla nápověda s překladem a významem těchto stavů, bylo zkritizováno, že stavy nejsou česky.

Stavy jsou tedy nově přeloženy do češtiny skrz celou aplikaci. Nově se stavy zobrazují jako **nedokončený**, **dokončený**, **odeslaný ke schválení**, **zamítnutý**, **schválený** a **zveřejněný**.

5.3.2 Upozornění o načítání aplikace

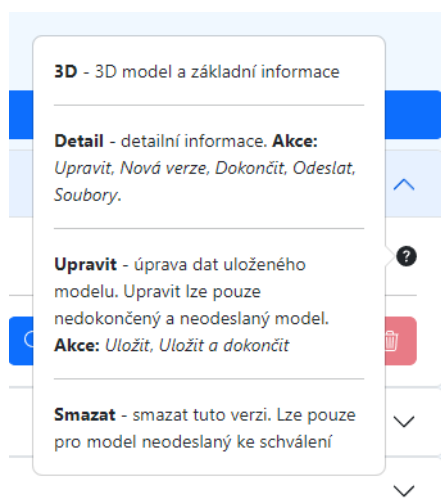
Již po prvním testování bylo zjištěno, že je vhodné uživatele upozornit o tom, že se aplikace načítá. Jedná se obzvláště o situace, kdy aplikace provádí API volání a čeká na odpověď.

Po prvním testování bylo implementováno okno, které uživateli oznámí, že se aplikace načítá. V grafu, který zobrazuje hodnocení rychlosti odezvy aplikace a jenž je vložen v přílohách, jsou vidět výsledky této úpravy. První tester, který toto okno neměl, ohodnotil rychlost špatně. Po implementaci načítacího okna se hodnocení zlepšilo na maximum.

5.3.3 Zlepšení nápověd

Během testování opakovaně docházelo k problémům v navigaci v aplikaci. Šlo například o to, že spousta testerů hledala akce s 3D objektem v **úpravě** nebo **3D detailu** místo **detailu**.

Byly proto rozšířeny nápovědy. Nyní jsou obsáhlejší, lépe formátované a obsahují například informace o tom, co všechno lze dělat na kterém odkaze. Příklad nápovědy, která je dostupná



■ **Obrázek 5.4** Nápopěda k tlačítkům u modelů modeláře

v přehledu modelů modeláře, je zobrazena na obrázku 5.4.

Například u nápopědy u **stavů 3D objektu**, kde jsou vysvětleny významy jednotlivých stavů, byly přidány informace o tom, kde lze stav změnit.

Dále u **verze** 3D objektu byla přidána nápopěda o tom, že se uživatel právě dívá na nejnovější verzi a starší verze nejsou dostupné.

5.3.4 Varování u nevratných akcí

U některých akcí bylo zjištěno, že by bylo vhodné uživatele explicitně upozornit na to, že to, co se snaží provést, je nevratná akce. Tato skutečnost vyvstala z toho, když se jeden tester odklonil od scénáře a u tvorby modelu stiskl tlačítko **Uložit a dokončit** místo tlačítka **Uložit**. Došlo k tomu navzdory tomu, že v nápopědě k akcím bylo vysvětleno, co tlačítko dělá.

U nebezpečných tlačítek, jako například **Dokončit**, bylo tedy přidáno do očí bijící varování v podobě rudé ikonky, která „on hover“ zobrazí text varování. Toto řešení je spíše provizorní.

5.3.5 Doplnění typu u nahrávání textur

Původní implementace aplikace zobrazovala při nahrávání textur políčko pro soubor a vedle něj „select box“ s výběrem možností pro typ textury. Tato implementace počítala s předpoklady pro uživatele modelář – tento uživatel by měl být znalý termínů z prostoru 3D modelování. Tudíž by měl rychle odvodit, k čemu je u souboru políčko s typem.

Při testování se téměř všichni testeři pozastavili nad tím, co je to typ textury a jak jej vyplnit. To se týkalo i testera, který má zkušenosti s 3D modelováním. Nikdo nakonec neměl problém s pochopením a vyplněním. Autor však došel k závěru, že bude vhodné do formuláře doplnit text informací o tom, že se očekává vyplnění souboru s texturou a její typ. Byl tedy doplněn nový vysvětlující text ve formulářích.

5.3.6 3D scéna

Při testování bylo zhruba v 50% pro testery problematické na první pokus pochopit, jak funguje pohyb ve 3D scéně. Toto je problém, který aplikace přijala společně s integrací liteprůvodce jako svého 3D zobrazovacího nástroje.

Tento problém není akutní a není prioritou jej řešit. Po prvním setkání s prostředím by již uživatelé měli zvládnout se v prostředí pohybovat. Bylo by však do budoucna vhodné tuto 3D scénu nahradit nějakým vhodnějším nástrojem.

5.3.7 Ikony

Na konci testování testeři dostali možnost se volně vyjádřit k aplikaci. Zazněly například připomínky, že použití **ikon** v aplikaci by mohlo vylepšit orientaci a lépe konotovat¹ význam tlačítek nebo akcí.

Do aplikace byly tedy přidány ikony ke všem důležitým tlačítkům. Nejen, že nyní aplikace působí lepším estetickým dojmem, ale také se zrychlila uživatelská orientace mezi tlačítky.

