

Implementace uživatelského rozhraní systému pro účastníky robotické soutěže

Pavel Ševčík

Bakalářská práce
2022



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
Ústav informatiky a umělé inteligence

Akademický rok: 2021/2022

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Pavel Ševčík**
Osobní číslo: **A19109**
Studijní program: **B3902 Inženýrská informatika**
Studijní obor: **Softwarové inženýrství**
Forma studia: **Prezenční**
Téma práce: **Implementace uživatelského rozhraní systému pro účastníky robotické soutěže**
Téma práce anglicky: **Implementation of User Interface for Robotic Tournament Event Management System**

Zásady pro vypracování

1. Prozkoumejte a popište systémy, používané pro správu robotických soutěží ve světě.
2. Prostudujte stávající uživatelské rozhraní a jeho implementaci v systému systému pro správu robotické soutěže FAI UTB ve Zlíně.
3. Vyberte platformu pro novou implementaci responzivního uživatelského rozhraní, které bude dobře ovladatelné na všech typech zařízení.
4. Implementujte nové funkce pro registraci soutěžících, týmů a robotů, správu uživatelského profilu, generování a zobrazení rozpisky soutěže („pavouka“), zobrazení výsledků atd. Věnujte také pozornost zabezpečení aplikace.
5. Vaši implementaci popište tak, aby bylo možné na základě tohoto popisu v budoucnu aplikaci rozšiřovat o další funkcionality.

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam doporučené literatury:

1. WAGNER, Gerd a Mircea DIACONESCU. Web applications with javascript or java: volume 2: associations and class hierarchies. 1. vyd. Boston: De Gruyter Oldenbourg, 2021. Webapp. ISBN 978-3-11-050024-0.
2. FRAIN, Ben. Responsive Web Design with HTML5 and CSS: Develop future-proof responsive websites using the latest HTML5 and CSS techniques, 3rd Edition. B.m.: Packt Publishing, 2020. ISBN 978-1-83921-156-0.
3. BANKS, Alex a Eve PORCELLO. Learning React: modern patterns for developing React Apps. Second edition. Beijing Boston Farnham Sebastopol Tokyo: O’Reilly, 2020. ISBN 978-1-4920-5172-5.
4. WILKEN, Jeremy. Angular in Action. 1st edition. Shelter Island, NY: Manning, 2018. ISBN 978-1-61729-331-3.
5. BOHNER, Michael. Ionic Framework: Building mobile apps with Ionic Framework. B.m.: CreateSpace Independent Publishing Platform, 2016. ISBN 978-1-5227-5352-0.

Vedoucí bakalářské práce: **Ing. Tomáš Dulík, Ph.D.**
Ústav informatiky a umělé inteligence

Datum zadání bakalářské práce: **3. prosince 2021**

Termín odevzdání bakalářské práce: **23. května 2022**

doc. Mgr. Milan Adámek, Ph.D. v.r.
děkan



prof. Mgr. Roman Jašek, Ph.D., DBA v.r.
ředitel ústavu

Ve Zlíně dne 24. ledna 2022

Prohlašuji, že

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – bakalářskou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne 22.5.2022

Pavel Ševčík v.r.

ABSTRAKT

Abstrakt česky

Tato práce se zabývá a pojednává o robotických soutěžích konaných na území České republiky a ve světě. Rozebírá se registrace, provedení soutěže a jejich celkový chod. Největší část práce je cílena zaměřením na ROBOGAMES na FAI. V praktické části je řešena problematika implementace a programování nové webové stránky.

Klíčová slova:

Robotická soutěž, Robot, Soutěž, Angular, Ionic

ABSTRACT

This theses is dealing with researching robotic tournamets in the Czech republic and in other countries. The research focuses on the form of registration of new competitors, organizing and course of the competitions. Largest part of this thesis is targeting robotic tournament called ROBOGAMES which takes place in the Faculty of applied informatics of Tomas Bata University in Zlín. The practical sector of this thesis deals with implementation of a new web application for this competition.

Keywords:

Robotic competetion, Robot, Competetion, Angular, Ionic

Děkuji panu Ing. Tomáši Dulíkovi, Ph.D. za vedení a konzultace této práce, taktéž za jeho fyzickou i morální podporu v časech, kdy vše nešlo, tak jak by mělo a nastaly komplikace v osobním životě. Taktéž děkuji kolegům slečně Elišce Obadalové a panu Martinu Krčmovi, za skvělou spolupráci při řešení tohoto společného rozsáhlého projektu, který probíhal skoro bez komplikací. Zároveň děkuji i mé rodině, která mě podporovala při tvoření této bakalářské práce a prožívala semnou nelehké období v těchto časech.

Prohlašuji, že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

OBSAH

ÚVOD	9
I TEORETICKÁ ČÁST	10
1 PRŮZKUM ROBOTICKÝCH SOUTĚŽÍ V ČR A VE SVĚTĚ	11
1.1 ROBOTICKÉ SOUTĚŽE V ČESKU.....	11
1.1.1 Robotická soutěž v Přerově.....	11
1.1.1.1 Řešení registrace	11
1.1.2 Robotická soutěž v Jedovnicích	11
1.1.2.1 Řešení registrace	12
1.1.3 Robotická soutěž v Praze na ČVUT	12
1.1.3.1 Řešení registrace	12
1.1.4 Robotiáda v Brně.....	13
1.1.4.1 Řešení registrace	13
1.2 ROBOTICKÉ SOUTĚŽE VE SVĚTĚ.....	13
1.2.1 Robotic Tournament v Polsku.....	13
1.2.1.1 Řešení registrace	14
2 PRŮZKUM ROBOGAMES NA FAI	15
2.1 VĚKOVÉ KATEGORIE SOUTĚŽÍCÍCH.....	15
2.1.1 Žáci.....	16
2.1.2 Studenti středních škol	16
2.1.3 Dospělí	16
2.2 KATEGORIE ROBOTŮ	16
2.2.1 Robosumo	16
2.2.2 Mini Robosumo.....	17
2.2.2.1 Kategorie lego.....	17
2.2.2.2 Kategorie MCU.....	18
2.2.3 Sledování čáry	18
2.2.3.1 Kategorie Lego	20
2.2.3.2 Kategorie MCU.....	21
2.3 ŘEŠENÍ DOPOSUD	21
2.4 NOVÁ KONCEPCE ŘEŠENÍ.....	24
3 VÝBĚR PLATFORMY	25
3.1 ANGULAR.....	26
3.2 IONIC.....	27
II PRAKTICKÁ ČÁST	28
4 ÚVOD	29
5 PŘIHLÁŠENÍ	30
5.1 POŽADAVKY	30
5.2 IMPLEMENTACE.....	30
5.2.1 OAuth2	32
5.3 DOMOVSKÁ STRÁNKA	35
6 REGISTRACE	37

6.1	POŽADAVKY.....	37
6.2	IMPLEMENTACE.....	37
6.2.1	Formulář.....	37
7	SPRÁVA UŽIVATELSKÉHO ÚČTU	38
7.1	POŽADAVKY.....	38
7.2	IMPLEMENTACE.....	38
8	SPRÁVA TÝMŮ.....	41
8.1	POŽADAVKY.....	41
8.2	IMPLEMENTACE.....	41
9	ROBOTI.....	44
9.1	POŽADAVKY.....	44
9.2	IMPLEMENTACE.....	44
10	ROZPISY ZÁPASŮ SKUPIN	47
11	POTVRZOVÁNÍ ROBOTŮ.....	49
	ZÁVĚR	50
	SEZNAM POUŽITÉ LITERATURY.....	51
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	54
	SEZNAM OBRÁZKŮ	55

ÚVOD

K tématu této bakalářské práce mě dovedly dlouholeté zkušenosti se soutěžením v robotických disciplínách, kterým se věnuji již od druhého ročníku střední školy až do nynější doby. Po nástupu na FAI UTB, kdy jsme se seznámili společně s Martinem Krčmou s panem doktorem Ing. Tomášem Dulíkem, PhD. a zjistili jsme, že se známe již dříve z robotických soutěží, kterých jsme se dohromady účastnili, tak jsme diskutovali, jak pozvednout úroveň Robogames na FAI UTB, které jsou již nyní na vysoké úrovni a po dlouhých debatách jsme dospěli k závěru, že by bylo vhodné udělat nové webové stránky, které by zajišťovali i registraci jinak než přes email, či Google formuláře, jako bylo řešeno doposud, či je řešeno u ostatních robotických soutěží v České republice. Dohodnuli jsme se tedy, že navážeme, na již rozdělanou práci pana Ing. Kříže [12], která však nebyla dokončena z důvodu rozsáhlé obtížnosti, a proto jsme si tuto problematiku rozdělili na tři části s mými kolegy, a to konkrétně se Eliškou Obadalovou a Martinem Krčmou, kdy každý řešíme určitou část problematiky tohoto webového rozhraní, které je responzivní a má být intuitivní. V mé části bakalářské práce řeším v teoretické části rozbor a průběh soutěží na domácí a světové scéně, kdy kladu hlavně důraz na soutěže, kterých jsme se aktivně s mým kolegou Martinem Krčmou účastnili, ať můžu popisovat soutěže co nejpřesněji mým osobním pohledem. Přičemž největší důraz byl kladen na ROBOGAMES na FAI UTB, kterého se celá bakalářská práce týká. V praktické části řeším implementaci uživatelského rozhraní systému pro účastníky robotických soutěží.

I. TEORETICKÁ ČÁST

1 PRŮZKUM ROBOTICKÝCH SOUTĚŽÍ V ČR A VE SVĚTĚ

Tato kapitola se zabývá srovnáním řešení a pořádání robotických soutěží v České republice a ve zbytku světa. Největší část je věnována ROBOGAMES na FAI. Rozhodl jsem se psát o robotických soutěžích, které znám osobně a mám s nimi víceleté zkušenosti až na výjimku Robotiády v Brně, které jsem se neúčastnil jako soutěžící ale pouze v podobě mentora a lektora dětí z kroužku ve kterém s kolegou Martinem Krčmou vyučujeme.

1.1 Robotické soutěže v Česku

Registrace na robotické soutěže v České republice jsou valnou většinou řešeny pomocí emailů nebo formulářů přes Google.

1.1.1 Robotická soutěž v Přerově

Tato soutěž se odehrává v moravském městě Přerov a oficiální název nese ROBOTIX a pořádá jej Střední průmyslová škola Přerov. V letošním roce probíhal již 4. ročník soutěže. Datum pořádání bývá většinou v lednu nebo v únoru. Soutěží se zde v disciplínách: Lego stopař, což je ekvivalent soutěže Lego line follower a je zde taktéž tato kategorie rozdělena do podkategorií Lego a MCU. Další disciplína je zde Lego sumo. Soutěž se odehrává v nové tělocvičně u školy. Po příchodu je provedena kontrola registrace a každý robot dostane své startovní číslo pod kterým je veden v tabulkách. Okolo soutěžních drah jsou rozmístěny stoly, na kterých můžou soutěžící zpravovat a vylepšovat své roboty. [13]

1.1.1.1 Řešení registrace

Na této soutěži je registrace řešena pomocí Google formuláře, přes který soutěžící vyplní údaje o týmu, kategoriích, ve kterých soutěží a robotech.

1.1.2 Robotická soutěž v Jedovnicích

Tato soutěž se také odehrává v moravském městě, a to v Jedovnicích. Oficiální název zní JedoBot. Konání této soutěže je na rozdíl od většiny posunuta až na květen, nevýhodou tohoto termínu je avšak, že se mnohdy kryje s maturitami a žáci čtvrtých ročníků maturitních oborů na něj tak často z tohoto důvodu nejezdí. Například i nám se v maturitním období kryla s termínem praktické zkoušky. Z tohoto důvodu je z mého hlediska ideálnější řešení v lednu nebo v únoru, kdy se termín soutěže nekryje s maturitami či jiným obdobím zaměřeným pro střední školy či děti navštěvující základní školy. Věkové kategorie jsou zde

rozděleny do základních škol, středních škol a kategorie open v níž je jsou zařazeni studenti vysokých škol a veřejnost. V případě malé naplnění kategorie může dojít ke sloučení. Z řad disciplín je zde nabídka širší než na mnoha jiných soutěžích. Kategorie, ve kterých se lze účastnit jsou: Lego sumo, Lego mini Sumo, sledovač čáry, který se dělí do kategorií Lego a MCU. Dále čárové bludiště, nově přidaná kategorie robo káry a Freestyle – tato kategorie je známá spíše ve světě než u nás a jedná se v ní o to, že soutěžící může postavit jakéhokoliv robota, kde fantazie není omezena a následně porota organizátorů a veřejnost hodnotí roboty a svými hlasy vybírají vítěze. Na této soutěži probíhá i seminář, kterého se lze zúčastnit a dozvědět se o novinkách, popřípadě se přiučit programování. Škola nabízí i nocleh pro účastníky.[14]

1.1.2.1 Řešení registrace

Předchozí ročníky byly řešeny pomocí emailové registrace, kdy organizátor zaslal Excelovský soubor, kde byly předvyplněné kolonky a účastník dopsal informace, které následně zaslal zpět emailem. Letošní ročník již pořadatelé přešli na variantu registrace skrze Google formulář.

