

ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA STROJNÍ

Studijní program: N0715A270012 Průmyslové inženýrství a management

DIPLOMOVÁ PRÁCE

Tvorba mobilní aplikace pro zobrazení stavu Fischertechnik modelu

Autor: Bc. Marek VELICH
Vedoucí práce: Ing. Pavel RAŠKA, Ph.D.

Akademický rok 2021/2022

ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta strojní

Akademický rok: 2021/2022

ZADÁNÍ DIPLOMOVÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Bc. Marek VELICH**
Osobní číslo: **S19N0147P**
Studijní program: **N0715A270012 Průmyslové inženýrství a management**
Téma práce: **Tvorba mobilní aplikace pro zobrazení stavu Fischertechnik modelu**
Zadávací katedra: **Katedra průmyslového inženýrství a managementu**

Zásady pro vypracování

1. Úvod do řešené problematiky
2. Analýza současného stavu
3. Návrh řešení
4. Testování řešení a analýza výsledků
5. Závěr

Rozsah diplomové práce: **50 – 70 stran**
Rozsah grafických prací: **0**
Forma zpracování diplomové práce: **tištěná**

Seznam doporučené literatury:

- USTUNDAG, Alp a Emre CEVIKCAN. *Industry 4.0: managing the digital transformation*. Cham: Springer International Publishing, 2018. Series in advanced manufacturing (Springer). ISBN 978-3-319-57869-9.
- NAKOV, S. *Fundamentals of computer programming with C# : The Bulgarian C# programming book*. Svetlin Nakov, 2013. ISBN 9789544007737.
- SCMALZ, M. *Using databases with C#*. O’ Reilly Media, 2011. ISBN 9781449309985.

Vedoucí diplomové práce: **Ing. Pavel Raška, Ph.D.**
Katedra průmyslového inženýrství a managementu

Konzultant diplomové práce: **Doc. Ing. Zdeněk Ulrych, Ph.D.**
Katedra průmyslového inženýrství a managementu

Datum zadání diplomové práce: **20. září 2021**

Termín odevzdání diplomové práce: **27. května 2022**

L.S.

Doc. Ing. Milan Edl, Ph.D.
děkan

Doc. Ing. Michal Šimon, Ph.D.
vedoucí katedry

Prohlášení o autorství

Předkládám tímto k posouzení a obhajobě diplomovou práci, zpracovanou na závěr studia na Fakultě strojní Západočeské univerzity v Plzni.

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně, s použitím odborné literatury a pramenů, uvedených v seznamu, který je součástí této diplomové práce.

V Plzni dne:

.....

podpis autora

ANOTAČNÍ LIST DIPLOMOVÉ PRÁCE

AUTOR	Příjmení Velich	Jméno Marek		
STUDIJNÍ PROGRAM	N0715A270012 Průmyslové inženýrství a management			
VEDOUCÍ PRÁCE	Příjmení (včetně titulů) Ing. Raška, Ph.D.	Jméno Pavel		
PRACOVISŤE	ZČU - FST - KPV			
DRUH PRÁCE	DIPLOMOVÁ	BAKALÁŘSKÁ	Nehodící se škrtněte	
NÁZEV PRÁCE	Tvorba mobilní aplikace pro zobrazení stavu Fischertechnik modelu			

FAKULTA	strojní	KATEDRA	KPV	ROK ODEVZD.	2022
----------------	---------	----------------	-----	--------------------	------

POČET STRAN (A4 a ekvivalentů A4)

CELKEM	66	TEXTOVÁ ČÁST	66	GRAFICKÁ ČÁST	0
---------------	----	---------------------	----	----------------------	---

STRUČNÝ POPIS (MAX 10 ŘÁDEK) ZAMĚŘENÍ, TÉMA, CÍL POZNATKY A PŘÍNOSY	<p>Cílem této práce je vytvoření mobilní aplikace pro zobrazení stavu Fischertechnik modelu. V práci je nejprve provedena rešerše na téma Průmysl 4.0, mobilních aplikací a jejich vývoje, dostupných technologií pro tvorbu mobilních aplikací, práce s databázemi a současného stavu Fischertechnik modelu. Práce dále pokračuje návrhem jednotlivých funkcionalit a GUI mobilní aplikace a její následnou tvorbou s využitím frameworku Xamarin.</p>
KLÍČOVÁ SLOVA ZPRAVIDLA JEDNOSLOVNÉ POJMY, KTERÉ VYSTIHUJÍ PODSTATU PRÁCE	<p>Mobilní aplikace, Xamarin, Multiplatformní, Fischertechnik</p>

SUMMARY OF DIPLOMA SHEET

AUTHOR	Surname Velich	Name Marek
STUDY PROGRAMME	N0715A270012 Industrial Engineering and Management	
SUPERVISOR	Surname (Inclusive of Degrees) Ing. Raška, Ph.D.	Name Pavel
INSTITUTION	ZČU - FST - KPV	
TYPE OF WORK	DIPLOMA	BACHELOR Delete when not applicable
TITLE OF THE WORK	Creating a mobile application to display the state of a Fischertechnik model	

FACULTY	Mechanical Engineering	DEPARTMENT	Industrial Engineering and Management	SUBMITTED IN	2022
----------------	------------------------	-------------------	---------------------------------------	---------------------	------

NUMBER OF PAGES (A4 and eq. A4)

TOTALLY	66	TEXT PART	66	GRAPHICAL PART	0
----------------	----	------------------	----	-----------------------	---

BRIEF DESCRIPTION TOPIC, GOAL, RESULTS AND CONTRIBUTIONS	The aim of this work is to create a mobile application to display the state of the Fischertechnik model. In this thesis, first a research is conducted on Industry 4.0, mobile applications and their development, available technologies for mobile application development, working with databases and the current state of the Fischertechnik model. The work continues with the design of individual functionalities and GUI of the mobile application and its subsequent creation using the Xamarin framework.
KEY WORDS	Mobile application, Xamarin, Multiplatform, Fischertechnik

Poděkování

Tímto bych chtěl poděkovat svému vedoucímu diplomové práce Ing. Pavlovi Raškovi, Ph.D. za odborné vedení, jeho rady a pomoc. Dále děkuji konzultantovi Doc. Ing. Zdeňkovi Ulrychovi, Ph.D. a Ing. Miroslavu Malagovi za ochotu při testování mobilní aplikace a svým kolegům Ing. Pavlovi Volkovinskému a Bc. Tomášovi Smíškovi za jejich cenné rady v oblasti databází a softwarového vývoje. Zároveň bych rád poděkoval mé rodině a blízkým za jejich podporu a trpělivost.

Seznam obrázků

Obr. 1 - Přehled průmyslových revolucí [3]	14
Obr. 2 - Schéma propojení modelu s databázi SQL [zdroj vlastní, 43, 44, 45, 46, 47, 48]	31
Obr. 3 - Fotografie pracoviště sklad.....	32
Obr. 4 - Fotografie pracoviště kontroly kvality.....	33
Obr. 5 - Fotografie manipulačního robota.....	34
Obr. 6 - Fotografie pracoviště obrábění	35
Obr. 7 - Fotografie svářecího pracoviště	36
Obr. 8 - Datový model k Fischertechnik modelu	37
Obr. 9 - Návrh GUI pro stránku Fischertechnik model.....	43
Obr. 10 - Návrh GUI pro stránky pracoviště.....	44
Obr. 11 - Vývojový diagram pro získání dat z databáze	48
Obr. 12 - Zdrojový kód pro získání dat z databáze	49
Obr. 13 - Ikony jednotlivých pracovišť	50
Obr. 14 - Ukázka zobrazení aktivního pracoviště	51
Obr. 15 - Informační panel stránky Fischertechnik model.....	52
Obr. 16 - Celková doba strávená na pracovištích.....	52
Obr. 17 - Historie kvality dílu u posledních deseti výrobních cyklů	53
Obr. 18 - Přehled počtu vyrobených dílů	53
Obr. 19 - Panel zobrazující podmínky okolního prostředí	54
Obr. 20 - Signalizace závady na pracovišti	54
Obr. 21 - Výsledná podoba stránky Fischertechnik model	55
Obr. 22 - Aktuální status pracoviště.....	56
Obr. 23 - Výsledná podoba stránek Sklad a Kontrola kvality.....	57
Obr. 24 - Výsledná podoba stránek Manipulační robot, Obrábění a Svařování	57
Obr. 25 – Výsledná podoba stránky export dat.....	58
Obr. 26 – Výsledná podoba informační stránky	59

Seznam tabulek

Tab. 1 - Porovnání technických zpracování mobilních aplikací [13, 14, 15, 16, 17, 18]	19
Tab. 2 - Porovnání nativního a multiplatformního vývoje [26, 27, 28]	23
Tab. 3 - Porovnání frameworků React Native, Flutter a Xamarin [29, 30, 31, 32, 33]	25

Seznam použitých zkratk

IoT	Internet of Things
AI	Artificial Intelligence
OS	Operační systém
IDE	Integrated Development Environment
API	Application Programming Interface
GUI	Graphical User Interface
TV	Televize
UWP	Universal Windows Platform
SŘBD	Systém Řízení Báže Dat
SQL	Structured Query Language
ZČU	Západočeská univerzita v Plzni
MSSQL	Microsoft SQL
s	sekunda
VPN	Virtual Private Network
IP	Internet Protocol
dB	decibel
MS	Microsoft
GB	gigabyte

Obsah

Seznam obrázků	8
Seznam tabulek	9
Seznam použitých zkratk.....	10
Úvod.....	13
1 Průmysl 4.0 a chytrá továrna.....	14
1.1 Průmyslové revoluce	14
1.2 Průmysl 4.0.....	15
1.3 Model Fischertechnik a Průmysl 4.0	16
2 Mobilní aplikace a jejich vývoj.....	17
2.1 Druhy mobilních aplikací	17
2.1.1 Nativní.....	17
2.1.2 Webové.....	18
2.1.3 Hybridní	18
2.2 Cílový operační systém	19
2.3 Strategie vývoje	20
2.4 Fáze vývoje.....	20
3 Technologie pro vývoj mobilních aplikací.....	22
3.1 Frameworky.....	22
3.1.1 Dostupné frameworky	23
3.1.2 Porovnání frameworků a výběr vhodné varianty	24
3.2 Programovací jazyky, jazyk C#.....	25
4 Práce s databázemi	28
5 Aktuální stav	31
5.1 Model Fischertechnik	31
5.2 Databáze	37
6 Návrh mobilní aplikace	38
6.1 Účel a využití mobilní aplikace	38
6.2 Návrh funkcionalit.....	38
6.3 Získání potřebných dat pro jednotlivé funkcionality.....	40
6.3.1 Pohled 1 – Aktuálně probíhající akce	41
6.3.2 Pohled 2 – Status kvality výrobků.....	41
6.3.3 Pohled 3 – Časy jednotlivých aktivit + chybovost frézky a vrtačky	41

6.3.4	Pohled 4 – Zmetkovitost + takt výroby jednoho OK dílu	41
6.3.5	Pohled 5 – Doba strávená na pracovištích	41
6.3.6	Pohled 6 – Přehled vyrobených dílů	42
6.4	Návrh GUI	42
7	Implementace	45
7.1	Struktura aplikace	45
7.2	Jazyk	45
7.3	Výběr šablony	45
7.4	Balíčky (knihovny)	45
7.5	Získávání dat z databáze	47
7.6	Tvorba ikon	50
7.7	Stránky mobilní aplikace a jejich funkcionality	50
7.7.1	Hlavní stránka – Fischertechnik model	50
7.7.2	Stránky pracovišť	55
7.7.3	Stránka export dat (DATA EXPORT)	58
7.7.4	Informační stránka (ABOUT)	59
7.8	Podmínky pro spuštění	59
8	Testování	60
8.1	Testování mobilní aplikace na fyzickém mobilním zařízení	60
8.2	Sumarizace výsledků a možná rozšíření	61
	Závěr	62
	Seznam literatury a dalších pramenů	63

Úvod

Doby, kdy z mobilního telefonu šlo pouze telefonovat, jsou dávno pryč. Od prvního veřejnosti dostupného modelu uběhlo již téměř 40 let, a zatímco tehdy bylo raritou jeden takový vlastnit, v dnešní době je tomu přesně naopak. Dle studie britské organizace Childwise z roku 2020, dostane 53 % dětí svůj první mobilní telefon okolo 7. roku života. Není proto divu, že před dovršením dospělosti má mobilní telefon téměř každý. Dalo by se tak říct, že mobilní telefon je věc naprosto běžná, například jako televize, nebo automobil. Díky tomu se začala vyvíjet zařízení, která lze s mobilním telefonem snadno propojit. Není už tedy žádnou novinkou, že lze pomocí telefonu ovládat určité funkcionality osobního automobilu nebo tzv. chytrou domácnost, která se může skládat ze zařízení, jako jsou např. chytré zásuvky, žárovky, spotřebiče, nebo i květináče. A když je takto možné ovládat automobil či domácnost, proč by nemohlo být možné ovládat celý obchod? A co teprve, kdyby bylo možné ovládat celý výrobní podnik. S rozvojem čtvrté průmyslové revoluce, která je založena na principu digitalizace a automatizace, by tento koncept výrobního podniku mohl být během několika let naprosto běžnou záležitostí. Již dnes jsou totiž k dispozici zcela automatizované výrobní linky, stroje nebo autonomní vysokozdvizné vozíky. Smyslem Průmyslu 4.0 je umožnit těmto zařízením a strojům mezi sebou komunikovat v reálném čase (tvorba kyberneticko – fyzikálních systémů) a přistupovat k nim pomocí vzdáleného přístupu, právě pomocí mobilních telefonů, tabletů nebo počítačů. [1]

Obsahem této diplomové práce je návrh a tvorba mobilní aplikace. Tato mobilní aplikace bude zobrazovat stavy a děje, které probíhají na modelu Fischertechnik. Tento model reprezentuje plně automatizovanou výrobní linku, jejíž stavy a děje, které zde probíhají, jsou zaznamenávány a ukládány do vzdálené databáze. K této databázi se bude mobilní aplikace připojovat a zobrazovat indikátory výkonu jednotlivých pracovišť. Diplomová práce je rozdělena na dvě části – teoretickou, kde je provedena rešerše dostupných technologií, aktuálního stavu modelu Fischertechnik a praktickou, ve které je postup návrhu a tvorby mobilní aplikace.

V první části jsou nejprve představeny klíčové pojmy Průmyslu 4.0 a chytrých továren a uvedení modelu Fischertechnik do kontextu s těmito pojmy. V práci jsou dále představeny druhy mobilních aplikací a strategie a fáze vývoje mobilních aplikací. Následná kapitola je věnována rešerši dostupných technologií pro tvorbu mobilních aplikací a popisu databáze. V závěru této části je provedena rešerše současného stavu modelu Fischertechnik a databáze, do které jsou ukládána data generovaná v modelu Fischertechnik.

Ve druhé části je vzhledem k současnému stavu Fischertechnik modelu proveden návrh mobilní aplikace, který se skládá z návrhu jednotlivých funkcionalit této mobilní aplikace a jejího grafického uživatelského rozhraní. Následně je popsána samotná tvorba, ve které jsou představeny technologie, o jejichž použití bylo rozhodnuto až v průběhu tvorby mobilní aplikace. Dále je popsána struktura aplikace a jednotlivé stránky v mobilní aplikaci. V závěru této části je popsáno testování mobilní aplikace.

1 Průmysl 4.0 a chytrá továrna

V dnešní době je kladen stále větší důraz na využití prvků Průmyslu 4.0 ve výrobních podnicích. Velká část výrobních podniků by velmi ráda dosáhla bodu, kdy bude jejich výroba zcela automatizována a nebude potřeba téměř žádné lidské pracovní síly. Dosažení takového stavu by mělo za výhodu zvýšení produktivity v důsledku snížení chyb a defektů způsobených lidským faktorem. Zároveň by došlo k úsporám finančních prostředků, které jsou vynaloženy na opravu takových chyb a defektů, na mzdy a na nutnost uzpůsobení výrobního podniku dle norem v závislosti na počtu pracovníků v daném závodě. V neposlední řadě by bylo možné takový výrobní podnik vzdáleně kontrolovat a v některých případech dokonce i ovládat. K tomuto stavu se ale rovněž váže i množství nevýhod. Tou největší je velmi vysoká investice, kterou je nutné na počátku vynaložit. Další nevýhodou je naprostá závislost na elektřině a také její násobně vyšší spotřeba. [2]

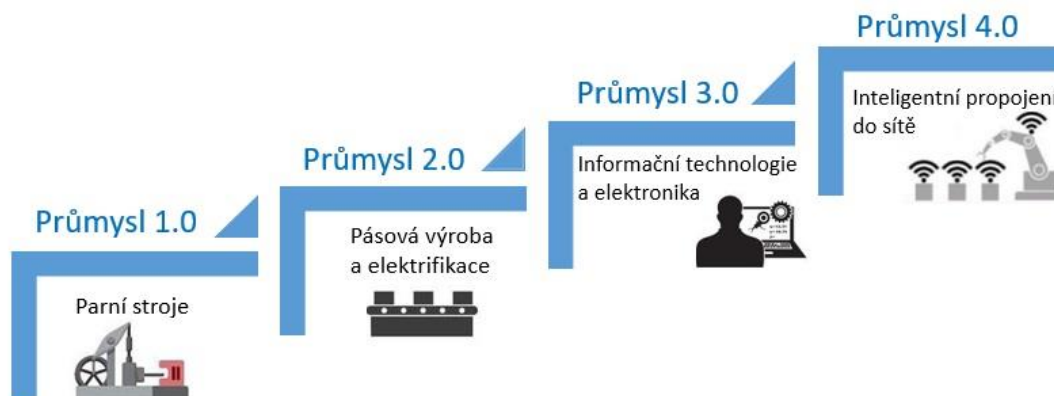
V následujících pododstavcích bude stručně představen Průmysl 4.0 a jeho prvky, které mohou být k takovéto automatizaci a přeměně továrny na „chytrou“ použity.

1.1 Průmyslové revoluce

Aby bylo možné zabývat se Průmyslem 4.0, je vhodné tento pojem zařadit do souvislosti s předchozími průmyslovými revolucemi. Ty v historii proběhly už celkem 3:

- Průmysl 1.0 – mechanizace, vodní energie, parní energie
- Průmysl 2.0 – montážní linky, elektrifikace
- Průmysl 3.0 – automatizace, informační technologie

První průmyslová revoluce probíhala od konce 18. století do poloviny 19. století a charakterizovat by se dala slovem industrializace, což je proces, při kterém dochází k přeměně předprůmyslové společnosti ve společnost industriální. Tradičními symboly této revoluce je parní stroj, mechanický tkalcovský stav a počátek masového využívání zdrojů energie. Druhá průmyslová revoluce, která navazovala na tu první, trvala až do poloviny 20. století. Je spojována především s elektrifikací, se vznikem montážních linek a tím i pásovou výrobou. Třetí průmyslová revoluce se nesla v duchu automatizace a rozvoje informačních technologií a trvala až do nedávna. Přesná data u jednotlivých revolucí nelze přesně určit. Přejít mezi nimi se většinou datuje k nejradikálnějším vynálezům té doby, jejich posouzení je však sporné. Přehled všech průmyslových revolucí je zachycen na Obr. 1 – Přehled průmyslových revolucí [3]. [3, 4, 5]



Obr. 1 – Přehled průmyslových revolucí [3]

Po třetí průmyslové revoluci nastává čtvrtá průmyslová revoluce, která probíhá v současné době a dle odhadů by měla probíhat ještě zhruba 10-30 let. [5]

1.2 Průmysl 4.0

Hlavní myšlenkou čtvrté průmyslové revoluce je snaha propojit jednotlivé prvky do sítě. V této síti spolu budou jednotlivé komponenty navzájem komunikovat, vyměňovat si informace, a to jak o produktech, tak o sobě samých, a ještě více tvořit jeden celek. Vztaheno k chytré továrně by takový celek v ideálním případě dokázal na základě zákaznickových objednávek projít sám veškerými procesy od objednávek zboží na základě stavu zásob, přes bezchybný výrobní proces až po naskladnění hotových výrobků do expedičního skladu nebo rovnou doručení zákazníkovi. Takový celek je v současné době pouze vizí, která by se ale dle některých předních průkopníků těchto myšlenek mohla v brzké době stát skutečností. [6]

Taková vize stojí na několika základních pilířích, které jsou stále více implementovány do různých výrobních i nevýrobních odvětví. Jako základní je možné považovat tyto:

- IoT (Internet of Things)
- AI (Artificial Intelligence)
- Digitální dvojče
- Simulace

IoT

Internet of Things, v překladu Internet věcí, označuje ekosystém počítačů a chytrých zařízení či strojů, které spolu mohou vzájemně komunikovat či spolupracovat bez nutné asistence člověka. V podstatě se jedná o běžná elektronická zařízení, jako je například lednička či televize, která přidáním operačního systému (dále jen OS) a síťové konektivity získávají zcela nová využití. Taková zařízení, díky svým senzorům a vnitřní logice, dokážou data získaná sledováním okolního prostředí či sebe samotných přetvářet v informace, na jejichž základě jsou prováděny činnosti, které hodnota těchto informací ovlivňuje. Nejznámějším příkladem jsou v současné době chytré hodinky, které dokážou získávat data o zdravotním stavu jejich nositele, tyto data vyhodnocovat a v případě nutnosti informovat jejich nositele o změně takových dat. [7, 8]

AI

Artificial Intelligence, v překladu umělá inteligence, je schopnost strojů napodobovat lidské schopnosti, jako je uvažování, učení se, plánování nebo kreativita. Taková inteligence je tak schopna reagovat na podněty z vnějšího okolí, přizpůsobovat jim své chování a učit se z minulosti. Umělou inteligenci je možné rozdělit do dvou základních skupin – softwarovou, což je druh inteligence, který je přítomen pouze v softwarové podobě, kde výstupem činnosti této inteligence jsou data či informace. Nejznámější zástupci této skupiny jsou virtuální asistenti, vyhledávače, nebo systémy pro rozpoznání řeči, nebo obličejů. Druhou skupinou je umělá inteligence zabudovaná v nějakém zařízení. Ta je rovněž přítomna pouze jako software, ale uvnitř nějakého fyzického zařízení, které nadále kromě datového či informačního výstupu provádí činnost, kterou je možno pocítit či pozorovat. Do této skupiny patří IoT, autonomní automobily či drony apod. [9]

Digitální dvojče

Digitální transformace umožňuje podnikům vytvářet digitální dvojče, které je virtuální replikou fyzického, skutečného objektu nebo systému. Digitální dvojče je konstruováno tak,

aby bylo schopno získávat data ze svého reálného, fyzického modelu. Sběr dat je většinou realizován pomocí různých čidel nebo senzorů. Digitální dvojče může být také navrženo na základě prototypu fyzického dílu, nebo může být i samo prototypem, ještě před vznikem prvního fyzického dílu. Tato technologie umožňuje využití v mnoha odvětvích. Jedním z nich je například zdravotnictví, kde díky sběru dat pomocí senzorů umístěných na těle subjektu dochází k predikci zdravotního stavu pacienta. Dále pak lze tuto technologii využít například v automobilovém průmyslu nebo výrobě. [10]

Simulace

Simulaci lze definovat jako napodobení skutečného (většinou značně komplexního a složitého) stavu, věci, nebo procesu. Při následném experimentování se simulačním modelem lze pozorovat jeho chování, což umožňuje zjistit jeho vlastnosti, silné a slabé stránky a vady. Optimalizací takového simulačního modelu je možno dojít k ideálnímu stavu, který pak může být převeden do reálné podoby. Jako příklad využití simulací v praxi lze uvést sledování využití kapacit a zdrojů, identifikaci kritických míst u silničních komunikací, nebo sledování doby trvání jednotlivých procesů/cyklů. [11]

1.3 Model Fischertechnik a Průmysl 4.0

Model Fischertechnik, pro který bude mobilní aplikace navržena, představuje model výrobní linky ve výrobním podniku, který má implementovány určité prvky Průmyslu 4.0. Linka je totiž sama schopna provést obráběný díl celým výrobním procesem od vyskladnění dílu ze skladu až po vyřazení hotového výrobku z linky. V průběhu výroby je rovněž schopna sama detekovat, o jaký druh výrobku se jedná a jestli je jeho kvalita vyhovující. Mobilní aplikaci, která propojí model Fischertechnik s uživatelem mobilní aplikace, lze vnímat jako podporu využívanou např. manažerem ve výrobním podniku, což podporuje koncept Průmyslu 4.0.

