Střední průmyslová škola elektrotechnická

a Vyšší odborná škola Pardubice

**STŘEDNÍ** **PRŮMYSLOVÁ** **ŠKOLA** **ELEKTROTECHNICKÁ**

**MATURITNÍ** **PRÁCE** **–** **PROGRAMOVÁNÍ**

**Piškvorky**

březen 2021 Jan Najman 4.D

*„Prohlašuji,* *že* *jsem* *maturitní* *práci* *vypracoval(a)* *samostatně* *a* *použil(a)* *jsem* *literárních*

*pramenů,* *informací* *a* *obrázků,* *které* *cituji* *a* *uvádím* *v* *seznamu* *použité* *literatury* *a* *zdrojů*

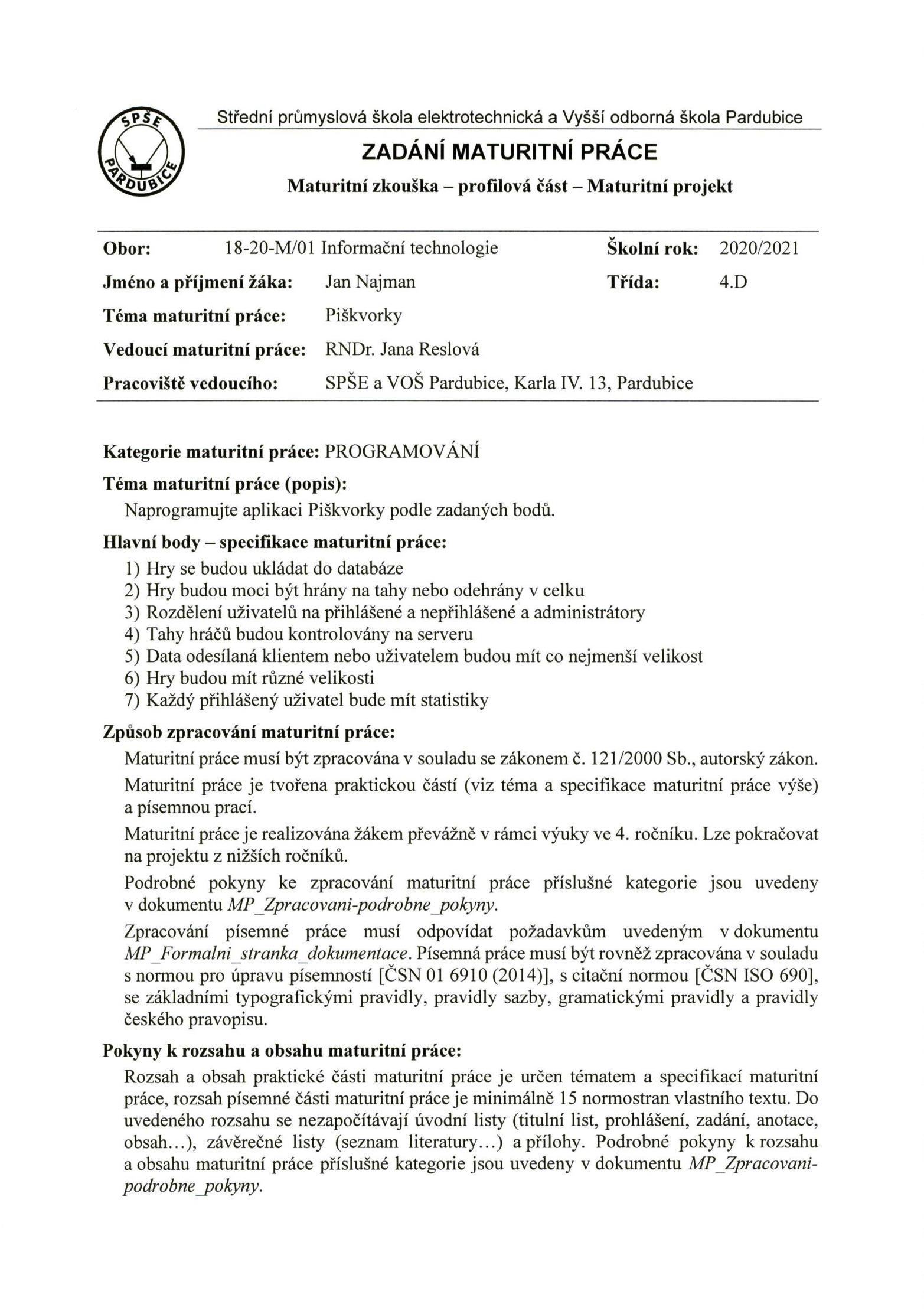
*informací* *a* *v* *seznamu* *použitých* *obrázků* *a* *neporušil* *jsem* *autorská* *práva.*