1.1.3 Robotická soutěž v Praze na ČVUT

Tato soutěž je na rozdíl od všech ostatních velmi specifická. Nejsou zde předem stanoveny kategorie, ve kterých se soutěží, ale téma zadání se každý rok mění a je odlišné. Zadání je zveřejněno několik měsíců dopředu aby byl dostatek času na to postavit funkčního robota. Věkové kategorie jsou zde rozděleny na první stupeň základních škol, druhý stupeň základních škol a střední školy. V každé kategorii jsou čtyři skupiny rozdělené na čtyři dny. Z těchto skupin postupují první čtyři týmy do finálového dne, kde se utkají o celkové vítězství. Kategorie pro střední školy je ozvláštněna o to, že do finálového dne jsou přidáni účastníci z řad studentů vysoké školy FEL ČVUT a celá soutěž je tímto sřížena. Dění se odehrává v místech katedry FEL ČVUT v Praze. Termíny robosoutěže pro žáky základních škol se odehrávají v dubnu a pro studenty středních škol v listopadu.[15]

1.1.3.1 Řešení registrace

Přihlašování je zde řešeno pomocí formuláře na stránkách soutěže, který si škola vytvořila sama. Obsah formuláře je složen z informací o soutěžích, kontaktu na školu, týmu, pedagoga a velikosti triček, které soutěžící zde za start obdrží. Dále je zde možnost zapůjčení Lego stavebnic, kde je na výběr EV3 nebo NXT, které zasílají poštou a vrací se na soutěži.

1.1.4 Robotiáda v Brně

Robotiáda v Brně je do počtu účastníků jedna z největších v České republice. V minulém ročníku se účastnilo více než 190 týmů a více než 700 účastníků, i z tohoto důvodu je počet týmů na školu limitován. Soutěží se zde v 9 disciplínách: Line follower, Lego sumo, Suo MCU, Sprint – tato disciplína taktéž není v Česku moc známá a je přezvána z ciziny, kde se těší velké popularitě. Úkolem je projet co nejrychleji 10 metrů dlouhou dráhu, která je rovná. Inspirací této soutěže jsou americké závody sportovních aut. Tato kategorie se taktéž dělí na roboty z Lega a roboty řízené MCU jednotkou. Dálkový medvěd, kde je za úkol pomoci robotu, který může být ovládán dálkovým ovladačem najít a zachránit medvěda. Je zde taktéž kategorie autonomního robota, kdy se nesmí používat dálkový ovladač, ale robot musí být plně autonomní a musí se rozhodovat sám na základě naprogramovaného kódu. Freestyle a Freestyle ze stavebnice WeDo. Tato stavebnice je volně prodejný kit, který je doporučený pro začátečníky a pohybuje se cenově kolem 1000 korun, tento set obsahuje všechny potřebné součástky pro začátek a naučení se s robotikou. Věkové kategorie jsou zde řešeny v rozmezích: 0-15 let, kam spadají žáci základních škol a 15-19 let, kam spadají studenti středních škol. [16]

1.1.4.1 Řešení registrace

Zde je registrace řešena pomocí formuláře na stránkách soutěže. Vyplňují se zde informace o soutěžících, jako je například jméno, věk, kategorie, které se účastní, škola, za kterou soutěží, či jméno zodpovědného vedoucího týmu. Je zde omezen počet týmů na jednotlivou školu v poměru 5 týmů z důvodu vysoké účasti soutěžících.

1.2 Robotické soutěže ve světě

Velikost robotických soutěží ve světě a v České republice je velikostí diametrálně na jiné úrovni. Pro porovnání si vezmeme například robotickou soutěž v Polském městě Rybnik, které se již několik let aktivně zúčastňujeme. Tato soutěž se nazývá taktéž jako mistrovství Evropy a je největší v Evropě.

1.2.1 Robotic Tournament v Polsku

Porovnávat České soutěže s Evropskými soutěži je prozatím jako házet hrách na zeď. V cizině tomuto odvětví věnují větší prostředky a časové úsilí než v našich krajích. I když postupe času, co se zde soutěže těší větší popularitě, tak zájem a úroveň se začíná zvětšovat i

zde. Tato soutěž se odehrává v Polském městě Rybnik asi hodinu od hranic. Letošního ročníku se účastnilo 98 týmů, 246 účastníků a 268 robotů z celé Evropy. Po Covidové pauze se jednalo o první ročník, což se podepsalo i na počtu účastníků, kdy za normálních okolností všechny počty registrací jsou téměř pětinasobné a zastoupení soutěžících ze zemí bývá o hodně širší. Soutěžních kategorií tato soutěž nabízí mnohem více než jakákoliv jiná soutěž. Zápasí se zde v: Line followeru, kde jsou opět kategorie LEGO a kategorie MCU. Kategorie Sumo zde čítá více odvětví: klasické Lego sumo, Mini sumo, které se dělí na Lego kategorii a kategorii s MCU, Micro sumo – tento robot může mít rozměry 50mm x 50mm, jeho maximální váha může být 100 gramů a Nano Sumo – zde jsou rozměry ještě menší, a to 25mm x 25mm a váha je stanovena maximálně na 25 gramech. Rozměry v těchto kategoriích jsou opravdu malé a roboti jsou řízeni pomocí MCU, jelikož Lego takto malé řídicí jednotky nenabízí. RoboDragRace a RoboDragRace MCU, RoboStrong – zde jde o to, že dva roboti jsou k sobě připoutáni řetězem a snaží se přetáhnout druhého robota ven ze stanoveného hřiště. Poslední disciplínou je zde Freestyle, který se již objevuje i v soutěžích v České republice ale roboti jsou zde o mnoho kreativnější. Výběr vítěze probíhá stejně, jako u nás skrze porotu organizátorů, hlasů soutěžících a veřejnosti. [17]

1.2.1.1 Řešení registrace

Opět narážíme na fakt, že soutěže v cizině jsou lépe řešeny. Přihlašování zde probíhá přes jejich webovou stránku, kde vyplníte informace o sobě, týmu, vyberete kapitána týmu a zvolíte kategorie ve kterých chcete soutěžit. Na rozdíl od řešení registrací u nás se Vám vytvoří účet, do kterého se můžete skrze přihlašovací údaje přihlásit a upravovat si svou registraci, jako například měnit členy týmu, přidávat roboty či odebírat ty, které již máte přihlášené

2 PRŮZKUM ROBOGAMES NA FAI

Robogames, kterou pořádá Fakulta Aplikované Informatiky pod Univerzitou Tomáše Bati se řadí k jedné z největších robotických soutěží v České republice. Této soutěže se účastní nejen studenti ale i učitelé z celé České republiky. Letos se koná již 4. ročník, přičemž první ročník soutěže se konal již v roce 2017. Průběh soutěže na dva roky pozdržela a přerušila pandemická opatření způsobené šířením Covidu-19. Pokud by soutěž nebyla pozdržena skrz koronavirové opatření, tak by soutěž již měla oslavený jubilejní 5. ročník soutěže a v tomto roce by se konal 6.ročník soutěže.

Soutěž se každoročně pořádá na přelomu ledna a února, většinou se snaží pořadatelé umístit den konání na den pololetních prázdnin, aby se zvýšila šance pro děti a mohli přijet. Taktéž je tento den vybírán z důvodu, aby nebyla zameškávána školní docházka soutěžících. Průběh probíhá následovně: Před hlavními vchodovými dveřmi fakulty je stolek s registrací, kde sedí příslušný tým rozhodčích, kteří týmy soutěžících uvítají, zkontrolují správnost registrovaných, ověří, zdali spadají do příslušné kategorie a nakonec přeměří a převáží každého z robotů, se kterým se daný tým chystá soutěžit. Každý z robotů musí splňovat příslušné požadavky jako jsou například omezená váha, rozměry, či nesmí obsahovat nepovolené komponenty pro danou kategorii ve které soutěží [2]. Po úspěšném splnění všech parametrů se soutěžící se svými roboty přesunou dále do foyer, kde jsou rozmístěny příslušné stanoviště pro soutěžící. Nejbližší vchodovým dveřím bývají umístěny ringy pro robosumo. V další části jsou umístěny dráhy pro line-followery, micromouse a robota uklízeče. Kolem drah jsou rozmístěny stoly se židlemi, kde mají týmy své zázemí pro úpravy, stavění a ladění svých robotů. Místa jsou týmům taktéž vyhrazena v přilehlých učebnách u foyer. Týmy jsou rozděleny do skupin, popřípadě pavouků, kde mají určené, s kým se setkají a svádí své boje o postup a vítězství. Po dobojování posledních finálových kol se všichni přesouvají do velké přednáškové místnosti, kde se koná vyhodnocení soutěže, které vede pan děkan doc. Ing. Jiří Vojtěšek, PhD. a pan doktor Ing. Tomáš Dulík, PhD.

2.1 Věkové kategorie soutěžících

Soutěž je členěna do různých věkových kategorií, aby byly spravedlivé podmínky pro všechny soutěžící a podmínky byly co nejvíce spravedlivé a vyrovnané.

Soutěží se v těchto věkových kategoriích.

2.1.1 Žáci

Věk soutěžícího je omezen na maximálně 16 let a kategorie je určena pro žáky základních škol [1]

2.1.2 Studenti středních škol

Tato kategorie je určena pro věkové rozmezí 16-19 let a spadají do ní studenti středních škol či učňovských oborů [1]

2.1.3 Dospělí

V kategorii dospělých věková hranice soutěžícího není nijak limitována, ale její spodní hranice začíná na 19 letech. Tato kategorie je určena pro studenty vysokých škol, popřípadě pro učitele či každého dospělého, který chce zkusit vyzkoušet svého robota a poměřit síly [1]

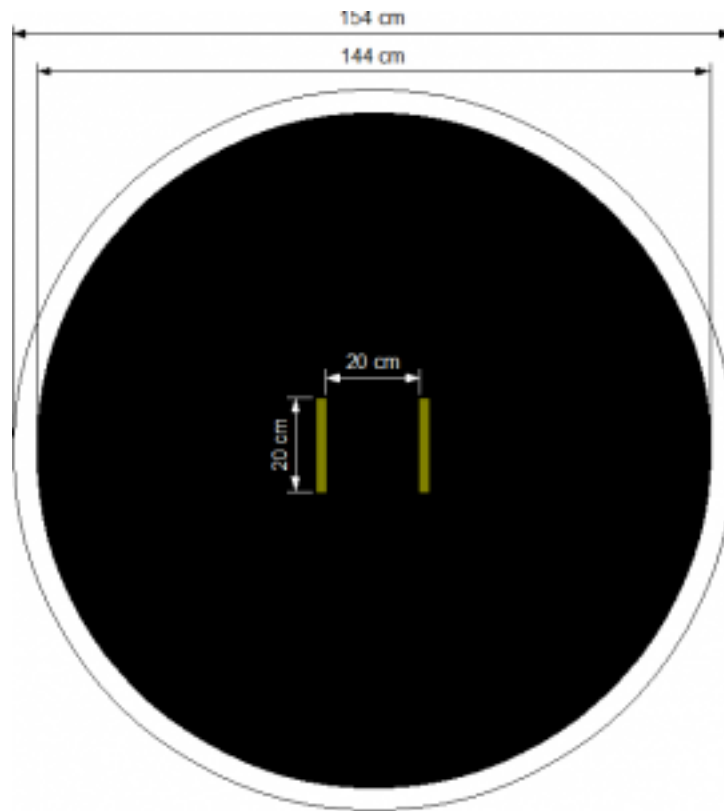
2.2 Kategorie robotů

Soutěžní kategorie robotů se dělí do těchto druhů.

2.2.1 Robosumo

Zápas probíhá tak, že proti sobě stojí dva roboti v kruhu, který je ohraničený bílým okrajem. V tomto místě má kruh rozměr 154 cm. Vnitřek kruhu je černý, rozměry černého pole jsou 144 cm. Cílem robota je vytlačit druhého robota z kruhu stejně jako v případě lidského suma. Souboj se koná maximálně celkem na 3 kola, přičemž vítězem se stává robot, který jako první získá dva body, to znamená, který robot jako první vytlačí druhého robota dvakrát z kruhu ven, jako první. Začátek souboje probíhá tak, že se roboti postaví k sobě do středu kruhu zády. Po odstartování musí 5 sekund nehybně stát a po tomto časovém úseku se každý z robotů musí otočit nejméně o 90 stupňů. Robot musí být autonomní, to znamená, že robot musí být naprogramovaný tak, aby vše prováděl sám a nesmí být řízen jakýmkoliv ovládačem či mobilním telefonem. Každé kolo trvá maximálně 3 minuty. Pokud souboj není v tomto časovém limitu ukončen je souboj považován za remízu. V případě, kdy se odehrává třetí kolo a stav je vyrovnaný 1:1, tak může rozhodčí nařídít prodloužení nebo přidělit vítězný bod za větší aktivitu robota, který vyvíjel větší činnost a byl aktivnější.

Maximální váha robota může být 1 kg. Rozměry robota jsou limitovány 25 cm na šířku a 25 cm na délku. Výška robota není limitována.[4]



Obrázek 1- Ring pro kategorii Lego Sumo [4]

2.2.2 Mini Robosumo

Mini Robosumo má stejná pravidla, jako jeho již zmiňovaná disciplína Robosumo, avšak se liší v pár detailech. Ring, ve kterém roboti zápasí má tentokrát rozměr pouze 77 cm v největším místě. Dále se tato kategorie na rozdíl od té větší dělí na dvě podkategorie, ve které se mohou používat jiné součástky a hlavní řídicí jednotka [6].

