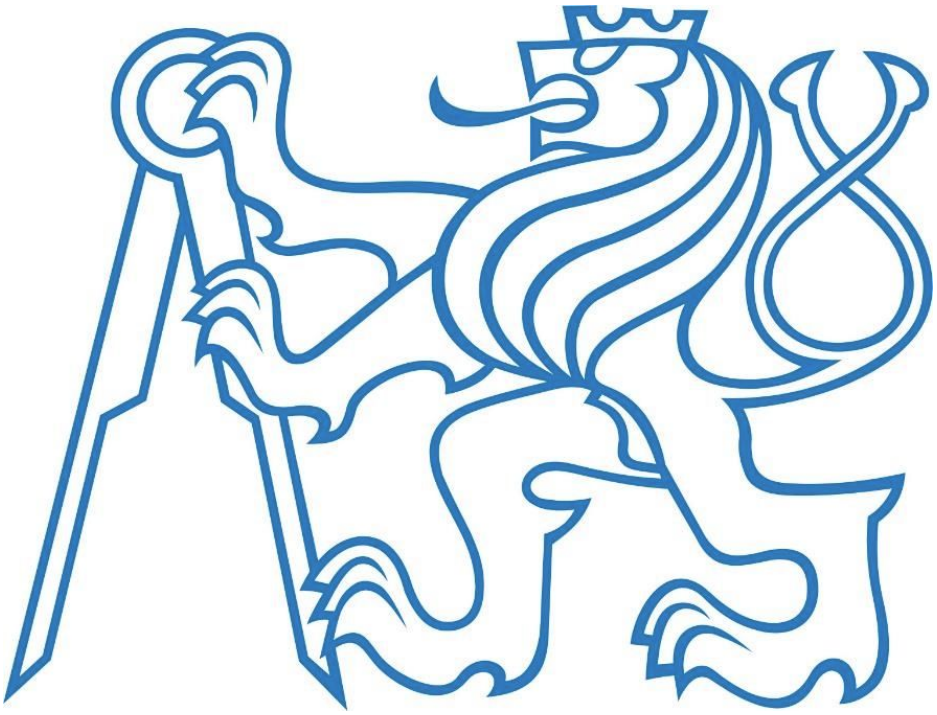
Semestrální projekt

NSS

Dokumentace



Tým:

Jakub Kiml
Ondřej Mareš
David Stražovan
Zuzana Štětinová
Petr Švec
Otto Vodvářka

Obsah

Motivace

Projekt vznikl za účelem vytvoření webové aplikace, která bude sloužit menším skupinkám lidí pro tvorbu a zaznamenávání průběhu lokálních turnajů typu “pavouka”, tedy vyřazovacích turnajů. Systém slouží hlavně pro ulehčení vytváření zápasů.

Tento systém je dále nazýván Spider.

Zadání projektu (“to be”)

Vytvořený systém bude umožňovat vytvoření, zobrazení a vyhodnocení turnajových zápasů. Dále bude uživatelsky přívětivý a co nejjednodušší na ovládání. Systém bude sloužit hlavně pro menší skupiny.

Pořadatel se bude moct do systému přihlašovat a bude tak mít vždy přístup k jeho poslednímu vytvořenému turnaji. Hráč naopak nebude nucen k vytváření profilu a bude se jen jednorázově přihlašovat pomocí údajů od pořadatele.

Hráč bude schopen zadávat výsledky turnajů, ve kterých je účastníkem, pořadatel bude schopen měnit výsledky libovolně.

Postup v turnaji bude viditelný pro pořadatele, hráče i diváky.

SWOT analýza

Silné stránky • Tým obsahuje zkušené členy • cíleno na jednoduchost použití • Rychlost vytvoření turnaje • Pro hráče bez nutnosti vytváření účtu	**Slabé stránky** • Marketing (O aplikaci se nikdo nedozví) • Nové odvětví, ve kterém nemáme zkušenosti • Malé množství typů turnajů (ve verzi 1 jen jeden typ) • Omezená funkcionalita (nelze prohlížet předešlé turnaje...)
Příležitosti • Pořadatelé pořádají více turnajů a díky tomu se o projektu dozví další lidé	**Hrozby** • Tým se rozpadne (první společný projekt) • Časová vytíženost členů týmu • Uživatelé nebudou přecházet z papírové formy a web nebudou používat

Analýza 5F

Konkurence

- Velké množství, ale konkurence je buď zastaralá, není vizuálně příjemná, ani interaktivní, nebo nabízí pro naší cílovou skupinu, zbytečně komplexní řešení, které je placené.

Síly dodavatelů

- Nejsme závislý na dodavatelích

Síly odběratelů

- Cílová skupina: Pořadatelé malých turnajů
- Mnoho odběratelů s malou “kupní” silou. (Software bude zdarma, tudíž se o kupní síle nedá mluvit)

Substituční produkty

- Substituční produkty nejsou možné

Nově příchozí

- Nízké vstupní náklady, žádná omezení
- Konkurence může snadno vzniknout.

