

Univerzita Hradec Králové
Fakulta informatiky a managementu
Katedra informačních technologií

Systemový přístup k návrhu vzdálené laboratoře
Diplomová práce

Autor: Bc. Hanna Kukharava
Studijní obor: im2-p Informační management

Vedoucí práce: Ing. Karel Mls, Ph.D.

Hradec Králové

srpen 2022

Prohlášení:

Prohlašuji, že jsem diplomovou práci zpracovala samostatně a s použitím uvedené literatury.

V Hradci Králové dne 13.8.2022

Hanna Kukharava

Poděkování:

Nejdřív bych ráda poděkovala vedoucímu diplomové práce Ing. Karlu Mlsovi, Ph.D. za odborné vedení, za čas, který mi věnoval, za pomoc a cenné rady při zpracování této práce.

Také obrovské poděkování patří mé rodině, která mi byla podporou po celou dobu mého studia.

Anotace

Laboratoře jsou důležitým nástrojem podporujícím technické a přírodovědné vzdělávání a poskytujícím praktické zkušenosti studentům na středních a vysokých školách. Neustálý rozvoj informačních technologií umožnil vznik možnosti vzdáleného přístupu k laboratorním prostřednictvím webových nebo mobilních aplikací. Vzdálené laboratoře nejen umožňují vzdálený přístup k experimentálním zařízením, ale mají řadu dalších výhod, jako jsou možnost lepšího rozvrhování a přizpůsobení individuálním požadavkům studentů nebo efektivní řízení nákladů.

Tato diplomová práce je věnována dané problematice a jejím primárním cílem je prozkoumat současné vzdálené laboratoře a na základě zjištěných poznatků navrhnout vlastní systém vzdálené laboratoře, který by umožnil studentům provádět experimenty na dálku. Pro modelování systému jsou využity základní nástroje, poskytované jazykem UML.

Klíčová slova: vzdálená laboratoř, UML, experimenty, virtuální laboratoř, internet věcí.

Annotation

Title: System approach to remote laboratory design

Laboratories are an important tool supporting technical and science education and providing practical experience to students in schools and universities. The ongoing development of information technologies and bringing new technology resources has enabled the possibility of remote access to laboratories through web or mobile applications. Remote laboratories not only allow remote access to experimental facilities, but have a number of other advantages, such as the better scheduling and

taking into consideration individual student requirements or effective cost management.

The main aim of the diploma thesis is to research on the existing remote laboratory systems and, based on the findings, design a new remote laboratory system, which would allow students to conduct experiments remotely. The basic tools provided by the UML language are used for system modeling.

Keywords: remote laboratory, virtual laboratory, experiments, IoT.

Obsah

1	Úvod.....	1
2	Cíl práce.....	3
3	Alternativy ke klasickým laboratořím	4
3.1	Virtuální laboratoře.....	4
3.2	Přenosná zařízení pro praktické experimenty	8
3.3	Vzdálené laboratoře	12
3.4	Shrnutí	15
4	Příklady vzdálených laboratoří.....	17
4.1	ILabs	17
4.2	WebLab-Deusto.....	21
4.3	iSES Remote Lab SDK.....	22
4.4	LiLa.....	25
4.5	Vzdáleně řízená laboratoř GymKT.....	27
5	UML.....	30
5.1	Vznik UML	31
5.2	Základní UML diagramy.....	32
5.2.1	Diagram případů užití - Use case diagram.....	32
5.2.2	Diagram tříd — Class diagram	33
5.2.3	Sekvenční diagram — Sequence Diagram	33
6	Systémový přístup k návrhu vzdálené laboratoře v oblasti vzdělávání	34
6.1	Fáze vývoje informačního systému dle metodiky RUP.....	34
6.2	Definování požadavků na systém	36
6.3	Analýza	39
6.3.1	Diagram případů užití.....	39
6.3.2	Analytický model tříd.....	43

6.4	Návrh.....	45
6.4.1	Návrhový model tříd	45
6.4.2	Sekvenční diagramy.....	46
6.5	Ukládání dat.....	50
6.6	Bezpečnost vzdálené laboratoře	51
7	Shrnutí výsledků a závěr	54
8	Seznam použité literatury.....	56
9	Přílohy	63

Seznam obrázků

Obrázek 1 - Simulace rozpouštění železa a zinku v kyselině chlorovodíkové ve virtuální laboratoři VirtuLab.....	6
Obrázek 2 - Simulace pohybu hmotného bodu v gravitačním poli ve virtuální laboratoři Algodoo	7
Obrázek 3 – Mobile Studio IOBoard™	8
Obrázek 4 – Zapojení Mobile Studio IOBoard™ do tabletu.....	8
Obrázek 5 – NI myDAQ	9
Obrázek 6 – Analog Discovery 2 - NI Edition	10
Obrázek 7 – Vernier Circuit Board 2	10
Obrázek 8 – Lab@Home	11
Obrázek 9 – Topologie dávkového experimentu založeného na sdílené architektuře iLab	18
Obrázek 10 – Topologie interaktivního experimentu založeného na sdílené architektuře iLab.....	19
Obrázek 11 – Experiment „The force on a dipole“, prováděný na iLab	20
Obrázek 12 – Obecné schéma architektury WebLab-Deusto	22
Obrázek 13 – Příklady vzdálených experimentů vzdálené laboratoře iSES Remote Lab SDK.....	24
Obrázek 14 – Rozhraní platformy LiLa	26
Obrázek 15 – Snímek obrazovky Wonderland simulace: avatary před projekční plochou vestavěného webového prohlížeče	27
Obrázek 16 – Ovládací panel vybraného experimentu na Remote-LAB GymKT	28
Obrázek 17 – Zařízení pro jeden z experimentu na Remote-LABu GymKT	29
Obrázek 18 – Proces vývoje informačního systému dle metodiky RUP	35
Obrázek 19 – Diagram případů užití vzdálené laboratoře.....	40
Obrázek 20 – Analytický model tříd systému vzdálené laboratoře	44
Obrázek 21 – Návrhový model tříd vzdálené laboratoře	46
Obrázek 22 – Sekvenční diagram k případu užití „Schedule experiment“	47
Obrázek 23 – Sekvenční diagram k případu užití „Carry out experiment“	49
Obrázek 24 – Sekvenční diagram k případu užití „View experiment results“	50

Seznam tabulek

Tabulka 1 – Specifikace k případu užití „Schedule experiment“	41
Tabulka 2 – Specifikace k případu užití „Carry out experiment“	42
Tabulka 3 – Specifikace k případu užití „Manage users“	42

1 Úvod

Při studiu na technických a přírodovědeckých oborech laboratorní praxe hraje značnou roli a posiluje teoretické koncepty poskytnuté během přednášek a seminářů. Nicméně pro provádění laboratorních prací vzdělávací instituce musí udržovat desítky jednotek laboratorního vybavení, což vyžaduje značné prostory, personál zodpovědný za údržbu laboratoře a finanční prostředky pro neustálou podporu a modernizaci vybavení (Sierra-Fernández et al., 2021). Vzdělávací instituce proto provádí laboratorní výuku pouze na dostupném zařízení podle přísného harmonogramu s omezenou dobou využití, a to nikoli individuálně, ale kolektivně, což znamená, že se nezohledňují individuální schopnosti a potřeby každého studenta.

Řešením dané situace je vytvoření a rozvoj systému vzdáleného praktického vzdělávání, který je založen na vzdálených laboratořích. Jejich vznik je spojen s rychlým rozvojem informačních a komunikačních technologií (ICT): vývoj v oblasti telekomunikací přinesl vysokorychlostní připojení k internetu a také mobilní technologie 3G, 4G (LTE Advanced), kdy obrovské množství dat mohlo být dostupné během několika sekund. Podstata vzdálených laboratoří spočívá v poskytování možnosti vzdáleného provádění experimentů v míře podobné fyzické interakci. To znamená, že studenti, vyučující a všichni ostatní odborníci budou mít přístup ke svému laboratornímu vybavení, interagovat s ním a prohlížet si výsledky, zatímco nebude nutná jejich osobní přítomnost u laboratorního zařízení.

Poslední tři roky se problematika vzdáleného přístupu k laboratořím stala ještě víc aktuální kvůli pandemii vyvolané šířením viru SARS-CoV-2. Aby se snížilo riziko infekce, musely vzdělávací instituce omezit množství laboratorních cvičení, vyhrazený čas a počet účastníků. Dle odhadů světové organizace pro vzdělání, vědu a kulturu UNESCO, bylo v roce 2020 tímto problémem postíženo více než 220 milionů studentů škol a univerzit (UNESCO, 2021), což přispělo k hledání alternativních možností vzdělávání. Mezi takové možnosti například patří použití videomateriálů demonstrujících experiment nebo různých druhů simulátorů (virtuálních laboratoří). Daný přístup má však značnou nevýhodu – student není

účastníkem procesu, ale pouze jeho pozorovatelem, což může negativně ovlivnit pochopení látky a její následnou aplikaci. Vzdálené laboratoře nemají výše popsané nevýhody, jelikož nejen naznačují studentovi, jaké vybavení potřebují a jaké jsou postupy experimentu, ale také umožňují tyto postupy implementovat (Gustavsson et al., 2009).

Jednu z prvních vzdálených laboratoří předvedla Oregonská státní univerzita (Bhandari, 1998). V České republice má vzdálené laboratoře jen několik vzdělávacích institucí, například na Matematicko-fyzikální fakultě Univerzity Karlovy¹. Potřeba jejich dostupnosti je aktuální pro město Hradec Králové, zejména pro Univerzitu Hradec Králové. Vzdálené laboratoře by se mohly využívat na Fakultě informatiky a managementu, například v rámci předmětů Počítačové sítě nebo Internet V, a byly by obzvláště užitečné na Přírodovědecké fakultě pro fyzikální a chemické experimenty.

¹ <https://eedu.eu/>

2 Cíl práce

Kvůli extrémnímu růstu požadavků na využití laboratoře se instituce potýkají s výzvami, jako je plánování, zabezpečení a nabízení vzdáleného přístupu. Cílem diplomové práce je představit a analyzovat vzdálené laboratoře jako alternativu k tradičním laboratořím. Na základě analýzy současných řešení poskytnout vlastní návrh informačního systému pro přístup ke vzdálené laboratoři pro vzdělávací instituce, čímž by se usnadnily procesy plánování, zabezpečení a poskytnutí možnosti vzdáleného přístupu zmíněné výše. Pro modelování systému byly využity základní nástroje poskytované jazykem UML.

Očekávaným přínosem diplomové práce jsou jednak strukturované teoretické poznatky z oblasti vzdálených laboratoří, tak také představení existujících vzdálených laboratoří. Z praktického pohledu vytvořený návrh systému by měl posloužit jako univerzální podklad pro implementaci vlastní vzdálené laboratoře.

3 Alternativy ke klasickým laboratořím

Laboratoř je důležitým nástrojem pro podporu vzdělání studentů i vědeckých výzkumů. Její rozvoj a údržba však může být náročná, protože potřebuje významné finanční a logistické investice. Požadavek na fyzický přístup k laboratornímu vybavení tradičně omezoval flexibilitu zařízení a jeho sdílení mezi více uživatelů.

V posledních desetiletích technologický pokrok otevřel dveře distančnímu vysokému vzdělávání, jehož kvalita je považována za srovnatelnou s tradiční prezenční výukou (Hansson, 2008). Přestože v distanční formě přednášky a cvičení nezpůsobují příliš velké potíže (hodně učitelů využívá možnosti komunikačních softwarů, jako jsou MS Teams, Skype, Discord, WhatsApp, Zoom), tak provádění laboratorních prací se v jejich klasickém smyslu stává dost problematické. V rámci distančního vzdělávání existují různé přístupy, které jsou snahou o poskytnutí praktických laboratorních výzkumů a jsou alternativou ke klasickým laboratořím. Mezi takové možnosti patří virtuální laboratoře, mobilní experimentální zařízení a vzdálené laboratoře, kterým je věnována tato kapitola.

3.1 Virtuální laboratoře

Jedna z alternativ ke klasickým laboratořím je založena na simulaci, to znamená provádění experimentů a laboratorních prací na virtuální ploše, obvykle v cloudové aplikaci. Tyto aplikace se nazývají virtuálními laboratořemi. Virtuální laboratoře simulují nástroje, vybavení, testy a postupy používané v chemii, biochemii, fyzice, biologii a dalších disciplínách, které zahrnují laboratorní složku. Klíčovou charakteristikou virtuálních laboratořích je jejich interaktivita – znázorněním akcí, reakcí a důsledků manipulace s materiály a vybavením poskytují virtuální laboratoře studentům způsob, jak se zapojit do výukových aktivit v laboratoři bez režie fyzické laboratoře.

Virtuální laboratoře fungují online, často jsou integrovány do systému pro řízení výuky². Studenti mají přístup k virtuálním reprezentacím vybavení a zásob, které by našli v klasické laboratoři. Například ve virtuální chemické laboratoři mohou studenti najít kádinky, pipety, zkumavky, lahvičky s kyselinami, Bunsenovy kahany, váhy a další potřebné vybavení. Studenti postupují podle kroků zadaného postupu, pozorují a zaznamenávají výsledky. Studenti vidí digitální simulace výsledků svých akcí a pokud udělají chybu, tak jsou na to upozorněni. Většina virtuálních laboratoří umožňuje uživatelům zastavit, spustit a přehrát aktivitu. Mnoho virtuálních laboratoří se integruje do osnov kurzu, zaznamenává pokrok studentů a přiděluje kredity a hodnocení. Virtuální laboratorní aplikace a obsah v nich lze zakoupit od dodavatelů nebo vyvinout interně. Některé nástroje dodavatelů umožňují instruktorům přidávat nebo upravovat obsah tak, aby vyhovoval jejich kurzům. Množství technických dovedností, které učitelé potřebují k účasti na vývoji nebo přizpůsobení virtuálních laboratoří se liší v závislosti na použitých nástrojích.

Typickým příkladem virtuálních laboratoří je online projekt *STAR*³ (*Software Tools for Academics and Researchers*), program Massachusettského technologického institutu (MIT) pro vývoj virtuálních laboratoří pro vzdělávání a výzkum. Program zahrnuje vývoj výukových a výzkumných aplikací v obecné biologii, biochemii, genetice, hydrologii a v oblasti distribuovaných výpočtů. Většina aplikací je implementována v jazyce Java a HTML.

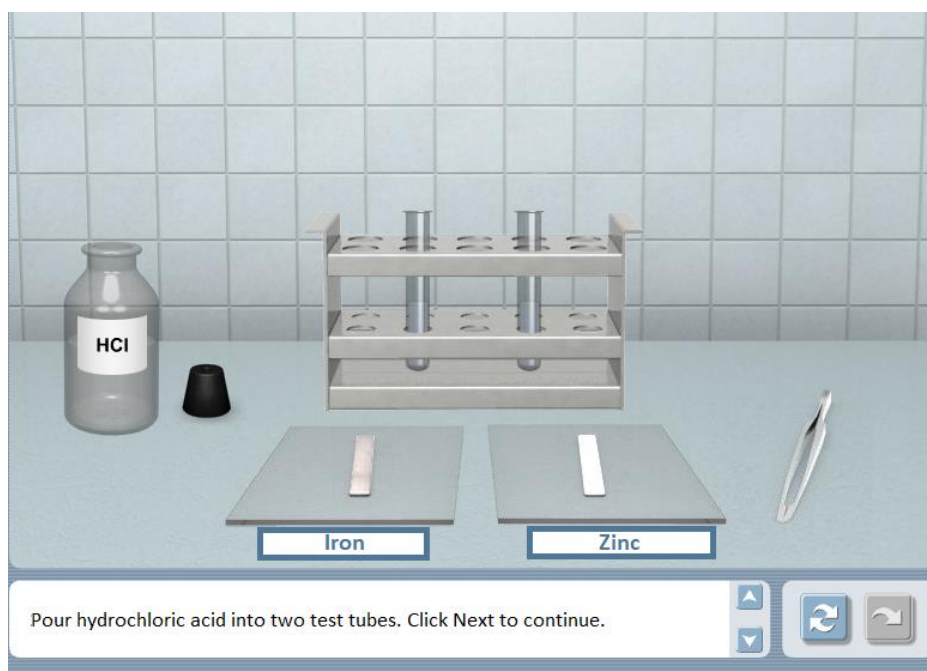
Dalším projektem rozvoje virtuální laboratorní práce pro studenty fyziky, chemie, biologie, ekologie je *VirtuLab*⁴. Tato virtuální laboratoř se liší úzkou

² **Systém pro řízení výuky (Learning management systém, LMS)** „patří mezi základní nástroje elektronického vzdělávání. Jsou to aplikace využívající řadu integrovaných nástrojů, které slouží ke komunikaci, správě jednotlivých kurzů, vkládání výukových materiálů nebo organizaci lekcí. Za běžné funkce systémů řízeného vzdělávání lze považovat následující moduly: evidence a správa žáků, evidence a správa kurzů, katalog výukových kurzů a objektů, správa studijních plánů, evidence hodnocení žáků“ (Kod'ousková, 2021).

³ <http://star.mit.edu/>

⁴ <http://www.virtulab.net/>

specializací, ve většině případů linearitou experimentu (postup a výsledky experimentu jsou předem dané). Každý experiment kromě vizualizace obsahuje podrobný popis včetně cílů laboratorní práce, postupu a doporučení pro vyučující. Zajímavou vlastností této virtuální laboratoře je, že u experimentů se zvýšenou úrovní nebezpečnosti, jako jsou například práce s kyselinami v chemii, jsou do postupu simulace zahrnuta vizualizovaná bezpečnostní pravidla. Produkty VirtuLab mají vzdělávací hodnotu a řeší problém provedení laboratorních prací při absenci potřebného vybavení, například „Studium interakce částic a jaderných reakcí“, „Vypočet měrné tepelné kapacity kovů“, „Studium Ohmova zákona pro celý obvod“. Na obrázku 1 je znázorněn snímek chemického experimentu (simulace) rozpouštění železa a zinku v kyselině chlorovodíkové, prováděný ve VirtuLab.

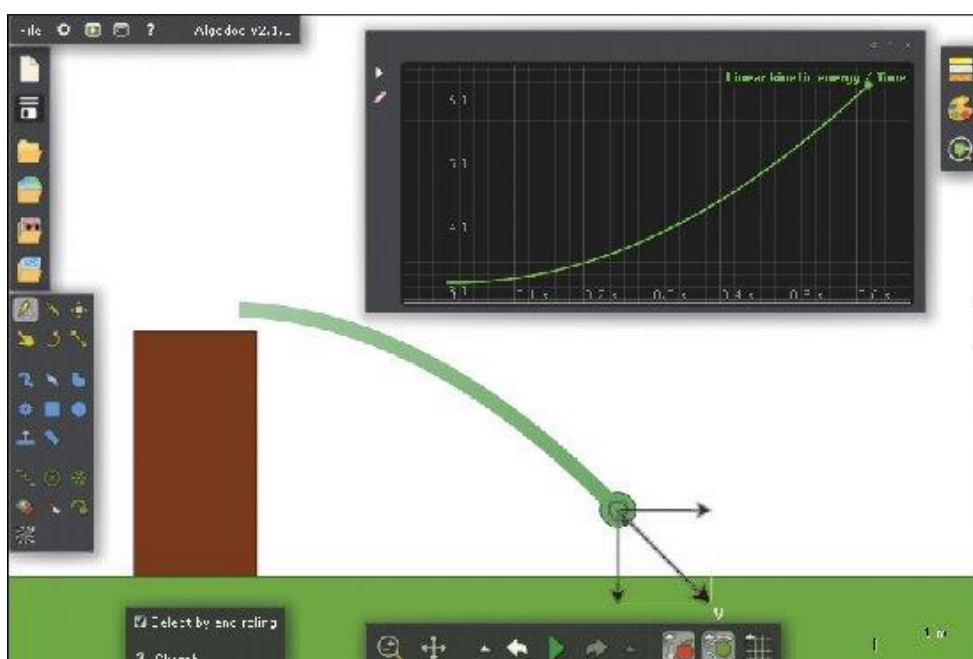


Obrázek 1 - Simulace rozpouštění železa a zinku v kyselině chlorovodíkové ve virtuální laboratoři VirtuLab.

Mezi virtuální laboratoře patří také 2D simulátor *Algodoo*⁵. Aplikace má bohatou sadu nástrojů pro 2D fyzikální simulace: vytváření různých objektů, mechanismů a systémů za účelem simulace jejich vlastností a fyzikální interakce.

⁵ <http://algodoo.com/>

Lze v ní například vytvořit model fungujících hodin, model roveru nebo vzduchovky. Program je schopen simulovat nejen mechanické procesy, ale i ty optické a možnost programování pomocí skriptovacího jazyka Thyme umožňuje vytvářet objekty s originálními fyzikálními vlastnostmi, různými funkcemi, efekty a jevy. Je také možné do něj nahrát obrázek, který se stane simulačním objektem a lze mu nastavit libovolné fyzické vlastnosti (Gregorcic, Bodin, 2017). Program poskytuje úložiště „algobox“, kde mohou uživatelé sdílet své modely. Aplikace je zcela zdarma.



Obrázek 2 - Simulace pohybu hmotného bodu v gravitačním poli ve virtuální laboratoři Algodoo

zdroj: Gregorcic, Bodin, 2017

Z představených příkladů plyne, že i přes řadu výhod, jako je bezpečnost provedení experimentů a možnost modelování a vizualizace procesů bez nutnosti pořizovat drahé zařízení, mají virtuální laboratoře dvě významné nevýhody: nejsou vázány na skutečnou laboratoř, a jak již bylo výše uvedeno, student není účastníkem procesu, ale je pouze jeho pozorovatelem. V některých oborech virtuální laboratoře neposkytují přesvědčivý a efektivní zážitek. Kromě toho komerční produkty pro virtuální laboratoře mohou být hodně nákladné a ne každá z nich poskytuje

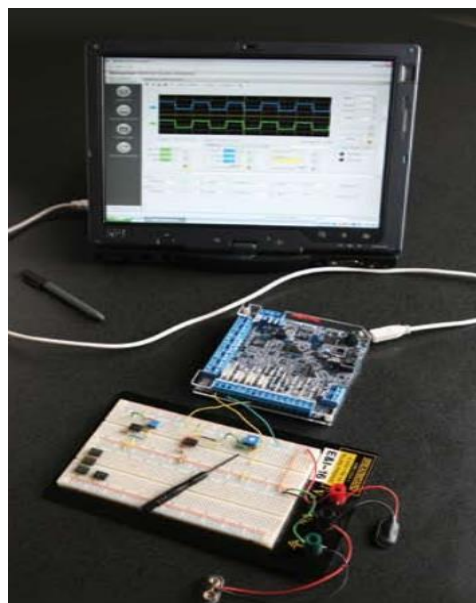
možnosti přizpůsobení specifickým potřebám zákazníka. U aplikací, které to umožňují, může tvorba a nastavování zabrat dost času, což přináší další náklady.

3.2 Přenosná zařízení pro praktické experimenty

Podstatou druhé alternativy k reálné laboratoře je poskytnout studentovi různé druhy zdrojů a technických řešení určených k provádění praktických experimentů mimo reálnou laboratoř a zvýšit flexibilitu během výzkumu (Monzo et al., 2021). Například v rámci projektu Mobile Studio⁶ byla vyvinuta nízkonákladová hardwarová platforma *Mobile Studio IOBoard™* (obrázky 3 a 4). Platforma je vhodná pro použití doma, ve třídě nebo ve vzdáleném prostředí. Po připojení k počítači přes USB port a v kombinaci se specializovaným softwarem *Mobile Studio Desktop™* zařízení poskytuje takovou funkcionalitu, která je podobná laboratornímu vybavení (osciloskop, generátor signálu, napájecí zdroje, digitální multimetr).