2 Mobilní aplikace a jejich vývoj

Mobilní telefon je vybaven celou řadou aplikací, a to buď přímo od výrobce mobilního telefonu, nebo od vývojářů OS, kterým je většina dnešních telefonů vybavena. Mobilní aplikace je ve své podstatě software určený pro mobilní telefon či tablet. Obvykle disponuje intuitivním uživatelským rozhraním, přístupností offline a rychlou odezvou. [12]

Mezi ty nejjednodušší známé aplikace, často dodávané právě výrobcem mobilního telefonu, nebo vývojářem OS patří poznámkový blok, kalkulačka nebo kalendář. Mezi ty složitější pak patří různé bankovní aplikace (např. ČSOB Smart), aplikace pro sociální sítě (např. LinkedIn, Facebook), nebo hry.

V následujících podkapitolách budou nejprve představeny druhy mobilních aplikací. Dále budou popsány nejpoužívanější OS pro mobilní telefony a výběr cílového OS pro tvorbu mobilní aplikace. V závěru této kapitoly budou představeny strategie, které je možné využít při tvorbě mobilní aplikace a jednotlivé fáze vývoje.

2.1 Druhy mobilních aplikací

Mobilní aplikace je možné dělit podle vícero kritérií. Těmi mohou být například účel (hry, sociální, utility), cílová skupina uživatelů (firemní, komerční), cena za použití (zdarma, placené) apod. Kritérií může být celá řada, avšak většina z nich je zohledňována především při komerčním vývoji. Pro mobilní aplikace pro zobrazení stavu Fischertechnik modelu nejsou tato kritéria považována jako stěžejní, a proto nebudou v této práci dále popisována. Kritérium, které ale může být označeno za stěžejní, je způsob technického zpracování. [13, 14, 15]

Technické zpracování má vliv na celkovou strukturu a využitelnost aplikace. V současné době lze mobilní aplikace dle druhu technického zpracování rozdělit na tyto: [13, 14, 15]

- Nativní
- Webové
- Hybridní

Každá z nich má své charakteristické rysy, výhody a nevýhody. Následující odstavce se budou věnovat popisu jednotlivých variant, jejich porovnání a výběru vhodné varianty v případě tvorby mobilní aplikace pro zobrazení stavu Fischertechnik modelu.

2.1.1 Nativní

Nativní mobilní aplikace jsou určeny a vyvíjeny pouze pro jednu konkrétní platformu. Platformou se v tomto případě rozumí OS mobilního telefonu. Mezi ty nejnámější a v současné době nejpoužívanější patří Android, iOS a Windows Phone. Takovou aplikaci je pak možné spustit pouze na jedné platformě. Pro spuštění na jiné platformě musí být aplikace přeprogramována. Jelikož nativní mobilní aplikace jsou účelně vytvářeny pro konkrétní platformu, umožňuje to vývojářům využití všech možností, která tato platforma nabízí. Tyto aplikace tak zpravidla poskytují nejlepší „user experience“ (volně přeloženo jako „uživatelský dojem/zážitek z užívání aplikace“), disponují největším výkonem a jednoduchostí užívání. Naopak, jelikož je potřeba pro každou platformu vyvíjet mobilní aplikaci zvlášť, jsou často finančně i časově náročnější než ostatní varianty. Jako příklad takové aplikace je možné uvést chatovou službu WhatsApp nebo navigační aplikaci Waze. [13, 14, 15, 16]

2.1.2 Webové

Webové aplikace nejsou na rozdíl od nativních vytvářeny pro konkrétní platformu. Jsou totiž vytvářeny tak, aby byly spustitelné v prohlížeči. Stačí tak pouze vývoj jedné aplikace, která bude následně spustitelná například na mobilním zařízení s OS Android, iOS nebo na počítači s OS Windows či Mac OS. Díky tomu je vývoj rychlejší a aplikace se tak může dříve dostat na trh. Výhodou je i to, že není nutné mobilní aplikaci stahovat ani instalovat, čímž dochází k úspoře paměti zařízení. Zároveň je díky uložení aplikace na serveru zaručena její aktuálnost na všech zařízeních. Nevýhodou takové aplikace je naopak závislost na internetovém připojení, bez kterého není aplikaci možné využívat. Dále pak skutečnost, že každé zařízení může využívat různé prohlížeče, mezi kterými mohou být určité rozdíly. To může způsobovat nestabilitu a vyžadovat častější aktualizace vzhledem k vývoji prohlížečů. V neposlední řadě může v některých případech negativně působit i limitovanost přístupu prohlížečů k některým funkcionalitám daného zařízení. Jako příklad takové aplikace je možné uvést streamovací službu Netflix, nebo aplikaci pro správu projektů Trello. [13, 14, 15, 17]

2.1.3 Hybridní

Hybridní aplikace je kombinací nativní a webové. Princip fungování je takový, že mobilní aplikace je nejprve nainstalována na dané zařízení a poté je v této aplikaci zobrazován webový obsah. Vývoj takových aplikací je rychlejší a levnější než nativní vývoj, není k jejich fungování potřeba prohlížeč a je možné je spustit na více platformách. Nevýhodou je opět možná nekompatibilita v důsledku multiplatformní aplikace, horší user experience než u nativních aplikací a vyšší cena než u webových aplikací. Jako příklad lze uvést sociální služby Instagram a Twitter. [13, 14, 15, 18]

Porovnání druhů technických zpracování

Každý druh technického zpracování má své určité pro a proti, které umožňují vytvoření představy o tom, kdy je vhodné, jakou variantu využít. Ty základní jsou shrnuty a porovnány v Tab. 1 - Porovnání technických zpracování mobilních aplikací [13, 14, 15, 16, 17, 18].

V případě tvorby mobilní aplikace pro zobrazení stavu Fischertechnik modelu byl zvolen nativní způsob. Volba nativní aplikace je především z důvodu možnosti využití veškerých funkcionalit, které dané zařízení nabízí, nezávislosti na připojení k internetu a také kvůli nevyššímu výkonu a nejlepší user experience z výše uvedených variant.

Tab. 1 - Porovnání technických zpracování mobilních aplikací [13, 14, 15, 16, 17, 18]

	Mobilní aplikace		
	Nativní	Webová	Hybridní
Multiplatformní	X	✓	✓
Nutná instalace	✓	X	✓
Časově náročný vývoj	✓	X	X
Závislé na prohlížeči	X	✓	X
Závislé na připojení k internetu	X	✓	X

2.2 Cílový operační systém

Při vývoji mobilní aplikace je nezbytné myslet na to, pro jaký OS je mobilní aplikace vytvářena. Těch je v současné době celá řada. Celosvětový trh je však z 99,3 % obsazen pouze dvěma - iOS a Android. Tato kapitola se bude věnovat krátkému představení těchto dvou OS a následnou volbou pro který z nich bude mobilní aplikace pro zobrazení stavu Fischertechnik modelu vytvořena. Ostatní OS zabírají na trhu tak malý podíl, že není považováno za podstatné se dále zabývat jejich rozbořem. [19]

Android

OS Android je založen na jádře Linuxu a je dostupný jako open source (otevřený systém). Nejčastěji je využíván v mobilních telefonech, tabletech a dalších chytrých zařízeních jako jsou například hodinky či televize. Jeho zastoupení na celosvětovém trhu je 71,6 % a je tak nejpoužívanějším mobilním OS. Na rozdíl od iOS je Android využíván na zařízeních od vícero výrobců (např. Xiaomi, Samsung, Realme). [19, 20]

iOS

Operační systém iOS byl vyvinut společností Apple a je využíván pouze na zařízeních od tohoto výrobce. Těmi jsou telefony, tablety, hudební přehrávače a televize. Na rozdíl od Androidu se jedná o closed source, což znamená, že zdrojový kód není veřejně dostupný a nelze ho upravovat a následně distribuovat. Na celosvětovém trhu mobilních telefonů se nachází v 27,68 % zařízeních a je tak druhým nejpoužívanějším OS pro mobilní telefony. [19, 21]

Volba cílového operačního systému

Mobilní aplikace pro zobrazení stavu Fischertechnik modelu bude vyvíjena pro OS Android. Hlavním důvodem výběru je procentuálně vyšší zastoupení tohoto OS na celosvětovém trhu a tím i možný větší dosah této aplikace. Dalším důvodem je možnost vývoje mobilních aplikací pro Android téměř nezávisle na OS zařízení, na kterém je vývoj prováděn. Android aplikace je totiž možné vyvíjet a testovat na obou OS, které mají celosvětově největší zastoupení – Windows (74,83 %) a Mac OS (15,37 %). Naopak vývoj aplikací pro iOS je nezbytné provádět pouze na zařízeních s Mac OS. [19, 22]

2.3 Strategie vývoje

K vývoji mobilních aplikací lze mimo jiné přistupovat dvěma hlavními způsoby, a to buď vodopádovým nebo agilním (spirálovým). Tyto způsoby se nepoužívají pouze u programování, ale v jakékoliv oblasti při práci vícečlenného týmu (nebo více týmů) na jednom projektu. Jejich význam lze odvodit od jejich názvu. U vodopádového stylu se mezi jednotlivými fázemi postupuje v daném pořadí a k předchozím fázím se nevrací, pokud to není nevyhnutelné. Simuluje to tedy vodopád, kdy voda padá pouze shora dolů, nahoru se nevrací. Agilní, též spirálový přístup, je naopak přístup, při němž se předpokládají kratší a menší kroky. Mezi jednotlivými fázemi se dá přecházet a tím i revidovat a měnit jednotlivé úkony. [23]

2.4 Fáze vývoje

Vývoj mobilních aplikací lze rozdělit do několika fází: [24, 25]

- Studie proveditelnosti
- Analýza požadavků
- Návrh designu (grafického prostředí)
- Programování
- Testování/uvvedení do provozu
- Podpora/údržba

Studie proveditelnosti

Studie proveditelnosti bývá první fází každého projektu. Jejím hlavním cílem je zodpovězení základních otázek, mezi které patří například: [25]

- Jsme schopni dosáhnout požadovaného výsledku v požadované kvalitě v požadovaný čas?
- Máme dostatečné vybavení a lidské zdroje k realizaci tohoto projektu?
- Je vhodnější realizovat projekt pouze s vlastními zdroji a možnostmi nebo je vhodnější využít externí zdroje?
- Není pro nás realizace projektu příliš riskantní?
- Máme aktuálně dostupné časové zdroje pro realizaci projektu?
- Bude pro nás mít projekt i jiné přínosy než ty finanční?

Na základě odpovědí na výše zmíněné otázky dochází k rozhodnutí, zda je vhodné projekt realizovat či nikoliv. Pro tvorbu mobilní aplikace pro zobrazení stavu Fischertechnik modelu není tato fáze projektu považována za klíčovou, a proto bude v samotné fázi vývoje mobilní aplikace vynechána.

Analýza požadavků

Analýza požadavků je klíčovou fází celého procesu vývoje mobilní aplikace. V této fázi jsou totiž navrženy veškeré funkcionality, kterými by aplikace měla disponovat. Zejména u vodopádového přístupu je nutné této fázi věnovat zvláštní pozornost. [24, 25]

Návrh designu

Fáze návrhu designu aplikace navazuje na fázi analýzy. Předpokládá se, že jsou známy veškeré funkcionality, které by měla mobilní aplikace obsahovat, a je tak možné navrhnout design aplikace – grafické uživatelské rozhraní (dále jen GUI). [24, 25]

Programování

Během fáze programování se píše program (kód) samotný. Jedná se o soubor příkazů zadané uživatelem (vývojářem), které slouží jako pokyny k vykonání pro počítač. Kód lze rozdělit na 2 druhy, frontend a backend. Rozdíl mezi nimi je ten, že ve frontend kódu se programuje GUI, tedy to, co uživatel uvidí. Naopak backend je kód, který dává grafickému prostředí „život“. Programují se v něm jednotlivé funkcionality, propojení apod. [24, 25]

Testování

Ve fázi testování probíhá ověření, zda je program schopný vykonávat požadované úlohy. V tomto případě bude testování probíhat na výrobním modelu Fischertechnik. Zde budou ověřeny veškeré požadované funkce a v případě jejich nedostatku bude provedena úprava kódu. [24, 25]

Uvedení do provozu

Během fáze uvedení do provozu dochází ke zpřístupnění produktu zadavateli, resp. uvedení produktu do běžného provozu. V této fázi je velice důležitý sběr dat a zpětných vazeb od uživatelů aplikace. Je totiž možné, že se během fáze testování nepodařilo odchytnit nějakou chybu. Zároveň se také stává, že uživatelům aplikace chybí nějaká funkcionality, která byla při analýze požadavků opomenuta. Na základě této zpětné vazby je pak vhodné aplikaci upravit. [24, 25]

Podpora/údržba

Ve fázi podpory/údržby jsou zapracovány podněty vzniklé v předchozí fázi. Jsou opraveny chyby, které uživatelé objevili během užívání aplikace. Zároveň jsou přidány funkcionality, které uživatelům chyběly a byly zadavatelem projektu shledány jako relevantní. Táto fáze má v běžné praxi dobu trvání dle dohody mezi zadavatelem projektu a realizátorem projektu. [24, 25]

3 Technologie pro vývoj mobilních aplikací

Na začátku vývoje každé mobilní aplikace je nutné zamyslet se nad tím, jaké zvolit technologie, které budou při vývoji mobilní aplikace využity. Pod pojmem technologie se v případě vývoje mobilních aplikací skrývá především framework, od kterého se odvíjí programovací jazyk, IDE (Integrated Development Environment), nároky na software a hardware apod. Tato kapitola je zaměřena na představení nejpobulárnějších frameworků používaných pro vývoj mobilních aplikací v současné době, výčet jejich výhod a nevýhod a porovnání. Kapitola dále pokračuje volbou vhodného frameworku pro tuto práci a představením programovacího jazyka, který je využit ve zvoleném frameworku.

3.1 Frameworky

Pojem framework v tomto případě představuje softwarovou strukturu, která přebírá řešení základních problémů a udává směr, kterým by se měl postup vývoje řídit. Framework poskytuje hotová řešení, která jsou vytvořena za účelem zrychlení vývoje. Nejčastěji obsahuje kompilátory, knihovny, API (Application Programming Interface) a různé podpůrné programy. V oblasti vývoje mobilních aplikací lze frameworky rozdělit na frameworky určené pro nativní vývoj a frameworky určené pro multiplatformní vývoj. Rozdíl mezi těmito druhy frameworků bude popsán v následující odstavcích, včetně výčtu jejich výhod a nevýhod. [26]

Frameworky pro nativní vývoj

Frameworky pro nativní vývoj jsou specifické tím, že umožňují vývoj mobilní aplikace pouze pro jednu konkrétní platformu, kde pod pojmem platforma se v tomto případě rozumí OS mobilního telefonu. To má za důsledek to, že pro vývoj mobilní aplikace, kterou bude možné spustit na mobilním telefonu s OS Android a iOS, je potřeba vytvořit dva unikátní projekty, jeden pro každý OS. Takové projekty jsou pak většinou tvořeny v různých programovacích jazycích s využitím různých IDE a je tak potřeba ovládat obě tyto technologie a doplňky s nimi spojené. Takový vývoj bývá často finančně i časově náročný. Existují proto frameworky, které vývoj pro více platforem zároveň značně usnadňují – frameworky pro multiplatformní vývoj. [27, 28]

Frameworky pro multiplatformní vývoj

Multiplatformní vývoj umožňuje vývoj pro více platforem najednou. Při vývoji mobilní aplikace pro dva a více OS je tak vytvořen pouze jeden projekt, který umožňuje využití jednoho zdrojového kódu dvěma či více aplikacemi, kde každá je určena pro jiný OS. Vytvořený kód však nikdy není 100 % využitelný pro oba OS. Vznikají zde totiž rozdíly v architektuře jednotlivých OS a zařízení a v GUI. Mobilní aplikace se tak skládají z přibližně 60-95 % sdíleného (stejného) kódu a zbylých 5-40 % je pro každou platformu doprogramován samostatně. [27, 28]

Porovnání nativního a multiplatformního vývoje

Obě z výše zmíněných platforem mají své výhody a nevýhody, které určují, kdy je která varianta vhodnější. Hlavní výhodou při vývoji nativních aplikací je lepší využitelnost funkcionalit, které daná platforma nabízí. Při multiplatformním vývoji je totiž při psaní kódu potřeba dbát na skutečnost, že přístup k některým funkcionalitám může být na různých platformách značně odlišný. Další výhodou nativního vývoje je vyšší výkon a rychlejší odezva než u multiplatformních aplikací. Nevýhodami jsou naopak delší vývoj kvůli nemožnosti využití kódu na více platformách a vyšší náklady na vývoj a údržbu.

U multiplatformních aplikací spočívají hlavní výhody v rychlosti vývoje, nižších nákladech, snazší údržbě, kdy je většinou potřeba upravit pouze jeden kód společný pro všechny platformy, namísto kódu pro každou platformu zvlášť. Naopak nevýhodou je potřeba detekce odlišností v jednotlivých platformách a přizpůsobení vývoje těmto odlišnostem a nižší výkon mobilní aplikace. Přehled všech výhod a nevýhod je uveden níže v Tab. 2 - Porovnání nativního a multiplatformního vývoje [26, 27, 28].

Tab. 2 - Porovnání nativního a multiplatformního vývoje [26, 27, 28]

		Vývoj	
		Nativní	Multiplatformní
+	Lepší využitelnost funkcionalit		Rychlejší vývoj
	Vyšší výkon aplikace		Levnější vývoj
	Zvýšené zabezpečení		Snazší údržba
-	Pomalejší vývoj		Nižší výkon aplikace
	Potřeba širšího spektra znalostí		Přizpůsobení vývoje odlišnostem jednotlivých platform
	Vyšší náklady na vývoj a údržbu		

Při pohledu na jednotlivé výhody a nevýhody nativního a multiplatformního vývoje lze vyvodit závěr, že multiplatformní vývoj se hodí pro vývoj mobilních aplikací určených pro více platform, kde je důraz kladen na vysokou rychlost a nízkou finanční náročnost vývoje. Naopak nativní vývoj je vhodný především pro aplikace, které chtějí uživateli umožnit nejlepší možný výkon aplikace a celkový dojem z aplikace s nižším ohledem na rychlost vývoje a náklady s ním spojené. [27, 28]

3.1.1 Dostupné frameworky

Předtím, než bude zahájen výběr frameworku, ve kterém bude mobilní aplikace pro zobrazení stavu Fischertechnik modelu vytvořena, je potřeba rozhodnout, zda se bude jednat o framework pro nativní nebo multiplatformní vývoj.

Ačkoliv v rámci této práce bude vytvářena mobilní aplikace pouze pro OS Android, předpokládá se v budoucnu rozšíření i na OS iOS. Zvolením multiplatformního frameworku bude následný vývoj pro iOS mnohem méně časově náročný. Následující odstavce se proto budou zabývat představením nejpoužívanějších multiplatformních frameworků na jehož konci proběhne hodnocení a zvolení nejvhodnějšího z nich.