Požadavky

Požadavky pořadatele

Pořadatel turnaje od systému požaduje, aby...

- ...mohl vytvořit turnaj s určitými vlastnostmi (např. jméno, poznámka…), možnostmi (např. souboj o 3. místo) a libovolným počtem hráčů, jejichž jména do něj zadá
- ...se mohl vrátit do průběhu posledního turnaje pomocí přihlášení
- ...mohl měnit veškeré výsledky svého turnaje, na kterých nezávisí již odehraná kola
- ...mohl psát k zápasům poznámky pro hráče, například čas a místo zápasu
- ...systém automaticky po zapsání výsledku určil vítěze a posunul jej dále, kvůli rychlejší organizaci
- ...hráč mohl zapisovat pouze výsledky zápasů, kterých se účastní, kvůli možným omylům a jasné kontrole

Požadavky hráče

Hráč v turnaji od systému požaduje, aby…

- …se mohl do turnaje přihlásit za jeho běhu - jednoduše a jednorázově bez registrace
- ...si mohl jednoduše zobrazit celého pavouka a viděl tak složení a výsledky turnaje
- ...si mohl zadat výsledek svého zápasu k urychlení organizace
- ...vždy věděl, proti komu nastupuje
- ...měl možnost změnit výsledek, pokud jej omylem zadal chybně

Požadavky diváka

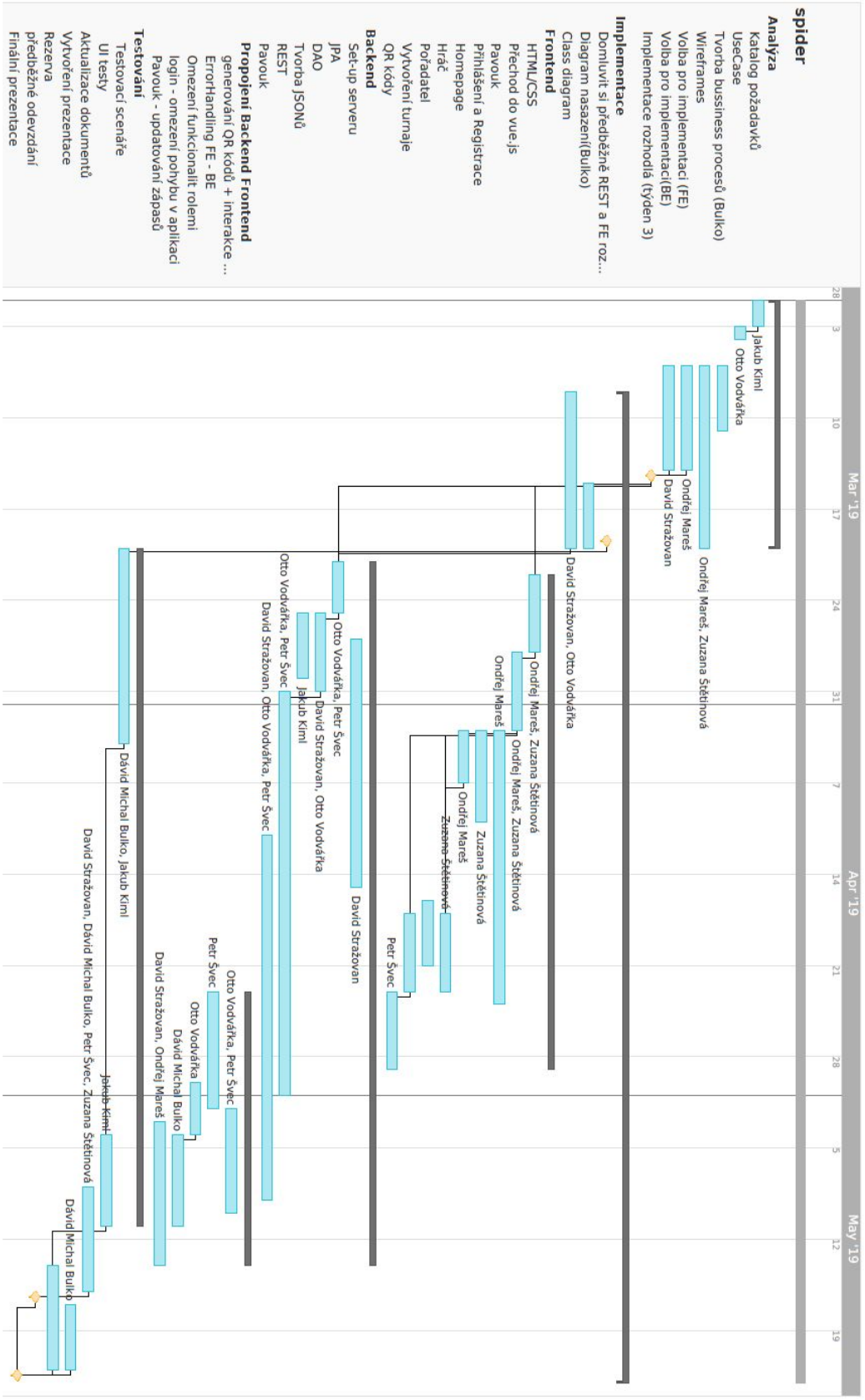
Divák turnaje od systému požaduje, aby…

- ...mu zobrazil zadané výsledky turnajů a postupy jednotlivých hráčů (pavouka)