2.2.2.1 Kategorie lego

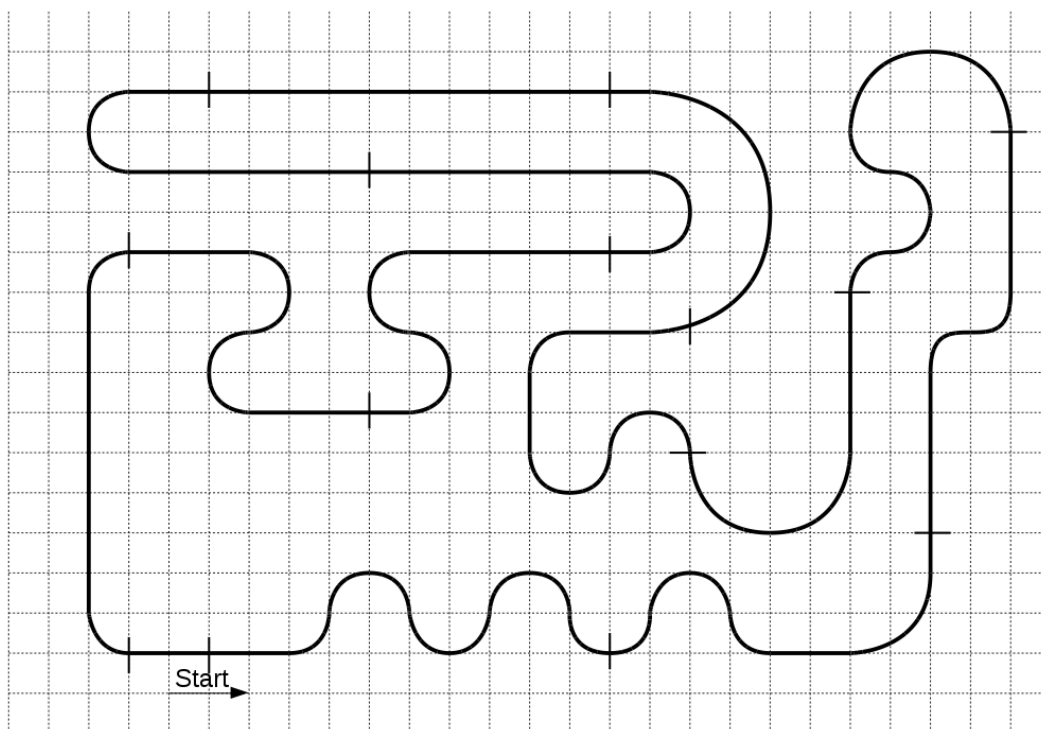
Zde jsou povoleny používat pouze součástky spadající pod Lego. Nesmí být použity žádné jiné komponenty. Rozměry robota mohou být 15 centimetrů na délku a 15 centimetrů na šířku. Výška opět není omezena.

2.2.2.2 Kategorie MCU

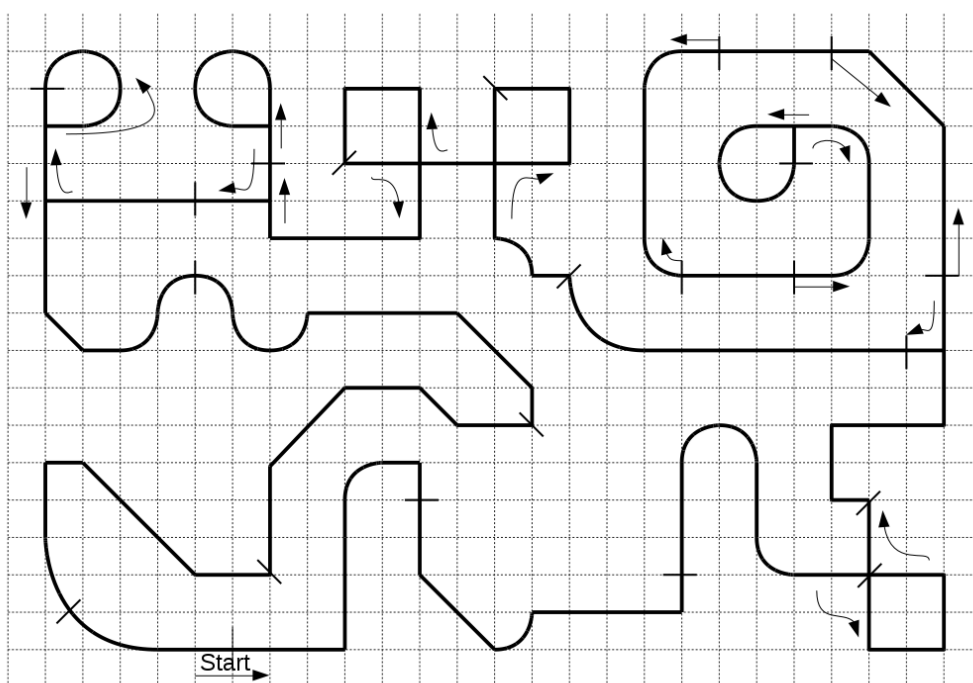
Tato kategorie je od Lego suma odlišná vtom, že zde mohou být používány jakékoliv komponenty a není omezení pouze na Lego součástky. Jako řídicí jednotka bývá velmi často používáno Arduino, Atmega či vývojové kity od firmy NXP. Rozměry jsou však menší než v předešlé kategorii a jsou 10 cm na šířku a 10 cm na délku. Výška opět není omezena.

2.2.3 Sledování čáry

Princip této soutěže spočívá v tom, že úkol robota je projet dráhu za co nejrychlejší čas a nevyjet mimo černou čáru. Dráha je vyrobena z bílého podkladu (nejčastěji z plastové folie nebo papíru), kde je předtištěna nebo nalepena stanovená trasa, kterou robot musí následovat a pomocí ní projet od startu až do cíle. Dráha má rozměry 280 x 200 cm. Velikost robota je omezena na 25 cm na délku a 25 cm na šířku, výška zde opět není nijak omezena. Konstrukce jsou, avšak velmi často menších rozměrů, protože takto velká konstrukce by mohla přidělávat problémy, jako je například nepřesnost či zpomalení robota. Váha je omezena na limit 1 kg. Zde platí taktéž nepsané pravidlo, kdy se roboti snaží být co nejlehčí z důvodu rychlejšího projetí dráhy. Tato disciplína se též dělí na zpracování konstrukce robotů z Lego součástek a MCU. Počet pokusů zde není limitován [7].



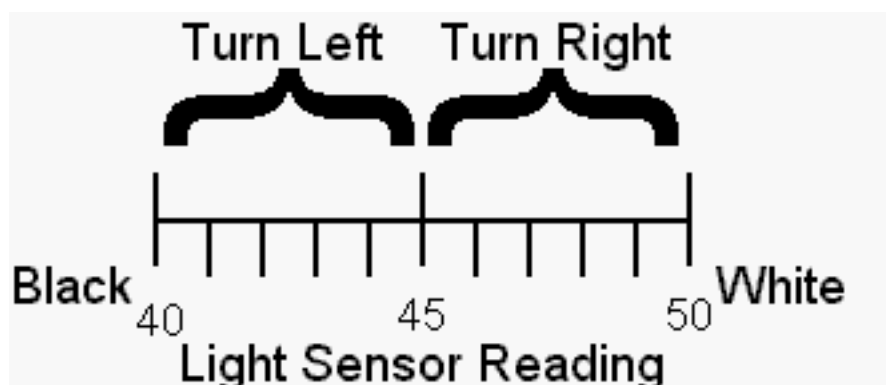
Obrázek 2 - Příklad dráhy pro sledovače čáry 1 [7]



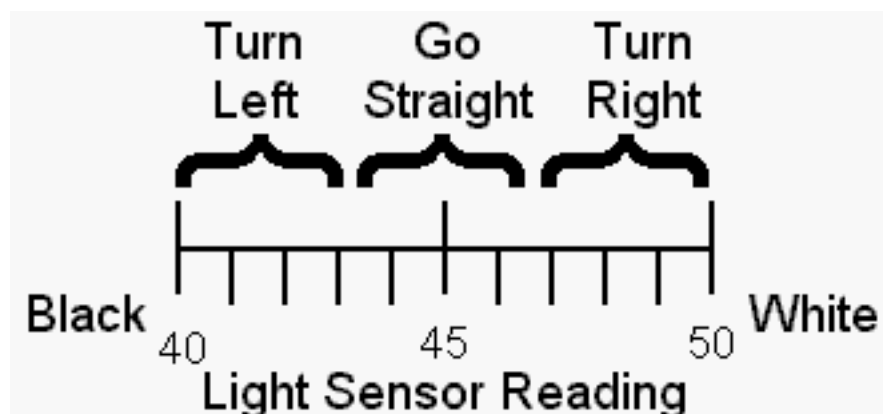
Obrázek 3 - Příklad dráhy pro sledovače čáry 2 [7]

2.2.3.1 Kategorie Lego

Zde mohou být použity pouze součásti ze stavebnice Lego a řídicí kostka musí být taktéž ze sady Lego. Řídicí kostky se používají ze série Lego Mindstorms, EV3 nebo nejnovější edice Inventor. Roboti, kteří jsou sestaveni z Lega bývají pomalejší než roboti, kteří jsou řízeni MCU jednotkou. Pro sledování čáry se používá Light sensor, který je součástí těchto stavebnic. Nejrozšířenějším případem je stav, kdy má robot pouze jedno čidlo, toto řešení je lehčí na programování, avšak je méně přesné a pomalejší. Druhá varianta je řešení pomocí více senzorů, kdy se využívá případ PID regulátoru, což je složení dvou nebo více čidel, které spolu spolupracují [10]. Toto řešení je rychlejší a přesnější, bohužel však složitější skrze programovou část a mnoho začínajících závodníků se do něj nepouští z důvodu programování v obrazovém editoru, kdy je téměř nereálné docílit správného naprogramování kvůli omezeným funkcím obrázkových bloků.



Obrázek 4 - PID regulátor pomocí dvou čidel [10]



Obrázek 5 - PID regulátor pomocí tří čidel [10]

2.2.3.2 Kategorie MCU

Ve variantě MCU se mohou používat jakékoliv součástky. Nejčastěji se využívá stejně jako v případě suma Arduino, Atmega a NXP. Prodávají se však i sety speciálně vymyšleny pro tuto kategorii. Roboti, kteří jsou v kategorii MCU jsou rychlejší než roboti z kategorie Lego.

2.3 Řešení doposud

Doposud je registrace řešena pomocí Google formuláře, ve kterém soutěžící vyplní údaje na o sobě například jméno, příjmení, věkovou kategorii a disciplíny ve kterých soutěží. Dotazník poté odešlou a data se zobrazí organizátorům, kteří s informacemi dále pracují při sestavování pavouků, tabulek a sběru dat. Informace o soutěži jsou zobrazeny na webové stránce <https://robogames.utb.cz>. Na této stránce se nachází údaje o datu konání, pravidla, disciplíny, organizační pokyny, odkaz pro registraci, kontakty a data z předchozích ročníků (jména vítězných týmu, fotky). Oznámení o nadcházejícím ročníku je realizováno pomocí zasílání emailů na adresy předchozích registrací účastníků, propagaci na Facebooku nebo Instagramu.

Robogames 2022 - registrace

Univerzita Tomáše Bati ve Zlíně, Fakulta aplikované informatiky, IČ: 70883521, se sídlem Nad Stráněmi 4511, Zlín (dále jen „UTB“) je správcem níže uvedených osobních údajů. UTB jako správce osobních údajů, zpracovává tyto osobní údaje za účelem přihlášení (registrace) do soutěže Robogames 2022 a poskytnutí informací týkajících se soutěže Robogames 2022. UTB bude tyto osobní údaje zpracovávat po dobu konání soutěže Robogames 2022. Po skončení soutěže budou tyto osobní údaje zničeny. Subjekt údajů má ve vztahu k osobním údajům práva, která jsou blíže specifikována v Zásadách ochrany osobních údajů. Tyto zásady jsou dostupné na webových stránkách UTB v sekci Ochrana osobních údajů (<https://www.utb.cz/ochrana-osobnich-udaju/>).