5.4 Postřehy z BI–TUR

V předmětu BI–TUR proběhla pro tento prototyp analýza návrhu, implementace a výsledků testování. Zde uvedu některé postřehy, které z prezentace prototypu a výsledků testování vyvstaly. Tyto návrhy zde autor uvádí jako možnosti na rozšíření do budoucna.

5.4.1 Indikátory stavu

Z důvodu lepší přehlednosti a sociální přijatelnosti by bylo vhodné, aby byly vylepšeny indikátory stavu. Pouhá barva není dostatečný nosič informace.

Aktuální návrh by nebyl vhodný například pro uživatele se slabším barevným vnímáním. Také například za zhoršených světelných podmínek je špatně patrný rozdíl mezi zelenou a šedou barvou – již při promítání na projektor byly tyto barvy téměř nerozeznatelné.

V budoucím návrhu by bylo vhodné implementovat lepší nosič informace o výsledku a barvu ponechat jako podpůrný nosič. Lze například udělat jednotlivé sloupce pro grafika a historika a místo „G“ a „H“ vložit do indikátorů ikony. Například vložit zelenou „fajfku“ pro schválení, zelený křížek pro zamítnutí a šedý otazník pro neohodnoceno.

5.4.2 Stahování souborů pro schvalovatele

Při prezentaci prototypu a výsledků testování v předmětu BI–TUR bylo diskutováno, zda by měl schvalovatel mít možnost stahovat soubory.

Do budoucna by uživatel v roli grafik nebo historik měl mít možnost si stáhnout soubory, které patří k 3D modelu, který uživatel právě schvaluje. Tato funkcionality je velice vhodná proto, že aktuálně není možné v aplikaci provádět pokročilejší zobrazování.

Uživatel by mohl mít potřebu například porovnat v aplikaci různé verze. Nebo uživatel v roli historik by do budoucna mohl mít potřebu 3D model porovnat s půdorysy nebo schémata – k tomu bude pravděpodobně potřebovat schvalovaný model importovat do externího nástroje. Z toho důvodu bude potřebovat si stáhnout soubory patřící k 3D modelu.

5.5 Akceptační testy

Akceptační testy mají za úkol zkontrolovat, zda aplikace naplňuje veškeré funkční požadavky. Zde akceptační testy probíhaly se zadavatelem práce, Ing. Jiřím Chludilem.

¹konotace = podprahové vyvolávání určitých dojmů nebo spojitostí (například slovo *černý* může někomu konotovat *tmu*, *zlo* nebo *eleganci*)

5.5.1 Průběh

Pro testování byl sestaven scénář, který kompletně pokrýval funkční požadavky určené v analýze. Autor práce postupně předváděl funkcionality aplikace a reprezentoval tím splnění jednotlivých funkčních požadavků.

Zadavatel sledoval prezentaci aplikace a postupně akceptoval funkčnost a nefunkčnost požadavků. Takto bylo u každého požadavku rozhodnuto, zda byl splněn, či ne.

5.5.2 Výsledek

Kompletně splněno: Funkční požadavky **F1**, **F2** a **F4** byly splněny kompletně, včetně veškerých svých bodů.

Částečně splněno: Funkční požadavek **F3** byl naplněn téměř kompletně. Jediný bod, který nebyl naplněn a nebyl akceptován, je požadavek **F3.7: Zaslání upozornění**.

Důvod nesplnění požadavku byl uveden v kapitole o nedokončených funkcionalitách (4.5). Stručně uvedeno, důvodem je, že v aplikaci nejsou, a ani nebudou, práva nastavena tak, aby upozornění přes email šlo provést.

Také bylo při akceptačních testech rozhodnuto, že funkcionalita pro zaslání emailů či jiných upozornění je dle „best practice“ zodpovědností backendu projektu, nikoliv frontendu. Je tomu tak hlavně z bezpečnostních důvodů. Z tohoto důvodu byla funkcionalita vynechána a byl vytvořen požadavek na backend, aby upozornění uživatelů do budoucna implementoval.

Kapitola 6

Příručky

V této kapitole bude uvedena instalační a uživatelská příručka. Příručky slouží ke snadné přípravě aplikace pro používání.

6.1 Instalační příručka

Zde bude čtenář proveden postupem, jak aplikaci lokálně spustit a sestavit pro případné nasazení.

1. **Naklonování GIT repozitáře** – tento repozitář je aktuálně privátní, jeho kopie bude poskytnuta v adresáři vedoucího. Adresář lze naklonovat v terminálu pomocí příkazu

```
git clone https://gitlab.fit.cvut.cz/...
```

kde `https` odkaz bude získán z repozitáře vedoucího.

Zdrojové kódy budou také přiloženy v při odevzdání práce.

2. **Instalace** probíhá pomocí nástroje `npm`. Je tedy předpokladem, aby uživatel měl nainstalovaný **Node.js**. V naklonovaném repozitáři je potřeba se přesunout do složky `vmck-approvals-frontend`. V případě jiného pojmenování složek je potřeba přesunout se do složky, kde jsou umístěny adresáře `src` a `public`. Zde se instalace spouští příkazem `npm install`. Tímto se nainstalují potřebné balíčky a knihovny.

3. **Nastavení konfiguračních souborů** je nutno provést tak, že uživatel si od autora nebo vedoucího práce vyžádá `.env` soubor. Tento soubor obsahuje citlivá konfigurační data k aplikaci. Tento soubor beze změny vloží do adresáře `vmck-approvals-frontend`. V případě přejmenování to opět bude složka, ve které je například adresář `src`. Soubor `.env` tedy musí být na stejné úrovni, jako je adresář `src`.