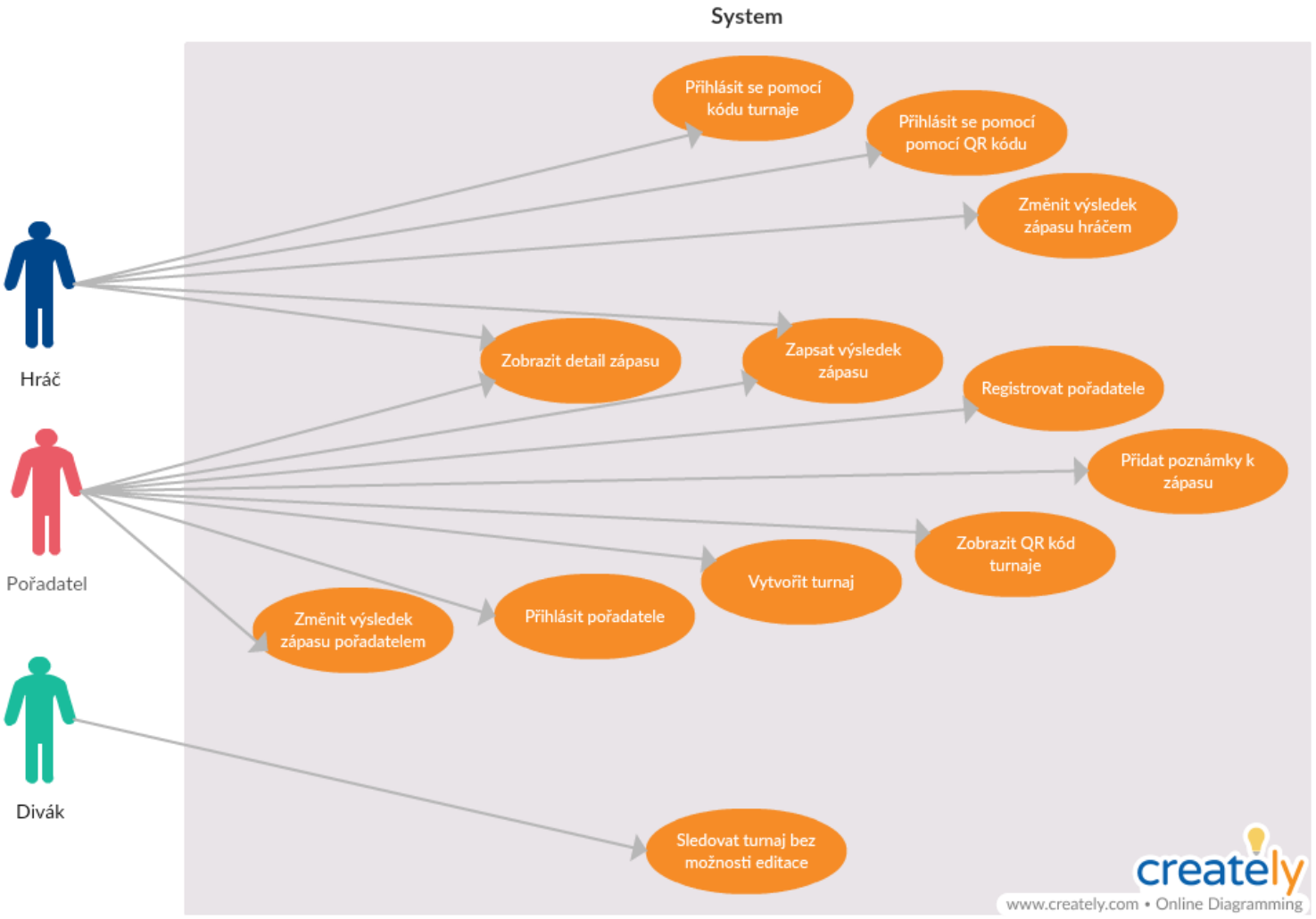
Nefunkční požadavky

- Aplikace musí běžet na současných verzích Chrome, Edge, Safari a Firefox
- Intuitivní ovládání a přehledené GUI
- Možná pozdější rozšiřitelnost

Projektový plán (Gantt)