Dle článků z internetových stránek blocshop.io [29] a medium.com [30] hodnotících multiplatformní frameworky pro vývoj mobilních aplikací se na prvních místech v různém

pořadí umisťují frameworky React Native, Flutter a Xamarin. Následující odstavce se proto budou věnovat právě jim.

React Native

React Native je framework vyvinutý v roce 2015 společností Meta Platforms, který umožňuje tvorbu aplikací pro Android, Android TV, iOS, Mac OS, tvOS, web, Windows a UWP (Universal Windows Platform). Tento framework je možné využívat ve vícero IDE, například Atom, Nuclide nebo Visual Studio Code. Díky tomu, že je React Native zcela zdarma i pro komerční účely a že využívá velmi populární programovací jazyk JavaScript, je ze všech tří frameworků nejpoužívanější. To má za následek velkou komunitní základnu, se kterou je možné komunikovat a radit se nad různými problémy, které v průběhu vývoje mobilních aplikací vznikají. [31, 32, 33]

Flutter

Flutter je framework vyvinutý v roce 2017 společností Google, který umožňuje tvorbu aplikací pro Android, iOS, Windows, MacOS, Linux, Google Fuchsia a web. Tento framework je rovněž možné využívat ve vícero IDE, kde nejpoužívanější jsou Visual Studio Code, Android Studio a IntelliJ IDEA. Flutter využívá programovací jazyk Dart a v posledních dvou letech je mezi frameworky pro tvorbu mobilních aplikací stále více populární. Jeho využití je rovněž zcela zdarma. [31, 32, 33]




Xamarin

Xamarin je framework vyvinutý v roce 2011 a je postaven na projektu zvaném Mono. V roce 2016 byl tento projekt odkoupen společností Microsoft, která jej spravuje dodnes. Xamarin umožňuje tvorbu aplikací pro Android, iOS a UWP. Tento framework je nejčastěji využíván v IDE Microsoft Visual Studio, ale je možné ho využít i v jiných IDE, například Rider. Xamarin využívá programovací jazyk C# a jeho použití je bezplatné pouze pokud se vývojářský tým neskládá z více než 5 členů, v opačném případě je nutné zakoupit licenci. [31, 32, 33]

3.1.2 Porovnání frameworků a výběr vhodné varianty

Porovnání konkrétních frameworků a vybrání nejvhodnějšího není tak jednoduché, jako v případě porovnání a výběru mezi nativním a multiplatformním vývojem. Obecně totiž nelze říci, jaké hodnotící kritérium je pro porovnání zásadní. Různé frameworky jsou totiž často v mnoha ohledech podobné a liší se jen v drobnostech. Záleží pak na konkrétním uživateli, jaký preferuje programovací jazyk, IDE nebo pro jakou platformu je daná mobilní aplikace určena. Ze shromážděných dat byla vytvořena Tab. 3 - Porovnání frameworků React Native, Flutter a Xamarin [29, 30, 31, 32, 33], která alespoň částečně zachycuje přehled vlastností, které tyto frameworky navzájem odlišují.

Tab. 3 - Porovnání frameworků React Native, Flutter a Xamarin [29, 30, 31, 32, 33]

	Framework		
	 React Native	 Flutter	 Xamarin
Programovací jazyk	JavaScript	Dart	C#
Množství sdíleného kódu mezi jednotlivými platformami	90 %	50-90 %	96 %
Náklady	Žádné	Žádné	Žádné / potřeba zakoupení licence pro tým o více než 5 členech

Pro tvorbu mobilní aplikace pro zobrazení stavu Fischertechnik modelu byl zvolen framework Xamarin. Ten byl vybrán ze dvou důvodů. Tím prvním je ten, že Xamarin umožňuje největší sdílení kódu mezi jednotlivými platformami. Tím druhým je skutečnost, že je při vývoji mobilních aplikací pomocí tohoto frameworku je využit programovací jazyk C#, který je v hojné míře vyučován na katedře Průmyslového inženýrství a managementu na Fakultě strojní Západočeské univerzity v Plzni. Lze tak předpokládat, že student či zaměstnanec, který bude v pozdějším stádiu vývoje vytvářet mobilní aplikaci i pro platformu iOS nebo přidávat nové funkcionality, bude s největší pravděpodobností disponovat právě znalostí jazyka C#. [31, 32, 33]

3.2 Programovací jazyky, jazyk C#

Tato kapitola je zaměřena na obecné představení programovacích jazyků a bližšího představení jazyka C#, který byl zvolen v kapitole *Porovnání frameworků a výběr vhodné varianty* pro vývoj mobilní aplikace zobrazující stav Fischertechnik modelu. V kapitole bude nejprve stručně popsáno, co je programovací jazyk a jaké je jeho základní dělení. Dále bude krátce popsán programovací jazyk C#.

Programovací jazyk

Programovací jazyk představuje prostředek pro zápis příkazů, instrukcí a jiných syntaxí za účelem tvorby programu, které pak počítač zpracovává. Lze ho tedy chápat jako komunikační nástroj mezi programátorem a počítačem. Programovací jazyky lze dělit mnoha způsoby, za základní rozdělení je možné považovat tyto: [34, 35]

- Nižší a vyšší
- Interpretované a kompilované

Jazyky nižší a vyšší

Rozdíl mezi nižším a vyšším jazykem je především ve srozumitelnosti pro člověka. Nižší programovací jazyk je tvořen příkazy, které nejsou pro člověka příliš srozumitelné (např. Assembler). Často se jedná o směs znaků a čísel, ze kterých není na první pohled patrné, co je úkolem daného programu. Uživatel tak musí disponovat znalostí daného hardware, který se ale kus od kusu liší. Jednoduchý program napsaný v nižším programovacím jazyce by mohl vypadat například takto: [34, 35]

```
2104
1105
3106
7001
0053
FFFE
0000
```

Naopak vyšší programovací jazyky dávají na první pohled představu o tom, co se od programu vyžaduje (např. Java, Basic, PHP, Python). Jsou tvořeny příkazy, které svou logikou a zápisem připomínají situace z reálného světa. Jednoduchý program napsaný ve vyšším programovacím jazyce by mohl vypadat například takto: [34, 35]

```
int main(void)
{
    int a, b, c;
    a = 83;
    b = -2;
    c = a + b;
    return 0;
}
```

Další rozdíl mezi těmito jazyky je v jejich přenositelnosti. Zatímco vyšší programovací jazyk není závislý na strojových principech počítače, nižší programovací jazyk je závislý na procesoru a není přenositelný na jiný procesor. [34, 35]

Jazyky interpretované a kompilované

Programovací jazyk lze rozdělit podle způsobu překladu kódu a spuštění programu na dva hlavní – interpretovaný programovací jazyk a kompilovaný programovací jazyk. Rozdíl mezi nimi je v tom, že kompilovaný jazyk celý kód nejprve přeloží pomocí kompilátoru, který kód přeloží do strojového kódu a až poté je program možné spustit. Oproti tomu program napsaný pomocí interpretovaného jazyka se překládá až při běhu programu. Mezi kompilované programovací jazyky je možné zařadit například Javu, Pascal nebo C++. Jejich výhodou je především v tom, že zkompilovaný jazyk je rychlejší, připravený ke spuštění a zdrojový kód není veřejný. Naopak nevýhodou spočívá v tom, že jazyk je závislý na platformě. Zkompilovaný program tak není možné spustit na jiné platformě, aniž by byl na této platformě zkompilován.

U interpretovaných jazyků, jako je například JavaScript nebo Python, dochází ke kompilaci až při běhu programu. Díky tomu je kód přenositelný na vícero platform a je snáze testovatelný. Nevýhodou je pomalost oproti kompilovaným jazykům a veřejnost zdrojového kódu. [34, 35, 36]

Programovací jazyk C#

Jazyk C# je programovací jazyk vyvinutý společností Microsoft v roce 2000. V průběhu let jsou postupně uvolňovány další verze tohoto jazyka. Jedná se o kompilovaný programovací jazyk vyšší úrovně, který je využíván především pro tvorbu her a desktopových, mobilních a webových aplikací. Nejčastěji je programovací jazyk C# využíván na platformě .NET, která poskytuje širokou škálu nástrojů a knihoven usnadňujících vývoj aplikací. Jazyk C# patří do skupiny objektově orientovaných programovacích jazyků. Pro ty je mimo jiné charakteristická tvorba tříd a metod, dědičnost nebo polymorfismus. Zároveň lze jazyk C# zařadit do skupiny silně typovaných programovacích jazyků, které se vyznačují tím, že je při nich nutné používat předem vymezené datové typy. V současné době je jazyk C# mezi pěti nejpopulárnějšími programovacími jazyky. [37, 38]

4 Práce s databázemi

Jelikož mobilní aplikace bude data pro zobrazení stavu Fischertechnik modelu získávat z databáze, je vhodné vysvětlit, co se pod tímto pojmem míní a jak lze s takovou databází zacházet.

Databáze je systém obsahující data, která mají pevně danou strukturu a které mají mezi sebou definované vztahy. Software, umožňující k nim přistupovat se nazývá Systém Řízení Báze Dat (dále jen SRBD). Za klíčové vlastnosti tohoto softwaru lze považovat: [39]

- Definice dat
- Údržbu dat
- Manipulaci s daty
- Zobrazování dat
- Integritu dat

V případě modelu Fischertechnik je využíván SRBD Microsoft SQL (Structured Query Language) Server.

Návrh a tvorba databáze

Pro návrh a tvorbu databáze je důležité znát několik základních pojmů. Pro databázi propojující mobilní aplikaci s modelem Fischertechnik lze definovat tyto základní prvky: [39]

- Databázový model
- Entita
- Atribut
- Záznam
- Primární a cizí klíč
- Relace

Databázový model

Pojem databázový model rozděluje databáze podle toho, jakou strukturu mají data uložená v databázi. V současné době je nejpoužívanějším modelem relační databáze, která je založena na relacích mezi tabulkami. Ty mezi sebou mohou sdílet jednotlivé záznamy díky klíčům a relacím. [40]

Entita

Entita je objekt, jehož vlastnosti jsou v databázi uloženy. Každá entita musí mít znatelnou diferenci od ostatních entit a musí existovat nezávisle na ostatních entitách. Entitou může být jakýkoliv objekt, jako příklad je možné uvést entitu „Žadatel“. [40]

Atribut

Pod pojmem atribut se myslí vlastnost daného objektu – entity. Pro výše zmíněný příklad entity „Žadatel“ je možné navrhnout například tyto atributy: Jméno, Příjmení, Věk, Zaměstnán. Atributy tedy umožňují definovat to, o jakých vlastnostech budou pro danou entitu vedeny záznamy. Pro každý atribut je před používáním databáze nutno definovat datový typ. Datový typ představuje obor hodnot, které smí tento atribut nabývat. V jazyce SQL (a nejen v něm), na němž je Microsoft SQL Server postaven (dále jen MSSQL Server), se rozlišuje velké množství datových typů. Pro tuto práci jsou za klíčové považovány tyto datové typy: [39, 40, 41]

- Int – Jedná se o datový typ, který umožňuje ukládání celých čísel, např.: 1, 85723, 43
- Float – Je podobný datovému typu int ale na rozdíl od něj umožňuje ukládání i desetinných čísel, např.: 0.432, 74329.89
- Bool (Boolean) – Jedná se o datový typ, který může nabývat pouze 2 hodnoty opačného významu: 0 a 1, kde 1 je chápána jako true a označuje hodnotu daného záznamu jako pravdivou, 0 je chápána přesně naopak
- Char(n) – Datový typ char je takový datový typ, který umožňuje uchovávat hodnoty typu znak a je využit pro práci s textem. Písmeno „n“ v závorce definuje jeho maximální velikost. Pro n=50 je tedy maximální délka záznamu 50 znaků. Příklad hodnoty datového typu char: „Západočeská univerzita v Plzni, Fakulta strojní“

Záznam

Jednotlivé řádky s hodnotami se nazývají záznamy. Záznam pro výše uvedené atributy „Jméno“ a „Věk“ pro entitu „Žadatel“ by mohly být například Jméno = „Karel“, Věk =“32“. [39, 40]

Primární a cizí klíč

V databázových systémech je často žádoucí, aby mezi sebou jednotlivé tabulky „komunikovaly“, tzn. aby se spojovaly záznamy mezi tabulkami. K tomu se využívají primární klíče, které se definují v jedné tabulce a cizí klíče, které se definují v druhé tabulce. Pro definování klíčů je nutné, aby záznamy daného atributu pro danou entitu byly jedinečné. Jako příklad lze uvést příjmení nebo věk, u kterých je vysoká pravděpodobnost, že v dané tabulce nebudou unikátní, tzn. pro danou entitu bude pod atributem příjmení nebo věk existovat více shodných záznamů. Naopak rodné číslo, nebo číslo občanského průkazu, je hodnota unikátní. Každý člověk má totiž od státu přiděleno právě jedno unikátní rodné číslo a jedno unikátní číslo občanského průkazu. [39, 40]

Relace

Relace představují typ vztahu, který mezi sebou jednotlivé tabulky mají. Ty existují celkem 4: [40, 42]

- 0 – žádný vztah
- 1:1 – jednomu záznamu v první tabulce odpovídá záznam z jiné tabulky
- 1:N – jednomu záznamu z první tabulky odpovídá N záznamů z jiné tabulky
- M:N – M záznamům v první tabulce odpovídá N záznamů z jiné tabulky

Vztah 1:1

Příkladem vztahu, kdy jednomu záznamu v první tabulce odpovídá záznam z druhé tabulky, může být člověk a jeho rodné číslo. Pro jednoho člověka existuje právě jedno rodné číslo a pro jedno rodné číslo je právě jeden člověk, jeden člověk tedy nemůže mít více rodných čísel a ani jedno rodné číslo nemůže být stejné pro více lidí. [42]

Vztah 1:N

Pro vztah 1:N, tedy vztah, kde jednomu záznamu z první tabulky odpovídá více záznamů z jiné tabulky, lze uvést příklad půjčovny kol. Jeden zákazník si může v daný okamžik půjčit více kol najednou, ale jedno kolo nemůže být v daný okamžik půjčeno více zákazníkům. [42]

Vztah M:N

Jako příklad pro vztah M:N, tedy vztah, kde M záznamům z první tabulky odpovídá N záznamů z druhé tabulky, lze uvést na příkladu tabulky objednávek a tabulky výrobků. V jedné objednávce totiž může být více výrobků a zároveň jeden výrobek může být ve více objednávkách. [42]

5 Aktuální stav

Ve snaze přiblížit studentům fungování výrobního podniku, který využívá technologie Průmyslu 4.0, byl vytvořen model Fischertechnik. Ten umožňuje pozorovat výrobu produktu na plně automatizované výrobní lince. Tato výrobní linka má však pouze demonstrativní charakter. Všechny operace mění stav výrobku, jako například svařování či vrtání, pouze zobrazují činnost, která na daném pracovišti probíhá a struktura a tvar výrobku tak zůstávají po celou dobu neměnné.

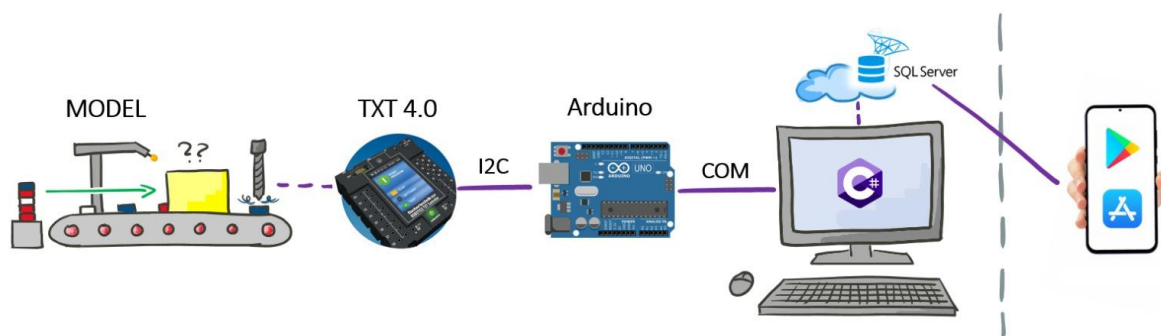
Tato kapitola je zaměřena na představení současného stavu modelu Fischertechnik, včetně popisu jednotlivých pracovišť, ze kterých se model skládá. V kapitole je rovněž popsán princip komunikace mezi mobilní aplikací a modelem Fischertechnik. V závěru kapitoly je představena již vytvořená databáze, do které jsou ukládány informace o Fischertechnik modelu.

5.1 Model Fischertechnik

Model Fischertechnik se skládá z 5 menších modelů, kde každý z nich představuje jedno z pracovišť ve výrobním podniku. Jedná se o model skladu, linky kontrolující kvalitu dílů, manipulačního robota, obráběcí linky a svářecí linky.

Komunikace mezi modelem Fischertechnik a mobilní aplikací

Princip komunikace a sběr dat, ze kterých bude mobilní aplikace čerpat informace, je zobrazen na Obr. 2 - Schéma propojení modelu s databází SQL [zdroj vlastní, 43, 44, 45, 46, 47, 48]. Jednotlivé pracoviště Fischertechnik modelu jsou propojeny kontrolérem TXT 4.0 od společnosti Fischertechnik. Tyto kontroléry jsou následně propojeny s jedním malým jednodeskovým počítačem Arduino pomocí I2C sběrnice. Následně je Arduino pomocí COM portu propojeno s počítačem, na kterém je vytvořena počítačová aplikace v jazyce C#, za jejíž pomoci jsou získaná data ukládána na SQL server. Z toho serveru bude mobilní aplikace data získávat a zobrazovat je.

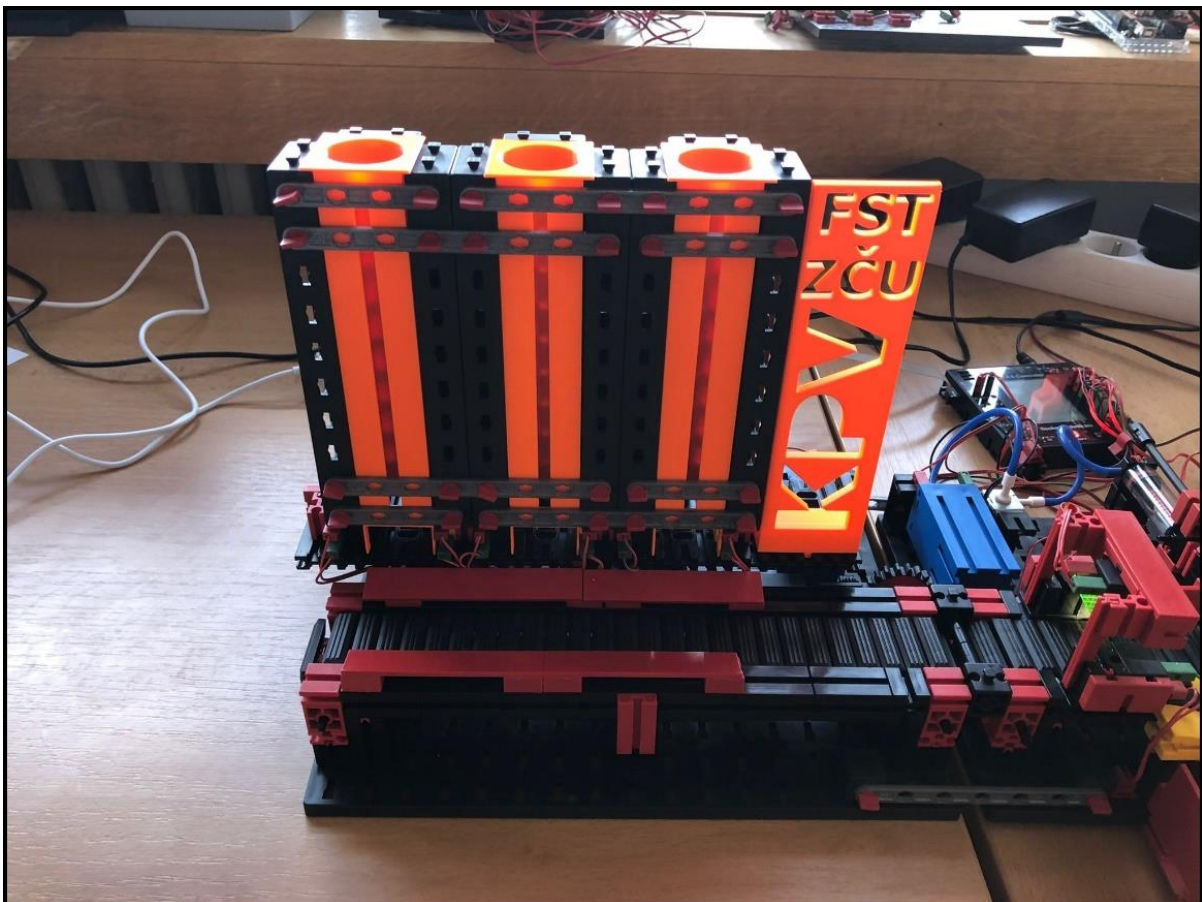


Obr. 2 - Schéma propojení modelu s databází SQL [zdroj vlastní, 43, 44, 45, 46, 47, 48]

Tento princip, jakým je v současné době model navržen, umožňuje pouze jednosměrnou komunikaci mezi modelem a databází. Vlivem toho je možné sbírat data a informace o Fischertechnik modelu a dále s nimi pracovat, není však možné zpracovávat data vložená do databáze modelem Fischertechnik. Tato vlastnost by dle plánů autorů tohoto modelu měla být v následujícím vývoji modelu pozměněna tak, aby byla umožněna obousměrná komunikace. V této práci se však bude vycházet ze současného stavu modelu Fischertechnik, tudíž i mobilní aplikace bude navržena pouze pro jednosměrnou komunikaci.