* Povinné

Souhlas se zpracováním osobních údajů a s pořizováním audiovizuálních záznamů

1. Přihlášením na akci souhlasíte

- se zpracováním Vašich osobních údajů, které zadáte do tohoto formuláře,
- s pořizováním fotografií a zvukových a zvukově obrazových záznamů osob Vašeho týmu a jeho robotů. Pořízené fotografie a audiovizuální záznamy mohou být pořadatelem použity pro účely propagace pořadatele, soutěže a podobné účely, a to bezplatně. *

Ano

Ne

Další

Stránka 1 z 3

Nikdy neprozrazujte svoje heslo. [Nahlásit porušení pravidel](#)

Obrázek 6 - Vzhled aktuálního formuláře pro přihlašování slide 1 [33]

Robogames 2022 - registrace

* Povinné

Údaje o vašem týmu

2. Název týmu
Pokud nebude vyplněn, budeme při soutěži jako název týmu uvádět Vaše příjmení.

3. Jméno a příjmení soutěžícího/soutěžících *

4. Email *

Emailová adresa jednoho člena týmu

5. Věková kategorie *

žáci: < 16 let

studenti SŠ: 16 - 19 let

dospělí: > 19 let

6. Zde vyberte soutěže, kterých se chcete zúčastnit. Můžete si vybrat i více soutěží.
Popisy soutěží a pravidla jsou na <http://robogames.utb.cz> *

Robosumo (25x25)

Mini robosumo LEGO (15x15)

Mini robosumo MCU (10x10)

Robot uklízeč varianta 1

Robot uklízeč varianta 2

Sledování číry LEGO

Sledování číry MCU

7. Název školy, kterou navštěvujete. *

8. Účastníte se seriálu soutěží Robo liga a souhlasíte s předáním Vašeho jména, příjmení, školy a Vašich výsledků v soutěži Robogames ke zpracování do seriálu soutěží Robo liga ? *

Ano

Ne

9. Souhlasíte s uchováním osobních údajů po ukončení soutěže ? *

Vaše osobní údaje budou uchovány pro potřeby organizátora soutěže - UTB ve Zlíně, abychom Vás mohli kontaktovat následující rok.

Ano

Ne

Kopii odpovědí si můžete po odeslání vytisknout

Stránka 1 z 1

Obrázek 7 - Vzhled aktuálního formuláře pro přihlašování slide 2 [33]

2.4 Nová koncepce řešení

Snaha o realizaci nového webu, zde již jednou proběhla, kdy na ní pracoval pan inženýr Kříž. V určité koncepci vycházíme z jeho předlohy, kdy však práce nebyla dotažena do finální verze, aby mohla být spuštěna. Proto jsme se s kolegou panem Martinem Krčmou, se kterým již dlouhodobě jezdíme na robotické soutěže a zúčastnili jsme se všech soutěží v České republice a mezinárodní soutěže v Polsku, která je brána jako Mistrovství Evropy, kterého se účastní týmy z celé Evropy, využít naše získané zkušenosti od roku 2016 a udělat nový web, který bude nabízet veškeré funkce pro správu Robogames. Do našeho týmu se přidala i kolegyně Eliška Obadalová, která se návrhu a implementaci nového webu také podílela. Jelikož se rovná o rozsáhlý projekt, na kterém je spousta práce, tak jsme se rozhodli práci po konzultaci s našim vedoucím práce, panem Ing. Tomášem Dulíkem, Ph.D. rozdělit práci na více částí, aby každý z nás mohl uplatnit své vědomosti a práce měla kvalitu a mohla reprezentovat tak významnou robotickou soutěž, jako je Robogames na FAI. Kolega Martin Krčma se ujmul práce na serverové části, na kterém naše aplikace běží a komunikuje s ní. Pro řešení serveru byla zvolena Java Spring framework. [11]

Kolegyně Eliška Obadalová řeší především uživatelské rozhraní pro organizátory, které není dostupné pro závodníky soutěže. Dále je součástí její problematiky generování diplomů pro účastníky. [18]

V mé části praktické části bakalářské práce řeším problematiku a implementování registrace soutěžících formou emailové adresy. V další části si soutěžící registruje svůj tým, roboty a může provádět správu již zadaných údajů, jako například změnu členů týmu, přidání či odebrání robotů v různých disciplínách. V části pro organizátory má práce obsahovat vytváření a generování rozpisů, skupin a pavouků pro konkrétní disciplíny.

Finální webová aplikace bude nasazena pro příští ročník Robogames, který bude symbolicky jubilejním 5. ročníkem.

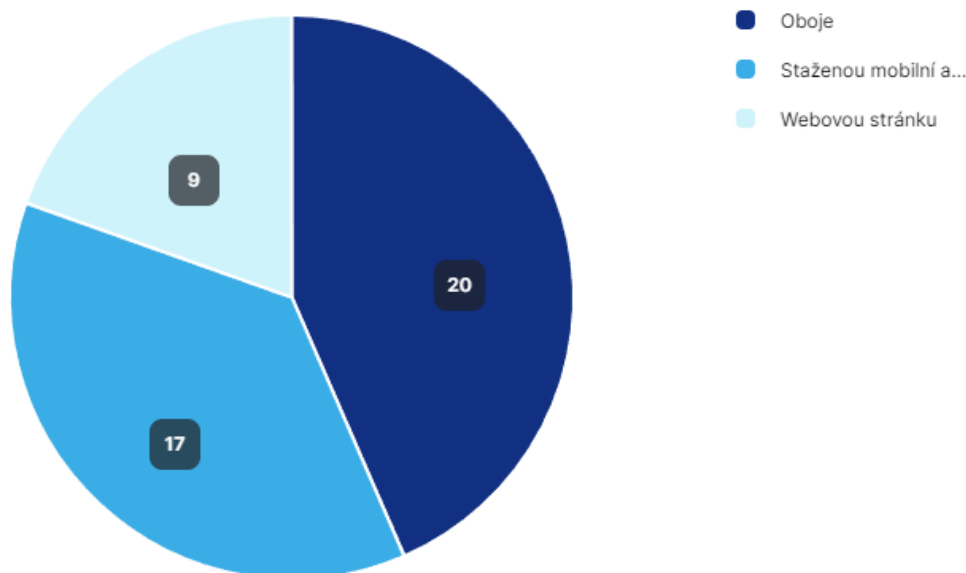
3 VÝBĚR PLATFORMY

Pro svou praktickou část bakalářské práce jsem si zvolil framework Ionic a v něm konkrétně Angular. Důvodem zvolení tohoto frameworku bylo více aspektů. První z aspektů, které mě vedli k výběru Angularu bylo to, že jsme se s ním učili ve škole, a to konkrétně na hodinách Mobilních aplikací, kde cvičení vedl pan Ing. Jakub Josef Forman a přednášky vedl pan Ing. Radek Vala, PhD. Oba tito učitelé dokázali skvěle podat problematiku této látky jak pan Ing. Forman v praktické části, tak i pan doktor Vala po teoretické části na přednáškách, a tak nás tento framework zaujal. Jedním z požadavků naší bakalářské práce bylo vytvořit responzivní aplikaci Angular patří k nejrozšířenějším nynějším frameworkům právě pro tento účel. Jelikož na klientské části spolupracuji s Eliškou Obadalovou, tak jsme se oba shodli, že toto bude ideální varianta. Jelikož má být web responzivní tak jsme v každé z našich částí kladli důraz na to, aby byla aplikace propracovaná, čitelná a mohl na ní v případě pracovat i někdo z nastupujících studentů, kteří se budou Robogames věnovat po nás, popřípadě pokud ji budou chtít upravit, předělat nebo rozšířit o další funkcionality.

Důvod výběru Ionicu byl také podnět od samotných účastníků robotické soutěže, pro které jsme vytvořili dotazník na webu www.survio.com ohledně jejich preferencí v různých tématech týkajících se této robotické soutěže. Jednou z našich otázek, kterou jsme se dotazovaných tázali byla na jakou platformu by novou webovou aplikaci uvítali.

Dostali jsme celkově 46 odpovědí, z nichž 20 odpověděla, že by webovou aplikaci uvítala na platformě webové tak mobilní. Jelikož Ionic framework se zaměřuje na vývoj webových aplikací, hybridních mobilních aplikací založených na webových technologiích, ale také vývoj webových aplikací, a také nabízí možnost vytvoření nativní aplikace, která by poté šla nahrát na oficiální obchody s aplikacemi ke stažení. Ionic tímto zároveň umožní, aby aplikace byla napsána jako jedna zdrojová základna pro všechny platformy, což ulehčí pozdější úpravy a správu aplikace.

8. Používali by jste raději:



Obrázek 8 – graf z výsledků dotazníku účastníků Robogames

3.1 Angular

Angular je framework pro tvorbu responzivních především single-page webových, mobilních a desktopových aplikací. Za jeho správu stojí společnost Google LLC a komunita jeho uživatelů. Je to framework založen na komponentách, obsahuje také velké množství knihoven, které pokrývají aspekty vývoje webových aplikací, jako například přesměrování, správu formulářů, komunikaci mezi serverem a klientem a další. Podporuje architekturu MVC (model-view-controller), která jednotlivé vrstvy uživatelského rozhraní a logiky aplikace odděluje. [20] [21]

Předchůdce moderního Angularu byl AngularJS, který vyvinula společnost Google v roce 2010. Tento krok byl revoluční a zaujmul spoustu vývojářů z důvodu nejvíce oceňované funkce, která byla obousměrná vazba mezi pohledem a modelem. Tento krok umožnil odstranit spoustu nepotřebného kódu mezi definicemi vazby a udržování pohledu a modelu tříd

vždy synchronizovány. V roce 2014 byl vyvinut Angular 2, který byl kompletně přepsán od nuly do TypeScriptu na rozdíl od AngularuJS, který vycházel z JavaScriptu. Další verze postupně rozšiřovaly funkcionality tohoto frameworku. [19]

3.2 Ionic

Ionic je multiplatformní framework zabývající se vývojem webových a hybridních mobilních aplikací. Hybridní aplikace je aplikace založená na webových technologiích s podporou nativních API zařízení. Poskytuje knihovnu pro komponenty uživatelského rozhraní pro iOS i Android zařízení. Je možné v něm využít frameworky Angular, React a Vue. Hybridní aplikace lze vytvořit pomocí Cordova nebo Capacitoru využitím jejich pluginů pro komunikaci s nativními prvky jako fotoaparátem, snímačem otisků prstů a podobně. [22]

II. PRAKTICKÁ ČÁST

4 ÚVOD

V této části jsem se věnoval implementaci aplikace ve vybraném frameworku Ionic a Angular. Specifikuji zde jednotlivé požadavky a jejich řešení.

Pro komunikaci se serverem využívá aplikace server pro správu robotických soutěží, která je praktickou částí bakalářské práce Martina Krčmy.[11] K jednotlivým funkcionalitám v aplikaci využívám API, jehož dokumentaci mi Martin Krčma poskytl. Má praktická část řeší funkcionality:

1. **Přihlášení a odhlášení uživatele**
2. **Registraci nového uživatele**
3. **Správu týmů – vytváření týmů, registraci týmů do ročníku soutěže**
4. **Správa robotů**
5. **Správa uživatelského účtu**
6. **Generování zápasů pro skupiny robotů**
7. **Potvrzování registrace robotů**

Dále je práce spojena dohromady s praktickou částí slečny Elišky Obadalové, která implementovala funkce pro organizátory soutěže. [18]

5 PŘIHLÁŠENÍ

5.1 Požadavky

Součástí přihlašovacího formuláře je pole na zadávání emailu a hesla. Na stránce se dále nachází pole pro vyplnění hesla, které si uživatel při registraci zvolí. Pod kolonkou heslo se nachází tlačítko pro přihlášení. Při vyplnění emailu a hesla při kliknutí na toto tlačítko dochází k odeslání požadavku na server, zda-li jsou vyplněná pole validní. Pokud ano, tak je uživatel puštěn dále. Pokud uživatel zadá neplatné údaje, tak vyzván aby došlo ke korekci emailu či hesla. Další možností bude registrace pomocí Google účtu metodou OAuth2, kterou Martin Krčma na svém serveru implementoval.

5.2 Implementace

Pro přihlášení jsem v Ionicu vytvořil novou stránku s názvem Login. Na této stránce jsem využil modul ReactiveForms, tento modul je součástí Angularu a validuje vstup uživatele, který zadává do formuláře. Po stisknutí tlačítka přihlásit se spustí metoda submitLogin(), na základě toho, jestli je formulář validní se odešle požadavek na server. [23] Serverové požadavky se v tomto projektu řeší skrze HttpModule voláním metod REST API . [30]

Pro využití ReactiveForms jsem nejdříve modul naimportoval do souboru pro moduly dané stránky a poté přidal FormBuilder, FormGroup a Validators do souboru s metodami stránky.

Vytvořil jsem instanci FormGroup třídy a v ní specifikoval oba údaje potřebné ve formuláři (email a password). Ke každé jsem přidal Validator, který určuje formát požadovaného vstupu. Pro Email jsem využil funkce pattern, který povolí vstupy pouze ve formátu emailu a metodu required, která zajistí, že vstup nesmí být nulová nebo prázdná hodnota. [24]

```
ngOnInit() {
  this.loginForm = this.formBuilder.group({
    password: ['', [Validators.required]],
    email: ['', [Validators.required, Validators.pattern('[a-z0-9._%+-]+@[a-z0-9.-]+\.[a-z]{2,3}$')]],
  })
}
```

Obrázek 9 – použití validátoru na ověření vstupu uživatele

← ROBOGAMES



E-mail
jannovak@seznam.cz

Heslo

PŘIHLÁSIT SE

PŘIHLÁSIT SE POMOCÍ GOOGLE ÚČTU

Nemáte účet?

REGISTROVAT SE

Obrázek 10 – vzhled formuláře pro přihlášení

Poté jsem v HTML souboru stránky vytvořil samotný formulář, kde jsem tagu formuláře přidal atribut `[formGroup] = loginForm` a ke každému prvku uvedl `formControlName`, který jsem nalinkoval na dané proměnné ve vytvořené `FormGroup` třídě. Mimo fomulář jsem zde také přidal odkaz na stránku s registrací, v případě, že by si uživatel chtěl založit účet nový.