Obrázek 3 – Mobile Studio IOBoard™
zdroj: mobilestudioproject.com

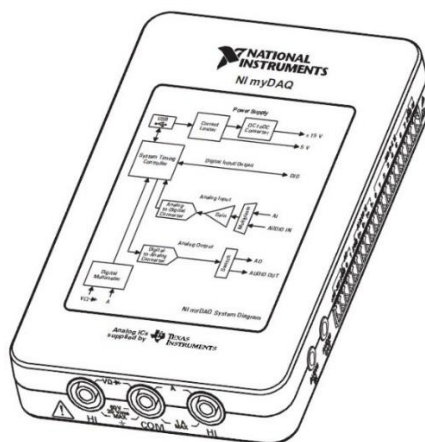


Obrázek 4 – Zapojení Mobile Studio IOBoard™ do tabletu
zdroj: transistor-man.com

⁶ <http://www.mobilestudioproject.com/>

Platforma Mobile Studio Desktop™ slouží pro vyučování fyziky, elektrotechniky, informatiky a lze ji využít i pro různé průmyslové a komerční účely. Účelem platformy je poskytnout možnost studovat principy, zařízení a systémy STEM (Science, Technology, Engineering & Mathematics) prakticky, což usnadňuje učení studentů.

Společnost National Instruments Corp. (USA) nabízí zařízení pro dálkové studium, jako je *NI myDAQ* (obrázek 5) a *Analog Discovery 2-NI Edition* (obrázek 6). NI myDAQ je přenosný učební nástroj navržený pro praktické experimentování v laboratorním a produkčním prostředí (Monzo et al., 2021). Kombinuje osm nejpoužívanějších laboratorních nástrojů typu plug-and-play, jako je digitální multimetr, osciloskop a funkční generátor. Student má k dispozici hotová softwarová zařízení určená pro experimenty a praktická cvičení s využitím Bodeho analyzátoru, generátoru funkcí DDS (přímé digitální syntézy), analyzátoru dynamického signálu (s podporou rychlé Fourierovy transformace), digitálních vstupů a výstupů. (National Instruments, 2011)



Obrázek 5 - NI myDAQ
zdroj: ni.com

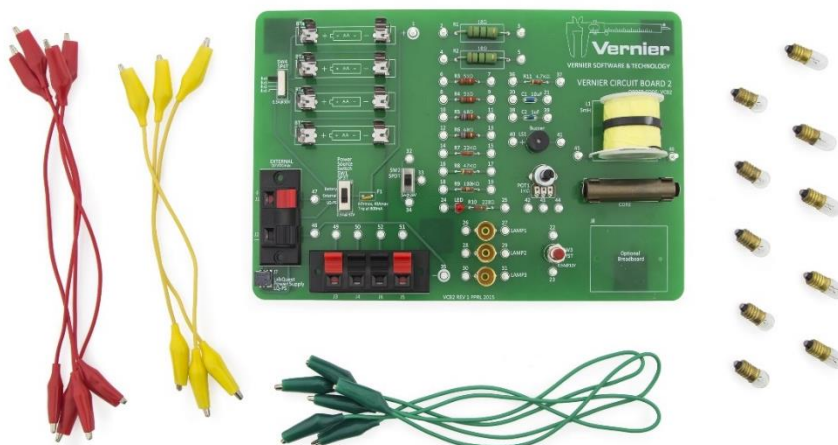
Analog Discovery 2 - NI Edition poskytuje studentům přístup k 100 MSa/s USB osciloskopu, USB logickému analyzátoru a šesti dalším přístrojům ve formě „kapesní laboratoře“. Analog Discovery 2 promění jakýkoli počítač v osciloskop a pracovní stanici pro elektrické navrhování. Toto zařízení napájené přes USB umožňuje studentům vytvářet a testovat analogové a digitální obvody v jakémkoli prostředí s funkcí tradičních stolních přístrojů. Kromě dvoukanalového

osciloskopu 100 MSa/s Analog Discovery 2 - NI Edition nabízí dvoukanálový generátor vzorů, šestnáctikanálový logický analyzátor, šestnáctikanálový generátor vzorů, spektrální analyzátor, síťový analyzátor a voltmetr (National Instruments, 2018).



Obrázek 6 – Analog Discovery 2 - NI Edition
zdroj: ni.com

Dalším příkladem „kapesní elektrotechnické laboratoře“ je zařízení od Vernier Software & Technology (USA) – *Vernier Circuit Board 2*⁷ (obrázek 7).

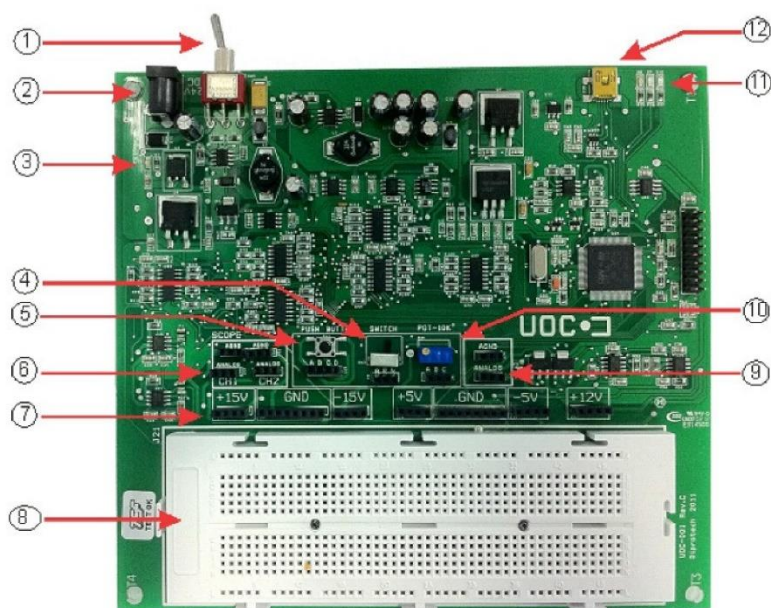


Obrázek 7 – Vernier Circuit Board 2
zdroj: vernier.com

⁷ <http://www.vernier.com>

Deska obsahuje sadu žárovek a kabelů, svorkovnici pro přidání vlastních komponent a předinstalované komponenty pro sledování chování sériových, paralelních a RLC obvodů.

Lab@Home je hardwarová platforma (obrázek 8) propojená s počítačem, která umožňuje tradiční laboratorní výzkumy v oblasti elektroniky, včetně sestavování elektronických obvodů a jejich testování, získávání analogového signálu a jeho měření (Monzo, 2020).



- | | | |
|--------------------------|------------------------|----------------------------|
| 1. ON/OFF main switch | 5. User push button | 9. Signal generator output |
| 2. 24V power supply jack | 6. Oscilloscope inputs | 10. User potentiometer |
| 3. ON/OFF indicator | 7. Power supplies | 11. Status indicator |
| 4. User switch | 8. Protoboard | 12. USB connector |

Obrázek 8 - Lab@Home

zdroj: Monzo, 2020

Výše uvedená hardwarová řešení umožňují provedení takových laboratorních prací, které nevyžadují komplexní a příliš drahé vybavení. Zpravidla jsou z oblasti výzkumu elektronických obvodů a jejich komponent, ale nejsou tím omezená: pro jednoduché chemické experimenty, jako je například měření teploty a pH chemických roztoků, existuje více přenosných zařízení od různých výrobců. Nicméně výběr zařízení pro experimenty z neelektronické oblastí není tak široký, což je nejspíše způsobeno zvýšenými bezpečnostními riziky při provádění takových

experimentů. Navíc mnoho vědeckých, technologických a inženýrských experimentů vyžaduje objemné a drahé laboratorní vybavení a pečlivě vybraná laboratorní prostředí. Například i pro provádění nejzákladnějších optických experimentů je nezbytné mít alespoň optický stůl s izolací vibrací, čočky, lasery a zrcadla (Zaman, 2021). Pro takové účely přenosná zařízení nejsou postačující a je potřeba najít komplexnější řešení.

3.3 Vzdálené laboratoře

Nejúplnějším řešením problému distančního vzdělávání v jeho praktickém paradigmatu by byla transformace celého prostředí skutečné laboratoře do její online verze. Tento požadavek splňuje další přístup k distanční výuce – tvorba a rozvoj vzdálených laboratoří.

Jak je patrné z názvu, vzdálená laboratoř je platforma, pomocí které její uživatelé vzdáleně přistupují ke svému laboratornímu prostředí a která představuje softwarový a hardwarový komplex (firmware), který umožňuje provádět experimenty bez fyzického kontaktu s reálnou instalací. Systém je přístupný prostřednictvím internetového prohlížeče, nebo pomocí aplikace a může být dostupný jak v rámci lokální sítě instituce, tak i zvenku, například z domova uživatele. Aby laboratoř byla snadněji dostupná, komunikace mezi laboratořemi a jejich uživateli je postavena na síti Internet, jelikož je nejrozšířenější počítačovou sítí, používanou v domácnostech, výukových institucích a firmách po celém světě. Technologie vzdálené laboratoře řeší většinu již zmíněných problémů, obav a potíží, se kterými se setkávají tradiční laboratoře.

Systém vzdálené laboratoře se skládá ze skutečné laboratoře, softwaru a hardwaru pro správu zařízení a digitalizace získaných dat a komunikačních nástrojů. Firmware vzdálené laboratoře používá architekturu klient-server. Klientskou stranou je webové rozhraní (zpravidla webový prohlížeč) pro usnadnění interakce uživatele se serverem. Strana serveru poskytuje experimentální nastavení (fyzický hardware) a také prostředí odpovědné za návrh, tvorbu a údržbu virtuální laboratoře. Vzdálený řídicí systém zajišťuje fungování komponent a rozhraní celého systému (Gerža, 2013). Pokud je pro experimenty nezbytná nebo

by byla užitečná interakce s někým ve fyzickém prostředí laboratoře v reálném čase, lze nastavit obousměrnou komunikaci, například formou videokonference.

Sám koncept vzdálené laboratoře není moc nový. Přesný rok jejich vzniku není znám, ale ve zkoumané literatuře se naznačuje, že první vzdálené laboratoře se objevily v polovině 90. let. Například na Univerzitě Tennessee v Chattanooga byla v tomto období založena laboratoř, která podporovala různé dálkově řízené laboratorní činnosti v řídicích systémech, chemickém inženýrství a strojírenství (Salihbegovic, Tanovic, 2008). Systém využíval desktopový software LabVIEW na každé řídicí stanici. Laboratoř byla přístupná přes internet a experimenty probíhaly v dávkovém (angl. *batch*) režimu.

V roce 1999 byla zřízena laboratoř pro dálkové ovládání fyzikálních experimentů, zabývajícími se určením rychlosti světla (Enloe et al., 1999). Ve stejném roce byla na Technologické a přírodovědecké univerzitě v Norsku (NTNU) vytvořena vzdálená laboratoř pro experimenty s polovodiči (Shen a et., 1999) Obě laboratoře byly určeny pro studenty vysokých škol. S časem a narůstající četností používání se vzdálené laboratoře staly komplexnějšími. Ukázalo se, že jejich použití poskytuje výhody, které běžné laboratoře nemají, jako je flexibilita přístupu, sdílení zdrojů a laboratoří a zabezpečení pro uživatele, data a zařízení (Lowe, 2013).

Současnou generaci vzdálených laboratoří lze rozdělit do dvou hlavních typů: dávkové laboratoře a interaktivní laboratoře.

Dávkové laboratoře jsou založeny na tom, že uživatelé zasílají popisy experimentálních operací, poté se experiment zařadí do fronty, a jakmile je hardware k dispozici, experiment se spustí asynchronně jako „dávková“ úloha (Harward et al., 2008). Výsledky jsou zaznamenány a později načteny uživatelem, který zadal experiment. V tomto scénáři uživatel vůbec neinteraguje s experimentální operací během jejího provádění a celková úroveň využití hardwaru je omezena pouze algoritmy zpracovávajícími frontu.

Interaktivní vzdálené laboratoře jsou běžnější a na rozdíl od dávkových laboratoří umožňují uživatelům synchronně monitorovat a upravovat běžící experiment. Očekává se, že uživatel bude vázán na laboratorní zdroj v době, kdy je tento zdroj dostupný. Tento distribuční proces je relativně standardním problémem plánování zdrojů. Plán rozvrhování konkrétní laboratoře bude záviset na řadě faktorů, včetně počtu uživatelů, počtu dostupných laboratorních zařízení, typické době trvání (a variabilitě) používání a záruky přístupu poskytované uživatelům. Problém optimalizace přístupu do vzdálené laboratoře je podrobněji popsán v práci Davida Lowe (2013).

Vlastnosti a hlavní výhody technologie vzdálené laboratoře jsou:

- *Vzdálený přístup.* Umožňuje jednotlivcům přístup do laboratoře bez nutnosti osobní přítomnosti. To zpřístupní laboratoř osobám, které jsou daleko od místa laboratoře, nebo když jejich přítomnost není povinná. Vzdálený přístup může také snížit obavy o bezpečnost a zabezpečení.
- *Přístup k vzácným vědeckým přístrojům.* Některé nástroje jsou tak drahé a specializované, že jich může existovat jen několik, nebo dokonce jen jeden. Poskytnutí vybavení, které studentům umožní zapojit se do autentických výukových zkušeností a ne jen do simulací nebo předem připravených cvičení využívajících archivní data, vytváří lepší příležitosti k učení. Práce se skutečnými nástroji oboru prohlubuje porozumění studentů konceptům v práci a připravuje je na kariéru s používáním těchto zařízení.
- *Úspora času.* Vzhledem k tomu, že není potřeba fyzicky navštěvovat laboratoře, ušetří se dost času, a místo toho, aby jednotlivci strávili čas dopravou k laboratořím, mohou ho věnovat hodnotnějším úkolům, jako je experiment a výzkum.
- *Rozvržení.* Laboratoř je přístupná v konkrétních datech a časech kvůli dané pracovní době, školní době a osobnímu časovému omezení. Systém vzdálené laboratoře je zpravidla přístupný neomezeně, přičemž více jednotlivců může pracovat se stejným laboratorním vybavením.

- *Efektivita nákladů.* Potřeba poskytovat stejné zařízení pro různé uživatele na různých místech vyžaduje velké náklady. V systému vzdálené laboratoře stačí mít jednu jedinou laboratoř a uživatel může přistupovat k požadovanému vybavení bez ohledu na to, kde se nachází.

Platforma vzdálené laboratoře má však také své *nevýhody* nebo *rizika*, mezi které patří otázka bezpečnosti a nákladů. Implementace a provoz specializovaného vybavení je nákladný a míra, do jaké přístroj vyžaduje dohled techniků na místě, tyto náklady zvyšuje. Každá organizace, která zpřístupňuje své nástroje, si to musí uvědomovat a počítat se zvýšenými náklady, které tento přístup nese. Na provoz některých zařízení musí dohlížet technici, což může zkomplikovat plánování experimentů. V praktických experimentech, které by měly být prováděny s lidskou interakcí nebo v případech, kdy jednotlivci potřebují provést některá nastavení a konfiguraci na zařízeních, nelze platformu vzdálené laboratoře považovat za naprosto vhodnou možnost, ledaže by k ní byli přiděleni i laboratorní asistenti. Proto musí být rozhraní navržena s ohledem na tyto problémy a omezovat přístup k funkcím, které by mohly neúmyslně poškodit přístroj.

3.4 Shrnutí

V této kapitole při použití odborné literatury byly definované a popsány tři alternativy ke klasickým fyzickým laboratořím. První variantou jsou virtuální laboratoře, které nemají přístup k reálným laboratořím, ale pouze simulují postupy experimentů, jako jsou STAR, VirtuLab, Algodoo. Jejich výhodou je, že jsou bezpečné, nepotřebují drahé fyzické zařízení a některá řešení jsou zcela zdarma. Nehledě na zvyšující se úroveň podobnosti reálnému světu se virtuální laboratoře nemohou rovnat plnému zážitku z práci se skutečnou laboratoří. Při provádění experimentů uživatel je pouze pozorovatelem, a proto jsou vhodné spíše pro demonstrační účely. Navíc většina platforem dostupných na trhu buď vůbec nejsou přizpůsobitelné, nebo jsou, ale za příplatek.

Další variantou jsou přenosná zařízení (například Mobile Studio IOBoard, NI myDAQ), která umožňují provedení takových laboratorních prací, které nevyžadují

složité a příliš drahé vybavení. Zároveň však kvůli ceně a dalším praktickým omezením nemůže být takový vědecký hardware zpřístupněn každému studentovi.

Třetí variantou jsou vzdálené laboratoře, které, na rozdíl od virtuálních, jsou propojené se skutečnou laboratoří. Této variantě je věnována zbývající část dané diplomové práce, jelikož je považována za nejvhodnější alternativu klasickým laboratořím pro vzdělávání na dálku.

4 Příklady vzdálených laboratoří

V této kapitole je představen přehled současných interaktivních vzdálených laboratoří, vytvořených ve Spojených Státech, Austrálii a v České republice.

4.1 iLabs

Projekt iLabs je systém vzdálené správy laboratoří vyvinutý v MIT v roce 1998 a podle autorů G. K. Herringa, L. T. Neustocka a L. Hesselinka (2018) byla první laboratoří, která se ovládala přes webový prohlížeč. Přes tuto platformu se provádějí experimenty z vědních oborů jako je fyzika nebo chemie. Byl to první systém vzdálené laboratoře, který zavedl koncept federace laboratoří: iLabs umožňuje sdílení laboratoří na úrovni architektury, rozděluje architekturu na prvky přítomné v kampusu na straně studentů a prvky přítomné v kampusu na straně laboratoře (Hardison et al., 2008). Je založen na sdílené architektuře iLab (Shared Architecture, ISA).

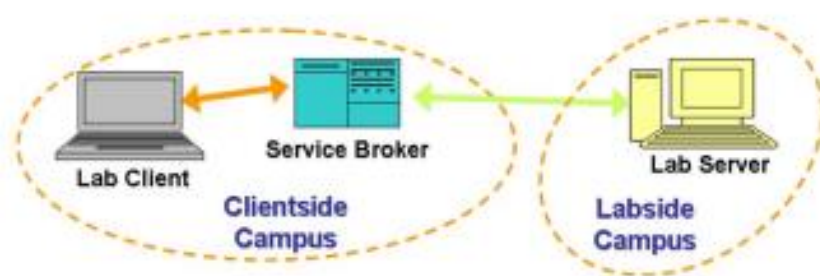
Dle architektury ISA každá online laboratoř je rozdělena na tři části:

- Lab Client (klient laboratoře),
- Service Broker (zprostředkovatel služeb)
- Lab Server (server laboratoře).

Lab Client je uživatelské rozhraní k iLabs, zatímco Lab Server se připojuje k laboratornímu hardwaru a spravuje provádění experimentů odeslaných uživatelem. Service Broker je zodpovědný za poskytování funkcí, které jsou generické (obecné) a užitečné pro všechny laboratoře iLab. Do této vrstvy middlewaru jsou zabudovány služby, jako jsou ukládání dat, autentizace a autentifikace uživatele. ISA poskytuje framework pro nasazení iLab distribuovaným způsobem pomocí webových služeb. To umožňuje, aby online laboratoře, vyvinuté na ISA, byly zpřístupněny uživatelům po celém světě pomocí standardních síťových protokolů.

ISA podporuje dva typy laboratoří (Hardison et al., 2008): dávkové a interaktivní laboratoře, které mají různé topologie a mechanismy plánování.

Dávkové iLabs jsou nasazeny s klienty Lab Client, zprostředkovateli Service Brokers a servery Lab Server, které komunikují prostřednictvím sítě internet pomocí webových aplikací. V tomto modelu, který je znázorněn na obrázku 9, spolu Lab Client a Lab Server komunikují výhradně prostřednictvím zprostředkovatele iLab (Service Broker). To zajišťuje, aby byla komunikace mezi těmito systémy standardizovaná: jakákoli vzdálená laboratoř dávkového typu, která je kompatibilní s iLab, bude kompatibilní a sdílená prostřednictvím jakéhokoli zprostředkovatele dávkových služeb na světě.

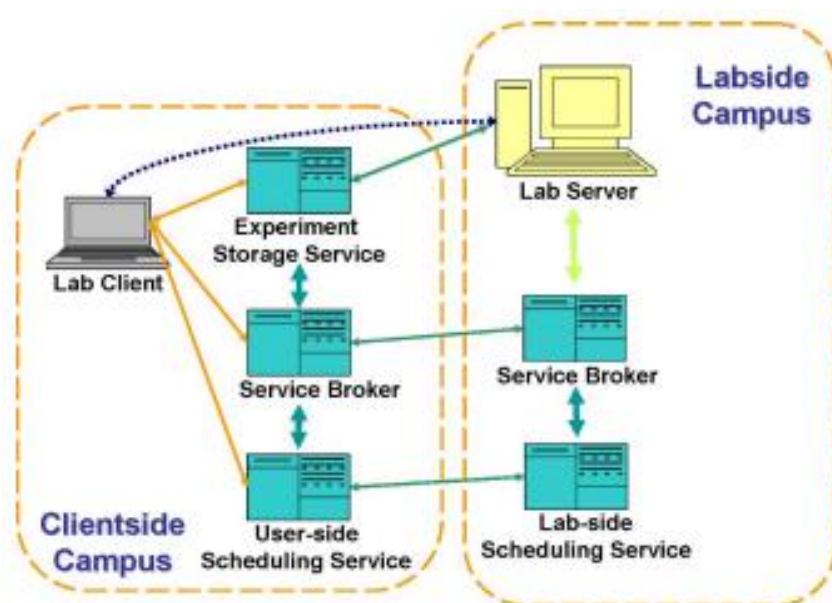


Obrázek 9 – Topologie dávkového experimentu založeného na sdílené architektuře iLab

zdroj: Hardison et al., 2008

Z uživatelského pohledu ve srovnání s dávkovými iLabs poskytuje interaktivní iLab vyšší stupeň kontroly: interaktivní experimenty vyžadují kontrolu laboratorního hardwaru, mezitímco uživatel nastavuje parametry a sleduje výsledky. Jak již bylo zmíněno v kapitole 3, v tom je rozdíl interaktivního modelu od dávkového, kde jsou experimenty řazeny do fronty a spuštěny až tehdy, když jsou k dispozici laboratorní zařízení (zdroje). Interaktivní experiment pak musí poskytnout laboratorní hardware jednotlivému uživateli na dobu trvání jeho sezení – obvykle 20 minut až hodinu – a také vyžaduje, aby sezení bylo předem naplánováno. Dalším klíčovým rozdílem mezi interaktivními a dávkovými laboratořemi je role Server Broker. Interaktivní experimenty vyžadují kontrolu v reálném čase a potenciálně mnohem rozsáhlejší šířku pásma (bandwidth) mezi laboratorním klientem a laboratorním serverem. Z tohoto důvodu v interaktivním iLab nebude Server Broker dávkové laboratoře, který používá webové služby ke směrování veškeré komunikace mezi laboratorním klientem a serverem, efektivně fungovat.

Topologie interaktivní iLab je znázorněna na obrázku 10. Jak je vidět, kromě Lab Client, Service Broker a Lab Sever jsou navíc přidány samostatné webové služby pro ukládání a správu experimentů a plánování laboratoří. Stejně jako v dávkovém modelu, v interaktivní laboratoři uživatel začíná svou relaci přihlášením přes Service Broker na své univerzitě a systém provádí jeho autentifikaci. Student si nejprve musí naplánovat čas, kdy bude laboratoř používat. V naplánovanou dobu se student přihlásí a je umožněn spustit klienta laboratoře. Tento klient komunikuje přímo s testovacím serverem, a jakmile se spustí testovací relace, Service Broker vystoupí z procesu.