V tomto souboru se definuje například adresa API nebo Client ID pro Google přihlašování.

4. **Spuštění aplikace** na `localhost` se provádí pomocí příkazu `npm start`. Příkaz je opět nutno provést ve složce `vmck-approvals-frontend`.
5. **Sestavení aplikace** se provádí pomocí příkazu `npm run build` spuštěného ve složce `vmck-approvals-frontend`.
6. **Nasazení aplikace** je aktuálně provedeno pomocí technologie `nginx` na DEV serveru projektu VMCK. Pro nasazení nové verze je potřeba aplikaci **sestavit** pomocí `npm run build` a následně obsah složky `build` zkopírovat do požadovaného nakonfigurovaného adresáře.

6.2 Uživatelská příručka

Uživatelská příručka je v této práci nahrazena rozborem stránek aplikace v kapitole **Stránky webové aplikace** (4.4). V této kapitole jsou uvedeny veškeré stránky aplikace a akce, které na nich uživatel může provádět. Kapitola se velice podrobně zabývá kompletním seznamem stránek a funkcionalitami, které jednotlivé stránky nabízí.

Dále byly v kapitole o výsledcích **testování** (5.3) některé nejasné stránky aplikace objasněny. Také byly do aplikace dodány velice rozsáhlé nápovědy, mimo jiné pro orientaci a vysvětlení pojmů.

Dále v kapitole s **testovacími scénáři** (5.1) lze získat dobrý přehled o tom, jaká je posloupnost akcí k dosažení některých uživatelských cílů.

Závěr

Tato práce se zabývala vývojem prototypu webové aplikace pro projekt Věnná města českých královen. Aplikace implementovaná v této práci měla za cíl umožnit členům projektu VMCK nahrávat 3D modely, podrobovat je schvalování a následně mít možnost označit schválené modely za veřejné.

Nejdříve byla provedena analýza projektu VMCK a Private API pro projekt VMCK. Také byla provedena analýza aplikací přidružených k projektu VMCK a analýza závěrečných prací, které se tematicky týkaly schvalování 3D modelů. Pomocí těchto zdrojů a instrukcí vedoucího byly sesbírány uživatelské požadavky a sepsány funkční a nefunkční požadavky na aplikaci. Také byl analyzován potenciál využít některé z jejich výstupů během návrhu a implementace. Tento potenciál se projevil jako značný. Dále byly analyzovány technologie, mezi kterými bylo potřeba před implementací volit.

V kapitole Návrh byl proveden návrh prototypu webové aplikace. Byly zde rozebrány procesy, které má aplikace zprostředkovat uživatelům. Dále byl proveden návrh stránek aplikace tak, aby aplikace pokrývala vytvořené funkční a nefunkční požadavky. Nakonec byla provedena kontrola, zda návrh aplikace pokrývá veškeré požadavky.

V kapitole Realizace byla nejdříve odůvodněna volba mezi analyzovanými technologiemi. Dále byly rozebrány některé problémy, které se během implementace objevily, a jejich dopad na tuto aplikaci. Dále byly popsány implementované stránky aplikace a vysvětlena jejich funkce. Na konci kapitoly byly rozebrány funkcionality, které nebylo možné dokončit a z jakého důvodu. Také zde bylo uvedeno, jak bude možno funkcionality opravit nebo dodělat v budoucnosti, až se zlepší podmínky pro implementaci.

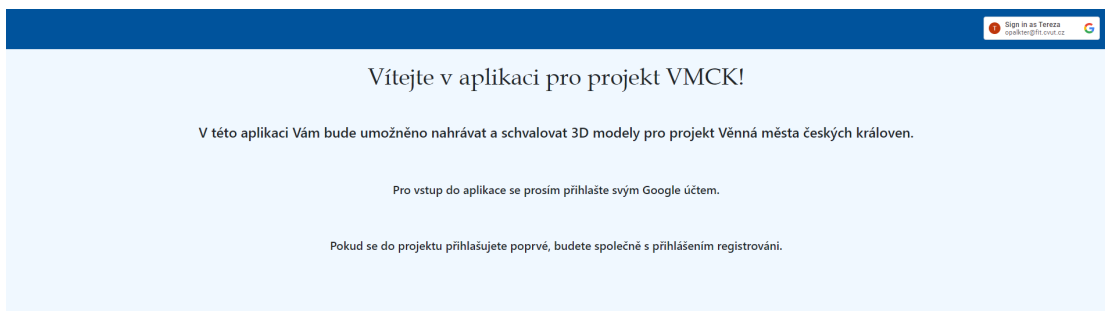
V kapitole Testování byl popsán průběh uživatelských testů, jimž byla aplikace podrobena. Byl zde uveden průběh testů a jejich výsledky. Na základě výsledků testů byly v aplikaci provedeny úpravy a ty zde byly také zdokumentovány. Byly zde také popsány akceptační testy pro aplikaci. K aplikaci byly vytvořeny příručky, instalační a uživatelská.

Dle autora byly splněny cíle vytyčené v této práci. Aplikace umožňuje uživatelům nahrávat 3D modely, podrobovat je schvalovacímu procesu, schvalovat je a nakonec vybrané schválené modely označit za veřejné. U funkcionalit, jež nebylo možno dokončit, byl řádně uveden důvod jejich nekompletnosti a byly rozebrány možné opravy v budoucnosti.