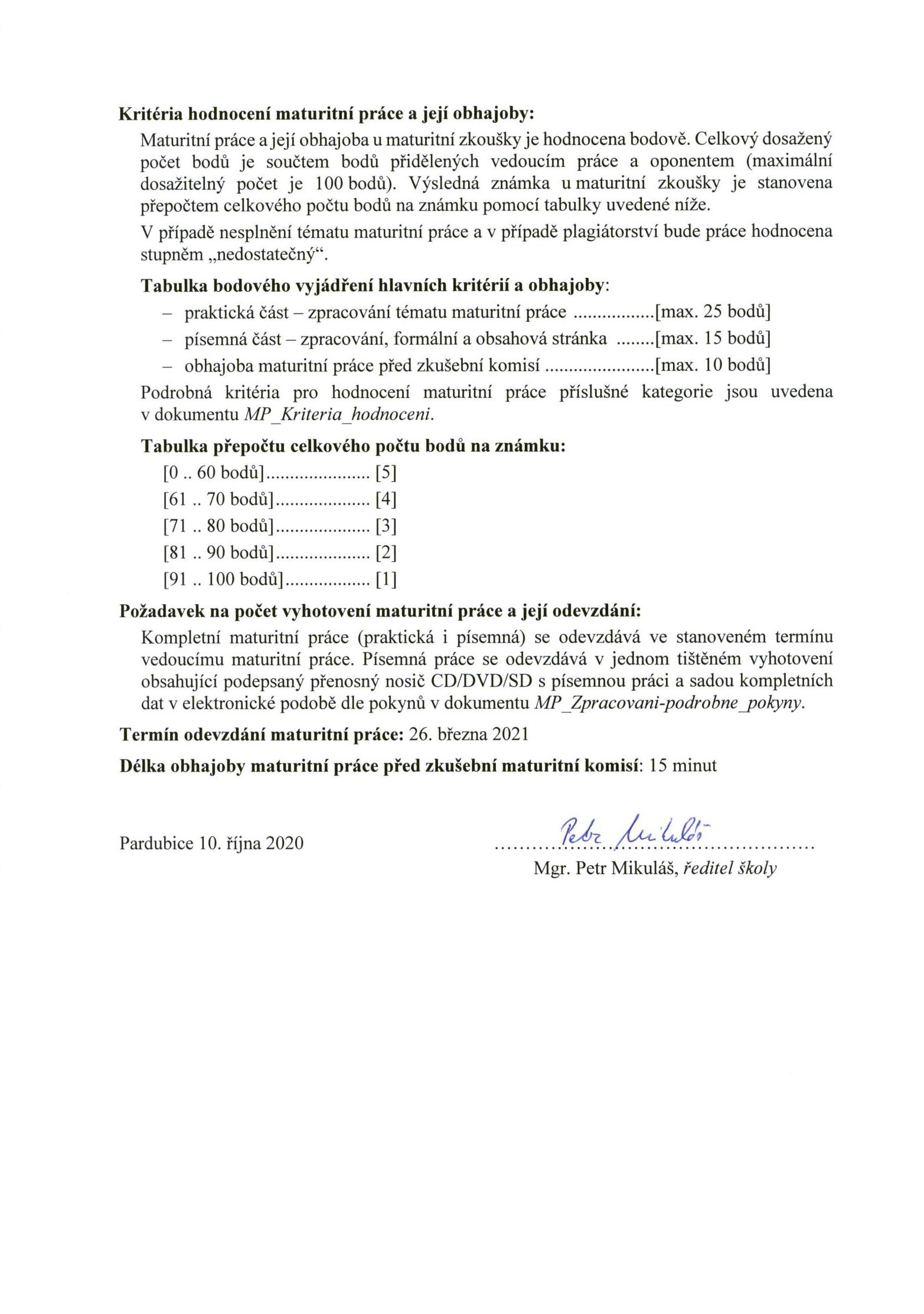
*Souhlasím* *s* *umístěním* *kompletní* *maturitní* *práce* *nebo* *její* *části* *na* *školní* *internetové*

*stránky* *a* *s* *použitím* *jejích* *ukázek* *pro* *výuku.“*

*V* *Pardubicích* *dne* *...........................* *................................................*

*podpis*





**Anotace**

Práce se zabývá rychle rostoucím a čím dál více populárním systémovým jazykem Rust. Ukazuje, jak je tento systémový programovací jazyk vyspělý a jaké má výhody oproti ostatním systémovým jazykům.

Tato práce konkrétně se zabývá programováním RESTful API serveru v programovacím jazyku Rust pomocí frameworku Actix Web a programováním front-endové webové aplikace pomocí frameworku Yew.