Sklad

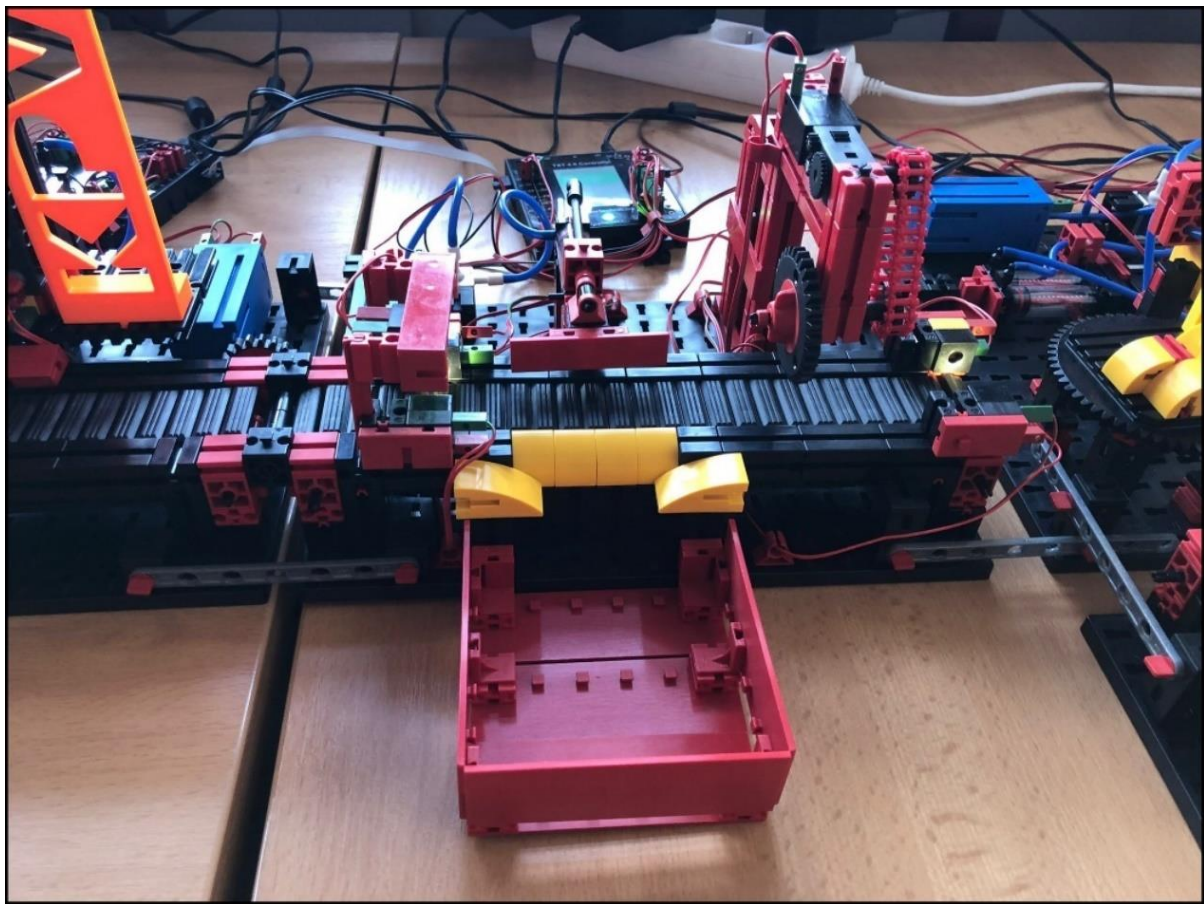
Prvním pracovištěm, které se na výrobní lince Fischertechnik modelu nachází, je sklad. Ten slouží jako vstupní brána pro materiál do celého výrobního procesu, kde materiálem jsou myšleny malé puky ve třech barevných variantách – modrá, červená a bílá. Tyto puky jsou ručně vloženy do jednoho ze tří zásobníků a pracoviště je automaticky odebírá na výrobní pás. Puky jsou ze zásobníků vyjímány v pořadí, v jakém jsou umístěny zásobníky. Nejprve dochází k vyjmutí všech puků z prvního zásobníku. Když je tento zásobník prázdný, pracoviště samo přistoupí k vyjímání dílů z druhého a následně třetího zásobníku až do té doby, dokud nejsou všechny zásobníky prázdné. Puky jsou po vyjmutí přepraveny pásovým dopravníkem na druhé pracoviště, kterým je kontrola kvality. Aktuální stav pracoviště je možné vidět na Obr. 3 - Fotografie pracoviště sklad.



Obr. 3 - Fotografie pracoviště sklad

Kontrola kvality

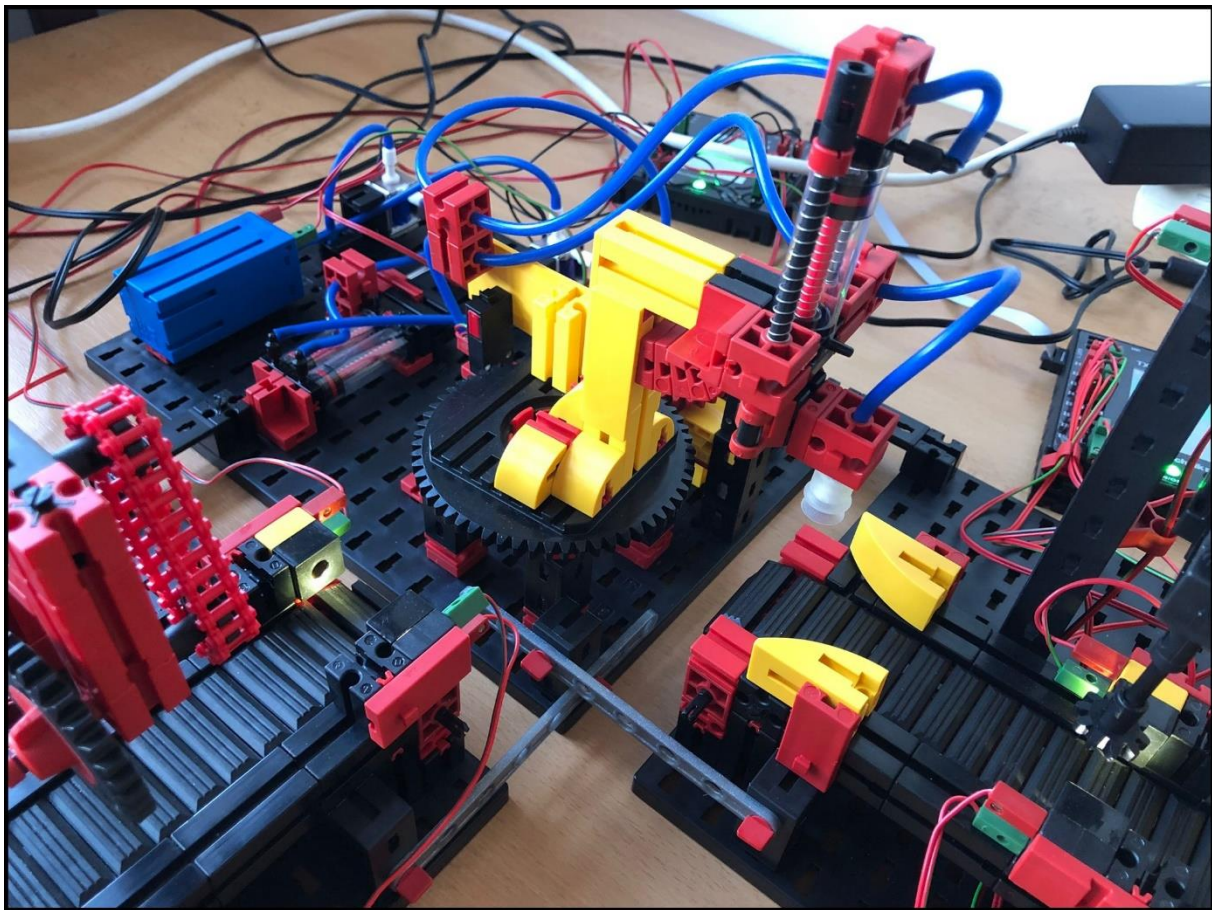
Z pracoviště sklad je obrobek (puk) transportován pásovým dopravníkem na pracoviště kontroly kvality. Zde je pomocí rentgenu provedena kontrola kvality daného výrobku. Ve skutečnosti je tato kontrola pouze demonstrativní a stav kvality nabývá pouze dvou hodnot - „vyhovuje“ a „nevyhovuje“. Stupeň kvality je počítačem generován náhodně na základě předem definované pravděpodobnosti. Po kontrole je výrobek dopraven před vyřazovací systém, který na základě stupně kvality obrobek vyřadí vyhozením z výrobní linky do připravené ohrádky v případě, že je stupeň kvality nevyhovující. Pokud je stupeň kvality vyhovující, je umožněno produktu vstup do další výrobní operace, tzn. obrobek je dále dopraven k řezacímu stroji. Tento stroj je ale na pracovišti umístěn pouze dočasně a nebude tak v další práci zohledněn. S pracovištěm bude dále zacházeno, jako by zde činnost řezání vůbec neprobíhala. Pracoviště kontroly kvality je možné vidět na Obr. 4 - Fotografie pracoviště kontroly kvality níže.



Obr. 4 - Fotografie pracoviště kontroly kvality

Manipulační robot

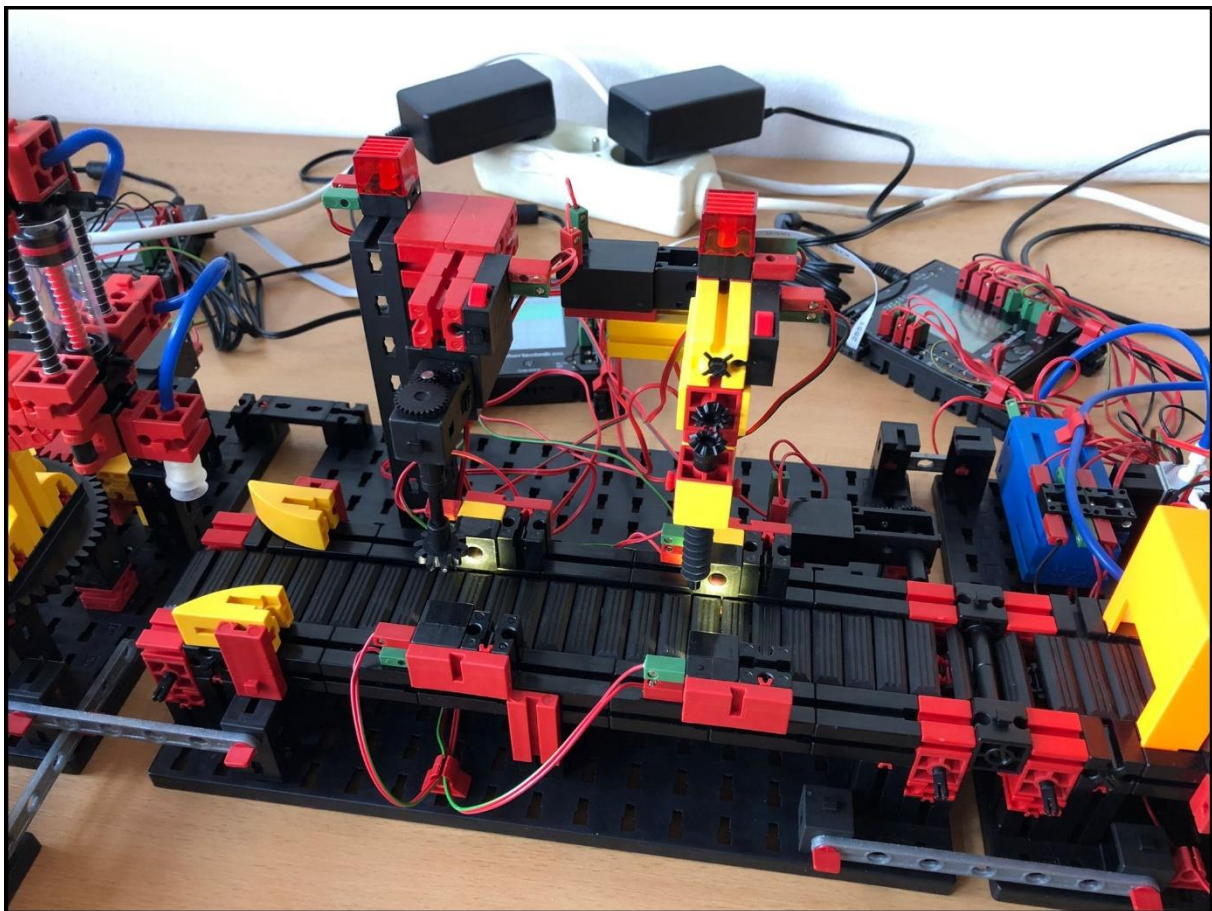
Výrobní linka je navržena do tvaru „L“ čímž je možné dosáhnout úspory místa při jejím umístění do učebny, ve které bude sloužit k výukovým účelům a také k jednoduššímu přemísťování a manipulaci. Tvar linky rovněž zachycuje skutečnost, že i výrobní linky v reálných podnicích mají mnohdy různé tvary a rozložení, které je nutno přizpůsobovat tvaru výrobní haly a jejímu layoutu. Tuto komplikaci je pak nutné překonávat pomocí dalších pásových dopravníků, manipulačních robotů, nebo využitím dalších zaměstnanců. V případě modelu Fischertechnik byl zvolen manipulační robot s podtlakovou přísavkou. Ten funguje tak, že v okamžik, kdy obrobek dojde na konec pracoviště kontroly kvality, přistaví manipulační robot pomocí ramen podtlakovou přísavku na povrch obrobku z horní strany. Vytvořením podtlaku k sobě obrobek přichytí a otočením o 90° proti směru hodinových ručiček přemístí obrobek nad pásový dopravník pracoviště obrábění, které je umístěno kolmo k předchozím dvěma pracovištím. Poté je podtlak uvolněn, čímž se výrobek oddělí od přísavky a pokračuje dále ve výrobním procesu. Aktuální stav manipulačního robota je možné vidět na Obr. 5 - Fotografie manipulačního robota.



Obr. 5 - Fotografie manipulačního robota

Obrábění

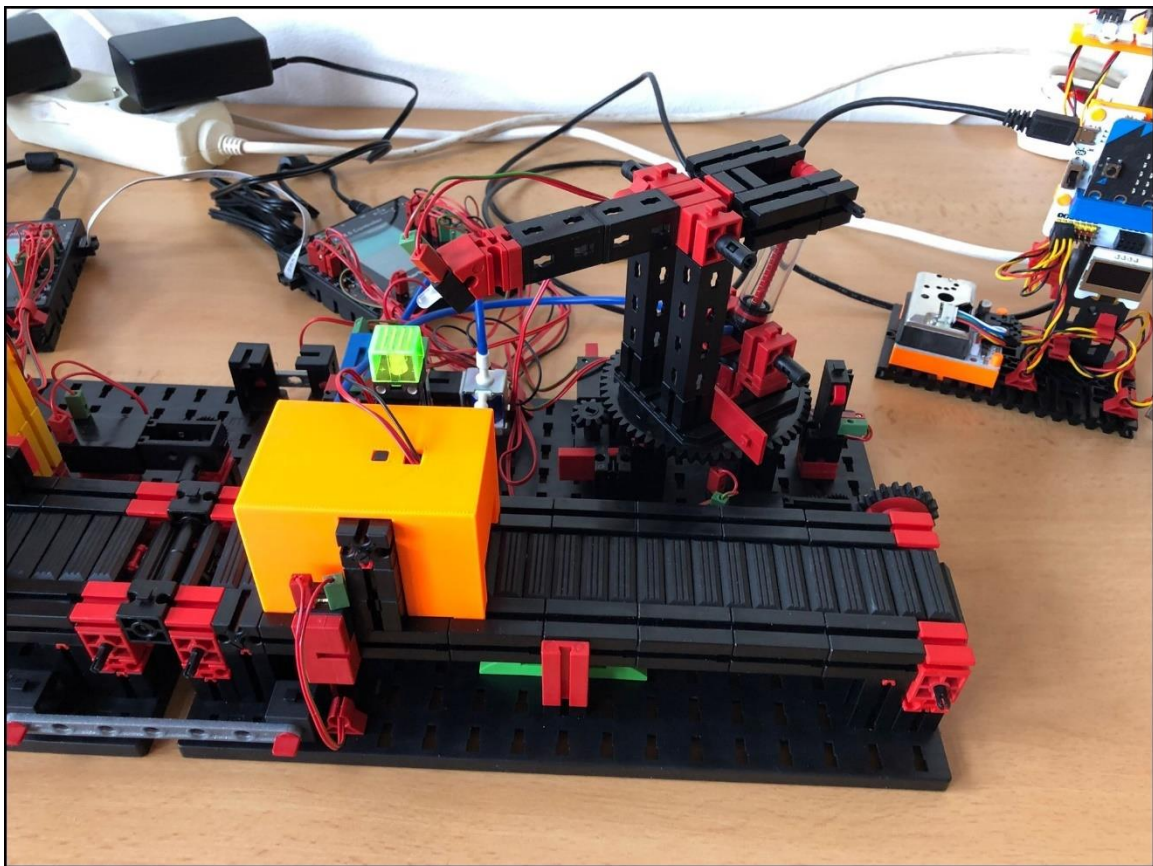
Manipulačním robotem je obrobek přesunut z pracoviště kontroly kvality na pracoviště obrábění. Zde je obrobek nejprve dopraven k frézce, kde dochází k opracování obrobku a následně je dopraven k vrtačce. U obou těchto obráběcích strojů je jistá pravděpodobnost výskytu závady. Tato závada simuluje skutečné závady, které mohou vzniknout v reálném výrobním podniku. V případě takové závady je celá linka pozastavena a nad daným obráběcím strojem je rozsvíceno červené světlo. V tomto stavu zůstává model Fischertechnik až do doby, než dojde ke stlačení tlačítka umístěného na obráběcím stroji. Toto stlačení symbolizuje opravu daného zařízení a výrobní linka se opět vrátí do provozuschopného stavu. Pracoviště obrábění je možné vidět na Obr. 6 - Fotografie pracoviště obrábění níže. Z pracoviště obrábění je poté obrobek přesunut na pracoviště sváření.



Obr. 6 - Fotografie pracoviště obrábění

Sváření

Na pracovišti sváření je obrobek nejprve přehřán na požadovanou teplotu v malé peci. Tato činnost je ale opět pouze demonstrativní. Ve skutečnosti zde neprobíhá přehřátí obrobku, nýbrž detekce jeho barvy. Jak totiž bylo zmíněno dříve, obrobek (v tomto případě puk), může mít jednu ze tří barev. Ta je pro výrobní linku po celou dobu výrobního procesu neznámá a tuto informaci získává až před poslední činností, kterou je sváření. Tato informace je pak využita k určení doby svařování. Pro každou barevnou variantu puku je totiž přednastavena doba, po kterou je výrobek svařován. Po ohřevu, respektive detekci barvy, je obrobek dopraven ke svářecímu robotu. Ten se aktivuje po dopravení obrobku přímo před něj. Svářecí robot pak ramenem přisune svářecí zařízení, které symbolicky provede sváření rozsvícením žárovky na konci svářecího zařízení. Po ukončení operace sváření je výrobní proces dokončen a obrobek je dopraven na konec výrobní linky. Pracoviště sváření je možné vidět Obr. 7 - Fotografie svářecího pracoviště níže.



Obr. 7 - Fotografie svářecího pracoviště

5.2 Databáze

Datový model použité databáze je zobrazen na Obr. 8 - Datový model k Fischertechnik modelu níže. Na obrázku jsou zobrazena jednotlivá datová pole tabulek a vzájemné relace mezi tabulkami. Pro tuto práci budou nejdůležitější tabulky ConversionTables, ModelEvents a ModelDevices. Tabulky SerialPortSettings a Models jsou určeny pro provozuschopnost celého modelu.

Tabulka ModelEvents poskytuje přehled o veškerých událostech, které na modelu proběhly, kde pod pojmem událost je myšlena akce nebo činnost, tj. výrobní proces, který byl na pracovišti vykonán (např. vyskladnění materiálu). Tato tabulka poskytuje informace o tom, kdy daná událost proběhla (sloupec SimTimeMin), jak dlouho trvala (sloupec ProcTimeMin) a identifikační číslo události ConversionTableID, pod kterým je možné v tabulce ConversionTables dohledat o jakou událost se jedná (sloupec Value), popis této události (sloupec Description) a identifikační číslo ModelDeviceID pod kterým je v tabulce ModelDevices možné dohledat, na jakém pracovišti tato událost proběhla (sloupec Device).



Obr. 8 - Datový model k Fischertechnik modelu

6 Návrh mobilní aplikace

Cílem této kapitoly je navržení mobilní aplikace pro zobrazení stavu Fischertechnik modelu. Na počátku kapitoly je určen účel a využití mobilní aplikace, na jejichž základě byl proveden návrh funkcionalit, kterými bude mobilní aplikace disponovat. Dále je popsán návrh GUI pro jednotlivé stránky mobilní aplikace. V závěru kapitoly je popsána tvorba jednotlivých pohledů pro získání potřebných informací z databáze.