Obrázek 10 - Topologie interaktivního experimentu založeného na sdílené architektuře iLab

zdroj: Hardison et al., 2008

Pro lepší přehlednost se uvádí příklad interaktivního experimentu „The force on a dipole“ (Síla působící na magnetický dipól). Experiment je navržen pro studenty prvního ročníku, studující fyziku na MIT. Tento experiment je součástí širšího projektu na rozšíření kurzů fyziky v prvním ročníku o praktické i vzdálené experimenty spojené s vizualizacemi za účelem ilustrování konceptů, které mohou být zcela abstraktní. Experiment se skládá z malého magnetu zavěšeného na pružině mezi dvěma vodorovně uloženými cívkami (Helmholtzovy cívky). Prostřednictvím uživatelského rozhraní (část Lab Client ve výše zmíněném

modelu), založeného na technologii LabVIEW mohou studenti měnit proud v cívkách tak, aby magnet osciloval. Laboratorní zařízení a snímek uživatelského rozhraní jsou znázorněny na obrázku 11. Na základě svých měření a několika známých systémových parametrů mohou studenti určit magnetický dipólový moment magnetu a pomocí vizualizací a videa laboratorního vybavení lépe pochopit praktické fungování elektromagnetických sil (Scheucher et al., 2009).



Obrázek 11 - Experiment „The force on a dipole“, prováděný na iLab
zdroj: Scheucher et al., 2009

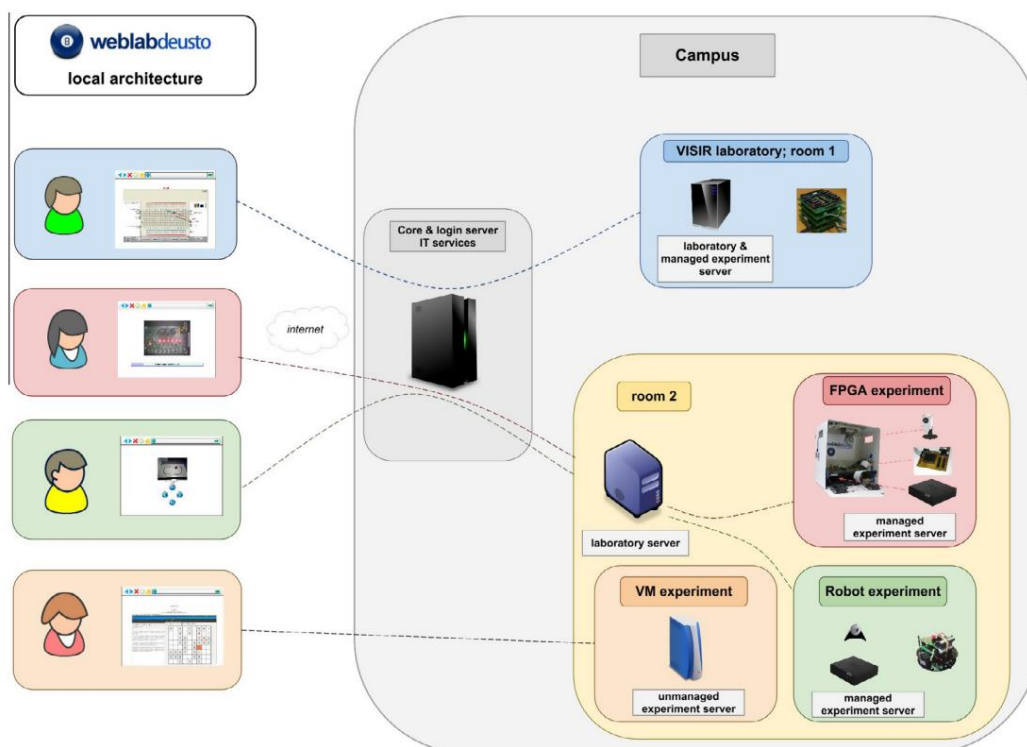
Za zmínku stojí, že mezi absolventy Stanfordské univerzity byl proveden online průzkum (Zaman, Neustock, Hesselink, 2021) s cílem posoudit kvalitu platformy iLabs jako vzdělávacího nástroje. K hodnocení se používala stupnice od 1 do 5, kde 1 znamenalo souhlas/pozitivní názor a 5 znamenalo nesouhlas/negativní názor. Názor absolventů byl celkově pozitivní, průměrné hodnoty se pohybovaly od 1,58 do 2,63. Zpětná vazba na intuitivnost uživatelského rozhraní byla výhradně pozitivní (průměr: 1,74, směrodatná odchylka: 0,55), jinými slovy nebyly žádné negativní recenze. Většina absolventů také kladně hodnotila účinnost a snadnost použití (průměrná hodnota účinnosti: 2,37, směrodatná odchylka: 0,65; průměrná hodnota snadnosti použití: 2,63, směrodatná odchylka: 1,09). Mnoho studentů také považovalo iLabs jako skvělý doplňkový výukový nástroj (průměr: 1,58, směrodatná odchylka: 0,75). Většina se také domnívala, že platforma iLabs povzbuzuje studenty k provádění experimentů a že díky své cenové dostupnosti může iLabs zvýšit počet poskytovaných experimentů.

Na základě průzkumu lze usoudit, že design platformy iLabs a její obecné určení splňují požadavky na vzdálené laboratoře.

4.2 WebLab-Deusto

WebLab-Deusto je systém vzdálené správy laboratoří vyvinutý na Univerzitě Deusto v roce 2005. Tak jako každá vzdálená laboratoř má i WebLab-Deusto architekturu klient-server (viz obrázek 12). Pro autentizaci studenti komunikují s přihlašovacím serverem (Login Server) a pro zbytek operací se základním serverem (Core Server). Mezi funkce Core serveru patří autentifikace uživatelů, plánování a ukládání informací zaslanych studenty a předání všech požadavků příslušné laboratoře. Za Core Serverem definuje architektura WebLab-Deusto další dva servery: laboratorní server (Laboratory Server) a experimentální server (Experiment Servers). První z nich má být umístěn v laboratorní místnosti, kde se nachází několik experimentálních serverů. Jeho hlavním účelem je zajistit bezpečnost experimentálních serverů a také pravidelně kontrolovat, zda stále fungují. Experiment Server je server, který fyzicky spolupracuje s hardwarem. Při nasazení experimentálních serverů lze použít těchto jazyků a technologií: C++, C, Java, Python, LabVIEW. Veškerá interakce mezi klientem a servery je odesílána prostřednictvím webových služeb. Používají se buď SOAP, JSON nebo HTTP a XML-RPC (Orduña et al., 2014).

Pomocí WebLab-Deusto lze vybudovat dva typy laboratoří: řízené a neřízené laboratoře. V řízených laboratořích je veškerá interakce mezi studentem a laboratoří řízena prostřednictvím všech popsaných vrstev. Ukládání je vyřešeno v Core serveru, a proto Experiment Server nemusí být na síťové úrovni veřejně dostupný. Nicméně v řízených i neřízených laboratořích musí být implementován protokol založený na příkazech. V neřízených laboratořích Experiment servery spravují další platformy, jako jsou virtuální stroje nebo vzdálené panely LabVIEW (Orduña et al., 2011). Studenti proto stále pracují se všemi vrstvami, ale hlavní interakci zajišťuje externí platforma (například virtuální počítač nebo vzdálená plocha). WebLab-Deusto nezodpovídá za zabezpečení a ukládání interakcí studentů.



Obrázek 12 – Obecné schéma architektury WebLab-Deusto

zdroj: Orduña et al., 2014

System plánování je založen na pluginech. Hlavním použitým pluginem je systém prioritních front. Systém přijímá různé kopie stejné laboratoře, čímž vyrovnává zatížení uživatelů mezi různými kopiemi, takže fronty lze snížit přidáním dalších kopií. Aby bylo možné federovat různé instance WebLab-Deusto, byl vytvořen nový plánovací plugin, který spoléhá na sadu externích instancí WebLab-Deusto. Každá instituce může stanovit vyšší prioritu pro svoje studenty (Orduña et al., 2014).

4.3 iSES Remote Lab SDK

System iSES (internet School Experimental studio) Remote Lab SDK je softwarovým studiem pro tvorbu vzdálených experimentů online s přístupem přes počítače i mobilní zařízení (Dvořák, Kuriščák, Lustig, 2022). Firma vyvíjí vzdálené laboratoře od roku 2012. První generace vzdálených laboratoří byly vytvořené na Java apletech. Ale od roku 2012 společnost Sun Microsystems (později Oracle), poskytovatel Javy, začala omezovat Java aplety kvůli objevujícím se bezpečnostním dírám. Proto se od roku 2013 při navrhování vzdálených experimentů začal

používat JavaScript a nový softwarový balík byl pojmenován „iSES Remote Lab SDK“.

Vzdálený experiment v rámci „iSES Remote Lab SDK“ je aplikace typu klient-server. Na straně klienta je počítač (nebo mobilní zařízení) s prohlížečem, který podporuje skriptovací jazyky a skripty musí být povoleny v nastavení prohlížeče. Na straně serveru je připojený k internetu počítač s experimentem. K němu je připojeno měřicí zařízení iSES nebo jiné měřicí systémy a zařízení s rozhraním RS 232 nebo USB, jako jsou například krokové motory, říditelné zdroje, multimetry (Schauer, Lustig, Ožvoldová, 2009). Skutečný experiment se také provádí ze strany serveru. Server provozuje i speciální programy, jako je MeasureServer a WEB server.

MeasureServer – to je server, který komunikuje s hardwarem měřicího zařízení. Jedná se o speciálně vytvořený program, který komunikuje se senzory, jako jsou sady iSES. MeasureServer také obsahuje vlastní logovací systém. V něm je možné spravovat vzdálené experimenty a měřicí informace každého uživatele. Do systému se zapisuje každá provedená operace s experimentem včetně podrobného popisu postupu měření daného studenta nebo vyučujícího. Z logovacích záznamů lze vyčíst, zda jsou přístupy na experimenty jenom „zkušební, testovací, bez úmyslu experiment naměřit a získat data z experimentu. Největší přínos ze vzdálených experimentů mají uživatelé, jejichž záznamy hlásí, že si experiment nejdříve otestují, potom naměří smysluplná data a nakonec si naměřená data stáhnou do svých počítačů přes Internet“ (Schauer, Lustig, Ožvoldová, 2009).

Jako WEB server se používá volně poskytovaný Nginx, který mimo jiné ukládá pro experiment vlastní webové stránky napsané pomocí HTML s vestavěnými JavaScript widgety z nové sady iSES Remote Lab SDK. V případech, kdy je potřeba mít podporu IP kamery, je spuštěn ImageServer (je součástí iSES Remote Lab SDK), který přenáší obraz streamovou metodou.



Obrázek 13 – Příklady vzdálených experimentů vzdálené laboratoře iSES Remote Lab SDK

zdroj: <https://www.ises.info/index.php/cs/systemises/sdkisesstudio>

Vzorové vzdálené experimenty vytvořené pomocí iSES Remote Lab SDK jsou umístěny na webu laboratoře iSES⁸ a na Matematicko-fyzikální fakultě Univerzity

⁸ <http://www.ises.info/index.php/cs/laboratory>

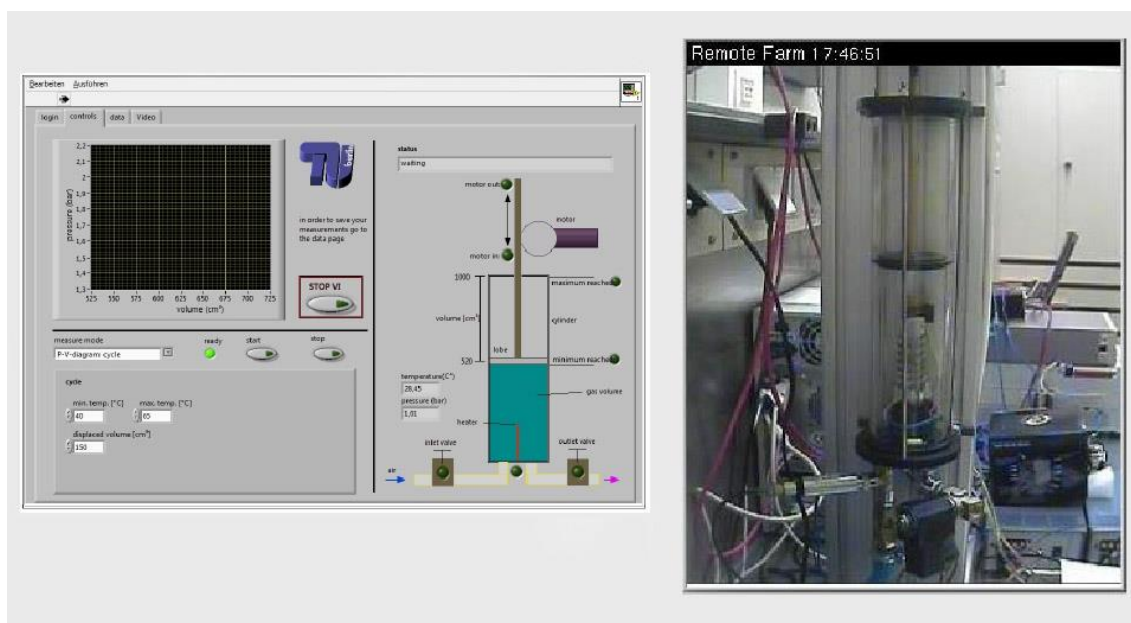
Karlove, kde je pomocí videí představeno jak se s experimenty pracuje. V současné době se pro střední a vysoké školy zdarma nabízí osmnáct vzdálených experimentů. Mezi ně například patří Usměrňovač, Mapa stanic monitorujících přirozené radiální pozadí, Meteorologická stanice, Ohyb na štěrbině – Heisenbergův princip, Vlastní a vynucené kmity na pružině apod. Na obrázku 13 jsou znázorněné některé z poskytovaných experimentů.

4.4 LiLa

Vývoj projektu LiLa (zkratka pro „Library of Labs“) probíhal od roku 2009 do roku 2011 a byl financován Evropskou komisí v rámci programu eContentplus. Primární cílovou skupinou LiLa jsou vyučující na vysokých školách a studenti bakalářského a magisterského přírodovědeckého a inženýrského studia. Vyučujícím by LiLa měla usnadnit přístup k laboratorním zařízením pro demonstrace experimentů. Projektové konsorcium zahrnuje jedenáct partnerů (univerzity a podniky) ze sedmi evropských zemí a při vývoji projektu byly řešeny následující úkoly:

- vytvořit technickou a organizační základnu pro vzájemnou výměnu experimentů v Evropě i mimo ni;
- vybudovat úložiště experimentů na centrálním serveru, odkud lze získávat výsledky experimentů;
- prezentovat experimenty ve formě opakovaně použitelných modulů, které lze provozovat v místních systémech pro řízení výuky a také na portálu LiLa;
- vybavit experimenty metadaty (popisem dat) a integrovat je do knihovných katalogů;
- zpřístupnit experimenty učitelům a studentům prostřednictvím systému kontroly přístupu a rezervace;
- poskytnout uživatelům možnost přidávat a měnit kurzy;
- sjednotit experimenty LiLa do struktury, která podporuje spolupráci mezi studenty a poskytuje příležitost k diskusi o experimentech.

Jinými slovy, cílem LiLa je poskytnout studentům možnost využívat na dálku jedinečná a drahá zařízení, jako jsou například ta, která se nacházejí ve vědeckých laboratořích v Basileji (Švýcarsko) a Soluni (Řecko), ke kterým by jinak neměli přístup (Richter, Boehringer, Jeschke, 2011). Studenti získají více příležitostí k experimentování, lepší porozumění a získají cenné dovednosti. Diskusní fórum umožní komunikaci v reálném čase mezi studenty po celé Evropě a mezi studenty a jejich vyučujícími.



Obrázek 14 – Rozhraní platformy LiLa
zdroj: Richter, Boehringer, Jeschke, 2011

V současnosti lze prostřednictvím portálu LiLa (obr. 14) provádět více než čtyřicet vzdálených experimentů v oblastech jako je fyzika, chemie a elektronika. LiLa poskytuje integrovaný webový portál, který uživatelům umožní najít vzdálené experimenty a virtuální laboratoře, získat k nim přístup, využít doprovodný výukový systém a najít doplňkové výukové materiály. Portál LiLa navíc vytvořil organizační rámec pro výměnu experimentů mezi institucemi a pro přístup k experimentálním nastavením. Pro zapojení vzdálených a virtuálních experimentů do sylabů předmětů LiLa poskytuje didaktickou pomoc pro lektory a šablony smluv pro instituce. Portál také obsahuje další funkce, například vyhledávací systém založený na běžných sadách metadat, integrovaných do knihovních katalogů. Řízení přístupu je řešeno systémem správy uživatelů a rezervačním systémem, potřebným

pro provádění vzdálených experimentů. LiLa se také zabývá vytvářením 3D prostředí pro společné použití vzdálených experimentů a simulací, založeného na „Wonderland“ společnosti Sun Microsystems (obrázek 15), nyní vlastněném společností Oracle.



Obrázek 15 – Snímek obrazovky Wonderland simulace: avatary před projekční plochou vestavěného webového prohlížeče

zdroj: Richter, Boehringer, Jeschke, 2011

4.5 Vzdáleně řízená laboratoř GymKT

Vzdáleně řízená laboratoř GymKT (Gymnázium Jaroslava Vrchlického Klatovy) je první středoškolskou vzdálenou laboratoří v Plzeňském kraji a první gymnaziální vzdálenou laboratoří v České republice, která byla vytvořena pro praktické vyzkoušení experimentální úlohy navržené v rámci diplomové práce na technické univerzitě v Liberci (Panzenberger, 2021). Časem se přidávaly další experimenty, které jsou především zaměřené na praktickou výuku fyziky. V současné době vzdálená laboratoř GymKT umožňuje provádět třináct experimentálních úloh, mezi které například patří určení horizontální složky magnetického pole Země, vzdálené ovládání robotické paže, studium difrakčních jevů. Všechny tyto experimenty jsou dostupné pro všechny zájemce 24/7 a jsou zcela zdarma. Ke každému experimentu se uvádí podrobné teoretické poznatky, včetně popisu aparatury a také informace o tom, pro koho je daný experiment vhodný. Experimenty jsou dostupné ve více jazycích: v češtině, angličtině, němčině, španělštině, slovenštině a ruštině. K úlohám se přistupuje přes internetový prohlížeč, který musí podporovat

a mít povolen servery JavaScript a AJAX. Pokud je dána úloha dostupná, lze vzdálený experiment hned spustit. V opačném případě je potřeba si zarezervovat experiment na daný termín přes rezervační formulář. Formulář je dostupný pouze pro autorizované uživatele.

Pro přehlednost je proces provádění experimentu na GymKT představen na vybraném experimentu, určeném pro sledování teplotní závislosti odporu kovu a polovodiče. Na obrázku 16 je představen snímek ovládacího panelu experimentu. Panel je rozdělen na několik sekcí: v sekci „Ohřev“ si uživatel nastaví teplotu pomocí 12 připravených hodnot, webová kamera zobrazuje displej digitálního teploměru s teplotou měřených prvků. Tuto teplotu (t) je potřeba odečíst z displeje a vložit do tabulky naměřených, dvě další hodnoty se vypočítávají systémem a taky se vkládají do tabulky, kterou lze exportovat do CSV souboru. Experiment lze dokončit buď kliknutím na „Ukončit měření a opustit aparaturu“, nebo automaticky po uplynutí stanoveného času (u daného experimentu je to 30 minut).

Obrázek 16 – Ovládací panel vybraného experimentu na Remote-LAB GymKT

Pokud jde o technickou stránku laboratoře, pak jsou momentálně všechny experimentální úlohy ovládané experimentální USB deskou K8055 od firmy Velleman. Do budoucna autoři plánují u stávajících experimentů využít modul

Arduino s rozšířením Ethernet Shield. Díky němu by se mělo dát zcela vynechat řídicí PC, což by umožnilo vytvářet ještě jednodušší a méně prostorově i finančně náročné experimenty. Kromě toho autoři chtějí sestavit úlohy využívající softwarové stavebnice ISES Web Control a Raspberry Pi. (Panzenberger, 2021)



Obrázek 17 – Zařízení pro jeden z experimentu na Remote-LABu GymKT
zdroj: <http://remote-lab.fyzika.net/>

5 UML

Jako programovací paradigma pro modelování vzdálené laboratoře bylo zvolené objektově orientované programování (angl. Object-oriented programming, OOP), které je v současné době jedno z nejvíce populárních paradigmat pro vytváření komplexních informačních systémů (Felgo, 2022).

OOP je založen na konceptu, že by systémy měly být sestaveny z kolekce opakovaně použitelných komponent nazývaných objekty. Každá z nich obsahuje potřebná data, jako jsou proměnné (atributy) a metody. Tento přístup má několik výhod: pomáhá zajistit opětovnou použitelnost kódu díky principu dědičnosti⁹, zajišťuje bezpečnost dat díky principu zapouzdření¹⁰ a abstrakci¹¹, a poskytuje flexibilitu a škálovatelnost díky principu polymorfizmu¹². Objektově orientované systémy se modelují pomocí unifikačního modelovacího jazyka – UML (angl. Unified Modeling Language).

UML je grafický popisný jazyk pro objektové modelování v oblasti vývoje softwaru, přijatý jako mezinárodní standard, schválený OMG¹³. K vytvoření modelu systému používá grafickou notaci. Tento jazyk byl vytvořen pro definování, vizualizaci, návrh a dokumentaci softwarových systémů a používá se také k modelování obchodních procesů a návrhu systému. Je nutno podotknout, že UML nemá striktní pravidla k tomu, který jazyk se má použít při modelování. Nicméně v UML diagramech v praktické části diplomové práce byla použita angličtina,

⁹ **Dědičnost** odkazuje na schopnost jedné třídy (podřízené třídy, potomku) zdědit identickou funkcionalitu jiné třídy (supertřídy, předka) a poté přidat novou vlastní funkci.

¹⁰ **Zapouzdření** představuje mechanismus schování vnitřní struktury objektů před okolím.

¹¹ **Abstrakce** - je koncept objektově orientovaného programování, které „ukazuje“ pouze podstatné atributy a „skrývá“ nedůležité informace (Engineering LibreText, 2020). Hlavním účelem abstrakce je skrytí nepotřebných detailů před uživateli.

¹² **Polymorfizmus (mnohotvárnost)** – „*princip, podle kterého jsou operace (metody), vyskytující u více objektů (tj. mají u různých objektů stejnou signaturu), ale mají odlišnou implementaci*“ (Franěk, 2014). Hlavním rozdílem mezi dědičností a polymorfizmem je, že dědičnost se aplikuje na třídy, zatímco polymorfizmus se aplikuje na operace.

¹³ **OMG (Object Management Group)** je mezinárodní standardizační organizace. Od roku 1989 vyvíjí a spravuje standardy v různých oborech a oblastech, jako je vývoj softwaru, systémové inženýrství, finance, úřady, zdravotnictví, robotika a mnoho dalších. Mezi nejznámější standardy patří CORBA, UML, OMG SysML a BPMN (Weillkiens, 2016).

vzhledem k tomu, že je angličtina obecně považována za nejlepší praxi při modelování a programování.

Dále se v této podkapitole uvádí stručný přehled historie vzniku tohoto jazyka a základních UML diagramů. Více detailní informace o technikách objektově orientovaného modelování lze nalézt v knize Jima Arlowa a Ila Neustadta „UML 2 a unifikovaný proces vývoje aplikací“ (2008), která je základním zdrojem pro tuto a další kapitoly, tykající se návrhu modelu vzdálené laboratoře.