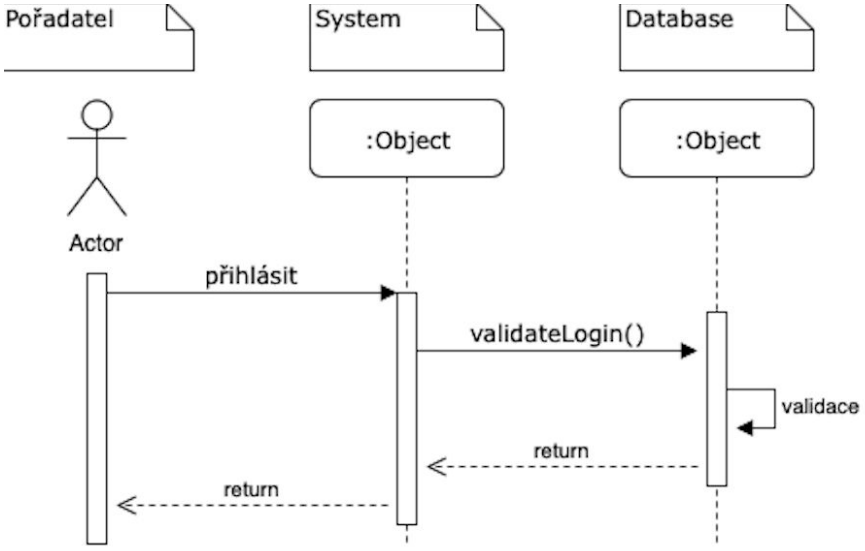


Případy užití

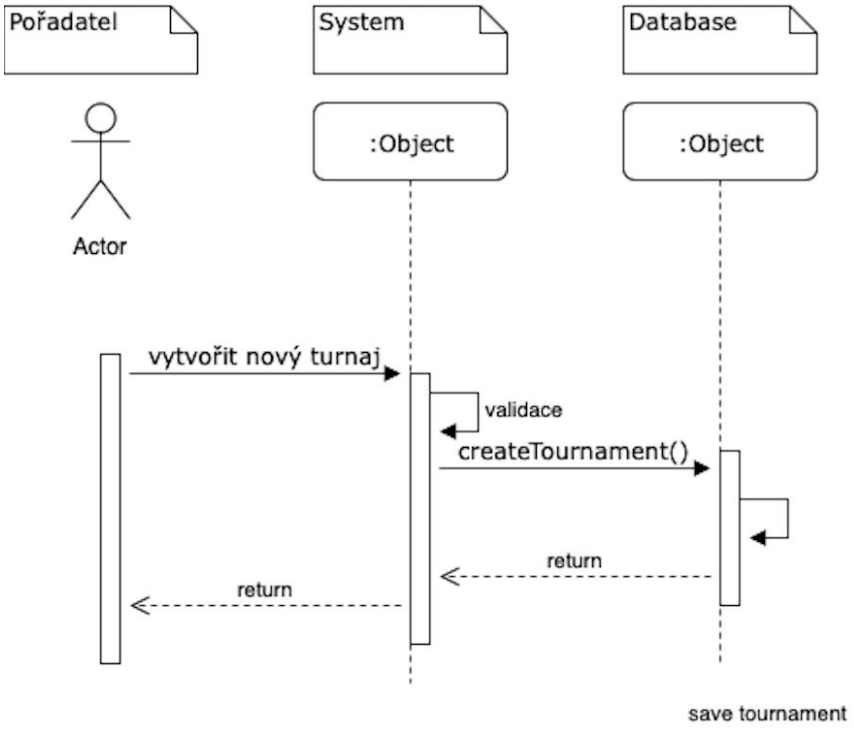


Sekvenční diagramy

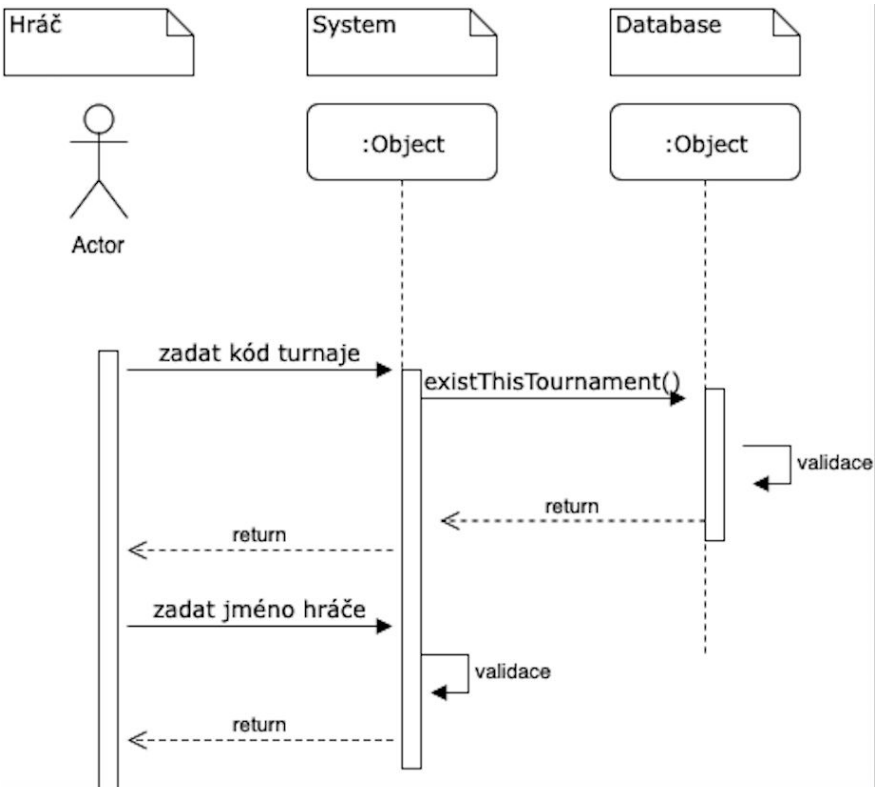
Přihlášení pořadatele