Stisknutí tlačítka přihlásit se se spustí metoda `submitForm()`. Tato metoda ověří, zda je formulář validní, a pokud ano, pošle jeho obsah jako požadavek metody `POST` na server. Odpověď serveru jsem poté získal metodou `subscribe`. Podle přijaté odpovědi je uživatel buď informován o chybě hesla nebo emailu, v případě validních dat server pošle autorizační token, který je následně uložen do lokální paměti prohlížeče. Tento token se poté odesílá s každým dalším požadavkem na server. Autorizační token slouží k zabezpečení zneužití

patříčných API serveru uživateli s nedostatečnými právy.[31] Poté je uživatel přesměrován na domovskou stránku.

```
submitLogin() {
  this.isSubmitted = true;
  if (!this.loginForm.valid) {
    console.log('Form invalid')
    return false;
  } else {
    console.log(this.loginForm.value)
    this.auth.login(this.loginForm.value).subscribe(res=>{
      if(res['data']!=='Incorrect email or password'){
        this.token=res['data'];
        localStorage.setItem('token',this.token);
        this.router.navigate(['/home']);
      }
      else{
        console.log(res);
        this.isLoggedIn = false;
      }
    },
    (err:HttpErrorResponse)=>{
      console.log(err)
      this.isLoggedIn= false;
    }
  )
}
```

Obrázek 11 – metoda submitLogin

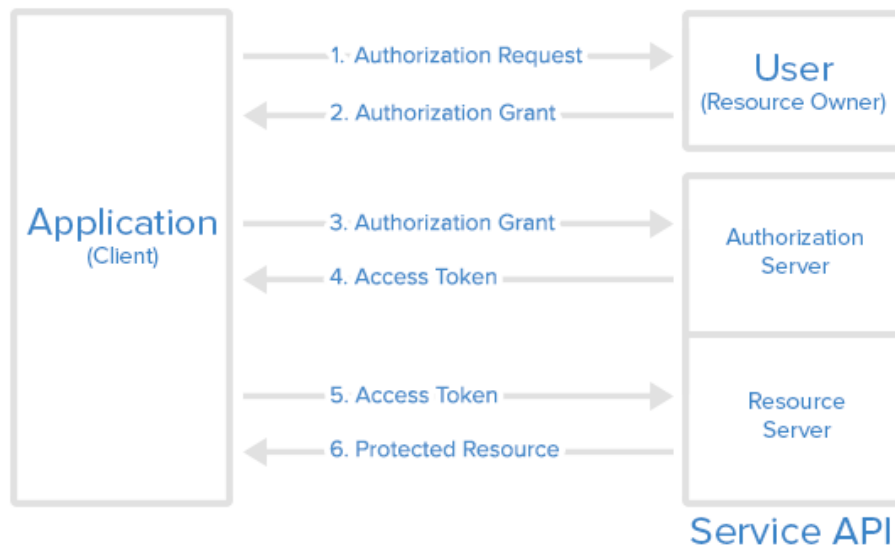
Zasílání požadavků je řešeno skrze tzv. službu, což je třída, která se importuje do komponentu a poté je možné používat její metody. Tyto metody jsem vytvářel do souboru auth.service.ts., kde se metoda pro přihlášení nazývá login. V parametru se předají hodnoty formuláře a následně se odešle post požadavek. Ten je poté vyhodnocen podle odpovědi serveru.

5.2.1 OAuth2

Přihlášení pomocí Google účtu jsem implementoval v UI pomocí tlačítka Přihlásit se pomocí účtu Google, které zavolá metodu pro přesměrování na autorizační stránku. Pro samotnou autorizaci a přesměrování se využívají 2 API poskytnuté serverem Martina Krčmy.

Volání těchto API jsem specifikoval v service auth.service.ts.

Abstract Protocol Flow



Obrázek 12 – jak funguje OAuth2 [25]

```

oAuthGet() {
|   return this.http.get(this._oAuthGet);
| }

oAuthToken(code, state){
|   return this.http.post<any>(this._oAuthToken+state+'&code='+code, '');
| }
  
```

Obrázek 13 – metody pro volání OAuth2 autorizace


Ukázaná `oAuthGet` metoda vrátí URL adresu, na kterou je uživatel přesměrován. Jde o API společnosti Google, kde se uživatel přihlásí svým Google účtem.

Po dokončení autorizace tato Google API zpátky přesměruje uživatele na danou URL, pro tento konkrétní server je to <https://localhost/auth/oauth2/code>, v rámci ladění této aplikace. Stránku `oauth` s touto konkrétní URL (`auth/oauth2/code`) jsem v projektu vytvořil, abych mohl zpracovat parametry, které jsou poslány autorizačním webem. Jde o parametry `state` a `code`, které se následně předají druhé OAuth2 API, které se pošle požadavek metodou `post`, s těmito hodnotami `state` a `code` formou URL parametrů. Pokud je autorizace úspěšná, vrátí tato API autorizační token a uživatel je přesměrován na domovskou stránku a načtou se uživatelská data.

```
this.code = this.route.snapshot.queryParamMap.get('code');
this.state = this.route.snapshot.queryParamMap.get('state');

this.auth.oAuthToken(this.code, this.state).subscribe(res=>{
  this.token = res['data']
  localStorage.setItem('token', this.token);
  this.auth.setNewUserInfo();
  this.router.navigate(['/home']);
})
}
```

Obrázek 14 – přihlášení pomocí OAuth2

 Přihlásit se přes Google

Přihlaste se

Pokračovat do aplikace [RoboCupMS](#)

[Zapomněli jste e-mail?](#)

[Vytvořit účet](#)

[Další](#)

Obrázek 15 – Autorizační stránka Google API

5.3 Domovská stránka

Stránka „home“ je defaultní stránkou této aplikace. Obsahuje odkazy na stránky se základními informacemi o soutěži, jako je kontakt, výsledky, odkaz na registraci, informace o disciplínách a tak dále. Všechny tyto stránky jsou specifikovány ve složce „info“.

Při vstupu na tuto stránku se zjišťuje, zda se v lokálním úložišti nachází autorizační token. Pokud token obsahuje, volá se metoda `getUserInfo` z importované služby `auth.service.ts`, která navrátí roli přihlášeného uživatele. Protože uživatel může mít několik rolí, specifikoval jsem, aby vracela roli s nejvyššími právy, kterou daný uživatel má.

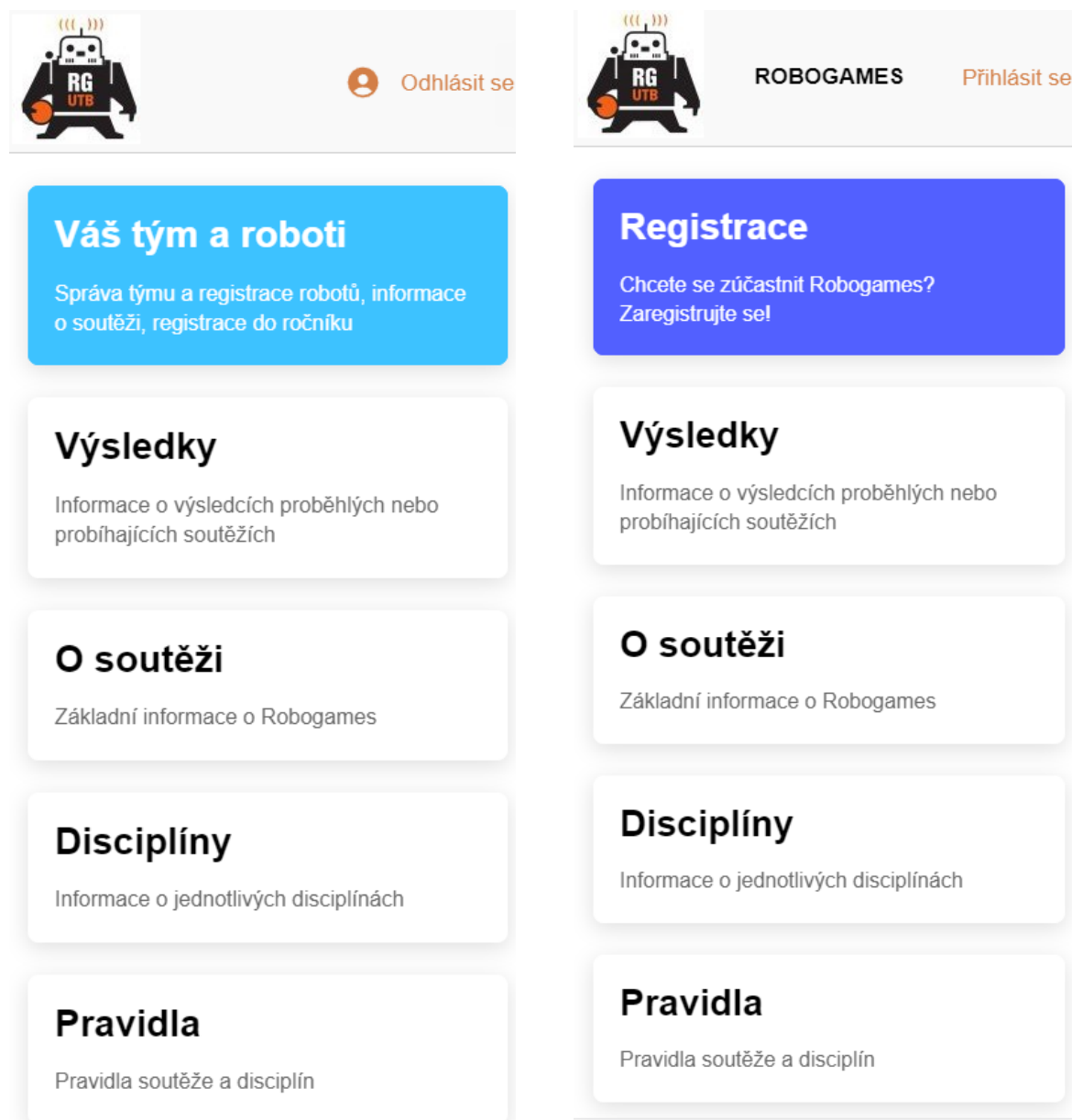
Na základě těchto informací se domovská stránka přizpůsobí tím, že pokud je uživatel přihlášen, bude mít v horní liště přístupnou ikonu pro správu uživatelského účtu a možnosti odhlášení. Toto dynamické přizpůsobení zobrazení na základě rolí jsem řešil použitím Angular HTML syntaxe „`ngIf`“, na základě které se daný prvek vykreslí nebo nevykreslí v závislosti na tom, zda je daná boolean hodnota `true` nebo `false`.

Pro uživatele typu `COMPETITOR` se odemkne možnost správy týmů, robotů a informace o zápasech a hřištích pro daný ročník. Pro uživatele s vyšším oprávněním (`ADMIN`, `LEADER`, `REFEREE`, `ASSISTANT`) se navíc umožní přístup ke funkcionalitám správy soutěže. Toto je řešeno v šablonové části komponentu v HTML souboru direktivo poskytovanou Angularem `ngIf`. [29].

Aby se stránka aktualizovala i po navrácení z předchozí stránky, implementoval jsem i metody životního cyklu Ionic stránky `ionViewWillEnter` a `ionViewDidEnter`, která po návratu uživatele na stránku obnoví informace o roli uživatele. [28]

Odhlášení je řešeno tlačítkem `Odhlásit se`, které zavolá metodu pro odhlášení. Na server se pošle požadavek pro odhlášení, po kterém se token uživatele stává neplatným. Tato metoda také vymaže veškerý obsah lokální paměti prohlížeče.

Dále jsem na stránku zahrnul všechny dostupné informace ze stávajícího webu `Robogames` rozdělených do jednotlivých témat.



Obrázek 16 – domovská stránka uživatele přihlášeného a nepřihlášeného

6 REGISTRACE

6.1 Požadavky

Nový uživatel se bude moci zaregistrovat pomocí poskytnutého formuláře. Tento formulář bude obsahovat jméno, příjmení, email, heslo a datum narození. Uživatel bude informován o úspěchu nebo chybě registrace na základě odpovědi serveru.

6.2 Implementace

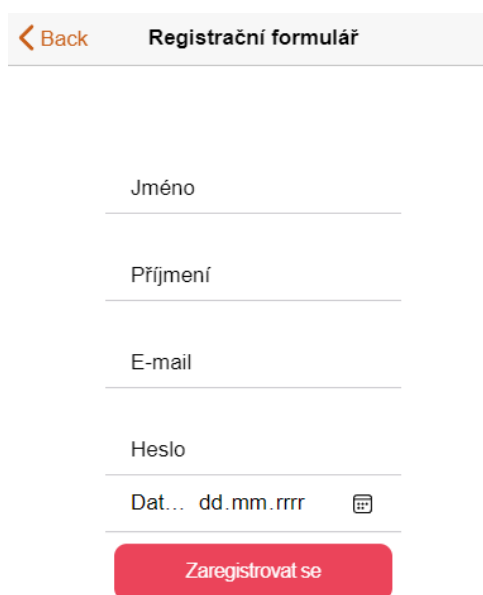
6.2.1 Formulář

Formulář pro registraci jsem vytvořil a validoval pomocí modulu Reactive Forms, jako v případě u přihlašování. Ve formuláři jsem validátory omezil heslo na minimální délku a všechny údaje nastavil jako povinné. [23] [24]

Pro jméno, příjmení a email jsem typ vstupu nastavil na text, u hesla na password a u datumu narození na typ date, která poskytuje přehledné rozhraní pro výběr datumu. [26]

Tlačítko zaregistrovat se spustí metodu, která opět ověří validitu formuláře a pošle jej pomocí dedikované API pro registraci na server.