5.1 Vznik UML

Před vznikem UML existovalo více než 50 jazyků a metodologií vizuálního modelování, z nichž každý měl své výhody a nevýhody. Nejpopulárnějšími byly Boochova metoda od Gradyho Boоче a technika objektového modelování (Object Modeling Technique, OMT) od Jamese Rumbaugh. Každou z těchto metod lze považovat za zcela holistickou a kompletní, i když každá z nich má nejen silné, ale i slabé stránky.

Oficiálně práce nad UML začala v říjnu 1994, kdy Rumbaugh přešel do Rational Software, kde pracoval Booch. Původním cílem bylo spojit Boochovou metodu a OMT. První zkušební verze 0.8 Unifikační metody (Unified Method) – tak se tehdy jmenoval UML, se objevila v říjnu 1995. Přibližně ve stejné době Jacobson přešel do společnosti Rational a projekt UML byl rozšířen o jazyk OOSE. Výsledkem společného úsilí byla v červnu 1996 vypuštěna UML verze 0.9. Během celého roku autoři sbírali zpětnou vazbu od velkých softwarových inženýrských společností. Během této doby se ukázalo, že většina těchto společností považuje UML za jazyk strategického významu pro jejich podnikání. V důsledku toho bylo založeno konsorcium UML, které zahrnovalo organizace ochotné přispět prostředky k práci na úplné definici UML.

Formální specifikace pro nejnovější verzi UML 2.0 byla zveřejněna v srpnu 2005. Do UML 2.0 bylo přidáno mnoho nových výkonných konceptů. Sémantika jazyka byla výrazně zpřesněna a rozšířena tak, aby podporovala metodologie Model Driven Development (MDD). Pracovalo se také na zlepšení vnitřního strukturování.

Nejnovější verze UML 2.4.1 byla zveřejněna v srpnu 2011 a byla přijata jako mezinárodní standard ISO/IEC 19505-1, 19505-2.

5.2 Základní UML diagramy

Mezi tři základní, v praxi nejčastěji používané UML diagramy pro vývoj softwaru patří diagram případů užití, sekvenční diagramy a diagramy tříd.

5.2.1 Diagram případů užití - Use case diagram

Podle specifikace UML diagram případů užití ukazuje vztahy mezi aktéry a případy užití v rámci systému. UC diagramy se často používají ke sdělení rozsahu vývojového projektu a modelování analýzy požadavků na systém ve formě modelu případu užití systému (Cockburn, 2000). Kromě toho UC diagramy slouží k ověření architektury systému, řízení implementace a tvorbě testovacích případů (scénářů).

Dle autorů knihy „UML 2 a unifikovaný proces vývoje aplikací“ modelování případů užití obvykle probíhá následovně:

1. Stanovují se hranice potenciálního systému.
2. Zjišťují se aktéři.
3. Zjišťují se případy užití:
 - a. definuje se případ užití;
 - b. zavádí se alternativní toky (vazby „include“ a „extend“).
4. Předchozí kroky se opakují, dokud se případy užití, aktéři a hranice systému nestabilizují.

Model případů užití se skládá z čtyř komponent:

1. *Hranice systému.* Graficky se představuje jako obdélník, který obsahuje případy užití pro označení hranice modelovaného systému. V UML 2 se tato hranice nazývá systémový kontext (subject).
2. *Aktér.* Soubor logicky souvisejících rolí vykonávaných při interakci s případy užití nebo entitami (systém, subsystém nebo třída). Aktérem může být osoba, role osoby v systému nebo jiný systém, subsystém nebo třída, která představuje něco mimo entitu.

3. *Případ užití*. Popis konkrétního aspektu chování systému z pohledu uživatele. Případ užití neukazuje „jak“ je dosaženo určitého výsledku, ale pouze „co“ to je.
4. *Vztahy*. Smysluplné vztahy mezi aktéry a případy užití.

Model případu užití je hlavním zdrojem objektů a tříd. To je hlavní vstup pro diagram tříd.

5.2.2 Diagram tříd — Class diagram

Třída představuje kategorii věcí, které mají společné atributy a operace. UML diagramy tříd ukazují třídy systému, operace a atributy tříd a vzájemné vztahy mezi třídami systému. Diagramy tříd se obvykle používají, i když ne ke všemu najednou, ke:

- zkoumání konceptu domén ve formě modelu domény;
- analýze požadavků ve formě analytického modelu;
- znázornění detailního návrhu objektově orientovaného softwaru.

Model třídy se skládá z jednoho nebo více diagramů tříd a podpůrných specifikací, které popisují prvky modelu včetně tříd, vztahů mezi třídami a rozhraními.

5.2.3 Sekvenční diagram — Sequence Diagram

Sekvenční diagram odráží dynamickou interakci objektů v čase. V tomto případě mají informace formu zpráv a interakce objektů znamená výměnu těchto zpráv v rámci scénáře. Logika scénáře může být součástí případu užití a může obsahovat alternativní toky. Používá se k upřesnění diagramů případů užití – popisuje aspekty chování systému.

6 Systémový přístup k návrhu vzdálené laboratoře v oblasti vzdělávání

Systémový přístup je samostatným směrem metodologie vědeckého poznání, je založen na reprezentaci jakéhokoli objektu jako systému, který lze zase rozdělit na podrobnější třídy prvků, prvky samotné a vztahy mezi nimi.

Existuje více definic pojmu „systém“. Obecně se systémem nazývá *„soubor prvků, které jsou ve vzájemných vztazích a spojeních, které tvoří určitou celistvost. Komplexní systém je systém skládající se z mnoha vzájemně se ovlivňujících komponent (subsystémů), v důsledku toho komplexní systém získává nové vlastnosti“* (Peterson, Davie, 2007). Pokud lze v systému rozlišit řadu jeho subsystémů, lze jej nazvat komplexním. V souladu s tímto přístupem lze téměř jakýkoli technický systém považovat za složitý. Systémový přístup je často používán technickými odborníky k vytváření složitých počítačových systémů, jako jsou operační systémy, sítě, aplikace. Jeho důležitou vlastností je dívat se na daný problém z různých úhlů pohledu, což se uplatňovalo během psaní této diplomové práce. Při návrhu vzdálené laboratoře je potřeba nejen vytvořit model systému a popsat jeho strukturu, ale i jeho chování (procesy) a jaký způsobem se budou ukládat data.

6.1 Fáze vývoje informačního systému dle metodiky RUP

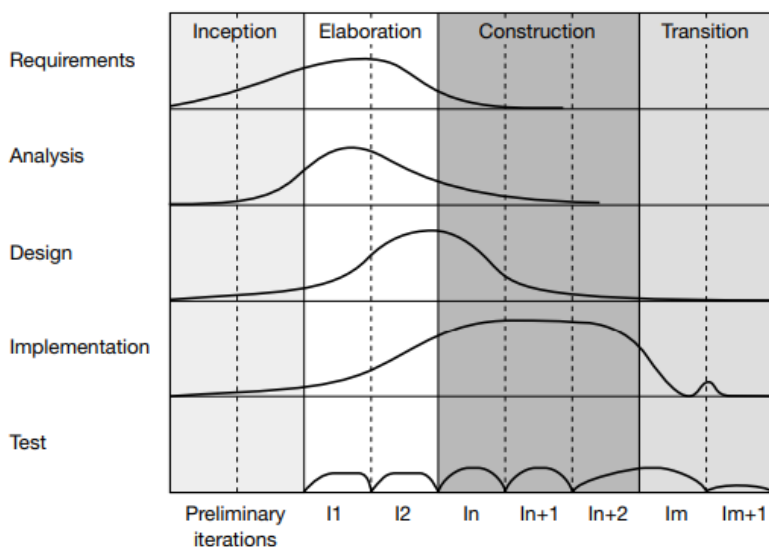
Proces tvorby vzdálené laboratoře jako komplexního systému je rozdělen do několika etap, omezených určitými časovými horizonty a končících vydáním konkrétního produktu (modelů, softwarových produktů, dokumentace). Názvy jednotlivých fází se liší podle toho, jaká metodika vývoje je použita, ale vždy by měly zahrnovat:

- tvorbu požadavků na systém;
- analýzu;
- návrh;
- implementaci;
- testování;
- zavedení do provozu.

Pro návrh vzdálené laboratoře byla zvolena metodika Rational Unified Process (RUP). Metodika používá OOP, je iterativní a agilní. (Bugajenko, 2021) Iterativní, protože všechny hlavní aktivity procesu se v průběhu projektu opakují. Agilní, protože komponenty, ze kterých se skládá proces, lze různě upravit a fáze cyklu se mohou opakovat, dokud software nesplní požadavky a cíle. Proces je znázorněn na obrázku 18 a obsahuje procesovou a časovou dimenzi.

Procesová dimenze je reprezentovaná svislou osou a představuje statický aspekt procesu. Je popsána z hlediska činností, artefaktů, pracovníků a pracovního postupu.

Časová dimenze je reprezentovaná vodorovnou osou a je vyjádřena pomocí fází. Jsou 4 fáze: zahájení, upřesnění, konstrukce a předání. Fáze zahájení obsahuje business zdůvodnění a rozsah projektu, požadavky na systém. Ve fázi upřesnění se definuje podrobnější popis požadavků, provádí se analýza a návrh pro vytvoření základní architektury systému. Fáze konstrukce se skládá z mnoha iterací. Každá iterace je mini projektem: analýza, návrh, implementace, testování jsou přiřazené ke každé iteraci. Výsledkem fáze konstrukce je plně dokončený softwarový systém. Fáze předání může zahrnovat testování beta verze, ladění výkonu a školení uživatelů.



Obrázek 18 – Proces vývoje informačního systému dle metodiky RUP
zdroj: Arlow, Neustadt, 2008

6.2 Definování požadavků na systém

Na začátku životního cyklu (fáze Zahájení) jakéhokoli projektu je nutné určit požadavky na systém. Definování požadavků je termín používaný k popisu činnosti, zahrnující identifikace, dokumentace a udržování sady požadavků na informační systém. Jde o zjištění toho, co zúčastněné strany od systému očekávají. Podle autorů článku „The Role of Requirements in the Success or Failure of Software Projects“ (Hussain, Mkpojiogu, Kamal, 2016) jsou neúplné požadavky a nedostatečné zapojení uživatelů dvěma hlavními důvody neúspěchu projektů. Oba problémy jsou výsledkem chyb ve vývoji požadavků. Vzhledem k tomu, že konečný softwarový systém je založen na sadě požadavků, efektivní generování požadavků je kritickým faktorem úspěchu v projektech vývoje softwaru.

Požadavky mohou být funkční a nefunkční. Funkční požadavky jsou popisem toho, co bude systém dělat, a jsou zohledněny v případech užití. Nefunkční požadavky jsou popisy omezení kladených na systém (například výkon, spolehlivost) a nelze je reprezentovat jako případy užití.

Jak již bylo několikrát zmíněno, navrhovaný systém bude sloužit pro vzdálený přístup k laboratořím a umožní studentům řízení experimentů na dálku. Systém by měl být dostupný odkudkoliv, proto celková myšlenka bude fungovat na bázi IoT¹⁴: budou využité robotické manipulátory, aktuátory a senzory. Systém bude oddělen od celkové školní infrastruktury. To pak například ovlivní proces správy uživatelů: databáze uživatelů nebude exportovaná z dalšího systému, ale bude vytvořená vlastní. Systém bude navržen obecně, aby ho bylo možné využít pro libovolnou školní laboratoř.

¹⁴ **Internet of Things (IoT)**, česky **internet věcí** – „je systém vzájemně propojených chytrých zařízení, která využívají vestavěné systémy, jako jsou procesory, senzory a komunikační hardware, ke shromažďování, odesílání a jednání s daty, která získávají ze svých prostředí. Někdy tato zařízení komunikují s dalšími souvisejícími zařízeními a jednají na základě informací, které od sebe získají. Zařízení vykonávají většinu práce bez lidského zásahu, ačkoli lidé mohou s nimi komunikovat – například je nastavovat, dávat jim pokyny nebo přistupovat k datům“ (Gillis, 2022).

Funkční požadavky lze rozdělit na business pravidla (angl. business rules) a funkce (angl. features). Mezi *business pravidla* pro navrhovanou vzdálenou laboratoř patří:

- Pouze registrovaný uživatel může provádět experimenty a pouze uživatel s roli „Vyučující“ nebo „Administrátor“ může přidávat a spravovat experimenty.
- Experiment má buď veřejný (jakýkoliv registrovaný uživatel má přístup) nebo soukromý přístup (pouze vybraní uživatelé mohou mít k němu přístup). Přístup definuje Vyučující při tvorbě experimentu.
- Formuláře pro tvorbu a editaci experimentu jsou validované systémem. Musí být vyplněna povinná pole, jako jsou název, popis a postup experimentu.
- Je možné si zvolit pouze dostupné termíny provádění experimentu, které jsou závislé na dostupnosti laboratoře.

Vzdálená laboratoř bude poskytovat následující *funkce*:

- systém vzdálené laboratoře musí provádět autentifikaci a autorizaci uživatele;
- systém vzdálené laboratoře musí umožňovat provádět experimenty;
- systém vzdálené laboratoře musí umožňovat přidávat nové experimenty;
- systém vzdálené laboratoře musí ukládat a vracet výsledky experimentů;
- systém vzdálené laboratoře musí umožňovat spravovat a rezervovat termíny experimentů;
- systém vzdálené laboratoře musí umožňovat spravovat zdroje a plánovat jejich dostupnost;
- systém vzdálené laboratoře musí umožňovat spravovat uživatele systému;
- systém vzdálené laboratoře musí poskytovat přehled o stavu laboratoře.

Pokud jde o *nefunkční požadavky*, pak první otázkou bylo, jak se bude *přístupovat* k systému a bylo rozhodnuto, že navržený systém bude na bázi webové

aplikace s přístupem přes internetový prohlížeč. Aplikace bude splňovat požadavky responsivenessy a proto se bude moci využívat i přes mobilní zařízení.

Bezpečnostní aspekty jsou hodně důležité při implementaci vzdálených laboratoří. Rizika mohou vznikat v důsledku špatného používání laboratoře, vniknutí malwaru, neautorizovaných přístupů, předstírání identity uživatelů na dálku a mnoha dalších hrozeb. Je proto nutné zajistit vysokou ochranu systému. Podrobněji je tomu věnován prostor v podkapitole 6.6 dané práce.

Dalším požadavkem je, aby *uživatelské rozhraní* zobrazovalo fyzické aspekty experimentu a neskrývalo je za příliš abstraktní, nepřírodně vypadající grafy a widgety GUI. Celý proces má být natočen a sdílen, aby uživatele vždy měli pocit, že jejich interakce je opravdová, nikoliv jen prostá simulace.

Klíčovým faktorem je také dobrá *dostupnost* (angl. accessibility), která je ovlivněná kvalitou internetového připojení a předpoklady klientského softwaru/hardware. Předpokládá se, že všechny experimenty musí mít video a audio monitorování v reálném čase, proto se vyžaduje vysokorychlostní (širokopásmové) připojení k internetu. Aby nebyla vyžadována speciální instalace klientského softwaru do systému uživatele, je nutné použití standardizovaných technologií nezávislých na platformě, jako je Javascript, AJAX, Flash (Go up, 2022).

Pro dobrou *udržitelnost* (angl. maintainability) systému je potřeba použít standardizované, dobře podporované softwarové technologie a standardní kvalitní hardware, které pomohou zajistit maximální provozní životnost laboratoře.

Infrastruktura vzdálené laboratoře by měla být navržena tak, aby byla *spolehlivá*, odolná vůči poruchám. Je potřeba zabezpečit, aby porucha nebo nesprávná funkce jedné součásti nezabránila použití celé laboratoře. K tomu se vyžaduje víceserverové prostředí (William, 2019). Použití víceserverového systému obecně pomáhá zaručit vysoký výkon a dobu provozuschopnosti, udržovat zabezpečení a umožňuje efektivnější alokaci zdrojů.

6.3 Analýza

Většina analytické práce začíná na konci fáze „Zahájení“ a hodně se překrývá s tokem definice požadavků. Účelem analytického pracovního postupu (z hlediska OO analýzy) je tvorba analytického modelu. Tento model se zaměřuje na popis toho, co by měl systém dělat. Podrobnosti o tom, jak to systém provede, jsou přenechávané návrhu (Arlow, Neustadt, 2008).

Výstupem analytické práce jsou dva klíčové artefakty:

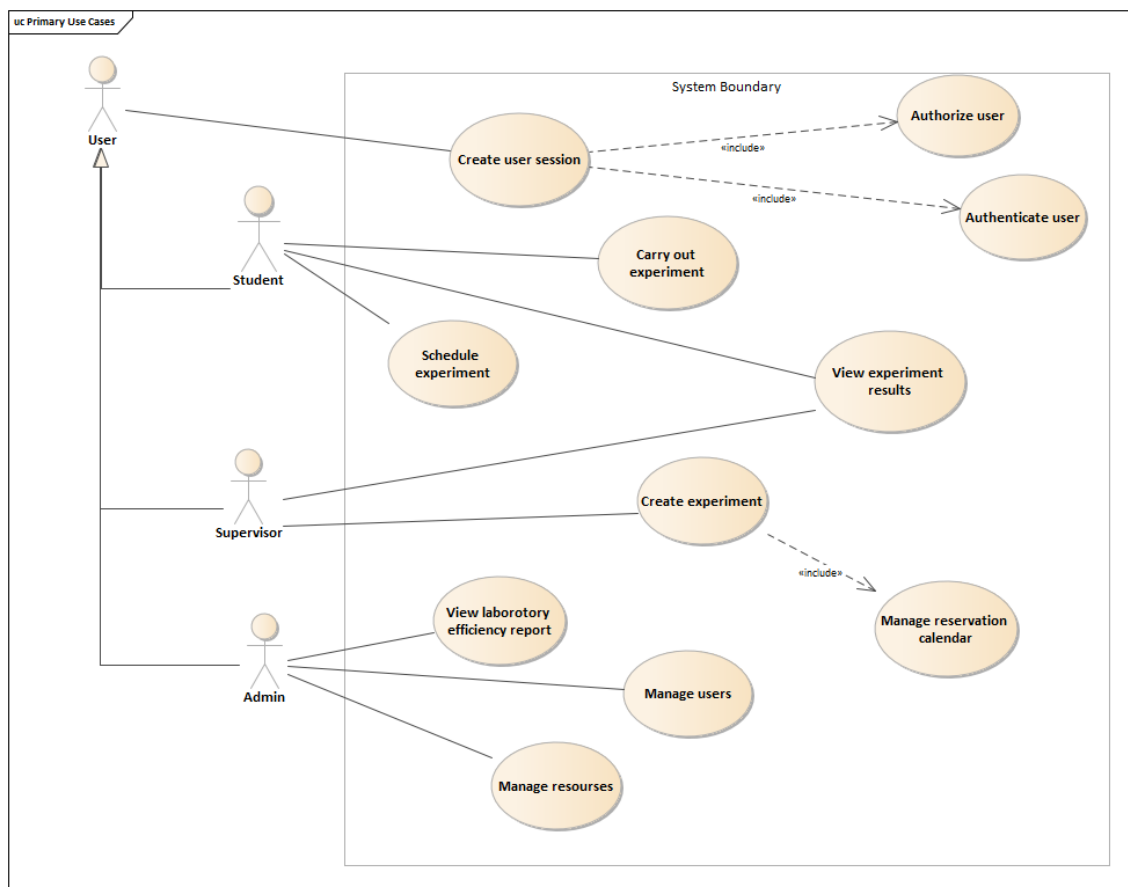
- diagram případů užití systému – zachycuje funkčnost systému tak, jak je vidí uživatelé
- analytický model tříd – klíčové pojmy, vyskytující se v případech užití.

6.3.1 Diagram případů užití

Na obrázku 19 je znázorněn diagram případů užití (UC) pro navrhovanou vzdálenou laboratoř. Aktéry systému jsou **Student**, **Supervisor** a **Admin**, kteří jsou umístěni mimo hranice systému. Před použitím laboratoře se každý aktér musí přihlásit a toto společné chování je vyčleněno do rodičovského aktéra **User**. K tomu se používá vazba generalizace. Společným případem užití pro všechny uživatele je „*Create user session*“ (Vytvoření uživatelské relace), které zahrnuje Autentifikaci („*Authenticate user*“) a Autorizaci uživatele („*Authorize user*“). Autorizace odkazuje na ověření, kdo se přihlašuje. Pro ověření se používá uživatelské jméno a heslo. Autentifikace odkazuje na to, co uživatel může dělat, jaké má práva. K případům užití „*Authenticate user*“ a „*Authorize user*“ je použita relace „include“. To znamená, že tento případ užití je začleněn do základního případu užití „*Create user session*“ a na základním UC se může pokračovat až po tom, co bude dokončen začleněný UC.

Aktér **Admin** spravuje uživatele systému („*Manage users*“) a zdroje laboratoře („*Manage resources*“). Správa uživatelů systému obsahuje přidání nových uživatelů, úpravu (například změna některých osobních údajů a práv), deaktivaci uživatelů. Správa zdrojů zahrnuje instalaci vybavení, aktualizaci softwaru a opravu poruch. Pro vyhodnocení efektivity laboratoře aktér **Admin**

může žádat o výkazy o provozu laboratoře (například kdo, kdy a jak používal laboratoř, jestli nastávaly poruchy systému).



Obrázek 19 – Diagram případů užití vzdálené laboratoře

Aktér **Supervisor** zadává do systému nové experimenty („*Create experiment*“). Během toho musí vybrat laboratorní zařízení, které bude potřeba využít během experimentu, a také uvádí specifikaci experimentu, jako je název, popis a postup provedení. Na základě vybraného zařízení systém ověří jejich dostupnost a nabízí aktérovi volné termíny. Aktér vybere řadu termínů, ze kterých si pak aktér **Student** může vybrat při plánování svého sezení. Jedná se o případ užití „*Manage reservation calendar*“, který je součástí případu užití „*Create experiment*“, ale zároveň se může provádět samostatně, pokud aktér Supervisor bude chtít změnit termíny rezervace.

Kromě toho spolu s aktérem **Student** může **Supervisor** prohlížet výsledky provedeného experimentu („*View experiment results*“). Tento případ užití mimo jiné

slouží pro porovnání výsledku experimentů, například v situaci, když se stejný experiment se prováděl několikrát kvůli obnově zařízení.

Jedním z požadavků vzdálené laboratoře je, aby se student mohl sám naplánovat, kdy chce využít laboratoř a provést experiment. Proto v rámci případu užití „*Schedule experiment*“ aktér **Student** může vybrat den a čas experimentu, který je předdefinován v kalendáři Vyučujícím. Ve vybraný termín se **Student** přihlásí do systému a provede experiment („*Carry out experiment*“).

Ke každému případu užití se uvádí jeho specifikace, včetně scénáře, který představuje posloupnost kroků spolupráce mezi aktérem a systémem. Specifikace vybraných případů užití jsou uvedeny v tabulkách 1 až 3, zbylé jsou přiloženy v příloze 1.

Tabulka 1 – Specifikace k případu užití „Schedule experiment“

Popis	Student zvolí datum a čas experimentu, který chce provést
Aktéři	Student
Podmínka pro spuštění	Student musí být přihlášen do systému
Základní tok	<ol style="list-style-type: none"> 1. Student klikne na „Naplánovat experiment“ 2. Systém načte dostupná data a čas 3. Student zvolí datum a čas experimentu a potvrdí volbu 4. Systém ověří dostupnost zdrojů 5. Systém rezervuje zdroje na zvolený termín 6. Systém zobrazí hlášku o uložení naplánovaného termínu
Alternativní tok	<p>5a Vybraný termín již není dostupný</p> <p>5a 1. Systém zobrazí hlášku, že sezení nebylo uloženo kvůli nedostupným zdrojům a vyzve aktéra zvolit jiný termín</p> <p>5a 2. Student klikne na možnost „Zvolit jiný termín“</p> <p>5a 3. Systém pokračuje od kroku 2 Základního scénáře</p>
Podmínky dokončení	Experiment je naplánován na termín vybraný uživatelem.