Klíčová slova: Rust, API, RESTful, Actix, Actix Web, Web, Server, Aplikace, Front-end, Front-end aplikace, programování, framework, Yew

**Annotation**

This work deals with the rapidly growing and increasingly popular system programming language Rust. It shows how advanced this system programming language is and what its advantages are over other system languages.

This work speciﬁcally deals with programming RESTful API server in Rust programming language using the Actix Web framework and programming a front-end web application using framework Yew.

Keywords: Rust, API, RESTful, Actix, Actix Web, Web, Server, Application, Front-end, Front-end applications, programming, framework, Yew

**Obsah**

**Úvod** **9**

**1** **Analýza** **obdobných** **aplikací** **10** 1.1 turtlediary . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 10 1.1.1 Kladné stránky . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 10

1.1.2 Záporné stránky . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 10 1.2 Ultimate Tic Tac Toe . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11

1.2.1 Kladné stránky . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11 1.2.2 Záporné stránky . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11

1.3 gametable . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 12 1.3.1 Kladné stránky . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 12 1.3.2 Záporné stránky . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 12

**2** **Návrh** **projektu** **13** 2.1 Obecná struktura . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 13 2.2 API . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 13 2.3 Redis . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 13 2.4 Databáze . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 13 2.5 Front-end . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 13

**3** **Zpracování** **praktické** **části** **14** 3.1 Back-end . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 14 3.1.1 Správa uživatelů . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 14

3.1.2 Vytváření žádostí o hru

3.1.3 Vytvoření hratelné hry

. . . . . . . . . . . . . . . . . . . . . . . . . 14

. . . . . . . . . . . . . . . . . . . . . . . . . 14

3.1.4 Hraní hry . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 14 3.2 Front-end . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 15

3.2.1 Registrace . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 15 3.2.2 Proﬁl . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 15 3.2.3 Výpis uživatelů . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 15

3.2.4 Výpis her

3.2.5 Hraní hry

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 15

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 15

3.2.6 Výpis pozvánek . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 16 3.2.7 Vytváření pozvánky . . . . . . . . . . . . . . . . . . . . . . . . . . . 16 3.2.8 Úprava uživatele . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 16

**4** **Závěr** **17**

**5** **Seznam** **použité** **literatury** **a** **zdrojů** **informací** **18**

**6** **Seznam** **použitých** **zkratek** **19**

**7** **Seznam** **obrázků,** **tabulek,** **příloh** **20**

**8** **Přílohy** **21**

**Úvod**

Poslední dobou se všichni pokouší optimalizovat své web servery již při jejich programování, kvůli náporu, který by nemusely stíhat. Kdo tak neprovede může toho litovat a snaží se tento problém obejít jinak. Tím že změní jazyk nebo vytvoří repliky své aplikace a dají na ně load-balancer, tuto možnost nakonec musí využít všichni, při velmi vysokém náporu.

Bohužel nepoužívanější jazyky k naprogramování web serveru jsou PHP, JS (Node.js) nebo Python (Flask, Django). Dá se v nich rychle udělat co potřebujete, ale mají spoustu nevýhod, a hlavně všechny tyto jazyky jsou tzv. interpretované jazyky. To znamená, že na pozadí běží nějaký engine (interpreter), který musí zpracovat daný kód za běhu. Jejich největším problémem oproti kompilovaným programovacím jazykům jako jsou Rust, C++, C# je jejich rychlost a možnost zjisti chybu při kompilaci.

Tabulka 1: Porovnání rychlosti jazyků

Jazyk

Rust C# Node.js PHP

Python 3.5

Python 2.7

Pomalejší než C++ (gcc -O2)