Tuto metodu pro odesílání požadavku jsem vytvořil v souboru služby auth.service.ts.



Registrační formulář

Jméno

Příjmení

E-mail

Heslo

Dat... dd.mm.rrrr

Zaregistrovat se

Obrázek 17 – registrační formulář

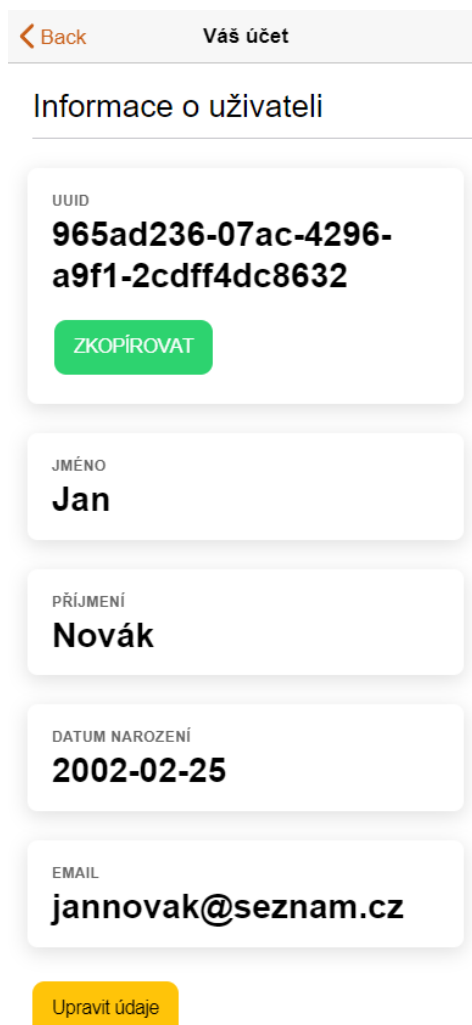
7 SPRÁVA UŽIVATELSKÉHO ÚČTU

7.1 Požadavky

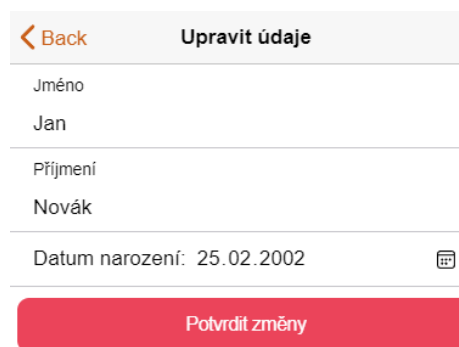
Uživatel si bude moci zobrazit základní informace o svém účtu. Bude možné tyto informace upravovat pomocí formuláře.

7.2 Implementace

Informace o svém účtu si uživatel může zobrazit po kliknutí na ikonu na horní liště aplikace, která ho odkáže na uživatelský účet. Pro tuto funkci jsem vytvořil stránku jménem user-profile. Jednotlivé informace se načítají z lokální paměti prohlížeče. Uživatel zde může vidět, své jméno, příjmení, UUID, email a datum narození. Pro upravování informací slouží tlačítko upravit údaje, po kterém je uživatel přesměrován na stránku edit-user-profile, která



Obrázek 19 – uživatelský účet



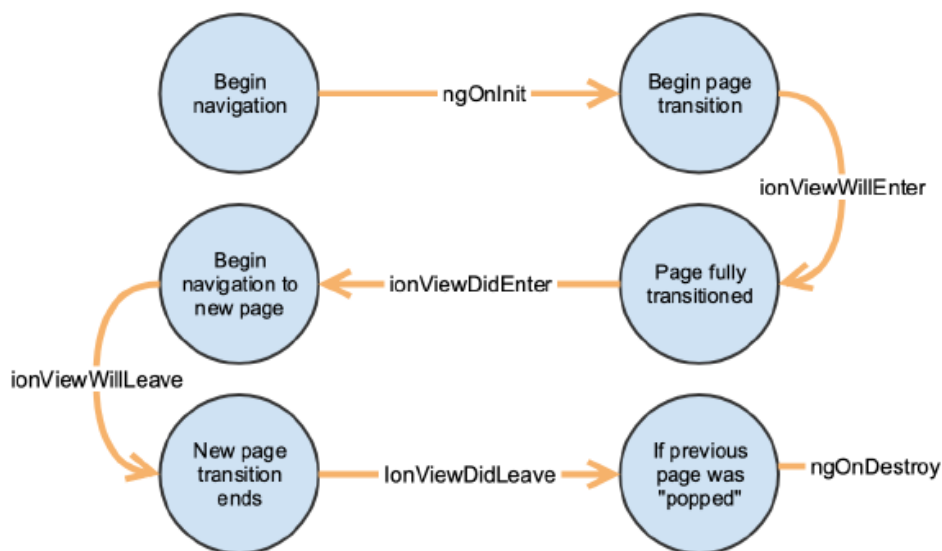
Obrázek 18 – úprava uživatelských údajů

obsahuje formulář pro změnu údajů. Po kliknutí na tlačítko pro potvrzení změn se data posílají na server.

K informaci UUID, jsem přidal možnost kliknutí na tlačítko zkopírovat, které uloží tento údaj do clipboardu. Toho jsem docílil přidáním knihovny ClipBoard Module do globálního souboru projektu pro moduly app.module.ts. Následně jsem na toto tlačítko přidal event handler, který spustí metodu copyUUID, ta pomocí funkce modulu ClipBoard zavolá copyFromContent(), tedy metodu pro zkopírování daného textu. [27]

Tuto funkci jsem zde přidal z důvodu, že UUID hodnota se později využívá při správě týmu, konkrétně u přidávání nového člena, a tato funkce usnadní uživatelům si toto UUID mezi sebou přeposílat.

Aby se data aktualizovala na stránce ihned po změně nějakého údaje, implementoval jsem metodu ionViewWillEnter a ionViewDidEnter, které znovu načtou nové hodnoty z lokálního úložiště, když na ni uživatel vstoupí, nebo se na ni vrátí. [28]



Obrázek 20 – životní cyklus Ionic stránky

```
ionViewWillAppear(){
    this.name = localStorage.getItem('name');
    this.surname = localStorage.getItem('surname');
    this.birthDate = localStorage.getItem('birthDate');
    this.email = localStorage.getItem('email');
}

ionViewDidEnter(){
    this.name = localStorage.getItem('name');
    this.surname = localStorage.getItem('surname');
    this.birthDate = localStorage.getItem('birthDate');
    this.email = localStorage.getItem('email');
}
```

Obrázek 21 – aktualizace informací o uživateli

8 SPRÁVA TÝMŮ

8.1 Požadavky

Uživatel bude moci v případě že není v žádném týmu vytvořit tým nový. Dále bude možná funkce přidání nového člena, kde tato funkce bude vyhrazena pouze pro zakladatele týmu. Dále bude možné zobrazit všechny členy týmu. Pokud není tým v daném ročníku registrován, bude se moci zaregistrovat do kategorie.

8.2 Implementace

Nejdříve jsem vytvořil stránku user-home, která zobrazuje všechny vytvořené ročníky soutěže a pobídne uživatele ke zvolení. Na základě této volby jsem nastavil parametr year, který se předá následující stránce competition-detail. Na této stránce jsem pomocí karet zobrazil možnosti správy týmu, robotů, informace o hřištích, výsledcích a zúčastněných týmů a robotů. Správa týmu se nachází na stránce my-team, kde se nejdříve na základě zasláního požadavku na server o informacích týmu uživatele vyhodnotí, zda je vůbec součástí nějakého týmu. Pro tuto hodnotu jsem vytvořil proměnnou isAMember. Pokud uživatel nemá tým, nabídne uživateli možnost vytvořit nový tým, kde pro toto slouží formulář na stránce team-create. Zde je uživatel vyzván k zadání jména týmu.

Pro výpis členů se posílá get požadavek na server, který vrátí informace o daném týmu včetně členů, kteří se následně vypíší do zobrazení do Ionic komponentu – listu.

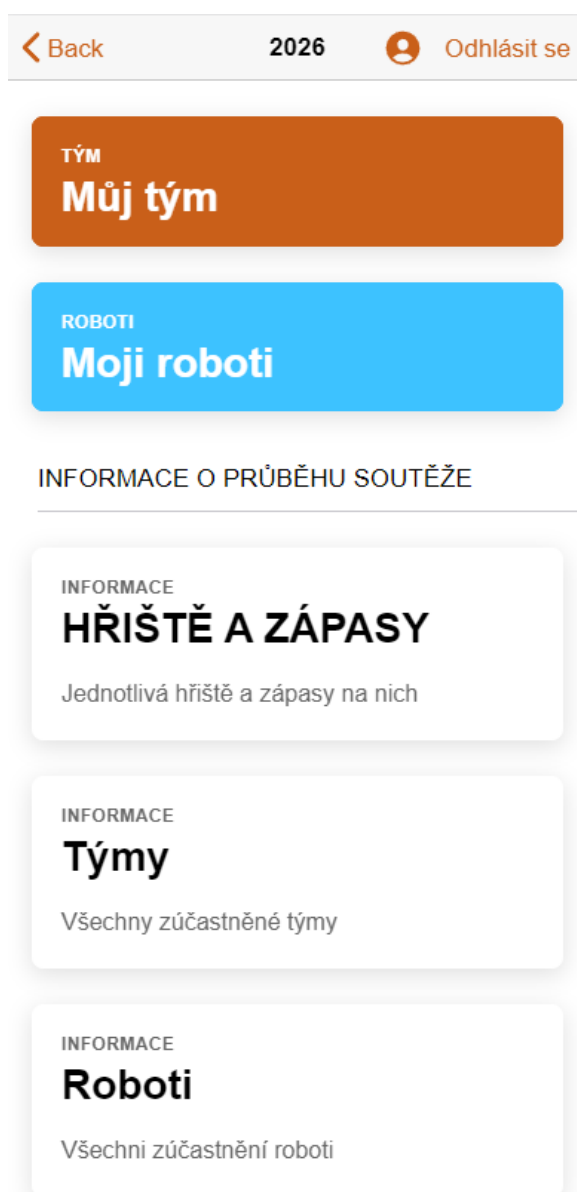
```
<div *ngIf="isAMember">
  <ion-item><h1>{{myTeamInfo.name}}</h1></ion-item>
  <ion-list-header> Členové: </ion-list-header>
  <ion-list *ngFor="let member of teamMembers" >
    <ion-item >
      <p>{{member.name + " " + member.surname}}</p>
      <p *ngIf="member.id == leaderID"> - vedoucí týmu</p>
    </ion-item>
  </ion-list>
</div>
```

Obrázek 22 – Výpis členů týmu

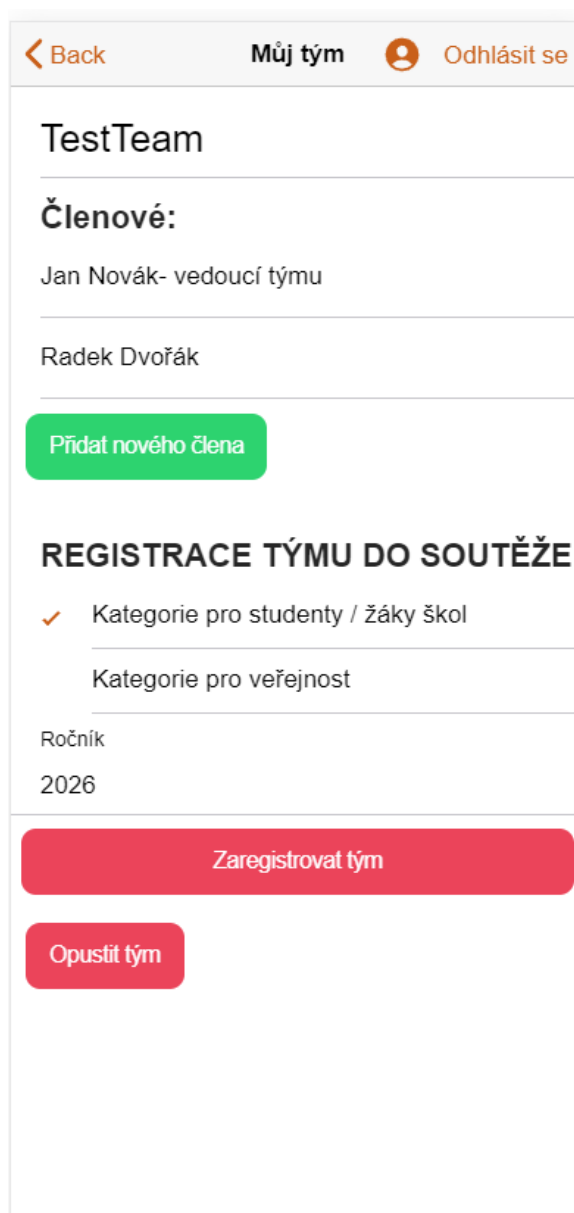
Dále jsem zde implementoval funkci pro registraci týmu do daného ročníku. Tento formulář se zobrazí pouze v případě, že daný tým není zatím v tomto roce registrován. Ve formuláři si týmový vedoucí vybere kategorii, ve které bude tým soutěžit.