Hlavním přínosem této práce je, že aplikace zpřístupnila uživatelům nahrávání 3D modelů a schvalovací proces pro 3D modely. Tento proces byl již dříve implementován, provádět jej bylo však vysoce nepraktické a nepřístupné uživatelům bez hluboké znalosti struktury projektu VMCK, jeho dokumentace a jeho API. Naproti tomu tato webová aplikace umožňuje intuitivní a jednoduché naplnění uživatelských cílů, jež jsou obsaženy v nahrávání modelů, schvalovacím procesu a publikaci modelů.

Další obrazovky aplikace

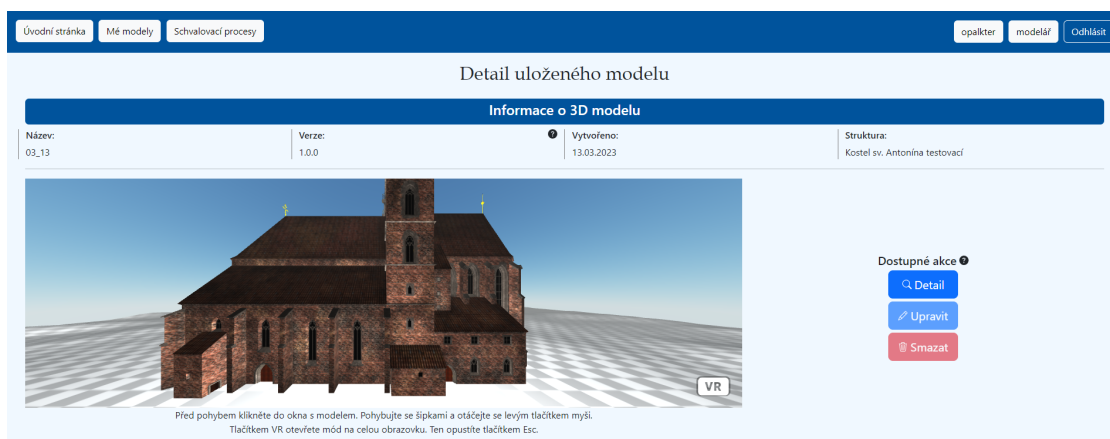
Zde budou uvedeny některé stránky, které byly v aplikaci implementovány, ale pro přehlednost nebyly jejich obrázky uvedeny v textu. Není to kompletní výčet veškerých stránek.



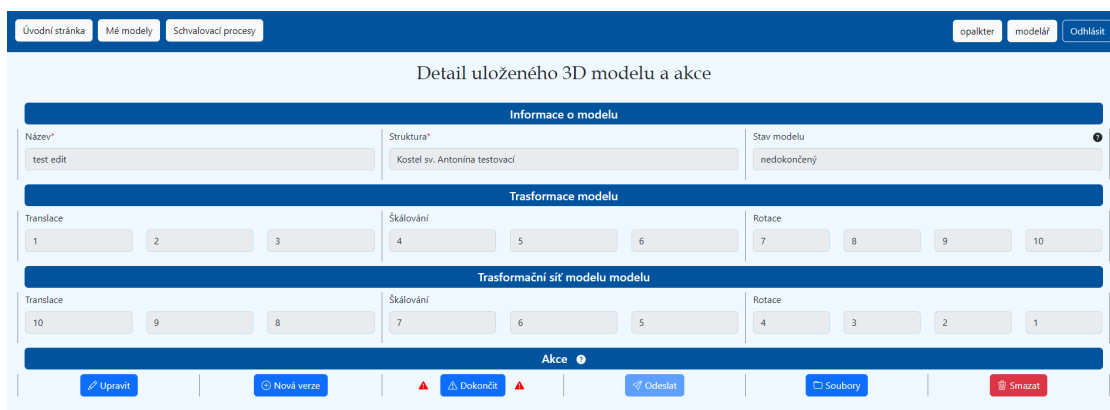
■ **Obrázek A.1** Úvodní obrazovka pro nepřihlášené uživatele



■ **Obrázek A.2** Úvodní obrazovka pro přihlášeného uživatele, zde modeláře



■ **Obrázek A.3** Obrazovka s detailem uloženého 3D objektu ve formě 3D zobrazení