Tabulka 2 – Specifikace k případu užití „Carry out experiment“

Popis	Ve zvolený termín student provede experiment
Aktéři	Student
Podmínka pro spuštění	Student musí být přihlášen do systému a musí mít rezervovaný termín experimentu
Základní tok	<ol style="list-style-type: none"> 1. Student iniciuje UC „Provést experiment“ 2. Systém načte dostupné experimenty 3. Student zvolí experiment a klikne na „Provést experiment“ 4. Systém alokuje zdroje a aktivuje zařízení 5. Student provádí interakce s laboratorním zařízením 6. Systém sbírá data a ukládá je do databáze 7. Student dokončuje experiment 8. Systém deaktivuje zařízení 9. Systém ukládá a načítá studentovi zprávu s výsledky experimentu
Podmínky dokončení	Uživatel dokončí experiment a systém deaktivuje zařízení

Tabulka 3 – Specifikace k případu užití „Manage users“

Popis	Administrátor laboratoře spravuje databázi s uživateli (přidává nové uživatele, upravuje jejich údaje a deaktivuje přístup uživatele k systému)
Aktéři	Administrátor (Admin)
Podmínka pro spuštění	Administrátor musí být přihlášen do systému a dostal požadavek o správě uživatelů od pověřené osoby
Základní tok	<ol style="list-style-type: none"> 1. Administrátor iniciuje UC „Správa uživatelů“ 2. Systém načte seznam dostupných možností správy uživatelů 3. Administrátor zvolí některou z možností 4. Systém uloží změny a zobrazí uživateli hlášku o úspěšném uložení
Alternativní tok	<p>3a Pokud Administrátor zvolí možnost přidat nového uživatele</p> <p>3a 1. Systém načte formulář</p> <p>3a 2. Administrátor zadá informace o uživateli, včetně jeho jména, emailové adresy (která se použije při</p>

	<p>přihlašování), nastaví uživateli roli (role) a klikne na „Uložit“.</p> <p>3a 3. Systém ověří správnost formuláře</p> <p>3a 4. Systém odešle novému uživateli na e-mail odkaz pro přihlášení do systému</p> <p>3a 5. Systém pokračuje od kroku 3 základního toku UC</p> <p>3b Pokud Administrátor zvolí možnost úpravy údajů uživatele</p> <p>3b 1. Systém načte seznam uživatelů</p> <p>3b 2. Administrátor zvolí uživatele a klikne u něj na „Upravit“</p> <p>3b 3. Systém načte detail vybraného uživatele</p> <p>3b 4. Administrátor provede změny a klikne na „Uložit“</p> <p>3b 5. Systém pokračuje od kroku 3 základního toku UC</p> <p>3c Pokud Administrátor zvolí deaktivovat uživatele</p> <p>3c 1. Systém načte seznam uživatelů</p> <p>3c 2. Administrátor zvolí uživatele a klikne na „Deaktivovat“</p> <p>3c 3. Systém vyzve aktéra potvrdit volbu</p> <p>3c 4. Administrátor potvrdí volbu</p> <p>3c 5. Systém pokračuje od kroku 3 základního toku UC</p>
Podmínky dokončení	Systém uloží provedené změny

Základní tok scénáře obsahuje ideální variantu průchodu případem užití. Pro lepší popis chodu se navíc používají alternativní toky, aby popsaly situaci, když nastala nějaká výjimka, chyba apod.

6.3.2 Analytický model tříd

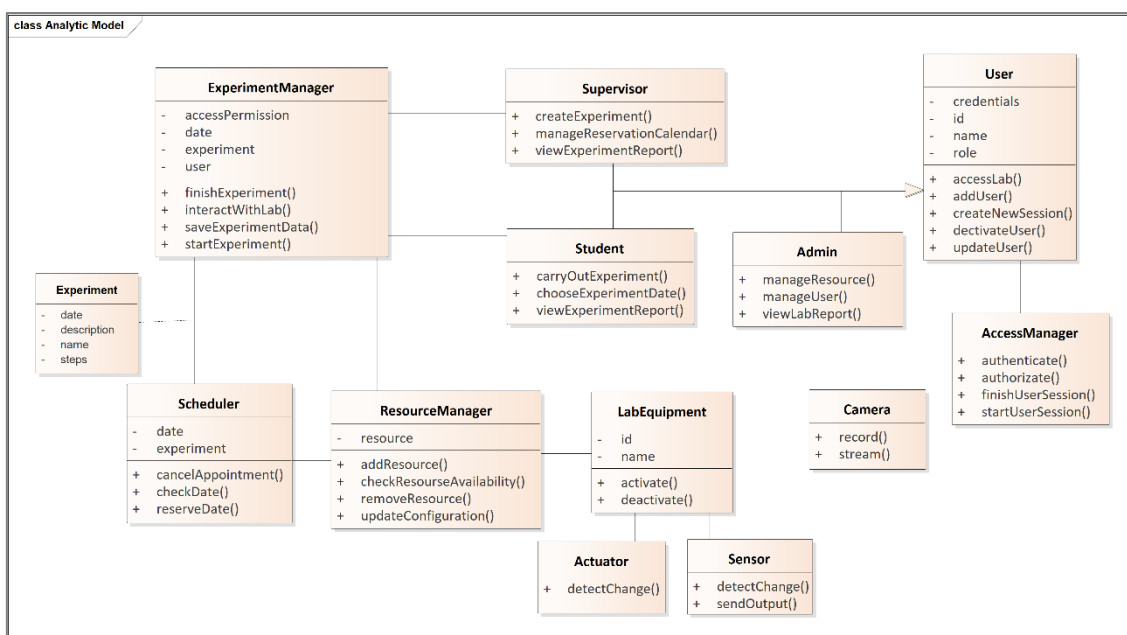
Třídy v analytickém modelu obsahují pouze klíčové atributy a operace, model se vyhýbá detailům implementace, které budou doplněné v návrhovém modelu.

Základní koncept struktury systému vzdálené laboratoře je znázorněn na obrázku 20. Jedním z prvků modelu je třída (neboli entita) *User*, sloužící pro evidenci uživatelů systému. Uživatelé se rozlišují podle rolí, proto je *User* propojen s třídami *Supervisor*, *Student* a *Admin* vztahem generalizace, jelikož třídy mají společné i rozdílné vlastnosti. Za ověření identity a autorizačních práv uživatelů zodpovídá třída *AccessManager*, proto je propojena s třídou *User*. V modelu mají entity *Supervisor* a *Student* přímou relaci na entitu *ExperimentManager*, přes

kteřou plánují (vytvářejí) a řídí experimenty. V reálném světě je mezi nimi grafické rozhraní. Model je ale zjednodušenou realitou a proto GUI není součástí analytického (ani návrhového) diagramu tříd.

Třída *ExperimentManager* obsahuje specifikace a výsledky experimentů a komunikuje s třídami *Scheduler* a *ResourceManager* pro plánování experimentu a alokování a řízení zdrojů. Třída *Experiment* spojuje třídy *Scheduler* a *ExperimentManager* a je asociační třídou: je vztahem, který má vlastní atributy. Třída *Scheduler* poskytuje služby související se správou termínů experimentů a je propojena s třídou *ResourceManager* pro ověřování dostupnosti konkrétního zdroje.

ResourceManager spravuje zdroje v databázi zdrojů udržováním informací o dostupnosti zdrojů v systému v reálném čase. Třída implementuje vyhledávací službu nad indexovaným seznamem zdrojů pomocí metadat spojených s každým zdrojem.



Obrázek 20 – Analytický model tříd systému vzdálené laboratoře

Analytická třída *LabEquipment* představuje zařízení fyzické laboratoře. Zařízení má relaci na třídy *Camera*, *Acuator* a *Sensor*. Reálná laboratoř může obsahovat i další zařízení (například manipulátory, kontroléry pro naprogramování

zdrojů, napájecí systém), které nejsou přítomné na daném diagramu pro jeho lepší čitelnost.

6.4 Návrh

Hlavním cílem analýzy bylo sestavit logický model systému, odrážející funkcionalitu, kterou musí systém poskytovat, aby vyhovoval požadavkům uživatelů. Účelem návrhu je určit, jak bude tato funkcionalita implementována (Arlow, Neustadt, 2008).

Výsledkem návrhu jsou dva artefakty:

- třídy návrhu – to jsou třídy, jejichž popisy jsou tak úplné, že je lze implementovat;
- sekvenční diagramy – interakce objektů a tříd návrhu, které implementují případ užití.

6.4.1 Návrhový model tříd

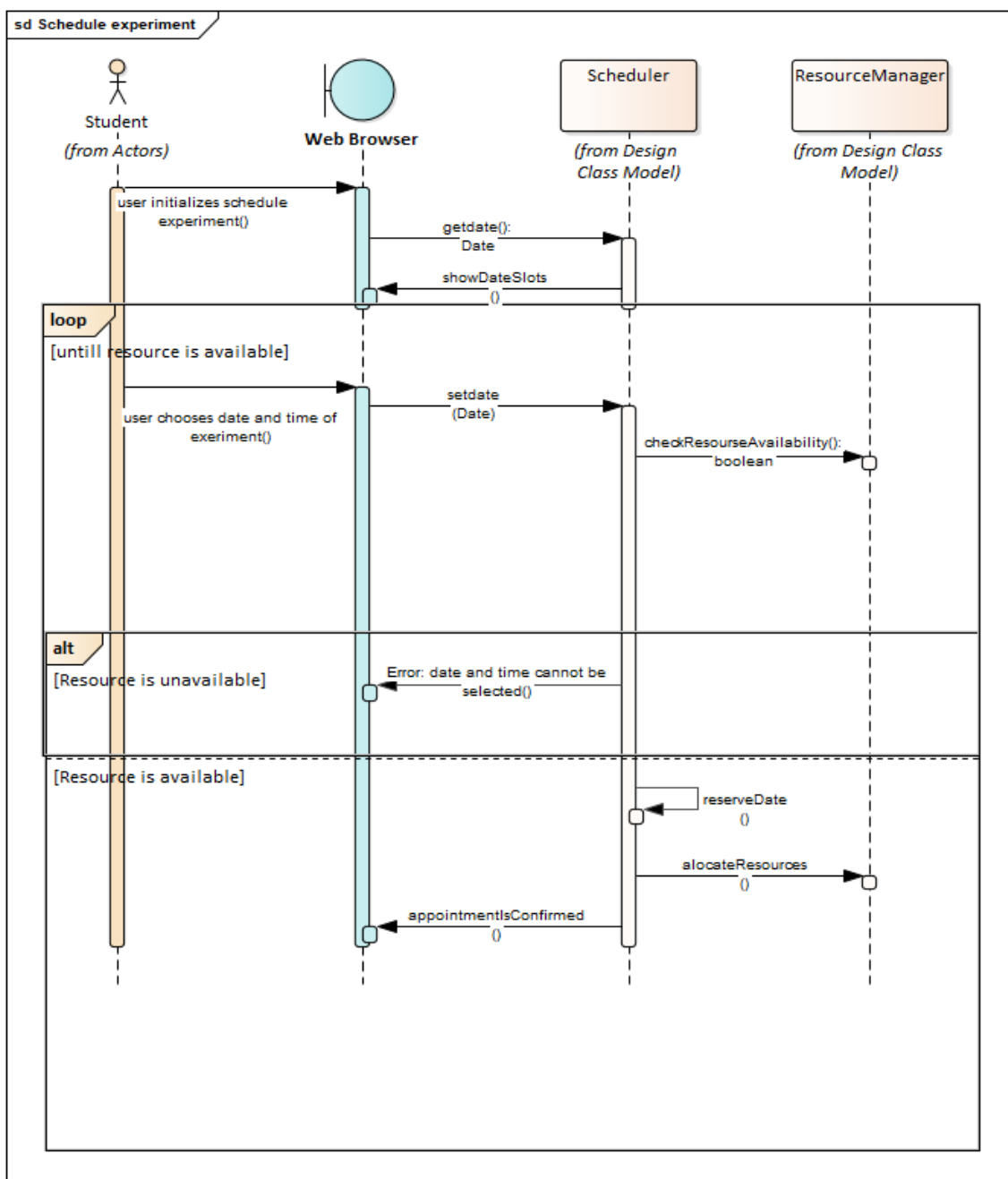
Rozšířením analytického modelu byl získán návrhový model systému vzdálené laboratoře, znázorněný na obrázku 21.

Třídy návrhu musí přesně definovat, jak bude každá třída plnit své „povinnosti“. V návrhovém modelu byl doplněn datový typ atributů a operací, tak zvané gettery a settery pro získání a nastavení atributů třídy. Kromě toho byly doplněné i některé chybějící operace. Pro definování vlastních datových typů byly přidány tak zvané pomocné třídy (angl. utility class): *Credentials* a *Resource*.

Pro znázornění relace mezi třídami v analytickém modelu byl použit obecný vztah asociace. V návrhovém modelu bylo nutné doplnit vztahy o multiplikaci a v případě potřeby transformovat vztah asociace na agregaci a kompozici. Například vztah asociace u tříd *Camera*, *Sensor* a *Actuator* k laboratornímu zařízení (*LabEquipment*) byl změněn na agregaci, sice patří fyzické laboratoře, ale mohly by se využít samostatně nebo v kombinaci s jiným typem zařízení. Asociační třída z analytického modelu *Experiment* byla transformována na obyčejnou třídu, je prostředníkem mezi *ExperimentManager* a *Scheduler* a je součástí obou. Navíc se

návrhu většiny projektů a není nutné, aby měl každý případ užití sekvenční diagram, pokud jsou ke každému případu užití sepsány již zmíněné scénáře. V případě, že by tyto scénáře nebyly dostatečně jednoznačné a přehledné, mohou se později při implementaci doplnit o sekvenční diagramy.

6.4.2.1 Plánování experimentu



Obrázek 22 – Sekvenční diagram k případu užití „Schedule experiment“

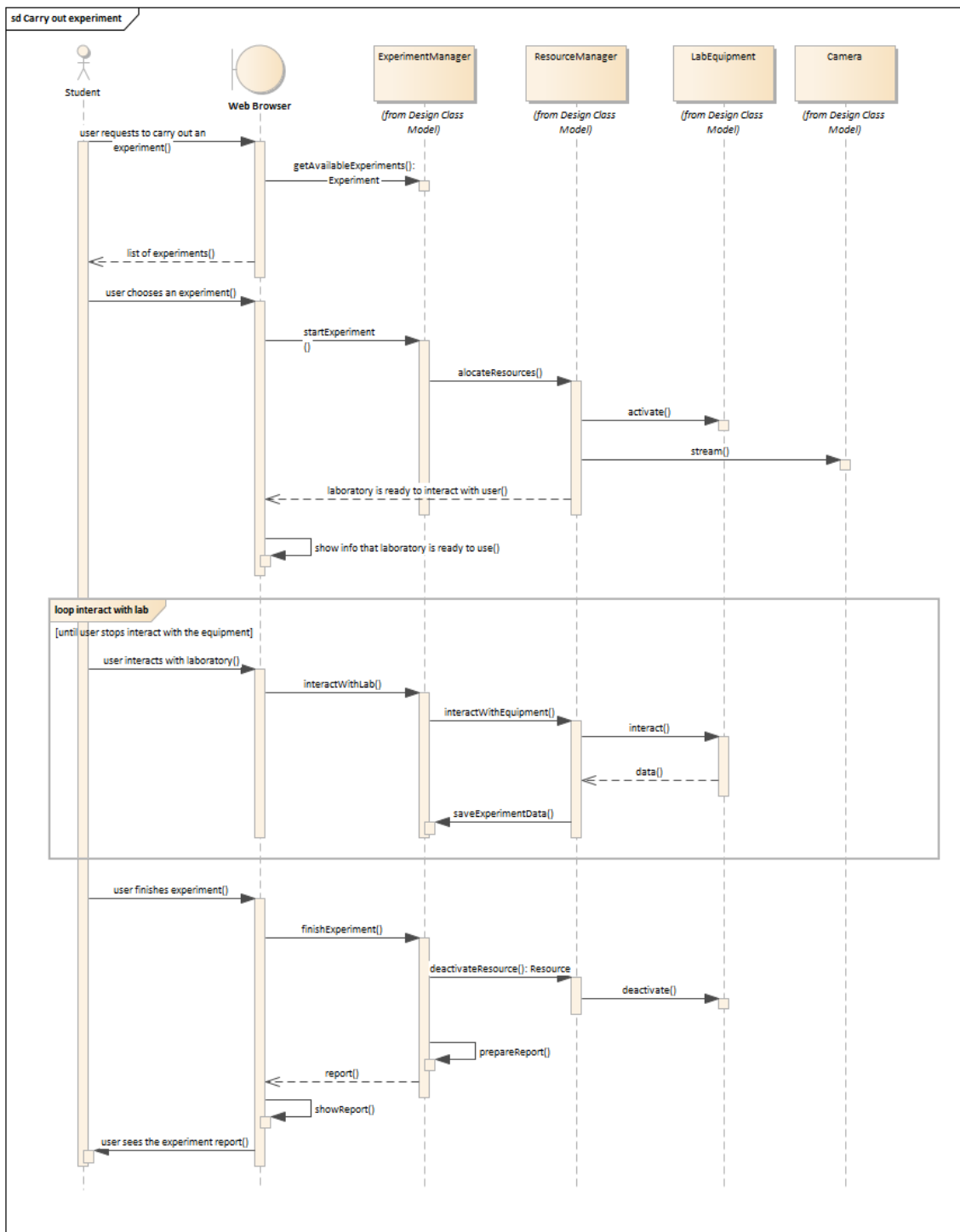
Případ užití „*Schedule experiment*“ je spuštěn aktérem **Student**, který ve webovém prohlížeči klikne na „Naplánovat experiment“. Webový prohlížeč odesílá zprávu entitě *Scheduler* pro zobrazení dostupných termínů. *Scheduler* načte termíny experimentu nastavené aktérem **Supervisor**, ze kterých si **Student** zvolí nejvíce hodící. *Scheduler* požádá *ResourceManager* o dodatečné ověření dostupnosti vyžadovaných zdrojů pro případ, že během procesu výběru nastaly změny a nějaký ze zdrojů již není dostupný. V případě, že není nějaký ze zdrojů dostupný, systém upozorní aktéra a požádá ho o zvolení jiného termínu. Celý proces s ověřením se znovu opakuje, dokud nejsou dostupné všechny vyžadované zdroje. Až se to stane, *Scheduler* zarezervuje termín a upozorní o to *ResourceManager* a aktéra.

Je potřeba poznamenat, že tento a další sekvenční diagramy zahrnuje vrstvu uživatelského rozhraní (GUI), který odděluje uživatele a systém. Tento prvek však není namodelován nijak detailně a v případě potřeby bude doplněn během implementace.

6.4.2.2 Provádění experimentu

Po úspěšném naplánování experimentu se ve vybraný termín **Student** připojí ke vzdálené laboratoři, aby provedl experiment. Následné kroky tohoto procesu znázorňuje sekvenční diagram na obrázku 23.

Případ užití začíná výběrem experimentu aktérem Student ve webovém prohlížeči. Systém požádá *ExperimentManager* o přístup k laboratoři a *ResourceManager* o alokaci zdrojů. *ResourceManager* aktivuje potřebná laboratorní zařízení a požádá kamery, umístěné v laboratorní místnosti, o video přenos. Jakmile budou zdroje připravené, uživatel obdrží notifikaci a zahájí přes uživatelské rozhraní interakci s laboratoří. Prostředníkem mezi entitou Student a *LabEquipment* jsou entity *ExperimentManager* a *ResourceManager*. Dokud Student nedokončí experiment, provádí se interakce s Aktivátory a Senzory a průběžně se ukládají výsledky. Až uživatel dokončí experiment, *ExperimentManager* upozorní *ResourceManager*, aby deaktivoval zařízení laboratoře. Následně *ExperimentManager* připraví zprávu s výsledky experimentu a zobrazí ji Studentovi.

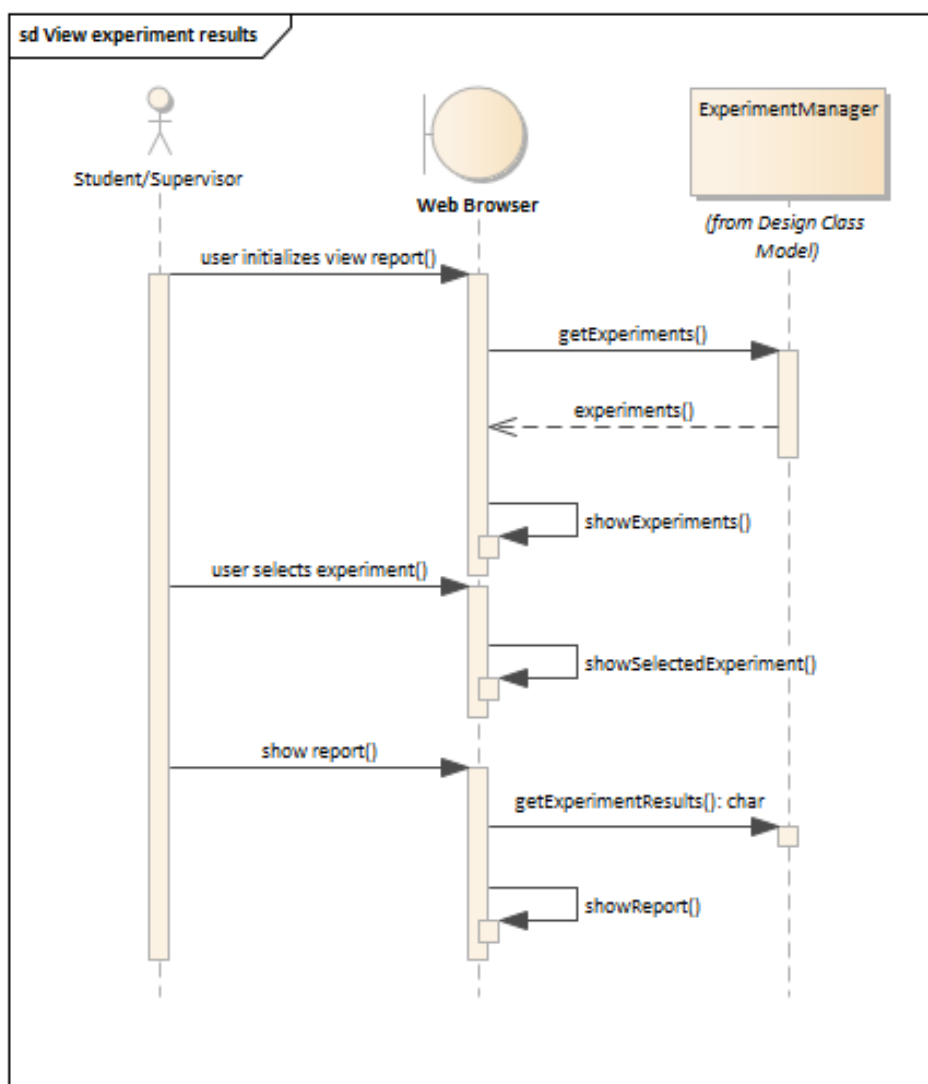


Obrázek 23 – Sekvenční diagram k případu užití „Carry out experiment“

6.4.2.3 Prohlížení zprávy o výsledcích experimentu

Tento případ užití je spuštěn buď aktérem Student, nebo aktérem Supervisor. Nejdřív ExperimentManager načte všechny experimenty přístupné danému uživateli. Uživatel si zvolí experiment, u kterého chce vidět výsledky. Správce

experimentu načte a zobrazí uživateli ve webovém prohlížeči výsledky experimentu. Výše popsaný proces je znázorněn na obrázku 24.