6.1 Účel a využití mobilní aplikace

Před začátkem samotného vývoje mobilní aplikace je vhodné definovat, za jakým účelem je mobilní aplikace vytvářena. Díky tomu je mnohem snazší provést návrh jednotlivých funkcionalit. V případě mobilní aplikace pro zobrazení stavu Fischertechnik modelu se jedná o zobrazení stavu jednotlivých výrobních i kontrolních pracovišť, které jsou reprezentovány jednotlivými modely sestavených z různých komponent stavebnice Fischertechnik. Zobrazením stavu se v tomto případě rozumí interpretace dat jak v grafické, tak textové podobě. Během chodu modelů jednotlivých pracovišť jsou data o výrobním procesu zasílána na vzdálenou databázi umístěnou na SQL Serveru. Hlavním účelem je zobrazení aktuálního stavu jednotlivých pracovišť, včetně zobrazení informace, které pracoviště je aktivní, tzn. na jakém pracovišti jsou vykonávány procesy. Protože jsou jednotlivé stavy a operace zaznamenány v databázi, lze vytvořit statistiku, a to jak krátkodobou, tak i dlouhodobou.

Model Fischertechnik je v současnosti využíván jako pomůcka (ukázka) při výuce předmětu Projektové Inženýrství a Počítačová podpora ve strojírenství na Západočeské univerzitě v Plzni (dále jen ZČU). Studenti mohou v rámci těchto předmětů, pochopit základní fyzikální a technické principy jednotlivých komponent modelu Fischertechnik. Další výhodou je imitace výrobního procesu, skládající se z jednotlivých operací a snímání stavů a jejich konverze do digitální podoby. Mobilní aplikace bude zejména využita k demonstrativním účelům během výuky. Cílem bude umožnit studentům vidět názornou ukázkou toho, jak může fungovat výrobní celek, jak je možné jednotlivá data z výroby snímat, zobrazovat, posílat a přistupovat k nim i z míst vzdálených od místa výroby a zejména jak je možné celý výrobní proces sledovat a analyzovat.

6.2 Návrh funkcionalit

Se znalostí, k jakým účelům bude mobilní aplikace využívána, je možné postoupit k návrhu jednotlivých funkcionalit vytvářené aplikace. Bylo definováno celkem pět základních funkcionalit, které budou popsány v následujících kapitolách.

Pro návrh mobilní aplikace byly navrženy tyto funkcionality:

- Zobrazení aktuálně probíhající činnosti na Fischertechnik modelu
- Zobrazení statistiky na základě dat získaných po celou dobu chodu Fischertechnik modelu – dlouhodobá statistika
- Zobrazení statistiky na základě dat získaných z konkrétního časového rozmezí – krátkodobá statistika
- Export zobrazovaných dat
- Upozornění v případě výskytu závady na frézce nebo vrtače

Zobrazení aktuálně probíhající činnosti na Fischertechnik modelu

Hlavní funkcionalitou, kterou bude mobilní aplikace disponovat, je zobrazení procesu, který bude na spuštěném modelu aktuálně probíhat. Účelem bude uživateli umožnit sledovat chod modelu bez nutnosti fyzické přítomnosti. Uživatel mobilní aplikace tak bude mít přehled o probíhajících procesech v reálném čase.

Zobrazení statistiky získané krátkodobým a dlouhodobým pozorováním modelu

Díky ukládání záznamů do databáze, je možné tyto záznamy využít pro tvorbu statistik o již proběhlých procesech na Fischertechnik modelu. Statistika se dělí na statistiku získanou krátkodobým pozorováním a dlouhodobým pozorováním (trendů). Statistika získaná dlouhodobým pozorováním je taková statistika, která zahrnuje veškeré hodnoty od prvního spuštění modelu. Zahrnuje tak všechny relevantní záznamy, které se v tabulce nachází. Naopak krátkodobá statistika využívá pouze hodnoty v určitém časovém rozmezí.

Dlouhodobá statistika modelu jako celku

Pro dlouhodobou statistiku modelu jako celku byly navrženy následující ukazatele:

- zmetkovitosti
- průměrné doby výroby jednoho OK (splňujícího požadavky na jakost) výrobku
- celkové doby strávené na pracovištích
- množstevní zastoupení jednotlivých dílů ve výrobě

Tyto ukazatele budou zobrazovat přehlednou statistiku o chování modelu jako celku. Ukazatel zmetkovitosti bude zobrazovat podíl špatných kusů a celkového počtu výrobků, které vstoupili do výrobního cyklu. Ukazatel průměrné doby výroby jednoho výrobku splňující požadavky na jakost, bude zobrazovat aktuální takt linky a budou brány v potaz i díly, které neprojdou přes kontrolu kvality. Ukazatel celkové doby strávené na pracovišti bude zobrazovat sumu časů všech činností vykonaných na daném pracovišti. Množstevní zastoupení jednotlivých dílů ve výrobě bude zobrazovat jaké díly a v jakém množství prošly výrobní linkou modelu Fischertechnik.

Dlouhodobá statistika jednotlivých pracovišť

Pro dlouhodobou statistiku jednotlivých pracovišť modelu byly navrženy tyto ukazatele:

- průměrná doba vyskladnění dílů ze skladu
- průměrná doba kvalitativní kontroly dílu
- průměrná doba frézování
- průměrná doba vrtání
- chybovosti frézky
- chybovosti vrtačky
- průměrná doba ohřevu dílu
- průměrná doba svařování

Tyto ukazatele budou zobrazovat chování jednotlivých pracovišť Fischertechnik modelu. Ukazatel průměrné doby vyskladnění dílu ze skladu bude zobrazovat, jak dlouho na pracovišti sklad trvá přesunout jeden díl ze zásobníku na dopravní pás. Průměrná doba kvalitativní

kontroly dílu bude zobrazovat, jak dlouho trvá zkontrolovat jeden díl na pracovišti a rozhodnout o něm, zda splňuje či nespĺňuje požadavky na jakost. Průměrná doba frézování a vrtání bude zobrazovat, jak dlouho trvá na konkrétním stanovišti obrobít jeden díl. Chybovost frézky a vrtačky bude zobrazovat v procentech, jak často se vyskytne chyba na konkrétním obráběcím stroji. Ukazatel průměrné doby ohřevu bude zobrazovat, jak dlouho trvá ohřev jednoho dílu před svařováním. Poslední ukazatel průměrné doby svařování bude ukazovat, jak dlouho trvá svařování jednoho dílu.

Krátkodobá statistika

Pro krátkodobou statistiku byl navržen ukazatel historie kvality dílů u posledních deseti výrobních cyklů. Ukazatel zmetkovitosti zobrazuje celkovou zmetkovitost od prvního spuštění modelu. Ta se s přibývajícím množstvím záznamů ustálí na určité hodnotě a krátkodobé zhoršení kvality výroby nebude z tohoto údaje čitelné. Je proto navržen ukazatel zobrazující krátkodobou historii kvality dílů, který bude umožňovat sledování kvality výroby u posledních deseti výrobních cyklů.

Export zobrazovaných dat

Aby mohli být data zpracovaná mobilní aplikací dále využita a uživatelé si je pro jejich uchování nemuseli nikam zapisovat, bude v mobilní aplikaci implementována možnost exportu zobrazovaných dat. Jednotlivá data, která budou exportována, jsou uvedena v kapitole 7.7.3 Stránka export dat (DATA EXPORT).

6.3 Získání potřebných dat pro jednotlivé funkcionality

Jak již bylo uvedeno u popisu aktuálního stavu Fischertechnik modelu, všechna vygenerovaná data jsou ukládána do MSSQL Server databáze. Z té bude mobilní aplikace data stahovat pomocí výběrového SQL příkazu. Aby bylo stahování dat co nejjednodušší (především pro zařízení, na kterém bude mobilní aplikace spuštěna), budou zdrojová data nejprve zjednodušena a zobrazena v tzv. pohledech (známější pod anglickým slovem Views). Pohled je databázový objekt, který zobrazuje data podle předem určených kritérií. Rozdíl mezi pohledem a tabulkou je ten, že tabulka data nejen zobrazuje, ale i uchovává. Naopak pohled je pouze nástroj, který se připojuje k tabulce a data pouze zobrazuje. Důvodem pro využití pohledu je skutečnost, že při využití pohledu nedochází k modifikaci původní tabulky. Je tak možné vytvářet několik různých pohledů se stejnou zdrojovou tabulkou. V následujících odstavcích budou popsány jednotlivé pohledy pro získání potřebných dat pro výše zmíněné funkcionality mobilní aplikace.

Pro mobilní aplikaci bylo navrženo celkem 6 pohledů, které byly tematicky rozděleny podle dat, která uchovávají. Důvodem rozdělení do více pohledů byla přehlednost zobrazených dat a skutečnost, že některé funkcionality využívají pouze jeden záznam v tabulce a některé naopak více. Pro mobilní aplikaci byly vytvořeny následující pohledy:

- Pohled 1 – Aktuálně probíhající akce
- Pohled 2 – Status kvality výrobků
- Pohled 3 – Časy jednotlivých aktivit + chybovost frézky a vrtačky
- Pohled 4 – Zmetkovitost + takt výroby 1 OK dílu
- Pohled 5 – Doba strávená na pracovištích
- Pohled 6 – Přehled vyrobených dílů

6.3.1 Pohled 1 – Aktuálně probíhající akce

První pohled byl vytvořen za účelem získání informace o aktuálním stavu modelu Fischertechnik. Tento pohled je navržen tak, aby v sestupném pořadí zobrazoval aktivity, které na modelu Fischertechnik proběhly. Na první řádce je tak umístěn nejnovější záznam, ze kterého jsou získávána data o právě probíhající aktivitě (ConversionTableID) a pracovišti (ModelDeviceID), na kterém tato aktivita probíhá. Dále je získáván popis aktivity (ActionName) a informace o podmínkách okolního prostředí (Conditions).

6.3.2 Pohled 2 – Status kvality výrobků

Tento pohled byl vytvořen pro funkcionalitu, která zobrazuje historii kvality dílů u posledních 10 výrobních cyklů. Pohled načítá z tabulky ModelEvents identifikační číslo aktivity (ConversionTableID), a to pouze u dvou stavů aktivit. Těmito stavy aktivity se rozumí informace, zda je daný výrobek zmetek či nikoliv. Tyto aktivity jsou poté seřazeny podle času, ve kterém proběhly, od nejnovějších záznamů po nejstarší.

6.3.3 Pohled 3 – Časy jednotlivých aktivit + chybovost frézky a vrtačky

V tomto pohledu je počítáno celkem osm hodnot. Prvních šest představuje výpočet průměrné doby, po kterou trvají jednotlivé akce, které byly blíže popsány v kapitole *Zobrazení statistiky získané krátkodobým a dlouhodobým pozorováním modelu*. Tyto hodnoty byly vypočteny jako suma všech časů jednotlivých úkonů (ProcTimeMin), které jsou potřeba pro vykonání konkrétní výrobní operace. Tato suma je následně vydělena počtem výrobních cyklů, které model Fischertechnik vykonal a následně vynásobena číslem 60 pro převod z minut na sekundy. Tento výpočet je možné zapsat pomocí vzorce následovně:

$$\emptyset \text{ doba akce} = \frac{\sum \text{časů jednotlivých úkonů}}{\text{počet výrobních cyklů}} * 60 \text{ [s]}$$

Poslední dvě hodnoty zobrazují chybovost frézky a vrtačky. Tyto hodnoty jsou získány vydělením počtu výskytu dané chyby celkovým počtem výrobních cyklů, které model Fischertechnik vykonal.

6.3.4 Pohled 4 – Zmetkovitost + takt výroby jednoho OK dílu

V tomto pohledu se počítají dvě hodnoty. První z nich je takt výroby jednoho OK dílu – díl v požadované kvalitě. V tomto výpočtu je nejprve časová hodnota trvání jednotlivých úkonů (ProcTimeMin) převedena na sekundy vynásobením číslem 60 a následně vydělena rozdílem mezi sumou OK výrobků a sumou NOK výrobků. Druhá hodnota, zmetkovitost, je počítána podle standartního vzorce pro výpočet zmetkovitosti, tj.:

$$\text{zmetkovitost} = \frac{\sum \text{NOK výrobků}}{\sum \text{všech výrobků}} * 100 \text{ [%]}$$

Suma OK, NOK a všech výrobků je stejně jako v předchozích pohledech získána na základě hodnoty uvedené v tabulce ModelEvents ve sloupci ConversionTableID, který zobrazuje ID jednotlivých aktivit.

6.3.5 Pohled 5 – Doba strávená na pracovištích

Doba strávená na pracovišti je počítána pro každé z pěti pracovišť stejně, pouze s jinými hodnotami. Jedná se o hodnotu, kde jsou nejprve získány doby jednotlivých událostí (ProcTimeMin) a k nim přiřazených ID jednotlivých aktivit (ConversionTableID). Z tabulky

ConversionTables jsou následně přiřazeny pracoviště (ModelDeviceID) a je provedena suma časů jednotlivých událostí pro jednotlivé pracoviště.

6.3.6 Pohled 6 – Přehled vyrobených dílů

Model Fischertechnik je v této fázi navržen tak, že se do výrobního cyklu mohou ze zásobníků vyskladnit tři různé varianty dílů – ve formě puku v barvě bílé, červené, nebo modré. Tato barva je v průběhu výrobního cyklu detekována a informace o barvě je uchována v databázi. Tento pohled slouží k získání tří hodnot, které zobrazují, kolikrát se daná barva ve výrobním cyklu nacházela.

6.4 Návrh GUI

Pakliže je známo, jaké funkcionality má mobilní aplikace obsahovat, je možné začít s návrhem GUI. To bylo navrženo tak, aby jeho ovládání bylo jednoduché a intuitivní a vzhled aplikace přehledný.

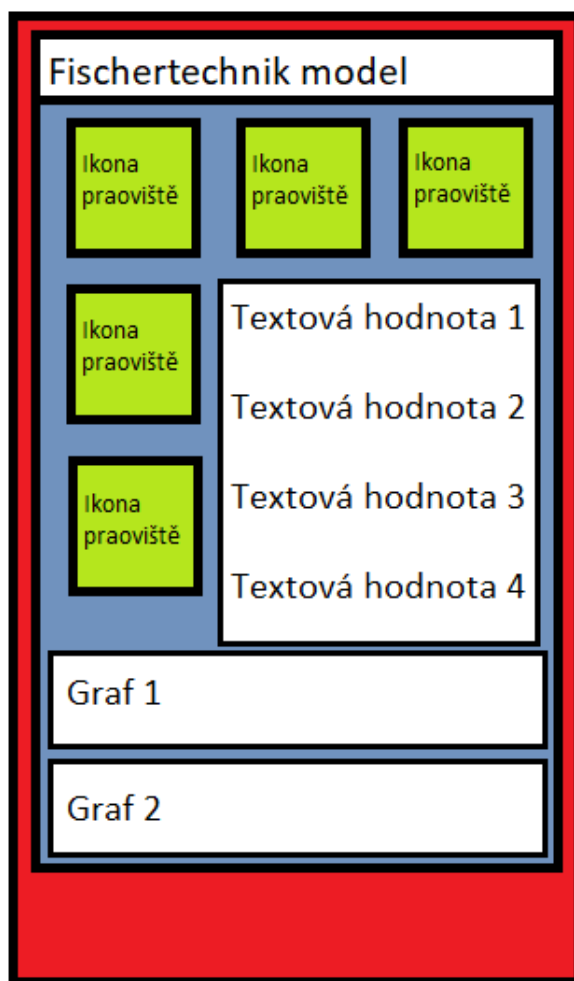
Mobilní aplikace bude mít celkem čtyři druhy stránek, kde pro každou z nich bude GUI navrženo samostatně. Jedná se o stránky – Fischertechnik model, informace, export dat a pracoviště. Stránky budou navrženy do přibližných tvarů displeje mobilního telefonu, který je na obrázcích zvýrazněn červenou barvou.

Stránka export dat a informační stránka

Stránka s informacemi o modelu a stránka umožňující export dat budou svým vzhledem podstatně jednodušší, než stránka pracoviště a Fischertechnik model. Na stránce s informacemi se budou nacházet pouze textové hodnoty obsahující popis Fischertechnik modelu. Na stránce umožňující export dat bude pouze tlačítko pro vygenerování obsahu. Z toho důvodu nebylo považováno za nezbytné tvořit pro tyto stránky grafický návrh.

Stránka Fischertechnik model

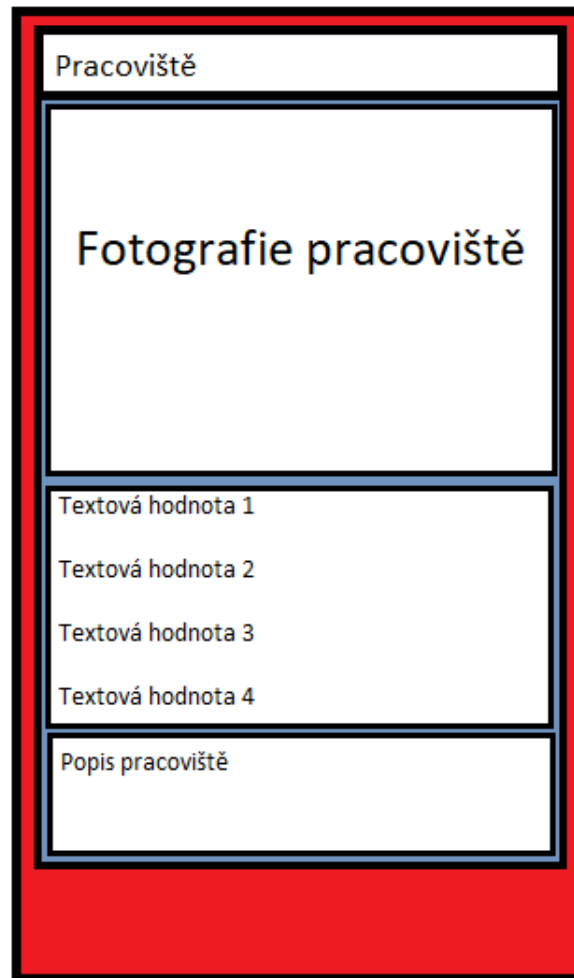
Tato stránka bude hlavní stránkou mobilní aplikace, která se zobrazí jako první při spuštění aplikace. Stránka bude zobrazovat celkový přehled o Fischertechnik modelu a bude možné se z ní dále dostat na stránky, které budou obsahovat informace o konkrétním pracovišti. Na této stránce se budou nacházet jednotlivá pracoviště, která budou umístěna ve tvaru, v jakém se ve skutečnosti nachází ve fyzickém modelu Fischertechnik. Takové rozložení je možné vidět na Obr. 9 - Návrh GUI pro stránku Fischertechnik model, na kterém jsou jednotlivá pracoviště znázorněna zeleným čtvercem s textem „ikona pracoviště“. Na stránce se dále budou nacházet informace získané z databáze. Těmi budou jak textové hodnoty (Textová hodnota 1-4), tak grafy (Graf 1-2).



Obr. 9 - Návrh GUI pro stránku Fischertechnik model

Stránky jednotlivých pracovišť

Stránka pracoviště bude pro každé pracoviště unikátní svými hodnotami, ale vzhled a rozložení budou stejné. Stránka je navržena tak, že jako první uživatel uvidí fotografii daného pracoviště fyzického modelu Fischertechnik (fotografie pracoviště). Pod touto fotografií se budou nacházet textové hodnoty získané z databáze (Textová hodnota 1-4). Poslední položkou bude popis pracoviště, ve kterém bude blíže popsána úloha, kterou dané pracoviště zastává v celém výrobním procesu. Návrh této stránky je možné vidět na Obr. 10 - Návrh GUI pro stránky pracoviště.



Obr. 10 - Návrh GUI pro stránky pracoviště

7 Implementace

Tato kapitola je zaměřena na implementaci navrženého řešení. Pro vývoj mobilní aplikace bude použit framework Xamarin s programovacím jazykem C#. V následujících podkapitolách bude nejprve popsána struktura mobilní aplikace a odůvodnění výběru anglického jazyka, jakožto výchozího jazyka mobilní aplikace. Kapitola bude dále pokračovat popisem jednotlivých technologií, které byly v průběhu tvorby mobilní aplikace využity, načež bude následovat samotná tvorba jednotlivých stránek včetně jejich funkcionalit. V závěru kapitoly budou shrnuty požadavky na mobilní zařízení pro správné fungování mobilní aplikace.

7.1 Struktura aplikace

Aplikace bude mít celkem osm stránek – Fischertechnik model, stránka pro export dat, informační stránka a celkem pět stránek pro pracoviště (jedna stránku pro každé z těchto pracovišť). Po spuštění aplikace bude otevřena hlavní stránka s celkovým náhledem Fischertechnik modelu. Navigace mezi stránkami Fischertechnik model, informační stránkou a stránkou pro export dat bude probíhat pomocí tzv. Flyout menu, které bude popsáno v podkapitole *Výběr šablony*. Ze stránky zobrazující celkový Fischertechnik model bude možné se kliknutím na ikonu konkrétního pracoviště přepnout na stránku daného pracoviště. Mobilní aplikace je navržena tak, aby v případě změny zdrojových dat pro většinu sledovaných parametrů došlo rovněž k jejímu automatickému aktualizování bez nutnosti zásahu uživatele.

Aby mohl uživatel používat mobilní aplikaci, nebude se k ní muset přihlašovat. Bude pouze zapotřebí provést autorizaci přihlášením k ZČU VPN (Virtual Private Network) síti, díky které bude mít mobilní aplikace na konkrétním zařízení přístup k serveru, na kterém jsou ukládána data získaná při chodu Fischertechnik modelu. Pakliže nebude uživatel v době užívání mobilní aplikace připojen k výše zmiňované síti, půjde mu sice mobilní aplikace spustit, ale ukáže mu nevalidní hodnoty a hlášku o tom, že není možné získat data z databáze.