Funkci přidání nového člena jsem vyřešil přidáním tlačítka Přidat nového člena, které je opatřeno podmínkou, že se zobrazí pokud se id přihlášeného uživatele shoduje se zakladatelem daného týmu. Toto tlačítko poté odkazuje na formulář, kde velitel týmu může zadat UUID uživatele, kterého chce do týmu přidat.

Dále je zde možnost opustit tým, realizovaný jako tlačítko.



Obrázek 24 – domovská stránka uživatele



Obrázek 23- menu pro správu týmu



Obrázek 25 – přidání nového člena

Veškeré API rozhraní pro manipulaci týmu jsem zkompletoval do jedné servisní třídy jménem `team.service.ts`. Obsahuje jednotlivé metody, které jsem při implementaci stránky pro správu týmu využil.

```
createNewTeam(form){
  return this.http.post<any>(this._createNewTeam, form,
    {headers :new HttpHeaders({'Authorization':'Bearer '+localStorage.getItem('token')}}});
}

addNewTeamMember(form){
  const headers = { 'Authorization': 'Bearer '+localStorage.getItem('token') };
  return this.http.put(this._addNewTeamMember+form.uuid, '', { headers });
}

leaveTeam(){
  const headers = { 'Authorization': 'Bearer '+localStorage.getItem('token') };
  return this.http.put(this._leaveTeam, '', { headers });
}
}
```

Obrázek 26 – ukázka metod pro správu týmů skrze server

9 ROBOTI

9.1 Požadavky

Každý tým, který je registrován do daného ročníku soutěže, bude moci přidávat a registrovat jednotlivé roboty do disciplín popřípadě jej odstranit. Robota a jeho informace bude možné upravit nebo ho zcela vymazat do doby, než byl oficiálně potvrzen vedením týmu. Tým bude mít možnost přehledně získat informace o svých robotech. Včetně proběhlých a plánovaných zápasů.

9.2 Implementace

Pro rozhraní správy robotů jsem vytvořil stránku my-robots. Ta obsahuje nejdříve ověření, zda je uživatel součástí týmu jedině tak je mu ze serverové části dovoleno spravovat roboty. Po splnění podmínek má každý člen k dispozici tlačítko pro přidání robota, které vede na formulář na tvorbu nového robota create-new-robot. Ke správě robotů jsem také vytvořil službu robot-service, která obsahuje veškeré možnosti komunikace se serverem konkrétně ohledně správy robotů. Zde konkrétně využívám metodu getMyRobots, kde jako parametr uvádím rok soutěže.

```
createNewRobot(registerInfo, year){
  const headers = { 'Authorization': 'Bearer '+localStorage.getItem('token') };
  return this.http.post<any>(this._createNewRobot+year, registerInfo, { headers });
}
getRobotById(id:any){
  return this.http.get(this._getRobotById+id, {headers :new HttpHeaders({'Authorization':'Bearer '
+localStorage.getItem('token')}})});
}
registerNewRobot(id:any, disciplineID:any){
  const headers = { 'Authorization': 'Bearer '+localStorage.getItem('token') };
  return this.http.put(this._registerRobot+id+"&disciplineID="+disciplineID, '', { headers });
}
```

Obrázek 27 – ukázka metod pro volání API

Dále je zde seznam již vytvořených robotů získaných z API. Každá položka vypisuje jméno robota, danou kategorii, a informace, zda je robot zaregistrován do disciplíny a jestli byl vedením soutěže potvrzen. Zároveň karta každého robota kliknutím uživatele nasměruje na správu detailů konkrétního robota.

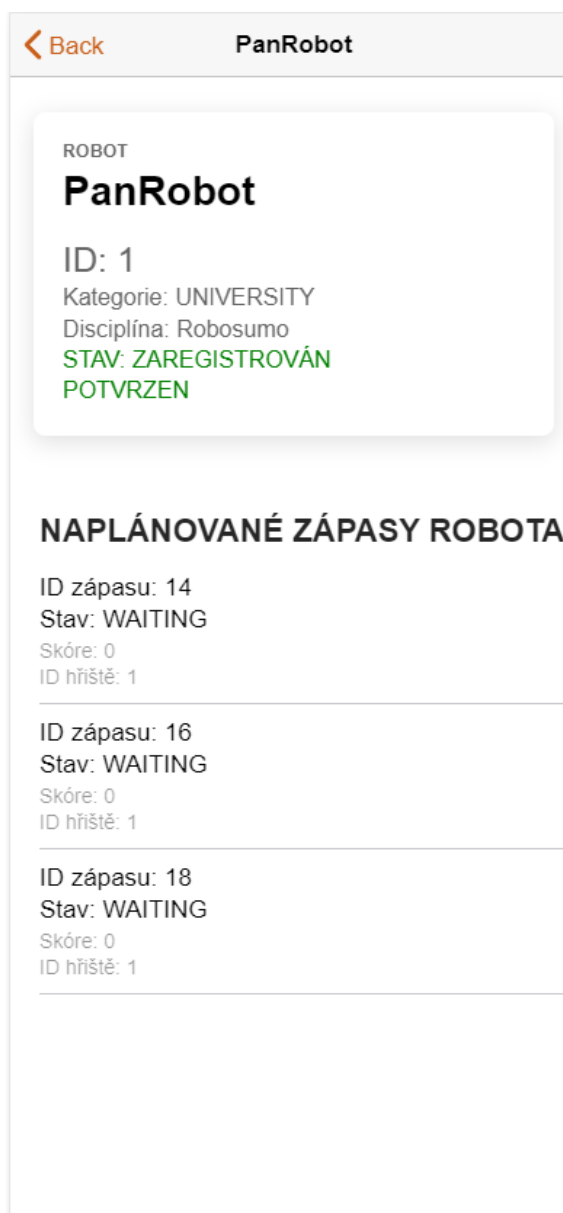
Pro ještě neregistrovaného robota jsem při zobrazení jeho detailu vytvořil formulář, který umožní vybrat jednu disciplínu a do ní robota zaregistrovat. Výběr disciplín je synchronizován s databází disciplín na serveru.



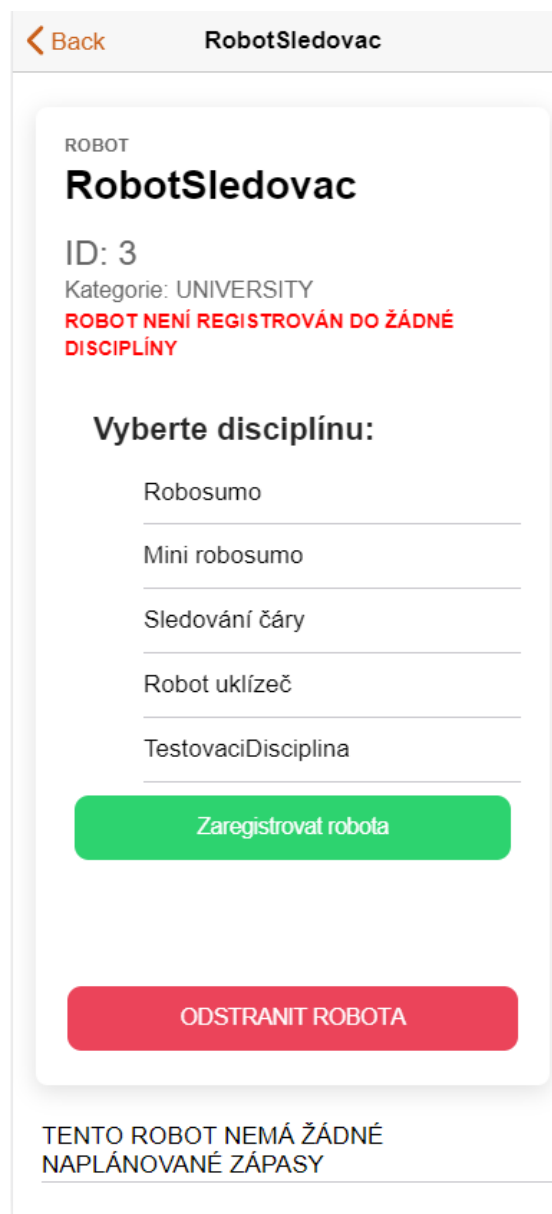
Obrázek 29 - Hlavní stránka správy robotů



Obrázek 28 – formulář přidání robota



Obrázek 31 – detail robota



Obrázek 30 – detail nezaregistrovaného robota

Další možnost, kterou serverové api nabízí, je zobrazování naplánovaných zápasů, na které se má robot dostavit. Tyto informace jsem vypsál na stránce robot-edit formou ionic listu. Informuje tak o skóre proběhlých zápasů a také kam se má robot dostavit na další zápas.

10 ROZPISY ZÁPASŮ SKUPIN

Pro generování zápasů každý s každým poskytuje API správcovského serveru, kde se vybere skupina robotů a hřiště na kterém se mají zápasy odehrát. Logiku této věci řeší server interně, proto jsem vytvořil odpovídající formulář, aby bylo generování zápasů uživatelsky přehledné. Pro výběr robotů a hřišť jsem vyřešil tak, že se nejprve se server požádá o informace o dostupných hřištích a robotech, kteří jsou zaregistrovaní a potvrzení a na základě těchto dat se vygenerují položky pro výběr. Výběr hřiště jsem implementoval formou radio-butonu, a roboty ionicem poskytovaným rozhraním pro výběr více položek. Zvolení roboti musí mít stejnou disciplínu, aby bylo generování serverem provedeno. Server následně vygeneruje po odeslání formuláře zápasy každý s každým pro zvolenou skupinu. Tyto zápasy se objeví u detailních informací každého ze zúčastněných robotů této skupiny.

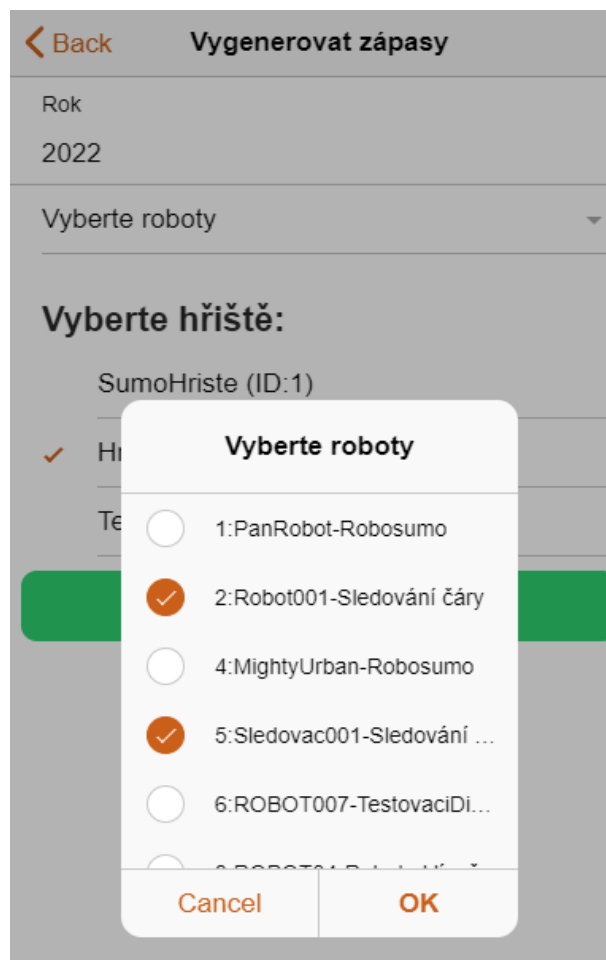
```
<ion-item>
  <ion-label>Vyberte roboty</ion-label>
  <ion-select interface="alert" multiple="true" formControlName="robots">
    <ion-label *ngFor="let robot of robots">
      <ion-select-option value={{robot.id}}>{{+robot.id+":"+robot.name+ "-" +robot.diciplineName}}</ion-select-option>
    </ion-label>
  </ion-select>
</ion-item>
```

Obrázek 32 - ukázka řešení multiple selectu robotů

Jednotlivé vygenerované zápasy se zobrazí na stránkách zápasů jednotlivých hřišť nebo na stránce právě probíhajících soutěží (currently-running page).