Obrázek 24 – Sekvenční diagram k případu užití „View experiment results“

Tento a další diagramy vytvořené v rámci praktické části diplomové práce jsou také přiložené v přílohách 2 až 7.

6.5 Ukládání dat

Důležitým aspektem tvořeného návrhu, který není zahrnutý v žádném z uvedených modelů, je to, jak se budou ukládat data. Pro správu uživatelů stačí využít tradiční *relační databáze*, která umožňuje ukládat velká množství dat. Konzistence, izolace (pro souběžný přístup) a perzistence jsou definujícími vlastnostmi relačních databázových transakcí (vkládání, aktualizace, vyhledávání). Tento typ databáze je

vhodný pro dlouhodobé ukládání dat, která je třeba v průběhu času aktualizovat (například upravit osobní nebo přihlašovací údaje uživatele, deaktivovat ho nebo upravit roli uživatele). Relační databáze by se taky mohla využít pro ukládání termínů naplánovaných experimentů a specifikace experimentů (tzn. postup, popis, seznam zařízení, které se použijí). Mezi tři nejpopulárnější relační databáze patří Oracle, MySQL a Microsoft SQL Server (DB-Engines, 2022).

Pro ukládání, vizualizaci a analýzu výsledků experimentů (údajů ze senzorů, kamer) a dat o celkovém provozu laboratoře nejsou relační databáze považovány za vhodné. Vhodnější je použít *databáze časových řád* (Time series database, TSDB). Jedná se o databáze, které jsou speciálně navrženy a optimalizovány pro práci s vysokými datovými rychlostmi a časovou značkou (angl. timestamp). Tato optimalizace se provádí několika způsoby, jednak efektivnější reprezentací časové značky, která vyžaduje menší kapacitu než běžná relační databáze, jednak díky schopnosti zapisovat miliony záznamů za sekundu do TSDB, což by většinou přetížilo databáze nečasových řád (Fadhel, Sekerinski, Yao 2019). Mezi nejčastěji používané databáze časových řád patří *InfluxDB*, *KDB+* a *Prometheus* (DB-Engines, 2022).

6.6 Bezpečnost vzdálené laboratoře

Při tvorbě jakéhokoliv informačního systému je důležitá otázka bezpečnosti tohoto systému. Jak bylo zmíněno v podkapitole 6.2, mezi požadavky na navrhovanou vzdálenou laboratoř patřilo, aby byl systém přístupný prostřednictvím sítě internet, přes internetové prohlížeče. Je známo, že vedle svých obrovských výhod je internet nejzranitelnější sítí, jaká kdy byla postavena (Lee, 2015). V daném projektu je navrhována laboratoř, která je zařízená drahým vybavením a nástroji. Obavy o bezpečnost jsou následující:

- Neoprávněný přístup na webovou stránku vzdálené laboratoře.
- Nedostatečné zabezpečení všech serverů souvisejících se vzdálenou laboratoří, jako je webový server, databázový server a víceúčelový server.
- Odposlech, jelikož se očekává, že navrhovaná webová aplikace přenáší všechna data ve formátu prostého textu.

Řešením první bezpečnostní otázky jsou výše zmíněné procesy autorizace a autentifikace, zabudované do systému. Uživatelé systému jsou přidávání administrátorem, který mimo jiné nastavuje uživateli roli, podle které se řídí přístupová práva. Zabezpečení sítě na druhé straně poskytne ochranu všech vzdálených laboratorních serverů a zařízení. Nebylo to součástí této diplomové práce, ale pro ochranu sítě se doporučují firewally (BasuMallick, 2022).

Jako metodu zabezpečení přenosu datových balíčků je nutné využít šifrování. Během minulých staletí se vyvinuly desítky šifrovacích metod, od Caesarovy šifry a Enigmy až po dnešní vysoce zabezpečené šifrovací mechanismy. Existovala možnost sepsání vlastního jednoduchého šifrovacího algoritmu a jeho integrace s navrhovaným systémem, ale protože je požadována vysoká úroveň zabezpečení systému, bylo nakonec rozhodnuto o použití hotových šifrovacích metod s výrazně vyšší bezpečností. Po provedení průzkumu bylo zjištěno, že nejbezpečnější a nejběžnější šifrovací metodou pro zabezpečení webových stránek je Secure Socket Layer (SSL). Jedná se o webové stránky, které používají protokol HTTPS (Secure Hypertext Transfer Protocol). HTTPS funguje na portu 443 a k zajištění svých připojení používá protokol TCP (Transmission Control Protocol). (GlobalSign, 2022) SSL je standardní bezpečnostní technologie pro vytvoření šifrovaného spojení mezi serverem a klientem, typicky webovým serverem (webem) a prohlížečem. SSL je založeno na sektoru standardizace telekomunikací (ITU-T) X.509 Mezinárodní telegrafní unie (ITU), což je standard pro kryptografii Public Key Infrastructure (PKI). SSL používá asymetrickou metodu šifrování, zatímco symetrická metoda je starší a snadno dešifrovatelná. V metodách šifrování soukromým (tajným) klíčem vysílač zašifruje dokument pomocí svého vlastního klíče a poté předá zašifrovaný balíček příjemci. Příjemce ale k odemknutí potřebuje stejný klíč, který šifruje data. Proto by měl vysílač odeslat také soukromý klíč. Jedná se o obrovskou bezpečnostní slabinu a existuje možnost, že by hacker mohl zachytit provoz a získat jak šifrovaná data, tak i klíč a tak data dešifrovat. Překonaným řešením této bezpečnostní slabiny je infrastruktura veřejného klíče (PKI), kterou SSL také využívá (Pilgrim, 2010).

Kromě toho může být zlepšena spolehlivost vzdálených laboratoří využitím složitějších plánovacích algoritmů, které berou v úvahu mezičas do selhání a požadavky na údržbu, jako je výměna baterie nebo materiálů.

7 Shrnutí výsledků a závěr

Důležitost laboratorního vzdělávání a praktického výcviku v mnoha vzdělávacích oblastech je nepochybná a je zdůrazněna v mnoha výzkumných článcích. Laboratorní praktické znalosti s praktickými zkušenostmi jsou nezbytné pro lepší pochopení teoretických konceptů. Klasické fyzické laboratoře však mají řádu omezení, hlavně omezený přístup k laboratoři, nedostatečnou flexibilitu a omezenou dostupnost vybavení. S nástupem pandemie COVID-19 se stalo dálkové studium povinné v mnoha školních institucích, a výuka předmětů obsahujících laboratorní práce byla obtížná, mnohdy došlo k jejímu úplnému zrušení. Proto byla v rámci této diplomové práce prozkoumány alternativy klasickým laboratořím, s cílem navrhnout vlastní systém pro učební účely, umožňující provádění experimentů na dálku. Mezi tyto alternativy patří virtuální laboratoře, mobilní zařízení a vzdálené laboratoře. Po zkoumání těchto variant bylo zjištěno, že vzdálené laboratoře jsou optimální variantou, jelikož umožňují jednotlivcům přístup k reálné laboratoři bez nutnosti osobní přítomnosti, což znamená, že uživatelé nejsou jenom pozorovateli procesu experimentu, ale jeho účastníky. Kromě toho vzdálené laboratoře umožňují studentům pracovat se speciálními vědeckými přístroji, poskytují možnost flexibilního rozvržení, úsporu času a efektivní rozvržení nákladů.

Praktická část práce pak byla věnována návrhu systému vzdálené laboratoře, která by měla být součástí výuky předmětů zahrnujících laboratorní práce na středních a vysokých školách. Proces návrhu byl založen na zkoumané odborné literatuře, zabývající se současnými vzdálenými laboratořemi. V rámci návrhu systému byly nejdříve definovány jeho funkční a nefunkční požadavky. Dále byl vytvořen UML model systému vzdálené laboratoře, zahrnující diagram případů užití, analytický a návrhový model tříd a sekvenční diagramy, které by měly být využité při implementaci systému.

Kromě strukturálního pohledu byl při zpracování diplomové práce využit datový pohled a byly představeny způsoby ukládání různých druhů dat. V systému vzdálené laboratoře by se měly propojit dva typy databáze: relační a časové řady.

Relační databáze se použije pro správu uživatelů laboratoře, specifikace experimentu a plánovaných termínů schůzek. Pro ukládání a analýzu dat ze senzorů, kamer a o celkovém chodu laboratoře, kdy je potřeba sledovat změny stavů v průběhu času, se využije databáze časových řád.

Velká pozornost byla věnována bezpečnosti systému, která je považována za hodně důležitý aspekt návrhu vzdálené laboratoře a dokonce i za jedno z jejích rizik. Rizika mohou vznikat v důsledku špatného používání laboratoře, neoprávněného přístupu, vniknutí malwaru a dalších hrozeb, které jsou vlastní systémům distribuovaným prostřednictvím jakéhokoli typu sítě. Proto jsou do modelu zahrnuté procesy autorizace a autentifikace, ověřující, kdo přistupuje k systému a jaká má přístupová práva. Kromě toho může být spolehlivost vzdálených laboratoří zlepšena pomocí firewallů, více serverů, šifrování dat a složitějších plánovacích algoritmů, které berou v úvahu mezičas do selhání a požadavky na údržbu.

Další práce na systému vzdálené laboratoře by měla být zaměřená na implementaci vytvořeného modelu vzdálené laboratoře. Je nutné vytvořit návrh uživatelského rozhraní systému a vybrat nástroje implementace: zvolit poskytovatele služeb databáze a rozhodnout, jakých programovacích jazyků a případně i frameworků se využije při vývoji. Potom by měl následovat vývoj databáze, klientského a serverového prostředí a konfigurace laboratorních zařízení včetně připojení IP kamer a příprava zařízení pro provádění experimentů a jejich propojení se systémem. Během tohoto procesu musí být celý systém neustále testován.

Po zavedení systému do provozu by se mohlo uvažovat o jeho rozšíření o jiné způsoby interakce a použití, například využít technologii rozšířené reality, která poskytuje značné příležitosti při výuce technických a přírodních věd online.

8 Seznam použité literatury

5 Most Popular Programming Languages & Paradigms. In: Felgo blog [online]. Jan 9, 2022 [cit. 2022-07-22]. Dostupné z: <https://blog.felgo.com/5-popular-programming-languages-examples>

ARLOW, Jim a Ila NEUSTADT. *UML 2 a unifikovaný proces vývoje aplikací: objektově orientovaná analýza a návrh prakticky*. 2., aktualiz. a dopl. vyd. Brno: Computer Press, 2007. ISBN 978-80-251-1503-9.

BASUMALLICK, Chiradeep. Top 10 Firewall Hardware Devices in 2022. In: Spiceworks [online]. 2022 [cit. 2022-07-22]. Dostupné z: <https://www.vox.com/2014/6/16/18076282/the-internet>

BUGAJENKO, Olga. What is the Rational Unified Process? - Methodology, Tools & Examples." Study.com, 19 June 2020, study.com/academy/lesson/what-is-the-rational-unified-process-methodology-tools-examples.html

BHANDARI, A. a M.H. SHOR. Access to an instructional control laboratory experiment through the World Wide Web. In: *Proceedings of the 1998 American Control Conference. ACC (IEEE Cat. No.98CH36207)* [online]. IEEE, 1998, 1998, 1319-1325 vol.2 [cit. 2022-07-21]. ISBN 0-7803-4530-4. Dostupné z: doi:10.1109/ACC.1998.703628

COCKBURN, Alistair. *Writing Effective Use Cases*. Pearson Addison Wesley Prof, 2000. ISBN 9780201702255.

DB-Engines Ranking of Relational DBMS. In: DB-Engines [online]. 2022 [cit. 2022-07-22]. Dostupné z: <https://db-engines.com/en/ranking/relational+dbms>

DB-Engines Ranking of Time Series DBMS. In: DB-Engines [online]. 2022 [cit. 2022-07-22]. Dostupné z: <https://db-engines.com/en/ranking/time+series+dbms>

Difference between Abstraction and Encapsulation in C++. In: Engineering: LibreTexts [online]. 25.09.2020 [cit. 2022-07-22]. Dostupné z: https://eng.libretexts.org/Courses/Delta_College/C_

[Data Structures/06%3A Abstraction Encapsulation/1.01%3A Difference between Abstraction and Encapsulation](#)

DVOŘÁK, Jiří, KURIŠČÁK, Pavel, LUSTIG, František. iSES Remote Lab SDK – internet School Experimental Studio for Remote Laboratory Software Development Kit [cit. 2022-07-21]. Dostupné z: <http://www.ises.info/index.php/en/systemises/sdkisesstudio>.

FADHEL, Muntazir, Emil SEKERINSKI a Shucaï YAO. A Comparison of Time Series Databases for Storing Water Quality Data. In: AUER, Michael E. a Thrasyvoulos TSIATSOS, ed. Mobile Technologies and Applications for the Internet of Things [online]. Cham: Springer International Publishing, 2019, 2019-04-18, s. 302-313 [cit. 2022-07-21]. Advances in Intelligent Systems and Computing. ISBN 978-3-030-11433-6. Dostupné z: doi:10.1007/978-3-030-11434-3_33

FRANĚK, Zdeněk, 2014. *Objektové metody modelování: výklad jazyka UML, jeho možnosti a praktické ukázky*. Projekt „Inovace studijních programů na Slezské univerzitě, Obchodně podnikatelské fakultě v Karviné“.

GARCIA-ZUBIA, J., D. LOPEZ-DE-IPINA a P. ORDUNA. Towards a canonical software architecture for multi-device WebLabs. In: 31st Annual Conference of IEEE Industrial Electronics Society, 2005. IECON 2005 [online]. IEEE, 2005, 2005, 6 pp.- [cit. 2022-07-21]. ISBN 0-7803-9252-3. Dostupné z: doi:10.1109/IECON.2005.1569236

GILLIS, Alexander S. What is the internet of things (IoT) In: *Techtarget* [online]. 2022 [cit. 2022-07-22]. Dostupné z: <https://www.techtargget.com/iotagenda/definition/Internet-of-Things-IoT>

GREGORCIC, Bor a Madelen BODIN. Algodoo: A Tool for Encouraging Creativity in Physics Teaching and Learning. *The Physics Teacher* [online]. 2017, 55(1), 25-28 [cit. 2022-01-26]. ISSN 0031-921X. Dostupné z: doi:10.1119/1.4972493

GUSTAVSSON, I., K. NILSSON, J. ZACKRISSON, et al. On Objectives of Instructional Laboratories, Individual Assessment, and Use of Collaborative Remote

Laboratories. IEEE Transactions on Learning Technologies [online]. 2009, 2(4), 263-274 [cit. 2022-01-26]. ISSN 1939-1382. Dostupné z: doi:10.1109/TLT.2009.42

GERŽA, Michal and Pavel POKORNÝ. A Visualisation of the Results of a Thermoforming Process Simulation in the Plastics Industry. In: 21st International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision: WSCG2013 - Poster Proceedings. Plzeň: Západočeská univerzita v Plzni, Katedra informatiky a výpočetní techniky, 2013 [cit. 2022-01-26]. ISBN 978-80-86943-76-3

HANSSON, H., M. JOHANSSON, P. WESTMAN a E. ÅSTRÖM. E-Learning Quality: Aspects and Criteria for Evaluation of E-Learning in Higher Education [online]. [cit. 2022-07-21]. Dostupné z: https://www.researchgate.net/publication/270158432_E-Learning_Quality_Aspects_and_Criteria_for_Evaluation_of_E-Learning_in_Higher_Education

HARDISON, James L., Kimberly DELONG, Philip H. BAILEY a V. Judson HARWARD. Deploying interactive remote labs using the iLab Shared Architecture. In: 2008 38th Annual Frontiers in Education Conference [online]. IEEE, 2008, 2008, S2A-1-S2A-6 [cit. 2022-07-21]. ISBN 978-1-4244-1969-2. Dostupné z: doi:10.1109/FIE.2008.4720536

HARWARD, V.J., J.A. DEL ALAMO, S.R. LERMAN, et al. The iLab Shared Architecture: A Web Services Infrastructure to Build Communities of Internet Accessible Laboratories. Proceedings of the IEEE [online]. 2008, 96(6), 931-950 [cit. 2022-07-21]. ISSN 0018-9219. Dostupné z: doi:10.1109/JPROC.2008.921607

HERRING, George K., Lars Thorben NEUSTOCK a Lambertus HESSELINK. Learning from the unexpected: Statistics and uncertainty in massively scalable online laboratories (MSOL). In: 2018 IEEE Global Engineering Education Conference (EDUCON) [online]. IEEE, 2018, 2018, s. 815-824 [cit. 2022-07-21]. ISBN 978-1-5386-2957-4. Dostupné z: doi:10.1109/EDUCON.2018.8363315

HONG SHEN, ZHENG XU, B. DALAGER, et al. Conducting laboratory experiments over the Internet. *IEEE Transactions on Education* [online]. 42(3), 180-185 [cit. 2022-01-26]. ISSN 00189359. Dostupné z: doi:10.1109/13.779896

How Does an SSL Certificate Work? In: *GlobalSign* [online]. [cit. 2022-07-22]. Dostupné z: <https://www.globalsign.com/en/ssl-information-center/what-is-an-ssl-certificate>

HUSSAIN, Azham, Emmanuel O. C. MKPOJIOGU a Fazillah Mohmad KAMAL. The Role of Requirements in the Success or Failure of Software Projects. *International Soft Science Conference* [online]. 2016 [cit. 2022-07-22]. ISSN 2146-4405. Dostupné z: www.econjournals.com

HYDER, A., S. K. CHOI a D. SCHAEFER. Remotely controlled laboratory experiments: Creation and examples. In: 2010 IEEE Systems and Information Engineering Design Symposium [online]. IEEE, 2010, 2010, s. 62-67 [cit. 2022-01-26]. ISBN 978-1-4244-7519-3. Dostupné z: doi:10.1109/SIEDS.2010.5469676

JavaScript, Java, AJAX & Flash. In: *Go up* [online]. [cit. 2022-07-22]. Dostupné z: <https://www.goup.co.uk/guides/javascript-java-ajax-flash/>

KOŽOUSKOVÁ, Barbora. Co je LMS a jaké výhody může přinést do vaší firmy? In: *Rascasone* [online]. 13.04.2021 [cit. 2022-07-22]. Dostupné z: <https://www.rascasone.com/cs/blog/co-je-lms-vyhody-firmy>

LEE, Timothy B. The internet, explained. In: *Vox Media* [online]. 2015 [cit. 2022-07-22]. Dostupné z: <https://www.vox.com/2014/6/16/18076282/the-internet>

LOWE, D. Integrating Reservations and Queuing in Remote Laboratory Scheduling. *IEEE Transactions on Learning Technologies* [online]. 2013, 6(1), 73-84 [cit. 2022-07-21]. ISSN 1939-1382. Dostupné z: doi:10.1109/TLT.2013.5

LUSTIG, F., BROM, P., KURIŠČÁK, P., SVOBODA, V.: Remote experiments as virtual labs? Examples with pendulum, In: Turčáni, M., Balogh, Z. et al. (eds.), *DIVAI 2020 - 13th international scientific conference on Distance Learning in Applied*

Informatics. 21.-23.9.2020. Štúrovo, Slovakia. Wolters Kluwer. 2020, pp. 395–405. ISBN: 978-80-7598-841-6. ISSN: 2464-7470 (Print), 2464-7489 (Online). 2020.

MONZO, Carlos, Germán COBO, José Antonio MORÁN, Eugènia SANTAMARÍA a David GARCÍA-SOLÓRZANO. Lab@Home: The Open University of Catalonia Hands-on Electronics Laboratory for Online Engineering Education. Electronics [online]. 2020, 9(2) [cit. 2022-01-26]. ISSN 2079-9292. Dostupné z: doi:10.3390/electronics9020222

MONZO, Carlos, Germán COBO, José Antonio MORÁN, Eugènia SANTAMARÍA a David GARCÍA-SOLÓRZANO. Remote Laboratory for Online Engineering Education: The RLAB-UOC-FPGA Case Study. Electronics [online]. 2021, 10(9) [cit. 2022-01-26]. ISSN 2079-9292. Dostupné z: doi:10.3390/electronics10091072

National Instruments. NI myDAQ User Guide and Specifications. Technical Report, National Instruments Corporation: Austin, TX, USA, 2011 [cit. 2022-01-26]. Dostupné z: <https://www.ni.com/pdf/manuals/373060g.pdf>

National Instruments, NI Analog Discovery 2—NI Edition Overview. Technical Report, National Instruments Corporation, 2018 [cit. 2022-01-26]. Dostupné z: <https://www.ni.com/pdf/manuals/AnalogDiscovery2NIDatasheet.pdf>

ORDUÑA, Pablo, Jaime IRURZUN, Luis RODRIGUEZ-GIL, Javier GARCIA-ZUBIA, Fabricio GAZZOLA a Diego LÓPEZ-DE-IPÍÑA. Adding New Features to New and Existing Remote Experiments through their Integration in WebLab-Deusto. International Journal of Online and Biomedical Engineering (ijOE) [online]. 2011, 7(S2), 33-39 [cit. 2022-07-21]. ISSN 2626-8493. Dostupné z: doi:10.3991/ijoe.v7iS2.1774

ORDUÑA, Pablo, Philip H. BAILEY, Kimberly DELONG, Diego LÓPEZ-DE-IPÍÑA a Javier GARCÍA-ZUBIA. Towards federated interoperable bridges for sharing educational remote laboratories. Computers in Human Behavior [online]. 2014, 30, 389-395 [cit. 2022-07-21]. ISSN 07475632. Dostupné z: doi:10.1016/j.chb.2013.04.029

PETERSON, Larry L. a Bruce S. DAVIE. *Computer networks: a systems approach*. 5th ed. Burlington: Morgan Kaufmann, 2011. ISBN 978-0-12-385059-1

PILGRIM, Mark. *HTML5: Up and Running*. O'Reilly Media, 2010. ISBN 9780596806026.

Remote-LAB GymKT [cit. 2022-07-21]. Dostupné z: <http://remote-lab.fyzika.net/>

RICHTER, Thomas, David BOEHRINGER a Sabina JESCHKE. LiLa: A European Project on Networked Experiments. In: JESCHKE, Sabina, Ingrid ISENHARDT a Klaus HENNING, ed. *Automation, Communication and Cybernetics in Science and Engineering 2009/2010* [online]. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, 2011-12-24, s. 307-317 [cit. 2022-07-21]. ISBN 978-3-642-16207-7. Dostupné z: doi:10.1007/978-3-642-16208-4_27

SALIHBEGOVIC, Adnan a Omer TANOVIC. Internet based laboratories in engineering education. In: ITI 2008 - 30th International Conference on Information Technology Interfaces [online]. IEEE, 2008, 2008, s. 163-170 [cit. 2022-07-21]. ISBN 978-953-7138-12-7. Dostupné z: doi:10.1109/ITI.2008.4588401

SCHAUER, F., OŽVOLDOVÁ, M., LUSTIG, F. : ISES - Internet School Experimental System for computer based laboratories in physics, to be published Innovations 2009(USA), World Innovations in Engineering Education and Research iNEER Special Volume 2009 [cit. 2022-07-21]

SCHEUCHER, B., BELCHER, J., P. BAILEY, F. SCHEUCHER, Christian GUETL a B. Evaluation results of a 3D virtual environment for internet-accessible physics experiments. In: Proceedings of the ICL 2009, pp. 1139-1150. [online]. Curtin Research Publications, 2009 [cit. 2022-07-21]. ISBN 9783899584813.