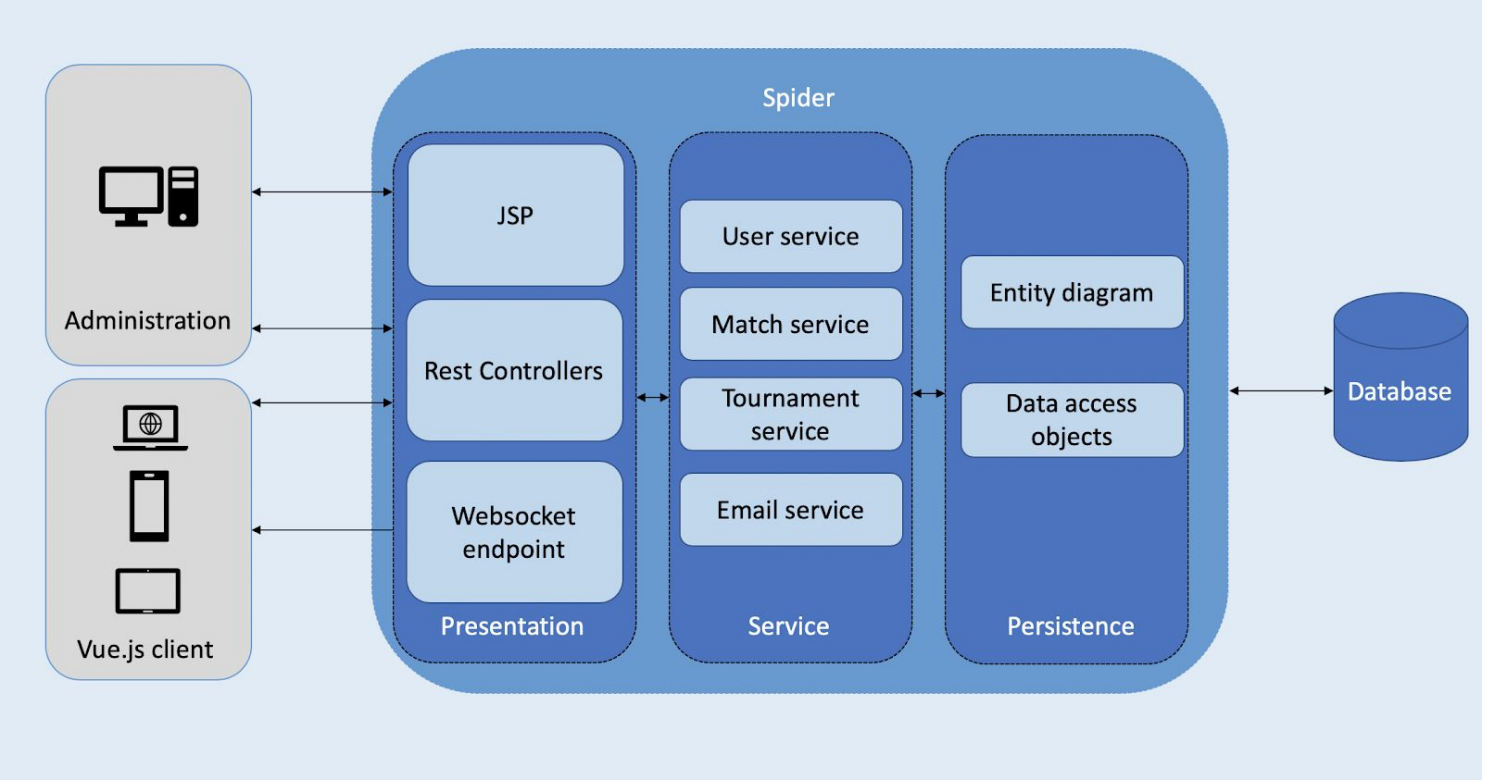
Vytvoření turnaje



Přihlášení hráče



Architektura



Zvolená architektura je klient-server. Komunikace mezi serverem a klientem probíhá dvěma způsoby:

- Request-response
- Posílání zpráv přes websockety

Request-response komunikace je použita pro základní operace ze strany klienta na server. Operacemi jsou například:

- Přihlášení
- Odhlášení
- Získání údajů o turnaji
- Aktualizace výsledků zápasu

Komunikace přes websockety je použita pro “živé” zobrazení turnaje. Při jakékoliv změně v turnaji, na který se uživatel dívá, server pošle klientovi zprávu obsahující změny a ten na základě nich zobrazená data aktualizuje. Tak odpadá potřeba průběžně kontrolovat, jestli se něco změnilo a zlepší se UX.

Aplikace - backend

Pro komunikaci s DB a modelování bylo zvoleno JPA, protože je dostatečné pro náš model a poskytlo nám možnost rychlé implementace.
V systému existují 4 služby (na úrovni kódu) poskytující základní funkcionalitu - Uživatelská, turnajová, zápasová a emailová. Dále existují další podpůrné služby jako např. služba pro vytváření QR kódů.

Velká část aplikační logiky je konfigurovatelná. V konfiguraci je možné přidávat skripty pro vytváření turnajů a zpracování aktualizací výsledků. Tyto se dají po dvojicích seskupovat do tzv. šablon ze kterých může uživatel vybírat při tvorbě turnaje. Toto nám dovoluje doplňovat nové turnajové formáty bez potřeby znovu sestavování aplikace a jejího nasazení. Stejně tak je možné konfigurovat používané emailové účty a šablony odesílaných emailů a další atributy systému.
Pro skriptování je použit jazyk Groovy, skripty jsou za běhu aplikace kompilovány a vyhodnocovány.

Aplikace - frontend

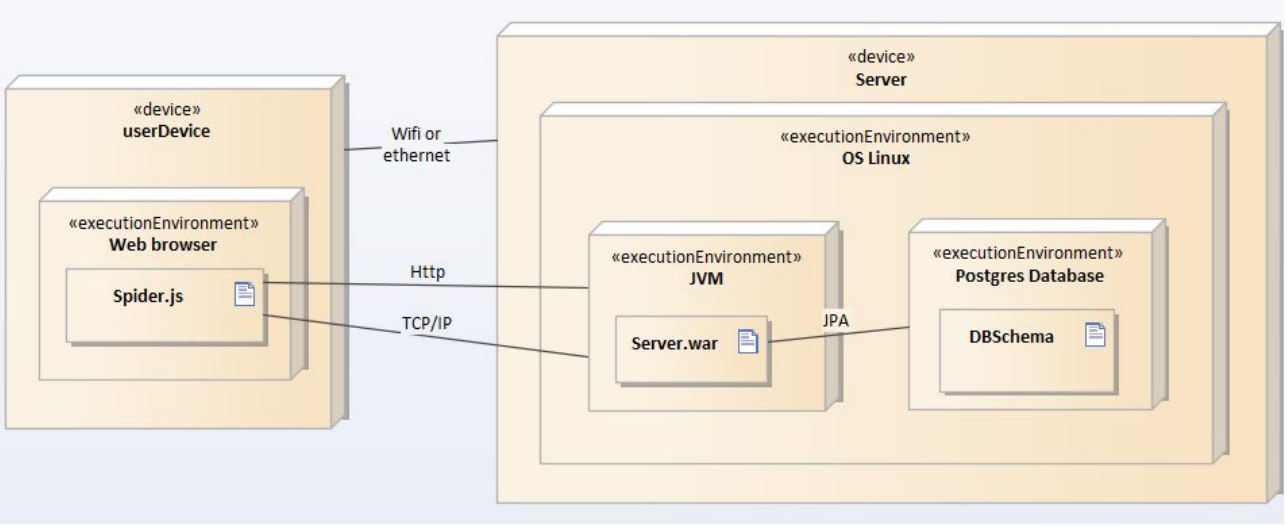
Administrace

Administrační část je implementovaná pomocí JSP a javascriptu a je součástí backendového balíčku. Jedná se o jednoduchou aplikaci, která se skládá převážně ze stránek s tabulkami a detaily pro jednotlivé záznamy.