7 % 78 % 93 % 596 %

1800 %

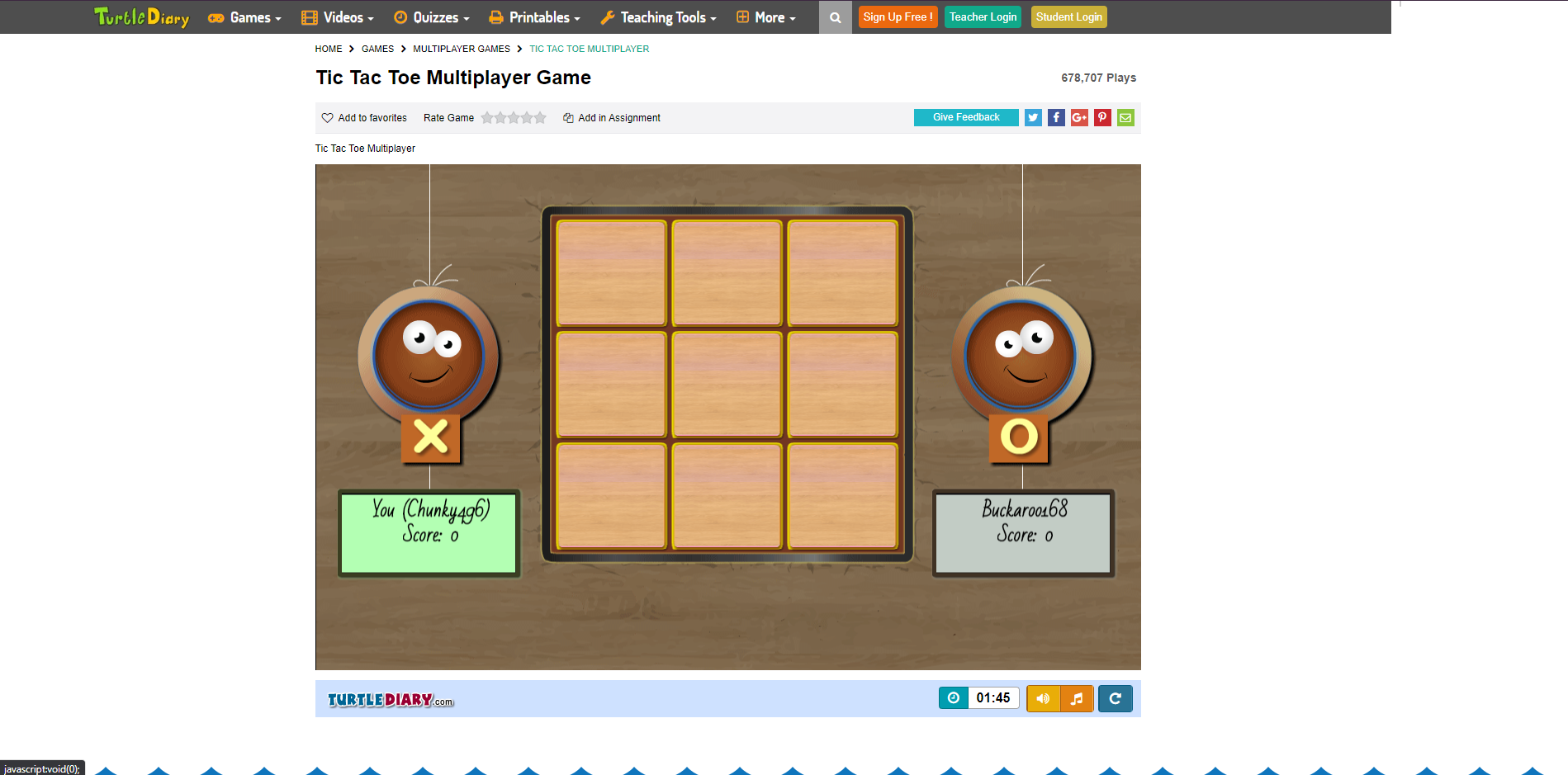
2562 %

Rozhodl jsem se ukázat, že systémový programovací jazyk Rust je na tolik vyspělý, že se nejen zvládne vše, co jiné jazyky, ale i to, že je rychlejší než konkurence.

Vybral jsem si programování RESTful API serveru a front-endové webové aplikace z důvodu, že jsou to dne nejpoužívanější technologie a mají budoucnost.

Téma Piškvorky jsem si vybral z více důvodů. Mělo by na něm jít perfektně předvést alespoň základy obou z frameworků. Tato hra by měla být každému povědomá a nemusím se zabývat vysvětlováním pravidel.

9



**1** **Analýza** **obdobných** **aplikací**

Analýzu obdobných aplikací je dobré provádět, abyste získali představu, jak má vaše aplikace vypadat. Co chcete, aby uměla a v čem byla lepší něž ostatní aplikace.

**1.1** **turtlediary**

Adresa: <https://www.turtlediary.com/game/tic-tac-toe-multiplayer.html>

Web nabízí možnost hrát piškvorky 3x3 s náhodnými lidmi, nebo s kamarádem.

**1.1.1** **Kladné** **stránky**

• ke hraní není potřeba registrace • vizuální rozhraní hry

**1.1.2** **Záporné** **stránky**

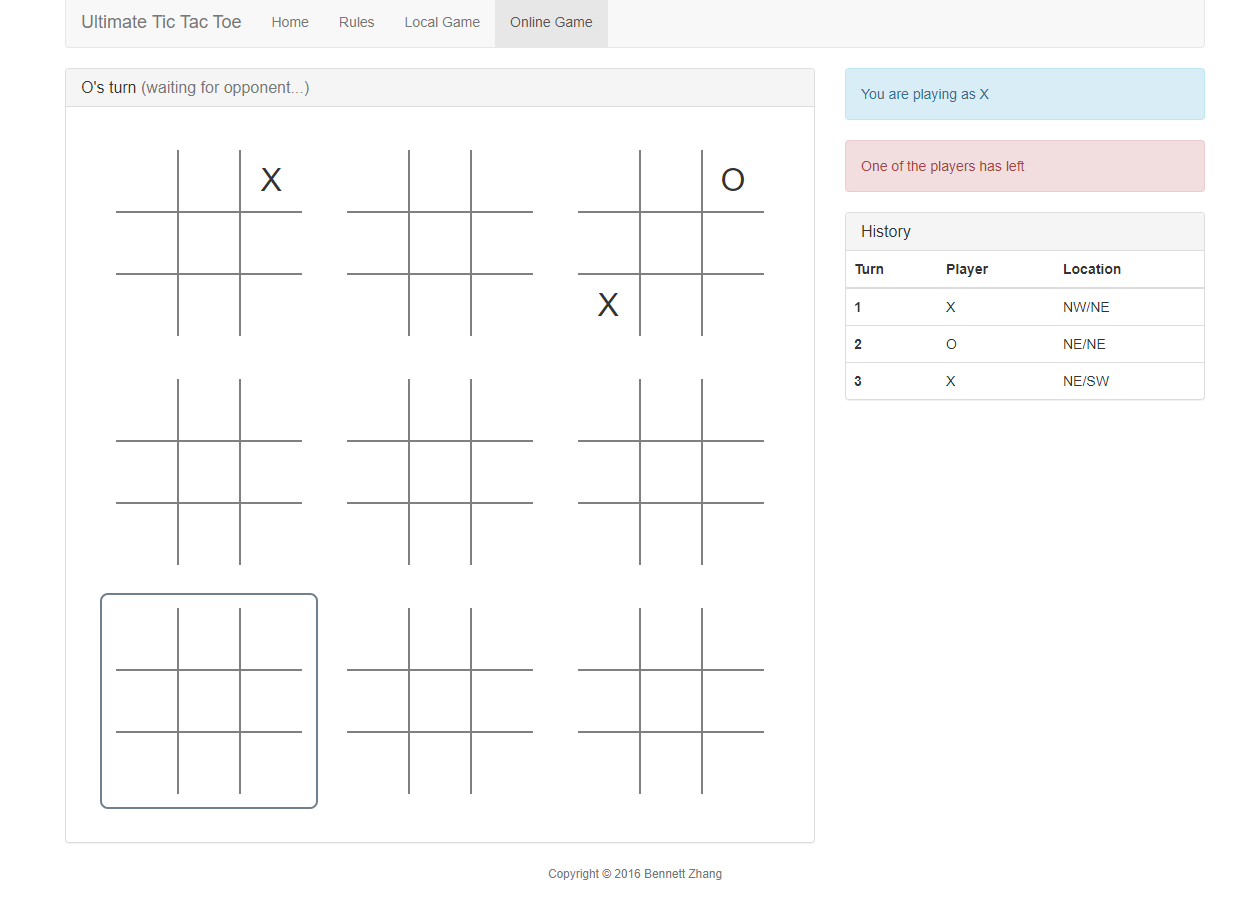
• web není responzivní

• hrací plocha je moc malá oproti zbylému volnému místu • web nemá statistiky nebo výpis nejlepších hráčů

• na mobilním telefonu není hrací plocha vidět celá

Obrázek 1: turtlediary - https://www.turtlediary.com/game/tic-tac-toe-multiplayer.html

10



**1.2** **Ultimate** **Tic** **Tac** **Toe**

Adresa: <https://ultimate-t3.herokuapp.com>

Web nabízí možnost hrát piškvorky 3x3 na více polích s kamarádem přes internet nebo lokálně.

**1.2.1** **Kladné** **stránky**

• ke hraní není potřeba registrace • čistý interface

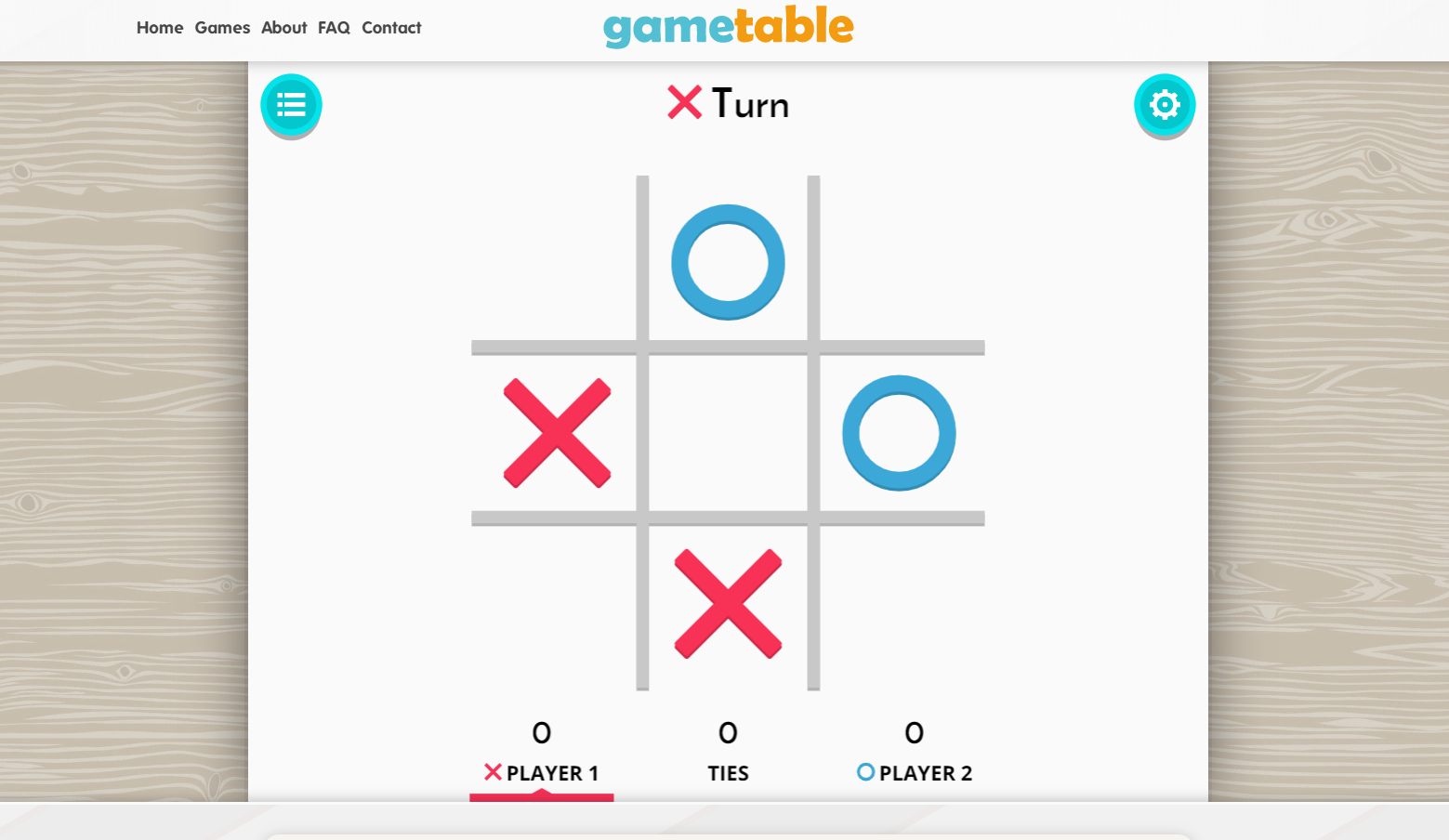
**1.2.2** **Záporné** **stránky**

• vice hracích ploch

• web nemá statistiky nebo výpis nejlepších hráčů

Obrázek 2: Ultimate Tic Tac Toe - https://ultimate-t3.herokuapp.com

11



**1.3** **gametable**

Adresa: <https://gametable.org/games/tic-tac-toe>

Web nabízí možnost hrát piškvorky 3x3 s kamarádem lokálně nebo proti AI.

**1.3.1** **Kladné** **stránky**

• ke hraní není potřeba registrace • čistý interface

• hra má vysvětleny pravidla pod hrací plochou

**1.3.2** **Záporné** **stránky**

• není možnost hrát s někým přes internet

• web nemá statistiky nebo výpis nejlepších hráčů

Obrázek 3: gametable - https://gametable.org/games/tic-tac-toe

12

**2** **Návrh** **projektu**

**2.1** **Obecná** **struktura**

Veškeré požadavky, které nebudou odkazovat na front-end zodpovídá API. API je poté napojené na databázi a redis.

Front-end nemá sám o sobě žádný přístup k databázi nebo redisu.

**2.2** **API**

API spojuje vše dohromady. Poskytuje veškeré informace front-endu a zpracovává veškeré příchozí informace.

**2.3** **Redis**

V redisu jsou ukládány uživatelské relace, aby k nim byl rychlý přístup a jejich ověření.

**2.4** **Databáze**

Databáze ukládá informace o uživatelích a hrách pro dlouhodobé uložení dat.

Diagram ﬁg. 4

**2.5** **Front-end**

Slouží jako graﬁcké zobrazení dat a druhořadá kontrola dat. Zobrazuje informace o uživatelích a hrách.

13

**3** **Zpracování** **praktické** **části**

**3.1** **Back-end**

**3.1.1** **Správa** **uživatelů**

Uživatelé jsou umístěni v tabulce *users*. Role jsou uloženy v tabulce *roles* a jsou k uživatelům přiřazovány skrz tabulku *roles\_to\_users*.

Pokud má uživatel roli *Admin* tak mohou upravovat kohokoli údaje bez omezení včetně rolí a hesla.

Pokud má uživatel roli *Banned* tak nemůže vytvářet pozvánky, ani nemůže být zahrnut do pozvánky jiným uživatelem.

**3.1.2** **Vytváření** **žádostí** **o** **hru**

Přihlášení uživatelé mají možnost vytvářet nové hry s různými parametry.

Nejprve se vytvoří žádost o hru, která se nachází v tabulce *game\_requests*. K dané žádosti na hru se přiřadí uživatelé skrz tabulku *users\_to\_game\_requests*.

Pozvánky nemohou vytvářet uživatelé s rolí *Banned*.

**3.1.3** **Vytvoření** **hratelné** **hry**

Po vytvoření žádosti o hru jí musí všichni hráči potvrdit a hra bude vytvořena, nebo někdo z pozvaných hráčů odmítne žádost a žádost o hru bude vymazána.

Jakmile je účast všech hráčů potvrzena, tak se vytvoří nová hra v tabulce *games*, přiřadí se k ní uživatelé skrz tabulku *games\_to\_users* a žádost o hru je poté vymazána.

**3.1.4** **Hraní** **hry**

Hrát můžete jen když jste na tahu a pokud hrané políčko ještě nebylo použito. Vyhraní hry se kontroluje na front-endu, pokud front-end usoudí, že hráč vyhrál tak výhru oznámí back-endu a ten výhru zkontroluje.

Back-end kontroluje, jestli je hráč na tahu, jestli hra neskončila, nebo jestli jeho tah je validní. V případě, že hráč ohlásí výhru, ale server zjistí, že lže, tak daný tah

14

zahodí a odpoví chybou.

**3.2** **Front-end**

**3.2.1** **Registrace**

Všechna pole jsou kontrolována. Pokud nějaké pole není validní, tak se nepošlou data na back-end.

Back-end data zkontroluje a pokud zjistí, že nejsou validní, tak žádost zahodí a vrátí chybu.

Pro hashování hesla se používá 128 znaková sůl a algoritmus Argon2.

**3.2.2** **Proﬁl**

Zobrazuje informace o uživateli a hry, ve kterých se nachází.

Zobrazuje také počet výher, proher a remíz. Z těchto dat poté vypočítá winrate (výhry / prohry).

Pokud je uživatel na svém proﬁlu, nebo pokud má uživatel roli *Admin*, tak se mu také zobrazí tlačítko na upravení proﬁlu.

**3.2.3** **Výpis** **uživatelů**

Zobrazuje všechny registrované uživatele a pár informací o nich.

Uživatelům s rolí *Admin* se navíc zobrazuje tlačítko upravení proﬁlu.

**3.2.4** **Výpis** **her**

Zobrazuje všechny rozehrané, nebo dohrané hry s jejich hráči.

**3.2.5** **Hraní** **hry**

Hry jsou hrány na síti 30x30.

Uživatelé jsou zobrazováni s jejich symbolem a za jménem je napsáno, jestli jsou na tahu.

15

Hrát můžou jen uživatelé, kteří jsou v dané hře, ale dívat se může kdokoli.

Tahy uživatelů jsou kontrolovány, jestli jsou validní a jestli nastala výhra nebo remíza.

**3.2.6** **Výpis** **pozvánek**

Zobrazuje název pozvánky (později název hry), počet tahů k vítězství.

Uživatel může pozvánku přijmout, nebo odmítnout.

**3.2.7** **Vytváření** **pozvánky**

Při vytváření pozvánky jsou skoro všechna pole kontrolována.

Uživatelé s rolí *Admin* mají práva na vypnutí skoro všech kontrol.

**3.2.8** **Úprava** **uživatele**

Uživatel může upravovat vše, kromě jeho rolí.

Uživatel s rolí *Admin* může upravovat vše a má možnost vypnout kontrolu, která je vyžadována po ostatních uživatelích.

16

**4** **Závěr**

17

**5** **Seznam** **použité** **literatury** **a** **zdrojů** **informací**

18

**6** **Seznam** **použitých** **zkratek**

Tabulka 2: Seznam použitých zkratek

Zkratka

API REST SPŠE

VOŠ

Význam

Application Programming Interface Representational State Transfer

Střední průmyslová škola elektrotechnická

Vysoká odborná škola

19

**7** **Seznam** **obrázků,** **tabulek,** **příloh**

**Seznam** **obrázků**

1 turtlediary - https://www.turtlediary.com/game/tic-tac-toe-multiplayer.html 10 2 Ultimate Tic Tac Toe - https://ultimate-t3.herokuapp.com . . . . . . . . . . 11 3 gametable - https://gametable.org/games/tic-tac-toe . . . . . . . . . . . . 12 4 ER Diagram . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 21

**Seznam** **tabulek**

1 Porovnání rychlosti jazyků

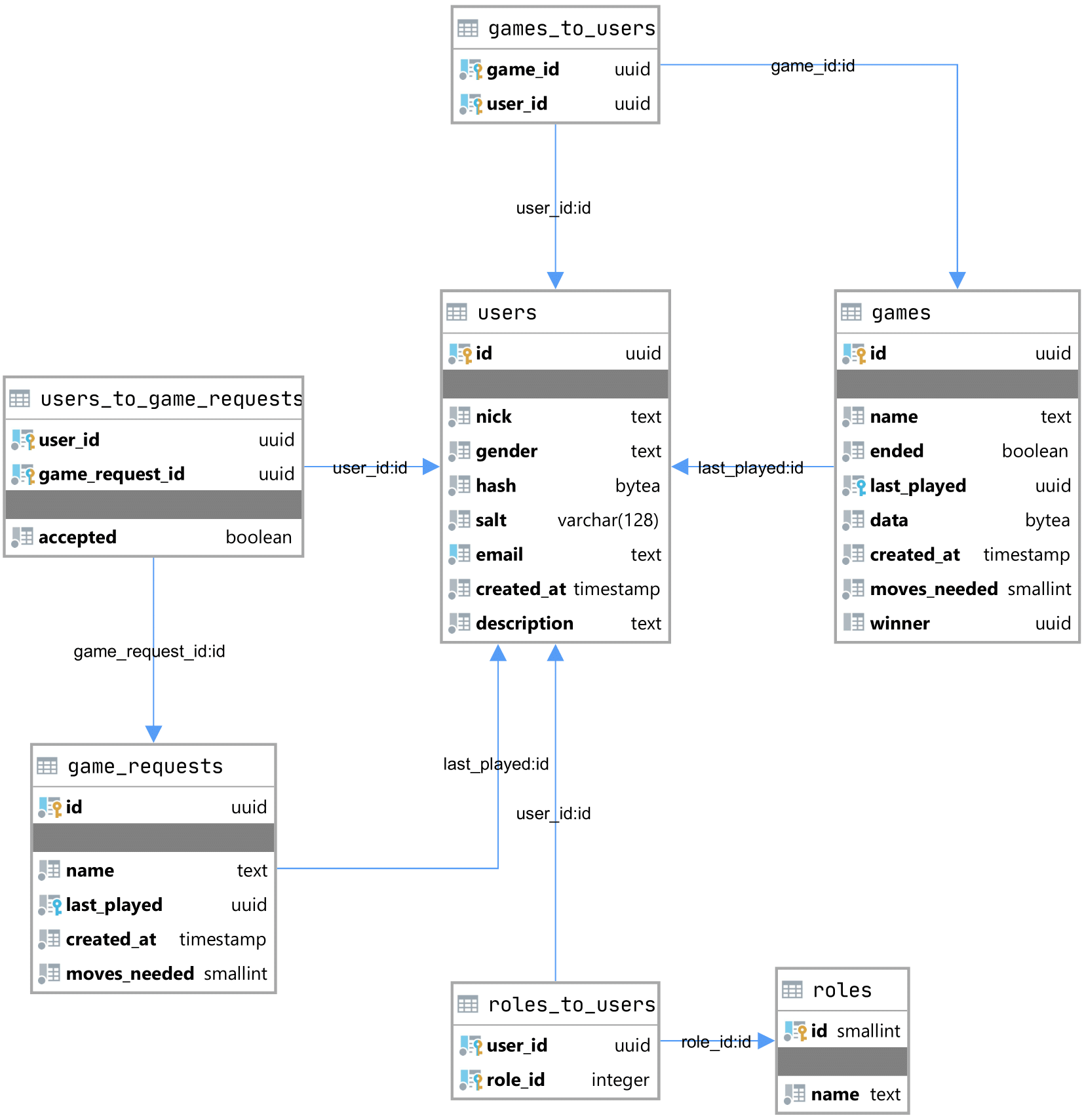
2 Seznam použitých zkratek

. . . . . . . . . . . . . . . . . . . . . . . . . . . 9

. . . . . . . . . . . . . . . . . . . . . . . . . . . 19

**Seznam** **příloh**

20



**8** **Přílohy**

Obrázek 4: ER Diagram

21