7.2 Jazyk

Všechny objekty, které budou v mobilní aplikaci zobrazeny, budou přeloženy do anglického jazyka. Důvodem je snaha vytvořit takovou mobilní aplikaci, kterou mohou případně využít i zahraniční studenti, kteří budou na ZČU studovat. Dalším důvodem je možnost zadat doprogramování mobilní aplikace rovněž těmto zahraničním studentům. Anglický jazyk použitý v mobilní aplikaci využívá termíny, které jsou vyučovány v rámci studia na Fakultě strojní.

7.3 Výběr šablony

Při zakládání nového projektu a vytváření nové mobilní aplikace může vývojář zvolit jednu ze tří šablon. Díky šabloně pak může mít mnoho užitečných věcí již přednastavených. V tomto případě byla zvolena šablona Flyout, která sama vytvoří hlavní stránku a navigační menu. Díky navigačnímu menu je pak možné přepínat mezi vybranými stránkami.

7.4 Balíčky (knihovny)

Ve vývojovém prostředí Visual Studio lze pro vývoj mobilních aplikací využít instalaci NuGet balíčků. NuGet balíčky představují jakýsi prostředek, který umožňuje implementaci různých prvků při programování a práci s nimi. Balíčky jsou nejčastěji vyvíjeny různými společnostmi či uživateli, kteří pak svůj vytvořený balíček umísťují na stránku nuget.org, kde se balíček stává veřejným a dostupným pro všechny uživatele Visual Studia. Balíčky ale mohou

být umístěny i na soukromé úložiště, které zajistí přístup k balíčkům pouze pro uživatele, kteří mají přístup k danému soukromému úložišti. Pro platformu Xamarin bylo zatím vyvinuto 1 242 takových balíčků. V mobilní aplikaci pro zobrazení stavu Fischertechnik modelu, bylo použito celkem šest balíčků, z toho tři byly nainstalovány Visual Studiem automaticky při vytváření projektu a tři během vývoje k zpřístupnění některých funkcionalit. Tato kapitola se bude dále zabývat popisem tří balíčků, které byly nainstalovány až v průběhu vývoje. [49, 50]

Knihovna Gembox.Spreadsheet

Název knihovny představuje jméno společnosti, Gembox, nalevo od tečky a jejich produkt napravo od tečky, Spreadsheet. Společnost Gembox se zabývá vývojem balíčků pro .Net aplikace, které jsou zaměřeny především na práci s dokumenty. Balíček Spreadsheet je určen pro práci s excelovými soubory ve formátu XLSX, XSL, ODS, textovými soubory (ve formátu CVS, TXT), HTML soubory, PDF soubory, XPS soubory a obrázkovými soubory. Ty lze, mimo jiné, na základě jejich formátu, číst, vytvářet, konvertovat, modifikovat nebo digitálně podepisovat. [51]

Pro tuto práci byl tento balíček implementován za účelem exportu zobrazovaných dat do souboru ve formátu XLSX.

Knihovna Microcharts.Forms

Tato knihovna byla vytvořena Aloísem Denielem za účelem jednoduché tvorby grafů. Knihovna je určena pro mobilní zařízení s OS Android, iOS a UWP a počítače s OS Mac OS. Při tvorbě grafu je nejprve potřeba nadefinovat zdrojová data. Definování je založeno na principu, kde každá hodnota v grafu je nadefinována zvlášť. Například pro sloupcový graf se třemi hodnotami je potřeba nadefinovat tři oddělené zdrojové záznamy. Tyto záznamy se skládají z několika hodnot. Klíčovými jsou celkem čtyři a to, číselná hodnota, která v grafu představuje velikost daného záznamu, název záznamu, textová hodnota záznamu (číselná hodnota představující velikost záznamu) a barva záznamu. Tyto záznamy jsou následně uvedeny jako zdrojová data při tvorbě grafů. Při té je možné vybírat celkem ze šesti druhů grafů. Mimo výše zmíněné hodnoty je díky této knihovně možné nastavovat barvy hodnot, barvy pozadí, animace, velikost textu apod. [52]

Pro tuto aplikaci byl balíček implementován za účelem zobrazování některých z navržených funkcionalit v grafickém provedení.

Knihovna System.Data.SqlClient

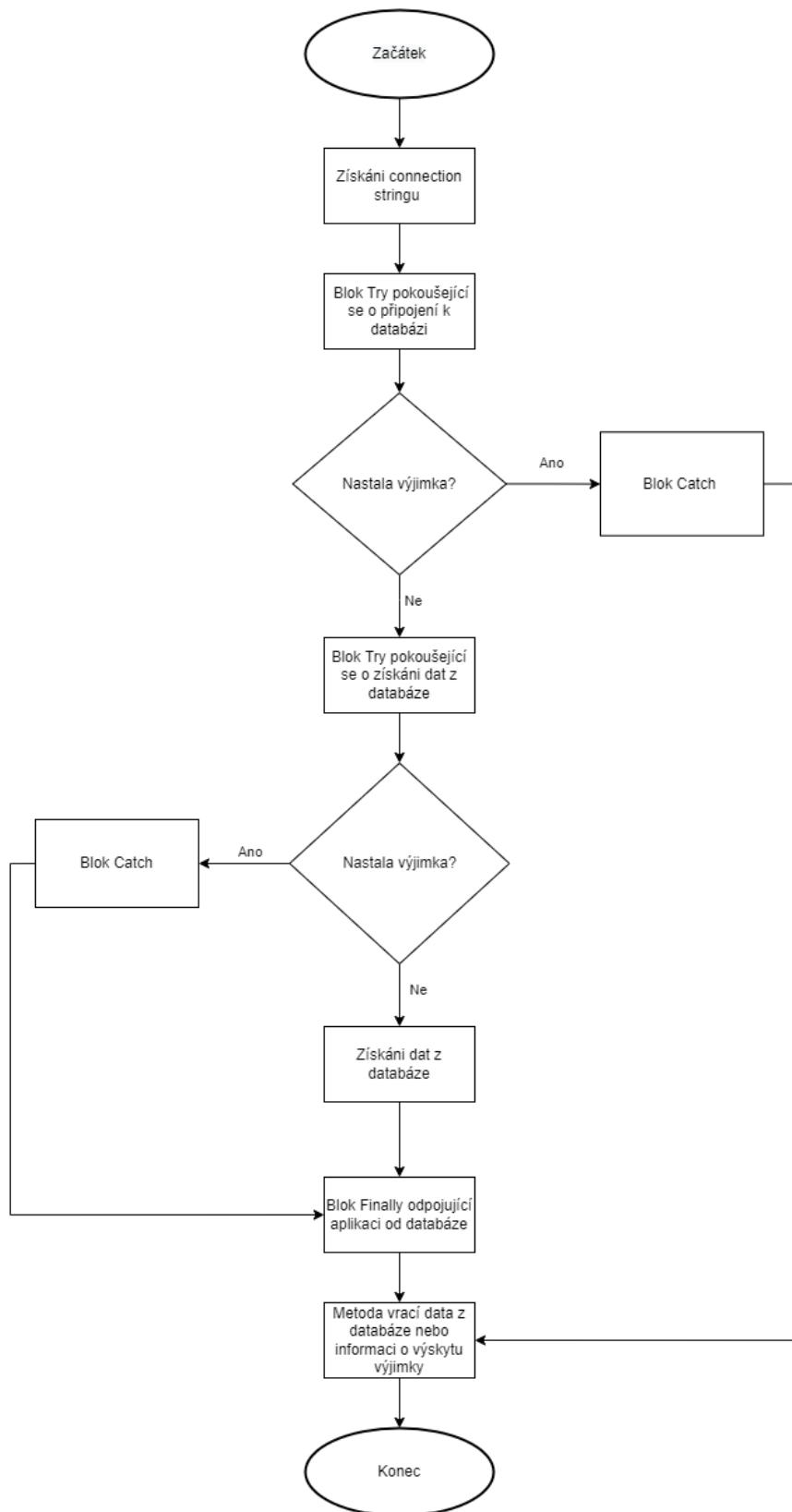
Knihovna System.Data.SqlClient byla vyvinuta společností Microsoft a je určena pro práci s SQL serverem. Pro vytváření mobilní aplikaci je tato knihovna implementována za účelem získávání dat z databáze. Kromě toho by v budoucnu mohla být využita k vytváření nových záznamů v databázi.

7.5 Získávání dat z databáze

S využitím knihovny System.Data.SqlClient je možné získat jakákoliv data z výše navržených pohledů. K přístupu k databázi je nejprve potřeba definovat connection string, což je textový řetězec, který obsahuje data o serveru, ke kterému se bude chtít zařízení připojit. Ten se v tomto případě skládá z IP (Internet Protocol) adresy serveru, názvu databáze a uživatelského jména a hesla potřebného pro přihlášení k databázi. Jako příklad lze uvést následující connection string, ve kterém jsou uvedeny fiktivní údaje (konkrétní údaje zde z bezpečnostních důvodů nejsou uvedeny):

```
connectionString = „Data Source=233.115.52.116\MSSQLSERVER2014; Initial  
Catalog=DPdb; User ID=marekvelich; Password=k9fE537o21“
```

S definovaným connection stringem je možné se připojit k databázi. To je provedeno využitím dvou bloků výjimek – vnější TryCatch a vnořené TryCatchFinally, které umožňují zpracování výjimek, jejichž výskyt je možné při pokusu o připojení k databázi předpokládat. V bloku TryCatch je nejprve v části za klíčovým slovem Try vytvořen požadavek o připojení k databázi. Pokud se mobilní aplikace k databázi nepřipojí, je spuštěna výjimka - blok Catch, ve kterém je uživatel informován o neúspěšném pokusu o připojení. Pokud se mobilní aplikace k databázi připojí, je spuštěn blok TryCatchFinally, ve kterém je nejprve v části Try vytvořen požadavek na čtení dat z databáze. Pokud je čtení úspěšné, je výsledná hodnota uložena do datového typu string a kód pokračuje k části Finally. Pokud je čtení neúspěšné, např. neobsahuje-li zdrojová tabulka daný sloupec, řádek apod., je uživatel informován o tom, že požadovaná data nejsou dostupná. Poté je spuštěna část Finally, ve které dochází k odpojení od databáze. Celý proces je zobrazen ve formě vývojového diagramu na Obr. 11 - Vývojový diagram pro získání dat z databáze a ve formě zdrojového kódu na Obr. 12 - Zdrojový kód pro získání dat z databáze níže.



Obr. 11 - Vývojový diagram pro získání dat z databáze


```
29 references
public string GetSQLData(string column, string table, string condition)
{
    Application.Current.Resources.TryGetValue("connectionString", out var connString);
    string connectionString = connString.ToString().Trim();
    string result = null;

    try
    {
        SqlConnection connection = new SqlConnection(connectionString);
        connection.Open();
        string query = $"Select {column} from {table} where {condition}";
        SqlCommand cmd = new SqlCommand(query, connection);
        SqlDataReader dr = cmd.ExecuteReader();

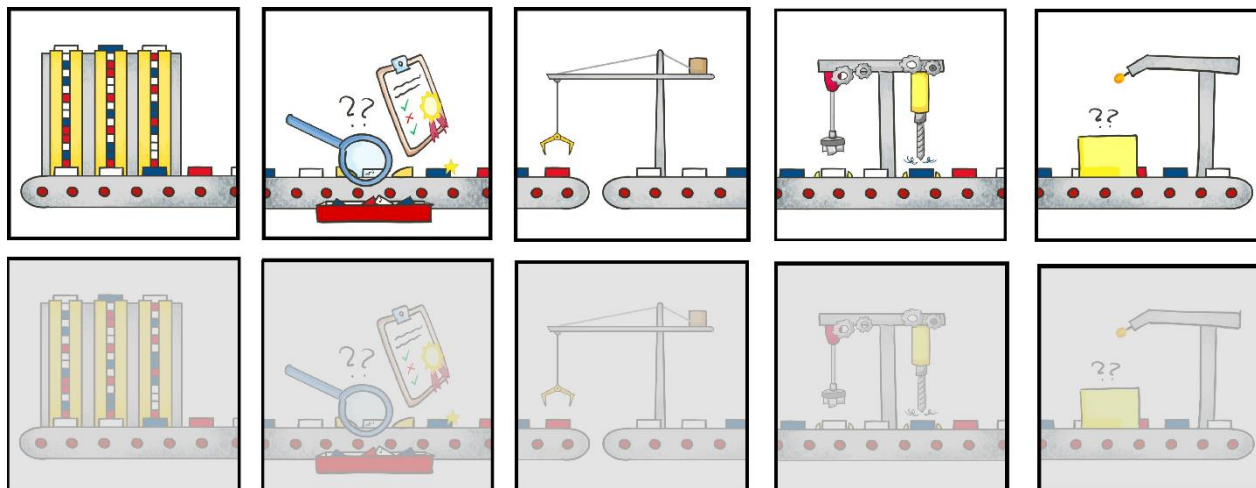
        try
        {
            while (dr.Read())
            {
                result = dr[column].ToString();
            }
        }
        catch (Exception)
        {
            result = "Data not available";
        }
        finally
        {
            connection.Close();
        }
    }
    catch (Exception)
    {
        result = "Database connection lost";
    }

    return result;
}
```

Obr. 12 - Zdrojový kód pro získání dat z databáze

7.6 Tvorba ikon

Pro hlavní stránku byly vytvořeny ikony, které znázorňují jednotlivá pracoviště. Tyto ikony jsou vytvořeny tak, aby bylo na první pohled jasné, čím se dané pracoviště zabývá. Nejsou tak na nich zobrazeny veškeré prvky a funkcionality daného pracoviště, ale především podstata pracoviště a jeho úloha v modelu jako celku. Ikony byly vytvořeny ve dvou variantách, čírá a s šedým filtrem (jejichž využití je popsáno v kapitole *Hlavní stránka – Fischertechnik model*). Ikony je možné vidět na Obr. 13 – Ikony jednotlivých pracovišť.



Obr. 13 - Ikony jednotlivých pracovišť

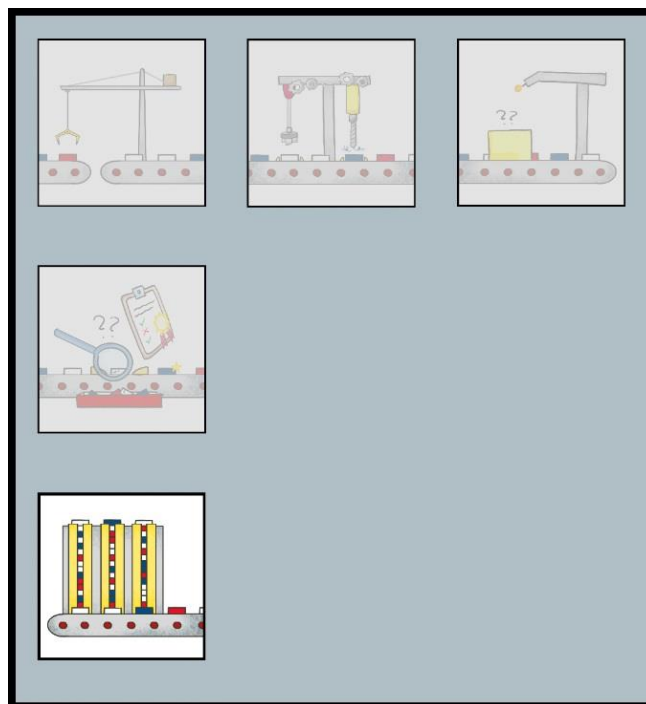
7.7 Stránky mobilní aplikace a jejich funkcionality

Obsahem této kapitoly bude představení jednotlivých stránek mobilní aplikace, včetně výčtu a popisu jejich funkcionalit, které byly navrženy v kapitole *Návrh funkcionalit*. V první podkapitole bude nejprve popsána hlavní stránka – Fischertechnik model, která se zobrazí po spuštění mobilní aplikace. Na tuto podkapitolu bude navazovat popis stránek pracovišť, kde budou nejprve popsány společné prvky, které využívají všechny stránky pracovišť. Dále budou popsány unikátní funkcionality, kterými disponují pouze konkrétní stránky. Po popisu stránek pracovišť bude popsána stránka exportu dat, včetně představení jednotlivých záznamů, které se budou ve vyexportovaném souboru nacházet. Na závěr této kapitoly bude popsána informační stránka mobilní aplikace.

7.7.1 Hlavní stránka – Fischertechnik model

Tato stránka se uživateli zobrazí jako první po spuštění mobilní aplikace. Dále se z ní může uživatel přepnout na stránky, které blíže popisují jednotlivá stanoviště a pomocí výše zmíněného flyout menu i na informační stránku a stránku export dat. Po grafické stránce byla tato stránka vytvořena podle návrhu uvedeného v kapitole *Návrh GUI*. Nachází se zde ikony jednotlivých pracovišť, které jsou rozmístěny podle layoutu fyzického modelu. Kromě zobrazení ikony pracoviště a přesměrování na stránku pracoviště, mají ikony další důležitou funkci. Tou je grafické znázornění aktuálně operujícího pracoviště. Ta využívá záznam o poslední aktivitě (respektive právě probíhajícího procesu) a na základě této informace je zobrazena ikona čírá v případě, že je dané pracoviště aktivní nebo ikona s šedým filtrem v případě, že je pracoviště neaktivní.

Ukázku, na které je graficky zobrazené aktivní pracoviště je možné vidět na Obr. 14 - Ukázka zobrazení aktivního pracoviště.



Obr. 14 - Ukázka zobrazení aktivního pracoviště

Informační panel

Napravo od ikon se nachází hlavní informační panel, který obsahuje funkcionality pro zobrazení dat z jednotlivých pohledů, které byly navrženy v kapitole *Návrh funkcionalit*. Jedná se o zobrazení:

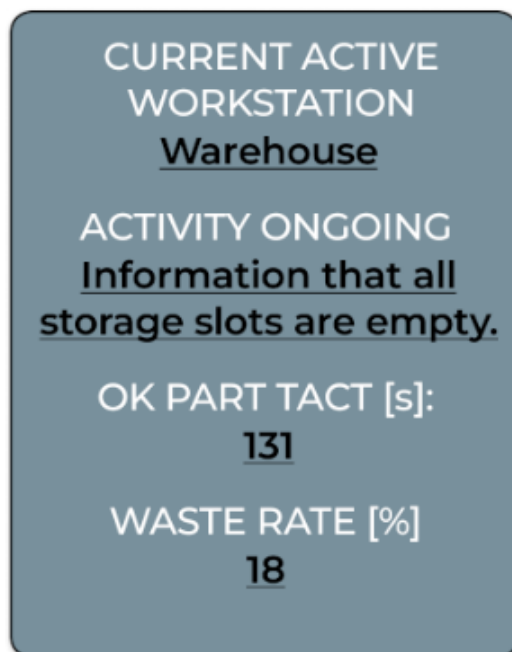
- Pracoviště, které je aktuálně v provozu
- Právě probíhající aktivita (proces) na daném stanovišti, které je aktuálně v provozu
- Takt výroby 1 OK dílu
- Zmetkovitost

První informace, která je v tomto informačním panelu zobrazena, je název pracoviště, které je aktuálně v provozu. Pracoviště, které je aktuálně v provozu, má stejný účel jako grafické znázornění aktuálně operujícího pracoviště. Jediný rozdíl je v tom, že tato funkcionality má textový výstup. Na první pohled by se mohla zdát jako duplicitní, jelikož se zde tato funkcionality již jednou nachází. Při tvorbě funkcionalit se nicméně předpokládalo, že by aplikaci mohli využívat i studenti z jiných fakult, nebo by mohla být demonstrována uchazečům o studium na Fakultě strojní, kteří by nemuseli být schopni rozlišit obsah ikony na první pohled a nebyli by tak schopni ji přiřadit správné označení. Informace o posledním záznamu je získávána z pohledu 1 – viz kapitola 6.3.1 Pohled 1 – Aktuálně probíhající akce.

Na zobrazení pracoviště, které je aktuálně v provozu, navazuje zobrazení konkrétní aktivity (procesu), která na stanovišti aktuálně probíhá. Ta je rovněž získávána z posledního záznamu ve vytvořeném pohledu 1. Tato informace je zobrazena v informačním panelu pod názvem pracoviště.

Pod informací o aktuálně probíhající aktivitě je zobrazena informace, která sděluje, za jakou dobu linka průměrně vyrábí jeden OK kus. Tato hodnota je udávána v sekundách a získávána z pohledu 4 – viz kapitola 6.3.4 Pohled 4 – Zmetkovitost + takt výroby jednoho OK dílu.

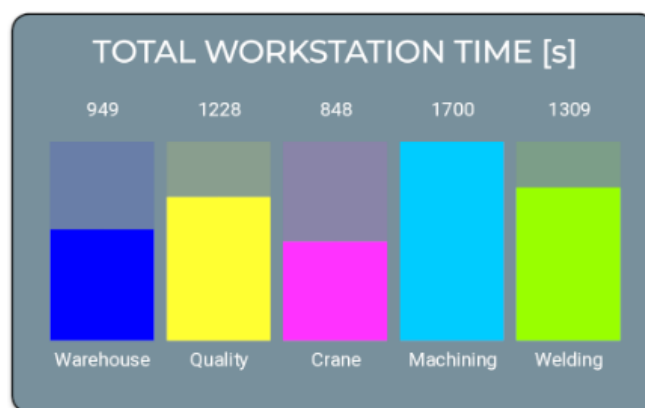
Poslední informací zobrazenou v informačním panelu je ukazatel zmetkovitosti. I jeho hodnota je brána z pohledu 4 a zobrazována je v procentech. Tento informační panel je možné vidět na Obr. 15 - Informační panel stránky Fischertechnik model níže.