Obrázek 33 – generování zápasů

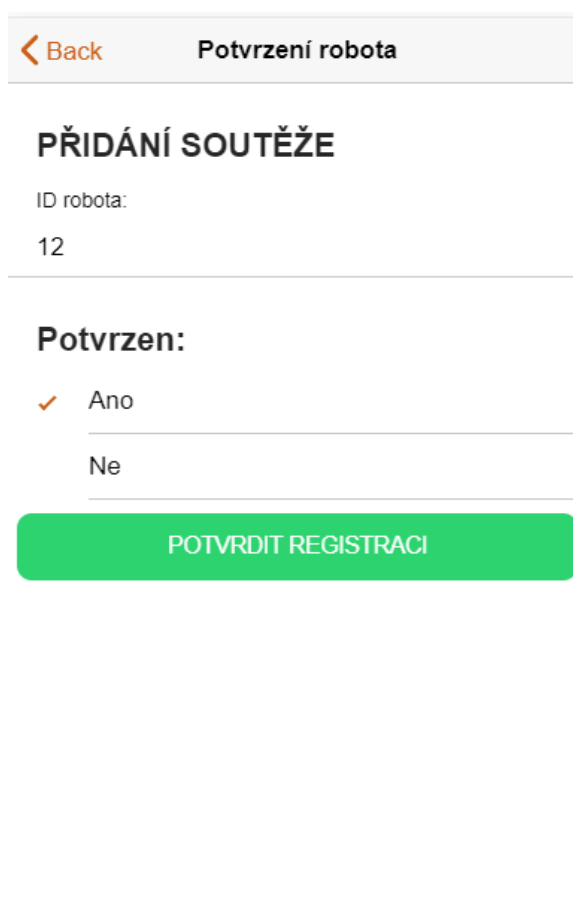


Obrázek 34 – Výběr robotů

11 POTVRZOVÁNÍ ROBOTŮ

Potvrzování robotů svojí funkcí spadá pod administrátorskou část aplikace, kde povolany uživatel na základě id robota zadává do systému, zda je daný robot v pořádku, především, že odpovídá pravidlům disciplíny. Pouze potvrzení roboti budou serverem povoleny k účasti se zápasů.

Tuto funkci jsem realizoval na stránce robot-confirm, který uživateli nabídne jednoduchý formulář pro vyplnění id robota a zda je potvrzen či nikoliv. Tímto formulářem jde i zpětně potvrzení zrušit.



< Back Potvrzení robota

PŘIDÁNÍ SOUTĚŽE

ID robota:
12

Potvrzen:

Ano
 Ne

POTVRDIT REGISTRACI

Obrázek 35 – formulář potvrzení robota

ZÁVĚR

Při tvoření této bakalářské práce jsem při psaní teoretické části hojně využil své zkušenosti ze soutěžení a navštěvování robotických soutěží s kolegou Martinem Krčmou, které se odrazily v popisování a psaní této část, která se věnovala především popisování konání soutěží v České republice a robosoutěží v Polsku. Největší důraz zde byl kladen na popsání ROBOGAMES na FAI, které se tato práce týká, zde jsem popisoval disciplíny, ve kterých se soutěží a registraci, která byla do tohoto roku využívána. Dále jsem vytvořil formulář, který jsme disponovali mezi možné účastníky soutěže abychom zjistili, které soutěže jsou pro ně nejvíce zajímavé a jakou aplikaci by nejvíce uvítali. Při vytváření aplikace byl kladen důraz na výsledky ankety, aby aplikace byla pro účastníky přívětivá a lehce ovladatelná. Při vytváření projektu pro praktickou část jsem konzultoval a spolupracoval s kolegy Eliškou Obadalovou a Martinem Krčmou, kteří řešili další odvětví této aplikace. Vytvořil jsem přihlašovací formulář, registraci, rozpisky soutěže a správu uživatelského účtu včetně správy robotů a týmů. Aplikace je psaná v Ionicu a Angularu, celá aplikace je řešena responzivní formou, aby byla možno spojit odvětví mých kolegů s mou prací v jeden funkční celek. Celý projekt bude nasazen na soutěž od dalšího ročníku, který bude v pořadí jubilejním pátým ročníkem této soutěže. Doufám, že tato aplikace bude hojně využita a sklídí pozitivní ohlasy z řad vyučujících na naší fakultě, organizátorů soutěže samotných účastníků a veřejnosti.

SEZNAM POUŽITÉ LITERATURY

- [1] *Pravidla: Věkové kategorie* [online]. Zlín: FAI UTB [cit. 2022-05-16]. Dostupné z: <https://robogames.utb.cz/pravidla/>
- [2] *Pravidla: Povolené díly a stavebnice* [online]. Zlín: FAI UTB, 2017 [cit. 2022-05-16]. Dostupné z: <https://robogames.utb.cz/pravidla/>
- [3] *Pravidla: Rozpiska soutěže* [online]. Zlín: FAI UTB, 2017 [cit. 2022-05-16]. Dostupné z: <https://robogames.utb.cz/pravidla/>
- [4] *Pravidla: Soutěže – Robosumo* [online]. Zlín: FAI UTB, 2017 [cit. 2022-05-16]. Dostupné z: <https://robogames.utb.cz/pravidla/>
- [5] *Pravidla: Soutěže – Mini Robosumo* [online]. Zlín: FAI UTB, 2017 [cit. 2022-05-16]. Dostupné z: <https://robogames.utb.cz/pravidla/>
- [6] *Pravidla: Soutěže – Sledování čáry* [online]. Zlín: FAI UTB, 2017 [cit. 2022-05-16]. Dostupné z: <https://robogames.utb.cz/pravidla/>
- [7] *Pravidla: Soutěže – Micromouse* [online]. Zlín: FAI UTB, 2017 [cit. 2022-05-16]. Dostupné z: <https://robogames.utb.cz/pravidla/>
- [8] *Pravidla: Soutěže – Robot uklízeč* [online]. Zlín: FAI UTB, 2017 [cit. 2022-05-16]. Dostupné z: <https://robogames.utb.cz/pravidla/>
- [9] *A PID Controller For Lego Mindstorms Robots* [online]. Lego at InPharmix dot com, 2009 [cit. 2022-05-16]. Dostupné z: https://www.inpharmix.com/jps/PID_Controller_For_Lego_Mindstorms_Robots.html
- [10] KRČMA, Martin. Implementace serverové části webové aplikace pro správu robotické soutěže, Zlín, 2022. Bakalářská práce. Univerzita Tomáše Bati ve Zlíně, Fakulta aplikované informatiky.
- [11] KŘÍŽ, Petr. Aplikace pro evidenci výsledků soutěže RoboGames [online]. Zlín, 2021 [cit. 2022-05-06]. Dostupné z: https://digilib.k.utb.cz/bitstream/handle/10563/46112/k%C5%99%C3%AD%C5%BE_2021_dp.pdf. Diplomová práce. Univerzita Tomáše Bati ve Zlíně, Fakulta aplikované informatiky.
- [12] ROBOTIX. *SPŠ PŘEROV* [online]. Přerov: SPŠ PŘEROV, 2018 [cit. 2022-05-21]. Dostupné z: <https://sites.google.com/sps-prerov.cz/robotikx/robotix-2022>
- [13] JEDOBOT. *Robotikahrave.cz* [online]. Jedovnice: SPŠ Jedovnice, 2015 [cit. 2022-05-21]. Dostupné z: <https://robotikahrave.cz/jedobot/>

- [14] Robosoutěž ČVUT. *Robosoutez.fel.cvut.cz/* [online]. Praha: ČVUT FEL, 2013 [cit. 2022-05-21]. Dostupné z: <https://robosoutez.fel.cvut.cz/>
- [15] ROBOTIÁDA. *Robotiada.cz* [online]. Brno: JCMM, 2018 [cit. 2022-05-21]. Dostupné z: <https://robotiada.cz/pravidla-souteze>
- [16] Robotic Tournament. *Robotictournament.pl* [online]. Polsko, Rybnik: Zespól Szkół Technicznych, 2015 [cit. 2022-05-21]. Dostupné z: <https://robotictournament.pl/about-tournament>
- [18] OBADALOVÁ, Eliška. Implementace responzivní webové aplikace pro pořadatele robotické soutěže , Zlín, 2022. Bakalářská práce. Univerzita Tomáše Bati ve Zlíně, Fakulta aplikované informatiky.
- [19] Výhody a nevýhody Angular. *Cs.quish.tv* [online]. Česko: cs.quish.tv, 2022 [cit. 2022-05-21]. Dostupné z: <https://cs.quish.tv/advantages-disadvantages-angular>
- [20] What are the Advantages and Disadvantages of Angular?. *Www.edureka.co* [online]. USA: Utkarsh Sidana, 2021 [cit. 2022-05-21]. Dostupné z: <https://www.edureka.co/blog/advantages-and-disadvantages-of-angular/>
- [21] What is Angular?. *Angular.io* [online]. USA: Google, 2022 [cit. 2022-05-21]. Dostupné z: <https://angular.io/guide/what-is-angular>
- [22] Ionic documentation. *Ionic.io* [online]. USA: Ionic, 2022 [cit. 2022-05-22]. Dostupné z: <https://ionic.io/docs>
- [23] How To Use Reactive Forms in Angular. *Digitalocean.com* [online]. New York: Alligator.io, 2022 [cit. 2022-05-22]. Dostupné z: <https://www.digitalocean.com/community/tutorials/angular-reactive-forms-introduction>
- [24] Reactive forms. *Angular.io* [online]. USA: Google, 2022 [cit. 2022-05-22]. Dostupné z: <https://angular.io/guide/reactive-forms>
- [25] An Introduction to OAuth 2. *Digitalocean.com* [online]. New York: Mitchell Anicas, 2014 [cit. 2022-05-22]. Dostupné z: <https://www.digitalocean.com/community/tutorials/an-introduction-to-oauth-2>
- [26] ion-input. *Ionicframework.com* [online]. USA: Ionic, 2022 [cit. 2022-05-22]. Dostupné z: <https://ionicframework.com/docs/api/input>
- [27] Clipboard. *Material.angular.io* [online]. Google, 2022 [cit. 2022-05-22]. Dostupné z: <https://material.angular.io/cdk/clipboard/overview>

- [28] Ionic Page Life Cycle. Ionicframework.com [online]. USA: Ionic, 2022 [cit. 2022-05-22]. Dostupné z: <https://ionicframework.com/docs/angular/lifecycle>
- [29] Built-in directives. Angular.io [online]. USA: Google, 2022 [cit. 2022-05-22]. Dostupné z: <https://angular.io/guide/built-in-directives>
- [30] Communicating with backend services using HTTP. Angular.io [online]. USA: Google, 2022 [cit. 2022-05-22]. Dostupné z: <https://angular.io/guide/http>
- [31] Angular Authentication With JSON Web Tokens (JWT): The Complete Guide. Angular Authentication With JSON Web Tokens (JWT): The Complete Guide [online]. USA: ANGULAR UNIVERSITY, 2022 [cit. 2022-05-22]. Dostupné z: <https://blog.angular-university.io/angular-jwt-authentication/>
- [32] Angular Navigation. Ionic.io [online]. USA: Ionic, 2022 [cit. 2022-05-22]. Dostupné z: <https://ionicframework.com/docs/angular/navigation>
- [33] Registrace. Robogames.utb.cz [online]. Zlín: Univerzita Tomáše Bati ve Zlíně [cit. 2022-05-22]. Dostupné z: <https://robogames.utb.cz/registrace/>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

HTML Hypertext Markup Language.

MCU Micro Controller Unit

REST Representational State Transfer

MVC Model-view-controller architektura

SEZNAM OBRÁZKŮ

Obrázek 1- Ring pro kategorii Lego Sumo [4]	17
Obrázek 2 - Příklad dráhy pro sledovače čáry 1 [7]	19
Obrázek 3 - Příklad dráhy pro sledovače čáry 2 [7]	19
Obrázek 4 - PID regulátor pomocí dvou čidel [10]	20
Obrázek 5 - PID regulátor pomocí tří čidel [10].....	20
Obrázek 6 - Vzhled aktuálního formuláře pro přihlašování slide 1 [33]	22
Obrázek 7 - Vzhled aktuálního formuláře pro přihlašování slide 2 [33]	23
Obrázek 8 – graf z výsledků dotazníku účastníků Robogames	26
Obrázek 9 – použití validátoru na ověření vstupu uživatele.....	30
Obrázek 10 – vzhled formuláře pro přihlášení	31
Obrázek 11 – metoda submitLogin.....	32
Obrázek 12 – jak funguje OAuth2 [25]	33
Obrázek 13 – metody pro volání OAuth2 autorizace	33
Obrázek 14 – přihlášení pomocí OAuth2	34
Obrázek 15 – Autorizační stránka Google API	34
Obrázek 16 – domovská stránka uživatele přihlášeného a nepřihlášeného	36
Obrázek 17 – registrační formulář	37
Obrázek 18 – úprava uživatelských údajů	38
Obrázek 19 – uživatelský účet	38
Obrázek 20 – životní cyklus Ionic stránky	39
Obrázek 21 – aktualizace informací o uživateli.....	40
Obrázek 22 – Výpis členů týmu	41
Obrázek 23- menu pro správu týmu	42
Obrázek 24 – domovský stránka uživatele	42
Obrázek 25 – přidání nového člena	43
Obrázek 26 – ukázka metod pro správu týmů skrze server	43
Obrázek 27 – ukázka metod pro volání API.....	44
Obrázek 28 – formulář přidání robota	45
Obrázek 29 - Hlavní stránka správy robotů	45
Obrázek 30 – detail nezaregistrovaného	46
Obrázek 31 – detail robota.....	46
Obrázek 32 - ukázka řešení multiple selectu robotů.....	47

Obrázek 33 – generování zápasů	48
Obrázek 34 – Výběr robotů	48
Obrázek 35 – formulář potvrzení robota.....	49