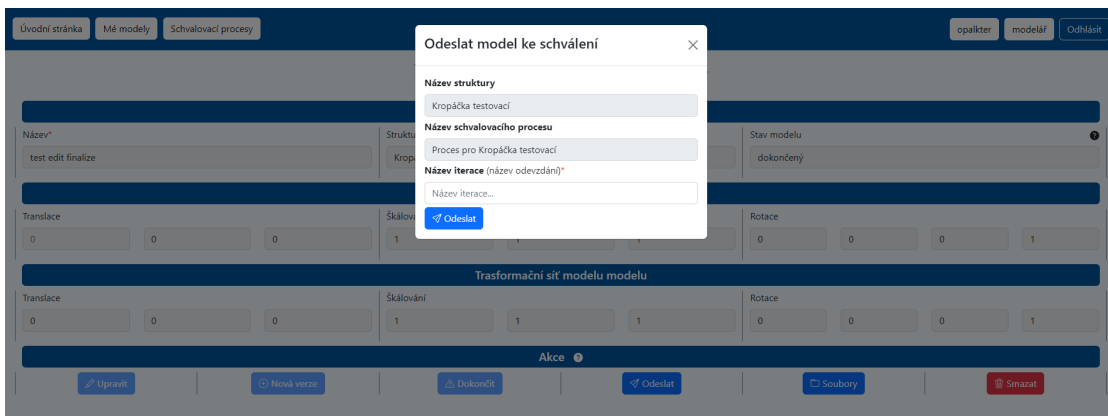


■ **Obrázek A.4** Obrazovka s detailem uloženého 3D objektu ve formě formuláře

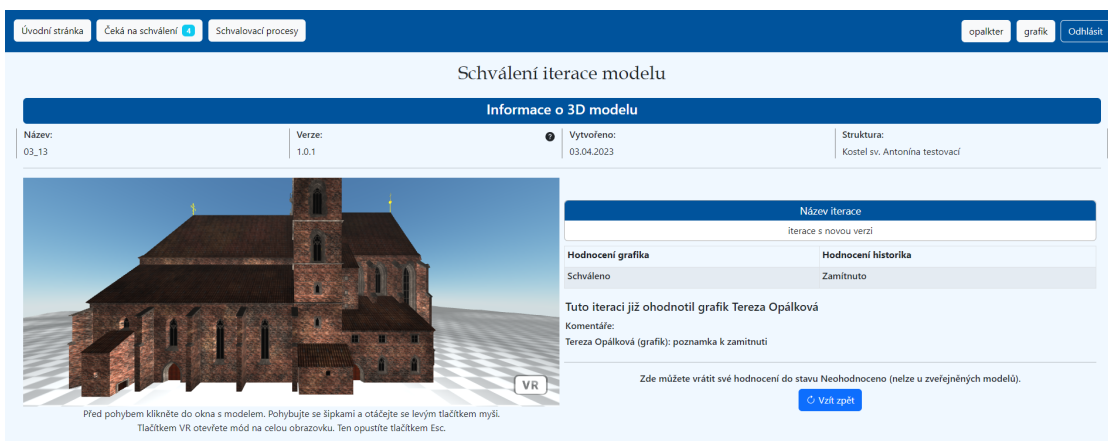
Obrázek A.5 Obrazovka s tvorbou nové verze pro existující 3D objekt

Obrázek A.6 Obrazovka s úpravou uloženého 3D objektu

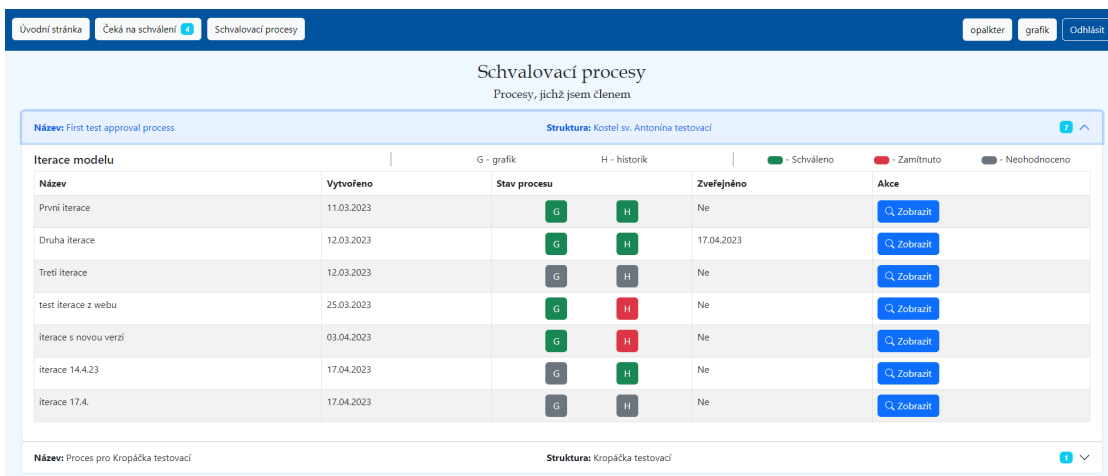
Obrázek A.7 Obrazovka se zobrazenými soubory uloženého 3D objektu



Obrazek A.8 Obrazovka s formulářem pro odeslání 3D objektu ke schválení



Obrazek A.9 Obrazovka s detailem iterace, která byla ohodnocena – zobrazena ve formátu pro schvalovatele



Obrazek A.10 Obrazovka s přehledem veškerých schvalovacích procesů schvalovatele

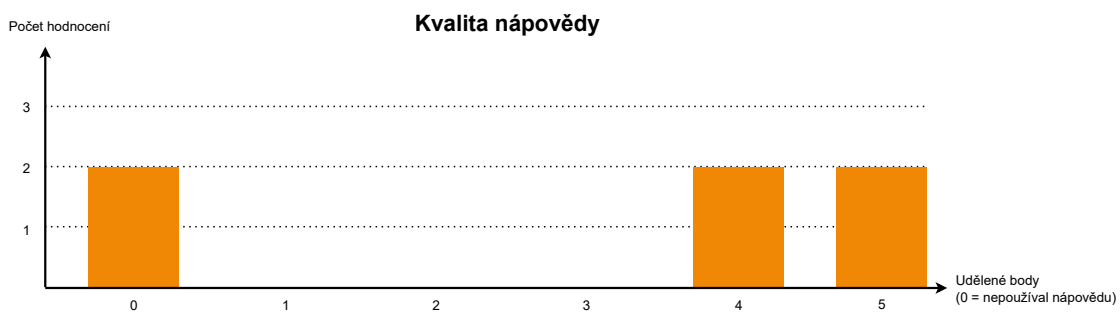
The screenshot displays a web application interface with a dark blue header. The header contains navigation links: 'Úvodní stránka', 'Čeká na publikaci', 'Veřejné modely', 'opalkter', 'administrátor', and 'Odhlásit'. The main content area is titled 'Zveřejněné modely' and includes a sub-header 'Tyto iterace a jejich modely byly schváleny a zveřejněny.' Below this, there is a summary bar with 'Název: First test approval process' and 'Struktura: Kostel sv. Antonína testovací'. The main section is titled 'Schválené iterace modelu' and contains a table with the following data:

Název	Vytvořeno	Zveřejněno	Akce
Druhá iterace	12.03.2023	17.04.2023	Zobrazit

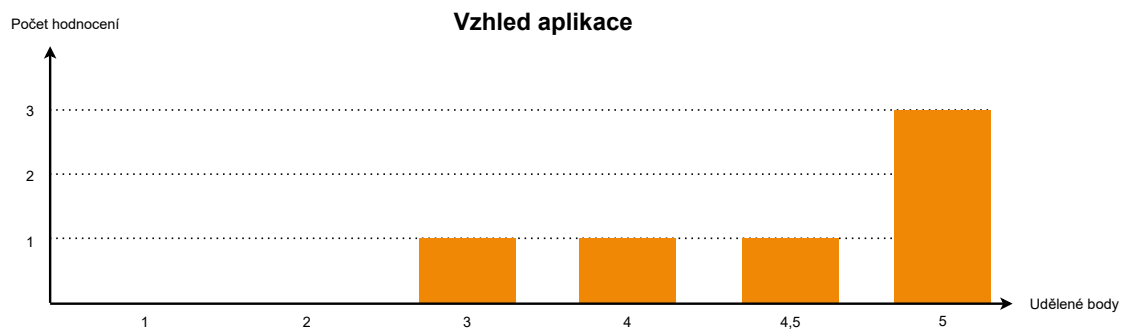
■ **Obrázek A.11** Obrazovka s přehledem publikovaných iterací

Další grafy z uživatelského testování

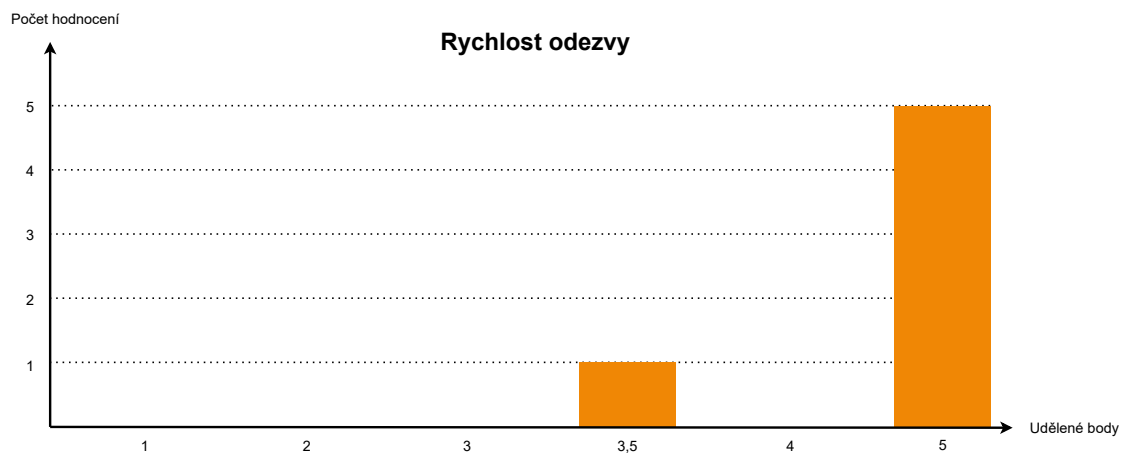
Zde bude uveden zbytek grafů vytvořených z dat získaných při uživatelském testování. Jedná se o „počet hvězdiček“, který testeři udělili jednotlivým bodům v aplikaci. Jsou zde uvedeny grafy, které pro přehlednost nebyly ukázány v textu.



■ **Obrázek B.1** Graf hodnocení testerů k bodu Kvalita nápovědy (otazníčků)



■ **Obrázek B.2** Graf hodnocení testerů k bodu Vzhled aplikace



■ **Obrázek B.3** Graf hodnocení testerů k bodu Rychlost odezvy aplikace

Emailové šablony

Zde budou uvedeny návrhy dvou emailových šablon pro službu EmailJS. Jedná se o dvě šablony. Každá šablona při použití dostane sadu parametrů, které se pokusí podle jména parametru vložit na místa pro proměnné, označené jako `{{prop_name}}`.

První šablona oznámí schvalovateli, že do jednoho z procesů, jichž je členem, byl odevzdán nový model (nová iterace).

Druhá šablona oznámí modeláři, že u jednoho jeho odevzdaného 3D modelu došlo k ohodnocení.

Schvalovatel odevzdání iterace

Playground Test It Save

Content Auto-Reply Attachments Contacts Settings

Subject *
Nové odevzdání v projektu VMCK

Content *

B I U S sans-serif 12pt **A** **I** **≡** **≡** **≡** **≡**

← → ☰ ☷ ☹ ☺ ☻ ☼ ☽ ☾ ☿ ☽ ☾ ☿ ☽ ☾ ☿ <>

Dobrý den,

máme pro Vás upozornění z aplikace pro projekt **Věnná města českých královen**.

Upozorňujeme Vás, že ve schvalovacím procesu s názvem "**{{process_name}}**" došlo k odevzdání nové iterace s názvem **{{iteration_name}}**.

Prosíme Vás o kontrolu a ohodnocení odevzdaného 3D objektu.

P **tiny**

To Email *
{{email_to}}

From Name
VMCK notifications

From Email *
 Use Default Email Address ⓘ

Reply To

Bcc

Cc

■ Obrázek C.1 Šablona pro zaslání upozornění schvalovateli

Modelář změna stavu

Playground Test It Save

Content Auto-Reply Attachments Contacts Settings

Subject *
Změna stavu schvalování v projektu VMCK

Content *

B I U S sans-serif 12pt **A** **I** **≡** **≡** **≡** **≡**

← → ☰ ☷ ☹ ☺ ☻ ☼ ☽ ☾ ☿ ☽ ☾ ☿ ☽ ☾ ☿ <>

Dobrý den,

máme pro Vás upozornění z aplikace pro projekt **Věnná města českých královen**.

[Odkaz na stránky aplikace](#)

Váš 3D objekt a názvem "**{{model_name}}**", jenž byl odevzdán v iteraci s názvem "**{{iteration_name}}**", byl ohodnocen jednou z autorit.

Nový stav Vašeho 3D objektu je nyní **{{model_new_status}}**.

P **tiny**

To Email *
{{email_to}}

From Name
VMCK notifications

From Email *
 Use Default Email Address ⓘ

Reply To

Bcc

Cc

■ Obrázek C.2 Šablona pro zaslání upozornění modeláři

Bibliografie

1. GRULICH, Petr et al. *Věnná města českých královen* [online]. 2023. [cit. 2023-04-11]. Dostupné z: <https://www.kralovskavennamesta.cz>.
2. GRULICH, Petr et al. *Věnná města českých královen, průvodce* [online]. 2023. [cit. 2023-04-11]. Dostupné z: <https://litepruvodce.kralovskavennamesta.cz/>.
3. DOMINIK SIVÁK, Jiří Chludil. *Private API for the project Věnná Města Českých Královen* [online]. 2023. [cit. 2023-04-11]. Dostupné z: <http://109.123.202.213:3003/>.
4. SIVÁK, Dominik. *Private API for the project Věnná Města Českých Královen* [online]. 2021. [cit. 2023-04-11]. Dostupné z: <https://app.swaggerhub.com/apis/sivakdom/private-api/3.0.5#>.
5. OPÁLKOVÁ, Tereza. *Konzultace s Ing. Jiřím Chludilem* [osobní konzultace]. 2023.
6. SIVÁK, Dominik. *Věnná města českých královen – Backend administrační části*. 2021. Dipl. pr. České vysoké učení technické v Praze, Fakulta informačních technologií.
7. INC., Postman. *Postman API Platform* [soft.]. 2023. [cit. 2023-04-15]. Dostupné z: <https://www.postman.com/>.
8. VANČURA, Daniel. *Věnná města českých královen - jádro*. 2020. Bak. práce. České vysoké učení technické v Praze, Fakulta informačních technologií.
9. MARTINEK, Michal. *Administrační rozhraní pro projekt Věnná města českých královen*. 2020. Dipl. pr. České vysoké učení technické v Praze, Fakulta informačních technologií.
10. KOŘOUŠKOVÁ, Barbora. *CO JE TIKETOVACÍ SYSTÉM A PROČ HO ZAVÉST DO FIRMY? Rascasone* [online]. 2021 [cit. 2023-04-11]. Dostupné z: <https://www.rascasone.com/cs/blog/co-je-tiketovaci-system>.
11. ANTOŠ, Pavel. *Věnná města českých královen – Webová aplikace pro schvalovací proces 3D modelů*. 2021. Bak. práce. České vysoké učení technické v Praze, Fakulta informačních technologií.
12. ANTOŠ, Pavel. *Dotazy k bakalářské práci* [elektronická pošta]. 2022. [cit. 2023-04-15]. Zasláno z: antospa2@fit.cvut.cz 4.12.2022.
13. A1CD et al. *Error message "error:0308010C:digital envelope routines::unsupported"* [online]. 2021. [cit. 2023-04-15]. Dostupné z: <https://stackoverflow.com/questions/69692842/error-message-error0308010cdigital-envelope-routinesunsupported/73465262#73465262>.
14. KLOFÁČ, Marek. *Věnná města českých královen - Schvalovací systém pro 3D modelu ve virtuální realitě*. 2021. Bak. práce. České vysoké učení technické v Praze, Fakulta informačních technologií.

15. NYPL, Tomáš; OPÁLKOVÁ, Tereza; PRAŽÁK, Michal; MARTIN, Švela; HIEU, Ta Minh. *Softwarový týmový projekt 1* [soft.]. České vysoké učení technické v Praze, Fakulta informačních technologií, 2022. [cit. 2023-04-15]. Dostupné z: <https://gitlab.fit.cvut.cz/vmck/web-manager>.
16. OPÁLKOVÁ, Tereza; PRAŽÁK, Michal; MARTIN, Švela; HIEU, Ta Minh. *Softwarový týmový projekt 2* [soft.]. České vysoké učení technické v Praze, Fakulta informačních technologií, 2022. [cit. 2023-04-15]. Dostupné z: <https://gitlab.fit.cvut.cz/vmck/web-manager>.
17. ARMSTRONG, Neil. *Mapy API verze 4.13* [online]. Seznam, 2020. [cit. 2023-04-15]. Dostupné z: <https://api.mapy.cz/>.
18. LUQMANUL, M. What Is an iFrame? *Hostinger Tutorials* [online]. 2022 [cit. 2023-04-15]. Dostupné z: <https://www.hostinger.com/tutorials/what-is-iframe/>.
19. OPÁLKOVÁ, Tereza. *Projekt v předmětu Tvorba uživatelských rozhraní*. České vysoké učení technické v Praze, Fakulta informačních technologií, 2023.
20. DIAGRAMS.NET. *Diagrams.net* [soft.]. 2012. [cit. 2023-04-16]. Dostupné z: <https://app.diagrams.net/>.
21. HERBERT, David. *What is React.js? (Uses, Examples, & More)* [online]. 2022. [cit. 2023-04-21]. Dostupné z: <https://blog.hubspot.com/website/react-js>.
22. ARANCIO, Stephen. *What is JSX?* [online]. 2021. [cit. 2023-04-21]. Dostupné z: <https://medium.com/@sjarancio/what-is-jsx-e3dda0af3490>.
23. KNOWLEDGEHUT. *What are the Pros and Cons of React* [online]. 2022. [cit. 2023-04-21]. Dostupné z: <https://www.knowledgehut.com/blog/web-development/pros-and-cons-of-react>.
24. SIMPLILEARN. *Guide To Using Typescript With React* [online]. 2023. [cit. 2023-04-21]. Dostupné z: <https://www.simplilearn.com/tutorials/reactjs-tutorial/react-typescript#:~:text=TypeScript%5C%20is%5C%20a%5C%20typed%5C%20JavaScript,JavaScript%5C%20to%5C%20generate%5C%20our%5C%20UIs..>
25. APPSDEVPRO. *React With TypeScript Vs JavaScript: Which Is Better?* [online]. 2022. [cit. 2023-04-16]. Dostupné z: [https://www.appsdevpro.com/blog/react-with-typescript-vs-javascript/#:~:text=React%5C%20with%5C%20TypeScript%5C%20offers%5C%20faster,fast%5C%20UI%5C%20\(User%5C%20Interface\)..](https://www.appsdevpro.com/blog/react-with-typescript-vs-javascript/#:~:text=React%5C%20with%5C%20TypeScript%5C%20offers%5C%20faster,fast%5C%20UI%5C%20(User%5C%20Interface)..)
26. DIAGRAMS.NET. *Diagrams.net* [online]. 2012. [cit. 2023-04-16]. Dostupné z: <https://www.diagrams.net/about>.
27. STEWART, Matt. *Designing the Ideal Navigation Bar* [online]. 2023. [cit. 2023-04-18]. Dostupné z: <https://www.thecreativemomentum.com/blog/designing-the-ideal-navigation-bar>. Přeloženo autorem.
28. JAVATTPOINT. *React create-react-app* [online]. 2021. [cit. 2023-04-23]. Dostupné z: <https://www.javatpoint.com/react-create-react-app#:~:text=The%5C%20create%5C%20react%5C%20dapp%5C%20is,on%5C%20writing%5C%20React%5C%20code%5C%20only..>
29. MUBARAKN. *Adding TypeScript* [online]. 2022. [cit. 2023-04-23]. Dostupné z: <https://create-react-app.dev/docs/adding-typescript/>.
30. EMAILJS. *EmailJS* [soft.]. 2022. [cit. 2023-04-23]. Dostupné z: <https://www.emailjs.com/>.
31. EMAILJS. *EmailJS Docs* [online]. 2023. [cit. 2023-04-16]. Dostupné z: <https://www.emailjs.com/docs/>.
32. THEIVORYCODER. *Send Emails In React - SUPER SIMPLE & FREE!* [online]. 2022. [cit. 2023-04-16]. Dostupné z: <https://www.youtube.com/watch?v=Lz8yx-zn-04>.

33. PRABHAT et al. *Send Email Directly From JavaScript using EmailJS* [online]. 2019. [cit. 2023-04-16]. Dostupné z: <https://stackoverflow.com/questions/58791656/send-email-directly-from-javascript-using-emailjs>.
34. BILGILI, Doğacan. *Using dangerouslySetInnerHTML in a React application* [online]. 2021. [cit. 2023-05-03]. Dostupné z: <https://blog.logrocket.com/using-dangerouslysetinnerhtml-in-a-react-application/>.
35. SCHMIDT, Jan. *TESTOVÁNÍ POUŽITELNOSTI* [online]. České vysoké učení technické v Praze, Fakulta informačních technologií, 2023. [cit. 2023-05-10]. Dostupné z: <https://courses.fit.cvut.cz/BI-TUR/lectures/files/TUR8usability.pdf>.