Obr. 15 - Informační panel stránky Fischertechnik model

Grafy

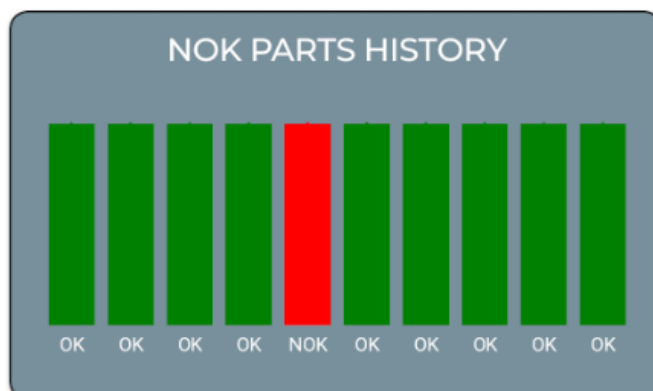
Pod informačním panelem a ikonami jednotlivých pracovišť se nachází 3 grafy. Ty jsou umístěny pod sebou tak, že každý z nich zabírá celou šířku displeje mobilního telefonu. První graf zobrazuje celkový čas strávený na jednotlivých pracovištích. Zde každé pracoviště představuje jeden sloupec, který nabývá hodnotu času (v sekundách), po kterou byla daná pracoviště v provozu. Mezi hodnotami je v grafu viditelný rozdíl a mohou tak být snadno porovnány. Hodnoty byly získány z pohledu 5 – viz kapitola 6.3.5 Pohled 5 – Doba strávená na pracovištích. Tento graf je zobrazen na Obr. 16 - Celková doba strávená na pracovištích níže.



Obr. 16 - Celková doba strávená na pracovištích

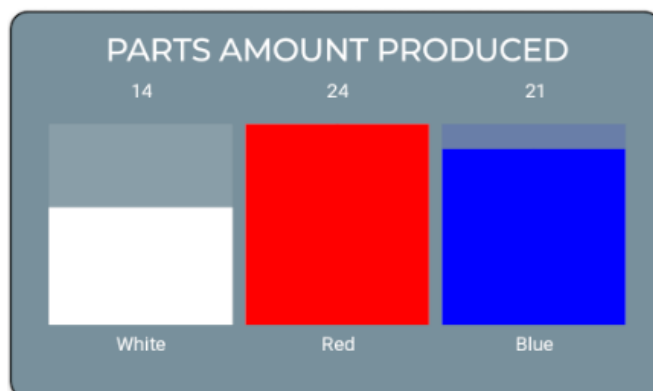
Druhý graf zobrazuje krátkodobou historii kvality dílů. Zde jsou z pohledu 2 (viz kapitola 6.3.2 Pohled 2 – Status kvality výrobků) získány informace o kvalitě výrobku pro posledních 10 výrobních cyklů. V grafu jsou pak tyto hodnoty zobrazeny ve sloupcích tak, že vpravo je díl, který byl vyroben jako poslední a vlevo díl, který byl vyroben jako první. Označení kvality

dílů je provedeno jak slovy, tak graficky. Pod každým sloupcem je uvedeno „OK“ v případě, že se jedná o kvalitní výrobek a „NOK“ v případě, že se jedná o zmetek. Graficky jsou správné výrobky označeny zelenou barvou a zmetky červenou barvou. Toto grafické znázornění kvality dílů je možné vidět na Obr. 17 - Historie kvality dílu u posledních deseti výrobních cyklů.



Obr. 17 - Historie kvality dílu u posledních deseti výrobních cyklů

Poslední graf na této stránce zobrazuje množstevní zastoupení jednotlivých variant výrobků, které vstoupily do výrobního cyklu. Ty jsou v této fázi modelu celkem tři – bílá, červená, modrá varianta obrobku (puku). V grafu se nachází tři sloupce, kde každý z nich nabývá hodnotu, která vyjadřuje, kolikrát se daný obrobek objevil ve výrobním cyklu. Jednotlivé sloupce jsou zároveň zbarveny do barvy výrobku, který reprezentují. Tento graf je možné vidět na Obr. 18 - Přehled počtu vyrobených dílů.



Obr. 18 - Přehled počtu vyrobených dílů

Podmínky okolního prostředí

Pod grafy se nachází informační panel, který zobrazuje podmínky prostředí v blízkém okolí modelu Fischertechnik. Těmi jsou teplota [°C], hluk [dB] a vlhkost vzduchu [%]. Jelikož se zdrojová data nachází ve tvaru, kdy jsou informace o všech podmínkách uloženy v jednom textovém řetězci, nebylo jejich přímé užití v mobilní aplikaci možné. Zdrojová data musela být nejprve načtena do mobilní aplikace, ve které jsou dále zpracována pomocí tzv. regulárního výrazu. Ten umožňuje vyhledávání konkrétních hodnot na základě předem definovaných parametrů. V tomto příkladě regulární výraz vyhledával jednomístnou až třímístnou celočíselnou hodnotu, která následovala ihned po definovaném textovém řetězci. Jako příklad lze uvést získání hodnoty pro okolní teplotu (temperature), kde regulární příkaz vypadal takto:

`(?<=.temperature. :).[0-9]{0,2}`

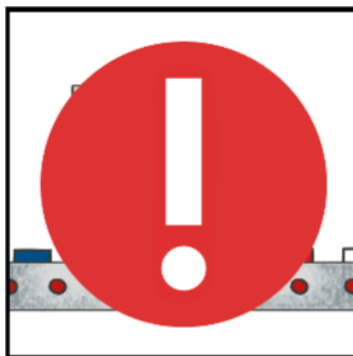
Výsledný panel je možné vidět níže na Obr. 19 - Panel zobrazující podmínky okolního prostředí.



Obr. 19 - Panel zobrazující podmínky okolního prostředí

Upozornění na závadu na pracovišti

Tato stránka rovněž disponuje funkcionalitou, jejíž efekt je možné vidět jen v určité situaci. Jedná se o upozornění uživatele na skutečnost, že se na jednom ze strojů na obráběcím pracovišti vyskytla závada, kterou je nutno odstranit. Takové upozornění je provedeno objevením symbolu vykřičníku v červeném kruhu přes ikonu pracoviště. Pokud je závada odstraněna, symbol vykřičníku zmizí. Tento jev je možné vidět na Obr. 20 - Signalizace závady na pracovišti níže.



Obr. 20 - Signalizace závady na pracovišti

Výsledná podoba této stránky Fischertechnik model je zachycena níže na Obr. 21 - Výsledná podoba stránky Fischertechnik model.



Obr. 21 - Výsledná podoba stránky Fischertechnik model

7.7.2 Stránky pracovišť

Stránek pro jednotlivá pracoviště bylo vytvořeno celkem pět – jedna pro každé pracoviště. Tyto stránky mají několik společných funkcionalit, které jsou na každé stránce stejné, pouze zobrazují různé hodnoty. V následujících odstavcích budou nejprve popsány společné funkcionality, na které bude navazovat popis funkcionalit, které jsou pro každou stránku unikátní.

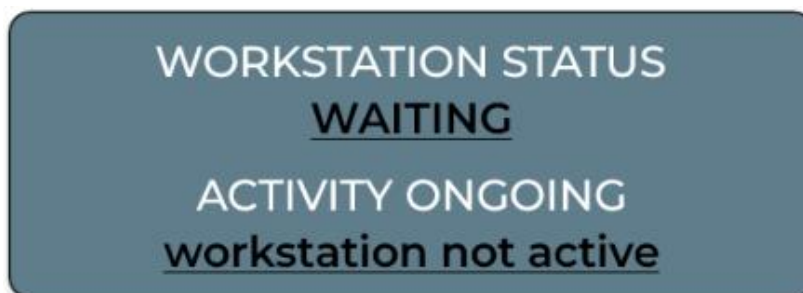
Společné funkcionality

Na stránky pracovišť je možné se dostat kliknutím na ikonu pracoviště na hlavní stránce – Fischertechnik model. Po otevření se v horní části obrazovky objeví fotografie daného pracoviště fyzického modelu Fischertechnik. Tyto fotografie je možné vidět v kapitole *Model Fischertechnik*, ve které byla jednotlivá pracoviště představena.

Pod fotografií se nachází informační panel, který zobrazuje informace o konkrétním pracovišti. Jako první je v informačním panelu umístěna informace o aktivitě pracoviště. Ta zobrazuje, jestli je zařízení aktuálně v činnosti či nikoliv. Nabývat může dvě hodnoty – „aktivní“ a „čeká“. Zobrazovaná hodnota je získávána z pohledu 1.

Na tuto informaci navazuje informace o aktuálně probíhající aktivitě v případě, že je pracoviště aktivní. Pokud aktivní není, je zobrazena hodnota „pracoviště není aktivní“. Princip získávání této informace je nastaven tak, že funkcionalita, která zajišťuje zobrazení dat, nejprve získá informaci o aktivním pracovišti. Pokud je hodnota shodná s názvem pracoviště,

funkcionalita se rovněž dotáže na aktivitu, která na tomto pracovišti probíhá. Pokud je ale hodnota pracoviště různá a aktivní pracoviště je jiné, funkcionalita automaticky přiřazuje hodnotu „pracoviště není aktivní“. Data pro tuto informaci jsou rovněž získávána z pohledu 1. Tuto funkcionalitu je možné vidět na Obr. 22 - Aktuální status pracoviště níže.

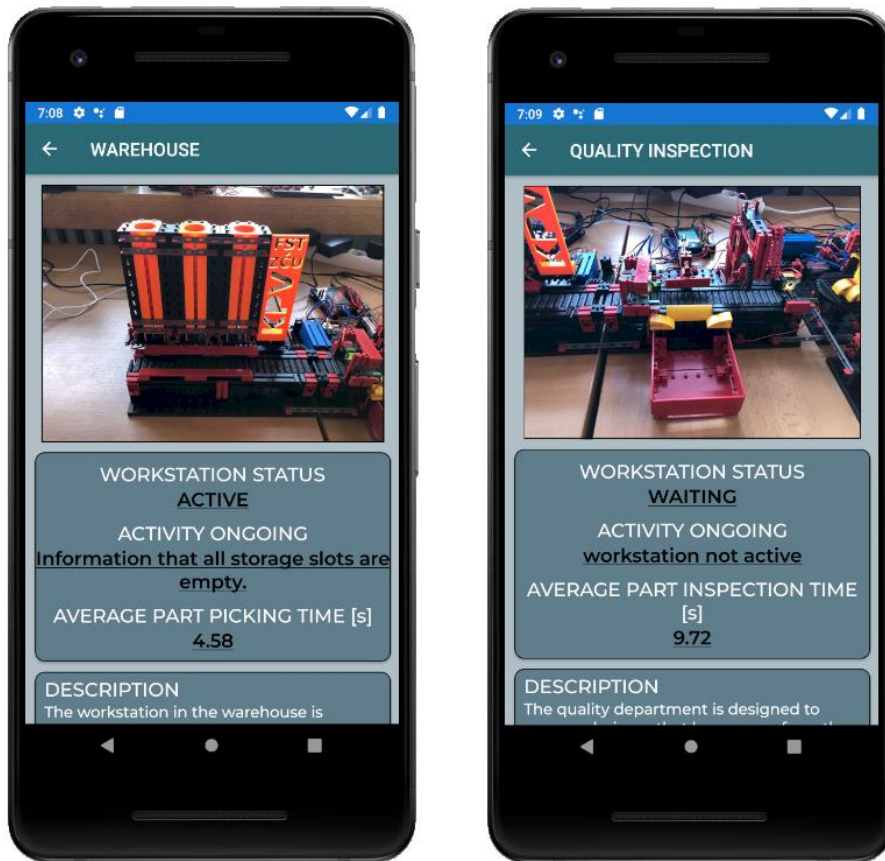


Obr. 22 - Aktuální status pracoviště

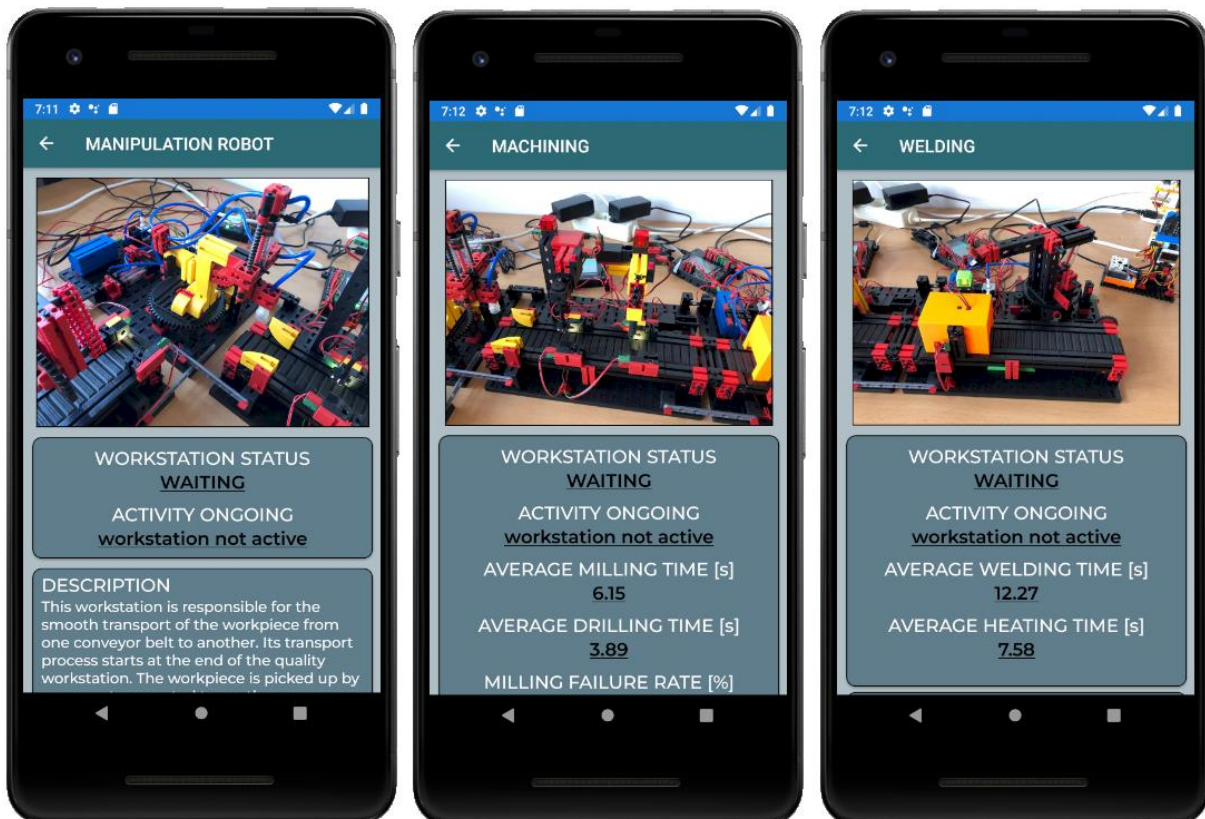
Unikátní funkcionality

Unikátní funkcionality jsou stránkám pracovišť přiřazeny podle činností, které vykonávají a podle strojů a zařízení, které se na jednotlivých pracovištích nacházejí. Jedná se především o funkcionality, které zobrazují statistický časový údaj (v sekundách) o daném pracovišti, který byl získán dlouhodobým pozorováním. Na pracovišti skladu se nachází funkcionalita, která zobrazuje informaci o průměrné době vyskladnění jednoho kusu polotovaru ze zásobníku na výrobní pás. Na pracovišti kvality se nachází informace o průměrné době kontroly polotovaru. Pracoviště obrábění obsahuje funkcionality celkem čtyři. První dvě zobrazují rovněž časové údaje, konkrétně průměrnou dobu frézování a průměrnou dobu vrtání. Druhé dvě zobrazují míru poruchovosti frézky a vrtačky udávanou v procentech. Poslední pracoviště – pracoviště svařování, zobrazuje dvě hodnoty. Těmi jsou průměrná doba svařování a průměrná doba ohřevu polotovaru.

Pod informačním panelem, který zobrazuje výše zmíněné informace, se nachází další informační panel, na kterém je umístěn popis konkrétního pracoviště. Výsledná podoba těchto stránek je zachycena na Obr. 23 - Výsledná podoba stránek Sklad a Kontrola kvality a Obr. 24 - Výsledná podoba stránek Manipulační robot, Obrábění a Svařování níže.



Obr. 23 - Výsledná podoba stránek Sklad a Kontrola kvality



Obr. 24 - Výsledná podoba stránek Manipulační robot, Obrábění a Svařování

7.7.3 Stránka export dat (DATA EXPORT)

Tato stránka byla vytvořena za účelem získání a uchování informací zobrazovaných v mobilní aplikaci. Na stránce je přítomno pouze jediné tlačítko, po jehož kliknutí se provede export dat a uživateli se otevře program MS Excel, ve kterém budou data zobrazena. Při exportu budou získány tyto informace:

- Takt výroby 1 OK dílu [s]
- Zmetkovitost [%]
- Průměrná doba vyskladnění polotovaru [s]
- Průměrná doba kontroly kvality polotovaru [s]
- Průměrná doba frézování [s]
- Průměrná doba vrtání [s]
- Poruchovost frézky [%]
- Poruchovost vrtačky [%]
- Průměrná doba svařování [s]
- Průměrná doba ohřevu polotovaru před svařováním [s]

Výsledná podoba této stránky je zachycena na Obr. 25 – Výsledná podoba stránky export dat níže.

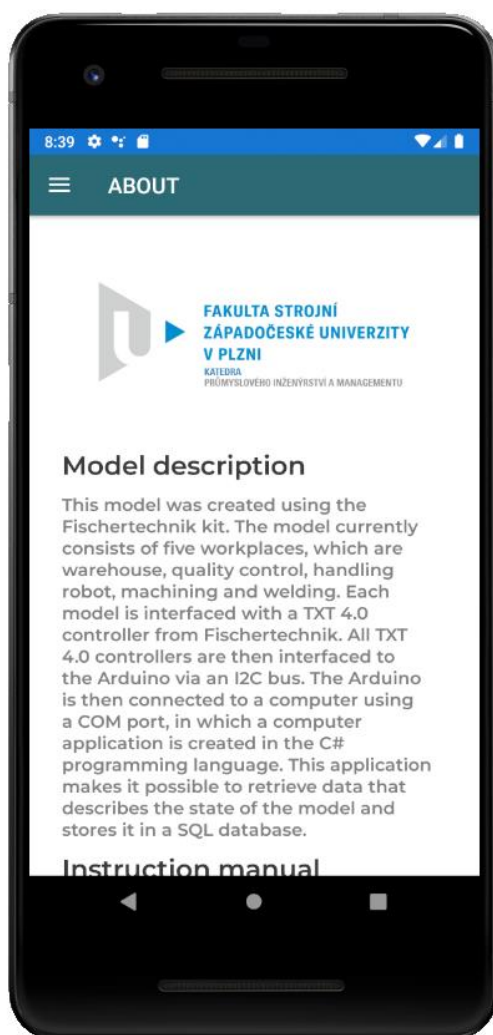


Obr. 25 – Výsledná podoba stránky export dat

7.7.4 Informační stránka (ABOUT)

Na této stránce se nachází základní informace o modelu Fischertechnik. Stránka je rozdělena do dvou částí. V té první je popsáno, co je model Fischertechnik a z jakých pracovišť se skládá. Dále je zde vysvětlen princip propojení modelu Fischertechnik s počítačem a následný sběr dat. Ve druhé části je popsána mobilní aplikace. Nachází se zde výčet stránek, které mobilní aplikace obsahuje a jejich účel. Rovněž je zde popsána struktura mobilní aplikace a jak s ní zacházet.

Výsledná podoba této stránky je zachycena na Obr. 26 – Výsledná podoba informační stránky níže.



Obr. 26 – Výsledná podoba informační stránky

7.8 Podmínky pro spuštění

Spuštění mobilní aplikace je možné na chytrých zařízeních s OS Android. Ten zároveň musí být ve verzi 5.0 Lollipop a vyšší. Ke správnému fungování mobilní aplikace je nutné být připojen ke školní síti. To je možné připojením ke školní Wi-Fi, nebo vzdáleným přístupem pomocí aplikace Cisco AnyConnect (VPN). Data získaná z modelu jsou tak dostupná pouze uživatelům, kteří jsou zaměstnanci nebo studenty na ZČU. Pro využití funkcionality exportu dat je nutné mít v mobilním zařízení nainstalovaný software MS Excel.

8 Testování

Mobilní aplikace byla vyvinuta v Microsoft Visual Studiu na notebooku s OS Windows 10, které bylo pomocí aplikace Cisco AnyConnect připojeno interní síti ZČU (vpn.zcu.cz). Na tomto zařízení byla tato aplikace rovněž testována. K tomu byl využit emulátor Visual Studia pro Android. V něm bylo vytvořeno virtuální mobilní zařízení s OS Android 9.0 Pie, RAM 1,5 GB a čtyřjádrovým procesorem, který svými parametry odpovídá spíše starším a méně výkonným mobilním telefonům. Dá se tedy předpokládat, že pokud bude mobilní aplikace plně funkční na zařízení s těmito parametry, nebude s největší pravděpodobností problém s jejím fungováním i na většině dnešních mobilních zařízeních. Testování pomocí emulátoru probíhalo po celou dobu tvorby mobilní aplikace, čímž byly ověřovány nově přidané funkcionality a formováno GUI aplikace.

Testování při spuštěném výrobním procesu na Fischertechnik modelu

Ověřování funkcionalit mobilní aplikace bylo potřeba provést za podmínek, kdy byl Fischertechnik model v provozu a v databázi se vytvářely nové záznamy. Díky tomuto testování bylo možné sledovat, že všechny funkcionality mobilní aplikace plnily účel, za jakým byly naprogramovány.

Jelikož mobilní aplikace bezchybně komunikovala s databází a všechny funkcionality dle pozorování fungovaly, mohla být provedena další fáze testování mobilní aplikace na fyzickém mobilním zařízení.