SIERRA-FERNÁNDEZ, Jose María, Agustin AGÜERA-PÉREZ, Jose Carlos PALOMARES-SALAS, Manuel Jesús ESPINOSA-GAVIRA, Olivia FLORANCIAS-OLIVEROS a Juan José González de la ROSA. Web-based Remote Lab System for Instrumentation and Electronic Learning. *Advances in Science, Technology and*

Engineering Systems Journal [online]. 2021, 6(4), 100-109 [cit. 2022-07-21]. ISSN 24156698. Dostupné z: doi:10.25046/aj060412

UNESCO. *COVID-19: reopening and reimagining universities, survey on higher education through UNESCO National Commissions* [online]. 2021 [cit. 2022-07-21]. Dostupné z: <https://unesdoc.unesco.org/ark:/48223/pf0000378174.locale=en>

WEILKIENS, Tim, Christian WEISS, Andrea GRASS a Kim Nena DUGGEN. Getting Started. In: *OCEB 2 Certification Guide* [online]. Elsevier, 2016, 2016, s. 1-13 [cit. 2022-07-21]. ISBN 9780128053522. Dostupné z: doi:10.1016/B978-0-12-805352-2.00001-7

WILLIAM, James. 6 Reasons Why You Should Start Using A Multi-Server Environment. In: *Temok* [online]. 2019 [cit. 2022-07-22]. Dostupné z: <https://www.temok.com/blog/multi-server-environment/>

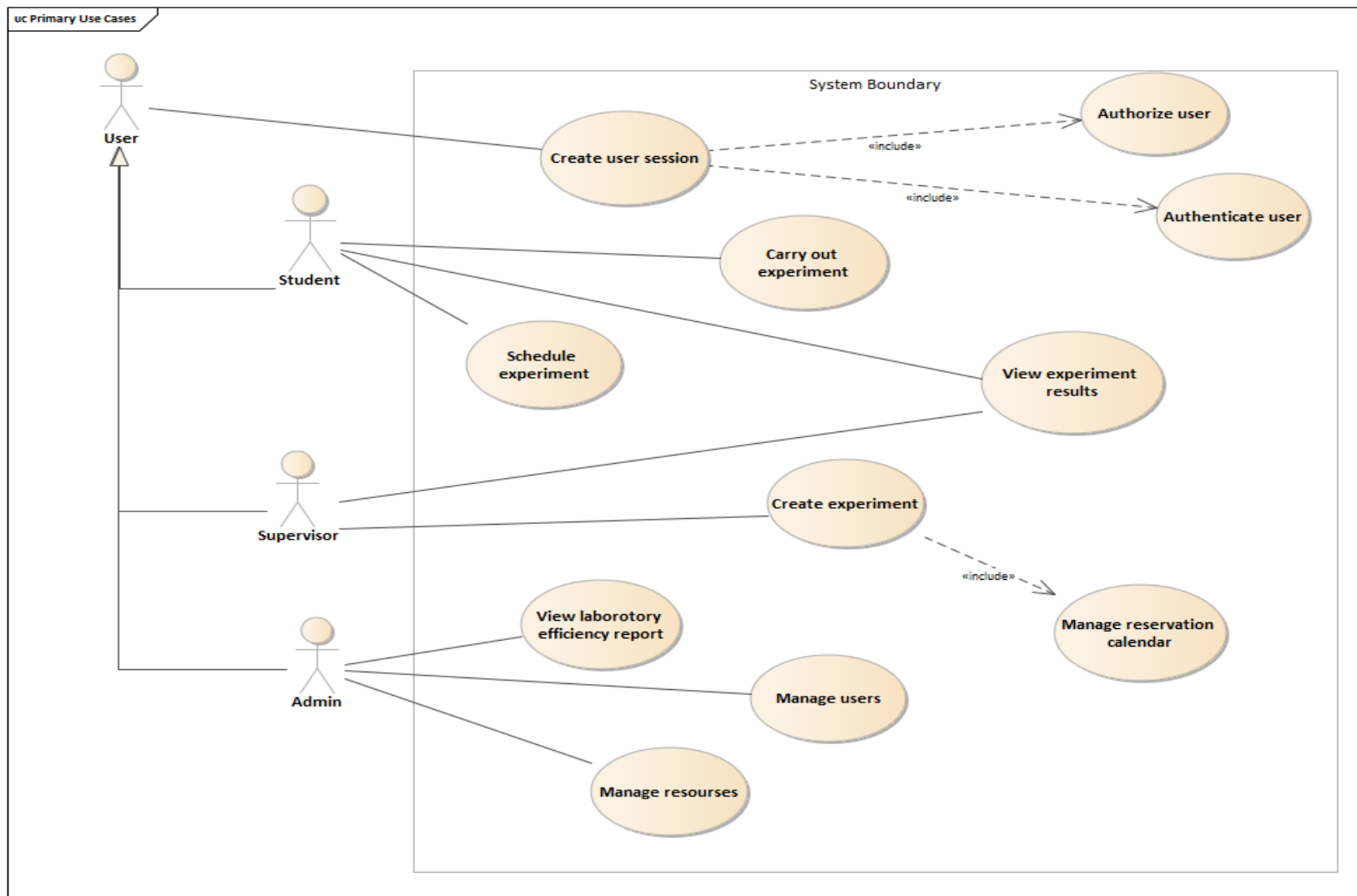
ZAMAN, Mohammad Asif, Lars Thorben NEUSTOCK a Lambertus HESSELINK. ILabs as an online laboratory platform: A case study at Stanford University during the COVID-19 Pandemic. 2021 IEEE Global Engineering Education Conference (EDUCON) [online]. IEEE, 2021, 2021-4-21, s. 1615-1623 [cit. 2022-01-26]. ISBN 978-1-7281-8478-4. Dostupné z: doi:10.1109/EDUCON46332.2021.945402

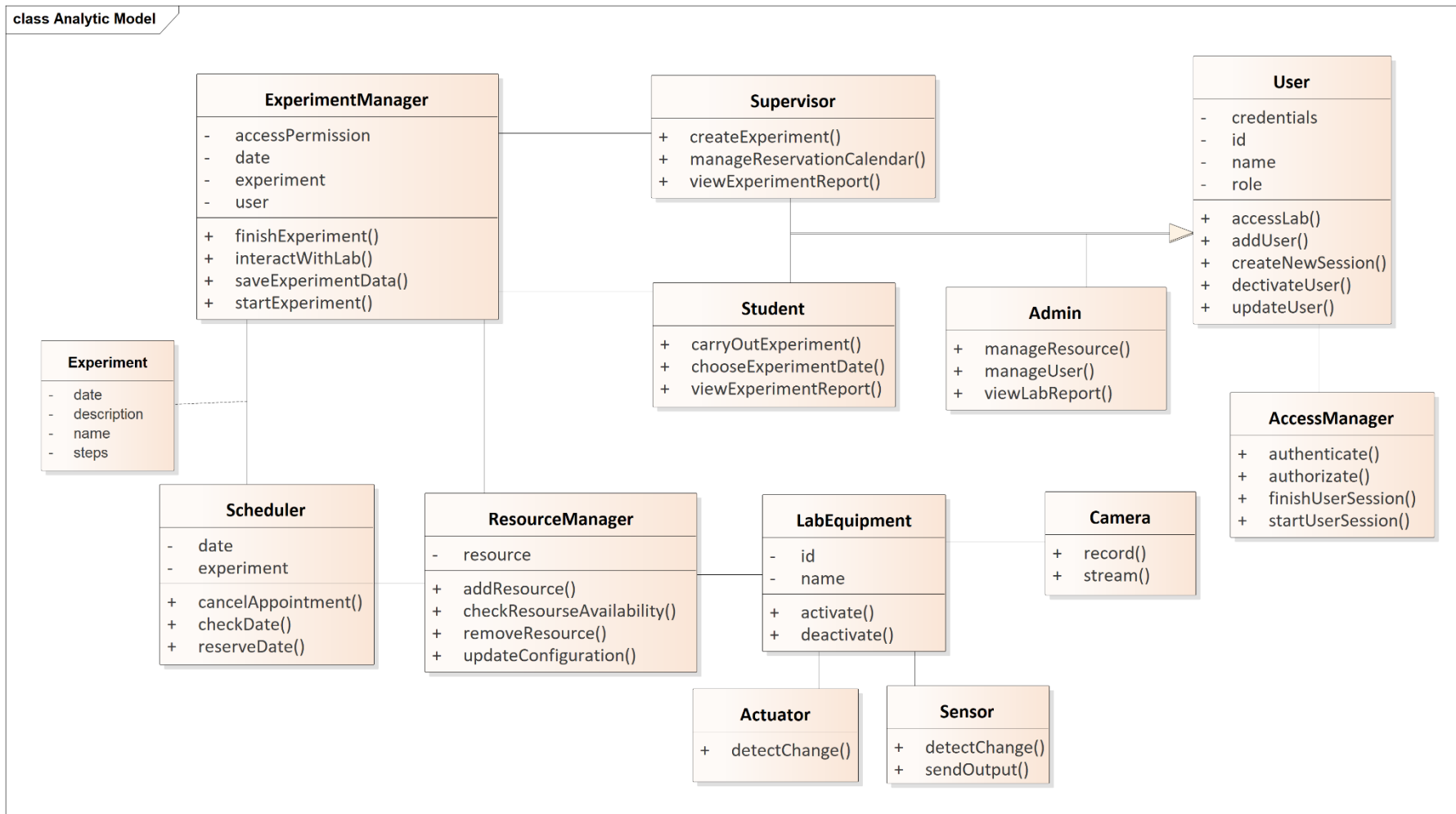
9 Přílohy

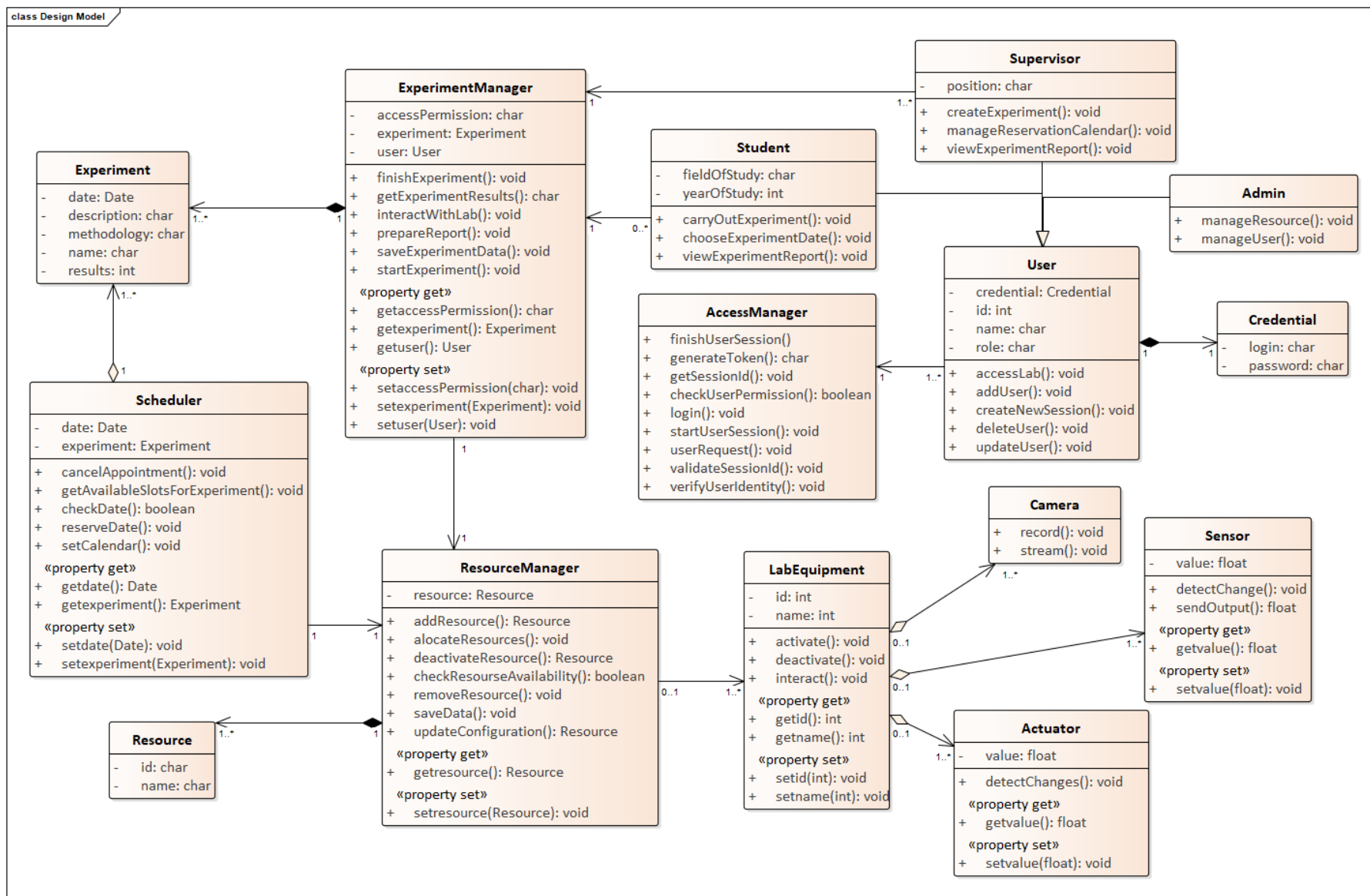
- 1) Specifikace případů užití systému vzdálené laboratoře
- 2) Diagram případů užití systému vzdálené laboratoře
- 3) Analytický model tříd systému vzdálené laboratoře
- 4) Návrhový model tříd systému vzdálené laboratoře
- 5) Sekvenční diagramy k případu užití „Schedule experiment“
- 6) Sekvenční diagramy k případu užití „Carry out experiment“
- 7) Sekvenční diagramy k případu užití „View experiment results“
- 8) Sekvenční diagramy k případu užití „Create user session“
- 9) Sekvenční diagramy k případu užití „Authenticate user“
- 10) Sekvenční diagramy k případu užití „Authorize user“
- 11) Sekvenční diagramy k případu užití „Manage reservation calendar“

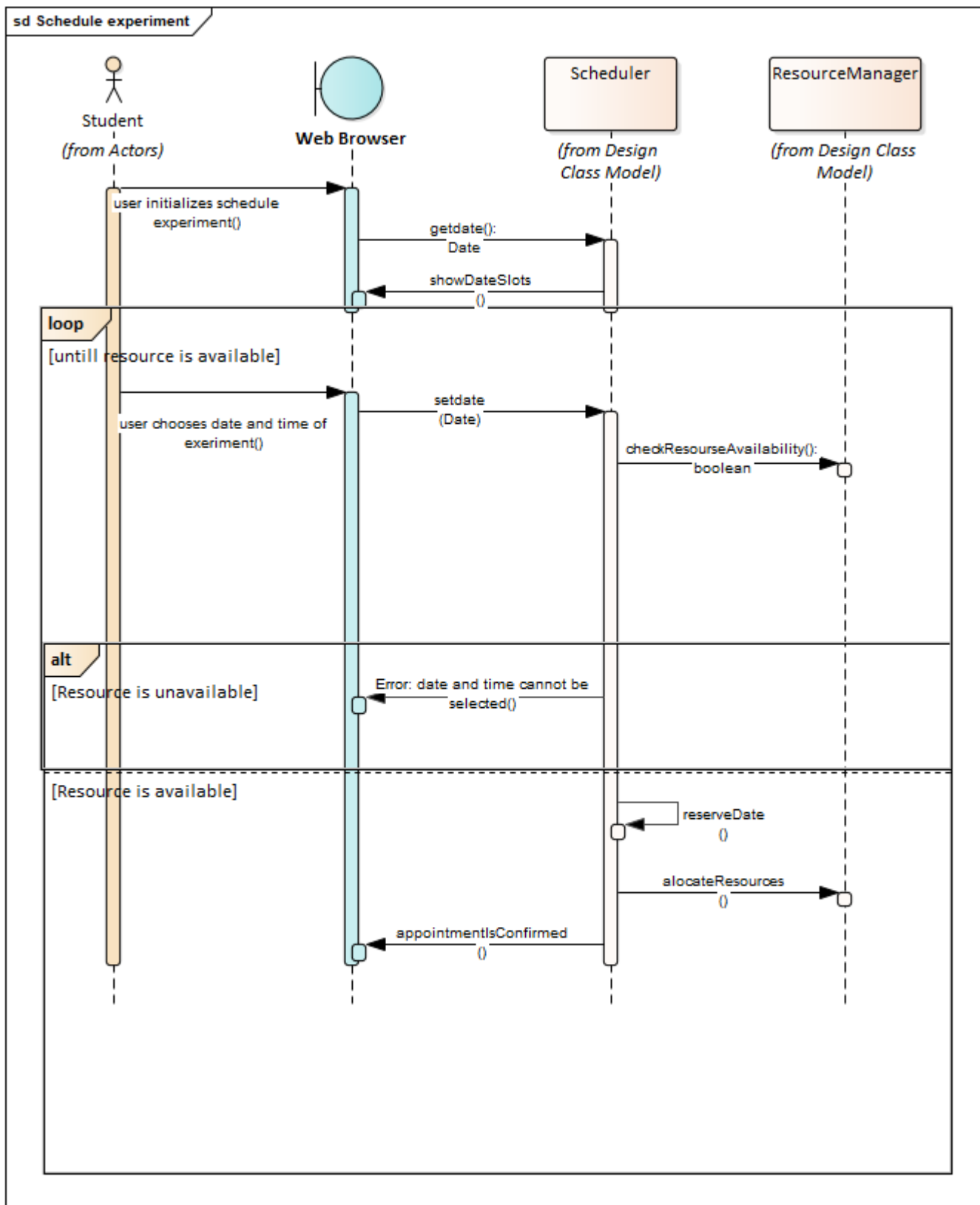
Create user session	
Description	User wants to access to the remote laboratory, so system checks, if user allowed to do that.
Actors	User
Basic path	<ol style="list-style-type: none"> 1. Actor initializes new user session 2. INCLUDE Authenticate user 3. System starts user session 4. INCLUDE Authorize user 5. System redirects user on initial page
Authorize user	
Description	System checks user permissions according to his role in order to hide or disable particular sections and actions of the application
Actors	User
Basic path	<ol style="list-style-type: none"> 1. Actor makes resource request 2. System gets session id 3. System validates session id 4. System checks user permissions
Alternative paths	<p>2a System doesn't get session id 2a1 System shows warning that user is not logged in</p> <p>3a System evaluates session id as invalid (wrong id or it is expired) 3a1 System shows warning that user is not logged in</p> <p>5a User doesn't have the permissions 5a1 System informs users that operation is not allowed</p>
Authenticate user	
Description	User logs in to the system
Actors	User
Basic path	<ol style="list-style-type: none"> 1. Actor chooses Log in option 2. System redirects to the log in page 3. Actor fills in its credentials and clicks on "Log in" button 4. System verifies user identity 5. System generates authentication token

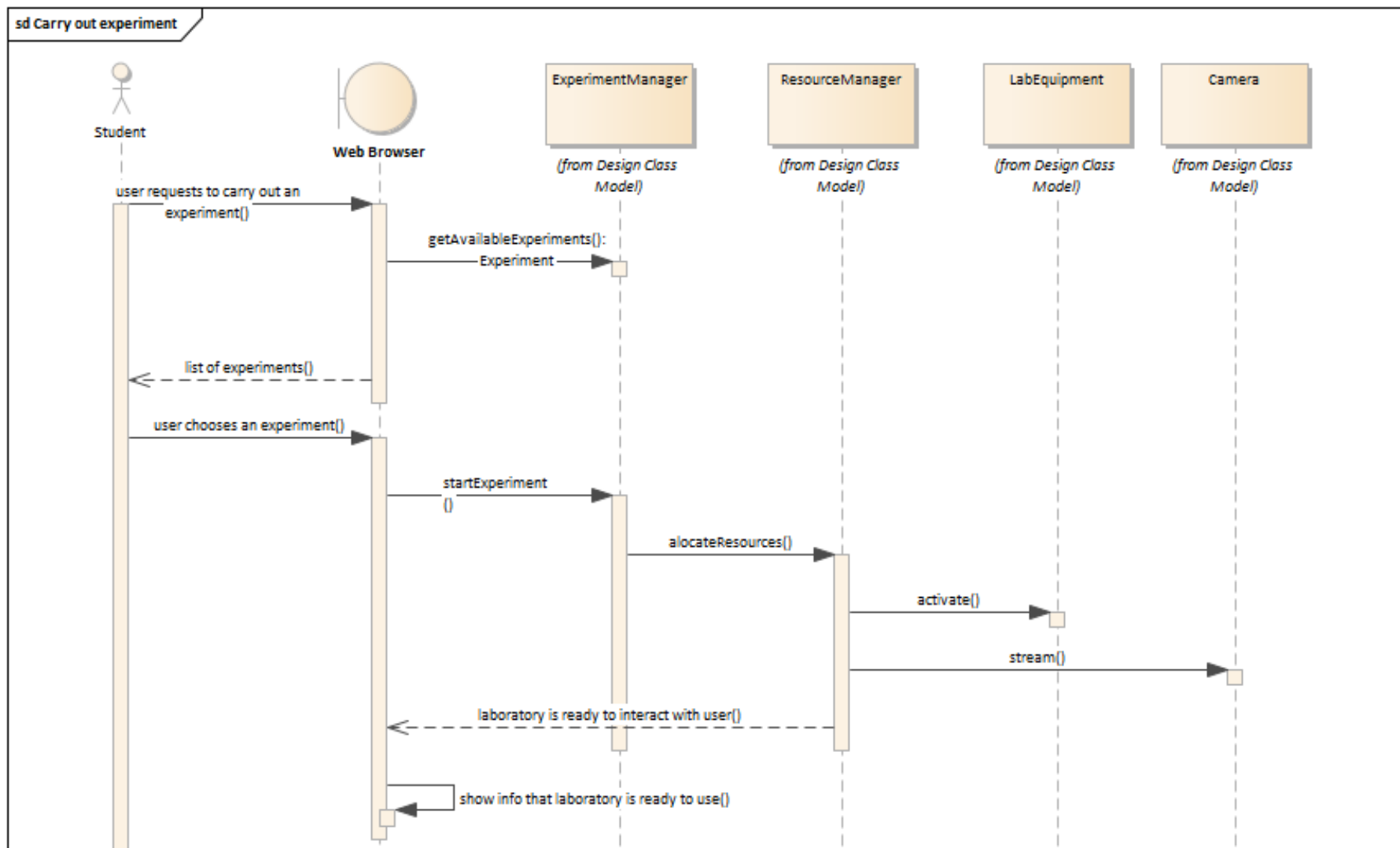
Alternative paths	5a System valuates user's credentials as invalid 5a1 System informs user, that combination username-password are not found
Create experiment	
Description	Actor adds new experiment to the remote laboratory
Actors	Supervisor
Basic path	<ol style="list-style-type: none"> 1. Actor initializes Create new experiment 2. System loads and show the new experiment form 3. Actor fills the form and clicks on Submit button 4. System validates the form 5. System suggests actor to choose needed for experiment equipment 6. Actor chooses equipment 7. System suggests actor to set data slots for experiment reservation 8. INCLUDE Manage reservation calendar 9. System creates new experiment
Alternative paths	5a System validates form as invalid 5a1 System asks user to fix the form data 5a2 System continues from the step 3 of a Basic path
Manage reservation calendar	
Description	Actor chooses list of dates when the selected experiment can be carried out
Actors	Supervisor
Basic path	<ol style="list-style-type: none"> 1. Actor initiates Manage reservation calendar 2. System check equipment availability 3. System loads free data slots for an experiment 4. Actor chooses list of data slots 5. System reserve resources for chosen dates 6. System continues on the basic UC
View laboratory efficiency report	
Description	Actor gets the laboratory efficiency report
Actors	Admin
Basic path	<ol style="list-style-type: none"> 1. Actor initializes to view laboratory efficiency report 2. System exports data about laboratory 3. System prepare efficiency report and loads it to user 4. Actor downloads report