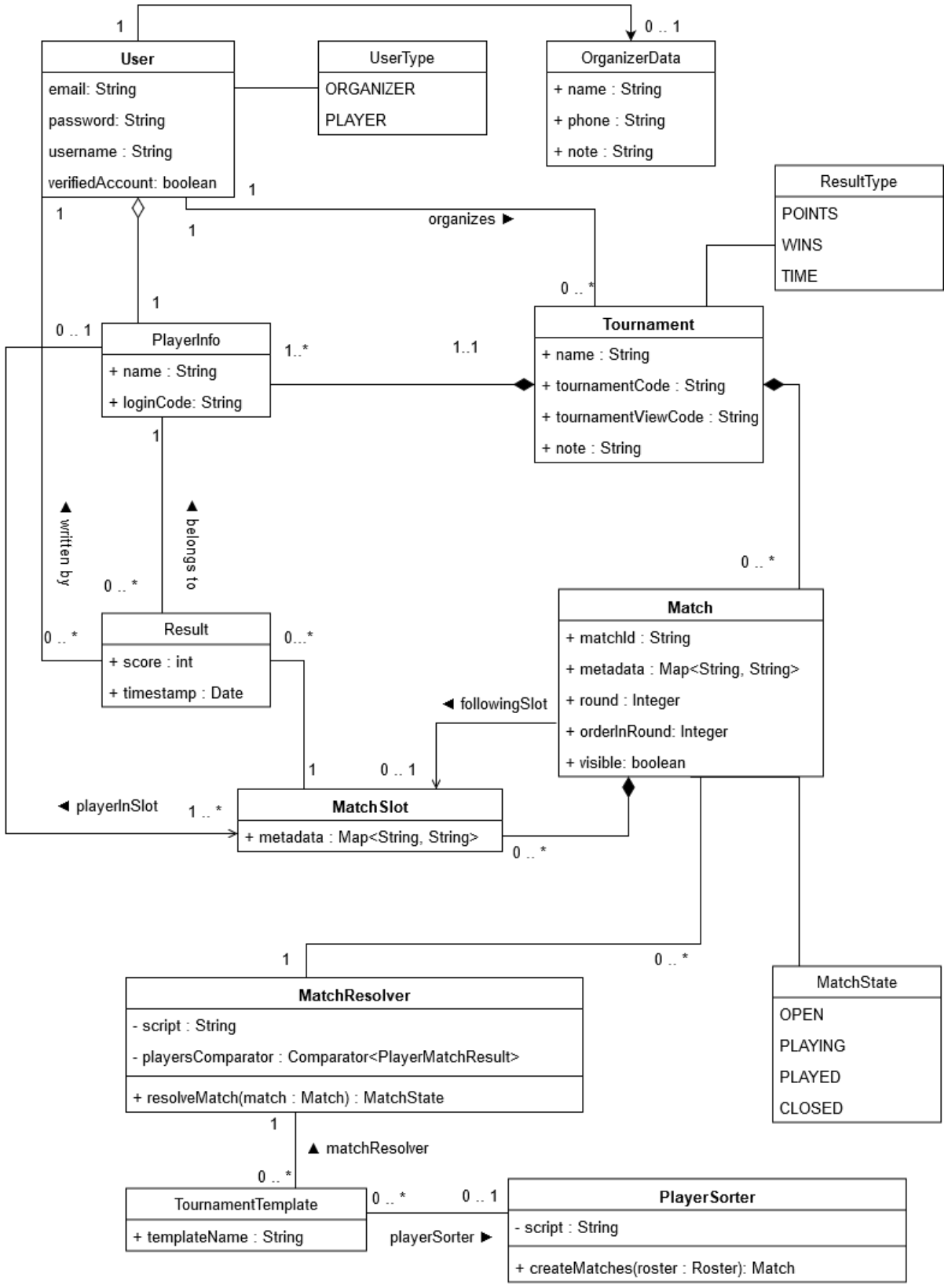
Uživatelský klient

Klientská aplikace je psaná v javascriptu. Pro implementaci klientské aplikace byl framework Vue.js. Ten nám dovoluje tvořit znovupoužitelné komponenty, které můžeme využívat na více stránkách.
Aplikaci je možné nasazovat samostatně a nezávisle na backendové části na http server (např. Apache nebo NGINX).

Diagram nasazení



Class diagram



Zvolená implementace

Backend:

- Spring boot
- REST rozhraní
- Komunikace přes websockety pro real-time promítání změn na frontendu

Fronted:

- Vue frontend

Použité patterny

- DAO / DTO
- State pattern
    - Vytváření zápasů vyhodnocování aktualizace výsledků
        - Algoritmus se liší podle toho, s jakou šablonou je turnaj vytvořen
- IoC
    - Některé třídy implementují interface Monitorable. Při startu aplikace se spustí komponenta, která z kontextu načte všechny beany s tímto interfacem a periodicky nad nimi volá metody interfacu Monitorable a sbírá a aktualizuje hodnoty.
        - “Don’t call us, we call you”
- Object pool
    - cache
- Chain of responsibility
    - CorsRequestFilter - serletový filtr
- Facade
    - QrCodeService - facade pro knihovnu na generování QR kódů

RACI Matice

	Petr Švec	Otto Vodvářka	Ondřej Mareš	David Stražovan	Zuzana Štětinová	Jakub Kiml
Katalog požadavků	C, I	I	I	I	I	R, A
Plánování	R, A	I	I	I	I	C, I
Use Case	C, I	R, A	I	I	I	I
Class Diagram	C, I	R, C	I	R, A	I	I
Analýza	C, I	I	C, I	I	C	R, A
Mock data	C, I	I	C, I	I	I	R, A
Test case	C, I	I	I	I	R, C, I	R, A
Testování	C, I	C, I	C, I	C, I	C, I	R, A
Frontend	I	I	R, A	I	R, A	C, I
Wireframes	I	I	R, C	I	R, A, C	C, I
Grafika	I	I	R, A	I	R, I	C, I
Pavouk	I	I	R, A	I	C, I	C, I
Úprava zápasů	I	I	R	I	R, A	C, I
Backend	R, A	R, A	I	R, A	I	C, I
Rest	R, A	I	C, I	R, C, I	R, I	R, C, I
JPA	R, C	R, A	I	R, C	I	I
DAO	R, C, I	R, A	I	R, C	I	I
Service	R, C, I	R, A	I	R, C	I	I
Databáze	C, I	C, I	I	R, A	I	I
Websockets	R, C, I	R, C, I	I	R, A	I	I

- R - responsible (odpovědná osoba)
- A - accountable (ručitel)
- C - consulted (konzultant)
- I - informed (informován)

Analýza rizik FMEA

SEV - nejhorší možné následky, které mohou vzniknout 1 - 10
OCC - pravděpodobnost výskytu každého režimu selhání 1 - 10
DET - metody detekce poruch 1 - 10
RPN - číslo priority rizika RPN = SEV * OCC * DET

Procesní část výskytu	Závada	Možný dopad na zákazníka	Možná příčina vzniku	SEV	OCC	DET	RPN	Doporučené opatření	Možný dopad na projekt	Odpovědná osoba
Analýza	Špatné porozumění zadání	Dodání něčeho jiného	Špatný sběr požadavků	8	5	3	120	Zaměřit se na sběr požadavků a analýzu systému	Dodání něčeho jiného opoždění vývoje	PM
Všechny	Špatná komunikace v takto velkém týmu	Opoždění dodání	Neústupnost jednotlivých členů týmu	7	4	4	112	Řešit problémy v klidu a naslouchat ostatním	Rozložení týmu	PM
Implementace frontendu	Zákazník nebude spokojen se vzhledem aplikace	Opoždění dodání	Špatné pochopení zadání	3	2	2	12	Časté konzultace se zákazníkem	Nutnost předělání	PM, frontend tým
Implementace	Špatná kompatibilita backendu a frontendu	Posunutí termínů	Špatná komunikace mezi backend a frontend	10	3	2	60	Jasně se domluvit na endpointech	Posunutí termínů	Frontend, backend
Plánování	Nedostatek času na testování	Posunutí termínů	Prodloužení implementace, špatné plánování	5	4	5	100	Kontrola termínů	Bugová aplikace	PM

Použité metriky

Název metriky	Hodnota
Počet případů užití	12
Čas spotřebovaný opravou defektu (průměrně)	0,5 MD
Velikost projektu (měřeno počtem tabulek v DB)	19

Plán odbavení

1. Zakoupení domény a nasazení aplikace na webovou stránku
2. Předání dokumentace a zdrojových kódů zadavateli
3. Seznámení zadavatele se základní funkčností aplikace

První bod je nejdůležitější, jelikož bez něj nemůže být aplikace nikým používána.

Plán podpory

1. Oprava problémů, které neodhalilo akceptační testování.
2. Vypracování nových požadavků, které se postupem času objeví, ale nebyly ve specifikaci.
3. Rozšíření aplikace na jiné druhy turnajů, například každý s každým apod.

První bod je spojen se specifikací projektu. U druhého bodu je však někdy problém určit, zda daná funkcionalita ještě spadá do specifikace. Zákazník má totiž často jinou představu o projektu a předpokládá i věci, které nespecifikoval.

Vyhodnocení

Zadání jsme zpracovali komplexněji než bylo nutné, ale snažili jsme se ho udělat jednoduše rozšiřitelné. Strávili jsme jsme relativně zbytečnou dobu na designu aplikace. Ovšem implementovali jsme všechny požadavky

Oproti plánu jsme měli odchylky hlavně v odhadu propojení frontendu pavouka s backendem. Museli jsme převzít práci zatíženého člena týmu a nechat ho dělat hlavně na tomto problému

Komunikace v týmu místy vázla a tak jsme změnili systém zadávání tasků a začali jsme více komunikovat přímo, například přes messenger. Celkově jsme na projektu strávili přibližně 270 hodin, ale trackování času nebylo úplně přesné, takže to je spíše odhad.

Celkově jsme pozorovali hlavně proměnlivou motivaci členů týmu během projektu. Zjistili jsme na vlastní kůži, jak důležitá je organizovanost a zapálení projektového vedoucího. Taky jsme zjistili, že je důležité se ujistit, zda všichni členové ví, jak se správně užívá použitá technologie jako např. vue components nebo git.