8.1 Testování mobilní aplikace na fyzickém mobilním zařízení

Pro testování mobilní aplikace na fyzickém zařízení bylo nejprve zapotřebí mobilní aplikaci archivovat a vytvořit z ní spustitelnou instalaci pro mobilní zařízení s OS Android – soubor s příponou *.apk. Mobilní aplikace byla poté přenesena na fyzické zařízení, kde může být provedena její instalace. K té bylo potřeba snížit úroveň zabezpečení mobilního telefonu. OS Android totiž od základu blokuje instalaci aplikací z jiných zdrojů, než je oficiální platforma pro získávání aplikací Google Play. Před spuštěním aplikace bylo mobilní zařízení připojeno ke školní Wi-Fi síti, aby mohla být umožněna komunikace s databází.

Samotné testování mobilní aplikace na fyzickém zařízení bylo velice podobné testování na notebooku. Po spuštění Fischertechnik modelu bylo sledováno, zda zobrazované hodnoty v mobilní aplikaci odpovídají aktuálnímu chování fyzického modelu Fischertechnik. Při testování bylo zpozorováno, že funkcionality, která graficky zobrazuje aktuálně operující pracoviště, nefunguje stejně jako při testování pomocí emulátoru. Funkce byla původně navržena tak, aby veškerá neaktivní pracoviště, na kterých v daný okamžik neprobíhá žádná aktivita, měla neprůhlednost ikony nastavenou na 50 % a naopak pracoviště, na kterém aktivita probíhá, měla neprůhlednost nastavenou na 100 %. Ikony s 50 % neprůhledností by propouštěly barvu pozadí a vytvářely efekt neaktivního pracoviště, ikona se 100 % neprůhledností by pak nepropouštěla barvu pozadí vůbec, zůstávala by jí jasná a čirá barva a vytvářela efekt aktuálně operujícího pracoviště. Jelikož je model Fischertechnik nastaven tak, že je prováděna pouze jedna aktivita najednou, zůstávala by se 100 % neprůhledností pouze jedna ikona.

Při testování na notebooku s využitím emulátoru tato funkce fungovala bezchybně. Problém však nastal při přenosu mobilní aplikace na fyzické mobilní zařízení. Zde tato funkcionality ze zatím nezjištěných příčin nefungovala a hodnotu neprůhlednosti automaticky nastavovala všem ikonám na 100 %. Z tohoto důvodu musela být funkce přeprogramována do podoby, ve které je grafický výsledek stejný, ale efektu se dosahuje jiným způsobem. Pro každý

obrázek ikony byl vytvořen ještě jeden obrázek, který byl v grafickém programu upraven tak, aby vypadal jako původní ikona s 50 % neprůhledností. Tyto obrázky pak byly uvedeny jako zdroj pro GUI mobilní aplikace na základě získané hodnoty aktuálně operujícího pracoviště z databáze, tzn. v případě, že je dané pracoviště v provozu, bude uživateli zobrazen čirý, neupravený obrázek, v opačném případě bude zobrazen upravený obrázek s šedým filtrem.

Žádné další závady nebyly při testování mobilní aplikace na fyzickém mobilním zařízení pozorovány.

8.2 Sumarizace výsledků a možná rozšíření

Vytvořená mobilní aplikace je navržena tak, aby uživateli umožnila mít přehled nad procesy, které na fyzickém modelu Fischertechnik aktuálně probíhají. Zároveň aplikace poskytuje přehlednou statistiku stěžejních ukazatelů výkonosti jak modelu jako celku, tak i pracovišť, jejichž povaha toto pozorování umožňuje. Užitečná je i možnost exportu dat, díky které si uživatel může uchovat data o modelu Fischertechnik, aniž by se musel k databázi připojovat, např. přes Microsoft SQL Server Management Studio.

Obousměrná komunikace

Vhodným rozšířením mobilní aplikace by mohlo být umožnění obousměrné komunikace mezi modelem Fischertechnik a mobilní aplikací. V současné době je možná pouze jednosměrná komunikace, kdy jsou data získaná pozorováním modelu ukládána do databáze, ze které je mobilní aplikace získává. Vhodnou úpravou jak modelu Fischertechnik, tak mobilní aplikace, by mohlo být docíleno obousměrné komunikace, při které by byl model Fischertechnik schopen přijímat data z mobilní aplikace. Tím by mohlo být docíleno například spouštěním modelu, jeho pozastavením nebo provedením „opravy“ strojů v případě výskytu poruchy.

Závěr

Tato diplomová práce se zabývala tvorbou mobilní aplikace pro zobrazení stavu Fischertechnik modelu, který byl vytvořen na katedře Průmyslového inženýrství a managementu Fakulty strojní na Západočeské univerzitě v Plzni. Tento model i mobilní aplikace samotná, slouží jako výuková pomůcka demonstrující možné využití prvků Průmyslu 4.0.

Práce je rozdělena na dvě části. V první části je nejprve vysvětlen pojem chytrá továrna. Dále jsou představeny jednotlivé průmyslové revoluce a jsou blíže popsány stěžejní prvky Průmyslu 4.0. Následující kapitola Mobilní aplikace a jejich vývoj je zaměřena na představení jednotlivých druhů mobilních aplikace z pohledu technického zpracování, kterými jsou nativní, webová a hybridní. Představení těchto druhů bylo zakončeno výčtem jejich výhod a nevýhod a následným výběrem nativní varianty, jakožto nejvhodnějšího řešení pro tvorbu mobilní aplikace pro zobrazení stavu Fischertechnik modelu. V této kapitole bylo rovněž provedeno představení nejpoužívanějších OS pro mobilní telefony, které vyústilo volbou OS Android, jakožto cílové platformy, pro který byla mobilní aplikace vytvořena. Kapitola se zabývá i rešerší na téma vývoje mobilních aplikací, ve které jsou představeny jednotlivé strategie a fáze vývoje, které je možné aplikovat při tvorbě mobilní aplikace. Stěžejní v této části je rešerše technologií, které je možno při tvorbě mobilních aplikací využít. Tato kapitola byla zaměřena především na dostupné frameworky, jejich porovnání a výběr nejvhodnější varianty. Pozornost byla věnována také tématu databází, ve které byly uvedeny základy tvorby databází a práce s nimi. Poslední kapitola této části provedla detailní rozbor současného stavu Fischertechnik modelu, který se zaměřil především na popis jednotlivých pracovišť, principu komunikace mezi modelem Fischertechnik a mobilní aplikací a popis současného stavu databáze, do které jsou záznamy o modelu sbírány.

Druhá část této práce je zaměřena na tvorbu mobilní aplikace. Zde je nejprve proveden návrh jednotlivých funkcionalit, kterými bude mobilní aplikace disponovat. Dále následoval návrh GUI a pohledů v databázi vytvořených za účelem snadného přístupu k datům. Práce dále pokračuje kapitolou, která se zabývá tvorbou mobilní aplikace. Zde jsou nejprve popsány technologie, o jejichž využití bylo rozhodnuto až v průběhu tvorby mobilní aplikace, metoda pro získávání dat z databáze, struktura mobilní aplikace a zvolená šablona. Poté byly popsány jednotlivé stránky mobilní aplikace, včetně jejich účelu a jednotlivých funkcionalit. Každá z těchto stránek byla poté graficky vyobrazena na fotografiích pořízených po dokončení vývoje mobilní aplikace. Poslední kapitola této části je věnována testování mobilní aplikace. To bylo provedeno jak na virtuálním mobilním zařízení v počítači, tak na fyzickém mobilním zařízení.

Seznam literatury a dalších pramenů

- [1] YouGov. (2020). *By the age of seven, 53% of the children in the UK own a mobile phone* [online]. [cit. 20. 2. 2022] Dostupné na: <https://yougov.co.uk/topics/resources/articles-reports/2020/01/31/age-seven-53-children-uk-own-mobile-phone>
- [2] The European Business Review. (2021). *Is a Smart Factory Right for Your Business?* [online]. [cit. 20. 2. 2022] Dostupné na: <https://www.europeanbusinessreview.com/is-a-smart-factory-right-for-your-business/>
- [3] Alena Szydlovská. (2017). *Nová průmyslová revoluce – Průmysl 4.0* [online]. Dostupné na: <https://www.datamix.eu/blog/nova-prumyslova-revoluce-prumysl-4-0/>
- [4] Amfg.ai. (2019). *Industry 4.0: 7 Real-World Examples of Digital Manufacturing in Action* [online]. [cit. 1. 3. 2022] Dostupné na: <https://amfg.ai/2019/03/28/industry-4-0-7-real-world-examples-of-digital-manufacturing-in-action/>
- [5] Andrea Cejnarová. (2015). *Od 1. průmyslové revoluce ke 4.* [online]. [cit. 1. 3. 2022] Dostupné na: https://www.technickytydenik.cz/rubriky/ekonomika-byznys/od-1-prumyslove-revoluce-ke-4_31001.html
- [6] IBM. *What is Industry 4.0?* [online]. [cit. 1. 3. 2022] Dostupné na: <https://www.ibm.com/topics/industry-4-0>
- [7] Oracle. *What is IoT?* [online]. [cit. 4. 3. 2022] Dostupné na: <https://www.oracle.com/cz/internet-of-things/what-is-iot/>
- [8] Barbora Kodřousková. (2021). *INTERNET VĚCÍ (IOT): DEFINICE, PŘÍKLADY VYUŽITÍ, PRODUKTY* [online]. [cit. 6. 3. 2022] Dostupné na: <https://www.rascasone.com/cs/blog/iot-internet-veci-definice-produkty-historie>
- [9] Evropský parlament. (2021). *Co je umělá inteligence a jak ji využíváme?* [online]. [cit. 6. 3. 2022] Dostupné na: <https://www.europarl.europa.eu/news/cs/headlines/society/20200827STO85804/umela-inteligence-definice-a-vyuziti>
- [10] TWI. *WHAT IS DIGITAL TWIN TECHNOLOGY AND HOW DOES IT WORK?* [online]. [cit. 15. 3. 2022] Dostupné na: <https://www.twi-global.com/technical-knowledge/faqs/what-is-digital-twin>
- [11] Martin Dlouhý. (2019). *Simulační modely* [online]. [cit. 15. 3. 2022] Dostupné na: https://nb.vse.cz/~dlouhy/Simulace_K1_P1.pdf
- [12] Barbora Kodřousková. (2021). *Vývoj mobilní aplikace: průběh, distribuce, propagace.* [online]. [cit. 15. 3. 2022] Dostupné na: <https://www.rascasone.com/cs/blog/vyvoj-mobilni-aplikace-ios-android>
- [13] Barbora Kodřousková. (2021). *TYPY MOBILNÍCH APLIKACÍ, KTERÝ VYBRAT PRO VLASTNÍ VÝVOJ?* [online]. [cit. 23. 3. 2022] Dostupné na: <https://www.rascasone.com/cs/blog/typy-mobilnich-aplikaci>
- [14] Charter Global. *UNDERSTANDING THE 3 TYPES OF MOBILE APPS: NATIVE, MOBILE, AND HYBRID* [online]. [cit. 23. 3. 2022] Dostupné na: <https://www.charterglobal.com/understanding-the-3-types-of-mobile-apps-development-services/>

- [15] Keith Shields. (2021). *Native App vs. Web App vs. Hybrid App: Which Is Best?* [online]. [cit. 24. 3. 2022] Dostupné na: <https://designli.co/blog/native-app-vs-web-app-vs-hybrid-app-which-is-best/>
- [16] Kim Ang Nguyen. (2021). *TOP 7 BEST NATIVE APP EXAMPLE IN 2022 THAT MERCHANTS CAN LEARN FROM* [online]. [cit. 24. 3. 2022] Dostupné na: <https://magenest.com/en/native-app-example/>
- [17] Keith Shield. (2021). *5 Great Web Application Examples* [online]. [cit. 25. 3. 2022] Dostupné na: <https://designli.co/blog/5-great-web-application-examples/>
- [18] Katie Iannace. (2021). *5 Best Hybrid App Examples* [online]. [cit. 25. 3. 2022] Dostupné na: <https://designli.co/blog/5-best-hybrid-app-examples/>
- [19] Statcounter. (2022). *Mobile Operating System Market Share Worldwide* [online]. [cit. 30. 3. 2022] Dostupné na: <https://gs.statcounter.com/os-market-share/mobile/worldwide>
- [20] Erica Mixon. (2020). *Android OS* [online]. [cit. 30. 3. 2022] Dostupné na: <https://www.techtarget.com/searchmobilecomputing/definition/Android-OS>
- [21] Brien Posey. (2020). *Apple iOS* [online]. [cit. 6. 4. 2022] Dostupné na: <https://www.techtarget.com/searchmobilecomputing/definition/iOS>
- [22] Statcounter. (2022). *Desktop Operating System Market Share Worldwide* [online]. [cit. 6. 4. 2022] Dostupné na: <https://gs.statcounter.com/os-market-share/desktop/worldwide>
- [23] Martina Jiříčková, Štefan Ondek. *Agilní přístupy v projektovém řízení* [online]. [cit. 6. 4. 2022] Dostupné na: <https://amfg.ai/2019/03/28/industry-4-0-7-real-world-examples-of-digital-manufacturing-in-action/>
- [24] Team Invento. (2021). *Mobile App Development Process: Step-by-Step Guide [2021]* [online]. [cit. 7. 4. 2022] Dostupné na: <https://www.invento.com/insights/mobile-app-development-process/>
- [25] Destin Learning. (2019). *What are the steps of the software development lifecycle?* [online]. [cit. 7. 4. 2022] Dostupné na: https://www.youtube.com/watch?v=gNmrGZSGK1k&t=177s&ab_channel=DestinLearning
- [26] Ritesh Ranjan. (2021). *What is a Framework in Programming & Why You Should Use One* [online]. [cit. 8. 4. 2022] Dostupné na: <https://www.netsolutions.com/insights/what-is-a-framework-in-programming/>
- [27] Anastasiya Marchuk. *Native Vs Cross-Platform Development: Pros & Cons Revealed* [online]. [cit. 8. 4. 2022] Dostupné na: <https://www.uptech.team/blog/native-vs-cross-platform-app-development>
- [28] Hanna Shnaider. (2021). *Native vs Cross-Platform Mobile App Development 2022 | 9 Factors to Consider* [online]. [cit. 11. 4. 2022] Dostupné na: <https://fortyseven47.com/blog/mobile-app-development-2/>
- [29] blocshop. (2021). *Cross-platform mobile app development: Tools & frameworks for 2022* [online]. [cit. 11. 4. 2022] Dostupné na: <https://www.blocshop.io/blog/cross-platform-app/>
- [30] javinpaul. (2020). *7 Best Frameworks & Libraries for Cross-Platform Android and iOS Apps in 2022* [online]. [cit. 11. 4. 2022] Dostupné na: <https://medium.com/javarevisited/top-5-frameworks-to-create-cross-platform-android-and-ios-apps-in-2020-d02edf3d01f1>

- [31] Mo Ali. (2022). *REACT NATIVE VS. FLUTTER VS. XAMARIN: THE BEST HYBRID APP DEVELOPMENT FRAMEWORK IN 2022* [online]. [cit. 12. 4. 2022] Dostupné na: <https://www.tekrevol.com/blogs/react-native-vs-flutter-vs-xamarin/>
- [32] Kiran Beladiya. (2021). *Flutter Vs React Native vs Xamarin – Top Cross Platform Mobile App Development Framework* [online]. [cit. 12. 4. 2022] Dostupné na: <https://theonetechnologies.com/blog/post/flutter-vs-react-native-vs-xamarin-top-cross-platform-mobile-app-development-framework>
- [33] Monisha Boparai. (2020). *ReactNative Vs Flutter Vs Xamarin- Best Cross-Platform Development*. [online] [cit. 12. 4. 2022] Dostupné na: <https://www.orafox.com/reactnative-vs-flutter-vs-xamarin-best-cross-platform-development/>
- [34] TechTerms.com. (2011). *Programming language* [online]. [cit. 12. 4. 2022] Dostupné na: https://techterms.com/definition/programming_language
- [35] info.spsnome.cz. (2015). *Dělení programovacích jazyků* [online]. [cit. 12. 4. 2022] Dostupné na: <http://info.spsnome.cz/Programovani/Zaklady-Programovaci-jazyky>
- [36] Coding Mentors. (2018). *Compiler and Interpreter: Compiled Language vs Interpreted Programming Languages* [online]. [cit. 12. 4. 2022] Dostupné na: https://www.youtube.com/watch?v=I1f45REi3k4&ab_channel=CodingMentors
- [37] David Čápka. *Lekce 1 - Úvod do C# a .NET frameworku* [online]. [cit. 17. 4. 2022] Dostupné na: <https://www.itnetwork.cz/csharp/zaklady/c-sharp-tutorial-uvod-do-jazyka-a-dot-net-framework>
- [38] Alexander S. Gillis, Sarah Lewis. *object-oriented programming (OOP)* [online]. [cit. 17. 4. 2022] Dostupné na: <https://www.techtarget.com/searcharchitecture/definition/object-oriented-programming-OOP>
- [39] Ing. Petr Kalčev. *Úvod do databází* [online]. [cit. 17. 4. 2022] Dostupné na: <http://people.fsv.cvut.cz/~dlaskpet/Help/UvodDoDatabazi.pdf>
- [40] sites.google.com. *Základní pojmy databáze* [online]. [cit. 17. 4. 2022] Dostupné na: <https://sites.google.com/site/xvinformatika/database/zakladni-pojmy-database>
- [41] w3schools.com. *SQL Data Types for MySQL, SQL Server and MS Access* [online]. [cit. 17. 4. 2022] Dostupné na: <https://sites.google.com/site/xvinformatika/database/zakladni-pojmy-database>
- [42] materialy-krystof.wz.cz. *Relační databáze* [online]. [cit. 21. 4. 2022] Dostupné na: <http://materialy-kristof.wz.cz/informatika/obsah/odkazy/database.html>
- [43] Adam Kos. [2018] *Aplikace na Iphone, které vám pomohou osvojit si zdravé návyky* [online]. Dostupné na: <https://macpodpora.cz/aplikace-na-iphone/>
- [44] Freia Labo. [2017] *Google just made very subtle change to its Play Store logo and icons* [online]. Dostupné na: <https://mashable.com/article/google-play-store-new-logo>
- [45] C Sharp (C#) *Logo Vector* [online]. Dostupné na: <https://seeklogo.com/vector-logo/363285/c-sharp-c>
- [46] pngwing.com [online]. Dostupné na: <https://www.pngwing.com/en/free-png-zoupl>
- [47] Seven_au. [2017]. *Arduino* [online]. Dostupné na: <https://pixabay.com/cs/illustrations/arduino-arduino-uno-technika-2168193/>
- [48] Amazon.de. *Fischertechnik TXT* [online]. Controller Dostupné na: <https://www.amazon.de/Fischertechnik-522429-TXT-Controller/dp/B00J8G9IQ4>

- [49] Vývojáři NuGet balíčků (2022). *An introduction to NuGet* [online]. [cit. 26. 4. 2022] Dostupné na: <https://docs.microsoft.com/en-us/nuget/what-is-nuget>
- [50] Nuget.org. *Xamarin* [online]. [cit. 26. 4. 2022] Dostupné na: <https://www.nuget.org/profiles/Xamarin>
- [51] GemBox Ltd. *GemBox.Spreadsheet* [online]. [cit. 26. 4. 2022] Dostupné na: <https://www.gemboxsoftware.com/spreadsheet>
- [52] Guest B (2017). *Microcharts: Elegant Cross-Platform Charts for Every App* [online]. [cit. 26. 4. 2022] Dostupné na: <https://devblogs.microsoft.com/xamarin/microcharts-elegant-cross-platform-charts-for-any-app/>
- [53] Pixel perfect. *Humidity - Free weather icons* [online]. Dostupné na: https://www.flaticon.com/free-icon/humidity_728093
- [54] Milinda Courey. *File, format, xlsx icon* [online]. Dostupné na: https://www.iconfinder.com/icons/644129/file_format_xlsx_document_download_extension_icon
- [55] MicrovOne. *Stock ilustrace Sada vektorových ikon průmyslových a rostlinných budov* [online]. Dostupné na: <https://www.istockphoto.com/cs/vektor/sada-vektorov%C3%BDch-ikon-pr%C5%AFmyslov%C3%BDch-a-rostlinn%C3%BDch-budov-gm613345652-105889383>
- [56] Mamhure Alp. *Industry 4.0* [online]. Dostupné na: <https://thenounproject.com/icon/industry-40-2334636/>
- [57] Akash Khatri. *Noise Icon - Transparent Speaker* [online]. Dostupné na: https://www.kindpng.com/imgv/iJiibb_noise-icon-transparent-speaker-icon-png-png-download/
- [58] *Cold Room Temperatures - Store At Room Temperature Icon* [online]. Dostupné na: https://www.pngkit.com/view/u2r5a9i1y3r5r5y3_cold-room-temperatures-store-at-room-temperature-icon/
- [59] *Red warning icon* [online]. Dostupné na: <https://www.iconsdb.com/red-icons/warning-icon.html>
- [60] Freepik. *Factory - Free buildings icons* [online]. Dostupné na: https://www.flaticon.com/free-icon/factory_440696
- [61] USTUNDAG, Alp a Emre CEVIKCAN. *Industry 4.0: managing the digital transformation*. Cham: Springer International Publishing, [2018]. Series in advanced manufacturing (Springer). ISBN 978-3-319-57869-9.
- [62] NAKOV, S. *Fundamentals of computer programming with C# : The Bulgarian C# programming book*. Svetlin Nakov, [2013]. ISBN 9789544007737
- [63] SCMALZ, M. *Using databases with C#*. O'Reilly Media, [2011]. ISBN 9781449309985