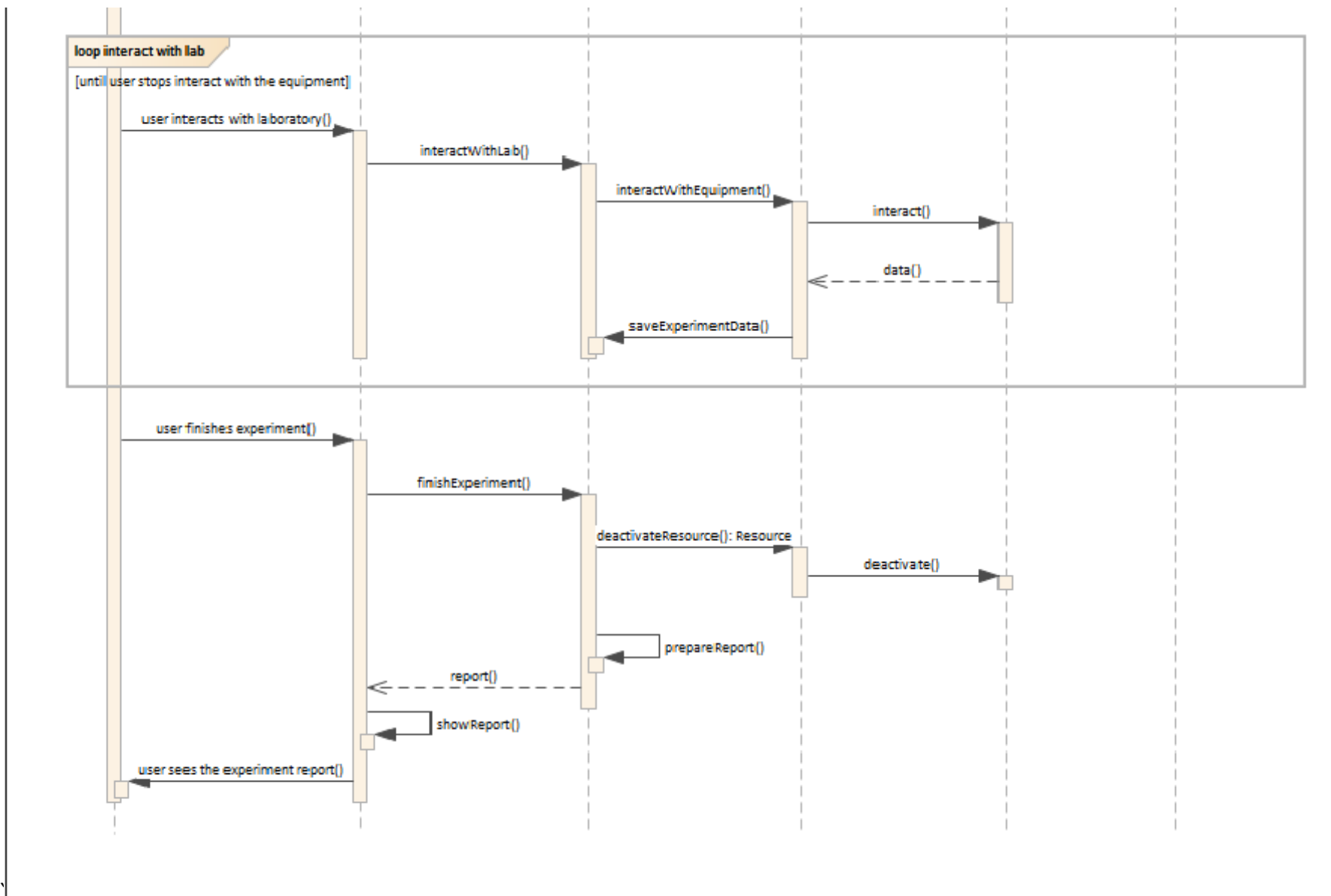


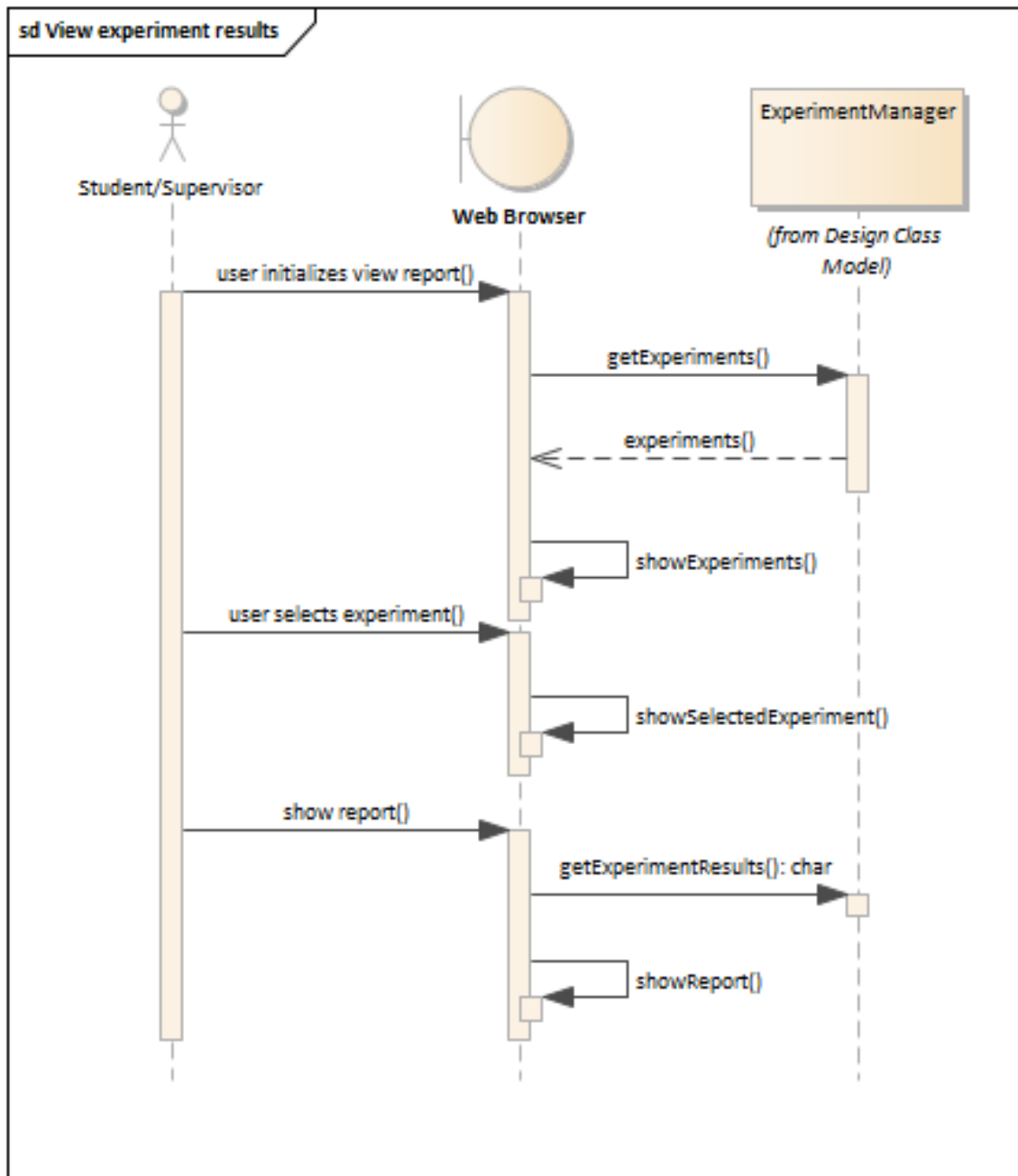


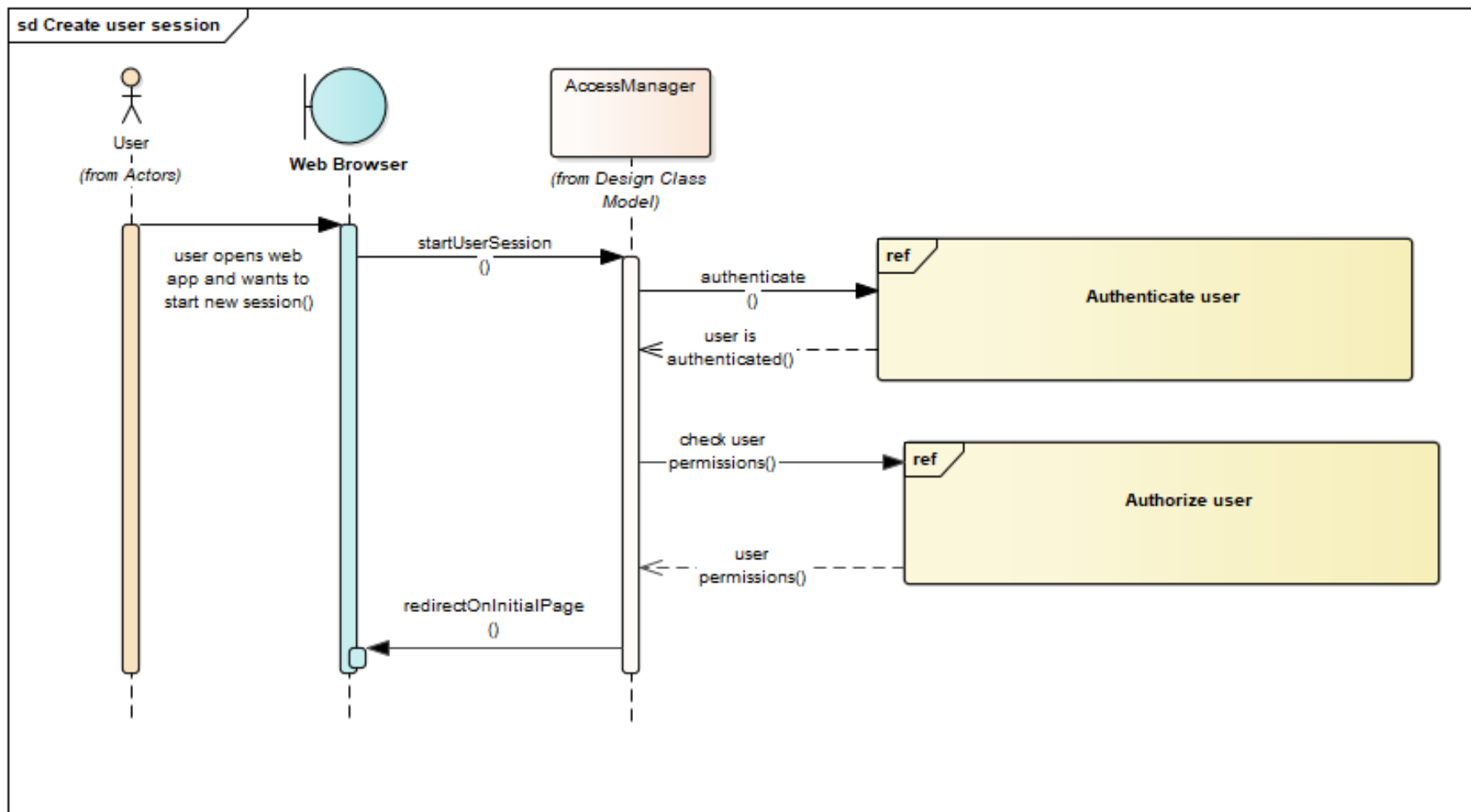


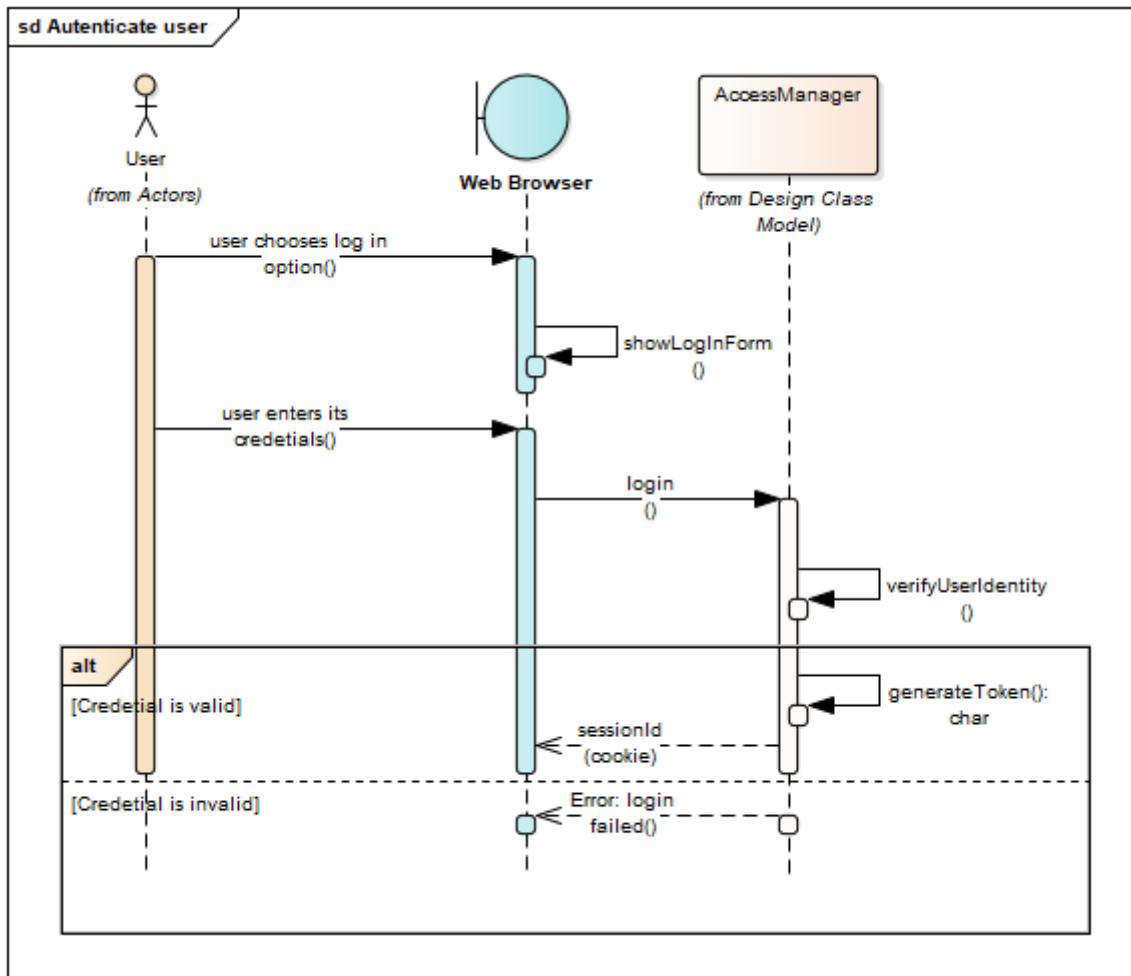


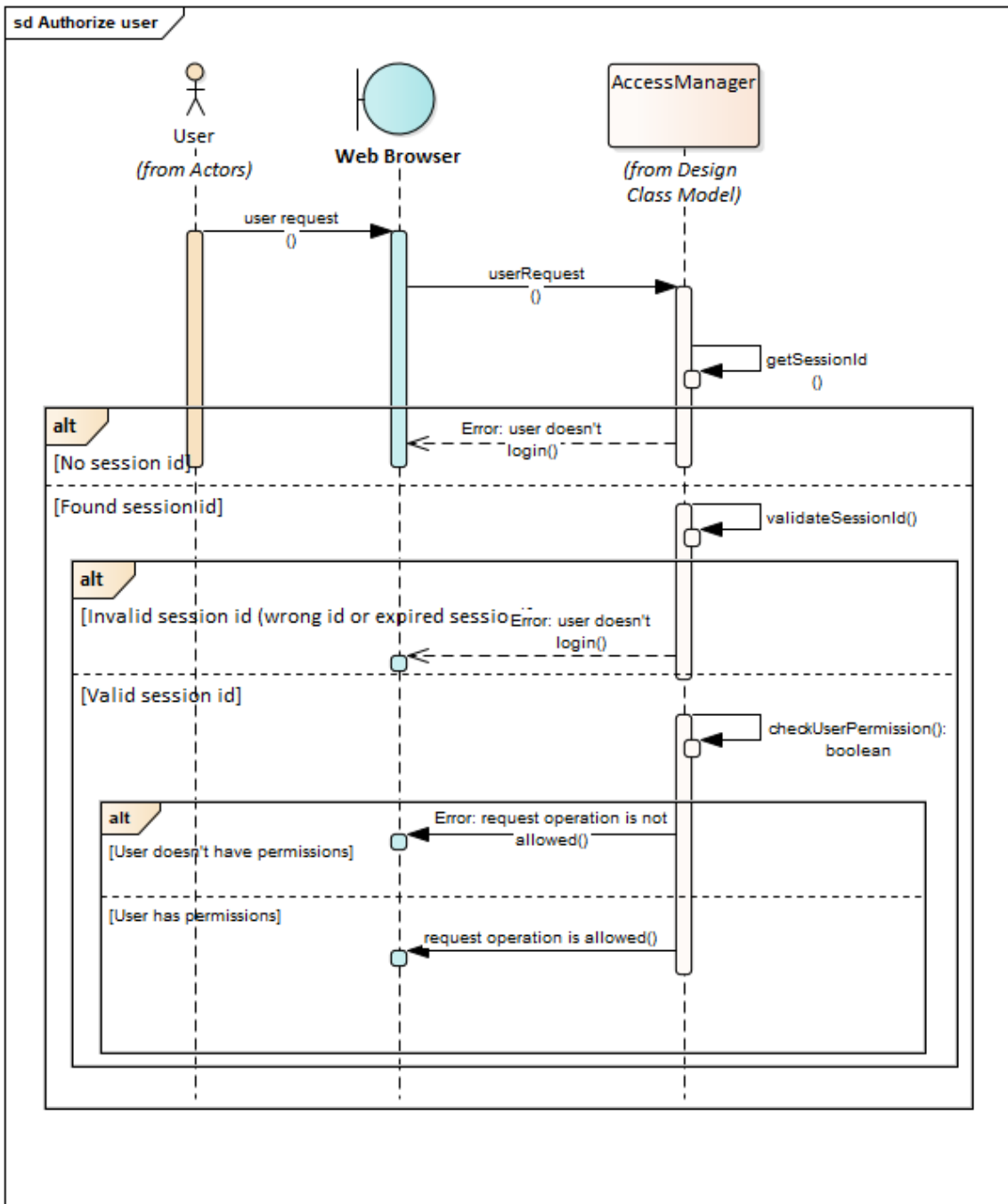


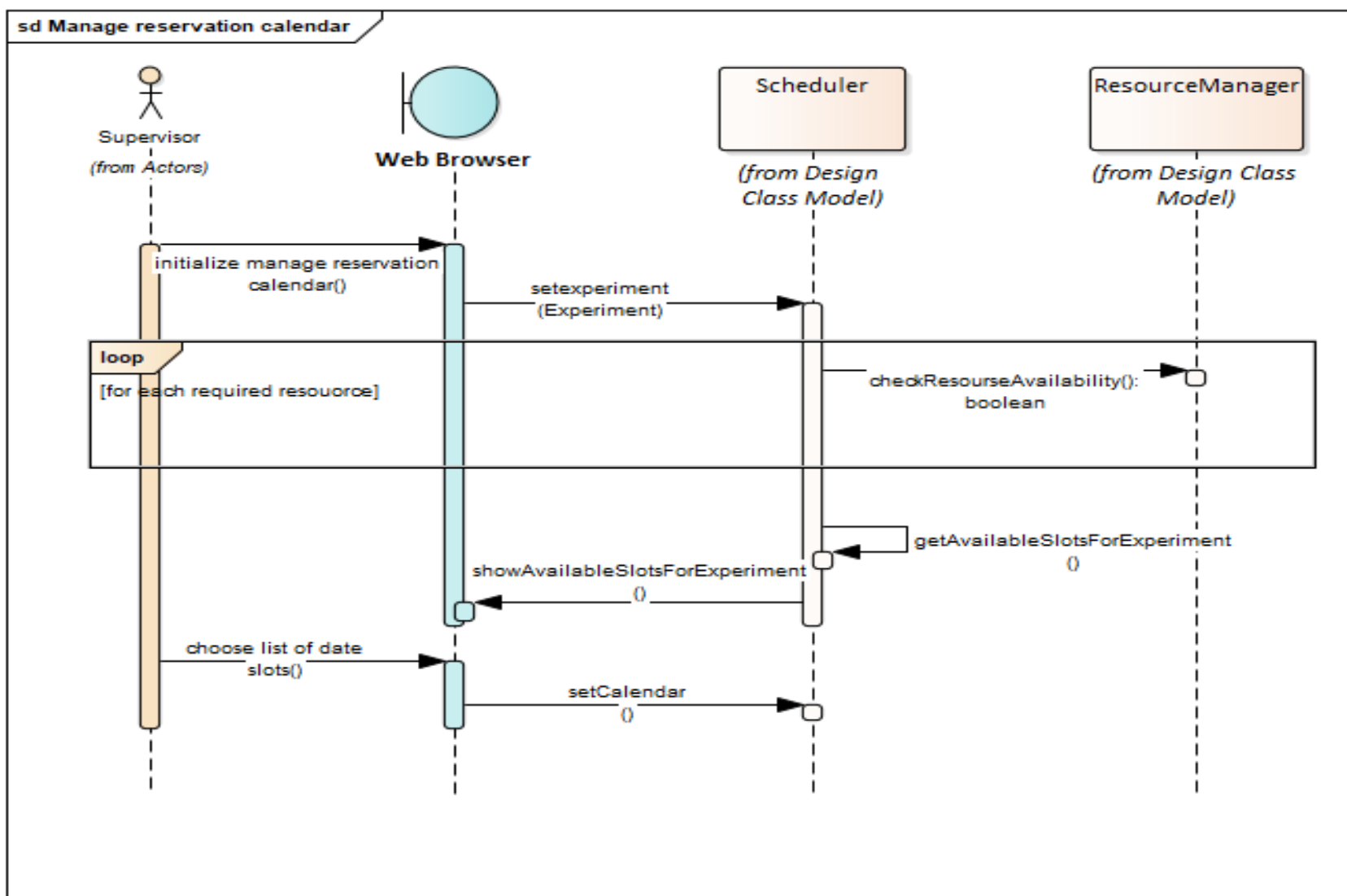












Podklad pro zadání DIPLOMOVÉ práce studenta

Jméno a příjmení: **Bc. Hanna Kukharava**
Osobní číslo: **I2000082**
Adresa: **Kejzlarova 1677, Hradec Králové – Nový Hradec Králové, 50012 Hradec Králové 12, Česká republika**
Téma práce: **Systémový přístup k návrhu vzdálené laboratoře**
Téma práce anglicky: **System approach to remote laboratory design**
Vedoucí práce: **Ing. Karel Mls, Ph.D.**
Katedra informačních technologií

Zásady pro vypracování:

Cíl: Na základě analýzy současného stavu využívání vzdálených laboratoří v různých oblastech navrhnout vlastní řešení včetně namodelování procesů přístupu a využití této laboratoře.

Osnova:

1. Úvod
2. Teoretické poznatky z oblasti vzdálených laboratoří
3. Analýza současného stavu využívání vzdálených laboratoří
5. Praktická část: návrh vlastní vzdálené laboratoře
6. Shrnutí výsledků a závěr

Seznam doporučené literatury:

GUSTAVSSON, Ingvar, et al. On objectives of instructional laboratories, individual assessment, and use of collaborative remote laboratories. *IEEE Transactions on learning technologies*, 2009, 2.4: 263-274.
PETERSON, Larry L.; DAVIE, Bruce S. *Computer networks: a systems approach*. Elsevier, 2007.

Podpis studenta:

Datum:

Podpis vedoucího práce:

Datum: